

**STUDI PENGGUNAAN METODE JARINGAN SYARAF TIRUAN
BACKPROPAGATION UNTUK MENENTUKAN KEMATANGAN BUAH
KOPI**

(SKRIPSI)

Oleh

Yeni Apriyana



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2019**

ABSTRACT

STUDY OF THE USE ARTIFICIAL NEURAL NETWORK BACKPROPAGATION METHOD FOR DETERMINING MATURE COFFEE

BY

YENI APRIYANA

Identification of coffee is a process in determining for maturity of coffee. When processing coffee, selection of coffee were done in manually by farmer on its determination. This makes the process slower and it makes no certain results for better quality of coffee, that's why need to develop a system to identify better quality of coffee. In this research, doing first study to develop system to identify maturity of coffee using Artificial Neural Network backpropagation method. The first step of designing system is preprocessing in picture processing before entering Artificial Neural Network backpropagation step. Preprocessing process is useful to increase contrast, to remove background in picture and read RGB's score for input in the next step. The next step is Artificial Neural Network backpropagation system, it doing the training after the system read the pattern correctly, after that, the system doing test using different data with training. The final result in this research is the system could determine the expected target and could read the training pattern correctly when doing testing. The level of system accuracy informed when the system succeeded testing the process, The 4th output conditions system are (mature, immature, over mature, and unknown) display the percentage results of succeed 96,66% from 60 samples of testing pictures. This research is expected to help in developing identification system of maturity coffee by using Artificial Neural Network backpropagation.

Keywords: coffee, Artificial Neural Network, backpropagation

ABSTRAK

STUDI PENGGUNAAN METODE JARINGAN SYARAF TIRUAN *BACKPROPAGATION* UNTUK MENENTUKAN KEMATANGAN BUAH KOPI

OLEH

YENI APRIYANA

Identifikasi kopi merupakan proses dalam menentukan kematangan kopi. Saat mengolah kopi, pemilihan kopi dilakukan secara manual oleh petani dalam penentuannya. Hal itu membuat proses menjadi lebih lambat dan tidak membuat kepastian untuk kualitas kopi yang bermutu baik, untuk itu perlu mengembangkan sistem untuk mengidentifikasi kualitas kopi yang baik. Dalam penelitian ini, dilakukan studi awal untuk mengembangkan sistem untuk mengidentifikasi kematangan kopi menggunakan metode JST *backpropagation*. Tahapan awal merancang sistem yaitu *preprocessing* dalam pemrosesan gambar sebelum memasuki langkah JST *backpropagation*. Proses *preprocessing* berguna untuk meningkatkan kontras, menghapus *background* dalam gambar dan membaca nilai RGB untuk input pada tahap berikutnya. Pada tahap selanjutnya adalah sistem JST *backpropagation*, melakukan *training* setelah sistem membaca polanya dengan benar, selanjutnya sistem melakukan *testing* menggunakan data yang berbeda dengan pelatihan. Hasil akhir dalam penelitian ini adalah sistem dapat menentukan target yang diharapkan dan dapat membaca pola pelatihan dengan benar pada saat melakukan *testing*. Tingkat akurasi sistem yang di informasikan ketika sistem pada proses *testing*, sistem memiliki 4 kondisi output yaitu (matang, tidak matang, terlalu matang dan tidak diketahui) menampilkan hasil presentase berhasil 96,66% dari 60 sample gambar testing. Penelitian ini diharapkan dapat membantu dalam mengembangkan sistem identifikasi kematangan kopi dengan menggunakan JST *backpropagation*.

Kata kunci : kopi, JST, *backpropagation*.

**STUDI PENGGUNAAN METODE JARINGAN SYARAF TIRUAN
BACKPROPAGATION UNTUK MENENTUKAN KEMATANGAN BUAH
KOPI**

Oleh

Yeni Apriyana

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK

Pada

Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2019**

Judul Skripsi : **STUDI PENGGUNAAN METODE
JARINGAN SYARAF TIRUAN
BACKPROPAGATION UNTUK
MENENTUKAN KEMATANGAN
BUAH KOPI**

Nama Mahasiswa : **Yeni Apriyana**

Nomor Pokok Mahasiswa : 1415061041

Program Studi : Teknik Informatika

Jurusan : Teknik Elektro

Fakultas : Teknik



Yessi Mulyani, S.T., M.T.
NIP 19731226 200012 2 001

Wahyu Eko S, S.T., M.Sc.
NIP 19651021 199512 2 001

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Dr. Herman Halomoan S, S.T., M.T.
NIP 19711130 199903 1 003

Ketua Program Studi
Teknik Informatika

Yessi Mulyani, S.T., M.T.
NIP 19731226 200012 2 001

MENGESAHKAN

1. Tim Penguji

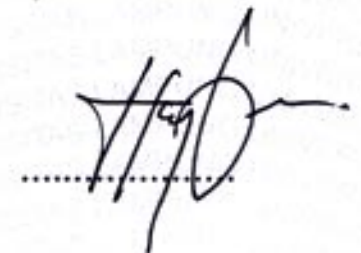
Ketua : **Yessi Mulyani, S.T., M.T.**



Sekretaris : **Wahyu Eko S, S.T., M.Sc.**



Penguji
Bukan Pembimbing : **Ing. Hery Dian Septama, S.T.**



2. Dekan Fakultas Teknik



Prof. Suharno, M.Sc., Ph.D.

NIP. 19620717 198703 1 002

Tanggal Lulus Ujian Skripsi : **6 Maret 2019**

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan oleh orang lain dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan di dalam daftar pustaka. Selain itu saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, Maret 2019



Yeni Apriyana

Yeni Apriyana

1415061041

RIWAYAT HIDUP



Penulis dilahirkan di Gisting pada tanggal 9 April 1996, sebagai anak kedua dari dua empat bersaudara, dari Bapak Syamsul Azhar dan Ibu Juwita Mulia.

Pendidikan Taman Kanak-kanak (TK) Aisyah diselesaikan tahun 2002, Sekolah Dasar (SD)

diselesaikan di SDN 1 Rejosari, Pringsewu pada tahun 2008, Sekolah Menengah Pertama (SMP) di SMPN 1 Pringsewu pada tahun 2011, dan Sekolah Menengah Akhir (SMA) di SMAN 2 Pringsewu pada tahun 2014.

Tahun 2014, penulis terdaftar sebagai mahasiswa Jurusan Teknik Elektro Program Studi Teknik Informatika Universitas Lampung melalui jalur UML. Selama menjadi mahasiswa penulis pernah menjadi asisten praktikum Pemrograman Web, Aktif di Organisasi Badan Eksekutif Mahasiswa Universitas (BEM U) dan Aktif di Organisa Badan Eksekutif Mahasiswa Fakultas Teknik (BEM FT) Universitas Lampung. Pada tahun 2017, Penulis melakukan Kerja Praktek di Balai Besar Wilayah Mesuji Sekampung.

PERSEMBAHAN

Skripsi ini kupersembahkan untuk:

- 1. Ayah dan Mama tersayang, yang telah membesarkan dan membimbing saya dari kecil, yang selalu memberikan motivasi dan nasihat dalam penyelesaian skripsi ini.*
- 2. Uwo Marisya, Dik Lely dan Adik Faiz, yang selalu memberikan motivasi positif dan semangat untuk menyelesaikan skripsi ini.*

SANWACANA

Bismillahirrahmanirrahim...

Puji syukur Penulis ucapkan kehadirat Allah SWT, karena atas rahmat dan hidayah-Nya skripsi ini dapat diselesaikan tepat pada waktunya. Skripsi yang berjudul “*Studi Penggunaan Metode Jaringan Syaraf Tiruan Metode Backpropagation Untuk Menentukan Kematangan Buah Kopi*” adalah salah satu syarat untuk memperoleh gelar sarjana Teknik di Universitas Lampung.

Dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Prof. Suharno, M.Sc. Ph.D selaku Dekan Fakultas Teknik;
2. Dr. Herman H Sinaga selaku Ketua Jurusan Teknik Elektro, Universitas Lampung;
3. Yessi Mulyani S.T., M.T., selaku Ketua Program Studi Teknik Informatika Universitas Lampung sekaligus selaku Pembimbing Utama, atas kesediaan waktunya memberikan bimbingan, saran, masukan, motivasi, dan pengarahan yang membangun dalam proses penyelesaian skripsi ini ;
4. Wahyu Eko S, S.T., M.Sc., selaku Pembimbing Kedua, yang telah meluangkan waktunya atas bimbingan, saran, masukan, serta kritik dalam menyelesaikan skripsi ini;
5. Ing Hery Dian Septama,S.T, selaku Penguji Utama, yang telah memberikan kritik, saran, dan koreksi dalam penyelesaian skripsi ini;
6. Bapak Dr. Herman H Sinaga, selaku Sekretaris Jurusan Teknik Elektro;
7. Seluruh Dosen Jurusan Teknik Elektro dan Program Studi Teknik Informatika Universitas Lampung, atas pengajaran dan bimbingan yang telah

diberikan kepada penulis selama menjadi mahasiswa Teknik Elektro Universitas Lampung;

8. Mbak Rika dan seluruh jajarannya atas semua bantuannya dalam menyelesaikan urusan administrasi di Jurusan Teknik Elektro Universitas Lampung;
9. Ayah, Mama, Uwo, dan Adik-adik yang selalu memberikan masukan, nasihat, dorongan, dan do'a kepada saya untuk menyelesaikan skripsi ini;
10. A. Aldino Rizki yang selalu merusuhi, buat kesal tapi tetap selalu mensupport saya dalam susah senang selama pembuatan skripsi ini;
11. Citra, Ipeh, dan Ka Tia yang selalu menemani mensupport mendampingi memberi motivasi di saat terpuruk dan selalu menjadi penyemangat.
12. Gin, Kadek, Fahreza, Meri dan teman-teman Teknik Informatika 2014 tercinta yang tidak bisa saya tuliskan satu persatu namanya atas dukungan dan telah memberikan semangat pada saya;

Akhir kata, Penulis meminta maaf karena menyadari masih ada kekurangan pada tugas akhir yang dikerjakan ini, untuk itu kritik dan saran yang membangun masih diperlukan demi kemajuan di masa depan. Semoga Allah SWT membalas kebaikan semua pihak yang telah membantu penulis dalam penyelesaian Tugas Akhir ini.

Bandar Lampung, Maret 2019

Penulis

Yeni Apriyana

DAFTAR ISI

	Halaman
DAFTAR ISI	i
DAFTAR GAMBAR	iv
DAFTAR TABEL	vi
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Kerja Penelitian.....	4
1.3 Rumusan Masalah	4
1.4 Batasan Masalah.....	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	5
II. TINJAUAN PUSTAKA	7
2.1 Kopi.....	7
2.2 Proses Pengolahan Biji Kopi	9
2.2.1 Pengolahan Cara Kering	9
2.2.2 Pengolahan Citra Basah	10
2.3 Definisi Citra.....	10
2.4 Pengolahan Citra Digital.....	11
2.4.1 Penskalaan	12
2.4.2 Representasi Warna	12
2.4.3 Format File Citra JPEG (.jpg).....	13
2.4.4 Ruang Warna RGB	13
2.4.5 Pemerataan Kontras	14
2.5 Kecerdasan Buatan	15
2.6 Jaringan Syaraf Tiruan.....	15
2.6.1 Arsitektur Jaringan.....	16
2.6.2 Algoritma Pembelajaran JST	17

2.6.3 Fungsi Aktifitasi, Bias dan Threshold	17
2.7 Arsitektur <i>Backpropagation</i>	19
2.7.1 Fungsi Aktivitasi <i>Backpropagation</i>	21
2.7.2 Pelatihan <i>Backpropagation</i>	22
2.8 Optimalitas Arsitektur <i>Backpropagation</i>	25
2.8.1 Pemilihan Bobot dan Bias Awal	26
2.8.2 Jumlah Pola Pelatihan	26
2.8.3 Lama Iterasi	26
2.8.4 Momentum	27
2.9 <i>Backpropagation</i> dengan Matlab.....	27
2.9.1 Matriks dan Vektor	27
2.10 Matlab	28
2.11 Paint	28
2.12 Penelitian Terkait	29
III. METODOLOGI PENELITIAN	31
3.1 Waktu dan Tempat.....	31
3.2 Alat dan Bahan.....	32
3.3 Tahapan Penelitian.....	33
3.3.1 Analisis dan Perancangan Sistem	33
3.3.1.1 Pengambilan Citra Gambar Buah Kopi.....	35
3.3.1.2 Tahap <i>Preprocessing</i>	35
3.3.1.3 <i>Scaling</i>	36
3.3.1.4 <i>Contrast- Limited Adaptive Histogram Equalization</i> (<i>CLAHE</i>).....	36
3.3.1.5 <i>Grayscale</i>	36
3.3.1.6 <i>Thresholding</i>	37
3.3.1.6.1 <i>In Biner</i>	38
3.3.1.6.1 Perbaikan Garis Tepi Citra.....	39
3.3.1.7 Penggabungan <i>CLAHE</i> dan <i>Threshold</i>	39
3.3.2 Jaringan Syaraf Tiruan <i>Backpropagation</i>	39
3.3.2.1 Arsitektur <i>JST BackPropagation</i>	41

3.3.2.2 Fungsi Aktifasi	41
3.3.2.3 Pemilihan Bobot	42
3.3.2.4 Jumlah Unit Tersembunyi	42
3.3.2.5 Iterasi	42
3.3.2.6 Penentuan <i>Goal</i> , <i>Learning Rate</i> dan <i>Momentum</i>	43
3.3.3 Pembuatan Sistem.....	44
IV. DATA DAN PEMBAHASAN	59
4.1 Pengujian Sistem.....	59
4.1.1 Proses Pengolahan Data Awal	59
4.1.2 Pembagian Data	60
4.1.3 Model Arsitektur Jaringan	63
4.1.4 Pelatihan	67
4.1.5 <i>Preprocessing</i>	72
4.1.6 Pengujian	73
4.2 Analisis Data	75
V. KESIMPULAN DAN SARAN.....	79
5.1 Kesimpulan	79
5.2 Saran	80
DAFTAR PUSTAKA	
LAMPIRAN.....	

