

SISTEM PENDETEKSI BUAH LADA BERBASIS *CONVOLUTIONAL NEURAL NETWORK (CNN)*

PROYEK AKHIR

Laporan proyek akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Abdur Rohim

NPM :

1051701

Reiva Marizka Harmie

NPM :

1051723

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2021**

LEMBAR PENGESAHAN

SISTEM PENDETEKSI BUAH LADA BERBASIS CONVOLUTIONAL NEURAL NETWORK (CNN)


Oleh:

Abdur Rohim	NPM	1051701
Reiva Marizka Harmie	NPM	1051723


Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma IV Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,


Pembimbing 1


Muhammad Iqbal Nugraha, M.Eng


Pembimbing 2


Indra Dwisaputra, M.T

Penguji 1


Charlotha, M.Tr.T

Penguji 2


Aan Febriansyah, M.T

Penguji 3


Yang Agita Rindri, M.Eng

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1: Abdur Rohim

NPM : 1051701

Nama Mahasiswa 2: Reiva Marizka Harmie

NPM : 1051723

Dengan Judul : Sistem Pendeteksi Buah Lada Berbasis *Convolutional Neural Network* (CNN)

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 09 Maret 2021

Nama Mahasiswa

1. Abdur Rohim
2. Reiva Marizka Harmie

Tanda Tangan


.....

.....

ABSTRAK

Pohon lada tumbuh merambat hingga ketinggian 4 meter dengan ditopang oleh pohon lain atau memakai tiang disebut dengan junjung. Permasalahan yang ada saat ini adalah saat proses pemanenan lada, para petani umumnya memetik dengan tangan. Pada tahun 2020 mahasiswa D3 membuat alat rancang bangun pemetik lada dengan menggunakan remote untuk mengontrol pergerakannya, alat ini belum bisa bergerak secara otomatis, dari persoalan tersebut dibuatlah sistem pendeteksi buah lada berbasis convolutional neural network (CNN). Dengan membuat sistem pendeteksi ini, maka robot diharapkan bisa mendeteksi objek buah lada sehingga kedepannya dapat digunakan untuk memetik buah lada secara otomatis. Sistem ini menggunakan metode Convolutional Neural Network (CNN). Implementasi CNN menggunakan tools Tensorflow dengan bahasa pemrograman Python. Jumlah dataset sebanyak 100 gambar buah lada dan 100 gambar non lada. Berdasarkan hasil pengujian tingkat precision prediksi sebesar 95 %, accuracy 89 %, dan recall 90 %, Jadi dapat disimpulkan bahwa sistem pendeteksi dengan menggunakan webcam dapat memprediksi buah lada dengan baik.

Kata kunci : Buah lada, Sistem pendeteksi, Buah lada

ABSTRACT

Pepper trees grow vines to a height of 4 meters supported by other trees or using poles called junjung. The current problem is that during the pepper harvesting process, farmers generally pick by hand. In 2020 D3 students made a pepper picker design tool using a remote to control its movement, this tool cannot move automatically, from this problem a convolutional neural network (CNN) based pepper fruit detection system was created. By making this detection system, the robot is expected to be able to detect objects of pepper fruit so that in the future it can be used to pick peppers automatically. This system uses the Convolutional Neural Network (CNN) method. CNN implementation uses the Tensorflow tools with the Python programming language. The number of datasets is 100 pictures of pepper fruit and 100 pictures of non pepper. Based on the test results, the level of prediction precision is 95%, accuracy 89%, and recall of 90%, so it can be concluded that the detection system using a webcam can predict pepper fruit well.

Keywords: *pepper fruit, detection system, pepper fruit.*

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat Allah SWT atas karunianya, sehingga penulis dapat menyelesaikan laporan tugas akhir ini yang berjudul "Sistem Pendeteksi Buah Lada Berbasis *Convolutional Neural Network* (CNN)". Serta shalawat dan salam kepada Rasulullah Muhammad S.A.W, yang telah membawa umat manusia ke dunia yang damai, terang dan penuh ilmu pengetahuan. Tujuan penulis membuat laporan tugas akhir ini adalah untuk memenuhi salah satu syarat kelulusan pendidikan Diploma IV di Politeknik Manufaktur Negeri Bangka Belitung. Dalam laporan tugas akhir, penulis membahas tentang hasil penelitian yang penulis laksanakan selama program proyek akhir berlangsung. Adanya sistem pendeteksi buah lada yang berbasis *Convolutional Neural Network* (CNN) ini diharapkan dapat mempermudah dan meringankan para petani dalam proses pengambilan buah lada.

Pada kesempatan ini, ucapan terima kasih juga disampaikan kepada pihak yang telah banyak membantu serta ikut memberi motivasi, sumbang saran, kritik yang tentunya sangat diharapkan dalam penyelesaian tugas akhir ini. Berikut ini adalah pihak-pihak yang ikut membantu baik secara langsung maupun tidak langsung, diantaranya:

1. Orang tua, keluarga, dan teman-teman yang telah banyak memberikan do'a dan dukungan.
2. Bapak Muhammad Iqbal Nugraha, M.Eng selaku dosen pembimbing 1 proyek akhir di Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Indra Dwisaputra, M.T selaku dosen pembimbing 2 proyek akhir di Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
5. Seluruh dosen dan instruktur yang telah banyak membantu dalam penyelesaian proyek akhir ini.
6. Seluruh rekan-rekan mahasiswa D4 Polman Babel.

7. Seluruh pihak-pihak yang tidak dapat penulis sebutkan satu per satu.

Penulis sangat berharap makalah ini bermanfaat dalam rangka menambah pengetahuan dan juga wawasan yang terdapat dalam tugas akhir ini. Penulis pun menyadari bahwa dalam laporan tugas akhir ini masih terdapat banyak kekurangan dan jauh dari kata sempurna. Oleh sebab itu, penulis mengharapkan banyak kritik dan saran demi perbaikan laporan penulis di masa yang akan datang, mengingat tidak ada kata sempurna tanpa saran membangun. Mudah – mudahan laporan tugas akhir sederhana ini dapat di pahami dan bermanfaat untuk semua orang khususnya bagi para pembaca makalah ini.

Sungailiat, 09 Maret 2021

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
PERNYATAAN BUKAN PLAGIAT	ii
ABSTRAK	iii
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah dan Batasan Masalah	2
1.2.1 Rumusan Masalah.....	2
1.2.2 Batasan Masalah	2
1.2.3 Tujuan Proyek Akhir.....	2
BAB II.....	3
DASAR TEORI	3
2.1 Jaringan Saraf Tiruan	3
2.2 Pengolahan Citra Digital	4
2.3 Arsitektur <i>Perceptron</i>	4
2.4 <i>Convolutional Neuran Network (CNN)</i>	5
2.4.1 Proses konvolusi	6
2.4.2 <i>Pooling Layer</i>	7
2.4.3 <i>Fully connected Layer</i>	8
2.5 <i>Computer Vision</i>	8
2.6 <i>Tensorflow</i>	9
2.7 Raspberry Pi	10
2.8 Webcam.....	11
BAB III	13
METODE PELAKSANAAN	13

3.1	Konsultasi dan <i>Study</i> Pustaka	13
3.2	Rancangan sistem	14
3.3	Sistem Kerja Alat	15
3.4	Arsitektur Jaringan Saraf Tiruan	16
BAB IV		17
PEMBAHASAN		17
4.1	Dataset	17
4.2	Pemrosesan Citra	19
4.3	Pemodelan Jaringan Syaraf Tiruan.....	19
4.4	Hasil.....	21
BAB V PENUTUP.....		26
5.1	Kesimpulan.....	26
5.2	Saran	26
DAFTAR PUSTAKA		27

DAFTAR GAMBAR

Gambar 1. 1 Pemetikan buah Lada	1
Gambar 2. 1 Arsitektur <i>perceptron</i> sederhana	5
Gambar 2. 2 Model CNN	6
Gambar 2. 3 Proses Konvolusi	6
Gambar 2. 4 Posisi Karnael dalam Konvolusi	6
Gambar 2. 5 <i>Fully Connected Layer</i>	8
Gambar 2. 6 Raspberry pi 3 B+	10
Gambar 3. 1 Survei ke perkebunan buah lada.....	13
Gambar 3. 2 Rancangan sistem.....	14
Gambar 3. 3 <i>Flowchart</i> Sistem Kerja Alat.....	15
Gambar 3. 4 Arsitektur CNN	16
Gambar 4. 1 Buah Lada	17
Gambar 4. 2 Sebelum di <i>resize</i> (b) Setelah di <i>resize</i>	19
Gambar 4. 3 Grafik <i>Accuracy</i> Model CNN	20
Gambar 4. 4 Grafik <i>Loss</i> Model CNN	20

