

Implementasi Deep Learning untuk Image Clasification menggunakan Arsitektur Convolutional Neural Network (CNN) pada Citra Sampah Hotel (Studi Kasus: Hotel Lombok Raya)

(Implementation of Deep Learning for Image Classification using Convolutional Neural Network (CNN) Architecture on Hotel Waste Images (Case Study: Hotel Lombok Raya))

I Gusti Ngurah Agung Bayu Adhipramana^[1], Ario Yudo Husodo^[2], Arik Aranta^[3]

^[1] Program Studi Teknik Informatika, Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: bayuadhi2121@gmail.com, [ario, arikaranta]@unram.ac.id

Abstract Waste is solid waste consisting of organic and inorganic materials that are considered no longer useful and must be recycled or managed properly to avoid harm to the environment. One of the causes of various environmental problems related to waste is tourism. Mandalika is one of the five super-priority tourist destinations located in West Nusa Tenggara (NTB), Indonesia. The development of tourism in NTB has led to an increase in hotels in various places, including Mataram city. Hotel Lombok Raya is one such hotel that has seen a significant increase in visitors. This increase aligns with the growing pile of waste. The continuously increasing amount of waste requires proper management, as approximately 1.9 million tons of waste in NTB remains unhandled. The increasing amount of waste, without proper processing, can have negative impacts on the environment, such as damaging water and air quality and becoming a source of diseases. The waste sorting problem can be addressed through various innovations, such as utilizing deep learning models like Convolutional Neural Network (CNN) with multiple architectures for image recognition. The main objective of this research is to implement a modified VGG16 architecture to achieve better accuracy in classifying hotel waste into three classes: organic, inorganic, and B3 (hazardous) waste. The total dataset used for this research comprises 1300 images. Based on the research results, the best model obtained an accuracy of 97%, precision rate of 98%, recall rate of 98%, and an F1-score of 98%.

Keywords: Trash, Tourism, Classification, Deep learning, CNN, VGG16.

I. PENDAHULUAN

Sampah menurut Kamus Besar Bahasa Indonesia (KBBI) adalah barang yang dibuang oleh pemiliknya karena tidak terpakai lagi (KBBI, 2006). Sampah terbentuk sebagai output seluruh kegiatan manusia sehingga peningkatan sampah tidak terlepas dari

peningkatan populasi dan aktivitas manusia. Pariwisata merupakan salah satu kegiatan yang dapat memicu semakin meningkatnya populasi dan aktivitas manusia di suatu daerah. Nofriya, menyatakan bahwa berbagai permasalahan lingkungan ditimbulkan oleh pariwisata terutama terkait dengan timbulan sampah yang diperburuk dengan perilaku buruk wisatawan [1].

Kementerian Pariwisata dan Ekonomi Kreatif menyatakan bahwa mandalika yang terletak di provinsi NTB merupakan salah satu dari lima destinasi wisata super prioritas. Pada tahun 2022 tercatat sebanyak 1,2 juta wisatawan yang mengunjungi NTB. Perkembangan pariwisata di NTB tentunya akan sejalan dengan meningkatnya hotel di berbagai tempat, salah satunya kota mataram yang memiliki hotel terbanyak kedua di NTB. Berdasarkan hasil observasi salah satu hotel yang ada di Mataram adalah Hotel Lombok Raya yang terpantau mengalami kenaikan pengunjung secara signifikan sebesar 34,85% pada tahun 2018 dan pada tahun 2022 sebesar 53,32%, hal ini sejalan dengan tumpukan sampah yang dihasilkan dan tidak terkelola dengan baik. Tidak hanya dari jumlah tamu, seluruh kegiatan operasional juga memberikan sumbangan sampah sehingga hal ini memerlukan penanganan mengingat sekitar 1,9 juta ton sampah di NTB masih tidak tertangani. Peningkatan jumlah sampah yang tidak disertai dengan pengolahan tersebut dapat memberikan dampak negatif bagi lingkungan seperti merusak kualitas air, udara dan dapat menjadi sumber penyakit [2].

Salah satu cara yang dapat digunakan untuk mengurangi timbulan sampah adalah dengan melakukan klasifikasi karena setiap jenis sampah memiliki cara pengolahan yang berbeda [3]. Untuk mendukung terlaksananya klasifikasi sampah tersebut Dinas

Lingkungan Hidup dan Kehutanan (DISLHK) provinsi NTB telah menetapkan kebijakan pemilahan sampah yang tercantum dalam Surat Keputusan Kepala DISLHK Nomor: 660/530/PSPP-DislHK/2022 tentang ketentuan sampah terpilah minimal terdiri dari 2 jenis sampah yaitu organik dan anorganik (PPID DISLHK NTB, 2022). Maka dari itu, seluruh kegiatan yang mengharuskan pengangkutan sampah harus melalui pemilahan atau klasifikasi terlebih dahulu termasuk sampah dari hotel di kota mataram.

Cara konvensional yang biasa dilakukan untuk memisahkan sampah adalah dengan menyiapkan tempat sampah sesuai dengan jenis sampah sehingga jika sampah dipisahkan menjadi dua jenis yaitu organik dan anorganik maka harus disediakan tempat sampah organik dan anorganik di setiap kamar hotel. Metode tersebut tentunya akan memakan biaya untuk pengadaan tempat sampah di setiap ruangan atau kamar. Meski telah memakan biaya, pemilahan sampah dengan menggunakan metode tersebut belum tentu maksimal mengingat kesadaran membuang sampah yang masih cenderung rendah dilihat dari data Badan Pusat Statistik (BPS) 2018 menunjukkan bahwa 79% masyarakat Indonesia tidak peduli dengan pengolahan sampah [4]. Hal tersebut ditunjukkan dengan masih banyaknya masyarakat yang membuang sampah tidak sesuai dengan jenisnya meskipun telah disediakan tempat sampah dan sosialisasi [5]. Permasalahan pemilahan sampah tersebut dapat ditanggulangi dengan berbagai inovasi, yaitu dengan memanfaatkan perkembangan teknologi sebagai dampak dari revolusi industri 4.0.

Salah satunya dengan memanfaatkan metode *deep learning*. Perkembangan dalam metode *deep learning* yang sekarang banyak digunakan dalam training, terutama dalam dunia citra yaitu menggunakan transfer learning. *Transfer learning* memungkinkan pelatihan *deep learning* mendapatkan akurasi yang tinggi bahkan dengan menggunakan jumlah sample yang sedikit. Salah satu arsitektur yang sering kali digunakan dalam transfer learning adalah VGG-16. VGG 16 merupakan arsitektur *deep convolutional neural network* yang memiliki jumlah layer sejumlah 16 yang terdiri dari *convolution layer* dan *fully connected layer*. VGG16 juga merupakan arsitektur transfer learning paling awal yang berhasil melakukan klasifikasi citra imageNet dengan akurasi yang tinggi, selain itu arsitektur VGG16 merupakan penyempurna dari arsitektur AlexNet [6].

Adapun klasifikasi sampah pernah dilakukan pada penelitian sebelumnya dan mendapatkan akurasi yang cukup bagus yaitu diatas 90% dan kurang bagus (dibawah 80%) hal ini dikarenakan dari dataset yang digunakan, hasil yang cukup bagus didapatkan karna dataset diambil dari situs seperti kaggle, TACO, UAVVaste dan lain sebagainya, sedangkan untuk hasil yang kurang bagus didapatkan dari data yang diambil sendiri oleh peneliti [7]. Maka dari itu, dalam penelitian ini akan dilakukan pengklasifikasian sampah dengan data dari hotel Lombok raya yang terdiri dari 3 jenis yaitu organik, anorganik, dan

B3 menggunakan arsitektur VGG16. Penelitian ini diharapkan dapat mempermudah dalam pengklasifikasian sampah hotel, dan mendapatkan akurasi yang lebih baik. Selain itu penelitian ini diharapkan dapat menjadi acuan awal dalam pemilahan sampah dalam rangka mengurangi timbulan sampah yang tidak tertangani di NTB.