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 Struktur Anatomi Buah Kopi	8
Gambar 2.2 Alur Pengolahan Kopi Secara Kering (<i>Dry Process</i>).....	9
Gambar 2.3 Proses Pengolahan Buah Kopi Secara Basah (<i>Wet Process</i>)	10
Gambar 2.4 R,G dan B kode 24 bit.....	14
Gambar 2.5 Jaringan Syaraf Biologi dan JST	16
Gambar 2.6 JST <i>Multi layer</i>	17
Gambar 2.7 JST Neuron.....	18
Gambar 2.8 Arsitektur <i>Backpropagation</i>	20
Gambar 2.9.1a Fungsi Sigmoid Biner.....	21
Gambar 2.9.1b Fungsi Sigmoid Bipolar	22
Gambar 3.1 Tahapan Penelitian	33
Gambar 3.2 Perancangan Sistem Identifikasi Kematangan Buah Kopi...	34
Gambar 3.3 Tahapan <i>Preprocessing</i>	35
Gambar 3.4 Proses Perubahan Nilai Pixel Pada Proses <i>Thresholding</i>	38
Gambar 3.5 Pembuatan Sistem GUI.....	45
Gambar 3.6 GUI <i>push Botton</i> Untuk <i>Train</i>	46
Gambar 3.7 <i>Source Code</i> untuk <i>Training JST Backpropagation</i>	46
Gambar 3.8 GUI Axes1 dengan Perintah Pilih Gambar	47
Gambar 3.9 <i>Source Code</i> untuk Fungsi Pilih Gambar.....	47
Gambar 3.10 GUI Axes2 dengan Perintah Pertajam Warna.....	48
Gambar 3.11 <i>Source Code</i> untuk Fungsi Pertajam Warna	49
Gambar 3.12 GUI axes3 dengan Perintah <i>Grayscale</i>	50
Gambar 3.13 <i>Source Code</i> untuk Perintah <i>Grayscale</i>	50
Gambar 3.14 GUI axes4 dengan Perintah Citra Biner.....	51
Gambar 3.15 <i>Source Code</i> untuk Perintah Citra Biner	51
Gambar 3.16 GUI axes5 dengan Perintah <i>In Biner</i>	52
Gambar 3.17 <i>Source Code</i> untuk Perintah <i>In Biner</i>	52
Gambar 3.18 GUI axes6 dengan Perintah Penghalusan <i>Biner</i>	53

Gambar 3.19 <i>Source Code</i> untuk Perintah Penghalus <i>Biner</i>	54
Gambar 3.20 GUI Hasil Citra dan Hasil Pembacaan Nilai RGB.....	55
Gambar 3.21 <i>Source Code</i> untuk Perintah Perintah Hasil	56
Gambar 3.22 Tampilan Program Saat Dijalankan	57
Gambar 4.1 Arsitektur Jaringan yang dibangun pada Uji 1.....	63
Gambar 4.2 Arsitektur Jaringan yan dibangun pada Penelitian.....	65
Gambar 4.3 <i>Syntax</i> Program Pelatihan <i>Backpro</i> pada Matlab	68
Gambar 4.4 Tampilan Proses <i>Training</i>	70
Gambar 4.5 Grafik <i>Train</i> Pelatihan.....	71
Gambar 4.6 Proses <i>Preprocessing</i> Sampel Pelatihan	72
Gambar 4.7 <i>Syntax</i> Program Pengujian <i>Backpro</i> pada MATLAB	73
Gambar 4.8 Hasil Pengujian JST <i>Backpro</i>	74

DAFTAR TABEL

Tabel	Halaman
Tabel 3.1 Jadwal Kegiatan Penelitian.....	31
Tabel 3.2 Arsitektur Penelitian	44
Tabel 4.1 Data Pelatihan sebagai Input JST <i>Backpropagation</i>	60
Tabel 4.2 Target Data Pelatihan.....	61
Tabel 4.3 Hasil Pengujian Sampel Uji	75

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kopi merupakan salah satu komoditas perkebunan yang banyak diperdagangkan dan masuk dalam komoditas strategis di Indonesia. Provinsi Lampung merupakan salah satu daerah yang menghasilkan produksi kopi yang cukup tinggi. Berdasarkan data dari Badan Pusat Statistik Perkebunan Indonesia (BPSPI) tahun 2015 – 2017, luas area dan produksi kopi tahun 2015 Lampung memiliki luas area perkebunan kopi sebesar 161.693 Ha dengan produksi mencapai 110.318 Ton, tahun 2016 dengan luas area 161.416 Ha dengan hasil produksi 110.354 Ton, dan pada tahun 2017 dengan luas area 162.020 Ha dengan produksi mencapai 110.368 Ton [1]. Berdasarkan data tersebut menunjukkan bahwa Provinsi Lampung memiliki potensi untuk melakukan pengembangan agroindustri kopi dengan semakin meningkatnya produksi kopi yang dihasilkan. Pengolahan kopi harus dilakukan dengan tata cara yang tepat agar menghasilkan kopi dengan kualitas dan mutu yang baik.

Proses pengolahan buah kopi menjadi biji kopi yang dilakukan oleh industri maupun petani dibagi menjadi dua tahapan yaitu pengolahan dengan proses basah dan pengolahan dengan proses kering. Pengolahan kopi dengan proses basah dimulai dari panen, sortasi buah kopi, pengupasan kulit buah merah, fermentasi, pencucian, pengeringan, pengupasan kulit tanduk dan kulit ari, sortasi biji kopi

pengemasan, dan penyimpanan. Sedangkan pengolahan kopi dengan proses kering terdiri dari panen, sortasi buah kopi, pengeringan, pengupasan, sortasi biji kopi, pengemasan, dan penyimpanan. Produk yang dihasilkan dari kopi tidak hanya sebagai kopi bubuk biasa tetapi juga dapat diturunkan kembali menjadi kopi instan bebas kafein, kopi dengan percampuran teh, es kopi instant, kopi putih, kopi dengan rasa buah – buahan dan masih banyak lagi.

Semakin meningkatnya permintaan pasar mengenai kopi menyebabkan banyak terjadi keterhambatan ketersediaan kopi yang diminta karena petani masih menggunakan peralatan manual serta dengan cara yang tradisional dalam proses sortasi buah kopi. Sedangkan pada tahap proses pengolahan kopi sortasi buah kopi sangat penting pada saat buah kopi dipanen. Buah kopi yang telah disortasi tidak dapat disimpan terlalu lama karena apabila penundaan pengolahan bisa memicu reaksi kimia yang akan menurunkan mutu kopi itu sendiri. Pemisahan buah dilakukan untuk membedakan kualitas biji kopi yang akan dihasilkan nanti untuk memilih buah yang superior (masak dan seragam) dari buah inferior (cacat, hitam, pecah, berlubang dan terserang hama/penyakit)

Tingkat kematangan buah kopi dapat dilihat dari warna kulitnya, apabila buah kopi telah berwarna merah berarti menandakan bahwa buah kopi tersebut telah matang penuh. Namun karena berbagai alasan, sortasi buah kopi secara manual sangat dipengaruhi oleh subjektivitas petani kopi sehingga menyebabkan tidak spesifiknya proses klasifikasi buah kopi. Hal tersebut menyebabkan rendahnya mutu kopi bubuk hasil olahan petani salah satunya disebabkan oleh proses sortasi buah kopi yang masih menggunakan cara manual seperti melakukan sortasi secara langsung oleh manusia, sehingga menyebabkan semakin banyaknya waktu dan sumber daya manusi yang dibutuhkan dalam proses sortasi buah kopi tersebut. Pada dasarnya terdapat tingkat kematangan buah kopi yaitu apabila buah kopi berwarna hijau dan hijau kekuningan menandakan bahwa kondisi

buah kopi masih muda dan tidak disarankan untuk dipetik, warna kuning kemerahan menunjukkan buah kopi sudah mulai matang namun belum memiliki aroma dan postur yang baik, warna merah penuh, menunjukkan buah telah matang sempurna atau memiliki aroma yang baik, dan yang terakhir warna merah tua atau kehitaman hitaman menandakan buah kopi tersebut sudah terlalu matang serta aroma dan posturnya yang mulai menurun [2].

Melihat pentingnya proses sortasi pada buah kopi untuk menghasilkan kopi yang bermutu baik yang berasal pada buah kopi yang sehat dan berwarna merah maka perlu dibangun sebuah sistem. Dengan perkembangan teknologi informasi saat ini memungkinkan untuk melakukan identifikasi buah berdasarkan ciri warna dengan bantuan komputer [3]. Pada cara komputasi tersebut dilakukan dengan pengamatan visual tidak langsung, dengan menggunakan kamera sebagai pengolahan citra dari gambar yang di foto (*image processing*) yang akan diolah menggunakan perangkat lunak komputer [4].

Berdasarkan uraian diatas maka dibangun suatu sistem identifikasi kematangan buah kopi dengan pengolahan citra serta pada proses indentifikasi buah kopi menggunakan Jaringan Syaraf Tiruan (JST) dengan metode pembelajaran *backpropagation*. Penggunaan pengolahan citra sendiri digunakan untuk melakukan proses *pre-processing* sebelum memasuki tahap identifikasi gambar. JST *backpropagation* merupakan topologi yang cukup populer dan banyak dipakai untuk berbagai aplikasi terutama dalam pengenalan pola [5]. JST *backpropagation* merupakan pelatihan jaringan untuk mengetahui kemampuan jaringan dalam mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan dalam melakukan respon yang benar terhadap pola masukan yang serupa dengan pola yang akan dipakai selama proses pelatihan [6]. Sistem ini dibangun dengan menggunakan masukan gambar buah kopi kemudian dilakukan pengolahan citra untuk menerangkan kontras warna gambar

kemudian dilakukan pembuangan background yang kemudian gambar akan di ambil nilai R, G dan B nya sebagai inputan dalam proses klasifikasi *backpropagation* yang nantinya akan mendapatkan 3 keluaran dari masing-masing warna berupa nilai biner 1/0.

1.2. Tujuan Penelitian

Tujuan dari penelitian ini sebagai berikut :

1. Membangun perangkat lunak berbasis GUI Matlab dengan metode JST *backpropagation* sebagai studi penelitian.
2. Menguji tingkat keberhasilan identifikasi kematangan buah kopi menggunakan metode JST *backpropagation* dengan 3 buah klasifikasi yaitu matang, tidak matang dan terlalu matang.

1.3. Rumusan Masalah

Rumusan masalah yang mendasari penelitian ini sebagai berikut :

1. Bagaimana membangun sebuah sistem identifikasi kematangan buah kopi menggunakan metode JST *backpropagation* dengan MATLAB?
2. Apakah sistem yang dibangun sudah dapat melakukan pengenalan pola pada saat pengujian dengan metode *backpropagation* sesuai dengan kriteria yang diharapkan?

1.4. Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut :

1. Pada penelitian ini hanya membahas mengenai metode JST *backpropagation* dan tidak membahas mengenai pengolahan citra digital selain pada proses *preprocessing*.

2. Menggunakan gambar buah kopi sebagai sample dengan ukuran 500 x 500 pixel dengan berbagai warna yang tersebar pada setiap sample.
3. Tidak membahas pemrograman matlab secara mendalam.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini sebagai berikut :

1. Penulis dapat lebih memahami dan mengaplikasikan ilmu kecerdasan buatan bidang JST metode *backpropagation*.
2. Menjadi penelitian awal untuk mendapatkan sebuah sistem penentuan kematangan buah kopi yang handal.

1.6. Sistematika Penulisan

Pada penulisan laporan ini untuk mempermudah penulisan dan pemahaman mengenai materi maka tulisan ini dibagi menjadi lima bab, yaitu :

BAB I Pendahuluan

Pada bab ini diuraikan hal-hal yang berhubungan dengan latar belakang, tujuan penelitian, rumusan masalah, batasan masalah, metode penelitian, dan sistematika penulisan.

BAB II Tinjauan pustaka

Bab ini merupakan landasan teori dari penyusunan penulisan yaitu prinsip, pengetahuan, rumus, dan teori penunjang tentang kopi, pengolahan citra, JST *backpropagation*, MATLAB dan lain-lain.

BAB III Metode penelitian

Bab ini berisi mengenai waktu penelitian, alat dan bahan penelitian , tahapan penelitian secara terperinci.

BAB IV Hasil dan PEMBAHASAN

Pada bagian ini memaparkan mengenai perancangan sistem menggunakan *backpropagation*, data hasil penelitian dan pembahasan dari data-data yang didapatkan saat pengujian.

BAB V Simpulan dan Saran

Berisikan kesimpulan dari hasil penelitian dan saran-saran mengenai perbaikan dan pengembangan penelitian ini .

Daftar pustaka**Lampiran**

BAB II

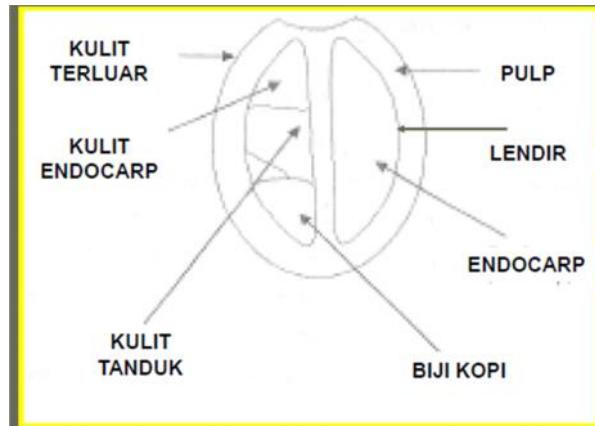
TINJAUAN PUSTAKA

2.1. Kopi

Kopi merupakan tanaman yang termasuk dalam famili *Rubiaceae*. Pada dasarnya kopi memiliki beberapa golongan, namun yang paling sering dibudidayakan yaitu *Coffea Arabica*, *Coffea Robusta*, dan *Coffea Liberica*. Tanaman kopi berasal dari Abessinia yang tumbuh di dataran tinggi dengan nama lain kopi yaitu *Coffea sp.* Berikut ini merupakan klasifikasi sebagai berikut [7] :

Regnum	: Plantae
Divisio	: Magnoliophyta
Sub Divisio	: Spermatophyta
Class	: Magnoliopsida
Sub Class	: Asteridae
Order	: Rubiales
Family	: Rubiaceae
Genus	: Coffea

Tanaman kopi pada umumnya berbunga setelah berumur sekitar dua tahun. Pada buah kopi terdiri dari daging buah dan biji. Pada bagian daging biji buah kopi terdiri dari 3 bagian yaitu lapisan kulit luar (*exocarp*), lapisan daging buah (*mesocarp*), dan lapisan kulit tanduk (*endoscarp*). Berikut ini merupakan gambaran struktur anatomi buah kopi yaitu :



Gambar 2.1. Struktur Anatomi Buah Kopi [7].

Pada umumnya buah kopi yang masih muda berwarna hijau, tetapi saat mulai tua menjadi kuning dan apabila telah masak warnanya menjadi merah. Kulit buah kopi sangatlah tipis dan mengandung klorofil serta zat-zat warna lainnya. Pada daging buah terdiri dari 2 bagian yaitu bagian luar yang lebih tebal dan keras dan pada bagian dalamnya yang bersifat seperti gel atau lendir. Pada lapisan lendir ini terdapat 85% air dalam bentuk terikat dan 15%nya merupakan bahan koloid yang tidak mengandung air. Pada bagian ini bersifat koloid hidrofilik yang terdiri dari $\pm 80\%$ pectin dan $\pm 20\%$ gula. Pada biji kopi kering mempunyai komposisi yaitu 12% air, 13% protein, 12% lemak, 9% gula, 1-1,5% *caffeine* (arabika), 2-2,5% (robusta), 9% *caffetanic acid cellulose* dan sejenisnya 35%, 4% abu, zat-zat lainnya yang larut dalam air 5% [8].

Pada umumnya buah kopi memiliki dua butir biji, namun pada kenyataannya terdapat pula buah kopi yang tidak menghasilkan biji atau hanya menghasilkan satu butir biji. Pada buah kopi biji kopi memiliki bagian-bagian penting yang terdiri dari kulit biji dan lembaga. Secara morfologi biji kopi berbentuk bulat atau oval seperti telur, memiliki warna putih kotor dan bertekstur keras [9].

