

Klasifikasi Cuaca Menggunakan Convolutional Neural Network

1st Muhammad Andriyansyah Malardy
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
andreansyahmalardy@gmail.com

2nd Rita Magdalena
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ritamagdalenat@telkomuniversity.ac.id

3rd Sofiah Saidah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sofiasaidahsf@telkomuniversity.ac.id

Abstrak — Cuaca merupakan salah satu faktor penting yang memengaruhi berbagai aktivitas manusia sehari-hari. Perubahan cuaca yang tidak menentu dapat berdampak pada sektor transportasi, pertanian, hingga penanggulangan bencana. Oleh karena itu, dibutuhkan sistem yang mampu mengklasifikasikan kondisi cuaca secara otomatis dan akurat. Salah satu solusinya adalah dengan memanfaatkan teknologi pengolahan citra dan kecerdasan buatan, khususnya menggunakan metode Convolutional Neural Network (CNN), yang mampu mengenali pola dalam gambar untuk menentukan jenis cuaca seperti cerah, berawan, atau hujan. Dalam penelitian ini, digunakan arsitektur CNN bernama MobileNetV2 yang dirancang untuk menghasilkan model yang ringan dan efisien, namun tetap memiliki tingkat akurasi yang tinggi. MobileNetV2 dipilih karena cocok digunakan dalam perangkat dengan kemampuan komputasi terbatas dan telah terbukti efektif dalam klasifikasi gambar. Proses yang dilakukan meliputi pengumpulan data dari situs Kaggle, pengolahan gambar agar memiliki ukuran seragam, pelatihan model menggunakan data latih, dan pengujian kinerja model untuk mengukur tingkat keberhasilannya dalam mengenali gambar cuaca. Dataset yang digunakan terdiri dari 768 gambar yang dibagi ke dalam tiga kategori: cerah (253 gambar), berawan (300 gambar), dan hujan (215 gambar). Sebanyak 80% data digunakan untuk pelatihan model, dan 20% sisanya untuk pengujian. Hasil terbaik diperoleh ketika menggunakan pengaturan: optimizer Stochastic Gradient Descent (SGD), learning rate 0,01, batch size 32, dan epoch 50. Dengan kombinasi tersebut, sistem berhasil mencapai akurasi sebesar 96,10%, dengan nilai presisi, recall, dan F1-score yang juga tinggi. Hal ini menunjukkan bahwa MobileNetV2 mampu memberikan hasil klasifikasi cuaca yang akurat dan efisien.

Kata kunci— Klasifikasi Cuaca, Convolutional Neural Network (CNN), MobileNetV2, Stochastic Gradient Descent (SGD), Learning Rate, Batch Size.

I. PENDAHULUAN

Cuaca merupakan kondisi pada atmosfer yang terjadi karena berbagai faktor, antara lain radiasi matahari, suhu udara, kecepatan dan arah angin. Jenis-jenis cuaca meliputi cuaca cerah, berawan, panas, dingin, berawan, dan hujan [1]. Cuaca sangat memengaruhi aktivitas kehidupan manusia. Klasifikasi unsur-unsur cuaca dapat berfungsi sebagai pedoman untuk keadaan cuaca di suatu lokasi. Klasifikasi unsur cuaca sering digunakan dalam peramalan cuaca dan pedoman deep learning dalam proses pengambilan keputusan. Sistem komputer memiliki kemampuan untuk mengamati kondisi cuaca dan menilainya berdasarkan kondisi cuaca yang sedang terjadi. Oleh karena itu, untuk memberikan hasil yang lebih akurat dan mengurangi kesalahan pengamat cuaca, sistem

harus mengklasifikasikan cuaca menurut kategorinya. Perubahan cuaca bergantung pada banyak faktor seperti suhu, kelembaban, angin, waktu, lokasi dan lainnya. Dari faktor ini dapat diperoleh jenis cuaca seperti cuaca cerah, berawan, hujan, hujan petir. Cuaca ekstrim berpotensi menjadi bencana seperti banjir, tanah longsor, kebakaran, penyebaran penyakit yang mempengaruhi daya tahan tubuh manusia. Dengan teknik klasifikasi cuaca yang baik dapat diprediksi kemungkinan perubahan cuaca yang terjadi dengan lebih akurat [2]. Pada tahun 2020, penelitian klasifikasi curah hujan dilakukan menggunakan metode Convolutional Neural Network (CNN) dengan arsitektur VGG16Net. Penelitian ini menggunakan 4 skenario pengujian. Hasil dari performa sistem yang diperoleh pengklasifikasi cuaca tidak mengalami overfitting maupun overshooting atau sistem mampu mengenali kondisi dari data citra latih dan data citra validasi dengan akurasi sebesar 87% [1]. Pada tahun yang sama di tahun 2020, penelitian klasifikasi untuk prediksi cuaca menggunakan metode Ensemble Learning. Penelitian ini menggunakan dataset klasifikasi cuaca di Australia yang terdiri dari 142.193 record dan 24 atribut, dengan data selama 10 tahun, hasil akurasi dan MSE dari metode ensemble learning adalah 81.21% sebagai akurasi dan 18.79% untuk 1 MSE [3]. Pada tahun 2022, penelitian klasifikasi cuaca provinsi DKI JAKARTA menggunakan algoritma Random Forest dengan teknik oversampling, eksperimen diperoleh akurasi model Random Forest mencapai 70% dengan teknik SMOTE. Hasil klasifikasi juga mampu memperbaiki prediksi setiap kelas minoritas 1, 2, 5, 6 dan 7 dengan rata-rata kenaikan prediksi benar mencapai 50% [2]. Pada penelitian ini terdapat perbedaan yaitu menggunakan Arsitektur MobileNetV2, serta Kaggle sebagai dataset yang akan digunakan. Salah satu metode yang populer saat ini adalah MobileNet. MobileNet memiliki kapabilitas yang terbaik untuk mengenali objek dengan akurasi yang tinggi dan kecepatan waktu deteksi [4].

II. KAJIAN TEORI

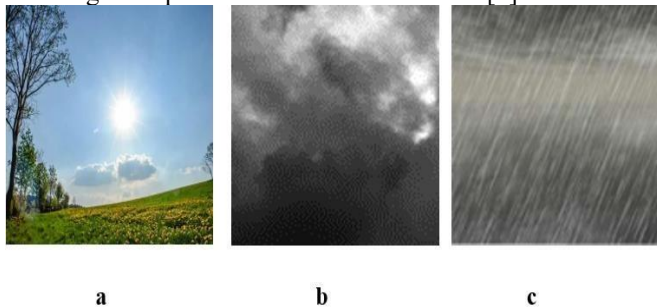
Menyajikan dan menjelaskan teori-teori yang berkaitan dengan variabel-variabel penelitian. Poin subjudul ditulis dalam abjad.

A. Meteorologi

Meteorologi adalah ilmu yang mempelajari cuaca di atmosfer bumi. Atmosfer adalah selubung gas yang menyelimuti bumi. Sedangkan cuaca adalah keadaan atmosfer pada saat yang pendek dan di tempat tertentu. Keadaan atmosfer gabungan dari berbagai unsur, antara lain suhu udara, kelembapan, kecepatan angin, dan presipitasi. Meteorologi membahas terjadinya perubahan dan gejala cuaca setiap saat menggunakan metode dan hukum-hukum fisika yang terjadi [5].