II. TINJAUAN PUSTAKA

Dilakukan penelitian sistem klasifikasi sampah anorganik kantor menggunakan *deep learning exception* berbasis nvidia *jetson nano* yang memiliki jumlah dataset 5.400 gambar selanjutnya di *training* sebanyak 320 *epoch* dan menghasilkan akurasi 91.67% [8].

Telah dilakukan analisis perbandingan algoritma *convolutional neural network* dan algoritma *multi-layer perceptron neural* dalam klasifikasi citra sampah dengan dataset sejumlah 1.933 gambar yang terdiri dari beberapa kelas. Dimana hasil *training* model *cnn* memiliki performa lebih baik dengan akurasi, presisi, *recall*, dan *f1-score* sebesar 0,98%. Model MLP hanya memiliki akurasi 0,43, presisi 0,41, *recall* 0,43, dan *f1-score* 0,39. Namun, CNN membutuhkan waktu lebih lama untuk pelatihan dan pengujian dibandingkan MLP [9].

Penelitian klasifikasi limbah menggunakan metode *convolutional neural network (cnn)* pada *web service* berbasis *framework flask*, penelitian ini menggunakan 9 macam *class* dataset dengan total 8.371 yang menghasilkan akurasi 69.77% dari hasil *training* 28 *epoch* [10].

Terdapat penelitian yang menggunakan metode *faster region based convolutional neural network* untuk klasifikasi sampah perkantoran, penelitian ini memiliki 453 dataset yang dibagi menjadi kertas, plastic dan kaleng. Dengan hasil akurasi berbeda disetiap *class* nya yaitu kertas sebesar 91,667%, botol *plastic* sebesar 91,667%, dan kaleng sebesar 83,33% [11].

Pendeteksian jenis-jenis sampah menggunakan metode CNN memiliki performa paling baik dalam memilah sampah dengan arsitektur yang digunakan yaitu VGG 16, Mobilenet VI, Inception V3, dan Resnet 50. Dataset yang digunakan yaitu 1.746 dengan 4 kelas itu plastic, besi, kaca, dan lainnya, dari hasil *training* didapatkan hasil akurasi sebesar 78% dan 90% [12].

Penelitian pengoptimasian *image classification* pada jenis sampah dengan data *augmentation* dan *convolutional neural network* tersebut menggunakan dataset yang berjumlah 1.190 data *train* dan 299 data *test* dengan nilai akurasi hasil *training* sebesar 90.97% [13].

Penelitian perbandingan *convolutional neural network* pada *transfer learning method* untuk mengklasifikasikan sel darah putih telah dilakukan. Dalam penelitian ini, menggunakan metode *transfer learning* dengan 4 arsitektur jaringan (AlexNet, GoogleNet, ResNet-50, VGG-16) untuk mengklasifikasikan citra sel darah putih. Dataset training terdiri dari 400 citra digital sel darah putih yang terbagi dalam 4 kelas (Eosinophil, Lymphocyte, Monocyte, Neutrophil). Hasil penelitian menunjukkan bahwa arsitektur ResNet-50 mencapai

akurasi tertinggi (94,17%), diikuti oleh GoogleNet, VGG-16, dan AlexNet.[14].

Berdasarkan penelitian-penelitian tersebut, dapat diketahui bahwa telah banyak dilakukan klasifikasi sampah menggunakan metode CNN dengan berbagai macam arsitektur salah satunya yaitu arsitektur VGG16. Arsitektur VGG16 memiliki tingkat akurasi yang baik untuk mengklasifikasikan berbagai data salah satunya citra sampah. oleh karena itu, penulis mengajukan penelitian klasifikasi sampah hotel menggunakan metode CNN dengan arsitektur VGG16.

A. Sampah

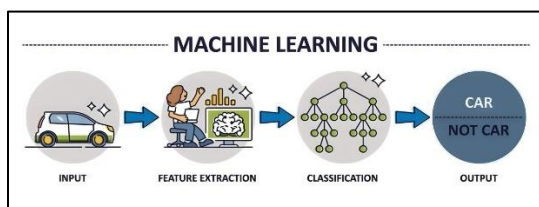
Sampah atau waste memiliki banyak pengertian dalam ilmu pengetahuan. Namun pada prinsipnya sampah adalah limbah yang bersifat padat yang terdiri dari bahan organik, anorganik, dan B3 yang sudah terbuang atau dibuang dari sumber hasil aktivitas manusia maupun alam dimana harus dikelola dengan sebaik mungkin agar tidak membahayakan lingkungan. Bentuk sampah bisa berada dalam setiap fase materi, yaitu padat, cair, dan gas [15]. Berikut beberapa jenis sampah yang digunakan dalam penelitian ini yaitu:

1. Sampah Organik
2. Sampah Anorganik
3. Sampah Bahan Berbahaya dan Beracun

B. Machine Learning

Machine learning adalah ilmu yang memungkinkan komputer berperilaku layaknya manusia, dimana komputer dapat meningkatkan pemahamannya dari pengalaman atau dengan berjalannya waktu secara otomatis (faggella, 2020). Kemampuan belajar diperoleh dengan cara mengakses data dan informasi secara terus-menerus.

Berdasarkan pengertian diatas, terdapat kata kunci yaitu, belajar, kecerdasan buatan, kode program tidak eksplisit, data masukan. Dengan kata kunci ini *machine learning* dapat di definisikan sebagai algoritma atau program komputer yang dapat membuat suatu sistem menjadi cerdas dengan mempelajari data-data yang ada dimana algoritma tersebut tidak didefinisikan secara eksplisit [16]. Proses *machine learning* agar dapat menghasilkan keluaran yang diinginkan dapat dilihat pada gambar berikut:



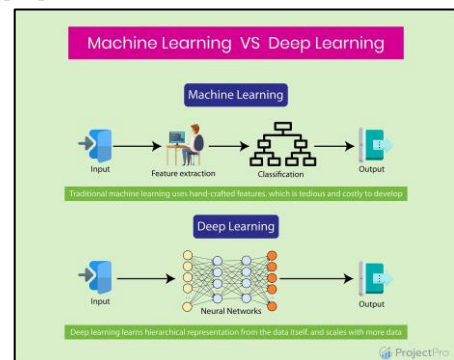
Gambar 1. Proses pada *machine learning* dari tahap masukan hingga keluaran

Pada Gambar 1 menunjukkan dalam *Machine Learning*, terdapat empat bagian utama yaitu *input*, *feature extraction*, *classification*, dan *output*. *Input* adalah data

yang akan digunakan dalam pembelajaran. *Feature extraction* menghasilkan fitur dari data. *Classification* menentukan klasifikasi data berdasarkan fitur yang telah diekstraksi. Metode pembelajaran *Machine Learning* dapat dibagi menjadi tiga kategori utama: *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning* [17].

