

TESIS

**OPTIMASI HYPERPARAMETER METODE CONVOLUTIONAL
NEURAL NETWORK DENGAN TRANSFER LEARNING UNTUK
KLASIFIKASI TINGKAT KEMATANGAN TANDAN BUAH SEGAR
KELAPA SAWIT**



Disusun oleh:

Nama : Jefri Zulkarnain
NIM : 21.51.2092
Konsentrasi : Digital Transformation Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2023

TESIS

**OPTIMASI HYPERPARAMETER METODE CONVOLUTIONAL
NEURAL NETWORK DENGAN TRANSFER LEARNING UNTUK
KLASIFIKASI TINGKAT KEMATANGAN TANDAN BUAH SEGAR
KELAPA SAWIT**

**HYPERPARAMETER OPTIMIZATION OF CONVOLUTIONAL
NEURAL NETWORK METHOD WITH TRANSFER LEARNING FOR
CLASSIFICATION OF MATURITY LEVELS OF OIL PALM FRESH
FRUIT BUNCHES**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Jefri Zulkarnain
NIM : 21.51.2092
Konsentrasi : Digital Transformation Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2023

HALAMAN PENGESAHAN

OPTIMASI HYPERPARAMETER METODE CONVOLUTIONAL NEURAL NETWORK DENGAN TRANSFER LEARNING UNTUK KLASIFIKASI TINGKAT KEMATANGAN TANDAN BUAH SEGAR KELAPA SAWIT

HYPERPARAMETER OPTIMIZATION OF CONVOLUTIONAL NEURAL NETWORK METHOD WITH TRANSFER LEARNING FOR CLASSIFICATION OF MATURITY LEVELS OF OIL PALM FRESH FRUIT BUNCHES

Dipersiapkan dan Disusun oleh

Jefri Zulkarnain

21.51.2092

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Jumat, 1 September 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 1 September 2023

Rektor

Prof. Dr. M. Suvanto, M.M.

NIK. 190302001

HALAMAN PERSETUJUAN

OPTIMASI HYPERPARAMETER METODE CONVOLUTIONAL NEURAL NETWORK DENGAN TRANSFER LEARNING UNTUK KLASIFIKASI TINGKAT KEMATANGAN TANDAN BUAH SEGAR KELAPA SAWIT

HYPERPARAMETER OPTIMIZATION OF CONVOLUTIONAL NEURAL NETWORK METHOD WITH TRANSFER LEARNING FOR CLASSIFICATION OF MATURITY LEVELS OF OIL PALM FRESH FRUIT BUNCHES

Dipersiapkan dan Disusun oleh

Jefri Zulkarnain

21.51.2092

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Jumat, 1 September 2023

Pembimbing Utama

Prof. Dr. Kusriani, M.Kom
NIK. 190302106

Anggota Tim Penguji

Dr. Arief Setyanto, S.Si., M.T.
NIK. 190302036

Pembimbing Pendamping

Tonny Hidayat, M.Kom., Ph.D
NIK. 190302182

Dr. Ferry Wahyu Wibowo, S.Si., M.Cs.
NIK. 190302235

Prof. Dr. Kusriani, M.Kom
NIK. 190302106

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

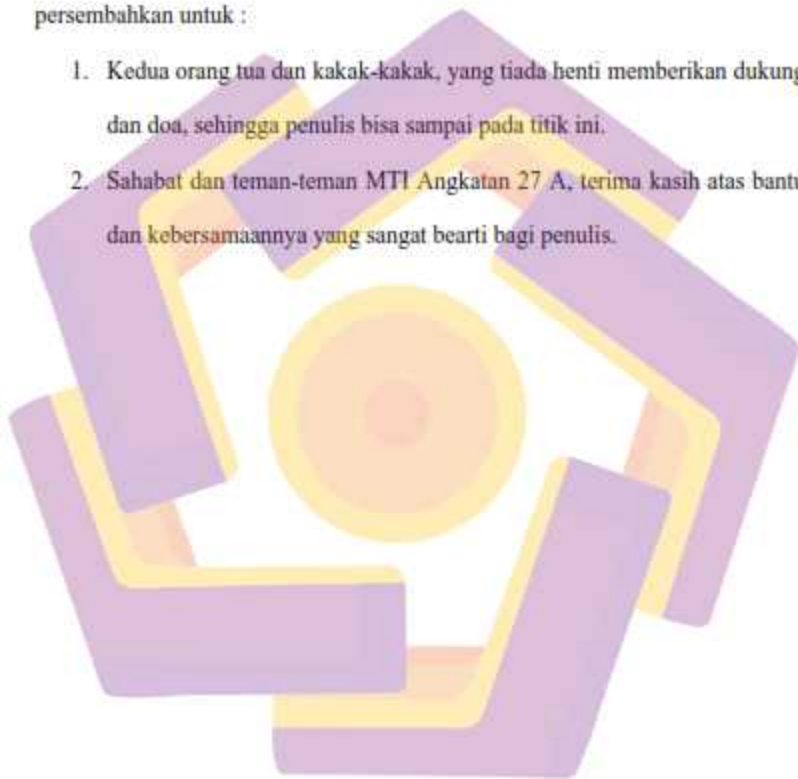
Yogyakarta, 1 September 2023
Direktur Program Pascasarjana

Prof. Dr. Kusriani, M.Kom.
NIK. 19030210

HALAMAN PERSEMBAHAN

Alhamdulillah, segala puji bagi Allah SWT yang telah melimpahkan rahmat serta hidayatNya sehingga tesis ini dapat terselesaikan. Tesis ini penulis persembahkan untuk :

1. Kedua orang tua dan kakak-kakak, yang tiada henti memberikan dukungan dan doa, sehingga penulis bisa sampai pada titik ini.
2. Sahabat dan teman-teman MTI Angkatan 27 A, terima kasih atas bantuan dan kebersamaannya yang sangat bearti bagi penulis.



HALAMAN MOTTO

Speak only when your words are more beautiful than the silence

– Imam Ali R.A

A man who is a master of patience is master of everything else

– George Savile



KATA PENGANTAR

Segala puji penulis panjatkan kepada Allah Subhanahu wa ta'ala atas segala limpahan Rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tesis ini dengan judul “Optimasi Hyperparameter Metode Convolutional Neural Network Dengan Transfer Learning Untuk Klasifikasi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit” ini.

Bimbingan dan bantuan berharga dari berbagai pihak tidak terlepas dari penyusunan tesis ini. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. M. Suyanto, M.M. selaku rektor Universitas Amikom Yogyakarta yang berkenan memberikan kesempatan untuk menimba ilmu.
2. Ibu Prof. Dr. Kusriani, M.Kom. sebagai pembimbing utama dan bapak Tonny Hidayat, M.Kom., Ph.D. sebagai pendamping yang telah mendampingi dalam penyusunan tesis ini yang telah banyak memberikan ilmu serta waktunya selama membimbing dalam penulis menyelesaikan tesis ini.
3. Bapak Dr. Arief Setyanto, S.Si., M.T. dan bapak Dr. Ferry Wahyu Wibowo, S.Si., M.Cs. selaku penguji yang telah memberikan masukan dan saran kepada penulis dalam menyelesaikan tesis ini.
4. Bapak/ibu dosen Magister Teknik Informatika Universitas Amikom Yogyakarta yang telah memberikan ilmunya kepada penulis selama menempuh studi Magister Teknik Informatika.

5. Keluarga tercinta bapak, ibu, serta kakak-kakakku terima kasih atas dukungan serta doa.
6. Sahabat dan teman-teman Magister Teknik Informatika Angkatan 27 A yang telah berbagi serta berjuang bersama menyelesaikan studi Magister Teknik Informatika.
7. Terakhir, kepada semua pihak yang telah membantu yang tidak dapat penulis sampaikan satu-persatu.

Ketidaktepatan dan kekurangan masih dapat ditemukan untuk menjadi patokan untuk kearah yang lebih baik lagi di masa yang akan datang. Selain itu, penulis meminta maaf apabila terdapat kata-kata yang salah atau salah dalam penyusunan. Terima kasih.

Yogyakarta, 1 September 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xv
INTISARI.....	xvii
<i>ABSTRACT</i>	xviii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	7
1.3. Batasan Masalah.....	8
1.4. Tujuan Penelitian.....	9
1.5. Manfaat Penelitian.....	9
BAB II TINJAUAN PUSTAKA.....	10
2.1. Tinjauan Pustaka.....	10
2.2. Keaslian Penelitian.....	15
2.3. Landasan Teori.....	21

BAB III METODE PENELITIAN	32
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	32
3.2. Metode Pengumpulan Data.....	32
3.3. Metode Analisis Data.....	34
3.4. Alur Penelitian.....	35
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	41
4.1. Pengumpulan Data.....	41
4.2. Skenario Penelitian.....	43
4.3. Preprocessing Data.....	47
4.3.1. Penyeimbangan Data.....	50
4.3.2. Penyesuaian Fully Connected Layer.....	55
4.4. Hasil dan Evaluasi Model Klasifikasi.....	60
4.4.1. Hasil Model Klasifikasi ResNet50.....	66
4.4.2. Pengaruh Optimizer ResNet50.....	68
4.4.3. Pengaruh Learning Rate ResNet50.....	68
4.4.4. Pengaruh Batch Size ResNet50.....	70
4.4.5. Pengaruh Epoch ResNet50.....	70
4.4.6. Pengaruh Dropout ResNet50.....	71
4.4.7. Pengaruh Alokasi Data ResNet50.....	72
4.4.8. Hasil Model Klasifikasi EfficientNetB0.....	72
4.4.9. Pengaruh Optimizer EfficientNetB0.....	74
4.4.10. Pengaruh Learning Rate EfficientNetB0.....	74
4.4.11. Pengaruh Batch Size EfficientNetB0.....	75

4.4.12. Pengaruh Epoch EfficientNetB0	76
4.4.13. Pengaruh Epoch EfficientNetB0	77
4.4.14. Pengaruh Dropout EfficientNetB0	78
4.4.15. Pengaruh Alokasi Data EfficientNetB0.....	78
4.5. Pengaruh Penyeimbangan Data	79
4.5.1. Hasil Synthetic Minority Oversampling Technique (SMOTE).....	80
4.5.2. Hasil Random Under Sampling (RUS)	83
4.6. Perbandingan Waktu Komputasi	87
4.7. Perbandingan Kondisi Terbaik Kedua Model.....	89
4.8. Evaluasi Hasil	92
4.9. Analisis Perbandingan Terhadap Studi Literatur.....	94
4.10. Pengujian Tambahan	97
4.10.1. Skenario 8 Warna Dan Tekstur	98
4.10.2. Skenario 9 Parameter Yang Sama	100
BAB V PENUTUP	104
5.1. Kesimpulan	104
5.2. Saran	105
DAFTAR PUSTAKA	107

DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian Optimasi Hyperparameter Metode Convolutional Neural Network Dengan Transfer Learning Untuk Klasifikasi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit	15
Tabel 3.1. Tingkat kematangan TBS kelapa sawit.....	33
Tabel 4.1. Detail dataset tingkat kematangan.....	41
Tabel 4.2. Sampel data citra TBS.....	42
Tabel 4.3. Sampel data citra TBS.....	43
Tabel 4.4. Hasil pengujian ResNet50.....	67
Tabel 4.5. Perbandingan hasil pengujian optimizer ResNet50.....	68
Tabel 4.6. Perbandingan hasil pengujian Learning Rate ResNet50.....	69
Tabel 4.7. Perbandingan hasil pengujian Batch Size ResNet50.....	70
Tabel 4.8. Perbandingan hasil pengujian Epoch ResNet50.....	70
Tabel 4.9. Perbandingan hasil pengujian Dropout ResNet50.....	71
Tabel 4.10. Perbandingan hasil pengujian Alokasi Data ResNet50.....	72
Tabel 4.11. Hasil pengujian EfficientNetB0.....	73
Tabel 4.12. Perbandingan hasil pengujian Optimizer EfficientNetB0.....	74
Tabel 4.13. Perbandingan hasil pengujian Learning Rate EfficientNetB0.....	75
Tabel 4.14. Perbandingan hasil pengujian Batch Size EfficientNetB0.....	76
Tabel 4.15. Perbandingan hasil pengujian Epoch EfficientNetB0.....	76
Tabel 4.15. Perbandingan hasil pengujian Epoch EfficientNetB0.....	77
Tabel 4.16. Perbandingan hasil pengujian Dropout EfficientNetB0.....	78

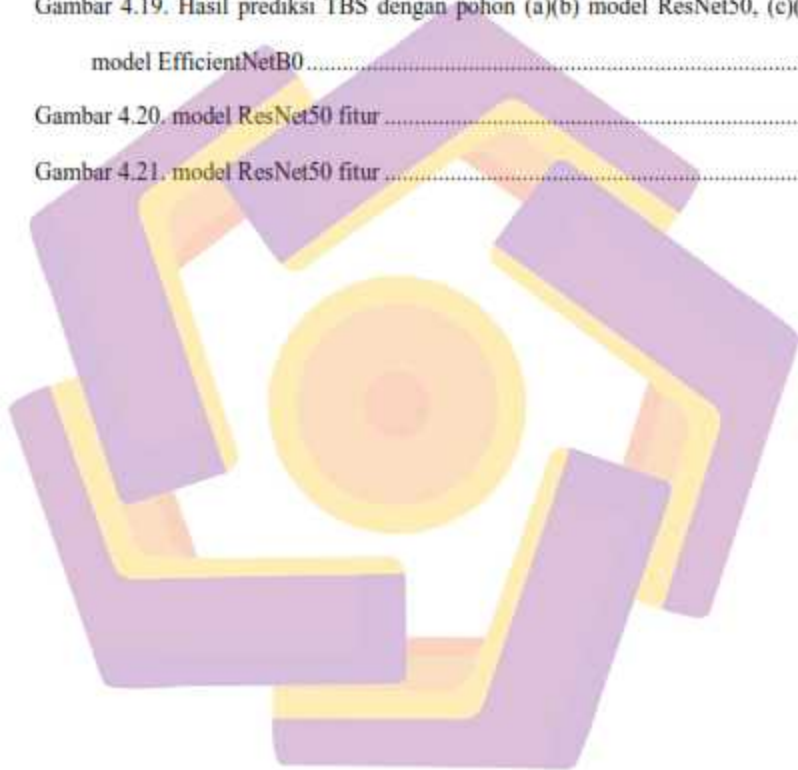
Tabel 4.17. Perbandingan hasil pengujian Alokasi Data EfficientNetB0.....	79
Tabel 4.18. Perbandingan sebelum dan sesudah penyeimbangan.....	81
Tabel 4.19. Perbandingan precision dan recal pada setiap kelas	82
Tabel 4.20. Hasil penerapan RUS	83
Tabel 4.21. Hasil precision dan recall masing-masing kelas RUS	86
Tabel 4.22. Perbandingan penelitian terkait.....	95
Tabel 4.23. Perbandingan gap penelitian terkait.....	96
Tabel 4.24. Pengujian tambahan Fitur warna dan tekstur.....	100
Tabel 4.25. Pengujian tambahan Parameter.....	101



DAFTAR GAMBAR

Gambar 2.1. Arsitektur CNN secara umum	28
Gambar 2.2. Residual Learning Block	30
Gambar 2.3. Arsitektur EfficientNet-B0	31
Gambar 3.1. Alur Penelitian	35
Gambar 3.2. Arsitektur ResNet50	37
Gambar 3.3. Arsitektur EfficientNetB0	39
Gambar 4.1. Labeling data	48
Gambar 4.2. Proses resize data	49
Gambar 4.3. Citra TBS sebelum dan setelah preprocessing	50
Gambar 4.4. Data tidak seimbang	51
Gambar 4.5. Smote pada python	53
Gambar 4.6. Dataset setelah dan sebelum penerapan smote	54
Gambar 4.7. Dataset setelah dan sebelum penerapan RUS	55
Gambar 4.8. CNN dengan transfer learning dalam penelitian ini	56
Gambar 4.9. ResNet50 sebelum penyesuaian	57
Gambar 4.10. ResNet50 setelah penyesuaian	58
Gambar 4.11. EfficientNetB0 sebelum penyesuaian	59
Gambar 4.12. EfficientNetB0 setelah penyesuaian	60
Gambar 4.13. Pengujian confusion matrix	62
Gambar 4.14. Perbandingan akurasi sebelum dan sesudah penerapan smote	80
Gambar 4.15. Perbandingan akurasi sebelum dan sesudah penerapan RUS	84

Gambar 4.16. Perbandingan waktu komputasi	88
Gambar 4.17. Perbandingan kondisi terbaik Resnet50 dan EfficientNetB0	91
Gambar 4.18. Hasil prediksi data baru (a)(b)(c)(d) model ResNet50, (e)(f)(g)(h) model EfficientNetB0	92
Gambar 4.19. Hasil prediksi TBS dengan pohon (a)(b) model ResNet50, (c)(d), model EfficientNetB0	93
Gambar 4.20. model ResNet50 fitur	99
Gambar 4.21. model ResNet50 fitur	99



INTISARI

Tandan Buah Segar (TBS) merupakan bahan utama dalam produksi minyak sawit. Kematangan menandakan bahwa Tandan Buah Segar (TBS) kelapa sawit adalah tanda kualitas buah terbaik. Untuk menentukan tingkat kematangan TBS dapat dinilai berdasarkan karakteristik eksternal seperti warna, tekstur, duri, ukuran, berat, dan bentuk TBS. Karena kemudahannya, ekstraksi ciri warna sering digunakan, tetapi hanya memperhatikan warna rata-rata yang kurang akurat. Sebaliknya, ekstraksi ciri tekstur memisahkan elemen yang mencerminkan sifat tekstur, tetapi seringkali kurang efektif dalam mengatasi noise dan variasi ukuran sampel tekstur. Untuk mengatasi masalah ini, metode yang lebih canggih menggunakan model deep learning digunakan untuk mengekstraksi fitur gambar yang lebih kompleks dan sensitif terhadap berbagai faktor. Dalam pekerjaan ini, dua algoritma arsitektur Convolutional Neural Network (CNN) dengan transfer learning diusulkan untuk melakukan klasifikasi tingkat kematangan. ResNet50 dan EfficientNetB0 diskenariokan dan dibandingkan dalam klasifikasi kematangan dengan empat tingkatan: mentah, kurang matang, matang dan terlalu matang. Dalam pekerjaan ini data yang diambil dari web Roboflow.com dilatih dan dilakukan penyeimbangan data antar kelas dengan metode Synthetic Minority Over-sampling Technique (SMOTE) dan Random Under-sampling (RUS) yang kemudian dilakukan hyperparameter tuning terhadap kedua model yang dilanjutkan dengan perbandingan akurasi, precision, recall dan waktu pelatihan. Selama tahap pengujian didapatkan kesimpulan bahwa arsitektur Convolutional Neural Network (CNN) dengan transfer learning cukup baik untuk mengklasifikasikan tingkat kematangan TBS dengan kelas berbeda, EfficientNetB0 dengan SMOTE dan hyperparameter tuning menunjukkan hasil yang cukup menjanjikan dengan akurasi 98.93%, precision 98.93%, recall 98.93%.

Kata kunci: Deep Learning, Transfer learning, Hyperparameter tuning, Maturity classification, FFB

ABSTRACT

Fresh Fruit Bunches (FFB) are the main ingredient in palm oil production. Ripeness indicates that oil palm Fresh Fruit Bunch (FFB) is a sign of the best quality fruit. To determine the level of maturity of FFB, it can be assessed based on external characteristics such as color, texture, thorns, size, weight and shape of FFB. Because of its simplicity, color feature extraction is often used, but only considering the average color is less accurate. In contrast, texture feature extraction separates elements that reflect texture properties, but is often less effective in dealing with noise and variations in texture sample size. To overcome this problem, more sophisticated methods using deep learning models are used to extract image features that are more complex and sensitive to various factors. In this work, two Convolutional Neural Network (CNN) architecture algorithms with transfer learning are proposed to perform maturity level classification. ResNet50 and EfficientNetB0 were scenario and compared in maturity classification with four levels: unripe, underripe, ripe and overripe. In this work, data taken from the Roboflow.com website was trained and data was balanced between classes using the Synthetic Minority Over-sampling Technique (SMOTE) and Random Under-sampling (RUS) methods, then hyperparameter tuning was carried out on the two models, followed by accuracy comparisons, precision, recall and training time. During the testing phase, it was concluded that the Convolutional Neural Network (CNN) architecture with transfer learning was good enough to classify FFB maturity levels with different classes, EfficientNetB0 with SMOTE and hyperparameter tuning showed quite promising results with 98.93% accuracy, 98.93% precision, 98.93% recall.

Keyword: Deep Learning, Transfer learning, Hyperparameter tuning, Maturity classification, FFB

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Transformasi digital, yang dianggap sebagai masa depan teknologi digital, menjadi prioritas utama di seluruh dunia. Salah satu tujuan transformasi ini adalah untuk membantu petani terhubung dengan ekonomi digital untuk membuat pertanian berkelanjutan. (Rijswijk et al., 2021). Secara efektif dan efisien, kemajuan teknologi informasi saat ini dapat digunakan untuk menentukan kualitas produk kelapa sawit. Teknik data mining dapat digunakan untuk menentukan klasifikasi kualitas kelapa sawit ini. Sebuah objek data diklasifikasikan ke dalam salah satu kategori yang telah didefinisikan sebelumnya melalui proses klasifikasi (Suryani et al., n.d.).

Tiga faktor yang dilaporkan mempengaruhi kualitas minyak sawit adalah kematangan TBS, asam lemak bebas yang rendah, dan tingkat ekstraksi minyak yang tinggi. Semua orang yang terlibat dalam bisnis ini percaya bahwa kematangan adalah tanda kualitas buah terbaik (Ghazalli et al., 2023). Menurut Dewan Minyak Kelapa Sawit Malaysia (MPOB), tingkat kematangan Tandan Buah Segar kelapa sawit terdiri dari mentah, kurang matang, matang, dan terlalu matang. Tandan mentah dan kurang matang yang dipanen tanpa mengontrol kematangan menyebabkan minyak kualitas rendah. Sebaliknya, minyak dengan kualitas tinggi dan nilai komoditas tinggi hanya dihasilkan dari tandan telah matang, kemudian

diikuti oleh tandan yang terlalu matang((Junkwon et al., 2009; Mansour et al., 2022; K. Sunilkumar & D. S. Sparjan Babu, 2013)).

Untuk menentukan tingkat kematangan TBS dapat dinilai berdasarkan karakteristik eksternal seperti warna, tekstur, duri, ukuran, berat, dan bentuk TBS (Alfatni et al., 2020). Perubahan warna kulit luar TBS dan jumlah atau persentase buah yang terlepas dari tandan buah segar adalah dua cara yang biasanya digunakan untuk menentukan tingkat kematangan TBS. Salah satu cara utama untuk mengetahui tingkat kematangan buah kelapa sawit adalah dengan melihat perubahan warnanya. Namun, persepsi manusia tentang warna dapat berubah-ubah karena kondisi fisik dan mental manusia. Buah yang terlepas dari TBS kelapa sawit dapat terjadi oleh factor lain, Jika TBS kelapa sawit tidak dipanen pada waktu yang tepat, buah dapat cepat berguguran ((Being et al., 2020)). Pelaksanaan metode konvensional seperti ini membutuhkan banyak tenaga kerja dan memerlukan waktu yang lama (Sinambela et al., 2020). Oleh karena itu, perlu adanya pengembangan klasifikasi tingkat kematangan TBS untuk mengoptimalkan kualitas minyak yang akan dihasilkan dari TBS.

Dalam melakukan klasifikasi TBS kelapa sawit telah digunakan metode seperti image processing, machine learning, and deep learning dengan menggunakan data citra untuk mendapatkan akurasi klasifikasi yang tinggi (Suharjito et al., 2021). Dua pendekatan yang paling sering digunakan dalam melakukan klasifikasi tingkat kematangan TBS kelapa sawit yaitu machine learning dan deep learning. Machine learning terdiri dari tiga proses tahapan yaitu akuisisi data, pembuatan model dan generalisasi, algoritma machine learning digunakan

ketika menangani masalah yang cukup kompleks dimana kompetensi manusia tidak mampu mengatasi masalah tersebut (Durai & Shamili, 2022).

Penelitian yang dilakukan (Hamdani et al., 2021), melakukan pengklasifikasian tingkat kematangan TBS kelapa sawit menggunakan ekstraksi fitur pada ruang warna L^*a^*b pada metode algoritma Latent Dirichlet Allocation (LDA), pada penelitian ini mereka berhasil melakukan pengklasifikasian dengan menghasilkan akurasi pengklasifikasian sebesar 98,88% (Raj et al., 2021). Di penelitian lain *fine k-nearest neighbors* (KNN) digunakan untuk mengklasifikasi TBS kelapa sawit, dikombinasikan dengan spektroskopi raman dengan menghasilkan akurasi 100%. (Himmah et al., 2020) dalam penelitian yang mereka lakukan untuk mengklasifikasi tingkat kematangan TBS kelapa sawit berdasarkan warna dan HSV pada metode K-Means Clustering, mereka menghasilkan akurasi sebesar 64,58%. Walaupun begitu machine learning masih memiliki kelemahan yaitu perlunya ekstraksi fitur diluar model, maka atas hal ini machine learning akan cukup memakan waktu jika diterapkan untuk mengklasifikasi tingkat kematangan TBS kelapa sawit yang memiliki fitur yang kompleks (Suharjito et al., 2021).

Teknik ekstraksi fitur warna, paling banyak digunakan karena mudahnya mengekstraksi fitur, namun hanya bergantung pada warna rata-rata, yang akan menghasilkan hasil yang tidak akurat. Ekstraksi ciri tekstur, Teknik ini digunakan untuk mengekstraksi ciri tekstur dengan memisahkan elemen yang menggambarkan sifat tekstur. Namun, teknik ini tidak dapat menangani sensitivitas terhadap noise dan ukuran sampel tekstur dengan baik. Untuk menyelesaikan masalah ini, tekstur gambar dapat diekstrak dengan menggunakan model deep learning. Metode ini

memungkinkan model untuk mempelajari cara mengekstrak fitur dari data yang telah diberikan. Deep learning terbukti efektif dan berhasil dengan cepat. Namun, untuk metode ini dalam mencapai akurasi yang tinggi dan mempelajari setiap tingkat kematangan, diperlukan data yang besar (Ghazalli et al., 2023). Algoritma dapat ditemukan di berbagai lapisan untuk menemukan fitur yang tidak penting dalam data. Semakin banyak fitur yang ditemukan, semakin akurat (Lai et al., 2023). Deep learning telah menunjukkan hasil yang cukup menjanjikan dalam melakukan klasifikasi citra pada kasus yang kompleks (Susanto et al., 2020).

Deep learning berbeda dari Machine learning karena menggunakan konsep pembelajaran representasi, yang memungkinkan model menggunakan data input mentah dalam berbagai tugas, seperti klasifikasi. Ini berarti bahwa model pembelajaran mendalam membentuk bentuk data abstraksi tingkat tinggi dengan mengekstraksi fitur input dari dalam model itu sendiri. Fitur-fitur ini diekstraksi melalui Konvolusi dan penyatuan, dan hasilnya dimasukkan ke dalam jaringan saraf. Convolutional Neural Network (CNN) menggunakan struktur ini (Lecun et al., 2015). CNN telah digunakan dalam banyak penelitian terkait klasifikasi dengan data gambar seperti klasifikasi kematangan TBS (Saleh & Liansitim, 2020, Prasetyo et al., 2020), pisang (Saranya et al., 2022), kanker kulit (Ali et al., 2022, Jaisakthi et al., 2022) hasil dalam penerapan berbagai model arsitektur cnn seperti VGG16, ResNet50 (Saranya et al., 2022) dan berbagai model EfficientNet (Ali et al., 2022, Jaisakthi et al., 2022) mendapatkan hasil dengan akurasi yang tinggi.

