

***COMPUTER VISION UNTUK MENGETAHUI KEMATANGAN
JAMBU KRISTAL MENGGUNAKAN METODE CONVOLUTIONAL
NEURAL NETWORK***

TUGAS AKHIR

Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika Universitas Pembangunan Nasional “Veteran” Yogyakarta



Disusun Oleh :

NAMA : MUHAMMAD ALIFADIN RAMADHAN

NIM : 123150128

**PROGRAM STUDI INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" YOGYAKARTA
2022**

***COMPUTER VISION UNTUK MENGETAHUI KEMATANGAN
JAMBU KRISTAL MENGGUNAKAN METODE CONVOLUTIONAL
NEURAL NETWORK***

TUGAS AKHIR



Disusun Oleh :

NAMA : MUHAMMAD ALIFADIN RAMADHAN

NIM : 123150128

**PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" YOGYAKARTA
2022**

HALAMAN PENGESAHAN PEMBIMBING

**COMPUTER VISION UNTUK MENGETAHUI KEMATANGAN JAMBU
KRISTAL MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK**

Disusun oleh:

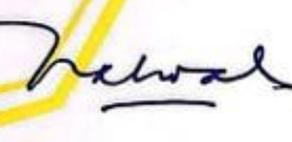
Muhammad Alifadin Ramadhan

123150128

Telah diuji dan dinyatakan lulus oleh pembimbing
pada tanggal: 17 Juni 2012

Menyetujui,
Pembimbing I

Menyetujui,
Pembimbing II



DR. Herlina Jayadianti, S.T., M.T.

NIK. 2 7708 02 0235 1

Nur Heri Cahyana, S.T., M.Kom.

NIP. 1960 09 22 1984 03 1001

Mengetahui,
Koordinator Program Studi

Dr. Herivanto, A.Md., S.Kom., M.Cs.

NIK 2 7706 06 11 0301 1

HALAMAN PENGESAHAN PENGUJI

**COMPUTER VISION UNTUK MENGETAHUI KEMATANGAN JAMBU
KRISTAL MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL
NETWORK***

Disusun oleh:

Muhammad Alifadin Ramadhan
123150128

Telah diuji dan dinyatakan lulus pada tanggal 7 Juni..... oleh :

Menyetujui,
Penguji I

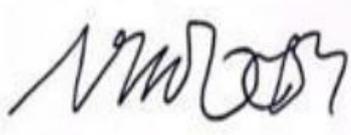
Penguji II


Dr. Herlina Jayadianti, S.T., M.T.
NIK. 2 7708 02 0235 1


Nur Heri Cahyana, S.T,M.Kom.
NIP. 1960 09 22 1984 03 1001

Penguji III

Penguji IV


Dr. Novrido Charibaldi, S.Kom, M.Kom
NIK. 2 6811 96 0066 1


Juwairiah, S.Si., M.T.
NIK. 2 7607 00 0230 1

PERNYATAAN BEBAS PLAGIAT

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Alifadin Ramadhan
NPM : 123150128
Fakultas/Prodi : Teknik Industri/Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir **COMPUTER VISION UNTUK MENGETAHUI KEMATANGAN JAMBU KRISTAL MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK*** Adalah hasil kerja saya sendiri dan benar bebas dari plagiat kecuali cuplikan serta ringkasan yang terdapat di dalamnya telah saya jelaskan sumbernya (sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Yogyakarta, 7 Juni 2022
Yang Membuat Pernyataan



Muhammad Alifadin Ramadhan
NPM. 123150128

SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertanda tangan di bawah ini, saya:

Nama : Muhammad Alifadin Ramadhan
No. Mhs : 123150128

Menyatakan bahwa karya ilmiah saya yang berjudul:

COMPUTER VISION UNTUK MENGETAHUI KEMATANGAN JAMBU KRISTAL MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK*
Merupakan karya asli saya dan belum pernah dipublikasikan di manapun. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa pun yang diberikan Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada : 7 Juni 2022


Muhammad Alifadin Ramadhan

HALAMAN PERSEMBAHAN

Puji Syukur kepada Tuhan yang Maha Esa

Puji syukur atas segala rahmatNya dalam keberhasilan menyelesaikan studi di Teknik Informatika UPN ‘Veteran’ Yogyakarta. Semoga karya tulis dapat bermanfaat bagi masyarakat umum. Kupersembahkan karya sederhana ini beserta ucapan terimakasih kepada orang yang sangat kusayangi.

1. Terkhusus untuk almarhumah Mamah saya yang meninggal pada 14 Januari 2015.
2. Papa dan adik-adik saya yang sangat berarti bagi kehidupan saya.
3. Dosen-Dosen Informatika yang membantu menyelesaikan masa studi saya terutama Bu Herlina, Pak Nur, Bu Juwai, Pak Edo dan dosen wali Pak Oliver.
4. Putri Avanny Jamaica yang selalu menjadi pengingat dan support saya dalam menjalani kehidupan yang lebih baik.
5. Keluarga besar IF 2015 yang menjadi teman suka dan duka selama di Teknik Informatika UPN ‘Veteran’ Yogyakarta.
6. Teman – teman Kantor Fams yang sangat berarti bagi saya, selalu menjalani suka dan duka dalam kehidupan dunia kampus.
7. Vincent Dian Asa Putra yang sudah membantu saya selama mengerjakan skripsi selama 2 tahun 2020-2022.
8. Teman – teman Team 7 yang hampir selalu menemani saya ketika bermain mobile legend.
9. Keluarga besar Mbah Salamun yang ada di Kab. Bogor tepatnya di Perum Pura Bojonggede yang sudah saya anggap saudara sendiri.

Aku bersyukur dan berterimakasih pada orang-orang yang masih mau dan ada untukku, serta tetap membersamai diriku, walaupun terkadang aku bersikap tidak baik terhadap mereka.

-

Dan semoga Allah menjaga, melindungi serta melimpahkan hidup mereka dengan nikmat juga rahmatNya

ABSTRAK

Jambu kristal merupakan varietas dari jambu biji (*Psidium Guajava L.*). Jambu kristal merupakan varietas baru sejak tahun 1998, hasil dari bantuan transfer teknologi Taiwan untuk Indonesia dalam menciptakan berbagai hasil rekayasa genetika (Wang, 2011). Jambu kristal memiliki beberapa keunggulan yaitu memiliki cita rasa yang segar, manis seperti buah apel dan pir, mudah dibudidayakan, berbuah sepanjang tahun, peluang wirausaha yang tinggi baik buah dan pembibitan, harga jual jambu ini di tingkat petani dan di supermarket cukup tinggi (Pakpahan, 2011). Dalam penentuan kematangan buah jambu kristal dapat dilihat dari parameter ciri warna. Dalam penentuan kematangan buah jambu kristal berdasarkan wana kulit buah secara visual mata manusia memiliki beberapa kekurangan diantaranya membutuhkan tenaga lebih banyak untuk menyortir kematangan buah dan penilaian kematangan oleh manusia bersifat subyektif dan tidak konsisten dalam penilaian penghilatan satu penilai dengan penilai lainnya sehingga mempengaruhi tingkat kualitas dalam pengklasifikasian. Maka itu, diperlukan sebuah aplikasi untuk mempermudah klasifikasi.

Pada penelitian ini, proses klasifikasi kematangan buah jambu kristal menggunakan metode Convolutional Neural Network (CNN). Hasilnya klasifikasi akan ditampilkan menggunakan Computer Vision dengan kamera secara realtime. Pengujian dilakukan pada 40 data uji dan 572 data latih. Hasil yang diperoleh dari pengujian menggunakan metode Convolutional Neural Network dan Computer Vision ini menghasilkan akurasi 95%. Berdasarkan nilai akurasi tersebut, maka dapat disimpulkan bahwa Computer Vision menggunakan metode Convolutional Neural Network mampu mengklasifikasi kematangan buah jambu kristal berdasarkan warna. Tujuan: Implementasi Computer Vision menggunakan Convolutional Neural Network dalam mengklasifikasikan kematangan buah jambu kristal.

Kata Kunci: Pengolahan Citra; Convolutional Neural Network; Computer Vision; Kematangan; Klasifikasi; Jambu Kristal.

ABSTRACT

Crystal guava (*Psidium guajava* L.) is a variety of guava. Crystal guava is a new variety since 1998, the result of technology transfer assistance from Taiwan to Indonesia in creating various genetically modified products (Wang, 2011). Crystal guava has several advantages, including having a fresh, sweet taste like apples and pears, easy to cultivate, bears fruit throughout the year, and give high opportunities for entrepreneurs both in fruit and in nurseries (Pakpahan, 2011). In determining the ripeness of crystal guava fruit, it can be seen from the color characteristics parameters. However, the determination of the ripeness of crystal guava fruit based on the color of the fruit skin visually by the human eye has several weaknesses, including requiring more energy to sort fruit ripeness, the ripeness assessment by humans is subjective and inconsistent in assessing the sight of one assessor with another so that it can affect the quality level of classification. Therefore, an application to facilitate classification is needed.

In this study, the Convolutional Neural Network (CNN) method is used to classify the ripeness of crystal guava fruit. The results of the classification will be displayed using Computer Vision with the camera in real time. Tests were carried out on 40 test data and 572 training data. The results obtained from testing using the Convolutional Neural Network and Computer Vision method obtain an accuracy of 95%. Based on the accuracy value, it can be concluded that Computer Vision using the Convolutional Neural Network method is able to classify the ripeness of crystal guava fruit based on color. Objective: Implementation of computer vision using deep learning convolutional neural network in classifying the ripeness of crystal guava fruit.

Keywords: Image Processing; Convolutional Neural Network; Computer Vision; Ripeness, Classification; Crystal Guava.

KATA PENGANTAR

Alhamdulillah, puji syukur kepada Allah SWT, atas karunia, rahmat, dan hidayah-Nya, penulis dapat menyelesaikan tugas akhir ini yang berjudul “COMPUTER VISION UNTUK MENGETAHUI KEMATANGAN JAMBU KRISTAL MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK”.

Penulisan skripsi ini diajukan untuk memenuhi salah satu syarat kelulusan dalam jenjang perkuliahan Strata 1 Universitas Pembangunan Nasional “Veteran” Yogyakarta. Dalam penulisan skripsi ini tidak lepas dari hambatan dan kesulitan, namun berkat bimbingan, bantuan, nasihat dan saran serta kerjasama dari berbagai pihak, khususnya pembimbing, segala hambatan tersebut akhirnya dapat diatasi dengan baik.

Dalam penulisan skripsi ini tentunya tidak lepas dari kekurangan, baik aspek kualitas maupun aspek kuantitas dari materi penelitian yang disajikan. Semua ini didasarkan dari keterbatasan yang dimiliki penulis.

Penulis menyadari bahwa skripsi ini jauh dari sempurna sehingga penulis membutuhkan kritik dan saran yang bersifat membangun untuk penelitian ini. Selanjutnya dalam penulisan skripsi ini penulis banyak diberi bantuan oleh berbagai pihak. Dalam kesempatan ini, penulis dengan tulus hati mengucapkan terimakasih kepada:

1. Papa, Mama, Ibu dan Adik-adik yang selalu memberikan dukungan dalam berbagai macam bentuknya dan selalu memberikan doa,
2. Ibu Dr. Herlina Jayadianti, S.T., M.T. selaku dosen pembimbing I,
3. Bapak Nur Heri Cahyana, S.T., M.Kom. selaku dosen pembimbing II,
4. Bapak dan Ibu dosen program studi Teknik Informatika Fakultas Teknik Industri yang telah memberikan ilmu dan pengalaman,
5. Berbagai pihak yang telah memberikan bantuan dan dorongan serta berbagipengalaman pada proses penyusunan skripsi ini.

Terakhir, semoga segala bantuan yang telah diberikan, sebagai amal soleh senantiasa mendapat Ridho Allah SWT. Sehingga pada akhirnya semoga skripsi ini dapat bermanfaat dalam bidang informatika dan dapat dikembangkan menjadi lebih baik lagi.

Yogyakarta, 7 Juni 2022

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN PENGUJI.....	iii
PERNYATAAN BEBAS PLAGIAT	iv
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR.....	v
HALAMAN PERSEMBAHAN	vi
ABSTRAK.....	vii
ABSTRACT	viii
KATA PENGANTAR.....	ix
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR MODUL PROGRAM.....	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Metodologi Penelitian	3
1.6.1. Studi Pustaka.....	3
1.6.2. Wawancara.....	3
1.6.3. Pengumpulan Data	3
1.6.4. Pengolahan Data	4
1.7. Metode Pengembangan Sistem	4
1.8. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1. Jambu Kristal	6
2.2. Metode Waterfall	7
2.3. Citra Digital	8
2.3.1. Definisi Citra Digital.....	8
2.3.2. Representasi Citra Digital	8
2.3.3. Pengolahan Citra Digital.....	8
2.4. Computer Vision.....	9
2.5. Image Processing	10
2.5.1. Resizing.....	11
2.5.2. Median Filter.....	11
2.5.3. Histogram.....	11
2.5.4. Histogram Equalization	12

2.6.	Deep Learning	12
2.7.	Neural Network.....	12
2.7.1.	Activation Function	13
2.7.1.1.	Rectified Linear Unit (ReLU).....	14
2.7.1.2.	Layer	14
2.8.	Convolutional Neural Network.....	14
2.8.1.	Prinsip Convolutional Neural Network.....	15
2.8.2.	Desain Arsitektur Convolutional Neural Network	16
2.8.3.	Arsitektur VGG.....	17
2.8.4.	Convolutional Layer	18
2.8.5.	Padding	18
2.8.6.	Pooling Layer.....	19
2.8.7.	Fully Connected Layer.....	20
2.8.8.	Training Data	20
2.9.	Object Detection	21
2.10.	Python	21
2.11.	TensorFlow	21
2.12.	Keras	22
2.13.	Confusion Matrix	22
2.14.	Studi Pustaka (State of The Art)	23
BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM.....		26
3.1.	Metodologi Penelitian	26
3.1.1.	Pengumpulan Data	26
3.1.2.	Data Preprocessing.....	29
3.1.3.	Convolutional Neural Network.....	32
3.1.4.	Pengujian Confusion Matrix Multi Class	42
3.2.	Analisis Kebutuhan	42
3.2.1.	Jenis dan Sumber Data.....	42
3.2.2.	Prosedur Pengumpulan Data.....	42
3.2.3.	Analisis Kebutuhan Data	43
3.2.4.	Kebutuhan Fungsional	43
3.2.5.	Kebutuhan Non-Fungsional	43
3.3.	Tahap Pengembangan Sistem	43
3.3.1.	Membangun Waterfall	45
3.3.2.	Membangun Arsitektur Sistem	45
3.3.3.	Flowchart Proses Keseluruhan Sistem.....	45
3.3.4.	Perancangan Antarmuka	47
3.4.	Perancangan Pengujian Model.....	49
BAB IV.....		51
4.1.	Hasil Penelitian	51
4.1.1.	Pembentukan Model	51
4.1.2.	Implementasi Model	58
4.2.	Pengujian Model	60

BAB V

KESIMPULAN	71
5.1. Kesimpulan	71
5.2. Saran	71

DAFTAR TABEL

Tabel 2.1 <i>Confusion Matrix</i>	22
Tabel 2.2 Studi Pustaka	23
Tabel 3.1 Tabel Wawancara	27
Tabel 3.2 Rincian Kelas Data	31
Tabel 3.3 Pengujian <i>Confussion Matrix Multi Class</i>	50
Tabel 4.1 Tabel Pengujian Model.....	60
Tabel 4.2 Tabel <i>Confusion Matrix</i>	70

DAFTAR GAMBAR

Gambar 2.1. Jambu Kristal (Cahyo, 2016).....	6
Gambar 2.2. Metode Prototype (Pressman, 2010).....	7
Gambar 2.3. Contoh Representasi Suatu Citra $f(x, y)$ Dalam 2 Dimensi.....	8
Gambar 2.4. Struktur Computer Vision (Amrizal & Aini, 2011).....	10
Gambar 2.5. Contoh Sederhana Arsitektur <i>Neural Network</i> (Ratnasari, 2018).....	13
Gambar 2.6. Grafik Fungsi ReLU (Novyantika, 2018).....	14
Gambar 2.7. <i>Main Process of CNN</i> (T. Liu et al., 2015).....	16
Gambar 2.8. <i>Architecture of CNN in Training Face</i> (T. Liu et al., 2015).....	17
Gambar 2.9. Arsitektur VGG-16 (Hindarto & Santoso, 2019).....	17
Gambar 2.10. Ilustrasi Proses Konvolusi.....	18
Gambar 2.11. <i>Padding</i>	19
Gambar 2.12. Perbedaan <i>Average Pooling</i> dan <i>Max Pooling</i> (Rawat & Wang, 2017).....	19
Gambar 2.13. <i>Fully Connected Layer</i>	20
Gambar 3.1. Tahapan Kerangka Kerja Penelitian.....	26
Gambar 3.2. Jambu Kristal Kebun Larasati.....	27
Gambar 3.3. Masing-masing Klasifikasi Jambu Kristal.....	28
Gambar 3.4. <i>Preprocessing</i>	29
Gambar 3.5. <i>Histogram Equalization</i>	30
Gambar 3.6. Proses <i>Labelling</i>	31
Gambar 3.7. Arsitektur VGG16.....	32
Gambar 3.8. Ilustrasi Perhitungan Lapisan Konvolusi.....	33
Gambar 3.9. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-1.....	35
Gambar 3.10. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-3.....	36
Gambar 3.11. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-5.....	36
Gambar 3.12. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-7.....	36
Gambar 3.13. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-9.....	37
Gambar 3.14. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-11.....	37
Gambar 3.15. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-15.....	37
Gambar 3.16. Lapisan Konvolusi <i>depthwise</i> dan <i>pointwise</i> ke-16.....	38
Gambar 3.17. <i>Fully Connected Layer</i>	39
Gambar 3.18. Flowchart Model CNN.....	40
Gambar 3.19. Proses Training.....	41
Gambar 3.20. Metode Pengembangan Sistem.....	44
Gambar 3.21. Arsitektur Sistem.....	45
Gambar 3.22. Flowchart Keseluruhan Sistem.....	45
Gambar 3.23. Flowchart Testing.....	46
Gambar 3.24. Antarmuka Halaman Utama.....	47
Gambar 3.25. Antarmuka Membuat Dataset.....	48
Gambar 3.26. Antarmuka <i>Training</i>	48
Gambar 3.27. Antarmuka Deteksi Jambu.....	49

Gambar 4.1. Halaman <i>Create Dataset</i>	52
Gambar 4.2. Isi File CSV	53
Gambar 4.3. Isi File <i>Train.csv</i>	54
Gambar 4.4. Isi File <i>Testing.csv</i>	54
Gambar 4.5. Halaman <i>Training</i>	55
Gambar 4.6. Proses Training	57
Gambar 4.7. Hasil Loss Classification	57
Gambar 4.8. Halaman Deteksi.....	58
Gambar 4.9. Halaman Hasil Deteksi	59
Gambar 4.10. Halaman Deteksi <i>Realtime</i>	59

DAFTAR MODUL PROGRAM

Modul Program 4.1. Source Code Histogram Equalization	54
Modul Program 4.2. Source Code Fungsi Konversi XML ke CSV	55
Modul Program 4.3. Source Code Split Data Train dan Testing.....	57
Modul Program 4.4. Source Code Pendefinisian Kelas	57
Modul Program 4.5. Source Code Generate TFRecord.....	58
Modul Program 4.6. Source Code Konfigurasi SSD Mobilenet	59
Modul Program 4.7. Source Code Create Model	59
Modul Program 4.8. Source Code Load Image	59
Modul Program 4.9. Source Code Start Webcam	60

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Jambu kristal merupakan varietas dari jambu biji (*Psidium Guajava L.*) yang memiliki sentra pengembangan didaerah Bengkulu (Bengkulu Utara), Jawa Barat (Depok, Majalengka, Bogor, Kuningan, Subang, Sukabumi, Sumedang), DI. Yogyakarta (Gunung Kidul), Papua (Kota Jayapura), Jawa Tengah (Sukoharjo), dan Papua Barat (Manokwari) (Nita, 2016). Jambu kristal merupakan varietas baru sejak tahun 1998, hasil dari bantuan transfer teknologi Taiwan untuk Indonesia dalam menciptakan berbagai hasil rekayasa genetika. Jambu ini memiliki ciri-ciri, yaitu rasa manis dengan kadar kemanisan 11–12° briks, bentuk buah bulat sedikit gepeng terkadang memiliki bentuk yang tidak simetris, kandungan biji kurang dari 3%, permukaan buah ada tonjolan tidak merata, bobot buah 100–500 gram per buah, warna kulit hijau muda, sedangkan daging buah putih, tekstur daging buah renyah seperti buah pir (Wang, 2011).

Jambu kristal memiliki beberapa keunggulan yaitu memiliki cita rasa yang segar, manis seperti buah apel dan pir, mudah dibudidayakan, berbuah sepanjang tahun, peluang wirausaha yang tinggi baik buah dan pembibitan, harga jual jambu ini di tingkat petani dan di supermarket cukup tinggi (Pakpahan, 2011). Kondisi ukuran buah jambu ini pada musim penghujan akan mempunyai ukuran buah yang lebih besar dari musim kemarau. Namun, rasa buah jambu kristal akan lebih manis pada musim kemarau. Dari uraian tersebut membuat jambu kristal menjadi komoditas unggulan petani untuk peningkatan perekonomian.

Sebagai komoditas unggulan petani tentu buah jambu kristal dilihat dari tingkat kematangannya. Kematangan buah biasanya ditentukan oleh beberapa parameter, diantaranya adalah parameter ukuran, berat, ciri warna, keharuman dari buah tersebut dan lain-lain (Chan et al., 2014). Pada kematangan buah jambu kristal dapat dilihat dari parameter ciri warna. Buah jambu kristal akan berubah warna ketika matang dari berwarna hijau menjadi berwarna hijau kekuningan proses tersebut *degreening*. Berikut beberapa penelitian sebelumnya yang menentukan kematangan buah jambu biji berdasarkan parameter warna oleh Adhi (Wibowo et al., 2021), Febry (Mulato, 2015) dan Hidayat (Hidayat et al., 2017). Pada beberapa penelitian tersebut menjelaskan dalam menentukan kematangan buah jambu berdasarkan warna kulit buah secara visual mata manusia memiliki beberapa kekurangan diantaranya dibutuhkannya tenaga lebih banyak untuk menyortir kematangan buah dan penilaian kematangan oleh manusia bersifat subyektif dan tidak konsisten dalam penilaian penghilatan satu penilai dengan penilai lainnya sehingga mempengaruhi tingkat kualitas dalam penyortiran.

Seiring berjalannya waktu, banyak penelitian pendeteksian kematangan buah berdasarkan warna menggunakan *deep learning* dapat dilihat dari beberapa penelitian sebelumnya yang menggunakan metode yang ada pada *deep learning* untuk mengklasifikasi tingkat kematangan buah berdasarkan warna yaitu Fahmil Hidayat

(Hidayat et al., 2017), Andi Baso Kaswar (Andi Baso Kaswar et al., 2020), Arvi Arkadia (Arkadia et al., 2021) dan Rhama Permadi Ahmad (Ahmad, 2020). Pada penelitian-penelitian sebelumnya dalam pendekteksian menggunakan *deep learning* metode *artificial neural network* dan metode *convolutional neural network* bertujuan sebagai pengganti kinerja otak manusia dalam mengenali kelas tingkat kematangan pada buah. Pada penelitian-penelitian tersebut menjelaskan *neural network* akan menerima informasi dari nodes atau informasi pada titik-titik yang terkumpul pada satu layer lalu diteruskan pada hidden layer. Dimana metode *artificial neural network* nodes terpisah satu sama lain sedangkan pada metode *convolutional neural network* nodes akan saling terhubung membuat metode *convolutional neural network* lebih hemat daya dalam komputasi atau pemrosesan informasi. Dalam hal akurasi untuk penelitian menggunakan *deep learning* metode *artificial neural network* menghasilkan akurasi rata-rata dibawah 90% dan untuk penelitian menggunakan *deep learning* metode *convolutional neural network* menghasilkan nilai rata-rata diatas 90%. Hal tersebut menunjukkan metode *convolutional neural network* memiliki kemampuan untuk mengklasifikasi kematangan buah lebih efisien.

Perkembangan teknologi yang pesat memunculkan teknologi *computer vision* yang dapat melihat dan menganalisis seperti layaknya mata manusia (Bahtiar, 2016). Beberapa penelitian sebelumnya oleh Oktaviano Yudha (Yudha, 2013) dan Nevalen Aginda Prasetyo (Prasetyo et al., 2021) menggunakan *camera* sebagai media *computer vision* untuk mengenali tingkat kematangan buah berdasarkan warna dan memiliki tingkat akurasi yang cukup tinggi dimana pada skripsi Oktaviano Yudha menghasilkan akurasi rata-rata diatas 70% dan pada penelitian Nevalen Aginda Prasetyo menghasilkan akurasi sebesar 98,4%. Hal tersebut menunjukkan penggunaan *computer vision* dapat digunakan dalam pengklasifikasian kematangan buah dan menghasilkan nilai akurasi yang tinggi.

Pada permasalahan diatas untuk mengurangi tenaga yang dibutuhkan dalam menyortir kematangan buah jambu kristal dan mendapatkan nilai tingkat kematangan yang pasti sehingga tidak menimbulkan penilaian subjektif dan penglihatan yang tidak konsisten terhadap kematangan buah jambu kristal guna meningkatkan kualitas dalam penyortiran juga dalam mendapatkan nilai akurasi yang tinggi maka, dilakukan penggabungan *deep learning* menggunakan metode *convolutional neural network* dengan *computer vision* dimana *deep learning* metode *convolutional neural network* sebagai pengganti kinerja otak manusia dan memiliki tingkat akurasi tinggi dan *computer vision* sebagai pengganti kinerja visual manusia.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, dapat dirumuskan permasalahan dari penelitian ini adalah mengetahui akurasi pada *deep learning* metode *convolutional neural network* dengan *computer vision* dalam mengklasifikasi kematangan jambu kristal.

1.3. Batasan Masalah

Setelah didapatkan rumusan masalah dibuatlah batasan permasalahan agar masalah yang dibahas menjadi lebih terfokus dan jelas. Batasan masalah penelitian ini adalah sebagai berikut:

1. Data citra diambil dari Kebun Buah Larasari Garden, Kabupaten Sukoharjo.
2. Mendeteksi kematangan jambu kristal berdasarkan warna.
3. Data gambar diambil pada jarak 10cm - 15cm.
4. Data gambar diambil dalam kondisi pencahayaan menggunakan *ring light* LED dengan cahaya *natural light* tingkat pencahayaan penuh.
5. Saat pengambilan data gambar menggunakan sisi buah yang memiliki warna mayoritas pada suatu tingkat kematangan.
6. Menggunakan kamera webcam sebagai penangkap gambar.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah mendapatkan hasil akurasi yang baik pada *computer vision* menggunakan *deep learning convolutional neural network* dalam mengklasifikasikan kematangan buah jambu kristal.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dengan dilakukannya penelitian ini yaitu agar dapat membantu petani jambu dan konsumen dalam mempermudah membandingkan tingkat kematangan jambu. Dan juga diperoleh suatu cara untuk melakukan klasifikasi kematangan buah jambu kristal selain cara manual, yaitu klasifikasi dengan bantuan teknologi, seperti komputer.

Selain itu, dapat memberikan informasi kepada akademisi sebagai referensi penelitian mengenai klasifikasi tingkat kematangan buah jambu kristal menggunakan *Computer Vision* menggunakan metode *Convolutional Neural Network*.

1.6. Metodologi Penelitian

Dalam penyusunan laporan tugas akhir ini metode penelitian menggunakan metode kuantitatif dengan tahapan:

1.6.1. Studi Pustaka

Pada tahap ini dilakukan studi terkait penelitian, jurnal, dan buku yang membahas mengenai buah jambu. Serta mempelajari hal-hal yang berhubungan dengan objek penelitian dan metode yang digunakan pada penelitian.

1.6.2. Wawancara

Dilakukan tahap wawancara kepada petani dan pemilik kebun jambu kristal mengenai kriteria panen buah jambu kristal.

1.6.3. Pengumpulan Data

Data yang didapatkan berasal dari Kebun Buah Larasati, Kecamatan Polokarto, Kabupaten Sukoharjo. Data dikumpulkan dengan cara mengambil citra menggunakan

kamera kemudian dipisahkan berdasarkan tingkat kematangan buah jambu kristal dan diberi label.

1.6.4. Pengolahan Data

Data yang sudah dikumpulkan kemudian di-*labelling* untuk mendapatkan anotasi dan dengan nilai .xml, lalu nilai .xml dikonversikan ke nilai .csv untuk mentrasfer data antar program. Kemudian dipisahkan menjadi dua yaitu data *training* dan data *testing* sebesar 80% untuk data training dan 20% untuk data testing.

1.7. Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini adalah metode *waterfall*. Tahapan dalam metode *waterfall* (Pressman, 2010).

1. Communication (Komunikasi)

Pada tahap komunikasi, pengumpulan data awal dengan menganalisis kebutuhan sistem (*analysis requirement*) dan mengidentifikasi permasalahan yang ada.

2. Planning (Perencanaan)

Pada tahap ini dilakukan dengan penentuan sumberdaya, spesifikasi dalam pengembangan berdasarkan kebutuhan sistem, dan tujuan berdasarkan hasil dari tahap komunikasi yang dilakukan agar pengembangan sesuai dengan yang diinginkan.

