

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Tinjauan pustaka merupakan acuan utama pada penelitian ini sebagai perbandingan penelitian yang akan penulis lakukan berupa beberapa studi yang pernah dilakukan berkaitan dengan penelitian.

Penelitian yang dilakukan Dila Deswari, Hendrik, Derisma (2013) mengidentifikasi kematangan buah tomat yang dilakukan berdasarkan pengamatan visual secara langsung pada buah yang akan diklasifikasi berdasarkan ciri warna. Komputasi yang dilakukan menggunakan webcam sebagai pengolah citra dari gambar yang direkam (*image processing*). Pada proses identifikasi untuk menentukan nilai bobot dikelompokkan berdasarkan masak, muda, dan setengah masak. Dari hasil identifikasi menghasilkan tingkat keberhasilan dari metode backpropagation 71.76%. dari identifikasi yang telah dilakukan menghasilkan 3 output yaitu tomat masak 85%, muda 45%, dan setengah masak 85%.

Penelitian Sella Kusumaningtyas dan Rosa Andrie Asmara (2016) mengidentifikasi kematangan buah tomat menggunakan metode perceptron dengan pendukung aplikasi ini dibantu media webcam sebagai pengambilan gambar tomat yang akan menentukan Histogram warnanya untuk memperoleh informasi citra dan dapat mengetahui jenis kematangan buah tersebut. Tingkat keberhasilan identifikasi kematangan buah tomat yang menggunakan metode perceptron 43,33%. Dari hasil identifikasi diperoleh menghasilkan 3 output yaitu mentah 10%, setengah matang 6,66%, dan matang 26,66%.

Penelitian Muara Widyaningsih (2017) identifikasi terhadap buah apel terhadap pengujian dari metode (GLCM) dengan 0^0 dengan menggunakan Euclidian Distance. Mengidentifikasi terhadap 7 data uji dari 12 data latih, memberikan informasi semua data dapat dikenali dengan nilai kemiripan data dengan menggunakan Euclidian Distance, dapat menjadi tolak ukur pengenalan sebuah objek citra queri terhadap pengujian data.

Aan Aldi Anarus Manangsang (2018) mengenali identifikasi kondisi daun kelapa sawit pada arsitektur JST yang diperoleh terdiri dari 45000 node input layer, hidden layer berjumlah 90 yaitu 32 iterasi dan learning rate 0.04 serta minimum error 0.0001 dengan durasi 0:01:57 detik dan nilai MSE 0.0000994. menghasilkan akurasi data saat pengujian sebesar 75%.

Yuda Pemadi dan Murinto (2015) mengenali tekstur kulit mentimun dengan menggunakan metode ekstraksi ciri statistik mendapatkan hasil deteksi kematangan dengan mempunyai presentase keberhasilan. Identifikasi kematangan mentimun berdasarkan tekstur kulit mendapatkan hasil sebesar 75 % dari pengujian 2 output yang didapat yaitu matang sebesar 70% sedangkan mentimun belum matang 80%.

Untuk memudahkan membandingkan tinjauan pustaka dengan penelitian sebelumnya, maka dibuat tabel 2.1.

Tabel 2.1 Perbandingan Penelitian Sebelumnya

Penulis dan Tahun	Objek	Metode Ekstraksi ciri	Metode pengenalan pola	Hasil
Dila Deswari, Hendrik, Derisma (2013)	Buah tomat	Segmentasi citra	Backpropagation	Tingkat identifikasi kematangan buah tomat menggunakan metode backpropagation ini berhasil dengan tingkat keberhasilan identifikasi 71,76%
Sella Kusumaningtyas dan Rosa Andrie Asmara (2016)	Buah Tomat	Sigmentasi citra	Perceptron	Tingkat keberhasilan identifikasi kematangan buah tomat yang menggunakan metode perceptron 43,33%. Dari hasil identifikasi diperoleh menghasilkan 3 output yaitu mentah 10%, stengah matang 6,66%, dan matang 26,66%
Maura Widyaningsih (2016)	Buah Apel	Geometri	Gray Level Co-occurrence Matrix (GLCM).	Dari data pengujian terhadap 7 data dapat disimpulkan bahwa identifikasi terhadap buah apel mendekati kebenaran terhadap pengujian dari metode GLCM dengan sudut 0^0 dengan menggunakan Eucledian Distance