Meskipun hasil model CNN memiliki akurasi yang cukup baik, tetapi masih dapat dioptimalkan untuk mendapatkan model yang lebih baik dengan mengatur

hyperparameter (Afis Julianto et al., 2022). Pengaturan yang diperlukan untuk mengarahkan algoritme pembelajaran mesin apa pun ini dikenal sebagai hyperparameter (Konar et al., 2020). Selama beberapa dekade terakhir, metode pengoptimalan hyperparameter telah digunakan dalam pembelajaran mesin dan pembelajaran mendalam untuk mengurangi kesalahan. Hyperparameter ini memiliki kemampuan untuk mengontrol istilah regularisasi, ukuran langkah pembelajaran, parameter peluruhan momentum, dan status inisialisasi. Tidak hanya penting, tetapi juga sulit untuk memilih hyperparameter terbaik (Sen & Ozkurt, 2020). Selain itu, perlu diperhatikan bahwa kurangnya pendekatan sistematis dan otomatis dapat memengaruhi proses pengembangan model secara keseluruhan dan kemampuan algoritma untuk belajar. Pendekatan ini dapat membantu mengembangkan model deep learning yang lebih efisien dengan hyperparameter yang lebih sesuai. (Ramirez-Gonzalez et al., 2022). Dalam penelitian terdahulu terkait pengoptimalan pada kinerja model neural network menggunakan beberapa hyperparameter diantaranya seperti learning rate (Konar et al., 2020; Motta et al., 2020; Sen & Ozkurt, 2020), batch size (Haque et al., 2022; Lee & Kim, 2023), optimizer (Lee & Kim, 2023; Motta et al., 2020), momentum (Lee & Kim, 2023; Sen & Ozkurt, 2020), epoch (Haque et al., 2022; Motta et al., 2020) dan dropout (Minarno et al., 2021).

Metode deep learning memiliki kekurangan, yaitu mereka membutuhkan jumlah data pelatihan yang besar untuk mencapai akurasi yang tinggi dan untuk menghindari overfitting, yaitu ketika jaringan saraf berfungsi dengan baik pada kumpulan data tertentu tetapi sebaliknya pada kumpulan data lainnya. Ada bukti

bahwa pembelajaran transfer dapat meningkatkan kinerja CNN dalam situasi di mana data yang sangat besar tidak dapat diakses. Akibatnya, kelemahan ini dapat diatasi dengan menggunakan pembelajaran transfer (Hentschel et al., 2016; Alzubaidi et al., 2020). Teknik transfer learning (TL) telah muncul di bidang deep learning baru-baru ini. Dalam teknik TL, bobot model pembelajaran mendalam yang telah dilatih dengan kumpulan data besar dapat ditransfer ke model jaringan lain untuk tugas baru yang serupa. Dengan demikian, bobot yang telah dilatih sebelumnya dapat digunakan untuk memulai model jaringan baru (Zein et al., 2021). Pembelajaran transfer menggunakan ImageNet secara khusus untuk transfer data objek umum (Deng et al., 2009). Untuk mendapatkan kinerja yang kuat pada dataset yang lebih kecil, arsitektur yang lebih kompleks dan mendalam perlu dikembangkan. Model kompleks dapat disesuaikan untuk mengekstraksi fitur unik dari gambar buah kelapa sawit (Sinambela et al., 2020).

Metode Sampling data juga dapat digunakan untuk mencegah overfitting. Metode Sampling data menghasilkan satu atau lebih kumpulan data yang masing-masing memiliki distribusi kelas yang berbeda dari kumpulan data aslinya. Oleh karena itu, undersampling dan oversampling adalah dua jenis resampling data utama yang digunakan (Azlim Khan & Ahamed Hassain Malim, 2023). random oversampling (ROS), dimana sampel dari kelas minoritas digandakan, dan random undersampling (RUS), dimana sampel dari kelas mayoritas dibuang (De Angeli et al., 2022). Teknik resampling data pada penelitian (Puri & Kumar Gupta, 2021) berfungsi dengan baik, menghasilkan nilai yang hilang yang lebih besar diantara set data kelas yang tidak seimbang. Pada penelitian (Azlim Khan & Ahamed Hassain

Malim, 2023) Ditunjukkan bahwa penggunaan Random Undersampling (RUS) sangat mempengaruhi kinerja model dalam memprediksi interaksi target obat. Ini terutama berlaku untuk dataset yang sangat tidak seimbang. Saat menangani kumpulan data yang tidak seimbang, SVM-SMOTE dapat digunakan sebagai metode resampling.

Maka berdasarkan latar belakang dan penelitian terdahulu, maka dari itu akan dilakukan penelitian untuk mengklasifikasi tingkat kematangan TBS kelapa sawit dengan memanfaatkan metode convolutional neural networks (CNN) dengan transfer learning dengan menggunakan hyperparameter tuning untuk mendapatkan hasil yang optimal. Pengklasifikasian dengan menggunakan metode convolutional neural networks (CNN) dengan transfer learning diharapkan dapat membantu mengoptimalkan kualitas minyak dari hasil pengolahan TBS kelapa sawit, maka untuk menilai metode klasifikasi yang diajukan dilakukan pengukuran metode dengan akurasi yang dihasilkan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dirumuskan, rumusan masalah dalam penelitian ini dijabarkan sebagai berikut:

- a. Metode apa yang dapat meningkatkan hasil akurasi dalam klasifikasi TBS kelapa sawit menggunakan convolutional neural networks dengan transfer learning?
- b. Faktor apa yang mempengaruhi performa terhadap optimasi klasifikasi dengan menggunakan metode arsitektur convolutional neural networks dengan transfer learning?

c. Berapa waktu komputasi yang dihasilkan dalam klasifikasi TBS kelapa sawit menggunakan convolutional neural networks dengan transfer learning?

1.3. Batasan Masalah

Beberapa hal yang menjadi batasan masalah dan ruang lingkup dalam penelitian ini, agar penelitian yang dilakukan terarah dan pembahasan yang relevan adalah sebagai berikut:

- a. Klasifikasi yang dilakukan terhadap TBS kelapa sawit adalah end-to-end klasifikasi.
- b. Pengklasifikasian TBS kelapa sawit menggunakan convolutional neural networks dengan transfer learning.
- c. Arsitektur convolutional neural networks yang akan digunakan adalah ResNet dan EfficientNet.
- d. Data citra TBS dikumpulkan dari website penyedia data citra roboflow.com dengan empat tingkat kematangannya, yaitu mentah, kurang masak, masak, dan terlalu masak.
- e. Untuk memudahkan dalam pelatihan, pengujian serta kebutuhan visualisasi data pada pengembangan metode convolutional neural networks dengan transfer learning akan menggunakan Google Collaboratory.
- f. Keluaran yang dihasilkan dari proses klasifikasi berupa hasil akurasi, recall, precision dengan menggunakan alat ukur confusion matrix.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mendapatkan model convolutional neural networks dengan transfer learning terbaik yang telah dilakukan optimasi dalam klasifikasi tingkat kematangan TBS kelapa sawit sebagai dasar upaya dalam peningkatan kualitas minyak kelapa sawit dan menghindari kesalahan dalam penilain tingkat kematangan.

1.5. Manfaat Penelitian

Berdasarkan tujuan yang ingin dicapai, penelitian ini diharapkan bermanfaat sebagai berikut:

- a. Model yang dihasilkan dapat menjadi alternatif dalam melakukan klasifikasi pada proses grading dan pemanen TBS kelapa sawit.
- b. Pemodelan yang dihasilkan sebagai wahana penambah wawasan dalam pengembangan dan evaluasi penelitian terkait pengklasifikasian pada penelitian selanjutnya.
- c. Membantu perusahaan dalam meningkatkan hasil akurasi klasifikasi tingkat kematangan sehingga mampu menghasilkan kualitas minyak yang baik.
- d. Membantu petani dalam peralihan pertanian cerdas dan berkelanjutan pada penerapan teknologi penilaian tingkat kematangan TBS kelapa sawit saat proses pemanenan.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Beberapa penelitian terdahulu yang dijadikan acuan dan tinjauan pustaka pada penelitian ini. Penelitian oleh (Suharjito et al., 2021) menggunakan metode deep learning yang dikombinasikan dengan transfer learning dan augmentasi data untuk melakukan klasifikasi pada TBS kelapa sawit. Penelitian ini mengimplementasikan empat transfer learning model cnn ringan dan metode augmentasi data baru yang diberi nama 9-angle crop, serta menggunakan kuantisasi pasca pelatihan untuk pengoptilan lebih lanjut. Mendapatkan model terbaik yaitu EfficientNet-B0 dengan kombinasi transfer learning, crop 9 angle, dan kuantisasi float16, mencapai akurasi pengujian keseluruhan sebesar 0,893 pada TensorFlow Lite dengan waktu klasifikasi 96 ms per gambar serta melampaui tiga model lainnya. Data pada penelitian ini menggunakan dataset dengan enam kelas kematangan yaitu mentah, kurang matang, matang, terlalu matang, tidak normal, dan tandan buah kosong. Saran pada penelitian ini untuk melakukan peningkatan lebih lanjut dalam mengklasifikasikan objek menggunakan deep learning neural networks dengan akurasi yang lebih tinggi dan waktu proses yang lebih cepat.

Pendeteksian dan perhitungan TBS kelapa sawit menggunakan metode R-CNN dilakukan oleh (Prasetyo et al., 2020), Penelitian yang dilakukan menyajikan sistem pendeteksian dan penghitungan TBS secara otomatis. Dengan membandingkan empat arsitektur jaringan yaitu inception V2, inception V2, resnet

50, dan resnet 101, dengan hasil keempat model arsitektur menghasilkan skor F1 diatas 80%. Pada penelitian ini terjadi tidak terdeteksinya TBS diakibatkan oleh tingginya okulasi yang disebabkan oleh tertutupnya TBS oleh objek lain.

Pendeteksian objek untuk klasifikasi tingkat kematangan TBS kelapa sawit selanjutnya dilakukan oleh (Mansour et al., 2022) menggunakan beberapa algoritma pendeteksian objek, yaitu MobileNetV2 SSD, EfficientDet (Lite0, Lite1 dan Lite2), dan YOLOv5 (YOLOv5n, YOLOv5s dan YOLOv5m). dalam penelitian ini disimulasikan menggunakan arsitektur yang berbeda untuk mengklasifikasikan tingkat kematangan TBS kelapa sawit yang berbeda, yaitu dengan empat kelas kematangan utama: matang, mentah, setengah matang, dan terlalu matang. Menghasilkan YOLOv5m menunjukkan hasil yang menjanjikan dengan presisi rata-rata 0,842. Adapun saran pada penelitian ini untuk melakukan penambahan pada dataset dengan menambahkan gambar TBS kelapa sawit di pohon dan meningkatkan akurasi pengujian waktu nyata algoritma serta pengoptimalan lebih lanjut menggunakan penyetelan hyperparameter agar sesuai dengan dataset untuk klasifikasi kematangan TBS kelapa sawit dan menggunakan teknik ansambel model untuk pengembangan algoritma.

(Lai et al., 2022) melakukan penelitian pendeteksian secara real-time untuk menentukan TBS kelapa sawit matang menggunakan Yolov4. Data yang digunakan pada penelitian ini diambil melalui tahapan akuisisi data dengan dua kelas yaitu matang dan tidak matang. Pengujian pendeteksian TBS matang dengan menggunakan Yolov4 tanpa melakukan perubahan yang dilakukan pada arsitektur model, terlihat bahwa dengan metodologi dan hyperparameter yang mereka pilih

untuk pelatihan, deteksi yang dihasilkan cukup memuaskan dengan hasil mean Average Precision (mAP) dan recall adalah 87,9% dan 82% serta deteksi memenuhi Intersect over Union (IoU) dengan lebih dari 0,5 setelah 2000 iterasi dan sistem beroperasi secara real-time kecepatan sekitar 21 Frame Per Second (FPS). Saran pada penelitian ini melakukan peningkatan dalam data pelatihan, arsitektur model, dan pengoptimalan hyperparameter.

Selanjutnya penelitian terdahulu yang dilakukan oleh (Hamdani et al., 2021), pada penelitian ini dilakukan klasifikasi menggunakan machine vision berdasarkan fitur warna dan tekstur pada TBS kelapa sawit. Jaringan saraf tiruan (JST) dengan algoritma back-propagation diterapkan untuk melakukan klasifikasi tingkat kematangan TBS mentah, matang, dan setengah matang. Ekstraksi fitur pada penelitian ini diterapkan pada lima ruang warna: RGB, HSI, HSV, YCbCr, dan YIQ dan menghasilkan sebanyak 40 fitur diekstraksi. Digunakan Tiga metode seleksi fitur analisis yaitu principal component analysis (PCA), correlation feature selection (CFS), dan gain ratio, yang menghasilkan five features (HYIQ, μ Cr, HYCbCr, μ I2, μ YIQ), two features (μ YCbCr, HYCbCr), and six features (HYCbCr, μ S, μ Cb, μ I2, μ Cr, γ YCbCr). Dari metode yang diusulkan, PCA dan JST mendapatkan hasil akurasi 98,3%, 98,4% dan 98,3% untuk precision dan recall. Pada penelitian ini terdapat kesalahan klasifikasi TBS setengah matang menjadi sebagai TBS matang, dikarenakan memiliki ruang warna yang hampir sama dan TBS matang terprediksi TBS mentah, terjadi ruang warna TBS matang tertutupi noise yang menyerupai warna TBS mentah. Saran pada penelitian ini untuk

meningkatkan teknik akuisisi citra dan menghilangkan noise yang menutupi TBS serta untuk membedakan TBS menjadi lebih dari tiga kelas.

Penelitian terdahulu menggunakan metode deep learning untuk melakukan klasifikasi dilakukan oleh (Ashari et al., 2022) menggunakan convolutional neural networks untuk mengklasifikasi pada tiga tingkat kematangan TBS mentah, matang, dan busuk. Menggunakan model CNN dengan 64 filter dan kernel matriks 3x3, kemudian pooling ukuran 2x2 dengan mask shift dua langkah. Pada tahap konvolusi kedua dengan menggunakan filter sebanyak 32 dan kernel matriks 2x2, yang terakhir dilanjutkan dengan flattening. Dari model yang diajukan menghasilkan nilai akurasi 98% untuk pelatihan, dan 76% untuk pengujian model. Kelemahan pada penelitian ini, pada pengujian buah busuk menggunakan background didapati akurasi 0,4481, hasil klasifikasi dapat dikatakan belum akurat.

Penelitian terdahulu yang dilakukan oleh (Saleh & Liansitim, 2020) untuk mengklasifikasi kelapa sawit masak menggunakan deep learning. Menggunakan CNN terhadap 628 yang terdiri dari 2 kelas berbeda mentah dan masak. Model yang diajukan dilakukan percobaan dengan 5 epoch dengan menghasilkan akurasi 98%. Saran penelitian ini pekerjaan di masa mendatang harus mendapatkan lebih banyak data karena algoritma deep learning sering bekerja lebih baik dengan lebih banyak data.

(Susanto et al., 2020) melakukan penelitian klasifikasi tingkat kematangan buah sawit dengan menggunakan deep learning dan perhatian visual. Dalam penelitian ini menggunakan 400 citra dengan tujuh tingkat kematangan, yang diuji pada model AlexNet, DenseNet + SE layer, ResAtt DenseNet, dan DenseNet

Sigmoid. Akurasi yang dihasilkan dari percobaan keempat model tertinggi 69% pada model ResAtt DenseNet, dan DenseNet Sigmoid, dengan recall dan precision tertinggi ResAtt DenseNet 0,6883 dan 0,7152. Saran pada penelitian ini pekerjaan masa depan dapat mengeksplorasi penggunaan strategi mekanisme perhatian lain yang dapat mengatasi masalah gradien hilang, seperti kerugian ekstra setelahnya.

Dari keseluruhan penelitian terdahulu dapat menyimpulkan beberapa hal, dimana dataset yang digunakan untuk klasifikasi merupakan data citra TBS yang kelasnya terbagi menjadi enam yaitu mentang, kurang matang, matang, terlalu matang, busuk, dan tandan kosong. Metode yang digunakan adalah metode deep learning dengan membandingkan arsitektur yang dimiliki oleh metode deep learning, dalam penelitian terdahulu menggunakan deep learning memiliki beberapa kelemahan dan saran yang akan dikembangkan dalam penelitian ini.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian
Optimasi Hyperparameter Metode Convolutional Neural Network Dengan Transfer Learning Untuk Klasifikasi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Oil palm fresh fruit bunch ripeness classification on mobile devices using	Suharjito, Gregorius Natanael Elwirehardja, Jonathan Sebastian Prayoga, Elsevier, 2021	Pembuatan aplikasi mobile untuk mengklasifikasikan tingkat kematangan TBS kelapa sawit menggunakan Convolutional Neural Network (CNN) yang ringan.	Menghasilkan sebuah aplikasi Android yang mampu mengklasifikasikan tingkat kematangan TBS kelapa sawit, dengan , EfficientNet-B0 telah terbukti tidak hanya sebagai arsitektur CNN paling efisien yang hanya membutuhkan waktu klasifikasi gambar 96 ms, tetapi juga sangat akurat, dengan akurasi pengujian 0,893.	Saran pada penelitian ini untuk melakukan peningkatan lebih lanjut dalam mengklasifikasikan objek menggunakan deep learning neural networks dengan akurasi yang lebih tinggi dan waktu proses yang lebih cepat.	Penelitian sebelumnya melakukan perbandingan antara MobileNetV1 dengan EfficientNet-B0 untuk klasifikasi TBS pada perangkat mobile. Penelitian ini akan mempertimbangkan EfficientNet dan ResNet dalam melakukan klasifikasi tingkat kematangan TBS.
2	Automatic Detection and Calculation of Palm Oil Fresh Fruit Bunches using Faster R-CNN	Novian Adi Prasetyo, Pranowo and Albertus Joko Santoso, International Journal of Applied	Menggunakan convolutional neural network (CNN) sebagai solusi untuk melakukan deteksi dan perhitungan TBS secara otomatis.	Hasil menunjukkan keberhasilan dalam menghitung TBS. Keberhasilan tersebut ditunjukkan dengan hasil evaluasi keempat model jaringan resnet	Tidak terdeteksinya TBS diakibatkan oleh tingginya okulasi yang disebabkan oleh tertutupnya TBS oleh objek lain dan melakukan Modifikasi	Pada penelitian ini convolutional neural network (CNN) digunakan untuk pendeteksian dan perhitungan TBS, dalam pendeteksian terdapat TBS yang tidak terdeteksi akibat

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Science and Engineering, 2021		inception V2, inception V2, resnet 50, dan resnet 101 dengan rata-rata skor F1 di atas 80%.	pada arsitektur jaringan dapat menjadi penemuan baru untuk deteksi dan perhitungan hasil TBS yang lebih baik.	tertutup objek lain. Pada penelitian yang akan dilakukan akan menggunakan dataset yang berbeda untuk menghindari tidak terdeteksinya TBS dan mengoptimalkan akurasi dengan model EfficientNet dan ResNet.
3	Object Detection Algorithms for Ripeness Classification of Oil Palm Fresh Fruit Bunch	Mohamed Yasser, Mohamed Ahmed Mansour, Katrina D. Dambul, Kan Yeep Choo, International Journal of Technology, 2022	Pendeteksian objek akan disefidiki untuk mengembangkan algoritma pendeteksian objek yang mampu mengklasifikasikan kematangan TBS kelapa sawit	Tiga algoritma deteksi objek yaitu MobileNetV2 SSD, EfficientDet, dan YOLOv5 dengan arsitektur yang berbeda diuji kinerjanya untuk mengklasifikasikan tingkat kematangan pada TBS kelapa sawit, YOLOv5m dengan prestasi rata-rata rata-rata 0,842 diusulkan menjadi model deteksi objek.	Melakukan peningkatan pada kumpulan data, seperti menambahkan gambar TBS kelapa sawit di pohon dan meningkatkan akurasi pengujian algoritma waktu nyata, menggunakan penyeteran hyperparameter agar sesuai dengan dataset untuk klasifikasi kematangan TBS kelapa sawit.	Pada penelitian sebelumnya MobileNetV2 SSD, EfficientDet, dan YOLOv5 digunakan untuk pendeteksian objek pada klasifikasi tingkat kematangan TBS namun masih membutuhkan dataset dengan berbagai latar belakang. Sementara pada penelitian yang akan dilakukan EfficientNet dan ResNet akan digunakan pada dataset dengan berbagai latar belakang serta menggunakan penyeteran pada

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
						hyperparameter sesuai untuk melihat pengaruh pada dataset.
4	Real-Time Detection of Ripe Oil Palm Fresh Fruit	Jin Wen Lai, Hafiz Rashidi Ramli, Luthfi Idzhar Ismail, Wan Zuha Wan Hasari, IEEE, 2022	Mengembangkan sistem untuk secara otomatis mendeteksi TBS matang dan belum dipanen secara real-time menggunakan kombinasi visi komputer dan kecerdasan buatan (AI).	Model YOLOv4 yang dilatih memperoleh mAP sebesar 87,9% dalam mendeteksi TBS matang, selama pengujian di perkebunan kelapa sawit, sistem beroperasi pada kecepatan real-time sekitar 21 FPS dan mencapai mAP 87,9%.	Performa dapat ditingkatkan lebih lanjut dengan perbaikan pada data pelatihan, arsitektur model, dan optimasi hyperparameter. Mengembangkan sistem untuk mengidentifikasi pelepah sawit yang mengelilingi TBS dan tangkai TBS yang perlu dipotong.	Pada penelitian sebelumnya YoloV4 digunakan untuk real-time deteksi terhadap TBS matang, penelitian ini hanya mendeteksi satu kelas tingkat kematangan TBS. Sementara pada penelitian yang akan dilakukan akan melakukan klasifikasi pada empat tingkatan kematangan TBS, dengan data latih akan diskensariokan 70:30, 80:20, dan 90:10.
5	Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features	Anindita Septiarini, Andi Sunyoto, Hamdani Hamdani, Anita Ahmad Kasim, Fitri Utaminigrum, Heliza Rahmania Hatta, Elsevier,	Menerapkan fitur warna dan tekstur pada metode yang diusulkan untuk mengklasifikasikan tingkat kematangan TBS	Hasil JST untuk membedakan tingkat kematangan TBS berdasarkan pengukuran presisi, recall, dan akurasi menunjukkan bahwa metode yang diusulkan	Kesalahan klasifikasi TBS setengah matang sebagai TBS matang terjadi karena perubahan warna buah yang tidak merata. Keterbatasan pekerjaan ini akan	Penelitian sebelumnya klasifikasi tingkat kematangan dengan jaringan syaraf tiruan pada data dengan kelas mentah, matang, dan setengah matang. Penelitian yang akan dilakukan

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		2021		berhasil mencapai nilai masing-masing 98,4%, 98,3%, dan 98,3%.	ditingkatkan dengan meningkatkan teknik akuisisi citra dan membuang noise detail yang menutupi area buah. Selain itu, metode klasifikasi masih terbuka lebar untuk pengembangan lebih lanjut untuk membedakan TBS menjadi lebih dari tiga kelas.	pengklasifikasian akan menggunakan enam kelas data yaitu mentah, kurang matang, matang, dan terlalu matang dengan penyeimbangan pada dataset.
6	Oil Palm Fruit Ripeness Detection using Deep Learning	Suci Ashari, Gomal Juni Yanris, Iwan Purnama, Sinkron, 2022	Mendeteksi kematangan TBS kelapa sawit berdasarkan citra digital dengan menerapkan algoritma deep learning sehingga tingkat kematangan dapat diklasifikasikan	Pendeteksian kematangan tandan buah segar kelapa sawit dengan metode CNN mendapatkan nilai akurasi yang tinggi yaitu 98% pada proses pelatihan dan 76% pada proses pengujian model.	Pengujian buah busuk menggunakan background didapati akurasi 0,4481, hasil klasifikasi pada buah busuk dapat dikatakan belum akurat.	Penelitian sebelumnya pada proses klasifikasi hanya menggunakan tiga tingkat kematangan dengan kelas mentah, matang, dan busuk. Penelitian yang akan dilakukan pengklasifikasian akan menggunakan empat kelas data yaitu mentah, kurang matang, matang, dan terlalu matang.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7	Palm oil classification using deep learning	Abdulrazak Yahya Salch, Ermawatih Liansim, Science in Information Technology Letters, 2020	Menerapkan dan menganalisis ketepatan deep learning untuk mengklasifikasikan buah masak pada buah kelapa sawit.	Berhasil mendeteksi dan membedakan kematangan buah kelapa sawit dengan metode CNN, dengan akurasi 98%.	Pekerjaan di masa mendatang harus mendapatkan lebih banyak data karena algoritma deep learning sering bekerja lebih baik dengan lebih banyak data.	Penelitian sebelumnya pada proses klasifikasi hanya menggunakan dua tingkat kematangan dengan kelas mentah dan matang. Penelitian yang akan dilakukan pengklasifikasian akan menggunakan enam kelas data yaitu mentah, kurang matang, matang, dan terlalu matang dengan menambahkan lebih banyak data citra TBS.
8	Oil Palm Fruit Image Ripeness Classification with Computer Vision using Deep Learning and Visual Attention	Herman, Albert Susanto, Tjeng Wawan Cenggoro, Suharjo and Bens Pardamean, Journal of Telecommunication, Electronic and Computer Engineering, 2020	Untuk mengembangkan model menggunakan mekanisme perhatian berbasis residual yang dapat mengenali perbedaan detail kecil antara gambar	dihasilkan secara residual dari jaringan saraf telah terbukti berguna sebagai perpanjangan untuk menghasilkan perhatian dengan melibatkan hasil dari setiap blok pada jaringan saraf yang	Pekerjaan masa depan dapat mengeksplorasi penggunaan strategi mekanisme perhatian lain yang dapat mengatasi masalah gradien hilang, seperti kerugian ekstra setelahnya.	Pada penelitian terdahulu AlexNet, DenseNet + SE Layer, ResAtt DenseNet, DenseNet Sigmoid untuk melakukan klasifikasi tingkat kematangan TBS, namun selama proses klasifikasi mengalami masalah gradien hilang. Pada penelitian yang akan

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				dalam seperti DenseNet. Akurasi yang dihasilkan dengan membandingkan empat model masih berada dibawah 70%.		dilakukan untuk mengatasi masalah gradien hilang, ResNet diajukan. melompati atau melewati beberapa layer ResNet berhasil menyelesaikan masalah yang ada atau dikenal Residual learning berhasil mengatasi gradien hilang.

2.3. Landasan Teori

Bagian ini memuat rangkuman teori-teori yang diambil dari buku atau literatur yang mendukung penelitian, serta memuat penjelasan tentang konsep dan prinsip dasar yang diperlukan untuk pemecahan permasalahan. Landasan teori dapat berbentuk uraian kualitatif, model matematis, atau tools yang langsung berkaitan dengan permasalahan yang diteliti. Sumber teori yang dirujuk pada bagian ini harus dicantumkan dalam kalimat atau pernyataan yang diacu dan dicantumkan dalam Daftar Pustaka.