3. Modelling (Pemodelan)

Pada tahap ini melakukan penerjemahan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Tahapan ini akan menghasilkan *software requirement* yang didalamnya terdapat rancangan struktur data, arsitektur *software*, representasi interface, dan detail (algoritma) prosedural.

4. Construction (Konstruksi)

Pada tahap ini dilakukan *coding* dimana tahap ini adalah tahap nyata dalam mengerjakan suara *software* atau sistem dikarenakan programmer menerjemahkan desain kedalam bahasa yang dapat dikenali komputer.

5. Deployment (Penyerahan sistem)

Tahap ini adalah tahapan akhir dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan *coding* maka *software* atau sistem akan digunakan oleh *user* dan dievaluasi dan dilakukan pemeliharaan secara berkala.

1.8. Sistematika Penulisan

Sistematika penulisan dalam penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi mengenai latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini berisi mengenai teori-teori yang berhubungan dengan pembahasan dari penelitian yang menjadi landasan berpikir serta dasar penyusunan penelitian. Teori-teori

tersebut didapat dari buku-buku referensi, penelitian sebelumnya, serta sumber-sumber informasi lainnya. Mencakup teori mengenai jambu kristal, konsep dasar dari metode *Convolutional Neural Network* sebagai metode dalam klasifikasi citra digital buah jambu dan *Computer Vision* sebagai alat yang digunakan untuk mengambil hasil citra digital, serta konsep dasar pendukung pembangunan sistem lainnya.

BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

Bab ini berisi tahapan metodologi penelitian dan pengembangan sistem yang digunakan dalam perancangan sistem. Bab ini juga menjelaskan tentang proses analisis data dan *design*, logika pembuatan program serta antarmuka pada sistem.

BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN

Bab ini menguraikan tentang implementasi dari analisis dan perancangan sistem menjadi sebuah perangkat lunak serta pengujian dan Analisa terhadap hasil yang diperoleh pada tampilan implementasi sistem.

BAB V PENUTUP

Pada bab penutup berisi mengenai kesimpulan (rangkuman keseluruhan isi yang sudah dibahas), dan saran (saran perluasan, pengembangan, dan pengkajian ulang).

BAB II TINJAUAN PUSTAKA

2.1. Jambu Kristal

Klasifikasi botani tanaman jambu kristal adalah sebagai berikut:

- Kingdom : *Plantae*
- Divisi : *Spermatophyta*
- Subdivisi : *Angiospermae*
- Kelas : *Dicotyledonae*
- Ordo : *Myrtales*
- Famili : *Myrtaceae*
- Genus : *Psidium*
- Spesies : *Psidium guajava L. Merr*

(Cahyo, 2016).



Gambar 2.1. Jambu Kristal (Cahyo, 2016)

Jambu kristal adalah varietas dari buah jambu biji (*Psidium Guajava L.*) merupakan hasil transformasi dari jambu Bangkok. Tanaman ini ditemukan oleh dua petani, Xi-Yao Lai dan Jiang-Ming Dong di Yanchao District, Kaohsiung, Taiwan pada 1991. Kaohsiung terkenal sebagai sentra hortikultura. Di Taiwan, jambu kristal dikenal dengan nama shuijing ba (shui-jing=kristal) (Wang, 2011). Buah jambu kristal diperkenalkan di Indonesia pada tahun 1991 melalui Misi teknik Taiwan yang bekerja sama dengan Institut Pertanian Bogor (IPB).

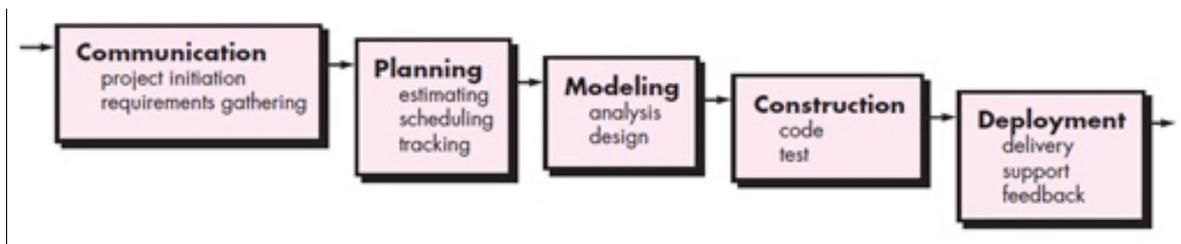
Jambu kristal merupakan kultivar jambu biji yang telah resmi dilepas oleh Kementerian Pertanian berdasarkan SK Mentan No.540/Kpts/SR.120/9/2007. Jambu kristal adalah jenis buah perdu yang berbuah sepanjang tahun secara menerus-menerus, berdaun selalu berwarna hijau, memiliki ketinggian pohon yang pada umumnya memiliki ketinggian 2 m – 2,5 m dan berdiameter 7,5 cm, memiliki banyak ranting, batang berbentuk silindris dan berwarna coklat (Balitbu, 2007).

Buah jambu kristal memiliki jumlah biji yang sangat sedikit yakni hanya 3% dari bagian buah. Bentuk buahnya bulat dan besar sedikit gepeng dengan warna kulit hijau muda ketika mulai matang dan menjadi hijau kekuningan ketika sudah matang. Warna dagingnya putih dengan tekstur renyah saat hampir matang dan empuk saat dipuncak kematangan. Kulitnya tipis, namun memiliki lapisan lilin yang tebal. Lapisan lilin membuat buah sulit ditembus hama (Cahyo, 2016).

Hampir seluruh bagian tanaman jambu kristal dapat dimanfaatkan. Tanaman jambu kristal memiliki kayu yang halus dan sangat padat sehingga baik bila digunakan sebagai ukiran atau patung bernilai tinggi. Buah jambu kristal dapat dikonsumsi dalam keadaan segar dan dapat pula diolah menjadi the, sirup, sari buah, nektar, buahvita, jeli, selai, kembang gula dan dodol. Buah jambu kristal juga dapat digunakan untuk resep pengobatan seperti diare, disentri, demam berdarah, gusi bengkak, sariawan, jantung dan diabetes (Saepudin, 2019).

2.2. Metode Waterfall

Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “Linear Sequential Model”, dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modelling*), konstruksi (*construction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan. (Pressman, 2010).



Gambar 2.2. Metode Waterfall (Pressman, 2010)

Berikut tahapan dalam metode *waterfall*:

1. *Communication* (Komunikasi)

Pada tahap komunikasi, pengumpulan data awal dengan menganalisis kebutuhan sistem (*analysis requirement*) dan mengidentifikasi permasalahan yang ada.

2. *Planning* (Perencanaan)

Pada tahap ini dilakukan dengan penentuan sumberdaya, spesifikasi dalam pengembangan berdasarkan kebutuhan sistem, dan tujuan berdasarkan hasil dari tahap komunikasi yang dilakukan agar pengembangan sesuai dengan yang diinginkan.

3. *Modelling* (Pemodelan)

Pada tahap ini melakukan penerjemahan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Tahapan ini akan menghasilkan *software requirement* yang didalamnya terdapat rancangan struktur data, arsitektur

software, representasi interface, dan detail (algoritma) prosedural.

4. *Construction* (Konstruksi)

Pada tahap ini dilakukan *coding* dimana tahap ini adalah tahap nyata dalam mengerjakan suara *software* atau sistem dikarenakan programmer menerjemahkan desain kedalam bahasa yang dapat dikenali komputer.

5. *Deployment* (Penyerahan sistem)

Tahap ini adalah tahapan akhir dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan *coding* maka *software* atau sistem akan digunakan oleh *user* dan dievaluasi dan dilakukan pemeliharaan secara berkala.

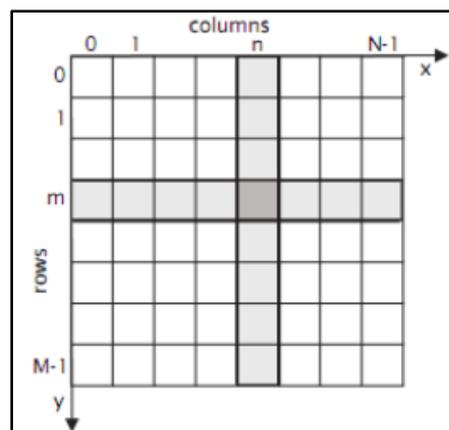
2.3. Citra Digital

2.3.1. Definisi Citra Digital

Citra dapat didefinisikan sebagai fungsi $f(x, y)$ kontinu dengan intensitas cahaya pada bidang dua dimensi dimana nilai x dan y menyatakan koordinat spasial dan elemen matriksnya menyatakan tingkat keabuan pada koordinat tersebut. Citra digital adalah citra $f(x, y)$ yang setiap elemennya direpresentasikan secara numerik dengan nilai-nilai diskrit (Kusumanto, 2011). Elemen citra digital disebut *piksel* (*picture elements*).

2.3.2. Representasi Citra Digital

Sebuah citra digital dapat diwakili dengan sebuah matriks dua dimensi $f(x, y)$ yang terdiri dari N kolom dan M baris. Setiap nilai *piksel* diwakili oleh dua buah bilangan bulat yang menunjukkan lokasi *piksel* dalam bidang citra seperti digambarkan pada Gambar 2.7. Pada gambar tersebut, koordinat $f(0, 0)$ digunakan untuk pojok kiri atas citra dan koordinat $f(N-1, M-1)$ digunakan untuk pojok kanan bawah dalam citra yang berukuran $N \times M$ *piksel*. Contoh representasi citra $f(x, y)$ dalam dua dimensi seperti yang ditunjukkan pada gambar 2.3.



Gambar 2.3. Contoh Representasi Suatu Citra $f(x, y)$ Dalam 2 Dimensi (Kusumanto, 2011)

2.3.3. Pengolahan Citra Digital

Pengolahan citra digital adalah suatu bidang ilmu yang mempelajari tentang teknik-teknik mengolah citra digital menggunakan komputer (Kusumanto, 2011). Karena pengolahan citra digital dilakukan menggunakan komputer, maka citra terlebih dahulu

ditransformasikan kedalam bentuk besaran diskrit berupa matriks dua dimensi yang mencakup tingkat keabuan serta lokasi *piksel*. Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi tidak akan sama persis seperti aslinya yang dipengaruhi oleh beberapa faktor. Faktor-faktor tersebut merupakan penurunan kualitas atau degradasi yang dapat berupa tingkat kontras yang terlalu tinggi atau terlalu rendah, efek distorsi, efek kabur (*Blur*), *noise* atau gangguan yang disebabkan oleh interfensi pengolah data citra maupun pada perangkat elektronik yang digunakan untuk mengambil data citra (Putri, 2016).

Teknik dan proses untuk mengurangi atau menghilangkan efek degradasi pada objek citra meliputi teknik perbaikan atau peningkatan citra (*image enhancement*), restorasi citra (*image restoration*) dan transformasi spesial (*special transformation*). Subjek lain dari pengolahan citra antara lain pengkodean citra, segmentasi citra, representasi dan deskripsi citra (Putri, 2016).

2.4. Computer Vision

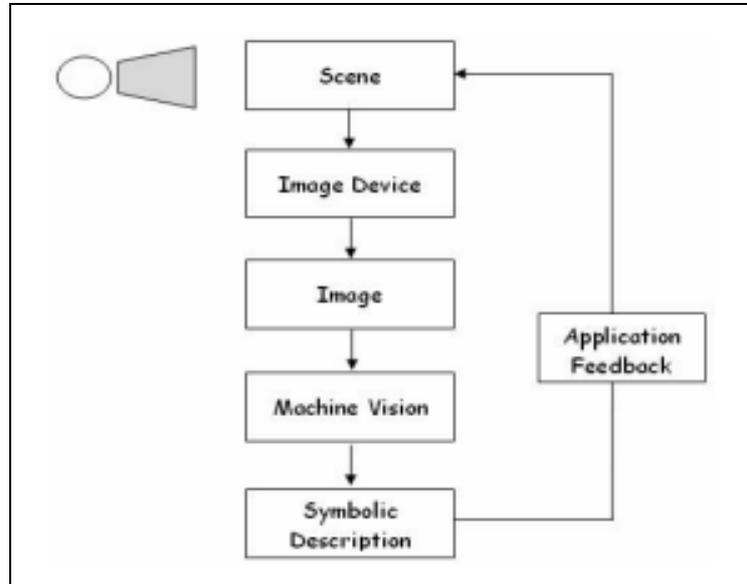
Computer Vision adalah suatu bidang yang mempelajari bagaimana komputer atau mesin dapat melihat dan menganalisis seperti layaknya mata dan otak manusia. Bidang dalam *computer vision* mencakup beberapa aspek penelitian yang antara lain bagaimana memperoleh, mengolah menganalisis, dan memahami data visual (citra atau video) bahkan dalam mengambil keputusan (Szeliski, 2010). *Computer vision* adalah kombinasi antara:

1. *Image Processing* atau Pengolahan citra, bidang ini membantu memperbaiki kualitas gambar agar komputer memahami dan menganalisis gambar lebih efisien.
2. *Pattern Recognition* atau Pengenalan pola, bidang ini melakukan proses identifikasi objek pada citra atau interpretasi citra dengan tujuan untuk mengekstrak informasi/pesan yang disampaikan oleh gambar.

Computer vision merupakan cabang dari teknik *artificial intelligence* yang mencoba meniru cara kerja sistem visual manusia. *Computer vision* tujuan utama untuk menganalisis objek fisik yang nyata berdasarkan image yang ditangkap dari sensor. Dalam prosesnya *computer vision* memiliki 4 tahapan yaitu:

1. *Image Acquisition*
Proses ini melakukan penangkapan informasi visual dan merubah sinyal analog menjadi data digital yang kemudian diproses oleh komputer.
2. *Image Processing*
Proses yang memiliki input dan output-nya berupa citra. Suatu citra dimodifikasi ke bentuk citra lain yang memiliki kualitas lebih baik.
3. *Image Analysis*
Proses yang memiliki input berupa citra dengan output bukan citra akan tetapi berupa hasil pengukuran terhadap citra tersebut.
4. *Image Understanding*
Proses ini merupakan proses yang memiliki input berupa citra dengan output-nya adalah gambaran tingkat tinggi dari citra tersebut.

Dalam menjalankan prosesnya *computer vision* memiliki 7 struktur yang melandasi elemen-elemen suatu *computer vision*, yaitu *light sources*, *scene*, *image device*, *machine vision*, *symbolic description*, dan *application feedback*. Berikut gambaran dari struktur *computer vision* pada gambar 2.3.



Gambar 2.4. Struktur Computer Vision (Amrizal & Aini, 2011)

Adapun komponen dari *computer vision*:

1. *light source* yang merupakan sumber cahaya yang digunakan dalam *computer vision*.
2. *scene* adalah kumpulan objek-objek, *image device* merupakan alat yang digunakan untuk mengubah letak benda yang direpresentasikan.
3. *image* adalah benda/objek yang digunakan.
4. *machine vision* merupakan sistem yang menafsirkan gambar melalui ciri-ciri, pola maupun objek yang dapat ditelusuri oleh sistem.
5. *symbolic description* adalah sistem dapat menggambarkan kinerja sistem kedalam simbol-simbol yang telah dimengerti oleh sistem.
6. *application feedback* merupakan respon penerimaan gambar dari sistem kepada pengguna.

2.5. Image Processing

Image processing adalah sebuah transformasi input data yang mentah untuk membantu kemampuan komputasional dan pencari ciri serta mengurangi noise (derau) (Utnasari, 2018). *Image processing* adalah sebuah teknik untuk menemukan orientasi dari sebuah citra, untuk menghilangkan noise kemudian meningkatkan kualitas dari sebuah citra sehingga memudahkan dalam proses pengklasifikasian dan penelitian lebih lanjut tentang sebuah citra.

2.5.1. Resizing

Sering dijumpai citra dengan berbagai ukuran yang perlu diubah sesuai dengan ukuran yang diinginkan. Secara umum, diperlukan pemetaan dari sumber citra ke citra *resized* tujuan menjadi semulus mungkin (Bradsky & Kaehler, 2008).

2.5.2. Median Filter

Median adalah nilai tengah dari kumpulan data (Yuwono, 2015). Untuk mencari median dari kumpulan data yang ganjil maka:

$$x = \frac{n+1}{2} \dots\dots\dots (2.1)$$

Keterangan:

n = Jumlah data

x = Nilai baru median

Untuk median *filtering*, data yang digunakan untuk menghitung median terdiri dari kumpulan data yang ganjil. Hal ini disebabkan dengan jumlah data yang ganjil maka piksel yang akan diproses dapat berada ditengah. Pada median *filtering* digunakan matrik berdimensi N x N. Dari matrik tersebut, kemudian data yang ada diurutkan dan dimasukkan dalam sebuah matrik berukuran 1X (N X N). Hal ini berguna untuk mempermudah menemukan median dari kumpulan data yang telah urut tersebut Sebagai contoh jika diketahui suatu matrik berdimensi 3X3 yang berisi piksel utama dan piksel-piksel di sekitarnya:

9	5	5
3	8	5
2	1	4

Matrik di atas harus diurutkan terlebih dahulu dan dimasukkan dalam sebuah matrik yang berukuran 1X (3X3) atau 1X9.

1	2	3	4	5	5	5	8	9
---	---	---	---	---	---	---	---	---

Dari matrik di atas dicari nilai piksel yang baru dengan menggunakan perhitungan median, maka nilai mediannya adalah $x = 5$. Nilai 5 ini akan menggantikan nilai 8 sehingga piksel utamanya akan memiliki warna yang berbeda dengan sebelumnya.

2.5.3. Histogram

Informasi penting tentang konten gambar digital dapat Ketahui dengan membuat histogram citra. Histogram citra adalah grafik menggambarkan penyebaran nilai-nilai Intensitas piksel gambar atau bagian ditentukan dalam gambar. dari histogram frekuensi relatif terjadinya dapat dilihat (relatif) intensitas dalam gambar. Menghasilkan data dari histogram Ini akan digunakan untuk pengolahan citra. (Munir, 2004).

Histogram biasanya merupakan distribusi atau distribusi frekuensi. Histogram gambar didefinisikan sebagai distribusi nilai skala abu-abu gambar Suatu histogram h dari citra grayscale dengan derajat keabuan $[0, L-1]$ akan memiliki sebanyak L derajat keabuan. Misalkan terdapat citra grayscale 8 bit memiliki $L= 2^8 =256$.

2.5.4. Histogram Equalization

Histogram Equalization adalah proses perataan histogram citra dengan cara mengubah nilai keabuan baru yang berbeda dari aslinya melalui suatu proses transformasi. Perataan histogram merupakan salah satu metode perbaikan citra yang sering digunakan untuk menyebarkan nilai intensitas pada citra yang terlalu terang atau terlalu gelap dan memiliki histogram yang sempit sehingga diperoleh penyebaran histogram yang merata.

Langkah-langkah dalam melakukan perbaikan citra dengan metode Histogram Equalization sebagai berikut:

- a. Menghitung frekuensi (n_k) setiap derajat keabuan (k) pada citra input.
- b. Menormalisasi setiap frekuensi dengan membagi nilai tersebut dengan jumlah seluruh piksel menggunakan persamaan 2.2.

$$P_k = \frac{n_k}{n} \dots \dots \dots (2.2)$$

- c. Menghitung penjumlahan kumulatif dari histogram ternormalisasi, seperti pada persamaan 2.3.

$$v_k = \sum_{k=0}^{L-1} P_k \dots \dots \dots (2.3)$$

- d. Menghitung nilai derajat keabuan (K) yang baru dari tiap piksel (k) serta mengubah setiap nilai derajat keabuan pada setiap piksel dari citra input (k). Nilai derajat keabuan yang baru dari tiap piksel dituliskan seperti pada persamaan 2.4

$$K = [(L - 1)v_k] \dots \dots \dots (2.4)$$

2.6. Deep Learning

Deep Learning atau pembelajaran mendalam adalah salah satu teknik dari *machine learning* yang memanfaatkan banyak lapisan (*layer*) pengolahan informasi *non-linear* untuk melakukan pengenalan pola, ekstraksi fitur dan klasifikasi (Deng & Yu, 2013).

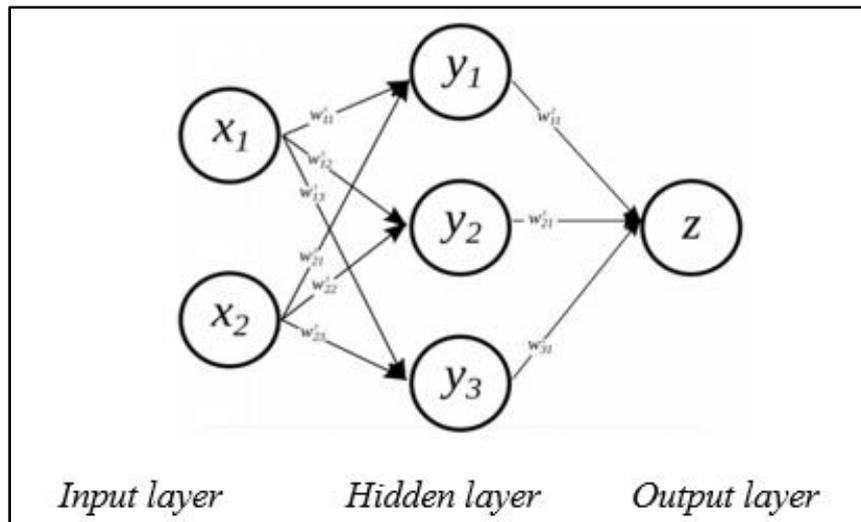
Deep Learning merupakan pengembangan dari *machine learning* yang telah lebih terarah pada *artificial intelligence* atau kecerdasan buatan. *Machine learning* memiliki keterbatasan kapasitas untuk pengenalan pola objek dan klasifikasi dalam memproses data dengan format aslinya, sehingga kemudian dikembangkan suatu teknik canggih untuk merepresentasikan pembelajaran lebih mendalam (Lecun et al., 2015).

Deep Learning memungkinkan model komputasi yang terdiri dari banyak lapisan untuk mempelajari representasi data dengan beberapa tingkatan abstraksi. Metode ini telah meningkatkan cakupan area penelitian secara signifikan seperti pengenalan suara, deteksi objek atau pengenalan objek visual. *Deep Learning* menemukan pola struktur yang rumit dalam suatu kumpulan data yang sangat besar menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana mesin harus mengubah bobot pada masing-masing parameter didalamnya. Bobot tersebut digunakan untuk menghitung representasi pada setiap lapisan dari representasi pada lapisan sebelumnya (Abadi et al., 2016).

2.7. Neural Network

Neural Network adalah suatu sistem pengenalan informasi dengan karakteristik yang sama dengan jaringan syaraf manusia (Fausett, 1994). Seperti halnya pada jaringan

syaraf manusia, informasi dalam *Neural Network* diproses pada elemen-elemen sederhana yang disebut *neuron*. Informasi yang berupa sinyal akan berpindah dari satu *neuron* ke *neuron* lainnya melalui sebuah penghubung. Setiap penghubung memiliki bobot yang akan mengubah sinyal yang ditransmisikan. Penentuan bobot tersebut dilakukan berdasarkan metode khusus yaitu pembelajaran atau pelatihan. Nilai dari bobot akan berubah seiring dengan proses pembelajaran atau pelatihan yang dilakukan sehingga sistem akan menghasilkan akurasi yang semakin baik dalam penyelesaian masalah. Setiap *neuron* memiliki fungsi aktivasi yang akan menentukan *output* dari *neuron*.



Gambar 2.5. Contoh Sederhana Arsitektur Neural Network (Ratnasari, 2018)

Pada gambar 2.5 adalah sebuah contoh sederhana arsitektur *Neural Network* yang terdiri dari 3 lapisan (*layer*) antara lain lapisan masukan (*Input Layer*), lapisan tersembunyi (*Hidden Layer*) dan lapisan keluaran (*Output Layer*) (Ratnasari, 2018). Pada gambar 2.5 terlihat bahwa *neuron-neuron* dalam masing-masing lapisan saling terhubung dengan bobot tertentu, misalkan *neuron* [x_1] pada *input layer* terhubung dengan *neuron* [y_1], [y_2] dan [y_3] pada *hidden layer* berbobot sejumlah [w_{11}], [w_{12}] dan [w_{13}]. Dalam mengimplementasikan konsep *neural network*, *hidden layer* dapat dibuat lebih dari dari satu lapisan yang membuat suatu *neural network* semakin dalam (Ratnasari, 2018).

Neural Network memiliki karakteristik yang membedakan satu dengan lainnya. Yang pertama adalah pola koneksi antar *neuron* atau biasa disebut arsitektur. Pembeda selanjutnya adalah metode yang digunakan dalam menentukan nilai bobot yang biasa disebut proses pembelajaran (*Data Training*). Pembeda terakhir adalah fungsi aktivasi yang digunakan.

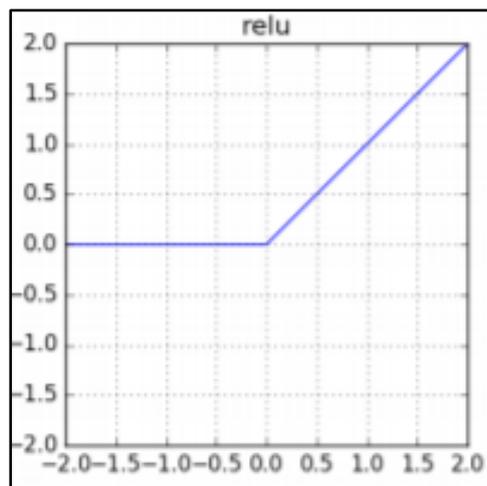
2.7.1. Activation Function

Activation Function atau fungsi aktivasi adalah suatu fungsi yang menunjukkan tingkat keaktifan suatu *neuron* dengan menerapkannya pada sinyal *input* terbobot. Nilai keaktifan hasil fungsi tersebut yang kemudian menjadi *output neuron* tersebut untuk diteruskan ke *neuron* lain pada lapisan selanjutnya atau menjadi penentu *output* sistem.

2.7.1.1. Rectified Linear Unit (ReLU)

Fungsi ReLU merupakan suatu fungsi aktivasi dengan perhitungan yang sederhana. Proses maju (*forward*) dan mundur (*backward*) dalam fungsi ReLU dapat dibuat hanya menggunakan kondisi *if*. Jika elemen x bernilai negatif, maka nilai akan diatur menjadi 0 dan tidak akan ada operasi eksponensial, perkalian atau pembagian.

Dengan karakteristik seperti yang sudah dijelaskan sebelumnya, fungsi ReLU memiliki kelebihan dalam hal waktu *training* dan *testing* data yang secara signifikan lebih cepat ketika berhadapan dengan jaringan yang memiliki *neuron* banyak.



Gambar 2.6. Grafik Fungsi ReLU (Novyantika, 2018)

Seperti yang dapat dilihat pada Gambar 2.6, ReLU setengah diperbaiki (dari bawah). Ini adalah $f(z)$ adalah nol bila z kurang dari nol dan $f(z)$ sama dengan z bila z di atas atau sama dengan nol. Rentang: (0 sampai tak terbatas) Fungsi dan turunannya keduanya bersifat monoton. Tapi masalahnya adalah semua nilai negatif menjadi nol segera yang menurunkan kemampuan model agar sesuai atau melatih dari data dengan benar. Itu berarti setiap masukan negatif yang diberikan pada fungsi aktivasi ReLU mengubah nilainya menjadi nol segera dalam grafik, yang pada gilirannya mempengaruhi grafik yang dihasilkan dengan tidak memetakan nilai negatif secara tepat (Novyantika, 2018).

2.7.1.2. Layer

Softmax layer merupakan lapisan yang biasanya menjadi lapisan terakhir dari sebuah *neural network* untuk mendapatkan output. Secara umum, *softmax layer* menerima input kemudian melakukan pembobotan dan penambahan *bias* dengan menerapkan fungsi *softmax*. Sehingga output pada *softmax layer* dapat dikatakan sebagai distribusi probabilitas.

2.8. Convolutional Neural Network

Pada tahun 1960an, ketika Hubel dan Wiesel meneliti neuron yang digunakan untuk orientasi sensitif lokal-selektif dalam sistem visual kucing, mereka menemukan bahwa struktur jaringan khusus dapat secara efektif mengurangi kompleksitas jaringan

saraf umpan balik dan kemudian mengusulkan *Convolutional Neural Network* (CNN). *Convolutional Neural Network* adalah algoritma pengenalan yang efisien dan banyak digunakan dalam pengenalan pola dan pengolahan citra. *Convolutional Neural Network* memiliki banyak fitur seperti struktur sederhana, parameter pelatihan yang kurang dan kemampuan beradaptasi. *Convolutional Neural Network* telah menjadi topik hangat dalam analisis suara dan pengenalan gambar. Bobotnya struktur jaringan bersama membuatnya lebih mirip dengan jaringan saraf biologis. Ini mengurangi kompleksitas model jaringan dan jumlah bobot (T. Liu et al., 2015).