C. Deep Learning

Deep learning adalah cabang dari *machine learning* yang menggunakan jaringan saraf (*neural network*) dengan banyak lapisan tersembunyi (*hidden layer*). Proses pembelajaran *deep learning* melibatkan banyak *layer*, dimulai dari fitur sederhana hingga fitur kompleks. *Deep learning* cocok untuk data yang tidak teratur seperti gambar, suara, dan teks. Perbedaannya dengan *machine learning* terletak pada ekstraksi fitur, di mana *machine learning* membutuhkan ekstraksi fitur manual, sedangkan *deep learning* dapat melakukan ekstraksi fitur dan klasifikasi secara otomatis. Gambar 2. Merupakan ilustrasi perbedaan antara *Machine Learning* dengan *Deep Learning* [18].



Gambar 2. Perbedaan *machine learning* dan *deep learning* dari segi ekstraksi fitur

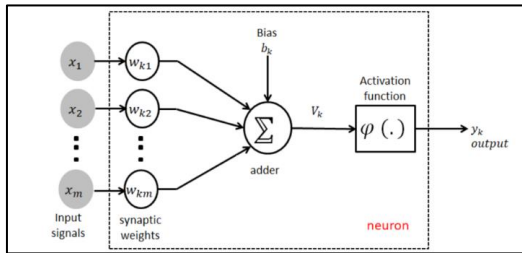
D. Data Augmentation

Data augmentation digunakan untuk memperkaya data dan mencegah *overfitting*. Metode yang umum digunakan termasuk *horizontal flip*, *shear range*, dan *zoom range*. *Horizontal flip* memutar gambar secara *horizontal*, *shear range* melakukan rotasi citra, dan *zoom range* memperbesar citra. Nilai *share range* dan *zoom range* adalah 0.2. Tujuan dari *data augmentation* di penelitian ini adalah untuk meningkatkan ukuran dataset pelatihan dan memperoleh variasi informasi yang lebih banyak [19].

E. Artificial Neural Network (ANN)

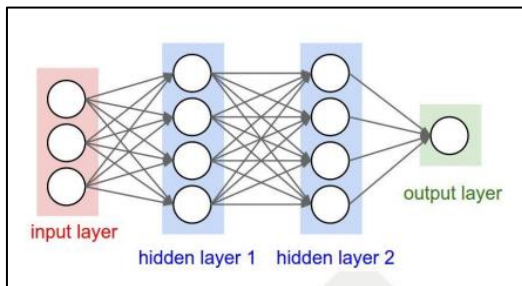
Artificial Neural Network (ANN) adalah teknik model matematika yang terinspirasi dari jaringan saraf biologis manusia. ANN menggunakan koneksi antar neuron buatan untuk memproses informasi dan memecahkan masalah seperti klasifikasi dan prediksi. ANN memiliki beberapa *layer*, termasuk *input layer*, *hidden layer*, dan *output layer*, dan setiap *layer* memiliki banyak *neuron* dengan bobot yang berbeda [20]. Neuron

dalam ANN dapat dicontohkan seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Neuron pada Artificial Neural Network.

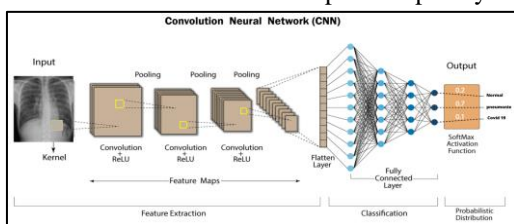
Adapun tiga bagian layer dalam ANN yang diatur dan saling terkait satu sama lain dapat dilihat pada Gambar 4.



Gambar 4. Lapisan (layer) pada ANN untuk multilayer Perceptron.

F. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan metode yang sukses dalam pengenalan pola, dikembangkan oleh Lecun pada tahun 1988. CNN memiliki arsitektur dengan puluhan hingga ratusan layer. Setiap layer melakukan pembelajaran dengan menggunakan output dari layer sebelumnya sebagai input. CNN menggunakan tahap pelatihan filter untuk mengekstrak fitur visual pada gambar. *Feature maps* kemudian dikurangi ukurannya melalui operasi pooling. Proses ini berulang hingga fitur terdalam diekstraksi. Selanjutnya, keputusan dibuat menggunakan classifier pada fitur ini. Struktur umum dari CNN melibatkan *convolutional layer*, *pooling layer*, dan *fully connected network*. *Convolutional layer* digunakan untuk ekstraksi fitur, sementara *fully connected layer* dapat menggunakan fungsi aktivasi softmax untuk klasifikasi multi kelas dan sigmoid untuk klasifikasi dua kelas pada output layer [18].

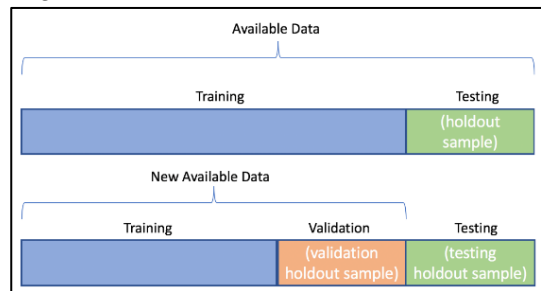


Gambar 5. Tahapan *deep learning* dengan arsitektur CNN.

Gambar 5. Merupakan tahapan dari arsitektur CNN. Yang dibagi menjadi 2 bagian yaitu *ekstraksi layer* dan *klasifikasi layer*.

G. Data splitting

Data splitting adalah metode untuk membagi data menjadi dua bagian: data *training* dan data *testing*. Data *training* digunakan untuk melatih model, sedangkan data *testing* digunakan untuk menguji model. Biasanya, persentase pembagian adalah 80% untuk data *training* dan 20% untuk data *testing*, tetapi juga dapat menggunakan pembagian seperti 70%:30% atau 50%:50%. Tujuannya adalah untuk memeriksa kinerja model akhir yang telah dilatih [21]. Berikut merupakan penerapan metode data *splitting*.



Gambar 6. Pembagian data dengan Data Splitting dengan tiga bagian.

H. Optimizer

Setiap model *deep learning* menggunakan algoritma optimasi untuk memetakan input ke output dan meminimalkan kesalahan. *Optimizer* berperan penting dalam meningkatkan akurasi dan kecepatan training model. Saat melatih model, bobot diperbarui setiap *epoch* untuk meminimalkan *loss function*. Pemilihan *optimizer* yang sesuai tergantung pada aplikasi yang digunakan [22].

I. Confusion Matrix

Confusion matrix digunakan untuk mengukur kinerja metode klasifikasi. Terdiri dari empat istilah: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Ini membandingkan hasil klasifikasi sistem dengan hasil yang seharusnya [23]. Evaluasi dan validasi hasil dihitung menggunakan rumus akurasi, *precision*, dan *recall* sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2-1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2-2)$$

$$Recall = \frac{TP}{TP+FN} \quad (2-3)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2-4)$$

J. Arsitektur CNN

Pengembangan arsitektur CNN yang optimal membutuhkan eksperimen berulang dan sumber daya komputasi yang signifikan. Beberapa perusahaan terkemuka, seperti Google dengan googlenet dan mobilenet, serta Facebook Research AI, dan perusahaan riset lainnya, terlibat dalam pengembangan model arsitektur. Pengembangan ini bertujuan untuk

memudahkan pengembang, peneliti, dan pihak lain dalam menerapkan *deep learning*.

III. METODE PENELITIAN

A. Alat

Dalam penelitian Tugas Akhir ini, diperlukan berbagai alat seperti perangkat keras dan perangkat lunak, serta data dan informasi yang relevan.

