

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penyusunan penelitian ini, ada beberapa penelitian terkait mengenai klasifikasi tingkat kematangan buah yang berkaitan dengan latar belakang pada penelitian ini.

Penelitian yang dilakukan oleh Ayu Damayanti dkk, [2] dengan judul “Klasifikasi Buah Mangga Badami untuk menentukan Tingkat Kematangan dengan Metode *Convolutional Neural Network (CNN)*” . terdapat sesuatu kasus dalam memastikan sebagian buah mangga yang telah melewati masa kematangan yang dimana buah mangga tersebut tidak layak lagi buat dikonsumsi. Pada riset ini dilakukan pemrograman sistem yang mendapatkan pendeteksian kematangan pada warga mangga badami dengan menerapkan metode *Convolutional Neural Network (CNN)* pada aplikasi pengolahan citra digital, sehingga dapat dipastikan buah mangga yang telah melewati masa kematangan untuk dikonsumsi. Pengujian ini dengan memakai citra sebanyak 25 citra sebagai citra uji dari 179 citra sebagai citra latih dari 204 total citra. Dengan akurasi pengujian model sebesar 97,2%.

Penelitian yang dilakukan oleh Yenansius, [1] dengan judul “ Klasifikasi Tingkat Kematangan Buah Cempedak dengan Menggunakan Metode *Convolutional Neural Network (CNN)*”. Metode *Convolutional Neural Network (CNN)* menggunakan model dasar yaitu *MobileNet* untuk bisa diimplementasikan diaplikasi android sehingga dengan memotret atau mengambil data dari galeri cempedak bisa

diketahui klasifikasi buah cempedak berdasarkan nilai akurasi. Penelitian melakukan *cross validation* akurasi terendah dengan *epoch* 10 adalah 61,87% dan yang tertinggi dengan *epoch* 50 dengan akurasi rata-rata 67,47%, dan untuk pengujian didapatkan akurasi 63,33% dengan presisi 62,6%.

Penelitian yang dilakukan oleh Aldho Helsaputra dkk, [5] dengan judul “Klasifikasi Tingkat Kematangan Buah Pepaya menggunakan metode *Convolutional Neural Network* (CNN)”. Dengan keakurasian 92% yang dimana terdapat 4 data citra uji yang tidak sesuai dengan kategori yang dikarenakan intensitas cahaya citra yang berbeda dengan data latih. Dari hasil penelitian ini memiliki persentasi keberhasilan sebesar 93,6% berhasil terdeteksi dan 6,4% berhasil terdeteksi.

Penelitian yang dilakukan oleh Budi Yanto, [4] dengan judul “Klasifikasi Tingkat Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan metode *Deep Learning Convolutional Neural Network* (CNN)”. Pada umumnya proses klasifikasi untuk menentukan jeruk yang layak dan tidak berkualitas masih menggunakan cara manual yaitu pengamatan langsung secara visual terhadap buah. Salah satu cara untuk mengklasifikasikan jeruk manis melalui sistem computer memakai algoritma *Deep Learning Convolutional Neural Network* (CNN). Pada penelitian ini dengan 100 *dataset* gambar jeruk manis menunjukkan tingkat akurasi klasifikasi maka hasilnya sebesar 97,5184%, dilakukan klasifikasi maka hasilnya sebesar 67,8221%. Pengujian sebesar 10 citra buah jeruk manis yang terbagi menjadi 5 citra jeruk bagus dan 5 citra jeruk busuk sebesar 96% untuk

training 92% untuk *testing* 68%.

Penelitian yang dilakukan oleh Indrabayu dkk, [3] dengan judul “ Klasifikasi Tingkat Kematangan stroberi berbasis Segmentasi Wana dengan Metode *Convolutional Neural Network* (CNN)”. Klasifikasi kematangan buah secara manual memiliki banyak keterbatasan karena dipengaruhi faktor subjektivitas manusia sehingga pengaplikasian pengolahan citra digital dan kecerdasan buatan menjadi lebih efektif dan efisien. penelitian ini bertujuan untuk membuat sistem klasifikasi kematangan buah stroberi secara otomatis yang dibagi menjadi 3 kategori yaitu belum matang, setengah matang dan matang. *Dataset* yang digunakan terdiri dari 158 gambar stroberi. Hasil penelitian menunjukkan bahwa klasifikasi kematangan stroberi menggunakan algoritma CNN adalah 97%.