Umumnya, struktur CNN mencakup dua lapisan satu adalah lapisan ekstraksi fitur, masukan setiap neuron terhubung ke bidang reseptif lokal dari lapisan sebelumnya, dan mengekstrak fitur lokal. Setelah fitur lokal diekstraksi, hubungan posisi antara itu dan fitur lainnya juga akan ditentukan. Yang lainnya adalah lapisan peta fitur, setiap lapisan komputasi jaringan terdiri dari sejumlah peta fitur. Setiap peta fitur adalah pesawat, berat neuron di pesawat sama. Struktur peta fitur menggunakan fungsi sigmoid sebagai fungsi aktivasi dari jaringan konvolusi, yang membuat peta fitur memiliki pergeseran *invariance*. Selain itu, karena neuron di pesawat pemetaan sama berbagi bobot, jumlah parameter bebas dari jaringan berkurang. Setiap lapisan konvolusi di jaringan syaraf konvolusi diikuti oleh lapisan komputasi yang digunakan untuk menghitung rata-rata lokal dan ekstrak kedua, dua lapisan ekstraksi fitur unik ini mengurangi resolusi. (T. Liu et al., 2015).

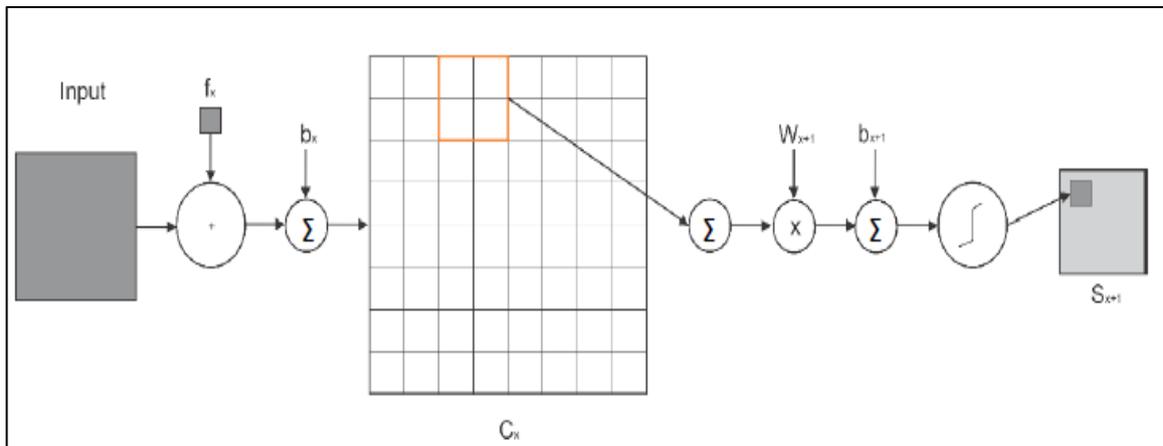
CNN terutama digunakan untuk mengidentifikasi perpindahan, *zoom* dan bentuk inversi grafis dua dimensi yang distortif. Karena lapisan deteksi fitur CNN belajar dengan data pelatihan, ia menghindari ekstraksi fitur eksplisit dan secara implisit belajar dari data pelatihan saat kita menggunakan CNN. Selanjutnya, neuron di bidang peta fitur yang sama memiliki bobot yang sama, sehingga jaringan bisa belajar secara bersamaan. Ini adalah keuntungan besar dari jaringan konvolusi sehubungan dengan jaringan neuron yang terhubung satu sama lain. Karena struktur khusus dari bobot berbagi lokal CNN membuatnya memiliki keunggulan unik dalam pengenalan suara dan pemrosesan gambar. Tata letaknya lebih dekat ke jaringan saraf sebenarnya. Bobot bersama mengurangi kompleksitas jaringan. Secara khusus gambar vektor input multi-dimensi dapat langsung masuk ke jaringan, yang menghindari kompleksitas rekonstruksi data dalam ekstraksi fitur dan proses klasifikasi (T. Liu et al., 2015).

2.8.1. Prinsip Convolutional Neural Network

Algoritma jaringan saraf konvolusi adalah *perceptron multilayer* yang merupakan desain khusus untuk identifikasi informasi gambar dua dimensi. Selalu memiliki lebih lapisan: lapisan masukan, lapisan konvolusi, lapisan sampel dan lapisan keluaran. Selain itu, dalam arsitektur jaringan yang dalam, lapisan konvolusi dan lapisan sampel dapat memiliki banyak. CNN tidak seperti mesin boltzmann yang dibatasi, harus sebelum dan sesudah lapisan neuron di lapisan yang berdekatan untuk semua koneksi, algoritma jaringan syaraf tiruan, setiap neuron tidak perlu melakukan citra global, hanya merasakan area lokal dari gambar. Selain itu, setiap parameter neuron disetel sama, yaitu pembagian

bobot, masing-masing neuron dengan kernel konvolusi yang sama dengan citra dekonvolusi (T. Liu et al., 2015).

Algoritma CNN memiliki dua proses: konvolusi dan sampling. Proses konvolusi: gunakan filter yang dapat dilatih f_x , dekonvolusi gambar masukan (Tahap pertama adalah gambar input, masukan dari konvolusi setelah adalah fitur gambar masing-masing *layer*, yaitu *Feature Map*), lalu tambahkan bias b_x , kita bisa mendapatkan lapisan konvolusi C_x . Proses sampling: n piksel setiap lingkungan melalui langkah penyatuan, menjadi piksel, dan kemudian dengan bobot skalar $W_x + 1$ tertimbang, tambahkan bias $b_x + 1$, dan kemudian oleh fungsi aktivasi, menghasilkan n kali lipat fitur peta $S_x + 1$ (T. Liu et al., 2015).

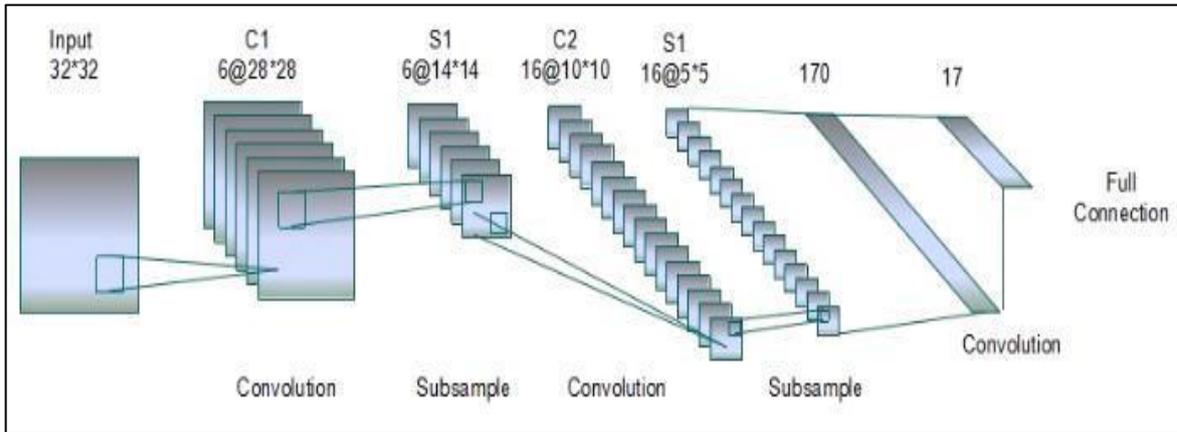


Gambar 2.7. Main Process of CNN (T. Liu et al., 2015)

Teknologi kunci CNN adalah *field* reseptif lokal, pembagian bobot, sub sampling berdasarkan waktu atau ruang, sehingga bisa mengekstrak fitur dan mengurangi ukuran parameter pelatihan. Keuntungan algoritma CNN adalah untuk menghindari ekstraksi fitur eksplisit, dan secara implisit untuk belajar dari data pelatihan. Bobot neuron yang sama pada permukaan pemetaan fitur, sehingga jaringan dapat dipelajari secara paralel, mengurangi kompleksitas jaringan. Mengadopsi struktur sub sampling berdasarkan waktu atau ruang, dapat mencapai tingkat ketahanan tertentu, skala dan deformasi perpindahan; Input informasi dan topologi jaringan bisa menjadi pertandingan yang sangat baik, Ini memiliki keunggulan unik dalam pengenalan suara dan pengolahan gambar (T. Liu et al., 2015).

2.8.2. Desain Arsitektur Convolutional Neural Network

Algoritma CNN membutuhkan pengalaman dalam desain arsitektur, dan perlu melakukan debug tanpa henti dalam aplikasi praktis, agar bisa mendapatkan arsitektur aplikasi CNN yang paling sesuai. Berdasarkan contoh pada gambar abu-abu dengan input 96x96. Pada tahap *preprocess*, mengubahnya menjadi 32x32 dari ukuran gambar. Desain kedalaman lapisan 7 model konvolusi: lapisan input, lapisan konvolusi C1, lapisan sub sampling S1, lapisan konvolusi C2, lapisan sampling S2, lapisan tersembunyi H dan lapisan keluaran F (T. Liu et al., 2015).

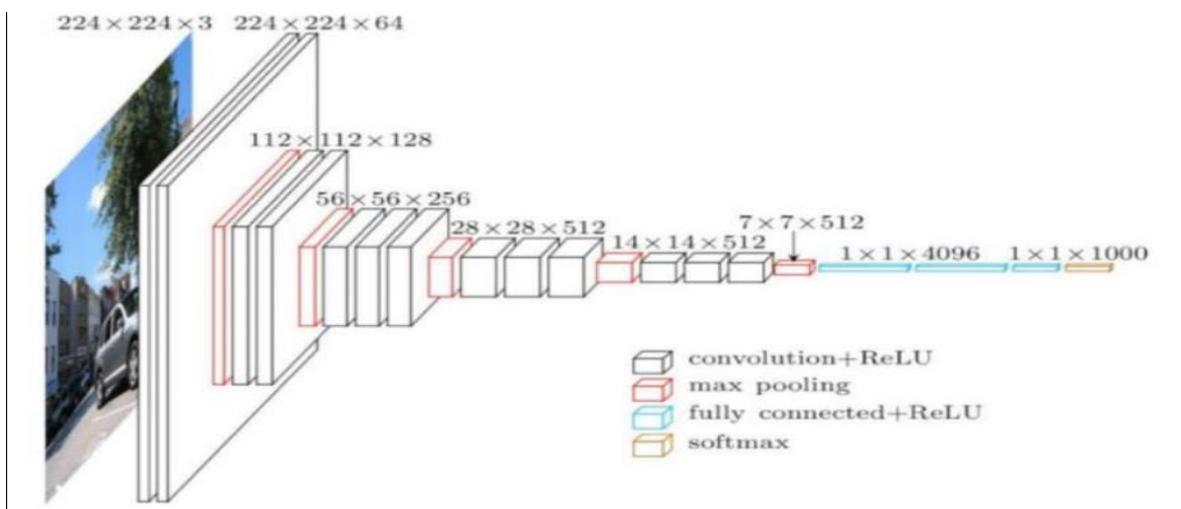


Gambar 2.8. Architecture of CNN in Training Face (T. Liu et al., 2015)

2.8.3. Arsitektur VGG

Jaringan VGG dibuat oleh peneliti dari *Oxford Visual Group*, Karen Simonyan dan Andrew Zisserman. Ada 2 jenis VGG yaitu VGG-16 yang memiliki kedalaman 16 layer dan VGG-19 dengan 19 layer. Jaringan ini terdiri dari lapisan konvolusi 3x3, selain lapisan *max pooling* yang digunakan untuk mengurangi ukuran volume dan lapisan terakhir adalah lapisan *fully connected* dengan 4096 neuron diikuti oleh lapisan *softmax preprocessing* yang dilakukan hanya pada input adalah pengurangan nilai rata-rata RGB yang dihitung pada training set setiap piksel. Proses pooling dilakukan dengan *max pooling* layer yang diikuti oleh beberapa lapisan konvolusi. Tidak semua lapisan konvolusi diikuti oleh *max pooling*. *Max pooling* dilakukan dengan *2x2 pixel window* dan *2 stride*. Aktivasi ReLU digunakan untuk setiap lapisan tersembunyi. Jumlah filter bertambah sesuai dengan kedalaman varian varian VGG (Shanmugamani, 2018).

VGG-16 adalah model yang terdiri dari 16 lapisan konvolusi dan *fully connected* yang biasanya digunakan untuk mengenali dan mengklasifikasikan gambar. Arsitektur VGG-16 memiliki 13 lapisan yang mewakili filter konvolusi 3x3 dengan lapisan *max pooling* untuk *downsampling*. VGG-16 juga memiliki dua lapisan yang terhubung penuh di lapisan tersembunyi (*hidden layer*) dengan total 4096 unit lapisan diikuti oleh lapisan *dense* 1000 unit, di mana setiap unit mewakili satu kategori gambar dalam *database ImageNet* (Shanmugamani, 2018). Berikut ilustrasi arsitektur VGG-16 dijelaskan pada Gambar 2.9.



Gambar 2.9. Arsitektur VGG-16 (Hindarto & Santoso, 2019)

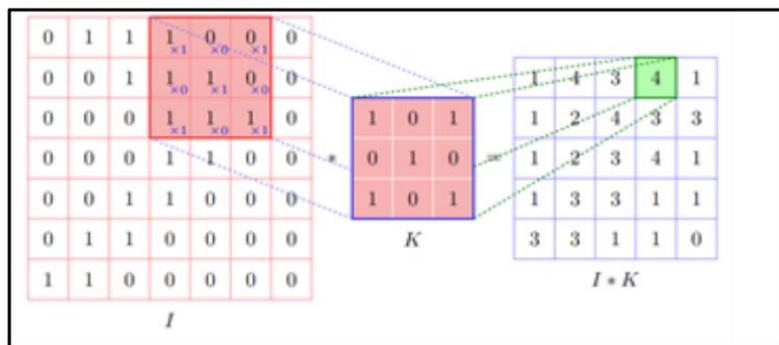
2.8.4. Convolutional Layer

Lapisan konvolusi atau *convolution layer* dalam CNN bertindak sebagai lapisan yang mengekstrak fitur, sekaligus mempelajari representasi fitur dari gambar yang di-*input*. *Neuron* dalam lapisan konvolusi tersusun membentuk *feature map* yang memiliki panjang dan tinggi (piksel). Setiap *neuron* dalam sebuah *feature map* memiliki sebuah penerima sinyal, dimana penerima sinyal tersebut terhubung dengan sekumpulan *neuron* dari lapisan sebelumnya yang telah memiliki bobot untuk melakukan *training* dan dalam beberapa kasus *neuron* tersebut ditunjuk sebagai patokan (Lecun et al., 2015).

Input tergabung dengan bobot yang telah di-*training* untuk melakukan *feature map* baru, dan hasil penggabungan tersebut dikirim melalui suatu fungsi aktivasi *non-linear*. Semua *neuron* dalam *feature map* akan memiliki bobot yang disamakan. Akan tetapi, *feature map* yang berbeda dalam lapisan konvolusi yang sama akan memiliki bobot yang berbeda pula, sehingga beberapa fitur dapat diekstrak dilokasi masing-masing *feature map* (Lecun et al., 2015). Fungsi aktivasi *non-linear* memungkinkan ekstraksi fitur *non-linear*. Fungsi aktivasi *non-linear* yang paling sering digunakan adalah fungsi *sigmoid*, *tanh* dan ReLU. Dalam pembentukan *feature map* dapat dituliskan dalam persamaan sebagai berikut:

$$FM[i]_{j,k} = (\sum_m \sum_n N_{(j-m,k-n)} F_{(m,n)}) + bF \dots\dots\dots(2.5)$$

Dimana pada persamaan (2.1) FM[i] adalah matriks feature map ke-I, N adalah matriks citra masukan, F adalah matriks filter konvolusi, bF adalah nilai bias pada filter, j dan k adalah posisi piksel pada matriks citra masukan serta m dan n adalah posisi piksel pada matriks filter konvolusi. Contohnya adalah *convolution layer* dengan ukuran panjang 3 piksel dan tinggi 3 piksel (3 x 3). Filter ini lalu digeser ke seluruh bagian gambar. Setiap filter ini bergeser sesuai dengan *strides*. Jika *strides* sebanyak 2 menunjukkan filter bergeser 2 piksel ke kanan dan ke baris selanjutnya sebanyak 2 kali. Dapat diilustrasikan dengan Gambar 2.10.



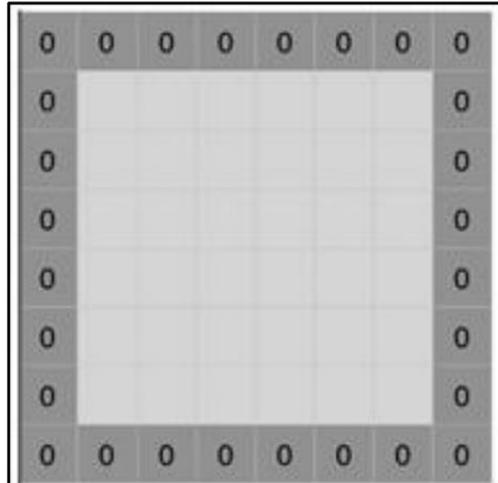
Gambar 2.10. Ilustrasi Proses Konvolusi

Dari Gambar 2.9 FM(0,3) sehingga didapat perhitungan sebagai berikut:

$$FM(0,3) = 1 \times 1 + 0 \times 0 + 0 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 1 \times 0 + 1 \times 1 = 4$$

2.8.5. Padding

Padding atau zero padding adalah parameter (piksel) menggunakan nilai 0 yg dibubuhi dalam sisi gambaran masukan buat memanipulasi dimensi menurut keluaran convolutional layer. Penambahan piksel pada setiap sisi berfungsi ketika proses konvolusi nir mengurangi dimensi masukan supaya ekstraksi fitur yg didapat lebih luas dan dalam. Penerapan padding dapat dilihat pada Gambar 2.11.



Gambar 2.11. *Padding*

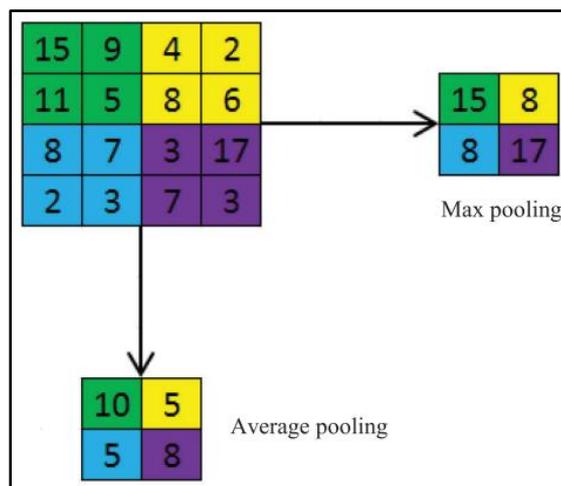
Dalam Gambar 2.10 memiliki pengertian *padding* bernilai 1. Dalam penerapan *padding* memiliki persamaan sebagai berikut:

$$n_{out} = \left(\frac{n_{in} - k + 2p}{s} \right) + 1 \dots \dots \dots (2.6)$$

Dimana pada persamaan (2.3) n_{out} adalah ukuran *feature map*, n_{in} adalah ukuran matriks masukan, k adalah ukuran matriks filter, p adalah ukuran *padding*, s adalah *stride*. Contohnya adalah saat melakukan konvolusi dengan ukuran matriks 433 x 433 dengan matriks filter 3 x 3, *padding* 1 dan *stride* 1, maka akan menghasilkan ukuran *feature map* yang sama yaitu 433 x 433.

2.8.6. Pooling Layer

Pooling layer dilakukan untuk mengurangi varians, mengurangi kerumitan perhitungan dan ekstrak fitur tingkat rendah dari lingkungan sekitar. *Pooling layer* terdapat dua jenis yaitu *Max pooling* dan *Average pooling*. *Max pooling* mengekstrak fitur yang paling penting seperti tepi sedangkan *average pooling* lebih halus dibandingkan dengan *max pooling* (Novyantika, 2018).



Gambar 2.12. Perbedaan *Average Pooling* dan *Max Pooling* (Rawat & Wang, 2017)

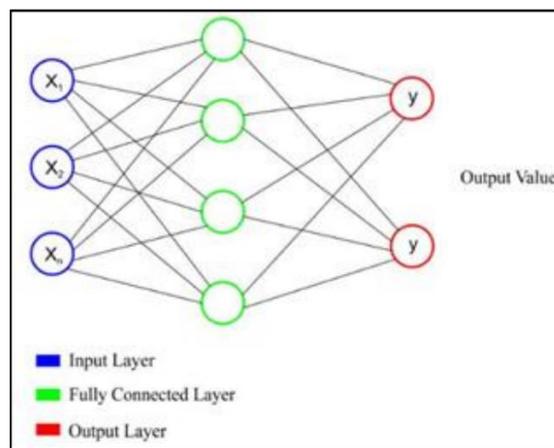
Pada Gambar 2.12 dapat dilihat bahwa *input* yang diberikan adalah gambar dengan ukuran 4 x 4, dengan menerapkan filter dan *stride* berukuran 2 x 2. Hasil *average pooling*

yang ditunjukkan pada gambar adalah nilai rata-rata yang dibulatkan dari setiap daerah sub sampel, sedangkan hasil *max pooling* adalah nilai maksimal dari setiap daerah sub sampel.

Meskipun keduanya digunakan untuk alasan yang sama, *max pooling* lebih baik untuk mengekstrak fitur ekstrem. *Average pooling* kadang-kadang tidak dapat mengekstrak fitur yang baik karena dibutuhkan semua untuk menghitung dan menghasilkan nilai rata-rata yang mungkin atau tidak penting untuk deteksi objek. *Average pooling* membawa semua ke dalam hitungan dan mengalirkannya ke lapisan berikutnya yang berarti semua nilai benar-benar digunakan untuk pemetaan fitur dan pembuatan keluaran yang merupakan perhitungan yang sangat umum. Jika tidak diperlukan semua masukan dari lapisan konvolusi maka akan mendapatkan akurasi yang buruk untuk *average pooling*. Pada klasifikasi objek *pooling layer* yang banyak digunakan adalah *average pooling*. Pemilihan *pooling layer* tergantung pada jenis dataset yang digunakan (Novyantika, 2018).

2.8.7. Fully Connected Layer

Feature map yang didapatkan sesudah melewati convolution layer berbentuk *multidimensional array*, sebagai akibatnya *feature map* tadi melalui proses *flatten* sebelum masuk ke termin *fully connected layer*. *Flatten* kemudian membuat sebuah vektor yang nantinya dipakai menjadi inputan berdasarkan *fully connected layer*. Setiap lapisan sebelumnya yang masuk ke *fully connected* perlu diubah ke pada matriks satu dimensi sebelum dihubungkan ke seluruh neuron pada lapisan buat diolah supaya bisa diklasifikasikan. Di dalam *fully connected layer* memiliki beberapa *hidden layer*, *action function*, *output layer*, dan *loss function* (Santoso & Ariyanto, 2018).



Gambar 2.13. Fully Connected Layer

2.8.8. Training Data

CNN secara umum menggunakan algoritma pembelajaran untuk menentukan parameter lepasnya; yaitu bias, bobot dan lain sebagainya, untuk mencapai hasil yang diinginkan. Algoritma pembelajaran yang paling umum digunakan adalah *backpropagation* (Deng & Yu, 2013). *Backpropagation* memperhitungkan gradien yang terdapat dalam fungsi objektif untuk menentukan bagaimana cara mengatur parameter dari jaringan tersebut supaya meminimalisir error yang berpengaruh pada performansi sistem (Rawat & Wang, 2017). Permasalahan yang biasa muncul dalam pelatihan data CNN

adalah *overfitting*, yang memiliki performa buruk ketika mencoba sistem dengan *test set* setelah melakukan pelatihan terhadap *dataset* yang kecil atau bahkan *dataset* yang besar. Permasalahan *overfitting* dapat mempengaruhi kemampuan model CNN dalam menyamaratakan data yang tidak terlihat (Rawat & Wang, 2017).

2.9. Object Detection

Object detection atau yang biasa disebut dengan deteksi objek merupakan cara computer untuk mendeteksi suatu objek yang terdapat pada gambar maupun video. Deteksi objek adalah tugas mengambil gambar *input* dan mengeluarkan label (matang, belum matang dll) atau kemungkinan label yang paling mirip dengan gambar. Bagi manusia, tugas pengenalan ini adalah salah satu kemampuan pertama yang dipelajari sejak saat dilahirkan dan merupakan salah satu yang datang secara alami dan mudah sebagai orang dewasa. Hanya saja kemampuan ini tidak serta merta dimiliki komputer. Apa yang manusia lihat dan komputer lihat jelas saja berbeda. Deteksi objek sendiri dapat dibagi menjadi dua, yaitu *soft detection* dan *hard detection*. *Soft detection* hanya mendeteksi adanya objek saja, sedangkan *hard detection* mendeteksi keberadaan dan lokasi objek (Jalied & Voronkov, 2016).

2.10. Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* juga didukung oleh komunitas yang besar (Syahrudin & Kurniawan, 2018).

Python mendukung pemrograman multi paradigma, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada *python* adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, *python* umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai *platform* sistem operasi (Syahrudin & Kurniawan, 2018).

2.11. TensorFlow

TensorFlow adalah perpustakaan perangkat lunak, yang dikembangkan oleh Tim Google Brain dalam organisasi penelitian Mesin Cerdas Google, untuk tujuan melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam. TensorFlow kemudian menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan. Fitur utamanya meliputi (TensorFlow):

1. Mendefinisikan, mengoptimalkan, dan menghitung secara efisien ekspresi matematis yang melibatkan array multi dimensi (*tensors*).

2. Pemrograman pendukung jaringan syaraf dalam dan teknik pembelajaran mesin
3. Penggunaan GPU yang transparan, mengotomatisasi manajemen dan optimalisasi memori yang sama dan data yang digunakan. Tensorflow bisa menulis kode yang sama dan menjalankannya baik di CPU atau GPU. Lebih khusus lagi, TensorFlow akan mengetahui bagian perhitungan mana yang harus dipindahkan ke GPU.
4. Skalabilitas komputasi yang tinggi di seluruh mesin dan kumpulan data yang besar.

2.12. Keras

Keras adalah perpustakaan berbasis sumber terbuka yang dirancang untuk menyederhanakan model dari kerangka *Deep Learning*. *Keras* ditulis dengan bahasa pemrograman *Python*, dapat dijalankan di atas *framework* (kerangka kerja) kecerdasan buatan seperti TensorFlow, Microsoft Cognitive Toolkit, dan Theano.

Saat ini, *Keras* dianggap sebagai salah satu perpustakaan pembelajaran mesin terbaik di Python. *Keras* juga menyediakan beberapa utilitas terbaik dalam hal menyusun model, memproses dataset, memvisualisasikan grafik, dan hal lainnya.

2.13. Confusion Matrix

Akurasi suatu model klasifikasi dapat diukur dengan menggunakan *Confusion Matrix* dari perhitungan hasil data testing (Haristu, 2019). Parameter yang digunakan *confusion matrix* dalam pengukuran suatu model dijelaskan pada tabel 2.1.

Tabel 2.1 Confusion Matrix

		Kelas Aktual	
		Positif	Negatif
Kelas Prediksi	Positif	TP (True Positive)	FP (False Positive)
	Negatif	FN (False Negative)	TN (True Negative)

Berdasarkan tabel matriks konfusi, terdapat nilai dalam matriks tersebut, yaitu true positive (TP), false positive (FP), false negative (FN), dan true negative (TN). Berdasarkan keempat nilai tersebut dapat dihitung akurasi, presisi, dan recall rate. Akurasi adalah perhitungan yang modelnya ditunjukkan pada persamaan (2.3). Akurasi menggambarkan tingkat akurasi model dalam memprediksi peristiwa positif dalam beberapa kegiatan prediksi. *Recall rate* adalah indeks yang mengukur sensitivitas sistem kelas.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(2.4)$$

Dimana TP (True Positive) adalah hasil prediksi benar yang benar dalam pengklasifikasian, TN (True Negative) adalah hasil prediksi salah namun benar dalam pengklasifikasian, FP (False Positive) adalah hasil prediksi benar namun salah dalam pengklasifikasian, FN (False Negative) adalah hasil prediksi salah dan juga salah dalam pengklasifikasian.

2.14. Studi Pustaka (State of The Art)

Keaslian dari penelitian ini berdasarkan pada penelitian terdahulu yang mempunyai topik pembahasan yang kurang lebih sama. Penelitian sebelum menjadi salah satu hal penting, sebagai suatu kajian dan referensi bagi penulis untuk mengetahui hubungan antara penelitian sebelumnya dengan penelitian yang dilakukan saat ini. Selain itu, hal tersebut dapat menghindari adanya duplikasi. Dampak yang dapat diberikan adalah menunjukkan bahwa penelitian yang dilakukan mempunyai arti penting, sehingga dapat memberikan kontribusi pada perkembangan ilmu pengetahuan. Berikut ini adalah beberapa penelitian terdahulu yang berhubungan dengan kematangan buah jambu kristal menggunakan *Computer Vision* dan metode *Convolutional Neural Network*.

