

TESIS

**KLASIFIKASI KACANG PISTACHIO MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK (CNN)**



Disusun oleh:

Nama : LISDA
NIM : 21.51.2103
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023**

TESIS

**KLASIFIKASI KACANG PISTACHIO MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK (CNN)**

**CLASSIFICATION OF PISTACHIO NUT USING
CONVOLUTIONAL NEURAL NETWORK (CNN)**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : LISDA
NIM : 21.51.2103
Konsentrasi : Business Intelligence

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

2023

HALAMAN PENGESAHAN

**KLASIFIKASI KACANG PISTACHIO MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK (CNN)**

**CLASSIFICATION OF PISTACHIO NUT USING
CONVOLUTIONAL NEURAL NETWORK (CNN)**

Dipersiapkan dan Disusun oleh

LISDA

21.51.2103

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 05 Oktober 2023

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

Yogyakarta, 05 Oktober 2023

Rektor

Prof. Dr. M. Suvanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

KLASIFIKASI KACANG PISTACHIO MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)

CLASSIFICATION OF PISTACHIO NUT USING CONVOLUTIONAL NEURAL NETWORK (CNN)

Dipersiapkan dan Disusun oleh

LISDA
21.51.2103

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 05 Oktober 2023

Pembimbing Utama

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

Pembimbing Pendamping

Dhani Ariatmanto, M.Kom., Ph.D.
NIK. 190302197

Anggota Tim Penguji

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

Hanafi, S.Kom., M.Eng., Ph.D.
NIK. 190302024

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

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

Yogyakarta, 05 Oktober 2023
Direktur Program Pascasarjana

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

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : LISDA
NIM : 21.51.2103
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
**KLASIFIKASI KACANG PISTACHIO MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK (CNN)**

Dosen Pembimbing Utama : Prof. Dr. Kusriani, M.Kom.,
Dosen Pembimbing Pendamping : Dhani Ariantanto, M.Kom., Ph.D.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai sumber dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 05 Oktober 2023
Yang Menyatakan,


LISDA

HALAMAN PERSEMBAHAN

Segala puji bagi Allah Subhanahu wa ta'ala atas segala limpahan rahmat karunia serta hidayahNya sehingga tesis ini dapat terselesaikan. Hasil ini saya persembahkan untuk :

1. Kedua orang tua saya yang saya cintai, dengan segala cinta dan kasih sayangnya, doa, harapan serta perjuangannya.
2. Saudari-saudari saya yang selalu mendukung dan mendoakan.
3. Sahabat dan teman – teman yang menjadi support sistem terbaik.



HALAMAN MOTTO

“Tidak masalah dianggap bukan siapa-siapa, dan tak perlu mati-matian membuktikan kita ini layak. Respect, persahabatan, kasih sayang, bahkan cinta sekalipun tidak baik dipaksakan melainkan tumbuh secara alami. Ketika kita melewatinya dengan tulus terus memperbaiki diri, maka esok lusa kita bisa jadi siapa-siapa”



KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah Subhanahu wa ta'ala atas limpahan rahmat, hidayah serta inayah-Nya, penulis masih diberi kesempatan dan kemudahan untuk menyelesaikan tesis ini.

Penulis mengucapkan terimakasih yang sebesar-besarnya, kepada :

1. Bapak Prof. Dr. M.Suyanto, MM, selaku rektor Amikom Yogyakarta
2. Ibu Prof. Dr. Kusriani, M.Kom, selaku pembimbing utama yang telah mendampingi dalam penyusunan tesis ini.
3. Bapak Dhani Ariatmanto, M.Kom., Ph.d, selaku pembimbing pendamping yang selalu memberi arahan kepada penulis dalam pengerjaan tesis ini.
4. Kedua orang tua tercinta yang tak pernah lelah dalam memberikan dukungan dan doanya.
5. Teman – teman MTI-2021 kelas A Universitas Amikom Yogyakarta yang telah memberikan dukungan. Dan kepada semua yang tidak bisa penulis sebutkan satu persatu, yang telah membantu proses tesis ini.

Semoga tesis ini dapat menambah pengetahuan dan memberi manfaat bagi para pembacanya serta dapat digunakan sebagai salah satu referensi untuk penelitian selanjutnya.

Yogyakarta, 05 Oktober 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
INTISARI.....	xv
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penelitian.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1. Tinjauan Pustaka.....	7
2.2. Keaslian Penelitian.....	10

2.3. Landasan Teori.....	15
2.3.1 Klasifikasi.....	15
2.3.2 Pistachio.....	15
2.3.3 Convolutional Neural Network (CNN).....	16
2.3.4 Gambaran Arsitektur Convolutional Neural Network (CNN).....	16
2.3.5 CNN's Layer.....	19
2.3.6 Hyperparameters.....	23
2.3.7 Pre Trained Model CNN.....	25
2.3.8 Confusion Matrix.....	27
BAB III METODE PENELITIAN.....	29
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	29
3.2. Metode Pengumpulan Data.....	29
3.3. Metode Analisis Data.....	29
3.4. Alur Penelitian.....	29
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	35
4.1. Pengumpulan Data.....	35
4.2. Preprocessing Data.....	36
4.2.1 Split Data.....	36
4.2.2 Augmentasi.....	36
4.3. Proses Klasifikasi Menggunakan CNN.....	40
4.3.1 Inception-V3.....	40
4.3.1 ResNet50.....	40
4.4. Skenario Percobaan.....	41

4.4.1 Skenario Percobaan Pertama.....	43
4.4.2 Skenario Percobaan Kedua	43
4.4.3 Skenario Percobaan Ketiga	43
4.4.4 Skenario Percobaan Keempat	45
4.5. Proses Pengujian	46
4.5.1 Pengujian Skenario Pertama	46
4.5.2 Pengujian Skenario Kedua	48
4.5.3 Pengujian Skenario Ketiga	50
4.5.4 Pengujian Skenario Keempat	52
4.6. Evaluasi Hasil Penelitian.....	55
4.6.1 Hasil Akurasi Inception-V3	64
4.6.2 Hasil Akurasi ResNet50.....	65
4.6.3 Perbandingan Hasil Akurasi Kedua Arsitektur	66
BAB V PENUTUP.....	68
5.1. Kesimpulan	68
5.2. Saran	69
DAFTAR PUSTAKA	70

DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian.....	10
Tabel 2.2. Pengujian menggunakan confusion matrix	27
Tabel 3.1. Detail pembagian dataset	32
Tabel 4.1. Sampel pistachio	35
Tabel 4.2. Skenario Percobaan.....	42
Tabel 4.3. Percobaan skenario 1	43
Tabel 4.4. Percobaan skenario 2	43
Tabel 4.5. Percobaan skenario 3	44
Tabel 4.6. Percobaan skenario 4	45
Tabel 4.7. Hasil pengujian	62

DAFTAR GAMBAR

Gambar 2.1. Arsitektur CNN	17
Gambar 2.2. Urutan Lapisan CNN.....	19
Gambar 2.3. Visualisasi input dan filter pada CNN.....	20
Gambar 2.4. Fungsi ReLu	22
Gambar 2.5. Sampel max pooling.....	23
Gambar 2.6. Arsitektur Inception V3.....	26
Gambar 2.7. Arsitektur ResNet50.....	27
Gambar 3.1. Alur penelitian.....	30
Gambar 3.2. Sampel pistachio jenis kirmizi	31
Gambar 3.3. Sampel pistachio jenis siirt.....	31
Gambar 3.4. Tahapan preprocessing	31
Gambar 4.1. Library import dataset	36
Gambar 4.2. Split dataset	36
Gambar 4.3. Sampel kirmizi dan siirt	37
Gambar 4.4. List kode proses augmentasi (flips).....	37
Gambar 4.5. Hasil augmentasi (flips)	38
Gambar 4.6. List kode proses augmentasi (rotations).....	38
Gambar 4.7. Hasil augmentasi (rotations).....	39
Gambar 4.8. List kode proses augmentasi (zoom).....	39
Gambar 4.9. Hasil augmentasi (zoom).....	39
Gambar 4.10. List kode fungsi arsitektur inception v3.....	40

Gambar 4.11. List kode fungsi arsitektur resnet50	41
Gambar 4.12. Kode penambahan hyperparameter skenario percobaan	44
Gambar 4.13. Hasil pengujian inception v3	47
Gambar 4.14. Hasil pengujian resnet50	49
Gambar 4.15. Hasil pengujian inception v3 + hyperparameter	51
Gambar 4.16. Hasil pengujian resnet 50 + hyperparameter	53
Gambar 4.17. Grafik persentase hasil percobaan skenario 1 dan 2	55
Gambar 4.18. Grafik persentase hasil percobaan skenario 3 epoch 50	57
Gambar 4.19. Grafik persentase hasil percobaan skenario 3 epoch 100	58
Gambar 4.20. Grafik persentase hasil percobaan skenario 4 epoch 50	59
Gambar 4.21. Grafik persentase hasil percobaan skenario 4 epoch 100	60
Gambar 4.22. Hasil arsitektur inception v3	64
Gambar 4.23. Hasil arsitektur resnet 50	65

INTISARI

Sektor pertanian merupakan bagian terpenting yang ikut membantu dalam pembangunan sebuah negara di abad ini dalam menyokong perekonomian. Seiring bertambahnya zaman, persaingan di sektor pertanian akan mengalami peningkatan persaingan yang dibuktikan dengan semakin tingginya produktivitas, mutu, dan efisiensi usaha. Inovasi-inovasi baru yang membantu sektor pertanian semakin dibutuhkan untuk meningkatkan kualitas dan kuantitas yang dihasilkan bisa maksimal. Salah satu sektor pertanian yang butuh inovasi baru adalah pistachio.

Pistachio merupakan jenis kacang kacangan yang memiliki kandungan gizi tinggi. Namun, beberapa jenis kacang pistachio memiliki bentuk yang sama, mulai dari cangkang, batang serta biji, sehingga sulit untuk dibedakan utamanya jenis kirmizi dan siirt. Hal ini mempengaruhi produktivitas kerja utamanya pada proses pengemasan serta pemasaran. Sehingga dibutuhkan inovasi baru untuk mengenali dan mengklasifikasikan jenis kacang pistachio terutama pistachio jenis kirmizi dan siirt agar proses pengemasan lebih mudah dan lebih cepat.

Pada penelitian ini, penulis melakukan klasifikasi pada kacang pistachio jenis kirmizi dan siirt menggunakan arsitektur CNN yaitu InceptionV3 dan ResNet50 dengan tujuan untuk mengetahui dan membandingkan tingkat akurasi dari arsitektur CNN yang digunakan. Dari hasil penelitian didapatkan nilai akurasi tertinggi pada penelitian ini terhadap klasifikasi kacang pistachio adalah arsitektur Inception-V3 dengan penerapan hyperparameter pada proses training, mendapatkan nilai akurasi sebesar 98.36%. Sedangkan pada penggunaan ResNet 50 mendapatkan nilai akurasi sebesar 89.65%. Kinerja inception v3 lebih baik dari resnet 50 sehingga penggunaan inception v3 dengan adjusting hyperparameter bisa dipertimbangkan untuk penelitian berikutnya.

Kata kunci: Klasifikasi, Arsitektur CNN, Kacang Pistachio, Perbandingan Akurasi

ABSTRACT

The agricultural sector is the most important part that helps in the development of a country in this century in supporting the economy. As time goes by, competition in the agricultural sector will experience increased competition as evidenced by higher productivity, quality and business efficiency. New innovations that help the agricultural sector are increasingly needed to increase the quality and quantity produced to the maximum. One agricultural sector that needs new innovation is pistachio.

Pistachios are a type of nut that has high nutritional content. However, several types of pistachio nuts have the same shape, starting from the shell, stem and seeds, so it is difficult to differentiate them, especially the scarlet and red types. This affects work productivity, especially in the packaging and marketing processes. So new innovations are needed to recognize and classify types of pistachio nuts, especially scarlet and red pistachios so that the packaging process is easier and faster.

In this research, the author carried out classification of scarlet and siirt pistachio nuts using the CNN architecture, namely InceptionV3 and ResNet50 with the aim of knowing and comparing the level of accuracy of the CNN architecture used. From the research results, it was found that the highest accuracy value in this study for the classification of pistachio nuts was the Inception-V3 architecture with the application of hyperparameters in the training process, obtaining an accuracy value of 98.36%. Meanwhile, using ResNet 50 gets an accuracy value of 89.65%. The performance of Inception v3 is better than Resnet 50 so that the use of Inception v3 with adjusting hyperparameters can be considered for future research.

Keyword: *Classification, CNN Architecture, Pistachio Nut, Accuracy Comparison*

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Sektor pertanian merupakan bagian terpenting yang ikut membantu dalam pembangunan sebuah negara di abad ini dalam menyokong perekonomian (Moeis Faizal Rahmanto et al., 2020). Seiring bertambahnya zaman, persaingan di sektor pertanian akan mengalami peningkatan persaingan yang dibuktikan dengan semakin tingginya produktivitas, mutu dan efisiensi usaha (Hartati Ery, 2020). Inovasi-inovasi baru yang membantu sektor pertanian semakin dibutuhkan untuk meningkatkan kualitas dan kuantitas yang dihasilkan bisa maksimal. Salah satu sektor pertanian yang butuh inovasi baru adalah pistachio.

Pistachio merupakan jenis tanaman penghasil biji-bijian yang biasa disebut sebagai kacang pistachio. Beberapa jenis kacang pistachio memiliki bentuk yang sama, mulai dari cangkang, batang serta biji. Pistachio umumnya tumbuh di daerah kering dan kondisi tanah yang tidak cukup baik (Mandalari Giuseppina et al., 2022) seperti Iran, Turkmenistan, dan Azerbaijan barat (Vidyarthi Sriram K. et al., 2020). Biji hijau cerah yang dihasilkan tanaman pistachio ini biasanya digunakan untuk menghias makanan atau juga digunakan dalam campuran olahan makanan seperti es krim (Bonifazi Giuseppe et al., 2021). Pistachio merupakan jenis kacang-kacangan yang memiliki kandungan gizi tinggi karena merupakan sumber protein, serat, asam lemak tak jenuh tunggal, mineral dan vitamin yang baik, serta

karotenoid, asam fenolik, flavonoid, dan antosianin (Mandalari Giuseppina et al., 2022).

Sebagai produk pertanian yang mahal, pistachio mencerminkan harga konsumen tergantung pada kualitas produk. Oleh karena itu, penentuan kualitas pistachio yang sudah dikupas merupakan isu penting dari segi pertanian, ekonomi, ekspor, dan pemasaran. Kualitas akan mengarah pada peningkatan konsumsi, produktivitas kerja utamanya pada proses pengemasan serta pemasaran. Selain itu, sama penting untuk menentukan kualitas pistachio dengan akurasi tinggi dan aplikasi yang mudah melalui sistem cerdas untuk mencegah kerugian ekonomi dalam hal ekspor dan pemasaran (Singh Dilbagh et al., 2022). Konsekuensinya, diperlukan metode dan teknologi baru untuk pemisahan dan klasifikasi dari pistachio.

Pada penelitian ini akan berfokus pada jenis Kirmizi dan Siirt. Kedua jenis tersebut memiliki nilai ekonomi dan pasar yang berbeda, dimana Kirmizi biasa digunakan oleh industri permen dan kue-kue manis karena memiliki warna hijau tua, rasa dan aromanya yang khas, sedangkan Siirt lebih disukai sebagai makanan ringan karena tingkat retaknya yang tinggi dan bentuknya yang bulat (Özkan İlker Ali et al., 2021). Namun, proses yang dilakukan untuk melakukan pemisahan kacang pistachio ini masih dilakukan dengan pengetahuan dasar yang dimiliki. Hal ini tentu saja sangat rawan menimbulkan kesalahan dalam melakukan klasifikasi karena kacang pistachio memiliki bentuk yang mirip untuk setiap jenis. Sehingga dibutuhkan inovasi baru untuk mengenali dan mengklasifikasikan jenis kacang

pistachio terutama pistachio jenis kirmizi dan siirt agar proses pengemasan lebih mudah dan lebih cepat.