2.2 Landasan Teori

2.2.1 Tanaman Tamarillo

Tamarillo atau terong belanda merupakan tanaman yang tergolong dalam keluarga *Solanaceae*. Terong belanda merupakan salah satu buah yang rasanya asam dan manis namun ketika diolah menjadi minuman segar seperti sirup dan jus rasanya akan semakin nikmat. Sepanjang tahun petani bisa memanen buah. Satu pohon dalam setahun bisa menghasilkan buah kira-kira 70 kg buah. Buah tamarillo berasal dari Andes Peru, Ekuador, Chili dan Bolivia [1]. Dan saat ini diyakini bahwa buah tamarillo sudah menghilang dari habitat aslinya, namun buah tamarillo sudah dikembangkan dan dibudidayakan di Argentina, Kolombi,

Venezuela, dan Brasil. Alasan tamarillo disebut terong belanda karena buah ini pertama kali dibawa dan dibudidayakan oleh tentara Belanda pada tahun 1941 di Indonesia. Buah ini pertama kali dibudidayakan di Indonesia oleh tentara Belanda di daerah Jawa Barat dan Bogor.

2.2.2 Klasifikasi Tanaman Tamarillo

Adapun klasifikasi tanaman tamarillo adalah sebagai berikut:

Kingdom	: <i>Plantae</i>
Subkingdom	: <i>Spermatophyta</i>
Kelas	: <i>Dicotyledonae</i>
Ordo	: <i>Sonales</i>
Subdivisi	: <i>Agiospermae</i>
Subkelas	: <i>Asteridae</i>
Genus	: <i>Solanum</i>
Spesies	: <i>Solanum betaceum cav</i>
Famili	: <i>Solanaceae</i>

2.2.3 Manfaat Tamarillo

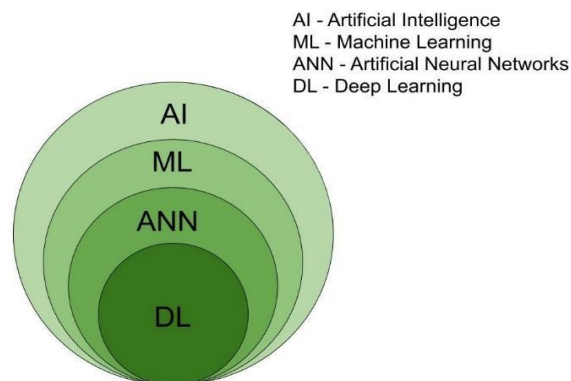
Tanaman tamarillo diketahui memiliki banyak manfaat antara lain :

1. Membantu mencegah obesitas
2. Sebagai anti oksidan
3. Meningkatkan daya tahan tubuh
4. Mengurangi resiko penyakit jantung

5. Merawat kesehatan kulit
6. Mencegah kanker
7. Meningkatkan penglihatan mencegah masalah pencernaan dan mengurangi sembelit karena banyak mengandung serat.

2.2.4 Deep Learning

Menurut Shafira dalam [7] *Deep Learning* adalah cabang dari *machine learning* yang menggunakan konteks manusia dengan menerapkan saraf buatan yang memiliki banyak *hidden layer*. *Deep learning* juga merupakan kunci pada kendaraan yang memiliki sistem kemudi secara otomatis, seperti pengenalan suara, gambar, dan deteksi ketidak normal/kelainan, serta pemrosesan bahasa lain tanpa memperkenalkan aturan kode atau pengetahuan dominan manusia. Pada *deep learning* tersedia solusi menyeluruh tanpa adanya pembagian pada proses fitur ekstraksi dan proses pengenalan menggunakan metode pembelajaran mesin. Gambar 2.1 merupakan diagram *venn deep learning*.



Gambar 2. 1 Venn Deep Learning [7]

2.2.5 Convolutional Neural Network (CNN)

Jaringan syaraf tiruan adalah algoritma yang sering digunakan pada konsep

deep learning karena dekat dengan konsep hirarki pada *deep learning* salah satu algoritma jaringan syaraf tiruan yaitu *Convolutional Neural Network* merupakan pengembangan dari *Multilayer Perceptron* (MLP) [1]. CNN juga merupakan algoritma yang memiliki layer convolutional yang berfungsi untuk ekstraksi data.