Tabel 2.1 Perbandingan Penelitian Sebelumnya

Penulis dan Tahun	Objek	Metode Ekstraksi ciri	Metode pengenalan pola	Hasil
Aan Aldi Anarus Manangsan (2018)	Daun kelapa sawit	Chromatocity diagram	Jaringan saraf tiruan backpropagation	Jenis penyakit tanaman, akurasi hasil identifikasi
Yuda Pemadi dan Murinto (2015)	Mentimum	Tekstur	Ekstraksi citi statistik	Kematangan mentimun dari aplikasi ini mempunyai presentase tingkat keberhasilan 75% dengan menggunakan metode ekstraksi ciri statistik dari tekstur kulit mentimun.
Penulis yang diajukan	Buah sawit	Segmentasi citra	Jaringan saraf tiruan backpropagation	Kematangan buah sawit.

2.2. Dasar Teori

2.2.1. Representasi Warna

Warna merupakan respon physiological dan intensitas yang berbeda. Persepsi warna dalam 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). Representasi warna ini terdiri dari tiga unsur utama yaitu merah (red), hijau (green),

dan biru (blue). Gabungan tiga warna ini membentuk warna-warna lainnya berdasarkan intensitas dari masing-masing warna tersebut dengan intensitas maksimal, dan warna hitam merupakan gabungan dari ketiga warna tersebut dengan intensitas minimal. Dalam aplikasi ini menggunakan model warna RGB. Tingkat RGB pola bit dikomposisikan dari tiga warna tersebut dan masing-masing warna mempunyai 28 atau 256 bit (0 - 255). Model warna RGB yang dapat dinyatakan dalam bentuk indeks warna RGB dengan cara menormalisasi setiap komponen warna dengan persamaan sebagai berikut (Dila Deswari, Hendrik, Derisma, 2013):

$$r = \frac{R}{R+G+B} \dots\dots\dots (2.2.3)$$

$$g = \frac{G}{R+G+B} \dots\dots\dots (2.2.4)$$

$$b = \frac{B}{R+G+B} \dots\dots\dots (2.2.5)$$

2.2.2. Pengolahan Citra Digital

Pengolahan citra adalah istilah umum untuk berbagai teknik keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah. Setiap foto dalam bentuk citra digital (misalkan berasal dari kamera digital) dapat diolah melalui perangkat lunak tertentu (Kadir, 2013). Pengolahan citra digital merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra dengan bantuan komputer. Secara umum tahapan pengolahan citra digital meliputi akuisisi citra, peningkatan kualitas citra, segmentasi citra, representasi dan uraian, pengenalan dan interpretasi.

2.2.3. Segmentasi

Menurut Tyagi (2018), segmentasi gambar adalah proses membagi gambar menjadi beberapa bagian yang berbeda. Setiap bagian menjadi homogen sesuai dengan fitur atau karakteristik yang terdefinisi dengan baik. Segmentasi gambar merupakan langkah penting dalam pemrosesan gambar, analisis gambar dan visi komputer. Keakuratan analisis gambar tergantung keakuratan segmentasi gambar.

Teknik segmentasi didasarkan pada dua sifat nilai intensitas area, yaitu diskontinuitas dan kesamaan. Teknik berbasis diskontinuitas membagi gambar berdasarkan perubahan intensitas yang tiba-tiba. Teknik berbasis kesamaan membagi gambar berdasarkan kesamaan menurut kriteria yang telah ditentukan.