DAFTAR TABEL

Tabel 2. 1 Tabel Spesifikasi Raspberry Pi 3 Model B+	10
Tabel 4. 1 Hasil Prediksi Model CNN	21
Tabel 4. 2 Hasil Nilai Prediksi	23
Tabel 4. 3 Hasil Prediksi dengan Jarak	24

DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup

Lampiran 2: Program Sistem Pendeteksi Buah Lada Berbasis Convolutional Neural Network (CNN)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Lada (*Piper nigrum*) merupakan salah satu rempah – rempah yang banyak digunakan dalam masakan di dunia. Sebagai daerah produsen utama lada putih di Indonesia, bagi Provinsi Kepulauan Bangka Belitung, lada juga merupakan produk unggulan ekspor, khususnya ditunjukkan melalui popularitas *branding* Muntok-*white paper* yang telah memiliki sertifikat indikasi geografis di pasar dunia. Usaha budi daya lada memiliki prospek dan peluang usaha yang tinggi dapat diarahkan sebagai unit bisnis yang mampu meningkatkan pendapatan bagi petani (Dinas Pertanian, Peternakan dan Perkebunan Provinsi Bangka Belitung, 2015) [1]. Ketinggian pohon lada dapat tumbuh sekitar 4 meter yang ditopang dengan pohon lain atau tiang disebut dengan junjung. Permasalahan yang ada saat ini adalah saat proses pemanenan lada, para petani umumnya memetik dengan tangan. Dalam satu musim panen, petani lada membutuhkan tenaga tambahan untuk memanen lada-lada tersebut dari kebunnya. Pekerja menggunakan tangga untuk menjangkau buah lada yang lebih tinggi sehingga semakin menambah kerumitan dalam pemanenan lada karena pekerja harus turun-naik tangga untuk memetik lada dari pohon satu ke pohon lainnya [1].



Gambar 1. 1 Pemetikan buah Lada [2]

Pada tahun 2020 mahasiswa D3 membuat alat rancang bangun pemetik lada dengan *remote* untuk pengontrolan pergerakan robot, dimana robot ini belum bisa bergerak secara otomatis dan belum bisa mendeteksi buah lada secara otomatis, jadi melalui proyek akhir ini kami mengusulkan untuk mengembangkan robot

pemetik buah lada khususnya pada bagian sistem pendeteksi buah lada dengan judul proyek akhir yaitu Sistem Pendeteksi Buah Lada Berbasis *Convolutional Neural Network* (CNN) . Dengan membuat sistem pendeteksi ini, maka robot diharapkan bisa mendeteksi objek buah lada sehingga suatu saat dapat digunakan untuk memetik buah lada secara otomatis.

1.2 Rumusan Masalah dan Batasan Masalah

1.2.1 Rumusan Masalah

Adapun rumusan masalah dalam sistem pendeteksi buah lada ini adalah sebagai berikut:

1. Bagaimana cara mengenali objek buah lada dengan tepat.
2. Bagaimana membuat suatu sistem pengenalan objek berbasis *convolutional neural network* (CNN).

1.2.2 Batasan Masalah

Adapun batasan masalah dalam sistem pendeteksi buah lada ini adalah sebagai berikut:

1. Sistem ini hanya mendeteksi buah lada tidak sampai pada pergerakan robot.
2. Sistem ini hanya mendeteksi ada atau tidaknya buah lada belum sampai membedakan tingkat kematangan buah lada.
3. Buah lada yang digunakan untuk pengujian hanya buah lada pada tangkai, tidak dalam bentuk butiran.

1.2.3 Tujuan Proyek Akhir

Adapun tujuan dari proyek akhir ini adalah:

1. Membuat sistem pengenalan objek buah lada berbasis *Convolutional Neural Network* (CNN).
2. Menggunakan kamera dalam pengambilan gambar objek buah lada.
3. Menampilkan hasil pendeteksian ditampilkan di layar monitor.
4. Menerapkan kecerdasan buatan dengan menggunakan Raspberry.

BAB II

DASAR TEORI

2.1 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (JST) merupakan sistem komputasi yang didasarkan pada pemodelan saraf biologis (*neuron*) atau saraf manusia. JST bisa dibayangkan berupa jaringan dengan elemen pengolah sederhana yang saling terhubung. Elemen pengolah berinteraksi melalui sambungan yang variabel, disebut bobot, dan bila diatur secara tepat dapat menghasilkan sifat yang diinginkan [3].

Pada jaringan saraf tiruan, lapisan *neuron* merupakan lapisan-lapisan *neuron* yang dihubungkan dengan lapisan sebelum dan sesudahnya. Lapisan tersembunyi (*hidden layer*) merupakan data yang diberikan pada lapisan masukan akan dirambatkan ke lapisan lainnya. Rambatan informasi bergantung pada algoritma pembelajarannya. Nilai bias digunakan untuk mengakomodasi informasi yang ada melalui data-data *input* maupun melalui bobot-bobotnya, maka biasanya pada JST ditambahkan bias yang senantiasa bernilai 1, berikut beberapa aplikasi jaringan saraf tiruan:

1. Pengenalan Pola (*Pattern Recognition*)

Jaringan syaraf tiruan juga sering digunakan untuk mengenali pola seperti huruf, angka, suara atau tanda tangan yang mengalami sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpai (mungkin wajah/bentuk tubuhnya sudah sedikit berubah).

2. *Signal Processing*

Jaringan syaraf tiruan (model *Backpropagation*) dapat dipakai untuk menekan *noise* dalam saluran telpon.

3. Peramalan

Jaringan syaraf tiruan juga dapat dipakai untuk meramalkan apa yang akan terjadi dimasa yang akan datang berdasarkan pola kejadian yang ada dimasa lampau. Ini dapat dilakukan mengingat kemampuan jaringan syaraf tiruan untuk mengingat dan membuat generalisasi dari apa yang sudah ada

sebelumnya [4].

2.2 Pengolahan Citra Digital

Pengolahan citra digital adalah manipulasi dan interpretasi digital dari citra dengan bantuan komputer. *Input* dari pengolahan citra adalah citra, sedangkan *output*-nya adalah citra hasil pengolahan[5]. pengolahan citra digital sering disebut sebagai pemrosesan citra dua dimensi . Pengolahan citra digital juga mencakup semua data dua dimensi.

Operasi-operasi pada pengolahan citra diterapkan bila:

1. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra (*image enhancement*). Contoh: warna semu, tepi objek, gelap/terang.
2. Adanya cacat pada citra sehingga perlu dihilangkan/diminimumkan (*image restoration*). Contoh: penghilangan *noise* dan *debluring*.
3. Elemen dalam citra perlu dikelompokkan, dicocokkan atau diukur (*image segmentation*).
4. Diperlukannya ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasian objek (*image analysis*). Contoh: pendeteksian tepi objek [5].

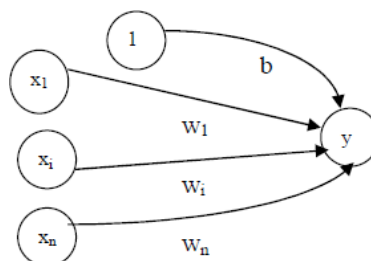
2.3 Arsitektur Perceptron

Perceptron adalah salah satu bentuk jaringan syaraf tiruan yang sederhana. Metode *Perceptron* merupakan metode pembelajaran dengan pengawasan dalam sistem jaringan syaraf. Hal yang perlu diperhatikan dalam merancang jaringan syaraf tiruan adalah jumlah spesifikai yang akan diidentifikasi. Jaringan *neuron* terdiri dari sejumlah *neuron* dan sejumlah masukan [6].

“*Perceptron* adalah bentuk paling sederhana dari JST yang digunakan untuk mengkasifikasikan pola khusus yang biasa disebut *linearly separable*, yaitu pola-pola yang terletak pada sisi yang berlawanan pada suatu bidang. Pada dasarnya *Perceptron* terdiri dari *neuron* tunggal dengan bobot-bobot sinaptik dan *threshold*

yang dapat diatur, *Perceptron* terbatas hanya untuk mengklasifikasikan dua kelas saja” [6].