1. Perangkat Keras

Tabel 1. Perangkat Keras

No	Nama Perangkat	Spesifikasi
1	Processor	AMD Ryzen 7 6800H
2	Memory	16GB DDR5
3	GPU	NVIDIA RTX 3060

2. Perangkat Lunak

Tabel 2. Perangkat Lunak

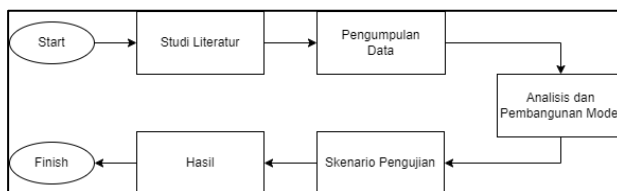
No	Nama Perangkat Lunak	Spesifikasi
1	Operating System	Windows 11 64-bit
2	Programming language	Python 3.8
3	Microsoft Office	Office 2021
4	Text Editor	Anaconda, vscode

B. Bahan Penelitian

Adapun sumber data sebagai bahan yang digunakan dalam penelitian Tugas Akhir ini adalah sebagai berikut.

- Berbagai literatur dari jurnal, buku elektronik, serta berbagai penulisan terkait yang berkaitan dengan klasifikasi sampah, *machine learning*, *neural network*, *activation function*, ANN, CNN, *convolution*, *dropout*, *pooling*, serta *optimizer*.
- Dataset yang digunakan pada penelitian tugas akhir ini terbagi menjadi 3 bagian, yaitu dataset latih dan uji. Dataset yang digunakan yaitu sampah hotel yang terbagi kedalam 3 jenis yakni organik, anorganik dan B3.

C. Diagram Alir Penelitian



Gambar 7. Diagram alir penelitian

Berdasarkan Gambar 7, terdapat beberapa tahapan yang dilakukan pada penelitian ini, tahap pertama dilakukan studi literatur, yaitu mempelajari sumber-sumber yang berkaitan dengan masalah yang diangkat sebagai dasar dalam melakukan penelitian. Kemudian tahapan selanjutnya yaitu pengumpulan *dataset* yang akan digunakan dalam pembuatan model, baik sebagai data latih maupun data uji. Setelah *dataset* didapatkan maka tahap selanjutnya yaitu pembangunan model menggunakan

metode CNN arsitektur VGG16. Setelah model dibuat, kemudian dilakukan pengujian terhadap model dengan menggunakan beberapa skenario pengujian. Tahap terakhir dilakukan analisis terhadap hasil berdasarkan skenario pengujian dan diambil kesimpulan penelitian.

D. Studi Literatur

Untuk mendukung penelitian, perlu dilakukannya studi literatur dengan mempelajari media-media seperti buku, jurnal penelitian, artikel, dan sumber lain yang berkaitan dengan permasalahan yang diangkat. Adapun materi yang dipelajari saat melakukan studi literatur ini berkaitan dengan penelitian-penelitian yang berhubungan dengan klasifikasi citra sampah, augmentasi, ANN, CNN beserta akurasi pada setiap arsitektur dan pengembangannya pada model, *activation function*, serta *confusion matrix*. Adapun studi literatur dilakukan dengan menonton video pembelajaran, membaca referensi melalui github, mencari jurnal-jurnal dan *e-book* melalui internet

E. Pengumpulan Data

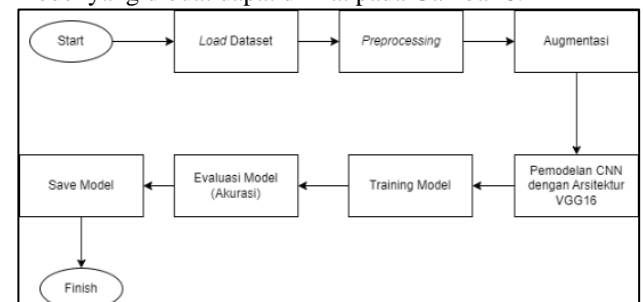
Data penelitian ini dikumpulkan dari hotel Lombok Raya di Mataram dengan bantuan *room boy*, dengan pengambilan data tiga kali dalam seminggu. Terdapat 600 gambar sampah yang terdiri dari 3 kelas: organik (200 citra), anorganik (200 citra), dan sampah B3 (200 citra), dengan latar belakang yang berbeda-beda. Contoh dataset dari setiap kelas dapat dilihat pada tabel 3:

Tabel 3. Contoh sampah hotel dari setiap class

Anorganik	Organik	B3

F. Analisis dan Pembangunan Model

Analisis dan pembangunan model merupakan langkah untuk memahami kinerja model yang akan dibangun dengan menganalisis kebutuhan secara intensif dan spesifik. Berikut merupakan gambar dari alur kerja model yang dibuat dapat dilihat pada Gambar 8.



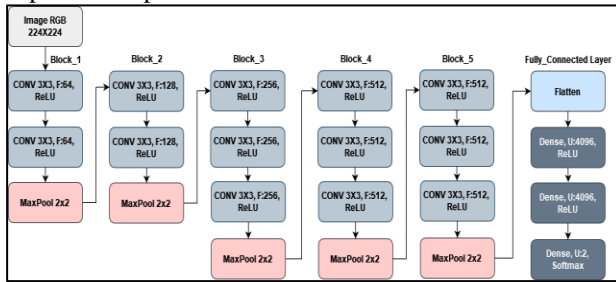
Gambar 8. Analisis dan pembangunan model

Dalam penelitian ini, langkah-langkah yang dilakukan adalah sebagai berikut:

1. *Load Dataset*: Dataset berupa gambar sampah dengan label 3 kelas (organik, anorganik, B3) di-*input*-kan.
2. *Preprocessing*: Melakukan penskalaan (*resize*), augmentasi, dan normalisasi pada data. Citra di-*resize* menjadi ukuran 224x224x3 *pixel*, tanpa konversi ke citra *grayscale*. Data dibagi menjadi 80% untuk *training* dan 20% untuk *validation*.
3. *Augmentasi*: Dilakukan menggunakan *TensorFlow* dan API *Keras* dengan modul *ImageDataGenerator*. Terdapat 7 jenis *augmentation*, termasuk *rescale* (1/255), *zoom range* (0.5), *rotation range* (45 derajat), *horizontal flip*, *vertical flip*, *shear range* (0.2), *brightness range* (0.1-0.2), *width shift range* (0.3), dan *height shift range* (0.3).

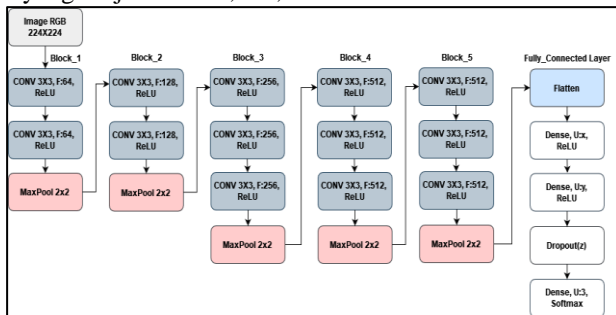
G. VGG16

Setelah dilakukannya *preprocessing* data, maka tahap selanjutnya yaitu melatih data menggunakan arsitektur VGG16 yang akan dimodifikasi pada layer *fully connected*, berikut dapat dilihat arsitektur dari VGG 16 dapat dilihat pada Gambar 9:



Gambar 9. Arsitektur VGG16

Berikut merupakan arsitektur dari VGG16 yang dimodifikasi dapat dilihat pada Gambar 3.5. Adapun bagian yang dimodifikasi terdapat pada *fully connected layer* tepatnya pada *dense layer* dengan mengubah jumlah *neuron* atau unit menjadi *x* dan *y*. Dimana nilai *x* yang akan digunakan berjumlah 4096, 1072, 1000 dan nilai *y* berjumlah 1072, 700, 512. Selain itu pada bagian *fully connected layer*, ditambahkan *layer dropout* dengan nilai *z* yang berjumlah 0.5, 0.3, dan 0.2.



