

Klasifikasi Genus Burung Hantu Berdasarkan Suara Menggunakan *Convolutional Neural Network*

1st Soraya Ihsanti
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

sorayaihsanti@student.telkomuniversity.ac.id

2nd Wikky Fawwaz Al Maki
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

wikkyfawwaz@telkomuniversity.ac.id

Abstrak — Burung memiliki peran penting dalam siklus kehidupan yang kebanyakan aktif di siang hari. Burung hantu merupakan salah satu jenis burung yang aktif di malam hari. Burung hantu berperan penting di bidang pertanian dalam mengusir hama. Meskipun penelitian tentang burung hantu sudah banyak, namun banyak aspek biologi, sejarah evolusi, dan taksonomi burung hantu yang tetap kurang diketahui. Penelitian ini dilakukan untuk mengenali suara burung hantu dan menggunakan *Convolutional Neural Network* (CNN) untuk mengklasifikasikannya berdasarkan genusnya. *Trimming* dan *noise reduction* dilakukan untuk membantu menghasilkan sistem dengan performansi yang optimal. Pengujian sistem dilakukan dengan skenario mencari nilai *epoch*, *learning rate*, dan *optimizer* dengan akurasi terbaik. Evaluasi performansi dilakukan dengan *hyperparameter* terbaik yang sudah didapat. Hasil dari pengujian mendapat akurasi sebesar 99.8%.

Kata kunci— *audio processing*, CNN, *sound classification*

I. PENDAHULUAN

Burung merupakan makhluk hidup yang memiliki banyak peran penting di kehidupan ini. Kebanyakan jenis burung merupakan hewan yang aktif di siang hari. Namun beberapa jenis burung juga terbiasa aktif di malam hari. Contohnya adalah burung hantu. Burung hantu termasuk hewan nokturnal dan mencari mangsanya pada malam hari. Namun, meskipun banyak penelitian, banyak aspek biologi, sejarah evolusi, dan taksonomi burung hantu tetap kurang diketahui [1]. Keadaan pengetahuan yang tidak lengkap digaris bawahi oleh peningkatan pesat jumlah spesies burung hantu yang diakui, dari 146 spesies pada tahun 1975 [2] menjadi 250 spesies pada tahun 2008 [3]. Berdasarkan data dari *International Union for Conservation of Nature* (IUCN) *Red List* [4], diketahui bahwa burung hantu (*Strigidae*) rata-rata memiliki status *Least Concern* yang artinya tidak terancam punah. Namun, beberapa jenis burung hantu termasuk hewan yang dilindungi.

Banyak penelitian yang mengklasifikasikan berbagai jenis burung menggunakan suara seperti yang dilakukan oleh Mangalam Sankupellay and Dmitry Konovalov yang menggunakan metode CNN dengan arsitektur *ResNet50* [5], namun belum mendapatkan hasil akurasi yang cukup besar,

yaitu sebesar 60%-72%. Hasil ini dikarenakan durasi inputan spektrogram yang digunakan belum cukup pendek. Ada pula penelitian yang dilakukan oleh Jimmy Ludeña-Choez, Raisa Quispe-Soncco, Ascensión Gallardo-Antolín yang menggunakan metode *Support Vector Machine* (SVM) [6]. Hasil yang didapat menggunakan MFCC feature adalah 69,63%.

Dalam penelitian *machine learning*, terdapat banyak metode yang digunakan untuk klasifikasi dan pengenalan suara. Salah satunya adalah *Convolutional Neural Network* (CNN). CNN sendiri merupakan metode yang paling banyak digunakan karena sangat efektif untuk tugas-tugas yang berkaitan dengan *computer vision*, *image recognition*, *object detection*, *image classification* dan tugas *machine learning* lainnya. Salah satu penelitiannya adalah klasifikasi audio skala besar [7]. Data yang digunakan merupakan dataset yang terdiri dari 100 juta video YouTube berdurasi rata-rata 4,6 menit tiap video dengan total 5,4 juta jam pelatihan. Arsitektur yang digunakan adalah *fully connected*, AlexNet, VGG, Inception V3, dan ResNet50. Namun karena ResNet50 mencapai nilai *mean Average Precision* (mAP) sebesar 21%, maka eksperimen selanjutnya menggunakan arsitektur ini. Banyak keuntungan dalam menggunakan metode CNN seperti pemrosesan gambar, video dan sinyal audio yang efisien, dapat menangani data dalam jumlah besar dan tingkat akurasi yang tinggi, tahan terhadap *noise*, serta *feature extraction* otomatis [8].

