

KLASIFIKASI JENIS FAUNA DENGAN MENGGUNAKAN *TRANSFER LEARNING* GOOGLNET

Asriyanik Asriyanik^[1]; Dennis Saputra Ariansyah^[2]

Teknik Informatika Fakultas Sains dan Teknologi^{[1][2]}
Universitas Muhammadiyah Sukabumi^{[1][2]}
asriyanik263@ummi.ac.id

INFO ARTIKEL	INTISARI
Diajukan : 25-04-2024	<i>Kemajuan teknologi yang pesat kini menjadi bagian tak terpisahkan dari kehidupan, termasuk dalam sektor pendidikan. Di era Revolusi 4.0, pendidikan Indonesia memerlukan adopsi teknologi informasi dan komunikasi (TIK) untuk mempercepat peningkatan kualitas pendidikan. Internet telah banyak digunakan dalam pembelajaran mulai dari usia dini hingga dewasa. Namun, banyak sekolah masih belum memanfaatkannya sepenuhnya, terutama dalam mengajar anak usia dini tentang hal-hal seperti pengenalan jenis fauna. Pengembangan model pengenalan jenis fauna untuk pembelajaran anak usia dini adalah salah satu solusinya. Terdapat beberapa metode dalam pengembangan model machine learning, dan pada penelitian ini digunakan metode SEMMA yang merupakan singkatan dari Sample, Explore, Modify, Model dan Assess, sedangkan algoritma yang akan digunakan adalah salah satu arsitektur Convolutional Neural Network (CNN) yaitu transfer learning GoogleNet. Dataset yang digunakan berasal dari pencarian gambar di Google Photo dan Kaggle yang terdiri dari 5500 dataset dan terbagi menjadi lima puluh kelas. Sebelum dimodelkan dataset dilakukan modifikasi untuk memastikan dataset ada pada kondisi yang baik. Hasil pemodelan dievaluasi dengan mengukur keakurasian dan tingkat loss (kerugian). Dari hasil evaluasi didapatkan nilai keakurasian model pada data training mencapai 98,36% sedangkan nilai kerugian (loss) adalah 0.06%. Dan nilai keakurasian model dari data validasi adalah 89,83% dan nilai kerugian (loss) adalah 38,84%. Untuk memudahkan penggunaan, hasil pemodelan ini diimplementasikan pada aplikasi berbasis web, sehingga mudah digunakan oleh berbagai pihak termasuk guru anak usia dini dalam proses pengajaran kepada siswa.</i>
Diterima : 15-05-2024	
Diterbitkan: 30-06-2024	
Kata Kunci : <i>Convolutional Neural Network, Deteksi Objek, Fauna, Googlenet, Transfer Learning,</i>	

I. PENDAHULUAN

Perkembangan Teknologi Informasi dan Komunikasi (TIK) telah mendorong berbagai perubahan pengelolaan dalam setiap aspek kehidupan termasuk bidang pendidikan (Putriani & Hudaidah, 2021). Beberapa pengaruh TIK dalam bidang pendidikan di antaranya adalah mempermudah akses informasi, pembelajaran interaktif, pembelajaran jarak jauh, personalisasi pembelajaran, efisiensi dan manajemen dalam pembelajaran, inklusivitas pendidikan dan lainnya (Widiasanti et al., 2023). Salah satu yang berkembang adalah penggunaan internet dalam media pembelajaran. Internet benar-benar banyak mengubah proses pembelajaran menjadi lebih mudah dan interaktif. Pada saat ini telah banyak media-media pembelajaran yang dengan mudah didapatkan melalui internet walaupun belum seluruhnya. Beberapa metode pembelajaran tradisional masih digunakan seperti penggunaan

poster, gambar-gambar cetak sebagai visual pembelajaran masih digunakan terutama untuk anak-anak.

