

Perbandingan Performa Algoritma VGG16 dan VGG19 Melalui Metode CNN Untuk Klasifikasi Varietas Beras

Adisaputra Zidha Noorizki ¹ dan Weny Indah Kusumawati ^{2*}

^{1,2*} Teknik Komputer, Fakultas Teknologi dan Informatika, Universitas Dinamika, Surabaya, Indonesia

* Korespondensi: weny@dinamika.ac.id

Abstrak: Beras merupakan jenis pangan utama yang paling populer di konsumsi di seluruh dunia. Berbagai negara pun melakukan proses impor untuk memenuhi kebutuhan penduduk di negaranya, salah satunya adalah Indonesia. Indonesia merupakan negara yang tergolong memiliki permintaan pasar pangan yang tinggi di ruang lingkup global. Masuknya berbagai varietas beras akibat adanya kegiatan impor di Indonesia membuat para importir mengalami kesulitan selama proses penyortiran produknya. Seiring berjalannya kemajuan teknologi, para importir juga harus terus beradaptasi, salah satunya dengan memanfaatkan salah satu teknologi yang akhir-akhir ini sedang marak dikembangkan oleh para peneliti. CNN atau *Convolutional Neural Network* merupakan salah satu teknologi yang memiliki kemampuan yang baik dalam mengklasifikasikan citra gambar. Penelitian ini secara khusus memfokuskan pada perbandingan kinerja dua algoritma CNN, yaitu VGG16 dan VGG19. Hasil penelitian ini menyatakan bahwa algoritma VGG16 mampu mencapai akurasi sebesar 98% dengan waktu *training* 73,405 detik, sementara algoritma VGG19 mencapai akurasi 97% dengan waktu *training* 78,098 detik. Hal ini menunjukkan bahwa algoritma VGG16 secara signifikan lebih unggul dibandingkan dengan algoritma VGG19 dalam mengklasifikasikan varietas beras menggunakan dataset yang terdiri atas lima kelas.

Kata kunci: CNN; Machine Learning; Varietas Beras.

1. Pendahuluan

Beras merupakan salah satu jenis bahan pangan yang secara luas dikonsumsi dan populer di seluruh dunia. Bijinya dihasilkan oleh tanaman padi yang dikenal dengan nama ilmiah *Oryza sativa*. Kandungan karbohidrat dalam beras, terutama dalam bentuk amilosa dan amilopektin, menjadikannya sebagai sumber utama energi bagi manusia dalam menjalani aktivitas sehari-hari [1]. Namun, keberagaman beras tidak hanya berhenti pada aspek nutrisi. Berbagai varietas beras yang ada memiliki peran krusial dalam pertanian dan pangan. Perbedaan-perbedaan ini memengaruhi produktivitas, kualitas, dan adaptasi tanaman terhadap lingkungan di berbagai wilayah. Dalam konteks pertanian, keberagaman varietas beras ini dapat memberikan hasil yang berbeda dalam hal rasa, tekstur, nilai gizi, serta kemampuan tahan terhadap kondisi lingkungan yang beragam pula [2]. Semua variasi ini memberikan konsumen dan pelaku industri pangan pilihan yang beragam sesuai dengan kebutuhan dan preferensi mereka. Dengan keadaan pasar beras yang kompleks, pentingnya pengenalan varietas yang akurat menjadi kunci dalam memenuhi permintaan konsumen dan menjaga kelangsungan pasokan dalam keadaan yang stabil.

Indonesia, sebagai salah satu kontributor penting dalam permintaan pasar pangan global, sangat bergantung pada beras karena mayoritas penduduknya menjadikan beras sebagai makanan pokok [3]. Penduduk asli Indonesia mengandalkan beras sebagai pilar utama dalam pola makan

mereka. Tingginya ketergantungan ini telah menghasilkan peningkatan permintaan yang konsisten, namun juga berdampak pada kesenjangan antara jumlah produsen beras dan konsumen di Indonesia. Hal ini mendorong negara untuk melakukan impor beras dari berbagai negara guna menjaga keseimbangan antara pasokan dan permintaan.

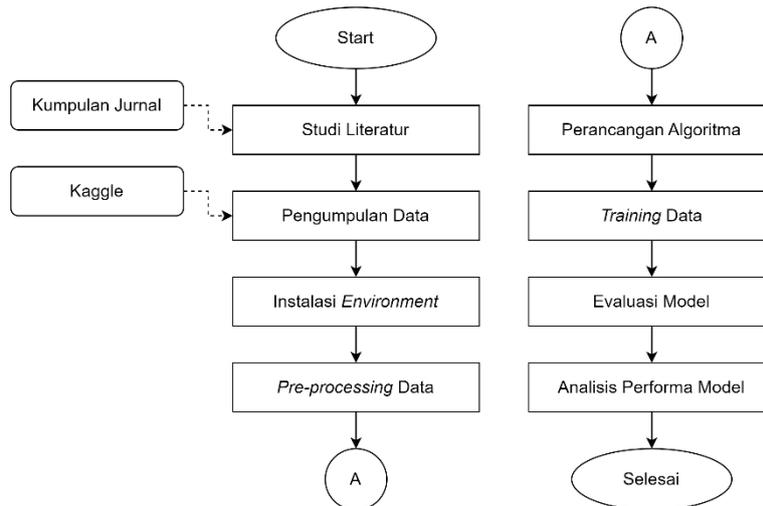
Masuknya berbagai macam varietas beras yang berasal dari berbagai negara pula yang mengakibatkan importir di Indonesia kewalahan saat melakukan kegiatan sortir produk. Berdasarkan data Badan Pusat Statistik (BPS) pada tahun 2022, negara India, Pakistan, Vietnam, Thailand, Myanmar dan Jepang menjadi negara-negara dengan angka kegiatan impor beras tertinggi ke Indonesia, ada sekitar 429,207 ton beras yang telah masuk ke negara Indonesia [4]. Penelitian ini akan difokuskan pada lima jenis beras, yaitu Arburio, Bastami, Ipsala, Jasmine, dan Karacadag. Namun, hingga saat ini belum ada data yang valid mengenai apakah jenis-jenis beras ini beredar di Indonesia. Butuh waktu yang cukup lama bagi beberapa importir baru untuk melakukan kegiatan sortir produk serta memastikan termasuk ke varietas mana beras yang saat itu sedang disortir. Sedangkan beras yang dijual di pasaran diberi harga sesuai dengan beberapa patokan yang umumnya digunakan antara lain tekstur, bentuk, warna, dan masih banyak lagi. Dengan demikian, proses sortir memang sudah semestinya dilakukan guna menentukan harga dari beras tersebut.

Seiring kemajuan teknologi saat ini, sudah semestinya para pelaku usaha terutama dibidang pertanian dan perdagangan terkhusus beras harus terus mengikuti kemajuan teknologi dan beradaptasi. Beberapa tahun terakhir, para ahli telah bersaing untuk melakukan penelitian tentang pengolahan dan klasifikasi citra menggunakan salah satu metode dalam *deep learning* yang dikenal sebagai *Convolutional Neural Network*, atau yang lebih umum dikenal sebagai *CNN*. *Convolutional Neural Network* (*CNN*) merupakan salah satu metode klasifikasi yang menggunakan lapisan konvolusi untuk memproses input dengan filter tertentu [5]. *CNN* merupakan variasi pengembangan dari metode sebelumnya yaitu jaringan syaraf tiruan yang memiliki beberapa lapisan tersembunyi dan memiliki bobot saja. *CNN* umumnya digunakan untuk melakukan tugas terkait dengan pengolahan data gambar yang diklasifikasikan dengan metode *supervised learning*. Pembelajaran yang diawasi/*supervised learning* merupakan jenis algoritma yang menghasilkan fungsi untuk memetakan input menuju output yang diinginkan [6]. Terdapat beberapa lapisan pada *model* dasar *CNN* antara lain lapisan *convolution*/konvolusi, fungsi aktivasi, lapisan *pooling*, lapisan *flatten*, dan yang terakhir adalah lapisan *fully connected* [7].