B. Cuaca

Cuaca adalah seluruh kejadian di atmosfer bumi yang merupakan bagian kehidupan sehari-hari manusia di dunia. Cuaca merupakan keadaan yang terjadi di permukaan bumi yang dipengaruhi oleh kondisi udara, yaitu tekanan dan suhu udara di langit maupun di bumi. Cuaca di setiap planet berbeda-beda tergantung pada jaraknya dari matahari dan pergerakan gas di setiap atmosfer planet-planet tersebut. Bahkan dalam suatu wilayah atau suatu perkotaan mempunyai jenis-jenis cuaca yang berbeda-beda dari daerah di sekelilingnya [5]. Cuaca yang terjadi di pagi hari yang cerah, radiasi matahari dengan cepat memanaskan daratan bumi kemudian memanaskan udara. Sedangkan cuaca yang terjadi di malam hari, tidak ada radiasi matahari sehingga terjadi pelepasan panas dari bumi yang besar sehingga mendinginkan permukaan daratan dan udara [1].



Gambar 1 Data set cuaca (a) Cerah, (b) Berawan, (c) Hujan

C. Citra

Pengertian citra secara umum adalah suatu gambar, foto ataupun berbagai tampilan dua dimensi yang menggambarkan suatu visualisasi objek. Citra dapat diwujudkan dalam bentuk tercetak ataupun digital. Citra digital adalah larik angka-angka secara dua dimensional. Citra digital tersimpan dalam suatu bentuk larik (array) angka digital yang merupakan hasil kuantifikasi dari tingkat kecerahan masing-masing piksel penyusun citra tersebut. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (continue) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, scanner dan lain sebagainya sehingga bayangan objek yang disebut citra tersebut terekam [6].

D. Citra RGB

RGB adalah suatu model warna yang terdiri atas 3 buah warna: merah (red), hijau (green), dan biru (blue), yang ditambahkan dengan berbagai cara untuk menghasilkan bermacam-macam warna [6].



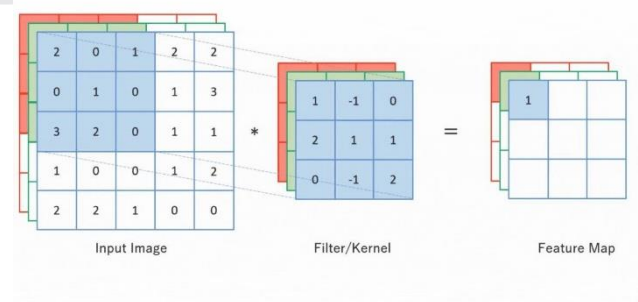
Gambar 2 Cuaca RGB

E. Convolutional Neural Network (CNN)

Convolutional Neural Network merupakan metode deep learning yang merupakan pengembangan dari Multilayer Perceptron (MLP) yang dapat mengolah data dua dimensi, seperti mengenali prediktif dari suatu objek gambar, teks, potongan suara. CNN terbagi menjadi dua bagian besar yaitu Feature Extraction dan Classification. Pada Feature Extraction terbagi atas Convolution dan Pooling. Pada proses CNN input citra diubah menjadi features berupa angka yang mempresentasikan suatu citra tersebut. Terdapat tiga layer pada CNN, yaitu Convolutional Layer, Pooling Layer, dan Fully Connected Layer. Arsitektur CNN terdiri dari beberapa jenis lapisan, yang mencakup lapisan konvolusi, lapisan aktivasi, dan lapisan pooling. Lapisan konvolusi memiliki fungsi utama dalam jaringan ini, bertanggung jawab untuk mengekstrak fitur dari gambar melalui teknik konvolusi. Filter yang digunakan dalam lapisan ini bergerak di atas gambar untuk mengidentifikasi pola visual seperti tepi, sudut, dan tekstur. Sementara itu, lapisan aktivasi berfungsi untuk menggabungkan informasi dengan menggunakan fungsi aktivasi seperti Rectified Linear Unit (ReLU), yang menambahkan unsur non-linearitas ke dalam model. Kemudian, lapisan pooling mengurangi dimensi spasial data dengan menurunkan resolusi gambar, sehingga memudahkan dalam pembelajaran fitur yang lebih tinggi dan menyederhanakan proses perhitungan [7].

F. Convolution Layer

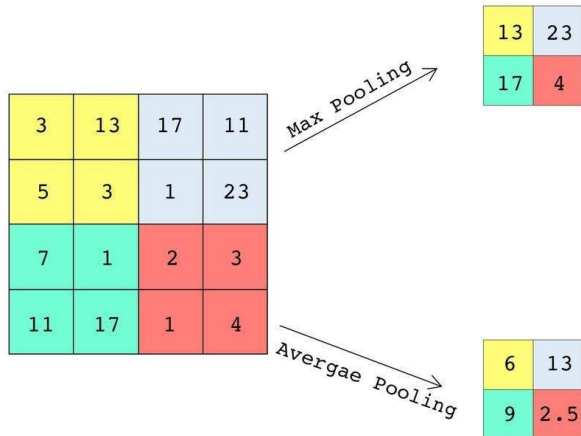
Convolutional layer adalah lapisan pertama pada CNN. Lapisan ini akan melakukan konvolusi citra masukan dengan *filter* yang telah didefinisikan tanpa merusak struktur citra awal. Fungsi lapisan ini yaitu mengambil fitur pada citra yang akan digunakan untuk melatih model.



Gambar 3 Contoh Convolutional Layer

G. Pooling layer

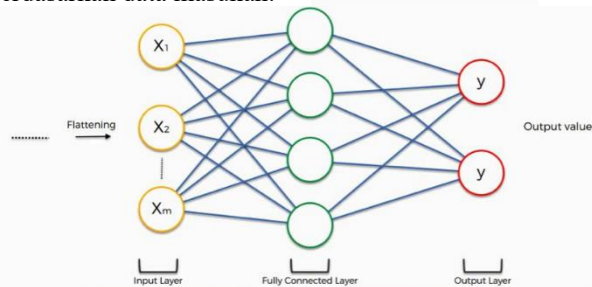
Pooling layer merupakan proses mengurangi jumlah parameter dan jumlah perhitungan dalam jaringan serta melakukan pencegahan *overfitting* pada citra. *Layer* ini terbagi menjadi *average pooling* dan *max pooling*. *Average pooling* yaitu mengambil nilai rata-rata dari area yang dipilih, sedangkan *max pooling* yaitu mengambil nilai terbesar dari area yang dipilih.



Gambar 4 Contoh *Max Pooling Layer* dan *Average Pooling Layer*

H. Fully Connected Layer

Fully connected layer adalah lapisan pada dimensi seluruh data akan diubah menjadi satu dimensi. Proses perubahan ukuran dimensi disebut *flatten*. Lapisan ini memiliki node yang saling berhubungan, memiliki bobot, dan memiliki fungsi aktivasi. Lapisan ini mengeluarkan sebuah prediksi berdasarkan data masukan.



