

OPTIMASI HYPERPARAMETER GRID SEARCH DAN RANDOM SEARCH PADA INCEPTION V3 UNTUK KEMATANGAN BUAH KELAPA SAWIT

Ramadani Zikri^{*1)}, Andi Sunyoto²⁾, Ainul Yaqin³⁾

1. Universitas Amikom, Yogyakarta
2. Universitas Amikom, Yogyakarta
3. Universitas Amikom, Yogyakarta

Article Info

Kata Kunci: Klasifikasi; Inception V3; Grid Search; Random Search; Kelapa sawit.

Keywords: Classification; Inception V3; Grid Search; Random Search; Oil Palm Fruit.

Article history:

Received 26 Agustus 2024

Revised 12 September 2024

Accepted 20 October 2024

Available online 1 December 2025

DOI :

<https://doi.org/10.29100/jipi.v10i4.6537>

* Corresponding author.

Ramadani Zikri

E-mail address:

rzikri@students.amikom.ac.id

ABSTRAK

Penelitian ini bertujuan untuk mengklasifikasikan tingkat kematangan buah kelapa sawit menggunakan model Inception V3 dengan pendekatan hyperparameter tuning menggunakan Grid Search dan Random Search. Dataset yang digunakan terdiri dari tiga kelas kematangan buah kelapa sawit: belum masak, masak, dan terlalu masak. Penelitian ini melibatkan beberapa kombinasi hyperparameter, termasuk learning rate, batch size, dan jumlah epoch, yang diuji untuk memperoleh konfigurasi terbaik. Hasil menunjukkan bahwa kombinasi terbaik dari Grid Search mencapai akurasi sebesar 91%, sedangkan Random Search menghasilkan akurasi tertinggi sebesar 92%. Evaluasi dilakukan melalui analisis confusion matrix, yang menunjukkan bahwa model mampu mengklasifikasikan tingkat kematangan buah dengan tingkat kesalahan yang rendah. Berdasarkan hasil ini, disimpulkan bahwa penggunaan metode hyperparameter tuning sangat penting dalam meningkatkan akurasi prediksi model. Penelitian ini juga memberikan rekomendasi untuk eksplorasi lebih lanjut dengan arsitektur model yang lebih kompleks serta data yang lebih bervariasi untuk meningkatkan generalisasi model.

ABSTRACT

This study aims to classify the ripeness levels of oil palm fruits using the Inception V3 model with hyperparameter tuning approaches through Grid Search and Random Search. The dataset used consists of three ripeness classes: unripe, ripe, and overripe. The research involved testing several hyperparameter combinations, including learning rate, batch size, and the number of epochs, to obtain the best configuration. The results showed that the best combination from Grid Search achieved an accuracy of 91%, while Random Search produced the highest accuracy of 92%. Evaluation was conducted using confusion matrix analysis, which demonstrated that the model was able to classify the ripeness levels with low error rates. Based on these findings, it is concluded that the use of hyperparameter tuning methods is crucial in improving model prediction accuracy. This study also provides recommendations for further exploration with more complex model architectures and more diverse data to enhance model generalization.

I. PENDAHULUAN

Klasifikasi kematangan buah kelapa sawit merupakan elemen krusial dalam industri kelapa sawit untuk menjamin kualitas hasil panen serta efisiensi proses produksi. Penentuan tingkat kematangan yang tepat sangat penting dalam menjaga mutu produk dan memaksimalkan hasil panen. Namun, proses klasifikasi ini tidak lepas dari berbagai tantangan, terutama dalam penerapannya menggunakan model deep learning. Salah satu kendala utama adalah variasi kondisi pencahayaan dan posisi buah dalam dataset yang digunakan untuk melatih model. Variasi ini seringkali menyebabkan model kesulitan menghasilkan klasifikasi yang konsisten. Sebagai contoh, perbedaan pencahayaan atau sudut pengambilan gambar dapat menyebabkan klasifikasi yang tidak akurat meskipun tingkat kematangan buah tersebut sama [1-4]. Oleh karena itu, teknik pre-processing yang efektif menjadi penting untuk mengatasi masalah ini, sehingga akurasi dan konsistensi klasifikasi dapat ditingkatkan [2], [4-10].

Penelitian ini menawarkan inovasi dalam bentuk pengembangan teknik pre-processing baru yang mencakup

augmentasi data adaptif dan normalisasi yang dirancang khusus untuk menangani variasi pencahayaan dan posisi buah. Selain itu, arsitektur model deep learning yang digunakan akan dioptimalkan untuk efisiensi komputasi, terutama untuk implementasi pada perangkat mobile. Pendekatan ini tidak hanya bertujuan untuk meningkatkan akurasi klasifikasi, tetapi juga untuk memastikan bahwa model dapat dijalankan secara efisien pada perangkat dengan daya komputasi terbatas [3], [11-13].

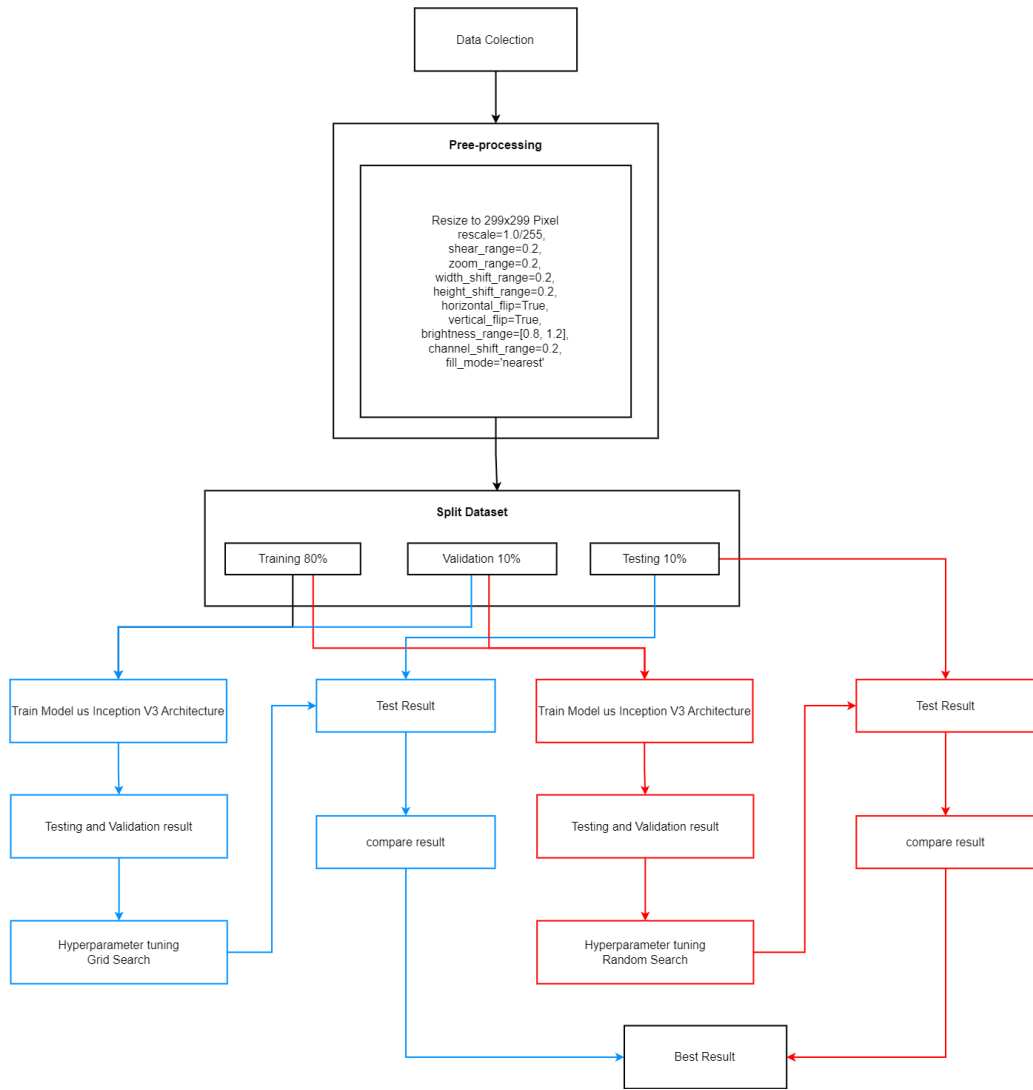
Selain itu, model deep learning juga menghadapi kesulitan dalam membedakan tingkat kematangan buah yang memiliki kemiripan visual, seperti antara buah yang matang dan terlalu matang. Tantangan ini sering kali disebabkan oleh kesamaan visual yang tinggi antara kelas-kelas tersebut, yang mengakibatkan model lebih mudah melakukan kesalahan klasifikasi [4-5], [14]. Pendekatan yang diusulkan dalam penelitian ini mencoba untuk mengatasi masalah ini dengan menggunakan teknik augmentasi data yang dirancang untuk meningkatkan variasi data secara artifisial, sehingga model dapat belajar lebih baik dalam membedakan kelas-kelas dengan kemiripan visual tinggi.