Penulisan riset ini didasarkan pada burung hantu yang jarang terlihat oleh manusia dapat dikenali lebih dalam menggunakan suaranya sebagai *input*. Penelitian ini akan mengklasifikasikan genus burung hantu menggunakan metode CNN karena pada beberapa riset mendapat hasil yang akurat. Harapan kedepannya riset ini dapat dikembangkan sehingga dapat membantu konservasi burung untuk melindungi burung hantu.

II. KAJIAN TEORI

A. Studi Terkait

Penelitian mengenai pengenalan suara burung telah dilakukan dalam penelitian [5] menggunakan *Deep*

Convolutional Neural Network dengan arsitektur ResNet50. Penelitian ini dilakukan menggunakan data yang diambil dari web Xeno-Canto dengan jumlah data dari 46 spesies burung sebanyak 2814 audio data. Dataset kemudian diubah menjadi spektrogram. Data yang sudah diubah dan diberi label, dibagi menjadi 2 bagian. 75% data untuk data train dan 25% digunakan untuk data *validation*. Sebelum pelatihan, model ResNet-50 dimuat dengan *ImageNet-trained weights* yang sesuai. Hasilnya, ResNet50 yang dilatih pada gambar berukuran 512 x 1024 yang dipotong secara acak mendapat hasil akurasi sebesar 65% untuk data *train* dan 57% untuk data *validation*. Eksperimen dilanjutkan dengan melatih model dengan gambar berukuran 512 x 512 dan diberikan *noise* secara acak. Hal ini meningkatkan akurasi menjadi sebesar 72% untuk data *train* dan 65% untuk data *validation* dengan nilai 400 *epoch*.

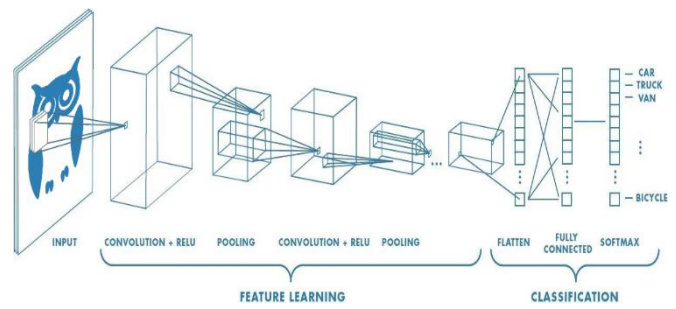
Klasifikasi dan pengenalan suara burung juga dilakukan pada penelitian [9] menggunakan CNN dengan arsitektur ResNet18 dan Inception V3. Penelitian ini menrancang sistem yang dapat mengenali 659 spesies burung dari 50.000 data audio. Klasifikasi dan pengenalan dilakukan dengan bantuan augmentasi dan normalisasi data. Arsitektur ResNet18 menggunakan *optimizer Stochastic Gradient Descent* (SGD), sementara arsitektur Inception V3 menggunakan *optimizer Adam*. Nilai performansi pada arsitektur Inception V3 memperoleh hasil sebesar 23% dan 13% pada arsitektur ResNet18. Karena arsitektur Inception V3 memperoleh hasil yang lebih besar, maka arsitektur ini dipakai juga untuk mengevaluasi per *running* sistem dengan skenario *running* pertama dan kedua tidak menambahkan *Gaussian noise* pada seluruh spektrogram dan menambahkan *Gaussian noise* pada *running* ketiga dan keempat. Nilai performansi yang didapat pada *running* pertama sebesar 14% dan naik pada *running* keempat dengan nilai 16%.

Adapun penelitian CNN untuk deteksi audio burung [10] yang menggunakan sebanyak 24.310 data audio dengan panjang audio rata-rata 10 detik dan total 68 jam. Metode CNN juga dibandingkan dengan *Convolutional Recurrent Neural Network* (CRNN) dan mendapat hasil sebesar 85.5% untuk metode CNN serta 88.5% untuk metode CRNN.

B. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image* dan biasa digunakan untuk mendeteksi dan mengenali objek pada sebuah *image* [11]. CNN secara otomatis dapat mengekstrak fitur tingkat tinggi dari fitur masukan mentah, yang jauh lebih kuat daripada fitur yang dirancang manusia.

Arsitektur *deep learning* yang digunakan dalam model prediksi sebagian besar adalah jaringan residual (ResNet), yang banyak digunakan di bidang seperti pemrosesan gambar dan pemrosesan bahasa alami. Ekstraksi fitur pada CNN lebih baik dalam pengenalan pola sehingga peningkatan kinerja model ini menjadi signifikan.



Gambar 1.
Model CNN

Pada Gambar 1, ditunjukkan model CNN yang paling umum. Pada proses *feature learning* terdapat layer *convolution Relu* dan *pooling* untuk melatih data dari fitur yang sudah didapatkan. Kemudian dilanjutkan dengan proses klasifikasi menggunakan beberapa layer seperti *fully connected* dan *softmax*.

CNN terdiri dari *input layer*, *hidden layer* dan *output layer*. *Hidden layer* CNN biasanya terdiri dari serangkaian *convolutional layer* dengan perkalian atau perkalian titik lainnya. Disebut *hidden layer* karena *input* dan *outputnya* ditutupi oleh fungsi aktivasi dan konvolusi akhir. Langkah-langkah yang terlibat dalam proses pengembangan CNN adalah :

1. Konvolusi diikuti dengan penerapan Fungsi Rectifier
2. *Pooling*
3. *Flattening*
4. *Fully connected*.

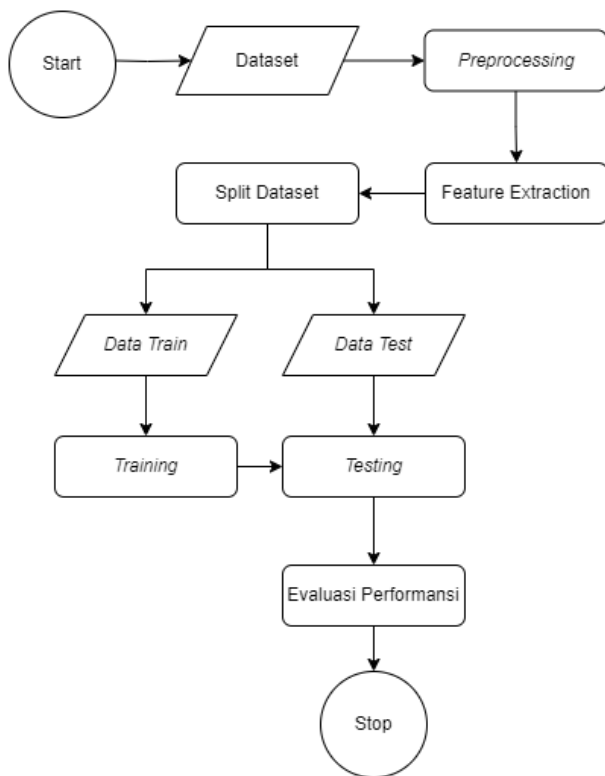
Convolutional layer menggabungkan *input* dan meneruskan hasilnya ke layer berikutnya. *Convolutional layer* dalam *Neural Network* harus memiliki atribut berikut:

- a. Kernel konvolusional ditentukan oleh lebar dan tinggi (*hyper-parameter*)
- b. Jumlah saluran masukan dan saluran keluaran (*hyper-parameter*)
- c. Kedalaman filter Konvolusi (saluran inputan) harus sama dengan saluran nomor (kedalaman) dari peta fitur masukan.

III. METODE

A. Studi Literatur

Sistem yang akan dibuat adalah sistem klasifikasi genus burung hantu yang memiliki performa akurasi yang tinggi dengan waktu komputasi yang cepat. Berikut ini adalah alur sistem klasifikasi genus burung hantu.



Gambar 2.
Flowchart model

Sistem yang akan dibuat bertujuan untuk mengenali suara dari burung hantu yang kemudian diproses menggunakan *audio processing* dan diklasifikasikan berdasarkan genusnya. Pada Gambar 2 di atas digambarkan alur pembuatan model sistem. Sistem ini akan dibangun menggunakan bahasa pemrograman Python dan beberapa *library* yang mendukung pembangunan sistem. Proses pembuatan model diawali dengan pengumpulan data, kemudian dilakukan *preprocessing*. Selanjutnya dilakukan proses *feature extraction* untuk mendapatkan label dan membagi data menjadi data *train* dan data *test*. Data *train* akan dilatih untuk mempelajari pola dan kemudian dilakukan *testing* untuk mengklasifikasi genus burung hantu dengan data *test*.