2.2. Proses Pengolahan Biji Kopi

Kopi yang telah dipetik sebaiknya segera diolah lebih lanjut dan tidak dianjurkan untuk dibiarkan begitu saja selama kurun waktu lebih dari 12-20 jam. Apabila buah kopi tidak segera diolah dalam jangka waktu tertentu maka kopi akan mengalami fermentasi dan proses kimia lainnya yang dapat mengakibatkan penurunan mutu kopi tersebut. Namun bila kopi benar-benar belum dapat diolah, maka kopi harus direndam terlebih dahulu dalam air bersih yang mengalir [10]. Proses pengolahan kopi pada dasarnya terbagi menjadi dua yaitu proses olahan kering (*dry process*) dan proses olahan basah (*wet process*) [11].

2.2.1. Pengolahan Cara Kering

Pada metode pengolahan kopi dengan cara kering banyak digunakan untuk pengolahan ditingkat petani dengan lahan yang tidak luas atau dengan kapasitas olahan yang kecil. Pada perkebunan yang memiliki lahan lebih besar pengolahan kopi dengan cara kering hanya khusus untuk kopi dengan buah yang berwarna hijau, kopi yang mengambang dan kopi yang terserang penyakit. Berikut ini merupakan tahapan pengolahan kopi dengan cara kering [11] :



Gambar 2.2 Alur Pengolahan Kopi Secara Kering (*Dry Process*).

2.2.2. Pengolahan Cara Basah

Pengolahan kopi dengan cara basah yaitu dengan cara buah kopi yang telah dipetik selanjutnya dimasukkan kedalam *pulper* untuk melepaskan kulitnya kemudian dibiarkan ke bak dan direndam selama beberapa hari untuk fermentasi. Kemudian setelah direndam buah kopi tersebut di cuci bersih dan dikeringkan. Pengeringan buah kopi dilakukan dengan dijemur dibawah sinar matahari atau menggunakan mesin pengering. Berikut ini merupakan tahapan pengolahan buah kopi dengan cara basah [11] :



Gambar 2.3. Proses Pengolahan Buah Kopi Secara Basah
(*Wet Process*).

2.3. Definisi Citra

Pada dasarnya pengertian citra secara visual yaitu representasi dari informasi yang terkandung di dalamnya sehingga mata manusia dapat menganalisis dan menginterpretasikan informasi tersebut sesuai dengan

tujuan yang diharapkan. Informasi yang terkandung pada sebuah citra dapat dibedakan menjadi dua bagian yaitu informasi dasar dan informasi yang bersifat abstrak. Pengertian informasi dasar yaitu informasi yang dapat diolah secara langsung tanpa membutuhkan bantuan tambahan pengetahuan yang khusus, seperti warna, bentuk dan tekstur. Sedangkan informasi abstrak yaitu informasi yang tidak secara langsung dapat diolah kecuali dengan bantuan tambahan pengetahuan khusus seperti ekspresi wajah di dalam sebuah citra yang dapat menggambarkan suasana hati seseorang (sedih, senang, marah, Murung dan lainnya). Pengertian citra secara sistematis didefinisikan sebagai fungsi dua dimensi $f(x,y)$, di mana x dan y merupakan koordinat spasial (*plane*) dan f merupakan nilai intensitas warna pada koordinat x dan y [14].

2.4. Pengolahan Citra Digital

Pengolahan citra digital merupakan suatu pemrosesan citra atau *image processing* dengan kegiatan memperbaiki kualitas citra agar mudah dikenali oleh manusia atau komputer sesuai dengan tujuan yang diharapkan [14]. Pengolahan citra digital (*Digital Image Processing*) merupakan disiplin ilmu yang mempelajari mengenai teknik-teknik mengolah citra (foto) maupun gambar bergerak (berasal dari *webcam*) dengan pengolahan citra/gambar dilakukan secara digital menggunakan komputer [15]. Pengolahan citra digital merupakan ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas citra, dengan kata lain pengolahan citra merupakan kegiatan memperbaiki kualitas citra agar dapat dengan mudah diterjemahkan oleh manusia atau mesin (komputer) [16].

Citra digital umumnya dua dimensi (2D) yang dinyatakan dengan matriks dengan jumlah elemen yang berhingga. Fungsi dua dimensi $f(x,y)$ di mana x dan y adalah koordinat spasial (*plane*) dan f adalah nilai intensitas warna

pada koordinat x dan y . Nilai f , x dan y semuanya merupakan nilai berhingga. Citra digital merupakan representasi piksel-piksel dalam ruang 2D yang dinyatakan dalam matriks berukuran N baris dan M kolom, dengan elemen matriks citra disebut piksel (*picture elemen, image elemen*) atau elemen terkecil dari sebuah citra [14].

2.4.1. Penskalaan

Penskalaan atau *Scaling* merupakan sebuah operasi geometri memberika efek memperbesar (*zoomin*) atau memperkecil (*zoomout*) sesuai dengan variabel penskalaan citranya. Proses penskalaan dilakukan dengan rumus :

$$P_o = S_p \times P_i \dots\dots\dots [1]$$

$$L_o = S_l \times L_i \dots\dots\dots [2]$$

Pada rumus tersebut (P_i , L_i) merupakan ukuran citra input, sedangkan (P_o, L_o) merupakan ukuran citra output, dan (S_p , S_l) merupakan variabel *scaling* yang diinginkan. Apabila variabel *scaling* bernilai lebih besar dari 1 maka hasil *scaling* akan memperbesar ukuran citra, namun apabila variabel *scaling* lebih kecil dari 1 maka hasilnya akan memperkecil ukuran citra [12].

2.4.2. Representasi Warna

Pada dasarnya mata manusia memiliki persepsi warna yang merupakan proses subjektif di mana syaraf visual otak merespon ketika cahaya masuk. Persepsi warna pada pengolahan citra tergantung pada tiga faktor, yaitu *spectral reflectance* (menentukan bagaimana suatu permukaan memantulkan warna), *spectral content* (kandungan warna dari cahaya yang menyinari permukaan) dan

spectral response (kemampuan merespon warna dari sensor dalam *imaging system*) [17].

Representasi warna terdiri dari tiga unsur warna utama yaitu merah (*red*), hijau (*green*) dan biru (*blue*) atau disingkat dengan warna RGB. Gabungan dari ketiga warna tersebut membentuk warna-warna lainnya berdasarkan intensitas dari setiap warna tersebut dengan intensitas maksimal dan warna hitam merupakan gabungan dari ketiga warna tersebut dengan intensitas minimal. Representasi warna dikembangkan dengan tujuan untuk memodelkan, menghitung dan memvisualisasikan informasi warna sehingga dapat dengan mudah komputer atau sistem digital lainnya untuk memproses informasi warna dan membedakan warna seperti halnya sistem visual manusia.

2.4.3. Format File Citra JPEG (.jpg)

Pada pengolahan citra terdapat format file citra standar yang digunakan yang terdiri dari beberapa jenis. Format file citra digunakan untuk menyimpan citra dalam sebuah file. Format yang digunakan pada penelitian ini JPEG atau .jpg merupakan salah satu format file citra yang banyak digunakan saat ini terutama untuk melakukan transmisi citra. Format .jpg digunakan sebagai tempat penyimpanan citra hasil kompresi dengan metode JPEG [12].

2.4.4. Ruang Warna RGB

Ruang warna RGB adalah ruang warna standar yang didasarkan pada hasil akuisisi frekuensi warna oleh sensor elektronik. Bentuk keluaran dari sensor ini adalah berupa sinyal analog, yang

kemudian intensitas amplitudonya di digitalisasi dan dikodekan dalam 8 bit untuk setiap warnanya. Setiap warna yang tampak merupakan kombinasi dari tiga komponen red, green dan blue. Gabungan tiga warna ini akan menggabungkan warna lain. Selain itu, R, G dan B memiliki intensitas yang sama, setiap titik atau pixel pada citra berwarna memiliki tiga komponen warna red, green dan blue yang masing-masing umumnya dikodekan dengan 8 bit atau total ketiganya adalah $3 \times 8 = 24$ bit [14].



Gambar 2.4. R, G dan B kode 24 bit [16].

Pada perhitungan nilai R, G dan B sendiri secara matematis dapat dihitung dengan rumus R :

$$R, G, \text{ dan } B = (R * 65536) + (G * 256) + B \dots\dots\dots [3]$$

Keterangan :

R = Pembulatan kebawah $(RGB / 65536 / 256 * 256)$

G = Pembulatan kebawah $(RGB / 256)$ Modulus

B = RGB Modulus 256.

Modulus merupakan sisa pembagian dari suatu bilangan.

2.4.5. Pemerataan Kontras

Penyamaan histogram adaptif merupakan pemerataan histogram adaptif dengan kontras terbatas atau *Contrast-Limited Adaptive Histogram Equalization* (CLAHE) dengan menggunakan fungsi *adapthisteq*. Histogram sendiri merupakan grafik yang mengindikasikan jumlah kemunculan setiap level keabuan pada suatu citra [18]. *Adapthisteq* beroperasi pada daerah kecil pada

gambar yang disebut ubin, adapthisteq berguna sebagai peningkatan kontras setiap ubin sehingga histogram dari wilayah output menjadi lebih terang dan jelas [19].

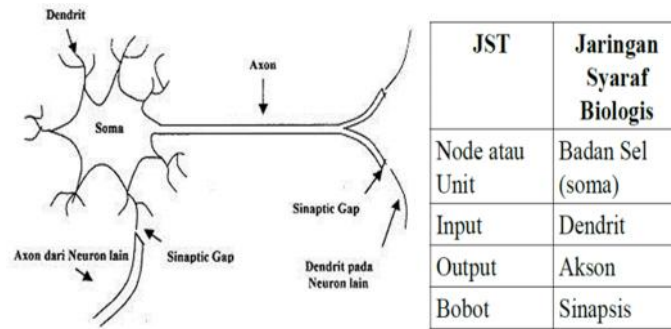
2.5. Kecerdasan Buatan

Kecerdasan buatan atau *Artificial Intelligence* (AI) merupakan cabang dari ilmu komputer yang mempelajari bagaimana membuat mesin/komputer yang dapat melakukan pekerjaan-pekerjaan yang memerlukan kecerdasan atau dapat melakukan pekerjaan layaknya manusia yang cerdas. Prinsip-prinsip ini meliputi struktur data yang digunakan dalam representasi pengetahuan, algoritma yang diperlukan untuk mengaplikasikan pengetahuan tersebut, serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya. Adapun beberapa bidang dari kecerdasan buatan yaitu jaringan syaraf tiruan, algoritma genetika, sistem pakar, komputer vision dan lain-lain.

2.6. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) merupakan sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi sehingga menjadikan JST cocok untuk menyelesaikan masalah dengan tipe yang menyerupai otak manusia. Otak manusia sendiri terdiri dari neuron-neuron dan penghubung yang disebut sinapsis. Neuron bekerja berdasarkan implus/sinyal yang diberikan pada neuron. Neuron meneruskannya pada neuron lain. Neuron memiliki 3 komponen penting yaitu dendrit, soma dan axon. Dendrit menerima sinyal dari neuron lain, sinyal tersebut berupa impuls elektrik yang dikirim melalui celah sinaptik melalui proses kimiawi. Sinyal tersebut dimodifikasi

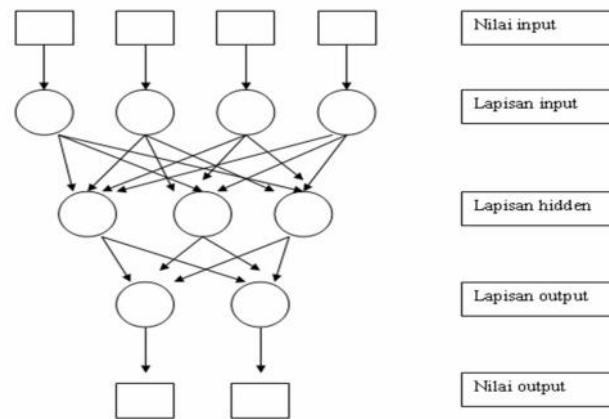
(diperkuat/diperlemah) di celah sinaptik. Soma menjumlahkan semua sinyal-sinyal yang masuk. Jika jumlah tersebut cukup kuat dan melebihi batas ambang (*threshold*), maka sinyal tersebut akan diteruskan ke sel lain melalui axon. Frekuensi penerusan sinyal berbeda-beda antara satu sel dengan yang lain [6].



Gambar 2.5. Jaringan Syaraf Biologi dan JST

2.6.1. Arsitektur Jaringan

Beberapa arsitektur jaringan yang sering dipakai dalam jaringan syaraf tiruan yaitu jaringan layar tunggal, jaringan layar jamak dan jaringan kompetitive. Pada JST *backpropagation* menggunakan arsitektur jaringan jamak (*multi layer*). Jaringan *multi layer* merupakan perluasan dari layar tunggal. Dalam jaringan ini, selain unit input dan output, ada unit-unit lain (sering disebut layar tersembunyi atau *hidden layer*). Seperti pada unit input dan output, unit-unit dalam satu layar tidak saling berhubungan, memiliki satu atau lebih lapisan input, satu atau lebih lapisan output dan lapisan tersembunyi. Dapat menyelesaikan masalah yang lebih kompleks karena lebih akurat dan fungsi pembelajaran lebih rumit.



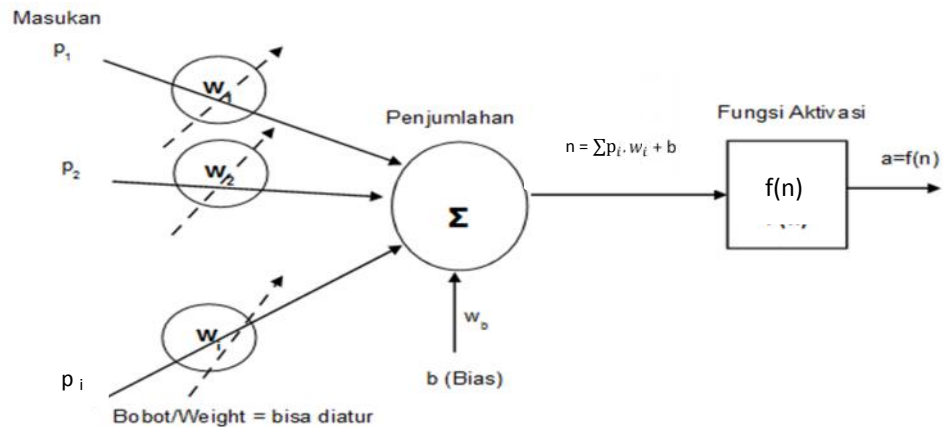
Gambar 2.6. JST *Multi Layer* [6].