Lebih lanjut, keterbatasan daya komputasi dan kebutuhan memori pada perangkat mobile menjadi masalah tambahan, mengingat model deep learning yang kompleks biasanya memerlukan sumber daya yang besar. Oleh karena itu, arsitektur model Inception V3 dipilih dan dioptimalkan untuk efisiensi, dengan pengurangan lapisan yang tidak perlu dan penerapan teknik fine-tuning untuk meminimalkan penggunaan sumber daya tanpa mengorbankan akurasi [6], [15-18]. Namun, implementasi klasifikasi kematangan buah kelapa sawit pada perangkat mobile sangat penting dalam operasi lapangan yang membutuhkan solusi cepat dan akurat. Oleh karena itu, diperlukan pengembangan model yang ringan dan efisien tanpa mengorbankan akurasi klasifikasi [3], [6], [19-20].

Penelitian ini mencoba menjawab tantangan-tantangan tersebut dengan menggabungkan dan menyeimbangkan dataset dari dua jurnal utama, yakni "Annotated Datasets of Oil Palm Fruit Bunch Piles for Ripeness Grading Using Deep Learning" [4] dan "Outdoor Oil Palm Fruit Ripeness Dataset" [5]. Dalam prosesnya, kelas-kelas dengan data yang tidak mencukupi dihapus, sehingga setiap kelas memiliki jumlah data yang seimbang. Selain itu, teknik augmentasi data seperti rotasi, flipping, dan penyesuaian kecerahan diterapkan untuk meningkatkan variabilitas dataset, yang diharapkan dapat membantu model dalam mengenali pola dengan lebih baik. Dengan pendekatan ini, model deep learning yang dikembangkan diharapkan mampu melakukan klasifikasi kematangan buah dengan akurasi yang lebih tinggi dan generalisasi yang lebih baik. Teknik-teknik ini dipilih karena kemampuannya untuk mengatasi variasi pencahayaan dan posisi buah yang sering menjadi tantangan utama dalam klasifikasi citra. Selain itu, pengembangan model yang efisien diharapkan dapat memberikan solusi yang lebih optimal, khususnya dalam implementasi pada perangkat mobile, sehingga dapat mendukung peningkatan kualitas dan efisiensi dalam industri kelapa sawit.

II. METODE PENELITIAN

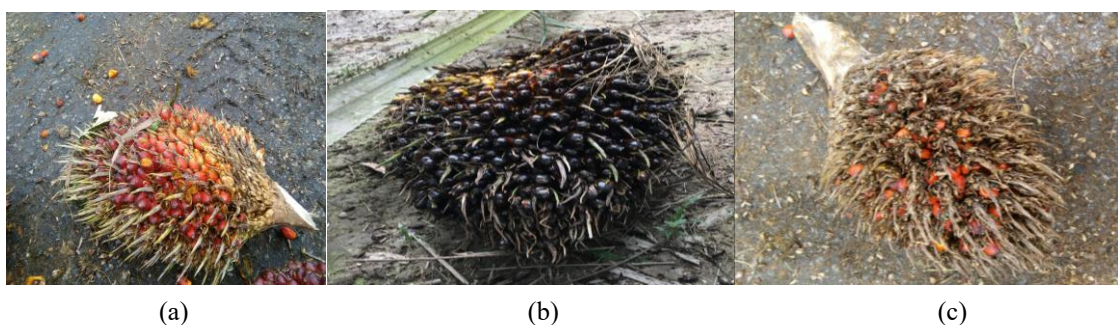
Penelitian ini menggunakan metode eksperimental yang berfokus pada klasifikasi citra buah kelapa sawit menggunakan model deep learning, khususnya arsitektur Inception V3. Langkah-langkah penelitian ini dijabarkan sebagai berikut:



Gambar 1. Tahap Penelitian

A. Pengumpulan Dataset

Pada tahap pertama, data yang digunakan dalam penelitian ini adalah data gambar buah kelapa sawit yang diperoleh dari dua dataset utama yaitu *Annotated Datasets of Oil Palm Fruit Bunch Piles for Ripeness Grading Using Deep Learning*[4] dan *Outdoor Oil Palm Fruit Ripeness Dataset*[5]. Kedua dataset ini digabungkan secara manual oleh peneliti untuk mendapatkan data yang relevan dan representatif. Proses penggabungan dilakukan dengan pemilihan data yang sesuai dengan kriteria penelitian, sehingga data yang digunakan bisa mendukung hasil klasifikasi yang lebih akurat. Data yang digunakan sebanyak tiga kelas yaitu buah kelapa sawit masak berwarna merah jingga atau oranye terang dan tekstur kulit lunak, kurang masak biasanya berwarna hijau gelap atau hitam kemerahan pada permukaan kulitnya dan biasanya posisinya sangat kuat menempel bagian tandan dan terlalu masak berwarna merah kehitaman teksturnya sangat lunak dan cenderung membusuk posisi buah dengan tandan biasanya sudah banyak berjatuhan dan hampir mengalami kebusukan, yang dapat dilihat pada gambar 2.



(a) Gambar 2. Gambar buah kelapa sawit a) masak b) kurang masak c) terlalu masak

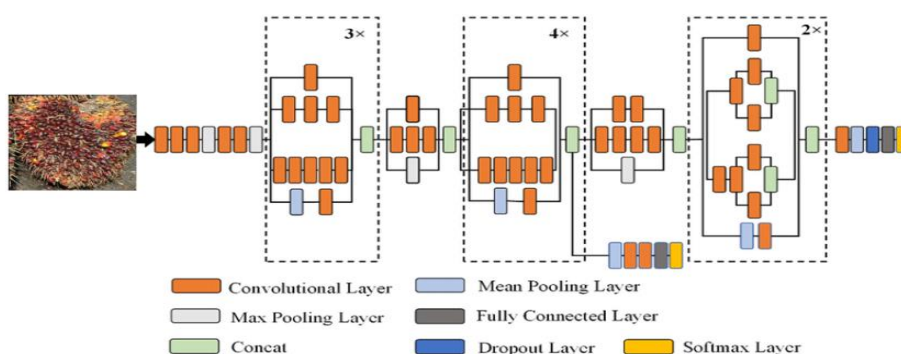
Bagian selanjutnya dikarenakan dibagi menjadi tiga bagian yaitu *training*, *validasi* dan *testing* data digabungkan berdasarkan kategori jenis buah untuk mempermudah tahapan selanjutnya. Oleh karena itu data yang digunakan pada kategori masak yaitu 440 gambar, buah kelapa sawit mentah yaitu 440 gambar dan buah kelapa sawit terlalu masak sebanyak 440 gambar dengan total keseluruhan dataset yaitu 1320 gambar buah kelapa sawit.