2.2.4. Segmentasi *Thresholding*

Menurut Tyagi (2018), segmentasi berbasis *thresholding* membagi piksel berdasarkan tingkat intensitas atau tingkat abu-abu dari suatu gambar. Metode ini hanya bisa diterapkan pada objek dalam gambar yang memiliki perbedaan distribusi tingkat abu-abu. Metode ini dapat membagi gambar menjadi area latar belakang dan latar depan. Area latar depan (objek) memiliki piksel dengan tingkat intensitas yang lebih besar dibandingkan dengan piksel di latar belakang.

2.2.5. Global *Thresholding*

Menurut Tyagi (2018), global *thresholding* merupakan metode *thresholding* yang menggunakan nilai ambang yang sama untuk seluruh gambar. Akurasi segmentasi tergantung pada pemilihan nilai ambang yang sesuai. Nilai ambang dapat dihitung menggunakan metode intensity histogram. Untuk gambar dengan variasi piksel tingkat rendah maka nilai ambang adalah rerata nilai puncak

histogram latar depan dan latar belakang. Untuk gambar dengan variasi piksel tingkat tinggi maka nilai ambang dihitung dengan langkah :

1. Menentukan perkiraan nilai ambang awal.
2. Membagi area berdasarkan nilai ambang.
3. Menghitung nilai rerata tingkat abu-abu pada area latar depan dan belakang.
4. Menghitung nilai ambang yang merupakan rerata tingkat abu-abu pada area latar depan dan belakang.
5. Ulangi pembagian area berdasarkan nilai ambang sampai nilai ambang terakhir lebih kecil dari nilai ambang sebelumnya.

2.2.6. Variable Thresholding

Menurut Tyagi (2018), *variable thresholding* merupakan metode *thresholding* yang menggunakan nilai ambang bervariasi. Ada dua jenis metode *variable thresholding*, yaitu *local thresholding* dan *adaptive thresholding*. Metode *local thresholding* menggunakan nilai T yang tergantung pada lingkungan piksel yang sedang diproses. Metode *adaptive thresholding* menggunakan nilai T dari fungsi koordinat piksel.

2.2.7. Histogram Citra

Histogram citra merupakan diagram yang menunjukkan distribusi frekuensi kemunculan nilai intensitas piksel. Pada histogram, sumbu-x menyatakan nilai intensitas piksel sedangkan sumbu-y menyatakan frekuensi kemunculan intensitas piksel. Nilai intensitas setiap piksel pada citra truecolor 24-bit berkisar antara 0 s.d 255. Masing-masing komponen warna pada citra RGB dapat diekstrak dan

ditampilkan histogramnya. Berikut ini merupakan cara mengekstrak masing-masing komponen warna citra RGB dan cara menampilkan histogramnya.

2.2.8. Jaringan Syaraf Tiruan (JST)

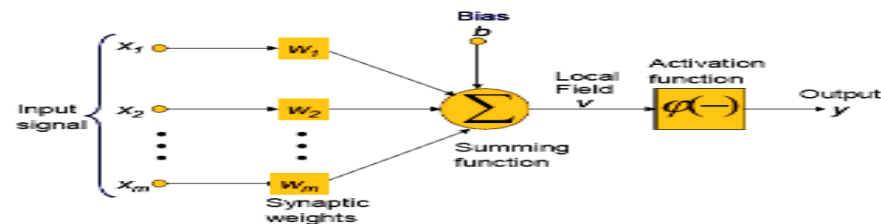
Menurut Fu (1994) sebuah jaringan syaraf tiruan merupakan simulasi dari otak biologis yang bertujuan untuk belajar mengenali pola data dan mensimulasikan proses belajar adaptif biologis, walau dalam skala yang sangat sederhana (Halim, 2012). Jaringan syaraf tiruan dibuat untuk bisa mengenali pola suatu data seperti kerja otak manusia karena jaringan syaraf tiruan adalah salah satu representasi otak manusia.

Pada jaringan syaraf tiruan, terdapat beberapa neuron-neuron yang kemudian akan di satukan dalam satu layer yang disebut dengan lapisan neuron. Neuron-neuron yang sudah disatukan akan dihubungkan dengan layer-layer yang lain. Jadi, semua informasi yang didapat dari neuron yang satu akan disampaikan dengan semua layer yang ada, proses penyampaian informasi akan dimulai dari lapisan *input* kemudian melewati lapisan *hidden* untuk menuju lapisan *output*.