CNN sendiri memiliki berbagai algoritma yang telah ditemukan oleh peneliti-peneliti terdahulu. Dalam penelitian ini, peneliti membandingkan kinerja antara algoritma *VGG16* dan *VGG19* dalam mengerjakan tugas untuk mengklasifikasikan varietas beras. Nama algoritma ini diambil dari singkatan dari *Visual Geometry Group*, yang disingkat menjadi *VGG* dan kemudian angka 16 dan 19 merupakan jumlah lapisan/*layer* dari algoritma tersebut. *VGG* dibuat oleh sekelompok peneliti Universitas Oxford, Inggris. Kelompok penelitian *VGG* dipimpin oleh Profesor Andrew Zisserman dan Profesor Karen Simonyan [8]. Algoritma *VGG* merupakan hasil perbaikan dan pengembangan dari *AlexNet* yang telah dipublikasikan pada tahun 2012. Arsitektur dari *VGG* berfokus pada meningkatkan proses ekstraksi fitur dengan menggunakan lapisan konvolusi yang lebih banyak [9].

2. Bahan dan Metode

Pada penelitian ini, urutan kegiatan yang dijalani oleh peneliti dapat dilihat dalam diagram alir yang terdapat pada Gambar 1. Penelitian dimulai dengan studi literatur melalui referensi berbagai jurnal, kemudian dilanjutkan dengan pengumpulan data yang digunakan dalam proses pelatihan menggunakan algoritma *VGG16* dan *VGG19*. Proses pelatihan dilakukan dengan beberapa jumlah iterasi. Setelah mendapatkan hasil dari proses pembelajaran, peneliti melakukan perbandingan untuk menentukan algoritma yang memiliki performa lebih baik dalam implementasinya pada klasifikasi citra varietas beras.



Gambar 1. Diagram alir penelitian

2.1. Studi Literatur

Penelitian ini dimulai dengan melakukan studi literatur yang mencakup berbagai jurnal yang relevan dengan penggunaan algoritma *VGG16* dan *VGG19* melalui metode *Convolutional Neural Network* (CNN) dalam konteks klasifikasi citra gambar dari berbagai objek. Serta dapat menggunakan hasil dari *model* yang telah dilatih pada penelitian-penelitian sebelumnya sebagai referensi untuk *model* yang digunakan dalam penelitian ini.

2.2. Pengumpulan Data

Dalam tahap ini, peneliti menghimpun data yang diperlukan untuk penelitian, memanfaatkan dataset yang tersedia secara terbuka dari sumber Kaggle. Dataset ini terdiri dari total 75,000 citra, dengan fokus pada gambar satu butir beras yang dibagi menjadi lima kelas, yaitu Arburio, Bastami, Ipsala, Jasmine, dan Karacadag. Jumlah data dalam setiap kelas adalah sekitar 15,000 citra [10]. Untuk memberikan gambaran visual tentang jenis citra yang ada dalam dataset ini, peneliti telah menyediakan Tabel 1 yang menggambarkan contoh-contoh dari setiap kelas. Serta setiap representasi citra dalam dataset ini memiliki ekstensi '.jpg'.

Tabel 1. Dataset varietas beras

Gambar Citra	Nama Kelas
	Arburio
	Bastami
	Ipsala
	Jasmine
	Karacadag

2.3. Instalasi Environment

Pada tahap ini, sebelum melanjutkan proses pengolahan data untuk meningkatkan efisiensi pelatihan data dalam *model Convolutional Neural Network (CNN)*, peneliti harus melakukan instalasi lingkungan yang sesuai. Dikarenakan penelitian ini menggunakan bahasa pemrograman *Python* dan berfokus pada klasifikasi gambar, beberapa *library* yang harus diinstal antara lain adalah *Keras*, *NumPy*, *Matplotlib*, *scikit-learn*, dan *OpenCV*. Perintah *install* dapat dilakukan dengan beberapa cara [11], salah satunya dengan menggunakan perintah '*pip*' yang mana dapat dilakukan pada *command prompt/terminal*, serta dapat dilihat pada Gambar 2.

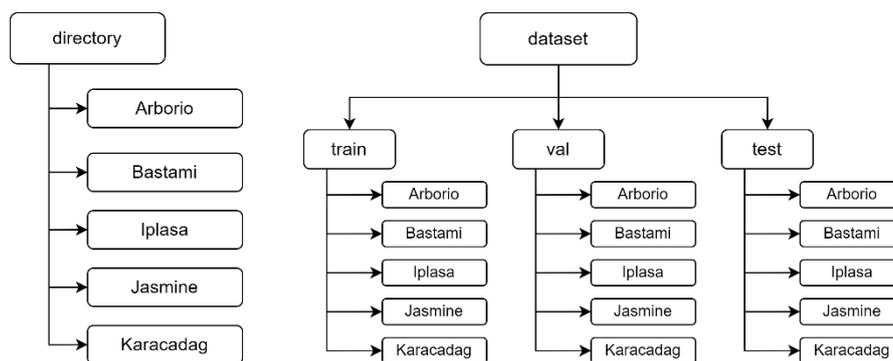
```
pip install keras
pip install numpy
pip install matplotlib
pip install scikit-learn
```

Gambar 2. Instalasi *library* menggunakan perintah '*pip*'

2.4. Pre-processing Data

Pada tahap ini, peneliti bertujuan untuk mengoptimalkan data agar lebih siap untuk proses pembelajaran menggunakan algoritma VGG16 dan VGG19. Dataset yang tersedia di Kaggle terdiri dari 15,000 citra gambar untuk setiap kelas, dan citra-citra tersebut telah dikelompokkan dalam folder sesuai dengan kelasnya. Langkah pertama yang dilakukan adalah memberikan label berupa angka dari 0 hingga 4 kepada setiap kelas. Dalam penelitian ini, peneliti tidak menggunakan seluruh data yang disediakan, namun peneliti menggunakan beberapa data saja, dengan menggunakan metode pengambilan sampel secara acak. Penelitian ini menggunakan data dengan jumlah 700 citra gambar dari setiap kelas.

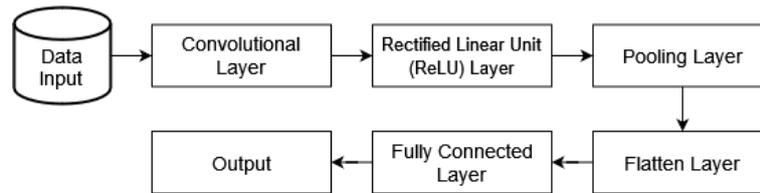
Selanjutnya, citra-citra ini dibagi kembali menjadi tiga bagian yang berbeda. Antara lain data latih (*training*), data validasi (*validation*), dan data uji (*testing*) seperti yang dapat dilihat dari Gambar 3 (a) dan (b). Pembagian ini akan dilakukan dengan proporsi 8:1.5:0.5 seperti yang telah diuji pada beberapa penelitian sebelumnya [12], atau dalam jumlahnya yaitu 560 citra gambar untuk data *training*, 105 citra gambar untuk data *validation*, dan 35 citra gambar untuk data *testing* setiap kelas varietas berasnya. Dengan total keseluruhan data yang digunakan pada penelitian kali ini sejumlah 3,500 citra gambar yang terbagi menjadi 5 kelas berbeda.