Output dari unit *assosiator* adalah biner vektor. Vektor digunakan sebagai sinyal *input* terhadap sinyal *output* atau *unit response*. Tujuan dari jaringan ini adalah mengklasifikasikan setiap pola *input* ke dalam kelas tertentu [7]. Apabila *output*-nya +1, maka *input* yang diberikan termasuk kelas tertentu, sebaliknya jika *output*-nya -1, maka *input* yang diberikan tidak masuk dalam kelas tertentu, berikut bentuk *Arsitektur perceptron* :



Gambar 2. 1 *Arsitektur perceptron* sederhana [7]

Keterangan:

$x_1 \dots x_i \dots x_n$ = *neuron input*

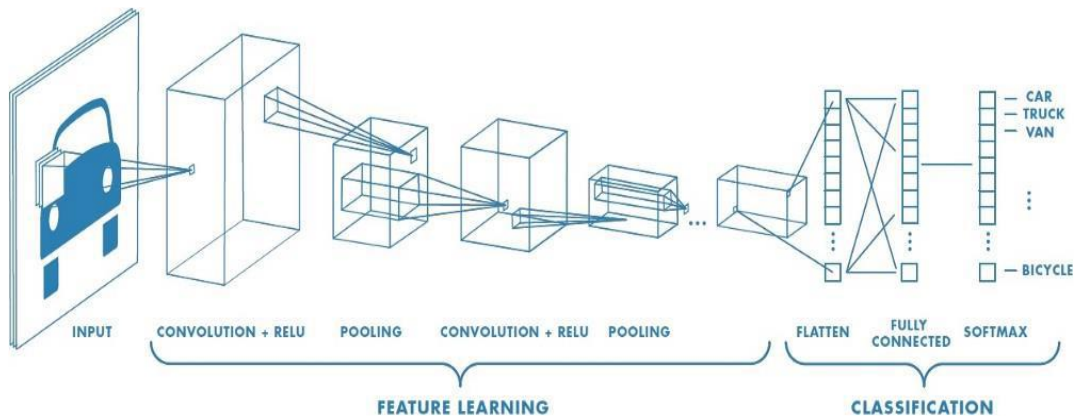
y = *neuron output*

b = *bias*

w_1, w_i, w_n = bobot [7]

2.4 *Convolutional Neuran Network (CNN)*

Convolutional Neural Network merupakan salah satu jenis jaringan saraf tiruan yang sering digunakan untuk pengolahan data gambar. *convolution* merupakan matriks yang memiliki fungsi melakukan *filter* pada gambar atau menggabungkan dua buah deret angka menghasilkan deret ketiga. Pada proses *training* terdapat 3 tahapan yaitu *Convolutional layer*, *Pooling layer*, dan *Fully connected layer*. [8]



Gambar 2. 2 Model CNN [8]

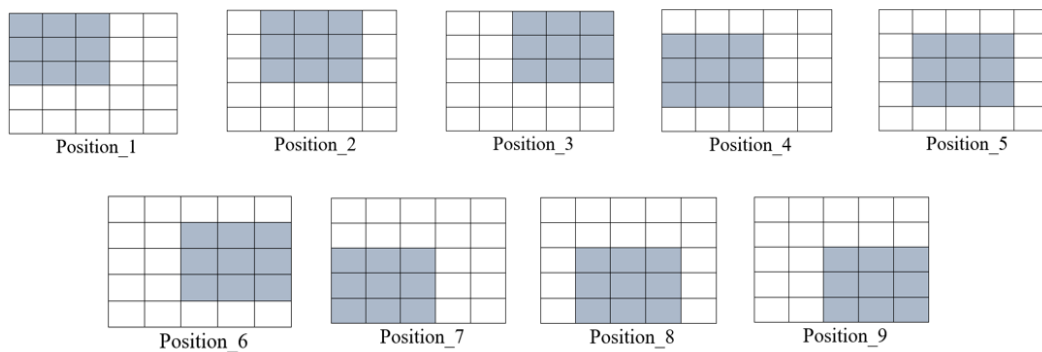
2.4.1 Proses konvolusi

Agar proses konvolusi dapat dipahami, berikut sampel dari sebagian matriks yang ada pada *input image*. Ukuran *input image* yang digunakan sebesar 80x80 *pixel*. Maka dari itu peneliti hanya menggunakan sebagian matriks dari *input image* untuk dijadikan contoh yang akan digunakan pada proses konvolusi.

$$\begin{bmatrix} 2 & 3 & 5 & 4 & 3 \\ 3 & 4 & 4 & 3 & 2 \\ 2 & 3 & 6 & 3 & 5 \\ 3 & 4 & 3 & 3 & 1 \\ 4 & 5 & 5 & 1 & 3 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -6 & -7 & 1 \\ -6 & 1 & -6 \\ -3 & -5 & 8 \end{bmatrix}$$

Gambar 2. 3 Proses Konvolusi [9]

Stride yang digunakan pada karnel adalah 1 dengan ukuran 3x3. *Stride* pergeseran yang digunakan bernilai 1. Visualisasinya adalah sebagai berikut:



Gambar 2. 4 Posisi Karnel dalam Konvolusi [9]

Berdasarkan gambar di atas merupakan proses konvolusi dengan kernel 3x3. Nilai *padding* yang digunakan adalah 1, agar terjadinya penambahan nilai 0 dinilai matriks inputan, yang nantinya berguna untuk menghasilkan nilai matriks *input* dan *output* yang sama. Perhitungan dari *dot product* adalah sebagai berikut:

$$\text{Position}_1 = (2*1) + (3*(-1)) + (2*1) + (3*(-1)) + (4*1) + (3*(-1)) + (5*1) + (4*(-1)) + (6*(-1)) = -6$$

$$\text{Position}_2 = (3*1) + (4*(-1)) + (3*1) + (5*(-1)) + (4*1) + (6*(-1)) + (4*1) + (3*(-1)) + (3*(-1)) = -7$$

$$\text{Position}_3 = (5*1) + (4*(-1)) + (6*1) + (4*(-1)) + (3*1) + (3*(-1)) + (3*1) + (2*(-1)) + (5*(-1)) = -1$$

$$\text{Position}_4 = (3*1) + (2*(-1)) + (3*1) + (4*(-1)) + (3*1) + (4*(-1)) + (3*1) + (2*(-1)) + (5*(-1)) = -6$$

$$\text{Position}_5 = (4*1) + (3*(-1)) + (4*1) + (4*(-1)) + (6*1) + (3*(-1)) + (3*1) + (3*(-1)) + (3*(-1)) = 1$$

$$\text{Position}_6 = (4*1) + (6*(-1)) + (3*1) + (3*(-1)) + (3*1) + (3*(-1)) + (2*1) + (5*(-1)) + (1*(-1)) = -6$$

$$\text{Position}_7 = (2*1) + (3*(-1)) + (4*1) + (3*(-1)) + (4*1) + (5*(-1)) + (6*1) + (3*(-1)) + (5*(-1)) = -3$$

$$\text{Position}_8 = (3*1) + (4*(-1)) + (5*1) + (6*(-1)) + (3*1) + (5*(-1)) + (3*1) + (3*(-1)) + (1*(-1)) = -5$$

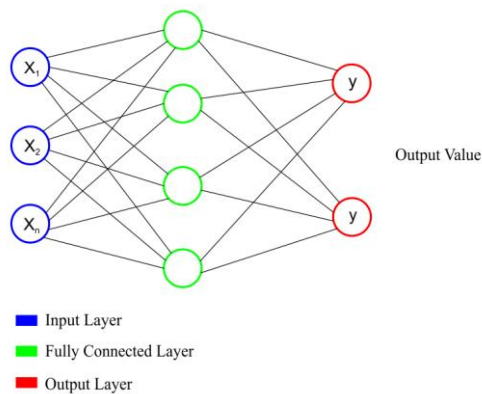
$$\text{Position}_9 = (6*1) + (3*(-1)) + (5*1) + (3*(-1)) + (3*1) + (1*(-1)) + (5*1) + (1*(-1)) + (3*(-1)) = 8$$

2.4.2 Pooling Layer

Pooling Layer biasanya berada setelah *Convolutional Layer*. Ada 2 jenis *Pooling Layer* yang sering digunakan yaitu *Max Pooling* dan *Average Pooling*. *Max Pooling* digunakan untuk mengambil nilai terbesar pada *area filter* yang digunakan, sedangkan *Average Pooling* digunakan untuk mengambil nilai rata-rata. *Pooling Layer* memiliki *filter* dan *stride* dengan ukuran tertentu, di mana setiap pergeserannya ditentukan oleh *stride* yang digunakan. [9]

2.4.3 Fully connected Layer

Feature map yang dihasilkan oleh tahap sebelumnya berbentuk *multidimensional array*. Sehingga, Sebelum masuk pada tahap *Fully Connected Layer*, *Feature Map* tersebut akan melalui proses “*flatten*” atau *reshape*. Fungsi dari proses *flatten* yaitu menghasilkan sebuah vektor yang akan dijadikan sebagai input. *Fully Connected Layer* memiliki beberapa *Hidden Layer*, *Action Function*, *Output Layer* dan *Loss Function* [8].