2.3.1. Klasifikasi

Klasifikasi merupakan proses pengelompokan suatu data ke dalam sebuah kelompok kelas suatu data tertentu dengan mencirikan konsep atau kelas data yang telah ada atau didefinisikan sebelumnya untuk mencapai tujuan tertentu (Andono et al., 2017; Suyanto, 2017). Sedangkan menurut Han dan Kamber klasifikasi didefinisikan sebuah proses mendapatkan sebuah model yang memiliki tujuan untuk menggambarkan, mengenali, dan membedakan kelas pada data, model yang telah didapatkan digunakan untuk memprediksi kelas pada objek yang belum diketahui kelasnya (Arhami & Nasir, 2020). Klasifikasi merupakan kegiatan sehari-hari yang sering dijumpai, seperti petani jeruk mengklasifikasi hasil panen berdasarkan warna kulit dan ukurannya, dokter mengklasifikasi penyakit berdasarkan gejala yang ada pada pasien, sistem keamanan pada email melakukan klasifikasi untuk membedakan jenis email berdasarkan spam atau bukan spam, dan masih banyak lagi.

Klasifikasi dibagi menjadi dua, yaitu terawasi (*supervised classification*) dan tidak terawasi (*unsupervised classification*). Klasifikasi terawasi terjadi jika sebelumnya kelas klasifikasi sudah diketahui, klasifikasi terawasi kembali terbagi menjadi dua kelompok, dimana berdasarkan digunakan atau tidaknya parameter statistik. Klasifikasi yang menggunakan ukuran statistik secara eksplisit dikenal sebagai klasifikasi parametrik dan klasifikasi yang tidak menggunakan ukuran statistik adalah klasifikasi non-parametrik. Sedangkan klasifikasi tidak terawasi terjadi jika kelas klasifikasi ditemukan selama tahapan training.

Tahapan yang terjadi pada saat pengklasifikasian biasanya terbagi menjadi dua tahapan, yaitu tahapan training dan testing. Didalam sebuah tahapan training sifat unik yang ada pada fitur dalam sebuah citra dipisahkan, hal ini dilakukan untuk mendapatkan deskripsi unik pada kelas training pada setiap kategori klasifikasi. Sedangkan pada tahapan testing deskripsi unik yang telah didapatkan pada tahapan training menjadi komponen penting dari sebuah proses klasifikasi, fitur citra testing dikelompokkan berdasarkan kelas yang didapatkan sebelumnya (Arnia & Munadi, 2018).

Model klasifikasi dibangun dengan menggunakan teknik pembelajaran mesin. Pembelajaran biasanya dilakukan secara otomatis pada suatu kumpulan data untuk menghasilkan sebuah model klasifikasi (fungsi target) yang akan memetakan objek data x (input) ke dalam sebuah kelas y yang telah didefinisikan sebelumnya. Proses pada sebuah pembelajaran biasanya terdiri dari masukan(input) yang berupa kumpulan data latih (training set) yang telah diberi label dan menghasilkan sebuah model sebagai output (Suyanto, 2017).

2.3.2. Definisi Citra

Secara visual, citra merupakan perwakilan dari sekumpulan informasi yang terkandung didalamnya, nantinya mampu dianalisis dan ditafsirkan oleh mata manusia sehingga informasi yang terkandung tersampaikan sesuai dengan tujuan yang ingin dicapai. Informasi yang terkandung dalam sebuah citra terbagi menjadi dua yaitu informasi dasar yang merupakan sebuah informasi yang dapat diolah secara langsung tanpa membutuhkan pengetahuan khusus, informasi dasar dapat berupa warna (*color*), bentuk (*shape*), dan tekstur (*texture*) analisis yang dilakukan pada informasi ini disebut *low level image analysis* dan informasi abstrak yang memerlukan bantuan pengetahuan khusus dalam proses pengolahan informasi.

Secara matematis, citra adalah fungsi dua dimensi $f(x, y)$, dan merupakan koordinat spasial (*plane*) dan f merupakan nilai intensitas warna pada koordinat x dan y , nilai yang terkandung di dalam fungsi dua dimensi adalah nilai terhitung. Didalam sebuah citra terdapat elemen matriks yang disebut *pixel* (*picture element*, *image element* atau *pel*), dimana setiap *pixel* f pada koordinat (x, y) mewakili warna. Dimana citra berwarna diwakili dengan kode 24 bit dengan tiga komponen warna RGB, sedangkan untuk citra *gray-level* dikodekan dengan 8 bit atau 1 bit citra biner (Madenda & Drajat, 2015).

2.3.3. Pengolahan Citra

Pengolahan citra yang merupakan bagian dari ilmu informatika berfungsi untuk mengoreksi kualitas citra agar didapati citra dengan kualitas yang baik, sehingga mampu diinterpretasi baik komputer maupun manusia dengan masukan berupa citra dan keluaran yang dihasilkan dari proses pengolahan juga berupa citra

(Hidayatullah, 2017). Menurut Efford, pengolahan citra merupakan istilah yang biasa disematkan pada teknik yang digunakan untuk memanipulasi dan modifikasi citra dengan berbagai metode. Foto merupakan citra dua dimensi yang pengolahannya dapat dilakukan dengan mudah, salah satunya citra digital (Kadir & Susanto, 2013).

Dalam pengolahan citra ada beberapa prinsip dasar yang digunakan, (Kadir & Susanto, 2013) dan (Putra, 2010) memiliki pendapat yang sama bahwa prinsip dasar dalam pengolahan citra yaitu peningkatan kecerahan dan kontras, penghilangan derau, dan pencarian bentuk objek, tetapi Putra mendefinisikan lebih lanjut dengan membagi dalam 3 kategori berikut:

1. Kategori rendah yang melibatkan operasi sederhana seperti pengurangan derau.
2. Kategori menengah yang melibatkan operasi klasifikasi dan segmentasi.
3. Kategori tinggi yang terjadi pada proses pengenalan dan deskripsi citra.

2.3.4. Evaluasi Kualitas Klasifikasi

Menurut (Andono et al., 2017) untuk mengukur kualitas sebuah citra terdapat dua cara, yang pertama dilakukan secara subjektif yaitu menilai dengan didasari pada persepsi manusia. Yang kedua dengan cara objektif, menilai dengan metode kuantitatif untuk menghasilkan pengukuran standar, cepat, mudah, dan murah. Pengukuran secara objektif dilakukan tanpa keterlibatan manusia, dengan menganalisis citra sebelum diproses dengan citra setelah diproses (citra terdistorsi). Pengukuran secara objektif dapat digunakan untuk membandingkan model klasifikasi yang dihasilkan, mengawasi klasifikasi untuk memastikan klasifikasi berjalan dengan baik. Salah satu pengukuran objektif yang digunakan untuk

evaluasi kinerja sistem klasifikasi adalah confusion matrix, yang berisi informasi hasil prediksi dan data aktual dari sebuah model klasifikasi. Terdapat empat istilah dalam sebuah tabel confusion matrix yaitu:

1. True positif (TP) merupakan jumlah hasil positif yang dilabeli benar oleh model klasifikasi.
2. True negatif (TN) merupakan jumlah hasil negatif yang dilabeli benar oleh model klasifikasi.
3. False positif (FP) merupakan jumlah hasil negatif yang salah dilabeli oleh model klasifikasi.
4. False negatif (FN) merupakan jumlah hasil positif yang salah dilabeli oleh model klasifikasi.

Evaluasi yang dilakukan terhadap sebuah model klasifikasi dilakukan dengan menghitung suatu ukuran pada kumpulan data uji yang tidak digunakan dalam pelatihan model klasifikasi. Ada beberapa ukuran yang digunakan untuk mengevaluasi model klasifikasi yaitu accuracy, error rate, recall, specificity, precision, dan F1 atau F-score (Suyanto, 2017).

1. Accuracy merupakan rasio dari jumlah hasil prediksi yang benar dan jumlah total data prediksi yang dirumuskan pada Persamaan 2.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots \dots \dots (2.1)$$

2. Recall berupa ukuran kelengkapan, dimana kelas positif yang diprediksi sebagai kelas positif dengan menggunakan Persamaan 2.2.

$$Recall = \frac{TP}{TP + FN} \dots \dots \dots (2.2)$$

3. Specificity merupakan jumlah kelas negatif yang diprediksi dengan benar, atau rasio jumlah data negatif yang diprediksi negatif dirumuskan pada Persamaan 2.3.

$$Specificity = \frac{TN}{TN + FP} \dots \dots \dots (2.3)$$

4. Precision merupakan ukuran kepastian, kelas positif yang diprediksi benar sebagai kelas positif dilakukan seperti pada persamaan 2.4.

$$Precision = \frac{TP}{TP + FP} \dots \dots \dots (2.4)$$

5. F-Score dapat dikatakan sebagai rata-rata bobot dari nilai recall dan precision, nilai F-score idealnya berada pada angka 1 dan angka 0 menjadi nilai terburuk. F-score pada persamaan 2.5 disebut juga f1.

$$F = \frac{2 \times precision \times recall}{precision + recall} \dots \dots \dots (2.5)$$

2.3.5. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) dikenalkan pertama kali pada tahun 1987, pada tahun tersebut teknologi pada saat itu masih sangat terbatas, baru pada tahun 2012 mulai populer melalui architecture AlexNet. Pada sebuah CNN dengan lapisan yang lebih dalam dari rekan aslinya, memenangkan kompetisi klasifikasi citra ILSVRC-2012. Sejak tahun 2012, banyak penelitian yang dilakukan menunjukkan peningkatan kinerja pada dataset ImageNet (Susanto et al., 2020).

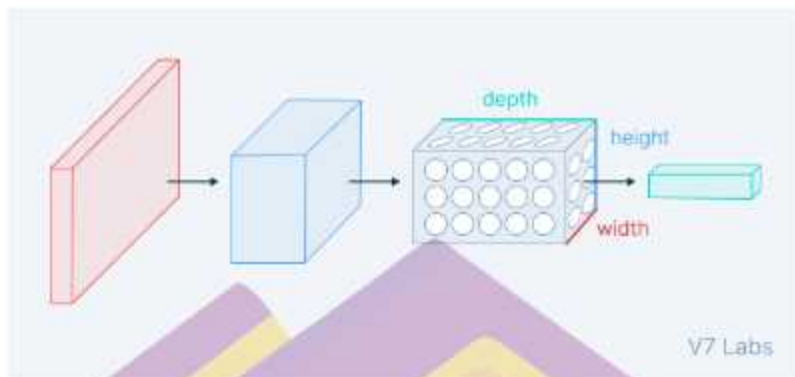
CNN salah satu metode dalam deep learning yang mampu mengklasifikasi dengan baik sebuah data citra, atas dasar hal tersebut CNN sering digunakan pada

pengaplikasi citra. CNN didasari pada proses biologi hubungan antar neurons yang menyerupai kumpulan visual cortex pada mamalia. CNN dapat belajar secara spontan dari masukan, ekstraksi fitur, dan detail kontekstual yang mampu mengurangi sedikit kesalahan saat pengenalan gambar. CNN secara khusus meniru cara otak manusia bekerja dalam mengenali sebuah objek (Primartha, 2021).

2.3.5. Arsitektur Convolutional Neural Network (CNN)

Arsitektur CNN terdiri dari 3 lapisan yaitu satu lapisan masukan (input layer), satu lapis keluaran (output layer), dan sejumlah lapisan tersembunyi (hidden layer). lapisan tersembunyi biasanya terdiri dari convolutional layers, pooling layer, normalization layer, ReLu layer, fully connected layers, dan loss layer. CNN juga menggunakan 3 arsitektur tiga dimensi yaitu lebar (width), tinggi (height), dan dalam (depth) yang divisualisasikan pada Gambar 2.1, dimana setiap lapisan CNN mentransformasikan volume masukan tiga dimensi pada volume keluaran tiga dimensi aktivasi sel saraf. Perkembangan yang terjadi pada CNN tergolong cepat dalam beberapa tahun, para ahli telah mendesain CNN dengan ratusan sampai ribuan lapisan (Suyanto et al., 2019). Banyak arsitektur yang telah dikembangkan oleh para ahli antara lain sebagai berikut:

1. LeNet-5 merupakan CNN pertama yang berhasil diaplikasikan pada pembacaan kode pos dan digit, LeNet-5 dibangun pada tahun 1990-an oleh Yann LeCun.

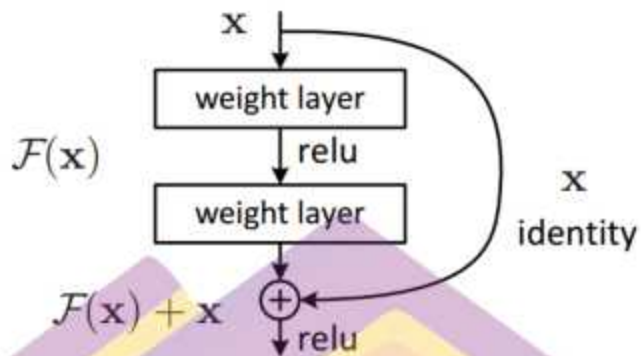


Gambar 2.1. Arsitektur CNN secara umum

2. AlexNet ditemukan oleh Alex Krizhevsky, Ilya Sutskever, dan Geoff Hinton pada tahun 2012 yang sangat populer untuk digunakan pada computer vision. AlexNet memiliki fitur yang hampir sama dengan LeNet tetapi terdapat pembaharuan pada convolutional layer yang ditumpuk pada bagian atas.
3. ZFNet merupakan salah satu arsitektur untuk perbaikan AlexNet, perbaikan ZFNet terdapat pada hyperparameters, melakukan perluasan pada convolutional layer serta memperkecil stride dan ukuran filter lapisan pertama. ZFNet dibangun oleh Matthew Zeiler dan Rob Fergus pada tahun 2013.
4. GoogleNet dibangun pada tahun 2014 oleh Szegedy dari Google. Pada GoogleNet dibangun inception Module yang berfungsi untuk mereduksi parameter menjadi 4 juta serta menggunakan average pooling untuk menggantikan fully connected layers pada bagian atas untuk menghilangkan parameter yang tidak dibutuhkan.
5. VGGNet dibangun pada tahun yang sama dengan GoogleNet yaitu pada tahun 2014 oleh Karen Simonyan dan Andrew Zisserman, dimana VGGNet menjadi juara

kedua pada ILSVRC 2014 hanya kalah dari GoogleNet di posisi pertama. VGGNet menunjukkan bahwa untuk menghasilkan performa yang tinggi memerlukan kedalaman jaringan, VGGNet memiliki jaringan terbaik dengan 16 convolutional fully connected layers dengan konvolusi 3x3 dan pooling 2x2 dari lapisan pertama dan terakhir. Namun parameter yang dihasilkan mencapai 140 juta yang membuat VGGNet membutuhkan banyak memori.

6. ResNet atau residual network, memiliki ciri khusus yaitu skip connections, batch normalization, dan menghilangkan fully connected layers yang berhasil menyelesaikan masalah yang ada pada arsitektur sebelumnya keadaan jika jaringan yang ada terlalu dalam mengakibatkan penurunan akurasi dan tingginya error yang dihasilkan. Dengan cara melompati atau melewati beberapa layer ResNet berhasil menyelesaikan masalah yang ada sebelumnya vanishing gradient problem, dan cara penyelesaian dikenal dengan residual learning yang divisualisasikan pada Gambar 2.2. ResNet dengan kemampuan tersebut menjadi standar dalam penggunaan secara praktis dan menjadi Arsitektur terbaik sejak 2016, yang dibangun oleh Kaiming pada tahun 2015.

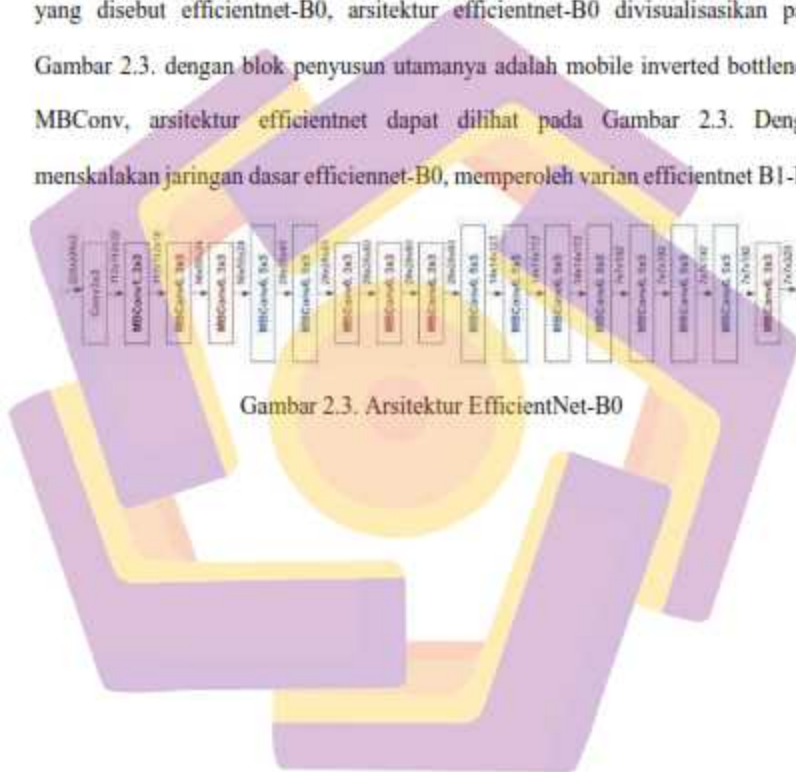


Gambar 2.2. Residual Learning Block

7. CapsNet atau dikenal juga sebagai Capsule network model yang ditemukan pada tahun 2017 oleh Hilton. Capsule merupakan kumpulan neuron yang terdiri dari beberapa parameter dan panjang dari sebuah vektor yang menunjukkan kemungkinan keberadaan suatu entitas. Kemampuan CapsNet tidak hanya pada melihat fitur tetapi juga mampu melihat posisi, arah yang dimiliki suatu entitas dengan cara ini CapsNet mampu menghasilkan akurasi yang tinggi.

8. EfficientNet merupakan model yang diusulkan untuk meningkatkan akurasi dan efisiensi pada CNN dengan menerapkan model penskalaan yang sama untuk semua dimensi yaitu kedalaman, lebar, dan resolusi dengan model yang lebih kecil. CNN pada umumnya untuk penurunan atau kenaikan dilakukan dengan menyesuaikan ke dalam jaringan, melakukan penskalaan pada jaringan akan meningkatkan akurasi namun pada model yang lebih kompleks dan besar terjadi penurunan akurasi. Agar tidak terjadi penurunan akurasi, penskalaan dilakukan pada setiap dimensi kedalaman, lebar, dan resolusi (Jaisakthi et al., 2022). (Tan & Le, 2019) mempelajari dan melakukan identifikasi penskalaan ConvNet mendapati bahwa

bagian terpenting adalah menyeimbangkan kedalaman, lebar, dan resolusi tetapi terdapat kehilangan pada beberapa potongan. Mereka mengembangkan dasar jaringan dengan memanfaatkan pencarian arsitektur neural multi-objektif yang mengoptimalkan akurasi dan flops, menghasilkan sebuah jaringan yang efisien yang disebut efficientnet-B0, arsitektur efficientnet-B0 divisualisasikan pada Gambar 2.3. dengan blok penyusun utamanya adalah mobile inverted bottleneck MBCConv, arsitektur efficientnet dapat dilihat pada Gambar 2.3. Dengan menskalakan jaringan dasar efficiennet-B0, memperoleh varian efficientnet B1-B7.



Gambar 2.3. Arsitektur EfficientNet-B0

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini menggunakan jenis penelitian eksperimental, pada hal ini dilakukan berbagai tindakan dimulai dari pengumpulan data, pre-processing data, ekstraksi fitur, kemudian dilakukan pemodelan data menggunakan beberapa metode, kemudian melalui proses training dan testing data. Adapun sifat penelitian yang dilakukan dalam hal ini adalah secara mandiri, dengan menggunakan metode deskriptif pada data yang telah diperoleh dilakukan pelatihan dan dilakukan pengujian. Kemudian Hasil dari pengujian dicari akurasi hasil klasifikasi tingkat kematangan.

Dilakukan pendekatan kuantitatif, dalam menghitung tingkat akurasi klasifikasi tingkat kematangan TBS kelapa sawit. Hasil tersebut akan menghasilkan akurasi pengenalan tingkat kematangan dan keberhasilan pada empat tingkat kematangan TBS kelapa sawit yang diuji. Pada klasifikasi tingkat kematangan TBS kelapa sawit menggunakan metode convolutional neural networks terhadap data citra TBS kelapa sawit dengan empat tingkatan kematangan yang telah dikumpulkan.

3.2. Metode Pengumpulan Data

Bagian ini memuat penjelasan secara lengkap dan terinci tentang cara-cara yang digunakan dalam proses pengumpulan data untuk jenis data yang diperlukan.

Misalnya melalui observasi, wawancara, eksperimen, atau penyebaran angket. Jika metode penyebaran angket digunakan, maka blangko angket harus dilampirkan dalam proposal. Untuk setiap metode pengumpulan data, harus dijelaskan tentang jenis data yang dikumpulkan dengan metode-metode yang terkait. Bagian ini juga memuat penjelasan secara lengkap dan terinci tentang jenis data yang diperlukan untuk analisis dalam pembahasan.

Data yang dibutuhkan pada penelitian ini adalah citra atau gambar dari tandan buah segar kelapa sawit. Data citra untuk penelitian ini bersumber dari data publik yang dapat diakses secara bebas. Data diperoleh dari website yang menyediakan berbagai macam data citra yaitu roboflow.com. Data yang diambil merupakan data berjenis citra tandan buah segar kelapa sawit dengan 4.938 data citra yang terbagi dalam empat tingkat kematangan. Tingkat kematangan yang digunakan pada data citra dijelaskan pada Tabel 2.1. Sebaran data pada dataset yang dikumpulkan terdiri dari mentah berjumlah 331 data, kurang matang berjumlah 1548, matang berjumlah 2236 data, dan yang terakhir terlalu matang 723 data.

Tabel 3.1. Tingkat kematangan TBS kelapa sawit.

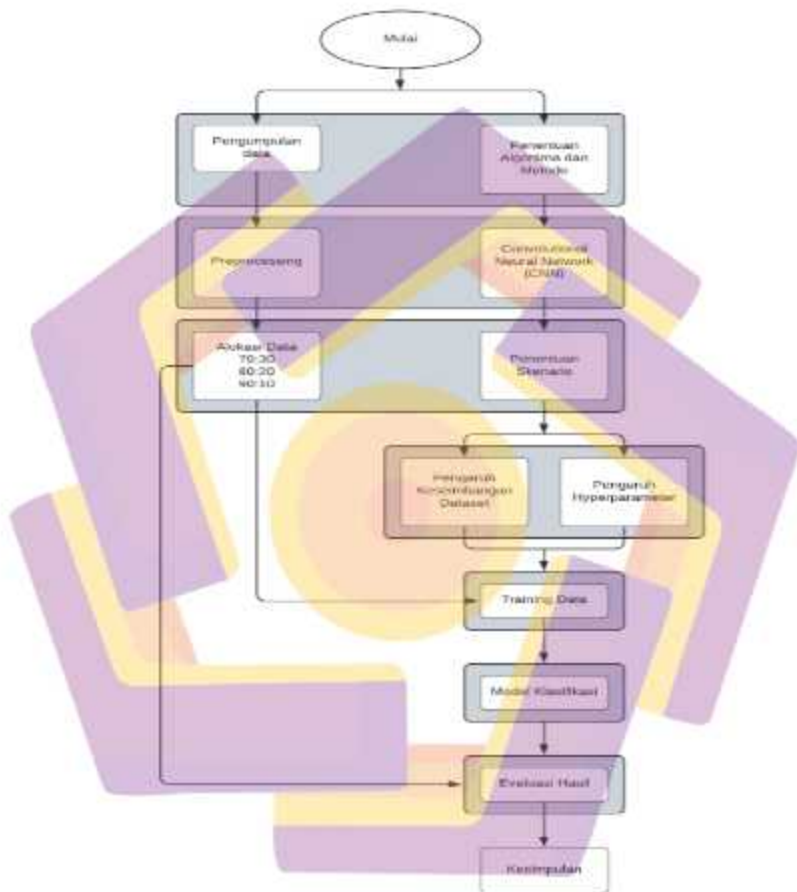
Kelas	Keterangan
Mentah	Buah pada tandan masih keras, berwarna hitam dan belum ada yang terlepas.
Kurang matang	Buah sudah berubah warna menjadi hitam kemerahan dan sudah ada buah yang terlepas satu sampai sembilan buah yang terlepas.
Matang	Buah pada tandan sudah mulai berubah menjadi jingga kemerahan dan 10 buah yang terlepas.
Terlalu matang	Buah pada tandan memiliki warna jingga kemerahan dan lebih dari 50 buah terlepas.

3.3. Metode Analisis Data

Pada penelitian ini dilakukan proses analisis data pada dataset yang telah diskenarioikan dalam penelitian ini yaitu 4.938 jumlah data tandan buah segar kelapa sawit yang terbagi dalam empat kelas terdiri dari mentah, kurang matang, matang dan terlalu matang. Metode analisis data yang akan digunakan pada penelitian ini adalah analisis kuantitatif dengan algoritma Convolutional Neural Network (CNN). Data yang telah dikumpulkan terlebih dahulu akan dilakukan proses preprocessing terhadap data sebelum dilanjutkan untuk dilakukan klasifikasi citra. Proses preprocessing yang akan dilakukan dimulai dengan menyamaratakan ukuran pixel dari citra TBS, melakukan labeling pada data citra dan melakukan augmentasi pada data. Pada data citra TBS yang telah dikumpulkan akan dilakukan penyesuaian kelas data dengan menambah data pada kelas minoritas dengan menggunakan metode algoritma Synthetic Minority Oversampling Technique (SMOTE) dan Random Under Sampling mengurangi jumlah sampel dalam kelas mayoritas sehingga proporsi antara kelas mayoritas dan minoritas menjadi lebih seimbang. Ini dicapai dengan menghilangkan beberapa sampel dari kelas mayoritas secara acak. Metode sampling digunakan untuk mengatasi ketidak seimbangan jumlah pada masing-masing kelas data.

Dalam penelitian ini percobaan dilakukan melalui pada 3 skenario utama terhadap klasifikasi tingkat kematangan menggunakan arsitektur CNN dengan transfer learning. Dalam melakukan klasifikasi masing-masing skenario akan diuji untuk mengetahui hasil akurasi dengan perhitungan menggunakan confusion matrix dan faktor yang mempengaruhi akurasi yang dihasilkan.

3.4. Alur Penelitian



Gambar 3.1. Alur Penelitian

Alur kerja dalam klasifikasi tingkat kematangan tandan buah segar kelapa sawit yang akan diteliti memiliki alur yang disesuaikan dengan alur Gambar 3.1 tersebut sebagai berikut:

1. Pengumpulan Data

Berdasarkan Gambar 3.1, Sebagai proses awal dilakukan pengumpulan dataset yang akan digunakan pada penelitian, data yang digunakan adalah tandan buah segar kelapa sawit yang terdiri dari empat kelas kematangan. Data ini diperoleh dari roboflow.com, penyedia data publik untuk computer vision. Selanjutnya, data ini akan digunakan untuk proses pelatihan dan pengujian model klasifikasi tingkat kematangan tandan buah kelapa sawit.

2. Pre-processing

Data yang sebelumnya dikumpulkan berasal dari dataset yang berbeda-beda, dimana data tersebut masih belum layak untuk digunakan pada proses training dan testing, maka akan dilakukan pre-processing terhadap data. Tahapan pre-processing data merupakan tahapan mengubah data mentah menjadi format data yang jelas. Tahapan dimulai dari load data. Untuk memastikan bahwa data sesuai dengan yang telah dikumpulkan sebelumnya, dilanjutkan dengan pengecekan data, disini pengecekan format data dilakukan dengan melakukan data check. Setelah selesai melakukan pengecekan data, maka dilakukan penyamaan ukuran data citra atau resizing citra pada proses penyeimbangan data. Pada penyeimbangan data juga dilakukan penyeimbangan kelas data pada kelas minoritas data latih dengan menggunakan algoritma Synthetic Minority Oversampling Technique (SMOTE).