A. Konsep *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) digolongkan kedalam *Deep Neural Network* karena banyak diaplikasikan pada citra dan memiliki kedalaman jaringan yang tinggi. Dalam klasifikasi citra pada awalnya dapat digunakan dengan MLP, akan tetapi penggunaan metode MLP kurang tepat karena menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik dan tidak menyimpan informasi spasial dari data citra. CNN pertama kali dilakukan oleh Hubel dan Wiesel (Hubel dan Wiesel, Tahun 1968) [7] tentang *visual cortex* pada indra penglihatan kucing. *Convolutional Neural Network* (CNN) merupakan struktur yang terdiri dari beberapa tahap dan dapat dilatih [1]. Tahapan *Convolutional Neural Network* (CNN) antara lain: Masukan (*input*), dan keluaran (*output*) setiap tahap terdiri dari beberapa *array* yang sering disebut dengan *feature map*. Setiap tahapan terdiri atas tiga layer yaitu konvolusi, fungsi aktivasi *layer* dan *pooling layer*.

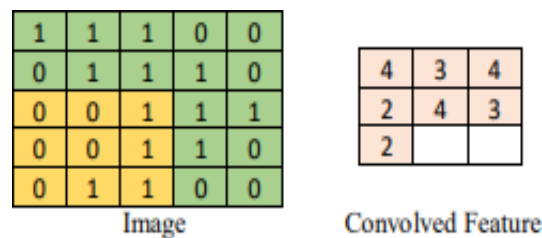
B. Struktur Jaringan *Convolutional Neural Network* (CNN)

Struktur *Convolutional Neural Network* (CNN) terdiri dari beberapa lapisan tersembunyi (*hidden layer*) yaitu berupa *Convolutional Layer*, *pooling Layer* dan

Fully Connected Layer [8]. Metode *Convolutional Neural Network* (CNN) bekerja secara hirarki sehingga output pada *Convolutional Layer* pertama akan digunakan sebagai input pada proses *Convolutional Layer*. Sedangkan pada proses klasifikasi terdiri dari *fully connected* dan *polling layer* yang *outputnya* berupa hasil klasifikasi.

1. *Convolutional Layer*

Dalam matematis konvolusi berarti mengaplikasikan output fungsi lain secara berulang. Pada pengolahan citra, konvolusi yaitu pengaplikasian sebuah kernel (kotak kuning) pada citra disemua offset yang memungkinkan. Cara kerja lapisan konvolusi yaitu bekerja dengan meniru sifat visual otak dan mempelajari filter gambar yang diinput [2]. Konvolusi berfungsi untuk mengekstraksi fitur gambar masukan, kemudian akan menghasilkan transformasi *linear* dari data masukan sesuai informasi spasial data.

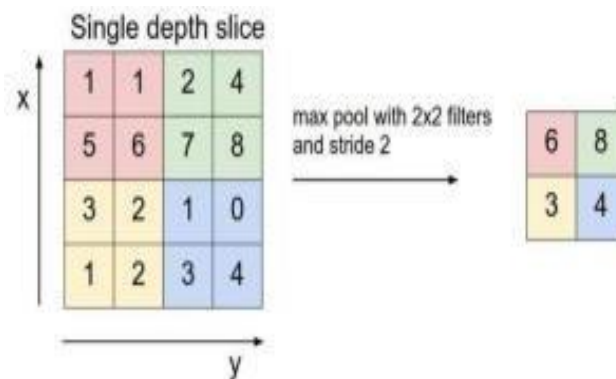


Gambar 2. 2 Operasi Konvolusi Layer[2]

Pada gambar 2.2 keseluruhan kotak hijau merupakan citra yang akan dikonvolusi. Kernel gambar 2.0 bergerak dari sudut kiri atas ke kanan bawah.