Tabel 2.2 Studi Pustaka

Penulis	Judul	Metode	Hasil Penelitian	Gap Penelitian
Adhi Wibowo, 2021	Deteksi Kematangan Buah Jambu Kristal Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna Hsv (Hue Saturation Value) Dan K-Nearest Neighbor	Pengolahan Citra, K-Nearest Neighbor	Penelitian ini menjelaskan untuk mengukur kematangan buah jambu kristal dapat dilakukan dengan memanfaatkan teknologi citra digital menggunakan <i>machine learning</i> metode KNN, dimana untuk mengukur tingkat kematangan dengan melihat warna dari jambu kristal tersebut. Penelitian ini menghasilkan nilai presentase 91,67% untuk kategori buah jambu matang, 90% untuk kategori buah jambu mentah dan nilai keseluruhan data mempunyai presentase yaitu sebesar 95%.	Metode yang digunakan KNN dan transformasi ruang warna HSV, tidak mendeteksi tingkat kematangan secara <i>real time</i> menggunakan <i>computer vision</i>
Febry Yuni Mulato, 2015	Klasifikasi Kematangan Buah Jambu Biji Merah (Psidium Guajava) Dengan Menggunakan Model Fuzzy	Pengolahan Citra, Fuzzy	Hasil penelitian ini dalam mengklasifikasi buah jambu biji menggunakan model fuzzy dibagi menjadi empat kategori yaitu mentah, setengah, matang dan busuk. Penelitian ini menghasilkan tingkat keakuratan untuk data training 94.67% dan keakuratan untuk data testing 83.3%.	Menggunakan jambu biji merah sebagai objek, metode yang digunakan model fuzzy, tidak mendeteksi tingkat kematangan secara <i>real time</i> menggunakan <i>computer vision</i>
Fahmil Ikhsan Hidayat, 2017	Identifikasi Kematangan Buah Jambu Biji Merah (Psidium Guajava) Dengan Teknik	Pengolahan Citra, ANN <i>Backpropagation</i>	Dari penelitian ini menjelaskan <i>deep learning</i> dapat digunakan untuk mengklasifikasi kematangan buah jambu berdasarkan warna dengan menggunakan metode ANN, dimana <i>deep learning</i> metode ANN memiliki tingkat akurasi	Objek yang digunakan jambu biji merah, menggunakan <i>deep learning</i> metode ANN, tidak mendeteksi tingkat kematangan secara

Tabel Lanjutan 2.2 Studi Pustaka

	Jaringan Syaraf Tiruan Metode Backpropagation		yang tinggi. Penelitian ini memiliki tingkat keberhasilan identifikasi kebenaran 83.3%.	<i>real time</i> menggunakan <i>computer vision</i>
Arvi Arkadia, 2021	Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN	Pengolahan Citra, <i>Convolutional Neural Network</i>	Penelitian tersebut menjelaskan <i>deep learning</i> metode CNN dapat digunakan untuk mengklasifikasikan tingkat kematangan buah mangga badami berdasarkan warna dengan tingkat akurasi yang sangat tinggi. Pada penelitian ini menghasilkan akurasi pengujian model sebesar 97,2%.	Menggunakan mangga badami sebagai objek, tidak mendeteksi tingkat kematangan secara <i>real time</i> menggunakan <i>computer vision</i>
Andi Baso Kaswar, 2020	Klasifikasi Tingkat Kematangan Buah Markisa Menggunakan Jaringan Syaraf Tiruan Berbasis Pengolahan Citra Digital	Pengolahan Citra, Jaringan Syaraf Tiruan	Hasil pengujian yang dilakukan penelitian ini dengan menggunakan <i>deep learning</i> metode ANN untuk klasifikasi kematangan buah markisa berdasarkan warna diperoleh hasil akurasi sebesar 80%	Objek yang digunakan dalam penelitian markisa, menggunakan <i>deep learning</i> metode ANN, tidak mendeteksi tingkat kematangan secara <i>real time</i> menggunakan <i>computer vision</i>
Rhama Permadi Ahmad, 2020	Klasifikasi Kematangan Buah Mangrove Menggunakan Metode <i>Deep Convolutional Neural Network</i>	Pengolahan Citra, <i>Deep Convolutional Neural Network</i>	Hasil penelitian ini dapat disimpulkan dengan menggunakan <i>deep learning</i> metode CNN untuk mengklasifikasikan kematangan buah mangrove dapat menghasilkan akurasi yang sangat tinggi dengan nilai akurasi sebesar 99,1%.	Menggunakan buah mangrove sebagai objek, menggunakan teknik grayscaling dalam pengolahan citra, tidak mendeteksi tingkat kematangan secara <i>real time</i> menggunakan <i>computer vision</i>
Hafid Bahtiar, 2016	Sistem Pendeteksi Helm Yang Dikenakan Pengendara Sepeda Motor Untuk Safety Riding Berbasis Raspberry Pi	Pengolahan Citra Digital, <i>Computer Vision</i>	Hasil penelitian ini menjelaskan penggunaan <i>computer vision</i> dalam mendeteksi suatu benda memiliki tingkat akurasi yang cukup tinggi. Pada penelitian ini mampu mendeteksi dengan akurasi yang cukup tinggi yaitu sebesar 93,7%.	Menggunakan mini-computer raspberry pi, tidak mendeteksi kematangan buah berdasarkan warna
Oktaviano Yudha	Aplikasi Komputer Vision Untuk Identifikasi	Pengolahan Citra Digital, <i>Computer</i>	Penelitian ini menjelaskan penggunaan <i>computer vision</i> dalam mendeteksi suatu benda dapat digunakan untuk	Objek yang digunakan jeruk nipis, dalam pemilahan

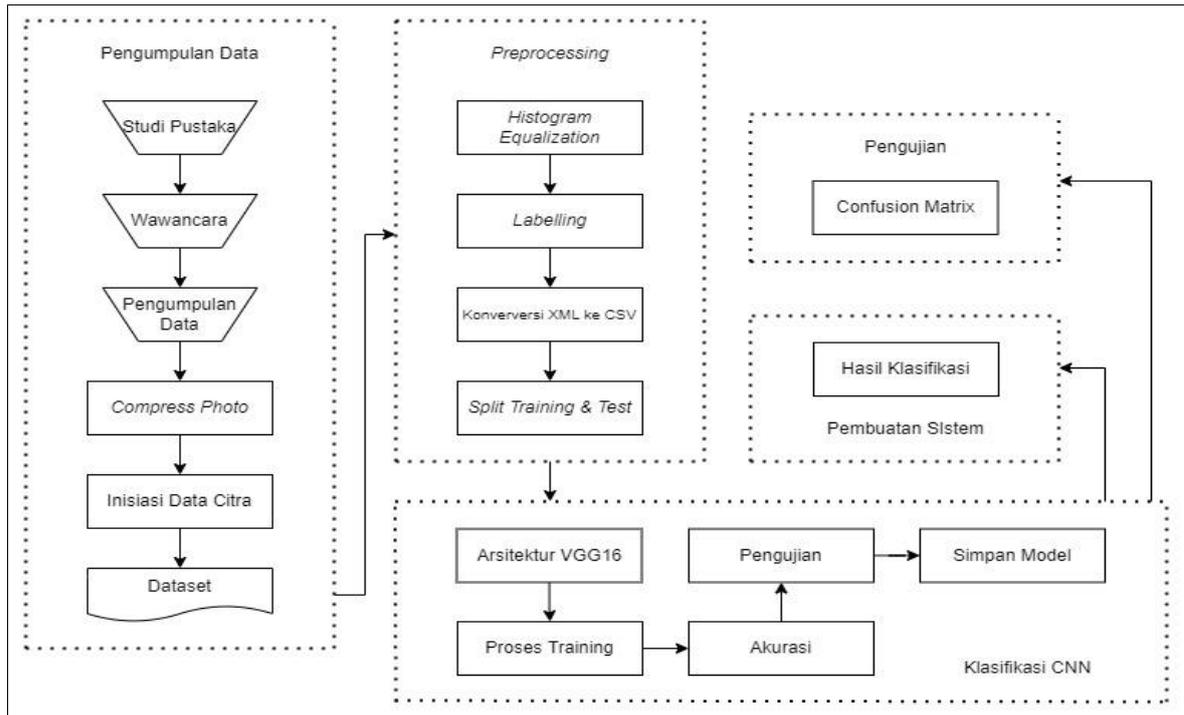
Tabel Lanjutan 2.2 Studi Pustaka

N, 2013	Kematangan Jeruk Nipis	<i>Vision, Jaringan Syaraf Tiruan</i>	mengklasifikasi kematangan buah berdasarkan warna. Pada penelitian ini menghasilkan tingkat akurasi sebesar 80% untuk kondisi buah matang, 70% untuk kondisi busuk dan 80% untuk kondisi random.	kematangan menggunakan <i>controller system</i> mekanik
Navalen Aginda Prasetyo, 2021	Aplikasi Komputer Vision Untuk Identifikasi Kematangan Jeruk Nipis	Pengolahan Citra Digital, Jaringan Syaraf Tiruan, <i>Computer Vision</i>	Dari penelitian ini menjelaskan penggunaan <i>computer vision</i> dalam mengklasifikasi tingkat kematangan buah dapat digabungkan dengan <i>deep learning</i> metode jaringan syaraf tiruan yang memiliki tingkat akurasi yang tinggi. Pada penelitian ini menghasilkan akurasi rerata sebesar 98,4%.	Menggunakan objek nanas, menggunakan <i>deep learning</i> metode ANN

BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

3.1. Metodologi Penelitian

Pada bab ini dibahas tentang metodologi penelitian kuantitatif dengan data primer dan metodologi pengembangan sistem yang digunakan sebagai acuan dari penelitian *Convolutional Neural Network* Klasifikasi Kematangan Buah Jambu Kristal. Berikut tahapan metodologi penelitian yang dilakukan dalam penelitian ini dalam gambar 3.1:



Gambar 3.1. Tahapan Kerangka Kerja Penelitian

3.1.1. Pengumpulan Data

Tahap pertama adalah studi pustaka, studi pustaka pada penelitian ini merupakan tahap mencari berbagai sumber referensi dari jurnal, buku, dan penelitian terdahulu yang terkait dengan penelitian yang dilakukan serta mencari *state of the art* dari penelitian yang dilakukan. Pada tahap ini juga digunakan sebagai pendalaman materi tentang apa saja yang diperlukan dan dibutuhkan untuk membuat penelitian yang dilakukan. Pada tahap ini dilakukan pencarian informasi mengenai metode dan teknik mengklasifikasikan kematangan buah jambu kristal dengan pengolahan citra digital untuk mencari data pembandingan dan informasi lainnya yang dapat mendukung pengembangan sistem.

Kemudian dilakukan tahap wawancara, wawancara pada penelitian ini digunakan untuk mendapatkan informasi tentang pengetahuan dan pembelajaran ilmu jambu kristal langsung dari ahli atau petani jambu kristal. Wawancara bersama Ibu Ketut Mahatma Aula dan Bapak Suroño merupakan pemilik Kebun Jambu Kristal dan petani jambu kristal di daerah Polokarto, Sukoharjo sebagai berikut:

Tabel 3.1 Tabel Wawancara

No.	Hari dan Tanggal	Pembahasan	Hasil
1.	Sabtu, 12 September 2020	Melakukan wawancara untuk mempelajari jambu kristal secara umum.	Memahami ilmu buah jambu kristal secara umum.
2.	Minggu, 20 September 2020	Mempelajari cara pemilihan buah jambu kristal yang sudah matang sebagai studi kasus penelitian ini.	Dapat membedakan buah yang matang atau belum matang dan mendapatkan data citra buah jambu kristal matang.
3.	Minggu, 27 September 2020	Mempelajari proses kematangan buah jambu kristal yang belum matang.	Mendapatkan data citra buah jambu kristal yang belum matang.

Pada wawancara tersebut menghasilkan kriteria panen buah jambu kristal. Menurut Ibu Ketut Mahatma Aula pada proses panen buah jambu kristal, buah tidak bisa matang ketika sudah dipetik dari pohon, tidak seperti buah lain pada umumnya buah jambu kristal harus sudah matang sepenuhnya pada pohon agar bisa dipanen/dipetik. Untuk menghasilkan panen yang lebih baik buah jambu kristal harus dibungkus oleh plastik untuk menghindari hama dan cuaca yang ekstrim pada kebun.

Dalam panen buah jambu kristal dalam setahun dilakukan 2 kali panen, untuk waktu panen membutuhkan waktu 4 bulan dari bunga pohon berubah menjadi buah jambu kristal. Kemudian setelah panen dilakukan proses perawatan tanaman dengan cara pemupukan tanaman dan *pruning* dahan dan daun dalam waktu 2 bulan, lalu dapat dilakukan kembali panen buah jambu kristal dan begitu seterusnya.

Dalam menentukan buah yang sudah matang dapat dilakukan dengan cara melihat warna kulit buah yang berubah menjadi warna hijau kekuningan serta memiliki sedikit bercak, bentuk buahnya lebih besar dari yang belum matang dan tidak memiliki pentil pada ujung buah. Bentuk jambu kristal memiliki bentuk yang tidak terlalu bulat. Berikut hasil foto jambu kristal dari Kebun Larasati yang dipetik langsung dari tempatnya.



Gambar 3.2. Jambu Kristal Kebun Larasati

Data yang digunakan dalam penelitian ini merupakan data citra yang berupa data primer yaitu data-data yang digunakan berasal dari sumbernya secara langsung, bukan berasal dari peneliti sebelumnya. Dalam pengumpulan data pada penelitian ini data yang dikumpulkan adalah foto jambu kristal matang dan jambu kristal belum matang dari beberapa sumber, yaitu Kebun Buah Jambu Kristal Larasati, Kabupaten Sukoharjo dan toko yang menjual buah jambu kristal. Kebun Buah Jambu Kristal Larasati merupakan sebuah kebun buah jambu kristal yang berada di didaerah Polokarto, Kabupaten Sukoharjo yang juga menjadi tempat rekreasi oleh warga sekitar Kebun Buah Larasati.

Proses pengambilan data tahap awal yang dilakukan adalah dengan cara memfoto jambu kristal menggunakan kamera *handphone* merek Xiaomi mi 10t dengan spesifikasi kamera *quad camera* 108MP, f/1.7 (*wide*), 13 MP, f/2.4 12mm (*ultrawide*) dan masing-masing 2MP f/2,4 (*depth* dan *macro*). Ukuran citra yang diambil menggunakan kamera ponsel adalah 5792x4344 *piksel* dengan jarak pengambilan gambar sekitar 10-20cm dengan intensitas cahaya yang cukup terang menggunakan *ringlight* untuk mendapatkan hasil gambar yang bagus.

Setelah semua data diambil citranya, selanjutnya citra di-*compress* kecuali data *testing* menggunakan aplikasi *Adobe Photoshop cc 2020* dengan tujuan mengurangi ukuran file citra tetapi tidak mengurangi kualitas asli citra. Hal tersebut bertujuan untuk memperingan dan mempercepat proses *training* dikarenakan ukuran file citra lebih kecil. Kemudian dilakukan proses inisiasi data citra jambu kristal dibagi kedalam 2 dataset sesuai klasifikasi jambu kristal matang dan belum matang. Untuk jambu kristal matang berjumlah 309 citra dan 263 untuk jambu kristal belum matang.

Dibawah ini adalah contoh hasil pengambilan citra yang didapat dari masing-masing klasifikasi jambu kristal:

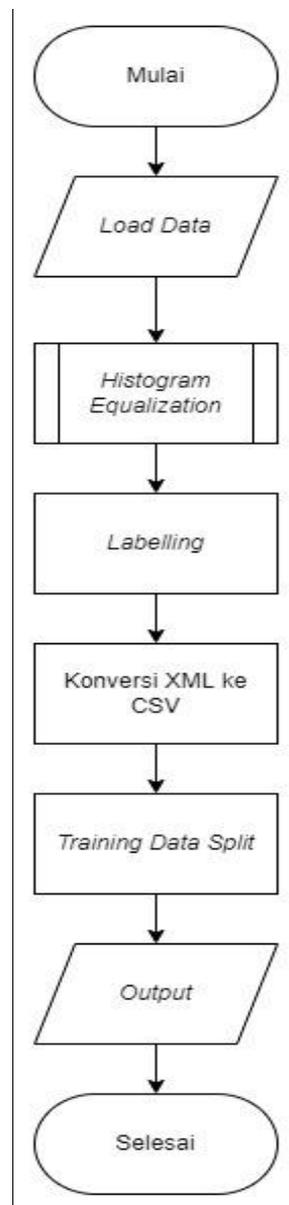


Gambar 3.3. Masing-masing Klasifikasi Jambu Kristal

Pada Gambar 3.3 diatas merupakan contoh dataset yang digunakan pada penelitian ini yang terdiri dari jambu kristal matang (sebelah kanan) dan jambu kristal belum matang (sebelah kiri). Citra jambu kristal matang dapat dilihat berwarna lebih terang lebih kekuningan, bagian tangkai buah sudah terlepas dan ukuran terlihat lebih besar. Sedangkan jambu kristal belum matang dapat dilihat berwarna lebih gelap dan belum kekuningan, bagian tangkai belum terkelupas seutuhnya dan ukuran relatif lebih kecil.

3.1.2. Data Preprocessing

Tahap data *preprocessing* merupakan tahap menyiapkan data yang akan digunakan pada proses *training* sebelum menjadi model. Pengolahan data pada penelitian ini yaitu terdiri dari empat langkah yaitu proses *histogram equalization*, *labelling*, konversi nilai XML ke CSV dan *training data split*. Flowchart pada pengolahan data akan ditampilkan sebagai berikut;



Gambar 3.4. *Preprocessing*

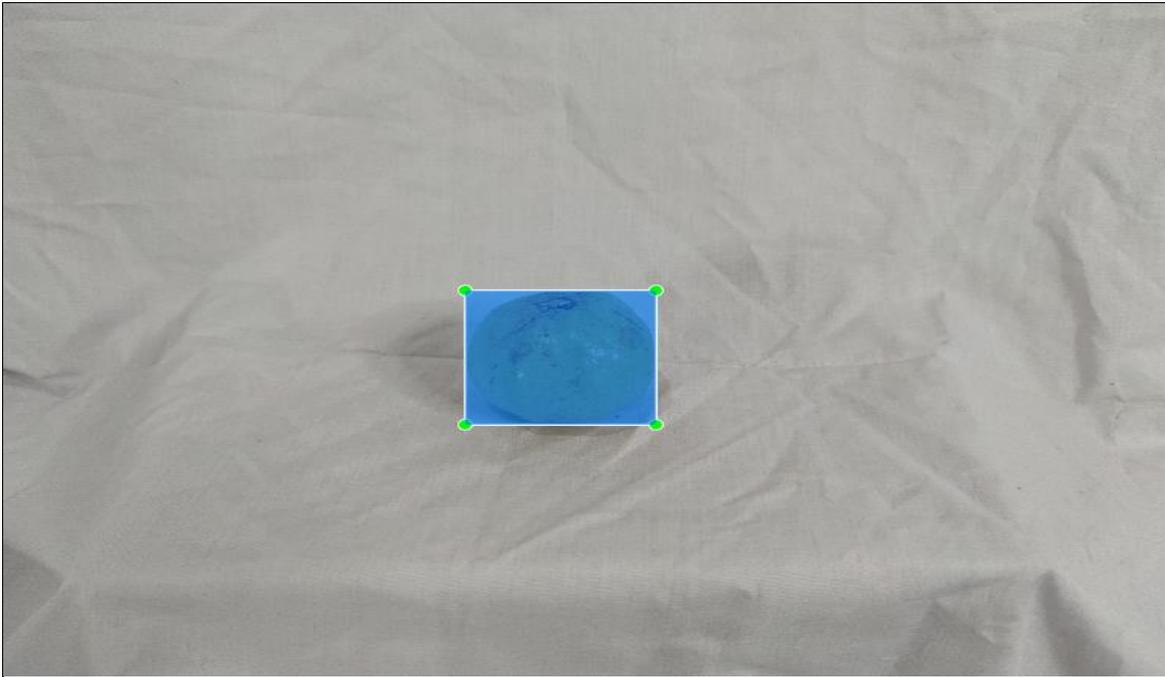
Pada proses pengolahan data, citra hasil pengambilan data yang sudah di-*compress* dan inisiasi dilakukan *histogram equalization* dimana didalamnya terdapat beberapa proses yang akan dijelaskan dalam *flowchart* berikut:



Gambar 3.5. Histogram Equalization

Pada *histogram equalization* dimulai dari menentukan nilai ukuran gambar yang diinput. Kemudian nilai RGB dari gambar yang diinput dikonversikan ke *grayscale* dan nilai tersebut dijadikan frekuensi nilai derajat keabuan. Frekuensi nilai tersebut dihitung nilai kumulatif dengan persamaan $\sum_{k=0}^{L-1} P_k$. Nilai kumulatif yang didapat digunakan untuk mencari nilai derajat keabuan baru pada tiap piksel. Lalu nilai derajat keabuan baru pada piksel dipetakan menjadi gambar bernilai RGB. Setelah *histogram equalization* dilakukan proses *labelling* dengan jenis teknik kotak pembatas untuk mengidentifikasi data citra jambu kristal dan menambah label yang memiliki makna dan nilai informatif matang atau

belum matang sehingga model pembelajaran mesin dapat mempelajarinya. Proses *labelling* akan ditampilkan sebagai berikut,



Gambar 3.6. Proses Labelling

Hasil dari proses *labelling* berupa anotasi data yang berbentuk *.xml* dimana anotasi data memiliki nilai label kematangan jambu dan nilai *xmin*, *xmax*, *ymin* dan *ymax* hasil dari teknik kotak pembatas. Nilai *xmin*, *xmax*, *ymin* dan *ymax* didalamnya mencakup nilai piksel RGB yang tersusun oleh tiga kanal warna yaitu kanal merah, kanal hijau dan kanal biru.

Setelah didapatkan data dalam nilai *.xml* kemudian dikonversikan kedalam nilai *.csv* dikarenakan nilai *.xml* hanya menyimpan data secara sederhana dan tidak dapat digunakan untuk mentransfer data antar program. Data dalam nilai *.csv* dapat digunakan untuk mentransfer data ke tensorflow yang datanya akan digunakan pada proses *training*. Kemudian data nilai *.csv* tersebut dibagi menjadi data latih (*training*) dan data validasi (*testing*). Sedangkan untuk data uji (*testing*) menggunakan data baru diluar dataset. Berikut rincian kelas data dalam penelitian ini,

Tabel 3.2 Rincian Kelas Data

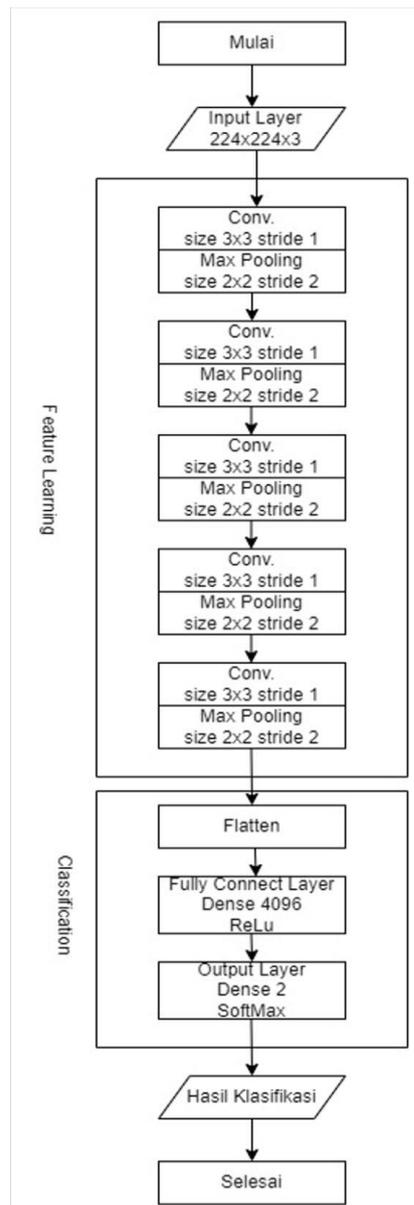
Data Citra	Data Training	Data Testing	Total
Jambu Kristal Matang	247	62	309
Jambu Kristal Belum Matang	210	53	263
Total	457	115	572

Pada Tabel 3.2 diketahui total dataset yang digunakan berjumlah 572 data citra. Keseluruhan data tersebut terdiri dari 309 citra jambu kristal matang dan 263 citra jambu belum matang. Pembagian data *training* dan data *testing* adalah 80%:20% sehingga terbagi 247 data *training* jambu kristal dan 62 data *testing*. Untuk jambu kristal belum matang

terbagi data *training* berjumlah adalah 210 data dan 53 data *testing*. Sehingga untuk total data *training* berjumlah 457 dan data *testing* berjumlah 115.

3.1.3. Convolutional Neural Network

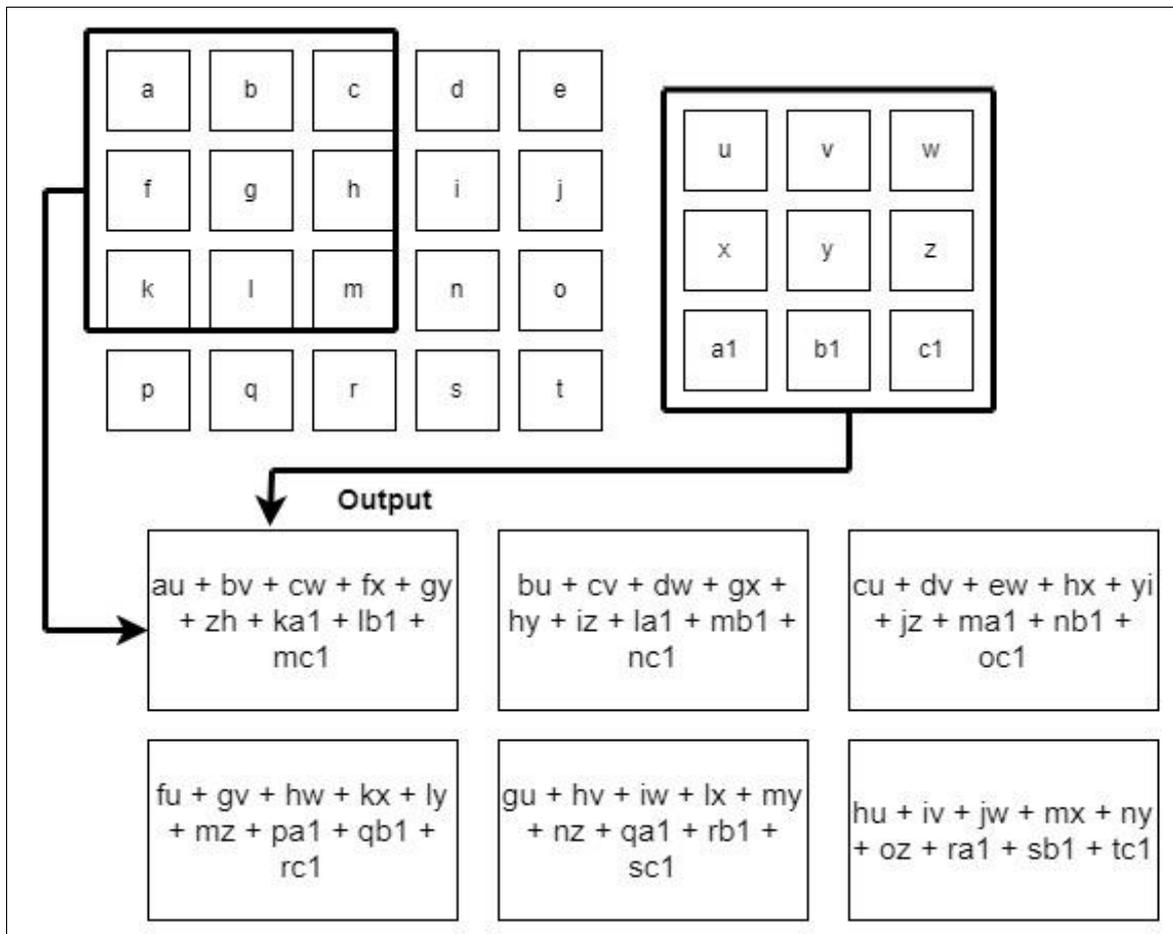
Pada penelitian ini digunakan *single shot detector* (SSD) dengan arsitektur *convolutional neural network* yang digunakan adalah VGG16. Model SSD dengan menggunakan *MobileNet* dapat berjalan dengan komputasi ringan (Liu et al., 2016). Dalam penelitian ini model SSD menggunakan *MobileNet* mengalami modifikasi dalam proses klasifikasi untuk mendapatkan *output* yang sesuai. Berikut *flowchart* arsitektur VGG16 pada penelitian ini ditampilkan:



Gambar 3.7. Arsitektur VGG16

Pada Gambar 3.7. menunjukkan pada arsitektur ini memiliki lima buah lapisan konvolusi, lima buah lapisan *pooling*, satu lapisan *flattening*, satu lapisan *fully connected layer* dan satu lapisan *output layer*. Pada lapisan konvolusi dan *fully connected* terdapat fungsi aktivasi yaitu fungsi aktivasi ReLu.

Arsitektur ini memiliki dua proses secara garis besar yaitu, *feature learning* dan *classification*. Pada arsitektur ini pertama input gambar yang sudah di-*resize* dengan resolusi 224 x 224 dengan warna RGB 3 channel atau disebut multidimensional array 224 x 224 x 3. Proses selanjutnya ada proses *feature learning* dimana setiap input gambar ditranslasikan menjadi *features* berdasarkan ciri dari input tersebut yang berbentuk angka-angka dalam *vector*. Lapisan yang terdapat pada *feature learning* ini terdiri dari lapisan konvolusi dan lapisan *pooling*. Berikut contoh perhitungan lapisan konvolusi;



Gambar 3.8. Ilustrasi Perhitungan Lapisan Konvolusi

Pada Gambar 3.8. Mendeskripsikan *convolutional layer*, dimana citra masukan akan dihitung menggunakan kernel/filter, dan nilainya sudah ditentukan sebelumnya sesuai citra masukan. Tujuannya adalah untuk menyederhanakan nilai matriks masukan tanpa menghilangkan nilai matriks masukan itu sendiri ke nilai intensi matriks baru yang disebut peta fitur. Data citra masukan matriks akan mengalami beberapa lapis proses konvolusi sesuai dengan model VGG16. Lapisan convolutional akan menghitung output dari neuron yang terhubung ke area lokal di input, dan setiap neuron menghitung produk titik antara bobotnya dan area kecil yang terhubung ke volume input.