3. Alokasi Data

Data yang akan digunakan pada proses modeling dan klasifikasi adalah data training dan data testing. Maka data yang akan dialokasikan hanya akan menjadi

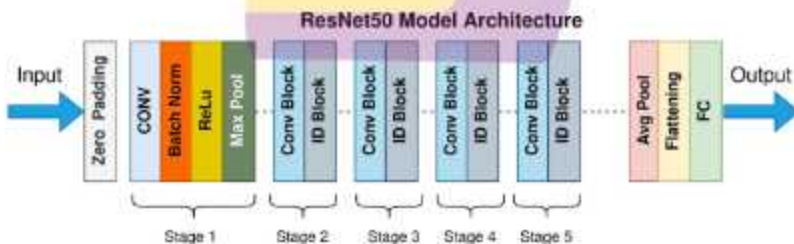
data training dan testing dengan tiga skenario pembagian data yaitu 70:30, 80:20, dan 90:10 alokasi data yang akan digunakan pada model.

4. Penentuan algoritma dan metode

Tahap ini merupakan penentuan algoritma yang akan digunakan, pada penelitian ini algorithm convolutional neural networks akan digunakan untuk melakukan klasifikasi tingkat kematangan tandan buah segar kelapa sawit.

a. ResNet50

ResNet-50 (Residual Network-50) diperkenalkan pada tahun 2015 oleh He et al. sebagai evolusi dari arsitektur ResNet yang lebih dalam dan sangat efektif dalam tugas pengenalan gambar. ResNet atau residual network, memiliki ciri khusus yaitu skip connections, batch normalization, dan menghilangkan fully connected layers yang berhasil menyelesaikan masalah yang ada pada arsitektur sebelumnya keadaan jika jaringan yang ada terlalu dalam mengakibatkan penurunan akurasi dan tingginya error yang dihasilkan. Dengan cara melompati atau melewati beberapa layer ResNet berhasil menyelesaikan masalah yang ada sebelumnya vanishing gradient problem, dan cara penyelesaian dikenal dengan residual learning.



Gambar 3.2. Arsitektur ResNet50

Pada Gambar 4.2, diilustrasikan tentang struktur arsitektur ResNet50 terdiri dari lima puluh lapisan, arsitektur ResNet-50 mencakup pooling, fungsi aktivasi ReLU, batch normalization, dan lapisan konvolusi. Selain itu, lapisan konvolusi dengan ukuran kernel 1×1 digunakan untuk mengurangi dimensi fitur dalam beberapa blok residu, sehingga mengurangi beban komputasi.

b. EfficientNetB0

EfficientNet merupakan model yang diusulkan untuk meningkatkan akurasi dan efisiensi pada CNN dengan menerapkan model penskalaan yang sama untuk semua dimensi yaitu kedalaman, lebar, dan resolusi dengan model yang lebih kecil. CNN pada umumnya untuk penurunan atau kenaikan dilakukan dengan menyesuaikan ke dalam jaringan, melakukan penskalaan pada jaringan akan meningkatkan akurasi namun pada model yang lebih kompleks dan besar terjadi penurunan akurasi. Agar tidak terjadi penurunan akurasi, penskalaan dilakukan pada setiap dimensi kedalaman, lebar, dan resolusi. Dilakukan pengembangan dasar jaringan dengan memanfaatkan pencarian arsitektur neural multi-objektif yang mengoptimalkan akurasi dan flops, menghasilkan sebuah jaringan yang efisien yang disebut efficientnet-B0. Struktur dasar EfficientNet-B0 terdiri dari blok berulang yang terdiri dari lapisan konvolusi, fungsi aktivasi, dan normalisasi batch. Selain itu, ada blok khusus yang disebut MBCnv (Mobile Inverted Residual Bottleneck Convolution), yang membantu mengekstraksi fitur dengan efisien.



Gambar 3.3. Arsitektur EfficientNetB0

EfficientNet-B0 memiliki 26 lapisan secara keseluruhan. Arsitektur ini terdiri dari berbagai jenis blok, seperti blok convolutional dan blok MBConv. Ini juga termasuk lapisan-lapisan tambahan, seperti lapisan penyesuaian, normalisasi batch, dan pooling yang diilustrasikan pada gambar 4.3.

5. Penentuan skenario

Pada penelitian ini akan menggunakan dua arsitektur CNN yaitu Resnet50 dan EfficientNetB0. Dalam penelitian ini skenario klasifikasi tingkat kematangan tandan buah segar kelapa sawit akan dilakukan dengan dataset yang seimbang dan tidak seimbang, skenario alokasi data dan hyperparameter tuning. Semua skenario yang akan dilakukan berjumlah 3 skenario pada masing-masing arsitektur CNN. Skenario percobaan pada penelitian ini untuk menarik analisis perbandingan dari hasil percobaan.

6. Training dan klasifikasi

Data yang telah melewati beberapa tahapan selanjutnya akan dilakukan tahapan training, tahapan ini bertujuan untuk mendapatkan model dari proses training data menggunakan arsitektur dari convolutional neural networks yaitu ResNet dan EfficientNet dengan menggunakan parameter yang berbeda pada proses training citra TBS. Setelah model didapatkan dari proses training, selanjutnya akan dilakukan tahapan testing, testing dilakukan untuk mendapatkan hasil kinerja dari

model yang telah didapatkan sebelumnya, dengan melakukan evaluasi kinerja apakah telah sesuai dengan hasil yang diharapkan.

7. Evaluasi

Tahapan selanjutnya pada alur penelitian ini adalah evaluasi. Evaluasi dilakukan untuk menarik hasil klasifikasi dan kesimpulan dari hasil pengujian model. Model evaluasi yang akan digunakan untuk proses evaluasi adalah confusion matrix, dengan menghitung akurasi, recall, dan precision untuk mendapatkan model terbaik dari model yang telah dilatih dan dilakukan testing.

8. Kesimpulan

Tahapan akhir dari penelitian ini adalah menarik kesimpulan dari hasil percobaan yang telah dilakukan. Pada tahapan ini akan memaparkan temuan terkait model mana dengan kondisi paling baik serta pengaruh keseimbangan data dan hyperparameter terhadap akurasi yang dihasilkan melalui scenario percobaan klasifikasi.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Bagian ini memberikan penjelasan tentang temuan dan hasil dari penelitian, yang dilakukan sesuai dengan alur penelitian yang digambarkan pada Gambar 3.1. Penelitian ini dimulai dengan tahapan pengumpulan data dan berakhir pada tahap evaluasi model klasifikasi untuk menentukan model yang paling cocok untuk klasifikasi tingkat kematangan tandan buah segar kelapa sawit.

4.1. Pengumpulan Data

Pada penelitian, data yang digunakan adalah tandan buah segar kelapa sawit yang terdiri dari empat kelas kematangan. Data ini diperoleh dari roboflow.com, penyedia data publik untuk computer vision. Selanjutnya, data ini akan digunakan untuk proses pelatihan dan pengujian model klasifikasi tingkat kematangan tandan buah kelapa sawit.





Tabel 4.1. Detail dataset tingkat kematangan.

Tingkat Kematangan	Jumlah Data	Sumber Data
Mentah	331	Roboflow
Kurang Matang	1548	
Matang	2336	
Terlalu Matang	723	
Total Data	4938	

Tabel 4.1 menunjukkan informasi lebih lanjut tentang data yang digunakan dalam penelitian ini, yang mencakup 4938 data citra untuk empat kelas kematangan, dengan sebaran data tingkat kematangan yang mencakup 331 data

kelas mentah, 1548 data kelas kurang matang, 723 data kelas matang, dan 723 data kelas terlalu matang. Tabel 4.2 menunjukkan sampel dari masing-masing kelas citra TBS.

Tabel 4.2. Sampel data citra TBS

Kelas Data	Citra TBS
Mentah	
Kurang Matang	
Matang	
Terlalu Matang	

Dalam pelaksanaannya, bagian ini dapat digunakan untuk membandingkan hasil-hasil penelitian yang diperoleh dalam penelitian yang sedang dilakukan terhadap hasil-hasil penelitian yang dilaporkan oleh peneliti terdahulu yang diacu pada penelitian ini. Secara ilmiah, hasil penelitian yang diperoleh dalam penelitian dapat berupa temuan baru atau perbaikan, penegasan, atau penolakan interpretasi suatu fenomena ilmiah dari peneliti sebelumnya.

4.2. Skenario Penelitian

Pada penelitian ini skenario percobaan menggunakan beberapa skenario yang berasal dari 3 skenario utama yaitu alokasi data, hyperparameter tuning dan teknik penyeimbangan data. Percobaan pada skenario yang dilakukan untuk mendapatkan hasil analisis untuk menentukan model klasifikasi terbaik pada tingkat kematangan TBS. Rincian terkait skenario percobaan yang akan dilakukan pada penelitian ini ditampilkan pada Tabel 4.3

Tabel 4.3. Sampel data citra TBS

No	Skenario	Keterangan
1	S1	Arsitektur+Optimizer
2	S2	Arsitektur+Learning rate
3	S3	Arsitektur+Batch size
4	S4	Arsitektur+Epoch
5	S5	Arsitektur+Alokasi Data
6	S6	Arsitektur+Penyeimbangandata
7	S7	Arsitektur+Dropout

Pada penelitian ini skenario percobaan menggunakan beberapa skenario yang berasal dari 3 skenario utama yaitu alokasi data, hyperparameter tuning dan

teknik penyeimbangan data. Percobaan pada skenario yang dilakukan untuk mendapatkan hasil analisis untuk menentukan model klasifikasi terbaik pada tingkat kematangan TBS. Rincian terkait skenario percobaan yang akan dilakukan pada penelitian ini ditampilkan pada Tabel 4.3

Skenario-skenario yang ditampilkan pada Tabel 4.3, berfungsi untuk mengidentifikasi model klasifikasi yang terbaik untuk tingkat kematangan tandan buah segar kelapa sawit. Penjelasan terkait skenario yang digunakan adalah sebagai berikut.

1. Skenario percobaan 1 (S1)

Pada skenario percobaan pertama, dilakukan pencari optimizer terbaik untuk model klasifikasi. Dua optimizer yang akan dibandingkan adalah Adam (Adaptive Moment Estimation) dan SGD (Stochastic Gradient Descent). Percobaan ini akan dilakukan pada dataset tandan buah segar kelapa sawit yang belum diseimbangkan, dengan menggunakan arsitektur model ResNet50 dan EfficientNetB0 seperti yang ditunjukkan pada Gambar 4. Pada percobaan ini, menggunakan parameter awal learning rate sebesar 0.0001, jumlah epoch sebanyak 100, batch size sebesar 16, dropout sebesar 0.5, dan alokasi data dengan perbandingan 80% untuk data training, 10% untuk data validasi, dan 10% untuk data testing. Dengan membandingkan kinerja optimizer Adam dan SGD, ingin menentukan optimizer mana yang memberikan hasil terbaik dalam melatih model klasifikasi untuk dataset kelapa sawit.

2. Skenario percobaan 2 (S2)

Pada skenario percobaan kedua, melakukan pencarian learning rate terbaik untuk model klasifikasi. Nilai learning rate yang akan dieksplorasi adalah 0.0001, 0.00035, 0.0005, 0.00075, dan 0.0009. Percobaan ini akan menggunakan parameter dan arsitektur model ResNet50 dan EfficientNetB0 seperti yang digunakan pada skenario sebelumnya. Namun, dalam skenario ini, hanya akan menggunakan optimizer Adam, yang telah diuji pada skenario sebelumnya, untuk proses pencarian learning rate terbaik. Percobaan ini ingin menentukan learning rate mana yang paling optimal dalam melatih model klasifikasi untuk dataset kelapa sawit.

3. Skenario percobaan 3 (S3)

Pada skenario percobaan ketiga, melakukan analisis pengaruh batch size terhadap performa model klasifikasi menggunakan arsitektur ResNet50 dan EfficientNetB0. Batch size mengacu pada jumlah gambar yang digunakan dalam satu iterasi pelatihan untuk melakukan propagasi maju dan mundur dalam algoritma pembelajaran. Pada percobaan ini, menggunakan batch size sebesar 8, 16, dan 32. Percobaan ini bertujuan untuk mengevaluasi bagaimana batch size dapat mempengaruhi kinerja model. Untuk melakukan ini, akan membandingkan hasil dari masing-masing ukuran batch untuk menentukan ukuran batch mana yang akan memberikan hasil terbaik dalam hal konvergensi model dan akurasi klasifikasi pada kumpulan data kelapa sawit yang digunakan dalam penelitian ini.

4. Skenario percobaan 4 (S4)

Dalam skenario percobaan keempat, hyperparameter yang lain akan dilakukan pengujian, yaitu epoch. Epoch menentukan berapa kali seluruh set data pelatihan akan diberikan kepada model selama proses pelatihan. Jumlah epoch yang

dipilih dapat mempengaruhi konvergensi dan kinerja model. Percobaan ini akan menguji tiga nilai epoch, yaitu 50, 100, dan 150. Tujuannya adalah untuk melihat bagaimana kinerja model akan berubah dengan nilai-nilai ini. Parameter pengujian lainnya, seperti arsitektur model, optimizer, learning rate, batch size, dan alokasi data, akan tetap menggunakan nilai yang sama seperti pada skenario sebelumnya. Dengan menganalisis hasil dari ketiga epoch yang diuji, akan dapat menentukan jumlah epoch yang memberikan hasil terbaik untuk dataset kelapa sawit yang digunakan.

5. Skenario percobaan 5 (S5)

Skenario percobaan kelima adalah mengevaluasi dampak penggunaan nilai dropout pada model arsitektur klasifikasi. Nilai dropout yang akan digunakan pada percobaan adalah 0.3 dan 0.5. Dropout adalah teknik regularisasi yang digunakan untuk menghindari overfitting pada model. Percobaan ini akan menunjukkan seberapa efektif penggunaan berbagai nilai dropout untuk memperbaiki performa dan mengurangi overfitting pada model arsitektur ResNet50 dan EfficientNetB0.

6. Skenario percobaan 6 (S6)

Dalam skenario ketujuh, untuk melihat pengaruh alokasi data pada model klasifikasi. Proses alokasi data melibatkan pembagian dataset yang tersedia menjadi subset yang berbeda, yaitu data training, data validasi, dan data testing. Dalam penelitian ini, kami akan menggunakan tiga skema alokasi data yang berbeda. Data dibagi menjadi 70% untuk training, 15% untuk validasi, dan 15% untuk testing dalam skema alokasi pertama. Dalam skema kedua, data dibagi menjadi 80% untuk training, 10% untuk validasi, dan 10% untuk testing. Dalam skema ketiga, data

dibagi menjadi 90% untuk training, 5% untuk validasi, dan 5% untuk testing. Percobaan ini menggunakan tiga skema alokasi data yang berbeda untuk melihat bagaimana alokasi data dapat mempengaruhi kinerja model klasifikasi.

7. Skenario percobaan 7 (S7)

Skenario percobaan yang terakhir untuk melihat pengaruh menyeimbangkan data training dan validasi akan dilakukan dengan menggunakan metode SMOTE (Synthetic Minority Oversampling Technique). Tujuan dari percobaan ini adalah untuk mengetahui bagaimana penyeimbangan data berdampak pada kinerja model. Arsitektur dan parameter pengujian yang akan digunakan akan sama dengan yang digunakan dalam skenario sebelumnya, yaitu arsitektur model, optimizer, learning rate, jumlah epoch dan batch size. Yang berbeda pada proses penyeimbangan data menggunakan SMOTE pada data training dan validasi sebelum melatih model. Tujuan penyeimbangan data ini adalah untuk mengetahui apakah kemampuan model untuk mengkategorikan kelas minoritas dapat ditingkatkan.

4.3. Preprocessing Data

Pada penelitian, data yang digunakan adalah tandan buah segar kelapa sawit yang terdiri dari empat kelas kematangan. Data ini diperoleh dari roboflow.com, penyedia data publik untuk computer vision. Selanjutnya, data ini akan digunakan untuk proses pelatihan dan pengujian model klasifikasi tingkat kematangan tandan buah kelapa sawit.

Salah satu proses penting pada bagian analisis data adalah preprocessing terhadap data. Preprocessing data adalah proses analisis data untuk meningkatkan kualitas data agar ideal untuk digunakan pada saat pemodelan.

```
[4] #create labels
classes=[]
filename='/content/gdrive/MyDrive/Thesis/Dataset/alldata98'
for sub_folder in os.listdir(os.path.join(filename,'train')):
    classes.append(sub_folder)
print(classes)

['Ripe', 'Unripe', 'Underripe', 'Overripe']
```

Gambar 4.1. Labeling data

Pada penelitian ini, tahapan pertama preprocessing adalah melakukan pelabelan terhadap data citra, proses pelabelan dilakukan dengan membuat daftar kosong kelas, selanjutnya membaca nama direktori sub-folder yang kemudian ditambahkan sebagai nama kelas ke dalam daftar kosong kelas sebelumnya seperti pada Gambar 4.1. Tahapan berikutnya melakukan resize terhadap data citra, karena data citra yang dikumpulkan memiliki ukuran yang berbeda. Ukuran yang digunakan untuk resize data citra adalah 224 x 224, ukuran ini digunakan mengacu pada penelitian terdahulu. Pada Gambar 4.2, proses dimana data citra dilakukan resize dan yang kemudian dilanjutkan dengan menggabungkan data training dan testing masing-masing kelas ke dalam satu folder yang sama.


```

[ ] #resize images and put together Training and Testing folder

X_train = []
y_train = []
image_size = 224
for i in classes:
    path_train = os.path.join(filename, 'train', i)
    for j in tqdm(os.listdir(path_train)): #instantly make your image show a smart progress enter
        img = cv2.imread(os.path.join(path_train, j), cv2.IMREAD_COLOR)
        img = cv2.resize(img, (image_size, image_size), interpolation=cv2.INTER_AREA)
        X_train.append(img)
        y_train.append(i)
    path_test = os.path.join(filename, 'test', i)
    for j in tqdm(os.listdir(path_test)):
        img = cv2.imread(os.path.join(path_test, j), cv2.IMREAD_COLOR)
        img = cv2.resize(img, (image_size, image_size), interpolation=cv2.INTER_AREA)
        X_train.append(img)
        y_train.append(i)

X_train = np.array(X_train)
y_train = np.array(y_train)

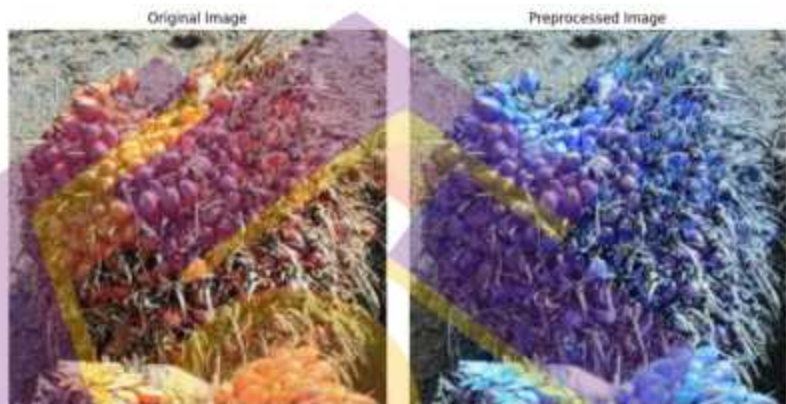
```

Gambar 4.2. Proses resize data

Tahapan setelah data diberi label dan resize, melakukan pembagian data untuk model klasifikasi. Pembagian data yang akan dilakukan menjadi data training dan testing. Ada 3 pembagian data atau alokasi data yang akan digunakan pada arsitektur ResNet50 dan efficientNetB0. Alokasi yang pertama yaitu 70/30, yang dimana 70% dari dataset akan digunakan untuk training model dan 30% dataset akan digunakan untuk testing, alokasi lainnya adalah 80/20. Proses alokasi data menggunakan fungsi sklearn model_selection import train_test_split untuk menjadi data training dan testing model.

Pada proses penggabungan dan resize gambar citra juga dilakukan memuat gambar dalam mode warna atau citra berwarna. Mode ini hanya mengambil saluran warna merah, hijau, dan biru dari gambar, mengabaikan saluran warna alfa (transparansi) menggunakan cv2.IMREAD_COLOR. Juga dilakukan proses interpolation=cv2.INTER_AREA, yaitu proses menghitung jumlah piksel baru yang ditemukan di bagian gambar yang salah setelah mengubah ukurannya. Metode

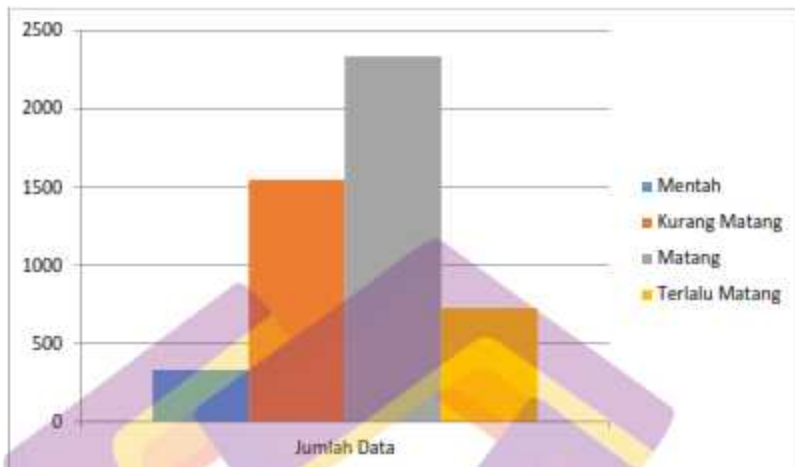
cv2.INTER_AREA melakukan interpolasi area piksel karena beberapa piksel akan dihilangkan ketika ukuran gambar dikurangi. Metode ini menghitung nilai piksel baru berdasarkan area piksel sekitarnya, Gambar hasil preprocessing direpresentasikan pada Gambar 4.3.



Gambar 4.3. Citra TBS sebelum dan setelah preprocessing

4.3.1. Penyeimbangan Data

Data yang dikumpulkan pada roboflow sebelumnya berjumlah 4938 data citra, dengan sebaran data antar kelas yang tidak seimbang. Ketidakseimbangan sebaran data dapat dilihat pada Gambar 4.4.



Gambar 4.4. Data tidak seimbang

Persebaran data pada dataset yang dikumpulkan menempatkan Data kelas mentah dan terlalu matang memiliki jumlah data paling sedikit pada dataset yang dikumpulkan; data mentah hanya 6,7% dari total data dan kelas terlalu matang 14,7%. Masalah keseimbangan data dapat diselesaikan dengan berbagai teknik penyeimbangan data, salah satunya adalah teknik augmentasi untuk menambah jumlah data gambar dan teknik sampling data dengan menambah data minoritas berdasarkan jumlah data mayoritas disebut *oversampling* atau melakukan reduksi data berdasarkan data minoritas dikenal dengan *undersampling*.

Ketidakseimbangan kelas terjadi ketika proporsi sampel dalam kumpulan data milik satu atau lebih kelas memiliki jarak secara signifikan (De Angeli et al., 2022). Pada penelitian ini proporsi antar kelas data tingkat keseimbangan data yang sangat ekstrim, dimana pada dua kelas data dari total keseluruhan data berada pada 6,7% dan 14,7%, proporsi data yang tidak seimbang ini tentunya akan

memunculkan masalah pada model klasifikasi. Masalah ketidakseimbangan kelas pada pengklasifikasian dapat diatasi dalam berbagai cara, seperti resampling data, teknik algoritmik dan hybrid. Teknik resampling data pada penelitian (Puri & Kumar Gupta, 2021) berfungsi dengan baik, menghasilkan nilai yang hilang yang lebih besar di antara set data kelas yang tidak seimbang. Dalam Teknik resampling data, dua paradigma paling penting adalah random oversampling (ROS), di mana sampel dari kelas minoritas digandakan, dan random undersampling (RUS), di mana sampel dari kelas mayoritas dibuang (De Angeli et al., 2022). Pada penelitian yang akan dilakukan untuk mengatasi masalah ketidakseimbangan kelas data, akan menggunakan teknik oversampling, yaitu Synthetic Minority Oversampling Technique (SMOTE). SMOTE merupakan teknik oversampling yang mengambil sampel data untuk minoritas dengan membuat sampel secara sintetik (Chawla et al., 2002). SMOTE sendiri bekerja secara acak memilih instansi minoritas p_i dari kelas minoritas P , dan kemudian menentukan m , sebagai tetangga terdekat p_i . Setelah menentukan m tetangga terdekat p_i , ia memilih tetangga acak m' : di mana $m' \in P$. Akhirnya, SMOTE membuat instance p_{new} baru menggunakan rumus berikut:

$$p_{new} = p_i + (m' - p_i) \times \delta, \delta \in [0, 1]$$

Rumus smote yang sebelumnya diterjemahkan ke dalam bahasa python divisualisasikan pada Gambar 4.5. Pada line pertama kode adalah untuk membuat objek smote yang akan digunakan untuk oversampling, smote diinisialisasikan dengan parameter default.

```

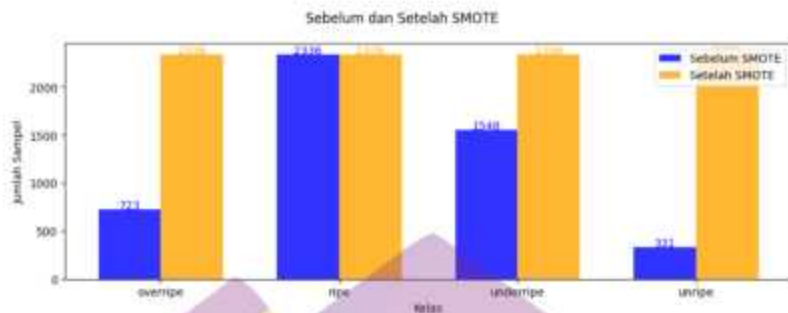
[ ] # Menggunakan SMOTE pada data training
smote = SMOTE()
X_train_resampled, y_train_resampled = smote.fit_resample(X_train.reshape(-1, image_size*image_size), y_train)

# kembali reshape data training setelah SMOTE
X_train_resampled = X_train_resampled.reshape(-1, image_size, image_size, 3)

```

Gambar 4.5. Smote pada python

Line kedua adalah untuk melakukan oversampling terhadap data dengan menggunakan metode `fit_resample()` dari Smote sebelumnya. Pada line ini `fit_resample()` mengambil dua parameter yaitu `X_train` dan `y_train`, pada line ini terjadi perubahan bentuk data `X_train` menjadi bentuk 2D dengan dimensi `(-1, image_size*image_size, 3)`. Perubahan ini bertujuan agar data dapat diterima oleh metode `fit_resample()`, yang kemudian disimpan pada variabel `X_train_resampled` dan `y_train_resampled`. Line yang terakhir berfungsi untuk mengembalikan bentuk data 2D ke bentuk aslinya yaitu 4D menggunakan `reshape()`. Data pada variabel `X_train_resampled` diubah menjadi 4D dengan dimensi `(-1, image_size*image_size, 3)` dengan melalui proses ini dataset telah melalui proses oversampling untuk penyeimbangan data. Dataset yang tidak seimbang sebelum SMOTE cenderung memiliki sejumlah besar sampel dari kelas mayoritas dan hanya sejumlah kecil sampel dari kelas minoritas.