Prinsip jaringan syaraf tiruan (Riadi. 2016) ditentukan oleh tiga elemen dasar model saraf, yaitu:

1. Satu set dari sinapsis, atau penghubung yang masing-masing digolongkan oleh bobot atau kekuatannya.
2. Sebuah penambah untuk menjumlahkan sinyal-sinyal input. Ditimbang dari kekuatan sinaptik masing-masing neuron.

3. Sebuah fungsi aktivasi untuk membatasi amplitudo output dari neuron.
Fungsi ini bertujuan membatasi jarak amplitudo yang diperbolehkan oleh sinyal output menjadi sebuah angka yang terbatas.



Gambar 2.1 Prinsip Dasar JST (Sumber: <https://media.neliti.com>)

Pada gambar 2.1, Y menerima masukan dari neuron x_1 , x_2 , dan x_3 , dengan bobot hubungan masing-masing adalah w_1 , w_2 , dan w_3 . Ketiga impuls neuron yang ada dijumlahkan dengan (persamaan 2.1)

$$\text{Net} = x_1w_1 + x_2w_2 + x_3w_3 \dots \dots \dots (2.1)$$

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y = f(\text{net})$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk mengubah bobot (Siang, 2004).

2.2.9. Backpropagation

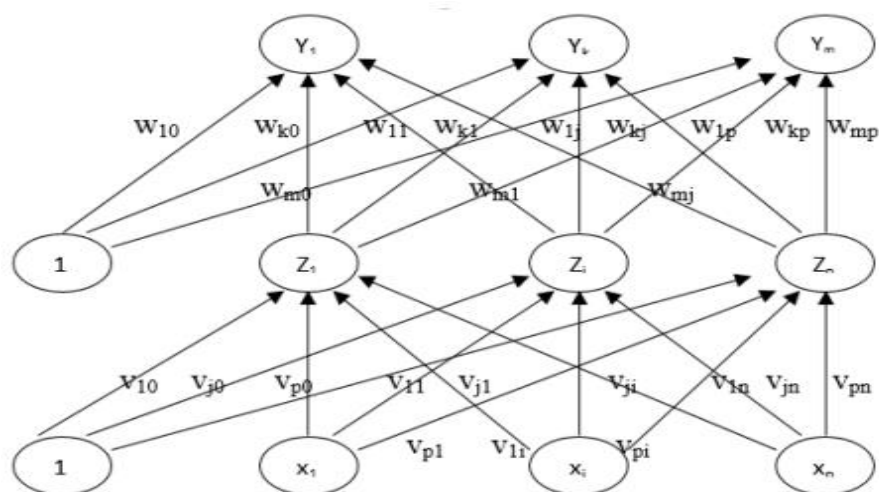
Salah satu metode pelatihan terawasi pada jaringan syaraf adalah metode *Backpropagation*. Algoritma *Backpropagation* merupakan sebuah algoritma yang dapat memperkecil tingkat suatu error dengan menyesuaikan bobot yang berdasarkan target serta output yang diharapkan. Secara umum, *Backpropagation* memiliki langkah-langkah utama diantaranya pengambilan input, penelusuran *error*, dan kemudian penyesuaian bobot. Dalam metode *backpropagation* algoritma

yang dikerjakan adalah inisialisasi bobot, komputasi *forward* dan *backpropagation* dan inisialisasi kondisi *stopping* berdasarkan nilai batas error atau jumlah batas *epoch*. Apa yang dimaksud dengan *epoch*? *Epoch* sendiri merupakan rangkaian langkah-langkah dari dalam pembelajaran ANN (*Artificial neural network*) atau dalam bahasa Indonesia disebut Jaringan saraf tiruan. Kemudian satu *epoch* dapat diartikan sebagai satu pembelajaran ANN itu sendiri (Fauzia, 2018).