Beberapa penelitian yang membahas tentang klasifikasi di bidang pertanian sebelumnya sudah pernah dilakukan. Penelitian dengan judul *Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models* (Singh Dilbagh et al., 2022), melakukan klasifikasi pistachio dengan menggunakan metode transfer learning yaitu AlexNet, VGG16, and VGG19. Penelitian dilakukan dengan menggunakan 1832 images, 916 tipe kirmizi dan 916 tipe siirt yang selanjutnya dibagi dengan perbandingan 80:20 untuk data training dan data testing. Klasifikasi dengan metode transfer learning AlexNet, VGG 16, and VGG 19 diperoleh hasil berturut-turut yaitu 94.42%, 96.84%, dan 96.14%. Performa model dievaluasi melalui metrik sensitivitas, spesifisitas, presisi, dan skor F-1. Selain itu, kurva ROC dan nilai AUC digunakan dalam evaluasi kinerja. Keberhasilan klasifikasi tertinggi dicapai dengan VGG16 yaitu 96,84%.

Penelitian yang dilakukan oleh (Bonifazi Giuseppe et al., 2021) yang berjudul "*Contaminant detection in pistachio nuts by different classification methods applied to short-wave infrared hyperspectral images*" penelitian ini bertujuan untuk membandingkan empat model klasifikasi untuk menentukan yang paling strategis untuk mengontrol kualitas otomatis kacang pistachio yang dapat dimakan. Empat model klasifikasi yang berbeda yang digunakan dalam penelitian ini, yaitu PLS-DA dengan efisiensi 0.86% bisa dimakan dan 0.80% untuk tidak bisa dimakan. PCA-DA dengan efisiensi 0.87% bisa dimakan dan 0.85% untuk tidak bisa dimakan, selanjutnya PCA-kNN dengan efisiensi 0.93% bisa dimakan

dan 0.92% untuk tidak bisa dimakan dan yang terakhir adalah CART dengan efisiensi 0.82% bisa dimakan dan 0.82% untuk tidak bisa dimakan.

Penelitian dengan judul "*Classification of Pistachio Species Using Improved K-NN Classifier*" (Ozkan İlker Ali et al., 2021), mengadopsi metode KNN untuk melakukan klasifikasi pistachio menjadi dua kelas yaitu Kirmizi dan Siirt dengan total citra yang digunakan sebanyak 2148 gambar. Pengimplementasian KNN untuk proses klasifikasi dilakukan dengan menggunakan tiga skenario yaitu dengan menggunakan KNN, Weighted KNN dan PCA based k-NN. Dari ketiga skenario tersebut, diperoleh hasil secara berturut-turut yaitu 83%, 87%, dan 94%. Keberhasilan klasifikasi tertinggi dicapai dengan menggunakan PCA yaitu dengan akurasi sebesar 94%.

Pada penelitian ini bertujuan untuk menerapkan model arsitektur untuk mengenali dan mengklasifikasikan jenis pistachio agar dapat meningkatkan efisiensi kecepatan dan kemudahan dalam proses pengemasan serta menangani kebutuhan pasar dengan sesuai. Untuk tujuan ini, penerapan computer vision dan metode pemrosesan gambar dapat digunakan secara bersama-sama. *Convolutional Neural Network (CNN)* adalah metode yang dapat digunakan untuk mengidentifikasi jenis pistachio Kirmizi dan Siirt. Pemanfaatan CNN dengan arsitektur Inception-V3 dan Resnet 50 diharapkan mampu mengklasifikasikan jenis pistachio menggunakan gambar dengan cepat dan efektif, sehingga pemanfaatannya mampu membantu dalam melakukan klasifikasi pistachio secara efektif dan efisien.

1.2. Rumusan Masalah

Dalam penelitian ini dirumuskan masalah-masalah sebagai berikut:

- a. Berapa tingkat akurasi arsitektur Inception-V3 dan Resnet 50 yang dihasilkan dalam proses klasifikasi pistachio?
- b. Manakah arsitektur yang menghasilkan nilai akurasi tertinggi dari dua arsitektur CNN yang digunakan dalam proses klasifikasi pistachio?

1.3. Batasan Masalah

Batasan masalah dalam suatu penelitian sangat diperlukan, sehingga penelitian dapat lebih terarah dan tidak menyimpang jauh. Adapun batasan masalah dalam penelitian ini, adalah:

- a. Jenis pistachio yang digunakan untuk data penelitian ini adalah Kirmizi dan siirt yang terdiri dari 1832 gambar. 916 untuk jenis Kirmizi dan 916 untuk jenis Siirt.
- b. Data objek gambar pistachio untuk proses training dan testing berasal dari *Kaggle.com* (<https://www.muratkoklu.com/datasets/>).
- c. Dilakukan preprocessing data dengan melakukan augmentasi, normalisasi, serta pembagian data training dan data testing secara manual.
- d. Ekstraksi fitur gambar berdasarkan ciri ciri bentuk dari objek citra pistachio dengan menggunakan algoritma CNN.
- e. Model arsitektur yang digunakan adalah Inception-V3 dan Resnet50.
- f. Maksimal nilai epoch adalah 100.

- g. Proses evaluasi model klasifikasi menggunakan metode *confusion matrix*.

1.4. Tujuan Penelitian

Tujuan dari penelitian yang penulis lakukan adalah untuk mengetahui tingkat akurasi dari arsitektur CNN yang digunakan pada klasifikasi kacang pistachio.

1.5. Manfaat Penelitian

Manfaat yang didapatkan dari penelitian ini, adalah:

- a. Rancangan arsitektur yang diterapkan dapat dijalankan dan dapat mengklasifikasikan gambar pistachio dengan akurasi yang baik sesuai dengan batasan yang telah ditentukan.
- b. Memberikan kontribusi penelitian terhadap gambar pistachio menggunakan metode CNN dengan dua model arsitektur terhadap nilai akurasi yang dihasilkan.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Beberapa penelitian yang membahas tentang klasifikasi di bidang pertanian sebelumnya terkait dengan kacang pistachio yang sudah pernah dilakukan antara lain, penelitian dengan judul "*Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models*" (Singh Dilbagh et al., 2022), melakukan klasifikasi dengan menggunakan metode transfer learning yaitu AlexNet, VGG16, and VGG19. Penelitian dilakukan dengan menggunakan 1832 images, 916 of Kirmizi type and 916 of Siirt type yang selanjutnya dibagi dengan perbandingan 80:20 untuk data training dan data testing. Klasifikasi dengan metode transfer learning AlexNet, VGG 16, and VGG 19 diperoleh hasil berturut-turut yaitu 94,42%, 96,84%, dan 96,14%. Performa model dievaluasi melalui metrik sensitivitas, spesifisitas, presisi, dan skor F-1. Selain itu, kurva ROC dan nilai AUC digunakan dalam evaluasi kinerja. Keberhasilan klasifikasi tertinggi dicapai dengan VGG16 yaitu 96,84%.

Penelitian dengan judul "*Classification of Pistachio Species Using Improved K-NN Classifier*" (Ozkan İlker Ali et al., 2021), mengadopsi metode KNN untuk melakukan klasifikasi pistachio menjadi dua kelas yaitu Kirmizi dan Siirt dengan total citra yang digunakan sebanyak 2148 gambar. Pengimplementasian KNN untuk proses klasifikasi dilakukan dengan

menggunakan tiga skenario yaitu dengan menggunakan KNN, Weighted KNN dan PCA based k-NN. Dari ketiga skenario tersebut, diperoleh hasil secara berturut-turut yaitu 83%, 87%, dan 94%. Keberhasilan klasifikasi tertinggi dicapai dengan menggunakan PCA yaitu dengan akurasi sebesar 94%.

Penelitian yang dilakukan oleh (Bonifazi et al., 2021) yang berjudul "*Contaminant detection in pistachio nuts by different classification methods applied to short-wave infrared hyperspectral images*" penelitian membandingkan empat model klasifikasi untuk menentukan yang paling strategis untuk mengontrol kualitas otomatis kacang pistachio yang dapat dimakan. Empat model klasifikasi yang berbeda yang digunakan dalam penelitian ini, yaitu PLS-DA dengan efisiensi 0.86% bisa dimakan dan 0.80% untuk tidak bisa dimakan, PCA-DA dengan efisiensi 0.87% bisa dimakan dan 0.85% untuk tidak bisa dimakan, selanjutnya PCA-kNN dengan efisiensi 0.93% bisa dimakan dan 0.92% untuk tidak bisa dimakan dan yang terakhir adalah CART dengan efisiensi 0.82% bisa dimakan dan 0.82% untuk tidak bisa dimakan.

Penelitian lain yang berjudul "*Prediction of size and mass of pistachio kernels using random Forest machine learning*" (Vidyarthi Sriram K. et al., 2020), memprediksi massa dan ukuran biji pistachio menggunakan metode *Random Forest Machine Learning*. Ukuran dan area satu pistachio mentah dicapai dalam bentuk piksel menggunakan algoritma pemrosesan gambar. Piksel kemudian diubah menjadi fitur vektor yang mewakili ukuran dan luas satu pistachio mentah, yang menjadi dasar untuk memprediksi panjang dan massa pistachio mentah atau kernel. Panjang rata-rata yang diukur dari 100 biji pistachio adalah 18,002 mm,

sedangkan panjang prediksi yang diperoleh dari algoritma pengolahan citra adalah 18,608 mm.

Penelitian lain yang dilakukan oleh (Menevşeođlu Ahmed et al., 2020), yang berjudul "*Non-targeted approach to detect green pea and peanut adulteration in pistachio by using portable FTIR, and UV-Vis spectroscopy*", penelitian ini bertujuan untuk menghasilkan metode non-target untuk spektrometer FTIR dan UV-Vis portable untuk mendeteksi pemalsuan kacang hijau dan kacang tanah pada pistachio. FT-IR memprediksi tingkat bahan pengganggu dengan 0,99 dan standar kesalahan masing-masing prediksi sebesar 0,66 untuk kacang hijau dan 0,80 untuk kacang tanah. Sedangkan prediksi dengan spektrometer UV-Vis menunjukkan bahwa kacang hijau diprediksi sebesar 0,58 dan kacang tanah diprediksi sebesar 0,14.

Penelitian lain yang berjudul "*Determination of green pea and spinach adulteration in pistachio nuts using NIR spectroscopy*" (Genis Huseyin Efe et al., 2021), mengembangkan teknik penentuan pemalsuan kacang hijau dan bayam dalam kacang pistachio dengan menggunakan spektroskopi inframerah dekat (NIR) ditambah dengan analisis kemometrik. Dengan membedakan sampel murni dan sampel yang dipalsukan. Model dari regresi kuadrat terkecil parsial (PLSR) dikembangkan untuk memprediksi tingkat pemalsuan kacang hijau dan bayam dengan nilai root mean square error of prediksi (RMSEP) yang ditemukan senilai 0,9968 untuk kacang hijau dan 4,69 untuk bayam.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian
Klasifikasi Kacang Pistachio Menggunakan Convolutional Neural Network (CNN)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models</i>	Singh Dilbag et al. <i>Journal electronics</i> , 2022.	Penelitian ini bertujuan untuk mengidentifikasi dua jenis kacang pistachio yang sering di tanam di Turki, dengan mengklasifikasikannya melalui tiga model arsitektur CNN yang berbeda.	Dari hasil klasifikasi yang dilakukan dengan dua jenis kacang pistachio yaitu kimmizi dan siirt masing-masing diperoleh keberhasilan klasifikasi 94,42% untuk model AlexNet, 96,84% untuk model VGG16, dan 96,14% untuk model VGG19. Dan klasifikasi tertinggi dicapai dengan model VGG16 sebesar 96,84%.	Untuk mencapai keberhasilan klasifikasi yang berbeda sebaiknya menggunakan beberapa model yang berbeda dengan memanfaatkan metode kecerdasan buatan dan juga tergantung pada jumlah gambar dalam dataset.	Perbedaan antara penelitian sebelumnya dengan penelitian yang akan dilakukan, yaitu pada penelitian ini menggunakan algoritma CNN dengan tiga jenis model arsitektur yang dipakai yaitu AlexNet, VGG16 dan VGG19.
2	<i>Classification of Pistachio Species Using Improved k-NN Classifier</i>	Ilker Ali Ozkan, Murat Koklu, & Ridvan Saracoglu. <i>Progress in Nutrition</i> , 2021	Untuk mengembangkan model klasifikasi yang berbeda dari metode pemisahan tradisional, berdasarkan pengolahan citra dan kecerdasan buatan	Hasil eksperimen menunjukkan bahwa pendekatan yang diusulkan mencapai keberhasilan klasifikasi sebesar 94,18%. Model klasifikasi kinerja	Sebaiknya sistem yang dikembangkan dapat lebih ditingkatkan dengan penggunaan metode pembelajaran mesin hybrid. Dikombinasikan dengan suara ultrasonik	Pada penelitian ini menggunakan algoritma k-NN dengan dua model yang berbeda yaitu model k-NN dan model k-NN berbobot.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)
 Klasifikasi Kacang Pistachio Menggunakan Convolutional Neural Network (CNN)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			yang mampu memberikan klasifikasi yang dibutuhkan	tinggi yang disajikan memberikan kebutuhan penting untuk pemisahan spesies pistachio dan meningkatkan nilai ekonomi spesies.	untuk membuat pengklasifikasi untuk kondisi cangkang terbuka dan cangkang tertutup pada pistachio.	
3	<i>Contaminant detection in pistachio nuts by different classification methods applied to short-wave infrared hyperspectral images</i>	Bonifazi Giuseppe et al. Elsevier. 2021.	Penelitian ini bertujuan untuk membandingkan empat model klasifikasi untuk menentukan yang paling strategis untuk mengontrol kualitas otomatis kacang pistachio yang dapat dimakan.	Empat model klasifikasi yang berbeda yang digunakan dalam penelitian ini, yaitu PLS-DA dengan efisiensi 0.86% bisa dimakan dan 0.80% untuk tidak bisa dimakan. PCA-DA dengan efisiensi 0.87% bisa dimakan dan 0.85% untuk tidak bisa dimakan, selanjutnya PCA-kNN dengan efisiensi 0.93% bisa dimakan dan 0.92% untuk tidak bisa dimakan dan yang terakhir adalah CART dengan efisiensi 0.82% bisa dimakan dan	Variabel data yang digunakan dalam penelitian ini belum maksimal untuk pengklasifikasian dalam mengontrol antara kacang pistachio yang dapat dimakan dan yang tidak dapat dimakan, karena memiliki sifat yang sama.	Pada penelitian ini, menggunakan empat jenis model klasifikasi yaitu, PLS-DA, PCA-DA, PCA-kNN dan CART dalam mengklasifikasikan kacang pistachio yang dapat dimakan.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)
 Klasifikasi Kacang Pistachio Menggunakan Convolutional Neural Network (CNN)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				0.82% untuk tidak bisa dimakan.		
4	<i>Prediction of size and mass of pistachio kernels using random Forest machine learning</i>	Vidyarthi Sriram K. et al. <i>Journal of Food Process Engineering</i> . 2020	Untuk memprediksi massa dan ukuran biji pistachio menggunakan metode <i>Random Forest Machine Learning</i> .	Ukuran dan area satu pistachio mentah dicapai dalam bentuk piksel menggunakan algoritma pemrosesan gambar. Piksel kemudian diubah menjadi fitur vektor yang mewakili ukuran dan luas satu pistachio mentah, yang menjadi dasar untuk memprediksi panjang dan massa pistachio mentah atau kernel. Panjang rata-rata yang diukur dari 100 biji pistachio adalah 18,00 mm, sedangkan panjang prediksi yang diperoleh dari algoritma pengolahan citra adalah 18,60 mm	Tidak boleh terjadi kesalahan dalam pengukuran karena, jika terjadi kesalahan pengukuran pada objek dapat memperbanyak kesalahan dalam estimasi ukuran yang selanjutnya dapat berkontribusi dalam korelasi antara massa pistachio yang diukur.	Penelitian ini menggunakan metode <i>Random Forest Machine Learning</i> dalam memprediksi massa dan ukuran biji pistachio.
5	<i>Non-targeted approach to detect green pea and peanut adulteration</i>	Menevseoglu Ahmed, Aykas Didem Peren, & Adal Eda. <i>Journal</i>	Tujuan dari penelitian ini adalah untuk menghasilkan metode non-target untuk	FT-IR memprediksi tingkat bahan pengganggu dengan 0,9% dan standar	Teknik perhitungan yang dipakai sebagai daya pembeda belum mampu membedakan	Pada penelitian ini menggunakan spektrometer FTIR dan UV-Vis portable, untuk menghasilkan metode