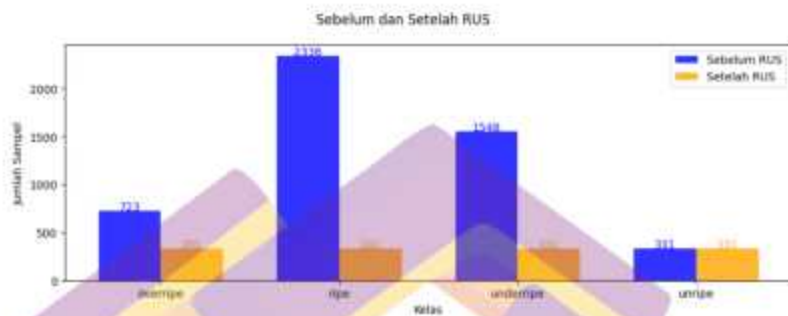


Gambar 4.6. Dataset setelah dan sebelum penerapan smote

Setelah menerapkan SMOTE, dataset yang tidak seimbang sebelumnya akan memiliki distribusi yang lebih seimbang antara kelas mayoritas dan kelas minoritas seperti pada Gambar 4.6. Persebaran data pada kelas minoritas sekarang sudah seimbang dengan kelas lainnya dimana setelah smote masing-masing kelas memiliki sejumlah 2336 data citra, sehingga setelah smote total keseluruhan data pada empat kelas menjadi 9344 data citra. Sampel sintesis yang dihasilkan oleh SMOTE akan meningkatkan representasi kelas minoritas dalam dataset, yang akan membantu model pembelajaran mesin belajar lebih baik dan lebih tepat dalam mengklasifikasikan kelas minoritas.

Pada proses penyeimbangan data, juga dilakukan metode random under sampling. Dimana data kelas mayoritas mengalami penyesuaian terhadap data minoritas, RUS sendiri merupakan kebalikan dari over sampling dimana sumber daya yang ada dibuang dan disesuaikan jumlahnya dengan data minoritas. Seperti yang ditampilkan pada Gambar 4.7, dimana data mayoritas seperti kelas data overripe, ripe dan under ripe disesuaikan jumlahnya dengan data kelas unripe,

dimana semua data pada setiap kelas menjadi 331 data citra sehingga total semua data menjadi 1.324 data citra.

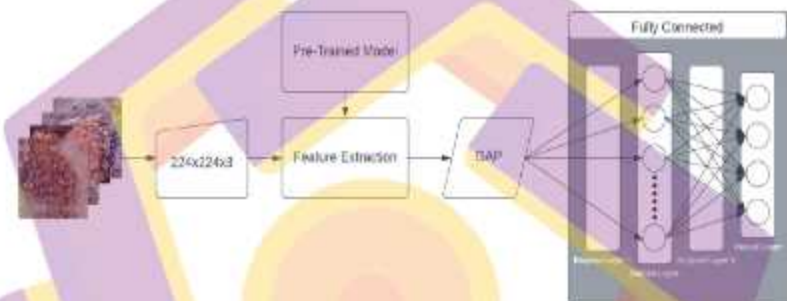


Gambar 4.7. Dataset setelah dan sebelum penerapan RUS

4.3.2. Penyesuaian Fully Connected Layer

Pada penelitian ini model yang akan digunakan terdiri dari tiga bagian ekstraksi fitur, Global Average Pooling (GAP) dan fully connected layer. Arsitektur CNN yang digunakan untuk ekstraksi fitur yang diterapkan dan dibandingkan *efficientnetb0* dan *resnet50*, arsitektur ini digunakan untuk ekstraksi fitur yang kemudian dilanjutkan dengan lapisan Global Average Pooling (GAP) dan lapisan klasifikasi pada layer selanjutnya. Pada proses klasifikasi ekstraksi fitur menggunakan arsitektur dengan transfer learning dengan tetap menggunakan bobot atau parameter asli dari ImageNet tanpa melakukan pelatihan ulang untuk memperbarui bobot yang ada pada lapisan masing-masing arsitektur, pada model ini yang akan mengalami perubahan bobot selama pelatihan hanya lapisan tambahan yang ada diatas arsitektur yaitu layer fully connected. Pada model ini dilakukan penyesuaian terhadap fully connected layer dengan modifikasi yang terdiri dari 1

layer dense tersembunyi, 1 layer dense keluaran dan 2 layer dropout sebelum dan sesudah layer tersembunyi. Lapisan dense tersembunyi berisi 60 neuron dengan aktivasi relu serta teknik inisialisasi bobot Glorotnormal, lapisan dropout masing-masing memiliki tingkat 0.5 dan lapisan output berisi 4 neuron yang mewakili 4 tingkat kematangan. Arsitektur CNN dengan transfer learning yang diusulkan divisualisasikan dalam Gambar 4.8.



Gambar 4.8. CNN dengan transfer learning dalam penelitian ini

Sebelum dilakukan proses training terhadap kedua arsitektur dilakukan tahapan penyesuaian, dilakukan dengan mengubah atau menambahkan struktur baru ke bagian tertentu dari jaringan. Sebagian atau seluruh jaringan disimpan dan kemudian dikirim ke domain target. Metode ini memiliki kemampuan untuk meningkatkan pelatihan pada data target dengan berbagai cara, seperti mengurangi biaya pelatihan dan mengatasi kebutuhan akan kumpulan data target yang luas. Untuk berbagai jenis tugas dan kumpulan data, seperti medis, mekanik, seni, fisika, keamanan, dll., metode ini sangat efektif (Ade Ramadhan 2020, Mohammadreza Iman 2023). Proses fine tuning pada model ini dengan mengambil bobot dari arsitektur yang telah dilatih dengan dataset imagenet, kemudian dilakukan

penyesuaian pada layer fully connected karena kedua arsitektur yang digunakan, menggunakan 1000 kelas pada imagenet sehingga perlu dilakukan penyesuaian kelas terhadap data yang digunakan penelitian ini dan menambah layer dense tersembunyi dan layer dropout.

conv5_block3_2_relu (Activation)	(None, 7, 7, 512)	0	['conv5_block3_2_bn[0][0]']
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1858624	['conv5_block3_2_relu[0][0]']
conv5_block3_3_bn (Batch Normalization)	(None, 7, 7, 2048)	8192	['conv5_block3_3_conv[0][0]']
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	['conv5_block2_out[0][0]', 'conv5_block3_3_bn[0][0]']
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	['conv5_block3_add[0][0]']
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	['conv5_block3_out[0][0]']
predictions (Dense)	(None, 1000)	2040000	['avg_pool[0][0]']

Total params: 25,636,712
 Trainable params: 25,583,592
 Non-trainable params: 53,120

Gambar 4.9. ResNet50 sebelum penyesuaian

Pada Gambar 4.9 ditunjukkan summary dari model arsitektur ResNet50 sebelum dilakukan proses penyesuaian, ekstraksi fitur yang dilakukan convolution layer dilanjutkan dengan global average pooling dan dilakukan prediksi pada layer dense dengan 1000 kelas. Dengan total parameter 25.636.712 yang terbagi menjadi parameter trainable 25.583.592 dan non-trainable 53.120 dengan jumlah layer dalam ResNet50 adalah 177 layer.

conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	['conv5_block3_2_relu[0][0]']
conv5_block3_3_bn (BatchNormal ization)	(None, 7, 7, 2048)	8192	['conv5_block3_3_conv[0][0]']
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	['conv5_block2_out[0][0]', 'conv5_block3_3_bn[0][0]']
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	['conv5_block3_add[0][0]']
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	['conv5_block3_out[0][0]']
dropout (Dropout)	(None, 2048)	0	['global_average_pooling2d[0][0]']
dense (Dense)	(None, 60)	122940	['dropout[0][0]']
dropout_1 (Dropout)	(None, 60)	0	['dense[0][0]']
dense_1 (Dense)	(None, 4)	244	['dropout_1[0][0]']

Total params: 23,716,896
 Trainable params: 23,657,776
 Non-trainable params: 53,120

Gambar 4.10. ResNet50 setelah penyesuaian

Setelah dilakukan penyesuaian terhadap arsitektur ResNet50 dapat dilihat pada summary yang ditampilkan Gambar 4.10, dapat dilihat setelah dilakukan penyesuaian setelah layer GAP terdapat tambahan layer dense tersembunyi dan 2 layer dropout. Dan layer klasifikasi telah disesuaikan dengan kelas yang ada pada dataset yaitu 4 neuron pada layer dense output yang melambangkan 4 kelas tingkat kematangan. Jumlah parameter juga telah mengalami penyesuaian menjadi 23.710.896 dengan trainable parameter 23.657.776 dan non-trainable 53.120. Dan jumlah layer bertambah menjadi 180 layer.

block7a_se_expand (Conv2D)	(None, 3, 1, 1152)	50688	['block7a_se_expand[0][0]']
block7a_se_excite (Multiply)	(None, 7, 7, 1152)	0	['block7a_activation[0][0]', 'block7a_se_expand[0][0]']
block7a_project_conv (Conv2D)	(None, 7, 7, 320)	308000	['block7a_se_excite[0][0]']
block7a_project_bn (BatchNormalizati	(None, 7, 7, 320)	1200	['block7a_project_conv[0][0]']
top_conv (Conv2D)	(None, 7, 7, 1200)	408000	['block7a_project_bn[0][0]']
top_bn (BatchNormalization)	(None, 7, 7, 1200)	5120	['top_conv[0][0]']
top_activation (Activation)	(None, 7, 7, 1200)	0	['top_bn[0][0]']
avg_pool (GlobalAveragePooling	(None, 1200)	0	['top_activation[0][0]']
top_dropout (Dropout)	(None, 1200)	0	['avg_pool[0][0]']
predictions (dense)	(None, 1000)	1201000	['top_dropout[0][0]']

Total params: 5,330,571			
Trainable params: 5,228,548			
Non-trainable params: 42,023			

Gambar 4.11. EfficientNetB0 sebelum penyesuaian

Proses penyesuaian yang terakhir dilakukan pada arsitektur EfficientNetB0, model sebelum dilakukan fine tuning ditampilkan pada Gambar 4.11. Dapat dilihat bahwa setelah layer GAP terdapat layer top dropout dan layer prediksi dengan 1000 neuron yang melambangkan jumlah kelas prediksi. Arsitektur ini memiliki berjumlah parameter 5.330.5761 dengan trainable parameter 5.228.548 dan nontrainable 42.023. Jumlah layer dalam arsitektur ini adalah 241 layer.

top_conv (Conv2D)	(None, 7, 7, 1280)	489600	['block7a_project_bn[0][0]']
top_bn (BatchNormalization)	(None, 7, 7, 1280)	5120	['top_conv[0][0]']
top_activation (Activation)	(None, 7, 7, 1280)	0	['top_bn[0][0]']
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0	['top_activation[0][0]']
dropout (Dropout)	(None, 1280)	0	['global_average_pooling2d[0][0]']
dense (Dense)	(None, 60)	76800	['dropout[0][0]']
dropout_1 (Dropout)	(None, 60)	0	['dense[0][0]']
dense_1 (Dense)	(None, 4)	264	['dropout_1[0][0]']

Total params: 4,126,675			
Trainable params: 4,084,652			
Non-trainable params: 42,023			

Gambar 4.12. EfficientNetB0 setelah penyesuaian

Pada Gambar 4.12 dapat dilihat bahwa setelah dilakukan proses penyesuaian, layer klasifikasi telah disesuaikan yang sebelumnya 1000 kelas setelah penyesuaian menjadi 4 kelas yang sesuai dengan dataset yang digunakan. Setelah penyesuaian juga ditambahkan 1 layer dense tersembunyi dan 2 layer dropout. Parameter pada arsitektur ini telah disesuaikan juga dengan total setelah penyesuaian berjumlah 4.126.675 dengan trainable parameter 4.084.652 dan nontrainable 42.023. Jumlah keseluruhan layer telah bertambah menjadi 243 layer.

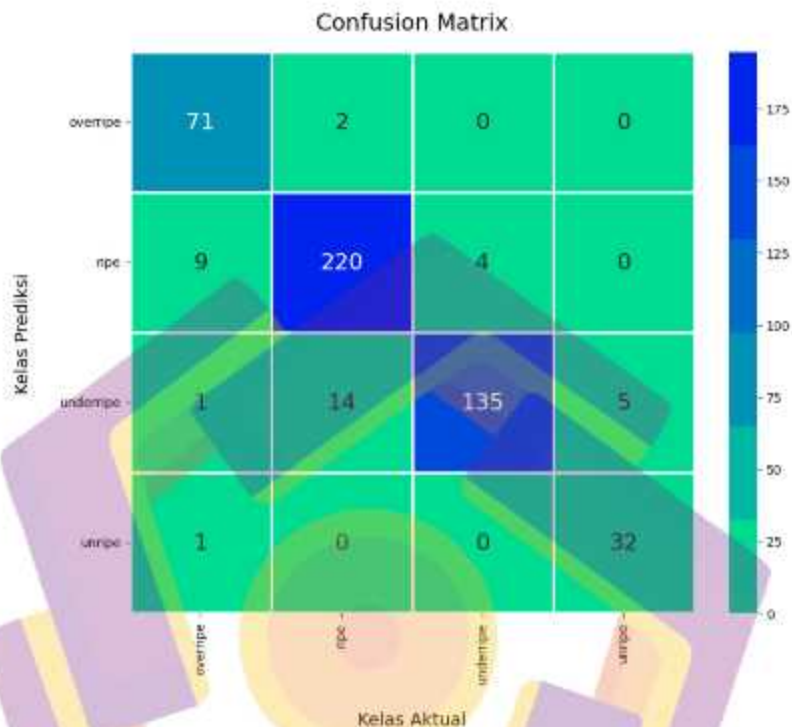
4.4. Hasil dan Evaluasi Model Klasifikasi

Setelah dilakukan preprocessing terhadap data dan proses pelatihan pada kedua model untuk mendapatkan model terbaik. Diperoleh model berdasarkan dari semua skenario yang diujikan dan merupakan penggambaran pengetahuan dari pembelajaran citra TBS kelapa sawit. Pada tahap pelatihan beberapa parameter digunakan seperti epoch sebesar 100, menggunakan batch size 16, menetapkan

angka learning rate pada angka 0.0001, dan alokasi data yang digunakan adalah 80/10/10 yang dibagi menjadi data training, validasi dan testing.

Pada kondisi ini arsitektur CNN, ResNet50 dan EfficientNetB0 telah berhasil melakukan identifikasi terhadap tingkat kematangan tandan buah segar kelapa sawit, proses ini terjadi pada layer konvolusi masing-masing model. Setelah ekstraksi fitur terjadi pada layer konvolusi, dilanjutkan dengan ekstraksi ciri lebih lanjut pada layer pooling dengan melakukan pengecilan dimensi hal ini agar fitur yang ada semakin terlihat lebih jelas. Setelah semua proses tadi, masuk pada tahap pengenalan serta identifikasi kelas citra tandan buah segar kelapa sawit pada layer fully connected. Bagian terakhir dari proses ini adalah pengujian model menggunakan data uji, hasil pengujian dengan menggunakan confusion matrix dapat dilihat pada Gambar 4.13.





Gambar 4.13. Pengujian confusion matrix

Pada penelitian, data yang digunakan adalah tandan buah segar kelapa sawit yang terdiri dari empat kelas kematangan. Data ini diperoleh dari roboflow.com, penyedia data publik untuk computer vision. Selanjutnya, data ini akan digunakan untuk proses pelatihan dan pengujian model klasifikasi tingkat kematangan tandan buah kelapa sawit.

Untuk penentuan terhadap nilai confusion matrix terdapat True positif (TP), True Negatif (TN), False Positif (FP) dan True Negatif (TN). Namun perhitungan confusion matrix pada multikelas berbeda, dimana berbentuk $n \times n$, dimana n adalah

jumlah kelas yang berbeda. Misalkan pada confusion matrix Gambar 4.9, pada kelas overripe. $TP(\text{overripe})$ merupakan prediksi benar untuk kelas overripe, $FP(\text{overripe})$ jumlah prediksi salah untuk kelas overripe, $FN(\text{overripe})$ merupakan jumlah prediksi dimana kelas overripe terprediksi sebagai kelas lain dan yang terakhir $TN(\text{overripe})$ jumlah prediksi yang benar pada kelas lain yang terprediksi sebagai kelas overripe. Berikut merupakan perhitungan confusion matrix pada Gambar 4.9.

1. Akurasi

Untuk pencarian akurasi menggunakan rumus berikut, total nilai TP dibagi dengan jumlah data pengujian. Jumlah nilai TP adalah $71 + 220 + 135 + 32 = 458$ dan jumlah data pengujian 494. Nilai akurasi yang dihasilkan adalah $458/494 = 0,9271$ atau 92,71%.

2. Precision

Untuk pencarian akurasi menggunakan rumus berikut, $TP/(TP + FP)$. Pada kasus multi class untuk mendapatkan nilai precision dilakukan perhitungan precision setiap kelas kemudian dibagi dengan jumlah kelas. Dan nilai precisionnya adalah

a. Overripe

$$P(\text{overripe}) = 71/(71+2+0+0)$$

$$P(\text{overripe}) = 71/73$$

$$P(\text{overripe}) = 0,9726 \text{ atau } 97,26\%$$

b. Ripe

$$P(\text{ripe}) = 220/(220+9+4+0)$$

$$P(\text{ripe}) = 220/233$$

$$P(\text{ripe}) = 0,9442 \text{ atau } 94,42\%$$

c. Underripe

$$P(\text{underripe}) = 135/(135+1+14+5)$$

$$P(\text{underripe}) = 135/155$$

$$P(\text{underripe}) = 0,8710 \text{ atau } 87,10\%$$

d. Unripe

$$P(\text{underripe}) = 32/(32+1+0+0)$$

$$P(\text{underripe}) = 32/33$$

$$e. P(\text{underripe}) = 0,9697 \text{ atau } ,96,97\%$$

Dan nilai precision yang dihasilkan adalah $(0,9726 + 0,9442 + 0,8710 + 0,9697)/4 = 0,9394$ atau 93,94%.

3. Recall

Untuk pencarian akurasi menggunakan rumus berikut, $TP/(TP + FN)$. Dalam menghitung recall juga akan menghitung recall pada masing-masing kelas, sebagai berikut.

a. Overripe

$$\text{recall}(\text{overripe}) = 71/(71+9+1+1)$$

$$\text{recall}(\text{overripe}) = 71/82$$

$$\text{recall}(\text{overripe}) = 0,8659 \text{ atau } 86,59\%$$

b. Ripe

$$\text{recall}(\text{overripe}) = 220/(220+2+14+0)$$

$$\text{recall}(\text{overripe}) = 220/236$$

$\text{recall}(\text{overripe}) = 0,9322$ atau 93,22%

c. Underripe

$\text{recall}(\text{underripe}) = 135/(135+0+4+0)$

$\text{recall}(\text{underripe}) = 135/139$

$\text{recall}(\text{underripe}) = 0,9712$ atau 97,12%

d. Unripe

$\text{recall}(\text{unripe}) = 32/(32+0+0+5)$

$\text{recall}(\text{unripe}) = 32/37$

$\text{recall}(\text{unripe}) = 0,8649$ atau 86,49%

Dan nilai Recall yang dihasilkan adalah $(0,8659 + 0,9322 + 0,9712 + 0,8649) / 4 = 0,9085$ atau 90,85%.

4. F1-score

Untuk pencarian akurasi menggunakan rumus berikut, $\text{F1-score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$. Perhitungan untuk F1-Score sebagai berikut.

$\text{F1-score} = 2 * (0,9394 * 0,9085) / (0,9394 + 0,9085)$

$\text{F1-score} = 1,7068898 / 1,8479$

$\text{F1-score} = 0,9237$ atau 92,37%.

Berdasarkan hasil perhitungan sebelumnya didapatkan bahwasanya confusion matrix pada Gambar 4.9, menghasilkan akurasi 92,71%, precision 93,94%, recall 90,85 dan F1-Score 92,37. Sedangkan untuk hasil model kalsifikasi ResNet50 dan EfficientNetB0 dijabarkan lebih lanjut pada sub bab 4.5.1 dan 4.5.2.

4.4.1. Hasil Model Klasifikasi ResNet50

Pada Tabel 4.4 merupakan hasil percobaan dari semua skenario yang diujikan pada model arsitektur ResNet50. Pada bagian ini, akan dilakukan 3 tahapan dalam percobaan skenario, tahapan pertama adalah mencari hyperparameter terbaik pada model klasifikasi ResNet50. Pada tahapan pertama percobaan didapatkan bahwa optimizer dengan akurasi terbaik adalah optimizer SGD, learning rate 0.00075 dengan akurasi terbaik dibandingkan learning rate lainnya pada percobaan ini, pada batch size didapati akurasi terbaik pada batch size 8. Untuk percobaan epoch dan dropout didapati bahwa epoch 100 dan dropout 0.5 dengan akurasi tertinggi dibandingkan yang lainnya, sehingga didapatkan hyperparameter pada ResNet 50 adalah optimizer SGD, Learning rate 0.00075, batch size 8, epoch 100 dan dropout 0.5. Tahapan kedua adalah melakukan percobaan terhadap hyperparameter yang telah didapatkan pada alokasi data 70/15/15, 80/10/10 dan 90/5/5, setelah melakukan percobaan didapati alokasi data 90/5/5 dengan hasil akurasi terbaik yaitu 96,36%. Dan yang terakhir adalah melihat pengaruh penyeimbangan data menggunakan smote pada dataset tandan buah segar kelapa sawit, setelah melakukan percobaan penyeimbangan data didapati akurasi yang dihasilkan lebih dari pada data yang tidak seimbang yaitu 98,29%. Dari ketiga tahapan percobaan ini, kondisi terbaik dari model ResNet50 adalah pada data alokasi 90/5/5, optimizer SGD, Learning rate 0.00075, batch size 8, epoch 100 dan dropout 0.5 serta pada sebaran data antar kelas yang seimbang hal ini dikarenakan

kenaikan hasil akurasi yang cukup signifikan, hal ini memberikan gambaran pengaruh penyeimbangan data menggunakan smote.

Tabel 4.4. Hasil pengujian ResNet50

ResNet50					
Skenario 1 Optimizer					
<i>Optimizer</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Adam	92,71%	0.3865	93,94%	90,85%	92,87%
SGD	95,34%	0.1640	96,97%	94,36%	95,65%
Skenario 2 Learning Rate					
<i>Learning rate</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
0.0001	95,34%	0.1640	96,97%	94,36%	95,65%
0.00035	95,14%	0.2619	96,85%	94,52%	95,67%
0.0005	95,95%	0.1596	96,05%	95,33%	95,69%
0.00075	96,36%	0.2287	97,15%	96,29%	96,72%
0.0009	94,94%	0.3799	93,74%	96,00%	94,86%
Skenario 3 Batch Size					
<i>Batch Size</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
8	95,75%	0.2238	96,92%	93,65%	95,26%
16	95,34%	0.1640	96,97%	94,36%	95,65%
32	91,70%	0.4036	92,44%	89,32%	90,85%
Skenario 4 Epoch					
<i>Epoch</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	93,32%	0.2787	94,33%	93,25%	93,78%
100	95,34%	0.1640	96,97%	94,36%	95,65%
150	93,52%	0.2400	94,26%	92,53%	93,38%
Skenario 5 Dropout					
<i>Dropout</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
0.5	95,34%	0.1640	96,97%	94,36%	95,65%
0.3	94,53%	0.2484	95,32%	93,76%	94,53%
Skenario 6 Alokasi Data					
<i>Alokasi data</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
70/15/15	94,33%	0.3253	96,06%	93,44%	94,74%
80/10/10	95,34%	0.3095	96,76%	94,54%	95,64%
90/5/5	96,36%	0.2000	95,93%	95,28%	95,60%
Skenario 7 Penyeimbangan Data					
<i>Penyeimbangan Data</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Smote	98,29%	0.1628	98,29%	98,31%	98,30%

4.4.2. Pengaruh Optimizer ResNet50

Dalam penelitian ini dilakukan perbandingan pada optimizer yang akan digunakan pada model arsitektur ResNet50, perbandingan ini bertujuan untuk mendapatkan optimizer dengan hasil paling optimal. Hasil perbandingan optimizer dapat dilihat pada Tabel 4.5.

Tabel 4.5. Perbandingan hasil pengujian optimizer ResNet50

Optimizer	Training	Validasi	Testing	Loss
SGD	99,90	94,74	95,34	0.1640
Adam	99,97	92,51	92,71	0.3865

Pada penelitian, data yang digunakan adalah tandan buah segar kelapa sawit yang terdiri dari empat kelas kematangan. Data ini diperoleh dari roboflow.com, penyedia data publik untuk computer vision. Selanjutnya, data ini akan digunakan untuk proses pelatihan dan pengujian model klasifikasi tingkat kematangan tandan buah kelapa sawit.

4.4.3. Pengaruh Learning Rate ResNet50

Pengaruh nilai learning dalam penelitian ini dicari dengan membandingkan nilai learning rate 0.0001, 0.00035, 0.0005, 0.00075 dan 0.0009. Untuk melakukan pencarian nilai learning rate yang paling optimal menggunakan parameter yang sebelumnya sudah dijelaskan pada sub bab 4.4 namun pada model arsitektur ResNet50 akan menggunakan SGD dalam proses pengujian ini. Untuk hasil terkait pengujian nilai learning rate dapat dilihat pada Tabel 4.6.

Tabel 4.6. Perbandingan hasil pengujian Learning Rate ResNet50

Learning Rate	Training	Validasi	Testing	Loss
0.0001	99,90	94,74	95,34	0.1640
0.00035	99,97	96,96	95,14	0.1462
0.0005	99,97	95,34	95,95	0.1562
0.00075	100	95,14	96,36	0.1580
0.0009	100	93,12	94,94	0.2949

Berdasarkan Tabel 4.6 nilai learning rate 0.0001, 0.00035 dan 0.0005 cenderung memiliki nilai yang hampir sama pada akurasi yang dihasilkan. Berbeda hal yang dengan nilai learning rate 0.00075, pada nilai learning rate ini mengalami kenaikan akurasi dibandingkan nilai learning rate yang lain meskipun nilai loss yang dihasilkan bukan yang terendah. Learning rate dengan akurasi terendah adalah 0.0009, meskipun mengalami kenaikan akurasi pada training namun pada validasi dan testing mengalami penurunan yang cukup banyak serta menjadi yang terendah diantara yang lain, loss yang dihasilkan pun menjadi yang paling tinggi. Dimana ini menandakan terjadinya overfitting, pada pengujian ini dapat dilihat bahwa penggunaan nilai learning rate yang besar menghasilkan akurasi yang rendah, namun pada learning rate yang terlalu kecil juga menghasilkan akurasi yang rendah. Hal ini terjadi karena nilai yang lebih tinggi berdampak pada proses pembelajaran model yang lebih cepat, tetapi tingkat ketelitian akan berkurang. Sebaliknya, nilai yang lebih rendah berdampak pada tingkat ketelitian neural network yang lebih tinggi, tetapi proses pembelajaran akan memerlukan lebih banyak waktu. Sehingga pada pengujian ini learning rate dengan nilai paling optimal adalah 0.00075.

4.4.4. Pengaruh Batch Size ResNet50

Nilai batch size 8, 16 dan 32 yang akan digunakan pada pengujian untuk melihat pengaruh terhadap nilai batch size pada penelitian ini, pengujian ini menggunakan parameter dasar yang sama pada pengujian sebelumnya. Hasil terkait pengujian dapat dilihat pada Tabel 4.7.