Berikut merupakan contoh perhitungan manual yang dilakukan pada proses konvolusi matriks CNN. Pada contoh perhitungan CNN menggunakan citra dengan ukuran 4x4 sebagai, dengan menggunakan kernel/filter 3x3 dengan *stride* = 1 serta menggunakan *zero padding* pada setiap gambar. Berikut contoh proses perhitungan dengan citra 4x4 dijabarkan sebagai berikut,

a. Input Citra

Dimulai dengan input citra berukuran 4x4.

$$\begin{bmatrix} 21 & 26 & 138 & 26 \\ 153 & 94 & 28 & 103 \\ 133 & 86 & 29 & 37 \\ 94 & 75 & 33 & 72 \end{bmatrix}$$

b. Padding

Padding merupakan proses dimana proses nilai citra masukan ditambah nilai matriks bernilai 0 pada setiap lapisan yang bertujuan memanipulasi dari lapisan konvolusi.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 21 & 26 & 138 & 26 & 0 \\ 0 & 153 & 94 & 28 & 103 & 0 \\ 0 & 133 & 87 & 29 & 37 & 0 \\ 0 & 94 & 75 & 33 & 72 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

c. Kernel/Filter

Proses kernel/filter yang digunakan berukuran 3x3 dengan jenis filter deteksi tepi.

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

d. Konvolusi

Proses *filtering* pada lapisan konvolusi dengan cara *stride* akan berpindah kekanan dan kebawah. *Stride* akan berpindah 1 piksel dikarena nilai *stride* bernilai 1. Matriks dengan berwarna merah adalah matriks yang akan dilakukan proses *filtering* yang digunakan untuk mendapatkan nilai *featured map* menggunakan persamaan 3.1.

$$\sum_{t=1}^n a_t b_t = a_1 b_1 + a_2 b_2 + \dots + a_n b_n \dots \dots \dots (3.1)$$

Dari persamaan 3.1. n merupakan jumlah perkalian matriks dengan kernel/filter, kemudian lakukan perkalian antara matriks input yang sudah melakukan proses padding dengan matriks filter sehingga menciptakan nilai matriks baru.

$$\text{Matriks input padding: } \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 21 & 26 & 138 & 26 & 0 \\ 0 & 153 & 94 & 28 & 103 & 0 \\ 0 & 133 & 87 & 29 & 37 & 0 \\ 0 & 94 & 75 & 35 & 97 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \text{filter: } \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{Stride (1): } \begin{bmatrix} 0 & 0 & 0 \\ 0 & 21 & 26 \\ 0 & 153 & 94 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{Stride (2): } \begin{bmatrix} 0 & 0 & 0 \\ 21 & 26 & 138 \\ 153 & 94 & 28 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{Stride (3): } \begin{bmatrix} 0 & 0 & 0 \\ 26 & 138 & 26 \\ 94 & 28 & 103 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Proses perhitungan diatas akan dilakukan dengan persamaan 3.1. sebagai berikut,

$$\sum_{t=1}^9 a_t b_t = (0 \times 1) + (0 \times 0) + (0 \times -1) + (0 \times 0) + (21 \times 0) + (153 \times 0) + (0 \times -1) + (26 \times 0) + (94 \times 1) = 94$$

$$\sum_{t=2}^9 a_2 b_2 = (0 \times 1) + (21 \times 0) + (153 \times -1) + (0 \times 0) + (26 \times 0) + (94 \times 0) + (0 \times -1) + (138 \times 0) + (28 \times 1) = -125$$

$$\sum_{t=3}^9 a_3 b_3 = (0 \times 1) + (26 \times 0) + (94 \times -1) + (0 \times 0) + (138 \times 0) + (28 \times 0) + (0 \times -1) + (26 \times 0) + (103 \times 1) = -9$$

Berikutnya perhitungan dilanjutkan hingga ke stride (16) sehingga perhitungan tersebut menghasilkan matriks baru dengan ukuran 4x4 sebagai berikut,

$$\text{Featured Map Layer: } \begin{bmatrix} 94 & -125 & -9 & -28 \\ 61 & -221 & -50 & 109 \\ -19 & 64 & -12 & -5 \\ -87 & 104 & 50 & 29 \end{bmatrix}$$

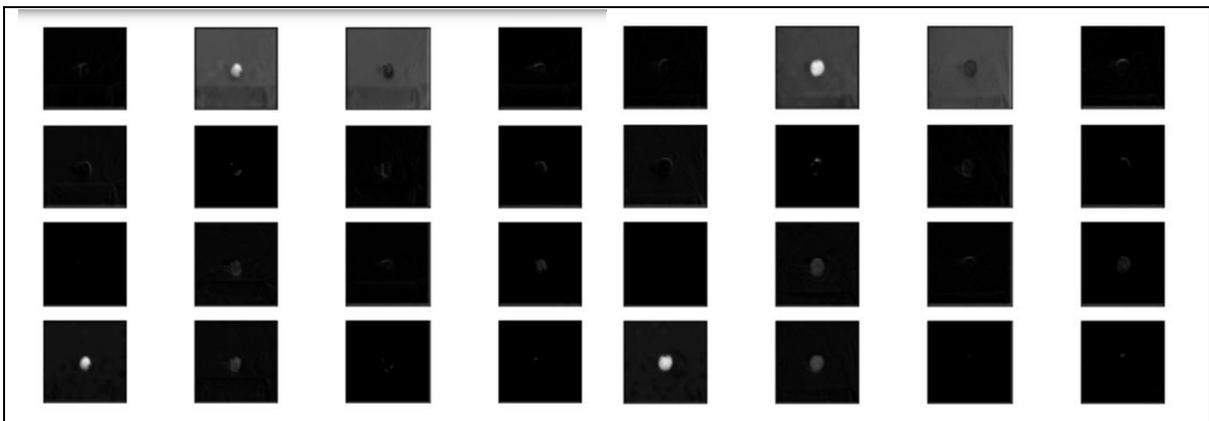
Setelah didapatkan *feature map layer* baru kemudian dilakukan proses aktivasi fungsi aktivasi *Rectified Linear Unit* (ReLU). Proses ini berfungsi merubah nilai piksel yang memiliki nilai kurang dari 0 akan dikoversikan nilainya menjadi 0, dengan rumus $f(x) = \max(0,x)$ maka akan menciptakan nilai matriks sebagai berikut,

$$\text{ReLU Featured Map: } \begin{bmatrix} 94 & 0 & 0 & 0 \\ 61 & 0 & 0 & 109 \\ 0 & 64 & 0 & 0 \\ 0 & 104 & 50 & 29 \end{bmatrix}$$

Proses selanjutnya adalah *pooling* dengan stride = 2 sehingga pada contoh perhitungan akan menghasilkan ukuran matriks 2x2. *Pooling layer* berfungsi untuk menyusutkan input secara spasial serta mengurangi jumlah parameter dengan operasi *downsampling*, sehingga menyusutkan dimensi dari *featured map* dan mempercepat komputasi karena ukuran parameter yang di-update semakin sedikit dan mengatasi *overfitting*. Pada penelitian ini digunakan metode *max pooling*. *Max pooling* berfungsi untuk menentukan nilai maksimum pada tiap pergeseran filter. Berikut ilustrasi *max pooling* hasil dari *ReLU featured map*,

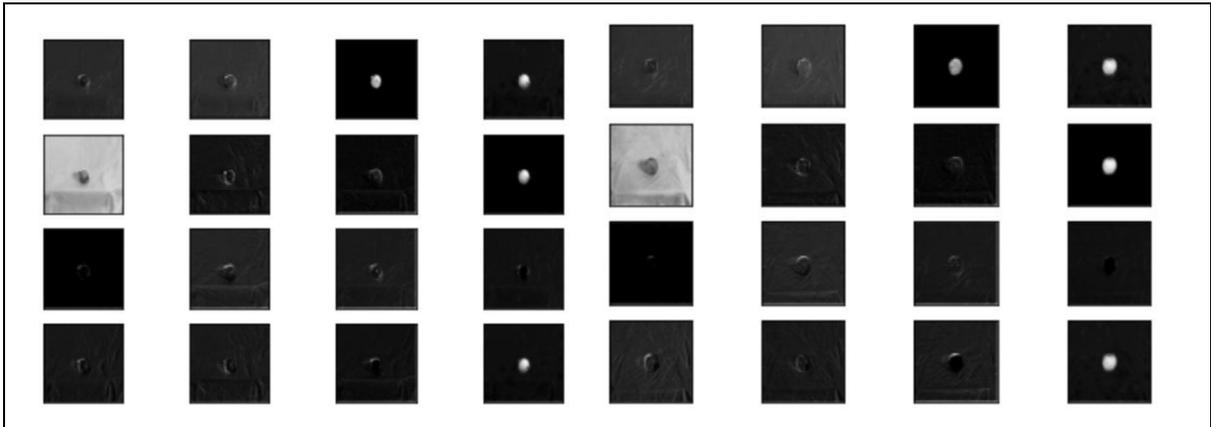
$$\text{Max Pooling: } \begin{bmatrix} 94 & 109 \\ 104 & 50 \end{bmatrix}$$

Kemudian perhitungan dilanjutkan kembali ketahap proses konvolusi, ReLu dan *pooling* dengan *featured map* baru. Berikut hasil dari proses konvolusi pada penelitian ini menggunakan *MobileNet*,

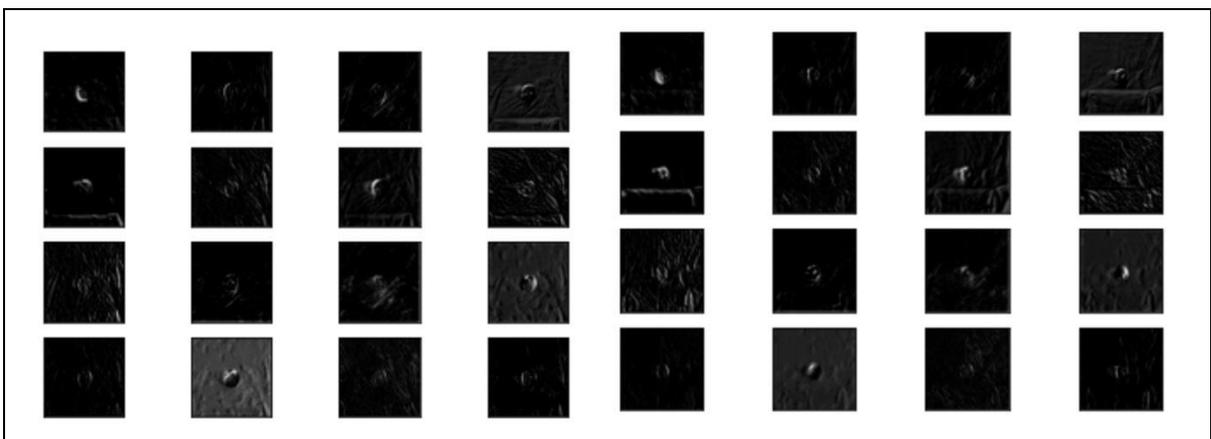


Gambar 3.9. Lapisan Konvolusi depthwise dan pointwise ke-1

Pada lapisan konvolusi pertama akan menghasilkan output citra yang masih terlihat jelas bentuk jambu kristal matang dan belum matang. Pada gambar 3.10 dapat dilihat hasil proses konvolusi dari lapisan *depthwise* dan *pointwise* ke-1.

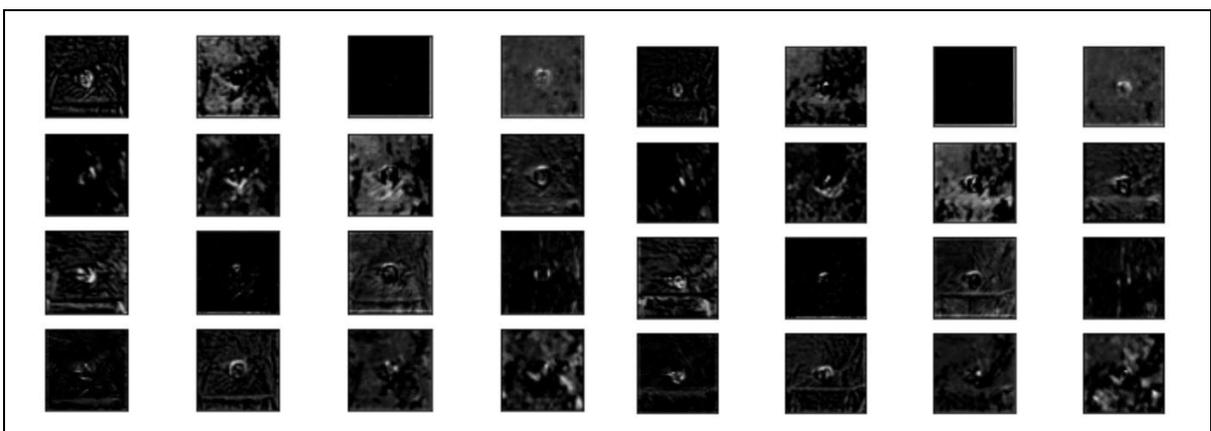


Gambar 3.10. Lapisan Konvolusi depthwise dan pointwise ke-3

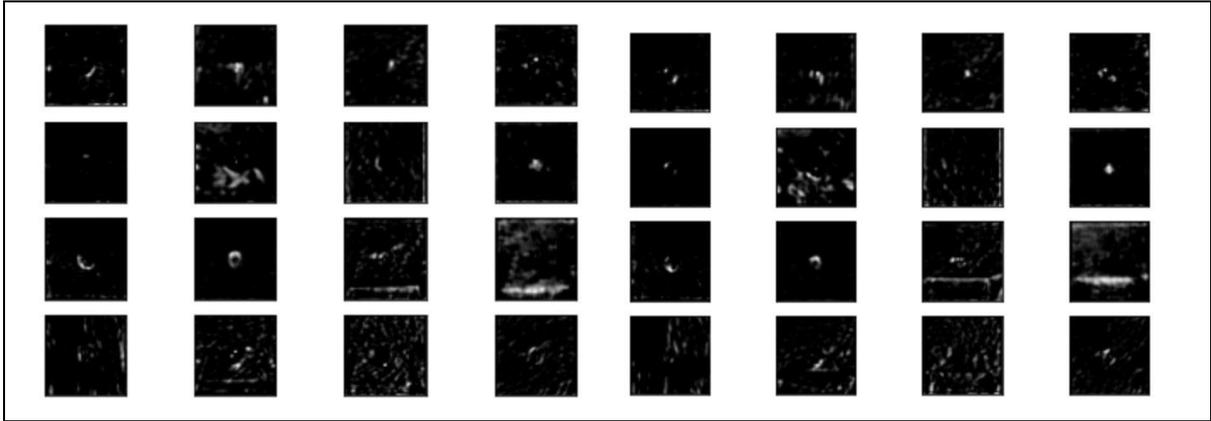


Gambar 3.11. Lapisan Konvolusi depthwise dan pointwise ke-5

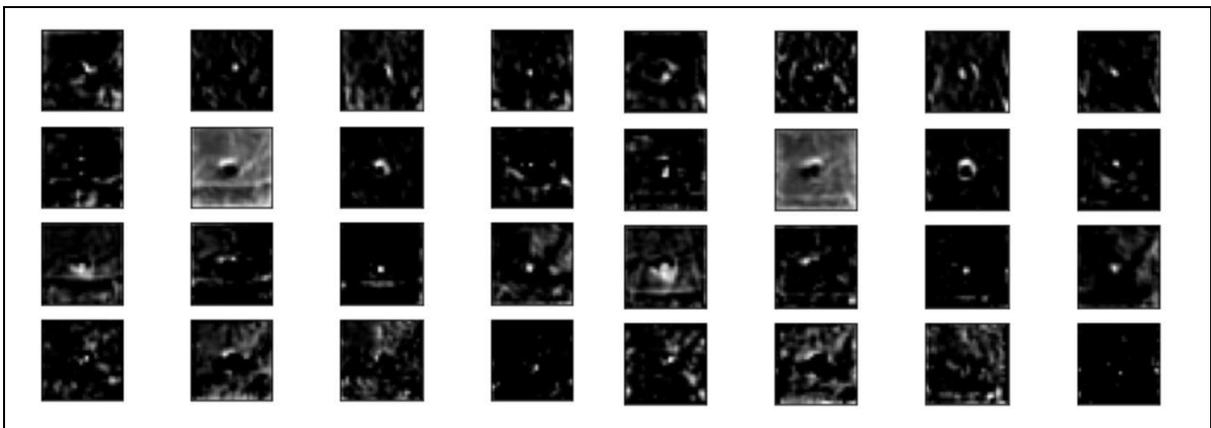
Pada lapisan konvolusi ke-3 serta ke-5 akan menghasilkan output citra yang masih terlihat cukup jelas bentuk jambu kristal matang dan belum matang. Pada lapisan ini, fitur dari citra jambu kristal matang dan belum matang telah sebagian diekstrak, sehingga gambar yang dihasilkan akan sedikit *blur*. Pada gambar 3.11 dan gambar 3.12 dapat dilihat hasil proses konvolusi dari lapisan *depthwise* dan *pointwise* ke-3 serta ke-5.



Gambar 3.12. Lapisan Konvolusi depthwise dan pointwise ke-7

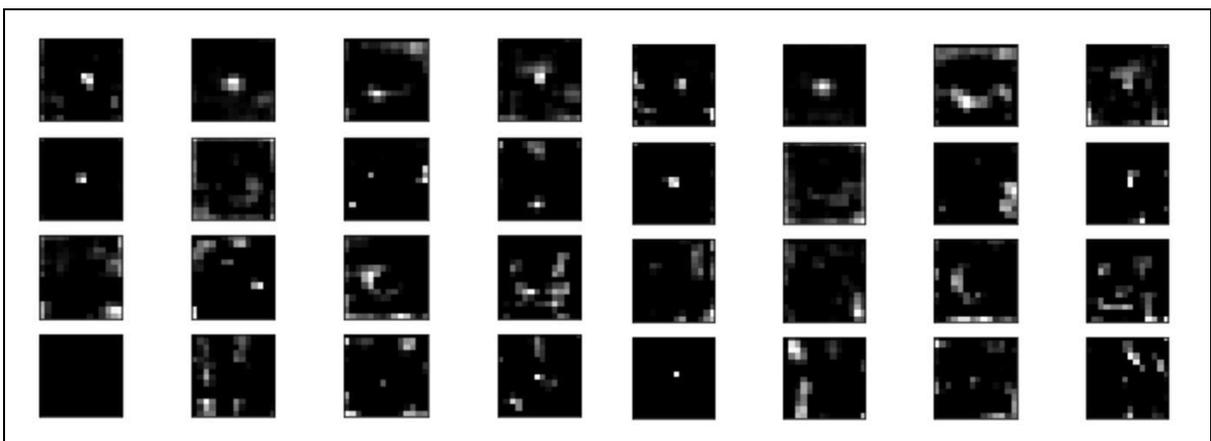


Gambar 3.13. Lapisan Konvolusi depthwise dan pointwise ke-9

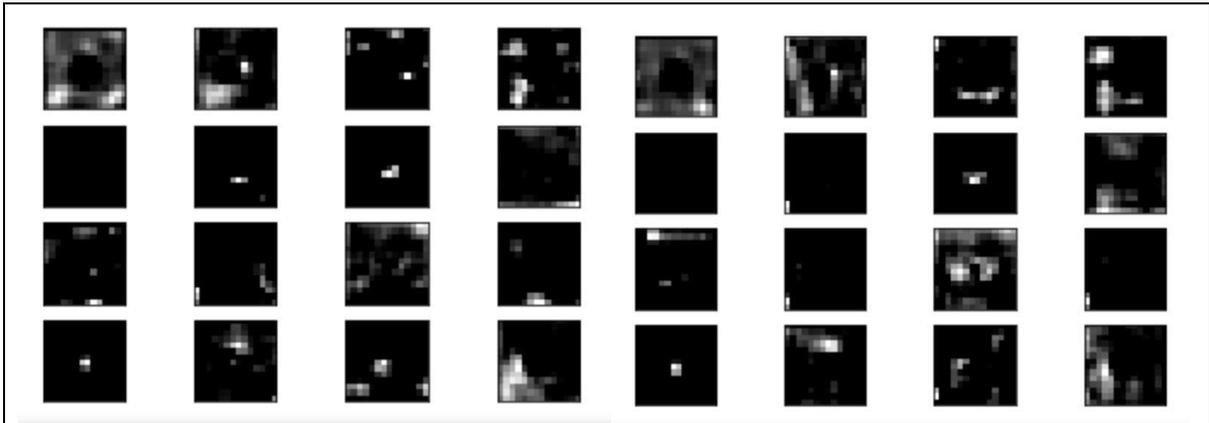


Gambar 3.14. Lapisan Konvolusi depthwise dan pointwise ke-11

Pada lapisan konvolusi ke-7, ke-9 serta ke-11, fitur dari citra jambu kristal matang dan belum matang telah sebagian diekstrak, sehingga gambar yang dihasilkan akan *blur*. Pada gambar 3.13, gambar 3.14 dan gambar 3.15 dapat dilihat hasil proses konvolusi dari lapisan *depthwise* dan *pointwise* ke-7, ke-9 serta ke-11.



Gambar 3.15. Lapisan Konvolusi depthwise dan pointwise ke-15



Gambar 3.16. Lapisan Konvolusi depthwise dan pointwise ke-16

Pada lapisan konvolusi ke-15 serta ke-16 akan menghasilkan output citra yang bentuk corak jambu kristal matang dan belum matang *pixelated* atau sangat terlihat bagian pikselnya. Namun jika diperhatikan lebih dekat, bagian dari jambu kristal matang dan belum matang masih dapat dilihat dengan jelas. Pada lapisan ini, fitur dari citra jambu kristal matang dan belum matang sebagian besar telah diekstrak, sehingga gambar yang dihasilkan akan lebih *blur*. Pada gambar 3.16 dan gambar 3.17 dapat dilihat hasil proses konvolusi dari lapisan *depthwise* dan *pointwise* ke-15 dan ke-16.

Kemudian proses dilanjutkan ketahap proses *classification*. Proses pertama dalam *classification* adalah proses *flattening* yaitu, proses *reshape featured map* dengan cara nilai matriks hasil *pooling* menjadi sebuah vector baris yang digunakan sebagai input dari *fully connected layer*. Baris pertama (295, 272, ...) digabung baris kedua (335, 348, ...) sampai matriks terakhir. Setelah proses *flattening*, selanjutnya nilai vector baris ini akan menjadi input layer di ANN (*artificial neural network*). Berikut ilustrasi proses *flattening* dengan menggunakan hasil proses *feature learning*,

$$\text{Max Pooling: } \begin{bmatrix} 94 & 109 \\ 104 & 50 \end{bmatrix} \quad \text{Flattening: } [94 \ 109 \ 104 \ 50]$$

Nilai diatas adalah salah satu hasil dari layer konvolusi dengan filter berukuran 3x3. Hasil *flattening* dipakai sebagai masukkan pada layer *full connected layer*. Layer ini mempunyai kecenderungan struktur seperti artificial neural network yaitu mempunyai *input layer*, *hidden layer* dan *output layer* yang masing-masing mempunyai neuron-neuron yang saling terhubung menggunakan neuron-neuron lain layer tetangganya.

Pada input layer memiliki fungsi menggabungkan holistik matriks *feature map* yg didapat berdasarkan proses *pooling layer*. Kemudian seluruh piksel tersebut direntangkan sebagai sebuah vektor sepanjang jumlah piksel berdasarkan matriks yg didapat dalam *pooling layer*. Kemudian semua nilai dalam input layer dipakai buat perhitungan dalam *hidden layer*. Pada perhitungan dalam *hidden layer* ini yaitu mengkalikan nilai-nilai yg terdapat pada input layer menggunakan bobot yg telah diinisialisasi sebelumnya kemudian ditambahkan menggunakan nilai bias. Rumus perhitungan pada *hidden layer* adalah sebagai berikut,

$$z_{in_i} = \sum_{j=1}^n X_j * V_{j,i} + V_{0,i} \dots \dots \dots (3.3)$$

Setelah dilakukan perhitungan diatas, selanjutnya masukkan fungsi aktivasi ReLu untuk semua hasil perhitungan. Hasil dari perhitungan ini akan digunakan pada proses perhitungan *output layer*. Perhitungan dalam *output layer* ini yaitu mengkalikan nilai-nilai

output perhitungan dalam *hidden layer* menggunakan bobot yg telah diinisialisasi sebelumnya kemudian dibubuhi menggunakan nilai bias. Rumus perhitungan pada *output layer* adalah sebagai berikut,

$$z_{in_i} = \sum_{j=1}^m Z_j * W_{j,i} + W_{0,i} \dots \dots \dots (3.4)$$

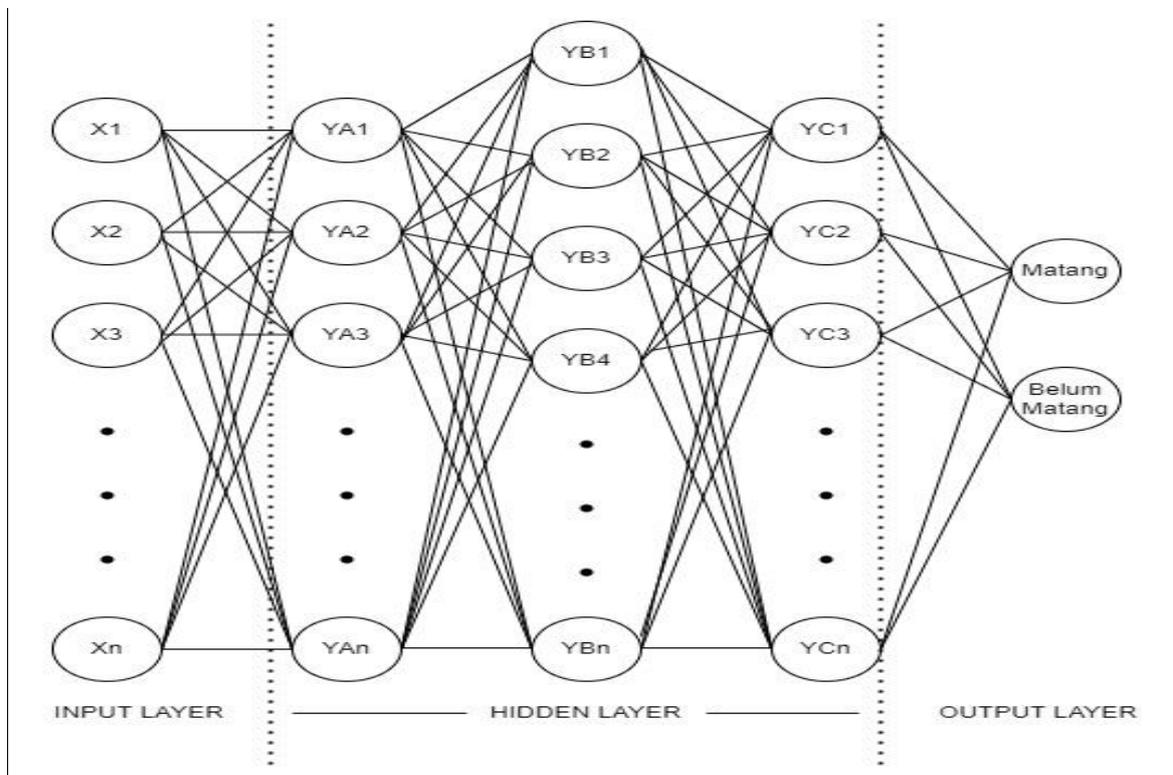
Setelah dilakukan perhitungan, selanjutnya tambahkan fungsi aktivasi *softmax* buat semua output yang berguna untuk mengklasifikasi nilai dominan dalam layer konvolusi sebagai akibatnya dapat mengkategorikan sebagai objek menurut kelas tertentu.

Pada bagian *output layer* terdapat beberapa kategori atau *class*. Pada penelitian ini terdapat 2 kelas yaitu kelas jambu matang dan jambu belum matang. Hasil dari prediksi yang mengungkapkan suatu citra berada di kelas jambu matang dan jambu belum matang ditampilkan dengan probabilitas. Suatu citra akan diprediksi dengan menghasilkan nilai probabilitas pada suatu kategori.

