KLASIFIKASI SPESIES BURUNG BERDASARKAN SUARA MENGGUNAKAN *DEEP LEARNING* DENGAN FITUR MEL-SPEKTOGRAM DAN *MEL FREQUENCY CEPTRAL COEFFICIENTS* (MFCC)

Daniswara Aditya Putra[1] dan Kartika Fithriasari[2]
[1,2]Departemen Statistika, Institut Teknologi Sepuluh Nopember
e-mail: [1]daniswara.dap@gmail.com

Abstrak— Burung merupakan indikator penting dalam menilai kesehatan ekosistem dan biodiversitas suatu wilayah. Perkembangan teknologi pengenalan suara berbasis deep learning memungkinkan pemantauan spesies burung dilakukan secara otomatis melalui data audio, tanpa perlu pengamatan langsung yang berisiko mengganggu habitat alami. Pada penelitian ini dilakukan klasifikasi suara enam spesies burung yang umum ditemukan di Indonesia, yaitu Kutilang, Gereja, Perkutut, Tekukur, Trucukan, dan Cendet, menggunakan lima arsitektur model deep learning, yaitu CNN (Mel-Spektrogram), CNN (MFCC), MobileNet (Mel-Spektrogram), MobileNet (MFCC), dan VGGish (waveform). Data yang digunakan merupakan rekaman suara dari situs xeno-canto.org yang telah melalui proses ekstraksi fitur. Model dibangun menggunakan framework TensorFlow dan Keras, kemudian dilatih dengan kombinasi hyperparameter yang bervariasi untuk memperoleh performa terbaik. Berdasarkan hasil evaluasi menggunakan metrik akurasi, precision, recall, dan F1-score pada data test, model CNN dengan fitur MFCC menunjukkan performa terbaik dengan akurasi mencapai 97,97%, precision 97,78%, recall 98,27%, dan F1-score 97,96%. Model ini tidak hanya mampu mengenali pola suara burung secara akurat, tetapi juga menunjukkan kemampuan generalisasi yang sangat baik terhadap data baru. Hasil penelitian ini menegaskan bahwa CNN dengan fitur MFCC merupakan pendekatan yang efektif dan andal untuk tugas klasifikasi suara burung, serta memiliki potensi besar untuk mendukung upaya konservasi berbasis teknologi di Indonesia.

Kata Kunci—Burung, CNN, Mel-Spektogram, MFCC, MobileNet, VGGish

I. PENDAHULUAN

Burung memiliki peran penting sebagai indikator kesehatan lingkungan karena populasinya mencerminkan kondisi udara, air, dan vegetasi [1]. Perubahan pada populasi burung bisa menjadi tanda awal gangguan ekosistem. Mengingat pentingnya keanekaragaman hayati, terutama di Indonesia yang kaya akan spesies burung, pendekatan seperti perekaman suara burung digunakan dalam penelitian karena tidak mengganggu habitat alaminya [2]. Namun, analisis suara ini memerlukan bantuan teknologi seperti deep learning [3].

Indonesia, sebagai negara dengan keanekaragaman burung yang tinggi, menjadi tempat yang relevan untuk penelitian ini. Enam jenis burung umum di pemukiman dipilih karena mudah diakses dan memiliki karakteristik vokal yang beragam, memudahkan pelatihan sistem pengenalan suara [4]. Identifikasi burung melalui suara memungkinkan pelestarian tanpa mengganggu habitat, serta mendukung pemantauan ekosistem secara efisien.

Klasifikasi suara burung dalam penelitian ini dilakukan menggunakan metode deep learning. Dalam penelitian ini menggunakan Convolutional Neural Network (CNN), MobileNet, dan VGGish. Sebelum digunakan sebagai input ke dalam model, data audio diproses terlebih dahulu untuk mengekstraksi fitur representatif yang dapat menggambarkan karakteristik suara secara efektif. Dua jenis fitur yang umum digunakan adalah Mel-spektrogram dan *Mel Frequency Cepstral Coefficients* (MFCC). Mel-spektrogram merepresentasikan distribusi energi suara dalam domain waktu-frekuensi pada skala Mel, sehingga membentuk citra dua dimensi yang memudahkan model deep learning dalam mengenali pola suara. Sedangkan MFCC merupakan representasi kompak dari spektrum suara yang mengekstraksi koefisien penting berdasarkan skala Mel, yang sering digunakan untuk menangkap karakteristik akustik suara secara efisien. Kedua fitur ini kemudian dijadikan input pada model deep learning seperti CNN, MobileNet, dan VGGish untuk proses klasifikasi suara burung [5]. CNN digunakan karena mampu mengenali pola fitur secara otomatis dan efisien dalam proses komputasi. MobileNet dipilih karena memiliki arsitektur yang ringan dan efisien, sehingga cocok untuk diterapkan pada perangkat dengan keterbatasan sumber daya seperti aplikasi mobile atau edge computing. Selain itu, MobileNet tetap mampu memberikan performa klasifikasi yang kompetitif meskipun dengan jumlah parameter yang lebih sedikit. Adapun VGGish merupakan model pre-trained berbasis arsitektur VGG yang telah dilatih khusus untuk data audio, sehingga mampu meningkatkan akurasi klasifikasi suara burung secara signifikan [6].

Berbagai penelitian sebelumnya mendukung penggunaan *Deep Learning* dalam klasifikasi suara burung. Irwandi et al. (2005) mengembangkan sonotaksonomi berdasarkan karakteristik suara. Ali (2020) menggunakan MFCC dan DTW untuk klasifikasi lovebird dengan akurasi 80%. Putra (2019) memadukan MFCC dan CNN untuk klasifikasi beberapa burung lokal, dengan akurasi 94%. Ihsanti dan Al Maki (2024) menggunakan mel-spektogram dan CNN untuk suara burung hantu dengan akurasi tinggi namun precision dan recall rendah. Penelitian ini bertujuan mengatasi keterbatasan terdahulu dan menghasilkan model klasifikasi suara burung yang lebih akurat dan seimbang [7-9].

II. TINJAUAN PUSTAKA

A. Suara dan Ciri Khas Burung

Suara merupakan fenomena fisik dari getaran yang merambat melalui medium dan dapat direpresentasikan dalam bentuk digital berdasarkan frekuensi, amplitudo, dan durasi [10]. Dalam pengolahan sinyal suara, data yang direkam melalui mikrofon diubah menjadi bentuk digital dan diekstraksi menjadi fitur seperti mel-spektogram atau MFCC agar dapat dianalisis oleh model deep learning, seperti CNN, yang mampu mengenali pola visual dari representasi suara tersebut [11]. Pendekatan ini telah banyak digunakan dalam sistem pengenalan suara otomatis karena efisiensinya dalam mendeteksi pola secara akurat. Dalam konteks klasifikasi suara burung, pendekatan ini menjadi relevan mengingat setiap spesies burung memiliki ciri khas vokal yang unik, seperti perbedaan frekuensi dan ritme kicauan, yang dapat diidentifikasi melalui representasi visual dari suara mereka [12].

Berikut merupakan spesies burung yang digunakan dalam penelitian ini.

1. Burung Kutilang (Pycnonotus aurigaster)
   Memiliki suara nyaring dan merdu dengan variasi seperti "cuk-cuk," "cang-kur," dan "ke-ti-lang." Suara ini digunakan untuk komunikasi wilayah, peringatan, dan menarik pasangan, menunjukkan kemampuan adaptasi tinggi.
2. Burung Gereja (*Passer domesticus*)
   Suara meningkat intensitasnya saat musim kawin, berfungsi dalam menarik perhatian betina. Kicauan lebih monoton dibanding spesies lain namun tetap penting dalam konteks reproduksi.
3. Burung Perkutut (*Geopelia striata*)
   Memiliki kicauan lembut dan berirama, dipercaya memberikan ketenangan. Suara jantan lebih kuat dan bervariasi dibanding betina, dengan perbedaan khas antara perkutut lokal dan Bangkok.
4. Burung Tekukur (*Streptopelia chinensis*)
   Dikenal dengan suara khas "te-kuk-kurrrr" yang berulang dan melengkung. Kicauan ini menjadi ciri utama spesies dan dapat dikenali dengan mudah di alam terbuka.
5. Burung Trucukan (*Pycnonotus goiavier*)
   Memiliki suara nyaring dan bervariasi, seperti "cok, cok, cok" disertai siulan cepat. Trucukan jantan lebih vokal dan aktif, sering menampilkan perilaku seperti mengangkat jambul saat berkicau.
6. Burung Cendet (*Lanius cristatus*)
   Memiliki variasi suara yang sangat luas, mulai dari siulan hingga pekikan tajam. Cendet jantan berkicau lebih lama dan sering, digunakan untuk menandai wilayah dan merespons lingkungan secara agresif.