Gambar 5 Ilustrasi Proses *Fully Connected Layer*

I. Rectified Linear Unit (RELU)

Rectified Linear Unit (RELU) adalah fungsi aktivasi pada CNN yang memiliki kelebihan mampu memproses data ukuran besar dengan cepat, yang digunakan di antara convolutional layer dan pooling layer. RELU menjaga hasil citra konvolusi pada domain definit positif, sehingga setiap nilai negatif yang berasal dari proses konvolusi akan melalui proses RELU, dan menjadikan nilai negatif tersebut sama dengan 0. Persamaan RELU ditunjukkan pada Persamaan 1

$$f(x) = \begin{cases} x, & \text{jika } x > 0 \\ 0, & \text{jika } x \leq 0 \end{cases} \quad (1)$$

J. Softmax

Softmax adalah fungsi aktivasi yang sering digunakan pada neural network yang memiliki banyak kategori output (multi-class). Fungsi ini mengubah suatu nilai angka menjadi nilai probability dengan skala yang proposional. Fungsi softmax mengubah nilai perhitungan menjadi nilai probability, hal ini menjadikan nilai hasil perhitungan bisa dibandingkan.

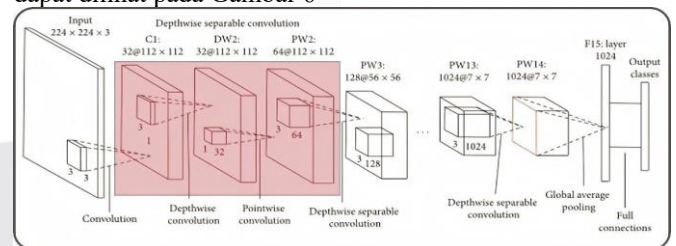
Dengan menggunakan fungsi softmax dapat terlihat kelas mana yang memiliki nilai kemungkinan terbesar. Kemudian, kemungkinan terbesar tersebut akan menjadi kelas terpilih dan masukan selanjutnya akan terklasifikasi menjadi kelas tersebut. Persamaan fungsi softmax ditunjukkan pada Persamaan 2

$$f_i(x) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}} \quad (2)$$

K. MOBILENETV2

MobileNet merupakan model dengan latensi rendah dan konsumsi daya yang minim, diadaptasi untuk memenuhi pembatasan sumber daya dalam berbagai skenario penggunaan. Menurut laporan penelitian, MobileNet V2 berhasil meningkatkan kinerja model seluler yang canggih dalam berbagai tugas dan uji coba, mencakup berbagai ukuran model. Sebagai ekstraktor fitur yang efektif untuk deteksi dan segmentasi objek, MobileNet V2 menunjukkan keunggulan, seperti dalam deteksi, di mana, ketika digunakan bersama dengan Single Shot Detector Lite, MobileNet V2 mencapai kecepatan sekitar 35 persen lebih tinggi dengan akurasi yang setara dengan MobileNet V1 [8].

Arsitektur MobileNet V2 terdiri dari blok dasar dengan konvolusi dan residu yang dapat dipisahkan dalam kedalaman. Struktur ini terdiri dari lapisan konvolusi awal dengan 32 filter, diikuti oleh 19 lapisan bottleneck residual. Dengan mengatur hyperparameter seperti resolusi gambar input dan penganda lebar, arsitektur ini telah disesuaikan untuk berbagai tingkat kinerja. Jaringan utama, yang memiliki resolusi $1,224 \times 224$, menggunakan 3,4 juta parameter dan membutuhkan biaya komputasi sekitar 300 juta perkalian tambahan. Biaya komputasi jaringan antara 7 dan 585 juta perkalian tambahan, dan ukuran model antara 1,7 dan 6,9 juta parameter. Contoh arsitektur MobileNet dapat dilihat pada Gambar 6



Gambar 6 MobileNet

L. Perbedaan MobileNetV1 dan V2

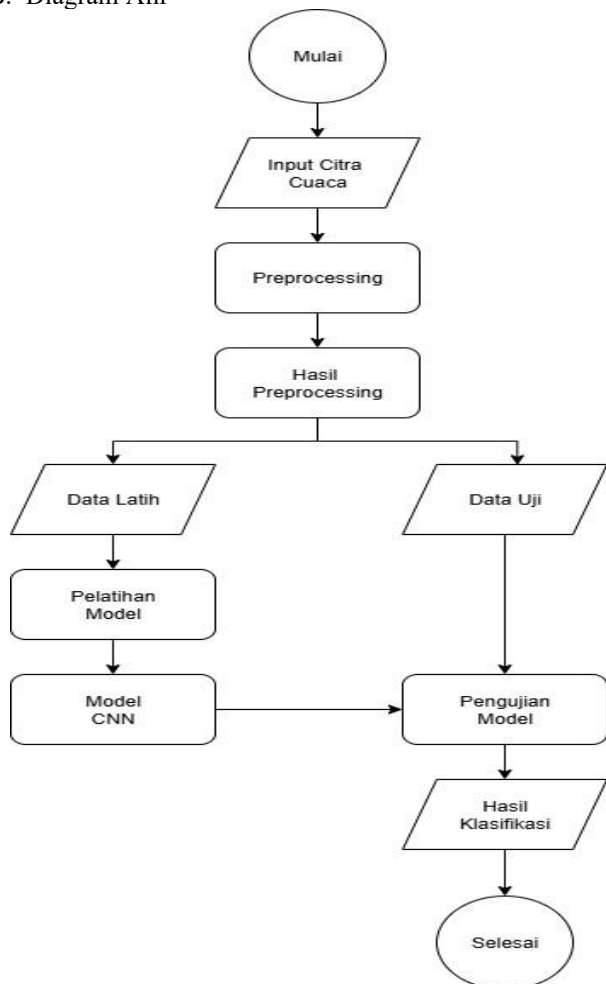
Perbedaan antara Model MobileNet V1 dan MobileNet V2 mencakup algoritma dan struktur masing-masing model. MobileNet V1 memiliki satu lapisan, sementara MobileNet V2 memiliki dua lapisan, menyebabkan waktu proses MobileNet V2 lebih cepat dibandingkan dengan MobileNet V1. MobileNet V2 menggunakan operasi dua kali lebih sedikit, memberikan akurasi yang lebih tinggi, membutuhkan parameter 30 persen lebih sedikit, dan mencapai kecepatan sekitar 30-40 persen lebih cepat (MobileNet V2) [8].

III. METODE

A. Konsep

Proses klasifikasi gambar cuaca akan menggunakan metode *CNN*, dan sistem pengolahan gambar digital akan menggunakan metode *Convolutional Neural Network* dengan arsitektur *MOBILENETV2*. Pada penelitian ini, klasifikasi cuaca dibagi ke dalam tiga kelas: berawan, hujan, dan cerah.