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)
Klasifikasi Kacang Pistachio Menggunakan Convolutional Neural Network (CNN)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	<i>in pistachio by using portable FT-IR, and UV-Vis spectroscopy</i>	<i>of Food Measurement and Characterization</i> 2020	spektrometer FTIR dan UV-Vis portable untuk mendeteksi pemalsuan kacang hijau dan kacang tanah pada pistachio	kesalahan masing-masing prediksi sebesar 0,66% untuk kacang hijau dan 0,80% untuk kacang tanah. Sedangkan prediksi dengan spektrometer UV-Vis menunjukkan bahwa kacang hijau diprediksi dengan sebesar 0,58%. Demikian pula, tingkat kacang tanah diprediksi sebesar 0,14%.	banyak sampel untuk keakuratan deteksi pemalsuan.	non-target dalam mendeteksi pemalsuan kacang hijau dan kacang tanah pada pistachio.
6	<i>Determination of green pea and spinach adulteration in pistachio nuts using NIR spectroscopy</i>	Genis Husayn Efe, Durna Sahin, & Boyaci Ismail Hakki. <i>Lwt-Food Science and Technology</i> . 2021	Tujuan dari penelitian ini adalah untuk mengembangkan teknik penentuan pemalsuan kacang hijau dan bayam dalam kacang pistachio dengan menggunakan spektroskopi inframerah dekat (NIR) ditambah	Dengan membedakan sampel murni dan sampel yang dipalsukan. Model dari regresi kuadrat terkecil parsial (PLSR) dikembangkan untuk memprediksi tingkat pemalsuan kacang hijau dan bayam dengan nilai root mean square error of prediksi (RMSEP)	Teknik yang digunakan bergantung pada jumlah data yang besar dari metode spektrofotometri, tidak dapat mengidentifikasi analit tertentu secara bersamaan dalam lingkungan yang rumit dengan menggunakan metode ini.	Pada penelitian ini mengembangkan teknik penentuan pemalsuan menggunakan spektroskopi inframerah dekat (NIR) ditambah dengan analisis kemometrik. Dengan menggunakan sampel kacang hijau dan bayam dicampur dengan kacang pistachio.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)
 Klasifikasi Kacang Pistachio Menggunakan Convolutional Neural Network (CNN)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			dengan analisis kemometrik.	yang ditemukan sebesar 0,9% untuk kacang hijau dan 4,6% untuk bayam.		

2.3. Landasan Teori

2.3.1. Klasifikasi

Dalam bukunya Santoso (2007) klasifikasi merupakan langkah atau cara dalam upaya membentuk suatu model atau fungsi yang digunakan dalam menjelaskan atau membedakan konsep kelas data. Dengan melakukan proses ini, suatu objek dapat dikenali dan dikelompokkan berdasarkan kelasnya dengan cara memperkirakan berdasarkan hasil dari model yang telah dibentuk.

2.3.2. Pistachio

Pistachio merupakan jenis tanaman penghasil biji-bijian yang biasa disebut sebagai kacang pistachio. Pistachio umumnya tumbuh di daerah kering dan kondisi tanah yang tidak cukup baik (Mandalari Giuseppina et al., 2022) seperti Iran, Turkmenistan, Turki dan Azerbaijan barat (Vidyarthi Sriram K. et al., 2020). Kacang pistachio memiliki kandungan gizi tinggi karena merupakan sumber protein, serat, asam lemak tak jenuh tunggal, mineral dan vitamin yang baik, serta karotenoid, asam fenolik, flavonoid, dan antosianin (Mandalari Giuseppina et al., 2022). Namun, beberapa jenis kacang pistachio memiliki kesamaan seperti jenis siirt dan kirmizi.

Kirmizi adalah salah satu jenis kacang pistachio yang berasal dari Turki. Kirmizi adalah kata dalam bahasa Turki yang berarti "merah", dan kacang pistachio ini dinamakan demikian karena warna kulitnya yang cenderung merah atau ungu. Kacang pistachio Kirmizi memiliki rasa yang khas dan sering kali dianggap memiliki rasa yang lebih kaya dibandingkan dengan beberapa varietas pistachio lainnya.

Siirt adalah sebuah kota di wilayah tenggara Turki yang terkenal dengan produksi kacang pistachio. Kacang pistachio yang berasal dari wilayah Siirt sering kali disebut sebagai "kacang pistachio Siirt". Jenis kacang ini sebenarnya masih varietas dari kacang kirmizi sehingga memiliki cangkang, biji dan batang yang sama dengan pistachio jenis kirmizi, bahkan warna yang sama yaitu warna kulit yang agak merah cerah dan daging yang kaya rasa. Adanya kesamaan terhadap kacang pistachio jenis kirmizi dan siirt seringkali sulit untuk dikenali, namun kacang pistachio Siirt sering dianggap sebagai varietas pistachio yang berkualitas tinggi (Mandalari Giuseppina et al., 2022).

2.3.3. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah algoritma klasifikasi hasil dari pengembangan *multilayer perceptron (MLP)* yang dibangun untuk memproses data dua dimensi. Karena kedalaman jaringan yang tinggi dan banyak diterapkan pada data gambar/citra sehingga CNN termasuk *deep neural network*. Algoritma ini pertama kali dikembangkan oleh Kunihiko Fukushima yaitu peneliti asal Jepang dan diberi nama NeoCognitron Fukushima (1980).

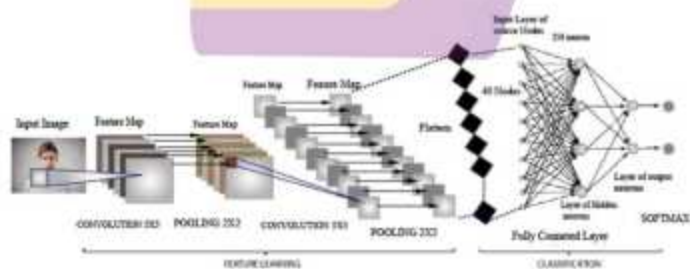
Convolutional Neural Network (CNN) merupakan salah satu model deep learning yang banyak digunakan untuk keperluan analisis citra/visual. Secara prinsip, CNN meniru visual cortex pada mamalia. CNN memiliki neuron-neuron yang disusun secara tiga dimensi, jadi memiliki panjang, lebar dan tinggi. Sehingga CNN sangat efektif dan efisien untuk menganalisis image/gambar. (Primartha, 2018).

2.3.4. Gambaran Arsitektur Convolutional Neural Network (CNN)

Analogi dari arsitektur Convolutional Neural Network (CNN) adalah seperti pola aktivitas neuron pada otak manusia yang saling terhubung antar neuron cara kerja ini diilhami oleh visual cortex. Dimana setiap neuron akan merespon setiap rangsangan dengan batas bidang visual yang dikenali sebagai bidang reseptif. Dari kumpulan bidang tersebut berupa tumpang tindih berguna menutupi seluruh area visual.

Dalam model arsitektur CNN proses feature learning sangat bergantung pada kedalaman suatu gambar. Semakin dalam suatu gambar maka semakin banyak mesin melakukan ekstraksi yang didapatkan sehingga informasi berupa pola yang di dapat akan semakin jelas terbentuk. Nilai dari pola berbentuk value inilah yang nantinya akan dikonversi menjadi vektor dan kemudian masuk pada tahap klasifikasi (Li & Zhang, 2018). Pada proses classifier ini model ConvErtis bekerja dalam melakukan tugas klasifikasi objek berdasarkan kelasnya.

Pada dasarnya berbagai arsitektur model CNN mengikuti arsitektur yang sama, pada Gambar 2.1, diilustrasikan menggunakan gambar sebagai input kemudian pada mesin model CNN melakukan operasi convolution, operasi pooling, diikuti oleh sejumlah layer yang terhubung sepenuhnya (fully connected layer).

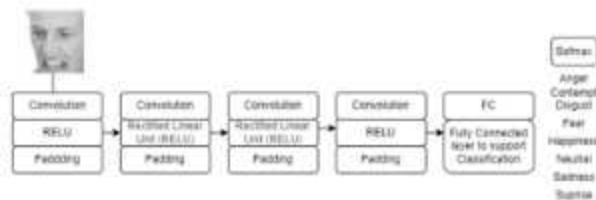


Gambar 2.1. Arsitektur *Convolutional Neural Network* (Alam, 2022).

Berdasarkan Gambar 2.1, arsitektur dari algoritma CNN dibagi menjadi 2 tahapan proses. Pada proses feature learning, secara umum ada 3 lapisan proses ekstraksi fitur. Lapisan-lapisan ini sering disebut dengan convolution layer, activation dan pooling layer. Lapisan-lapisan ini akan melakukan operasi matematika untuk membentuk kedalaman data agar mendapatkan pola secara spesifik. Ketiga layer tersebut memiliki urutan proses yang tidak harus selalu sama, dalam artian prosesnya bisa dimodifikasi sesuai dengan kebutuhan. Tapi umumnya proses feature learning ini diawali dengan melakukan proses konvolusi antara matriks input dengan kernel ukuran tertentu kemudian dilanjutkan oleh filter untuk mendapatkan informasi dari gambar sehingga dihasilkan feature map proses akan berlanjut sesuai kedalaman layer model CNN yang ditentukan.

Dengan menggunakan gambar grayscale sebagai nilai input, CNN membatasi arsitektur dengan cara yang lebih sederhana untuk diterapkan. Secara lebih khusus, layer-layer model CNN mempunyai neuron yang saling yang mana tersusun dalam ruang tiga dimensi yaitu: Lebar (W), Tinggi (H), Kedalaman (D). Kedalaman arsitektur CNN pada tiga dimensi volume aktivasi, bukan kedalaman dari jaringan neural penuh, yang dapat merujuk pada jumlah total layer dalam jaringan.

Algoritma CNN memanfaatkan proses konvolusi dengan mengalikan sebuah filter atau kernel konvolusi (filter) berukuran tertentu sebuah inputan berupa gambar. Sesuai dengan arsitektur model pada Gambar 2.2,



Gambar 2.2. Urutan Lapisan CNN (Alam, 2022)

2.3.5. CNN's Layer

Layer-layer yang sangat penting dalam membangun arsitektur Convolutional Neural Network (CNN) sebagai berikut: Convolutional Layers (CONV), ReLU, Pooling Layer (POOL) dan Fully Connected Layer (FC).

2.3.5.1 Convolutional Layer

Convolutional layer merupakan lapisan terpenting dalam lapisan ini. Seperti namanya, lapisan konvolusional ini terdiri dari blok bangunan dasar yang disebut konvolusi. Terapkan konvolusi ke sebagian kecil gambar, buat sampel nilai piksel di area itu, lalu ubah menjadi satu piksel. Menerapkan ke setiap area piksel pada gambar untuk menghasilkan gambar baru yang disebut feature maps.

Idenya adalah bahwa piksel dalam gambar baru menyertakan informasi tentang piksel sekitarnya untuk mencerminkan seberapa baik fitur tersebut diekspresikan di area tersebut. Konvolusi memiliki dua sifat penting yaitu size dan step size. Konvolusi berikut diterapkan pada gambar 3×3 dan langkah 1, yang memiliki ukuran 3×3 karena berjalan pada kisi 3×3 piksel. Ukuran langkahnya adalah 1 karena jarak antar daerah yang diterapkan konvolusi adalah 1. Dengan kata lain, pra-konvolusinya adalah 1 per suku. Ukuran dan ukuran langkah konvolusi menentukan ukuran gambar keluaran. Misalnya, lakukan konvolusi 3×3 dengan

langkah 1 untuk mengubah gambar 5×5 menjadi gambar 3×3 . Pertanyaan yang muncul adalah bagaimana konvolusi mengubah grid piksel menjadi satu piksel (Indolia et al., 2018). pada dasarnya melewati grid piksel melalui perceptron, dengan kata lain, mendapatkan jumlah piksel input yang dibobot. Bagian kanan bawah piksel menunjukkan bobot yang diterapkan ke piksel. Pada konvolusi berikut, tidak ada bias (atau offset), tetapi seperti halnya perceptron, konvolusi biasanya juga memiliki output dari konvolusi yang diterapkan ke input tunggal akan tetap sebagai piksel tunggal, dan output dari konvolusi yang diterapkan ke seluruh gambar 3 dimensi akan menjadi gambar 2 dimensi.

Kita mungkin berpikir bahwa ini berarti gambar kehilangan kedalaman, padahal sebenarnya tidak. Secara teoritis ketika gambar melewati lapisan konvolusional, kedalaman biasanya meningkat. Setiap lapisan konvolusi tidak diterapkan konvolusi ke gambar, namun akan menghasilkan beberapa keluaran 2D untuk setiap konvolusi, yang ditumpuk bersama untuk menghasilkan gambar 3D baru. Oleh karena itu, kedalaman keluaran dari lapisan konvolusional adalah jumlah konvolusi yang diterapkan (Indolia et al., 2018).

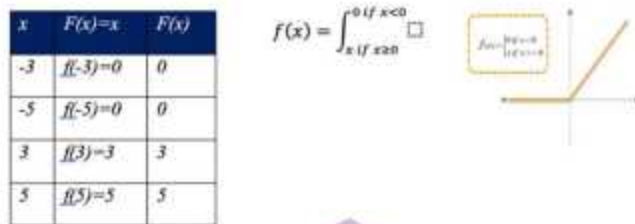


Gambar 2.3. Visualisasi Input dan Filter pada Convolutional Neural Network (CNN) (Alam, 2022).

Dari visualisasi Gambar 2.3 digambarkan di sebelah kiri merupakan input dari convolutional layer dengan ukuran 7×7 . Sedangkan di sebelah kanan merupakan convolutional filter yang juga disebut sebagai kernel. Ini merupakan convolutional 3×3 berdasarkan dari filternya. Setelah melakukan perkalian convolutional layer dengan filter maka akan mendapatkan matriks yang dikenal sebagai *feature map*. Operasi *convolution* dilakukan dengan menggeser filter ini dalam input.

2.3.5.2 *Active Function Rectified Linear Unit (ReLU)*

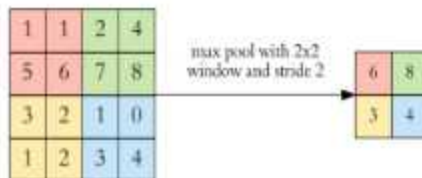
Setelah model melakukan proses konvolusi dalam convolution layer maka layer berikutnya adalah ReLU layer. Pada Gambar 2.4 aktivasi ReLU (Rectified Linear Unit) adalah lapisan aktivasi yang saat ini banyak digunakan untuk model CNN yang mengaplikasikan fungsi $f(x) = \max(0, x)$ yang berarti fungsi ini melakukan thresholding dengan nilai nol terhadap nilai piksel pada input citra. Aktivasi ini membuat seluruh nilai piksel yang bernilai kurang dari nol pada suatu citra akan dijadikan 0 (Indolia et al., 2018). Layer ini meningkatkan sifat non linear dari model dan jaringan keseluruhan tanpa mempengaruhi bidang reseptif dari layer convolutional. Hasil operasi convolution dimasukkan ke dalam fungsi aktivasi ReLU. Jadi nilai-nilai di feature map akhir sebenarnya bukan jumlah, tetapi merupakan penerapan fungsi ReLU (Karpathy, n.d.).