B. Preprocessing Data dan Ekstraksi Fitur (Mel-Spektogram dan MFCC)

Tahap preprocessing merupakan langkah penting dalam pengolahan data audio sebelum dimasukkan ke dalam model klasifikasi. Pada penelitian ini, proses preprocessing mencakup beberapa tahapan utama, yaitu konversi format, normalisasi durasi, deteksi aktivitas suara (VAD), serta ekstraksi fitur mel-spektogram dan MFCC.

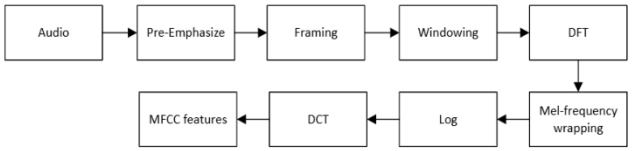
Langkah pertama adalah memastikan semua data audio memiliki format dan durasi yang seragam. File audio dikonversi ke format WAV untuk menjaga konsistensi input. Proses ini juga mencakup normalisasi volume dan penghapusan noise statis.

Salah satu tahapan penting dalam preprocessing adalah Voice Activity Detection (VAD). VAD merupakan teknik yang digunakan untuk mendeteksi keberadaan suara dalam sinyal audio, dengan menghilangkan jeda atau noise berdasarkan analisis energi sinyal. Pendekatan ini meningkatkan efisiensi dan kualitas data karena hanya segmen dengan aktivitas suara yang diproses lebih lanjut. Dalam penelitian ini, VAD diimplementasikan menggunakan fungsi 'librosa.effects.split' yang mendeteksi segmen aktif berdasarkan ambang energi rata-rata lokal [13]. Persamaan yang digunakan dalam deteksi aktivitas suara oleh Librosa dapat dituliskan sebagai

$$E_t = \frac{1}{T}\sum_{h=1}^{T} x(h)^2 \quad (1)$$

dimana E_t adalah energi rata-rata sinyal suara pada frame ke-*t*, T adalah jumlah sampel dalam satu rekaman, dan *x(h)* adalah nilai amplitudo sinyal pada sampel ke-*h*. Jika energi dalam suatu segmen melebihi nilai ambang batas ($E_{threshold}$), segmen tersebut dianggap sebagai bagian yang mengandung suara dan dipertahankan, sedangkan bagian lain yang berada di bawah ambang batas akan dihapus.

Setelah segmen suara burung yang aktif diperoleh melalui proses VAD, data audio kemudian ditransformasikan menjadi bentuk numerik yaitu mel-spektogram dan MFCC. Pada Gambar 1 menunjukkan alur dari data audio menjadi mel-spektogram dan MFCC.

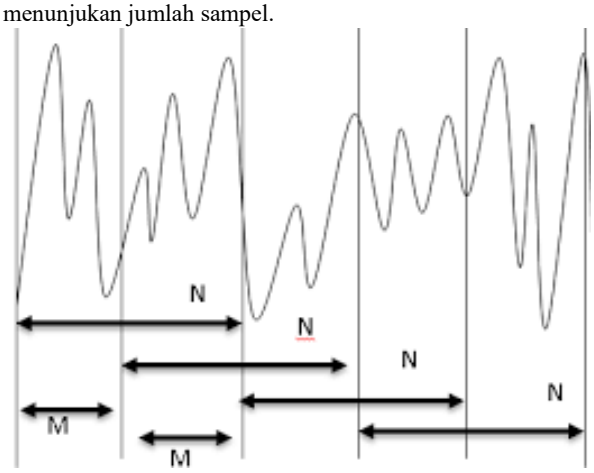


Gambar 1. Alur Pembuatan dari Audio Menjadi Mel-Spektogram dan MFCC

Berikut adalah alur pembuatan mel-spektogram dan MFCC yang digunakan dalam penelitian ini.

1. Pre-emphasis
   Pre-emphasis sendiri bertujuan untuk mengurangi noise ratio pada sinyal serta menyeimbangkan spektrum dari voiced sound. Secara perhitungan tahap ini dapat dirumuskan dalam dalam persamaan sebagai
   $$x_{pre}(t) = x(t) - \alpha x(t-1) \quad (2)$$
   dimana $x_{pre}(t)$ adalah sinyal hasil filter pre-emphasis ke-t, $x(t)$ adalah sinyal sebelum pre-emphasis ke-*t,* α merupakan koefisien pre-emphasis (0,95), $x(t-1)$ merupakan sinyal sebelum pre-emphasis ke *(t-1), t* merupakan indeks waktu yang menunjukkan posisi sampel dalam sinyal audio.

2. Framing
   Sinyal suara yang telah di pre-emphasis akan dilakukan proses framing. Pada langkah ini sinyal akan terbagi menjadi beberapa frame dengan masing-masing frame memuat N sampel sinyal dan frame yang saling berdekatan dipisahkan sejauh M sampel. Panjang frame yang membagi sampel menjadi beberapa frame berdasarkan waktu terletak di antara 20ms sampai 40ms. Pada Gambar 2 menunjukan M adalah panjang frame sedangkan N menunjukan jumlah sampel.



Gambar 2. Proses Framing

3. Windowing
   Proses windowing mempunyai tujuan untuk mengurangi efek diskontinu pada ujung frame yang dihasilkan oleh frame blocking. Fungsi window (w(t)) sendiri ada banyak, contohnya adalah rectangular window dan hamming window tetapi pada penelitian ini menggunakan hamming window yang dapat dituliskan sebagai
   $$w(t) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi t}{T-1}) , & 0 \le t \le T \\ 0 , & lainnya \end{cases} \quad (3)$$
   lalu dari fungsi window tersebut diinputkan terhadap sinyal suara yang dapat dituliskan sebagai
   $$x_{frame}(t) = x_{pre}(t).w(t) \quad (4)$$

dimana $x_{frame}(t)$ adalah sinyal output hasil windowing ke-t, $w(t)$ adalah fungsi window ke-t, T adalah jumlah sampel yang akan diproses.

4. Fast Fourier Transform (FFT)

Fast Fourier Transform adalah pengembangan dari algoritma Discrete Fourier Transform (DFT) yang digunakan untuk mengubah sinyal yang semula time domain menjadi frequency domain. Secara perhitungan tahap ini dapat dirumuskan sebagai

$$X(t,k) = \sum_{t=0}^{T-1} x_{frame}(t) e^{-j\left(\frac{2\pi}{T}\right)tk} \quad ; \quad k = 0,1,2,\dots,T-1 \qquad (5)$$

lalu dilakukan perhitungan magnitudo kuadrat dari spektrum untuk mendapatkan spektrum daya yang dapat dituliskan sebagai

$$SD(t,k) = |X(t,k)|^2 \qquad (6)$$

dimana $X(t,k)$ adalah hasil perhitungan FFT indeks frame ke-t dan frekuensi ke-k, k adalah indeks frekuensi diskrit yang bernilai $(k = \frac{T}{2}, k \in T)$, $SD(t,k)$ adalah spektrum daya indeks frame ke-t dan frekuensi ke-k.

5. Mel-Scale dan Filter Bank

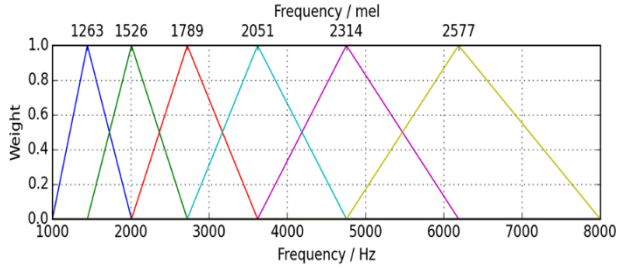
Pada tahap ini dilakukan wrapping terhadap spectrum yang dihasilkan Mel-scale untuk menyesuaikan resolusi frekuensi terhadap properti pendengaran manusia. Kemudian Mel-scale dikelompokan menjadi sejumlah critical ban menggunakan filter bank. Adapun untuk menghitung mel-scale dapat dituliskan sebagai

$$m = 2595 \log(1 + \frac{f}{700}) \qquad (7)$$

Setelah memperoleh nilai mel dari frekuensi minimum dan maksimum, nilai-nilai tersebut kemudian digunakan untuk membagi spektrum menjadi sejumlah critical band melalui filter bank. Filter segitiga ini memiliki tiga titik penting yaitu batas bawah (k_{m-1}), puncak (k_m), dan batas atas (k_{m+1}). Pembentukan masing-masing filter segitiga dilakukan dengan persamaan sebagai

$$H_m(k) = \begin{cases} 0, & k < k_{m-1} \\ \dfrac{k - k_{m-1}}{k_m - k_{m-1}}, & k_{m-1} \le k \le k_m \\ \dfrac{k_{m+1} - k}{k_{m+1} - k_m}, & k_m \le k \le k_{m+1} \\ 0, & k > k_{m+1} \end{cases} \qquad (8)$$

dimana $H_m(k)$ adalah filter segitiga untuk filter ke-m. Dengan demikian, frekuensi yang telah dipetakan ke dalam skala mel ini digunakan untuk mendistribusikan filter segitiga secara merata dalam bank filter mel. Pada Gambar 3 merupakan contoh gambar dari bank filter mel.