Fungsi yang bekerja buat menghitung nilai probabilitas berdasarkan setiap kelas sasaran atas seluruh kelas sasaran yg memungkinkan dan akan membantu buat memilih kelas sasaran buat input yg diberikan. Fungsi aktivasi *softmax* dipakai buat menghitung hasil klasifikasi. Rumus perhitungan pada aktivasi *softmax* adalah sebagai berikut,

$$y_i = \frac{e^{y_{in_i}}}{\sum_{i=1}^m e^M} \dots \dots \dots (3.5)$$

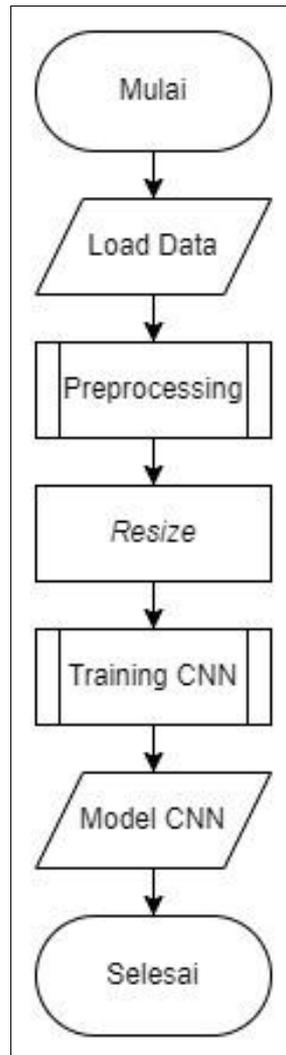
Softmax memakai eksponensial (*e-power*) berdasarkan nilai input yg diberikan dan jumlah nilai eksponensial berdasarkan seluruh nilai pada input. Sehingga rasio eksponensial berdasarkan nilai input dan jumlah nilai eksponensial merupakan hasil berdasarkan fungsi *softmax*.



Gambar 3.17. Fully Connected Layer

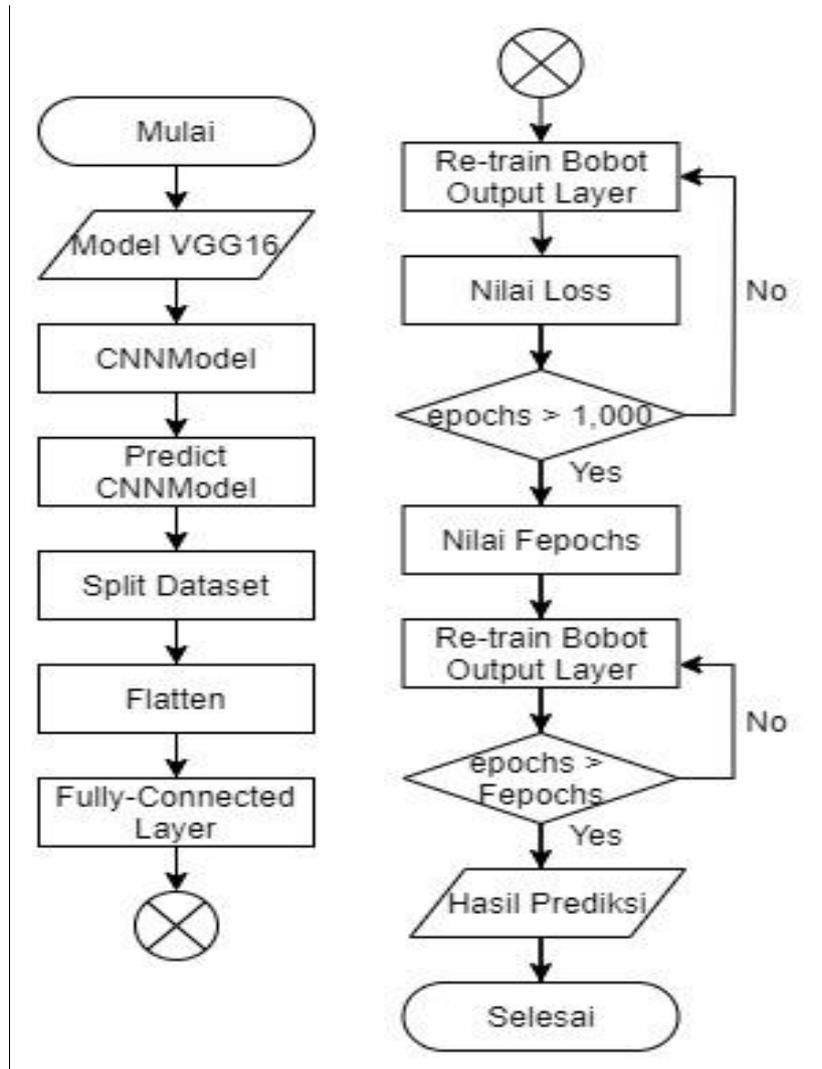
Pada Gambar 3.17. menjelaskan *input layer*, *hidden layer* dan *output layer* yang saling terhubung pada *fully connected layer*. *Layer-layer* ini umumnya memiliki struktur yang sama dengan jaringan syaraf tiruan dimana setiap *layer* berisikan *neuron* dengan *neuron* yang saling berhubungan. *Input layer* adalah hasil dari proses *flattening*, kemudian *input layer* tersebut saling dihubungkan pada proses *hidden layer* sehingga menghasilkan *output layer* yaitu, jambu kristal matang dan jambu kristal belum matang.

Perancangan Flowchart Model CNN secara garis besar dibagi menjadi sub program seperti pada Gambar 3.18.



Gambar 3.18. Flowchart Model CNN

Pada Gambar 3.18 pertama *load dataset* yang terbagi menjadi dua kelas yaitu jambu matang dan jambu belum matang. Gambar yang akan dilatih harus dapat dibaca oleh program sehingga dilakukan proses *labelling* dan proses merubah nilai XML ke CSV yang digunakan untuk mentransfer data ke tensorflow yang datanya akan digunakan pada proses *training*. Kemudian gambar di *resize* menjadi ukuran matriks 224x224 piksel yang dimana gambar digunakan sebagai input pada proses CNN dengan arsitektur yang digunakan yaitu VGG16 untuk dilakukan proses *training* dan *testing* menggunakan metode CNN dan akhirnya didapatkan model CNN.



Gambar 3.19. Proses Training

Proses CNN pada penelitian ini memiliki beberapa tahap dilihat pada Gambar 3.19. Pertama *load pre-trained* model VGG16 dengan bobot *imageNet* yang digunakan untuk mengklasifikasikan gambar baru. Kemudian sebagaimana dari *pre-trained* model digunakan untuk memproses gambar sebelumnya dan mengekstrak fitur yang relevan. Lalu diintegrasikan kedalam model baru dan lapisan model yang dilatih sebelumnya dibekukan selama pelatihan dengan cara *transfer learning* dengan *2x2 max pool layer block 5*. Kemudian diintegrasikan kembali dan lapisan model yang sebelumnya dilatih bersama dengan model baru sehingga didapatkan model prediksi CNN. Model prediksi CNN yang sudah dilatih dilakukan *split* dengan pembagian 80% untuk *training* dan 20% untuk validasi dengan ukuran *output 7x7x512* dan dilakukan *flatten* untuk membentuk ulang fitur menjadi sebuah vektor yang akan digunakan sebagai *input* dari *fully-connected layer* yang berisikan jumlah neuron yang sama dengan kelas dataset dan kemudian dilakukan fungsi aktivasi *softmax* yaitu untuk mengonversi angka menjadi probabilitas yang berjumlah satu, dengan demikian mengeluarkan vektor yang mewakili distribusi probabilitas dari daftar hasil potensial. Kemudian bobot *output layer* (*training* dan validasi) hasil dari aktivasi *softmax* dilakukan *retrain* sehingga didapatkan nilai *loss* dimana jika nilai *loss* (*testing loss* dan *testing accuracy*) diatas 1,000 maka akan didapatkan nilai *epochs* baru dan apabila nilai dibawah 1,000 maka akan dilakukan proses *retrain*. Setelah didapatkan nilai *epochs*

baru (*training* dan *testing*) dilakukan kembali proses *retrain* dari bobot *epochs* yang didapatkan. Jika nilai *epochs* hasil *retrain* bernilai kurang dari *epochs* sebelum proses *retrain* maka akan dilakukan kembali proses *retrain* dan apabila *epochs* hasil *retrain* bernilai diatas *epochs* sebelum proses *retrain* maka akan didapatkan hasil prediksi dari model CNN.

3.1.4. Pengujian Confusion Matrix Multi Class

Confusion matrix multi class adalah metode buat mengukur kemampuan sistem pada pembagian terstruktur mengenai data menggunakan cara menghitung nilai aktualnya dibandingkan menggunakan jumlah semua data & akan dihasilkan nilai akurasi. Alasan dipilihnya confusion matrix multi class merupakan lantaran data diklasifikasikan sebagai beberapa kelas.

Metode *confusion matrix multi class* mempunyai 4 istilah untuk menampilkan hasil klasifikasi yaitu TP (true positive) yang berarti jumlah data positif yang terklasifikasi benar oleh sistem, TN (true negative) berarti jumlah data negatif yang terklasifikasi benar oleh sistem, FP (false positive) berarti jumlah data positif yang terklasifikasi salah oleh sistem, FN (false negative) berarti jumlah data negatif yang terklasifikasi salah oleh sistem.

Perhitungan akurasi dalam metode ini diperoleh berdasarkan jumlah nilai *diagonal confusion matrix* dibagi menggunakan total nilai prediksi.

3.2. Analisis Kebutuhan

Analisis kebutuhan merupakan proses analisa terhadap kebutuhan yang dibutuhkan dalam pembuatan penelitian dengan tujuan mendapatkan hasil yang baik. Analisis kebutuhan yang digunakan adalah sebagai berikut:

3.2.1. Jenis dan Sumber Data

Jenis data pada penelitian yang dilakukan adalah data kuantitatif. Data kuantitatif adalah data yang dapat diukur oleh numerik. Penelitian yang dilakukan menggunakan data kuantitatif dikarenakan data yang akan diambil berupa citra buah jambu kristal yang dicari nilai matriksnya dan dapat dihitung menggunakan rumus.

Sumber data yang akan digunakan dalam penelitian adalah data primer, dikarenakan data yang akan digunakan dalam penelitian adalah citra buah jambu kristal yang diambil langsung dari kebun buah jambu kristal. Hal ini menunjukkan kesesuaian dengan pengertian data primer, dimana data primer atau data mentah adalah data pertama kali yang dikumpulkan oleh peneliti melalui upaya pengambilan data di lapangan langsung.

3.2.2. Prosedur Pengumpulan Data

Proses pengumpulan data, dimulai dengan melakukan foto untuk menghasilkan citra buah jambu kristal. Kemudian memilah citra jambu kristal yang matang dengan jambu kristal yang belum matang.

3.2.3. Analisis Kebutuhan Data

Analisis kebutuhan sistem bertujuan untuk mendeskripsikan kebutuhan apa saja yang dilakukan sistem agar sistem dapat berjalan. Untuk memudahkan analisis kebutuhan sistem, kebutuhan sistem dibagi menjadi yaitu, kebutuhan fungsional dan kebutuhan nonfungsional.

3.2.4. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang akan dikerjakan oleh sistem. Kebutuhan fungsional yang akan dimiliki oleh sistem adalah:

1. Sistem dapat menampilkan gambar buah jambu kristal.
2. Sistem dapat dapat mengklasifikasikan buah jambu kristal matang dan belum matang.
3. Sistem dapat memberi nilai sebagai tingkat akurasi terhadap model yang dibuat.

3.2.5. Kebutuhan Non-Fungsional

1. Analisis kebutuhan perangkat keras

Penelitian dilakukan dengan dukungan perangkat keras, yaitu satu buah unit laptop ASUS TUF Gaming FX505GT dengan spesifikasi sebagai berikut:

- a. Intel Core i5 9300H-2.4Ghz upto 4.1Ghz
- b. NVIDIA GeForce GTX1650-4GB
- c. *Random Access Memory* (RAM) 8 GB DDR 4
- d. *Solid State Disk* 512 GB NVME
- e. Resolusi *monitor* 1920x1080 IPS

2. Analisis kebutuhan perangkat lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

- a. Windows sebagai *operating system*
- b. *Phyton* sebagai Bahasa pemrograman
- c. *PhyCharm* sebagai *Development Enviroment*

3. Analisis kebutuhan pengguna

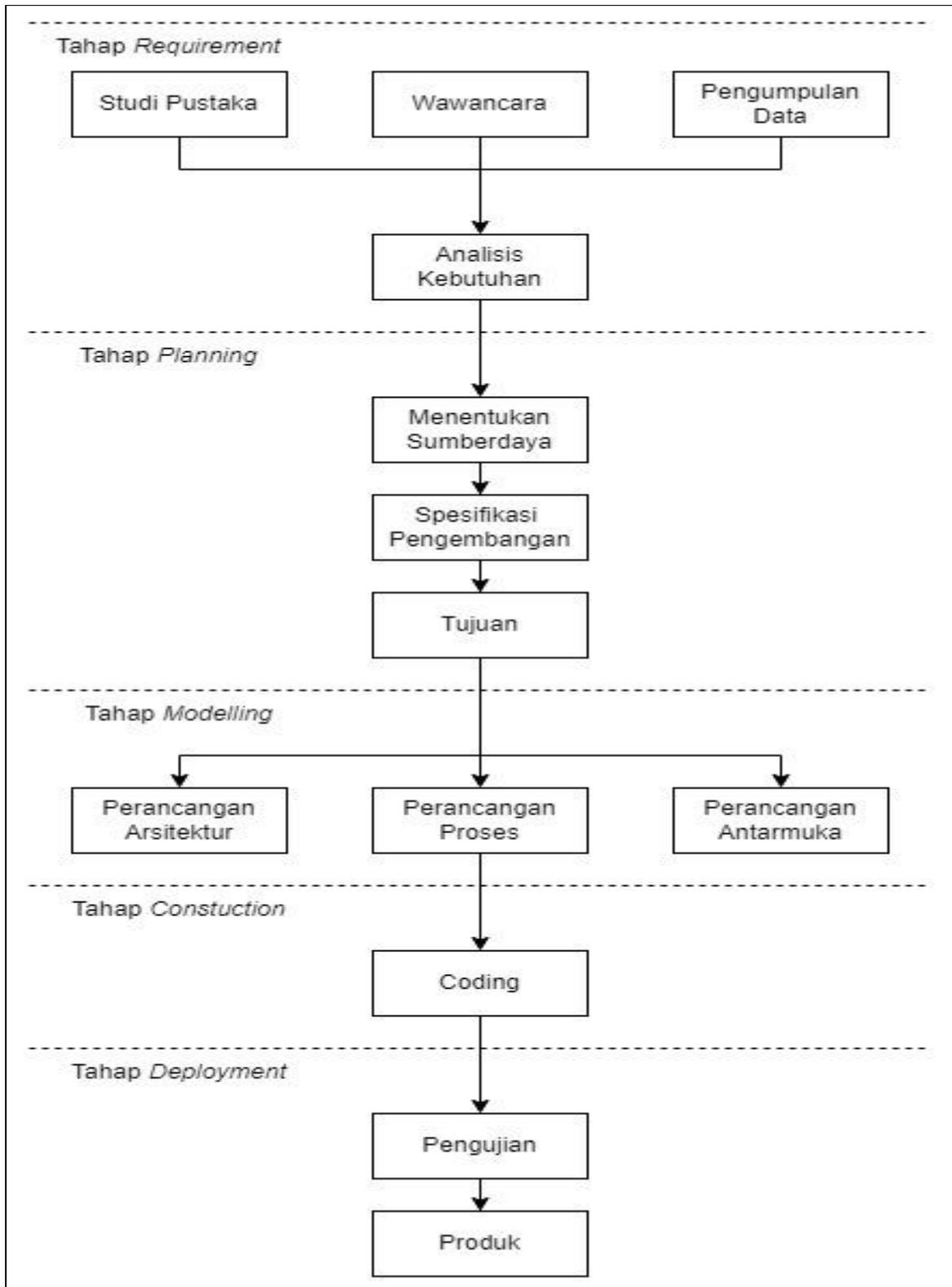
Kebutuhan pengguna dalam *system* ini adalah sebagai berikut:

Pengguna dapat membedakan buah jambu kristal yang matang dan belum matang.

3.3. Tahap Pengembangan Sistem

Metodologi pengembangan sistem yang digunakan pada penelitian ini yaitu metode *waterfall*. Metodologi *waterfall* merupakan salah satu model dalam perancangan piranti lunak. Model *waterfall* adalah sebuah contoh dari proses perencanaan, dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. Proses dari metode *waterfall* antara lain *Communication, Planning, Modeling,*

Construction dan Deployment (Pressman, 2010). Untuk gambaran proses pengembangan sistem dapat dilihat pada Gambar 3.20.



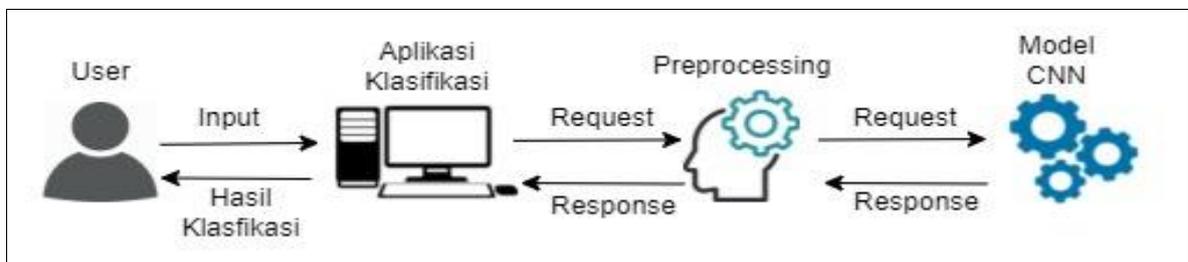
Gambar 3.20. Metode Pengembangan Sistem

3.3.1. Membangun Waterfall

Membangun *waterfall* adalah proses yang dibuat dengan rancangan temporer dengan sesuai kebutuhan klien berdasarkan hasil wawancara serta masukan saat berkomunikasi yang digunakan untuk pembangunan arsitektur sistem dan persyaratan fungsional.

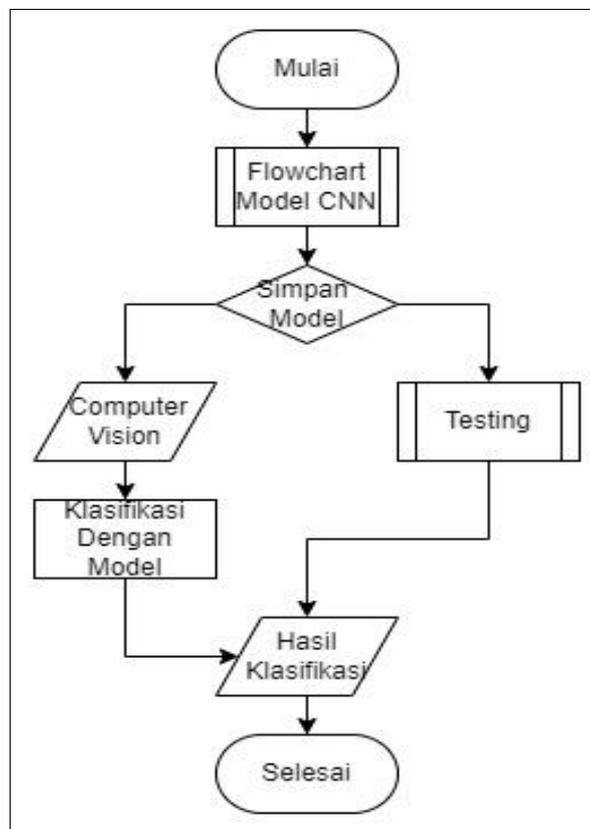
3.3.2. Membangun Arsitektur Sistem

Arsitektur sistem adalah gambaran jalannya sistem, pada penelitian ini proses awal pada sistem yaitu *user* melakukan *input* foto jambu kristal terhadap aplikasi klasifikasi kematangan buah jambu kristal. Kemudian sistem akan melakukan *preprocessing* pada foto tersebut, setelah itu data input akan dilakukan proses perhitungan dan proses klasifikasi menggunakan metode *convolutional neural network*. Data hasil perhitungan dan klasifikasi tersebut akan disimpan dalam model *convolutional neural network*, dimana hasilnya akan ditampilkan pada aplikasi klasifikasi kematangan jambu kristal kepada *user*. Berikut contoh arsitektur sistem pada penelitian ini ditampilkan pada Gambar 3.21.



Gambar 3.21. Arsitektur Sistem

3.3.3. Flowchart Proses Keseluruhan Sistem

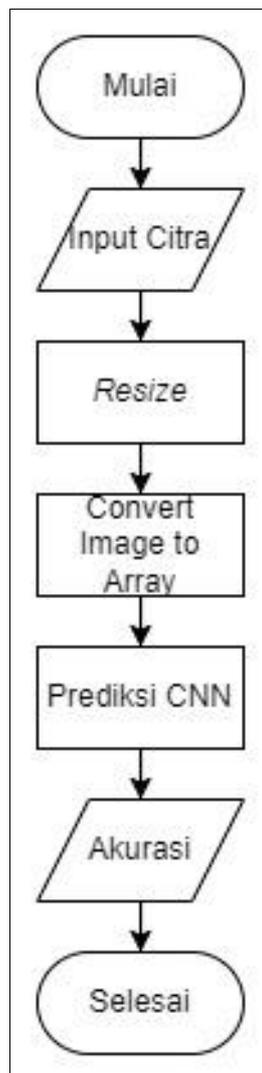


Gambar 3.22. Flowchart Keseluruhan Sistem

Gambar 3.22 menunjukkan proses keseluruhan pada penelitian ini. Pertama input data jambu kristal yang digunakan sebagai media pada proses *preprocessing*, kemudian dilakukan perancangan arsitektur CNN dan dilakukan pembelajaran agar model *convolutional neural network* dapat mengetahui objek citra jambu kristal matang dan jambu kristal belum matang berdasarkan tiap indeks dari kelas yang dimasukkan. Proses pembelajaran terus dilakukan hingga didapatkan model *convolutional neural network* yang cocok dengan tujuan yang diinginkan. Setelah didapatkan model *convolutional neural network* yang diinginkan, kemudian model disimpan dan dilakukan proses klasifikasi dengan *computer vision* dan *testing*.

Pada penelitian ini klasifikasi dengan *computer vision* menggunakan kamera sebagai sensor untuk memperoleh image objek jambu kristal, lalu diklasifikasikan dengan model CNN yang sudah disimpan dengan melihat dari informasi warna, bentuk dan identitas dari image objek jambu kristal sehingga diperoleh hasil klasifikasi jambu kristal matang atau belum matang.

Pada tahap klasifikasi *testing* bertujuan sebagai pengujian untuk mengetahui kinerja dari model *convolutional neural network* menggunakan data diluar dataset dalam pengklasifikasian citra jambu kristal matang dan jambu kristal dan belum matang.

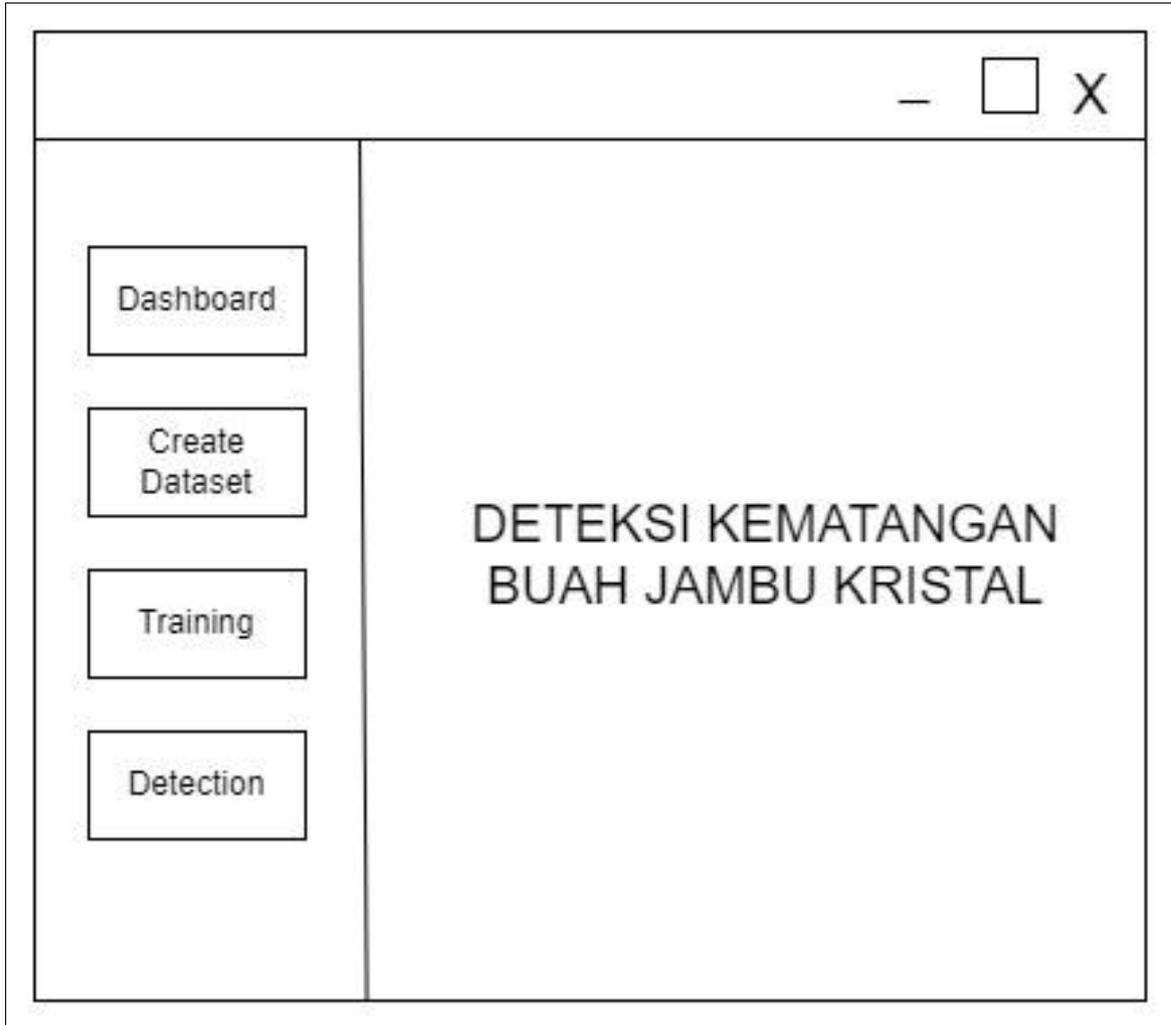


Gambar 3.23. Flowchart Testing

Pada tahap dimulai dengan menginputkan gambar yang akan diuji kemudian gambar dilakukan *resize* dengan ukuran 224x224 piksel sesuai ukuran citra yang dipakai sebagai data input ketika proses *training*. Selanjutnya citra diubah ke array dengan tujuan nilai array dicocokkan dengan model CNN yang sudah di-*training*, sehingga memunculkan memunculkan hasil prediksi antara buah jambu kristal matang atau jambu kristal belum matang dan akan diketahui akurasi dari hasil klasifikasi tersebut.

3.3.4. Perancangan Antarmuka

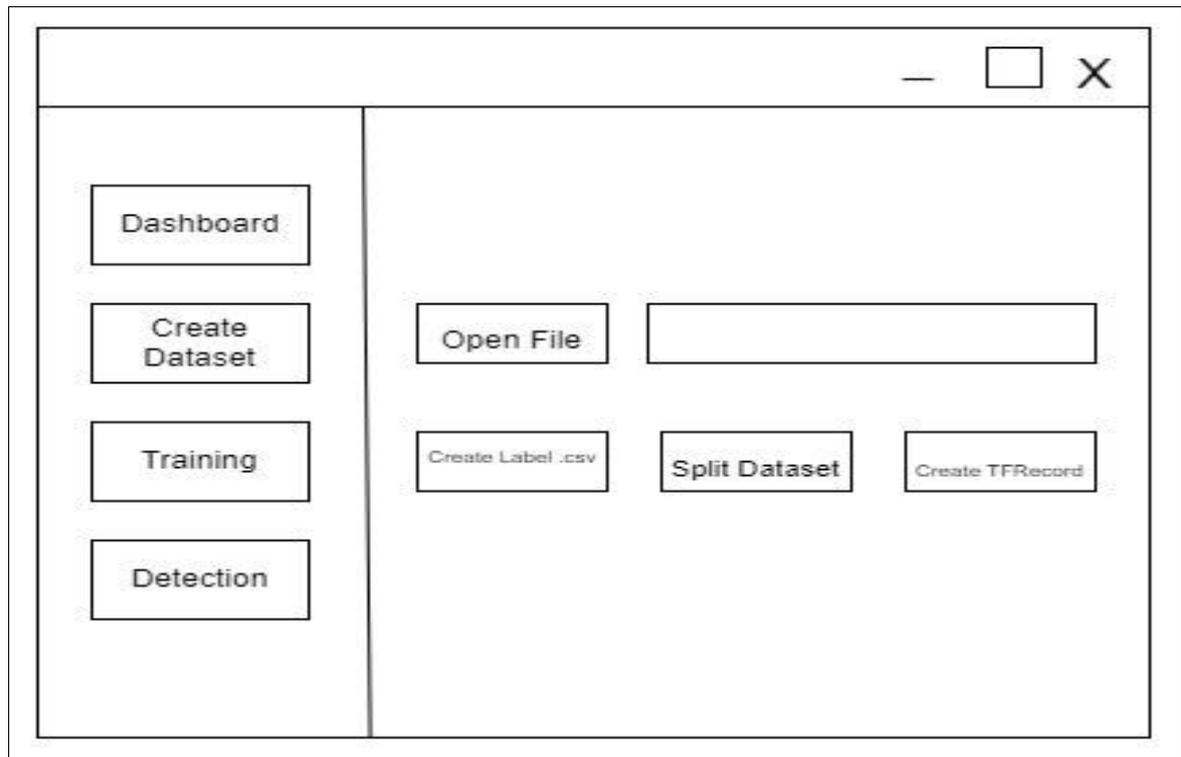
a. Antarmuka Halaman Utama



Gambar 3.24. Antarmuka Halaman Utama

Rancangan antarmuka halaman utama ini akan menampilkan tampilan pertama kali ketika pengguna menggunakan aplikasi deteksi kematangan buah jambu kristal. Pada halaman ini menampilkan judul dari aplikasi.