(a) Struktur folder asli

(b) Struktur folder setelah *pre-processing* data

Gambar 3. Struktur folder untuk data citra gambar varietas beras



Gambar 4. Diagram alir algoritma *Convolutional Neural Network* (CNN)

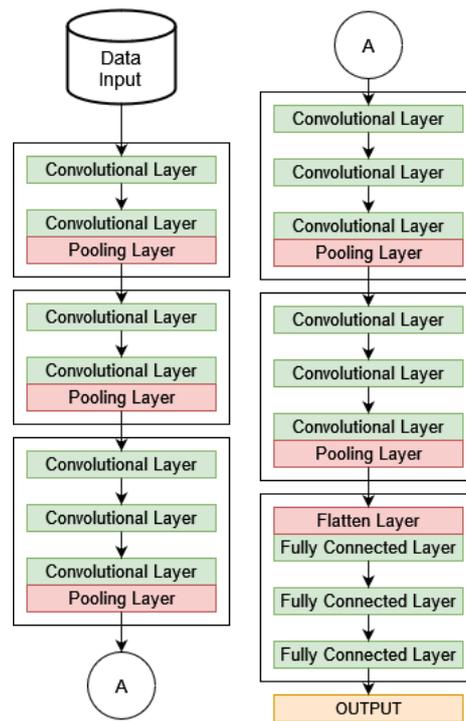
2.5. Training Data dengan metode CNN

Pada tahap ini, peneliti menentukan jumlah dari setiap lapisan/*layer* pada masing-masing algoritma yang digunakan. Dalam pengambilan keputusan ini, peneliti mempertimbangkan beberapa *model* yang telah dilakukan pengujian pada penelitian-penelitian sebelumnya. Hal ini memungkinkan peneliti untuk memanfaatkan *model* yang sudah terbukti berhasil dalam penelitian terdahulu sebagai acuan dalam merancang model pada penelitian kali ini. Dengan demikian, peneliti dapat memastikan bahwa pendekatan yang diambil sudah teruji dan memiliki potensi untuk memberikan hasil yang baik dalam klasifikasi citra gambar yang dilakukan. Demi memberikan gambaran yang lebih komprehensif terkait proses algoritma yang digunakan, peneliti juga menyertakan blok diagram yang mengilustrasikan struktur dan urutan operasi dari proses algoritma *Convolutional Neural Network* (CNN).

Blok diagram di atas memperlihatkan bagaimana data masukan melalui berbagai lapisan yang memiliki fungsi dan tugasnya masing-masing. *Layer*/lapisan yang pertama kali dilalui oleh data input adalah lapisan konvolusi yang memiliki fungsi untuk mengekstrak fitur-fitur dari gambar input melalui operasi konvolusi. Bertujuan untuk mendeteksi fitur-fitur seperti sudut, tepi, dan tekstur pada gambar input. Hasil proses konvolusi akan melewati lapisan ReLU yang merupakan salah satu jenis fungsi aktivasi di mana berperan untuk mengubah nilai-nilai negatif dari hasil konvolusi menjadi nol, sementara nilai-nilai positif akan tetap atau tidak berubah. Selanjutnya, keluaran dari lapisan ReLU akan mengalir ke lapisan *Pooling* yang memiliki tanggung jawab untuk mengurangi dimensi dari gambar. Dengan beberapa cara, diantaranya yaitu mengambil nilai rata-rata atau nilai maksimum dari setiap data pada gambar. Lapisan *Pooling* sangat membantu selama proses algoritma CNN berjalan, sebab lapisan ini dalam mempertahankan fitur-fitur penting sambil mengurangi kompleksitas dan sumber daya komputasi yang dibutuhkan. Setelah melewati lapisan konvolusi, ReLU, dan *Pooling*, hasilnya akan diarahkan ke lapisan yang bernama *Flatten* yang memiliki fungsi untuk merubah representasi spasial dari gambar menjadi vektor satu dimensi. Selanjutnya, akan melewati lapisan bernama *Fully Connected*, disetiap neuron di lapisan ini terhubung dengan setiap neuron di lapisan sebelumnya. Lapisan ini memiliki fungsi untuk mengintegrasikan informasi dari fitur-fitur yang telah dikenali dalam tahapan sebelumnya. Selain itu, lapisan ini juga melakukan proses pembelajaran (*learning*) untuk mengenali pola-pola kompleks dalam data dan menyesuaikan bobot-bobot antar-neuron. Setelah melalui lapisan *Fully Connected*, hasil akhirnya akan diteruskan ke lapisan *Output*. Pada lapisan ini, jumlah neuron akan disesuaikan dengan jumlah kelas yang ingin diprediksi menggunakan algoritma CNN. Dengan demikian, diagram pada Gambar 4 akan memberikan gambaran secara umum bagaimana tahapan atau proses yang terjadi pada algoritma *Convolutional Neural Network* (CNN).

2.5.1. VGG16 Model

Proses yang terjadi dalam *CNN-VGG16* pada dasarnya tidak berbeda jauh dengan uraian yang telah dijelaskan sebelumnya oleh peneliti. Namun, ada sedikit perbedaan dalam cara menghitung jumlah lapisan serta menamai setiap lapisan tersebut. Pada Gambar 4, peneliti memberikan penjelasan global mengenai lapisan-lapisan mana yang akan dilewati oleh data gambar hingga menghasilkan nilai pada lapisan *output*.



Gambar 5. Diagram alir CNN-VGG16

Sementara pada Gambar 5, terlihat bahwa beberapa nama lapisan tidak tercantum. Hal ini bukan berarti lapisan-lapisan tersebut tidak ada, namun proses-proses yang sebelumnya telah dijelaskan akan tergabung pada satu lapisan. Sehingga satu lapisan dapat memuat lebih dari satu proses yang berlangsung. Dengan demikian, *CNN-VGG16* memiliki jumlah lapisan sebanyak 16, di mana terdiri atas 13 lapisan konvolusi dan 3 lapisan *Fully Connected* yang dapat dilihat pula pada Gambar 5 secara visualnya. Lapisan Rectified Linear Unit (ReLU) pada Gambar 5 nampak tidak ada, sebab lapisan ini sudah sekaligus dijalankan prosesnya menjadi satu pada setiap lapisan konvolusi sebelum dilanjutkan ke tahap berikutnya.

Perancangan *model CNN-VGG16* dalam penelitian ini akan menggunakan input berukuran $224 \times 224 \times 3$, yang berarti memiliki panjang 224 piksel, lebar 224 piksel, dan *depth* 3, yang mewakili warna RGB. Selanjutnya, untuk jumlah filter, digunakan 64, 128, 256, dan 512 filter. *Kernel* yang digunakan berukuran 3×3 , dan lapisan *max pooling* memiliki ukuran 2×2 . Secara lengkap, perancangan algoritma *CNN-VGG16* dalam penelitian ini dapat ditemukan dalam Tabel 2.