2. Pooling Layer

Pooling adalah mengurangi ukuran matriks menggunakan operasi *Pooling*. *Pooling layer* terdiri dari filter dengan *stride* dan ukuran tertentu yang akan bergantian bergeser pada kegiatan *map* [4]. Pada metode *Convolutional Neural Network* (CNN) menggunakan *max pooling*, dimana *max pooling* membagi output konvolusional menjadi beberapa kisi kecil, kemudian nilai maksimum dari setiap kisi yang diambil untuk menyusun gambar yang dikurangi. Operasi *max pooling* dapat dilihat pada gambar 2.3.



Gambar 2. 3 Pooling Layer [4]

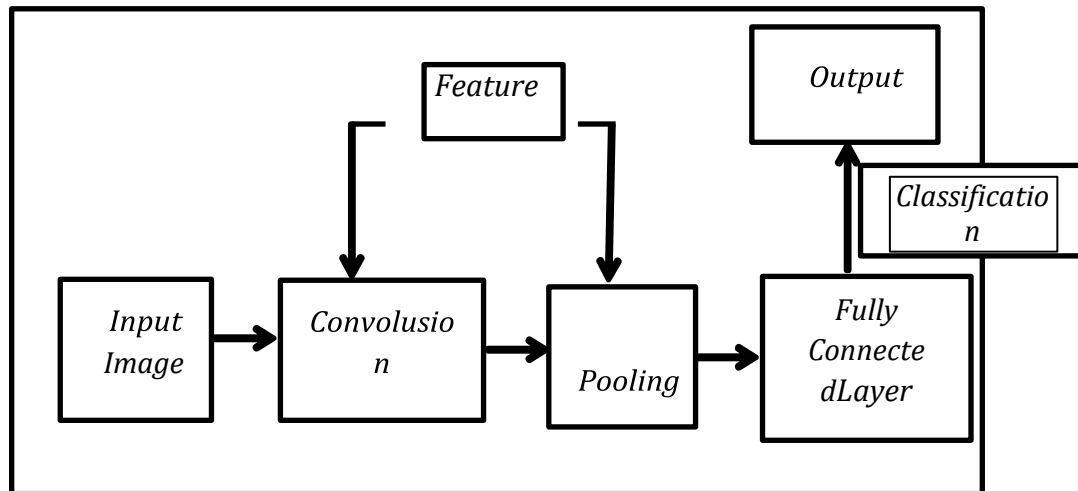
3. Fully Connected Layer

Fully Connected merupakan layer yang biasanya dipakai dalam implementasi MLP yang digunakan untuk melakukan transformasi pada dimensi data agar data dapat dikelompokkan secara *linear*. Setiap *neural* pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected* layer [3]. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak

reversibel, *fully connected layer* hanya dapat diimplementasikan di akhir jaringan.

C. Langkah-Langkah *Convolutional Neural Network* (CNN)

Langkah-langkah proses CNN yaitu:



Gambar 2. 4 Langkah-Langkah *Convolutional Neural Network*

- a) Langkah yang pertama dilakukan ialah memecah gambar menjadi gambar yang tumpang tindih yang lebih kecil [2]. Kemudian hasil konvolusi dari setiap gambar yang kecil dijadikan input untuk mendapatkan sebuah hasil representasi fitur untuk memberikan kemampuan pada *Convolutional Neural Network* (CNN) dalam mengenali sebuah objek, oleh karena itu dimanapun posisi objek tersebut muncul pada suatu gambar disebut dengan *translation invariance*.
- b) Setiap gambar yang lebih kecil dimasukkan kedalam *small neural network*. Semua bagian gambar yang telah dikonvolusi dilakukan pada proses ini, dengan menggunakan filter yang sama [2]. Dalam proses ini setiap bagian

gambar akan memiliki faktor penggali yang sama, atau *neural network* yang disebut sebagai *weights sharing*. Jika ada hal yang menarik pada setiap gambar, maka akan diberi tanda sebagai *object of interest*. Hasil dari proses *weights sharing* merupakan sebuah *activation map*. Untuk menghasilkan representasi gambar yang lebih baik, maka proses konvolusi diulang menggunakan filter yang berbeda untuk mendapatkan *activation map* lain