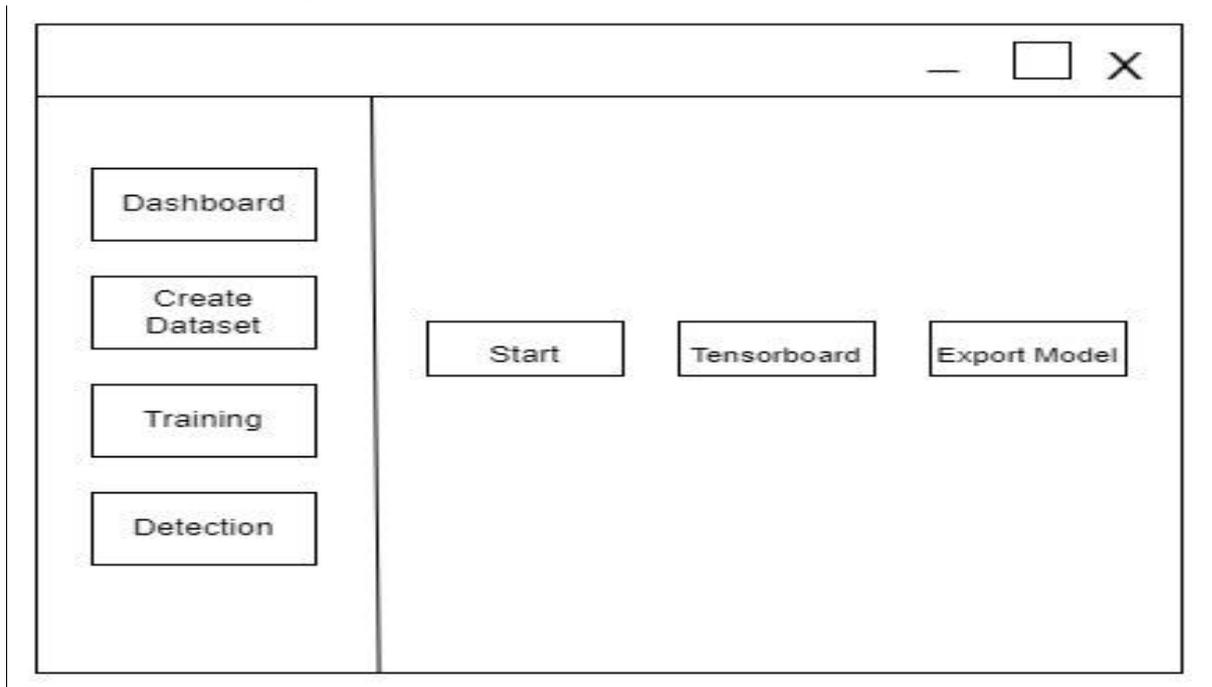
b. Antarmuka Membuat Dataset



Gambar 3.25. Antarmuka Membuat Dataset

Dapat dilihat rancangan antarmuka membuat dataset terdapat fungsi pemanggilan file *excel* dalam bentuk label. Kemudian dibawahnya terdapat tiga tombol fungsi yang masing-masing adalah membuat file *excel* tersebut menjadi *.csv*, *split dataset training* dan *testing* 80%:20% dan membuat *TFRecord* (*tensorflow record*) yang berguna untuk proses pelatihan model.

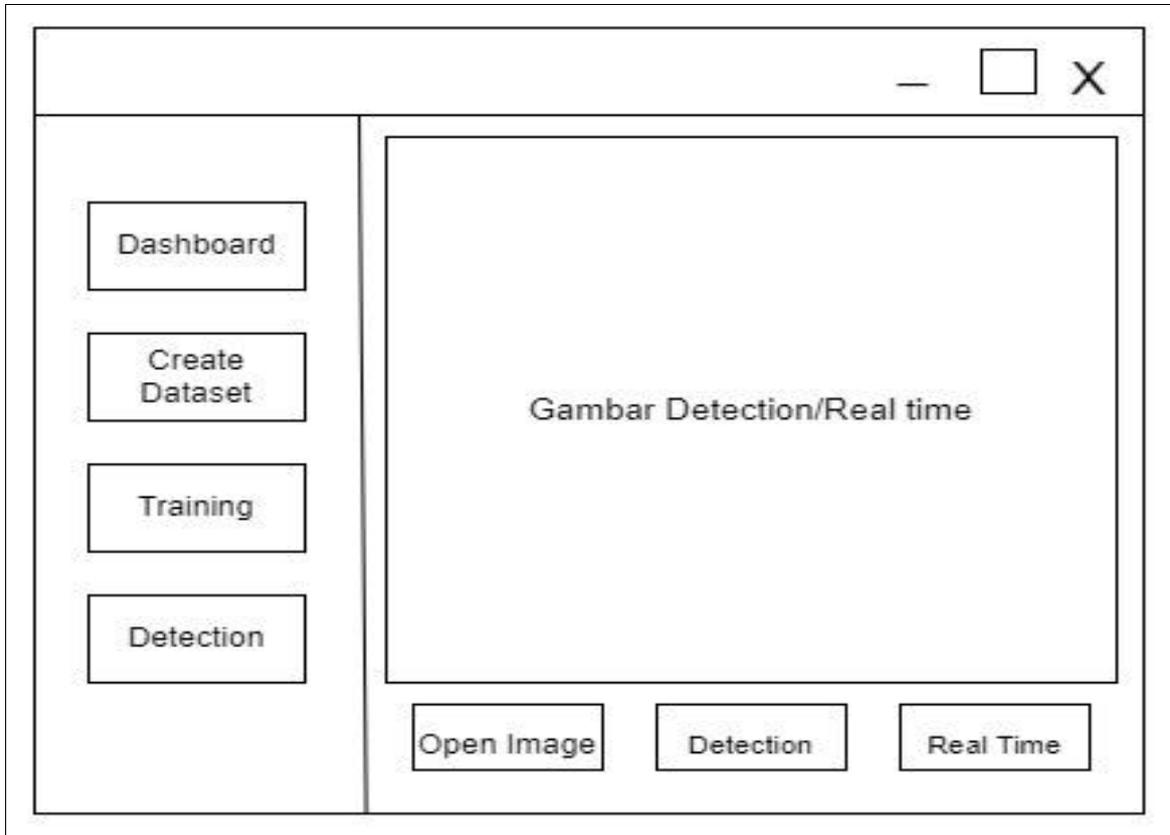
c. Antarmuka Training



Gambar 3.26. Antarmuka Training

Pada antarmuka *training* menampilkan tombol *start training* untuk memulai *training* data yang sudah diolah, *tensorboard* yang akan menampilkan hasil dari *training* dan *export model* hasil dari *training*.

d. Antarmuka Deteksi Jambu



Gambar 3.27. Antarmuka Deteksi Jambu

Dapat dilihat pada rancangan antarmuka deteksi jambu diatas, akan ditampilkan hasil klasifikasi dan nilai akurasi dari *testing* dan *real time*. Kemudian dibawahnya ada tiga tombol fungsi pertama *open image* untuk membuka gambar yang digunakan oleh *testing*, kemudian *detection* dan *real time*.

3.4. Perancangan Pengujian Model

Proses pengujian pada penelitian ini menggunakan pengujian model yang dilakukan untuk menguji kebenaran dari dataset pada metode *convolutional neural network*. Lakukan pengujian untuk menentukan apakah sistem bekerja dengan baik. Proses pengujian menggunakan dua divisi data, yaitu data latih dan data uji. Distribusi data latih dan data uji adalah 80% dan 20%. Pada data pelatihan, kumpulan data berlabel digunakan untuk melakukan proses pelatihan pada sistem, dan model akan dibangkitkan nanti. Model ini akan digunakan dalam proses klasifikasi tahap pengujian menggunakan data yang tidak berlabel. Tahap pengujian digunakan untuk mengetahui seberapa baik model yang dihasilkan dari pelatihan. Proses pengujian dalam penelitian ini menggunakan matriks *confussion matrix multi class*. Pengujian menggunakan *confussion matrix multi class* untuk menguji mengukur akurasi. Rancangan metode pengujian pada penelitian ini dapat dilihat pada Tabel 3.3

Tabel 3.3 Pengujian *Confussion Matrix Multi Class*

		Kelas Aktual		Total Prediksi
		Matang	Belum Matang	
Kelas Prediksi	Matang			
	Belum Matang			
Total Aktual				

BAB IV HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Berikut merupakan hasil implementasi dari perancangan program. Hasil dari penelitian ini terdiri dari dua bagian yaitu implementasi klasifikasi kematangan buah jambu kristal dan implementasi aplikasi. Setelah tahap implementasi akan dilakukan pengujian terhadap aplikasi serta dilihat juga kekurangan yang ada pada aplikasi untuk pengembangan selanjutnya.

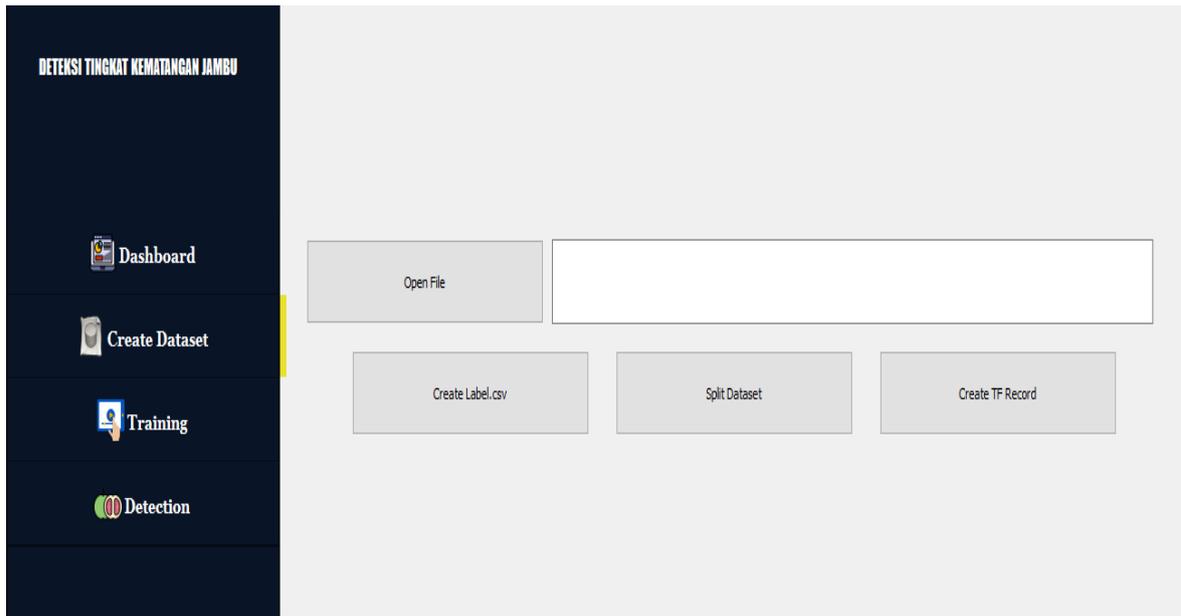
4.1.1. Pembentukan Model

Membuat sebuah file direktori *folder* yang berisi gambar jambu kristal matang dan belum matang. Data yang digunakan dalam penelitian ini berjumlah 572 citra jambu kristal yang terdiri dari 309 citra jambu kristal matang dan 263 jambu kristal belum matang. Dalam penelitian ini untuk mendapatkan informasi terhadap citra dan mempermudah pengenalan jambu kristal matang dan belum matang menggunakan *histogram equalization*, dimana *histogram equalization* dapat meningkatkan kontras citra dan kualitas citra.

```
def cont_stretch(im, levels):  
  
    im_out = np.zeros((im.shape[0],im.shape[1],3), dtype=np.uint8)  
    a, b = 0, levels-1  
    c, d = im.min(), im.max()  
    if imutil.is_gray(im):  
  
        im_out = np.zeros((im.shape[0],im.shape[1]),  
dtype=np.uint8)  
        im = imutil.correct_gray(im)  
        h, w = im.shape  
        im_out[0:h, 0:w] = (im[0:h, 0:w] - c)*((b - a)/(d-c)) + a  
    else:  
  
        h, w, ch = im.shape  
        img = imutil.rgb2ycbcr(im)  
        dMat = img[:, :, 0]  
        dMat[0:h, 0:w] = (dMat[0:h, 0:w] - c)*((b - a)/(d-c)) + a  
        img[:, :, 0] = dMat  
        im_out = imutil.ycbcr2rgb(img)  
  
    return im_out
```

Modul Program 4.1 Source Code Histogram Equalization

Dalam direktori *folder* yang berisi gambar jambu kristal tersebut dibuat folder yang didalamnya berisi anotasi dari gambar tersebut. Anotasi berisi keterangan label gambar jambu kristal matang dan belum matang berekstensi *xml* dimana dalam pembentukan anotasi menggunakan aplikasi LabelImg (*library* pengolahan untuk *labelling* gambar).



Gambar 4.1. Halaman *Create Dataset*

Halaman *create dataset* berisikan tombol *open file*, *create label to csv*, *split dataset* dan *create TFRecord*. Tombol *open file* digunakan untuk membaca direktori file berisi anotasi *image*. Tombol *create label to csv* untuk mengubah nilai anotasi *xml* kedalam bentuk *csv*. Tombol *split dataset* untuk membagi data *csv* menjadi 2 file yaitu data *training* dan data *testing* dimana masing-masing berbentuk *csv*. Tombol *create TFRecord* untuk membuat file *TFRecord* dari data *training* dan *testing* berbentuk *csv*.

```
def xml_to_csv(path):
    xml_list = []
    for xml_file in glob.glob(path + '/*.xml'):
        tree = ET.parse(xml_file)
        root = tree.getroot()
        for member in root.findall('object'):
            value = (root.find('filename').text,
                    int(root.find('size')[0].text),
                    int(root.find('size')[1].text),
                    member[0].text,
                    int(member[4][0].text),
                    int(member[4][1].text),
                    int(member[4][2].text),
                    int(member[4][3].text))
            xml_list.append(value)
    column_name = ['filename', 'width', 'height', 'class', 'xmin',
                  'ymin', 'xmax', 'ymax']
    xml_df = pd.DataFrame(xml_list, columns=column_name)
    return xml_df
```

Modul Program 4.2 *Source Code Fungsi Konversi XML ke CSV*

File anotasi dalam bentuk *xml* dikonversi ke bentuk *csv*. Proses konversi *xml* ke *csv* dengan membaca direktori tempat *file xml* tersimpan. Jumlah *file xml* dalam folder anotasi sesuai dengan jumlah data citra jambu kristal matang dan belum matang yaitu 572. Dalam file *csv* tersebut terdapat informasi nama file, lebar, tinggi, kelas, *xmin*, *ymin*, *xmax* dan *ymax* dari proses pelabelan citra jambu kristal.

	A	B	C	D	E	F	G	H	I
1	filename	width	height	class	xmin	ymin	xmax	ymax	
2	BelumMatang (1).JPG	4032	3024	BelumMatang	1042	864	2579	2448	
3	BelumMatang (10).JPG	4032	3024	BelumMatang	1115	853	2479	2253	
4	BelumMatang (100).JPG	4032	3024	BelumMatang	1305	690	2894	2264	
5	BelumMatang (101).JPG	4032	3024	BelumMatang	1310	617	2905	2159	
6	BelumMatang (102).JPG	4032	3024	BelumMatang	1273	617	2826	2180	
7	BelumMatang (103).JPG	4032	3024	BelumMatang	1215	843	2800	2348	
8	BelumMatang (104).JPG	4032	3024	BelumMatang	1147	722	2668	2264	
9	BelumMatang (105).JPG	4032	3024	BelumMatang	1147	680	2710	2253	
10	BelumMatang (106).JPG	4032	3024	BelumMatang	1310	680	2815	2180	
11	BelumMatang (107).JPG	4032	3024	BelumMatang	1252	648	2726	2243	
12	BelumMatang (108).JPG	4032	3024	BelumMatang	1200	685	2689	2180	
13	BelumMatang (109).JPG	4032	3024	BelumMatang	1073	790	2615	2259	
14	BelumMatang (11).JPG	4032	3024	BelumMatang	1100	848	2436	2259	
15	BelumMatang (110).JPG	4032	3024	BelumMatang	794	790	2415	2385	
16	BelumMatang (111).JPG	4032	3024	BelumMatang	805	774	2421	2353	
17	BelumMatang (112).JPG	4032	3024	BelumMatang	994	796	2579	2306	
18	BelumMatang (113).JPG	4032	3024	BelumMatang	889	901	2589	2348	
19	BelumMatang (114).JPG	4032	3024	BelumMatang	989	769	2589	2243	

Gambar 4.2. Isi File CSV

File yang terdapat dalam folder anotasi akan menjadi satu file csv yang berisikan data hasil labelling gambar. Filename merupakan nama file dari gambar. Ukuran gambar disimpan dalam kolom width berupa lebar gambar dan height berupa tinggi gambar. Class merupakan informasi label gambar, dalam penelitian ini menggunakan 2 kelas yaitu Matang dan BelumMatang. xmin, ymin, xmax dan ymax adalah informasi titik - titik dari proses labelling gambar. Isi dari file csv berjumlah sesuai dengan file xml yaitu 572 yang terdiri dari 309 kelas jambu kristal matang dan 263 kelas jambu kristal belum matang.

```

data_all= pd.read_csv('label.csv')
X=data_all.drop('class', axis=1)y=data_all['class']
X_train, X_test, y_train, y_test = train_test_split(X, y,
train=0.80, test_size=0.20, random_state=1010)
y_train=pd.DataFrame(y_train)
y_test=pd.DataFrame(y_test)
data_train=pd.merge(X_train,y_train,left_index=True,right_index=True)
e)

data_test=pd.merge(X_test,y_test,left_index=True,right_index=True)
data_train.to_csv('data/train.csv')
data_test.to_csv('data/test.csv')

```

Modul Program 4.3 Source Code Split Data Train dan Testing

File label csv terbagi menjadi dua yaitu data train dan data test. File data train berisi data *training* dan file test berisi data *testing*. File *training* berisi 457 data dengan rincian 247 data citra jambu kristal matang dan 210 citra jambu kristal belum matang. File *testing* berisi 115 data citra dengan rincian 62 data citra jambu kristal matang dan 53 data citra jambu belum matang. Dalam penelitian ini menggunakan rasio perbandingan data *training* dan data *testing* 80:20. `train=0.80`, `test_size=0.20`, digunakan untuk membagi data training dengan rasio 0.8 dan data testing dengan rasio 0.2 Berikut merupakan hasil dari pembagian data *training* dan *testing* berupa file train.csv dan test.csv.

A	B	C	D	E	F	G	H	I	J
	filename	width	height	xmin	ymin	xmax	ymax	class	
158	BelumMatang (241).JPG	4032	3024	1005	806	2473	2374	BelumMatang	
267	Matang (102).JPG	4032	3024	1171	719	2799	2476	Matang	
390	Matang (213).JPG	4032	3024	1365	858	2730	2343	Matang	
413	Matang (234).JPG	4032	3024	1080	886	2580	2358	Matang	
365	Matang (191).JPG	4032	3024	1265	758	2858	2358	Matang	
554	Matang (83).JPG	4032	3024	932	748	2585	2348	Matang	
119	BelumMatang (206).JPG	4032	3024	1105	774	2515	2122	BelumMatang	
234	BelumMatang (73).JPG	4032	3024	1131	674	2668	2269	BelumMatang	
370	Matang (196).JPG	4032	3024	1037	843	2658	2479	Matang	
561	Matang (9).JPG	4032	3024	1161	1024	2690	2471	Matang	
255	BelumMatang (92).JPG	4032	3024	1105	553	2689	2169	BelumMatang	
201	BelumMatang (43).JPG	4032	3024	1226	774	2989	2338	BelumMatang	
509	Matang (42).JPG	4032	3024	966	733	2628	2529	Matang	
542	Matang (72).JPG	4032	3024	980	729	2494	2267	Matang	
134	BelumMatang (22).JPG	4032	3024	921	769	2436	2211	BelumMatang	
243	BelumMatang (81).JPG	4032	3024	1268	653	2763	2227	BelumMatang	
476	Matang (291).JPG	4032	3024	1130	879	2365	2279	Matang	
431	Matang (250).JPG	4032	3024	1137	801	2587	2286	Matang	
551	Matang (80).JPG	4032	3024	913	805	2471	2433	Matang	

Gambar 4.3. Isi File *Train.csv*

A	B	C	D	E	F	G	H	I	J
	filename	width	height	xmin	ymin	xmax	ymax	class	
364	Matang (190).JPG	4032	3024	1265	715	2915	2293	Matang	
136	BelumMatang (221).JPG	4032	3024	1147	780	2721	2311	BelumMatang	
78	BelumMatang (17).JPG	4032	3024	1257	680	2673	2201	BelumMatang	
408	Matang (23).JPG	4032	3024	794	667	2537	2586	Matang	
44	BelumMatang (139).JPG	4032	3024	1147	1080	2573	2380	BelumMatang	
454	Matang (271).JPG	4032	3024	1015	822	2437	2208	Matang	
170	BelumMatang (252).JPG	4032	3024	1073	864	2779	2522	BelumMatang	
538	Matang (69).JPG	4032	3024	1009	824	2556	2395	Matang	
360	Matang (187).JPG	4032	3024	1037	858	2437	2236	Matang	
391	Matang (214).JPG	4032	3024	1287	801	2651	2222	Matang	
462	Matang (279).JPG	4032	3024	1037	779	2587	2229	Matang	
292	Matang (125).JPG	4032	3024	1251	838	2699	2252	Matang	
562	Matang (90).JPG	4032	3024	880	657	2618	2305	Matang	
452	Matang (27).JPG	4032	3024	675	590	2580	2490	Matang	
362	Matang (189).JPG	4032	3024	1244	801	2801	2415	Matang	
226	BelumMatang (66).JPG	4032	3024	1342	748	2863	2396	BelumMatang	
14	BelumMatang (111).JPG	4032	3024	805	774	2421	2353	BelumMatang	
450	Matang (268).JPG	4032	3024	1030	851	2522	2236	Matang	

Gambar 4.4. Isi File *Testing.csv*

Dataset dalam bentuk file *train.csv* dan *test.csv* dikonversi dalam bentuk *TF record*. *TF record* berisi urutan string biner. *TF record* diimplementasikan untuk proses training menggunakan *library* Tensorflow. Dalam pembentukan file *TF record* diperlukan pendefinisian nama kelas dalam bentuk angka agar mempermudah pembacaan saat *training*.

```
def class_text_to_int(row_label):
    if row_label == 'BelumMatang':
        return 1
    elif row_label == 'Matang':
        return 2
    else:
        return None
```

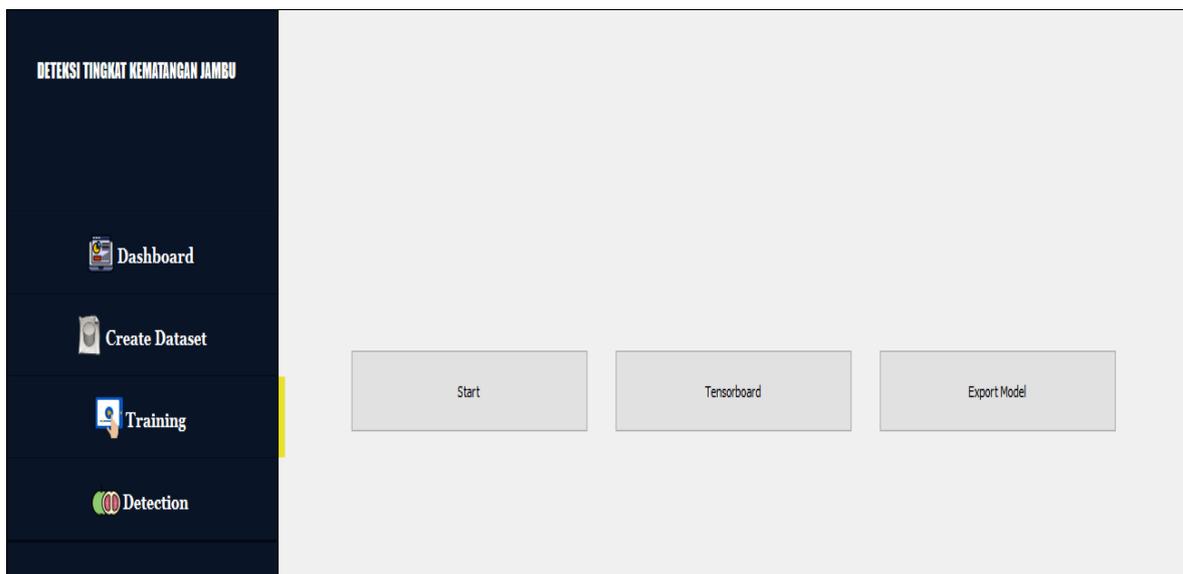
Modul Program 4.4 Source Code Pendefinisian Kelas

`row_label` sesuai dengan nama kelas pada saat labelling dilakukan. Data citra jambu kristal belum matang akan bernilai 1. Data citra jambu kristal matang akan bernilai 2. Jika `row_label` tidak didefinisikan sesuai dengan nama label kelas pada saat *labelling* maka data tidak akan terbaca dan mempengaruhi proses *training*.

```
def generate_tfrecord(self, csv_input, output_path):
    writer = tf.python_io.TFRecordWriter(output_path)
    path = os.path.join(os.getcwd(), 'Detection/images')
    examples = pd.read_csv(csv_input)
    grouped = self.split(generate, examples, 'filename')
    for group in grouped:
        tf_example = self.create_tf_example(generate, group, path)
        writer.write(tf_example.SerializeToString())
    writer.close()
    output_path = os.path.join(os.getcwd(), output_path)
```

Modul Program 4.5 Source Code Generate TFRecord

Pembuatan TF *record* menggunakan fungsi dari *library* tensorflow. Definisi variabel *writer* sebagai tempat direktori TF *record*. Pada proses pembuatan TF *record* diperlukan pembacaan folder yang berisi citra jambu kristal matang dan tnelum matang. File csv yang berisi filename, xmins, xmaxs, ymins, ymaxs, dan class dikonversi, dimana xmins dan xmaxs dikonversi menjadi lebar gambar dan ymins, juga ymaxs dikonversi menjadi tinggi gambar. Gambar yang dibentuk sesuai dengan pelabelan yang dilakukan saat proses labelling. File yang dikonversi adalah train.csv dan test.csv menghasilkan train.*record* dan test.*record*.



Gambar 4.5. Halaman Training

Pada halaman *training* terdapat tombol *start*, *tensorboard* dan *export model*. Tombol *start* digunakan untuk memulai proses *training* menggunakan *command prompt*. Tombol *tensorboard* untuk melihat grafik selama proses *training* yang sudah berlangsung. Tombol *export model* berfungsi untuk membuat model hasil dari proses *training*.

Kode program fungsi pada proses *convolution neural network* menggunakan konfigurasi *training data* `ssd_mobilenet.config`. Konfigurasi *mobilenet* merupakan arsitektur *convolution neural network* VGG16 yang digunakan untuk membuat model. Beberapa pengaturan utama konfigurasi `ssd_mobilenet.config` adalah jumlah *batch size* dan *num steps* yang saat proses *training*.

```

train_config: {
  batch_size: 4
  optimizer {
    rms_prop_optimizer: {
      learning_rate: {
        exponential_decay_learning_rate {
          initial_learning_rate: 0.004
          decay_steps: 800720
          decay_factor: 0.95
        }
      }
    }
    momentum_optimizer_value: 0.9
    decay: 0.9
    epsilon: 1.0
  }
}
fine_tune_checkpoint: ""
from_detection_checkpoint: true
num_steps: 10000
data_augmentation_options {
  random_horizontal_flip {
  }
}
data_augmentation_options {
  ssd_random_crop {
  }
}
}

```

Modul Program 4.6 Source Code Konfigurasi SSD Mobilenet

Batch size adalah jumlah sampel data yang disebarkan ke *neural network*. Dalam penelitian ini, 612 data dan *batch size* adalah 4 maka algoritma menggunakan 4 sampel data pertama dari 612 data yang kita miliki (^{ke}1, ^{ke}2, ^{ke}3, ^{ke}4) lalu disebarkan atau ditraining oleh *Neural Network* sampai selesai kemudian mengambil kembali 4 sampel data kedua dari 612 data (^{ke}19, ^{ke}20, ^{ke}21, ^{ke}22, ^{ke}23, ^{ke}24, ^{ke}25, ^{ke}26, ^{ke}27, ^{ke}28, ^{ke}29) dan begitu seterusnya sampai 4 sampel data ke 153($612/4=153$).

Epoch adalah ketika dataset melalui pelatihan pada *neural network* sampai dikirim kembali ke awal putaran karena *epoch* terlalu besar untuk muat komputer, jadi kita harus membaginya menjadi potongan-potongan unit kecil (*batches*). *Num steps* tersebut dimaksudkan untuk diproses ketika kumpulan data lengkap telah melalui pelatihan data pada *neural network* sampai dikembalikan ke kepala untuk putaran. Ini juga bertujuan untuk memungkinkan mesin mempelajari cara mengenali data selama pelatihan. *Num steps* yang lebih tinggi tidak selalu berarti akurasi yang tinggi, mungkin ada *overfitting* atau *underfitting*. Dalam penelitian ini menggunakan langkah hingga 10.000 langkah.