2.6.2. Algoritma Pembelajaran JST

JST pada dasarnya memiliki dua macam pelatihan yang dilakukan yaitu pelatihan supervisi (*supervised*) dan tanpa supervisi (*unsupervised*). Pelatihan *supervised* yaitu pelatihan yang dilakukan dengan melatih jaringan untuk mendapatkan bobot yang diinginkan sesuai dengan data masukan dan target yang digunakan sebagai acuan untuk menentukan pola jaringan. Sedangkan pelatihan *unsupervised* merupakan pelatihan dengan pengelompokkan input yang serupa dan tidak menyediakan target. Pada pelatihannya perubahan bobot jaringan dilakukan berdasarkan parameter tertentu dan jaringan dimodifikasi sesuai dengan parameter tersebut [6].

2.6.3. Fungsi Aktifasi, Bias dan Threshold

JST ditentukan dengan pola hubungan antar neuron (disebut arsitektur jaringan), metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*) dan fungsi aktifasi. Berikut ini merupakan contoh neuron [6] :



Gambar 2.7. JST Neuron.

Fungsi aktivasi digunakan untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya). Jika $net = \sum p_i \cdot w_i$ maka fungsi aktivasinya adalah $f(net) = f(\sum p_i \cdot w_i)$. Pada JST ditambahkan unit masukan yang nilainya selalu = 1 (bias). Bias dapat dipandang sebagai sebuah input yang nilainya = 1. Bias berfungsi untuk mengubah nilai *threshold* menjadi = 0.

$$net = b + \sum p_i \cdot w_i \dots\dots\dots [4]$$

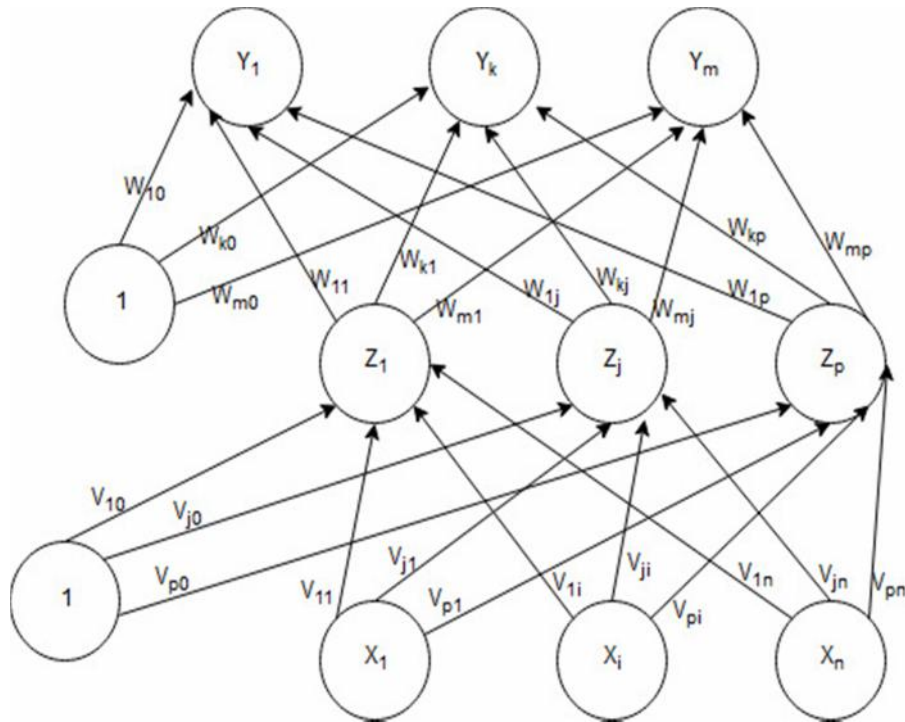
Neuron – neuron pada JST akan dikumpulkan dalam lapisan yang disebut dengan *layers*. Neuron dalam satu lapisan akan dihubungkan dengan neuron pada lapisan lainnya, terkadang muncul juga *layer* tersembunyi (*hidden layer*) untuk menambahkan keakuratan pelatihan. Neuron terdiri dari 3 elemen pembentuk yaitu himpunan unit-unit yang dihubungkan dengan jalur koneksi dan memiliki bobot yang berbeda-beda, unit penjumlahan yang menjumlahkan input-input sinyal yang sudah dikalikan dengan bobotnya dan yang terakhir yaitu fungsi aktifasinya yang akan menentukan apakah sinyal dari input neuron akan diteruskan ke neuron lain atau tidak.

Kelebihan JST yaitu mampu mengakuisisi pengetahuan meskipun nilai tidak pasti (*fault tolerance*), gangguan dapat dianggap sebagai

noise (derau/gangguan) saja. Kemudian mampu melakukan generalisasi (kesimpulan secara umum pada sebuah kejadian) dan ekstraksi (pemisahan) dari suatu pola data tertentu, JST dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar (*self organizing*). Lalu kemampuan perhitungan secara paralel sehingga proses lebih singkat dan mampu bekerja secara paralel layaknya otak manusia bekerja. Kelemahan JST yaitu kurang mampu untuk melakukan operasi-operasi numerik dengan *presisi* (keadaan pasti) tinggi, keakuratan yang tinggi sulit diperoleh, bekerja berdasarkan pola yang terbentuk pada inputnya, dan lamanya proses *training* yang mungkin terjadi dalam waktu yang sangat lama untuk jumlah data yang besar [6].

2.7. **Arsitektur *Backpropagation***

Backpropagation merupakan metode dengan melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan responnya yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan. *Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Berikut ini merupakan gambar arsitektur *backpropagation* [6] :



Gambar 2.8. Arsitektur *Backpropagation* [6].

Pada gambar 2.8. merupakan arsitektur *backpropagation* dengan n unit input (X_1, X_2, X_n), p buah layer tersembunyi (Z_1, Z_j, Z_p), dan m buah output (Y_1, Y_k, Y_m). Dengan urutan pembelajaran pada JST *backpropagation* pertama melakukan inisialisasi bobot, bias dan penetapan *epoch* (satu siklus pelatihan yang melibatkan semua pola), laju pembelajaran (*learning rate*) dan batas toleransi *error*.

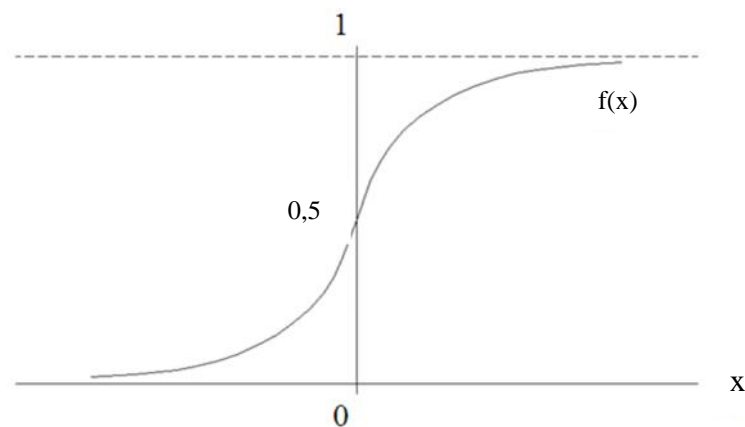
Selanjutnya dilakukan perhitungan keluaran pada tiap unit tersembunyi (Z), kemudian hitung keluaran pada unit keluaran (Y) tahapan ini disebut propagasi maju. Tahapan kedua yaitu menghitung *error* pada unit keluaran berdasarkan kesalahan pada unit keluaran (Y), selanjutnya hitung *error* pada unit tersembunyi (Z) tahapan ini disebut propagasi mundur. Terakhir yaitu tahapan perubahan bobot yaitu dengan menghitung semua perubahan bobot. Proses ini dilakukan secara berurutan dan proses ini akan dilakukan secara berurutan dan berulang kali hingga syarat berhenti terpenuhi.

2.7.1. Fungsi Aktivasi *Backpropagation*

Pada *backpropagation* terdapat fungsi aktivasi yang digunakan harus mengikuti beberapa syarat yaitu kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memiliki ketiga syarat tersebut sehingga sering dipakai adalah fungsi sigmoid biner yang memiliki range (0,1) [6].

$$f(x) = \frac{1}{1+e^{-x}} \text{ dengan turunan } f'(x) = f(x)(1-f(x)) \dots\dots\dots[5]$$

Grafik fungsinya tampak pada gambar di bawah ini :

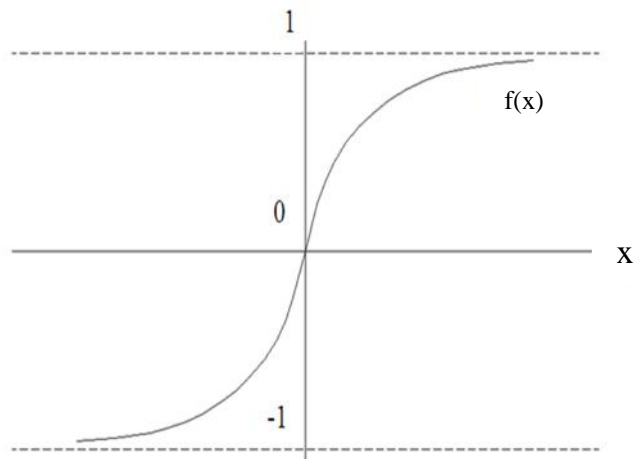


Gambar 2.9.1a. Fungsi Sigmoid Biner [6].

Selain fungsi sigmoid biner terdapat pula fungsi sigmoid bipolar yang memiliki bentuk fungsinya mirip dengan fungsi sigmoid biner, tetapi berbeda range (-1, 1).

$$f(x) = \frac{2}{1+e^{-x}} - 1 \text{ dengan turunan } f'(x) = \frac{(1+f(x))(1-f(x))}{2} \dots [6]$$

Grafik fungsinya tampak pada gambar di bawah ini :



Gambar 2.9.1b. Fungsi Sigmoid Bipolar [6].

2.7.2. Pelatihan *Backpropagation*

Backpropagation memiliki pelatihan dengan 3 fase. Fase pertama yaitu fase maju dengan pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua yaitu fase mundur, selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasi mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga yaitu modifikasi bobot untuk menurunkan kesalahan yang terjadi [6].

- **Fase I : Propagasi maju**

Pada propagasi maju, sinyal masukan ($= x_i$) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi ($= z_j$) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan

keluaran jaringan ($= y_k$). Berikutnya, keluaran jaringan ($= y_k$) dibandingkan dengan target yang harus dicapai ($= t_k$). Selisih $t_k - y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi [6].

- **Fase II : Propagasi mundur**

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor δ_k ($k = 1, 2, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit y_k . δ_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor δ_j di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor δ_i di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung [6].

- **Fase III : Perubahan bobot**

Setelah semua faktor δ_i dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ_i neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δ_k yang ada di unit keluaran [6].

Pada ketiga fase tersebut dilakukan perulangan secara terus menerus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang

ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diizinkan. Berikut ini merupakan tahapan algoritma pelatihan untuk jaringan dengan satu layer tersembunyi (fungsi aktivasi sigmoid biner) adalah sebagai berikut:

Langkah 0 : Memberikan nilai semua bobot dengan bilangan acak kecil.

Langkah 1 : Lakukan langkah 2 - 9.

Langkah 2 : Pada data pelatihan, lakukan langkah 3– 8.

Fase I : Propagasi maju

Langkah 3 : Tiap unit masukkan menerima sinyal dan meneruskan ke unit tersembunyi di atasnya.

Langkah 4 : Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$) :

$$z_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \dots\dots\dots [7].$$

$$z_j = f(z_net_j) = \frac{1}{1+e^{-z_net_j}} \dots\dots\dots [8].$$

Langkah 5 : Hitung keluaran pada jaringan di unit y_k ($k=1,2, \dots, m$) :

$$y_net_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \dots\dots\dots [9].$$

$$y_k = f(y_net_k) = \frac{1}{1+e^{-y_net_k}} \dots\dots\dots [10].$$

Fase II : Propagasi mundur

Langkah 6 : Hitung faktor unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1,2, \dots, m$)

$$\delta_k = (t_k - y_k) f'(y_net_k) = (t_k - y_k) y_k (1 - y_k) \dots\dots\dots [11].$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer di bawahnya (langkah 7). Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan .

$$\Delta w_{kj} = \delta_k z_j \quad ; \quad k = 1,2, \dots, m ; \quad j = 0,1, \dots, p$$

Langkah 7 : Hitung faktor unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$_{net_j} = \sum_{k=1}^m \delta_k w_{kj} \dots\dots\dots [12].$$

Faktor unit tersembunyi :

$$_{net_j} = \delta_j f'(z_{net_j}) = \delta_j z_j (1 - z_j) \dots\dots\dots [13].$$

Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk merubah bobot v_{ji})

$$\Delta v_{ji} = \delta_j x_i \quad ; \quad j = 1, 2, \dots, p ; \quad i = 0, 1, \dots, n$$

Fase III : Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \delta_j \dots\dots\dots [14].$$

Perubahan bobot garis yang menuju ke unit tersembunyi

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \delta_j \dots\dots\dots [15].$$

Kemudian jika telah selesai pelatihan maka jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan. Apabila fungsi aktivasi yang dipakai bukan simoid biner, maka langkah 4 dan 5 harus disesuaikan. Demikian juga turunannya pada langkah 6 dan 7 [6].

2.8. Optimalitas Arsitektur *Backpropagation*

Pada *backpropagation* masalah utama yang sering dihadapi yaitu lamanya iterasi yang harus dilakukan. *Backpropagation* tidak dapat memberikan kepastian tentang berapa *epoch* yang harus dilalui untuk mencapai kondisi yang diinginkan [6].

2.8.1. Pemilihan Bobot dan Bias Awal

Pemilihan bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global. Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil. Sebab itu standar *backpropagation*, bobot dan bias diisi dengan bilangan acak kecil [6].

2.8.2. Jumlah Pola Pelatihan

Backpropagation tidak dapat memastikan mengenai kepastian berapa banyak pola yang diperlukan untuk jaringan agar dapat dilatih dengan sempurna. Jumlah pola yang dibutuhkan dipengaruhi oleh banyaknya bobot dalam jaringan serta tingkat akurasi yang diharapkan.

2.8.3. Lama Iterasi

Pada *backpropagation* tujuan utama penggunaan yaitu mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (data pengujian). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Pada umumnya data dibagi menjadi 2 bagian saling asing, yaitu pola data yang dipakai sebagai pelatihan dan data yang dipakai untuk pengujian. Perubahan bobot dilakukan

berdasarkan pola pelatihan. Akan tetapi selama pelatihan, kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan akan terus dijalankan. Akan tetapi jika kesalahannya meningkat maka pelatihan tidak ada gunanya untuk diteruskan lagi [6].

2.8.4. Momentum

Pada metode *backpropagation* momentum digunakan untuk mengurangi perubahan bobot yang mencolok atau data yang sangat berbeda dengan yang lainnya. Momentum merupakan perubahan bobot yang didasarkan pada gradien pada pola terakhir dan pola sebelumnya atau pola terakhir saja yang digunakan [6].

2.9. *Backpropagation* dengan MATLAB

Matlab dibuat untuk memudahkan perhitungan tersebut. Matlab juga menyediakan fungsi-fungsi khusus untuk menyelesaikan model jaringan syaraf tiruan metode *backpropagation*.