Gambar 3. *Filter Mel Bank*

Setelah mendapatkan bank filter Mel dengan menggunakan filter segitiga yang telah ditentukan berdasarkan titik-titik kritis langkah berikutnya adalah menghitung mel spektrogram yang dapat dihitung dengan persamaan sebagai

$$S(t,m) = \sum_{k=1}^{K} H_m(k).SD(t,k) \qquad (9)$$

dimana $S(t,m)$ adalah mel spektogram indeks frame ke-t dan filter ke-m, m adalah indeks koefesien mel filter bank, dan M adalah jumlah channel dalam filter bank.

6. Discrete Consine Transform (DCT)

DCT adalah langkah terakhir pada proses ekstrasi ciri MFCC. Konsep dasar dari DCT adalah mendekorelasi mel spectrum sehingga menghasilkan representasi yang baik dari properti spektral vokal [14]. Adapun DCT dapat dihitung dengan persamaan sebagai

$$C(a,t) = \sum_{m=1}^{M} (\log_{10} S_{mel}(t,m) \cos[a\left(m - \frac{1}{2}\right)\frac{\pi}{M}]) \qquad (10)$$

dimana $C(a,t)$ adalah koefisien MFCC ke-a dan frame ke-t dan a adalah indeks koefisien MFCC (a =1,2,…,A)

C. Model Deep Learning

Deep learning adalah cabang dari machine learning yang menggunakan jaringan saraf tiruan bertingkat (deep neural networks) untuk mempelajari representasi data secara otomatis [15]. Model deep learning terdiri dari banyak layer tersembunyi yang mampu mengekstrak fitur kompleks dari data yang tidak terstruktur, seperti gambar, teks, atau audio [16]. Dalam konteks klasifikasi suara, deep learning sangat efektif karena mampu mengenali pola visual pada representasi suara (misalnya mel-spektogram atau MFCC) tanpa perlu fitur yang ditentukan secara manual [17]. Pada penelitian ini menggunakan tiga arsitektur model yang mewakili karakteristik dan keunggulan masing-masing, yaitu Convolutional Neural Network (CNN) sebagai model dasar, MobileNet sebagai model ringan dan efisien, serta VGGish sebagai model pre-trained yang telah dioptimalkan khusus untuk data audio.

1. CNN

CNN merupakan jenis deep neural network yang dirancang untuk mengenali pola spasial dalam data dua dimensi, seperti gambar atau spektogram suara [18]. Arsitektur CNN terdiri dari beberapa layer utama, yaitu convolutional layer, activation layer (ReLU), pooling layer, dan fully connected layer. Perhitungan pada convolutional layer dilakukan dengan menerapkan filter (kernel) pada input untuk menghasilkan feature map yang dapat dituliskan sebagai

$$c_{p^*,q^*}^{(layer)} = f_{ReLU}\left(B^{(layer)} + \sum_{i=1}^{I}\sum_{j=1}^{J} W_{i,j}^{(layer)} X_{p^*+i-1,q^*+j-1}^{(layer-1)}\right) \qquad (11)$$

Pada proses konvolusi dalam jaringan saraf konvolusional (CNN), output dari convolutional layer pada posisi baris ke-p^* dan kolom ke- q^* dilambangkan dengan $c_{p^*,q^*}^{(layer)}$. Nilai ini diperoleh dari hasil operasi konvolusi antara bobot kernel $W_{i,j}^{(layer)}$ dengan input dari layer sebelumnya, yaitu $X_{p^*+i-1,q^*+j-1}^{(layer-1)}$, yang merupakan representasi mel-spektogram atau MFCC. Kernel konvolusi memiliki ukuran panjang I dan lebar J, yang menunjukkan dimensi filter yang digunakan [19]. Setelah hasil konvolusi dijumlahkan dan ditambahkan dengan bias $B^{(layer)}$ atau simpangan pada feature map, nilai tersebut kemudian dilewatkan melalui fungsi aktivasi ReLU yang dilambangkan sebagai f_{ReLU} [20]. Fungsi ReLU dapat dituliskan sebagai

$$f_{ReLU}(Q) = \max(0, Q) \qquad (12)$$

Kemudian, dilakukan *max-pooling* untuk mengurangi dimensi yang perhitungannya dapat dituliskan sebagai

$$d_{p^*,q^*}^{(layer+1)} = \max_{i,j}\{c_{p^*+i-1,q^*+j-1}^{(layer)}\} \quad (13)$$

dimana $d_{p^*,q^*}^{(layer+1)}$ adalah output pooling layer dan $c_{p^*+i-1,q^*+j-1}^{(layer)}$ adalah input dari layer sebelumnya. Setelah itu dilakukan perhitungan layer fully connected, tetapi output tersebut terlebih dahulu akan dipipihkan (flatten) menjadi vektor satu dimensi. Perhitungan layer fully connected dapat dituliskan sebagai

$$\rho_r = \sum_{s=1}^{S} W_{r,s} g_s + B_r \quad (14)$$

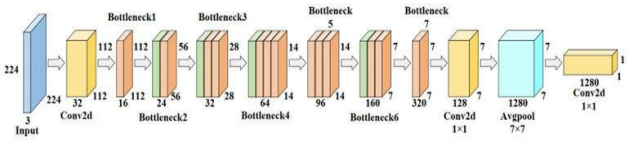
dimana $W_{r,s}$ merupakan bobot (weight) yang menghubungkan input ke-s dengan neuron ke-r, sementara g_s adalah nilai input ke-s dari layer sebelumnya. Nilai bias atau simpangan untuk neuron ke-r ditunjukkan dengan B_r. Indeks s merepresentasikan posisi input dari layer sebelumnya, dan S menunjukkan jumlah total neuron pada layer sebelumnya yang terhubung dengan neuron ke-r di layer fully connected. Layer ini berfungsi untuk menggabungkan dan menginterpretasi fitur yang telah diekstraksi oleh layer-layer sebelumnya (misalnya convolutional layer), sehingga memungkinkan model untuk melakukan klasifikasi berdasarkan kombinasi fitur yang telah dipelajari [21]. Lalu output dari fully connected layer akan dihitung menggunakan fungsi softmax untuk mengetahui hasil klasifikasi yang dapat dihitung dengan persamaan sebagai

$$f_{softmax}(h) = \frac{e^{\rho_h}}{\sum_{r=1}^{R} e^{\rho_r}} \quad (15)$$

dimana $f_{softmax}(h)$ merupakan probabilitas output untuk kelas ke-h [22].

2. MobileNet

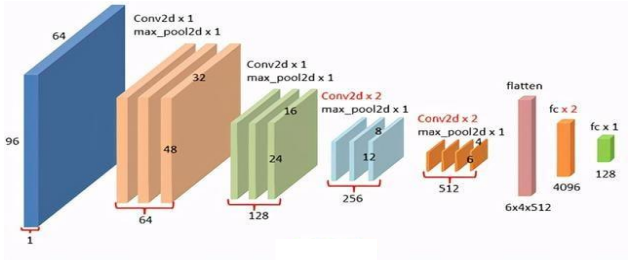
MobileNet adalah arsitektur CNN yang dikembangkan oleh Google pada tahun 2017 dengan tujuan menciptakan model yang ringan dan efisien untuk perangkat mobile dan embedded. Keunggulannya terletak pada penggunaan depthwise separable convolution, yang secara signifikan mengurangi jumlah parameter dan beban komputasi dibandingkan CNN konvensional. Dalam penelitian ini, MobileNet digunakan dengan pendekatan transfer learning, di mana bobot hasil pelatihan awal pada dataset besar seperti ImageNet dimanfaatkan sebagai feature extractor, lalu dilatih ulang pada lapisan akhir sesuai jumlah kelas suara burung. Pendekatan ini mempercepat pelatihan, mengurangi kebutuhan data, dan tetap menghasilkan akurasi tinggi, sehingga sangat cocok untuk tugas klasifikasi berbasis mel-spektogram dan MFCC [23]. Untuk arsitektur MobileNet dapat dilihat pada Gambar 4.