Selama proses pelatihan data, data citra RGB sebelumnya akan ditransformasikan oleh aplikasi untuk mendapatkan nilai array berdasarkan nilai array perpixel yang bersangkutan. Gambar juga akan diubah ukurannya menjadi 224x224 piksel dengan 3 *channel* RGB. Selama proses pelatihan data, data citra RGB sebelumnya akan ditransformasikan oleh aplikasi untuk mendapatkan nilai array berdasarkan nilai array perpixel yang bersangkutan. Gambar juga akan diubah ukurannya menjadi 224x224 piksel dengan 3 *channel* RGB.

Kemudian data citra RGB akan dilakukan proses *feature learning* dan *classification* dimana pada *feature learning* terdapat proses *convolutional layer* dan *pooling* hingga nilai

array menjadi nilai vector baris melalui proses *flattening*. Pada proses *classification* terdapat proses *fully connected layer* dimana nilai vector baris akan saling terhubung sepenuhnya. Setelah terhubung adalah tahap klasifikasi menggunakan fungsi *softmax*. Fungsi ini akan membantu vector baris yang terhubung mengklasifikasikan input kedalam 2 kelas yaitu, Matang dan BelumMatang.

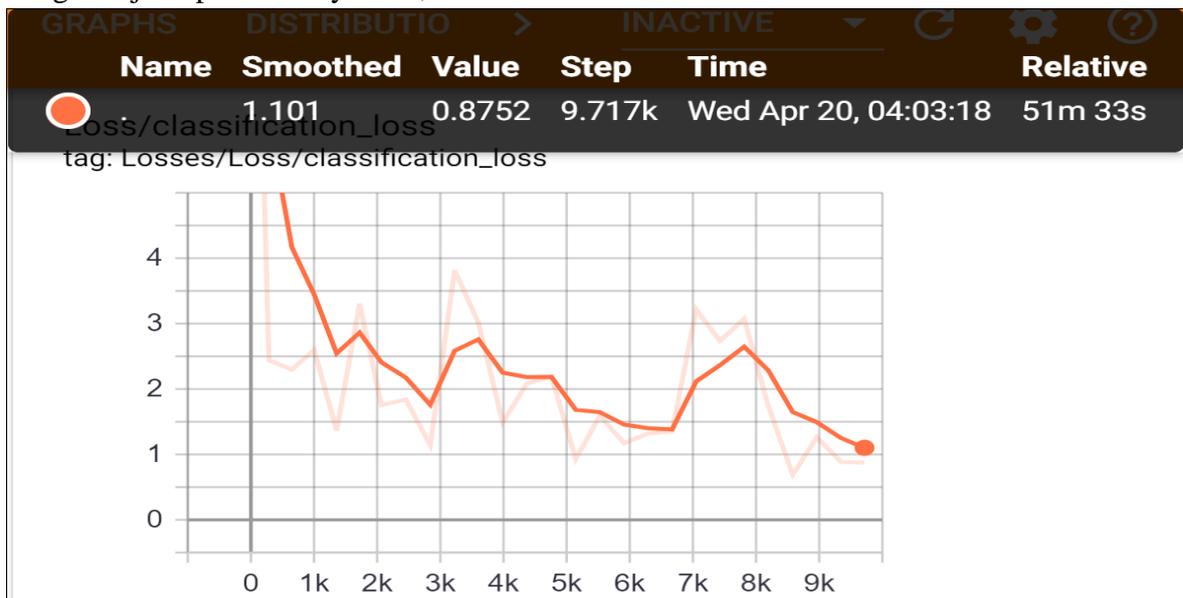
```

INFO:tensorflow:global step 9962: loss = 0.7232 (2.832 sec/step)
INFO:tensorflow:global step 9963: loss = 0.7845 (2.863 sec/step)
INFO:tensorflow:global step 9964: loss = 1.8398 (2.863 sec/step)
INFO:tensorflow:global step 9965: loss = 0.7733 (2.848 sec/step)
INFO:tensorflow:global step 9966: loss = 0.7317 (2.863 sec/step)
INFO:tensorflow:global step 9967: loss = 0.8878 (2.848 sec/step)
INFO:tensorflow:global step 9968: loss = 0.9922 (2.863 sec/step)
INFO:tensorflow:global step 9969: loss = 0.8768 (2.832 sec/step)
INFO:tensorflow:global step 9970: loss = 0.6652 (2.863 sec/step)
INFO:tensorflow:global step 9971: loss = 0.9989 (2.879 sec/step)
INFO:tensorflow:global step 9972: loss = 0.9939 (2.848 sec/step)
INFO:tensorflow:global step 9973: loss = 0.9992 (2.816 sec/step)
INFO:tensorflow:global step 9974: loss = 1.8921 (2.832 sec/step)
INFO:tensorflow:global step 9975: loss = 0.9921 (2.848 sec/step)
INFO:tensorflow:global step 9976: loss = 0.7852 (2.863 sec/step)
INFO:tensorflow:global step 9977: loss = 0.9849 (2.157 sec/step)
INFO:tensorflow:global step 9978: loss = 0.9455 (2.173 sec/step)
INFO:tensorflow:global step 9979: loss = 1.8765 (2.863 sec/step)
INFO:tensorflow:global step 9980: loss = 0.9755 (2.478 sec/step)
INFO:tensorflow:global step 9981: loss = 0.7961 (2.376 sec/step)
INFO:tensorflow:global step 9982: loss = 0.7544 (2.423 sec/step)
INFO:tensorflow:global step 9983: loss = 0.9757 (3.978 sec/step)
INFO:tensorflow:Recording summary at step 9983
INFO:tensorflow:global step 9984: loss = 0.9842 (2.798 sec/step)
INFO:tensorflow:global step 9985: loss = 0.6847 (2.188 sec/step)
INFO:tensorflow:global step 9986: loss = 0.7685 (2.118 sec/step)
INFO:tensorflow:global step 9987: loss = 0.8097 (2.894 sec/step)
INFO:tensorflow:global step 9988: loss = 0.7989 (2.879 sec/step)
INFO:tensorflow:global step 9989: loss = 0.9288 (2.832 sec/step)
INFO:tensorflow:global step 9990: loss = 0.8296 (2.879 sec/step)
INFO:tensorflow:global step 9991: loss = 0.8372 (2.863 sec/step)
INFO:tensorflow:global step 9992: loss = 0.9947 (2.832 sec/step)
INFO:tensorflow:global step 9993: loss = 0.9819 (2.847 sec/step)
INFO:tensorflow:global step 9994: loss = 1.1824 (2.894 sec/step)
INFO:tensorflow:global step 9995: loss = 1.8788 (2.879 sec/step)
INFO:tensorflow:global step 9996: loss = 0.7886 (2.894 sec/step)
INFO:tensorflow:global step 9997: loss = 0.8988 (2.208 sec/step)
INFO:tensorflow:global step 9998: loss = 0.9288 (2.124 sec/step)
INFO:tensorflow:global step 9999: loss = 1.8748 (2.832 sec/step)
INFO:tensorflow:global step 10000: loss = 0.8584 (2.118 sec/step)
INFO:tensorflow:Stopping Training.
INFO:tensorflow:Finished training! Saving model to disk.

```

Gambar 4.6. Proses Training

Pelatihan dimulai dengan memanggil *library training* dari tensorflow. Data pelatihan dengan 10.000 langkah membutuhkan waktu sekitar 1 hari. Output yang akan dihasilkan dalam data proses pelatihan akan dimasukkan ke dalam profil pelatihan. Nilai kerugian adalah ukuran kuantitatif dari penyimpangan antara perkiraan dan keluaran aktual (label). Nilai loss berbanding terbalik dengan akurasi model. Semakin besar nilai kerugiannya, semakin tidak akurat modelnya. Nilai loss selama proses pelatihan ini sesuai dengan tujuan penelitian yaitu 0,8752.



Gambar 4.7. Hasil Loss Classification

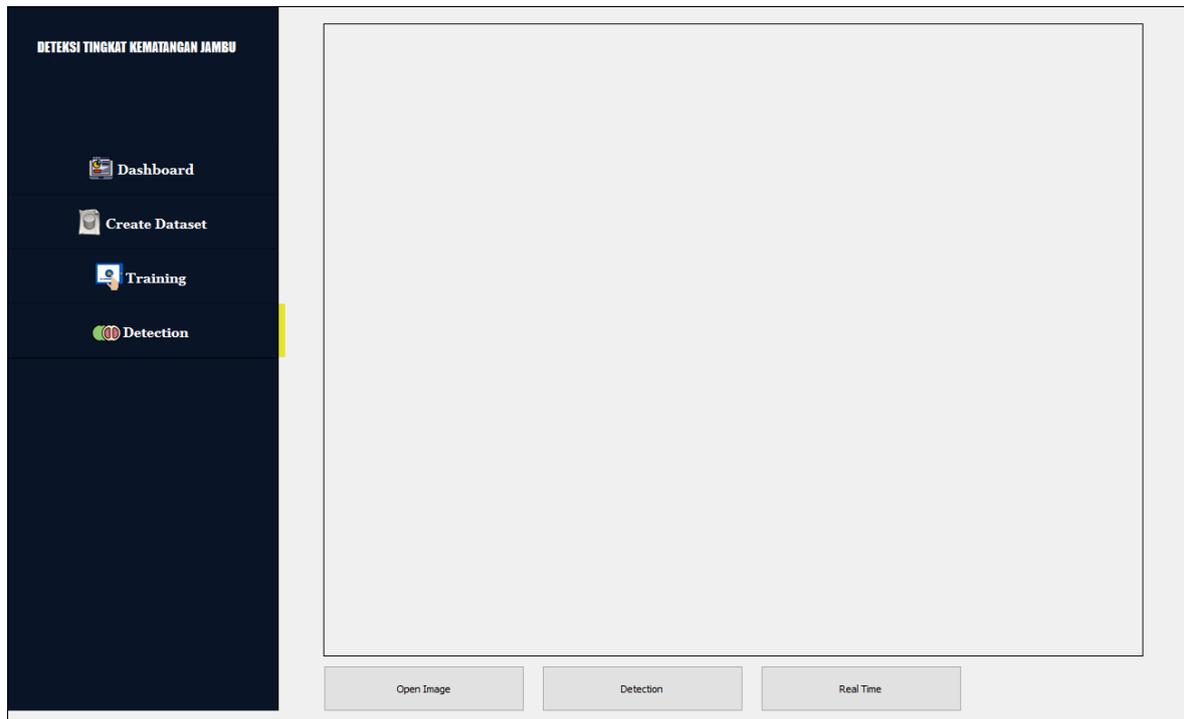
Hasil pelatihan diekspor dalam bentuk model dimana model yang digunakan sebagai pengenalan terhadap kematangan buah jambu kristal. Tahap selanjutnya adalah pengujian model. Pengujian model dilakukan dengan membandingkan model dan input gambar diluar data gambar *training* maupun *testing* saat proses pembentukan model.

```
def create_model(self, trained_checkpoint_prefix):
    pipeline_config = pipeline_pb2.TrainEvalPipelineConfig()
    with tf.gfile.GFile(FLAGS.pipeline_config, 'r') as f:
        text_format.Merge(f.read(), pipeline_config)
    if FLAGS.input_shape:
        input_shape = [
            int(dim) if dim != '-1' else None
            for dim in FLAGS.input_shape.split(',')
        ]
    else:
        input_shape = None
    exporter.export_inference_graph(FLAGS.input_type,
    pipeline_config, trained_checkpoint_prefix, FLAGS.output_directory, input_shape)
```

Modul Program 4.7 Source Code Create Model

4.1.2. Implementasi Model

Implementasi model dibuat dalam bentuk aplikasi deteksi kematangan jambu kristal menggunakan *computer vision*. Aplikasi deteksi terdiri dari beberapa halaman seperti pada rancangan antarmuka pada bab 3.



Gambar 4.8. Halaman Deteksi

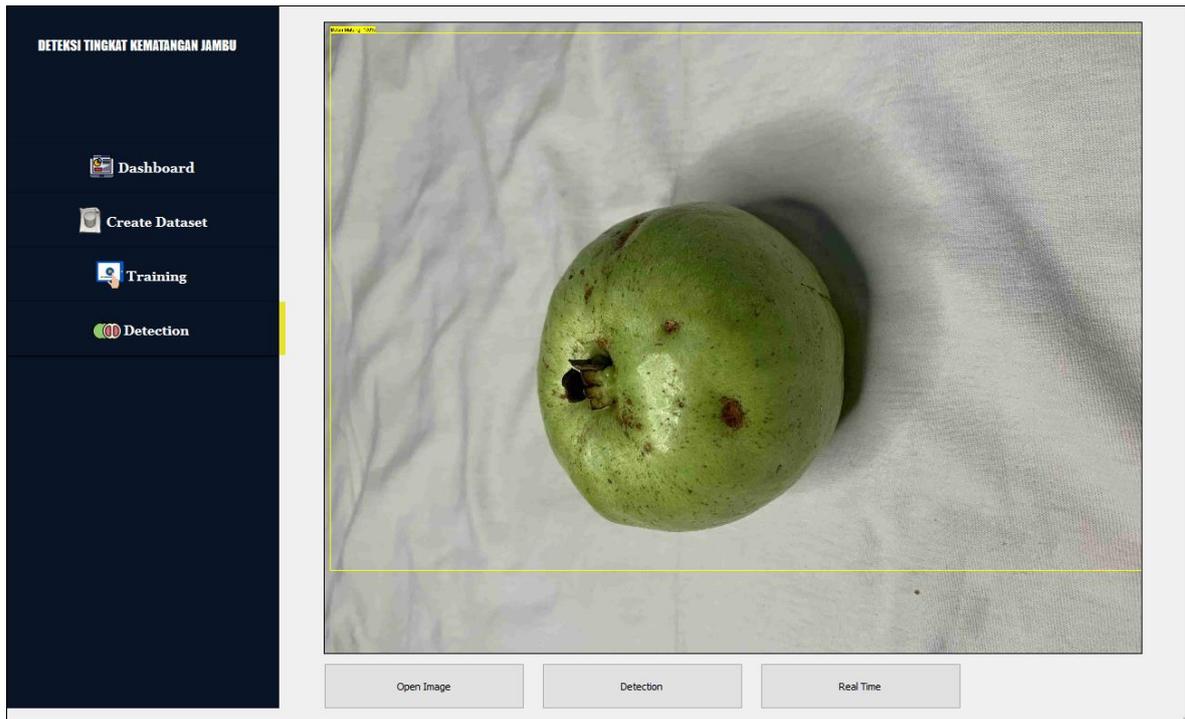
Halaman deteksi terdapat beberapa tombol yaitu tombol *open image*, tombol *detection* dan tombol *real time*. Tombol *open image* digunakan untuk memanggil citra jambu kristal yang sudah tersimpan pada direktori file PC/Laptop. Tombol *detection* digunakan dalam mendeteksi kematangan citra jambu kristal dari citra yang dipanggil pada

proses *open image*. Tombol *realtime* berguna untuk mendeteksi kematangan secara langsung menggunakan *camera/webcam*.

```
def loadImage(self, fname):  
    self.processedImage = fname.copy()  
    self.previewImage = cv2.resize(self.processedImage, (450, 450))  
    self.displayImage(self.previewImage)
```

Modul Program 4.8 Source Code Load Image

Source code diatas digunakan untuk memanggil dan membaca gambar input untuk ditampilkan di halaman berikutnya.



Gambar 4.9. Halaman Hasil Deteksi

Halaman hasil deteksi menampilkan hasil deteksi kematangan jambu kristal yang matang dan jambu kristal belum matang yang digunakan sebagai data *testing*.



Gambar 4.10. Halaman Deteksi *Realtime*

Halaman deteksi *realtime* diambil melalui kamera/*webcam* yang digunakan sebagai input citra jambu kristal yang disambungkan ke laptop/pc. Kemudian hasil kematangan akan ditampilkan beserta persentasenya.

```
def start_webcam(self):
    if self.capture is None:
        self.capture = cv2.VideoCapture(0)
        self.capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
        self.capture.set(cv2.CAP_PROP_FRAME_WIDTH, 1080)
    self.timer.start()
```

Modul Program 4.9 Source Code Start Webcam

Source code diatas adalah fungsi tombol *realtime* untuk menampilkan layer kamera ke komputer. Fungsi *self.capture* berfungsi untuk menangkap *frame* pada layer kamera untuk ditampilkan di layar komputer.

4.2. Pengujian Model

Dalam pengujian menggunakan *confusion matrix*. Cara ini dapat digunakan untuk mengukur performa pada kasus klasifikasi yang didalamnya minimal terdapat dua keluaran atau lebih. Penelitian ini menggunakan data sebanyak 40 citra jambu kristal yang diambil dari Kebun Larasati Garden dengan setiap citra jambu kristal memiliki 4 nilai empat kombinasi berbeda dalam table dari nilai prediksi dan nilai aktual diantaranya True Positif, True Negatif, False Positif dan False Negatif.

Tabel 4.1 Tabel Pengujian Model

No	Citra Jambu Kristal	Nilai Prediksi	Nilai Aktual
1.		Belum Matang	Belum Matang
2.		Matang	Belum Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

3.		Belum Matang	Belum Matang
4.		Belum Matang	Belum Matang
5.		Belum Matang	Belum Matang
6.		Belum Matang	Belum Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

7.		Belum Matang	Belum Matang
8.		Belum Matang	Belum Matang
9.		Belum Matang	Belum Matang
10.		Belum Matang	Belum Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

11.		Belum Matang	Belum Matang
12.		Belum Matang	Belum Matang
13.		Belum Matang	Belum Matang
14.		Belum Matang	Belum Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

15.		Belum Matang	Belum Matang
16.		Belum Matang	Belum Matang
17.		Belum Matang	Belum Matang
18.		Belum Matang	Belum Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

19.		Belum Matang	Belum Matang
20.		Belum Matang	Belum Matang
21.		Matang	Matang
22.		Matang	Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

23.		Matang	Matang
24.		Matang	Matang
25.		Belum Matang	Matang
26.		Matang	Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

27.		Matang	Matang
28.		Matang	Matang
29.		Matang	Matang
30.		Matang	Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

31.		Matang	Matang
32.		Matang	Matang
33.		Matang	Matang
34.		Matang	Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

35.		Matang	Matang
36.		Matang	Matang
37.		Matang	Matang
38.		Matang	Matang

Tabel Lanjutan 4.1 Tabel Pengujian Model

39.		Matang	Matang
40.		Matang	Matang

Dari hasil pengujian diatas didapatkan table nilai *confusion matrix* sebagai berikut:

Tabel 4.2 Tabel *Confusion Matrix*

		Kelas Aktual		Total Prediksi
		Matang	Belum Matang	
Kelas Prediksi	Matang	19	1	20
	Belum Matang	1	19	20
Total Aktual		20	20	40

Dari hasil table *confusion matrix* diatas didapatkan perhitungan akurasi nilai *accuracy*. *Accuracy* menggambarkan seberapa akurat model dalam melakukan klasifikasi secara benar. Nilai *accuracy* pada table dibawah diperoleh berdasarkan persamaan.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \dots\dots\dots(4.1)$$

Dari persamaan diatas maka perhitungan akurasi sebagai berikut;

$$\frac{19 + 19}{19 + 1 + 1 + 19} \times 100\% = 95\%$$

Akurasi yang didapat sebesar 95% dengan rincian 18 data kematangan jambu kristal belum matang bernilai benar, 19 data kematangan jambu kristal matang bernilai benar. Data yang salah adalah 2 data kematangan jambu kristal belum matang dan 1 data kematangan jambu kristal matang. Data yang salah tersebut tidak dapat diklasifikasi karena noise dan pencahayaan yang kurang ketika pengambilan gambar sehingga warna pada jambu kristal tidak dapat terdeteksi dengan benar.

BAB V

KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil percobaan yang telah dilakukan dalam pengklasifikasian kematangan buah jambu kristal menggunakan metode *convolutional neural network* dan *computer vision* dapat diambil kesimpulan sebagai berikut;

1. Metode *convolutional neural network* menggunakan *computer vision* dapat digunakan untuk mengklasifikasi kematangan buah jambu kristal dengan baik, dengan akurasi sebesar 95% dari 40 data uji buah jambu kristal matang dan jambu kristal belum matang.
2. Penggunaan *histogram equalization* pada pengolahan data dapat meningkatkan kontras citra dan kualitas citra sehingga dalam proses *training* dapat menghasilkan model yang lebih baik.
3. Kekurangan pada penelitian ini terdapat pada *computer vision* saat uji coba secara *realtime*. Saat pengujian *camera/webcam* resolusinya tidak seperti *camera* saat pengambilan data sehingga kurang maksimal dalam pengujian secara *realtime*.

5.2. Saran

Saran untuk penelitian selanjutnya berdasarkan apa yang dialami pada penelitian ini adalah sebagai berikut;

1. Menggunakan *camera/webcam* yang memadai sehingga saat pengujian menghasilkan hasil yang maksimal.
2. Penelitian selanjutnya dapat dilakukan dengan jambu selain jenis jambu kristal.
3. Dalam pembuatan dataset, harus memperhatikan klasifikasi jambu kristal sehingga tidak terjadi kesalahan pada hasil akhir klasifikasi.

DAFTAR PUSTAKA

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2016). *Tensorflow: Large-Scale Machine Learning On Heterogeneous Distributed Systems*. [Http://Arxiv.Org/Abs/1603.04467](http://Arxiv.Org/Abs/1603.04467)
- Ahmad, R. P. (2020). *Klasifikasi Kematangan Buah Mangrove Menggunakan Metode Deep Convolutional Neural Network*.
- Amrizal, V., & Aini, Q. (2011). *Kecerdasan Buatan*.
- Andi Baso Kaswar, Risal, A. A. N., Fatiah, F., & Nurjannah, N. (2020). Klasifikasi Tingkat Kematangan Buah Markisa Menggunakan Jaringan Syaraf Tiruan Berbasis Pengolahan Citra Digital. *Jessi*, 01(May), 1–8. [Https://Ojs.Unm.Ac.Id/JESSI/Article/View/13505](https://Ojs.Unm.Ac.Id/JESSI/Article/View/13505)
- Arkadia, A., Damayanti, S. A., & Prasvita, D. S. (2021). *Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan Dengan Metode CNN*. *September*, 158–165.
- Bahtiar, H. (2016). *Sistem Pendeteksi Helm Yang Dikenakan Pengendara Sepeda Motor Untuk Safety Riding Berbasis Raspberry Pi*.
- Balitbu. (2007). *Deksripsi Jambu Biji Varietas Kristal*. *September*, 6–7.
- Bradsky, G., & Kaehler, A. (2008). Learning Opencv---Computer Vision With The Opencv Library (Bradski, G.R. Et Al.; 2008)[On The Shelf]. In *IEEE Robotics & Automation Magazine* (Vol. 16, Issue 3). [Https://Doi.Org/10.1109/Mra.2009.933612](https://Doi.Org/10.1109/Mra.2009.933612)
- Cahyo, L. D. (2016). *Pengembangan Metode Pengukuran Tingkat Kematangan Buah Jambu Kristal Menggunakan Spektroskopi Nir (Near Infrared)*.
- Chan, A., Liem, P., Wong, N. P., & Gunawan, T. (2014). Segmentasi Buah Menggunakan Metode K-Means Clustering Dan Identifikasi Kematangannya Menggunakan Metode Perbandingan Kadar Warna. *JSM (Jurnal SIFO Mikroskil)*, 15(2), 91–100.
- Deng, L., & Yu, D. (2013). Deep Learning: Methods And Applications. *Foundations And Trends In Signal Processing*, 7(3–4), 197–387. [Https://Doi.Org/10.1561/20000000039](https://Doi.Org/10.1561/20000000039)
- Fausett, L. (1994). *Fundamentals Of Neural Networks Architectures, Algorithms And Applications*.
- Hidayat, F. I., Harahap, L. A., & Panggabean, S. (2017). Identifikasi Kematangan Buah Jambu Biji Merah (Psidium Guajava) Dengan Teknik Jaringan Syaraf Tiruan Metode Backpropagation (Identification Of Guava Maturity With Artificial Neural Network Backpropagation Method). *Keteknikan Pertanian J.Rekayasa Pangan Dan Pert., Vol.5 No. 4 Th. 2017*, 5(4), 826–835.
- Hindarto, D., & Santoso, H. (2019). *Plat Nomor Kendaraan Dengan Metode Convolutional Neural Network*. *September 2021*, 1–12.
- Jalled, F., & Voronkov, I. (2016). *Object Detection Using Image Processing*. 1–6. [Http://Arxiv.Org/Abs/1611.07791](http://Arxiv.Org/Abs/1611.07791)
- Kusumanto, R., & Alan Novi, T. (2011). Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi Rgb. *Studies In*

- Environmental Science*, 2011(Semantik), 7.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/Nature14539>
- Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (2015). *Implementation Of Training Convolutional Neural Networks*. <http://arxiv.org/abs/1506.01195>
- Liu, W., Anguelov, D., Erhan, D., & Szegedy, C. (2016). SSD: Single Shot Multibox Detector. *ECCV*, 1, 21–37. <https://doi.org/10.1007/978-3-319-46448-0>
- Mulato, F. Y. (2015). *Klasifikasi Kematangan Buah Jambu Biji Merah (Psidium Guajava) Dengan Menggunakan Model Fuzzy*. 1–155.
- Munir, R. (2004). *Pengolahan Citra Digital Dengan Pendekatan Algoritmik* (1st Ed.). Informatika.
- NITA TRI DAMAYANTI. (2016). Potensi Pengembangan Tanaman Jambu Kristal (*Psidium Guajava* L) Berdasarkan Aspek Agroklimat Di Jawa Barat. *Repository IPB*, 1–10.
- Novyantika, R. D. (2018). *Deteksi Tanda Nomor Kendaraan Bermotor Pada Media Streaming Dengan Algoritma Convolutional Neural Network Menggunakan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Jurusan Statistika Disusun Oleh : Rizky Dwi Novyantika*. March.
- Pakpahan, T. E. (2011). Kajian Teknik Mencangkok Perbanyakkan Jambu Kristal. *Sekolah Tinggi Penyuluhan Pertanian Medan Sekolah Tinggi Penyuluhan Pertanian Medan*.
- Prasetyo, N. A., Surtono, A., Junaidi, J., & Pauzi, G. A. (2021). Sistem Identifikasi Tingkat Kematangan Buah Nanas Secara Non-Destruktif Berbasis Computer Vision. *Journal Of Energy, Material, And Instrumentation Technology*, 2(1), 1–10. <https://doi.org/10.23960/Jemit.V2i1.26>
- Pressman, R. S. (2010). *Software Quality Engineering: A Practitioner's Approach*. In *Software Quality Engineering: A Practitioner's Approach* (Vol. 9781118592). <https://doi.org/10.1002/9781118830208>
- Putri, A. R. (2016). Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 1(01), 1–6. <https://doi.org/10.29100/jipi.V1i01.18>
- Ratnasari, M. C. D. (2018). *Deep Learning Convolutional Neural Network Untuk Klasifikasi Pengenalan Objek Menggunakan Mxnet*. 61.
- Rawat, W., & Wang, Z. (2017). *Deep Convolutional Neural Networks For Image Classification: A Comprehensive Review Waseem*. 2733(October), 2709–2733. <https://doi.org/10.1162/NECO>
- Saepudin, E. (2019). *Budidaya Jambu Kristal*. 1–13. http://distan.jabarprov.go.id/distan/uploads/files_download/budidaya_jambu_kristal.pdf
- Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, 18(01), 15–21. <https://doi.org/10.23917/Emitor.V18i01.6235>
- Shanmugamani, R. (2018). *Deep Learning For Computer Vision: Expert Techniques To*

Train Advanced Neural Networks Using Tensorflow And Keras (S. M. Moore (Ed.)). Packt Publishing.

- Syahrudin, A. N., & Kurniawan, T. (2018). Input Dan Output Pada Bahasa. *Jurnal Dasar Pemrograman Python STMIK*, January, 1–7.
- Szeliski, R. (2010). Computer Vision. *Computer Science Handbook, Second Edition*, 43-1-43–23. <https://doi.org/10.4324/9780429042522-10>
- Utnasari, I. (2018). *Penerapan Jaringan Syaraf Tiruan Pada Pengenalan Backpropagation Intan Utnasari. 01*, 7–11.
- Wang. (2011). *Taiwan Guava Production Manual* (1st Ed., Issue November). Ministry Of Agriculture Research And Development Center, Central Farm And The Technical Mission Of The Republic Of China (Taiwan).
- Wibowo, A., Hermanto, D. M. C., Lestari, K. I., & Wijoyo, H. (2021). Deteksi Kematangan Buah Jambu Kristal Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna Hsv (Hue Saturation Value) Dan K-Nearest Neighbor. *INCODING: Journal Of Informatics And Computer Science Engineering*, 1(2), 76–88. <https://doi.org/10.34007/Incoding.V2i1.131>
- Yudha, O. (2013). Aplikasi Komputer Vision Untuk Identifikasi Kematangan Jeruk Nipis. *Skripsi*, 1–8. <http://digilib.its.ac.id/public/ITS-Undergraduate-17648-Paper-525217.pdf>
- Yuwono, B. (2015). Image Smoothing Menggunakan Mean Filtering, Median Filtering, Modus Filtering Dan Gaussian Filtering. *Telematika*, 7(1). <https://doi.org/10.31315/Telematika.V7i1.416>