B. Dataset

Dataset yang digunakan merupakan data yang diambil secara manual dari Xeno-Canto. Data ini merupakan data yang bebas digunakan oleh siapapun. Data yang diambil, dipilah secara manual dengan mengambil lima (5) genus yaitu *Otus*, *Megascopus*, *Bubo*, *Glaucidium*, dan *Strix*. Data yang diambil merupakan data audio yang memiliki *noise* paling rendah. Jumlah data yang digunakan berjumlah sekitar 99 file suara. Durasi awal suara sekitar 10-20 detik. Jenis file yang digunakan berformat .MPEG. Data yang terkumpul dapat dilihat dari rincian sebagai berikut.

- Bubo : 10 data audio
- Glaucidium : 13 data audio
- Megascopus : 19 data audio
- Otus : 14 data audio
- Strix : 43 data audio

C. Preprocessing

Preprocessing merupakan tahap awal dari proses sistem klasifikasi yang bertujuan untuk menghilangkan *noise* sehingga dapat meningkatkan performansi sistem. Setelah data terkumpul, data disiapkan dengan membagi audio menjadi potongan-potongan audio burung hantu berdurasi 1-3 detik dari dataset. Pembagian audio ini membagi sinyal yang panjang menjadi segmen yang lebih pendek. Proses ini disebut juga *trimming* audio dataset. Pada tahap ini juga dilakukan *noise reduction* yang masih tersisa dari pemilihan dataset secara manual dan mengubah format dataset menjadi .MP3 menggunakan *software* Audacity dan Wondershare Filmora. Dari hasil *trimming* didapat data dengan rincian pada Tabel 1.

Tabel 1.
Perbandingan Jumlah Data

| Genus | Jumlah data awal | Jumlah data setelah <i>trimming</i> |
|-------------------|------------------|-------------------------------------|
| <i>Bubo</i> | 10 | 104 |
| <i>Glaucidium</i> | 13 | 109 |
| <i>Megascopus</i> | 19 | 101 |
| <i>Otus</i> | 14 | 139 |
| <i>Strix</i> | 43 | 158 |
| Total | 99 | 611 |

D. Feature Extraction

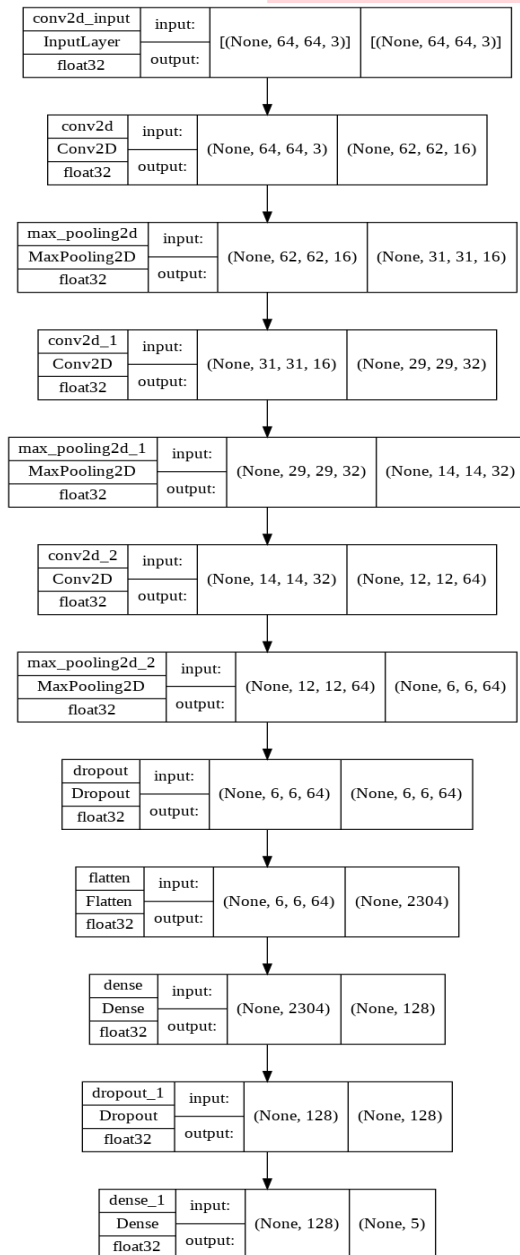
Feature extraction merupakan proses penting dalam klasifikasi audio. Terdapat banyak fitur yang dapat digunakan untuk *preprocessing* audio data. Beberapa yang digunakan dalam penelitian ini adalah sebagai berikut.