Gambar 2. 5 *Fully Connected Layer* [8]

2.5 Computer Vision

Computer Vision merupakan suatu ilmu yang mempelajari tentang cara pandang komputer terhadap gabungan dari *digital image processing*, *pattern recognition*, *computer graphics* dan *machine learning*. *Computer vision* juga merupakan kombinasi antara sistem pencahayaan (*lightning system*) dan analisa citra / gambar (*image analysis*). Otak manusia dapat mengenali wajah sesama manusia meskipun baru pertama kali bertemu dan dapat mengenali objek dengan cepat. Bagi komputer, sebuah gambar hanya merupakan sekumpulan *pixel*. *Computer Vision* bertujuan untuk mengenali sebuah gambar dengan baik, sehingga dapat menghasilkan informasi-informasi yang penting dalam sebuah gambar. Dengan kata lain tujuan *Computer Vision* adalah untuk mengajarkan komputer bisa membuat *pixel-pixel* menjadi hidup dan terasa seperti dunia nyata [10].

Computer Vision menjadikan komputer “*act like human sight*”, sehingga mendekati kemampuan manusia dalam menangkap informasi visual.

Kemampuan itu antaranya adalah:

1. *Object Detection*: Mengenali sebuah objek pada *scene* dan mengetahui dimana batasnya.
2. *Recognition*: Menempatkan label pada objek.
3. *Description*: Menugaskan properti adegan 3D dan 2D yang dilihat.
4. *Interpreting motion*: Menafsirkan gerakan [10].

Jadi pada proyek akhir ini penulis menggunakan open cv untuk memproses seluruh input gambar.

2.6 Tensorflow

Tensorflow merupakan salah satu *framework deep learning* dan juga salah satu *library* untuk *data science* yang bersifat *free open source* yang dikembangkan oleh para peneliti dari tim Google. *Tensorflow* dapat digunakan dalam berbagai bidang. Dalam bidang *object detection* terdapat *framework tensorflow object detection API* yang merupakan suatu alat yang dapat digunakan untuk mempermudah proses *constructing, training* dan *deployment* pada suatu model *object detection*. *Framework tensorflow object detection API* menyediakan *pretrained object detection* model bagi *user*, namun memungkinkan jika *user* ingin menggunakan *pretrained object detection* model yang lain, seperti *Faster R-CNN, SSD, Retinanet, Resnet50* dan masih banyak lagi. [11]

Tensorflow menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan. Fitur utamanya meliputi:

1. Mendefinikan, mengoptimalkan dan menghitung secara efisien ekspresi matematis yang melibatkan *array multidimension (tensor)*.
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

khususnya lagi, *Tensorflow* akan mengetahui bagian perhitungan yang harus dipindahkan ke GPU.

4. Skalabilitas komputasi yang tinggi di seluruh mesin dan kumpulan data yang besar [10].

2.7 Raspberry Pi

Raspberry Pi, sering disingkat dengan nama Raspi, adalah komputer papan tunggal (*single-board circuit*; SBC) yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry merupakan komponen paling penting dalam proyek akhir ini karena fungsinya untuk menjalankan seluruh program yang telah dibuat [12].



Gambar 2. 6 Raspberry pi 3 B+ [12]

Tabel 2. 1 Tabel Spesifikasi Raspberry Pi 3 Model B+ [12]

Komponen	Spesifikasi
<i>Processor</i>	Broadcom BCM2837B0, Cortex- A53 (ARMv8) 64-bit SoC @ 1.4GHz
<i>Memory</i>	1GB LPDDR2 SDRAM
<i>Power Supply</i>	5V/2.5A DC power input (microUSB)

<i>Wireless</i>	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac <i>wireless</i> LAN, <i>Bluetooth</i> 4.2, BLE
<i>Ethernet</i>	Gigabit Ethernet over USB 2.0 (<i>maximum throughput</i> 300 Mbps)
GPIOs	<i>Extended</i> 40-pin GPIO <i>header</i>
<i>Video Output</i>	Full-size HDMI
<i>Audio Output</i>	4-pole <i>stereo output</i> dan <i>port</i> <i>video</i> Komposit
USB <i>Port</i>	4 USB 2.0 <i>ports</i>
CSI <i>camera port</i>	Untuk menghubungkan kamera Raspberry Pi.
DSI <i>display port</i>	Untuk menghubungkan tampilan layar sentuh Raspberry Pi.
Micro SD port	Untuk membuat sistem operasi dan menyimpan data Anda
Power-over-Ethernet (PoE)	Memerlukan dukungan PoE HAT terpisah.
Suhu operasional	0~50°C
Dimensi	120mm x 75mm x 34mm

2.8 Webcam

Pada proyek akhir ini salah satu komponen yang digunakan yaitu kamera jenis webcam, webcam adalah sebuah *periferal* berupa kamera sebagai pengambil

citra/gambar dan mikropon (*optional*) sebagai pengambil suara/audio yang dikendalikan oleh sebuah komputer atau oleh jaringan komputer. Gambar yang diambil oleh webcam ditampilkan ke layar monitor, karena dikendalikan oleh komputer maka ada *interface* atau *port* yang digunakan untuk menghubungkan webcam dengan komputer atau jaringan. Ada beberapa orang mengartikan Webcam sebagai *web pages + camera*, karena dengan menggunakan webcam untuk mengambil gambar video secara aktual bisa langsung *di-upload* bila komputer yang mengendalikan terkoneksi internet [5].

BAB III

METODE PELAKSANAAN

3.1 Konsultasi dan *Study Pustaka*

Pertama melakukan konsultasi ke dosen pembimbing membahas tentang apa saja yang berkaitan dengan proyek akhir, Setelah melakukan konsultasi, melakukan *Study pustaka* dilakukan untuk mencari tahu tentang Sistem Pendeteksi Berbasis *Convolutional Nueral Network* (CNN) yang sudah pernah dibuat oleh orang lain guna untuk mendukung tugas akhir ini. Proses yang dilakukan pada tahap ini yaitu dengan mencari berbagai referensi baik dari *searching* di internet dan juga konsultasi dengan dosen pembimbing, Setelah itu kami langsung melakukan survei ke lapangan di perkebunan warga Desa Terentang III, Koba, Bangka Tengah. Ada beberapa hal yang kami tanyakan yaitu :