Gambar 4. Arsitektur Model MobileNet

3. VGGish

VGGish adalah model deep learning adaptasi dari arsitektur VGG yang dikembangkan oleh Google pada tahun 2017 untuk ekstraksi fitur audio. Model ini dilatih pada dataset AudioSet yang besar dan beragam, sehingga mampu menghasilkan representasi audio yang kuat dan general. Input berupa waveform audio diubah menjadi mel-spektogram berdurasi sekitar 0,96 detik, lalu diproses oleh jaringan konvolusi dengan struktur VGG, menghasilkan embedding berdimensi 128 yang merepresentasikan fitur audio tingkat tinggi. Embedding ini kemudian dapat digunakan sebagai input untuk model klasifikasi atau deteksi lanjutan. VGGish mendukung pendekatan transfer learning, meningkatkan efisiensi pelatihan dan akurasi meskipun dengan dataset yang lebih kecil [24]. Model ini telah banyak digunakan dalam aplikasi audio seperti pengenalan suara burung, musik, dan lingkungan. Arsitektur model VGGish dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur Model VGGish

D. Evaluasi Model

Evaluasi model merupakan langkah penting dalam mengukur sejauh mana model klasifikasi mampu bekerja secara efektif, tidak hanya pada data pelatihan, tetapi juga pada data baru yang belum pernah dilihat sebelumnya. Dalam penelitian ini, evaluasi performa dilakukan menggunakan beberapa metrik utama, yaitu *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*, yang memberikan gambaran kuantitatif tentang kekuatan dan kelemahan model.

Confusion matrix digunakan untuk membandingkan hasil prediksi model dengan label sebenarnya, menghasilkan empat kategori evaluasi yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) [25]. *Confusion matrix* untuk *multiclass classification* diilustrasikan pada Gambar 6.

	Prediksi Kelas 1	**Prediksi Kelas 2**	...	**Prediksi Kelas N**
Aktual Kelas 1	$C_{1,1}$	$C_{1,2}$	...	$C_{1,N}$
Aktual Kelas 2	$C_{2,1}$	$C_{2,2}$	...	$C_{2,N}$
⋮	⋮	⋮	⋱	⋮
Aktual Kelas N	$C_{N,1}$	$C_{N,2}$	...	$C_{N,N}$

Gambar 6. Confusion Matrix

Keterangan:

- Ci,i adalah jumlah data yang benar diklasifikasikan untuk kelas i (True Positive untuk kelas i).
- Ci,j dengan $i \neq j$ adalah jumlah data dari kelas i yang salah diklasifikasikan sebagai kelas j (False Negative untuk kelas i, dan False Positive untuk kelas j).
- True Negative untuk kelas i yaitu jumlah data yang bukan kelas i dan tidak diprediksi sebagai kelas i.

Berdasarkan nilai-nilai dari confusion matrix, dapat dihitung beberapa metrik evaluasi untuk setiap kelas ke-i:

1. *Precision*

Precision mengukur proporsi prediksi positif yang benar terhadap seluruh prediksi positif yang dibuat model, dirumuskan sebagai

$$Precision_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i} = \frac{C_{i,i}}{\sum_{j=1}^{N} C_{j,i}} \quad (16)$$

2. *Recall*

Recall mengukur proporsi data positif yang berhasil diidentifikasi oleh model dari seluruh data positif yang sebenarnya, dirumuskan sebagai

$$Recall_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} = \frac{C_{i,i}}{\sum_{j=1}^{N} C_{i,j}} \quad (17)$$

3. *F1-score*

F1-score merupakan rata-rata harmonis dari precision dan recall, dan digunakan saat dibutuhkan keseimbangan antara keduanya, dirumuskan sebagai

$$F1 - score_i = \frac{2(Precision_i \times Recall_i)}{Precision_i + Recall_i} \quad (18)$$

4. *Accuracy*

Accuracy menunjukkan proporsi total prediksi yang benar dibandingkan seluruh data yang diuji, dirumuskan sebagai

$$Accuracy = \frac{\mathrm{TP}_i + \mathrm{TN}_i}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{TN}_i + \mathrm{FN}_i} = \frac{\sum_{i=1}^{N} C_{i,i}}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}} \quad (19)$$

Meskipun *accuracy* merupakan metrik yang umum digunakan, dalam kasus dataset yang tidak seimbang, *precision*, *recall*, dan *F1-score* memberikan gambaran performa yang lebih komprehensif [26]. Oleh karena itu, semua metrik ini digunakan bersamaan untuk mengevaluasi model CNN, MobileNet, dan VGGish dalam mengklasifikasikan suara burung secara optimal.

III. METODOLOGI

A. Dataset

Penelitian ini menggunakan data sekunder berupa rekaman suara burung yang diperoleh dari situs https://xeno-canto.org. Dataset terdiri dari 60 rekaman audio dari 6 spesies burung, masing-masing sebanyak 10 rekaman. Durasi rekaman bervariasi antara 11 hingga 115 detik. Spesies burung yang dipilih antara lain yaitu Burung Kutilang, Burung Gereja, Burung Perkutut, Burung Tekukur, Burung Trucukan, dan Burung Cendet. Pemilihan keenam spesies ini didasarkan pada keberadaan mereka yang umum di Indonesia serta karakteristik vokal yang khas, sehingga sesuai untuk tugas klasifikasi berbasis suara.

Setiap rekaman dianotasi berdasarkan label kelas (spesies burung), dan kemudian diolah menjadi format input fitur melalui proses ekstraksi mel-spektogram dan MFCC. Struktur data suara yang akan diproses disajikan pada Tabel 1.

Tabel 1
Struktur Data Suara (Waveform)

Data	X_1	X_2	…	X_T	Y
1	$X_{1,1}$	$X_{1,2}$	…	$X_{1,T}$	Y_1
2	$X_{2,1}$	$X_{2,2}$	…	$X_{2,T}$	Y_2
⋮	⋮	⋮	⋱	⋮	⋮
N	$X_{N,1}$	$X_{N,2}$	…	$X_{N,T}$	Y_{sp}

Sementara untuk struktur data fitur mel-spketogram (128 koefisien per frame) disajikan pada Tabel 2.

Tabel 2
Struktur Data Mel-Spektogram

Data	*Frame*	S_1	S_2	…	S_{128}	Y
1	1	$S_{1,1,1}$	$S_{1,1,2}$	…	$S_{1,1,128}$	Y_1
	2	$S_{1,2,1}$	$S_{1,2,2}$	…	$S_{1,2,128}$	
	⋮	⋮	⋮	⋱	⋮	
	T_1	$S_{1,T1,1}$	$S_{1,T1,2}$	…	$S_{1,T1,128}$	
2	1	$S_{2,1,1}$	$S_{2,1,2}$	…	$S_{2,1,128}$	Y_1
	2	$S_{2,2,1}$	$S_{2,2,2}$	…	$S_{2,2,128}$	
	⋮	⋮	⋮	⋱	⋮	
	T_2	$S_{2,T2,1}$	$S_{2,T2,2}$	…	$S_{2,T2,128}$	
⋮	⋮	⋮	⋮	⋱	⋮	⋮
N	1	$S_{N,1,1}$	$S_{N,1,2}$	…	$S_{N,1,128}$	Y_{sp}
	2	$S_{N,2,1}$	$S_{N,2,2}$	…	$S_{N,2,128}$	
	⋮	⋮	⋮	⋱	⋮	
	T_N	$S_{N,TN,1}$	$S_{N,TN,2}$	…	$S_{N,TN,128}$	

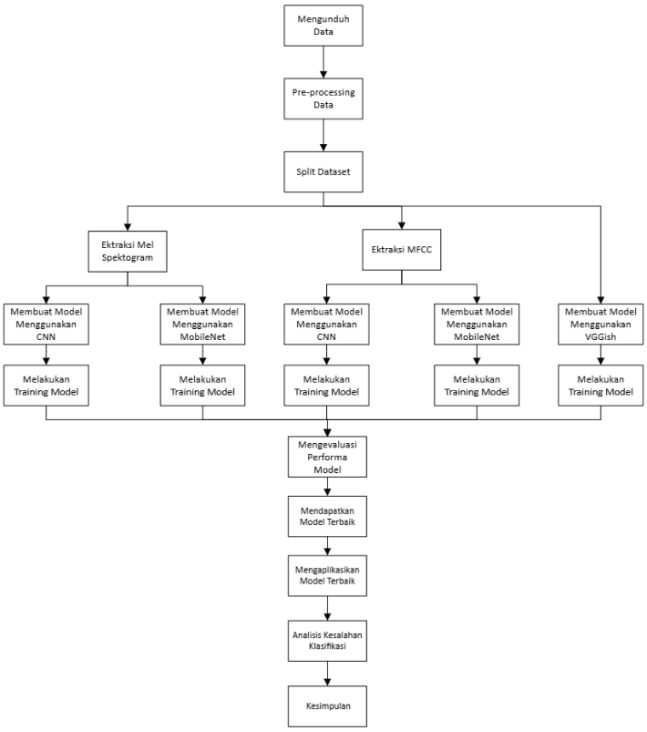
Kemudian untuk struktur data MFCC (40 koefisien per frame) disajikan pada Tabel 3.

