

BAB 2

LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam penelitiannya (Nurmila, Sugiharto, & Sarwoko, 2010) mengenai pengenalan pola karakter huruf Jawa menggunakan algoritma *Backpropagation Neural Network* menunjukkan bahwa masing-masing bagian sampel memiliki karakteristik *Backpropagation* yang berbeda-beda untuk mendapatkan hasil pelatihan yang paling baik pada pengenalan karakter huruf Jawa dan rata-rata keakuratan BPNN dalam mengenali pola karakter huruf Jawa adalah sebesar 99.563% untuk data sampel berupa data pelatihan, 61.359% untuk data sampel di luar dan pelatihan, dan 75% untuk data sampel data pelatihan dan di luar data pelatihan.

Dalam penelitiannya (Alqurni & Muljono, 2016) mengenai pengenalan tanda tangan menggunakan metode *Perceptron* dan *Backpropagation* didapatkan hasil dari 400 data latih dan 100 data uji adalah keakurasian pelatihan sebesar 100% berdasarkan hasil evaluasi menggunakan data latih dan akurasi uji sebesar 86% dengan banyak pola benar yang dapat dikenali sejumlah 86 citra. Sedangkan untuk metode *Perceptron* mendapatkan hasil akurasi training 100% berdasarkan hasil evaluasi menggunakan data latih dan akurasi testing 76% dengan 76 citra yang dapat dikenali. Jadi berdasarkan hasil tersebut, maka dapat diambil kesimpulan bahwa penggunaan metode jaringan syaraf tiruan *Backpropagation* lebih baik dibandingkan dengan metode *Perceptron* yang digunakan dalam pengenalan tanda tangan.

Dalam penelitiannya (W & Al Farqi, 2015) membuat sistem pengenalan karakter aksara caraka Madura yang telah dilakukan pengujian dengan arsitektur Jaringan Syaraf Tiruan *Backpropagation* dengan jumlah data *training* sebanyak 400 yang mewakili 20 karakter, jumlah *input* 256 *neuron*, *hidden layer* 3 *neuron*,

output layer 1 neuron, dan *learning rate* 0,2. Hasil pengujian ini menunjukkan nilai keakuratan karakter sebesar 78,333%. Karena banyak karakter yang memiliki ciri mirip, maka mereka menyarankan agar ditambah data latih agar pengenalan lebih akurat.

Dalam penelitiannya (Hidayatno, Isnanto, & Buana, 2008) mengenai indentifikasi tanda tangan menggunakan *backpropagation*. Pada penelitiannya, dilakukan ekstraksi ciri pada tanda tangan menjadi angka biner. Angka tersebut kemudian akan digunakan menjadi input dalam node *backpropagation* yang akan dilatih hingga mendapatkan nilai yang sudah ditargetkan. Hasil kesimpulan yang diperoleh *backpropagation* dalam mengidentifikasi tanda tangan memiliki tingkat ketepatan 88%.

Dalam penelitiannya (Sukmawati & Pujiyanta, 2014) mengenai deteksi penyakit tulang menggunakan *backpropagation*, terkenal dengan baik pada galat 0.5, *learning rate* (α) = 0.1 dan 42 *neuron hidden*, fungsi aktivasi yang dipakai adalah fungsi *logsig (sigmoid biner)*, dan metode pelatihan yang dipakai adalah metode *trial and error*, didapatkan hasil akurasi keberanaran pengujian aplikasi ini adalah 90%.

Selanjutnya Dalam penelitiannya (Hartadi, Santoso, & Hidayatno, 2011) mengenai deteksi kanker payudara pada mammogram, GLCM digunakan untuk mencirikan tekstur massa tumor, normal, serta gugus mikroklasifikasi pada area mammogram. Untuk memperbaiki kualitas citra dilakukan tahap pengurangan derau, perbaikan kontras dan operasi morfologi. Setelah tahap pra pressing dilakukan proses ekstraksi ciri GLCM. Pada penelitian ini tingkat keberhasilan mencapai 86%.

Dalam melakukan penelitiannya (Astuti, 2008) untuk pengklasifikasian motif ppla batik menggunakan metode representasi bentuk nilai ekstraksi ciri yang telah diterapkan, *eccentricity* yang merupakan rasio dari perbandingan antara panjang *minor axis* dengan *major axis* dan *compactness* yang merupakan suatu obyek yang dapat diukur melalui analisis faktor bentuk tak berdimensi dengan

ketidaksamaan *isoperimetic*. *Compactness* dapat dihitung dengan menggunakan area dan perimeternya. Dan keduanya telah diterapkan dalam melakukan penelitian untuk pengenalan bentuk geometri benda. Program yang telah dibuatnya adalah untuk mengidentifikasi bulat, persegi dan segitiga dengan nilai rasio.