Gambar 2.4. ReLU *Function* (Karpathy, 2018).

2.3.5.3 *Pooling Layer*

Pooling Layer merupakan lapisan yang menggunakan fungsi dengan *Feature Map* sebagai nilai input kemudian mengolahnya dengan berbagai macam operasi statistik berdasarkan nilai piksel terdekat. Pada model CNN, lapisan *Pooling* biasanya disisipkan secara teratur setelah beberapa lapisan konvolusi. Lapisan *Pooling* yang dimasukkan di antara lapisan konvolusi secara berturut turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume output pada *Feature Map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, dan untuk mengurangi *Overfitting*. Lapisan *Pooling* bekerja di setiap tumpukan *Feature Map* dan mengurangi ukurannya. Bentuk lapisan *Pooling* yang paling umum adalah dengan menggunakan filter atau *kernel* berukuran 2x2 yang diaplikasikan dengan langkah sebanyak 2 dan kemudian beroperasi pada setiap irisan dari input. Bentuk seperti ini akan mengurangi *Feature Map* hingga 75% dari ukuran aslinya (Indolia et al., 2018). Contoh operasi *Max Pooling* ditunjukkan dalam Gambar 2.5.



Gambar 2.5. *Max Pooling Sample* (Karpathy, 2018).

2.3.5.4 *Fully Connected Layer*

Dalam model CNN setelah proses convolution layer dan ReLE kemudian dilanjutkan downsampling dengan max pooling dalam lapisan pooling. Fully Connected layer dengan menggunakan vektor fitur dapat mengklasifikasikan gambar input ke dalam beberapa kelas berdasarkan pada dataset pelatihan yang berlabel, fully connected layer berupa vektor angka 1D. Dengan meratakan output dari layer kumpulan terakhir ke vektor dan itu menjadi input ke fully connected layer. Sehingga pada saat meratakan merubah isi angka 3D menjadi sebuah vektor 1D (Alam, 2022).

2.3.5.5 *Softmax Classifier*

Softmax Classifier atau biasa disebut dengan fungsi aktivasi *softmax* merupakan bentuk lain dari algoritma logistic regression yang dapat digunakan untuk klasifikasi lebih dari dua kelas. Standar klasifikasi yang umum dilakukan oleh algoritma Logistic Regression adalah tugas untuk klasifikasi kelas biner (Alam, 2022).

2.3.6. *Hyperparameters*

Hyperparameters adalah pengaturan yang dapat diubah dalam proses pelatihan model machine learning yang tidak dipelajari secara otomatis dari data

selama proses pelatihan. Pengaturan ini memengaruhi bagaimana model dipelajari dan bagaimana model tersebut akan berperfa. Variabel hyperparameters dapat ditentukan sebelum proses pelatihan. *Hyperparameters* berdasarkan keterhubungannya dapat ditentukan menjadi *hyperparameter* yang menentukan model struktur jaringan dan hyperparameters yang sangat berpengaruh terhadap jaringan pelatihan (Putra, dkk. 2023).

2.3.6.1 *Hyperparameters* Training

Tanpa penyesuaian hyperparameter pada proses training seringkali tidak akan menghasilkan model yang optimal, beresiko mengalami overfitting (model terlalu cocok dengan data pelatihan) atau underfitting (model terlalu sederhana untuk menggambarkan data). Dengan melakukan penyesuaian hyperparameter pada proses training dapat mencoba berbagai kombinasi nilai hyperparameter untuk mencari kombinasi terbaik yang menghasilkan kinerja terbaik. Berikut ini adalah hyperparameter yang sangat berpengaruh dalam hal proses training model (Hanin, dkk. 2021).

1. *Epoch* merupakan hyperparameter ketika model dalam proses training di mana menentukan berapa kali model algoritma pembelajaran akan bekerja mengolah seluruh dataset training. Satu epoch berarti bahwa setiap sampel dalam dataset training memiliki kesempatan untuk memperbarui parameter model internalnya.
2. *Batch Size* merupakan sub sampel yang diberikan ke sebuah jaringan setelah pembaruan terjadi. Secara default untuk ukuran dalam proses fitting 32. Ukuran batch yang dapat digunakan 32, 64, dan seterusnya.

3. *Learning Rate* merupakan salah satu parameter proses training berjalan dalam sebuah model. Bekerja dengan mengoreksi nilai koreksi bobot ketika proses training berjalan. Rentang dari parameter learning rate adalah 0 sampai 1. Ketika nilai learning rate kecil maka proses training berjalan lambat tapi memperbaharui informasi dengan cara yang halus. Semakin tinggi nilai learning rate maka proses training berjalan cepat begitu juga dengan sebaliknya.

2.3.6.2 *Loss Function*

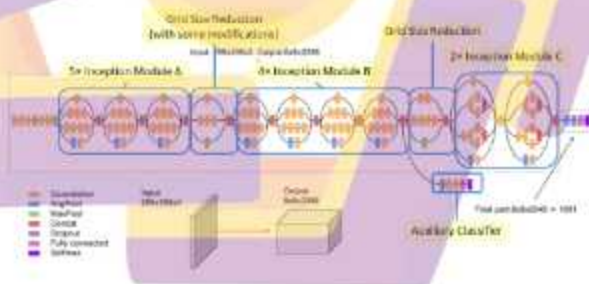
Loss function merupakan lapisan terakhir dalam CNN, menentukan bagaimana pelatihan memberikan penalti atas penyimpangan antara hasil prediksi dan label. Terdapat jumlah variasi *loss function*, diantaranya adalah *softmax loss* yang digunakan untuk memprediksi satu dari sejumlah kelas yang saling eksklusif, *sigmoid cross-entropy* yang digunakan untuk memprediksi sejumlah nilai probabilitas dalam interval $[0, 1]$, dan *euclidean loss* yang digunakan untuk regresi nilai kontinu.

2.3.7. **Pre Trained Model CNN**

Pre-trained model CNN (Convolutional Neural Network) adalah model CNN yang telah dilatih sebelumnya pada tugas pengenalan gambar atau tugas penglihatan komputer lainnya menggunakan dataset gambar yang besar dan beragam. Model CNN ini telah mempelajari fitur-fitur hierarkis dari gambar-gambar dalam dataset pelatihan awalnya dan telah mengembangkan representasi-fitur yang kuat untuk berbagai jenis objek, pola, dan fitur dalam gambar (Singh, et al. 2022).

2.3.7.1 Inception V3

Pada Inception biasanya memiliki tiga ukuran lapisan konvolusi yang berbeda, dengan ukuran lapisan penyatuan menjadi satu. Layer yang digunakan pada model Inception V3 meliputi beberapa lapisan dalam arsitektur ini termasuk lapisan konvolusi yang digunakan untuk membagi massa menjadi perselisihan kecil, dan pergeseran ini adalah operasi untuk mengolah berbagai fitur dari masukan yang diberikan (Dong et al., 2020) *pooling layer* membantu mengurangi dimensi fungsi folder dengan mempercepat komputasi dan mengurangi *overfitting* (Nour et al., 2020), *concat layer* menggabungkan modul sebelumnya dengan yang berikutnya, lapisan yang berfungsi penuh adalah semua neuron dari lapisan modul ke lapisan yang berfungsi penuh, lapisan aktivasi yang digunakan adalah aktivasi *softmax*, yang membantu mengatasi beberapa masalah klasifikasi kelas. Gambar 2.6 merupakan arsitektur dari model Inception-V3.

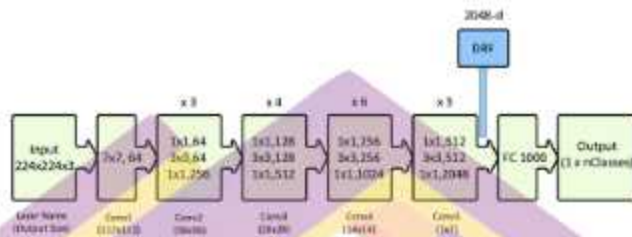


Gambar 2.6. Arsitektur Inception-V3

2.3.7.2 Resnet50

Pada Resnet50, digunakan 5 jenis konvolusi pada tahap pelatihan dengan menormalisasikan gambar menjadi ukuran 224 x 224 pixel [15]. Aktivasi yang digunakan pada *fully-connected layer* adalah fungsi *flatten* dengan mengubah keluaran

multidimensional array dari proses *training* menjadi array satu dimensi untuk kemudian menghasilkan masukan fungsi aktivasi *softmax* yang digunakan untuk menghitung peluang data hasil *training* terhadap objek yang terdiri dari 5 kelas tersebut



Gambar 2.7. Arsitektur Resnet50

2.3.8. Confusion Matrix

Dalam buku Sokolova & Lapalme (2009) *Confusion matrix* adalah salah satu metode yang digunakan untuk mengevaluasi kinerja algoritma klasifikasi. Berikut ini gambaran sederhana untuk mempermudah pemahaman tentang istilah *confusion matrix* dalam keluaran klasifikasi.

Tabel 2.2. Tabel Pengujian menggunakan *Confusion matrix*

		Kelas Prediksi	
		Positif	Negatif
Kelas sesungguhnya	Positif	TP	FN
	Negatif	FP	TN

Nilai *true negatif* (TN) merupakan data yang diklasifikasi dengan tepat sebagai keluaran negatif atau salah. *True positive* (TP) adalah data yang diklasifikasi dengan tepat sebagai keluaran positif atau benar. *False positive* (FP) adalah data yang diklasifikasi dengan kurang tepat apabila keluaran berupa positif atau benar. *False negative* (FN) adalah data yang diklasifikasi dengan kurang tepat.

$$Presisi = \frac{\sum_i^n \frac{TP_i}{TP_i + FP_i}}{n} \quad (1)$$

Persamaan (1) merupakan perhitungan rata-rata nilai *precision* yaitu dari data hasil klasifikasi seberapa banyak data yang benar antara nilai sebenarnya dengan prediksi yang diberikan oleh sistem.

$$Recall = \frac{\sum_i^n \frac{TP_i}{TP_i + FN_i}}{n} \quad (2)$$

Persamaan (2) merupakan perhitungan rata-rata nilai *recall* yaitu dari seluruh data benar seberapa banyak data yang keluar dalam hasil klasifikasi. Evaluasi *recall* digunakan apabila lebih memilih nilai *false positive* daripada *false negative* Ghoneim (2019).

$$Akurasi = \frac{\sum_i^n \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}}{n} \quad (3)$$

Persamaan (3) merupakan perhitungan rata-rata nilai akurasi untuk menunjukkan tingkat efektifitas per kelas dari sebuah klasifikasi. Sokolova & Lapalme (2009).

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian yang digunakan adalah penelitian eksperimental, yaitu menerapkan serangkaian tindakan untuk membuktikan suatu konsep yang diusulkan. Sedangkan sifat penelitian ini adalah penelitian eksperimental komputasi. Penelitian ini melakukan suatu eksperimen untuk klasifikasi kacang pistachio menggunakan metode *Convolutional Neural Network*.

3.2. Metode Pengumpulan Data

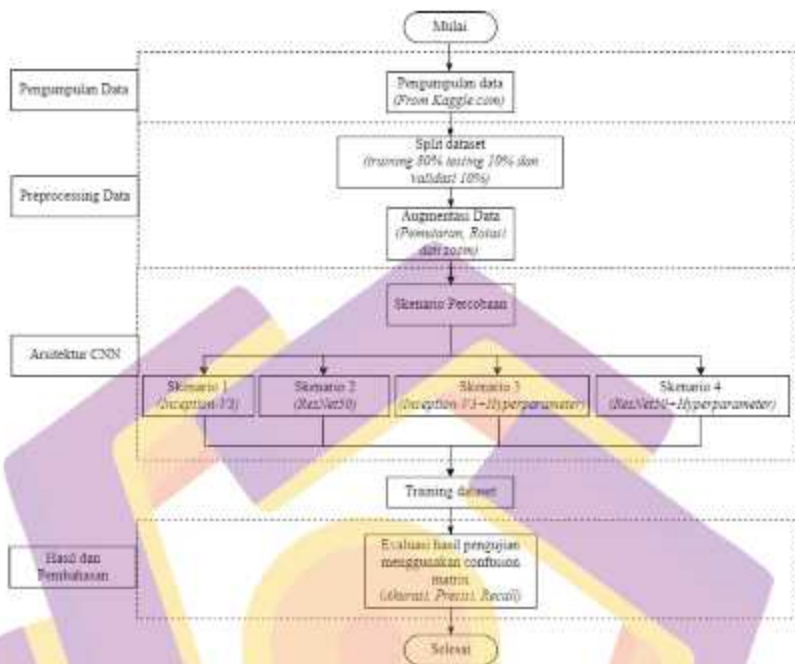
Data yang dibutuhkan untuk penelitian ini berupa gambar pistachio. Metode pengumpulan data dilakukan dengan cara mengambil data gambar pistachio dari *Kaggle.com* (<https://www.kaggle.com/muratkoklu/datasets/>).

3.3. Metode Analisis Data

Metode analisis yang digunakan dalam penelitian ini adalah analisis kuantitatif menggunakan algoritma *Convolutional Neural Network*. Data yang sudah terkumpul menjadi dataset gambar pistachio selanjutnya akan dilakukan *preprocessing* namun, sebelum dilakukan klasifikasi, akan dilakukan beberapa proses, yaitu *preprocessing* data dengan *split* dataset dan *augmentasi* data untuk selanjutnya dilakukan proses klasifikasi dengan menggunakan arsitektur yang akan digunakan.

3.4. Alur Penelitian

Berikut adalah bagan dari alur penelitian:



Gambar 3.1. Alur penelitian

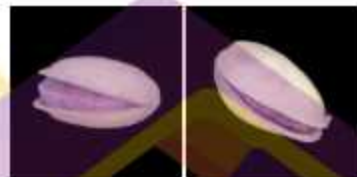
Pada penelitian ini terdiri dari lima tahap utama seperti pada gambar 3.8 dengan rincian alur sebagai berikut:

1. Pengumpulan data

Setelah membaca beberapa referensi yang relevan, tahap selanjutnya adalah proses pengumpulan data berupa gambar pistachio dari sumber *Kaggle.com* (<https://www.muratkoklu.com/datasets/>). Dalam penelitian ini jenis pistachio yang digunakan adalah kirmizi dan siirt. Berikut di bawah ini contoh gambar kirmizi dan siirt.



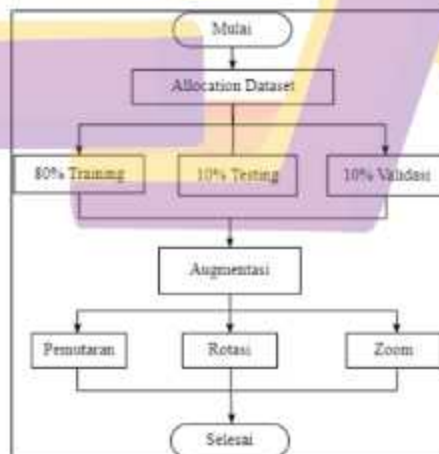
Gambar 3.2. Sampel pistachio Jenis Kirmizi



Gambar 3.3. Sampel pistachio Jenis Siirt

2. *Preprocessing*

Pada tahap ini, dataset yang didapatkan akan disiapkan untuk diproses oleh sistem. Tahapan preprocessing ini mensplit dataset sebanyak 80% untuk data training, 10% untuk data testing dan 10% untuk data validasi, kemudian melakukan augmentasi data.