Model *CNN-VGG16* pada penelitian ini menggunakan *Stochastic Gradient Descent* (SGD) sebagai *optimizer* dan untuk *loss function* menggunakan *Categorical Crossentropy*. Total parameter dalam *model* ini adalah sebanyak 35.626.053. Sedangkan lapisan/*layer Batch Normalization* berguna untuk membantu menjaga stabilitas dan kecepatan konvergensi selama pelatihan data oleh model terkait. Dan untuk *layer Dropout* berguna untuk mengurangi *overfitting*, yaitu kondisi di mana model terlalu menghafal data pelatihan dan tidak dapat menggeneralisasi dengan baik pada data baru. Nilai *Dropout* yang peneliti gunakan pada penelitian ini adalah sebesar 0.25.

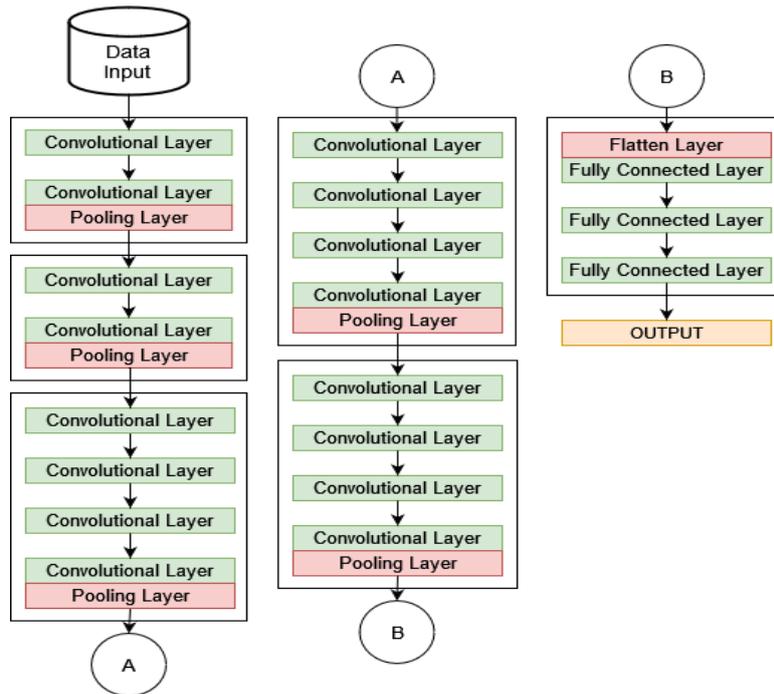
Tabel 2. Algoritma CNN-VGG16

<i>Layer</i>	<i>Output</i>
Input	224 x 224 x 3
64 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 64
64 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 64
<i>Max Pooling</i> (2x2)	224 x 244 x 64
128 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 128
128 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 128
<i>Max Pooling</i> (2x2)	224 x 244 x 128
256 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 256
256 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 256
256 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 256
<i>Max Pooling</i> (2x2)	224 x 244 x 256
256 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 256
256 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 256
256 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 256
<i>Max Pooling</i> (2x2)	224 x 244 x 256
512 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 512
512 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 512
512 Conv <i>Filter</i> dan Kernel (3x3)	224 x 244 x 512
<i>Max Pooling</i> (2x2)	224 x 244 x 512
<i>Flatten</i>	25088
<i>Dense</i>	1024
<i>Dense</i>	512
<i>Batch Normalization</i>	512
<i>Dropout</i> (rate=0.25)	512
<i>Output</i>	5

2.5.2. VGG19 Model

Proses yang terjadi dalam CNN-VGG19 pada dasarnya mengikuti prinsip yang serupa dengan CNN-VGG16 sebagaimana dijelaskan sebelumnya oleh peneliti. Namun, terdapat perbedaan signifikan pada jumlah lapisan yang dimiliki. Dalam struktur CNN-VGG19, terdapat tambahan lapisan konvolusi yang memungkinkan model untuk mengekstraksi fitur-fitur dengan tingkat kompleksitas yang lebih tinggi. CNN-VGG19 terdiri dari 19 lapisan, yang terdiri atas 16 lapisan konvolusi dan 3 lapisan *Fully Connected*. Lapisan konvolusi di dalam VGG19 berperan penting dalam mendeteksi pola-pola kompleks pada berbagai tingkat abstraksi. Selain itu, lapisan-lapisan konvolusi ini juga telah dilengkapi juga dengan *Rectified Linear Unit* (ReLU) sama seperti CNN-VGG16.

Tidak ada perbedaan yang signifikan antara perancangan model CNN-VGG16 dan CNN-VGG19 dalam penelitian ini. Perbedaan utama terletak pada jumlah lapisan yang digunakan. Model CNN-VGG19 memiliki total 19 lapisan, yang terdiri dari lapisan-lapisan yang sama seperti yang telah dijelaskan pada model CNN-VGG16 sebelumnya. Pada Tabel 3, peneliti telah paparkan secara lengkap algoritma CNN-VGG19 yang digunakan pada penelitian ini. Akibat dari perbedaan jumlah lapisan yang dimiliki, maka jumlah dari parameter model ini adalah 39.166.021.



Gambar 6. Diagram alir CNN-VGG19

Tabel 3. Algoritma CNN-VGG19

Layer	Output
Input	224 x 224 x 3
64 Conv Filter dan Kernel (3x3)	224 x 244 x 64
64 Conv Filter dan Kernel (3x3)	224 x 244 x 64
Max Pooling (2x2)	224 x 244 x 64
128 Conv Filter dan Kernel (3x3)	224 x 244 x 128
128 Conv Filter dan Kernel (3x3)	224 x 244 x 128
Max Pooling (2x2)	224 x 244 x 128
256 Conv Filter dan Kernel (3x3)	224 x 244 x 256
256 Conv Filter dan Kernel (3x3)	224 x 244 x 256
256 Conv Filter dan Kernel (3x3)	224 x 244 x 256
256 Conv Filter dan Kernel (3x3)	224 x 244 x 256
Max Pooling (2x2)	224 x 244 x 256
256 Conv Filter dan Kernel (3x3)	224 x 244 x 256
256 Conv Filter dan Kernel (3x3)	224 x 244 x 256
256 Conv Filter dan Kernel (3x3)	224 x 244 x 256
256 Conv Filter dan Kernel (3x3)	224 x 244 x 256
Max Pooling (2x2)	224 x 244 x 256
512 Conv Filter dan Kernel (3x3)	224 x 244 x 512
512 Conv Filter dan Kernel (3x3)	224 x 244 x 512
512 Conv Filter dan Kernel (3x3)	224 x 244 x 512
512 Conv Filter dan Kernel (3x3)	224 x 244 x 512
Max Pooling (2x2)	224 x 244 x 512
Flatten	25088
Dense	1024
Dense	512
Batch Normalization	512
Dropout (rate=0.25)	512
Output	5

2.6. Evaluasi

Pada tahap ini, peneliti memilih untuk menggunakan metode *Confusion Matrix* juga mencakup empat metrik evaluasi kinerja, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, untuk mengevaluasi kinerja klasifikasi pada 5 kelas berbeda yang mewakili varietas beras. Selain itu, terdapat istilah-istilah penting lainnya, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* [13], yang digunakan untuk merepresentasikan hasil dari proses pembelajaran klasifikasi. Dengan istilah-istilah ini, diharapkan dapat mengidentifikasi berapa banyak hasil prediksi yang benar (positif dan negatif) dan berapa banyak yang salah dari model yang telah melalui proses pembelajaran/*training*.