Dalam penelitiannya (Raheja, Das, & Chaudhary, 2011) mengenai pendeteksi ujung jari dari tangan manusia secara cepat. Mereka melakukan segmentasi tangan dengan berdasarkan *hue*, *saturation* dan *value* (HSV) dan akan didapatkan sebuah gambar biner. Kemudian dilanjutkan dengan pendeteksi pergelangan tangan dengan intensitas warna yang telah dihitung. Gambar akan dipotong dari pergelangan tangan sehingga hanya akan didapatkan gambar tangan. Ujung jari akan dideteksi dari pencarian gambar yang telah dipotong. Hasil dari penelitian ini adalah sistem sudah siap digunakan untuk digabungkan dengan implementasi seperti mengendalikan robot menggunakan *gesture* tangan.

Dalam meninjau pada kelima jurnal diatas maka penulis akan melakukan penelitian yang mengacu pada kedelapan jurnal tersebut. Penulis ingin menerapkan metode *backpropagation* pada sebuah sistem untuk mengklasifikasi jenis buah *pear*, GLCM untuk ekstraksi ciri tekstur, *eccentricity* dan *compactness* untuk ekstraksi ciri bentuk dan HSV untuk ekstraksi ciri warna.

2.2. Landasan Teori

Untuk memperkuat penelitian mengenai pengklasifikasian jenis buah *pear*, penulis menggunakan beberapa landasan teori terkait kasus tersebut.

2.2.1. Pear

Menurut (Adiyanto, 2009) pir merupakan tanaman yang berasal dari Eropa Barat, Afrika Utara, dan Asia Timur. Pir juga merupakan tanaman yang tahan terhadap suhu dingin dan dapat tumbuh di daerah pesisir ataupun daerah yang

bertemperatur sejuk. Pir juga kaya berbagai macam vitamin antara lain vitamin A, B1, B2, C, E, K. *Pear* memiliki banyak jenis, namun pada penelitian ini digunakan 4 jenis yaitu *pear jenis packham, singo, sweet* dan *xianglie*.

a. *Pear Packham*



Gambar 2.1 *Pear Packham*

Gambar 2.1 menunjukkan buah *pear packham*. *Pear Packham* yang berasal dari Australia adalah jenis buah *pear* golongan hijau. Saat di makan kondisi daging buah *pear* ini keras, terasa manis mirip dengan jambu bengkak atau jambu mutiara. Karakteristik buah ini adalah berwarna hijau, kulit agak kasar dikelilingi spot warna coklat alami, tekstur keras lagi renyah.

b. *Pear Singo*

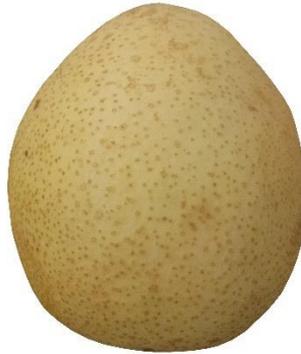


Gambar 2.2 *Pear Singo*

Gambar 2.2 adalah gambar dari *Pear Singo*. *Singo Pear* berasal dari negara Cina, bentuknya yang menyerupai buah apel ini berwarna coklat, dengan tekstur yang renyah. Buah ini juga memiliki kandungan serat yang sangat tinggi, dan dapat mengobati gangguan pencernaan,

mengatasi sembelit, juga mencegah terjadinya kanker kolon (kanker kolon dan rektum adalah kanker yang menyerang usus besar dan rektum).

c. *Pear Sweet*



Gambar 2.3 *Pear Sweet*

Gambar 2.3 menunjukkan *pear* jenis *Sweet*. *Pear Sweet* merupakan jenis *pear* yang berasal dari negara Cina. Sweet pear atau yang biasa disebut pear madu ini memiliki tekstur yang agak keras, namun untuk rasa, pear ini sangat manis, pear ini salah satu jenis buah yang mengandung cukup banyak vitamin c dan k.

d. *Pear Xianglie*



Gambar 2.4 *Pear Xianglie*

Gambar 2.4 menunjukkan buah *pear Xianglie*. Berasal dari negara china, pear xiang lie atau yang biasa disebut dengan pear hijau ini adalah terbaik dikelasnya karna dikenal mengandung banyak air pada daging

buahnya, memiliki rasa yang manis, tekstur daging yang lembut, dan kadar air yang melimpah.