Tabel 4.7. Perbandingan hasil pengujian Batch Size ResNet50

Batch Size	Training	Validasi	Testing	Loss
8	99,90	94,53	95,75	0.2238
16	99,90	94,74	95,34	0.1640
32	99,92	91,70	91,70	0.4036

Dapat dilihat berdasarkan Tabel 4.7 semakin tinggi nilai batch size yang digunakan akurasi yang dihasilkan terus mengalami penurunan. Sehingga batch size terkecil yang paling optimal pada pengujian ini dengan menghasilkan akurasi yang lebih baik dari pada nilai batch size yang lainnya, meskipun loss bukan yang terendah tapi masih lebih baik dari loss yang dihasilkan oleh nilai batch size 32.

4.4.5. Pengaruh Epoch ResNet50

Penggunaan jumlah epoch yang digunakan pada proses pelatihan juga akan dilihat dalam penelitian ini. Untuk itu, pengujian ini akan menggunakan tiga jumlah epoch dengan nilai terendah 50 dan tertinggi 150. Hasil dari pengujian jumlah epoch data dilihat pada Tabel 4.8.

Tabel 4.8. Perbandingan hasil pengujian Epoch ResNet50

Epoch	Training	Validasi	Testing	Loss
50	99,62	93,32	93,32	0.2787
100	99,90	94,74	95,34	0.1640
150	99,97	93,52	93,52	0.2490

Tabel 4.8 dapat dilihat bahwa kenaikan jumlah epoch yang digunakan diikuti juga oleh kenaikan akurasi, namun pada jumlah epoch tertinggi mengalami overfitting hal ini dapat dilihat berdasarkan akurasi validasi dan testing yang mengalami penurunan angka yang signifikan dan loss yang mengakibatkan lebih besar. Sehingga epoch dengan jumlah 100 menjadi yang paling optimal pada pengujian ini.

4.4.6. Pengaruh Dropout ResNet50

Untuk mencoba mengurangi overfitting yang terjadi, teknik regulasi diterapkan pada penelitian ini. Teknik regulasi yang akan digunakan pada penelitian ini adalah dropout, dengan membandingkan dua nilai dropout 0.3 dan 0.5. Untuk hasil tersebut dapat dilihat pada Tabel 4.9.

Tabel 4.9. Perbandingan hasil pengujian Dropout ResNet50

Dropout	Training	Validasi	Akurasi	Loss
0.5	99,90	94,74	95,34	0.1640
0.3	100	93,32	94,53	0.2484

Berdasarkan Tabel 4.9 dapat dilihat dropout dengan nilai dropout 0.5 dengan hasil yang lebih baik dari pada dropout 0.3, sehingga dropout 0.5 akan digunakan pada mode dalam penelitian ini. Sehingga dengan penggunaan dropout 0.5, Selama setiap iterasi pelatihan, setiap unit atau node di lapisan tersembunyi, juga dikenal sebagai lapisan tersembunyi, memiliki kesempatan 50% untuk dinonaktifkan secara acak dan 50% lain 50 unit akan tetap aktif dan digunakan untuk terus melakukan pelatihan.

4.4.7. Pengaruh Alokasi Data ResNet50

Setelah dilakukan pencarian hyperparameter dengan tujuan untuk mendapatkan hyperparameter yang optimal, dilakukan tahap pengujian pada alokasi kata 70/15/15, 80/10/10 dan 90/5/5 dengan optimizer SGD, learning rate 0.0075, batch size 8, epoch 100 dan pada dropout 0.5 yang telah didapatkan dari proses sebelumnya. Hasil pengujian pada alokasi data dapat dilihat pada Tabel 4.10.

Tabel 4.10. Perbandingan hasil pengujian Alokasi Data ResNet50

Alokasi Data	Training	Validasi	Testing	Loss
70/15/15	100	93,79	94,33	0.3253
80/10/10	100	95,34	95,34	0.3095
90/5/5	100	94,33	96,36	0.2000

Berdasarkan Tabel 4.10 dapat dilihat bahwa semakin bertambahnya jumlah data pelatihan nilai akurasi yang dihasilkan semakin baik dan nilai loss juga semakin mengalami penurunan. Penggunaan data pelatihan yang lebih kecil dapat menyebabkan overfitting dan bisa menyebabkan kurang representasi terhadap kelas minoritas. Sehingga alokasi 90/5/5 memberikan performa terbaik dibandingkan alokasi data lainnya pada penelitian ini dengan akurasi testing tertinggi yaitu 96,36 dan loss terendah untuk klasifikasi tingkat kematangan tandan buah segar pada data yang belum diseimbangkan.

4.4.8. Hasil Model Klasifikasi EfficientNetB0

Pada percobaan menggunakan model klasifikasi EfficientNetB0 memiliki tahapan percobaan yang sama pada model ResNet50 sebelumnya yaitu tahapan percobaan pencarian hyperparameter, tahapan percobaan alokasi data dan melihat pengaruh penyeimbangan data. Hasil pengujian ketiga tahapan percobaan yang

telah dilakukan terdapat pada Tabel 4.5, pada tabel tersebut didapatkan bahwa kondisi terbaik dari keseluruhan percobaan model EfficientNetB0 adalah pada optimizer Adam, epoch 100, learning rate 0.0001, batch size yang digunakan 32, dropout 0.5 pada alokasi data 90/5/5 dengan data yang seimbang dengan akurasi terbaik 98,93%.

Tabel 4.11. Hasil pengujian EfficientNetB0

EfficientNetB0					
Skenario 1 Optimizer					
Optimizer	Accuracy	Loss	Precision	Recall	F1-Score
Adam	94,33%	0.1939	93,89%	92,57%	93,23%
SGD	97,57%	0.1038	97,07%	97,46%	97,26%
Skenario 2 Learning Rate					
Learning rate	Accuracy	Loss	Precision	Recall	F1-Score
0.0001	97,57%	0.1038	93,89%	92,57%	93,23%
0.00035	96,96%	0.2619	97,25%	94,59%	95,90%
0.0005	96,96%	0.1596	97,54%	96,04%	96,79%
0.00075	96,15%	0.2287	96,35%	95,17%	95,76%
0.0009	93,12%	0.3799	94,04%	90,65%	92,32%
Skenario 3 Batch Size					
Batch Size	Accuracy	Loss	Precision	Recall	F1-Score
8	96,36%	0.2210	97,80%	95,22%	96,49%
16	96,56%	0.1953	93,89%	92,57%	93,23%
32	97,98%	0.1070	97,83%	97,83%	97,83%
Skenario 4 Epoch					
Epoch	Accuracy	Loss	Precision	Recall	F1-Score
50	95,95	0.2512	96,28%	95,19%	95,73%
100	97,57	0.1038	93,89%	92,57%	93,23%
150	96,76	0.1305	96,06%	95,60%	95,83%
Skenario 5 Dropout					
Dropout	Accuracy	Loss	Precision	Recall	F1-Score
0.5	97,57%	0.1038	93,89%	92,57%	93,23%
0.3	97,17%	0.2285	96,28%	97,08%	96,68%
Skenario 6 Alokasi Data					
Alokasi data	Accuracy	Loss	Precision	Recall	F1-Score
70/15/15	95,68%	0.2788	97,02%	94,04%	95,51%
80/10/10	95,55%	0.3097	95,35%	95,04%	95,19%
90/5/5	98,79%	0.0697	99,15%	97,99%	98,56%
Skenario 7 Penyeimbangan Data					
Penyeimbangan Data	Accuracy	Loss	Precision	Recall	F1-Score
Smote	98,93%	0.1094	98,93%	98,93%	98,93%

4.4.9. Pengaruh Optimizer EfficientNetB0

Dalam penelitian ini dilakukan perbandingan pada optimizer yang akan digunakan pada model arsitektur EfficientNetB0, perbandingan ini bertujuan untuk mendapatkan optimizer dengan hasil paling optimal. Parameter dasar yang digunakan untuk pengujian ini menggunakan parameter yang telah dijelaskan pada sub bab 4.4. Hasil perbandingan optimizer dapat dilihat pada Tabel 4.12.

Tabel 4.12. Perbandingan hasil pengujian Optimizer EfficientNetB0

Optimizer	Training	Validasi	Testing	Loss
SGD	96,00	94,74	94,33	0.1939
Adam	99,90	97,57	97,57	0.1038

Pada Tabel 4.12 menyajikan hasil pengujian dari dua optimizer yaitu SGD dan Adam, dari tabel tersebut didapati bahwa optimizer terbaik adalah optimizer Adam. Hal ini dikarenakan akurasi yang dihasilkan oleh optimizer Adam lebih dan stabil serta angka loss yang dihasilkan lebih rendah dari optimizer SDG, dengan akurasi yang dihasilkan saat testing 97,57. Hal ini menunjukkan bahwa model yang dilatih menggunakan optimizer Adam cenderung melakukan tugas klasifikasi dengan lebih baik daripada SGD.

4.4.10. Pengaruh Learning Rate EfficientNetB0

Pengaruh nilai learning dalam penelitian ini dicari dengan membandingkan nilai learning rate 0.0001, 0.00035, 0.0005, 0.00075 dan 0.0009. Untuk melakukan pencarian nilai learning rate yang paling optimal menggunakan parameter yang sebelumnya sudah dijelaskan pada sub bab 4.4 namun pada model arsitektur

EfficientNetB0 akan menggunakan Adam dalam proses pengujian ini. Untuk hasil terkait pengujian nilai learning rate dapat dilihat pada Tabel 4.13.

Tabel 4.13. Perbandingan hasil pengujian Learning Rate EfficientNetB0

Learning Rate	Training	Validasi	Testing	Loss
0.0001	99,90	97,57	97,57	0.1038
0.00035	99,97	95,14	96,96	0.2619
0.0005	99,95	97,57	96,96	0.1596
0.00075	99,90	97,98	96,15	0.2287
0.0009	99,90	94,13	93,12	0.3799

Pada Tabel 4.13 dapat dilihat bahwa semakin besar nilai dari learning rate akurasi yang dihasilkan pada testing terus mengalami penurunan dan peningkatan nilai pada loss. Berdasarkan data tersebut, kombinasi learning rate 0.0001 dan 0.0005 menunjukkan yang lebih tinggi pada data validasi dan pengujian serta loss yang lebih rendah dibandingkan dengan nilai learning rate lainnya. Sebaliknya, kombinasi learning rate 0.0009 menunjukkan kinerja yang lebih buruk dengan akurasi yang lebih rendah pada data validasi dan pengujian serta loss tertinggi yang menandakan terjadinya overfitting. Sehingga learning rate 0.0001 menjadi learning rate yang paling optimal, dengan akurasi testing 97,57 dan loss yang rendah, hasil ini menjadi yang terbaik dari nilai learning rate lainnya.

4.4.11. Pengaruh Batch Size EfficientNetB0

Nilai batch size 8, 16 dan 32 yang akan digunakan pada pengujian untuk melihat pengaruh terhadap nilai batch size pada penelitian ini, pengujian ini menggunakan parameter dasar yang sama pada pengujian sebelumnya. Hasil terkait pengujian dapat dilihat pada Tabel 4.14.

Tabel 4.14. Perbandingan hasil pengujian Batch Size EfficientNetB0

Bacth Size	Training	Validasi	Testing	Loss
8	99,90	95,55	96,36	0.2210
16	99,90	97,57	97,57	0.1953
32	99,90	96,76	97,98	0.1070

Jumlah batch 16 dan 32 memberikan akurasi dan loss yang lebih rendah daripada ukuran batch 8. namun, ukuran batch yang lebih besar cenderung membutuhkan lebih banyak ruang memori dan komputasi, seperti yang ditunjukkan pada Tabel 4.14. dapat dilihat juga semakin tinggi jumlah batch size akurasi yang dihasilkan juga terus mengalami peningkatan. Ukuran batch, mengacu pada jumlah sampel data yang digunakan dalam satu iterasi pelatihan sebelum parameter model diperbarui. Berdasarkan performa dari ketiga batch size tersebut, batch dengan kondisi terbaik adalah batch size 32 dengan akurasi yang lebih baik dan loss yang paling rendah pada pengujian ini.

4.4.12. Pengaruh Epoch EfficientNetB0

Penggunaan jumlah epoch yang digunakan pada proses pelatihan juga akan dilihat dalam penelitian ini. Untuk itu, pengujian ini akan menggunakan tiga jumlah epoch dengan nilai terendah 50 dan tertinggi 150. Hasil dari pengujian jumlah epoch data dilihat pada Tabel 4.15.

Tabel 4.15. Perbandingan hasil pengujian Epoch EfficientNetB0

Epoch	Training	Validasi	Testing	Loss
50	99,77	95,55	95,95	0.2512
100	99,90	97,57	97,57	0.1038
150	99,97	97,37	96,76	0.1305

Pada Tabel 4.15 dapat dilihat bahwa tingkat akurasi pelatihan dan validasi cenderung meningkat dengan peningkatan jumlah epoch. Namun, ada sedikit penurunan akurasi pada data pengujian setelah 100 epoch, menunjukkan bahwa model memiliki tingkat akurasi yang tinggi dan kehilangan yang rendah pada jumlah epoch 100, dan peningkatan jumlah epoch lebih lanjut tidak menghasilkan peningkatan yang signifikan dalam performa model. Sehingga epoch 100 menjadi epoch dengan hasil terbaik.

4.4.13. Pengaruh Epoch EfficientNetB0

Pada penelitian, data yang digunakan adalah tandan buah segar kelapa sawit yang terdiri dari empat kelas kematangan. Data ini diperoleh dari roboflow.com, penyedia data publik untuk computer vision. Selanjutnya, data ini akan digunakan untuk proses pelatihan dan pengujian model klasifikasi tingkat kematangan tandan buah kelapa sawit.

Tabel 4.15. Perbandingan hasil pengujian Epoch EfficientNetB0

Epoch	Training	Validasi	Testing	Loss
50	99,77	95,55	95,95	0.2512
100	99,90	97,57	97,57	0.1038
150	99,97	97,37	96,76	0.1305

Pada Tabel 4.15 dapat dilihat bahwa tingkat akurasi pelatihan dan validasi cenderung meningkat dengan peningkatan jumlah epoch. Namun, ada sedikit penurunan akurasi pada data pengujian setelah 100 epoch, menunjukkan bahwa model memiliki tingkat akurasi yang tinggi dan kehilangan yang rendah pada jumlah epoch 100, dan peningkatan jumlah epoch lebih lanjut tidak menghasilkan

peningkatan yang signifikan dalam performa model. Sehingga epoch 100 menjadi epoch dengan hasil terbaik.

4.4.14. Pengaruh Dropout EfficientNetB0

Untuk mencoba mengurangi overfitting yang terjadi, teknik regulasi diterapkan pada penelitian ini. Teknik regulasi yang akan digunakan pada penelitian ini adalah dropout, dengan membandingkan dua nilai dropout 0.3 dan 0.5. Untuk hasil tersebut dapat dilihat pada Tabel 4.16.

Tabel 4.16. Perbandingan hasil pengujian Dropout EfficientNetB0

Dropout	Training	Validasi	Testing	Loss
0.5	99,90	97,57	97,57	0.1038
0.3	99,97	96,76	97,17	0.2285

Pada Tabel 4.16, dapat dilihat bahwa dropout dengan nilai 0.5 menunjukkan akurasi yang tinggi dan kehilangan yang rendah pada data validasi dan pengujian. Namun, Dropout dengan nilai 0.3 menunjukkan akurasi dan kehilangan yang sedikit lebih rendah daripada hilangnya nilai 0.5. Dalam pengujian nilai dropout terbaik, yang dapat menghasilkan hasil terbaik adalah dropout dengan nilai 0,5.

4.4.15. Pengaruh Alokasi Data EfficientNetB0

Setelah dilakukan pencarian hyperparameter dengan tujuan untuk mendapatkan hyperparameter yang optimal, dilakukan tahap pengujian pada alokasi kata 70/15/15, 80/10/10 dan 90/5/5 dengan optimizer Adam, learning rate 0.001, batch size 32, epoch 100 dan pada dropout 0.5 yang telah didapatkan dari proses sebelumnya. Hasil pengujian pada alokasi data dapat dilihat pada Tabel 4.17.

Tabel 4.17. Perbandingan hasil pengujian Alokasi Data EfficientNetB0

Alokasi Data	Training	Validasi	Testing	Loss
70/15/15	100	95,82	95,68	0.2788
80/10/10	99,90	97,17	95,55	0.3097
90/5/5	99,98	98,38	98,79	0.0697

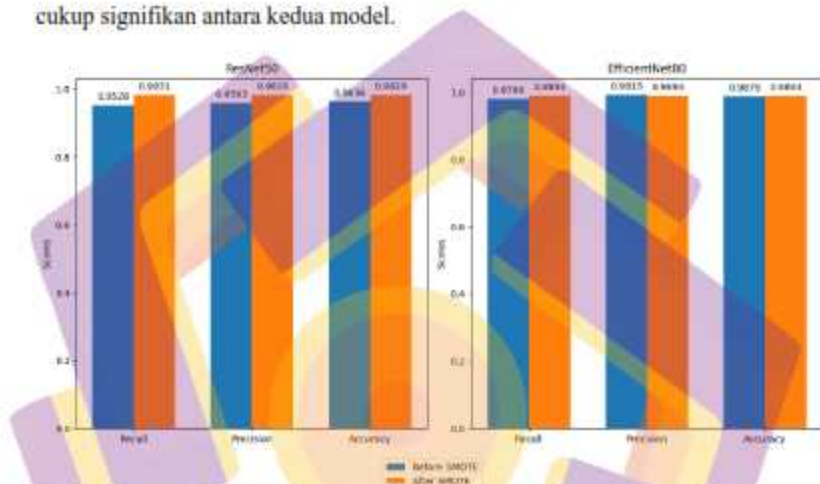
Pada Tabel 4.17 ditunjukkan dari hasil pengujian tersebut bahwa alokasi data 90/5/5 memiliki akurasi yang paling tinggi dan kehilangan yang paling rendah pada data validasi dan pengujian dan dengan ini menjadi kondisi terbaik klasifikasi tingkat kematangan tandan buah segar kelapa sawit pada model arsitektur EfficientNetB0 pada data yang belum diseimbangkan. Ini menunjukkan bahwa alokasi pelatihan yang lebih besar dapat membantu model belajar dengan lebih baik dan melakukan lebih baik pada data yang tidak terlihat sebelumnya.

4.5. Pengaruh Penyeimbangan Data

Penelitian ini tidak hanya menentukan kondisi terbaik untuk masing-masing model, tetapi juga melakukan penyeimbangan data pada kelas data minoritas untuk menghindari bias dalam pengklasifikasian yang mungkin mengarah ke kelas data mayoritas. Untuk melihat dampak penyeimbangan data pada masing-masing model, kedua model akan diuji menggunakan kondisi terbaik yang telah ditentukan sebelumnya. Selanjutnya, teknik resampling digunakan untuk menguji penyeimbangan data: oversampling pada kelas minoritas dengan Teknik Resampling Minoritas Sintetis (SMOTE) dan undersampling pada kelas mayoritas.

4.5.1. Hasil Synthetic Minority Oversampling Technique (SMOTE)

Gambar 4.9 menunjukkan representasi hasil pengujian setelah menggunakan SMOTE. Seperti yang ditunjukkan pada Gambar 4.14, percobaan penyeimbangan data meningkatkan kinerja kedua model. Namun, peningkatan ini cukup signifikan antara kedua model.



Gambar 4.14. Perbandingan akurasi sebelum dan sesudah penerapan smote

Sebelum penyeimbangan data, atau data raw, akurasi model klasifikasi ResNet50 adalah sebesar 96.36%, dan naik sebesar 1.93% menjadi 98.29% setelah penyeimbangan data menggunakan metode SMOTE. Akurasi model klasifikasi EfficientNetB0, di sisi lain, hanya meningkat sebesar 98.79% sebelum dan setelah penyeimbangan data menggunakan metode SMOTE.

Selain meningkatkan akurasi, model klasifikasi ResNet50 juga meningkatkan nilai precision dan recall. Sebaliknya, nilai precision model klasifikasi EfficientNetB0 menurun, sementara nilai recall meningkat.

Akibatnya, setelah penerapan SMOTE, ResNet50 menunjukkan peningkatan yang lebih besar dalam hal akurasi dan beberapa metrik lainnya, sedangkan EfficientNetB0 tidak mengalami peningkatan yang sama.

Tabel 4.18. Perbandingan sebelum dan sesudah penyeimbangan

Resnet50	Raw	SMOTE	Gap(%)
Accuracy	96.36	98.29	1.93
Loss	0.2000	0.1628	0.0372
Precision	95.93	98.29	2.36
Recall	95.28	98.31	3.03
F1-score	95.60	98.30	2.70
EfficientNetB0	Raw	SMOTE	Gap(%)
Accuracy	98.79	98.93	0.14
Loss	0.0697	0.1094	-0.0397
Precision	99.15	98.93	-0.22
Recall	97.99	98.93	0.94
F1-score	98.56	98.93	0.37

Tabel 4.18 menunjukkan perbandingan hasil model sebelum dan setelah penyeimbangan data. Terlihat bahwa setelah penerapan metode penyeimbangan, model ResNet50 mengalami peningkatan akurasi sebesar 1.93%, ketepatan sebesar 2.36%, recall sebesar 3.03%, dan skor F1-nya sebesar 2.70%. Selain itu, nilai kerugian turun sebesar 0.0372.

Sebaliknya, setelah dilakukan penyeimbangan, model EfficientNetB0 mengalami penurunan pada metrik akurasi, dengan penurunan sebesar -0.22% dan peningkatan pada nilai kehilangan sebesar 0.0397. Namun, meskipun mengalami penurunan pada akurasi dan kehilangan, EfficientNetB0 tetap menghasilkan tingkat akurasi yang baik, dengan peningkatan akurasi sebesar 0.14%. Selain itu, nilai recall meningkat sebesar 0.94% dan skor F1 meningkat sebesar 0.37%.

Meskipun ResNet50 meningkat dalam semua faktor setelah penerapan metode penyeimbangan, akurasi yang dihasilkan oleh EfficientNetB0 masih lebih rendah daripada yang dihasilkan oleh ResNet50. Akurasi EfficientNetB0 mencapai 98,93% untuk data yang telah melalui SMOTE dan 98,79% untuk data asli (raw), sedangkan akurasi ResNet50 mencapai 98,29% untuk data yang telah melalui SMOTE dan 96,36% untuk data asli (raw).

Tabel 4.19. Perbandingan precision dan recall pada setiap kelas

RESNET50						
Kelas	Precision			Recall		
	Raw	Smote	Gap	Raw	Smote	Gap
Terlalu matang	97,22%	100%	2,78%	92,11%	98,32%	6,21%
Matang	96,58%	97,44%	0,86%	96,58%	100,00%	3,42%
Kurang Matang	96,15%	95,73%	-0,42%	98,68%	98,25%	-0,43%
Mentah	93,75%	100%	6,25%	93,75%	96,69%	2,94%
EfficientNetB0						
Kelas	Precision			Recall		
	Raw	Smote	Gap	Raw	Smote	Gap
Terlalu matang	100%	100%	0%	100,00%	99,15%	-0,85
Matang	99,15%	97,44%	-1,71%	99,15%	98,28%	-0,87
Kurang Matang	97,44%	98,29%	0,85%	98,70%	98,20%	-0,41
Mentah	100%	100%	0%	94,12%	100,00%	5,88

Untuk kelas mentah, matang, dan terlalu matang, metrik precision dan recall meningkat setelah menerapkan metode SMOTE pada model ResNet50. Namun, untuk kelas kurang matang, nilai precision turun sebesar 0,42% dan nilai recall juga turun sebesar 0,43%.

Selain itu, setelah SMOTE diterapkan pada model EfficientNetB0, metrik ketepatan dan recall berubah. Kelas terlalu matang dan mentah memiliki nilai

ketepatan 100%, sedangkan kelas kurang matang meningkatkan ketepatan sebesar 0,85%, sementara kelas matang kehilangan ketepatan sebesar -1.71%.

Selain itu, perlu dicatat bahwa pada model EfficientNetB0, nilai recall untuk kelas terlalu matang, matang, dan kurang matang turun, tetapi nilai recall untuk kelas mentah meningkat sebesar 5.88%.

Dengan temuan ini, dapat disimpulkan bahwa ResNet50 mengungguli EfficientNetB0 dalam hal ketepatan dan recall dibandingkan dengan EfficientNetB0. Ini terutama benar setelah penerapan metode SMOTE, karena kelas terlalu matang, matang, dan mentah menjadi lebih baik.

4.5.2. Hasil Random Under Sampling (RUS)

Seperti yang ditunjukkan dalam Tabel 4.20, hasil setelah penerapan metode penyeimbangan RUS menunjukkan bahwa model arsitektur ResNet50 memiliki akurasi tertinggi sebesar 88.06% pada data pengujian, sementara model arsitektur EfficientNetB0 memiliki akurasi sebesar 82.09% pada data pengujian. Dengan nilai akurasi dan kehilangan ini, ResNet50 dianggap sebagai model yang memiliki hasil terbaik dalam pengujian dengan metode penyeimbangan data menggunakan RUS.

Tabel 4.20. Hasil penerapan RUS

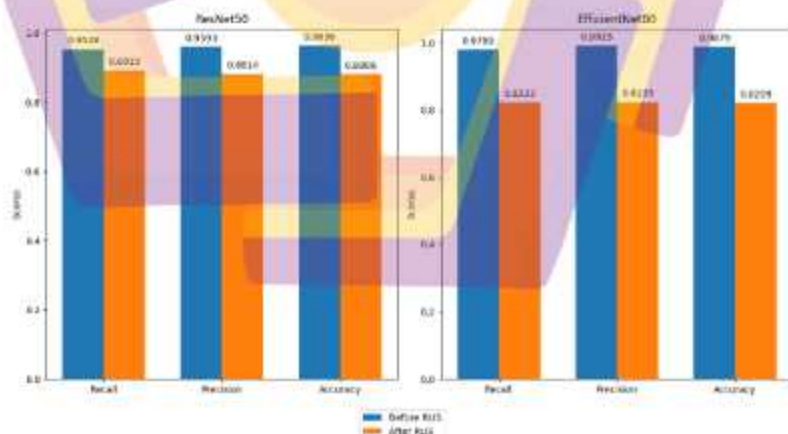
Metode Penyeimbangan	Training	Validasi	Testing	Loss
ResNet50				
RUS	100	89.39	88.06	0.7573
EfficientNetB0				
RUS	100	84.85	82.09	0.9800

Gambar 4.15 menunjukkan hasil dari penggunaan metode Random Under-Sampling (RUS) pada dataset untuk melakukan penyeimbangan. Setelah RUS

diterapkan, kedua arsitektur mengalami penurunan pada berbagai metrik pengukuran, seperti yang ditunjukkan pada grafik tersebut. Penurunan ini berdampak pada semua metrik yang dihasilkan oleh kedua model.

Penurunan akurasi setelah penerapan RUS pada model arsitektur EfficientNetB0 lebih besar daripada pada ResNet50, dengan penurunan sekitar 16.7% dibandingkan dengan 8.3% pada ResNet50. Ini menunjukkan bahwa penerapan RUS menghasilkan penurunan akurasi yang lebih besar, sekitar dua kali lipat dibandingkan dengan penurunan pada ResNet50.

Hasil ini menunjukkan bahwa penerapan RUS tidak menghasilkan peningkatan akurasi dan bahkan dapat menyebabkan overfitting dengan penurunan akurasi pada data validasi dan pengujian. Nilai kehilangan yang dihasilkan juga meningkat setelah RUS diterapkan, dengan nilai ResNet50 sekitar 0.7573 dan EfficientNetB0 sekitar 0.9800.