Dalam algoritma *backpropagation* menggunakan lapisan-lapisan untuk menghasilkan sebuah keluaran atau *output*, yaitu:

1. Lapisan masukan (*layer input*)
2. Lapisan tersembunyi (*layer hidden*)
3. Lapisan keluaran (*layer output*)

Layar *input* tidak melakukan proses apapun, pada layer *input* hanya memasukkan ke layer *hidden*, layer *hidden* berfungsi sebagai ruang untuk menyesuaikan bobot yang untuk mendapatkan nilai yang mendekati target *output* yang diinginkan. Arsitektur jaringan syaraf tiruan *backpropagation* bisa dilihat pada gambar 2.3.



Gambar 2.2 Arsitektur JST Backpropagation(Sumber: jurnal.unpad.ac.id)

Penggunaan metode *backpropagation* memiliki 2 tahap yaitu tahap pelatihan atau tahap pembelajaran dimana pada tahap ini digunakan untuk melatih dengan berbagai data yang diberikan. Selanjutnya tahap pengujian atau penggunaan pada tahap ini dilakukan untuk menguji pada tahap pelatihan. Namun pada tahap pelatihan terdiri dari tiga langkah, yaitu:

1. Data dimasukkan ke input jaringan (*feedforward*)
2. Perhitungan dan propagasi balik dari error yang bersangkutan
3. Pembaharuan (*adjustment*) bobot bias

Pada fase *feedforward*, pola yang dimasukkan akan dihitung maju yang mulai dari lapisan *input* sampai lapisan *output* menggunakan fungsi aktivasi. Dalam pelatihan ini fungsi aktivasi yang digunakan adalah fungsi sigmoid biner, fungsi ini memiliki range nilai 0 sampai 1 karena. Dalam fase *backpropagation* (langkah kedua) menghitung nilai kesalahan, kemudian akan di propagasikan mundur mulai dari garis yang berhubungan langsung dengan unit-unit di layer *output*. Fase ke tiga (*adjustment*) bertujuan untuk memodifikasi bobot dengan tujuan menurunkan kesalahan yang terjadi. Fase-fase ini akan terus berlanjut dan hanya akan berhenti jika kondisi untuk berhenti terpenuhi.

Fungsi sigmoid biner

$$f_1(x) = 1/(1+e^{-x}) \dots\dots\dots (2.2)$$

dengan turunan

$$f_1'(x) = f_1(x)(1-f_1(x)) \dots\dots\dots (2.3)$$

Algoritma *Backpropagation*:

Step 0 : Inisialisasi bobot dan bias baik bobot maupun bias dapat diset dengan sembarang angka (acak) dan biasanya angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif)

Step 1 : Jika *stopping* condition masih belum terpenuhi, jalankan step 2-9.

Step 2 : Untuk setiap data training, lakukan step 3-8.

Umpan maju (feedforward)

Step 3 : Setiap unit yang dipakai di sini adalah *input* training data yang sudah di skalakan.

Step 4 : Setiap *hidden* unit ($Z_j, j=1, \dots, p$) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya dengan (persamaan 2.4)

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (2.4)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari *hidden* unit yang bersangkutan, melalui (persamaan 2.5),

$$z_j = f(z_{in_j}) \dots\dots\dots (2.5)$$

lalu mengirim sinyal *output* ini ke seluruh unit pada unit output

Keterangan:

x = *input* vektor pelatihan $x = (x_1, \dots, x_i, \dots, x_n)$

v_{0j} = bias pada unit tersembunyi j

Z_j = unit tersembunyi j *input* jaringan Z_j disimbolkan dengan Z_{in_j}

Σ = sigma

n = Jumlah data latih

Step 5 : Setiap unit *output* ($Y_k, k=1, \dots, m$) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya dengan (persamaan 2.6),

$$y_{in_k} = w_{ok} + \sum_{j=1}^p z_j w_{jk} \dots\dots\dots (2.6)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari unit *output* yang bersangkutan, melalui (persamaan 2.7):

$$y_k = f(y_{in_k}) \dots\dots\dots (2.7)$$

Keterangan:

Y_k = unit *output input* jaringan ke Y_k disimbolkan dengan y_{in_k}

w_{ok} = nilai bias

Propagasi balik error (backpropagation of error)

Step 6 : Setiap unit *output* (Y_k , $k=1, \dots, m$) menerima suatu target (*output* yang diharapkan) yang akan dibandingkan dengan *output* yang dihasilkan dengan (persamaan 2.8).