- *Root Mean Square Energy* (RMSE) merupakan fitur yang berguna dalam tugas segmentasi audio dan klasifikasi [12].
- *Spectral Centroid* berguna untuk mengukur posisi dan bentuk spektral dari titik keseimbangan distribusi energi. Secara sederhana *spectral centroid* disebut gravitasi dari spektrum [13].
- *Spectral Bandwith* merupakan rata-rata tertimbang jarak pita frekuensi dari *Spectral Centroid* [12].
- *Spectral Rolloff* berguna untuk menentukan nilai frekuensi yang terletak dibawah nilai distribusi spektrum (biasanya bernilai 90%) terkonsentrasi [13]
- *Mel-Frequency Cepstrum Coefficients* (MFCC) adalah salah satu bentuk representasi cepstral dari sinyal dimana frekuensi didistribusi berdasarkan skala mel bukan secara pendekatan linear [13].
- *Zero-Crossing Rate* (ZCR) adalah nilai perubahan sinyal dalam frame atau berapa kali sinyal berubah nilai dari positif ke negatif atau sebaliknya dibagi dengan panjang frame [13].

E. Classification

Setelah mendapat dataset audio akhir, dataset tersebut diubah menjadi spektrogram untuk merepresentasikan visual dari spektrum suara atau sinyal lain yang bervariasi terhadap waktu. Dataset yang sudah diubah kemudian dilakukan *feature extraction*. Fitur yang digunakan adalah RMS (*Root Mean Square*), *spectral centroid*, *spectral bandwidth*, *spectral rolloff*, *zero crossing rate*, dan MFCC.

Untuk klasifikasi, dataset dibagi menjadi 2, yaitu data *train* dan data *test*. Data *train* diambil sebanyak 90% dari data audio dan 10% sisanya untuk data *test*. Kemudian data *train* dilatih menggunakan CNN. Model arsitektur CNN yang dibuat meliputi *MaxPooling2D*, *Convolution2D*, *Dropout*, *Flatten* dan *Dense layer*. *Dropout Layer* digunakan agar tidak terjadi *overfit* pada data yang masuk ke *Flatten Layer*. Data akan di *dropout* sebesar 20%. Model arsitektur yang digunakan dalam proses pengujian dijelaskan pada Gambar 3.



GAMBAR 3. Layer CNN

F. Testing

Pengujian sistem dilakukan melalui tiga skenario sebagai berikut.

1. Skenario Pertama

Skenario pertama dilakukan pengujian pengaruh dari nilai *epoch* terhadap performansi sistem yang bertujuan untuk memperoleh nilai *epoch* dengan hasil akurasi yang tinggi. Nilai *epoch* yang dibandingkan pada skenario ini yaitu 50, 100, dan 150. Pengujian dilakukan dengan parameter *learning rate* dengan nilai 0.05 dan menggunakan *optimizer Adagrad*.

2. Skenario Kedua

Skenario kedua dilakukan pengujian pengaruh dari nilai *learning rate* terhadap performansi sistem yang bertujuan untuk memperoleh nilai *learning rate* dengan hasil akurasi yang tinggi. Nilai *learning rate* yang akan dibandingkan pada skenario ini yaitu 0.05, 0.001, dan 0.0005. Pengujian dilakukan dengan parameter *epoch* dengan nilai 100 dan menggunakan *optimizer Adagrad*.

3. Skenario Ketiga

Skenario ketiga dilakukan pengujian pengaruh dari nilai *optimizer* terhadap performansi sistem yang bertujuan untuk memperoleh nilai *optimizer* dengan hasil akurasi yang tinggi. Nilai *optimizer* yang akan dibandingkan pada skenario ini yaitu *Adam*, *Adagrad*, dan *RMSprop*. Pengujian dilakukan dengan parameter *epoch* dengan nilai 100 dan *learning rate* dengan nilai 0.05.