Gambar 10. Modifikasi Arsitektur VGG16

Arsitektur VGG16 yang telah dimodifikasi digunakan dalam penelitian ini untuk melakukan klasifikasi jenis sampah. Arsitektur ini terdiri dari beberapa tahapan, yaitu:

1. Data gambar sampah organik, anorganik, dan B3 dengan ukuran 224x224 dan format RGB digunakan sebagai input.

2. *Block 1*: Terdiri dari 2 lapisan konvolusi 3x3 dengan 64 filter, diikuti oleh aktivasi ReLU, dan dilakukan *max pooling 2x2* dengan *stride 2*.
3. *Block 2*: Terdiri dari 2 lapisan konvolusi 3x3 dengan 128 filter, diikuti oleh aktivasi ReLU, dan dilakukan *max pooling 2x2* dengan *stride 2*.
4. *Block 3*: Terdiri dari 3 lapisan konvolusi 3x3 dengan 256 filter, diikuti oleh aktivasi ReLU, dan dilakukan *max pooling 2x2* dengan *stride 2*.
5. *Block 4*: Terdiri dari 3 lapisan konvolusi 3x3 dengan 512 filter, diikuti oleh aktivasi ReLU, dan dilakukan *max pooling 2x2* dengan *stride 2*.
6. *Block 5*: Terdiri dari 3 lapisan konvolusi 3x3 dengan 512 filter, diikuti oleh aktivasi ReLU, dan dilakukan *max pooling 2x2* dengan *stride 2*.
7. *Fully Connected Layer*: Keluaran dari *max pooling* terakhir dihubungkan ke dua lapisan *Dense layer* dengan jumlah *neuron* yang telah ditentukan, diikuti oleh aktivasi ReLU. Dilakukan pengurangan neuron menggunakan *dropout regularization*. Selanjutnya, dilakukan klasifikasi dengan fungsi aktivasi *softmax* untuk menentukan kelas sampah yang tepat.

H. Skenario Penelitian

1. Pengujian dilakukan dengan memvariasi jumlah *neuron* pada *dense layer* pertama dan kedua.
2. Pengujian dilakukan dengan memvariasi nilai *dropout* pada *hidden layer* menjadi 0.2, 0.3, dan 0.5.
3. Pengujian dilakukan dengan memvariasi nilai *batch size* menjadi 8, 16, dan 32.
4. Pengujian dilakukan dengan memvariasi jumlah *Epoch* dimana *Epoch* yang digunakan yaitu 25, 50, dan 100.
5. Pengujian dilakukan dengan menggunakan data yang telah diaugmentasi dan tidak diaugmentasi.
6. Pengujian dilakukan dengan memvariasi pembagian jumlah data untuk *testing* dan *training* dimana digunakan 8:1:1, 7:2:1, dan 6:3:1.
7. Pengujian dengan menggunakan jumlah data seimbang dan tidak pada data organik, anorganik, dan B3.

IV. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Pada penelitian ini, model menggunakan dataset yang dikumpulkan oleh peneliti dari Hotel Lombok Raya dengan bantuan *room boy*. Pengambilan gambar dilakukan 3 kali seminggu dengan variasi latar belakang. Jumlah total dataset saat ini mencapai 1300 gambar, terdiri dari 3 kelas (organik, anorganik, dan B3) dengan rata-rata 400 gambar per kelas.

B. Mekanisme Penelitian

Penelitian ini melibatkan beberapa tahapan untuk mendapatkan model terbaik. Tahap awal melibatkan

pemilihan citra dengan rata-rata setiap kelas berjumlah 400 citra dengan tujuan meratakan penyebaran citra. Selanjutnya, citra dibagi menjadi *training*, *validation*, dan *testing* dengan pembagian 80%:10%:10%. Dilakukan juga *pre-processing* dan augmentasi. Pengujian dilakukan dengan variasi jumlah *neuron* pada *dense layer*, nilai *dropout*, *batch size*, jumlah *epoch*, penggunaan data yang telah diaugmentasi atau tidak, pembagian jumlah data untuk *testing* dan *training*, serta keseimbangan data organik, anorganik, dan B3. Skenario pengujian dilakukan secara berurutan dengan menggunakan parameter terbaik dari skenario sebelumnya. Hasil terbaik ditentukan berdasarkan akurasi, presisi, dan *recall* terbaik. Pengujian dilakukan menggunakan Visual Studio Code.

C. Pre-processing

Dalam penelitian ini, dilakukan tahap *pre-processing* dengan melakukan *resize* citra menjadi ukuran 224 x 224 piksel menggunakan library TensorFlow. Tujuan *resize* citra adalah agar semua citra memiliki ukuran yang sama dan dapat diolah secara efisien oleh model.

D. Augmentasi

Selain *pre-processing* dataset, penelitian ini juga melakukan augmentasi data untuk meningkatkan keragaman dan menghindari *overfitting*. Pada penelitian ini augmentasi data yang dilakukan berbeda dari yang telah disampaikan pada Bab sebelumnya hal ini karena hasil akurasi yang didapatkan menggunakan semua augmentasi tersebut kurang baik, oleh karena itu peneliti hanya memilih beberapa teknik augmentasi seperti *zoom_range*, *rotation_range*, *horizontal_flip*, *vertical_flip*, dan *shear 50*. Adapun proses augmentasi menggunakan *ImageDataGenerator* dari library *tensorflow*.

E. Arsitektur CNN

Penelitian ini menggunakan arsitektur VGG16 yang terdiri dari 5 blok. Setiap blok memiliki berbagai layer konvolusi dengan jumlah filter yang berbeda-beda, mulai dari 64 hingga 512 filter. Ukuran *kernel* yang digunakan adalah 3x3 dengan 1 *stride*. *Zero padding* juga diterapkan agar ukuran citra tetap atau tidak berkurang saat melewati layer konvolusi. Dalam proses konvolusi, dilakukan normalisasi fitur dengan ReLU untuk menghindari nilai negatif pada citra. Setelah itu, dilakukan *pooling layer* dengan ukuran *kernel* 2x2 dan *stride* 2 untuk mengurangi ukuran fitur menjadi setengahnya. Hasil *pooling* terakhir diubah menjadi vektor 1 dimensi (*flatten*). Vektor ini masuk ke 2 *dense layer* dengan jumlah *neuron* yang berbeda. *Dense layer* pertama memiliki variasi *neuron* (4096, 1072, 1000) dan *dense layer* kedua memiliki variasi *neuron* (1072, 700, 512). Dilakukan *dropout* dengan variasi (20%, 30%, 50%) untuk mengurangi *overfitting*. Terakhir, dilakukan klasifikasi dengan fungsi aktivasi *softmax*.

F. Training

Pada penelitian ini, dilakukan *training* dengan 50 *epoch* dan *batch size* 16 untuk memperbarui bobot model. *Cost* dari bobot dihitung dan digunakan untuk evaluasi dan optimasi model. Dengan demikian, model mengalami peningkatan akurasi setiap *epoch*.

V. HASIL PENGUJIAN

A. Hasil Pengujian

Pada pengujian, parameter awal yang digunakan adalah jumlah *neuron* pada *dense layer* pertama dan kedua, yaitu 4096 dan 1072. *Dropout* yang digunakan adalah 0.5. Model terbaik ditentukan berdasarkan kombinasi performa akurasi, presisi, *recall*, dan *F1-score*.