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \dots\dots\dots (2.8)$$

Faktor δ_k ini digunakan untuk menghitung koreksi error (ΔW_{jk}) yang nantinya akan dipakai untuk memperbaharui W_{jk} , dengan (persamaan 2.9):

$$\Delta w_{jk} = \alpha \delta_k z_j \dots\dots\dots (2.9)$$

Selain itu juga dihitung koreksi bias ΔW_{ok} yang nantinya akan dipakai untuk memperbaharui W_{ok} , dengan (persamaan 2.10):

$$\Delta w_{ok} = \alpha \delta_k \dots\dots\dots (2.10)$$

Faktor δ_k ini kemudian dikirimkan ke layer di depannya

Keterangan:

y = *output neuron*

t = target data latih

δ_k = informasi tentang kesalahan pada unit Y_k yang disebarkan kembali ke unit tersembunyi.

α = *learning rate*

Δ = delta

ΔW_{jk} = koreksi error

Step 7 : Setiap *hidden* unit ($Z_j, j=1, \dots, p$) menjumlah *input* delta (yang dikirim dari layer pada step 6) yang sudah berbobot, (persamaan 2.11):

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \dots\dots\dots (2.11)$$

Hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi error δ_j , (persamaan 2.12):

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \dots\dots\dots (2.12)$$

Faktor δ_j ini digunakan untuk menghitung koreksi error (ΔV_{ij}) yang nantinya akan dipakai untuk memperbaharui V_{ij} , (persamaan 2.13):

$$\Delta v_{ij} = \alpha \delta_j x_i \dots\dots\dots (2.13)$$

Selain itu juga dihitung koreksi bias ΔV_{0j} yang nantinya akan dipakai untuk memperbaharui V_{0j} , (persamaan 2.14):

$$\Delta v_{0j} = \alpha \delta_j \dots\dots\dots (2.14)$$

Keterangan:

δ_j = informasi tentang kesalahan dari lapisan *output* ke unit tersembunyi Z_j

X_i = unit *input* I

ΔV_{ij} = koreksi error bias pada unit tersembunyi j

Pembaharuan bobot dan bias:

Step 8 : a. Setiap unit output ($Y_k, k=1, \dots, m$) akan memperbaharui bias dan bobotnya dengan setiap hidden unit. (persamaan 2.15):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots\dots\dots (2.15)$$

b. Demikian pula untuk setiap hidden unit akan memperbaharui bias dan bobotnya dengan setiap unit input. (persamaan 2.16):

$$v_{ij}(\text{baru}) = v_{ij} + \Delta v_{ij} \dots\dots\dots (2.16)$$

Keterangan:

v_{0j} = bias pada unit tersembunyi j

Step 9 : Memeriksa stopping condition, jika stop condition telah terpenuhi, maka pelatihan jaringan dapat dihentikan.

Stopping Condition

Untuk menentukan stopping condition terdapat dua cara yang biasa dipakai, yaitu:

a. Membatasi iterasi yang ingin dilakukan.

Misalnya jaringan akan dilatih sampai iterasi yang ke-500. Yang dimaksud dengan satu iterasi adalah perulangan step 3 sampai step 8 untuk semua training data yang ada.

b. Membatasi error.