Menurut Asosiasi Penyelenggaraan Jasa Internet Indonesia (APJII) “penggunaan *internet* di Indonesia mencapai 77,02% pada tahun 2021-2022. Berdasarkan kelompok umur, anak-anak yang berusia 5-12 tahun telah mencapai 62,43%” (Pahlevi, 2022). Berdasarkan data tersebut sangat memungkinkan penggunaan media pembelajaran berbasis internet diterapkan pada anak-anak termasuk anak usia dini.

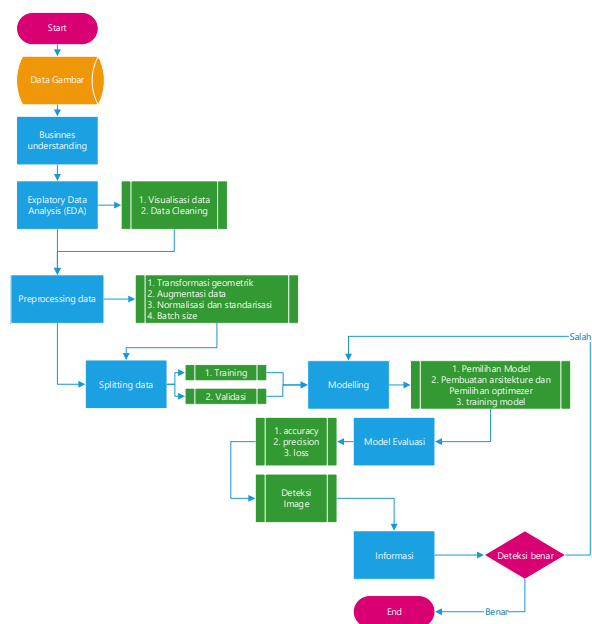
Salah satu pembelajaran pada anak usia dini atau taman kanak-kanak adalah pengenalan jenis fauna yang saat ini masih menggunakan media visual seperti gambar poster, namun metode ini masih terbatas dan tidak dapat dilakukan secara *realtime* terhadap jenis fauna yang baru. Untuk menyelesaikan hal ini dapat dilakukan pengenalan objek baru dengan membuat pemodelan klasifikasi

fauna dengan algoritma CNN (*Convolutional Neural Network*) (Babulal & Das, 2022).

CNN terdiri dari beberapa jenis arsitektur di antaranya GoogleNet, AxelNet, RestNet, Inception RestNet dan VGGNet (Sabilla, 2020). Dari arsitektur tersebut, GoogleNet memiliki tingkat kesalahan yang lebih rendah dibandingkan dengan yang lainnya yaitu mencapai 6.67% sedangkan AxelNet 15,3% (Ifantiska, 2022). Penelitian-penelitian yang lain yang telah dilakukan yang berkaitan dengan deteksi objek yaitu klasifikasi fauna fauna terbang (Inaya et al., 2024) dan klasifikasi jenis-jenis anjing. Dari penelitian tersebut, pengenalan fauna masih dilakukan secara parsial yaitu hanya untuk fauna terbang belum secara menyeluruh dan belum diukur berapa nilai keakuratannya. Pengukuran keakuratan model sangat perlu dilakukan untuk memastikan model yang dihasilkan memiliki kualitas yang baik dalam proses klasifikasi jenis fauna. Pada penelitian ini dilakukan deteksi untuk seluruh jenis fauna sehingga dapat mencapai area klasifikasi objek yang lebih luas yang menggunakan arsitektur GoogleNet dan hasil pemodelan dilakukan pengukuran keakuratan dengan *confussion matrix*.

II. BAHAN DAN METODE

Metode yang digunakan dalam proses penelitian adalah dengan menggunakan metode SEMMA yang merupakan salah satu bentuk metodologi pengembangan *data science* ataupun machine learning dengan tahapan 1) *Sample*, 2) *Explore*, 3) *Modify*, 4) *Model* dan 5) *Asses* (Palacios et al., 2017) yang tersaji pada Gambar 1.