B. Pre Procressing Data

Pada tahap pra-pemrosesan, gambar-gambar yang telah dikumpulkan diubah ukurannya menjadi 299x299 piksel untuk memastikan konsistensi input ke dalam model Inception V3. Arsitektur Inception V3 dipilih karena telah terbukti efektif dalam berbagai tugas klasifikasi citra, khususnya dalam mendeteksi objek dengan berbagai skala dan aspek rasio. Dengan menggunakan modul 'Inception', arsitektur ini dapat memproses informasi pada skala yang berbeda secara paralel, yang sangat berguna dalam mengklasifikasikan citra buah kelapa sawit yang memiliki variasi ukuran dan bentuk. Selain itu, Inception V3 memiliki efisiensi komputasi yang baik, menjadikannya pilihan ideal untuk implementasi pada perangkat dengan daya komputasi terbatas, seperti perangkat mobile. Selain itu, dilakukan juga augmentasi data seperti rotasi, flipping, zooming, dan penyesuaian kecerahan. Untuk rotasi, digunakan parameter rotasi acak hingga 40 derajat; flipping dilakukan secara horizontal; zooming diterapkan dengan skala antara 80%-120%; dan penyesuaian kecerahan dilakukan dengan variasi intensitas antara 70%-130%. Teknik-teknik augmentasi ini dipilih untuk meningkatkan robustnes model terhadap variasi pencahayaan dan sudut pandang, yang umum terjadi dalam lingkungan pengambilan gambar di lapangan. Dengan melakukan augmentasi ini, diharapkan model dapat mengenali pola lebih baik, meskipun terdapat perubahan kondisi visual pada data yang diujikan. Setelah itu, dataset dibagi menjadi tiga bagian: 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Pembagian ini dilakukan untuk memastikan bahwa model tidak hanya belajar dari data pelatihan, tetapi juga mampu memprediksi dengan akurat pada data yang belum pernah dilihat sebelumnya [19], [21-23].

C. Pelatihan Model

Model Inception V3 digunakan sebagai arsitektur dasar untuk klasifikasi gambar buah kelapa sawit. Pada tahap pelatihan, model dilatih menggunakan dua pendekatan hyperparameter tuning yaitu Grid Search dan Random Search. Hyperparameter yang diuji meliputi learning rate, batch size, dan jumlah epoch. Rentang nilai yang dipertimbangkan untuk learning rate adalah antara 0.0001 hingga 0.001, untuk batch size adalah 32 hingga 64, dan untuk jumlah epoch adalah 30 hingga 50. Pemilihan kombinasi hyperparameter terbaik didasarkan pada akurasi model pada data validasi. Grid Search mencoba semua kombinasi hyperparameter yang telah ditentukan, sementara Random Search memilih kombinasi secara acak dalam rentang yang lebih luas. Dalam Grid Search, semua kombinasi yang mungkin dari parameter yang ditentukan diuji secara sistematis, sedangkan Random Search memilih kombinasi acak dari rentang yang lebih luas, yang memungkinkan untuk menemukan kombinasi optimal dengan lebih cepat. Tujuan dari kedua metode ini adalah untuk menemukan kombinasi hyperparameter yang memberikan performa terbaik dalam klasifikasi [15], [24].



Gambar 3. Arsitektur Inception V3

D. Hyperparameter Deep Learning

Dalam deep learning, hyperparameter adalah variabel yang menentukan struktur jaringan saraf dan proses pelatihannya. Hyperparameter ini berbeda dari parameter yang dipelajari langsung oleh model, seperti bobot dan bias, karena mereka diatur oleh pengguna sebelum pelatihan dimulai. Contoh hyperparameter penting dalam deep learning meliputi jumlah hidden layer, batch size, learning rate, dan jumlah epoch [25-27].

E. Grid Search

Grid Search adalah metode optimasi hyperparameter yang bekerja dengan mencoba setiap kombinasi yang mungkin dari parameter yang telah ditentukan sebelumnya. Pendekatan ini sangat sistematis dan memastikan bahwa semua kombinasi diuji untuk menemukan yang terbaik berdasarkan metrik evaluasi seperti akurasi atau loss. Kelebihan utama dari Grid Search adalah cakupan pencarian yang menyeluruh, yang memungkinkan identifikasi kombinasi hyperparameter yang optimal. Namun, kelemahannya adalah waktu komputasi yang sangat besar, terutama ketika ruang pencarian hyperparameter besar dan kompleks [28].

F. Random Search

Random Search adalah metode optimasi hyperparameter yang memilih kombinasi hyperparameter secara acak dari ruang pencarian. Meskipun tidak mencoba setiap kombinasi, Random Search sering kali lebih efisien dan cepat karena hanya perlu mencoba sebagian dari keseluruhan ruang pencarian untuk menemukan kombinasi yang baik. Penelitian menunjukkan bahwa dalam banyak kasus, Random Search dapat menemukan kombinasi hyperparameter yang optimal dengan lebih cepat dibandingkan Grid Search, terutama ketika ruang pencarian hyperparameter sangat besar dan hanya beberapa hyperparameter yang memiliki pengaruh signifikan terhadap kinerja model. Hal ini menjadikannya pilihan yang lebih hemat waktu dan sumber daya dibandingkan Grid Search, terutama untuk model dan dataset yang besar [27].

G. Evaluasi dan Pengujian

Setelah proses pelatihan selesai, model dievaluasi menggunakan data pengujian yang terpisah. Hasil pengujian ini dibandingkan untuk menentukan metode tuning mana yang menghasilkan performa terbaik. Evaluasi dilakukan dengan mengukur akurasi model pada data pengujian, yang kemudian digunakan untuk menyimpulkan efektivitas pendekatan yang digunakan dalam penelitian ini [24], [26], [29].

III. HASIL DAN PEMBAHASAN

A. Hasil Pelatihan model

Pada kombinasi hyperparameter yang diuji menggunakan *Grid Search*, model Inception V3 menghasilkan performa yang bervariasi. Hal ini dapat dilihat pada Gambar 4 di mana parameter training accuracy dan validation accuracy mengalami fluktuasi pada setiap epoch. Terdapat beberapa kombinasi yang menunjukkan tanda-tanda overfitting atau underfitting.