Tabel 3
Struktur Data MFCC

Data	*Frame*	C_1	C_2	…	C_{40}	Y
1	1	$C_{1,1,1}$	$C_{1,1,2}$	…	$C_{1,1,40}$	Y_1
	2	$C_{1,2,1}$	$C_{1,2,2}$	…	$C_{1,2,40}$	
	⋮	⋮	⋮	⋱	⋮	
	T_1	$C_{1,T1,1}$	$C_{1,T1,2}$	…	$C_{1,T1,40}$	
2	1	$C_{2,1,1}$	$C_{2,1,2}$	…	$C_{2,1,40}$	Y_1
	2	$C_{2,2,1}$	$C_{2,2,2}$	…	$C_{2,2,40}$	
	⋮	⋮	⋮	⋱	⋮	
	T_2	$C_{2,T2,1}$	$C_{2,T2,2}$	…	$C_{2,T2,40}$	
⋮	⋮	⋮	⋮	⋱	⋮	⋮
N	1	$C_{N,1,1}$	$C_{N,1,2}$	…	$C_{N,1,40}$	Y_{sp}
	2	$C_{N,2,1}$	$C_{N,2,2}$	…	$C_{N,2,40}$	
	⋮	⋮	⋮	⋱	⋮	
	T_N	$C_{N,TN,1}$	$C_{N,TN,2}$	…	$C_{N,TN,40}$	

B. Langkah Penelitian

Langkah penelitian ini dimulai dengan pengumpulan data rekaman suara burung dari situs *xeno-canto.org*, diikuti dengan preprocessing yang mencakup trimming menggunakan Voice Activity Detection (VAD), serta ekstraksi fitur mel-spektogram dan MFCC. Data kemudian diubah ke dalam format dan dimensi yang sesuai untuk input model, termasuk proses normalisasi. Dataset dibagi menjadi data training dan validation dengan proporsi 80:20. Selanjutnya, dilakukan pelatihan dan evaluasi model klasifikasi menggunakan CNN, MobileNet, dan VGGish. Evaluasi performa dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score, serta dibandingkan untuk menentukan model terbaik.



Gambar 7. Diagram Alir Penelitian

IV. HASIL DAN PEMBAHASAN

A. Preprocessing dan Ekstraksi Fitur

Tahap awal dalam penelitian ini adalah preprocessing rekaman suara burung. Setiap rekaman dikonversi ke format .wav dan dipotong menggunakan metode Voice Activity Detection (VAD) dengan ambang energi -45 dB untuk menghilangkan bagian yang tidak mengandung suara burung. Hasil trimming menghasilkan segmen pendek berdurasi 1–3 detik yang hanya berisi suara aktif. Proses ini

tidak hanya meningkatkan jumlah data, tetapi juga kualitas representasi untuk setiap kelas spesies burung, sebagaimana terlihat pada Tabel 4.

Tabel 4
Jumlah Rekaman Suara Burung Sebelum dan Setelah Trimming

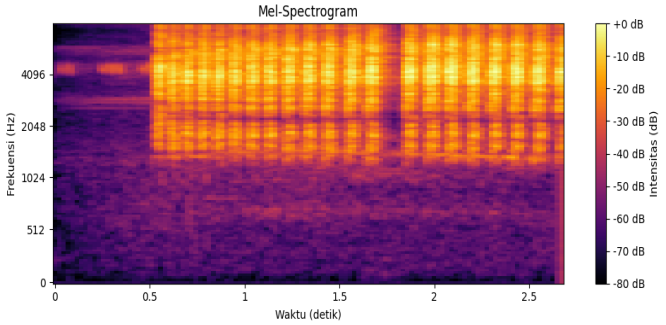
Spesies	Rekaman Asli	Rekaman Hasil Trimming
Kutilang	10	97
Gereja	10	63
Perkutut	10	150
Tekukur	10	78
Trucukan	10	76
Cendet	10	95

Dataset kemudian dibagi menjadi data training validation dan testing dengan proporsi 80 % (80% training dan 20% validation) : 20 % secara stratifikasi yang disajikan pada Tabel 5.

Tabel 5
Hasil Pembagian Data Training dan Testing

Jenis Burung	Data			Total
	Training	Validation	Testing	
Kutilang	62	16	19	97
Gereja	40	10	13	63
Perkutut	96	24	30	150
Tekukur	50	12	16	78
Trucukan	49	12	15	76
Cendet	61	15	19	95
Total	358	89	112	559

Setelah itu, dilakukan ekstraksi fitur audio berupa mel-spektogram dan Mel Frequency Cepstral Coefficients (MFCC) menggunakan library Librosa. Mel-spektogram diperoleh dengan membagi sinyal menjadi frame pendek, menerapkan STFT, dan memetakan hasilnya ke skala mel menggunakan 128 filter. Visualisasi hasil ekstraksi menunjukkan pola energi frekuensi yang khas untuk tiap spesies burung yang dapat dilihat pada Gambar 14 yang merupakan salah satu contoh ilustrasi mel-spektogram dan salah satu perhitungan perhitungan mel-spektogram menggunakan library Librosa dengan ditunjukkan pada Tabel 6.

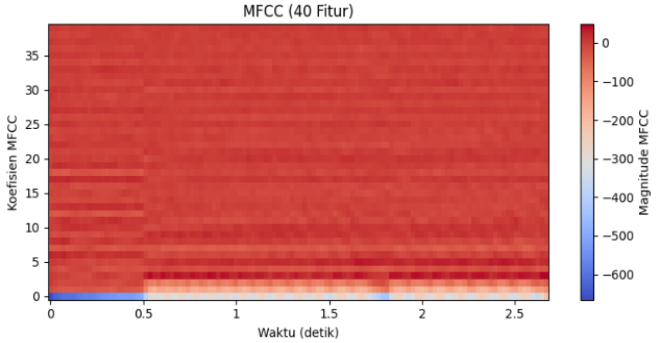


Gambar 8. Visualisasi Mel-Spektogram dari Audio Segmen 1 Cendet2.wav

Tabel 6
Hasil Mel-Spektogram untuk Audio Segmen 1 Cendet2.wav

Frame ke-	Mel ke-1	Mel ke-2	…	Mel ke-128
1	-69,199	-75,341	…	-72,787
2	-68,113	-78,818	…	-69,254
3	-68,803	-77,263	…	-69,282
⋮	⋮	⋮	⋱	⋮
116	-43,600	-43,777	…	-20,992

Sementara itu, MFCC diperoleh dari log energi mel-spektrogram yang kemudian ditransformasi dengan Discrete Cosine Transform (DCT), menghasilkan 40 koefisien per frame. Untuk visulisasi salah satu contoh MFCC disajikan pada Gambar 15 dan salah satu perhitungan perhitungan MFCC menggunakan library Librosa dengan ditunjukkan pada Tabel 7.



Gambar 9. Visualisasi MFCC dari Audio Segmen 1 Cendet2.wav

Tabel 7
Hasil MFCC untuk Audio Segmen 1 Cendet2.wav

Frame ke-	MFCC ke-1	MFCC ke-2	…	MFCC ke-40
1	-666,193	-15,322	…	6,118
2	-630,269	-20,513	…	6,198
3	-619,503	-29,231	…	4,175
⋮	⋮	⋮	⋱	⋮
116	-215,160	-112,700	…	0,149

Kedua fitur ini mampu menangkap karakteristik vokal burung secara rinci dan menjadi input utama dalam model klasifikasi CNN dan MobileNet.

B. Pemodelan dan Evaluasi Klasifikasi dengan CNN

Model CNN digunakan sebagai baseline dengan input berupa fitur mel-spektogram dan MFCC yang disusun dalam bentuk matriks dua dimensi. Arsitektur terdiri dari tiga lapisan Conv2D yang diikuti MaxPooling2D dan Batch Normalization, lalu Flatten dan Dense sebagai klasifikasi akhir. Fungsi aktivasi softmax digunakan pada output untuk memetakan prediksi ke enam kelas burung. Detail arsitektur lengkap model CNN ini dapat dilihat pada Tabel 8.