Gambar 1. Tahapan Penelitian Klasifikasi Jenis Fauna dengan Metode SEMMA

Tahap pertama dilakukan pengumpulan sampel atau *dataset* tentang berbagai jenis fauna dari Kaggle dan Google Photo. Tahap kedua adalah melakukan eksplorasi (*explore*) *dataset* yang meliputi deskripsi dan visualisasi data yang telah didapatkan (Alizah et al., 2020), sehingga dapat diketahui kondisi *dataset* fauna yang ada agar memudahkan tahapan selanjutnya.

Tahap ketiga yaitu memodifikasi (*modify*) *dataset* fauna yang telah didapatkan dengan cara melakukan augmentasi gambar untuk meningkatkan variasi data pelatihan (*training*) tanpa perlu mengumpulkan data tambahan. Proses augmentasi gambar meliputi rotasi, translasi, skalasi, flip horizontal dan vertikal, pemotongan, perubahan kecerahan, ukuran gambar dan lainnya (Ardiansyah & Hasan, 2023). Tahap keempat adalah melakukan pemodelan yang akan dibantu dengan arsitektur CNN yaitu GoogleNet. Dan tahap terakhir adalah dilakukan evaluasi (*asses*) dengan mencari nilai akurasi (*accuracy*), presisi (*precision*), kehilangan (*loss*) dan *recall* dan *f1-score* dari data uji yang sudah dideteksi (Sanjaya et al., 2023). Persamaan evaluasi akurasi model yaitu ada pada persamaan (1), (2) dan (3).

$$Precision = \frac{TP}{TP + FP} \dots \dots \dots (1)$$

$$Recall = \frac{TP}{TP + FN} \dots \dots \dots (2)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \dots \dots \dots (3)$$

Dimana:
TP = True Positive
FP = False Positive
FN = False Negative

Model klasifikasi fauna yang telah didapatkan akan disimulasikan dalam sistem berbasis *web*, sehingga dapat lebih mudah digunakan. Sistem ini dapat diakses melalui perangkat laptop maupun *smartphone*.

III. HASIL DAN PEMBAHASAN

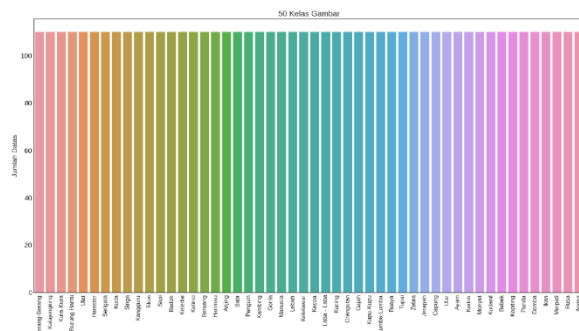
1. Pengumpulan *Dataset*

Dataset dikumpulkan dengan cara manual yaitu melalui proses pencarian contoh gambar dari Google foto dan Kaggle dengan kata kunci “hewan” dan “*Animal Image Dataset*”. *Dataset* yang didapatkan berjumlah 5500 gambar yang terbagi menjadi 110 data setiap kelasnya. Jenis fauna dibagi menjadi 50 kelas yaitu kelas angsa, anjing, ayam, babi, badak, bebek, berang-berang, beruang, buaya, burung hantu, capung, domba, gajah, gorila,

hamster, harimau, ikan, jerapah, kalajengking, kambing, kangguru, kecoa, keledai, kelelawar, kelinci, kepiting, kodok, kucing, kuda, kudaniil, kupu-kupu, kura-kura, laba-laba, lebah, lumba-lumba, manusia, merpati, monyet, orang utan, panda, penguin, rusa, sapi, serigala, singa, tikus, tupai, ular, ulat, dan zebra.

2. Eksplorasi Dataset (Explore)

Tahap *eksplorasi* data ini dilakukan memastikan jumlah data pada masing-masing kelas memiliki jumlah yang sama, jika ada kelas yang memiliki jumlah data melebihi atau kurang dari data rata-rata maka akan dilakukan proses *resampling* atau *downsampling*. Tahap ini juga dilakukan visualisasi data menggunakan *library* matplotlib dan seaborn, visualisasi dihasilkan ditampilkan pada Gambar 2.