1. Bentuk buah lada.
2. Warna buah lada saat belum matang, setengah matang, dan matang.

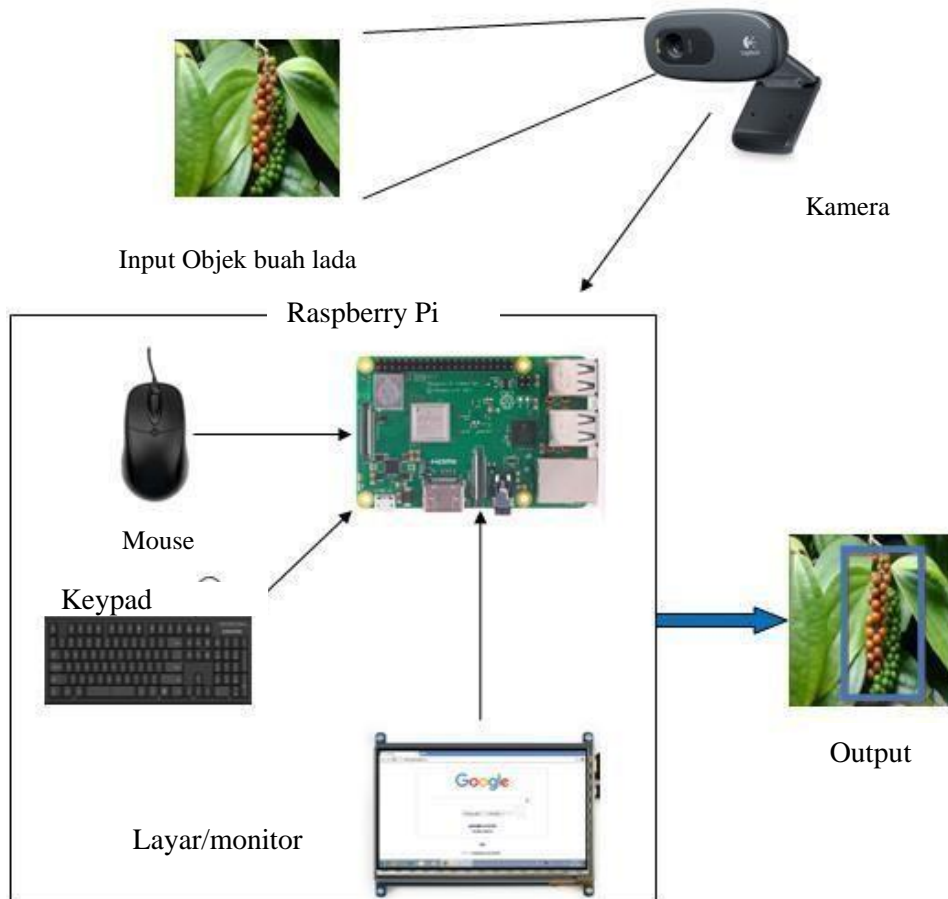


Gambar 3. 1 Survei ke perkebunan buah lada

Mengumpulkan data yang telah di dapatkan dari konsultasi ke pembimbing, *study pustaka*, dan survei. Data yang terkumpul dijadikan referensi untuk proses selanjutnya dalam pembuatan proyek akhir yang berjudul “Sistem Pendeteksi Buah Lada Berbasis *Convolutional Neural Network* (CNN)” dan juga sebagai sumber pembuatan makalah proyek akhir.

3.2 Rancangan sistem

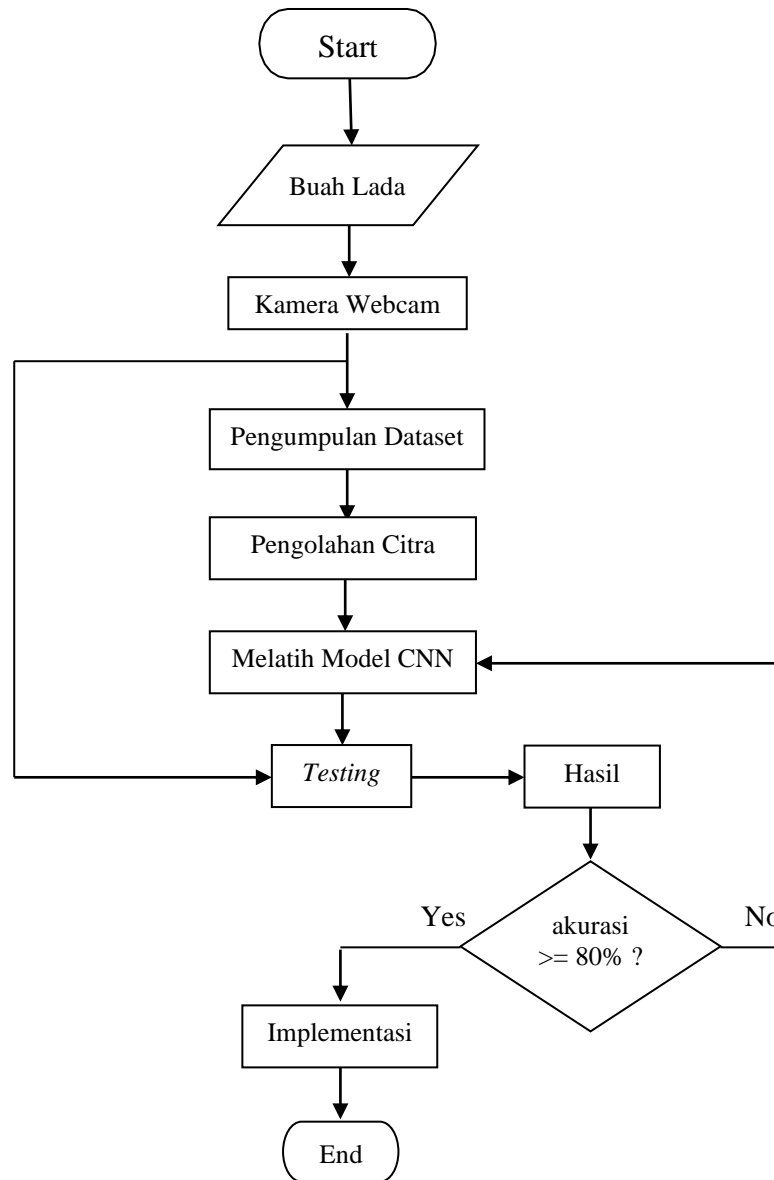
Pada gambar 3.2 merupakan rancangan sistem yang digunakan pada proyek akhir ini dengan menggunakan webcam dengan pendeteksi objek, setelah objek didapatkan maka *image* tersebut akan di proses oleh mini PC / Raspberry, di dalam mini PC, image akan diproses dalam beberapa metode yaitu *image prosesing* dan *neural network*. Seluruh proses metode tadi dilakukan maka hasil *output*-nya akan keluar seperti gambar 3.1, barulah sistem ini dapat mendeteksi objek buah lada.



Gambar 3. 2 Rancangan sistem

3.3 Sistem Kerja Alat

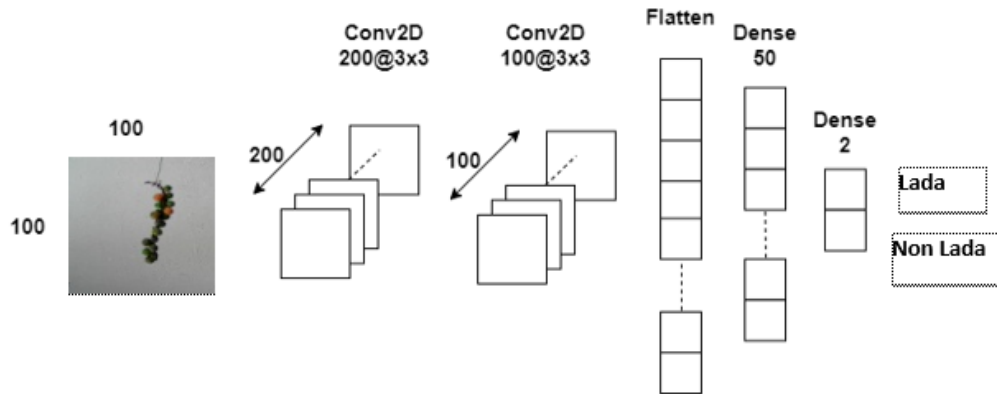
Pada pengerjaan proyek akhir ini terdapat cara kerja alat, Berikut ini adalah *flow chart* cara kerja alat yang ditunjukkan pada gambar 3.2.



Gambar 3. 3 *Flowchart* Sistem Kerja Alat

3.4 Arsitektur Jaringan Saraf Tiruan

Dalam proyek akhir ini, metode yang digunakan yaitu CNN (*Convolutional Neural Network*) untuk pengolahan citra gambar, bentuk model CNN yang digunakan sebagai berikut :



Gambar 3. 4 Arsitektur CNN

BAB IV

PEMBAHASAN

Pada bab ini, penulis akan menguraikan proses pengerjaan proyek akhir ini berdasarkan metode yang telah disebutkan pada bab sebelumnya. Penulis menggunakan aplikasi jenis *Python* untuk membuat program pada sistem yang akan digunakan pada proyek akhir ini. Berikut tahapan – tahapan yang dilakukan pada pembahasan kali ini menguraikan tentang :

1. Dataset
2. Pemrosesan Citra
3. Pemodelan Jaringan Saraf Tiruan
4. Hasil

4.1 Dataset

Pada proyek ini, penulis pertama sekali mengumpulkan data berupa set gambar buah lada, Dataset ini dijadikan *input* dalam proses sistem jaringan syaraf tiruan. Dataset gambar buah lada di dapatkan dari hasil penangkapan kamera tipe webcam.