Gambar 4.15. Perbandingan akurasi sebelum dan sesudah penerapan RUS

Tidak ada peningkatan akurasi pada kedua model yang digunakan ketika Random Under-Sampling (RUS) digunakan. Ini disebabkan oleh kegagalan model untuk menemukan semua kasus positif yang sebenarnya dan mengidentifikasinya dengan benar. Lebih khusus lagi, dibandingkan dengan kondisi sebelum penerapan RUS, metrik precision, yang mengukur kemampuan model untuk mengidentifikasi kasus positif dengan benar, menurun. Ini terlihat pada model arsitektur ResNet50. Sebaliknya, kelompok minoritas, terutama kelompok "Kurang Matang", memiliki tingkat akurasi yang rendah setelah penerapan RUS. Meskipun penerapan RUS dapat membantu meningkatkan recall, hanya dua kelompok minoritas lainnya yang menunjukkan peningkatan.

Selain itu, metrik recall berubah. Pada kelas mayoritas, atau kelas "Matang", yang memiliki jumlah sampel yang lebih besar, penerapan RUS justru menghasilkan akurasi yang lebih rendah. Ini mungkin disebabkan oleh fakta bahwa RUS mengurangi jumlah sampel dalam kelas mayoritas, sehingga sulit untuk menemukan semua kasus positif dalam kategori tersebut.

Ini sebanding dengan temuan pada model EfficientNetB0. Setelah penerapan Random Under-Sampling (RUS), akurasi dalam mengidentifikasi kasus positif baik pada kelas mayoritas maupun minoritas (yaitu, kelas "Matang" dan "Kurang Matang") menurun dibandingkan dengan kondisi sebelum RUS digunakan. Hasil metrik recall juga menunjukkan pola yang serupa, dengan hanya kelas "Mentah" yang mengalami peningkatan akurasi dalam mengidentifikasi kasus. Tiga kelas lainnya, di sisi lain, mengalami penurunan akurasi yang cukup besar setelah penerapan RUS. Tabel 4.21 menampilkan detail hasil precision dan recall

untuk masing-masing kelas. Akibatnya, kesimpulan yang dapat diambil adalah bahwa, terutama dalam hal kelas mayoritas dan beberapa kelas minoritas, penerapan RUS tampaknya tidak secara konsisten meningkatkan performa model.

Tabel 4.21. Hasil precision dan recall masing-masing kelas RUS

RESNET50						
Kelas	Precision			Recall		
	Raw	RUS	Gap	Raw	RUS	Gap
Terlalu matang	97.22%	93.75%	-3.58%	92.11%	93.75%	1.78%
Matang	96.58%	76.47%	-20.81%	96.58%	86.67%	-10.24%
Kurang Matang	96.15%	94.12%	-2.11%	98.68%	76.19%	-22.75%
Mentah	93.75%	88.24%	-5.88%	93.75%	100%	6.67%
EfficientNetB0						
Kelas	Precision			Recall		
	Raw	RUS	Gap	Raw	RUS	Gap
Terlalu matang	100%	100%	2.78%	100.00%	76.19%	-17.31%
Matang	99.15%	52.94%	-45.07%	99.15%	81.82%	-15.27%
Kurang Matang	97.44%	76.47%	-20.44%	98.70%	76.47%	-22.49%
Mentah	100%	100%	6.67%	94.12%	94.44%	0.74%

Dalam penerapan teknik penyeimbangan data menggunakan SMOTE (Teknik Penyelidikan Minoritas Sintetis) dan RUS (Teknik Penyelidikan Acak) pada dua model arsitektur CNN dengan transfer learning, ResNet50 dan EfficientNetB0, terlihat dampak yang berbeda pada kinerja kedua model tersebut.

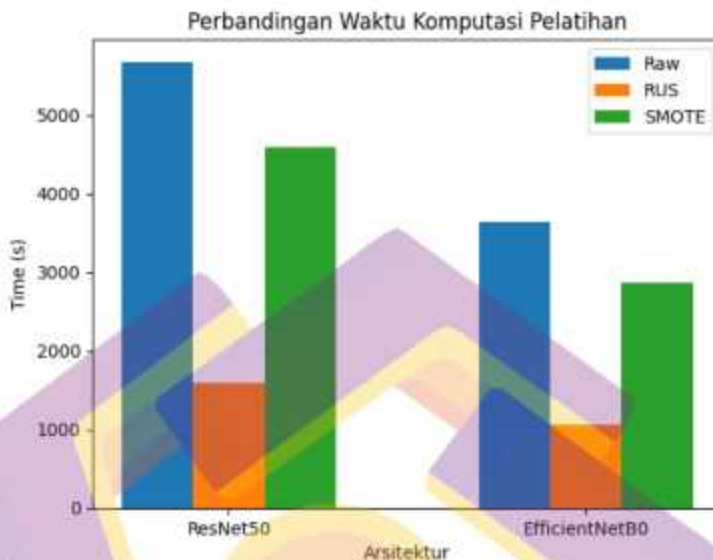
Secara keseluruhan, penggunaan SMOTE meningkatkan akurasi, ketepatan, dan recall. Namun, perlu dicatat bahwa beberapa kelas minoritas kehilangan akurasi dibandingkan dengan kondisi sebelum SMOTE digunakan. Namun, temuan ini tidak mengurangi kemampuan model untuk mengidentifikasi berbagai kelas yang ada dalam tugas klasifikasi tingkat kematangan tandan buah segar kelapa sawit.

Berbeda dengan metode penyeimbangan data menggunakan SMOTE setelah penerapan, penggunaan RUS untuk penyeimbangan data menghasilkan penurunan akurasi yang cukup besar. Tampaknya model tidak dapat mengidentifikasi mayoritas dan minoritas kelas dalam dataset, yang menyebabkan penurunan ini.

Akibatnya, kesimpulan yang dapat diambil adalah bahwa, meskipun dengan beberapa penurunan pada kelas minoritas, penggunaan teknik penyeimbangan data seperti SMOTE dapat meningkatkan performa model secara keseluruhan. Sementara itu, RUS cenderung mengakibatkan penurunan yang signifikan dalam akurasi model, menunjukkan bahwa pemilihan metode penyeimbangan data harus dipertimbangkan dengan hati-hati sesuai dengan karakteristik dataset dan tujuan pemodelan.

4.6. Perbandingan Waktu Komputasi

Selain itu, dalam penelitian ini dicatat waktu komputasi yang dibutuhkan oleh sebuah model selama proses pembelajaran, terutama dalam hal ekstraksi fitur dari masing-masing kelas data. Kami membandingkan waktu komputasi antara data asli (raw) dan data yang telah mengalami proses penyeimbangan menggunakan metode RUS dan SMOTE, yang dapat dilihat dalam Gambar 4.16.



Gambar 4.16. Perbandingan waktu komputasi

Saat melihat waktu komputasi untuk kedua arsitektur, diukur dalam detik, Gambar 4.16 menunjukkan bahwa metode penyeimbangan data RUS memiliki waktu komputasi paling cepat dibandingkan dengan arsitektur ResNet50 dan EfficientNetB0. Ini karena metode RUS mengurangi jumlah data yang signifikan, yang menghasilkan proses pembelajaran yang lebih cepat.

Namun, metode SMOTE memiliki waktu komputasi yang lebih cepat daripada data asli (raw), meskipun data yang dihasilkan oleh SMOTE lebih banyak. Ini disebabkan oleh penggunaan GPU yang berbeda selama proses pembelajaran: GPU T4 digunakan untuk data asli dan RUS, sedangkan GPU V100 digunakan untuk data SMOTE. Penggunaan GPU yang berbeda ini disebabkan oleh kebutuhan metode penyeimbangan data SMOTE untuk daya komputasi yang lebih besar.

Selain factor dari kompleksitas dari masing-masing model, ada faktor eksternal seperti penggunaan GPU dan jaringan yang digunakan, ada faktor internal dalam model yang memengaruhi waktu komputasi. Faktor internal ini termasuk pengaturan parameter pada kedua arsitektur model; dalam hal ini, kondisi parameter seperti optimizer, tingkat pembelajaran, dan ukuran batch yang digunakan berbeda, yang dapat memengaruhi waktu komputasi antara kedua model.

4.7. Perbandingan Kondisi Terbaik Kedua Model

Setelah melakukan semua percobaan skenario 1-7 didapati kondisi terbaik dari kedua model klasifikasi tingkat kematangan tandan buah segar kelapa sawit. Dimana kondisi terbaik ResNet50 adalah pada data alokasi 90/5/5, optimizer SGD, Learning rate 0.00075, batch size 8, epoch 100 dan dropout 0.5 pada data yang seimbang. Sedangkan kondisi terbaik EfficientNetB0 pada optimizer Adam, epoch 100, learning rate 0.0001, batch size yang digunakan 32, dropout 0.5 pada alokasi data 90/5/5 dengan data yang seimbang. Kondisi terbaik kedua model ini memiliki kesamaan pada penggunaan alokasi data, epoch, dropout dan data yang seimbang.

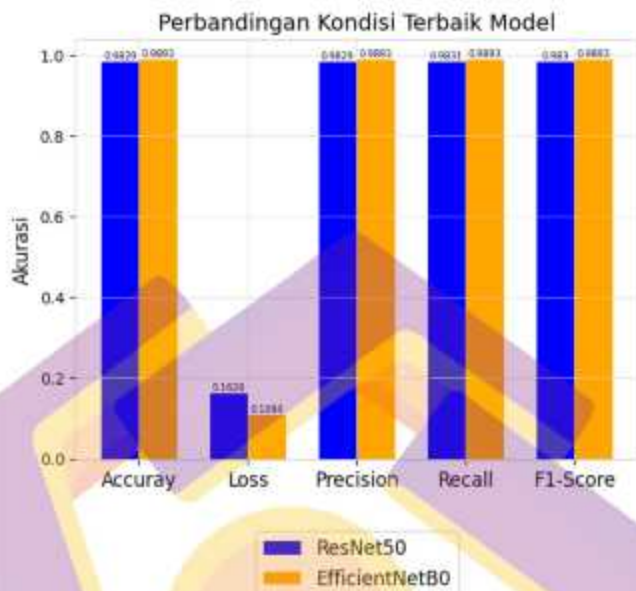
Menemukan kondisi terbaik untuk kedua model klasifikasi tingkat kematangan tandan buah segar kelapa sawit setelah melakukan semua percobaan dalam skenario 1-7. Dalam dataset yang seimbang, kondisi terbaik untuk ResNet50 adalah dengan menggunakan alokasi data 90% untuk pelatihan, 5% untuk validasi, dan 5% untuk pengujian, dengan pengaturan optimizer SGD, tingkat pembelajaran 0.00075, ukuran batch 8, jumlah epoch 100, dan dropout 0.5.

Namun, kondisi optimal untuk EfficientNetB0 adalah dengan menggunakan alokasi data yang sama, yaitu 90% untuk pelatihan, 5% untuk validasi, dan 5% untuk pengujian. Pengaturan optimizer Adam mencakup epoch 100, rasio pembelajaran 0.0001, ukuran batch 32, dan hilangnya 0.5, serta pada dataset yang seimbang. Dalam hal alokasi data, jumlah epoch, pengaturan dropout, dan penggunaan dataset yang seimbang, kondisi terbaik kedua model ini memiliki kesamaan.

Gambar 4.17 menunjukkan perbandingan nilai confusion matrix yang dihasilkan oleh kedua model berdasarkan kondisi terbaik. Dari grafik tersebut, terlihat bahwa EfficientNetB0 mengungguli ResNet50 dalam hal akurasi, kehilangan, ketepatan, recall, dan skor F1 pada kondisi terbaik.

Tingkat akurasi EfficientNetB0 sebesar 98,93% dengan nilai kehilangan sebesar 0,0094; skor ketepatan, recall, dan F1 semua mencapai 98,93%. Ini menunjukkan bahwa EfficientNetB0 memiliki kinerja yang sangat baik dalam klasifikasi tingkat kematangan tandan buah kelapa sawit segar.

Meskipun demikian, perlu dicatat bahwa model klasifikasi ResNet50 menghasilkan nilai yang sangat baik, dengan akurasi sebesar 98,39%, ketepatan sebesar 98,29%, recall sebesar 98,315%, dan nilai F1-score sebesar 98,30%. Dengan nilai-nilai ini, ResNet50 juga berhasil mengklasifikasikan tingkat kematangan tandan buah sawit untuk keempat kelas kematangan yang diuji.



Gambar 4.17. Perbandingan kondisi terbaik Resnet50 dan EfficientNetB0

Perbedaan hasil yang didapatkan dapat dikarenakan faktor dari kompleksitas dari masing-masing model, serta pengaturan parameter pada kedua arsitektur model; dalam hal ini, kondisi parameter seperti optimizer, tingkat pembelajaran, dan ukuran batch yang digunakan berbeda, yang dapat memengaruhi hasil yang didapatkan antara kedua model.

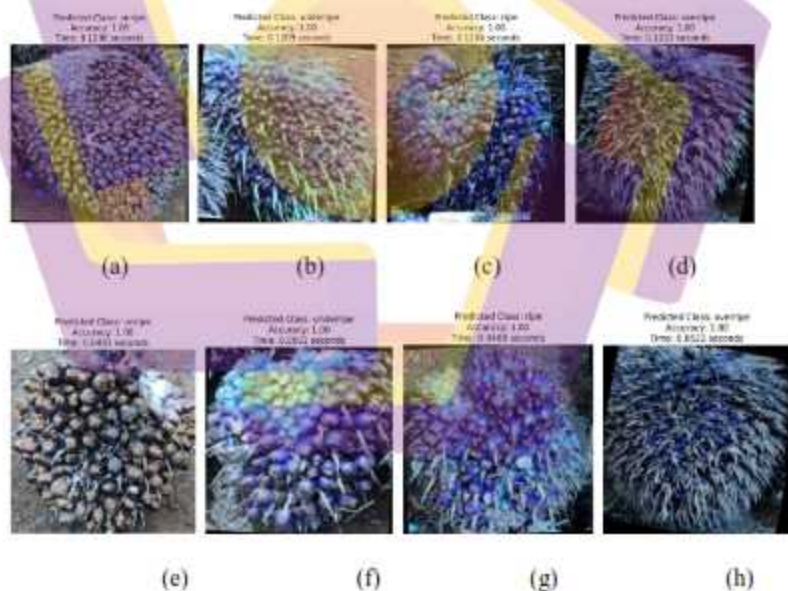
Secara umum, model EfficientNetB0 lebih baik daripada ResNet50 karena memiliki jumlah parameter yang lebih banyak dan kompleksitas yang lebih rendah. Ini membuatnya bekerja dengan lebih efisien dan memiliki kinerja yang lebih baik.

Namun, untuk banyak kasus penggunaan, terutama yang memerlukan efisiensi waktu dan sumber daya, EfficientNetB0 telah terbukti lebih baik dengan mempertahankan atau meningkatkan performa sambil mengurangi kompleksitas

model. Oleh karena itu, pemilihan model bergantung pada sumber daya dan kebutuhan khusus dari tugas. ResNet-50 tetap memiliki kinerja yang baik, terutama dalam situasi di mana sumber daya komputasi tidak terlalu dibatasi.

4.8. Evaluasi Hasil

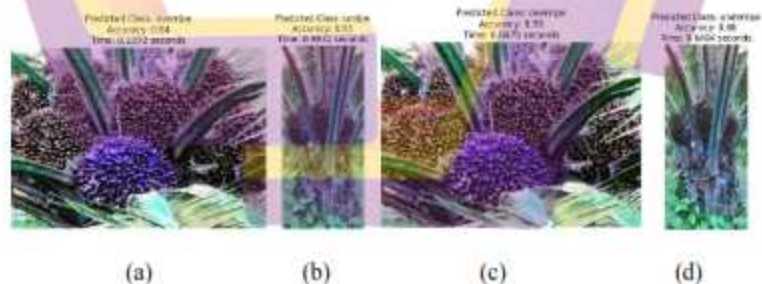
Selain itu, pengujian yang dilakukan dengan data baru dilakukan dalam penelitian ini. Tujuan dari pengujian ini adalah untuk mengetahui berapa lama waktu yang diperlukan oleh model untuk dimuat atau dipenuhi, untuk membuat prediksi tentang gambar baru, dan untuk menentukan kelas data yang tepat. Hasil dari pengujian ini ditunjukkan dalam Gambar 4.18.



Gambar 4.18. Hasil prediksi data baru (a)(b)(c)(d) model ResNet50, (e)(f)(g)(h) model EfficientNetB0

Hasil pengujian pada data baru menggunakan model ResNet50 menunjukkan bahwa waktu yang dibutuhkan untuk memprediksi tingkat kematangan berkisar antara 0.1204 dan 0.1223 detik, dengan kelas "ripe" memprediksi tingkat kematangan tercepat, seperti yang ditunjukkan pada Gambar 4.18. Namun, pengujian tingkat kematangan menggunakan model arsitektur EfficientNetB0 membutuhkan waktu lebih cepat daripada model ResNet50, dengan waktu 0.0469–0.0522 detik.

Kompleksitas model dan jumlah parameter yang digunakan menyebabkan perbedaan waktu ini. EfficientNetB0 memiliki arsitektur CNN yang lebih ringan, sehingga lebih cepat dan efisien. Selain itu, EfficientNetB0 menunjukkan kinerja yang baik bahkan dengan parameter yang lebih sedikit dibandingkan ResNet50. Pada pengujian data baru ini, prediksi kelas TBS yang masih ada pada pohon kelapa sawit juga dilakukan. Hasil pengujian TBS pada pohon ditunjukkan dalam Gambar 4.19.



Gambar 4.19. Hasil prediksi TBS dengan pohon (a)(b) model ResNet50, (c)(d), model EfficientNetB0

Kedua model mengalami kesulitan dalam melakukan prediksi, seperti yang ditunjukkan oleh hasil pengujian pada tandan buah segar (TBS) dengan banyak

buah kelapa sawit di dalamnya, seperti yang ditunjukkan pada Gambar 4.19. Meskipun ResNet50 berhasil meramalkan kelas "Unripe", masih ada masalah. Ini karena kedua model arsitektur CNN yang digunakan dalam klasifikasi tingkat kematangan ini tidak memasukkan proses deteksi objek.

Dalam penelitian ini, kedua model langsung menerima data input gambar dan menggambarkan seluruh gambar tanpa melakukan tahap pendeteksian TBS terlebih dahulu. Metode ini membuat tugas klasifikasi menjadi lebih sulit ketika ada banyak TBS dalam satu gambar, yang dapat mengakibatkan jenis gambar yang lebih tidak akurat.

4.9. Analisis Perbandingan Terhadap Studi Literatur

Setelah analisis yang dilakukan terhadap model arsitektur ResNet50 dan EfficientNetB0 untuk memaksimalkan kondisi masing-masing model dalam penelitian ini, langkah selanjutnya adalah menyajikan rangkuman studi literatur yang menjadi dasar penelitian ini. Tujuan dari rangkuman ini adalah untuk membandingkan hasil penelitian ini dengan temuan dalam literatur yang digunakan sebagai referensi.

Tabel 4.21 menunjukkan hasil perbandingan antara hasil penelitian ini dan studi literatur lainnya. Membandingkan hasil penelitian ini dengan penelitian sebelumnya dapat membantu kita memahami lebih baik bagaimana penelitian ini telah membantu mengklasifikasikan tingkat kematangan tandan buah segar kelapa sawit.

Tabel 4.22. Perbandingan penelitian terkait

Penulis	Metode	Distribusi Data	Jumlah Kelas	Akurasi (%)
(Suharjito et al., 2021)	EfficientNetB0	Separate validation	6 kategori	89.3
(Prasetyo et al., 2020)	ResNet50	Cross validation	-	86
(Mansour et al., 2022)	YOLOv5m	Separate validation	4 kategori	84.2
(Lai et al., 2022)	YOLOv4	Single validation	2 kategori	87.9
(Septiarini et al., 2021)	Artificial Neural Network (ANN)	Single validation	3 Kategori	98.3
(Ashari et al., 2022)	CNN	Single validation	3 kategori	92
(Saleh & Liansitim, 2020)	CNN	Single validation	2 kategori	98
(Susanto et al., 2020)	DenseNet Sigmoid	Separate validation	7 kategori	69
The proposed method	ResNet50	Separate validation	4 kategori	98.29
The proposed method	EfficientNetB0	Separate validation	4 kategori	98.93

Dibandingkan dengan semua penelitian sebelumnya, penelitian ini menggunakan dataset yang berbeda. Penelitian ini menggunakan metode validasi terpisah yang juga digunakan oleh Suharjito, Mansour, dan Susanto, serta model yang digunakan oleh Suharjito.

Hasil penelitian Suharjito mencapai akurasi sebesar 89,3%, penelitian Mansour mencapai 84,2%, penelitian Susanto mencapai akurasi sebesar 69,5%, dan penelitian ini mencapai akurasi yang tinggi sebesar 98,93%.

Selain itu, penelitian lain dengan fokus serupa telah dilakukan oleh Prasetyo, Lai, Septiarini, Ashari, dan Saleh. Penelitian Prasetyo mencapai akurasi sebesar 86%, sedangkan penelitian lain mencapai akurasi antara 87,9% dan 98,3%, dan penelitian Saleh bahkan mencapai akurasi sebesar 98%.

Dengan perbandingan ini, dapat disimpulkan bahwa, meskipun menggunakan dataset yang berbeda dan menggunakan metode validasi terpisah, penelitian ini berhasil mengklasifikasikan tingkat kematangan tandan buah kelapa sawit dengan sangat akurat. Hasil ini menunjukkan kemajuan yang signifikan dalam pengembangan model klasifikasi ini.

Tabel 4.23. Perbandingan gap penelitian terkait

Penulis	Akurasi(%)	The proposed method(EfficientNetB0)%	Gap(%)
(Suharjito et al., 2021)	89.3	98.93	9.63
(Prasetyo et al., 2020)	86	98.93	12.93
(Mansour et al., 2022)	84.2	98.93	14.73
(Lai et al., 2022)	87.9	98.93	11.03
(Septiarini et al., 2021)	98.3	98.93	0.63
(Ashari et al., 2022)	92	98.93	6.93
(Saleh & Liansitim, 2020)	98	98.93	0.93
(Susanto et al., 2020)	69	98.93	29.93

Dalam Tabel 4.22, dapat dilihat bahwa ada perbedaan antara penelitian sebelumnya dan model yang digunakan dalam penelitian ini. Perbedaan ini mengacu pada perbedaan antara hasil akurasi yang ditemukan dalam penelitian ini dan hasil dari penelitian sebelumnya; dalam hal ini, penelitian yang menggunakan model yang sama dengan penelitian ini, yaitu penelitian Suharjito, menunjukkan perbedaan sebesar 9.63%.

Dalam perbandingan ini, penelitian Septiarini memiliki gap akurasi paling rendah, yaitu hanya 0.63, menunjukkan bahwa model yang digunakan dalam

penelitian ini mendekati hasil penelitian Septiarini. Sebaliknya, penelitian Susanto memiliki gap akurasi terbesar, yaitu 29.93.

Berdasarkan perbandingan yang telah dilakukan dan perbedaan dalam hasil penelitian, dapat disimpulkan bahwa model arsitektur EfficientNetB0 yang digunakan dalam penelitian ini sangat baik untuk mengklasifikasikan tingkat kematangan tandan buah kelapa sawit segar dengan akurasi sebesar 98,93%. Hasil ini mengungguli hasil penelitian sebelumnya, tetapi perlu diperhatikan bahwa semua data dalam perbandingan ini berasal dari dataset yang berbeda.

4.10. Pengujian Tambahan

Tujuan dari pengujian tambahan dalam skenario ini adalah untuk memperoleh pemahaman dan informasi yang berguna tentang cara mengklasifikasikan tingkat kematangan tandan buah kelapa sawit. Tujuan dari pengujian ini adalah untuk mengetahui apakah ekstraksi fitur warna dan tekstur tambahan memiliki dampak yang signifikan terhadap keakuratan klasifikasi.

Selain itu, pengujian kedua melibatkan penyesuaian pada parameter yang sama pada kedua platform arsitektur CNN menggunakan teknik transfer learning. Ini dilakukan karena, dalam kondisi terbaik, kedua arsitektur memiliki parameter yang berbeda. Tujuan dari pengujian tambahan ini adalah untuk mengetahui bagaimana perubahan parameter berdampak pada kemampuan untuk mengklasifikasikan tingkat kematangan tandan buah kelapa sawit.

Hasil dari kedua pengujian tambahan ini akan meningkatkan pemahaman tentang komponen yang mempengaruhi kinerja model klasifikasi. Mereka juga

akan membantu kita membuat keputusan tentang konfigurasi parameter dan metode ekstraksi fitur terbaik.

4.10.1. Skenario 8 Warna Dan Tekstur

Dalam pengujian ini, fitur diekstraksi di luar arsitektur kedua model utama, yaitu fitur warna dan tekstur. Untuk mengekstraksi fitur warna, menggunakan metode YBcbr dan LBP (Local Binary Pattern). Untuk mendapatkan fitur warna YBcbr dan LBP yang telah diekstraksi sebelumnya, kami menambah lapisan GAP2d (Global Average Pooling 2D) dan lapisan dense tersembunyi. Gambar 4.20 dan 4.21 menunjukkan model ini.

Percobaan dilakukan pada dua arsitektur menggunakan parameter optimizer Adam: epoch 100, rate pembelajaran 0.0001, ukuran batch 32, dan dropout 0.5. Untuk arsitektur tanpa fitur tambahan, parameter yang sama digunakan untuk alokasi data 90/5/5. Dua input digunakan dalam pengujian ini: gambar TBS kelapa sawit dan fitur YBcbr dan LBP yang diekstrak dari gambar TBS. Tabel 23 menunjukkan hasil dari tes ini.

Layer (type)	Output Shape	Param #	Connected to
image_input (InputLayer)	(None, 224, 224, 3)	0	[]
resnet50 (Functional)	(None, 7, 7, 2048)	2358771	['image_input[0][0]']
combined_features_input (InputLayer)	(None, 770)	0	[]
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2048)	0	['resnet50[0][0]']
dense_2 (Dense)	(None, 128)	80712	['combined_features_input[0][0]']
concatenate_1 (Concatenate)	(None, 2176)	0	['global_average_pooling2d_1[0][0]', 'dense_2[0][0]']
dropout_2 (Dropout)	(None, 2176)	0	['concatenate_1[0][0]']
dense_3 (Dense)	(None, 96)	206208	['dropout_2[0][0]']
dropout_3 (Dropout)	(None, 96)	0	['dense_3[0][0]']
output (Dense)	(None, 4)	244	['dropout_3[0][0]']

Total params: 2183209 (88.88 MB)
 Trainable params: 2176210 (86.96 MB)
 Non-trainable params: 53198 (207.30 KB)

Dumlah layer dalam model gabungannya: 10

Gambar 4.20. model ResNet50 fitur

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	(None, 224, 224, 3)	0	[]
efficientnetb0 (Functional)	(None, 7, 7, 1280)	660971	['input_3[0][0]']
input_4 (InputLayer)	(None, 770)	0	[]
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1280)	0	['efficientnetb0[0][0]']
dense_1 (Dense)	(None, 128)	80712	['input_4[0][0]']
concatenate_1 (Concatenate)	(None, 1408)	0	['global_average_pooling2d_1[0][0]', 'dense_1[0][0]']
dropout_1 (Dropout)	(None, 1408)	0	['concatenate_1[0][0]']
dense_2 (Dense)	(None, 96)	84340	['dropout_1[0][0]']
dropout_2 (Dropout)	(None, 96)	0	['dense_2[0][0]']
dense_3 (Dense)	(None, 4)	244	['dropout_2[0][0]']

Total params: 4,234,807
 Trainable params: 4,192,844
 Non-trainable params: 41,963

Dumlah layer dalam model gabungannya: 10

Gambar 4.21. model ResNet50 fitur

Tabel 4.24. Pengujian tambahan Fitur warna dan tekstur

	Training	Val	Test	Precision	Recall	F1-Score	Time(s)
Resnet50	100	95.95	94.33	94.48	94.40	94.44	4659.19
EfficientNetB0	99.98	98.38	98.79	99.15	97.99	98.5	3645.69
ResNet50 warna dan tekstur	99.98	95.55	95.55	94.31	96.43	95.36	4837.98
EfficientNetB0 warna dan tekstur	99.91	97.98	96.76	97.76	97.96	97.86	3702.13

Berdasarkan Tabel 4.23, terlihat bahwa pengujian fitur warna dan tekstur pada kedua arsitektur menghasilkan tingkat akurasi yang cukup tinggi, namun hasil ini belum mencapai tingkat akurasi yang diberikan oleh model arsitektur EfficientNetB0 tanpa ekstraksi fitur tambahan.