2.9.1. Matriks dan Vektor

Pada proses JST banyak berhubungan dengan matriks dan vektor untuk itu perlu memahami mengenai matriks dan vektor. Matriks merupakan kumpulan bilangan-bilangan yang disusun dalam larik baris dan kolom. Umumnya matriks diberi notasi huruf kapital A,B,...Matriks A yang terdiri dari m baris dan n kolom (sering disebut dengan ordo $m \times n$), maka akan ditulis sebagai :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ a_{m1} & a_{m2} & a_{mn} \end{bmatrix}$$

Matriks yang terdiri dari 1 kolom sama dengan suatu vektor. Vektor merupakan ruang dimensi n dengan bilangan-bilangan riil. Notasinya dengan menggunakan huruf-huruf kecil seperti x, y, z. Pada vektor tidaklah penting dinyatakan dalam sebuah baris ataupun sebuah kolom.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_n \end{bmatrix} \quad \text{dengan } x_1, x_2, \dots, x_n \text{ adalah bilangan – bilangan riil.}$$

2.10. Matlab

Matlab merupakan salah satu alat yang digunakan untuk komputasi matrik. Matlab adalah salah satu perangkat lunak yang melibatkan penggunaan matriks dan vektor. Fungsi-fungsi pada toolbox dibangun untuk memudahkan pengguna dalam perhitungan tersebut [6]. Pada fungsi matlab terdapat model Jaringan Syaraf Tiruan (JST) yang menggunakan manipulasi matriks atau vektor dalam iterasinya. Maka matlab dianggap perangkat lunak yang cocok digunakan sebagai perancangan sistem ini karena menggunakan metode JST *Backpropagation*.

2.11. Paint

Microsoft Paint merupakan editor grafis sederhana yang disediakan oleh microsoft yang dimiliki semua versi pada Microsoft Windows [13]. Paint merupakan program yang berfungsi untuk membuka, mengedit dan menyimpan file dalam berbagai macam format file seperti bitmap, JPEG,

GIF dan lainnya. Pada penelitian ini *paint* digunakan sebagai *resize* gambar dengan ukuran sama yaitu 500x500 *pixel*.

2.12. Penelitian Terkait

Identifikasi Kematangan Buah Tomat Menggunakan Metode *Backpropagation* oleh Desa Deswari dkk, 2013. Penelitian ini bertujuan mengidentifikasi tingkat kematangan buah tomat berdasarkan warna dengan jaringan syaraf tiruan *backpropagation* untuk menghasilkan nilai bobot. Nilai bobot yang dihasilkan digunakan untuk identifikasi buah tomat masak, muda, dan setengah masak [14].

Sistem Otomasi Dalam Penyortiran Tomat Dengan Image Processing Menggunakan Metode Deteksi RGB oleh Ihsan Nugraha dkk, 2015. Penelitian ini bertujuan mengidentifikasi warna tomat yang matang secara tampilan visual didapat dari proses mendeteksi nilai RGB (Red, Green dan Blue) pada warna tomat. Identifikasi tekstur warna pada tomat dengan ekstraksi cirinya dengan menggunakan metode deteksi warna dengan informasi yang didapatkan berupa nilai-nilai ciri statistik. Ekstraksi ciri citra diakuisisi dari kamera kemudian diklasifikasi dengan *euclidean distance* [20].

Pemrosesan Citra Digital Untuk Klasifikasi Mutu Buah Pisang Menggunakan Jaringan Syaraf Tiruan, 2014. Penelitian ini bertujuan mengolah citra pisang dengan pemrosesan citra digital untuk mengekstrak fitur warna dan tekstur buah pisang. Klasifikasi mutu pisang menggunakan jaringan syaraf tiruan *backpropagation* dengan parameter luas cacat, nilai red, green, blue, energy, homogeneity, dan contrast. Sistem klasifikasi mutu pisang dengan laju pembelajaran 0,3 dan jumlah neuron pada lapisan tersembunyi sebanyak 10 neuron dengan tingkat keberhasilan 94% dari 100% data uji pisang [21].

Peningkatan Kontras Menggunakan Metode *Contrast Limited Adaptive Histogram Equalization* Pada Citra *Underwater*, 2015. Penelitian ini bertujuan mengubah kualitas citra *underwater* yang rendah karena cahaya matahari sulit menembus kedalaman air metode CLAHE (*Contrast Limited Adaptive Histogram Equalization*) metode yang mampu meningkatkan kontras citra tetapi tidak berlebihan. Metode CLAHE beroperasi pada tile, kontras yang terdapat pada tiap – tiap tile akan diperbaiki dan mendapatkan hasil yang sesuai dengan yang diinginkan. Hasil penerapan metode CLAHE terbaik akan ditentukan dengan nilai MSE dan PSNR tertinggi [22].

BAB III METODOLOGI PENELITIAN

3.1. Waktu dan Tempat

Penelitian dan pembuatan Tugas Akhir ini dilakukan pada :

Waktu : Juni 2018 – Maret 2019.

Tempat : Kebun kopi Sinarwaya Kab. Pringsewu Lampung

Jadwal kegiatan penelitian yang dilakukan dapat dilihat pada Tabel 3.1 dibawah ini :

Tabel 3.1 Jadwal kegiatan penelitian.

No.	Aktifitas	Jun	Jul	Agu	Sep	Okt	Nov	Des	Jan	Feb	Mar
1	Studi <i>Literatur</i>										
2	Desain Sistem										
3	Penulisan										
4	Seminar Usul										
5	Pengujian Program										
6	Seminar Hasil										
7	Ujian Komprehensif										

3.2. Alat dan Bahan

Berikut ini merupakan alat yang digunakan pada penelitian dan pembuatan Tugas Akhir ini adalah :

1. Alat Penelitian

a. Perangkat Keras

- ✓ Laptop Dell latitude 6320 dengan spesifikasi *Processor* Intel(R) Core(TM) i7-2620M CPU @270GHz, RAM 4GB, System 64-bit.
- ✓ Kamera handphone Xiomi Note 4 dengan spesifikasi kamera 13MP.

b. Perangkat Lunak

- ✓ Sistem operasi Windows 7.
- ✓ MATLAB r2015a, digunakan untuk pembuatan sistem.
- ✓ Paint, digunakan untuk mengubah ukuran citra (*scaling*).
- ✓ Power Designer, digunakan untuk pembuatan *flowchart*.

2. Bahan Penelitian

Buah Kopi yang dipetik langsung dari kebun kopi yang berada di sinarwaya pringsewu lampung dengan 120 gambar sample yang terdiri dari 60 sample untuk pelatihan dan 60 sample gambar untuk pengujian. Pada 120 gambar sample terdiri dari 2 klasifikasi metode pengambilan gambar dengan melihat waktu pengambilan gambar yaitu siang atau saat cahaya matahari sudah memancarkan cahaya dengan baik dan saat siang namun cuaca sedang mendung atau cahaya yang dihasilkan tidak terlalu baik. Klasifikasi ini dilakukan untuk melihat apakah pencahayaan yang dilakukan saat pengambilan sample dapat mempengaruhi tingkat pembacaan warna pada sistem.

3.3. Tahapan Penelitian

Pada penelitian ini tahapan penelitian atau langkah-langkah yang dilakukan yaitu sebagai berikut :

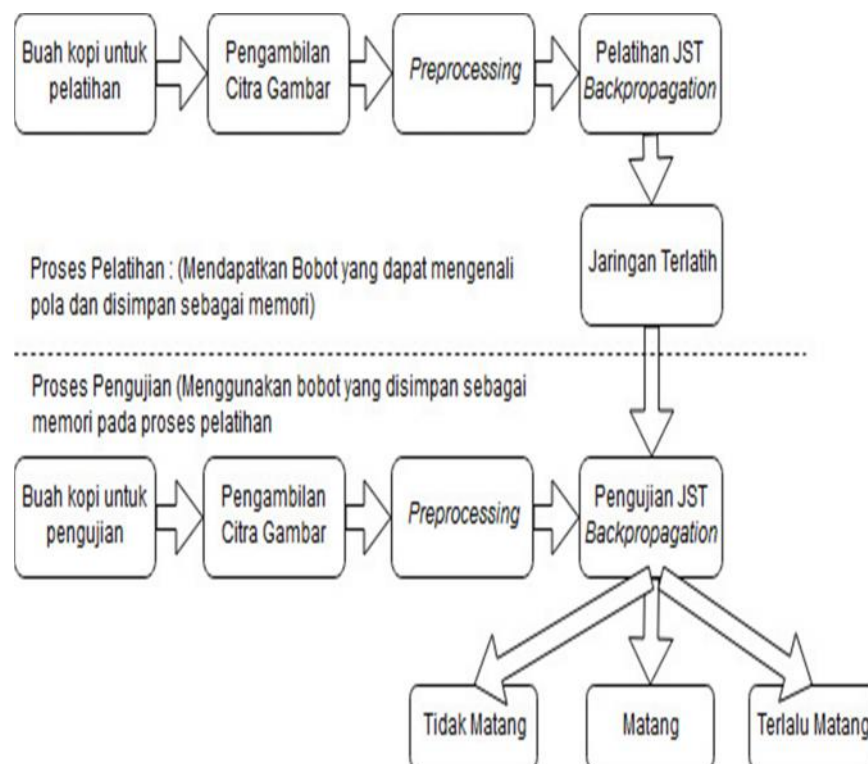


Gambar 3.1. Tahapan Penelitian

3.3.1. Analisis dan Perancangan Sistem

Analisis yang dilakukan pada penelitian ini dengan penjabaran penyusunan sistem dan mengidentifikasi masalah dengan menentukan nilai masukan, nilai keluaran, arsitektur jaringan, nilai ambang MSE (*Mean Squared Error*) sebagai kondisi berhenti,

learning rate, goal dan menentukan momentum. Penentuan nilai bobot ditentukan secara acak pada interval misalnya $[-1, +1]$ atau $[0.5, +0.5]$ ataupun lainnya (tidak adanya aturan baku mengenai interval ini). Pada perancangan sistem ini dilakukan pengolahan citra digital dengan ekstraksi ciri citra warna RGB (*preprocessing*), kemudian dilakukan pemrosesan dengan JST backpropagation yang terdiri dari 2 proses, yaitu proses *training* (pelatihan) dan proses *testing* (pengujian). Pada proses pelatihan dilakukan untuk melatih jaringan syaraf tiruan agar mampu mengenali dan mengidentifikasi warna buah kopi yang matang keseluruhan saat proses sortasi. Buah kopi sendiri disortasi untuk meningkatkan mutu dan kualitas kopi yang dihasilkan dengan mengklasifikasi 3 output keluaran yaitu tidak matang, matang dan terlalu matang. Berikut ini merupakan diagram alir untuk cara kerja dari sistem untuk memperjelas langkah-langkah kerja yang akan dilakukan pada sistem:



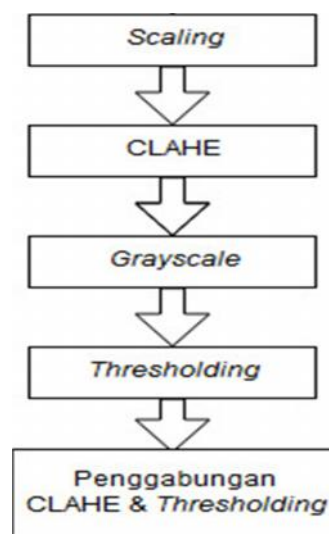
Gambar 3.2. Perancangan Sistem Identifikasi Buah Kopi

3.3.1.1. Pengambilan Citra Gambar Buah Kopi

Pada proses ini dilakukan pengambilan citra buah kopi menggunakan kamera handphone Xiaomi Note 4 dengan metode pengambilan sampel *random* (acak) yaitu berwarna merah tua, merah, merah muda, hijau kekuningan dan hijau. Citra yang diambil terdiri dari 5 citra warna dengan posisi pengambilan acak dan terdiri dari lebih dari satu buah kopi dengan format JPEG (*Joint Photographic Experts Group*). Kemudian citra disimpan dalam folder dengan nama Sample

3.3.1.2. Tahap *Preprocessing*

Pada tahapan *preprocessing* (pengolahan citra) dilakukan untuk menghilangkan *background* pada gambar agar sistem dapat terfokus membaca nilai warna RGB pada objek yaitu buah kopi dan meningkatkan kontras warna pada gambar. Berikut ini langkah-langkah yang dilakukan yaitu sebagai berikut :



Gambar 3.3. Tahapan *Preprocessing*.

3.3.1.3. *Scaling*

Pada proses ini *scaling* atau penskalaan merupakan suatu proses mengubah ukuran citra dengan tujuan untuk menyamakan seluruh ukuran citra. Ukuran citra pada data latih dan data uji akan menjadi sama yaitu dengan ukuran 500 x 500 piksel dengan tujuan agar citra tidak memiliki banyak nilai untuk diolah pada proses klasifikasi. Pada proses ini dilakukan dengan cara manual yaitu menggunakan aplikasi *Microsoft Paint* dengan melakukan *resize* ukuran pixel gambar yang kemudian file disimpan.

3.3.1.4. *Contrast- Limited Adaptive Histogram Equalization (CLAHE)*

Image Enhancement (Perbaikan Citra) dengan algoritma CLAHE membagi citra menjadi beberapa daerah kontekstual dan menerapkan pemerataan histogram dari masing – masing daerah tersebut. Metode ini akan meratakan distribusi nilai warna yang digunakan sehingga ciri dan kontras citra bisa terlihat, terutama pada daerah yang homogen dan menghindari penguatan dan gangguan (*noise*) yang ada pada citra. Pada penggunaannya pada Matlab menggunakan perintah *Adapthisteq*.

3.3.1.5. *Grayscale*

Pada proses ini citra akan melalui tahap *grayscale* yang merupakan perubahan citra warna menjadi citra keabuan setelah melalui proses *scaling*. Perubahan warna

gambar menjadi *grayscale* untuk memudahkan pada proses selanjutnya yaitu *thresholding*. Untuk mengubah citra berwarna menjadi *grayscale* yaitu dengan cara mengkonversi nilai matrik dengan mengambil rata-rata dari nilai *r*, *g* dan *b* seperti pada rumus di bawah ini :

$$\text{Grayscale} = \frac{r+g+b}{3} \dots\dots\dots [16]$$

Keterangan :

- r* = Nilai *Red*
- g* = Nilai *Green*
- b* = Nilai *Blue*

3.3.1.6. *Thresholding*

Proses *thresholding* merupakan tahapan perubahan kuantitas pada citra atau perubahan citra menjadi citra biner atau hitam putih. Perubahan ini digunakan untuk memisahkan antara objek dan *background* pada gambar. Hal ini digunakan untuk memfokuskan sistem membaca warna hanya pada objek yang diinginkan. *Thresholding* pada proses ini digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Menggunakan *thresholding* dilakukan dengan langkah berikut ini :

- a. Nilai *threshold* (*T*) ditentukan dengan rentan 0 -255, pada penelitian ini diambil nilai $T = 165$ dengan ketentuan semua objek yang terdapat pada semua *sample* tidak hilang bersama *background*.
- b. Nilai piksel jika didapat lebih dari atau sama dengan 165 maka nilai piksel pada citra diubah menjadi 1, sedangkan nilai piksel yang didapat kurang dari 138 maka diubah menjadi 0.