Tabel 8
Arsitektur Model CNN

Layer Type	Output Shape	Parameter
input (input layer)	(None, 224, 224, 3)	0
conv2d_1 (Conv2D)	(None, 222, 222, 32)	320
max_pooling2d_1 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_2 (Conv2D)	(None, 109, 109, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_3 (Conv2D)	(None, 52, 52, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten_1 (Flatten)	(None, 86528)	0
dense_1 (Dense)	(None, 64)	5537856
dense_2 (Dense)	(None, 6)	390
Total Parameter		**16892756**

Setelah memuat detail arsitektur CNN, dilakukan proses pelatihan model menggunakan dataset suara burung dengan fitur mel-spektogram dan MFCC sebagai input. Model dilatih menggunakan data training dan divalidasi untuk memantau akurasi dan loss, serta mencegah overfitting melalui dropout dan stratifikasi data. Setiap model kombinasi dilatih selama maksimal 150 epoch dengan penerapan mekanisme *callbacks* untuk mengoptimalkan proses pelatihan dan mencegah *overfitting*. Setiap kombinasi model dilatih sebanyak 30 kali secara independen, sehingga distribusi hasil evaluasi mendekati normal dan estimasi rata-rata performa lebih stabil.

1. Klasifikasi Model CNN dengan Mel-Spektogram

Model CNN dilatih menggunakan fitur mel-spektogram dari suara burung, dengan pengujian beberapa kombinasi hyperparameter untuk memperoleh konfigurasi terbaik. Hyperpameter kombinasi model ditampilkan pada Tabel 9 dan hasil pelatihan ditampilkan pada Tabel 10. Pada setiap kombinasi, digunakan nilai learning rate yang sama sebesar 0,00001 dan dropout sebesar 0,5.

Tabel 9
Hyperparemeter Kombinasi Model CNN (Mel-Spektogram)

No.	Hyperparameter
	Batch
1.	16
2.	32

Tabel 10
Hasil Kombinasi Model CNN (Mel-Spektogram)

No.	Hasil			
	Mean* Akurasi *Training	***Mean Loss Training***	***Mean* Akurasi *Validation***	***Mean Loss Validation***
1.	0,99	0,002632	0,9134	0,453622
2.	1,00	0,002472	0,9190	0,365826

Kombinasi kedua (learning rate 0.000001, batch size 16, dropout 0.5) memberikan hasil dengan akurasi training 100% dan akurasi validasi 91,90%, serta loss validasi yang lebih rendah dibandingkan kombinasi pertama (learning rate 0.000001, batch size 32, dropout 0.5).

Perbedaan performa pada data validasi menunjukkan bahwa pemilihan batch size berpengaruh terhadap generalisasi model. Kombinasi pertama dipilih sebagai konfigurasi optimal karena menjaga keseimbangan antara akurasi tinggi dan overfitting yang rendah.

2. Klasifikasi Model CNN dengan MFCC

Setelah pelatihan model CNN menggunakan fitur mel-spektogram, pelatihan serupa dilakukan dengan fitur MFCC untuk mengevaluasi efektivitas alternatif representasi audio. Proses pelatihan menggunakan arsitektur CNN yang sama, dengan beberapa kombinasi hyperparameter untuk menentukan konfigurasi terbaik. Hyperpameter kombinasi model ditampilkan pada Tabel 11 dan hasil pelatihan ditampilkan pada Tabel 12. Pada setiap kombinasi, digunakan nilai learning rate yang sama sebesar 0,00001 dan dropout sebesar 0,5.

Tabel 11
Hyperparemeter Kombinasi Model CNN (MFCC)

No.	Hyperparameter
	Batch
1.	16
2.	32

Tabel 12
Hasil Kombinasi Model CNN (MFCC)

No.	Hasil			
	Mean* Akurasi *Training	***Mean* Loss *Training***	***Mean* Akurasi *Validation***	***Mean* Loss *Validation***
1.	1,00	0,000368	0,9797	0,110764
2.	1,00	0,000789	0,9762	0,109140

Kombinasi pertama (batch size 16) menghasilkan akurasi training 100% dan validasi 97,97%, dengan loss validasi 0,11. Kombinasi kedua, dengan batch size 32, menghasilkan akurasi validasi lebih rendah (97,62%) dan loss validasi lebih (0,109). Hasil ini menunjukkan bahwa konfigurasi pertama lebih optimal, terutama dalam hal generalisasi. Performa model dengan input MFCC cukup sebanding dengan mel-spektogram, namun tetap menunjukkan bahwa pemilihan batch size berpengaruh terhadap akurasi validasi.

C. Pemodelan dan Evaluasi Klasifikasi dengan MobileNet

Model klasifikasi kedua menggunakan pendekatan transfer learning dengan arsitektur MobileNet, yang menerima input berupa fitur mel-spektogram dan MFCC dari sinyal audio suara burung. Fitur diubah ukurannya agar sesuai dengan input MobileNet (224×224×3), kemudian dimasukkan ke model yang telah dilatih sebelumnya pada dataset besar (ImageNet). Hanya lapisan akhir yang dilatih ulang agar sesuai dengan klasifikasi enam spesies burung.

MobileNet dipilih karena efisien secara komputasi dan cocok untuk pemrosesan pada perangkat terbatas. Total parameter dalam model ini sebanyak 4.253.312, dengan sebagian besar berasal dari lapisan konvolusi terpisah (depthwise separable convolution) yang mempercepat proses pelatihan. Arsitektur lengkap ditampilkan pada Tabel 13.

Tabel 13
Arsitektur Model MobileNet

Layer (Type)	Output Shape	Parameter
InputLayer	(None, 224, 224, 3)	0
Conv2D (Conv1)	(None, 112, 112, 32)	864
BatchNormalization	(None, 112, 112, 32)	128
ReLU	(None, 112, 112, 32)	0
DepthwiseConv2D (dw_1)	(None, 112, 112, 32)	288
BatchNormalization	(None, 112, 112, 32)	128
ReLU	(None, 112, 112, 32)	0
Conv2D (pw_1)	(None, 112, 112, 64)	2048
BatchNormalization	(None, 112, 112, 64)	256
ReLU	(None, 112, 112, 64)	0
⋮	⋮	⋮
Conv2D (pw_8)	(None, 7, 7, 1280)	1310720
BatchNormalization	(None, 7, 7, 1280)	5120
ReLU	(None, 7, 7, 1280)	0
GlobalAveragePooling2D	(None, 1280)	0
Dense (Softmax, 6 classes)	(None, 6)	7686
Total Parameter		**4253894**

Model kemudian dilatih dengan data suara burung, serupa dengan model CNN sebelumnya. Selama pelatihan, akurasi dan loss dimonitor pada data training dan validasi untuk memastikan generalisasi model tetap terjaga dan tidak overfitting. Setiap model kombinasi dilatih selama maksimal 150 epoch dengan penerapan mekanisme *callbacks* untuk mengoptimalkan proses pelatihan dan mencegah *overfitting*. Setiap kombinasi model dilatih sebanyak 30 kali secara independen, sehingga distribusi hasil evaluasi mendekati normal dan estimasi rata-rata performa lebih stabil. Hasil pelatihan dibandingkan dengan CNN untuk melihat keunggulan penggunaan transfer learning.

1. Klasifikasi Model MobileNet dengan Mel-Spektogram

Model MobileNet dilatih menggunakan fitur mel-spektogram dari suara burung, dengan pengujian beberapa kombinasi hyperparameter untuk memperoleh konfigurasi terbaik. Hyperpameter kombinasi model MobileNet dengan fitur mel-spektogram ditampilkan pada Tabel 14 dan hasil pelatihan ditampilkan pada Tabel 15. Pada setiap kombinasi, digunakan nilai learning rate yang sama sebesar 0,00001 dan dropout sebesar 0,5.

Tabel 14
Hyperparemeter Kombinasi Model MobileNet (Mel-Spektogram)

No.	*Freeze*	Hyperparameter
		Batch
1.	Ya	16
2.	Ya	32
3.	Tidak	16
4.	Tidak	32

Tabel 15
Hasil Kombinasi Model MobileNet (Mel-Spektogram)

No.	Hasil			
	Mean* Akurasi *Training	***Mean Loss Training***	***Mean* Akurasi *Validation***	***Mean Loss Validation***
1.	0,99	0,030630	0,8958	0,339251
2.	0,99	0,031818	0,8988	0,335059
3.	1,00	0,000039	0,9681	0,142865
4.	1,00	0,000013	0,9675	0,148537

Dari Tabel 15 terlihat bahwa model tanpa freeze layer memberikan performa terbaik. Kombinasi ketiga (tanpa freeze, batch size 16, dropout 0,5, dan learning rate 0,000001) menghasilkan akurasi validasi 96,81% dan loss 0,1428, lebih tinggi dibanding konfigurasi lain. Sebaliknya, konfigurasi dengan freeze layer menghasilkan akurasi validasi lebih rendah (89,88%) dan loss lebih tinggi. Hasil ini menunjukkan pentingnya membuka semua layer MobileNet saat fine-tuning agar model dapat belajar pola baru dari fitur suara burung secara optimal.