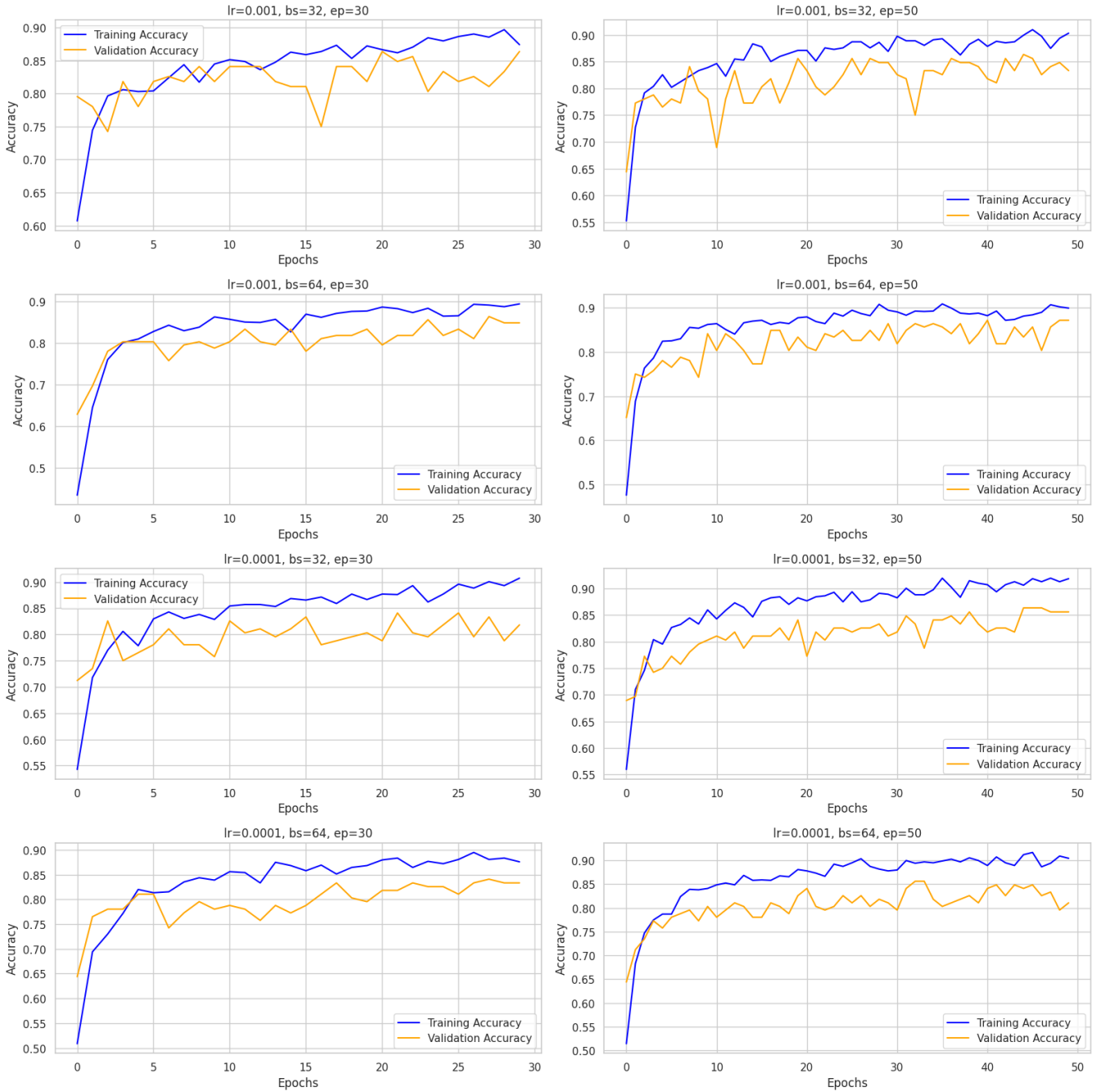
Misalnya, pada kombinasi dengan learning rate 0.001, batch size 32, epochs 30, model menghasilkan akurasi validasi yang lebih rendah dibandingkan dengan akurasi pelatihan, menunjukkan bahwa model mungkin mengalami overfitting. Overfitting ini mungkin disebabkan oleh beberapa faktor seperti ukuran dataset yang mungkin tidak cukup besar untuk mendukung pembelajaran model yang kompleks, atau kompleksitas model itu sendiri yang terlalu tinggi untuk dataset ini. Selain itu, kurangnya variasi dalam data training juga bisa menjadi penyebab overfitting, di mana model menjadi sangat spesifik terhadap data training dan tidak dapat melakukan generalisasi dengan baik pada data baru.

Namun, ada beberapa kombinasi yang menunjukkan hasil yang lebih stabil dan baik, seperti pada *learning rate 0.0001, batch size 64, epochs 50*. Pada kombinasi ini, perbedaan antara akurasi pelatihan dan validasi tidak terlalu besar, dan loss pada kedua data menunjukkan tren yang stabil tanpa fluktuasi yang signifikan. Hal ini menunjukkan bahwa hyperparameter yang dipilih mampu menjaga keseimbangan antara underfitting dan overfitting, di mana model tidak terlalu sederhana atau terlalu kompleks untuk dataset ini. Teknik augmentasi data yang digunakan juga berperan dalam membantu meningkatkan variasi data dan mencegah overfitting.

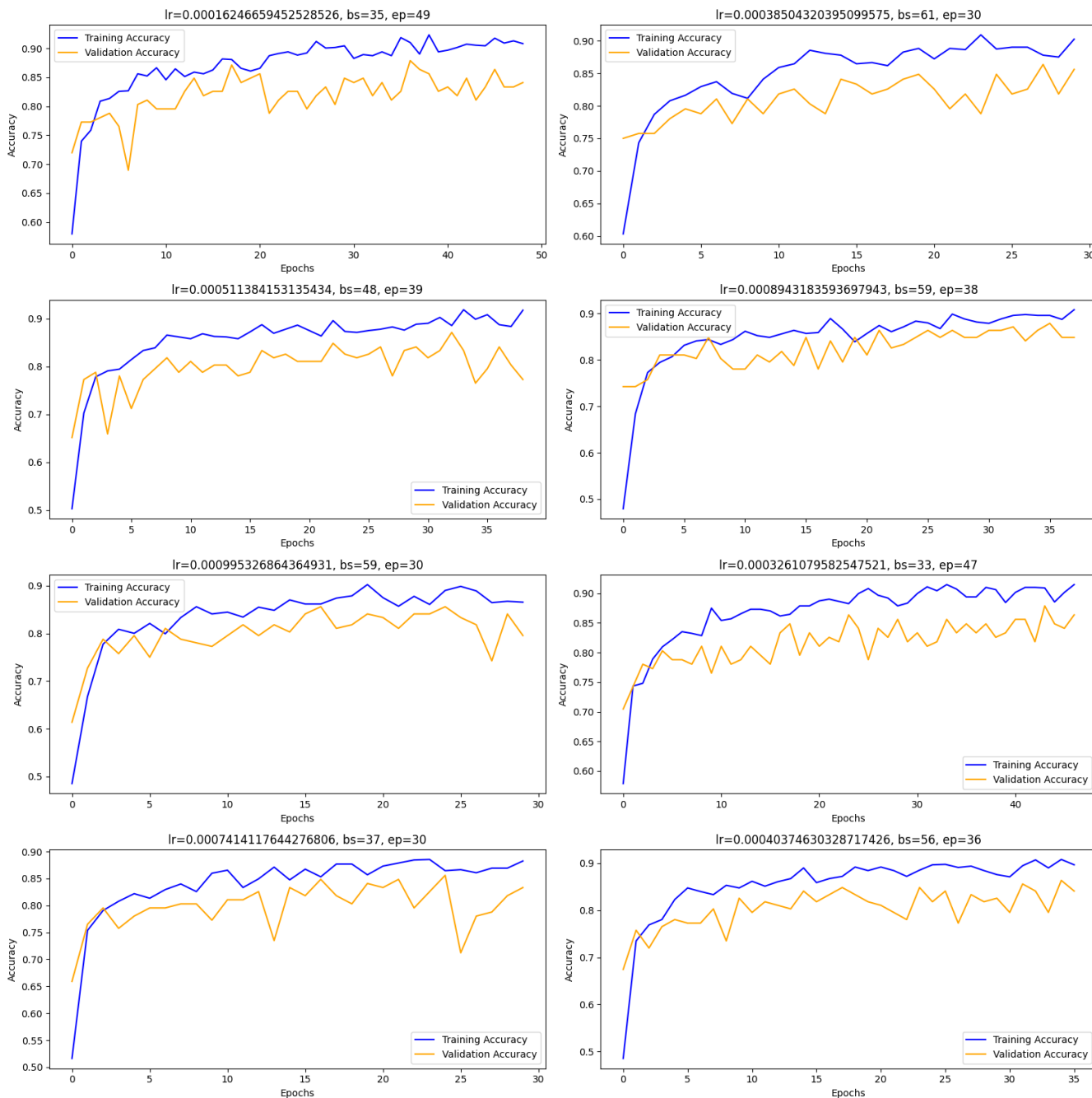
Secara keseluruhan, hasil Grid Search ini menunjukkan bahwa pemilihan hyperparameter yang tepat sangat penting untuk mendapatkan model yang baik dan menghindari masalah overfitting atau underfitting. Beberapa kombinasi hyperparameter menghasilkan model dengan performa yang lebih konsisten, yang dapat diandalkan untuk prediksi pada data baru.