Accuracy merupakan nilai yang mengukur sejauh mana *model* klasifikasi dapat memprediksi dengan benar label kelas dari data uji. Dalam konteks klasifikasi, *accuracy* mengukur jumlah prediksi yang benar baik positif maupun negatif (*True Positive (TP)*/*True Negative (TN)*) dibagi dengan jumlah total data uji. Jika dituliskan secara matematis dapat dilihat pada persamaan 1.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision merupakan nilai untuk mengukur sejauh mana model dapat mengidentifikasi dengan benar kasus positif dari semua kasus yang diprediksi sebagai positif. Jika dituliskan secara matematis dapat dilihat pada persamaan 2.

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

Recall (sensitivitas) merupakan nilai untuk mengukur sejauh mana model dapat mengidentifikasi dengan benar semua kasus positif. Jika dituliskan secara matematis dapat dilihat pada persamaan 3.

$$recall = \frac{TP}{TP+FN} \quad (3)$$

F1-Score merupakan metrik yang menggabungkan *Precision* dan *Recall* menjadi satu angka tunggal untuk memberikan gambaran keseluruhan tentang kinerja model. Jika dituliskan secara matematis dapat dilihat pada persamaan 4.

$$f1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

2.7. Analisis Performa

Pada tahap ini, peneliti melakukan perbandingan antara nilai-nilai metrik yang telah ditemukan seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Di mana sebelumnya telah dilakukan pada tahap evaluasi selama analisis performa setiap *model*. Peneliti memulai dengan menggambarkan hasil perbandingan ini untuk memberikan pemahaman lebih lanjut tentang kinerja masing-masing *model*, baik untuk algoritma *CNN-VGG16* maupun *CNN-VGG19*.

3. Hasil dan Pembahasan

Hasil yang telah ada, diperoleh setelah melakukan beberapa langkah yang telah dipaparkan pada subbab sebelumnya. Beberapa proses yang melibatkan penggunaan bahasa pemrograman *Python* pada penelitian ini, dilakukan dengan memanfaatkan bantuan perangkat lunak/*software* bernama *Jupyter-Lab*. Peneliti mencoba membandingkan nilai hasil data *training* dengan jumlah *epoch* sebanyak 5, 10, 15, 20, 25, dan 30. Dengan tujuan untuk melihat pada jumlah *epoch* beberapa, *model* mencapai performa terbaiknya. Setelah mengetahui *model* terbaik dari setiap algoritma, selanjutnya peneliti melakukan perbandingan antara nilai-nilai dalam metrik *Confusion Matrix* dari masing-masing algoritma. Untuk mengetahui performa algoritma mana yang lebih unggul dalam mengklasifikasikan varietas beras.

3.1. Hasil Proses *Training Model***Tabel 4.** Hasil proses *training model CNN-VGG16*

<i>Epoch</i>	<i>Average duration per Step (s)</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Validation Accuracy</i>	<i>Validation Loss</i>
5	2608	0,9089	0,2900	0,5867	1,1597
10	2341	0,9350	0,1907	0,7810	0,6993
15	2341	0,9557	0,1210	0,8743	0,4647
20	2341	0,9654	0,1069	0,9200	0,2279
25	2341	0,9714	0,0796	0,9276	0,1809
30	2341	0,9757	0,0699	0,9581	0,1013

Tabel 5. Hasil proses *training model CNN-VGG19*

<i>Epoch</i>	<i>Average duration per Step (s)</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Validation Accuracy</i>	<i>Validation Loss</i>
5	2562	0,8886	0,3253	0,5962	1,2456
10	2802	0,9307	0,2046	0,7905	0,8318
15	2698	0,9525	0,1242	0,7429	0,5536
20	2668	0,9604	0,0642	0,8648	0,3102
25	2628	0,9682	0,0414	0,8705	0,3409
30	2603	0,9707	0,0317	0,9486	0,1066

Hasil proses *training* dari setiap model, baik *CNN-VGG16* dan *CNN-VGG19* pada *epoch* 5, 10, 15, 20, 25, dan 30 dapat dilihat melalui Tabel 4 dan 5. Di mana pada kedua tabel yang sebelumnya telah disebutkan, memiliki istilah yaitu *epoch*. *Epoch* merupakan satu kali proses lengkap di mana seluruh dataset pelatihan digunakan untuk melatih suatu model jaringan saraf [14]. Proses pelatihan pada umumnya melibatkan beberapa *epoch* untuk memungkinkan model mempelajari pola-pola dalam data dengan lebih baik. Sedangkan untuk *loss* mengacu pada nilai seberapa baik *model* dalam memprediksi *output* yang benar untuk *input* tertentu pada *dataset* pelatihan. *Neural network* pada dasarnya melakukan proses belajar untuk meminimalkan jarak antara label yang diprediksi dan label sebenarnya. Untuk mengurangi jarak ini, *neural network* melakukan proses belajar secara berkali-kali dengan menyesuaikan nilai bobot dan bias agar mengurangi nilai *loss* [15].

Melalui Tabel 4 di atas, nilai rata-rata lama proses *training* per-*epoch* hanya pada *epoch* 5 saja yang berbeda, yaitu sekitar 2608 detik, lebih lama dibandingkan dengan rata-rata pada jumlah *epoch* lainnya. Sementara untuk *epoch* 10, 15, 20, 25, dan 30 memiliki rata-rata yang sama, yaitu sekitar 2341 detik untuk setiap *epoch*. Tingkat akurasi *train* dan *validation* antar *epoch* tidak memiliki perbedaan yang signifikan. Namun, perlu diperhatikan bahwa tingkat *loss validation* pada *epoch* 5 memiliki kesenjangan yang cukup besar dibandingkan dengan *epoch* yang lain. Secara keseluruhan, *model CNN-VGG16* telah menunjukkan performa yang baik dalam mengklasifikasikan varietas beras. Tingkat akurasi yang tinggi pada data latih dan validasi serta penurunan *validation loss* seiring berjalannya waktu menunjukkan bahwa model memiliki potensi yang kuat untuk digunakan dalam tugas pengklasifikasian varietas beras.

Melalui Tabel 5 di atas menunjukkan hasil pelatihan *model CNN-VGG19*, yang merupakan kelanjutan dari percobaan yang telah dilakukan pada *model CNN-VGG16* sebagaimana yang telah dijelaskan dalam Tabel 4 sebelumnya. Terdapat beberapa perbedaan yang menarik antara kedua *model* ini. Dalam hal durasi per-*epoch*, *model CNN-VGG19* tidak menunjukkan konsistensi yang lebih baik daripada model sebelumnya. *CNN-VGG19* memiliki durasi yang lebih beragam selama proses pelatihan. Hal ini dapat mengindikasikan bahwa model ini tidak lebih baik pada *model CNN-VGG16*.

Pada akurasi, *model CNN-VGG19* menunjukkan kinerja yang kuat. Pada akhir pelatihan pada *epoch* ke-30, model ini mencapai akurasi yang sangat tinggi sekitar 99.07% pada data latih. Dan untuk nilai *validation loss* pada *epoch* ke-30, menunjukkan angka yang cukup rendah yaitu sekitar 0.1066.