B. Diagram Alir



Gambar 7 Diagram Alir

C. Akuisisi Data

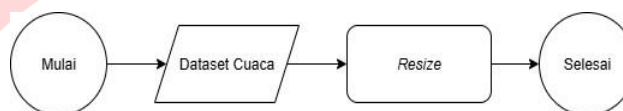
Akuisisi data adalah proses mengumpulkan dan menyimpan informasi dari berbagai sumber, salah satu nya melalui situs [Kaggle.com](https://www.kaggle.com/pratik2901/multiclassweather-dataset). Identifikasi sumber data, pengumpulan, penyimpanan, pemrosesan, dan pembersihan data, serta pengelolaan data adalah langkah-langkah umum dalam akuisisi data. Dataset citra yang digunakan berjudul Multi-class Weather Dataset, dibuat oleh Prateek. Dataset tersebut dapat diakses pada link <https://www.kaggle.com/pratik2901/multiclassweather-dataset>. Akuisisi data sangat penting dalam berbagai bidang, termasuk ilmu pengetahuan, teknik, bisnis, dan teknologi informasi, karena memungkinkan pengambilan keputusan yang akurat dan relevan. Terdapat 3 kelas cuaca, yaitu Cerah, Berawan, dan Hujan. Tabel 1 menunjukkan jumlah dataset dari setiap kelas cuaca. Dataset yang digunakan diresize menjadi ukuran 224 x 224 sebelum dilakukan training dan testing. Dimana untuk data training model yang dipakai 80% dan data testing 20%.

Tabel 1 Detail jumlah dataset dari setiap kelas cuaca

Kelas	Jumlah
Hujan	215
Berawan	300
Cerah	253

D. Preprocessing

Pada tahap *preprocessing* adalah langkah awal yang sangat penting untuk mempersiapkan citra daun kopi sebelum masuk ke tahapan selanjutnya. Pada tahap ini, citra cuaca yang akan digunakan dimasukkan sebagai data awal. Langkah pertama adalah mengubah ukuran citra (*resize*) 224 x 224 agar memiliki dimensi yang seragam. Proses ini dilakukan untuk memastikan semua gambar memiliki ukuran yang sama sehingga lebih mudah diproses oleh sistem, serta mengurangi risiko kesalahan akibat perbedaan ukuran citra.



Gambar 8 Tahap *Preprocessing*

E. Model Pelatihan dan Pengujian

Pelatihan dan pengujian model dilakukan agar sistem mampu mengenali pola dalam data dan membuat prediksi yang akurat. Tahap pelatihan dirancang untuk mengajarkan model memahami ciri khas dari setiap data yang diberikan, sedangkan pengujian bertujuan untuk memastikan model dapat bekerja dengan baik ketika dihadapkan pada data baru. Dengan adanya kedua tahap ini, model diharapkan tidak hanya memberikan performa yang baik pada data pelatihan tetapi juga dapat diandalkan dalam situasi nyata.

Pada proses pelatihan, model *MobileNetV2* dilatih menggunakan dataset yang telah diproses sebelumnya. Dataset ini dibagi menjadi data pelatihan dan pengujian untuk memastikan pembagian yang seimbang. Selama pelatihan, model belajar mengenali ciri-ciri spesifik seperti pola awan guna mengklasifikasikan cuaca menjadi cerah, berawan, atau hujan. Setelah pelatihan selesai, tahap pengujian dilakukan untuk mengevaluasi performa model menggunakan matrix seperti akurasi, presisi, recall, dan *F1-score*. Selain itu, confusion matrix juga digunakan untuk melihat kesalahan klasifikasi yang terjadi. Hasil dari pengujian ini menjadi penilaian akhir terhadap kemampuan model dalam mengklasifikasikan kondisi cuaca.

F. Parameter Performa

Untuk mengevaluasi kinerja model yang telah dilatih, dilakukan pengukuran dengan menggunakan beberapa parameter performa. Parameter ini memberikan gambaran tentang kemampuan model dalam melakukan klasifikasi secara akurat dan efektif. Lima parameter utama yang

digunakan adalah *Confusion Matriks*, Akurasi, Loss, Presisi, Recall.

G. Confusion Matriks

Confusion matrix atau disebut juga error matrix merupakan sebuah matriks yang memberikan informasi perbandingan hasil klasifikasi oleh model dengan hasil aktual. Confusion matrix memiliki tabel dengan 4 kombinasi yang berbeda dari hasil klasifikasi model dan hasil aktual.

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada confusion matrix yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN).

1. True Positive (TP): data positif yang diprediksi benar.
2. True Negative (TN): data negatif yang diprediksi benar.
3. False Negative (FN): data positif namun diprediksi sebagai data negatif.
4. False Positive (FP): data negatif namun diprediksi sebagai data positif

H. Akurasi

Akurasi adalah parameter yang digunakan untuk mengetahui seberapa akurat model mendeteksi objek dengan benar. Akurasi dapat dihitung dengan persamaan 3

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \quad (3)$$

Dimana TP singkatan dari *True Positive*, TN singkatan dari *True Negative*, FP singkatan dari *False Positive*, dan FN singkatan dari *False Negative*.

I. Loss

Loss mengukur rata-rata kesalahan model selama pelatihan atau pengujian. Loss digunakan untuk mengetahui seberapa jauh prediksi model dari nilai aktual. Nilai loss yang lebih rendah menunjukkan performa model yang lebih baik.

Rumus loss tergantung pada fungsi loss yang digunakan, seperti categorical cross-entropy untuk klasifikasi multi-kelas:

$$(p, q) = -\frac{1}{N} \sum_{i=1}^N \sum_k p_{i,k} \log q_{i,k} \quad (4)$$

J. Presisi

Presisi adalah perbandingan antara *True Positive* dan Semua *positive* yang ada. Presisi menggambarkan ketepatan antara data yang diminta dengan hasil prediksi oleh model. Presisi dapat dihitung dengan persamaan 5

$$Presisi = \frac{TP}{TP + FP} \times 100\% \quad (5)$$

Dimana TP singkatan dari *True Positive* dan FP singkatan dari *False Negative*

K. Recall

Recall adalah keberhasilan sebuah model dalam mendeteksi kembali sebuah informasi. *Recall* dapat dihitung dengan persamaan 6

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

Dimana TP singkatan dari *True Positive* dan FN singkatan dari *False Negative*.

L. F1-Score

F1-Score adalah perbandingan rata-rata dari nilai precision dan recall. F1-

Score mempunyai nilai tertinggi sebesar 1 dan terendah sebesar 0, jika Nilai F1- Score semakin mendekati 1, menunjukkan bahwa kinerja sistem bekerja dengan baik. F1-Score dapat dihitung dengan persamaan 7

$$F1Score = 2X \frac{Recall \times Presisi}{Recall + Presisi} \times 100\% \quad (7)$$

IV. HASIL DAN PEMBAHASAN

A. Skenario Pengujian Sistem

Dalam tugas akhir ini, akan dilakukan 4 skenario pengujian. Setiap pengujian memiliki tujuan untuk mendapatkan hasil kinerja sistem terbaik. Hasil dari setiap pengujian akan digunakan kembali pada pengujian selanjutnya. Tujuannya adalah untuk mendapatkan hasil yang paling optimal. Berikut ini rencana pengujian yang akan dilakukan:

1. Skenario pertama

Skenario pengujian pertama melibatkan proses evaluasi terhadap berbagai optimizer. Tujuannya adalah mengidentifikasi optimizer yang menghasilkan akurasi tertinggi, yang kemudian akan digunakan pada tahapan pengujian selanjutnya.

2. Skenario kedua

Skenario kedua pengujian ini difokuskan pada learning rate. Beberapa nilai learning rate akan diuji menggunakan optimizer terbaik pada scenario pertama untuk menentukan nilai yang learning rate terbaik.