Pada kombinasi hyperparameter yang diuji menggunakan *Random Search*, model Inception V3 juga menghasilkan performa yang bervariasi. Hal ini dapat dilihat pada Gambar 5 di mana parameter training accuracy dan validation accuracy mengalami fluktuasi pada setiap epoch. Beberapa kombinasi menunjukkan tanda-tanda *overfitting*, terutama pada kombinasi dengan learning rate yang lebih kecil dan batch size yang lebih besar. Misalnya, pada kombinasi $lr=0.0003261079852547521$, $bs=33$, $epochs=47$, model menunjukkan akurasi training yang lebih tinggi dibandingkan akurasi validasi, yang menandakan bahwa model mungkin mengalami overfitting. Hal ini bisa disebabkan oleh batch size yang lebih besar, yang membuat model belajar terlalu banyak detail dari data training, serta learning rate yang terlalu kecil yang menyebabkan model terlalu lama dalam mencapai konvergensi, sehingga overfitting terjadi.

Di sisi lain, terdapat beberapa kombinasi yang menunjukkan performa yang lebih stabil, seperti $lr=0.00038504320395099575$, $bs=61$, $epochs=30$, di mana akurasi training dan validasi tidak memiliki perbedaan yang signifikan. Ini menunjukkan bahwa kombinasi hyperparameter tersebut mampu menjaga stabilitas antara akurasi training dan validasi, yang penting untuk menghindari overfitting. Hasil dari Random Search ini memberikan wawasan bahwa memilih hyperparameter secara acak dapat memberikan variasi hasil yang cukup luas, dan dalam beberapa kasus, dapat menemukan kombinasi yang lebih optimal dibandingkan Grid Search.



Gambar 4. Visualisasi akurasi training setiap kombinasi grid search



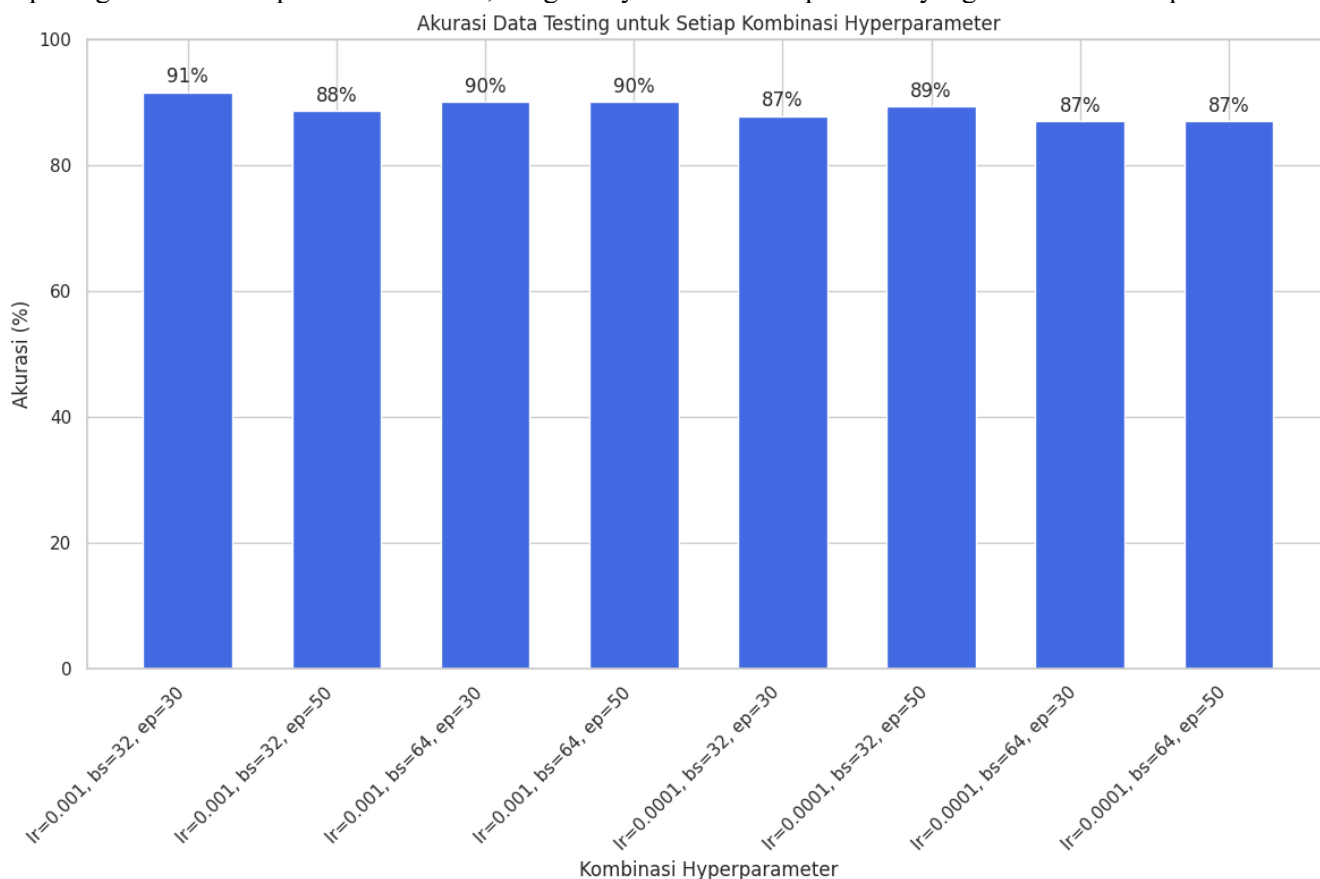
Gambar 5. Visualisasi akurasi raining setiap kombinasi random search

B. Hasil Evaluasi Model

Pada tahap evaluasi model, akurasi dari setiap kombinasi hyperparameter yang diuji dengan Grid Search dianalisis untuk menentukan performa terbaik dari model yang telah dilatih. Berdasarkan grafik yang ditampilkan, terlihat bahwa kombinasi hyperparameter yang berbeda memberikan variasi akurasi yang tidak terlalu signifikan, dengan kisaran akurasi antara 87% hingga 91%. Ini menunjukkan bahwa model relatif stabil terhadap perubahan hyperparameter, meskipun ada beberapa kombinasi yang memberikan hasil lebih baik dibandingkan yang lain. Stabilitas ini penting karena menunjukkan bahwa model dapat menangani variasi data dengan baik, tanpa terlalu banyak terpengaruh oleh perubahan kecil dalam parameter.

Kombinasi hyperparameter terbaik dengan akurasi 91% menggunakan *learning rate 0.001*, *batch size 32*, dan *30 epochs*. Evaluasi ini menegaskan bahwa penyesuaian batch size dan jumlah epoch memiliki dampak yang cukup besar terhadap hasil akhir model. Batch size yang lebih kecil memungkinkan model untuk memperbarui bobotnya lebih sering dengan lebih banyak iterasi, sementara jumlah epoch yang tepat membantu memastikan bahwa model tidak terlalu lama dilatih yang dapat menyebabkan *overfitting*. Selain itu, kombinasi hyperparameter dengan batch size yang lebih besar atau jumlah epoch yang lebih tinggi cenderung menunjukkan performa yang sedikit lebih rendah, yang mungkin disebabkan oleh *overfitting* atau kurangnya variasi dalam data yang dilatih.

Secara keseluruhan, hasil evaluasi model dengan Grid Search memperlihatkan bahwa pendekatan ini dapat membantu mengidentifikasi pengaturan hyperparameter yang paling optimal. Grid Search memiliki kelebihan dalam memberikan pemahaman yang lebih sistematis mengenai pengaruh setiap hyperparameter, namun kelemahannya adalah membutuhkan banyak waktu dan sumber daya komputasi terutama jika ruang pencarian sangat besar. Meski begitu, evaluasi ini juga menekankan pentingnya keseimbangan antara parameter yang digunakan, karena perubahan kecil dapat menghasilkan variasi dalam akurasi. Model yang dihasilkan dari kombinasi terbaik ini kemudian dapat digunakan untuk prediksi data baru, dengan keyakinan bahwa performa yang dihasilkan cukup konsisten.