Sumber: Hasil Pengolahan Data
Gambar 2. Visualisasi Jumlah Data untuk Setiap Kelas Fauna

Hasil visualisasi pada Gambar 2 menunjukkan bahwa setiap kelas memiliki jumlah data yang sama sehingga jumlah *dataset* telah seimbang.

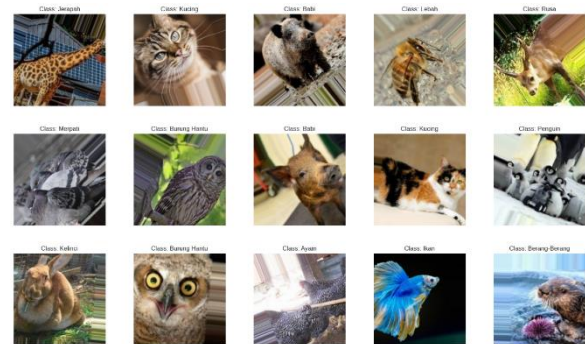
Selain itu juga dilakukan analisis karakteristik gambar dan variasi visual *dataset* jenis fauna dengan mengambil sampel dari setiap kelas seperti pada Gambar 3. Proses ini dilakukan dengan cara mengubah gambar keseluruhan dengan ukuran lebar dan tinggi, normalisasi gambar, mengubah gambar menjadi RGB (*Red, Green, Blue*) untuk mempermudah proses *augmentasi* yang akan dilakukan pada dataset.



Sumber: Hasil Pengolahan Data
Gambar 3. Visualisasi Contoh Dataset sebelum Modifikasi

3. Modifikasi Dataset (Modify)

Tahap modifikasi dataset yang dilakukan pada penelitian ini adalah *augmentasi* gambar yaitu mempertajam gambar, mengubah posisi gambar baik dengan cara rotasi maupun flip horizontal dan vertikal, juga mengubah ukuran gambar dan lainnya. Pada Gambar 4 tersaji *dataset* yang telah dilakukan *augmentasi*. *Augmentasi* yang dilakukan pada dataset fauna meliputi: 1) mengubah skala intensitas piksel gambar menjadi 224, 2) memutar gambar hingga 45 derajat, 3) menggeser gambar secara horizontal dan vertikal sebesar 20%, 4) menerapkan transformasi shear hingga 20%, 5) mengisi piksel kosong dalam gambar, 6) membalik gambar secara acak, 7) membagi data validasi sebesar 10%, 8) mengubah ukuran semua gambar dalam dataset menjadi 224, dan 9) membagi data menjadi 80% untuk pelatihan dan 20% untuk evaluasi.



Sumber: Hasil Pengolahan Data
Gambar 4. Hasil Augmentasi Dataset Fauna

4. Pemodelan Data (Model)

Tahap pemodelan dilakukan berdasarkan arsitektur *Convolutional Neural Network* (CNN) dengan proses memasukkan data menggunakan *batch size* 32, lebar gambar 224, tinggi gambar 224, dan 3 saluran warna dari *dataset* gambar. *Feature maps* menggunakan fungsi aktivasi ReLU, sementara proses klasifikasi menggunakan aktivasi *softmax* untuk menangani data kategori lebih dari dua kelas. Proses klasifikasi dilakukan dengan metode *multi-class* menggunakan *batch size* 32 dan langkah setiap *epoch* sebanyak 20

```
1. last_output = base_model.output
2. x = GlobalAveragePooling2D()(last_output)
3. x = Flatten()(x)
4. x = Dense(512, activation = 'relu')(x)
5. x = Dense(1024, activation = 'relu')(x)
6. x = Dropout(0.1)(x)
7. outputs = Dense(50, activation = 'softmax')(x)
```

```
8. model = Model(inputs =  
base_model.inputs, outputs = outputs)
```