2.2.2. Preprocessing

Menurut (Anggraini, 2014) *preprocessing* merupakan suatu proses pengolahan citra untuk mendapatkan informasi dari citra dan menghasilkan citra yang diinginkan. Citra harus diproses terlebih dahulu sebelum masuk ke proses klasifikasi menggunakan Jaringan Syaraf Tiruan.

2.2.3. Resize

Menurut (Styawan & Supriyanto, 2015) *resize* merupakan perubahan resolusi citra digital dari yang sebelumnya $N \times M$ diubah menjadi $N' \times M'$. Tujuan dari tahapan ini adalah untuk menyeragamkan ukuran dari semua citra digital yang ada selama proses pengolahan citra.

2.2.4. Grayscale

Menurut (Santi, 2011) *grayscale* merupakan proses awal dalam *image processing* yaitu mengubah citra berwarna menjadi citra *grayscale*. Sebuah citra terdiri dari tiga *layer* matrik warna yaitu *Red*, *Green* dan *Blue*. Proses perhitungan menggunakan tiga *layer* matrik tersebut akan dilakukan sebanyak tiga kali dengan perhitungan yang sama, untuk menyederhanakan perhitungan matrik, ketiga *layer* matrik diubah menjadi satu *layer* yang merupakan sebuah matrik *grayscale*.

Citra warna dapat diubah menjadi citra *grayscale* dengan cara mengalikan elemen warna Merah (R), Hijau (G) dan Biru (B) dengan persamaan rumus sebagai berikut:

$$grayscale = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad [2.1]$$

2.2.5. Thresholding

Citra yang telah melalui tahap *grayscale* kemudian akan masuk kedalam proses *thresholding*. *Thresholding* atau pengembangan akan menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih. Setiap pixel pada citra dipetakan menjadi dua nilai yaitu 1 dan 0. Secara umum fungsi pengembangan citra *grayscale* untuk menghasilkan citra biner adalah sebagai berikut (Munir, 2004).

$$g(x, y) = \begin{cases} 1 & \text{if } fo(x, y) \geq T \\ 0 & \text{if } fo(x, y) \leq T \end{cases} \quad [2.2]$$

Keterangan:

$g(x, y)$: citra biner

$fo(x, y)$: citra *grayscale*

T : nilai ambang batas penentu

2.2.6. Complement

Citra negatif merupakan citra yang nilai pikselnya berkebalikan dengan citra aslinya. Untuk citra *grayscale 8-bit*, apabila citra asli disimbolkan dengan I , maka negatif dari citra tersebut adalah $I' = 255 - I$. Contoh perhitungan nilai piksel dari citra negatif ditunjukkan pada Gambar 2.5.

| I | | |
|-----|-----|-----|
| 0 | 200 | 150 |
| 42 | 50 | 255 |
| 100 | 10 | 74 |

komplemen

| I' = 255-I | | |
|------------|-----|-----|
| 255 | 55 | 105 |
| 213 | 205 | 0 |
| 155 | 245 | 181 |

Gambar 2.5 Contoh perhitungan nilai piksel dari citra negatif
Dikutip dari : <https://pemrogramanmatlab.com/tag/complement-image/>

2.2.7. Compactness dan Eccentricity

Compactness merupakan sebuah metode untuk mencari ratio antara luas dari objek gambar dengan luas lingkaran yang mengelilingi objek tertentu. Adapun rumus metode *compactness* adalah sebagai berikut (Russ, 2011)

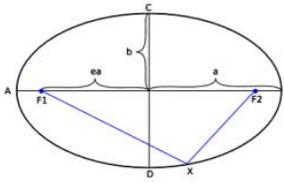
$$Compactness = \frac{\sqrt{\left(\frac{4}{\pi}\right)Area}}{MaxDiameter} \quad [2.3]$$

Keterangan:

Max Diameter = *Major Axis*

Area = Luas obyek gambar

Eccentricity merupakan nilai rasio perbandingan antara nilai *major axis* dengan nilai *minor axis* suatu obyek. *Eccentricity* memiliki rentang nilai antara 0 hingga 1. Obyek yang berbentuk memanjang/ mendekati bentuk garis lurus, nilai *eccentricity* mendekati angka 1, sedangkan obyek yang berbentuk bulat/lingkaran memiliki nilai *eccentricity* mendekati angka 0 (Felixiana, 2015). Penghitungan dari *eccentricity* diilustrasikan pada Gambar 2.6. dan dapat dilihat pada persamaan [2.4].