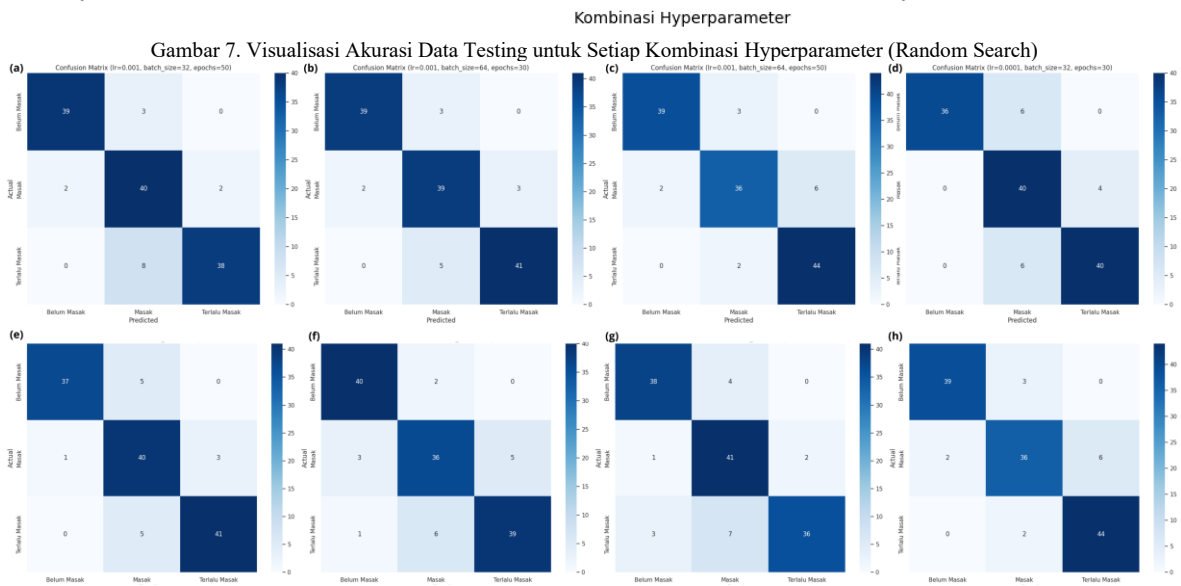
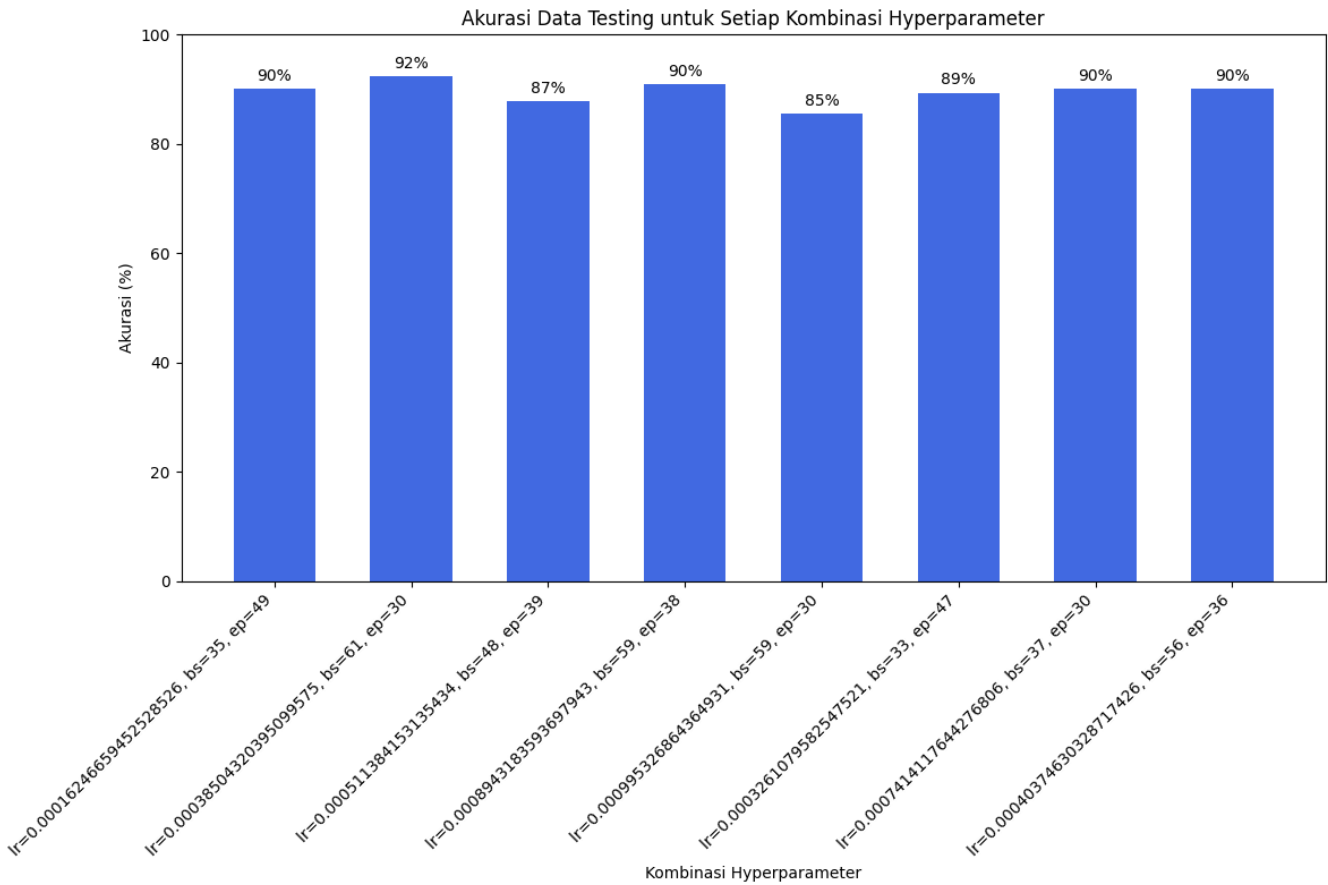


Gambar 6. Hasil evaluasi model Grid Search

Pada tahap evaluasi model dengan metode Random Search, berbagai kombinasi hyperparameter diuji untuk mendapatkan performa terbaik pada data testing. Grafik berikut menampilkan hasil akurasi dari setiap kombinasi hyperparameter yang diuji.

Dari hasil evaluasi, dapat dilihat bahwa kombinasi hyperparameter dengan *learning rate* 0.0003840320395099576, *batch size* 61, dan 30 *epochs* menghasilkan akurasi tertinggi sebesar 92%. Random Search mampu menemukan kombinasi hyperparameter yang lebih optimal meskipun hanya menggunakan sebagian kecil dari ruang pencarian yang mungkin. Hasil ini menunjukkan bahwa Random Search dapat menemukan kombinasi hyperparameter yang lebih optimal dibandingkan dengan Grid Search, meskipun dengan jumlah iterasi yang lebih sedikit. Namun, kekurangannya adalah Random Search tidak menjamin akan mengeksplorasi seluruh kombinasi yang mungkin secara sistematis seperti Grid Search. Selain itu, akurasi model tetap konsisten di sekitar angka 85%-92%, menunjukkan stabilitas model terhadap perubahan hyperparameter yang signifikan.

Dengan membandingkan hasil ini dengan Grid Search, dapat disimpulkan bahwa Random Search dapat menjadi alternatif yang efisien untuk pencarian hyperparameter, terutama ketika ruang pencarian sangat besar dan sumber daya komputasi terbatas.