- c) Proses *feature learning* terdiri dari 3 bagian yaitu, *Convolution* (konvolusi), *Rule* (fungsi aktivitas *layer*) dan *pooling layer*. Proses ini dilakukan berulang kali sampai menghasilkan peta fitur yang cukup untuk dilanjutkan ketahap *Classification* (klasifikasi) [2]. *Convolutional layer* tersusun atas *neural* yang sedemikian rupa sehingga membentuk sebuah filter dengan tinggi (pixel) dan panjang. Operasi “*dot*” dilakukan pada setiap pergeseran antara input dan nilai dari filter sehingga menghasilkan output atau yang sering disebut sebagai *feature map* atau *activation map*. *Feature map* dapat dihitung dengan rumus sebagai berikut.

$$Output = \frac{W-F+2P}{S+1} \quad (2,1)$$

Keterangan:

F = Ukuran Filter

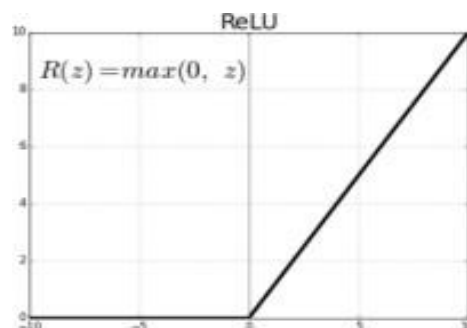
S = *Stride* (ukuran pergeseran)

P = Jumlah *Padding* yang di gunakan

W = Ukuran volume gambar

Dari persamaan rumus tersebut, maka ukuran spasial dari volume *output* dapat dihitung dimana *hyperparameter* yang digunakan ialah ukuran *filter* (F), *Stride* yang diterapkan (S), jumlah *padding* nol yang digunakan (P) dan volume (V) [4]. *Stride* yaitu nilai yang berfungsi untuk menggeser filter melalui *input* citra dan *Zero padding* yaitu nilai untuk mendapatkan angka nol disekitar border citra.

Fungsi *Rule* (*Rectified linear unit*) merupakan nilai keluaran/*output* dari *neural* dapat dinyatakan sebagai nol apabila inputnya adalah *negative*. Sedangkan *pooling layer* biasanya terletak setelah *convolutional layer* dimana prinsip *pooling layer* tersusun dari sebuah filter dengan ukuran dan *stride* tertentu yang bergeser pada seluruh area *feature map* yang berfungsi untuk mengurangi dimensi dari *feature map* (*downsampling*), sehingga mempercepat komputasi karena parameter yang harus di *update* semakin sedikit dan mengatasi *overfitting*. *Pooling* dibagi atas 2 bagian yaitu *Average Pooling* (nilai rata-rata) dan *Max Pooling* (nilai tertinggi). Gambar 2.4 menampilkan fungsi aktivitas *Rule*.



Gambar 2. 5 Fungsi Aktivitas *Rule* [5]

d) Hasil dari *feature learning* ialah *feature map* yang masih berbentuk multidimensional *array*, sehingga harus melakukan “*flatten*” atau “*reshape feature map*” menjadi sebuah vector sebagai bisa digunakan sebagai input dari *fully-connected* layer [5]. Lapisan *fully-connected* merupakan suatu lapisan dimana seluruh *neural* aktivitas dari lapisan sebelumnya semua terhubung dengan *neural* dilapisan selanjutnya seperti pada jaringan syaraf tiruan yang sering digunakan pada metode *multi lapisan perceptron* yang berfungsi untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan lapisan konvolusi dan lapisan *fully-connected* ialah pada lapisan konvolusi *neural* dilapisan konvolusi hanya terhubung ke daerah tertentu pada input. Sedangkan *fully-connected* mempunyai *neural* secara keseluruhan terhubung. Tetapi, kedua lapisan tersebut masih mengoperasikan produk *dot*, sehingga fungsinya hampir sama.

2.2.6 Arsitektur MobileNetv3

MobileNetv3 adalah model yang berbasis *Convolutional Neural Network* (CNN) paling populer untuk aplikasi smartphone dan perangkat yang tertanam. Teknologi ini menggabungkan jaringan seluler ke CPU sehingga di dalam memilih konfigurasi secara optimal dan mudah untuk menemukan atau mengoptimalkan arsitektur jaringan [5]. *MobileNetv3* memiliki struktur yang lebih sederhana daripada jaringan saraf biasa. Bagian konvolusional terdiri dari satu lapisan sederhana *MobileNetv3* memberikan banyak peningkatan kinerja. Model *MobileNetv3* Residual terbaik dan struktur *Bottleneck Linier* diperkenalkan atas

dasar model *MobileNetv3*, yang tidak hanya meningkatkan kecepatan model *Neural Network*, tetapi juga mengurangi kompleksitas jaringan syaraf.