3. Skenario ketiga

Skenario pengujian ketiga yaitu proses evaluasi terhadap batch size menggunakan optimizer dan learning rate terbaik yang diperoleh dari scenario sebelumnya. Tujuannya adalah mengidentifikasi batch size yang mendapatkan nilai dengan akurasi tertinggi.

4. Skenario keempat

Skenario pengujian keempat yaitu proses evaluasi terhadap epoch untuk mendapatkan nilai yang terbaik menggunakan optimizer, learning rate dan batch size terbaik yang diperoleh dari scenario sebelumnya.

B. Hasil Pengujian

Data yang akan diuji adalah klasifikasi citra cuaca yang berjumlah 768 citra dan telah dilakukan *resize* dengan ukuran 224 x 224 piksel. Terdapat beberapa parameter awal yang akan diukur selama pengujian meliputi *optimizer*, *learning rate*, *batch size*, dan *epoch*. Untuk melihat parameter awal pada pengujian pertama, dapat dilihat pada Tabel 2

Tabel 2 Parameter Awal

Ukuran Citra	224 x 224
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	0,01
<i>Batch Size</i>	8
<i>Epoch</i>	50

Pada parameter *optimizer*, empat *optimizer* yang akan diterapkan adalah Adam, Nadam, RMSprop dan Stochastic Gradient Descent (SGD). Untuk parameter *learning rate*, terdapat tiga variasi nilai yang akan diuji, yaitu 0.01, 0.001 dan 0,0001. Pada parameter *batch size*, akan digunakan empat variasi, yakni 8, 16, dan 32. Sedangkan pada parameter parameter *epoch*, akan digunakan variasi nilai sebanyak empat, yaitu 50, 60, 70 dan 80. Proses pelatihan model akan memanfaatkan 80% dari total 1664 data, sementara proses pengujian akan menggunakan sisa dari proses pelatihan yaitu 20%. Pengujian ini bertujuan untuk mendapatkan nilai akurasi, *loss*, *precision*, *recall*, dan *F1-score*.

Skenario Pertama, Pengujian *Optimizer* Terbaik

Pada skenario pertama ini akan dilakukan pengujian terhadap *optimizer* dengan parameter awal seperti pada Table 3. Pengujian ini dilakukan dengan tujuan untuk mendapatkan *optimizer* yang paling efisien untuk mengklasifikasi citra cuaca. *Optimizer* yang akan diuji yaitu Adam, Nadam, RMSprop dan *Stochastic Gradient Descent* (SGD). Hasil dari pengujian *optimizer* tersebut dapat dilihat pada Tabel 4

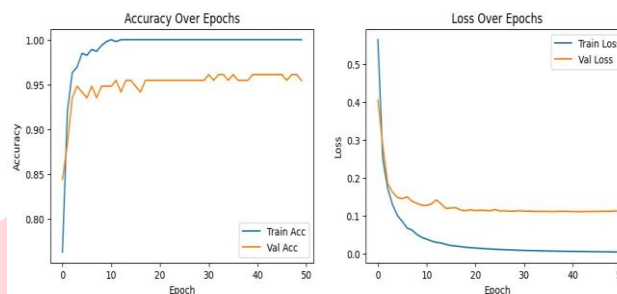
Tabel 3 Hasil Perbandingan Dan Pengujian *Optimizer*

Optimize	Training	Validation	Testing
Adam	100%	96,10%	92,21%
Nadam	100%	97,40%	92,86%
SGD	100%	95,45%	95,45%
<u>RMSprop</u>	100%	96,10%	94,81%

Tabel 4 Hasil Uji *Optimizer*

Optimizer	Akurasi	Persisi	Recall	F1-Score
Adam	92,21%	92,33%	92,21%	92,25%
Nadam	92,86%	92,92%	92,86%	92,82%
SGD	95,45%	95,46%	95,45%	95,45%
<u>RMSprop</u>	94,81%	95,13%	94,81%	94,77%

Berdasarkan Tabel 4, optimisasi Stochastic Gradient Descent (SGD) memiliki akurasi paling baik jika dibandingkan dengan optimisasi Adam, Nadam, dan RMSprop dengan nilai akurasi sebesar 95,45%, presisi sebesar 95,46%, recall sebesar 95,45%, dan F1-score sebesar 95,45%. Sedangkan akurasi optimisasi Adam sebesar 92,21%, Nadam sebesar 98,17%, dan RMSprop sebesar 98,17%. Untuk pengujian terbaik dapat dilihat pada Gambar 9

Gambar 9 Grafik Pengujian *Optimizer*

Berdasarkan Gambar 9, grafik menunjukkan bahwa optimisasi Stochastic Gradient Descent (SGD) memperoleh akurasi tertinggi. Hal ini disebabkan oleh efisiensinya yang tinggi serta proses komputasi yang relatif ringan. Selain itu, optimisasi Stochastic Gradient Descent (SGD) juga memiliki kompatibilitas yang baik dalam proses optimasi sistem klasifikasi katarak. Oleh karena itu, optimisasi *Stochastic Gradient Descent* (SGD) dipilih untuk digunakan pada skenario pengujian berikutnya.

Skenario Kedua, Pengujian *Learning Rate* Terbaik

Pada skenario kedua, dilakukan pengujian terhadap nilai *learning rate* untuk menentukan nilai terbaik serta mengetahui pengaruhnya terhadap kinerja sistem. Nilai *learning rate* yang diuji dalam skenario ini meliputi 0,01, 0,001, dan 0,0001. Nilai *learning rate* yang menghasilkan performa terbaik nantinya akan digunakan pada skenario pengujian selanjutnya. Proses pengujian ini dilakukan menggunakan parameter yang diasumsikan dan hasil terbaik dari skenario sebelumnya, yaitu optimisasi *Stochastic Gradient Descent* (SGD), *batch size* yang ditetapkan sebesar 8, dan *epoch* sebanyak 50. Hasil dari pengujian skenario kedua ditampilkan pada Tabel 5 dan 6

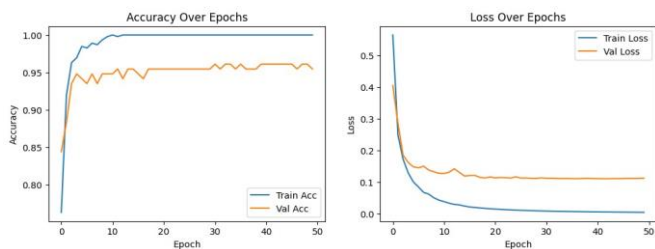
Tabel 5 Hasil Perbandingan Dan Pengujian *Learning rate*

Learning Rate	Training	Validation	Testing
0,01	100%	95,45%	95,45%
0,001	98,91%	94,16%	94,16%
0,0001	87,39%	88,31%	86,36%

Tabel 6 Hasil Uji *Learning Rate*

Learning Rate	Akurasi	Persisi	Recall	F1-Score
0,01	95,45%	95,46%	95,45%	95,45%
0,001	94,16%	94,22%	94,16%	94,18%
0,0001	86,36%	86,23%	86,36%	86,28%

Berdasarkan Tabel 5, *learning rate* 0,01 memiliki akurasi paling baik jika dibandingkan dengan *learning rate* 0,001 dan 0,0001 dengan nilai akurasi sebesar 95,45%, presisi sebesar 95,46%, *recall* sebesar 95,45%, dan *F1-score* sebesar 95,45%. Sedangkan akurasi 0,001 sebesar 94,16% dan 0,0001 sebesar 86,36%. Untuk pengujian terbaik dapat dilihat pada Gambar 10.