IV. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan skenario membandingkan nilai *hyperparameter epoch*, *learning rate*, dan *optimizer* pada data. Pengukuran performansi dilakukan pada proses *testing* menggunakan metrik akurasi, presisi, *recall*, dan *F1-Score* yang diperoleh dari *Confusion Matrix*. Nilai akurasi, presisi, *recall*, dan *F1-Score* secara berurutan diperoleh dari persamaan 1, 2, 3, dan 4.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$Precision = \frac{TP}{FP+TP} \tag{3}$$

$$F1\ Score = 2 \times \frac{precision \times recall}{precision+recall} \tag{4}$$

Pengujian sistem dilakukan melalui tiga skenario dengan hasil sebagai berikut.

1. Skenario Pertama

Skenario pertama adalah membandingkan nilai *epoch* antara 50, 100 dan 150. Hasil dari skenario ini dapat dilihat pada Tabel 2.

TABEL 2. Hasil Skenario Pertama

| Epoch | Accuracy | Loss |
|-------|----------|--------|
| 50 | 99.2% | 0.0315 |
| 100 | 99.8% | 0.0052 |

| | | |
|-----|-------|--------|
| 150 | 93.3% | 0.2065 |
|-----|-------|--------|

| | | |
|----------------|-------|--------|
| <i>RMSprop</i> | 24.6% | 1.5954 |
|----------------|-------|--------|

Berdasarkan Tabel 2 terlihat hasil pengujian skenario memiliki nilai yang beragam untuk setiap nilai *epoch* yang diujikan. Pada *epoch* 50 didapatkan nilai akurasi 99.2% dengan nilai *loss* 0.0315, nilai *precision* 22%, nilai *recall* 23%, dan nilai *f1-score* 22%. Pada *epoch* 100 didapatkan nilai akurasi 99.8% dengan nilai *loss* 0.0052, nilai *precision* 19%, nilai *recall* 20%, dan nilai *f1-score* 19%. Pada *epoch* 150 didapatkan nilai akurasi 93.3% dengan nilai *loss* 0.2065, nilai *precision* 18%, nilai *recall* 18%, dan nilai *f1-score* 17%.

Dari hasil pengujian dapat dilihat bahwa nilai *epoch* 150 membuat hasil akurasi menurun meskipun tidak terlalu jauh. Dengan nilai *epoch* 100 saja sudah cukup untuk mencapai hasil yang baik. Maka untuk skenario berikutnya akan menggunakan nilai *epoch* dengan hasil akurasi terbaik yaitu *epoch* dengan nilai 100.

2. Skenario Pertama

Skenario kedua adalah membandingkan nilai *learning rate* antara 0.05, 0.01 dan 0.005. Hasil dari skenario ini dapat dilihat pada Tabel 3.

TABEL 3.
Hasil Skenario Kedua

| <i>Learning rate</i> | <i>Accuracy</i> | <i>Loss</i> |
|----------------------|-----------------|-------------|
| 0.05 | 99.2% | 0.0288 |
| 0.01 | 95.6% | 0.1244 |
| 0.005 | 89.4% | 0.3030 |

Berdasarkan Tabel 3 menunjukkan hasil pengujian skenario memiliki nilai yang beragam. Pada nilai *learning rate* 0.05 didapatkan nilai akurasi 99.2% dengan nilai *loss* 0.0288, nilai *precision* 22%, nilai *recall* 23%, dan nilai *f1-score* 22%. Pada nilai *learning rate* 0.01 didapatkan nilai akurasi 95.6% dengan nilai *loss* 0.1244, nilai *precision* 21%, nilai *recall* 21%, dan nilai *f1-score* 21%. Pada nilai *learning rate* 0.005 didapatkan nilai akurasi 89.4% dengan nilai *loss* 0.3030, nilai *precision* 32%, nilai *recall* 31%, dan nilai *f1-score* 29%.

Dari hasil pengujian dapat dilihat bahwa semakin kecil nilai *learning rate* maka akan membuat hasil akurasi semakin menurun. Maka untuk skenario berikutnya akan menggunakan nilai *learning rate* dengan hasil akurasi terbaik yaitu *learning rate* dengan nilai 0.05.