Gambar 3.4. Tahapan Preprocessing

a. Split dataset

Seperti yang terlihat pada Gambar 3.1 proses split dataset dalam penelitian ini dilakukan dengan membagi dataset berupa data *training* dengan *rasio* 80%, data testing 10% data validasi 10%. Berikut di bawah ini tabel pembagian dataset.

Tabel 3.1. Detail Pembagian Dataset

Jenis Pistachio	Pembagian Dataset		
	Training (80%)	Testing (10%)	Validasi (10%)
Kirmizi	548	184	184
Siirt	548	184	184
Total	1096	368	368

b. Augmentasi

Tahap selanjutnya adalah augmentasi, teknik ini digunakan dalam pengolahan data untuk membuat variasi baru dari data latihan yang ada. Penerapan augmentasi dapat membantu meningkatkan keberagaman data, mengurangi *overfitting*, dan membuat model pembelajaran mesin lebih umum dan mampu menangani data baru yang belum pernah dilihat sebelumnya. Ada 3 jenis augmentasi yang digunakan dalam penelitian ini, diantaranya pemutaran, rotasi, dan zoom.

Pemutaran (*Flipping*) adalah teknik augmentasi di mana gambar atau data lainnya diputar secara horizontal atau vertikal tergantung pada sumbu tertentu. Pemutaran horizontal mengubah gambar dari kiri ke kanan atau sebaliknya, sementara pemutaran vertikal mengubah gambar dari atas ke bawah atau sebaliknya. Pemutaran membantu model

mengenali objek atau pola dari sudut pandang yang berbeda dan meningkatkan keberagaman data.

Rotasi merupakan teknik augmentasi di mana gambar atau data lainnya diputar sekitar titik pusatnya dalam berbagai sudut. Seperti yang telah dijelaskan sebelumnya, ini membantu model mengenali objek atau pola dalam berbagai orientasi, membuatnya lebih tahan terhadap variasi sudut dalam data latihan dan data uji.

Zoom berfungsi untuk memperbesar atau memperkecil untuk membuat variasi dari skala gambar. Dengan zoom in, bagian dari gambar diperbesar sehingga detailnya lebih jelas, sedangkan dengan zoom out, gambar menjadi lebih kecil. Teknik ini membantu model mengenali objek pada berbagai skala dan meningkatkan ketahanannya terhadap variasi ukuran objek pada data uji.

3. Arsitektur CNN

Karena jenis penelitian ini bersifat eksperimen maka alur penelitian ini disediakan langkah skenario, sebelum peneliti memulai tahap skenario ada beberapa tahap sebelumnya yang dilakukan. Salah satu tahapnya adalah menyiapkan data gambar dengan preprocessing sebagai data training dan evaluasi. Untuk dapat mengetahui performa dari peningkatan kualitas gambar, maka dilakukan 4 skenario menggunakan algoritma Convolutional Neural Network yaitu Inception-V3 dan ResNet50.

4. Training data

Setelah menentukan skenario yang ada yaitu transfer learning menggunakan Inception-V3 dan ResNet50, kita mulai melakukan percobaan menggunakan dataset gambar kacang Pistachio yang telah dipersiapkan sebelumnya. Training data bertujuan untuk melatih skenario yang ada sehingga saat tahap evaluasi dapat menghasilkan nilai akurasi yang tinggi.

5. Analisa dan pembahasan

Tahap ini merupakan tahap untuk melakukan evaluasi (penilaian) menggunakan confusion matrix berdasarkan tahap pelatihan menggunakan Inception-V3 dan ResNet50. Ada 4 evaluasi yang dilakukan, yaitu mengetahui performa penerapan arsitektur Inception-V3 dan ResNet 50 dan performa penerapan arsitektur inception v3 dan resnet 50 dengan adjusting hyperparameter, selanjutnya membandingkan akurasi dari arsitektur Inception-V3 dan ResNet 50 pada peningkatan kualitas gambar.









BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah gambar kacang pistachio yang terdiri dari dua jenis yaitu, kacang Kirmizi dan Siirt. Data pistachio tersebut didapatkan dari sumber *kaggle.com* (<https://www.muratkoklu.com/datasets/>). Total data berjumlah 1832, kelas Kirmizi berjumlah 916 dan kelas Siirt berjumlah 916. Adapun ukuran gambar data pistachio memiliki resolusi sebesar 200 x 200 piksel. Berikut tabel di bawah ini menyajikan beberapa sampel gambar dari kacang pistachio sesuai dengan kelasnya.

Tabel 4.1. Contoh sampel Pistachio

Kelas	Gambar Pistachio			
Kirmizi				
Siirt				

Setelah gambar pistachio dikumpulkan, selanjutnya gambar diupload ke dalam *drive* dan ditempatkan di dalam masing masing folder untuk selanjutnya ditampilkan di dalam *google colab* dengan *library os* dan *drive*. Berikut di bawah ini gambar list kode dari *library* yang digunakan untuk menampilkan dataset.

```
from google.colab import drive
import os
drive.mount('/content/drive')
```

Gambar 4.1. *Library* import dataset

4.2. Preprocessing

Setelah data gambar pistachio diimport, tahap selanjutnya adalah tahap *preprocessing* dengan melakukan proses split data terlebih dahulu.

4.2.1 Split Data

Split dataset digunakan untuk membagi dataset menjadi dua atau lebih subset yang berbeda. Dataset dibagi menjadi subset pelatihan (training set), subset pengujian (test set) dan subset validasi (validation set). Pada penelitian ini, peneliti melakukan split dataset yaitu sebanyak 80% untuk data training dan 10% untuk data testing, 10% data validasi. Seperti pada Gambar di bawah ini.

```
print(f'Train: {total_X_train}')
print(f'Validation: {total_X_val}')
print(f'Test: {total_X_test}')

Train: 1099
Validation: 366
Test: 367
```

Gambar 4.2. Split dataset

4.2.2 Augmentasi

Setelah gambar dimuat, berbagai transformasi yang akan diterapkan pada gambar. Beberapa transformasi yang digunakan diantaranya pemutaran atau (*flips*), rotasi (*rotations*), dan memperbesar dan memperkecil gambar secara proporsional (*zoom*).



Gambar 4.3. Sampel Kirmizi dan Siirt

1. Pemutaran (*Flips*)

```
datagen = default_datagen()
datagen.horizontal_flip = True
datagen.vertical_flip = True
plot_augmentation(datagen, data)
```

Gambar 4.4. List kode proses Augmentasi (*flips*)

Terlihat pada Gambar 4.4 untuk proses pemutaran gambar. Pertama, *default_datagen()* digunakan untuk membuat objek *datagen* dengan pengaturan *default*. *Datagen* ini akan digunakan untuk menghasilkan data augmented.

Selanjutnya, *datagen.horizontal_flip = True* digunakan untuk mengaktifkan augmentasi *flip* horizontal. Ini berarti setiap gambar dalam data akan diputar secara horizontal dengan membuat versi *flipped* dari gambar-gambar tersebut. Kemudian, *datagen.vertical_flip = True* digunakan untuk mengaktifkan augmentasi *flip* vertikal. Dengan mengatur nilai ini menjadi *true*, setiap gambar dalam data akan diputar secara vertikal dengan membuat versi *flipped* dari gambar-gambar tersebut. Terakhir, *plot_augmentation(datagen, data)* digunakan untuk memvisualisasikan hasil augmentasi data. Fungsi *plot augmentation* akan menerima objek *datagen* dan data asli (*data*), dan menghasilkan plot yang menunjukkan perbedaan antara gambar asli dan versi augmented dari gambar tersebut. Seperti pada Gambar 4.5 hasil dari pemutaran gambar.



Gambar 4.5. Hasil Augmentasi (*flips*)

2. *Rotations*

```
datagen = default_datagen()
datagen.rotation_range = 15
plot_augmentation(datagen, data)
```

Gambar 4.6. List kode proses *rotations*

Pada proses rotasi digunakan `datagen.rotation_range=15`, mengatur rentang rotasi yang akan diterapkan pada data saat dilakukan augmentasi. Dalam hal ini, rentang rotasi adalah 15 derajat. Artinya, setiap gambar dalam data dapat dirotasi secara acak antara -15 derajat hingga +15 derajat. Setelah mengatur parameter augmentasi, kode tersebut kemungkinan besar akan memanggil fungsi `plot_augmentation()` untuk memvisualisasikan perubahan yang dihasilkan oleh augmentasi pada data, seperti pada Gambar 4.7.

Gambar 4.7. Hasil Augmentasi (*rotations*)

3. *Zoom*

```
datagen = default_datagen()
datagen.zoom_range = [0.15, 1.5]
plot_augmentation(datagen, data)
```

Gambar 4.8. List kode proses zoom

Pada proses augmentasi selanjutnya adalah zoom, fungsi yang digunakan untuk menginisialisasi objek *datagen* dengan pengaturan default yang sesuai untuk augmentasi data adalah *datagen.zoom_range*. Pada baris ini, rentang zoom untuk augmentasi data diatur pada objek *datagen*. Rentang zoom didefinisikan dalam bentuk daftar (*min_zoom, max_zoom*). Rentang zoom diatur dari 0.15 hingga 1.5, yang berarti gambar-gambar akan di-zoom-in atau di-zoom-out dalam rentang tersebut selama proses augmentasi.

Gambar 4.9. Hasil proses augmentasi (*zoom*)

4.3. Proses Klasifikasi Menggunakan Convolutional Neural Network

Penelitian ini menggunakan percobaan terhadap 2 arsitektur yang telah teruji tingkat kinerja dari masing-masing arsitektur.

4.3.1 Inception-V3

Ukuran input shape dasar untuk arsitektur inception V3 adalah 299x299x3, namun dalam penelitian ini menggunakan 224x224x3, include top = true, weights = none, input tensor = none, pooling= none, kelas= 2 dan klasifikasi kelas menggunakan sigmoid karena data yang digunakan hanya berjumlah dua kelas.

```

model = tf.keras.applications.InceptionV3(
    input_shape= [224, 224, 3],
    include_top=False,
    weights=None,
    input_tensor=None,
    pooling=None,
    classes=2,
    classifier_activation="sigmoid",
)

```

Gambar 4.10. List kode Fungsi Arsitektur Inception-V3

4.3.2 ResNet50

Setelah tahap preprocessing dan data telah dibagi menjadi data training dan data testing, maka gambar memasuki proses pelatihan model pada arsitektur ResNet-50. Pada tahap pertama, gambar berukuran 224 x 224 piksel yang berasal dari layer input di konvolusi pada lapisan konvolusi dengan ukuran filter 7×7 dan stride 2. Konvolusi tersebut menghasilkan feature map yang kemudian dinormalisasi oleh Batch Normalization. Hasil dari normalisasi selanjutnya masuk ke layer aktivasi, di dalamnya terdapat fungsi ReLU yang digunakan untuk membuat hasil dari ekstraksi fitur menjadi non-linear. Selanjutnya nilai output dari

fungsi ReLU direduksi pada lapisan max pooling sebelum diteruskan pada tahap konvolusi kedua. Di antara tahap konvolusi kedua sampai dengan kelima, proses ekstraksi fitur dilakukan oleh kombinasi convolutional block dan identity block. Setelah proses ekstraksi fitur selesai, kemudian feature map diproses ke dalam fully connected layer untuk melakukan proses prediksi dibantu dengan fungsi aktivasi sigmoid.

```
resnet50 = tf.keras.applications.ResNet50(  
    include_top=True,  
    weights=None,  
    input_tensor=None,  
    input_shape=[224, 224, 3],  
    pooling=None,  
    classes=2,  
    classifier_activation="sigmoid",  
)
```

Gambar 4.11. List kode Fungsi Arsitektur ResNet50

Seperti yang terlihat pada Gambar 4.11 dalam penelitian ini menggunakan input shape = [224, 224, 3], include top = true, input tensor = None, pooling = None, weights = None, aktivasi kelas menggunakan sigmoid, dan kelas = 2, karena data yang digunakan penulis dalam penelitian ini hanya berjumlah 2 kelas.

4.4. Skenario Percobaan

Untuk menghasilkan hasil yang diharapkan, diperlukan beberapa skenario percobaan yang akan diterapkan. Dalam penelitian ini penulis menentukan 4 skenario percobaan yang dapat dilihat pada Tabel 4.2 berikut.

Tabel 4.2 Skenario percobaan

No	Skenario	Arsitektur
1	Skenario 1	InceptionV3
2	Skenario 2	ResNet50
3	Skenario 3	InceptionV3 + Hyperparameter (Epoch 50/100) (Batch Size 32/64) (Lr 0.1/0.01/0.001/0.0001/0.00001)
4	Skenario 4	ResNet50 + Hyperparameter (Epoch 50/100) (Batch Size 32/64) (Lr 0.1/0.01/0.001/0.0001/0.00001)

Empat skenario yang ditunjukkan pada Tabel 4.2 akan diterapkan terhadap kedua arsitektur yang telah penulis sebutkan sebelumnya. Pada skenario percobaan ketiga dan keempat diterapkan *adjusting* hyperparameter yaitu epoch, batch size dan learning rate, hal ini ditinjau pada penelitian sebelumnya yang dilakukan oleh (Singh Dilbagh et al., 2022), dalam penelitian sebelumnya hanya melakukan sekali pelatihan terhadap data yang cukup banyak, sehingga terjadi stagnasi pada hasil akurasi (tidak mengalami penurunan dan juga tidak mengalami peningkatan) dan tidak ada percobaan dalam mencari kombinasi nilai hyperparameters terbaik pada proses training yang digunakan.

Pada skenario pertama dan kedua ditetapkan penggunaan epoch 100, batch size 32 dan learning rate 0.001 dengan merujuk ke penelitian sebelumnya terkait hyperparameter pada proses training yang dilakukan oleh (Hanin, dkk. 2021), dengan penggunaan epoch 100, batch size 32 dan learning rate 0.001, mendapatkan nilai akurasi yang cukup baik. Setiap skenario percobaan di atas akan diujikan dalam pembelajaran data dengan rasio pembagian data latih 80%, 10% data uji dan

10% untuk data validasi. Dalam penelitian ini penulis ingin mengetahui nilai akurasi tertinggi dari arsitektur Inception-V3 dan ResNet.

4.4.1 Skenario Percobaan Pertama

Setelah tahap preprocessing selesai dan data dibagi dengan rasio perbandingan 80% data latih, 10% data uji dan 10%, maka selanjutnya dilakukan proses training data dengan menggunakan arsitektur Inception-V3.

Tabel 4.3. Percobaan skenario 1

Skenario 1	Epoch	Batch Size	Learning Rate	Nilai Akurasi	Nilai Validasi	Nilai Loss
Inception-V3	100	32	0.001	97.00%	96.45%	12.55%

Seperti yang terlihat pada Tabel 4.3, percobaan skenario 1 menggunakan arsitektur Inception V3 dengan epoch 100, batch 32 dan learning rate 0.001. Mendapatkan nilai akurasi sebesar 97.00% dan nilai validasi sebesar 96.45%.

4.4.2 Skenario Percobaan Kedua

Pada skenario kedua menggunakan ResNet 50, dengan penggunaan epoch 100, batch size 32 dan learning rate 0.001 seperti pada skenario pertama.

Tabel 4.4. Percobaan skenario 2

Skenario 2	Epoch	Batch Size	Learning Rate	Nilai Akurasi	Nilai Validasi	Nilai Loss
ResNet50	100	32	0.001	85.56%	84.43%	44.51%

Seperti yang terlihat pada Tabel 4.4, pada percobaan skenario kedua mendapatkan nilai akurasi sebesar 85.56% dan nilai validasi sebesar 84.43%.