Gambar 10 Grafik Pengujian *Learning Rate*

Berdasarkan hasil pada Gambar 10, performa terbaik diperoleh saat menggunakan *learning rate* sebesar 0,01 dalam memprediksi nilai TP dan TN. Nilai *precision* yang dihasilkan menunjukkan persentase prediksi TP terhadap seluruh hasil prediksi positif sebesar 95,45%. Kondisi ini terjadi karena semakin kecil nilai *learning rate*, maka ketelitian model meningkat, sedangkan nilai *learning rate* yang terlalu besar justru menurunkan ketelitian jaringan. Berdasarkan hasil tersebut, pengujian selanjutnya akan menggunakan *learning rate* sebesar 0,01 karena memberikan performa paling optimal dalam skenario pengujian nilai *learning rate*.

Skenario Ketiga, Pengujian *Batch Size* Terbaik

Pada skenario ketiga, dilakukan pengujian terhadap *Batch size* untuk menentukan nilai terbaik serta mengetahui pengaruhnya terhadap kinerja sistem. *Batch size* yang diuji dalam skenario ini meliputi 8, 16, dan 32. *Batch size* yang menghasilkan performa terbaik nantinya akan digunakan pada skenario pengujian selanjutnya. Proses pengujian ini dilakukan menggunakan parameter yang diasumsikan dan hasil terbaik dari skenario sebelumnya, yaitu optimisasi *Stochastic Gradient Descent* (SGD), *epoch* sebanyak 50, dan nilai *learning rate* 0,01. Hasil dari pengujian skenario ketiga ditampilkan pada Tabel 7 dan 8

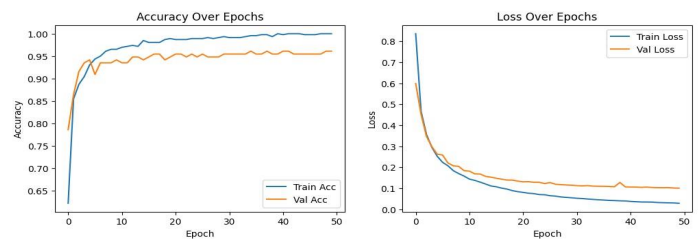
Tabel 7 Hasil Perbandingan Dan Pengujian *Batch Size*

Batch Size	Training	Validation	Testing
8	100%	95,45%	95,45%
16	100%	94,81%	94,81%
32	100%	96,10%	96,10%

Tabel 8 Hasil Uji *Batch Size*

Batch Size	Akurasi	Persisi	Recall	F1-Score
8	95,45%	95,46%	95,45%	95,45%
16	94,81%	94,83%	94,81%	94,80%
32	96,10%	96,14%	96,10%	96,11%

Berdasarkan Tabel 8, *batch size* 32 memiliki akurasi paling baik jika dibandingkan dengan *batch size* 8 dan 16, dengan nilai akurasi sebesar 96,10%, presisi sebesar 96,14%, *recall* sebesar 96,10%, dan *F1-score* sebesar 96,11%. Sedangkan akurasi *batch size* 8 sebesar 95,45%, dan 16 sebesar 98,33%. Untuk pengujian terbaik dapat dilihat pada Gambar 11.



Gambar 11 Grafik Pengujian *Batch Size*

Berdasarkan Gambar 11, performa terbaik diperoleh saat menggunakan *batch size* sebesar 32. Dari hasil pengujian skenario ketiga, dapat disimpulkan bahwa *batch size* 32 menjadi nilai paling optimal dalam meningkatkan akurasi sistem klasifikasi katarak. Hal ini karena sistem mampu mencapai *accuracy* prediksi benar untuk TP dan TN sebesar 96,10%, dengan *precision* prediksi TP terhadap seluruh prediksi positif sebesar 96,14%, dan *recall* prediksi TP terhadap seluruh data positif sebenarnya sebesar 96,10%. Kesimpulannya, semakin besar nilai *batch size* yang digunakan, cenderung menurunkan hasil *accuracy*, *precision*, *recall*, dan *F1-score* yang diperoleh.

Scenario Keempat, Pengujian *Epoch* Terbaik

Pada skenario keempat, dilakukan pengujian terhadap *epoch* untuk menentukan nilai terbaik serta mengetahui pengaruhnya terhadap kinerja sistem. *epoch* yang diuji dalam skenario ini meliputi *epoch* 50, 60, 70, dan 80. *Epoch* yang menghasilkan performa terbaik nantinya akan menghasilkan skenario pengujian terbaik. Proses pengujian ini dilakukan menggunakan parameter yang diasumsikan dan hasil terbaik dari skenario sebelumnya, yaitu optimisasi *Stochastic Gradient Descent* (SGD), *batch size* 32, dan nilai *learning rate* 0,01. Hasil dari pengujian skenario kedua ditampilkan pada Tabel 9 dan 10

Tabel 9 Hasil Perbandingan Dan Pengujian *Epoch*

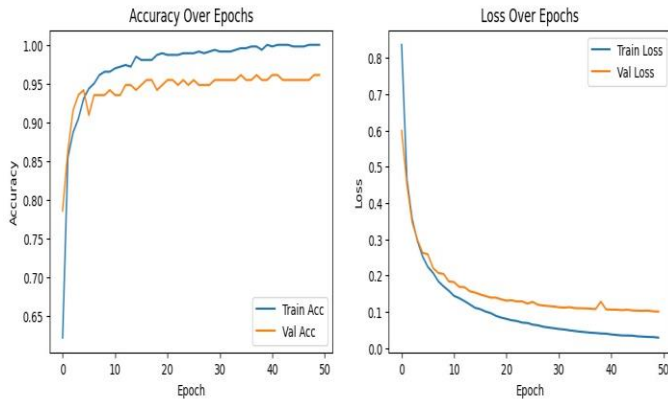
Epoch	Training	Validation	Testing
50	100%	96,10%	96,10%
60	100%	96,75%	94,16%
70	100%	96,75%	95,45%
80	100%	96,10%	94,81%

Tabel 10 Hasil Uji *Epoch*

Epoch	Akurasi	Persisi	Recall	F1-Score
50	96,10%	96,14%	96,10%	96,11%
60	94,16%	94,25%	94,16%	94,12%
70	95,45%	95,58%	95,45%	95,44%
80	94,81%	94,78%	94,81%	94,79%

Berdasarkan Tabel 10, *epoch* 50 memiliki akurasi paling baik jika dibandingkan dengan *epoch* 60, 70, dan 80. dengan nilai akurasi sebesar 96,10%, presisi sebesar

96,14%, *recall* sebesar 96,10%, dan *F1-score* sebesar 96,11%. Sedangkan akurasi *epoch* 60 sebesar 94,16%, 70 sebesar 95,58% dan 80 sebesar 94,81%. Untuk pengujian terbaik dapat dilihat pada Gambar 12.