2. Klasifikasi Model MobileNet dengan MFCC

Model MobileNet juga dilatih menggunakan fitur MFCC untuk membandingkan efektivitasnya dengan Mel-spektogram. Hyperpameter kombinasi model MobileNet dengan fitur MFCC ditampilkan pada Tabel 16 dan hasil pelatihan ditampilkan pada Tabel 17. Pada setiap kombinasi, digunakan nilai learning rate yang sama sebesar 0,00001 dan dropout sebesar 0,5.

Tabel 16
Hyperparemeter Kombinasi Model MobileNet (MFCC)

No.	*Freeze*	Hyperparameter
		Batch
1.	Ya	16
2.	Ya	32
3.	Tidak	16
4.	Tidak	32

Tabel 17
Hasil Kombinasi Model MobileNet (MFCC)

No.	Hasil			
	Mean* Akurasi *Training	***Mean* Loss *Training***	***Mean* Akurasi *Validation***	***Mean* Loss *Validation***
1.	1,00	0,007361	0,9366	0,193864
2.	1,00	0,006201	0,9300	0,202851
3.	1,00	0,000063	0,9646	0,171753
4.	1,00	0,000015	0,9631	0,175985

Dari Tabel 17, konfigurasi terbaik diperoleh pada kombinasi tanpa freeze layer, dengan batch size 16, dan learning rate 0,000001, menghasilkan akurasi validasi 96,46% dan loss 0,171. Sebaliknya, konfigurasi dengan freeze layer menunjukkan penurunan akurasi validasi hingga 93%, serta peningkatan loss. Hal ini menunjukkan bahwa membiarkan seluruh layer MobileNet dilatih ulang lebih efektif dalam mempelajari fitur MFCC.

D. Pemodelan dan Evaluasi Klasifikasi dengan VGGish

Model VGGish dilatih menggunakan input waveform audio mentah, dengan arsitektur yang telah dimodifikasi pada bagian output untuk menyesuaikan enam kelas spesies burung. Pada Tabel 18 menunjukkan arsitektur dari model VGGish.

Tabel 18
Arsitektur Model VGGish

Layer Type	Output Shape	Parameter
input (input layer)	(None, 96, 64, 1)	0
conv1 (Conv2D)	(None, 96, 64, 64)	640
MaxPooling2D-1	(None, 48, 32, 64)	0
conv2 (Conv2D)	(None, 48, 32, 128)	73856
MaxPooling2D-2	(None, 24, 16, 128)	0
conv3_1 (Conv2D)	(None, 24, 16, 256)	295168
conv3_2 (Conv2D)	(None, 24, 16, 256)	590080
MaxPooling2D-3	(None, 12, 8, 256)	0
conv4_1 (Conv2D)	(None, 12, 8, 512)	1180160
conv4_2 (Conv2D)	(None, 12, 8, 512)	2359808
MaxPooling2D-4	(None, 6, 4, 512)	0
fc1_1 (Dense)	(None, 12288)	50331648
fc1_2 (Dense)	(None, 4096)	16777216
fc2 (Dense)	(None, 128)	524416
dense_4 (Dense)	(None, 64)	8256
dense_5 (Dense)	(None, 6)	390
Total Parameter		**72141184**

Pendekatan transfer learning ini memanfaatkan bobot pre-trained dari AudioSet dan melakukan fine-tuning pada lapisan klasifikasi akhir. Model kemudian dilatih dengan data suara burung, berbeda dengan model-model sebelumnya, pada model VGGish inputnya merupakan data suara mentah (waveform). Setiap model kombinasi dilatih selama maksimal 150 epoch dengan penerapan mekanisme *callbacks* untuk mengoptimalkan proses pelatihan dan mencegah *overfitting*. Setiap kombinasi model dilatih sebanyak 30 kali secara independen, sehingga distribusi hasil evaluasi mendekati normal dan estimasi rata-rata performa lebih stabil. Selama pelatihan, akurasi dan loss dimonitor pada data training dan validasi untuk memastikan generalisasi model tetap terjaga dan tidak overfitting. Dilakukan pengujian dengan beberapa kombinasi hyperparameter untuk memperoleh konfigurasi terbaik. Hyperpameter kombinasi model VGGish dengan input waveform ditampilkan pada Tabel 19 dan hasil pelatihan ditampilkan pada Tabel 20. Pada setiap kombinasi, digunakan nilai learning rate yang sama sebesar 0,00001 dan dropout sebesar 0,5.

Tabel 19
Hyperparemeter Kombinasi Model VGGish (Waveform)

No.	Hyperparameter
	Batch
1.	16
2.	32

Tabel 20
Hasil Kombinasi Model VGGish (Waveform)

No.	Hasil			
	Mean* Akurasi *Training	***Mean Loss Training***	***Mean* Akurasi *Validation***	***Mean Loss Validation***
1.	1,00	0,009292	0,9485	0,174700
2.	1,00	0,008162	0,9467	0,169665

Dari Tabel 20, kombinasi terbaik diperoleh pada batch size 16 dan dropout 0,5, dengan akurasi validasi 94,67% dan loss 0,169. Sementara kombinasi batch size 32 menunjukkan penurunan akurasi validasi dan peningkatan loss. Hasil ini menegaskan bahwa batch size kecil lebih optimal untuk data audio mentah karena memberikan pembaruan bobot yang lebih presisi.

E. Pemilihan Model Terbaik

Penelitian ini mengevaluasi tiga model klasifikasi utama, yaitu CNN, MobileNet, dan VGGish, menggunakan input berbeda yaitu Mel-spektogram, MFCC, dan waveform. CNN dan MobileNet memanfaatkan fitur hasil ekstraksi, sedangkan VGGish langsung memproses waveform mentah melalui arsitektur pre-trained dari AudioSet. Tujuan utamanya adalah menentukan model dengan akurasi dan generalisasi terbaik untuk klasifikasi enam spesies burung berdasarkan suara. Hasil evaluasi performa terbaik dari masing-masing model ditampilkan pada Tabel 21 dan hasil evaluasi pada test ditampilkan pada Tabel 22.

Tabel 21
Hasil Performa Terbaik Masing-Masing Model

Model	*Mean Train Accuracy*	*Mean Train Loss*	*Mean Validation Accuracy*	*Mean Validation Loss*
CNN (Mel-Spektogram)	1,00	0,002472	0,9190	0,365826
CNN (MFCC)	1,00	0,000368	0,9797	0,110764
MobileNet (Mel-Spektogram)	1,00	0,000039	0,9681	0,142865
MobileNet (MFCC)	1,00	0,000063	0,9646	0,171753
VGGish (Waveform)	1,00	0,009292	0,9485	0,174700

Tabel 22
Hasil Evaluasi Setaip Model pada Data Test

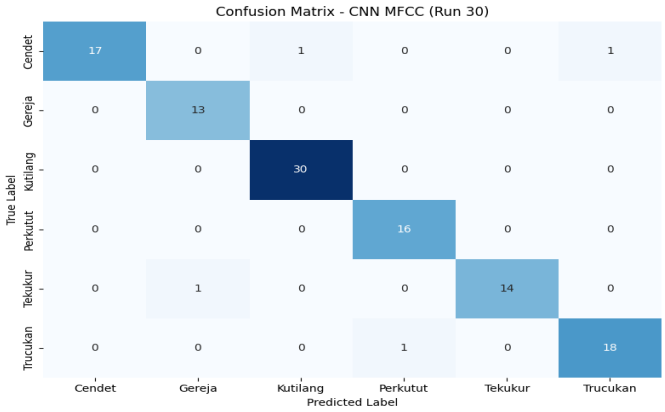
Model	*Test Accuracy*	*Test Recall*	*Test Precision*	*Test F1-Score*
CNN (Mel-Spektogram)	0,91161	0,917915	0,90800	0,910245
CNN (MFCC)	0,97976	0,982743	0,97782	0,979612
MobileNet (Mel-Spektogram)	0,96934	0,971222	0,96598	0,967385
MobileNet (MFCC)	0,96607	0,968407	0,96376	0,964899
VGGish (Waveform)	0,95029	0,952212	0,94742	0,948519

Berdasarkan Tabel 21 dan Tabel 22, model CNN dengan fitur MFCC dipilih sebagai model terbaik karena menghasilkan akurasi validasi tertinggi (97,97%) dan nilai loss terendah (0,1107), serta akurasi test yang konsisten tinggi (97,97%). Keberhasilan ini didukung oleh efisiensi arsitektur CNN dan representasi spektral dari MFCC, yang efektif dalam menangkap ciri khas suara burung. Dengan performa tinggi pada seluruh metrik evaluasi dan jumlah data yang terbatas, model ini dianggap paling optimal dalam penelitian klasifikasi suara burung yang dilakukan.