4.4.3 Skenario Percobaan Ketiga

Pada skenario percobaan ketiga menggunakan arsitektur inception v3 ditambah penyetelan hyperparameter pada proses pelatihan, seperti kode yang terlihat pada Gambar 4.12.

```
# Define hyperparameter
batch_size = 32/64
epochs = 50/100
optimizer = Adam(learning_rate=0.1
                 0.01
                 0.001
                 0.0001
                 0.00001)
```

Gambar 4.12. Kode penambahan hyperparameter pada skenario percobaan

Tabel 4.5. Percobaan skenario 3

Skenario 3	Epoch	Batch Size	Learning Rate	Nilai Akurasi	Nilai Validasi	Nilai Loss
InceptionV3 + Hyperparameter	Epoch 50	Batch Size 32	Lr 0.1	50.41%	48.09%	69.39%
			Lr 0.01	96.46%	96.45%	10.60%
			Lr 0.001	96.73%	96.45%	10.39%
			Lr 0.0001	95.91%	96.45%	10.34%
			Lr 0.00001	96.19%	95.63%	10.39%
		Batch Size 64	Lr 0.1	50.40%	47.81%	69.32%
			Lr 0.01	96.19%	95.90%	10.39%
			Lr 0.001	96.19%	96.72%	10.39%
			Lr 0.0001	96.19%	96.17%	10.395
			Lr 0.00001	97.00%	95.63%	10.04%
	Epoch 100	Batch Size 32	Lr 0.1	49.59%	52.19%	67.81%
			Lr 0.01	49.59%	52.19%	67.81%
			Lr 0.001	98.36%	98.20%	8.01%
			Lr 0.0001	95.64%	97.27%	12.54%
			Lr 0.00001	97.00%	96.17%	9.70%
		Batch Size 64	Lr 0.1	49.59%	52.19%	68.64%
			Lr 0.01	95.64%	96.72%	12.21%
			Lr 0.001	97.28%	97.54%	9.51%
			Lr 0.0001	97.55%	97.27%	8.21%
			Lr 0.00001	97.28%	96.72%	9.51%

Jumlah epoch yang digunakan dalam skenario ketiga adalah epoch 50 dan 100, batch size 32 dan 64 dengan learning rate dari rentan nilai maksimum 0.1 hingga nilai minimum 0.00001, 10 kali pelatihan menggunakan epoch 50/100, dan 10 kali menggunakan batch size 32/64. Pada skenario ketiga memperoleh nilai akurasi tertinggi menggunakan epoch 100, batch size 32 dengan learning rate 0.001 yaitu nilai akurasi sebesar 98,36%, dan nilai validasi sebesar 98,20%. Jumlah total proses pelatihan secara keseluruhan dapat dilihat pada Tabel 4.5.

4.4.4 Skenario Percobaan Keempat

Tabel 4.6. Percobaan skenario 4

Skenario 4	Epoch	Batch Size	Learning Rate	Nilai Akurasi	Nilai Validasi	Nilai Loss
ResNet50 Parameter	Epoch 50	Batch Size 32	Lr 0.1	50.41%	47.81%	69.32%
			Lr 0.01	49.59%	52.19%	69.35%
			Lr 0.001	85.56%	83.88%	50.46%
			Lr 0.0001	84.20%	81.97%	40.99%
			Lr 0.00001	83.65%	81.69%	40.22%
		Batch Size 64	Lr 0.1	49.59%	52.19%	69.35%
			Lr 0.01	50.41%	47.81%	69.32%
			Lr 0.001	75.75%	76.23%	44.23%
			Lr 0.0001	89.10%	84.97%	35.71%
			Lr 0.00001	86.92%	82.51%	35.24%
	Epoch 100	Batch Size 32	Lr 0.1	49.59%	52.19%	69.33%
			Lr 0.01	50.41%	47.81%	69.32%
			Lr 0.001	50.41%	47.81%	37.74%
			Lr 0.0001	87.74%	82.24%	40.32%
			Lr 0.00001	85.01%	82.51%	37.74%
		Batch Size 64	Lr 0.1	49.59%	52.19%	69.32%
			Lr 0.01	88.01%	84.70%	37.46%
			Lr 0.001	89.65%	85.52%	36.75%
			Lr 0.0001	80.38%	78.96%	44.95%
			Lr 0.00001	49.59%	52.19%	69.51%

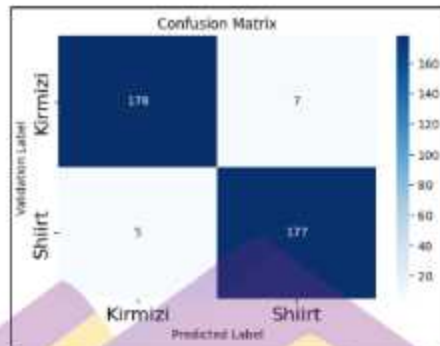
Proses yang diterapkan pada skenario keempat yaitu menggunakan arsitektur ResNet 50 dengan penambahan hyperparameter, sama seperti pada proses skenario ketiga, menggunakan epoch 50 dan 100, batch size 32 dan 64 serta penggunaan learning rate dari nilai maksimum 0.1 hingga nilai minimum 0.00001. Seperti yang terlihat pada Tabel 4.6 total proses pelatihan sebanyak 20 kali, 10 kali menggunakan epoch 50/100 dan 10 kali menggunakan batch size 32/64. Pada skenario keempat mendapatkan nilai tertinggi dari epoch 100, batch size 64 dan learning rate 0.001 sebesar 89,65% untuk nilai akurasi dan 85,52% untuk nilai validasi.

4.5. Proses Pengujian

Penelitian ini menggunakan *confusion matrix* sebagai teknik atau metode dalam menguji kinerja dari model yang telah dibuat. Dalam metode *confusion matrix* ini dapat digunakan dalam mencari nilai akurasi, presisi, recall dari model klasifikasi yang telah dibuat.

4.5.1 Pengujian Skenario Pertama

Pada pengujian skenario pertama menggunakan arsitektur inception v3. Hasil pengujian skenario pertama disajikan pada Gambar 4.13. Dalam menentukan nilai akurasi, presisi dan recall sebagai berikut.



Gambar 4.13. Hasil pengujian InceptionV3

1) Akurasi

Untuk menentukan nilai akurasi model melalui gambar tabel *confusion matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih. Seperti pada gambar 4.13 jumlah nilai TP adalah $178 + 177 = 355$ dan jumlah data latih yang digunakan adalah 367. Sehingga nilai akurasinya adalah $355/367 = 0.97$ atau 97%.

2) Presisi

Rumus yang digunakan dalam menentukan nilai presisi adalah $TP/(TP+FP)$. Pada kelas banyak diperlukan perhitungan presisi terhadap masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas. Mengacu pada gambar 4.13 adalah nilai posisinya:

a. Nilai presisi kelas Kirmizi

$$P(\text{Kirmizi}) = 178/(178+5) \Rightarrow 178/183 \Rightarrow 0.97$$

b. Nilai presisi kelas Siirt

$$P(\text{Siirt}) = 177/(177+7) \Rightarrow 177/184 \Rightarrow 0.96.$$

Sehingga nilai presisi untuk skenario keempat adalah $(0.97 + 0.96) / 2$
 $\Rightarrow 0.97$.

3) Recall

Rumus yang digunakan dalam menentukan nilai recall adalah $TP/(TP+FN)$. Pada penentuan recall ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan recall terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

a. Nilai recall kelas Kirmizi

$$R(\text{Kirmizi}) = 178/(178+7) \Rightarrow 178/185 \Rightarrow 0.96$$

b. Nilai recall kelas Siirt

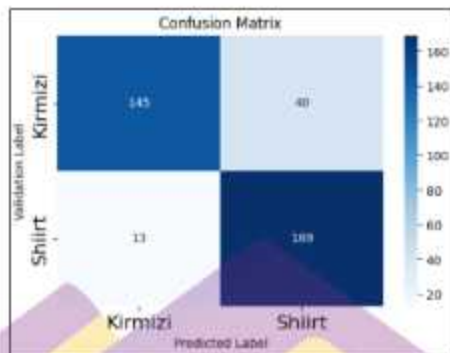
$$R(\text{Siirt}) = 177/(177+5) \Rightarrow 177/182 \Rightarrow 0.97$$

Sehingga nilai recall yang diperoleh untuk skenario kedua adalah $(0.96 + 0.97) / 2 \Rightarrow 0.97$.

4.5.2 Pengujian Skenario Kedua

Pada pengujian skenario kedua menggunakan arsitektur Resnet 50. Hasil pengujian skenario kedua menggunakan arsitektur resnet 50 disajikan pada Gambar

4.14. Dalam menentukan nilai akurasi, presisi dan recall sebagai berikut.



Gambar 4.14. Hasil pengujian ResNet50

1) Akurasi

Untuk menentukan nilai akurasi model melalui gambar tabel *confusion matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih. Seperti pada gambar 4.14 jumlah nilai TP adalah $145 + 169 = 314$ dan jumlah data latih yang digunakan adalah 367. Sehingga nilai akurasinya adalah $314/367 = 0.8555$ atau 85.56%.

2) Presisi

Rumus yang digunakan dalam menentukan nilai presisi adalah $TP/(TP+FP)$. Pada kelas banyak diperlukan perhitungan presisi terhadap masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas. Mengacu pada gambar 4.14 adalah nilai presisinya:

a. Nilai presisi kelas Kirmizi

$$P(\text{Kirmizi}) = 145/(145+13) \Rightarrow 145/158 \Rightarrow 0.92$$

b. Nilai presisi kelas Shiirt

$$P(\text{Shiirt}) = 169/(169+40) \Rightarrow 169/209 \Rightarrow 0.81$$

Sehingga nilai presisi untuk skenario kedua adalah $(0.92 + 0.81) / 2 \Rightarrow 0.86$.

3) Recall

Rumus yang digunakan dalam menentukan nilai recall adalah $TP/(TP+FN)$. Pada penentuan recall ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan recall terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

a. Nilai recall kelas Kirmizi

$$R(\text{Kirmizi}) = 145/(145+40) \Rightarrow 145/185 \Rightarrow 0.78$$

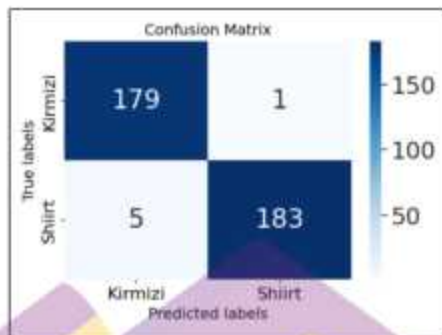
b. Nilai recall kelas Siirt

$$R(\text{Siirt}) = 169/(169+13) \Rightarrow 169/182 \Rightarrow 0.93$$

Sehingga nilai recall yang diperoleh untuk skenario kedua adalah $(0.78 + 0.93) / 2 \Rightarrow 0.86$.

4.5.3 Pengujian Skenario Ketiga

Pada pengujian skenario ketiga terdapat 20 kali proses pengujian namun hanya menampilkan hasil pengujian dari nilai presisi, recall dan akurasi tertinggi. Hasil pengujian skenario ketiga menggunakan arsitektur Inception-V3 ditambah hyperparameter disajikan pada Gambar 4.15. Dalam menentukan nilai akurasi, presisi dan recall sebagai berikut.



Gambar 4.15. Hasil pengujian InceptionV3+hyperparameter

Berikut hasil pengujian arsitektur Inception-V3 dengan penambahan hyperparameter epoch, batch size dan learning rate yang disajikan pada Gambar 4.15. Dalam menentukan nilai akurasi, presisi dan recall sebagai berikut.

1) Akurasi

Untuk menentukan nilai akurasi model melalui gambar tabel *confusion matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih. Seperti pada gambar 4.15 jumlah nilai TP adalah $179 + 183 = 362$ dan jumlah data latih yang digunakan adalah 368. Sehingga nilai akurasinya adalah $362/368 = 0.9836$ atau 98,36%.

2) Presisi

Rumus yang digunakan dalam menentukan nilai presisi adalah $TP/(TP+FP)$. Pada kelas banyak diperlukan perhitungan presisi terhadap masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas. Mengacu pada gambar 4.15 adalah nilai presisinya:

- a. Nilai presisi kelas Kirmizi

$$P(\text{Kirmizi}) = 179/(179+5) \Rightarrow 179/184 \Rightarrow 0.97$$

b. Nilai presisi kelas Siirt

$$P(\text{Siirt}) = 183/(183+1) \Rightarrow 183/184 \Rightarrow 0.99$$

Sehingga nilai presisi untuk skenario ketiga adalah $(0.97 + 0.99) / 2 \Rightarrow 0.98$.

3) Recall

Rumus yang digunakan dalam menentukan nilai recall adalah $TP/(TP+FN)$. Pada penentuan recall ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan recall terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

a. Nilai recall kelas Kirmizi

$$R(\text{Kirmizi}) = 179/(179+1) \Rightarrow 179/180 \Rightarrow 0.99$$

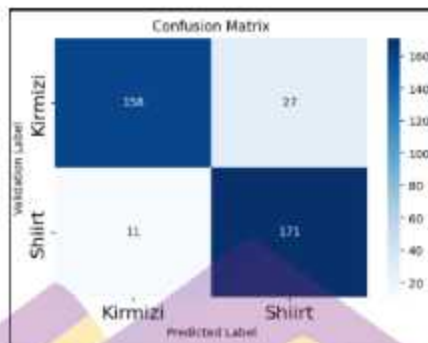
b. Nilai recall kelas Siirt

$$R(\text{Siirt}) = 183/(183+5) \Rightarrow 183/188 \Rightarrow 0.97$$

Sehingga nilai recall yang diperoleh untuk skenario ketiga adalah $(0.99 + 0.97) / 2 \Rightarrow 0.98$.

4.5.4 Pengujian Skenario Keempat

Pada pengujian skenario keempat juga terdapat 20 kali proses pengujian, namun yang akan ditampilkan adalah hasil pengujian yang mendapatkan nilai akurasi, presisi dan recall tertinggi. Berikut hasil pengujian ResNet50 dengan penyetelan hyperparameter dalam menentukan nilai akurasi, presisi, dan recall yang disajikan pada Gambar 4.16.



Gambar 4.16. Hasil pengujian ResNet50+hyperparameter

Seperti halnya pada skenario keempat, terjadi 20 kali proses pengujian namun pengujian ditampilkan adalah yang memperoleh nilai tertinggi. Seperti akurasi, presisi dan recall sebagai berikut.

1) Akurasi

Untuk menentukan nilai akurasi model melalui gambar tabel *confusion matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih.

Seperti pada gambar 4.16 jumlah nilai TP adalah $158 + 171 = 329$ dan jumlah data latih yang digunakan adalah 367. Sehingga nilai akurasinya adalah $329/367 = 0.8965$ atau 89,65%.

2) Presisi

Rumus yang digunakan dalam menentukan nilai presisi adalah $TP/(TP+FP)$. Pada kelas banyak diperlukan perhitungan presisi terhadap masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas. Mengacu pada gambar 4.16 adalah nilai presisinya:

a. Nilai presisi kelas Kirmizi

$$P(\text{Kirmizi}) = 158/(158+11) \Rightarrow 158/169 \Rightarrow 0.93$$

b. Nilai presisi kelas Siirt

$$P(\text{Siirt}) = 171/(171+27) \Rightarrow 171/198 \Rightarrow 0.86$$

Sehingga nilai presisi untuk skenario keempat adalah $(0.93 + 0.86) / 2 \Rightarrow 0.90$.