Gambar 12 Grafik Pengujian Epoch

Berdasarkan Gambar 12, performa terbaik diperoleh saat *Epoch* 50. Dari hasil pengujian skenario keempat, dapat disimpulkan bahwa *epoch* 50 menjadi nilai paling optimal dalam meningkatkan akurasi sistem klasifikasi katarak, dikarenakan *epoch* 60, 70, dan 80 tidak memberikan hasil akurasi yang optimal.

C. Pengujian Dataset yang Sudah DiBalancing

Pada pengujian ini, dilakukan pengujian terhadap citra cuaca yang sudah dibalancing untuk menentukan nilai terbaik serta mengetahui pengaruhnya terhadap kinerja sistem. Data citra yang digunakan setelah balancing bisa dilihat pada Tabel 11

Tabel 11 Detail Jumlah Dataset Yang Sudah Dibalancing

Kelas	Jumlah
Hujan	215
Berawan	215
Cerah	215

Proses pengujian ini dilakukan menggunakan parameter yang diasumsikan dan hasil terbaik dari skenario sebelumnya, yaitu optimisasi *Stochastic Gradient Descent* (SGD), *batch size* 32, *learning rate* 0,01, dan nilai *epoch* 50. Hasil dari pengujian skenario kedua ditampilkan pada Tabel 12 dan 13.

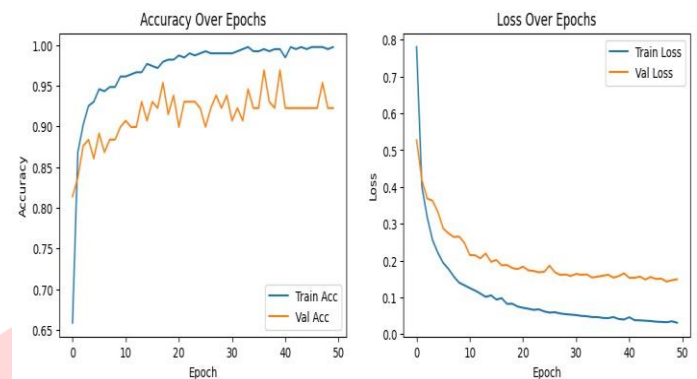
Tabel 12 Hasil Pengujian Dataset Balancing

Training	Validation	Testing
99,78%	92,25%	92,25%

Tabel 13 Hasil Uji Dataset Balancing

Akurasi	Pesisi	Recall	F1-Score
92,25%	92,80%	92,25%	92,28%

Berdasarkan Tabel 13, data yang sudah di balancing mendapatkan nilai akurasi 92,25%, nilai presisi 92,80%, nilai recall 92,25%, dan nilai *F1-score* 92,28%. Untuk grafik dari data pengujian bisa dilihat pada Gambar 13.



Gambar 13 Grafik Hasil Uji Dataset Balancing

D. Analisa Hasil

Dari pengujian terhadap keempat skenario tersebut, dapat disimpulkan bahwa konfigurasi optimal untuk model klasifikasi citra daun kopi ini adalah menggunakan optimizer *Stochastic Gradient Descent* (SGD), *learning rate* 0.01, *batch size* 32, dan *epoch* 50. optimizer *Stochastic Gradient Descent* (SGD) adalah sebagai optimizer yang paling optimal, sementara *learning rate* 0.01 memberikan keseimbangan terbaik antara kecepatan konvergensi dan stabilitas model, *batch size* 52 menawarkan keseimbangan terbaik antara kinerja dan efisiensi penggunaan memori, *Epoch* 50 memungkinkan model untuk belajar secara optimal tanpa mengalami *overfitting*. Tabel 15 merupakan skenario terbaik yang diperoleh dari pengujian ini.

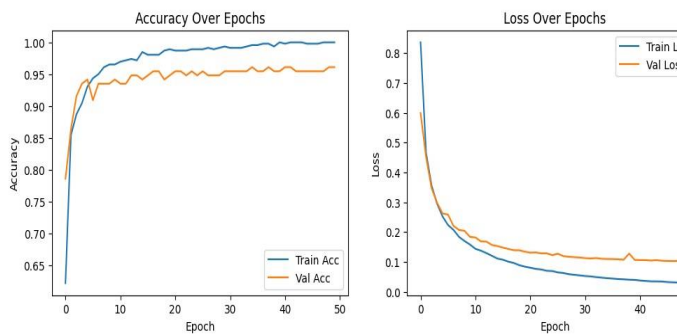
Tabel 14 Hasil Pengujian Dan Perbandingan Skenario Terbaik

Skenario Terbaik		Training	Validation	Testing
Optimizer	SGD	100%	95,45%	95,45%
Learning Rate	0,01	100%	95,45%	95,45%
Batch Size	32	100%	96,10%	96,10%
Epoch	50	100%	96,10%	96,10%

Tabel 15 Hasil Uji Skenario Terbaik

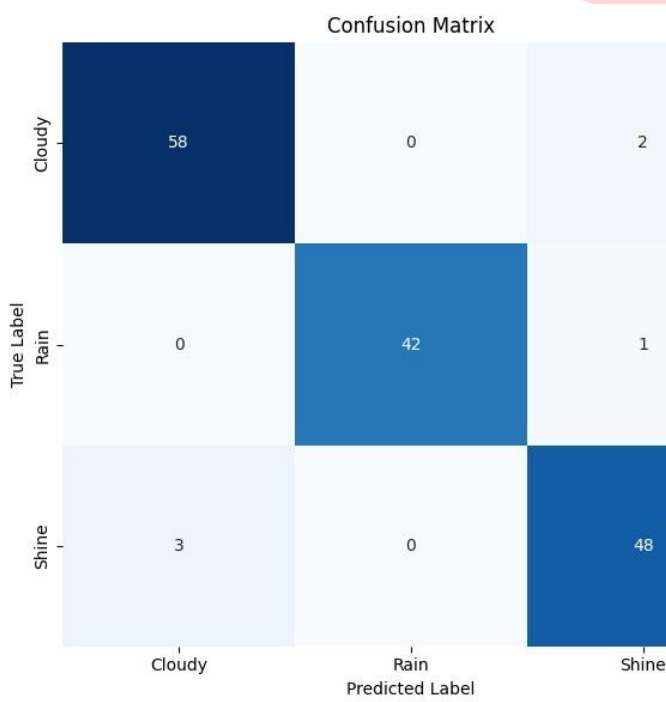
Skenario Terbaik		Akurasi	Pesisi	Recall	F1-Score
Optimizer	SGD	95,45%	95,46%	95,45%	95,45%
Learning Rate	0,01	95,45%	95,46%	95,45%	95,45%
Batch Size	32	96,10%	96,14%	96,10%	96,11%
Epoch	50	96,10%	96,14%	96,10%	96,11%

Konfigurasi ini menghasilkan kinerja terbaik dengan akurasi mencapai 100% dan *loss* 0.027. Pada grafik akurasi tidak mengalami *overfitting*, menunjukkan peningkatan yang stabil dan konsisten dikarenakan penggunaan kombinasi parameter yang optimal, yang dapat disimpulkan bahwa pemilihan parameter yang telah di uji efektif dalam mengoptimalkan kinerja model dan mencapai akurasi yang tinggi. Hasil grafik akurasi dan loss ditunjukan dari Gambar 14 berikut.