Misalnya menentukan besar Mean Square Error antara output yang dikehendaki dan output yang dihasilkan oleh jaringan. Jika terdapat sebanyak m training data, maka untuk menghitung Mean Square Error digunakan (persamaan 2.17)::

$$\text{MSE} = 0,5X\{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \dots + (t_{km} - y_{km})^2\} \dots (2.17)$$

Keterangan:

MSE = Mean Square Error

$t = \text{output vector target } t = (t_1, \dots, t_k, \dots, t_m)$

$Y_k = \text{unit output}$

Tahap pengujian & Penggunaan

Mengimplementasikan metode backpropagation pada tahap pengujian sama seperti proses belajar, tetapi hanya pada bagian umpan majunya saja :

Step 0 : Inisialisasi bobot sesuai dengan bobot yang telah dihasilkan pada proses pelatihan di atas.

Step 1 : Untuk setiap input, lakukan step 2-4.

Step 2 : Untuk setiap input $i=1, \dots, n$ skalakan bilangan dalam range fungsi aktivasi seperti yang dilakukan pada proses pelatihan di atas.

Step 3 : Untuk $j=1, \dots, p$:

$$z_{in_j} = v_{oj} + \sum_{i=1}^n x_i v_{ij} \dots \dots \dots (2.18)$$

$$z_j = f(z_{in_j}) \dots \dots \dots (2.19)$$

Step 4 : Untuk $k=1, \dots, m$:

$$y_{in_k} = w_{ok} + \sum_{j=1}^p z_j w_{jk} \dots \dots \dots (2.20)$$

$$y_k = f(y_{in_k}) \dots \dots \dots (2.21)$$

2.2.10. Akurasi dan Error Rate

Pengukuran terhadap kinerja suatu sistem klasifikasi merupakan hal yang penting. Kinerja sistem klasifikasi menggambarkan seberapa baik sistem dalam mengklasifikasikan data. *Confusion matrix* merupakan salah satu metode yang

dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya.

Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, *True Positive* (TP) merupakan data positif yang terdeteksi benar. *False Negative* (FN) merupakan kebalikan dari *True Positive*, sehingga data positif, namun terdeteksi sebagai data negatif (Achmad, 2017).

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

Tabel 2.2 *Confusion Matrix*

Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) dapat diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Sedangkan *Error Rate* adalah kasus yang diidentifikasi salah dengan sejumlah semua kasus. Untuk memperoleh akurasi dapat menggunakan rumus berikut:

$$\text{Akurasi} \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots (2.22)$$

$$\text{Error Rate} \frac{FP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots (2.23)$$

2.2.11. Matlab

Matlab adalah sebuah bahasa dengan kinerja tinggi untuk komputasi masalah teknik. MATLAB mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar (Febri, 2016). Penggunaan Matlab meliputi

1. Matematika dan komputasi
2. Pembentukan algoritma
3. Akusisi data
4. Pemodelan, simulasi, dan pembuatan *prototype*
5. Analisa data, explorasi, dan visualisasi
6. Grafik keilmuan dan bidang rekayasa

Nama MATLAB merupakan singkatan dari *matrix laboratory*. Dalam lingkungan perguruan tinggi teknik, MATLAB merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tinggi, pengembangan dan analisisnya. Fitur-fitur MATLAB sudah banyak dikembangkan, dan lebih kita kenal dengan nama *toolbox*. Sangat penting bagi seorang pengguna MATLAB, *toolbox* mana yang mendukung untuk *learn* dan *apply technology* yang sedang dipelajarinya. *Toolbox* ini merupakan kumpulan dari fungsi-fungsi MATLAB (*M-files*) yang telah dikembangkan ke suatu lingkungan kerja MATLAB untuk memecahkan masalah dalam kelas particular. Area-area yang sudah bisa dipecahkan dengan toolbox saat

ini meliputi pengolahan sinyal, system kontrol, *neural networks*, *fuzzy logic*, *wavelets*, dan lain-lain (Febri, 2016).