Meskipun begitu, pada ResNet dengan tambahan fitur ekstraksi, terlihat dampak yang positif terhadap hasil akurasi dibandingkan dengan ResNet50 tanpa fitur ekstraksi tambahan. Hal ini mengindikasikan bahwa ekstraksi fitur warna dan tekstur dapat meningkatkan performa model ResNet.

4.10.2. Skenario 9 Parameter Yang Sama

Kondisi terbaik untuk kedua arsitektur berbeda, jadi pengujian ini akan dilakukan dengan parameter yang digunakan oleh arsitektur lain; dalam hal ini, ResNet50 akan digunakan untuk EfficientNetB0 dan sebaliknya. Selanjutnya, hasil pengujian akan dibandingkan dengan kinerja arsitektur pada parameter aslinya.

Untuk pengujian ini, optimizer Adam akan digunakan untuk mengkonfigurasi ResNet50 dengan epoch 100, rasio pembelajaran 0.0001, ukuran batch 32, dan dropout 0.5. Data akan didistribusikan dengan proporsi 90 persen

untuk pelatihan, 5 persen untuk validasi, dan 5 persen untuk pengujian. Di sisi lain, EfficientNetB0 akan menggunakan alokasi data yang sama untuk pelatihan, validasi, dan pengujian. Tetapi arsitektur ini akan dikonfigurasi dengan optimizer SGD, rate learning sebesar 0.00075, ukuran batch sebesar 8, jumlah epoch sebanyak 100, dan dropout sebesar 0.5. Untuk menguji kedua arsitektur, metode penyeimbangan data SMOTE akan digunakan. Hasil percobaan dapat dilihat dalam Tabel 24.

Tabel 4.25. Pengujian tambahan Parameter

Kondisi	Training	Val	Test	Precision	Recall	F1-Score	Time(s)
Resnet50	100	98.29	98.29	98.29	98.31	98.30	4594.27
EfficientNetB0	100	99.14	98.93	98.93	98.93	98.93	2867.34
ResNet50 Parameter EfficientNetB0	99.98	97.43	96.37	96.37	96.44	96.40	2717.75
EfficientNetB0 Parameter ResNet50	99.90	99.14	98.72	98.72	98.74	98.73	4665.84

Pengujian tambahan yang melibatkan penggunaan parameter yang berbeda dari parameter yang digunakan sebelumnya pada masing-masing arsitektur ResNet50 telah dilakukan, seperti yang terlihat pada Tabel 4.23. Hasilnya menunjukkan bahwa penerapan parameter EfficientNetB0 pada arsitektur ResNet50 meningkatkan waktu komputasi, yang menjadi lebih cepat dibandingkan dengan waktu yang dihasilkan dengan parameter asli pada ResNet50 dan bahkan lebih singkat.

Namun, perlu diperhatikan bahwa penggunaan parameter EfficientNetB0 pada ResNet50 tidak menghasilkan peningkatan nilai akurasi yang signifikan dibandingkan dengan penggunaan parameter asli pada ResNet50 atau

EfficientNetB0 asli. Namun, akurasi yang diperoleh tetap tinggi, mencapai 96.37%, yang cukup untuk tugas klasifikasi. Oleh karena itu, dapat disimpulkan bahwa penggunaan parameter EfficientNetB0 pada ResNet50 lebih berpengaruh terhadap peningkatan waktu komputasi daripada peningkatan akurasi.

Di sisi lain, pengujian yang melibatkan penggunaan parameter ResNet50 pada EfficientNetB0 menunjukkan hasil yang berbeda dari pengujian sebelumnya yang melibatkan penggunaan parameter ResNet50 pada EfficientNetB0. Dalam pengujian ini, penggunaan parameter ResNet50 pada EfficientNetB0 meningkatkan akurasi dibandingkan dengan kinerja EfficientNetB0 asli, tetapi akurasi yang dicapai masih di bawah tingkat akurasi yang dapat diterima.

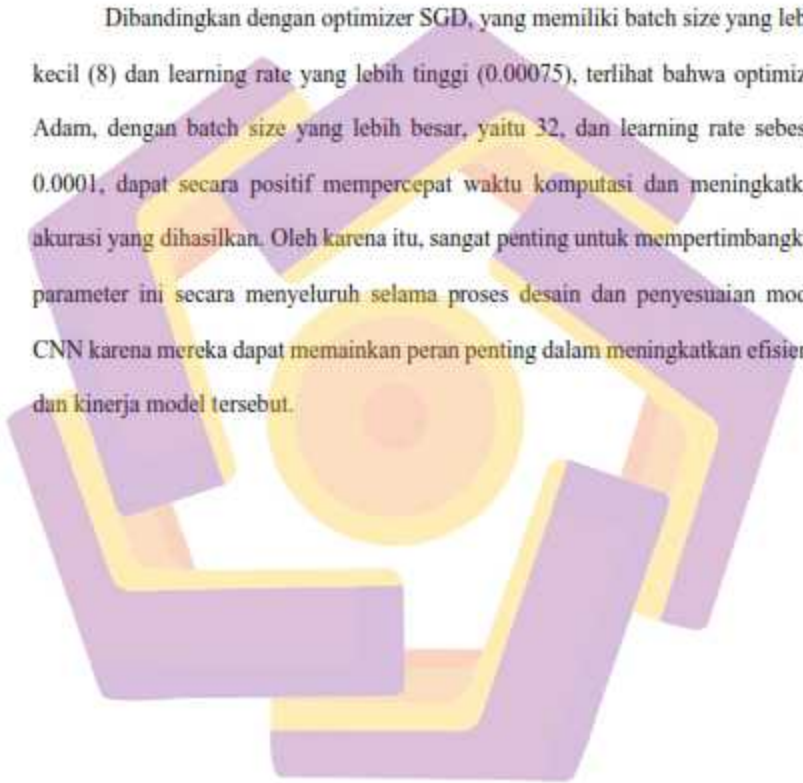
Sangat penting untuk dicatat bahwa perbedaan akurasi EfficientNetB0 dengan parameter ResNet50 dan EfficientNetB0 asli sangat kecil, hanya sekitar 0,21%, menunjukkan bahwa meskipun penggunaan parameter ResNet50 dapat meningkatkan akurasi, perbedaan ini tidak signifikan.

Namun, ada konsekuensi yang harus dipertimbangkan, yaitu waktu yang diperlukan untuk komputasi. Dalam pengujian ini, penggunaan parameter ResNet50 pada EfficientNetB0 membutuhkan waktu komputasi yang lebih lama daripada penggunaan parameter asli pada EfficientNetB0. Oleh karena itu, tingkat akurasi dan efisiensi waktu komputasi harus dipertimbangkan saat memilih model klasifikasi untuk menentukan tingkat kematangan tandan buah kelapa sawit segar.

Hasil penelitian ini menunjukkan bahwa penerapan parameter yang tepat dalam arsitektur Convolutional Neural Network (CNN) dapat berdampak signifikan pada kinerja kedua model arsitektur tersebut; perbedaan yang

paling mencolok antara kedua model ini terletak pada penggunaan optimizer, ukuran batch, dan nilai rasio pembelajaran. Pada akhirnya, dampak yang signifikan dari penerapan parameter yang tepat akan timbul pada kinerja kedua model tersebut.

Dibandingkan dengan optimizer SGD, yang memiliki batch size yang lebih kecil (8) dan learning rate yang lebih tinggi (0.00075), terlihat bahwa optimizer Adam, dengan batch size yang lebih besar, yaitu 32, dan learning rate sebesar 0.0001, dapat secara positif mempercepat waktu komputasi dan meningkatkan akurasi yang dihasilkan. Oleh karena itu, sangat penting untuk mempertimbangkan parameter ini secara menyeluruh selama proses desain dan penyesuaian model CNN karena mereka dapat memainkan peran penting dalam meningkatkan efisiensi dan kinerja model tersebut.



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan dari hasil penelitian yang telah dilakukan maka dapat disimpulkan bahwa:

1. Rancangan metode arsitektur terbaik yang dapat meningkatkan akurasi pada penelitian ini terhadap klasifikasi tingkat kematangan tandan buah segar kelapa sawit adalah model arsitektur EfficientNetB0 dikombinasikan dengan Hyperparameter tuning dan proses penyeimbangan data menggunakan Synthetic Minority Oversampling Technique (SMOTE). Nilai yang dihasilkan adalah akurasi 98.93%, precision 98.93%, recall 98.93%.
2. Penyetelan hyperparameter berpengaruh dalam meningkatkan akurasi pada model arsitektur ResNet50 dan EfficientNetb0 dengan transfer learning. Dalam penelitian ini dilakukan 5 penyetelan hyperparameter yaitu optimizer, learning rate, batch size, epoch dan dropout.
3. Penggunaan alokasi data yang tepat dapat meningkatkan nilai akurasi pada klasifikasi TBS kelapa sawit. Dalam penelitian ini 3 alokasi data dilakukan percobaan, pada ResNet50 dengan hasil akurasi 94.33% untuk alokasi 70/15/15, 95.34% untuk alokasi 80/10/10 dan 95.93% untuk alokasi 90/5/5. Pada EfficeintNetB0 dengan hasil akurasi 95.68% untuk alokasi 70/15/15, 95.55% untuk alokasi 80/10/10 dan 98.79% untuk alokasi 90/5/5.

4. Penggunaan penerapan penyeimbangan data menggunakan teknik oversampling Synthetic Minority Oversampling Technique (SMOTE) mempengaruhi dalam meningkatkan akurasi kinerja model yang dibuktikan pada skenario 7 baik pada model klasifikasi Resnet50 dengan peningkatan akurasi 1.93% sedangkan pada EfficientNetB0 dengan peningkatan akurasi 0.14%.

5. Hasil waktu komputasi terbaik pada klasifikasi TBS kelapa sawit adalah pada kombinasi EfficientNetB0 dengan kombinasi RUS dengan waktu komputasi 1064.75 detik.

6. Penggunaan Random Under Sampling (RUS) terhadap penyeimbangan data untuk klasifikasi TBS kelapa sawit berpengaruh terhadap akurasi dan waktu komputasi. Pada akurasi pengaruh RUS kurang baik karena menurunkan akurasi dibandingkan data sebelum menggunakan RUS, untuk waktu komputasi memberikan pengaruh yang sangat baik dengan waktu komputasi yang lebih cepat dari data raw dan penggunaan SMOTE.

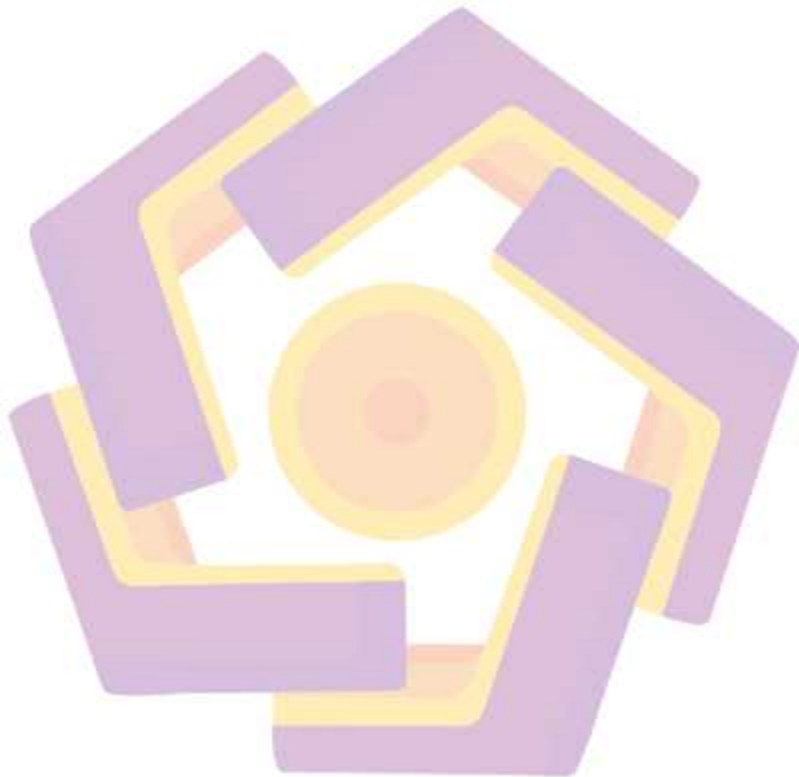
7. Kinerja model klasifikasi TBS kelapa sawit kurang bisa melakukan klasifikasi TBS dengan background pohon kelapa sawit. Hal lain adalah model klasifikasi tidak dapat melakukan klasifikasi pada gambar dengan banyak TBS di dalamnya.

5.2. Saran

Adapun beberapa saran yang dapat dijadikan rujukan bagi pengembangan penelitian ini, yaitu:

1. Jumlah dan varian dataset dapat diperbanyak agar yang didapat nantinya lebih optimal, seperti varian TBS yang berbeda dan data dengan background pohon sawit.

2. Melakukan eksplorasi lebih lanjut dalam melakukan klasifikasi arsitektur ringan lainnya seperti mobilenet dan lainnya.
3. Penelitian berikutnya dapat terlebih dahulu melakukan deteksi terhadap TBS kelapa sawit kemudian dilakukan klasifikasi terhadap TBS.



DAFTAR PUSTAKA

PUSTAKA BUKU

- Andono, P. N., T. Sutojo, & Muljono. (2017). *Pengolahan Citra Digital*. Penerbit Andi.
- Arhami, M., & Nasir, M. (2020). *Data Mining - Algoritma dan Implementasi*. Penerbit Andi.
- Arnia, F., & Munadi, K. (2018). *Pengantar Teknik Pengolahan Citra & Visi Komputer*. Penerbit Tombak.
- Hidayatullah, P. (2017). *Pengolahan Citra Digital: Teori dan Aplikasi Nyata*. Penerbit Informatika.
- Kadir, A., & Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Penerbit Andi.
- Madenda, S., & Drajat, A. M. (2015). *Pengolahan citra dan video digital: teori, aplikasi dan pemrograman menggunakan MATLAB*. Erlangga.
- Primartha, R. (2021). *Algoritma Machine Learning*. Informatika Bandung.
- Putra, D. (2010). *Pengolahan Citra Digital*. Penerbit Andi.
- Suyanto. (2017). *Data mining : untuk klasifikasi dan klasterisasi data*. Penerbit Informatika .
- Suyanto, Ramadhani, K. N., & Mandala, S. (2019). *Deep learning : modernisasi machine learning untuk big data*. Penerbit Informatika.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Abbasi, R., Martinez, P., & Ahmad, R. (2022). The digitization of agricultural industry – a systematic literature review on agriculture 4.0. *Smart Agricultural Technology*, 2, 100042. <https://doi.org/10.1016/j.atech.2022.100042>
- Ancin, M., Pindado, E., & Sánchez, M. (2022). New trends in the global digital transformation process of the agri-food sector: An exploratory study based on Twitter. *Agricultural Systems*, 203. <https://doi.org/10.1016/j.agsy.2022.103520>

- Ashari, S., Yanris, G. J., & Purnama, I. (2022). Oil Palm Fruit Ripeness Detection using Deep Learning. *Sinkron*, 7(2), 649–656. <https://doi.org/10.33395/sinkron.v7i2.11420>
- Being, C. Y., Hashim, N., Maringgal, B., & Wondi, M. H. (2020). A review of non-destructive techniques applied for measuring quality of oil palm fresh fruit bunches. *Journal of Agricultural and Food Engineering*, 1(1), 1–6. <https://doi.org/10.37865/jafe.2020.0002>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16(Sept. 28), 321–357. <https://arxiv.org/pdf/1106.1813.pdf> <http://www.snopes.com/horrors/insects/telamonia.asp>
- De Angeli, K., Gao, S., Danciu, I., Durbin, E. B., Wu, X. C., Stroup, A., Doherty, J., Schwartz, S., Wiggins, C., Damesyn, M., Coyle, L., Penberthy, L., Tourassi, G. D., & Yoon, H. J. (2022). Class imbalance in out-of-distribution datasets: Improving the robustness of the TextCNN for the classification of rare cancer types. *Journal of Biomedical Informatics*, 125(November 2021), 103957. <https://doi.org/10.1016/j.jbi.2021.103957>
- Durai, S. K. S., & Shamili, M. D. (2022). Smart farming using Machine Learning and Deep Learning techniques. *Decision Analytics Journal*, 3, 100041. <https://doi.org/10.1016/j.dajour.2022.100041>
- Hamdani, H., Septiarini, A., Sunyoto, A., Suyanto, S., & Utaminigrum, F. (2021). Detection of oil palm leaf disease based on color histogram and supervised classifier. *Optik*, 245. <https://doi.org/10.1016/j.ijleo.2021.167753>
- Himmah, E. F., Widyarningsih, M., & Maysaroh, M. (2020). Identifikasi Kematangan Buah Kelapa Sawit Berdasarkan Warna RGB Dan HSV Menggunakan Metode K-Means Clustering. *Jurnal Sains Dan Informatika*, 6(2), 193–202. <https://doi.org/10.34128/jsi.v6i2.242>
- Jaisakthi, S. M., Mirunalini, P., Aravindan, C., & Appavu, R. (2022). Classification of skin cancer from dermoscopic images using deep neural network architectures. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-022-13847-3>
- Junkwon, P., Takigawa, T., Okamoto, H., Hasegawa, H., Koike, M., Sakai, K., Siruntawineti, J., Chaeychomsri, W., Vanavichit, A., Tittinuchanon, P., & Bahalayodhin, B. (2009). Hyperspectral imaging for nondestructive determination of internal qualities for oil palm (*Elaeis guineensis* Jacq. var. *tenera*). In *Agricultural Information Research* (Vol. 18, Issue 3). www.jstage.jst.go.jp/

- Lai, J. W., Ramli, H. R., Ismail, L. I., & Hasan, W. Z. W. (2022). Real-Time Detection of Ripe Oil Palm Fresh Fruit Bunch Based on YOLOv4. *IEEE Access*, 10, 95763–95770. <https://doi.org/10.1109/ACCESS.2022.3204762>
- Lye Chew, C., Yong Ng, C., Onn Hong, W., Yeong Wu, T., Lee, Y.-Y., Ee Low, L., San Kong, P., & Seng Chan, E. (1947). Improving Sustainability of Palm Oil Production by Increasing Oil Extraction Rate: a Review. <https://doi.org/10.1007/s11947-020-02555-1> Published
- Mansour, M. Y. M. A., D. Dambul, K., & Choo, K. Y. (2022). Object Detection Algorithms for Ripeness Classification of Oil Palm Fresh Fruit Bunch. *International Journal of Technology*, 13(6), 1326. <https://doi.org/10.14716/ijtech.v13i6.5932>
- Prasetyo, N. A., Pranowo, & Santoso, A. J. (2020). Automatic detection and calculation of palm oil fresh fruit bunches using faster R-CNN. *International Journal of Applied Science and Engineering*, 17(2), 121–134. [https://doi.org/10.6703/IJASE.202005_17\(2\).121](https://doi.org/10.6703/IJASE.202005_17(2).121)
- Puri, A., & Kumar Gupta, M. (2021). Knowledge discovery from noisy imbalanced and incomplete binary class data. *Expert Systems with Applications*, 181(March 2020), 115179. <https://doi.org/10.1016/j.eswa.2021.115179>
- Raj, T., Hashim, F. H., Huddin, A. B., Hussain, A., Ibrahim, M. F., & Abdul, P. M. (2021). Classification of oil palm fresh fruit maturity based on carotene content from Raman spectra. *Scientific Reports*, 11(1). <https://doi.org/10.1038/s41598-021-97857-5>
- Rijswijk, K., Klerkx, L., Bacco, M., Bartolini, F., Bulten, E., Debruyne, L., Dessen, J., Scotti, L., & Brunori, G. (2021). Digital transformation of agriculture and rural areas: A socio-cyber-physical system framework to support responsabilisation. *Journal of Rural Studies*, 85, 79–90. <https://doi.org/10.1016/j.jrurstud.2021.05.003>
- Saleh, A. Y., & Liansitim, E. (2020). Palm oil classification using deep learning. *Science in Information Technology Letters*, 1(1), 1–8. <https://doi.org/10.31763/sitech.v1i1.1>
- Sinambela, R., Mandang, T., Subrata, I. D. M., & Hermawan, W. (2020). Application of an inductive sensor system for identifying ripeness and forecasting harvest time of oil palm. *Scientia Horticulturae*, 265. <https://doi.org/10.1016/j.scienta.2020.109231>
- Suharjito, Elwirehardja, G. N., & Prayoga, J. S. (2021). Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. *Computers and Electronics in Agriculture*, 188. <https://doi.org/10.1016/j.compag.2021.106359>

- Susanto, A., Wawan Cenggoro, T., & Pardamean, B. (2020). Oil Palm Fruit Image Ripeness Classification with Computer Vision using Deep Learning and Visual Attention. https://farm3.static.flickr.com/2337/2387965665_1d4278c6
- Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. <http://arxiv.org/abs/1905.11946>
- Fitra Maulana, F., & Rochmawati, N. (n.d.). Klasifikasi Citra Buah Menggunakan Convolutional Neural Network.
- Ali, K., Shaikh, Z. A., Khan, A. A., & Laghari, A. A. (2022). Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer. *Neuroscience Informatics*, 2(4), 100034. <https://doi.org/10.1016/j.neuri.2021.100034>
- K. Sunilkumar, & D. S. Sparjan Babu. (2013). Surface color based prediction of oil content in oil palm (*Elaeis guineensis* Jacq.) fresh fruit bunch. *African Journal of Agricultural Research*.
- Saranya, N., Srinivasan, K., & Kumar, S. K. P. (2022). Banana ripeness stage identification: a deep learning approach. *Journal of Ambient Intelligence and Humanized Computing*, 13(8), 4033–4039. <https://doi.org/10.1007/s12652-021-03267-w>
- Afis Julianto, Andi Sunyoto, & Ferry Wahyu Wibowo. (2022). Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, 3(2), 98–105. <https://doi.org/10.46764/teknimedia.v3i2.77>
- Haqee, R., Ho, S. B., Chai, L., & Abdullah, A. (2022). Parameter and Hyperparameter Optimisation of Deep Neural Network Model for Personalised Predictions of Asthma. *Journal of Advances in Information Technology*, 13(5), 512–517. <https://doi.org/10.12720/jait.13.5.512-517>
- Konar, J., Khandelwal, P., & Tripathi, R. (2020). Comparison of Various Learning Rate Scheduling Techniques on Convolutional Neural Network. 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science, SCEECS 2020. <https://doi.org/10.1109/SCEECS48394.2020.94>
- Lee, S., & Kim, T. (2023). Impact of Deep Learning Optimizers and Hyperparameter Tuning on the Performance of Bearing Fault Diagnosis. *IEEE Access*, 11(May), 55046–55070. <https://doi.org/10.1109/ACCESS.2023.3281910>
- Minarno, A. E., Hazmi Cokro Mandiri, M., Munarko, Y., & Hariyady, H. (2021). Convolutional Neural Network with Hyperparameter Tuning for Brain Tumor Classification. *Kinetik: Game Technology, Information System, Computer*

Network, Computing, Electronics, and Control, 4.
<https://doi.org/10.22219/kinetik.v6i2.1219>

- Motta, D., Bandeira Santos, A. Á., Souza Machado, B. A., Vicente Ribeiro-Filho, O. G., Arriaga Camargo, L. O., Valdenegro-Toro, M. A., Kirchner, F., & Badaró, R. (2020). Optimization of convolutional neural network hyperparameters for automatic classification of adult mosquitoes. *PLoS ONE*, 15(7), 1–30. <https://doi.org/10.1371/journal.pone.0234959>
- Ramirez-Gonzalez, M., Segundo Sevilla, F. R., Korba, P., & Castellanos-Bustamante, R. (2022). Convolutional neural nets with hyperparameter optimization and feature importance for power system static security assessment. *Electric Power Systems Research*, 211(April), 108203. <https://doi.org/10.1016/j.epsr.2022.108203>
- Sen, S. Y., & Ozkurt, N. (2020). Convolutional Neural Network Hyperparameter Tuning with Adam Optimizer for ECG Classification. *Proceedings - 2020 Innovations in Intelligent Systems and Applications Conference, ASYU 2020*, 978. <https://doi.org/10.1109/ASYU50717.2020.9259896>
- Alfatni, M. S. M., Mohamed Shariff, A. R., Ben Saeed, O. M., Albhbah, A. M., & Mustapha, A. (2020). Colour Feature Extraction Techniques for Real Time System of Oil Palm Fresh Fruit Bunch Maturity Grading. *IOP Conference Series: Earth and Environmental Science*, 540(1). <https://doi.org/10.1088/1755-1315/540/1/012092>
- Azlim Khan, A. K., & Ahamed Hassain Malim, N. H. (2023). Comparative Studies on Resampling Techniques in Machine Learning and Deep Learning Models for Drug-Target Interaction Prediction. In *Molecules* (Vol. 28, Issue 4). MDPI. <https://doi.org/10.3390/molecules28041663>
- Ghazalli, S. A., Selamat, H., Khamis, N., & Haniff, M. F. (2023). Short Review on Palm Oil Fresh Fruit Bunches Ripeness and Classification Technique. *Journal of Advanced Research in Applied Mechanics*, 106(1), 37–47. <https://doi.org/10.37934/aram.106.1.3747>
- Lai, J. W., Ramli, H. R., Ismail, L. I., & Wan Hasan, W. Z. (2023). Oil Palm Fresh Fruit Bunch Ripeness Detection Methods: A Systematic Review. In *Agriculture (Switzerland)* (Vol. 13, Issue 1). MDPI. <https://doi.org/10.3390/agriculture13010156>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. <https://doi.org/10.1038/nature14539>
- Sinambela, R., Mandang, T., Subrata, I. D. M., & Hermawan, W. (2020). Application of an inductive sensor system for identifying ripeness and

forecasting harvest time of oil palm. *Scientia Horticulturae*, 265. <https://doi.org/10.1016/j.scienta.2020.109231>

Suryani, D., Yulianti, A., Maghfiroh, E. L., & Alber, J. (n.d.). SISTEMASI: Jurnal Sistem Informasi Klasifikasi Kualitas Produk Kelapa Sawit Menggunakan Metode Naïve Bayes Quality Classification of Palm Oil Products Using Naïve Bayes Method. <http://sistemasi.ftik.unisi.ac.id>

Zein, O. M. E., Soliman, M. M., Elkholy, A. K., & Ghali, N. I. (2021). Transfer Learning Based Model for Pneumonia Detection in Chest X-ray Images. *International Journal of Intelligent Engineering and Systems*, 14(5), 56–66. <https://doi.org/10.22266/ijies2021.1031.06>