Data yang tertulis pada Tabel 4 dan 5 di atas merupakan nilai akurasi dan *loss* yang muncul saat pertama kali *model neural network* dijalankan. Nilai-nilai tersebut terus berubah apabila proses *training* dijalankan, sebab data *training* dan *validation* yang digunakan setiap proses *training* selalu dipilih secara acak. Sebagai contoh, pada data latih pertama, data yang digunakan mungkin nomor 1, 3, 6, 9, 12, dan seterusnya. Dan untuk data tesnya mungkin nomor 21, 24, 27, 28, dan seterusnya. Pada percobaan kedua, data latih yang diambil adalah nomor 2, 5, 8, 11, dan seterusnya. Sedangkan untuk data tesnya 8, 11, 15, 18, dan seterusnya. Hal ini yang membuat data latih tidak beraturan atau acak. Nilai akurasi dan *loss* dapat lebih tinggi dan bahkan tidak menutup kemungkinan dapat lebih rendah dibandingkan dengan proses sebelumnya [16].

3.2. Hasil *Confusion Matrix* dan *Classification Report Model*

3.2.1. *Confusion Matrix Model*

Berikut merupakan nilai dari *confusion matrix* kedua model yang sebelumnya telah dilakukan proses *training* yang disajikan dengan menggunakan tabel, yang dapat dilihat pula pada Tabel 6. Pada Tabel 6, disajikan nilai dari *confusion matrix* untuk dua model yang diuji, yaitu *VGG16* dan *VGG19*. Matriks kebingungannya menggambarkan kinerja model dalam mengklasifikasikan sampel-sampel ke dalam kelas yang benar. Kedua model, baik *VGG16* maupun *VGG19*, diperoleh hasil yang menunjukkan performa yang sangat baik. Dari matriks kebingungannya, terlihat bahwa kedua model tidak menghasilkan *False Positive* (FP) atau *False Negative* (FN) untuk kelas apapun. Hal ini dapat diartikan bahwa tidak ada kasus di mana model memprediksi kelas yang sebenarnya salah. Selain itu, keduanya memiliki nilai *True Positive* (TP) yang tinggi, menandakan bahwa kedua model sangat mampu mengidentifikasi dan mengklasifikasikan sampel yang sebenarnya termasuk dalam kelas tertentu. Demikian pula, *True Negative* (TN) juga menunjukkan bahwa kedua model dapat membedakan dengan benar sampel yang tidak termasuk dalam kelas tertentu. Secara keseluruhan, hasil dari *confusion matrix* menunjukkan bahwa baik model *VGG16* maupun *VGG19* memberikan performa yang baik dalam tugas klasifikasi yang diuji. Kedua model ini memiliki tingkat akurasi yang tinggi dan mampu membedakan dengan baik antara berbagai kelas yang diuji.

Tabel 6. Nilai *confusion matrix model*

	<i>VGG16</i>	<i>VGG19</i>
True Positive (TP)	34	35
False Positive (FP)	0	0
False Negative (FN)	0	0
True Negative (TN)	34	35

Tabel 7. *Classification report model CNN-VGG16*

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
'Arborio' : 0	1.00	0.97	0.99	35
'Basmati' : 1	0.94	0.97	0.96	35
'Ipsala' : 2	1.00	1.00	1.00	35
'Jasmine' : 3	0.97	0.94	0.96	35
'Karacadag' : 4	0.97	1.00	0.99	35
accuracy			0.98	175
macro avg	0.98	0.98	0.98	175
weighted avg	0.98	0.98	0.98	175

Tabel 8. Classification report model CNN-VGG19

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
'Arborio' : 0	0.97	1.00	0.99	35
'Basmati' : 1	0.90	1.00	0.95	35
'Ipsala' : 2	1.00	1.00	1.00	35
'Jasmine' : 3	1.00	0.89	0.94	35
'Karacadag' : 4	1.00	0.97	0.99	35
accuracy			0.97	175
macro avg	0.97	0.97	0.97	175
weighted avg	0.97	0.97	0.97	175

3.2.2. Classification Report Model

Setelah melakukan proses *training* pada setiap *model CNN* dan mengamati hasil pada Tabel 4 dan 5, serta melihat nilai-nilai *confusion matrix* pada Tabel 6 di atas, nilai yang akan digunakan adalah hasil dari *training* pada *epoch* ke-30, baik pada *model CNN-VGG16* maupun *CNN-VGG19*. Berikut ini adalah *classification report* dari setiap *model* menggunakan bantuan *library 'sklearn.metrics'* yang terdapat pada Tabel 7 dan 8 di bawah ini.

Tabel 7 menunjukkan nilai-nilai dari variabel *precision*, *recall*, *f1-score*, dan *support* untuk *model VGG16*. Beberapa kelas menunjukkan performa yang baik karena nilainya mendekati angka 1.00 dan bahkan ada juga pada nilai-nilai di kelas tertentu yang memiliki nilai 1.00. Dapat dilihat bahwa beberapa kelas memiliki performa yang sangat baik, dengan nilai *precision* dan *recall* mendekati 1.00. Sebagai contoh, kelas 'Ipsala' memiliki *precision*, *recall*, dan *f1-score* sebesar 1.00, menunjukkan bahwa *model* dengan sempurna mengenali dan mengklasifikasikan semua sampel dari kelas tersebut. Hasil yang baik juga terlihat pada kelas 'Arborio' dengan nilai *precision* sekitar 1.00 dan *recall* sekitar 0,97. Secara keseluruhan, *model CNN-VGG16* menunjukkan performa yang sangat baik dengan akurasi sekitar 0.98, yang berarti sekitar 98% dari prediksi *model* benar. Hasil ini menunjukkan bahwa *model* ini efektif dalam mengklasifikasikan berbagai varietas beras.

Tabel 8 menunjukkan hasil dari evaluasi klasifikasi untuk *model CNN-VGG19*. Pada keseluruhan performa *model*, dapat dilihat bahwa *model* ini juga cukup memberikan hasil yang baik. Beberapa kelas menunjukkan performa yang hampir sempurna, terutama 'Ipsala' yang memiliki nilai *precision*, *recall*, dan *f1-score* sebesar 1.00. Dapat diartikan bahwa, *model* dengan sempurna mengenali dan mengklasifikasikan semua sampel dari kelas tersebut. Kelas 'Arborio' dan 'Basmati' juga memiliki hasil yang sangat baik dengan nilai *recall* dan *f1-score* mendekati 1.00. Namun, ada beberapa kelas seperti 'Jasmine' dan 'Karacadag' yang memiliki perbedaan antara *precision* dan *recall*. Ini menandakan bahwa *model* cenderung lebih baik dalam mengklasifikasikan kelas-kelas lain dibandingkan dengan kelas 'Jasmine' dan 'Karacadag'. Kelas 'Jasmine' memiliki *recall* sedikit lebih rendah, sehingga *model* mungkin membutuhkan peningkatan dalam mengidentifikasi semua kasus 'Jasmine'. Secara keseluruhan, akurasi *model* ini adalah sekitar 0.97, yang berarti sekitar 97% dari prediksi *model* benar. Hasil ini mengindikasikan bahwa *model CNN-VGG19* cukup efektif dalam mengklasifikasikan berbagai varietas beras.