F. Implementasi Model Terbaik pada Data Test

Model terbaik yang di peroleh dari penelitian ini, yaitu model CNN dengan fitur MFCC, diimplementasikan pada data test yang terdiri dari 112 sampel suara dari enam jenis burung lokal Indonesia. Tujuan dari pengujian ini adalah mengevaluasi kemampuan model dalam mengklasifikasikan suara burung pada data yang benar-benar baru dan tidak dilibatkan dalam proses pelatihan maupun validasi.

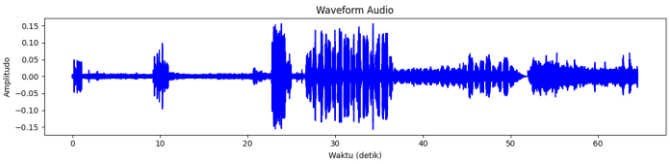
Setelah data diuji diproses ke bentuk Mel-Spektogram, model melakukan prediksi untuk masing-masing sampel. Hasil klasifikasi divisualisasikan dalam confusion matrix yang ditampilkan pada Gambar 10 yang menunjukkan performa tinggi dengan akurasi keseluruhan mencapai 97,97%.



Gambar 10. Confusion Matrix Prediksi Model Terbaik pada Data Test

Model menunjukkan klasifikasi sempurna pada beberapa jenis burung seperti Kutilang, Perkutut, dan Trucukan, meskipun terdapat sedikit kesalahan pada kategori Cendet, Kutilang, dan Gereja.

Untuk menguji performa lebih lanjut dalam skenario nyata, model juga diterapkan pada rekaman suara berdurasi panjang (53 detik) yang gelombang audionya ditampilkan pada Gambar. Rekaman ini dipotong menjadi segmen pendek menggunakan metode Voice Activity Detection (VAD), kemudian diklasifikasikan menggunakan model yang telah dilatih. Hasil deteksi frekuensi kemunculan burung ditampilkan pada Tabel 23.



Gambar 11 Gelombang Audio dari Suara yang akan Diprediksi

Tabel 23
Hasil Prediksi Berdasarkan Jumlah Deteksi masing-masing Jenis Burung

No.	Jenis Burung	Jumlah Terdeteksi
1.	Trucukan	7
2.	Tekukur	9
3.	Gereja	5
4.	Kutilang	2

Hasil ini menunjukkan bahwa model mampu mengidentifikasi jenis burung dan menghitung frekuensi kemunculannya secara akurat, yang sangat relevan untuk aplikasi pemantauan biodiversitas. Pendekatan ini memungkinkan pengamatan populasi burung secara non-invasif dan efisien, serta membuka peluang besar untuk pengembangan sistem monitoring otomatis berbasis suara di masa depan.

V. KESIMPULAN

Penelitian ini menggunakan lima model deep learning—CNN (Mel-Spektrogram), CNN (MFCC), MobileNet (Mel-Spektrogram), MobileNet (MFCC), dan VGGish (waveform)—untuk mengklasifikasikan suara enam spesies burung umum di Indonesia. Data rekaman suara diperoleh dari situs xeno-canto.org dan diproses melalui ekstraksi fitur sesuai kebutuhan masing-masing model. Model dilatih dan dioptimasi menggunakan TensorFlow dan Keras dengan tuning hyperparameter untuk mencapai performa terbaik. Evaluasi dilakukan menggunakan metrik akurasi dan loss pada data training dan validasi, serta akurasi, precision, recall, dan F1-score pada data test. Hasil evaluasi menunjukkan bahwa model CNN dengan fitur MFCC

memberikan performa terbaik, dengan akurasi validasi sebesar 97,97% dan loss validasi 0,1108, serta akurasi test 97,97%. Model ini juga mencapai precision 97,78%, recall 98,27%, dan F1-score 97,96% pada data test, mengungguli model lain seperti CNN (Mel-Spektrogram), MobileNet (Mel-Spektrogram dan MFCC), dan VGGish. Tingginya metrik evaluasi pada data test menegaskan kemampuan generalisasi model CNN MFCC yang kuat terhadap data baru. Dengan performa konsisten dan akurasi tinggi pada seluruh tahap evaluasi, CNN dengan MFCC dipilih sebagai model terbaik untuk pengembangan sistem klasifikasi suara burung yang akurat dan andal.

DAFTAR PUSTAKA

[1] Soraya, "Klasifikasi Genus Burung Hantu Berdasarkan Suara Menggunakan Convolutional Neural Network," *Journal eproc,* 2024.

[2] A. Y. Rahman, "Klasifikasi Citra Burung Jalak Menggunakan Artificial Neural Network dan Random Forest," *Jurnal Edukasi dan penelitian Informatika,* 2022.

[3] G. A. Hatma, Klasifikasi Suara Jenis Burung Menggunakan Deep Learning Berbasis Compressive Sensing, Bandung: Universitas Telkom, 2024.

[4] A. Afida, "Klasifikasi jenis burung berdasarkan suara menggunakan algoritme Support Vector Machine," *MATHunesa Jurnal Ilmiah Matematika 8(1),* pp. 1-6, 2020.

[5] R. Ponnusamy, "A Review of Image Classification Approaches and Techniques," *International Journal of Recent Trends in Engineering and Research,* pp. 3(3), 1-5, 2017.

[6] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie dan L. Farhan, "Review of deep learning: concepts, CNN architechtures, challenges, applications, future directions," *Journal of Big Data,* p. 8(1), 2021.

[7] Irwandi, Marwan, A. Hadi Mahmud dan Abdullah, "Upaya Pemanfaatan Rekaman Suara Burung dan Analisis Spektrogram untuk Menyusun Metode Klasifikasi Berdasarkan Suara (Sonotaksonomi)," *Biosfera,* p. 18–24, 2005.

[8] M. F. Ali, Klasifikasi Jenis Burung Berdasarkan Suara Burung Berbasis Deep Learning, Surabaya: Institut Teknologi Sepuluh Nopember, 2020.

[9] B. P. A. Putra, Klasifikasi suara untuk menentukan kualitas burung lovebird menggunakan metode Mel Frequency Cepstral Coefficients dan Dynamic Time Warping, Yogyakarta: Universitas Pembangunan Nasional Veteran Yogyakarta, 2019.

[10] T. Mitchell, Machine Learning and Sound Processing, Cambridge University Press, 2019.

[11] F. Imaroh, "Cara Kerja Deep Learning dalam Pengenalan Suara," 2024. [Online]. Available: https://blog.advan.id/15138/cara-kerja-deep-learning-dalam-pengenalan-suara/.

[12] H. Wijaya, "Teknologi Pengenalan Suara tentang Metode, Bahasa dan Tantangan," *Bit-Tech, 7(2),* p. 534–540, 2024.

[13] J. Ramírez, J. C. Segura, C. Benítez, L. García dan A. Rubio, "Voice Activity Detection with Noise Reduction and Long-Term Spectral Divergence Estimation for Robust Speech Recognition," *IEEE Transactions on Speech and Audio Processing, 12(3),* p. 265–275, 2004.

[14] F. Patmadi, Klasifikasi Emosi pada Suara dengan Ektraksi Ciri Mel Fequency Ceptral Coefficients Menggunakan Metode 1D Convolutional Neural Network, Jakarta: UIN Syarif Hidayatullah, 2022.

[15] A. Buono, Deep Learning: Dasar Teori dan Penerapannya, Jakarta: Penerbit Informatika, 2020.

[16] J. van deer Laak, G. Litjens dan F. Ciompi, "Deep learning in histopathology: the path to the clinic," *Nat Med 27,* p. 775–784, 2021.

[17] R. Baldock, H. Mennel dan B. Nesyhabur, "Deep Learning Through the Lens of Example Difficulty," in *35th Conference on Neural Information Processing Systems*, 2021.

[18] Y. LeCun, L. Bottou, Y. Bengio dan P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* pp. 86(11), 2278-2324, 1998.

[19] I. Goodfellow, Y. Bengio dan A. Courville, Deep Learning, MIT Press, 2016.

[20] X. Glorot, A. Bordes dan Y. Bengio, "Deep Sparse Rectifier Neural Networks," *In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011),* pp. pp. 315-323, 2011.

[21] B. Md Anwar Hossain, M. Shahriar Alam Sajib, M. Anwar Hossain dan M. Shahriar Alam Sajib, Classification of Image using Convolutional Neural Network (CNN), 2019.

[22] Y. Ho dan S. Wookey, The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling., 2020.

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto dan H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.048614,* 2017.

[24] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss dan K. Wilson, "CNN architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[25] N. Ali, D. Neagu dan P. Trundle, "Evaluation of k-nearest neighbour classifier," *SN Applied Sciences, 1(12).,* p. 1(12), 2019.

[26] Confusion matrix in machine learning, "Retrieved from https://www.geeksforgeeks.org/confusion-matrix-machine-learning/ (Diakses: 2024-10-16)," 2021.