Gambar 2.6 Ilustrasi Eccentricity

$$e = \sqrt{1 - \frac{b^2}{a^2}} \quad [2.4]$$

Keterangan: e = eccentricity, a = Major axis, b = Minor axis

2.2.8. Tekstur

Dalam penelitiannya (Budiarso, 2010) menyatakan bahwa tekstur (*Textures*) adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat tersebut dapat berulang dalam daerah tersebut. Pengertian dari tekstur dalam hal ini adalah keteraturan pola-pola tertentu yang terbentuk dari susunan pixel-pixel dalam citra. Suatu permukaan dikatakan mempunyai informasi tekstur, jika luasannya diperbesar tanpa mengubah skala, maka sifat-sifat permukaan hasil perluasan mempunyai sifat kemiripan dengan permukaan asalnya. Syarat terbentuknya tekstur.

Adanya pola-pola primitif yang terdiri dari satu atau lebih piksel. Bentuk-bentuk pola primitif ini dapat berupa titik, garis, luasan dan sebagainya yang merupakan elemen dasar dari suatu bentuk. Ini artinya bahwa tekstur harus mempunyai elemen tekstur di dalamnya (disebut *texel* atau *texton*), meskipun elemen tekstur tersebut hanya terdiri dari sebuah pixel.

Pola-pola primitif tadi muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi karakteristik pengulangannya. Ini berarti bahwa harus ada seperangkat aturan yang dapat menjelaskan bagaimana elemen-elemen tekstur ini muncul secara berulang-ulang.

Dengan adanya dua syarat tersebut maka dapat dikatakan bahwa tekstur suatu permukaan dapat dibentuk melalui aturan-aturan yang berlaku atau sebaliknya, yaitu suatu permukaan dapat dianalisis teksturnya dengan aturan yang sama untuk diperbandingkan satu sama lain, atau untuk keperluan interpretasi keperluan citra digital.

2.2.9. GLCM (*Gray Level Co-Occurrence Matrix*)

Salah satu metode yang akan digunakan pada penelitian ini untuk mendeteksi tekstur adalah GLCM (*Gray Level Co-Occurrence Matrix*). GLCM merupakan metode statistika untuk membentuk fitur/ciri yang tidak didasarkan pada nilai piksel semata dan hubungan ketetanggaan piksel (Kadir & Susanto, 2013).

Pada pernyataan (Eskaprianda, Isnanto, & Santoso, 2011) bahwa langkah-langkah untuk membuat GLCM simetris ternormalisasi secara berurutan adalah sebagai berikut.

1. Membuat area kerja matriks.
2. Menentukan hubungan spasial antara piksel referensi dengan piksel tetangga, berapa nilai sudut θ dan jarak d .
3. Menghitung jumlah kookurensi dan mengisikannya pada area kerja.
4. Menjumlahkan matriks kookurensi dengan transposenya untuk menjadikannya simetris.
5. Normalisasi matriks untuk mengubahnya ke bentuk probabilitas.

Contoh bagaimana cara membentuk GLCM:

Misal ada citra dengan ukuran 4 x 4 pixel, 2 bit (Gambar 2.7).

| | | | |
|---|---|---|---|
| 0 | 2 | 2 | 1 |
| 0 | 3 | 3 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 2 | 1 | 0 |

Gambar 2.7 Contoh matrix citra

Karena matriks contoh memiliki empat aras keabuan, maka jumlah nilai piksel tetangga dan nilai piksel referensi pada area kerja matriks berjumlah empat. Berikut adalah area kerja matriks (Gambar 2.8).

| Nilai piksel tetangga \ Nilai piksel referensi | 0 | 1 | 2 | 3 |
|--|-----|-----|-----|-----|
| 0 | 0,0 | 0,1 | 0,2 | 0,3 |
| 1 | 1,0 | 1,1 | 1,2 | 1,3 |
| 2 | 2,0 | 2,1 | 2,2 | 2,3 |
| 3 | 3,0 | 3,1 | 3,2 | 3,3 |

Gambar 2.8 Area kerja matriks

Jika ditetapkan jaraknya 1 piksel dan arahnya adalah 0, 45, 90, dan 135 derajat, maka GLCM nya adalah sebagai berikut:

Untuk arah 0 derajat (Gambar 2.9):

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 2 | 2 | 1 | 0 |
| 2 | 0 | 2 | 1 | 0 |
| 3 | 1 | 0 | 0 | 1 |

Gambar 2.9 GLCM arah nol derajat dengan jarak 1 piksel

Setelah mendapatkan GLCM seperti di atas (Gambar 2.9), selanjutnya menambahkan hasil matriks kookurensi yang telah didapat dengan matriks *transposenya* untuk menjadikannya simetris terhadap sumbu diagonal (Gambar 2.10).