2.2.7 Python

Python adalah bahasa pemrograman yang sifatnya *open source*. bahasa pemrograman ini dioptimalisasikan untuk *companion, integration developer productivity software quality* dan *program portability* [2]. Python sudah banyak digunakan dalam pengembangan berbagai perangkat lunak, seperti *user interface, numeric programming, product customization, system programming* dan *internet scripting*.

2.2.8 Open source computer vision library (open CV)

Open source computer vision library merupakan sebuah *library* yang berfungsi untuk mengolah video dan gambar sampai mampu mengestrak, informasi yang didalam [1]. *OpenCV* memiliki fitur seperti: deteksi wajah, pengenalan wajah, *kalman filtering*, pelacakan wajah, dan masih banyak metode AI (*Artificial Intellegence*) lainnya. *OpenCV* dapat berjalan diberbagai bahasa pemrograman diantaranya: *Matlab, Java, C,C++,Python* serta mendukung banyak *platform* seperti *Windows, Mac OS, Ios, Android* dan *Linux*.

Cara kerja *OpenCV* ialah mencoba meniru cara kerja sistem visual manusia atau penglihatan, dimana objek didapat melalui penglihatan mata kemudian citra yang ditangkap oleh mata akan diteruskan ke otak [4]. Citra yang didapat oleh otak akan di proses atau di interpretasi sehingga dapat dimengerti objek yang dilihat. Setelah itu hasil dari proses penglihatan bisa

digunakan untuk mengambil keputusan atau tindakan selanjutnya. Sehingga komputer *vision* ini termasuk dalam artificial intelligence atau kecerdasan buatan.

Keunggulan dari *OpenCV* :

- a) Manipulasi data citra
- b) Kalibrasi kamera
- c) Pelabelan citra
- d) Pemrosesan citra fundamental
- e) Analisis struktur
- f) Citra dan video *I/O*
- g) Pengenalan objek
- h) Data struktur dinamis
- i) Analisis gerakan
- j) *Graphical User Interface*

2.2.9 Confusion Matrix

Penelitian ini menggunakan *Confusion Matrix* yang disebut *error matrix* dalam pengujian tingkat akurasi. *Confusion matrix* adalah sebuah tabel yang menyatakan jumlah data uji benar dan salah di klasifikasi. Dalam *confusion matrix* terdapat 4 istilah yang digunakan sebagai representasi hasil proses klasifikasi antara lain:

1. *True negatif* (TN) yaitu data negatif yang diprediksi benar.

2. *True positif* (TP) yaitu data positif yang diprediksi benar.
3. *False negatif* (FN) yaitu positif namun diprediksi sebagai data negatif, yang kemungkinan seluruh kejadian sebenarnya positif (P) dan kemungkinan seluruh kejadian sebenarnya negatif (N).
4. *False positif* (FP) yaitu data negatif tetapi diprediksi sebagai data positif.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Gambar 2. 6 *Convulsion Matriks*

Nilai yang ada pada gambar 2.5 diatas bisa digunakan untuk menghitung akurasi dengan persamaan.

$$\text{Presisi prediksi} = \frac{TP}{TP+FP} \quad (2,2)$$

Presisi merupakan penggambaran seberapa tepat model dalam memprediksi kajadian positif saat kegiatan prediksi. Kemudian untuk menghitung akurasi menggunakan persamaan

$$\text{Akurasi} = \frac{TP+TN}{P+N} \quad (2,3)$$

Akurasi berfungsi sebagai parameter yang akurat pada suatu model dalam melakukan klasifikasi. *Recall* dapat dihitung menggunakan persamaan.