Berikut ini merupakan persamaan yang digunakan untuk mengubah citra grayscale menjadi biner dengan metode thresholding yaitu :

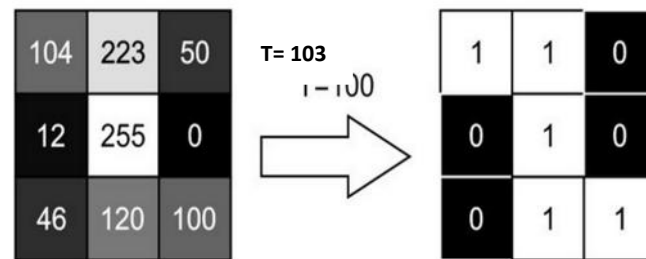
$$g(x,y) = \begin{cases} 1, & \text{jika } f(x,y) \geq T \\ 0, & \text{jika } f(x,y) < T \end{cases} \dots\dots\dots [17]$$

Keterangan :

$f(x,y)$: Nilai citra grayscale

$y(x,y)$: Nilai citra biner

T : Nilai *thresholding*



Gambar 3.4. Proses Perubahan Nilai Pixel pada Proses *Thresholding*.

3.3.1.6.1. *In Biner*

Pada proses *in biner* merupakan citra negatif yang merupakan citra yang berkebalikan dengan citra asli. Pada matlab fungsi ini biasa digunakan untuk mengubah citra biner yang sebelumnya objek bernilai 1 atau putih menjadi nilai 0 atau hitam dan *background* menjadi kebalikannya. Fungsi ini pada matlab sudah tersedia dengan menggunakan perintah *imcomplement*.

3.3.1.6.2. Perbaikan Garis Tepi Citra

Proses ini merupakan salah satu proses pengolahan citra yaitu *edge detection* yaitu proses menentukan lokasi titik-titik yang merupakan tepi objek. Fungsi ini digunakan untuk memperbaiki garis tepi objek yang diinginkan dan melakukan perbaikan citra agar objek yang diharapkan dapat terambil dengan baik.

3.3.1.7. Penggabungan CLAHE dan *Threshold*

Pada proses ini dilakukan penggabungan gambar antara gambar yang telah dilakukan CLAHE dengan gambar yang telah dilakukan *threshold*. Pada *sourcecode* Matlab menggunakan fungsi `cat(3,R,G,B)` untuk menggabungkan kedua fungsi tersebut, sehingga nantinya akan didapatkan gambar yang telah dipertajam warnanya dan dihilangkan *backgroundnya*.

3.3.2. Jaringan Syaraf Tiruan *Backpropagation*

Pada proses ini JST *backpropagation* mula-mula memisahkan data untuk pelatihan dan pengujian (*training* dan *testing*) jaringan. Umumnya *data* set terbagi menjadi dua yaitu pelatihan dan pengujian. Pelatihan dilakukan untuk menentukan bobot dan untuk mengklasifikasikan data untuk membentuk suatu pola. Selanjutnya *mechine learning* akan mempelajari pola dan akan disimpan. Kemudian dilakukan pengujian untuk melihat tingkat keberhasilan

yang dilakukan sistem, untuk itu data yang digunakan pada proses *training* dan *testing* harus berbeda sehingga dapat diketahui apakah pola yang dibaca oleh sistem sudah dapat membaca dengan benar atau tidak.

Pada proses pelatihan JST bertujuan untuk menentukan nilai bobot yang menghubungkan antara input ke *hidden layer* ke output dan meminimalkan kesalahan yang terjadi dan mengidentifikasi apakah model JST *backpropagation* ini sudah sesuai dengan target yang diharapkan. Pelatihan ini memerlukan data input dan data target. Pada pelatihan JST *backpropagation* terdapat dua cara umum pelatihan yaitu *backpropagation* dan *backpropagation* dengan momentum. Berikut ini merupakan persamaan dengan cara *backpropagation* momentum :

$$w_{kj}(t+1) = w_{kj}(t) + z_j + \mu(w_{kj}(t) - w_{kj}(t-1)) \dots\dots\dots [18].$$

Perubahan bobot menuju ke unit tersembunyi

$$v_{ji}(t+1) = v_{ji}(t) + z_j + \mu(v_{ji}(t) - v_{ji}(t-1)) \dots\dots\dots [19].$$

- μ merupakan momentum yang nilainya antara 0 dan 1.
- $w_{kj}(t+1) / v_{ji}(t+1)$ merupakan perubahan bobot dengan momentum.
- $w_{kj}(t-1) / v_{ji}(t-1)$ merupakan bobot mula-mula pada iterasi pola pertama (bobot awal).

Pelatihan JST *backpropagation* memiliki dua hal penting yaitu pada *learning rate* dan *momentum*. Fungsi *learning rate* yaitu untuk menentukan seberapa cepat jaringan tersebut dapat mempelajari pola dari data pelatihan. Fungsi *momentum* yaitu untuk meningkatkan kecepatan menemukan nilai yang diharapkan. Penambahan *momentum* digunakan untuk menghindari bobot yang terlalu mencolok akibat adanya data yang sangat berbeda. Penentuan nilai keduanya harus ditentukan dengan benar untuk meminimalkan proses pembelajaran yang terlalu lama jika nilai terlalu kecil dan

akan terjadi kesalahan pembacaan pola jika nilai terlalu besar. Momentum menunjukkan bahwa penentuan bobot pada iterasi sebelumnya mempengaruhi bobot saat ini.

3.3.2.1. **Arsitektur JST *Backpropagation***

Pada penelitian ini arsitektur JST *backpropagation* yang digunakan untuk *training* terdiri dari 60 sample gambar yang di ambil secara *random* dengan isi buah kopi lebih dari 1 buah kopi. Pada penelitian ini dibuat parameter-parameter jaringan sehingga dapat menghasilkan output yang sesuai dengan target yang diharapkan. Berikut ini merupakan parameter yang ditentukan untuk menghasilkan output yang optimal dan sesuai dengan target yang diharapkan :

- Parameter jumlah iterasi (*epoch*)
- Parameter untuk batas minimal nilai *error* atau mse.
- Parameter *learning rate* untuk menentukan seberapa cepat jaringan dapat mengenali pola.
- Parameter *momentum* untuk kecepatan menemukan nilai yang diharapkan.

3.3.2.2. **Fungsi Aktifasi**

Menurut Siang (2005), pada *backpropagation* fungsi aktifasi harus memenuhi beberapa syarat dalam penentuannya yaitu bersifat kontinu, terdiferensial dengan mudah dan bukan merupakan fungsi turun [6]. Fungsi aktifasi yang digunakan pada penelitian ini dan memenuhi

ketiga syarat tersebut yaitu fungsi *sigmoid biner* yang memiliki range (0,1).

3.3.2.3. Pemilihan Bobot

Pada penentuan bobot pada penelitian ini dilakukan secara acak dan terdiri dari bilangan acak kecil, karena apabila bobot menghasilkan nilai turunan aktivasi kecil dapat menyebabkan perubahan bobot menjadi sangat kecil. Demikian apabila nilai bobot awal terlalu besar maka nilai turunan fungsi aktivasi menjadi sangat kecil pula.

3.3.2.4. Jumlah Unit Tersembunyi

Penentuan jumlah *hidden layer* dapat ditentukan dengan *trial end error*. Namun pada dasarnya jaringan dengan sebuah layer tersembunyi sudah cukup untuk JST *backpropagation* untuk mengenali suatu pola antara masukan dan target. Akan tetapi, penambahan *hidden layer* kadangkala membuat pelatihan lebih mudah. Pada penelitian ini terdiri dari 3 buah input dan 3 buah output. Penentuan jumlah *hidden layer* dan jumlah neuron pada tiap *hidden layer* ditentukan saat *trial end error* pelatihan.

3.3.2.5. Iterasi

Penggunaan JST *backpropagation* bertujuan untuk mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar. Jaringan dapat dilakukan pelatihan

secara terus menerus hingga pola dapat dikenali dengan benar. Namun hal tersebut tidak menjamin jaringan mampu mengenali pola pengujian dengan tepat. Menurut Siang (2005), Melakukan iterasi secara terus menerus tidaklah bermanfaat hingga semua kesalahan pola pelatihan bernilai 0. Kesalahan yang terjadi pada semua data (pelatihan dan pengujian) apabila kesalahan menurun pelatihan dapat terus dijalankan, namun jika kesalahan terus meningkat pelatihan tidak ada gunanya untuk diteruskan karena jaringan telah mengambil sifat yang hanya dimiliki secara spesifik oleh data pelatihan (tidak dimiliki oleh data pengujian) dan mulai kehilangan kemampuan untuk *generalisasi* (menyamarkan) [6].

3.3.2.6. Penentuan *Goal*, *Learning Rate* dan *Momentum*

Pada proses belajar jaringan penentuan batasan nilai *error* penting karena dapat membatasi iterasi mse yang dilakukan. Iterasi akan berhenti jika $mse < \text{batas}$ yang ditentukan yaitu 0.0001. Parameter ini cukup untuk membatasi nilai mse pada pembelajaran dan pelatihan pada penelitian ini. Pada penelitian ini digunakan nilai sebesar 0.01 sebagai nilai *learning rate*. Digunakan nilai tersebut karena *defaultnya* (standar) yang digunakan pada Matlab. Parameter penentuan nilai momentum yaitu berdasarkan *defaultnya* yaitu 0.9.

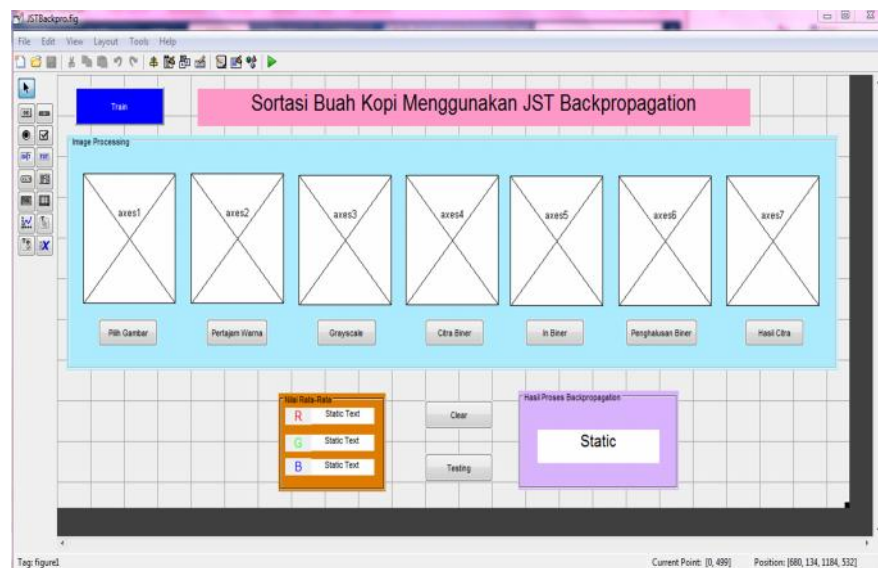
Tabel 3.2. Arsitektur penelitian.

Karakteristik	Spesifikasi
- Jumlah <i>hidden layer</i>	3
- Neuron lapisan input	3
- Neuron lapisan tersembunyi 1	40
- Neuron lapisan tersembunyi 2	15
- Neuron lapisan tersembunyi 3	5
- Neuron lapisan output	3 (Red, Green, Blue)
- Fungsi aktivasi lapisan <i>hidden layer</i>	<i>Sigmoid Biner</i>
- Fungsi aktivasi lapisan output	<i>Sigmoid Biner</i>
- Parameter <i>epoch</i>	1.000
- Parameter <i>goal</i>	0,0001
- Parameter <i>learning rate</i>	0,01
- Parameter <i>momentum</i>	0,9
- Fungsi pelatihan jaringan	<i>Trainrp</i>
- Fungsi perubahan bobot	<i>Learngdm</i>
- Fungsi perhitungan error	<i>mse</i>

3.3.3. Pembuatan Sistem

Pada penelitian ini sistem dibangun dengan menggunakan aplikasi MATLAB R2015a. Alasan menggunakan aplikasi Matlab karena Matlab dianggap lebih fleksibel dalam pembuatan sistem JST dan Matlab memiliki fungsi JST terutama JST *backpropagation* [6].

Keunggulan yang dimiliki ini sehingga membuat penulis memilih aplikasi ini sebagai dasar pembuatan sistem dalam penelitian ini. Sistem yang akan dibangun dirancang dengan menggunakan *Graphical User Interface* (GUI) untuk memudahkan penggunaan sistem. Berikut ini merupakan rancangan GUI yang telah dibangun menggunakan Matlab.



Gambar 3.5. Pembuatan Sistem GUI

Pada gambar 3.5. merupakan rancangan GUI yang akan ditampilkan pada saat program dijalankan. Program ini dimulai dengan melakukan *training* atau pelatihan jaringan dengan data latih dan target yang telah diinputkan. Kemudian program ini terdiri pengolahan citra digital (*preprocessing*) dengan 7 langkah pengolahan yaitu pemilihan citra gambar yang akan diolah, pertajam warna (CLAHE), perubahan warna menjadi *grayscale*, mengubah citra menjadi citra biner (*black and white*), merubah citra menjadi citra *in biner* (membalikkan citra dari 0 menjadi 1 dan sebaliknya), penghalusan citra in biner, dan terakhir penggabungan dari keseluruhan proses yaitu citra hasil yang berfungsi untuk mengambil nilai warna *red*, *green*, dan *blue* yang telah dipertajam warnanya dan dihilangkan *backgroundnya*. Kemudian sample uji yang telah

melalui tahap *preprocessing* dilakukan *testing JST backpropagation* menggunakan bobot akhir pada *training*. Pembuatan GUI pada gambar berguna sebagai *User Interface* untuk memudahkan pengguna dalam melakukan identifikasi kematangan buah kopi. Tahap pertama pada saat ingin menjalankan program ini yaitu melakukan *training JST backpropagation* dengan menekan tombol “Train”.



Gambar 3.6. GUI Push Button untuk Train.

Pada gambar diatas merupakan *pushbutton* untuk melakukan *training JST backpropagation* dengan *sourcecode* berikut ini.

```

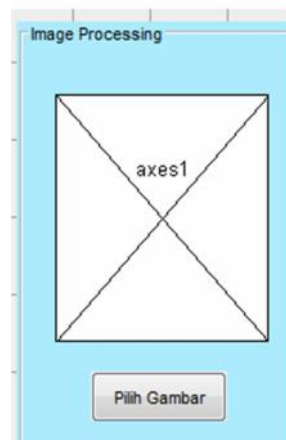
77 % --- Executes on button press in pushbutton8.
78 function pushbutton8_Callback(hObject, eventdata, handles)
79 % hObject handle to pushbutton8 (see GCBO)
80 % eventdata reserved - to be defined in a future version of MATLAB
81 % handles structure with handles and user data (see GUIDATA)
82
83 proyek=guidata(hObject);
84 a = load('inputdatalatih.mat'); %data training
85 t = load('output.mat'); %target
86
87 %membangun jaringan
88 net=newff(minmax(a.inputdatalatih),[40 15 5 3], {'logsig' 'logsig' 'logsig' 'logsig'},'trainrp', 'learnqdm', 'mse') %neural dan l
89
90 %melihat bobot awal
91 bobotawal_input = net.IW(1,1) %bobot akan direndom oleh matlab
92
93 %penentuan parameter training
94 net.trainParam.epochs=1000; %jumlah iterasi
95 net.trainParam.goal=0.0001; %minimalnilai error
96 net.trainParam.lr=0.01; %learning rate
97 net.trainParam.mc=0.9; %momentum
98
99 %melakukan training
100 net=train(net,a.inputdatalatih,t.output)
101
102 %melihat bobot akhir
103 bobotakhir = net.IW(1,1)
104 bias_akhir = net.b(1)
105 bobotunit_tersembunyi = net.LW(2,1)
106 biasunit_tersembunyi = net.b(2)
107 bobotunit_tersembunyi_ke_output = net.LW(3,2)
108 biasunit_tersembunyi_ke_output = net.b(3)
109
110 save net.mat net
111
112 hasil=sim(net,a.datalatih)
113
114 f = msgbox('Training Completed','Success');

```

Gambar 3.7. Source Code untuk Training JST Backpropagation.