3. Skenario Ketiga

Skenario ketiga adalah membandingkan nilai *optimizer* antara *Adam*, *Adagrad* dan *RMSprop*. Hasil dari skenario ini dapat dilihat pada Tabel 4.

TABEL 4.
Hasil Skenario Ketiga

| <i>Optimizer</i> | <i>Accuracy</i> | <i>Loss</i> |
|------------------|-----------------|-------------|
| <i>Adam</i> | 25.9% | 1.5935 |
| <i>Adagrad</i> | 99.8% | 0.0052 |

Berdasarkan Tabel 4 menunjukkan hasil pengujian skenario memiliki nilai yang beragam. Pada nilai *optimizer Adam* didapatkan nilai akurasi 25.9% dengan nilai *loss* 1.5935, nilai *precision* 5%, nilai *recall* 20%, dan nilai *f1-score* 8%. Pada nilai *optimizer Adagrad* didapatkan nilai akurasi 99.8% dengan nilai *loss* 0.0052, nilai *precision* 21%, nilai *recall* 21%, dan nilai *f1-score* 21%. Pada nilai *optimizer RMSprop* didapatkan nilai akurasi 24.6% dengan nilai *loss* 1.5954, nilai *precision* 5%, nilai *recall* 20%, dan nilai *f1-score* 8%.

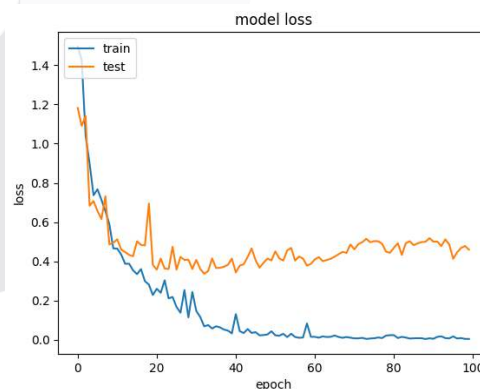
Dari hasil pengujian dapat dilihat *optimizer Adam* dan *RMSprop* memberikan hasil akurasi yang jauh lebih kecil dibandingkan *optimizer Adagrad*. Dengan demikian disimpulkan bahwa *optimizer Adagrad* adalah *optimizer* yang optimal untuk sistem yang dibangun sehingga akan digunakan sebagai parameter terbaik.

Dari hasil pengujian tiga skenario sebelumnya, diperoleh parameter-parameter terbaik. Tabel 5 menunjukkan parameter-parameter terbaik untuk sistem yang dibangun.

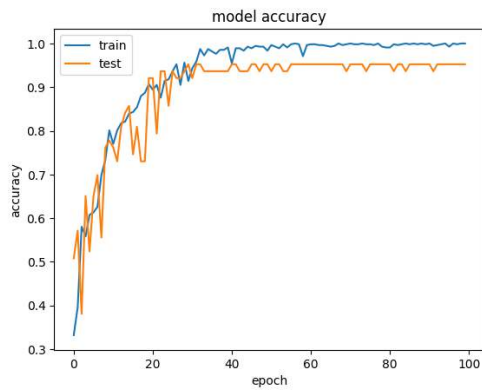
TABEL 5.
Hasil Parameter Terbaik

| <i>Hyperparameter</i> | Parameter Terbaik |
|-----------------------|-------------------|
| <i>Epoch</i> | 100 |
| <i>Learning rate</i> | 0.05 |
| <i>Optimizer</i> | <i>Adagrad</i> |

Pengujian menggunakan parameter terbaik diperoleh nilai akurasi sebesar 99.8% dengan nilai *loss* sebesar 0.0052. Gambar 4 dan Gambar 5 menunjukkan grafik akurasi dan grafik *loss*.



GAMBAR 4.
Grafik Akurasi



GAMBAR 5. Grafik Loss

Performansi sistem pengujian dengan parameter terbaik diukur dengan empat nilai yaitu nilai *precision*, *recall*, *f1-score*, dan *confusion matrix*. Tabel 6 dan Gambar 6 menunjukkan hasil dari pengukuran performansi.