3.3. Analisis Performa Model

Tabel 9. Perbandingan nilai *precision* model VGG16 dan VGG19

	VGG16	VGG19	Gap
'Arborio' : 0	1.00	0.97	0.3
'Basmati' : 1	0.94	0.90	0.4
'Ipsala' : 2	1.00	1.00	0
'Jasmine' : 3	0.97	1.00	0.3
'Karacadag' : 4	0.97	1.00	0.3

Tabel 10. Perbandingan nilai *recall* model VGG16 dan VGG19

	VGG16	VGG19	Gap
'Arborio' : 0	0.97	1.00	0.3
'Basmati' : 1	0.97	1.00	0.3
'Ipsala' : 2	1.00	1.00	0
'Jasmine' : 3	0.94	0.89	0.5
'Karacadag' : 4	1.00	0.97	0.3

Setelah memeriksa performa dari kedua model, seperti yang dapat diamati pada Tabel 7 dan 8 di atas, peneliti akan melakukan analisis lebih lanjut dengan membandingkan nilai-nilai dari *confusion matrix*. *Confusion matrix* meliputi metrik-metrik penting seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Dengan membandingkan nilai-nilai dari *confusion matrix* ini, peneliti dapat membuat keputusan yang lebih terarah mengenai pilihan *model* terbaik untuk tugas klasifikasi jenis padi ini. Hal ini penting untuk memastikan bahwa *model* yang dipilih mampu memberikan hasil yang konsisten dan akurat dalam aplikasi praktis.

Dalam tabel 9, terdapat perbandingan nilai *precision* antara *model* VGG16 dan VGG19 untuk lima kelas beras yang berbeda. Dalam konteks ini, nilai *precision* berkisar antara 0 hingga 1, di mana apabila nilai semakin mendekati nilai 1.00 menunjukkan kinerja yang lebih baik. Berdasarkan hasil yang tercatat dalam tabel, dapat diamati bahwa pada semua kelas beras, *model* VGG19 menunjukkan nilai *precision* yang lebih tinggi dibandingkan dengan VGG16. Sebagai contoh, untuk kelas 'Arborio', VGG16 memiliki nilai *precision* sebesar 0.97, sedangkan VGG19 memiliki nilai *precision* 1.00. Hal serupa juga terlihat pada kelas 'Basmati', di mana VGG16 memiliki *precision* sekitar 0.94 sedangkan VGG19 memiliki nilai yang lebih tinggi yaitu 0.90. Fenomena ini juga teramati pada kelas-kelas lainnya.

Tabel 11. Perbandingan nilai *f1-score* model VGG16 dan VGG19

	VGG16	VGG19	Gap
'Arborio' : 0	0.99	0.99	0
'Basmati' : 1	0.96	0.95	0.1
'Ipsala' : 2	1.00	1.00	0
'Jasmine' : 3	0.96	0.94	0.2
'Karacadag' : 4	0.99	0.99	0

Tabel 12. Perbandingan nilai *accuracy* model VGG16 dan VGG19

	VGG16	VGG19
<i>accuracy</i>	0.98	0.97

Hal ini mengindikasikan bahwa *model VGG19* cenderung memberikan hasil yang lebih akurat dalam mengklasifikasikan varietas beras yang diuji dalam eksperimen ini jika dibandingkan dengan *VGG16*. Oleh karena itu, berdasarkan nilai *precision*, dapat disimpulkan bahwa *model VGG19* menunjukkan performa yang lebih baik dibandingkan dengan *VGG16* dalam tugas klasifikasi varietas beras ini.

Dalam Tabel 10, terdapat perbandingan nilai *recall* antara *model VGG16* dan *VGG19* untuk lima kelas beras yang berbeda. Seperti *precision*, nilai *recall* juga berkisar antara 0 hingga 1, dan semakin mendekati 1 menunjukkan kinerja yang lebih baik. Berdasarkan hasil yang tercatat dalam tabel, dapat diamati bahwa kinerja *VGG16* dan *VGG19* berbeda tergantung pada kelas beras yang diuji. Sebagai contoh, pada kelas 'Jasmine', *VGG16* memiliki nilai *recall* sebesar 0.94, sementara *VGG19* memiliki nilai *recall* 0.89. Hal ini menunjukkan bahwa *VGG16* mampu mengenali lebih banyak kasus positif 'Jasmine' dibandingkan *VGG19*. Namun, pada kelas 'Basmati' dan 'Ipsala', kedua model menunjukkan hasil *recall* yang serupa, yaitu 1.00, menunjukkan kemampuan yang sangat baik dalam mengenali kasus positif. Khusus untuk kelas 'Karacadag', *VGG16* memiliki *recall* sekitar 1.00 sedangkan *VGG19* memiliki *recall* 0.97. Meskipun keduanya memiliki kinerja yang tinggi, *VGG16* mungkin sedikit lebih unggul dalam mengenali kasus positif untuk kelas ini.

Secara keseluruhan, berdasarkan nilai *recall*, tidak ada *model* yang secara konsisten lebih unggul dibandingkan yang lain. Performa keduanya cenderung serupa dan tergantung pada kelas beras yang diuji. Tetapi apabila dilihat pada *gap* dari nilai *recall* kedua *model*, *VGG16* dapat dikatakan memiliki performa yang lebih baik, walaupun hanya sedikit *gap* nilainya.

Dalam Tabel 11, terdapat perbandingan nilai *f1-score* antara *model VGG16* dan *VGG19* untuk lima kelas padi yang berbeda. Berdasarkan hasil yang tercatat dalam tabel, dapat diamati bahwa performa *VGG16* dan *VGG19* tampak sangat serupa dalam klasifikasi jenis padi ini. Secara keseluruhan, tidak terdapat perbedaan yang signifikan antara keduanya dalam hal *f1-score* untuk setiap kelas. Semua *gap* nilai *f1-score* antara keduanya adalah kurang dari 0.2, menunjukkan bahwa keduanya memiliki kinerja yang serupa. Walau dengan kondisi seperti ini, *gap* yang dimiliki kedua model juga dapat terlihat jelas bahwa nilai-nilai *f1-score* yang lebih mendekati angka satu pada setiap kelasnya adalah *model VGG16*. Maka dari itu, dapat disimpulkan bahwa *model VGG16* pada nilai *f1-score* lebih baik dibandingkan dengan *model VGG19*.

Dalam Tabel 12, terdapat perbandingan nilai akurasi (*accuracy*) antara *model VGG16* dan *VGG19*. Akurasi mengukur sejauh mana *model* dapat melakukan klasifikasi dengan benar secara keseluruhan. Berdasarkan hasil yang tercatat dalam tabel, dapat diamati bahwa *VGG16* memiliki tingkat akurasi sebesar 98%, sementara *VGG19* memiliki tingkat akurasi 97%. Hal ini menunjukkan bahwa *VGG16* memiliki sedikit keunggulan dalam hal akurasi dibandingkan dengan *VGG19*.

4. Diskusi

Berdasarkan hasil dan pembahasan terkait perbandingan performa antara algoritma *VGG16* dan *VGG19* dalam proses mengklasifikasikan varietas beras dengan lima kelas, dapat disimpulkan bahwa keduanya menghasilkan hasil yang memuaskan. Proses *training model Convolutional Neural Network (CNN)* dilakukan menggunakan *dataset* dengan perbandingan 8:1.5:0.5, atau dengan total 3500 citra gambar. Hasil evaluasi menunjukkan bahwa algoritma *VGG16* mencapai tingkat akurasi sebesar 98% dengan waktu *training* selama 73,405 detik. Sementara itu, algoritma *VGG19* mencapai tingkat akurasi sebesar 97% dengan waktu *training* 78,098 detik. Selain akurasi, nilai-nilai metrik lain seperti *precision*, *recall*, dan *f1-score* juga dievaluasi. Algoritma *VGG19* menonjol pada nilai *precision*, yang mengukur akurasi dari hasil klasifikasi positif terhadap seluruh prediksi positif yang sebenarnya. Namun, dalam hal *recall* dan *f1-score*, algoritma *VGG16* menunjukkan performa lebih baik. Di mana *recall* merupakan nilai yang mengukur kemampuan model untuk menemukan kembali seluruh kasus positif yang sebenarnya dan untuk *f1-score* merupakan metrik yang menilai keseimbangan antara *precision* dan *recall* dari suatu *model*.