Gambar 4. 1 Buah Lada

Berikut program untuk pengumpulan *dataset*:

1. Membuat folder *dataset* yang akan dikumpulkan, berikut program sumber yang dibuat:

```
import cv2
import os

if not os.path.exists("Dataset"):
```

```

os.makedirs("Dataset")
os.makedirs("Dataset/Lada")
os.makedirs("Dataset/Non Lada")

dirs=" Dataset/"

```

Keterangan:

Pada program diatas digunakan untuk membuat folder penyimpanan buah lada dengan nama folder yaitu Dataset, di dalam folder Dataset terdapat folder lagi yaitu folder Lada dan Non Lada. Folder Lada digunakan untuk tempat penyimpanan buah lada, sedangkan folder Non Lada digunakan untuk menyimpan gambar yang bukan buah lada.

2. Menyimpan gambar di folder di dataset dengan program sebagai berikut:

```

cap=cv2.VideoCapture(0)
i=0
a=0

while(True);
    _, frame = cap.read()
    x1=int(0.5*frame.shape[1])
    y1=10
    x2=int(0.5*frame.shape[1])
    cv2.rectangle(frame, (x1-1,y1-1), (x2+1,y2+1), (255,0,0),1)
    roi=frame[y1:y2 , x1:x2]
    roi=cv2.resize(roi, (100,100))

    cv2.imshow("kamera",frame)
    cv2.imshow("roi",roi)

    interrupt=cv2.waitKey(10)
    if interrupt & 0xFF == 27:
        break
    if interrupt & 0xFF == ord("1"):
        cv2.imwrite(dirs+"Lada/"+str(i)+".jpg",roi)
        i+=1
    if interrupt & 0xFF == ord("2"):
        cv2.imwrite(dirs+"Non Lada/"+str(a)+".jpg",roi)
        a+=1

```

Keterangan:

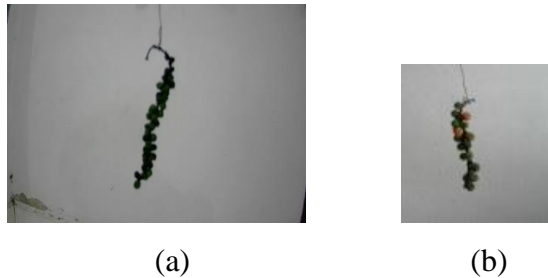
Pada program di atas digunakan untuk menyimpan gambar yang di tangkap dari kamera tipe webcam, untuk proses penyimpananya bisa dilakukan dengan menekan angka 1 dan 2 pada keyboard. Pada saat menekan angka 1 maka gambar akan tersimpan di folder Lada, sedangkan angka 2 maka gambar akan tersimpan di folder Non Lada.

4.2 Pemrosesan Citra

Setelah melakukan pengumpulan dataset berupa gambar buah lada, selanjutnya melakukan proses citra dengan tahapan - tahapan sebagai berikut :

1. *Resize* Gambar

Ukuran pada gambar sangat mempengaruhi dalam proses pengolahan data *training*, semakin besar ukuran gambar yang digunakan maka semakin sulit dalam melakukan proses pengolahan data *trainig*. Ukuran yang di simpan oleh kamera tipe webcam adalah 640 x 360 piksel, ukuran ini sangat besar dan juga terlalu berat jika digunakan sehingga dilakukan *resize* data sebelum memasuki tahap *training*. Ukuran gambar yang digunakan yaitu 100 x 100



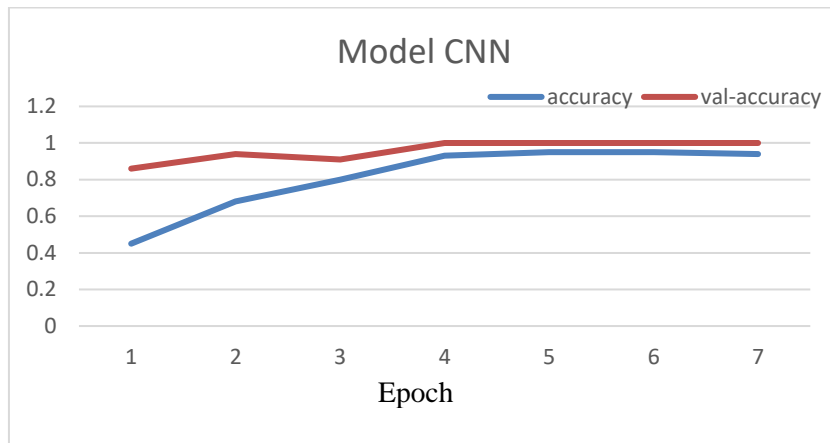
Gambar 4. 2 Sebelum di resize (b) Setelah di resize

2. *Grayscale*

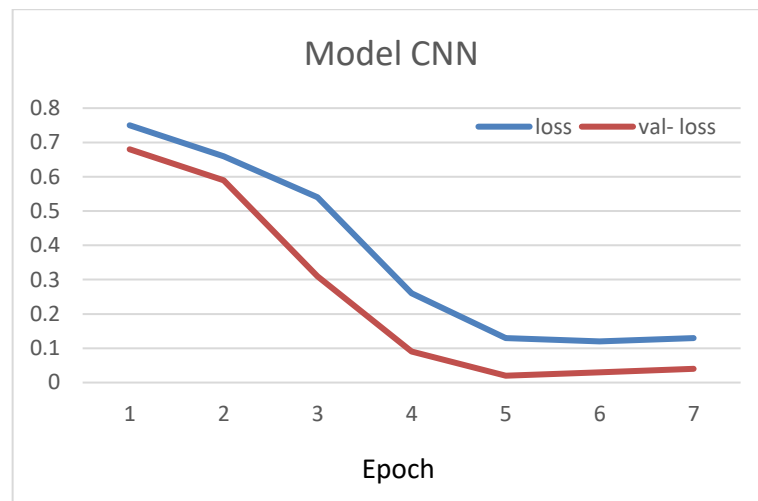
Setelah melakukan proses *Resize* data, selanjutnya dataset gambar yang telah di *resize* di *convert* ke *grayscale*. Gambar RGB lebih sulit diproses dibandingkan dengan gambar *grayscale*, jadi dalam proses ini gambar RGB akan di *convert* ke *Grayscale*.

4.3 Pemodelan Jaringan Syaraf Tiruan

Proses selanjutnya melakukan pelatihan pada jaringan syaraf tiruan untuk mempelajari suatu pola, dimana dapat menghasilkan suatu pengenalan deteksi objek yang sesuai dengan apa yang dilatih, Setelah melakukan proses citra pada gambar, selanjutnya melakukan proses *training* jaringan syaraf tiruan. Algoritma yang digunakan yaitu *Convolutional Neural Network* (CNN) yang merupakan *tools* dari Tensorflow, berikut hasil model yang telah di *training* dibawah ini:



Gambar 4. 3 Grafik *Accuracy* Model CNN



Gambar 4. 4 Grafik *Loss* Model CNN




Proses pelatihan model CNN menggunakan 2 *class*, untuk *class* pertama berupa gambar buah lada sedangkan *class* kedua berupa gambar non lada, Banyak gambar masing – masing *class* yaitu 100 gambar dengan ukuran 100 x 100 piksel. Terdapat 2 gambar grafik yaitu grafik *accuracy* dan grafik *loss*, jadi nilai *accuracy* digunakan untuk mengetahui tingkat keberhasilan model yang telah dibuat. Sedangkan nilai *loss* merupakan nilai *error* dari model yang telah dibuat. Berdasarkan grafik di atas jumlah *epoch* yang di *training* yaitu 7 *epoch*. *Epoch* merupakan proses training keseluruhan dataset dari awal sampai akhir dalam 1 kali putaran, Jadi untuk mengetahui model CNN yang bagus digunakan yaitu nilai *accuracy* semakin tinggi sedangkan nilai *loss* semakin rendah. Berdasarkan








gambar grafik di atas dapat disimpulkan bahwa semakin banyak model CNN di latih maka semakin baik pula model CNN dalam memprediksi objek yang diinginkan. Ada sebanyak 7 *epoch* yang di *training*, waktu yang dibutuhkan saat proses *training* yaitu 8 jam dan hasil model yang telah dilatih menunjukkan hasil yang bagus berdasarkan dari nilai *accuracy* dan *loss*.