Pada proses pemodelan diterapkan lapisan *drop out* dengan tingkat 10% untuk membantu mengurangi *overfitting* selama pelatihan model. Base model yang digunakan dari *transfer learning* GoogleNet dengan mengambil nilai rata-rata dari setiap fitur jenis fauna pada *output* spasial dua dimensi (2D). Lalu *input* data multidimensi diubah menjadi satu dimensi dan lapisan *neuron* yang terhubung penuh menerima *neuron* di lapisan sebelumnya.

```
[1]: In[9]: Map  
# Save model.h5 dengan nama model yang sudah  
model_path = "/content/gdrive/MyDrive/colab_models/model.h5"  
  
# Simpan file  
with h5py.File(model_path, "w") as f:  
# Simpan layer dan dataset dalam file  
f.put("layer", layer)  
f.put("dataset", dataset)  
print("Done!")  
  
model_weights/activation_106_output_group/model_weights/activation_106_output_group (8 members)  
model_weights/activation_107_output_group/model_weights/activation_107_output_group (8 members)  
model_weights/activation_108_output_group/model_weights/activation_108_output_group (8 members)  
model_weights/activation_109_output_group/model_weights/activation_109_output_group (8 members)  
model_weights/activation_110_output_group/model_weights/activation_110_output_group (8 members)  
model_weights/activation_111_output_group/model_weights/activation_111_output_group (8 members)  
model_weights/activation_112_output_group/model_weights/activation_112_output_group (8 members)  
model_weights/activation_113_output_group/model_weights/activation_113_output_group (8 members)  
model_weights/activation_114_output_group/model_weights/activation_114_output_group (8 members)  
model_weights/activation_115_output_group/model_weights/activation_115_output_group (8 members)  
model_weights/activation_116_output_group/model_weights/activation_116_output_group (8 members)  
model_weights/activation_117_output_group/model_weights/activation_117_output_group (8 members)  
model_weights/activation_118_output_group/model_weights/activation_118_output_group (8 members)  
model_weights/activation_119_output_group/model_weights/activation_119_output_group (8 members)  
model_weights/activation_120_output_group/model_weights/activation_120_output_group (8 members)  
model_weights/average_pooling2d_13_output_group/model_weights/average_pooling2d_13_output_group (8 members)  
model_weights/average_pooling2d_14_output_group/model_weights/average_pooling2d_14_output_group (8 members)  
model_weights/average_pooling2d_15_output_group/model_weights/average_pooling2d_15_output_group (8 members)  
model_weights/average_pooling2d_16_output_group/model_weights/average_pooling2d_16_output_group (8 members)  
model_weights/average_pooling2d_17_output_group/model_weights/average_pooling2d_17_output_group (8 members)  
model_weights/average_pooling2d_18_output_group/model_weights/average_pooling2d_18_output_group (8 members)  
model_weights/batch_normalization_106_output_group/model_weights/batch_normalization_106_output_group (8 members)  
model_weights/batch_normalization_107_output_group/model_weights/batch_normalization_107_output_group (8 members)  
model_weights/batch_normalization_108_output_group/model_weights/batch_normalization_108_output_group (8 members)  
model_weights/batch_normalization_109_output_group/model_weights/batch_normalization_109_output_group (8 members)  
model_weights/batch_normalization_110_output_group/model_weights/batch_normalization_110_output_group (8 members)  
model_weights/batch_normalization_111_output_group/model_weights/batch_normalization_111_output_group (8 members)  
model_weights/batch_normalization_112_output_group/model_weights/batch_normalization_112_output_group (8 members)  
model_weights/batch_normalization_113_output_group/model_weights/batch_normalization_113_output_group (8 members)
```

Sumber: Hasil Pengolahan Data
Gambar 5 Isi Model.h5

Pada Gambar 5, model terbaik selama pelatihan disimpan dalam file *model.h5* dengan menggunakan fitur *ModelCheckpoint*. Selain itu, *EarlyStopping* diterapkan untuk menghentikan pelatihan jika tidak terjadi peningkatan dalam kinerja model.