Hasil penelitian menunjukkan bahwa tidak selalu arsitektur yang memiliki lapisan yang lebih kompleks atau jumlah lapisan yang lebih banyak akan memberikan performa yang lebih baik. Dalam

penelitian ini, VGG16 terbukti memiliki performa yang lebih unggul dibandingkan dengan VGG19. Faktor-faktor beragam dapat mempengaruhi hasil akhir perbandingan dua algoritma ini. Salah satu faktor yang paling berpengaruh adalah ketersediaan data atau dataset seperti yang disebutkan oleh peneliti sebelumnya. Jumlah data memiliki dampak yang signifikan terhadap pilihan algoritma yang tepat. Oleh karena itu, dalam kasus ini VGG16 dapat dianggap lebih dapat diandalkan dengan jumlah data sekitar 3500 citra gambar, karena memiliki kecenderungan *overfitting* yang lebih rendah dibandingkan dengan VGG19.

5. Kesimpulan

Hasil penelitian ini menunjukkan bahwa dalam kasus mengklasifikasikan varietas beras dengan lima kelas, algoritma VGG16 telah terbukti lebih unggul daripada algoritma VGG19. Meskipun *model* VGG16 memiliki jumlah lapisan yang lebih sedikit dibandingkan VGG19, namun hal ini tidak selalu menjadi indikator mutlak. Pada kasus ini, VGG16 menonjol dengan akurasi mencapai 98%, sedangkan VGG19 mencapai akurasi sebesar 97%. Selain *accuracy*, kinerja model juga diukur dengan metrik-metrik lain seperti *precision*, *recall*, dan *f1-score*. Meskipun VGG19 menunjukkan keunggulan pada nilai *precision*, VGG16 tampil lebih baik dalam hal *recall* dan *f1-score*. Keunggulan performa VGG16 mungkin disebabkan oleh kompleksitas model yang lebih rendah dan kemampuannya untuk menghindari *overfitting* pada dataset yang tersedia. Jumlah lapisan yang lebih sedikit mungkin memungkinkan VGG16 untuk lebih efisien dalam mempelajari fitur-fitur penting dari gambar beras. Oleh karena itu, dalam konteks penelitian ini, VGG16 terbukti menjadi pilihan arsitektur yang lebih optimal dibandingkan dengan VGG19.

Referensi

- [1] A. R. Sari, Y. Martono, and F. S. Rondonuwu, "Identifikasi Kualitas Beras Putih (*Oryza sativa* L.) Berdasarkan Kandungan Amilosa dan Amilopektin di Pasar Tradisional dan 'Selepan' Kota Salatiga," *Titian Ilmu J. Ilm. Multi Sci.*, vol. 12, no. 1, pp. 24–30, 2020, doi: 10.30599/jti.v12i1.599.
- [2] H. Windiyani, "Keragaman Varietas Unggul Baru Padi Fungsional Mendukung Ketahanan Pangan dalam Menghadapi Pandemi COVID-19," in *Komoditas Sumber Pangan untuk Meningkatkan Kualitas Kesehatan di Era Pandemi Covid-19*, 2020, pp. 449–456. [Online]. Available: <http://conference.unsri.ac.id/index.php/lahansuboptimal/article/view/1964/0>
- [3] G. Prayitno, *Ruang Berketahanan Pangan: Menjawab Tantangan Produksi Pangan Berkelanjutan dengan Optimasi Keruangan Menuju Indonesia Berdaulat*, 1st ed. Malang: UB Press, 2022. [Online]. Available: <https://edeposit.perpusnas.go.id/collection/ruang-berketahanan-pangan-sumber-elektronis-menjawab-tantangan-produksi-pangan-berkelanjutan-dengan-optimasi-keruangan-menuju-indonesia-berdaulat/112492#>
- [4] A. Ahdiat, "Indonesia Impor Beras 429 Ribu Ton Pada 2022, Ini Negara Asalnya," 2023. <https://databoks.katadata.co.id/datapublish/2023/04/14/indonesia-impor-beras-429-ribu-ton-pada-2022-ini-negara-asalnya#:~:text=Menurut laporan Badan Pusat Statistik,Myanmar seperti terlihat pada grafik.>
- [5] D. Iswanto and D. Handayani UN, "Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN)," *J. Ilm. Univ. Batanghari Jambi*, vol. 22, no. 2, p. 900, 2022, doi: 10.33087/jiubj.v22i2.2065.
- [6] F. S. Pamungkas, B. D. Prasetya, and I. Kharisudin, "Perbandingan Metode Klasifikasi Supervised Learning pada Data Bank Customers Menggunakan Python," *Prism. Pros. Semin. Nas. Mat.*, vol. 3, pp. 689–694, 2020, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/prisma/article/view/37875>
- [7] A. Mulyanto, E. Susanti, F. Rossi, W. Wajiran, and R. I. Borman, "Penerapan Convolutional Neural

- Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR)," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 1, p. 52, 2021, doi: 10.26418/jp.v7i1.44133.
- [8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, no. 6, pp. 1–14, Sep. 2014, [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [9] E. Tanuwijaya and A. Roseanne, "Modifikasi Arsitektur VGG16 untuk Klasifikasi Citra Digital Rempah-Rempah Indonesia," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 1, pp. 189–196, 2021, doi: 10.30812/matrik.v21i1.1492.
- [10] M. Koklu, I. Cinar, and Y. S. Taspinar, "Classification of rice varieties with deep learning methods," *Comput. Electron. Agric.*, vol. 187, p. 106285, Aug. 2021, doi: 10.1016/j.compag.2021.106285.
- [11] A. Huda, *DASAR-DASAR PEMROGRAMANBERBASIS PYTHON*, 1st ed. Padang: Tim editor UNP Press, 2020.
- [12] M. Ezar Al Rivani and A. Giovri Riyadi, "Perbandingan Arsitektur LeNet dan AlexNet Pada Metode Convolutional Neural Network Untuk Pengenalan American Sign Language," *J. Komput. Terap.*, vol. 7, no. Vol. 7 No. 1 (2021), pp. 53–61, 2021, doi: 10.35143/jkt.v7i1.4489.
- [13] T. Dwi Antoko, M. Azhar Ridani, and A. Eko Minarno, "Klasifikasi Buah Zaitun Menggunakan Convolution Neural Network," *Komputika J. Sist. Komput.*, vol. 10, no. 2, pp. 119–126, 2021, doi: 10.34010/komputika.v10i2.4475.
- [14] F. Masykur, M. B. Setyawan, and K. Winangun, "Epoch Optimization on Rice Leaf Image Classification Using Convolutional Neural Network (CNN) MobileNet," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 7, no. 2, p. 581, 2022, doi: 10.24114/cess.v7i2.37336.
- [15] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized Cut Loss for Weakly-Supervised CNN Segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1818–1827, 2018, doi: 10.1109/CVPR.2018.00195.
- [16] A. Khamparia, D. Gupta, V. H. C. de Albuquerque, A. K. Sangaiah, and R. H. Jhaveri, "Internet of health things-driven deep learning system for detection and classification of cervical cells using transfer learning," *J. Supercomput.*, vol. 76, no. 11, pp. 8590–8608, Nov. 2020, doi: 10.1007/s11227-020-03159-4.