Sebagai sebuah sistem, MATLAB tersusun dari 5 bagian utama:

1. **Development Environment**, merupakan sekumpulan perangkat dan fasilitas yang membantu kita untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah **Graphical User Interfaces (GUI)**. Termasuk didalamnya adalah *MATLAB desktop* dan *Command Window*, *Command History*, sebuah editor dan *debugger*, dan *browsers* untuk melihat *help*, *workspace*, *files*, dan *search path*.
2. **Matlab Mathematical Function Library**, merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: *sum*, *sin*, *cos*, dan *complex arithmetic*, sampai dengan fungsi-fungsi yang lebih kompleks seperti *matrix inverse*, *matrix eigenvalues*, *Bessel functions*, dan *fast Fourier transforms*.
3. **Matlab Language**, merupakan suatu *high-level matrix/array language* dengan *control flow statements*, *functions*, *data structures*, *input/output*, dan *fitur-fitur object-oriented programming*. Ini memungkinkan bagi kita untuk melakukan kedua hal baik “pemrograman dalam lingkup sederhana” untuk mendapatkan hasil yang cepat, dan “pemrograman dalam lingkup yang lebih besar” .
4. **Graphics**, matlab memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatu grafik. Didalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga

dimensi, *image processing*, *animation*, dan *presentation graphics*. Ini juga melibatkan fungsi level rendah yang memungkinkan bagi kita untuk membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan graphical user interfaces pada aplikasi matlab.

5. **Matlab Application Program Interface (API)**, merupakan suatu *library* C yang memungkinkan program yang telah kita tulis dalam bahasa dan *Fortran* mampu berinteraksi dengan matlab.

2.2.12. Kelapa Sawit

Tanaman kelapa sawit (*Elaeis Quineensis* Jacq) berasal dari Nigeria, Afrika Barat. Meskipun demikian kelapa sawit hidup subur diluar daerah asalnya. Seperti Malaysia, Thailand, Papua nugini. Bahkan mampu memberi hasil produksi perhektar yang lebih tinggi (Fauzi, 2002). Pendapat ini diperkuat oleh suyatno (1994) bahwa kelapa sawit saat ini telah berkembang pesat. Di Asia Tenggara khususnya Indonesia dan Malaysia justru bukan di Afrika Barat atau Amerika yang dianggap daerah asalnya. Tanaman Kelapa Sawit umumnya memiliki batang yang tidak bercabang. Pada pertumbuhan awal setelah fase muda (seedling) terjadi pembentukan batang yang melebar tanpa terjadi pemanjangan internodia (ruas). Titik tumbuh batang kelapa saawit terletak di pucuk batang, terbenam didalam tajuk daun. Di batang terdapat pangkal pelepah-pelepah dain uang melekat kuku (Sunarko, 2008).

Tumbuhan ini termasuk tumbuhan pohon, tingginya bisa mencapai 24 meter. Bunga dan buahnya berupa tandan, serta bercabang banyak. Buahnya kecil

dan apabila masak, berwarna merah kehitaman. Daging buahnya padat. Daging dan kulit buahnya mengandung minyak. Minyaknya sering digunakan sebagai bahan minyak goreng, sabun, dan lilin. Sedangkan ampasnya dimanfaatkan untuk makanan ternak, khususnya sebagai salah satu bahan pembuatan makanan ayam. Tempurungnya digunakan sebagai bahan bakar dan arang.

Warna buah sawit bervariasi, mulai dari hitam, ungu, hingga merah. Ini tergantung dari bibit yang dipakai. Buahnya ini bergerombol dalam tandan yang muncul dari masing-masing pelepah. Buah sawit ini terdiri dari tiga lapisan, yakni: Eksoskarp adalah bagian kulit buah yang warnanya kemerahan dan licin. Mesoskarp yang merupakan serabut buah. Dan Endoskarp adalah cangkang pelindung inti. Inti sawit ini merupakan endosperm dan embrio dengan kandungan minyak inti berkualitas tinggi. Dan adapun jenis-jenis buah pada tahap perubahan warna untuk menentukan tingkat-tingkat kematangannya yaitu :

A. Mentah/belum matang



Gambar 2.3 Buah Mentah

B. Mengkal/setengah matang



Gambar 2.4 Buaha Mengkal

C. Matang



Gambar 2.5 Buah Matang