Pada gambar diatas merupakan *source code* untuk melakukan pelatihan pada jaringan *JST backpropagation* dengan mengambil

data pada `datalatih.mat` dan `target.mat`. Kemudian membangun jaringan dengan JST *backpropagation* dengan format yang telah disediakan oleh Matlab `newff` dengan 3 buah *hidden layer* yang berisikan 40 neuron pada *hidden layer* pertama, 15 neuron pada *hidden layer* ke dua dan 5 *hidden layer* ke tiga. Menggunakan fungsi aktivasi sigmoid biner (`logsig`), fungsi pelatihan, fungsi bobot dan fungsi perhitungan *error*. Kemudian terdapat source code untuk melihat bobot awal input, penentuan parameter *training* dan melakukan *training*, kemudian melihat fungsi bobot, hasil pelatihan JST *backpropagation* dengan data latih dan yang terakhir menampilkan *message box* “Training Completed” “*Succes*”. Pada gambar 3.5 terdapat 7 axes yang setiap axesnya berfungsi untuk menampilkan gambar yang akan dilakukan pengolahan citra. Pada Axes 1 akan menampilkan gambar yang dipilih dengan perintah tombol “Pilih Gambar”.



Gambar 3.8. GUI Axes1 dengan Perintah Pilih Gambar.

Pada gambar 3.8 berfungsi untuk menampilkan citra gambar yang akan diolah pada proses pengolahan citra selanjutnya dengan *sourcecode* program berikut ini .

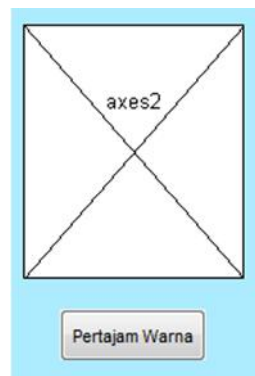

```

129 % --- Executes on button press in pushbutton1.
130 function pushbutton1_Callback(hObject, eventdata, handles)
131 % hObject handle to pushbutton1 (see GCBO)
132 % eventdata reserved - to be defined in a future version of MATLAB
133 % handles structure with handles and user data (see GUIDATA)
134
135 proyek=guidata(gcbo);
136 [namafile,direktori]=uigetfile({'*.jpg'; '*.bmp'; '*.png'; '*.tif'}, 'Buka Gambar')
137 if isequal(namafile,0)
138 return;
139 end
140 eval(['cd '' direktori ''']);
141 I=imread(namafile);
142 set(proyek.figure1, 'CurrentAxes', proyek.axes1);
143 set(imshow(I));
144
145 set(proyek.figure1, 'Userdata', I);
146 set(proyek.axes1, 'Userdata', I);
147

```

Gambar 3.9. *Source Code* untuk Fungsi Pilih Gambar.

Pada gambar 3.9 merupakan source code yang digunakan dengan perintah membuka direktori file, kemudian dilakukan pemilihan gambar yang diinginkan pada direktori dengan format gambar .jpeg, .bmp, .tif. kemudian sistem akan menampilkan gambar pada Axes1 yang selanjutnya gambar akan disimpan pada program dengan nama I untuk memudahkan pemanggilan gambar pada proses selanjutnya. Selanjutnya pada Axes2 dengan push button “Pertajam Warna”.



Gambar 3.10. GUI Axes2 dengan perintah pertajam warna.

Pada gambar 3.10. merupakan tampilan GUI dengan perintah perbaikan kontras warna pada gambar dengan menggunakan metode CLAHE. Fungsi ini sendiri digunakan pada pengolahan citra untuk memperbaiki kontras warna citra namun tidak dengan berlebihan sehingga memudahkan dalam proses klasifikasi saat pembacaan nilai RGB pada citra. Berikut ini merupakan source code yang digunakan.

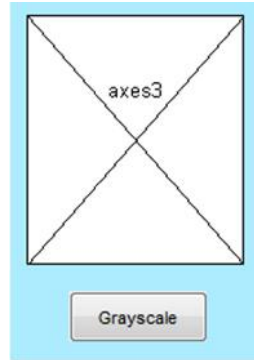
```

149 % --- Executes on button press in pushbutton2.
150 function pushbutton2_Callback(hObject, eventdata, handles)
151 % hObject handle to pushbutton2 (see GCBO)
152 % eventdata reserved - to be defined in a future version of MATLAB
153 % handles structure with handles and user data (see GUIDATA)
154 proyek=guidata (gcbo);
155 I=get(proyek.axes1,'Userdata');
156
157 Red= I(:, :,1);
158 Green= I(:, :,2);
159 Blue= I(:, :,3); %memisahkan setiap warna RGB
160
161 reds= adapthisteq(Red);
162 greens= adapthisteq(Green);
163 blues= adapthisteq(Blue);
164 %mempertajam kontrasnya dengan Contrast Limited Adaptive Histogram
165 %Equalization (CLAHE) pada tiap layar warnanya
166
167 edgeSharp(:, :,1)= reds;
168 edgeSharp(:, :,2)= greens;
169 edgeSharp(:, :,3)= blues; %menggabungkan kembali ketiga warna
168 edgeSharp(:, :,2)= greens;
169 edgeSharp(:, :,3)= blues; %menggabungkan kembali ketiga warna
170
171 set(proyek.figure1,'CurrentAxes',proyek.axes2);
172 set(imshow(edgeSharp));
173 set(proyek.figure1,'Userdata',edgeSharp);
174 set(proyek.axes2,'Userdata',edgeSharp);
175

```

Gambar 3.11. *Source Code* untuk Fungsi Pertajam Warna.

Pada gambar 3.11. merupakan source code pertajam warna citra dengan menggunakan metode CLAHE pada gambar diatas dapat dilihat pada line 155 merupakan perintah untuk memanggil data pada axes1 yang sebelumnya telah disimpan dengan nama I. Selanjutnya dilakukan pemisahan warna Red, Green , dan Blue pada citra dengan perintah pada line 157-159, setelah dilakukan pemisahan warna RGB selanjutnya digunkan metode CLAHE dengan sorce code pada Matlab yaitu *adapthisteq*, fungsi ini digunakan untuk mempertajam kontras setiap warna pada RGB citra. Kemudian setelah setiap warna RGB dipertajam dengan metode CLAHE perintah selanjutnya yaitu menggabungkan kembali ketiga warna tersebut dan ditampilkan pada axes2 dengan nama file yang disimpan yaitu edgeSharp. Berikut ini gambar GUI untuk perintah Grayscale.



Gambar 3.12. GUI Axes3 dengan perintah Grayscale

Pada gambar 3.12. merupakan tampilan GUI untuk menampilkan citra grayscale. Fungsi ini sendiri digunakan pada untuk merubah warna citra RGB menjadi warna abu-abu sehingga akan lebih mudah untuk proses selanjutnya yaitu proses *threshold*. Berikut ini merupakan source code yang digunakan pada proses ini.

```

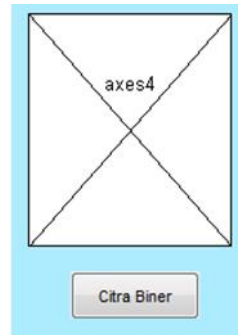
177 % --- Executes on button press in pushbutton3.
178 function pushbutton3_Callback(hObject, eventdata, handles)
179 % hObject handle to pushbutton3 (see GCBO)
180 % eventdata reserved - to be defined in a future version of MATLAB
181 % handles structure with handles and user data (see GUIDATA)
182
183 - proyek=guidata (gcbo);
184 - edgeSharp=get(proyek.axes2,'Userdata');
185 - Igray=rgb2gray(edgeSharp);
186 - set(proyek.figure1,'CurrentAxes',proyek.axes3);
187 - set(imshow(Igray));
188 - set(proyek.figure1,'Userdata',Igray);
189 - set(proyek.axes3,'Userdata',Igray);
190

```

Gambar 3.13. Source Code untuk Perintah Grayscale.

Pada gambar di atas dapat dilihat bahwa sistem menggunakan perintah untuk memanggil data yang sebelumnya disimpan yaitu *edgeSharp* yang sebelumnya telah ditampilkan pada *axes2* fungsi itu terdapat pada line 184. Kemudian pada line 185 dapat dilihat digunakan fungsi *rgb2gray* fungsi ini merupakan salah satu fungsi otomatis yang dimiliki Matlab untuk mengubah citra warna RGB menjadi citra grayscale atau abu-abu. Kemudian sistem menyimpan file grayscale ini dengan nama *Igray* yang memudahkan saat

pemanggilan data pada proses pengolahan citra selanjutnya. Berikut ini merupakan tampilan GUI dengan *push button* citra biner.



Gambar 3.14. GUI Axes4 dengan Perintah Citra Biner.

Pada 3.14. merupakan tampilan GUI dengan menampilkan citra biner yang berfungsi untuk mengubah citra yang mana piksel-piksel citra diubah dan hanya memiliki dua buah nilai derajat keabuan (*grayscale*) yaitu hitam dan putih atau pixel-pixel citra akan bernilai 1 pada objek dan bernilai 0 pada latar belakang. Fungsi ini digunakan untuk memudahkan dalam pembacaan nilai RGB citra dengan cara memisahkan objek gambar dengan *background*. Berikut ini source code yang digunakan pada perintah ini.

```

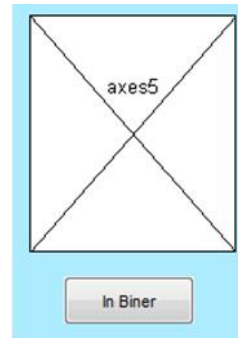
198 - proyek=guidata(gcbo);
199 - Igray=get(proyek.axes3,'Userdata');
200 - Ibw = im2bw(Igray,165/255); %mengkonversikan citra gray menjadi citra biner
201 -                               %dengan threshold
202 - set(proyek.figure1,'CurrentAxes',proyek.axes4);
203 - set(imshow(Ibw));
204 - set(proyek.figure1,'Userdata',Ibw);
205 - set(proyek.axes4,'Userdata',Ibw);
206 -

```

Gambar 3.15. *Source Code* untuk Perintah Citra Biner.

Pada gambar 3.15. merupakan source code untuk mengubah citra grayscale menjadi citra biner. Pada line 199 sistem memanggil file sebelumnya yaitu Igray yang kemudian diolah gambarnya dengan fungsi *im2bw* dengan *threshold*. Pada line 200 dapat dilihat *Ibw = im2bw(Igray, 165/255)* merupakan fungsi yang digunakan untuk mengkonversi citra grayscale menjadi citra biner dengan *threshold*

citra keabuan bernilai 165/255. Fungsi dari nilai 165 merupakan perubahan nilai piksel ≥ 165 pada proses thresholding akan bernilai 1 atau putih pada citra biner. Sedangkan nilai yang kurang dari 165 akan bernilai 0 atau berwarna hitam. Berikut ini merupakan gambar axes4 dengan perintah in biner.



Gambar 3.16. GUI Axes5 dengan perintah *In Biner*.

Pada gambar 3.16 merupakan tampilan GUI untuk menampilkan in biner atau perintah untuk membalikkan nilai biner dengan tujuan untuk menampilkan objek yang akan digunakan dengan membalik nilai biner sebelumnya bernilai 1 menjadi 0 dan nilai 0 menjadi 1. Berikut ini source code yang digunakan pada push button tersebut.

```

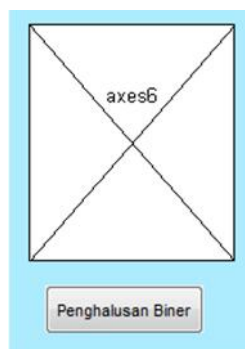
208 % --- Executes on button press in pushbutton5.
209 function pushbutton5_Callback(hObject, eventdata, handles)
210 % hObject handle to pushbutton5 (see GCBO)
211 % eventdata reserved - to be defined in a future version of MATLAB
212 % handles structure with handles and user data (see GUIDATA)
213
214 proyek=guidata(gcbo);
215 Ibw=get(proyek.axes4,'Userdata');
216 Ibw = imcomplement(Ibw);
217 set(proyek.figure1,'CurrentAxes',proyek.axes5);
218 set(imshow(Ibw));
219 set(proyek.figure1,'Userdata',Ibw);
220 set(proyek.axes5,'Userdata',Ibw);
221

```

Gambar 3.17. Source Code untuk Perintah *In Biner*.

Pada gambar 3.17. sama halnya dengan proses sebelumnya yaitu sistem membaca proses sebelumnya yaitu Ibw atau citra biner yang

telah di threshold pada axes4 kemudian pada line 216 digunakan fungsi imcomplement (Ibw) yang berfungsi untuk membalikkan nilai biner sehingga tampilan yang akan dimunculkan pada axes5 merupakan kebalikan dari axes4 yaitu background yang semula berwarna putih menjadi hitam dan object yang semula berwarna hitam berubah menjadi warna putih. Berikut ini merupakan perintah untuk menampilkan proses selanjutnya pada sistem ini yaitu penghalusan gambar biner.



Gambar 3.18. GUI Axes6 dengan Perintah Penghalusan *Biner*.

Pada gambar 3.18. merupakan perintah untuk menampilkan proses penghalusan nilai tepi biner sehingga memudahkan dalam proses selanjutnya yaitu memisahkan background dengan objek yang dibutuhkan. Berikut ini merupakan source code yang digunakan pada proses ini.