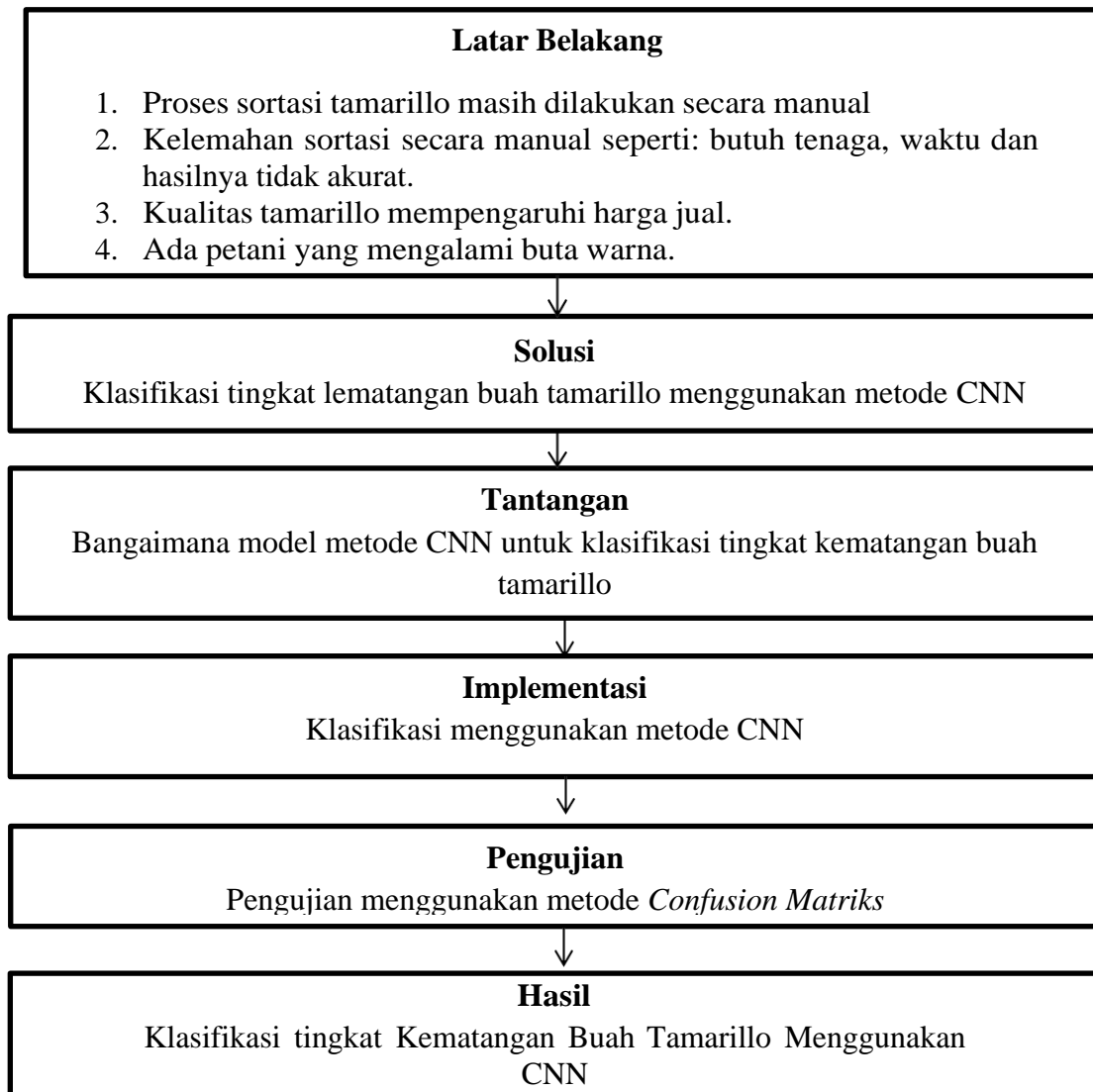
$$Recall = \frac{TP}{TP+FN} \quad (2,4)$$

F1 Score merupakan rata-rata dari *precision* dan *recall* yang dapat dihitung dengan persamaan.

$$F1 = 2x \frac{precision \times Recall}{precision+Recall} \quad (2,5)$$

2.3 Kerangka Pikir

Kerangka pikir dalam penelitian ini mengenai klasifikasi tingkat kematangan buah tamarillo menggunakan metode *Convolutional Neural Network* (CNN) dapat dilihat pada Gambar 2.6



Gambar 2. 7 Kerangka Pikir