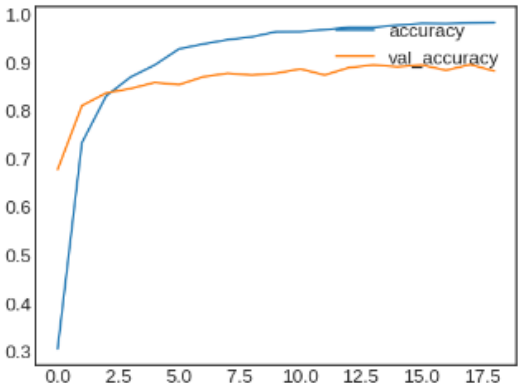
```
1. model_name = 'model.h5'  
2. checkpoint =  
ModelCheckpoint(model_name,  
3. monitor =  
'val_loss',  
4. mode =  
'min',  
5. save_best_o  
nly = True,  
6. verbose =  
1)  
7.  
8. earlystopping = EarlyStopping(monitor =  
'val_loss', min_delta = 0, patience =  
5, verbose = 1, restore_best_weights =  
True)
```

Train model menggunakan optimizer dari library keras SGD dengan *learning rate* 0.001, *Momentum* 0.9 dan *loss* menggunakan *categorical_crossentropy*.

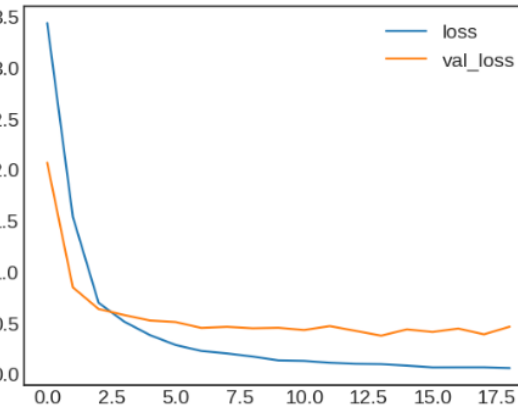
5. Penilaian Akurasi (Asses)

Model klasifikasi fauna yang telah dibuat sebelumnya dievaluasi untuk menilai kinerja dalam mendeteksi gambar. Evaluasi akhir mencakup pengukuran seperti akurasi, loss, presisi, dan recall. Berdasarkan Gambar 6 dan Gambar 7, hasil evaluasi akhir menunjukkan akurasi sebesar 98,36%, loss sebesar 0,06%, *val_accuracy* sebesar 89,83%, dan *val_loss* sebesar 38,84%.

Dari evaluasi tersebut menunjukkan bahwa 98.36% dari prediksi yang dibuat oleh model klasifikasi jenis fauna adalah benar dan nilai kesalahannya adalah 0,06%. Dan sebanyak 89,83% prediksi yang dibuat oleh model pada *dataset* validasi benar dan rata-rata kesalahan model pada *dataset* validasi adalah 38,84%.



Sumber: Hasil Pengolahan Data
Gambar 6. Hasil Evaluasi Akurasi Klasifikasi Jenis Fauna



Sumber: Hasil Pengolahan Data
Gambar 7. Hasil Evaluasi Loss Klasifikasi Jenis Fauna

Serta untuk hasil evaluasi akhir setiap kelas tersaji pada Tabel 1 yang menggambarkan *presicion*, *recall*, *F1-score* dari 50 kelas fauna.