3) Recall

Rumus yang digunakan dalam menentukan nilai recall adalah $TP/(TP+FN)$. Pada penentuan recall ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan recall terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

a. Nilai recall kelas Kirmizi

$$R(\text{Kirmizi}) = 158/(158+27) \Rightarrow 158/185 \Rightarrow 0.85$$

b. Nilai recall kelas Siirt

$$R(\text{Siirt}) = 171/(171+11) \Rightarrow 171/182 \Rightarrow 0.94$$

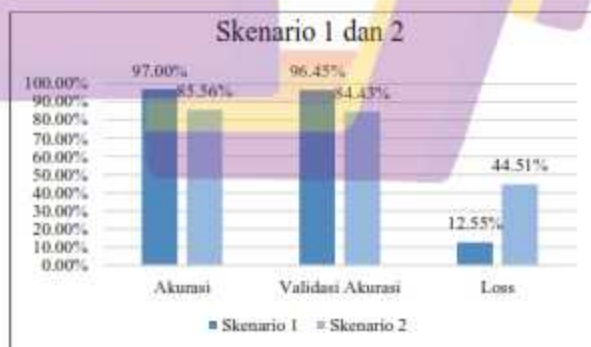
Sehingga nilai recall yang diperoleh untuk skenario keempat adalah $(0.85 + 0.94) / 2 \Rightarrow 0.90$.

Berdasarkan tabel confusion matrix, hasil pengujian yang dilakukan terhadap keseluruhan skenario, masih mengalami error detection, dimana pada skenario 1 jumlah kirmizi yang terdeteksi salah sebanyak 5 sedangkan jumlah siirt terdeteksi salah sebanyak 7. Pada pengujian skenario 2, jumlah siirt yang terdeteksi sebanyak 40 dan jumlah kirmizi terdeteksi salah sebanyak 13. Selanjutnya, untuk pengujian skenario 3 jumlah siirt terdeteksi salah sebanyak 1 dan jumlah kirmizi

terdeteksi salah sebanyak 5. Dan untuk pengujian yang terakhir yaitu pengujian skenario 4, jumlah kirmizi yang terdeteksi salah sebanyak 11 dan jumlah siirt terdeteksi salah sebanyak 27. Jika ditinjau kembali tinjauan literatur, kacang pistachio jenis kirmizi dan siirt memiliki kesamaan mulai dari warna, cangkang, biji serta batang. Hal ini menjadi faktor dalam pemrosesan data seperti data mengalami ambiguitas, dimana beberapa objek gambar sulit untuk dibedakan secara visual, sehingga model mengalami kesulitan untuk mengklasifikasikan gambar dengan benar. Dan juga faktor kebisingan dalam memproses data, ini dapat berasal dari berbagai sumber seperti pencahayaan yang buruk pada gambar, artefak dalam gambar serta kualitas gambar yang rendah dan kurang jelas.

4.6. Evaluasi Hasil Penelitian

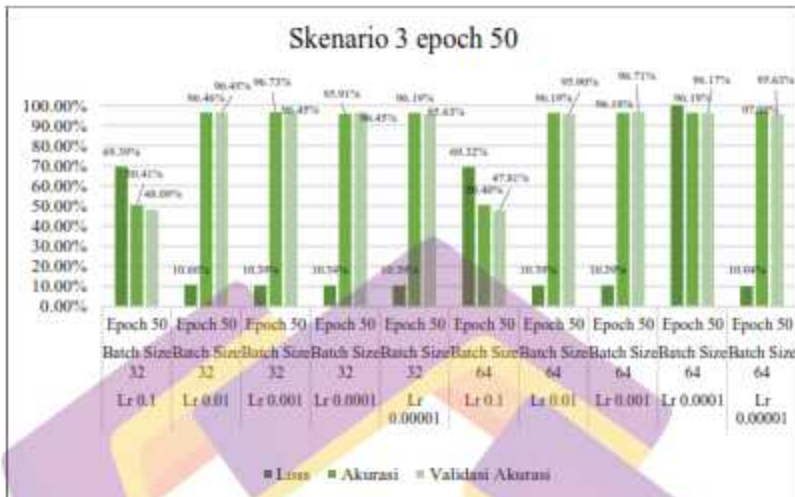
Setelah dilakukan pengolahan data (*preprocessing*) dan proses pembelajaran model terhadap skenario percobaan, diperoleh beberapa model klasifikasi sebanyak 4 skenario yang digunakan. Hasil percobaan keseluruhan skenario disajikan dalam grafik persentase, berikut di bawah ini.



Gambar 4.17. Grafik persentase hasil percobaan skenario 1 dan 2

Dapat dilihat pada grafik persentase hasil percobaan skenario 1 dan 2 pada Gambar 4.17 Dimana skenario 1 menggunakan Resnet 50 dan Skenario 2 menggunakan Inception V3. Penggabungan hasil percobaan skenario 1 dan 2 dalam grafik persentase dikarenakan pada percobaan skenario 1 dan 2 menggunakan epoch, batch size serta nilai lr yang sama, yaitu epoch 100, batch size 32 dan lr 0.001. Pada skenario 1 menggunakan inception v3, memperoleh nilai akurasi sebesar 97.00%, sedangkan pada skenario 2 menggunakan resnet 50, mengalami penurunan hasil akurasi dan validasi akurasi sebesar 85.56% dan 84.43%, namun pada nilai loss, skenario 2 mengalami peningkatan dari skenario 1 yaitu sebesar 44.51%, sedangkan pada skenario 1 memperoleh nilai loss jauh lebih kecil daripada skenario 2, yaitu sebesar 12.55%.

Pada skenario percobaan 3 menggunakan inception v3 dengan adjusting hyperparameters pada proses training, yaitu epoch 100/50, batch 32/64, dan lr dengan rentan nilai dari maksimum 0.1 sampai dengan nilai minimum 0.00001. terdapat 20 kali proses percobaan dimana 10 kali menggunakan epoch 100/50 dan 10 kali menggunakan batch size 32/64, sehingga dalam hal ini penyajian hasil persentase percobaan pada skenario 3 disajikan dalam 2 grafik yang berbeda.



Gambar 4.18. Grafik Persentase hasil percobaan skenario 3 epoch 50

Dapat dilihat pada Gambar 4.18. Grafik persentase skenario percobaan 3 menggunakan inception v3 + hyperparameter epoch 50 memperoleh nilai akurasi tertinggi pada penggunaan batch size 64 dan lr 0.00001 yaitu sebesar 97.00% dengan nilai validasi sebesar 95.63% dan nilai loss sebesar 10.04%, sedangkan nilai akurasi terendah diperoleh pada penggunaan batch size 64 lr 0.1 sebesar 50.40%, nilai validasi sebesar 47.81% dan memperoleh nilai loss tertinggi sebesar 69.32%.

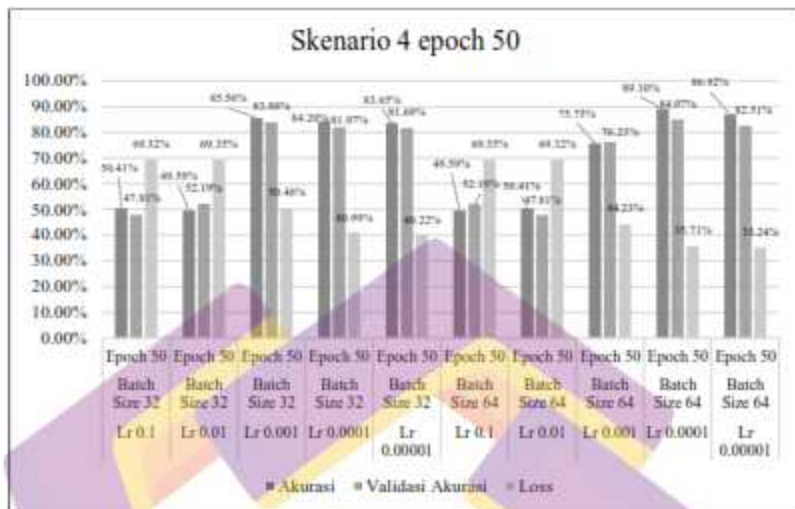
Pada skenario 3 penggunaan epoch 100 seperti yang terlihat pada Gambar 4.19 memperoleh nilai akurasi tertinggi pada batch size 32 dan lr 0.001 dengan akurasi sebesar 98.36%, nilai validasi sebesar 98.20% dan memperoleh nilai loss paling rendah sebesar 5.01%, sedangkan hasil akurasi terendah diperoleh pada penggunaan batch size 32 lr 0.1, 0.01, dan batch size 64, lr 0.1 masing masing memperoleh nilai akurasi, validasi akurasi serta nilai loss yang sama yaitu sebesar 49.59%, nilai validasi sebesar 52.19% dan nilai loss sebesar 67.81%.



Gambar 4.19. Grafik persentase hasil percobaan skenario 3 epoch 100

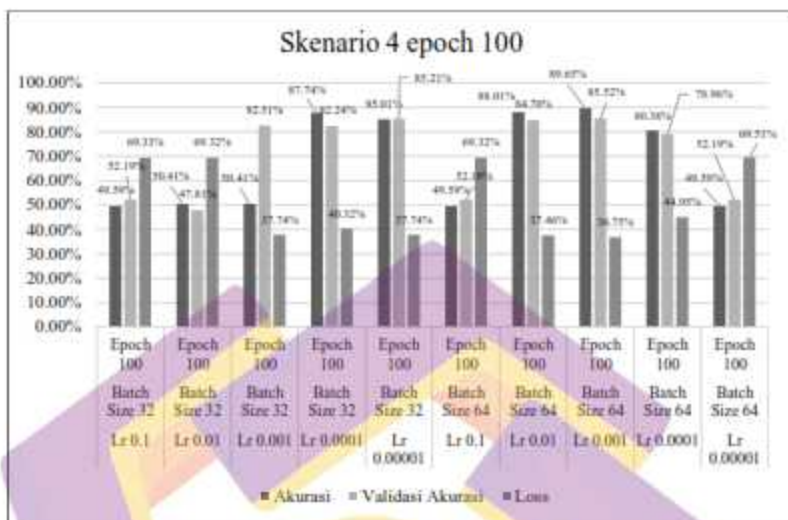
Jika dilihat hasil keseluruhan percobaan pada skenario 3 menggunakan inception v3 ditambah hyperparameter, nilai tertinggi diperoleh dengan penggunaan epoch 100 batch size 32 dan lr 0.001.

Pada skenario percobaan 4 menggunakan ResNet 50 juga diterapkan adjusting hyperparameters pada proses training, yaitu epoch 100/50, batch 32/64, dan lr dengan rentan nilai dari maksimum 0.1 sampai dengan nilai minimum 0.00001. terdapat 20 kali proses percobaan dimana 10 kali menggunakan epoch 100/50 dan 10 kali menggunakan batch size 32/64, sehingga dalam hal ini penyajian hasil persentase percobaan pada skenario 4 disajikan dalam 2 grafik yang berbeda.



Gambar 4.20. Grafik persentase hasil percobaan skenario 4 epoch 50

Dapat dilihat pada Gambar 4.20, Grafik persentase hasil percobaan skenario 4 epoch 50 mengalami nilai baik akurasi, validasi serta nilai loss yang naik turun dimana nilai tertinggi diperoleh pada penggunaan batch size 64 lr 0.00001 yaitu akurasi sebesar 86.92%, nilai validasi sebesar 82.51%, dan nilai loss sebesar 35.24% sedangkan nilai terendah diperoleh dengan penggunaan batch size 32 lr 0.01, dan batch size 64 lr 0.1 memperoleh nilai akurasi yang sama yaitu sebesar 49.69%, nilai validasi sebesar 52.19% dan nilai loss sebesar 69.35%.



Gambar 4.21. Grafik persentase hasil percobaan skenario 4 epoch 100

Pada Gambar 4.21 merupakan grafik hasil persentase nilai akurasi, validasi akurasi dan loss pada skenario 4 epoch 100, terlihat pada gambar grafik tersebut mengalami nilai yang naik turun seperti halnya pada skenario 4 epoch 50 sebelumnya, nilai loss terendah diperoleh pada penggunaan batch size 64 lr 0.001, dan nilai loss meningkat pada penggunaan batch size 64 dan lr 0.0001, sedangkan nilai akurasi tertinggi masih diperoleh pada batch size 64 lr 0.001, begitupun nilai validasi diperoleh sebesar 85.52%, namun mengalami penurunan hasil akurasi dan validasi pada penggunaan batch 32/lr 0.1, batch size 64/lr 0.1 dan batch size 64/lr 0.00001 mendapatkan hasil yang sama yaitu 49.59% untuk nilai akurasi dan 52.19% untuk nilai validasi akurasi. Jika dilihat dari grafik persentase skenario epoch 50 dan 100, maka nilai tertinggi diperoleh epoch 100, batch size 64 dan lr 0.001.

Dapat disimpulkan berdasarkan grafik persentase hasil percobaan terhadap 2 arsitektur bahwa penerapan adjusting terbukti memberikan hasil yang lebih signifikan jika dibandingkan dengan hasil percobaan tanpa adjusting hyperparameter memperoleh nilai akurasi sebesar 85.56% untuk resnet 50, dan 97.00% untuk inception v3, setelah diterapkan adjusting hyperparameter pada proses training seperti epoch, batch size dan lr, hasil akurasi inception v3 meningkat menjadi 98.36% dan hasil akurasi resnet 50 meningkat menjadi 89.65%. Ini membuktikan bahwa kinerja hyperparameter pada proses training dapat memberikan hasil akurasi yang lebih baik, namun perlu digaris bawahi bahwa dalam mendapatkan akurasi tertinggi pada adjusting hyperparameter, perlu dilakukan beberapa kali pelatihan dalam mencari kombinasi nilai hyperparameter yang terbaik.

Setelah didapatkan fitur pada gambar, kemudian dilakukan proses pengenalan dan pengidentifikasian pada layer fully connected layer yang akan menentukan kelas dari gambar pistachio. Tahap selanjutnya model tersebut akan dilakukan pengujian menggunakan confusion matrix untuk mengetahui nilai nilai true positif, false positif, true negatif dan false negatif untuk selanjutnya digunakan untuk mengevaluasi perhitungan seperti akurasi, recall dan presisi. Berikut adalah hasil keseluruhan pengujian berdasarkan akurasi, presisi dan recall terhadap model klasifikasi dapat dilihat pada Tabel 4.7.