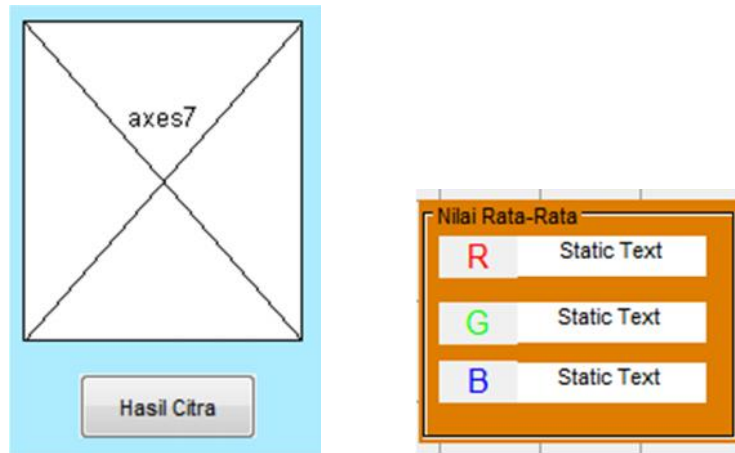
```

223 % --- Executes on button press in pushbutton6.
224 function pushbutton6_Callback(hObject, eventdata, handles)
225 % hObject handle to pushbutton6 (see GCBO)
226 % eventdata reserved - to be defined in a future version of MATLAB
227 % handles structure with handles and user data (see GUIDATA)
228
229 proyek=guidata(gcbo);
230 Ibw=get(proyek.axes5,'Userdata');
231 Ibw = imfill(Ibw,'holes'); %perbaiki garis tepi citra
232 Ibw = bwareaopen(Ibw,100); %untuk menghilangkan object kecil pada matriks
233 %citra dengan menghilangkan luasan yang kurang dari 100
234 str = strel('disk',5);
235 Ibw = imerode(Ibw,str);
236 set(proyek.figure1,'CurrentAxes',proyek.axes6);
237 set(imshow(Ibw));
238 set(proyek.figure1,'Userdata',Ibw);
239 set(proyek.axes6,'Userdata',Ibw);
240

```

Gambar 3.19. *Source Code* untuk Perintah Penghalusan Biner.

Pada gambar 3.19. merupakan source code yang digunakan pada proses penghalusan citra biner fungsi tersebut digunakan untuk memperbaiki garis tepi citra dengan source code yang digunakan pada line 231 atau pada Matlab digunakan fungsi `imfill` dan pada line 232 terdapat fungsi untuk menghilangkan objek kecil citra dengan menghilangkan luasan citra yang nilainya kurang dari 100. Sehingga citra yang akan ditampilkan akan lebih halus pada tepi gambarnya yang memisahkan antara objek dan background dan tidak adanya objek kecil berwarna hitam pada bagian-bagian citra objek yang berwarna putih. Selanjutnya digunakan fungsi `strel` yang berguna sebagai elemen struktur atau penentu bentuk pada line 234 fungsi `strel` yang digunakan yaitu 'disk', 5 yang artinya elemen struktur berbentuk cakram dengan radius 5. Kemudian pada fungsi `imerode(Ibw,str)` yaitu sebagai fungsi yang digunakan untuk operasi erosi dalam menyusutkan elemen dengan menggunakan elemen `strel` yang sebelumnya telah dibentuk atau `strel('disk', 5)` pada gambar `Ibw`. Berikut ini merupakan GUI hasil dari pengolahan citra yang telah dilakukan pada beberapa proses sebelumnya.



Gambar 3.20. GUI Hasil Citra dan Hasil Pembacaan Nilai RGB.

Pada gambar 3.20. merupakan GUI yang akan menampilkan hasil dari keseluruhan proses pengolahan citra digital yang telah dilakukan pada proses-proses sebelumnya pada axes 7. Berikut ini merupakan source code yang digunakan pada push button hasil citra.


```

268 % --- Executes on button press in pushbutton7.
269 function pushbutton7_Callback(hObject, eventdata, handles)
270 % hObject handle to pushbutton7 (see GCBO)
271 % eventdata reserved - to be defined in a future version of MATLAB
272 % handles structure with handles and user data (see GUIDATA)
273
274 proyek=guidata(gcbo);
275 edgeSharp=get(proyek.axes2,'Userdata');
276 Ibw=get(proyek.axes6,'Userdata');
277
278 R = edgeSharp(:,:,1);
279 G = edgeSharp(:,:,2);
280 B = edgeSharp(:,:,3);
281
282 R(~Ibw) =0; %pembacaan nilai tepi RGB
283 G(~Ibw) =0;
284 B(~Ibw) =0;
285
286 RGB_stretch = cat(3,R,G,B);
287
288 set(proyek.figure1,'CurrentAxes',proyek.axes7);
289 set(imshow(RGB_stretch));
290 set(proyek.figure1,'Userdata',RGB_stretch);
291 set(proyek.axes7,'Userdata',RGB_stretch);
292
293 sumI = sum(sum(RGB_stretch)); %jumlah nilai RGB
294 s = nnz(RGB_stretch); %fungsi nonzeros pada matriks
295 rata_rataI = sumI*3/s; %rumus menghitung jumlah piksel
296
297 rata_rataR = rata_rataI(1);
298 handles.rata_rataI(1)=rata_rataR;
299 rata_rataG = rata_rataI(2);
300 handles.rata_rataI(2)=rata_rataG;
301 rata_rataB = rata_rataI(3);
302 handles.rata_rataI(3)=rata_rataB;
303
304 guidata(hObject,handles);
305 set(handles.text3,'string',rata_rataR);
306 set(handles.text4,'string',rata_rataG);
307 set(handles.text5,'string',rata_rataB);
308

```

Gambar 3.21. *Source Code* untuk Perintah Hasil.

Pada gambar 3.21. source code yang digunakan untuk menampilkan hasil citra dengan meningkatkan kontras warna gambar dan menghilangkan background pada gambar. Fungsi yang digunakan pada source code diatas yaitu program memanggil 2 buah data yang disimpan yaitu data edgeSharp yang ditampilkan pada axes2 dan data Ibw yang ditampilkan pada axes6. Selanjutnya dilakukan pemisahan nilai Red, Green, dan Blue pada data edgeSharp seperti pada line 278-280, kemudian dilakukan pembacaan nilai tepi pada RGB menggunakan file Ibw dan yang terakhir pada line 286

digunakan perintah cat (3,R,G,B) yaitu menggabungkan kedua buah file sebelumnya yaitu edgeSharp dan Ibw menjadi satu gambar yang utuh dan ditampilkan pada axes7. Setelah itu dilakukan pembacaan nilai objek gambar berupa nilai Red, Green, dan Blue dengan rumus yang terdapat pada line 293. Fungsi rumus tersebut digunakan sebagai acuan untuk menentukan nilai rata-rata dari setiap warna yaitu Red, Green, dan Blue, namun karena telah dilakukan penghilangan nilai pada *background* maka nilai saat ini *background* bernilai 0 dan berpengaruh pada saat dilakukan penghitungan rata-rata karena jumlah pixel *background* akan tetap dihitung. Maka digunakan fungsi nnz (*none zero*) yang berfungsi untuk tidak menambahkan jumlah pixel zero dan membuahkan nilai hasil rata-rata nilai warna menjadi lebih maksimal yang kemudian ditampilkan pada table yang telah disediakan pada tampilan GUI sistem ini. Berikut ini merupakan tampilan sistem pada saat dijalankan.



Gambar 3.22. Tampilan Program Saat Dijalankan.

Pada gambar 3.22. merupakan tampilan sistem identifikasi kematangan buah kopi pada saat dijalankan dalam Matlab. Pada tombol paling kiri merupakan perintah untuk menjalankan *pushbutton1* atau “Pilih Gambar” maka sistem dapat memilih ataupun mengambil gambar dengan format .bmp atau .jpeg dan file

akan menyimpan nilai variabel kemudian ditampilkan pada axes1. Selanjutnya tombol kedua dari kiri yaitu fungsi *pushbutton2* atau “Pertajam warna”. Kemudian tombol *pushbutton3* atau “Grayscale” untuk mengubah warna citra menjadi *greyscale*, fungsi *pushbutton4* atau “Citra Biner” untuk mengubah citra *grayscale* menjadi citra biner, fungsi *pushbutton5* atau “In Biner” untuk membalikkan citra objek dan citra *background*. Kemudian fungsi *pushbutton6* atau “Penghalusan Biner” untuk menghaluskan nilai tepi pada citra *in biner*, fungsi *pushbutton6* atau “Hasil Citra” untuk menampilkan citra yang telah dipertajam dan dihilangkan *backgroundnya*, serta membaca nilai warna citra yaitu *red*, *green* dan *blue*.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berikut ini merupakan kesimpulan yang diperoleh dari penelitian ini:

1. Pada GUI Matlab identifikasi kematangan buah kopi menggunakan metode *backpropagation* sudah dapat berjalan dengan baik dan telah mencapai tingkat kesalahan minimum. Pada pelatihan sistem menggunakan 60 sample data latih dan target yang diharapkan dengan menggunakan 3 buah *hidden layer* yang terdiri dari 40 neuron pertama, 15 neuron kedua, dan 5 neuron ketiga. serta parameter yang ditentukan pada pelatihan yaitu *epoch* = 1000, *goal* atau batas *mse* = 0,0001, *learning rate* atau laju pemahaman = 0,1 dan momentum = 0,9 sebagai batasan sistem untuk melakukan *stop condition*.
2. Pada penelitian ini telah dilakukann penelitian awal untuk menguji jaringan syaraf tiruan metode *backpropagation* yang digunakan sebagai identifikasi kematangan buah kopi. Pada penelitian ini sistem menunjukkan tingkat akurasi benar mencapai 96,66% dengan 4 buah kondisi yaitu matang, tidak matang, terlalu matang dan tidak diketahui menggunakan 60 sample baru yang telah dibedakan dengan sample latih dan sistem dapat membaca pola dengan benar sebanyak 58 buah sample dan 2 buah sample terbaca dengan salah.

5.2. Saran

Berikut ini merupakan beberapa saran untuk menyempurnakan penelitian ini yaitu :

1. Pada sistem identifikasi kematangan buah kopi yang telah dibangun menggunakan metode *JST backpropagation* sistem yang dibangun dapat mengenali pola pelatihan dan pengujian yang telah dilakukan, namun untuk tahap saat ini sistem belum dapat digunakan oleh petani kopi dan masih menjadi studi awal untuk pengujian metode *JST backpropagation*.
2. Pada penelitian dan pembuatan sistem ini hanya dilakukan pada identifikasi warna pada gambar buah kopi sehingga diharapkan pada penelitian selanjutnya dapat menambahkan identifikasi lainnya seperti tekstur dan ukuran buah kopi menggunakan metode *JST backpropagation*.

DAFTAR PUSTAKA

- [1] Direktorat Jendral Perkebunan.2015.*Statistikk Perkebunan Indonesia*. Jakarta: Dirjen Perkebunan.
- [2] Kementerian Pertanian Direktorat Jenderal Perkebunan. 2014. *Pedoman Teknis Budidaya Kopi yang Baik (Good Agriculture Practices/GAP On Coffee)*. Jakarta: Kementerian Pertanian Dirjen perkebunan.
- [3] Noor,M. Helmi dan Moch. Hariadi. 2009. Seminar Nasional Informatika UPN “Veteran”. *Image Cluster Berdasarkan Warna Untuk Identifikasi Kematangan Buah Tomat Dengan Metode Valley Tracing*. Sub A: No 15-24.
- [4] Kastaman,Roni dan Fadhil Abdulfatah. 2009. *Analisis Kinerja Perangkat Lunak Pengolahan Citra dengan Menggunakan Beberapa Metode Klasifikasi untuk Menentukan Kualitas Buah Manggis*. [Internet]. Tersedia di: <https://repository.ipb.ac.id/jspui/handle/123456789/31881>.
- [5] Jumarwanto, A. Hartanto, R. Prastiyanto, D.2009. *Jurnal Teknik Elektro. Aplikasi Jaringan Syaraf Tiruan Backpropagation untuk Memprediksi Penyakit THT di Rumah Sakit Mardi Rahayu Kudus*. Vol 1 No 1.
- [6] Siang, Jong J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan Matlab*. Andi. Yogyakarta.
- [7] Cronuist, A. 1981. *An Integrated System Of Classification Of Flowering Plants*. [Internet]. Tersedia di: https://books.google.co.id/books/about/An_Integrated_System_of_Classification_o.html?id=EB6qflbmcrc&redir_esc=y.
- [8] Wachjar,A.1984. *Pengantar Budidaya Kopi*. Bogor : Fakultas pertanian.
- [9] Najiyati,S dan Danarti. 2001. *Kopi Budidaya dan Penanganan Lepas Panen*. Jakarta : Penebar Swadaya Jakarta.
- [10] Rahardjo, Pudji. 2012. *Panduan Budidaya dan Pengolahan Kopi Arabika*

dan Robusta. Jakarta. Penebar Swadaya.

- [11] Ciptadi dan MZ Nasution. 1985. *Pengolahan Kopi*. Bogor. Agro Industri Press.
- [12] Putra, Darma. 2010. *Pengolahan Citra Digital*. Andi Offset. Yogyakarta.
- [13] Microsoft Windows. 1985. *Microsoft Paint* di: https://en.wikipedia.org/wiki/Microsoft_Paint (di akses 10 Oktober 2018).
- [14] Madenda, Sarifuddin. 2015. *Pengolahan Citra dan Video Digital*. Erlangga. Jakarta.
- [15] Kusumanto, RD dan Tompunu, Alan Novi. 2011. *Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Warna Model Normalisasi RGB*. [Internet]. Tersedia di: <https://publikasi.dinus.ac.id/index.php/semantik/article/view/153>.
- [16] Deswari, Dila. Hendric. Derisma. 2010. *Identifikasi Komatangan Buah Tomat Menggunakan Metode Backpropagation*. [Internet]. Tersedia di: repo.unand.ac.id/269/
- [17] Arham, Zaenul dkk. 2004. *Evaluasi Mutu Jeruk Nipis (Citrus Aurantifolia Swingle) Dengan Pengolahan Citra Digital Dan Jaringan Syaraf Tiruan*. Bogor. [Internet]. Tersedia di: <https://repository.ipb.ac.id/handle/123456789/8087>
- [16] Rapidtables. *RGB Color* di: https://www.rapidtables.com/web/color/RGB_Color.html (di akses 10 Oktober 2018).
- [17] Tim Parker, Rhodes.Fay,dkk. 2002. *Flash MX Designer's Action Script Reference*. [Internet]. Tersedia di: <https://www.apress.com/gp/book/9781590591659>.
- [18] Irwan, Muhamad. Sianipar, RH. 2018. *Pengantar Pengolahan Citra Digital*. Sparta Publisher.
- [19] MathWorks.*Adaptive Histogram Equalization*. Tersedia di: <https://www.mathworks.com/help/images/adaptive-histogram-equalization.html>. (di akses 11 November 2018).
- [20] Nugraha, Drs Suwandi M. Si, Bethaningtyas. 2015. *Sistem Otomasi Dalam Penyortiran Tomat Dengan Image Processing Menggunakan Metode Deteksi RGB*. [Internet]. Tersedia di:

<https://openlibrary.telkomuniversity.ac.id/.../sistem-otomasi-dalam-penyortiran-tomat>. (di akses 11 November 2018).

- [21] Harjoko, Agus. 2014. *Pemrosesan Citra Digital Untuk Klasifikasi Mutu Buah Pisang Menggunakan Jaringan Saraf Tiruan*. Yogyakarta. Gadjah Mada University.
- [22] Indriana, Dina dan Nurtatio A, Pulung. 2015. *Peningkatan Kontras Menggunakan Metode Contrast Limited Adaptive Histogram Equalization Pada Citra Underwater*. [Internet]. Tersedia di: eprints.dinus.ac.id/16854/. (di akses 11 November 2018).