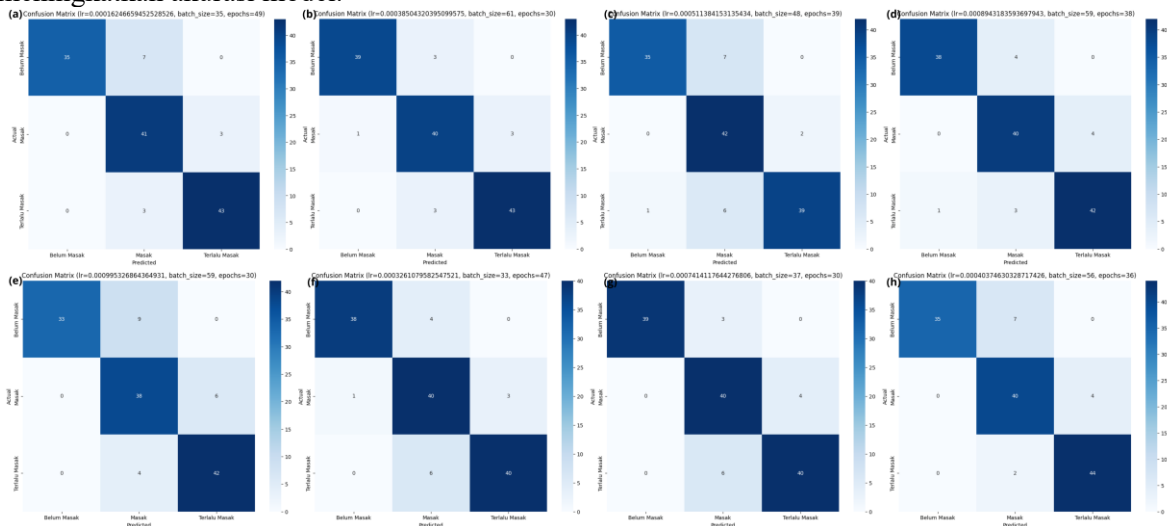


Gambar 8. Hasil Confusion Matrix Setiap Kombinasi Grid Search

Model dengan *learning rate* 0.001, *batch size* 32, dan 30 *epochs* pada Gambar 8a menunjukkan performa yang cukup baik dengan hanya sedikit kesalahan prediksi. Misalnya, terdapat 38 prediksi benar untuk kelas "Belum Masak", 40 untuk "Masak", dan 43 untuk "Terlalu Masak". Namun, ada beberapa kesalahan seperti 4 buah "Belum Masak" yang diprediksi sebagai "Masak". Pada Gambar 8b, performa model sedikit lebih baik dengan peningkatan jumlah prediksi yang benar untuk "Belum Masak". Namun, terdapat beberapa prediksi "Terlalu Masak" yang masih salah diklasifikasikan sebagai "Masak". Gambar 8c menunjukkan akurasi yang cukup stabil di ketiga kelas, meskipun masih ada beberapa kesalahan prediksi antara "Masak" dan "Terlalu Masak". Gambar 8d menunjukkan kesalahan yang serupa dengan kombinasi sebelumnya, namun dengan sedikit peningkatan di kelas "Masak". Gambar 8e dengan *learning rate* yang lebih kecil, model ini menunjukkan sedikit penurunan akurasi di kelas "Belum Masak", namun tetap memiliki prediksi yang cukup baik di kelas lainnya. Gambar 8f pada kombinasi ini, ada sedikit perbaikan di prediksi kelas "Belum Masak" dibandingkan dengan kombinasi sebelumnya, namun masih ada beberapa kesalahan di kelas lainnya. Gambar 8g menunjukkan bahwa model memiliki keseimbangan yang baik antara akurasi di semua kelas, meskipun masih terdapat kesalahan kecil di kelas "Masak". Gambar 8h menunjukkan

model mengalami sedikit kesalahan prediksi, khususnya pada kelas "Masak" yang terkadang diklasifikasikan sebagai "Terlalu Masak".

Secara keseluruhan, confusion matrix memberikan pandangan yang lebih mendalam mengenai bagaimana model menangani prediksi untuk setiap kelas. Dari hasil ini, kita dapat melihat bahwa meskipun model umumnya memiliki akurasi tinggi, ada beberapa kelas yang cenderung lebih sulit diprediksi dengan benar, seperti kesalahan klasifikasi antara "Masak" dan "Terlalu Masak". Hal ini bisa menjadi area untuk perbaikan lebih lanjut dalam penelitian masa depan, misalnya dengan menambahkan lebih banyak data pelatihan atau menggunakan teknik augmentasi data untuk meningkatkan akurasi model.



Gambar 9. Hasil Confusion Matrix Setiap Kombinasi Random Search

Pada Gambar 9a hingga 9h, masing-masing confusion matrix menunjukkan performa model dalam mengklasifikasikan tiga kategori buah kelapa sawit, yaitu belum masak, masak, dan terlalu masak, berdasarkan kombinasi hyperparameter yang berbeda pada Random Search. Gambar 9a ($lr=0.00016246659452528526$, $batch_size=35$, $epochs=49$) menunjukkan kesalahan kecil dalam klasifikasi buah belum masak dan masak. Gambar 9b ($lr=0.00038504320395099575$, $batch_size=61$, $epochs=30$) memperlihatkan peningkatan akurasi pada buah belum masak. Gambar 9c ($lr=0.000511384153135434$, $batch_size=48$, $epochs=39$) menunjukkan sedikit peningkatan kesalahan pada buah terlalu masak. Gambar 9d ($lr=0.0008943183593697943$, $batch_size=59$, $epochs=38$) memperlihatkan hasil yang lebih konsisten dengan penurunan kesalahan pada klasifikasi buah masak dan terlalu masak. Gambar 9e ($lr=0.000995326864364931$, $batch_size=59$, $epochs=30$) mengalami sedikit penurunan akurasi dalam klasifikasi buah belum masak. Gambar 9f ($lr=0.0003261079582547521$, $batch_size=33$, $epochs=47$) memiliki hasil yang lebih baik dengan penurunan kesalahan pada buah masak, tetapi tetap ada sedikit kesalahan pada buah belum masak. Gambar 9g ($lr=0.0007414117644276806$, $batch_size=37$, $epochs=30$) menunjukkan penurunan akurasi terutama pada buah masak dan terlalu masak. Terakhir, Gambar 9h ($lr=0.00040374630328717426$, $batch_size=56$, $epochs=36$) memperlihatkan perbaikan dalam klasifikasi buah terlalu masak dengan pengurangan kesalahan klasifikasi untuk buah belum masak.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Dalam penelitian ini, telah dilakukan eksplorasi dan pengujian berbagai kombinasi hyperparameter menggunakan Grid Search dan Random Search untuk mengklasifikasikan kematangan buah kelapa sawit menggunakan model Inception V3. Hasil evaluasi menunjukkan bahwa kombinasi hyperparameter tertentu dapat memberikan akurasi yang lebih baik dalam memprediksi tingkat kematangan buah kelapa sawit. Pada umumnya, model yang menggunakan Grid Search dan Random Search menunjukkan performa yang memadai, dengan beberapa variasi dalam akurasi tergantung pada kombinasi hyperparameter yang digunakan. Confusion matrix yang dihasilkan menunjukkan bahwa model secara konsisten dapat mengklasifikasikan buah kelapa sawit dengan tingkat kesalahan yang rendah.

B. Saran

Untuk penelitian selanjutnya, disarankan untuk mengeksplorasi arsitektur model yang lebih kompleks atau menggunakan metode ensemble untuk meningkatkan akurasi prediksi. Selain itu, penggunaan data yang lebih

bervariasi dan representatif dari berbagai kondisi lingkungan dan varietas kelapa sawit dapat membantu dalam memperbaiki generalisasi model. Penerapan teknik augmentasi data yang lebih agresif juga dapat menjadi strategi yang efektif dalam menghadapi keterbatasan dataset.