4.4 Hasil

Setelah model dari Jaringan CNN di dapat selanjutnya melakukan pengujian secara langsung dengan menggunakan kamera tipe webcam, kami menggunakan beberapa *sample* buah atau benda seperti buah lada, buah cabai dan buah mengkudu untuk melihat seberapa akurat model CNN yang telah dilatih. berikut tabel hasil pengujian dengan mengambil beberapa gambar yang telah diuji dari 85 gambar :

Tabel 4. 1 Hasil Prediksi Model CNN

No	Nama Benda	Gambar	Prediksi
1.	Buah Lada		Benar
2.	Buah Lada		Benar
3.	Buah Lada		Benar

4.	Buah Lada		Benar
5.	Buah Cabai		Benar
6.	Buah Mengkudu		Benar
7.	Daun dan tangkai		Benar
8.	Daun lada		Benar
9.	Buah lada		Salah
10.	Daun dan tangkai		Salah

Berdasarkan dari hasil pengujian diatas sebanyak 85 gambar yang telah diuji,

terdapat 5 jenis gambar yang diuji yaitu buah lada, daun lada, tangkai lada, buah cabai dan buah mengkudu, Dari jenis – jenis gambar tersebut akan dibandingkan dengan buah lada itu sendiri, apakah sistem ini bisa berjalan dengan baik. Terdapat 75 gambar dengan prediksi benar sedangkan sisanya salah prediksi. Perhitungan nilai *Accuracy*, *Recall*, dan *Precision* dari hasil pengujian diatas sebagai berikut :

Tabel 4. 2 Hasil Nilai Prediksi

<i>True Positive</i> (TP)	<i>True Negative</i> (TN)	<i>False Positive</i> (FP)	<i>False Negative</i> (FN)
59	17	3	6

$$Precision = \frac{TP}{TP+FP} \times 100\%$$

$$Precision = \frac{59}{59+3} \times 100\% = 95\%$$

$$Recall = \frac{TP}{TP+FN} \times 100 \%$$


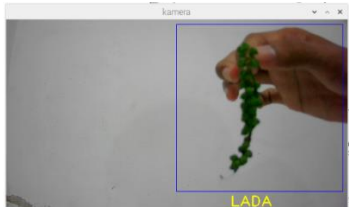
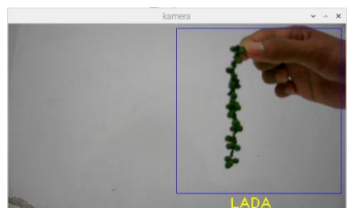
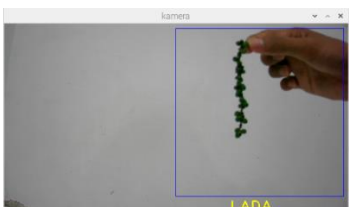
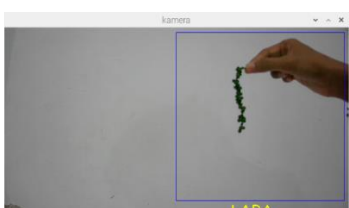
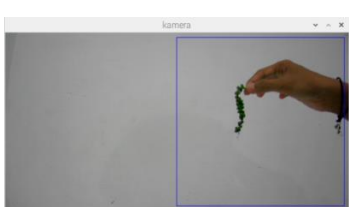
$$Recall = \frac{59}{59+6} \times 100\% = 90\%$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

$$Accuracy = \frac{59+17}{59+17+3+6} \times 100 \% = 89\%$$

Jadi hasil *Accuracy*, *Recall*, dan *Precision* dari model yang digunakan dengan *input* gambar ukuran 100 x 100 piksel yaitu *Accuracy* 89%, *Recall* 90% dan presisi 95%. Hasil pengujian sudah menunjukkan hasil yang baik. Cara meningkatkan akurasi dari model CNN dalam memprediksi objek buah lada dapat menambah *input* gambar atau dataset lebih banyak lagi sehingga tingkat akurasi lebih tinggi

Tabel 4. 3 Hasil Prediksi dengan Jarak

No	Jarak	Gambar	Asil
1.	9 cm		Tidak terdeteksi
2.	10 cm		Terdeteksi
3.	15 cm		Terdeteksi
4.	25 cm		Terdeteksi
5.	30 cm		Terdeteksi
6.	31 cm		Tidak Terdeteksi

7. 35 cm



Tidak Terdeteksi

Berdasarkan dari hasil tabel di atas jika jarak buah lada dengan kamera kurang dari 10 cm maka objek yang ditangkap oleh kamera terlalu besar sehingga sistem tidak dapat mengenali buah lada dengan baik, jika jarak buah lada dengan kamera lebih dari 30 cm maka objek yang ditangkap oleh kamera terlalu kecil sehingga sistem tidak dapat mengenali buah lada dengan baik, jadi dapat disimpulkan bahwa sistem pendeteksi dapat mengenali buah lada dengan jarak 10 cm – 30 cm dari kamera webcam.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan dari pembahasan dan hasil analisa yang telah dilakukan dalam proyek akhir ini dapat disimpulkan sebagai berikut :

1. Sistem pendeteksi buah lada berbasis *Convolutional Neural Network* (CNN) yang telah dibuat di proyek akhir ini dapat memprediksi buah lada dengan tingkat akurasi 88% yang ditampilkan di layar monitor.
2. Semakin banyak proses *training* model dilatih maka semakin baik juga tingkat akurasi dalam memprediksi objek yang diinginkan, sebaliknya jika proses training sedikit dilatih maka tingkat akurasinya kurang bagus dalam memprediksi objek.
3. Ukuran gambar sangat mempengaruhi proses pengolahan data training, semakin besar ukuran gambar yang digunakan maka semakin lama dalam melakukan proses pengolahan data training.
4. Input gambar yang dilatih pada pembahasan diatas sebanyak 100 gambar untuk setiap *class*, jumlah ini terbilang sedikit untuk proses pelatihan model CNN, untuk mendapatkan model CNN yang bagus maka diperlukan input / dataset yang banyak.
5. bahwa sistem pendeteksi dapat mengenali buah lada dengan jarak 10 cm – 30 cm dari kamera webcam berdasarkan hasil pengujian.

5.2 Saran

Dari hasil proyek akhir ini penulis memiliki beberapa saran untuk pengembangan yang lebih lanjut sebagai berikut :

1. Lebih banyak menambahkan input gambar atau dataset untuk melatih model CNN serta dapat memperoleh akurasi yang tinggi
2. Untuk kedepannya bisa mengenali suatu objek melalui kombinasi warna dan bentuk
3. Lebih baik menggunakan kamera webcam dengan kualitas yang bagus.