Gambar 14 Hasil Grafik Akurasi dan *Loss*

Pada Gambar 15 dibawah terlihat bahwa *confusion matrix* menunjukkan kinerja model yang sangat baik dalam mengklasifikasikan citra cuaca, di mana sebagian besar citra diklasifikasikan dengan benar, ditunjukkan oleh nilai tinggi pada diagonal utama yang mewakili *true positive* setiap kelas. Total untuk citra yang diklasifikasi adalah 768 citra yang terbagi menjadi 3 kelas yaitu Cerah 253, Berawan 300, Hujan 215.



Gambar 15 Confusion Matrix

Meskipun terdapat beberapa kesalahan klasifikasi *minor* (*false positive* dan *false negative*) di luar diagonal, jumlahnya relatif kecil, menunjukkan kemampuan model yang baik dalam membedakan kelas-kelas citra cuaca. Model mungkin sedikit lebih sulit membedakan antara kelas "*Shine*" dan "*Cloudy*", tetapi secara keseluruhan, model menunjukkan performa yang sangat baik dengan akurasi tinggi dalam klasifikasi citra cuaca.

E. Penjelasan Mengapa Nilai Akurasi dan Recall Sama
Hasil pengujian klasifikasi cuaca ini mendapatkan nilai akurasi dan recall yang sama, dikarenakan model prediksi disetiap kelas (*Cloudy*, *Rain*, *Shine*) positif atau semua prediksi benar, yang mengakibatkan nilai akurasi dan recall menjadi sama.

V. KESIMPULAN

Berdasarkan rangkaian proses perancangan, implementasi, dan pengujian sistem klasifikasi cuaca yang telah dilakukan, diperoleh beberapa poin kesimpulan sebagai berikut:

Sistem klasifikasi cuaca berbasis citra berhasil dikembangkan dengan memanfaatkan metode Convolutional Neural Network (CNN) menggunakan arsitektur MobileNetV2. Sistem ini mampu mengenali dan membedakan gambar cuaca ke dalam tiga kategori, yaitu cerah, berawan, dan hujan, melalui tahapan seperti pengambilan data, pengolahan awal, pelatihan model, dan evaluasi hasil

Evaluasi performa sistem dilakukan menggunakan beberapa parameter seperti akurasi, presisi, recall, dan F1-score. Dari serangkaian uji coba yang melibatkan berbagai kombinasi parameter, ditemukan konfigurasi terbaik dengan penggunaan: 1. Optimizer: Stochastic Gradient Descent (SGD), 2. Learning rate: 0,01, 3. Batch size: 32, 4. Epoch: 50

Dengan konfigurasi tersebut, sistem mampu mencapai akurasi pengujian sebesar 96,10%, menunjukkan kemampuan model dalam mengenali pola cuaca secara konsisten dan akurat.

Hasil analisis kinerja menunjukkan bahwa MobileNetV2 sebagai arsitektur CNN memberikan keunggulan dalam hal efisiensi dan akurasi. Model ini mampu bekerja dengan baik meskipun pada lingkungan komputasi yang terbatas, sehingga layak diterapkan untuk sistem klasifikasi cuaca yang membutuhkan kecepatan serta keakuratan dalam pengambilan keputusan.

REFERENSI

- [1] R. P. S. S. Randhy Sulistyo Budi, "KLASIFIKASI CUACA MENGGUNAKAN METODE CONVOLUTIONAL (CNN)," *e-Proceeding of Engineering*, 2021.
- [2] I. A. D. Faqih Hamami, "KLASIFIKASI CUACA PROVINSI DKI JAKARTA MENGGUNAKAN," *TEKNOINFO*, vol. 16, 2022.
- [3] T. S. F. A. F. I. k. Amril Mutoi Siregar, "Klasifikasi untuk Prediksi Cuaca Menggunakan Ensemble Learning," *PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika*, vol. 13, 2020.
- [4] D. H. Jasen Jonathan, "Penentuan Epochs Hasil Model Terbaik: Studi Kasus Algoritma YOLOv8," *Digital Transformation Technology (Digitech)*, 2024.
- [5] D. I. J. R. M. ., S. A. W. P. Atina Nur Azizah, "PERKIRAAN CUACA BERBASIS ANALISIS DATA MENGGUNAKAN METODE COARSE TO FINE SEARCH DAN FUZZY LOGIC STUDI KASUS CUACA BERPOTENSI HUJAN," 2019.
- [6] D. A. A. M. Dedy Agung Prabowo, "DETEKSI DAN PERHITUNGAN OBJEK BERDASARKAN

WARNA MENGGUNAKAN COLOR OBJECT TRACKING," *Jurnal Pseudocode*, vol. 2, 2018.

[7] R. N. Keiron O'Shee, "An Introduction to Convolutional Neural Networks," *arXiv:1511.08458v2 [cs.NE]*, 2015.

[8] M. Y. W. I. F. Achmad Reza Fahrurroji, "IMPLEMENTASI ALGORITMA CNN MOBILENET UNTUK KLASIFIKASI GAMBAR SAMPAH DI BANK SAMPAH," *PROSISKO*, vol. 11, 2024.

[9] E. H. Arum Tiara Sari, "Penerapan Convolutional Neural Network Deep Learning dalam Pendeteksian Citra Biji Jagung Kering," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 2021.

[10] T. S. F. A. F. I. k. Amril Mutoi Siregar, "Klasifikasi untuk Prediksi Cuaca Menggunakan Ensemble Learning," *Jurnal Pengkajian dan Penerapan Teknik Informatika*, vol. 13, 2020.

[11] A. A. K. D. I Gede Aris Gunadi, "Klasifikasi Curah Hujan di Provinsi Bali Berdasarkan Metode," *Wahana Matematika dan Sains: Jurnal Matematika, Sains, dan Pembelajarannya*, vol. 12 no. 1, 2018.

[12] Y. H. M. Hendriyana, "Identifikasi Jenis Kayu menggunakan Convolutional Neural Network," *JURNAL RESTI*, vol. 1, 2017.

[13] F. N. S. M. A. A. M. Firaz Kasfillah Hanif, "PERAMALAN CUACA DI KABUPATEN BANDUNG MENGGUNAKAN ALGORITMA ITERATIVE DICHOTOMISER TREE (ID3)," 2019.

[14] WIKIPEDIA, [Online]. Available: https://id.m.wikipedia.org/wiki/Berkas:The_three_primary_colors_of_RGB_Color_Model_%28Red,_Green,_Blue%29.png. [Accessed 10 December 2024].

[15] L. N. S. A. W. Muhammad Dafa Maulana, "Evaluasi Kinerja YOLOv8 dalam Identifikasi Kesegaran Ikan dengan Metode Deteksi Objek," *eProceeding of Engineering*, vol. 11, 2024.

[16] N. R. T. S. Imam Maulana, "ANALISIS PENGGUNAAN MODEL YOLOV8 (YOU ONLY LOOK ONCE) TERHADAP DETEKSI CITRA SENJATA BERBAHAYA," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, 2023.