DAFTAR PUSTAKA

- [1] C. L. Chew, B. A. Tan, J. Y. S. Low, N. I. N. Mohd Hakimi, S. F. Kua, and C. M. Lim, "Exogenous ethylene application on postharvest oil palm fruit bunches improves crude palm oil quality," *Food Sci Nutr*, vol. 9, no. 10, pp. 5335–5343, Oct. 2021, doi: 10.1002/fsn3.2423.
- [2] Suharjito, G. N. Elwirehardja, and J. S. Prayoga, "Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches," *Comput Electron Agric*, vol. 188, Sep. 2021, doi: 10.1016/j.compag.2021.106359.
- [3] Herman, A. Susanto, T. W. Cenggoro, D. Ariansyah, and B. Pardamean, "DEEP LEARNING WITH AUXILIARY LEARNING ATTENTION MECHANISM FOR OIL PALM FRUIT IMAGE RIPENESS CLASSIFICATION," *ICIC Express Letters*, vol. 17, no. 12, pp. 1303–1310, Dec. 2023, doi: 10.24507/icieel.17.12.1303.
- [4] Suharjito *et al.*, "Annotated Datasets of Oil Palm Fruit Bunch Piles for Ripeness Grading Using Deep Learning," *Sci Data*, vol. 10, no. 1, Dec. 2023, doi: 10.1038/s41597-023-01958-x.
- [5] Z. Omar, A. P.P. Abdul Majeed, M. Rosbi, S. A. Ghazalli, and H. Selamat, "Outdoor oil palm fruit ripeness dataset," *Data Brief*, vol. 55, Aug. 2024, doi: 10.1016/j.dib.2024.110667.
- [6] E. Salim and Suharjito, "Hyperparameter optimization of YOLOv4 tiny for palm oil fresh fruit bunches maturity detection using genetics algorithms," *Smart Agricultural Technology*, vol. 6, Dec. 2023, doi: 10.1016/j.atech.2023.100364.
- [7] A. Indrawati, A. Rahman, E. Pane, and Muhathir, "Classification of Diseases in Oil Palm Leaves Using the GoogLeNet Model," *Baghdad Science Journal*, vol. 20, pp. 2508–2520, 2023, doi: 10.21123/bsj.2023.8547.
- [8] N. S. Tajudin, M. H. Musa, I. A. Seman, and C. N. A. Che Amri, "Quantifying spatial variability of soil and leaf nitrogen, phosphorous and potassium of basal stem rot infected oil palms using geospatial information system," *J Oil Palm Res*, vol. 32, no. 3, pp. 427–438, Sep. 2020, doi: 10.21894/jopr.2020.0047.
- [9] E. P. Atika, A. Sunyoto, and E. T. Luthfi, "Genetic Algorithm and K-Nearest Neighbors for Oil Palm Leaf Disease Classification," in *ICOIACT 2022 - 5th International Conference on Information and Communications Technology: A New Way to Make AI Useful for Everyone in the New Normal Era, Proceeding*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 447–451. doi: 10.1109/ICOIACT55506.2022.9971854.
- [10] H. Hamdani, A. Septiariini, A. Sunyoto, S. Suyanto, and F. Utaminigrum, "Detection of oil palm leaf disease based on color histogram and supervised classifier," *Optik (Stuttg)*, vol. 245, Nov. 2021, doi: 10.1016/j.jilleo.2021.167753.
- [11] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, and A. Gertych, "Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization," *Heliyon*, vol. 10, no. 5, Mar. 2024, doi: 10.1016/j.heliyon.2024.e26586.
- [12] G. Wang, Y. Sun, and J. Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning," *Comput Intell Neurosci*, vol. 2017, 2017, doi: 10.1155/2017/2917536.
- [13] Q. Zeng, X. Ma, B. Cheng, E. Zhou, and W. Pang, "GANS-based data augmentation for citrus disease severity detection using deep learning," *IEEE Access*, vol. 8, pp. 172882–172891, 2020, doi: 10.1109/ACCESS.2020.3025196.
- [14] N. D. Roslan, I. N. A. M. Azni, and S. Sundram, "MOLECULAR IDENTIFICATION AND PHYLOGENETIC ANALYSIS OF FUNGI AND BACTERIA ASSOCIATED TO COMMON SPEAR ROT DISEASE IN MALAYSIA," *J Oil Palm Res*, vol. 34, no. 3, pp. 580–589, Sep. 2022, doi: 10.21894/jopr.2021.0039.
- [15] M. Ahmed, N. Afreen, M. Ahmed, M. Sameer, and J. Ahamed, "An inception V3 approach for malware classification using machine learning and transfer learning," *International Journal of Intelligent Networks*, vol. 4, pp. 11–18, Jan. 2023, doi: 10.1016/j.ijin.2022.11.005.
- [16] Y. A. Bezabh, A. M. Ayalew, B. M. Abuhayi, T. N. Demlie, E. A. Awoke, and T. E. Mengistu, "Classification of mango disease using ensemble convolutional neural network," *Smart Agricultural Technology*, vol. 8, Aug. 2024, doi: 10.1016/j.atech.2024.100476.
- [17] I. Ahmad, M. Hamid, S. Yousaf, S. T. Shah, and M. O. Ahmad, "Optimizing pretrained convolutional neural networks for tomato leaf disease detection," *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/8812019.
- [18] J. H. Ong, P. Ong, and W. K. Lee, "Image-based Oil Palm Leaf Disease Detection using Convolutional Neural Network," *Journal of Information and Communication Technology*, vol. 21, no. 3, pp. 383–410, Jul. 2022, doi: 10.32890/jict2022.21.4.
- [19] S. Puttinaovarat, S. Chai-Arayalert, and W. Saetang, "Oil Palm Bunch Ripeness Classification and Plantation Verification Platform: Leveraging Deep Learning and Geospatial Analysis and Visualization," *ISPRS Int J Geoinf*, vol. 13, no. 5, May 2024, doi: 10.3390/ijgi13050158.
- [20] A. Darwish, D. Ezzat, and A. E. Hassanien, "An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis," *Swarm Evol Comput*, vol. 52, Feb. 2020, doi: 10.1016/j.swevo.2019.100616.
- [21] S. W. Chun *et al.*, "Deep learning algorithm development for early detection of Botrytis cinerea infected strawberry fruit using hyperspectral fluorescence imaging," *Postharvest Biol Technol*, vol. 214, Aug. 2024, doi: 10.1016/j.postharvbio.2024.112918.
- [22] C. C. Olisah *et al.*, "Convolutional neural network ensemble learning for hyperspectral imaging-based blackberry fruit ripeness detection in uncontrolled farm environment," *Eng Appl Artif Intell*, vol. 132, Jun. 2024, doi: 10.1016/j.engappai.2024.107945.
- [23] A. Sainani, P. R. Anish, V. Joshi, and S. Ghaisas, "Extracting and Classifying Requirements from Software Engineering Contracts," in *Proceedings of the IEEE International Conference on Requirements Engineering*, IEEE Computer Society, Aug. 2020, pp. 147–157. doi: 10.1109/RE48521.2020.00026.
- [24] T. Islam, M. S. Hafiz, J. R. Jim, M. M. Kabir, and M. F. Mridha, "A systematic review of deep learning data augmentation in medical imaging: Recent advances and future research directions," Jun. 01, 2024, *Elsevier Inc*. doi: 10.1016/j.health.2024.100340.
- [25] I. Jamalededdin, R. El ayachi, and M. Biniz, "An improved approach to Arabic news classification based on hyperparameter tuning of machine learning algorithms," *Journal of Engineering Research (Kuwait)*, vol. 11, no. 2, Jun. 2023, doi: 10.1016/j.jer.2023.100061.
- [26] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, and A. Gertych, "Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization," *Heliyon*, vol. 10, no. 5, Mar. 2024, doi: 10.1016/j.heliyon.2024.e26586.
- [27] B. Xi *et al.*, "Modeling and random search optimization for the polysilicon CVD reactor," *Results in Control and Optimization*, vol. 13, Dec. 2023, doi: 10.1016/j.rico.2023.100320.
- [28] M. Ogunsanya, J. Isichei, and S. Desai, "Manufacturing Letters Grid Search Hyperparameter Tuning in Additive Manufacturing Processes-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>) Peer-review under responsibility of the Scientific Committee of the NAMRI/SME," 2023. [Online]. Available: www.sciencedirect.com
- [29] M. A. Islam, S. I. Rashid, N. U. I. Hossain, R. Fleming, and A. Sokolov, "An integrated convolutional neural network and sorting algorithm for image classification for efficient flood disaster management," *Decision Analytics Journal*, vol. 7, Jun. 2023, doi: 10.1016/j.dajour.2023.100225.