DAFTAR PUSTAKA

- [1] Ardan Adhi Chandra, "Awalnya Iseng, Pengusaha Ini Bisa Kantongi Rp 6M/ Bulan dari BIsnis Lada Putih," *detikFinance*, Rabu 08 Juni 2016. [Online]. Available: <https://finance.detik.com/berita-ekonomi-bisnis/d-3228160/awalnya-iseng-pengusaha-ini-bisa-kantongi-rp-6-m-bulan-dari-bisnis-lada-putih>. [Accessed 26 November 2020].
- [2] Marisa, "Budidaya Lada," 17 Januari 2021. [Online]. Available: <https://www.teknikbudidaya.com/budidaya-lada/>. [Accessed 08 Maret 2021].
- [3] A. Prasajo, "PENGENALAN KARAKTER ALFABETMENGUNAKAN JARINGAN SARAF TIRUAN," p. 2.
- [4] M. F. T. Hakim, "PENGENALAN KARAKTER ALFABETMENGUNAKAN JARINGAN SARAF TIRUAN," pp. 26-27, 2018.
- [5] D. A. Probowo, D. Abdullah and A. Manik, "DETEKSI DAN PERHITUNGAN OBJEK BERDASARKAN WARNA MENGGUNAKAN COLOR OBJECT TRACKING," *Jurnal Pseudocode*, vol. 5, no. 2, pp. 85-91, 2018.
- [6] D. A. Ulandari and D. Swanjaya, "Perbandingan Transformasi Data pada Penentuan Peserta Bimbingan Belajar Menggunakan Metode Perceptron," vol. 4, no. 3, p. 2, 2020.
- [7] A. Pujjyanta, "PENGENALAN CITRA OBJEK SEDERHANA DENGAN JARINGAN SYARAF TIRUAN METODE PERCEPTRON," *Jurnal Informatika Ahmad Dahlan*, vol. 3, no. 1, p. 3, 2009.
- [8] A. Santoso, "Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah," Program Studi Informatika Fakultas Komunikasi dan Informatika, Surakarta, 2018.
- [9] W. Anggraini, "Deep Larning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (CNN) Dengan Tensorflow," Fakultas Tarbiyah Dan Keguruan Universitas Islam Negeri Ar-Raniry, Banda Aceh, 2020.
- [10] S. R. Dewi, "DEEP LEARNING OBJECT DETECTION PADA VIDEO MENGGUNAKAN TENSORFLOW DAN CONVOLUTIONAL NEURAL," pp. 1-95, 2018.
- [11] D. Manajang, S. R. U. A. Sompie and A. Jacobus, "Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor," *Jurnal Teknik Informatika*, vol. 15, no. 3, p. 173, 2020.
- [12] L. Elektronika, "MENGENAL SINGLE BOARD KOMPUTER RASPBERRY Pi 3 MODEL B+," Selasa, 5 Juni 2018. [Online]. Available: <http://www.labelektronika.com/2018/06/mengenal-raspberry-pi-3-model-b-plus.html>. [Accessed 21 11 2020].
- [13] M. N. Firmansyah, "PENGENALAN KARAKTER

- ALFABETMENGGUNAKAN JARINGAN SARAF TIRUAN," p. 8, 2019.
- [14] A. Santoso, "Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah," Program Studi Informatika, Surakarta, 2018.

LAMPIRAN

LAMPIRAN 1
DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Abdur Rohim
Tempat & Tanggal lahir : Terantang III, 17 September
1998
Alamat : Desa Terantang III, Koba, kep.
Bangka Belitung.
Jenis Kelamin : Laki-laki
Agama : Islam
Telp : -
Hp : 081293172208
E-mail : caxem17@gmail.com
Hobi : Bulutangkis



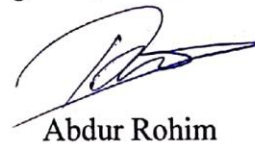
2. Riwayat Pendidikan

SDN 10 Koba Lulus Tahun 2011
SMPN 1 Koba Lulus Tahun 2014
SMKN 1 Koba Lulus Tahun 2017

3. Riwayat Pendidikan Non Formal

-

Sungailiat, 09 Maret 2021



Abdur Rohim

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Reiva Marizka Harmie
Tempat & Tanggal lahir : Sungailiat, 21 Maret 1999
Alamat : Jl. Bukit Rempuding No. 25
kec. Sungailiat, kab. Bangka,
kep. Bangka Belitung.
Jenis Kelamin : Perempuan
Agama : Islam
Telp : -
Hp : 081278826568
E-mail : reivamh@gmail.com
Hobi : Menggambar



2. Riwayat Pendidikan

SDN 3 Sungailiat 2011
SMP Setia Budi Sungailiat 2014
SMAN 1 Sungailiat 2017

3. Riwayat Pendidikan Non Formal

-

Sungailiat, 09 Maret 2021

A handwritten signature in black ink, appearing to read 'Reiva'.

Reiva Marizka Harmie

LAMPIRAN 2
PROGRAM SISTEM PENDETEKSI BUAH LADA BERBASIS
CONVOLUTIONAL NEURAL NETWORK (CNN)

```

import cv2
import os
if not os.path.exists("Dataset"):
    os.makedirs("Dataset")
    os.makedirs("Dataset/Lada")
    os.makedirs("Dataset/Non Lada")
dirs="Dataset/"
cap=cv2.VideoCapture(0)
i=0
a=0
while(True):
    _, frame = cap.read()
    x1=int(0.5*frame.shape[1])
    y1=10
    x2=int(frame.shape[1]-10)
    y2=int(0.5*frame.shape[1])
    cv2.rectangle(frame, (x1-1,y1-1), (x2+1,y2+1), (255,0,0),1)
    roi=frame[y1:y2 , x1:x2]
    roi=cv2.resize(roi, (100,100))
    cv2.imshow("kamera",frame)
    cv2.imshow("roi",roi)
    interrupt=cv2.waitKey(10)
    if interrupt & 0xFF == 27:
        break
    if interrupt & 0xFF == ord("1"):
        cv2.imwrite(dirs+"Lada/"+str(i)+".jpg",roi)
        i+=1
    if interrupt & 0xFF == ord("2"):
        cv2.imwrite(dirs+"Non Lada/"+str(a)+".jpg",roi)
        a+=1
cap.release()

```

```

import cv2,os
data_path='dataset'
categories=os.listdir(data_path)
labels=[i for i in range(len(categories))]
label_dict=dict(zip(categories,labels))
print(label_dict)
print(categories)
print(labels)
img_size=100
data=[]
target=[]
for category in categories:
    folder_path=os.path.join(data_path,category)
    img_names=os.listdir(folder_path)
    for img_name in img_names:
        img_path=os.path.join(folder_path,img_name)
        img=cv2.imread(img_path)
        try:
            gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
            resized=cv2.resize(gray,(img_size,img_size))
            data.append(resized)
            target.append(label_dict[category])

        except Exception as e:
            print('Exception:',e)

import numpy as np
data=np.array(data)/255.0
data=np.reshape(data,(data.shape[0],img_size,img_size,1))
target=np.array(target)
from keras.utils import np_utils
new_target=np_utils.to_categorical(target)

np.save('data',data)
np.save('target',new_target)

```

```

import numpy as np

data=np.load('data.npy')
target=np.load('targe.npy')

from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Activation
from keras.layers import Conv2D, MaxPooling2D
from keras.callbacks import ModelCheckpoint
model = Sequential()

model.add(Conv2D(200, (3,3), input_shape=data.shape[1:]))
model.add(Activation('relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(100, (3,3)))
model.add(Activation('relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dropout(0.5))

model.add(Dense(50,activation='relu'))
model.add(Dense(2,activation='softmax'))

model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
from sklearn.model_selection import train_test_split
train_data,test_data,train_target,test_target=train_test_split(dat
a,target,test_size=0.1)
checkpoint = ModelCheckpoint('model-
{epoch:03d}.model',monitor='val_loss',verbose=0,save_best_only=True,
e,mode='auto')
history=model.fit(train_data,train_target,epochs=50,callbacks=[che
ckpoint],validation_split=0.2)

```



```
from matplotlib import pyplot as plt
plt.plot(history.history['loss'],'r',label='training loss')
plt.plot(history.history['val_loss'],'r',label='validation loss')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
plt.plot(history.history['accuracy'],'r',label='training
accuracy')
plt.plot(history.history['val_accuracy'],'r',label='validation
accuracy')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()

print(model.evaluate(test_data,test_target))
```

```

import cv2
import numpy as np
import tensorflow as tf
import operator
import sys, os

model = tf.keras.models.load_model("model-003.model")
cap=cv2.VideoCapture(0)

while(True):
    _, frame = cap.read()

    frame = cv2.flip(frame, 1)

    x1=int(0.5*frame.shape[1])
    y1=10
    x2=int(frame.shape[1]-10)
    y2=int(0.5*frame.shape[1])

    cv2.rectangle(frame, (x1-1,y1-1), (x2+1,y2+1), (255,0,0),1)
    roi=frame[y1:y2 , x1:x2]
    roi=cv2.resize(roi, (100,100))

    roi=cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
    reshaped=np.reshape(roi, (-1,100,100,1))
    prediksi=model.predict(reshaped)

    print(prediksi)

    prediction = {'LADA': prediksi[0][1],
                  'NON LADA': prediksi[0][0]}

    # Sorting based on top prediction
    prediction = sorted(prediction.items(),
key=operator.itemgetter(1), reverse=True)

```

```
# Displaying the predictions
cv2.putText(frame, prediction[0][0], (x1+100, y2+30),
cv2.FONT_HERSHEY_PLAIN, 2, (0,255,255), 2)

cv2.imshow('kamera',frame)
cv2.imshow('roi',roi)
interrupt=cv2.waitKey(10)
if interrupt & 0xFF == 27:
    break

cap.release()
cv2.destroyAllWindows()
```