Tabel 4.7. Hasil Pengujian

Skenario 1				
Keterangan	Lr	Akurasi	Precision	Recall
Inception-V3 Epoch 100 Batch Size 32	0.001	97.00%	0.97	0.97
Skenario 2				
Keterangan	Lr	Akurasi	Precision	Recall
ResNet50 Epoch 100 Batch Size 32	0.001	85.56%	0.86	0.86
Skenario 3				
Keterangan	Lr	Akurasi	Precision	Recall
InceptionV3+Parameter Epoch 50 Batch Size 32	Lr 0.1	50.41%	0.25	0.50
	Lr 0.01	96.46%	0.96	0.96
	Lr 0.001	96.73%	0.97	0.97
	Lr 0.0001	95.91%	0.96	0.96
	Lr 0.00001	96.19%	0.96	0.96
Epoch 50 Batch Size 64	Lr 0.1	50.40%	0.25	0.50
	Lr 0.01	96.19%	0.96	0.96
	Lr 0.001	96.19%	0.96	0.96
	Lr 0.0001	96.19%	0.96	0.96
	Lr 0.00001	97.00%	0.97	0.97
Epoch 100 Batch Size 32	Lr 0.1	49.59%	0.25	0.50
	Lr 0.01	49.59%	0.25	0.50
	Lr 0.001	98.36%	0.98	0.98
	Lr 0.0001	95.64%	0.96	0.96
	Lr 0.00001	97.00%	0.96	0.96
Epoch 100	Lr 0.1	49.59%	0.25	0.50
	Lr 0.01	95.64%	0.96	0.96

Tabel 4.7. Lanjutan Hasil Pengujian

Batch Size 64	Lr 0.001	97.28%	0.97	0.97
	Lr 0.0001	97.55%	0.97	0.97
	Lr 0.00001	97.28%	0.97	0.97
Skenario 4				
Keterangan	Lr	Akurasi	Precision	Recall
ResNet50+Parameter	0.1	50.41%	0.25	0.50
Epoch 50 Batch Size 32	0.01	49.59%	0.25	0.50
	0.001	85.56%	0.86	0.86
	0.0001	84.20%	0.86	0.84
	0.00001	83.65%	0.85	0.84
Epoch 50 Batch Size 64	0.1	52.19%	0.25	0.50
	0.01	50.41%	0.25	0.50
	0.001	75.75%	0.25	0.50
	0.0001	89.10%	0.89	0.90
	0.00001	86.92%	0.25	0.50
Epoch 100 Batch Size 32	0.1	49.59%	0.25	0.50
	0.01	50.41%	0.25	0.50
	0.001	85.01%	0.87	0.85
	0.0001	87.74%	0.88	0.88
	0.00001	85.01%	0.87	0.85
Epoch 100 Batch Size 64	0.1	49.59%	0.25	0.50
	0.01	88.01%	0.88	0.88
	0.001	89.65%	0.90	0.90
	0.0001	80.38%	0.85	0.80
	0.00001	49.59	0.25	0.50

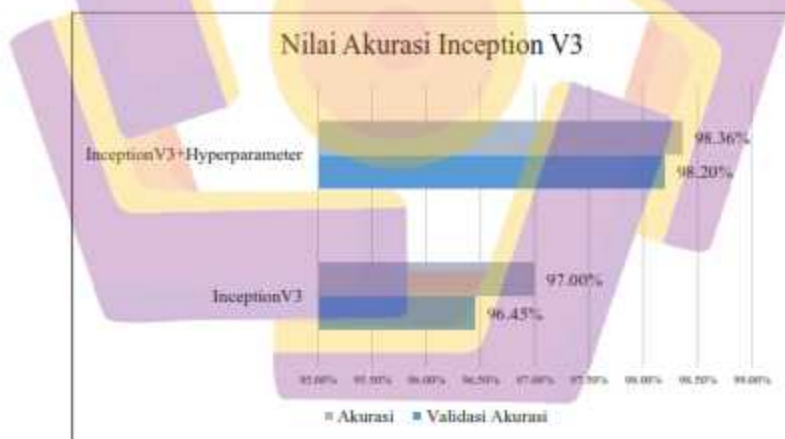
Pengujian diterapkan pada kedua arsitektur yang digunakan, jika dilihat pada Tabel 4.7 hasil pengujian dari keseluruhan skenario, nilai akurasi tertinggi diperoleh pada pengujian skenario ketiga dengan adjusting hyperparameter pada

arsitektur inception v3, mendapatkan nilai akurasi sebesar 98.36%, sedangkan presisi mendapatkan nilai sebesar 98, begitupun recall mendapatkan nilai sebesar 98.

Jika melihat hasil pengujian yang didapatkan dapat disimpulkan bahwa kinerja inception v3 lebih baik daripada resnet 50 jika dilihat dari nilai akurasi, presisi dan recall pada tabel pengujian. Meskipun demikian, hasil pengujian keseluruhan skenario masih mengalami error deteksi sehingga hal ini bisa menjadi topik untuk penelitian selanjutnya.

4.6.1 Hasil Akurasi Inception V3

Berikut adalah hasil perbandingan akurasi arsitektur inception v3, dan inception v3 dengan penerapan adjusting hyperparameter.



Gambar 4.22. Hasil Arsitektur Inception V3

Seperti yang terlihat pada Gambar 4.22 penggunaan inception v3 dalam penelitian ini menghasilkan nilai akurasi 97.00% dan nilai validasi sebesar 96.45%, jika dibandingkan hasil akurasi inception v3 dengan penerapan hyperparameter

mengalami peningkatan yang tidak signifikan, hanya sebesar 1% dan nilai validasi mengalami peningkatan sebesar 2% yaitu 98.20%. Namun bisa dikatakan hasil akurasi maupun hasil validasi yang dihasilkan, telah mencapai kinerja terbaik pada penggunaan adjusting hyperparameter.

4.6.2 Hasil Akurasi ResNet 50

Berikut sajian tabel perbandingan untuk hasil akurasi menggunakan ResNet50.



Gambar 4.23. Hasil Arsitektur ResNet 50

Gambar 4.23 merupakan perbandingan hasil akurasi ResNet 50 dengan ResNet 50 ditambah penyetulan hyperparameter. Dapat diperhatikan bahwa hasil akurasi ResNet 50 mengalami kenaikan yang cukup baik yaitu sebesar 4% dari 85.56% meningkat pada penyetulan hyperparameter sebesar 89.65%. Akan tetapi pada nilai validasi hanya meningkat sebesar 1% yaitu sebesar 85.52%, hal ini dikarenakan resnet 50 memiliki kompleksitas arsitektur dengan mengkonsumsi waktu pelatihan yang lebih lama sehingga mempengaruhi akurasi yang dihasilkan.

4.6.3 Perbandingan Hasil Akurasi Kedua Arsitektur

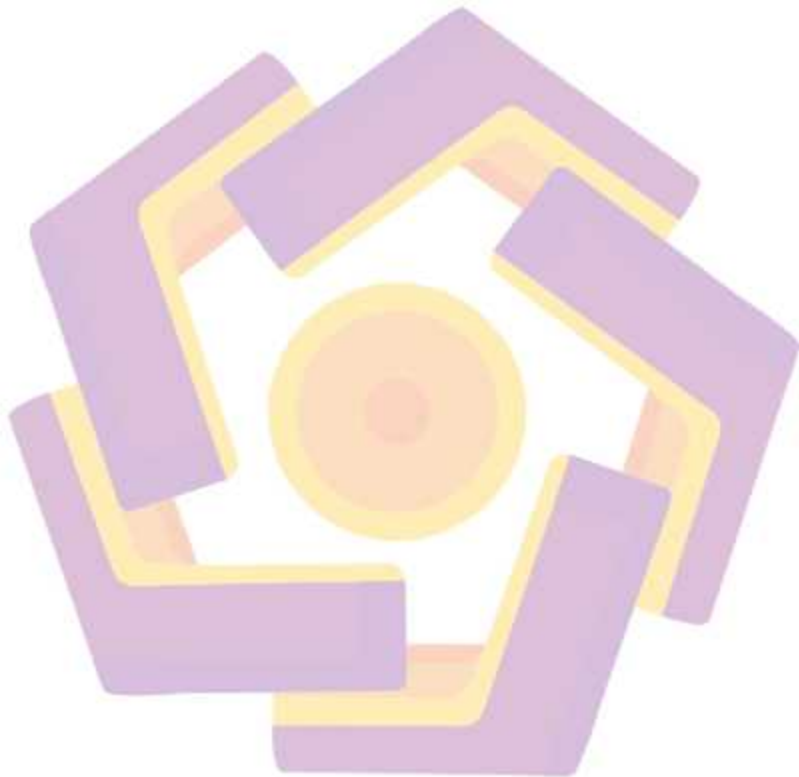
Pemilihan arsitektur Inception V3 dan ResNet 50 pada penelitian ini untuk mengetahui performa kedua model dalam mengklasifikasikan kacang pistachio yang hasilnya nanti agar dapat dipertimbangkan digunakan sebaik mungkin untuk penelitian selanjutnya. Parameter akurasi menjadi perbandingan utama dalam penelitian ini. Nilai akurasi dapat menggambarkan bagus tidaknya suatu model klasifikasi akan digunakan untuk mengklasifikasi data kacang pistachio.

Jika dilihat pada gambar grafik hasil arsitektur inception v3 dan resnet 50 performa terbaik dalam penelitian ini adalah inception v3 dengan penerapan hyperparameter pada proses training untuk mencari kombinasi terbaik dari penggunaan epoch, batch size, dan lr. Dimana penggunaan epoch 100, batch size 32 dan lr 0.001 yang menghasilkan nilai akurasi yang tertinggi, yaitu sebesar 98.36%.

Ditinjau pada penelitian sebelumnya yang dilakukan oleh (Singh Dilbagh et al., 2022), mendapatkan akurasi tertinggi pada penggunaan arsitektur VGG16 sebesar 96.84%, jika dibandingkan dengan penelitian saat ini yang dilakukan, berhasil mendapatkan akurasi tertinggi sebesar 98.37% menggunakan arsitektur inception v3 dengan penggunaan adjusting hyperparameter dalam mencari kombinasi terbaik terhadap hasil akurasi yang dihasilkan. Sehingga penggunaan arsitektur inception v3 dapat dipertimbangkan untuk penelitian selanjutnya.

Adapun permasalahan pada penelitian sebelumnya, telah diperbaiki berdasarkan hasil yang dipaparkan, telah dilakukan perbaikan pada kurangnya pelatihan model yang tidak memiliki cukup kesempatan untuk belajar dari seluruh

data pelatihan serta tidak ada percobaan yang dilakukan dalam menetapkan kombinasi nilai hyperparameters pada training yang digunakan. Hal ini mengakibatkan model mengalami stagnasi, tidak mengalami penurunan dan tidak mengalami peningkatan hasil akurasi.



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan analisa hasil pada pengujian yang telah dilakukan terhadap arsitektur inception-v3 dan arsitektur resnet 50 dapat disimpulkan bahwa:

1. Nilai akurasi tertinggi pada penelitian ini terhadap klasifikasi kacang pistachio adalah arsitektur Inception-V3 dengan penerapan Hyperparameter pada proses training, mendapatkan nilai akurasi sebesar 98.36%. Sedangkan pada penggunaan ResNet 50 mendapatkan nilai akurasi sebesar 89.65%. Kinerja inception v3 lebih baik dari resnet 50 sehingga penggunaan inception v3 dengan adjusting hyperparameter bisa dipertimbangkan untuk penelitian berikutnya.
2. Pada proses pengujian masih mengalami error dalam mendeteksi kacang kirmizi dan siirt. Dikarenakan pada gambar kirmizi dan siirt memiliki biji dan cangkang bahkan warna yang sama. Hal ini menjadi faktor dalam pemrosesan data seperti data mengalami ambiguitas, dimana beberapa objek gambar sulit untuk dibedakan secara visual, sehingga model mengalami kesulitan untuk mengklasifikasikan gambar dengan benar. Dan juga faktor kebisingan dalam memproses data, ini dapat berasal dari berbagai sumber seperti pencahayaan yang buruk pada gambar, artefak dalam gambar serta kualitas gambar yang rendah dan kurang jelas.

3. Penerapan hyperparameter pada proses training untuk mencari kombinasi terbaik dapat meningkatkan hasil akurasi yang lebih baik dalam tugas klasifikasi.
4. Memerlukan waktu dan sumber daya komputasi yang signifikan. Jika jumlah kombinasi yang diuji banyak, maka proses pelatihan dapat menjadi sangat lambat dalam hal waktu pelatihan dan kebutuhan komputasi.

5.2. Saran

Berikut ini adalah beberapa saran yang dapat dijadikan pedoman untuk melakukan pengembangan penelitian ini, diantaranya adalah:

1. Melakukan eksplorasi dan percobaan pada beberapa arsitektur CNN yang lain untuk mengetahui hasil yang maksimal.
2. Pengumpulan data yang berkualitas tinggi, untuk mencegah ambiguitas dan kebisingan dalam pemrosesan data.
3. Mempertimbangkan penggunaan hyperparameter pada proses training untuk menentukan kombinasi yang terbaik.

DAFTAR PUSTAKA

PUSTAKA BUKU

- Santoso, B. 2007, *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Graha Ilmu, Yogyakarta.
- Suyanto, 2018, *Machine Learning Tingkat Dasar dan Lanjut*, Edisi pertama, Informatika Bandung, Bandung.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- ALAM, I. N. (2022). METODE TRANSFER LEARNING PADA DEEP CONVOLUTIONAL NEURAL NETWORK (DCNN) UNTUK PENGENALAN EKSPRESI WAJAH.
- Bonifazi Giuseppe, Capobianco Giuseppe, Gasbarrone Riccardo, & Serranti Silvia. (2021). Contaminant detection in pistachio nuts by different classification methods applied to short-wave infrared hyperspectral images. *Food Control*.
- Dong, N., Zhao, L., Wu, C. H., & Chang, J. F. (2020). Inception v3 based cervical cell classification combined with artificially extracted features. *Applied Soft Computing*, 93, 106311. <https://doi.org/10.1016/j.asoc.2020.106311>.
- Fitra Maulana, F., & Rochmawati, N. (n.d.). *Klasifikasi Citra Buah Menggunakan Convolutional Neural Network*.
- Genis Huseyin Efe, Durna Sahin, & Boyaci Ismail Hakki. (2021). Determination of green pea and spinach adulteration in pistachio nuts using NIR spectroscopy. *LWT-Food Science and Technology*. <https://doi.org/https://doi.org/10.1016/j.lwt.2020.110008>.
- Hartati Ery. (2020). PENGGUNAAN KLASIFIKASI SAYUR SEGAR DAN SAYUR BUSUK MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE. *Jurnal Teknik Informatika Dan Sistem Informasi*, Vol. 7 No. 3, 678-687.
- Hanin, M. A., Patmasari, R., & Fuâ, R. Y. N. (2021). Sistem Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network (cnn). *eProceedings of Engineering*, 8(1).
- Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, 132, 679-688. <https://doi.org/10.1016/j.procs.2018.05.069>.

- Karpathy, A. (n.d.). Convolutional Neural Networks for Visual Recognition.
- Li, H., & Zhang, L. (2018). MULTI-EXPOSURE FUSION WITH CNN FEATURES Hui Li and Lei Zhang Department of Computing , The Hong Kong Polytechnic University. 2018 25th IEEE International Conference on Image Processing (ICIP), 1723–1727.
- Mandalari Giuseppina, Barreca Davide, Gervasi Teresa, Roussel Michael A., Klein Bob, Feeney Mary Jo, & Carughi Arianna. (2022). Pistachio Nuts (*Pistacia vera* L.): Production, Nutrients, Bioactives and Novel Health Effects. *Plants*. <https://doi.org/10.3390/plants11010018>.
- Menevseoglu Ahmed, Aykas Didem Peren, & Adal Eda. (2020). Non-targeted approach to detect green pea and peanut adulteration in pistachio by using portable FT-IR, and UV-Vis spectroscopy. *Journal of Food Measurement and Characterization*.
- Moeis Faizal Rahmanto, Dartanto Teguh, Moeis Josy Prananta, & Ikhsan Mohamad. (2020). A longitudinal study of agriculture households in Indonesia: The effect of land and labor mobility on welfare and poverty dynamics. *World Development Perspectives*.
- Nour, M., Cömert, Z., & Polat, K. (2020). A Novel Medical Diagnosis model for COVID-19 infection detection based on Deep Features and Bayesian Optimization. *Applied Soft Computing*, 97. <https://doi.org/10.1016/j.asoc.2020.106580>.
- Putra, R. R., Isa, I. G. T., & Malyan, A. B. J. (2023). *Buku Ajar Pengantar Deep Learning dalam Pemrosesan Citra*, Penerbit NEM.
- Özkan İlker Ali, Köklü Murat, & Saraçoğlu Rıdvan. (2021). Classification of Pistachio Species Using Improved k-NN Classifier. *Progress in Nutrition*, Vol. 23 No. 2.
- Singh Dilbag, Taspınar Yavuz Selim, Kursun Ramazan, Cınar İlkay, Koklu Murat, Ozkan İlker Ali, & Lee Heung-No. (2022). Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models. *Electronics*.
- Vidyarathi Sriram K., Tiwari Rakhee, Singh Samrendra K., & Xiao Hong-Wei. (2020). Prediction of size and mass of pistachio kernels using random Forest machine learning. *Food Process Engineering*.

PUSTAKA ELEKTRONIK

- Ghoneim, S. (2019, April 2). Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on? Retrieved October 2022.