TABEL 6. Classification Report Pengujian Parameter Terbaik

| Kelas | Precision | Recall | F1-Score | Jumlah data |
|-------------------|-----------|--------|----------|-------------|
| <i>Bubo</i> | 11 | 9 | 10 | 11 |
| <i>Glaucidium</i> | 25 | 27 | 26 | 11 |
| <i>Megascopus</i> | 33 | 36 | 35 | 11 |
| <i>Otus</i> | 8 | 7 | 7 | 14 |
| <i>Strix</i> | 18 | 19 | 18 | 16 |
| Total | 18 | 19 | 19 | 63 |



GAMBAR 6. Confusion Matrix Pengujian Parameter Terbaik

Tabel 6 memperlihatkan sistem pada pengujian memakai parameter terbaik dengan nilai *precision* sebesar 18%, *recall* 19%, dan *f1-score* 19%. Gambar 6 memperlihatkan dari 63 data audio dari data test yang digunakan hanya ada 11 gambar yang terdeteksi sesuai dengan kelasnya, yaitu 1 data audio kelas *Bubo*, *Glaucidium* dan *Megascopus*, 3 data audio kelas *Strix*, serta 5 data audio kelas *Otus*.

V. KESIMPULAN

Dari hasil pengujian tiga skenario yang telah dilakukan, didapat hasil bahwa sistem yang telah dirancang menggunakan *Convolutional Neural Network (CNN)* dapat mengklasifikasikan burung hantu ke dalam lima kelas yaitu *Otus*, *Megascopus*, *Bubo*, *Glaucidium*, dan *Strix*. Pengujian terbaik diperoleh menggunakan parameter pengujian nilai *epoch* 100, *learning rate* 0.05, dan dengan menggunakan *optimizer Adagrad*. Hasil yang diperoleh adalah akurasi sebesar 99.8% dengan *loss* 0.0052, *precision* 16%, *recall* 16%, dan *f1-score* 16%. Meski mendapat nilai akurasi yang tinggi, pengujian tersebut mendapat nilai *precision*, *recall*, dan *f1-score* yang sangat rendah. Ini menyebabkan model yang dibangun belum mencapai hasil yang baik karena jumlah dataset tiap kelas yang tidakimbang.

REFERENSI

- [1] Sangster, G., King, B.F., Verbelen, P., Trainor, C.R., "A New Owl Species of the Genus *Otus* (Aves: Strigidae) from Lombok, Indonesia," *PLoS ONE*, vol. 8(2), 2013.
- [2] Morony, J.J., Bock, W.J., dan Farrand, J., Reference list of the bird of the world, New York: Department of Ornithology, American Museum of Natural History, 1975.
- [3] König, C., Weick, F., *Owls: a guide to the owls of the world*. Second Edition, Pica Press, 2008.
- [4] "IUCN Red List," [Online]. Available: <https://www.iucnredlist.org/search/stats?taxonomies=22673078&searchType=species>. [Diakses 13 November 2020].
- [5] Sankupellay, M., dan Konovalov, D., "Bird call recognition using deep convolutional neural network ResNet50," vol. 7, pp. 1-8, November 2018.
- [6] J. Q.-S. R. G.-A. A. Ludeña-Choez, "Bird sound spectrogram decomposition through Non-Negative Matrix Factorization for the acoustic classification of bird species," *PLoS ONE*, vol. 12(6), 2017.
- [7] Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, Malcolm., Weiss, R. J, Wilson, K., "CNN architectures for large-scale audio classification," 2017.
- [8] "GeeksforGeeks," [Online]. Available: <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>. [Diakses 9 September 2023].
- [9] Koh, C. Y., Chang, J. Y., Tai, C. L., Huang, D. Y., Hsieh, H. H., & Liu, Y. W., "Bird Sound Classification Using Convolutional Neural Networks," *CLEF*, September 2019.
- [1] C,akır, E., Adavanne, S., Parascandolo, G., Drossos, K., 0] Virtanen, T., "Convolutional Recurrent neural networks for bird audio detection," 2017.
- [1] "Medium," 13 November 2017. [Online]. Available: <https://medium.com/@samuelsena/pengenalan-deep-1>

learning-part-7-convolutional-neural-network-cnn-b003b477dc94. [Diakses 28 November 2020].

- [1] “Devopedia,” 2021. [Online]. Available:
- 2] <https://devopedia.org/audio-feature-extraction>. [Diakses 7 September 2023].

- [1 “BINUS University Master of Computer Science,”
- 3] [Online]. Available:
<https://mti.binus.ac.id/2017/11/13/ekstraksi-fitur-untuk-genre-musik/>. [Diakses 7 September 2023].