1. Pengujian Terhadap Pengaruh Jumlah Neuron Pada Dense Layer

Pengujian ini menguji pengaruh jumlah *neuron* dalam *dense layer* terhadap performa model. Tiga variasi digunakan untuk setiap *dense layer*: 4096, 1072, dan 1000 untuk *dense layer* pertama, serta 1072, 700, dan 512 untuk *dense layer* kedua. Jumlah *neuron* yang lebih sedikit atau lebih banyak tidak selalu meningkatkan akurasi model.

Tabel 4. Hasil pengujian terhadap jumlah *neuron*

Jumlah Neuron 1	Jumlah Neuron 2	Akurasi Testing (%)	Precision (%)	Recall (%)	F1 Score (%)
4096	1072	92%	92%	90%	90%
	700	93%	90%	89%	89%
	512	93%	91%	90%	90%
1072	1072	92%	91%	90%	90%
	700	92%	89%	88%	88%
	512	93%	90%	89%	89%
1000	1072	92%	92%	91%	91%
	700	92%	91%	89%	89%
	512	90%	92%	91%	91%

Dalam pengujian, ditemukan performa terbaik dengan kombinasi 1000 *neuron* pada *dense layer* pertama dan 1072 *neuron* pada *dense layer* kedua. Hasil pengujian menunjukkan akurasi *testing* sebesar 92%, presisi 92%, *recall* 91%, dan *F1-score* 91%. Meskipun demikian, peningkatan atau penurunan jumlah *neuron* tidak selalu menghasilkan peningkatan akurasi. Namun kombinasi dari kedua *neuron* memiliki pengaruh terhadap total parameter, semakin kecil kombinasi yang digunakan semakin berkurang total parameter yang dihasilkan. Adapun hasil pengujian untuk parameter dapat dilihat pada Tabel 5 sebagai berikut:

Tabel 5. Hasil parameter dari kombinasi *neuron*

Jumlah Neuron 1	Jumlah Neuron 2	Total Parameter
4096	1072	121.874.435
	700	120.246.079
	512	119.476.343
1072	1072	42.763.571
	700	42.363.299

	512	42.161.011
1000	1072	40.879.979
	700	40.506.491
	512	40.317.739

Dalam Tabel 5, dapat dilihat bahwa kombinasi *neuron* 1000 dengan 512 menghasilkan total parameter yang paling kecil. Namun, perlu dicatat bahwa meskipun kombinasi ini memiliki total parameter yang lebih kecil, hasil akurasi *testing* dan presisinya lebih rendah sebesar 2% dibandingkan dengan kombinasi *neuron* 1000 dengan 1072. Kemudian dapat dilihat bahwa kombinasi *neuron* 1000 dengan 1072 memiliki total parameter yang hampir sama dengan total parameter kombinasi *neuron* 1000 dengan 512 yakni masih dikisaran 40 juta.

Dengan begitu peneliti akan mempertimbangkan total parameter dan hasil akurasi dalam memilih kombinasi *neuron* yang optimal. Dimana, meskipun kombinasi *neuron* 1000 dengan 512 memiliki total parameter yang lebih sedikit, hasil akurasi yang lebih rendah membuat peneliti memilih kombinasi *neuron* 1000 dengan 1072 yang memiliki total parameter yang cukup serupa, tetapi memberikan kinerja yang lebih baik. Sehingga pengujian selanjutnya akan menggunakan arsitektur dengan *neuron* berjumlah 1000 dan 1072 pada *dense layer* secara berurutan.

2. Pengujian Terhadap Pengaruh Bobot *Drop Out*

Dropout ditambahkan pada model untuk mengurangi *overfitting* pada data *testing*. Bobot *dropout* yang digunakan dalam penelitian ini adalah 0.2, 0.3, dan 0.5.

Tabel 6. Hasil pengujian terhadap pengaruh bobot *dropout*

Bobot Drop Out	Akurasi Train (%)	Akurasi Testing (%)	Precision (%)	Recall (%)	F1 Score (%)
0.2	92%	92%	90%	89%	89%
0.3	94%	91%	91%	90%	90%
0.5	91%	92%	92%	91%	91%

Hasil pengujian menunjukkan bahwa *dropout* dengan bobot 0.5 memiliki performa terbaik. Dalam pengujian tersebut, diperoleh akurasi sebesar 91%, presisi 92%, *recall* 91%, dan F1-score 91%. Penambahan bobot *dropout* dapat meningkatkan performa model. Namun untuk membuktikan apakah memang benar penambahan bobot *dropout* dapat meningkatkan performa model, maka dilakukan tambahkan pengujian dimana menggunakan bobot *dropout* 0.6 dan 0.7, berikut hasilnya dapat dilihat pada Tabel 7.

Tabel 7. Hasil pengujian tambahkan terhadap bobot *dropout*

Bobot Drop Out	Akurasi Train (%)	Akurasi Testing (%)
0.6	89%	89%
0.7	84%	86%

Berdasarkan hasil ini maka dapat diketahui bahwa semakin besar nilai bobot ternyata tidak meningkatkan performa model, hal ini dibuktikan ketika bobotnya ditambah menjadi 0.6 dan 0.7 hasil akurasi menjadi semakin kecil. Dengan begitu dapat dikatakan bahwa semakin besar nilai bobot *dropout* juga dapat menghilangkan sebagian besar informasi yang dapat menghambat kemampuan model untuk mempelajari pola-

pola yang penting. Dengan begitu dapat disimpulkan bahwa bobot *dropout* yang optimal didapatkan pada 0.5. Selanjutnya bobot *dropout* 0.5 akan digunakan dalam model untuk melihat pengaruh nilai *batch size* pada model yang memiliki performa terbaik.

3. Pengujian Terhadap Nilai *Batch Size*

Pengujian dilakukan pada nilai *batch size* (jumlah sampel data pelatihan per iterasi) dengan variasi 8, 16, dan 32 untuk menentukan performa terbaik dalam pelatihan model.

Tabel 8. Hasil pengujian terhadap nilai *Batch Size*

Batch Size	Akurasi Train (%)	Akurasi Testing (%)	Presisi (%)	Recall (%)	F1 Score (%)
8	90%	91%	90%	87%	87%
16	91%	92%	92%	91%	91%
32	89%	92%	90%	89%	89%

Berdasarkan pengujian pada Tabel 8, *batch size* 16 memberikan performa terbaik dibandingkan dengan *batch size* 8 dan 32. Dengan menggunakan *batch size* 16, model mencapai akurasi 91%, 1% lebih tinggi dari *batch size* 8, dan 2% lebih tinggi dari *batch size* 32. Pemilihan *batch size* yang tepat memiliki pengaruh signifikan terhadap performa model. Oleh karena itu, *batch size* 16 dipilih untuk pengujian lebih lanjut terkait jumlah *epoch* pada model dengan performa terbaik.