Tabel 1 Hasil Evaluasi Akurasi dari Setiap Kelas Fauna

Jenis Fauna	Precision	Recall	F1-Score
Angsa	0.88	1.00	0.94
Anjing	0.95	0.91	0.93
Ayam	0.72	0.82	0.77
Babi	1.00	0.73	0.84
Badak	0.84	0.95	0.89
Bebek	0.85	0.55	0.67
Berang-berang	0.75	0.95	0.84
Beruang	0.88	1.00	0.94
Buaya	0.84	0.95	0.89
Burung hantu	0.96	1.00	0.98
Capung	0.92	1.00	0.96
Domba	0.71	1.00	0.83
Gajah	0.95	0.91	0.93
Gorila	0.88	0.95	0.91
Hamster	1.00	0.73	0.84
Harimau	1.00	1.00	1.00
Ikan	1.00	0.95	0.98
Jerapah	0.91	0.95	0.93
Kalajengking	0.85	1.00	0.92
Kambing	0.73	0.73	0.73
Kangguru	0.87	0.59	0.70
Kecoa	1.00	1.00	1.00
Keledai	0.83	0.86	0.84
kelelawar	0.95	0.95	0.95
Kelinci	0.84	0.84	0.84
Kepiting	0.95	0.95	0.95
Kodok	0.95	0.95	0.95
Kucing	0.95	0.91	0.93
Kuda	0.79	0.86	0.83
Kudanil	0.91	0.95	0.93
Kupu-kupu	1.00	0.91	0.95
Kura-kura	1.00	0.91	0.95
Laba-laba	1.00	0.86	0.93
Lebah	1.00	1.00	1.00
Lumba-lumba	0.87	0.91	0.89
Manusia	0.95	0.95	0.95
Merpati	0.82	0.82	0.82
Monyet	0.95	0.86	0.90
Orang utan	1.00	1.00	1.00
Panda	0.92	1.00	0.96
Penguin	0.96	1.00	0.98
Rusa	0.86	0.82	0.84
Sapi	0.83	0.68	0.75
serigala	0.86	0.82	0.84
Singa	0.95	0.86	0.90
Tikus	0.76	0.77	0.85
Tupai	0.94	0.77	0.85
Ular	0.91	0.95	0.93
Ulat	0.89	0.77	0.83
Zebra	1.00	1.00	1.00

Sumber: Hasil pengolahan data

Dari Tabel 1 dapat diketahui bahwa rata-rata akurasi dan adalah 0.9 yang menunjukkan model klasifikasi jenis fauna memiliki kinerja sangat baik yang berarti 90% dari prediksi positif yang dibuat model benar-benar relevan untuk masing-masing kelas yang dievaluasi.

6. Simulasi Model pada Aplikasi Web

Tujuan awal dari penelitian adalah untuk mempermudah anak-anak dalam mengenal jenis fauna, oleh karena itu hasil model klasifikasi fauna disimulasikan dalam sistem berbasis web. Aplikasi web ini dibuat dengan Bahasa Pemrograman HTML, CSS dan Python. Aplikasi web ini dibuat bertujuan mempermudah anak-anak belajar mengenal jenis fauna dengan mudah tanpa memakan memory pada perangkat yang digunakan.

Gambar 8 menunjukkan tampilan awal web berisi dua menu *Home* dan *Scan*.



Gambar 8. Tampilan Awal Aplikasi Web Klasifikasi Jenis Fauna

Fungsi menu *home* untuk menunjukkan halaman depan dan menu *scan* untuk melakukan deteksi gambar fauna yang diinginkan. Contoh hasil klasifikasi deteksi jenis fauna ada pada Gambar 9 yang menunjukkan deteksi gambar kucing dan sistem dapat menjawab dengan benar bahwa gambar tersebut adalah kucing.



Gambar 9 Hasil Deteksi Menggunakan Model Klasifikasi Jenis Fauna

Hasil implementasi pemodelan klasifikasi fauna ini dapat mudah digunakan oleh setiap orang yang memiliki akses ke *internet* untuk mendeteksi jenis fauna melalui kamera laptop maupun *smartphone*. Hal ini bisa menjadi alternatif media dalam mengenal fauna secara lebih interaktif.