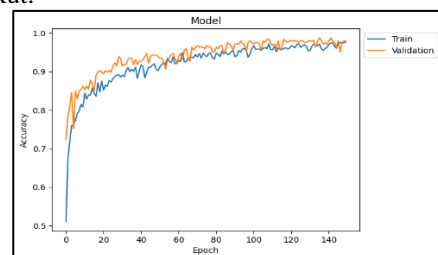
4. Pengujian Terhadap Pengaruh Jumlah *Epoch*

Pengujian *epoch* pada pelatihan model untuk mengetahui pengaruh jumlah *epoch* terhadap performa model. Terdapat 3 variasi jumlah *epoch* yang digunakan: 25, 50, dan 100. Tujuannya adalah untuk menemukan jumlah *epoch* yang memberikan performa terbaik pada model

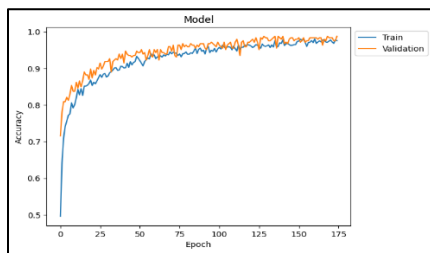
Tabel 9. Hasil pengujian terhadap jumlah *epoch*

Epoch	Akurasi Train (%)	Akurasi Testing (%)	Precision (%)	Recall (%)	F1 Score (%)
25	88%	88%	87%	84%	84%
50	91%	92%	92%	91%	91%
100	96%	97%	98%	98%	98%

Berdasarkan hasil pengujian pada Tabel 9, performa terbaik ditemukan pada *epoch* 100, dengan akurasi sebesar 88%. Penambahan jumlah *epoch* dari 25 ke 50 meningkatkan akurasi sebesar 3%, sedangkan penambahan dari 50 ke 100 meningkatkan akurasi sebesar 5%. Hasil ini menunjukkan bahwa peningkatan jumlah *epoch* saat pelatihan dapat meningkatkan akurasi dengan peningkatan yang tidak signifikan. Untuk membuktikan hal ini maka dilakukan kembali pengujian untuk jumlah *epoch* dimana menggunakan *epoch* sebesar 150 dan 175. Adapun hasil pengujiannya dapat dilihat pada Gambar 11 dan Gambar 12 berikut:



Gambar 11. Diagram *epoch* 150



Gambar 12. Diagram epoch 175

Berdasarkan Gambar 11 dan Gambar 12 diketahui bahwa hasil akurasi *training* dan *testing* mengalami kenaikan yakni hampir menyentuh 100%, namun dilihat dari epoch 150 dan 175 tidak terdapat perubahan pada peningkatan akurasi dengan begitu dapat disimpulkan epoch yang optimal pada penelitian ini didapat pada epoch ke 150.

5. Pengujian Terhadap Penggunaan Data Augmentasi

Pada pengujian ini, data augmentasi digunakan untuk menghasilkan variasi baru dalam dataset guna mengurangi *overfitting* dan memperkaya jumlah sampel. Tujuannya adalah untuk mengevaluasi pengaruh data yang telah diaugmentasi dibandingkan dengan data yang tidak diaugmentasi terhadap kinerja model.

Tabel 10. Hasil pengujian terhadap penggunaan data augmentasi

Augmentasi Data	Akurasi Train (%)	Akurasi Testing (%)	Presisi (%)	Recall (%)	F1 Score (%)
Augmentasi	96%	97%	98%	98%	98%
Tanpa augmentasi	100%	91%	90%	88%	88%

Hasil pengujian menunjukkan bahwa akurasi *training* tanpa data augmentasi mencapai 100%, tetapi akurasi *testing* hanya sebesar 91%, menunjukkan adanya *overfitting*. Selain itu, presisi, *recall*, dan F1-Score juga lebih rendah untuk data tanpa augmentasi dibandingkan dengan data yang telah diaugmentasi. Penggunaan augmentasi mengurangi selisih antara akurasi *training* dan *testing* menjadi 1%, sedangkan selisih pada data tanpa augmentasi adalah 9%. Penggunaan data augmentasi dapat mencegah *overfitting*. Selanjutnya, dilakukan percobaan untuk menguji pengaruh perbandingan data *split*.

6. Pengujian Terhadap Perbandingan Split Data

Pengujian data *split* dilakukan untuk evaluasi model. Dua perbandingan yang digunakan adalah 8:1:1 dan 7:2:1, yang menunjukkan pembagian dataset menjadi *training set*, *validation set*, dan *testing set*. Perbandingan 8:1:1 memiliki 80% *training set*, 10% *validation set*, dan 10% *testing set*, sementara 7:2:1 memiliki 70% *training set*, 20% *validation set*, dan 10% *testing set*, dan terakhir perbandingan 6:3:1. Pemilihan perbandingan data *split* yang optimal sangat penting karena dapat mempengaruhi performa model. Perbandingan yang tepat dapat membantu model untuk belajar dengan baik.

Tabel 11. Hasil pengujian terhadap Terhadap Perbandingan Split Data

Split Data	Akurasi Train (%)	Akurasi Testing (%)	Presisi (%)	Recall (%)	F1 Score (%)
8:1:1	96%	97%	98%	98%	98%
7:2:1	94%	91%	91%	89%	90%

6:3:1	94%	90%	90%	89%	90%
-------	-----	-----	-----	-----	-----

Berdasarkan hasil pengujian pada Tabel 11, dapat diketahui bahwa hasil terbaik didapatkan dari pembagian data sebesar 80% untuk *training*, 10% untuk *validation* dan 10% untuk *testing*. Dimana akurasi yang didapatkan sebesar 96%, lebih besar 2% dari *split* data yang lainnya. Presisi sebesar 98% dan *recall* sebesar 98%. Sedangkan untuk *split* data yang lainnya dimana ketika persentase data *training* berkurang 10% dan persentase data *validation* bertambah 10% akurasi dari model semakin berkurang. Pengurangan untuk akurasi *training* memang tidak terlalu besar namun perlu ditinjau juga dari persentase presisi, *recall*, dan F1-score dimana persentase presisi dengan perbandingan data (8:1:1) memiliki nilai lebih besar 7% dari perbandingan data (7:2:1) dan 8% lebih besar dari perbandingan (6:3:1). Sedangkan persentase untuk *recall* pada perbandingan data (8:1:1) memiliki nilai lebih besar 9% dari perbandingan data (7:2:1) dan (6:3:1). Dengan begitu pada penelitian ini *split* data yang optimal digunakan pada perbandingan data (8:1:1).

7. Pengujian Terhadap Data Imbalance pada setiap kelas

Pengujian data yang tidak seimbang (*imbalanced data*) adalah evaluasi performa model dengan dataset yang memiliki distribusi kelas yang tidak seimbang. Ketidakseimbangan tersebut dapat mempengaruhi performa model karena kecenderungan model untuk memprediksi kelas mayoritas dengan akurasi tinggi, sementara kelas minoritas mungkin tidak terdeteksi dengan baik. Hal ini dapat menyebabkan bias pada model dan kurangnya kemampuan dalam mengenali atau memprediksi kelas minoritas yang penting dalam data.

Tabel 10. Pengujian terhadap imbalance

Data Setiap Kelas	Akurasi Train (%)	Akurasi Testing (%)	Precision (%)	Recall (%)	F1 Score (%)
Balance	96%	97%	98%	98%	98%
Imbalance	90%	93%	92%	87%	88%

Data balance memiliki pengaruh positif terhadap akurasi atau performa model dengan peningkatan sebesar 6%. Akurasi yang dicapai adalah 96%, dengan presisi dan *recall* sebesar 98%. Setelah pengujian dengan 7 parameter yang berbeda, model terbaik telah ditemukan. Hasil *confusion matrix* dari performa terbaik yang didapatkan dapat dilihat pada Tabel 11 dan Tabel 12.

Tabel 11. Hasil Confusion Matrix Pengujian Model Terbaik

		Prediksi		
		B3	Organik	Anorganik
Aktual	B3	40	0	1
	Organik	0	47	0
	Anorganik	1	0	44

Tabel 12. Hasil Performa Pengujian Model Terbaik

Akurasi Testing (%)	Presisi (%)	Recall (%)	F1 Score (%)
97%	98%	98%	98%

Tabel 12 menunjukkan bahwa model terbaik berdasarkan parameter-parameter pengujian sebelumnya dapat melakukan klasifikasi citra dengan sangat baik dengan tingkat akurasi 97%, presisi 98%, *recall* 98%, dan F1-score sebesar 98%.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan terdapat beberapa hal yang dapat disimpulkan, antara lain sebagai berikut:

- 1) Model terbaik yang dihasilkan pada penelitian ini yaitu menggunakan kombinasi neuron pada *dense* layer sebesar 1000 dan 1072, *dropout* 0.5, *batch size* 32, jumlah *epoch* 100, dan perbandingan data *training* sebesar 80%, *validation* sebesar 10%, dan *test* sebesar 10%.
- 2) Pengujian menggunakan model terbaik yang didapatkan melalui beberapa tahap pengujian menghasilkan akurasi sebesar 96%, tingkat presisi sebesar 98%, dan tingkat *recall* sebesar 98%.
- 3) Penggunaan data *balance* untuk setiap kelas dapat memberikan akurasi yang lebih baik dari pada data *imbalanced* atau data tidak seimbang untuk setiap kelasnya, akurasi yang didapatkan sebesar 6% lebih baik dari pada data *imbalanced*.

B. Saran

Berdasarkan hasil analisis penelitian ini penulis mendapatkan pertimbangan pengembangan pada penelitian selanjutnya diantaranya:

- 1) Menambah variasi dataset untuk setiap kelas dan menambah scenario pengujian agar semakin variatif sehingga dapat meningkatkan performa model.
- 2) Menambah modifikasi pada arsitektur VGG16 untuk mengurangi jumlah parameter sehingga daya komputasi semakin rendah.
- 3) Melakukan beberapa percobaan kombinasi pada augmentasi data untuk mendapatkan hasil yang lebih baik.

REFERENCES

- [1] Issn et al., "Dampak: Jurnal Teknik Lingkungan Universitas Andalas Attribution-NonCommercial 4.0 International. Some rights reserved Artikel Penelitian," UNIVERSITAS ANDALAS, vol. 16, no. 2, pp. 86–94, 2019, doi: 10.25077/dampak.16.2.86-94.2019.
- [2] I Ketut Bagiastra, Si Luh Putu Damayanti, and I Ketut Purwata, "Kesadaran Pengelolaan Sampah Untuk Mendukung Pengembangan Hygiene Sanitasi Kawasancity Hotel Kota Mataram," Open Journal Systems, vol. 15, Aug. 2020.
- [3] U. Azmiyati, G. T. Rancak, W. J. Upaya, P. Pengelolaan, and S. Lambda, "Upaya Peningkatan Pengelolaan Sampah Berdasarkan Komposisi Sampah Di Kota Mataram," Jurnal Pendidikan MIPA dan Aplikasinya, vol. 2, no. 2, pp. 36–42, Aug. 2022.
- [4] Badan Pusat Statistik, "Laporan Indeks Perilaku Ketidakterpedulian Lingkungan Hidup Indonesia 2018," Sep. 2018.
- [5] P. Hariyono, S. Suciarto, V. Kusdiartini, and E. E. Listiati, "Kesadaran Pemilahan Sampah Rumah Tangga Pada Masyarakat Kota Semarang Dan Yogyakarta," SERI KAJIAN ILMIAH, vol. 15, no. 1, Jan. 2013.
- [6] K. Kualitas Buah Salak dengan Transfer Learning Arsitektur VGG and A. Luthfiarta, "VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification," Jurnal Informatika dan Teknologi Informasi, vol. 18, no. 1, pp. 37–48, 2021, doi: 10.31515/telematika.v18i1.4025.
- [7] H. Abdu and M. H. Mohd Noor, "A Survey on Waste Detection and Classification Using Deep Learning," IEEE Access, vol. 10, pp. 128151–128165, 2022, doi: 10.1109/ACCESS.2022.3226682.
- [8] R. Bijak, N. Kholis, and F. Utaminingrum, "Rancang Bangun Sistem Klasifikasi Sampah Anorganik Kantor menggunakan Deep Learning Arsitektur Xception berbasis NVIDIA Jetson Nano," vol. 6, no. 6, pp. 2681–2686, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [9] K. L. Kohsasih et al., "Analisis Perbandingan Algoritma Convolutional Neural Network Dan Algoritma Multi-Layer Perceptron Neural Dalam Klasifikasi Citra Sampah," no. 2, 2021, [Online]. Available: <http://ejournal.stmik-time.ac.id>
- [10] P. N. Dacipta and R. E. Putra, "Sistem Klasifikasi Limbah Menggunakan Metode Convolutional Neural Network (CNN) Pada Web Service Berbasis Framework Flask," Journal of Informatics and Computer Science, vol. 03, 2022.
- [11] D. M. Yafi and F. Utaminingrum, "Sistem Klasifikasi Sampah Perkantoran menggunakan Metode Faster Region Based Convolutional Neural Network berbasis NVIDIA Jetson Nano," vol. 6, no. 7, pp. 3122–3127, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [12] Stephen, Raymond, and Handri Santoso, "Aplikasi Convolution Neural Network Untuk Mendeteksi Jenis-Jenis Sampah," Jurnal Sistem Informasi dan Telematika, vol. 10, no. 2, 2019.
- [13] R. Permana, H. Saldy, and D. I. Maulana, "Optimasi Image Classification Pada Jenis Sampah Dengan Data Augmentation Dan Convolutional Neural Network," Jurnal Sistem Informasi dan Informatika (Simika), vol. 5, 2022.
- [14] D. M. Wonohadidjojo, "Perbandingan Convolutional Neural Network pada Transfer Learning Method untuk Mengklasifikasikan Sel Darah Putih," Ultimatics : Jurnal Teknik Informatika, vol. 13, no. 1, p. 51, 2021.
- [15] Tim Penulis PS, Penanganan dan Pengolahan Sampah. Niaga Swadaya, 2008.
- [16] Purba Daru Kusuma, Machine Learning Teori, Program, Dan Studi Kasus. Deepublish, 2020.
- [17] Dr. Yay Heryadi and Dr. Edi Irwansyah, Deep Learning dan Aplikasinya di Bidang Informasi Geospasial. PT. Artifisia Wahan Informa Teknologi, 2020.
- [18] Wahyudi Setiawan, Deep Learning Menggunakan Convolutional Neural Network Teori dan Aplikasi, vol. 1. Tim MNC Publishing, 2020.
- [19] M. Farid Naufal and S. Ferdiana Kusuma, "Pendeteksi Citra Masker Wajah Menggunakan Cnn Dan Transfer Learning," vol. 8, no. 6, pp. 1293–1300, 2021, doi: 10.25126/jtiik.202185201.
- [20] Eko Cahyono Putro and Rolly Maulana Awangga, Tutorial Gender Classification Using the You Look Only Once (Yolo), vol. 1. Kreatif, 2020.
- [21] Nisa Hanum Harani and Fikri Aldi Nugraha, Segmentasi Pelanggan Menggunakan Python. Bandung: Kreatif, 2020.

- [22] S. Kom. , M. K. Doni Putra Purbawa, S. Kom. , M. K. M. Syauqi Hanif Ardani, Ph. D. Prof. Drs. Ec. Ir. Riyanarto Sarno M. Sc., and S. K. Dr. Shoffi Izza Sabilla, Machine Learning dan Deep Learning-Konsep dan Pemrograman Python. Yogyakarta: Penerbit Andi, 2022.
- [23] Karsito and Santi Susanti, "Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naïve Bayes Di Perumahan Azzura Residencia," Jurnal Teknologi Pelita Bangsa, vol. 9, no. 3, 2019.