IV. KESIMPULAN

Dengan menerapkan metode transfer learning GoogleNet, diperoleh model klasifikasi jenis fauna dengan akurasi tinggi, mencapai 98,36% dan loss

sebesar 0,06%, menggunakan inputan berukuran 32 batch size dengan dimensi gambar 224x224. Model klasifikasi ini kemudian diimplementasikan dalam aplikasi web untuk memudahkan anak-anak belajar mengenal fauna. Namun, mengingat banyaknya jenis fauna yang ada, menambah jumlah kelas fauna dalam model dapat meningkatkan keragaman dan karakteristik klasifikasi.

V. REFERENSI

- Alizah, M. D., Nugroho, A., Radiah, U., & Gata, W. (2020). Sentimen Analisis Terkait Lockdown pada Sosial Media Twitter. *Indonesian Journal on Software Engineering (IJSE)*, 6(2), 223–229. <https://doi.org/10.31294/ijse.v6i2.8991>
- Ardiansyah, A., & Hasan, N. F. (2023). Deteksi dan Klasifikasi Penyakit Pada Daun Kopi Menggunakan Yolov7. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 12(1), 30–35. <https://doi.org/10.32736/sisfokom.v12i1.1545>
- Babulal, K. S., & Das, A. K. (2022). Deep Learning-Based Object Detection: An Investigation. In: Singh, P.K., Wierzchoń, S.T., Chhabra, J.K., Tanwar, S. (eds) *Futuristic Trends in Networks and Computing Technologies*. In P. K. Singh (Ed.), *Lecture Notes in Electrical Engineering* (pp. 697–711). Springer. https://doi.org/https://doi.org/10.1007/978-981-19-5037-7_50
- Ifantiska, D. (2022). *Implementasi Arsitektur Googlenet Dan Xception Untuk Identifikasi Penyakit Pada Daun Tanaman Kelapa Sawit*.
- Inaya, A. N., Rahma, A. U., Jannah, M., Arafah, L. R. K., Ishak, L., & Edy, M. R. (2024). Klasifikasi Citra Dengan Pendekatan Transfer Learning Pada Gambar Fauna Terbang. *Jurnal MediaTIK*, 7(1), 85–89. <https://journal.unm.ac.id/index.php/MediaTIK/article/view/2785>
- Pahlevi, R. (2022). *Penetrasi Internet di Kalangan Remaja Tertinggi di Indonesia*. <https://databoks.katadata.co.id/datapublish/2022/06/10/penetrasi-internet-di-kalangan-remaja-tertinggi-di-indonesia>
- Palacios, H. J. G., Toledo, R. A. J., Pantoja, G. A. H., & Navarro, Á. A. M. (2017). A comparative between CRISP-DM and SEMMA through the construction of a MODIS repository for studies of land use and cover change. *Advances in Science, Technology and Engineering Systems*, 2(3), 598–604. <https://doi.org/10.25046/aj020376>
- Putriani, J. D., & Hudaidah, H. (2021). Penerapan Pendidikan Indonesia Di Era Revolusi Industri 4.0. *Edukatif: Jurnal Ilmu Pendidikan*, 3(3), 831–838. <https://edukatif.org/index.php/edukatif/article/view/407>
- Sabilla, I. A. (2020). *Arsitektur Convolutional Neural Network (Cnn) Untuk Klasifikasi Jenis Dan Kesegaran Buah Pada Neraca Buah*. In *Tesis*. https://repository.its.ac.id/73567/1/05111850010020-Master_Thesis.pdf
- Sanjaya, M. O., Bukhori, S., & Furqon, M. `Ariful. (2023). Virtual Assistant for Thesis Technical Guide Using Artificial Neural Network. *Indonesian Journal of Artificial Intelligence and Data Mining*, 6(2), 188. <https://doi.org/10.24014/ijaidm.v6i2.23473>
- Widiasanti, I., Ramadhan, N. A., Alfarizi, M., Fairus, N., Oktafiani, A. W., & Thahur, D. (2023). Pemanfaatan Sarana Multimedia dan Media Internet sebagai Alat Pembelajaran yang Efektif. *Edukatif: Jurnal Ilmu Pendidikan*, 5(3), 1355–1370. <https://doi.org/10.31004/edukatif.v5i3.4939>