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 0 & 1 \\ 0 & 2 & 2 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 1 & 2 \\ 2 & 4 & 3 & 0 \\ 1 & 3 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}$$

$I \quad + \quad I^T = \quad I \text{ simetris}$

Gambar 2.10 GLCM simetris

Matriks yang telah simetris selanjutnya dinormalisasikan, elemennya dinyatakan dengan probabilitas. Nilai elemen untuk masing-masing sel dibagi dengan jumlah seluruh elemen spasial (Gambar 2.11).

| | | | |
|--------|--------|--------|--------|
| 0 | 0.0833 | 0.0417 | 0.0833 |
| 0.0833 | 0.1667 | 0.125 | 0 |
| 0.0417 | 0.125 | 0.0833 | 0 |
| 0.0833 | 0 | 0 | 0.0833 |

Gambar 2.11 Matriks kookurensi simetrik ternormalisasi

Setelah mendapatkan GLCM, kemudian dihitung berdasarkan fitur yang terdapat pada GLCM. Pada penelitian ini, penulis menggunakan empat fitur, antara lain:

1. Kontras (*Contrast*)

Informasi statistik ini menunjukkan ukuran penyebaran (momen inersia) elemen-elemen matriks citra. Jika letaknya jauh dari diagonal utama, nilai kekontrasan besar. Secara visual, nilai kekontrasan adalah ukuran variasi antar derajat keabuan suatu daerah citra.

$$Contrast = \sum_{i,j} P(i,j)(i-j)^2 \quad [2.5]$$

Keterangan:

$P(i,j)$ = nilai probabilitas matriks GLCM

i = nilai piksel referensi

j = nilai piksel tetangganya

2. Korelasi (*Correlation*)

Menunjukkan ukuran ketergantungan linear derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra.

$$Correlation = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j} \quad [2.6]$$

Keterangan:

$P(i,j)$ = nilai probabilitas matriks GLCM

i = nilai piksel referensi

- j = nilai piksel tetangganya
- μ_i = mean (i)
- μ_j = mean (j)
- σ_i = Standar deviasi (i)
- σ_j = Standar deviasi (j)

3. Energi (*Energy*)

Menyatakan distribusi intensitas piksel terhadap jangkauan aras keabuan.

$$Energy = \sum_{i,j} P(i,j)^2 \quad [2.7]$$

Keterangan:

- $P(i,j)$ = nilai probabilitas matriks GLCM
- i = nilai piksel referensi
- j = nilai piksel tetangganya

4. Homogenitas (*Homogeneity*)

Digunakan untuk mengukur kehomogenan variasi intensitas citra. Nilai homogenitas akan semakin membesar bila variasi intensitas dalam citra mengecil.

$$Homogeneity = \sum_{i,j} \frac{P(i,j)}{1+|i-j|} \quad [2.8]$$

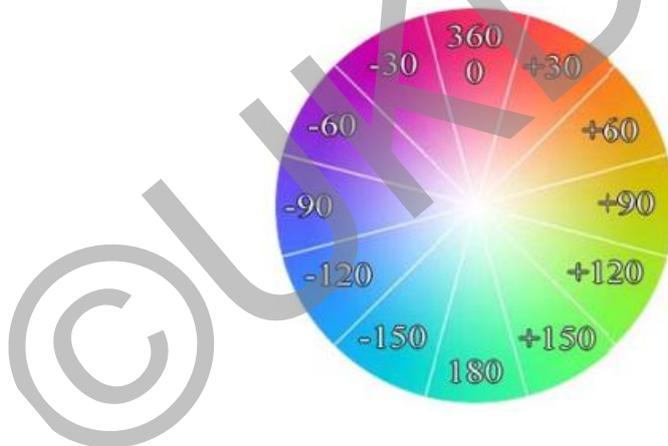
Keterangan:

- $P(i,j)$ = nilai probabilitas matriks GLCM
- i = nilai piksel referensi
- j = nilai piksel tetangganya

2.2.10. HSV (*Hue, Saturation, Value*)

Segmentasi warna merupakan pemisahan segmen dalam suatu citra berdasarkan warna yang terkandung dalam citra. Pada penelitian ini penulis menggunakan segmentasi warna dengan metode deteksi warna HSV. Ruang lingkup warna HSV terdiri dari 3 elemen yaitu *Hue* mewakili warna, *Saturation* mewakili tingkat dominasi warna, dan *Value* mewakili tingkat kecerahan. Dengan demikian metode ini cenderung mendeteksi warna dan tingkat dominasi serta kecerahannya (Putranto, 2010).

Hue merupakan salah satu elemen dalam ruang warna HSV yang mewakili nilai warna sehingga nilai toleransi *hue* juga akan mempengaruhi nilai warna terseleksi dalam proses segmentasi. Nilai *hue* direpresentasikan dalam bentuk lingkaran dan memiliki rentang berupa sudut antara 0° - 360° , penggambaran elemen warna *hue* dapat dilihat pada Gambar 2.12 di bawah ini.

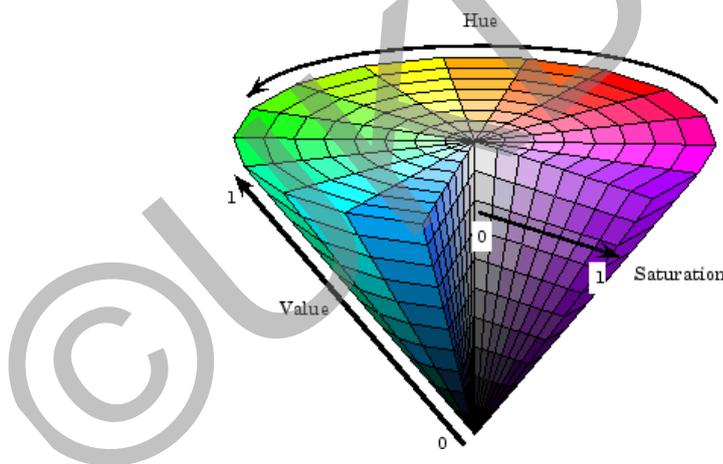


Gambar 2.12 Lingkaran elemen warna *hue*

Dikutip dari : Jurnal Informatika , Vol. 6(2)

Oleh karena elemen warna *hue* berupa lingkaran dan dituliskan dalam sudut, maka setiap operasi yang berkaitan dengan elemen warna *hue* (penambahan/ pengurangan, perhitungan toleransi, filter warna) merupakan operasi sudut.

Saturation merupakan salah satu elemen warna HSV yang mewakili tingkat intensitas warna. Pada nilai tingkat kecerahan (*value*) yang sama nilai *saturation* akan menggambarkan kedekatan suatu warna pada warna abu-abu. Pada sistem nilai *saturation* memiliki rentang antara 0 (minimum) dan 1 atau 100% (maksimum). Dalam ruang warna HSV, untuk merepresentasikan tingkat kecerahan warna digunakan elemen *value*. Pada nilai *value* maksimum warna yang dihasilkan adalah warna dengan tingkat kecerahan maksimum sedangkan pada *value* minimum dihasilkan warna dengan tingkat kecerahan minimum (warna hitam). Beberapa nilai *hue* dan *saturation* warna, jika nilai *value* yang dimiliki adalah 0 (minimum) maka warna yang dihasilkan adalah warna hitam. Nilai *value* maksimum adalah 1 (100%), dimana warna yang dihasilkan memiliki tingkat kecerahan maksimum. Nilai toleransi elemen *value* mempengaruhi tingkat kecerahan warna objek yang ikut terseleksi dalam proses segmentasi warna. Gambar 2.13 menunjukkan derajat warna pada HSV.



Gambar 2.13 Derajat warna HSV

Dikutip dari : <http://www.charisfauzan.net/2015/01/ruang-warna-hue-saturation-value-hsv.html>

2.2.11. Jaringan Syaraf Tiruan

Jaringan saraf neural merupakan landasan teori utama dalam penelitian ini. Berdasarkan pendapat (Rao, 1995), *Neural Network* atau yang biasa disebut juga *Artificial neural network* merupakan jaringan saraf tiruan dari otak manusia

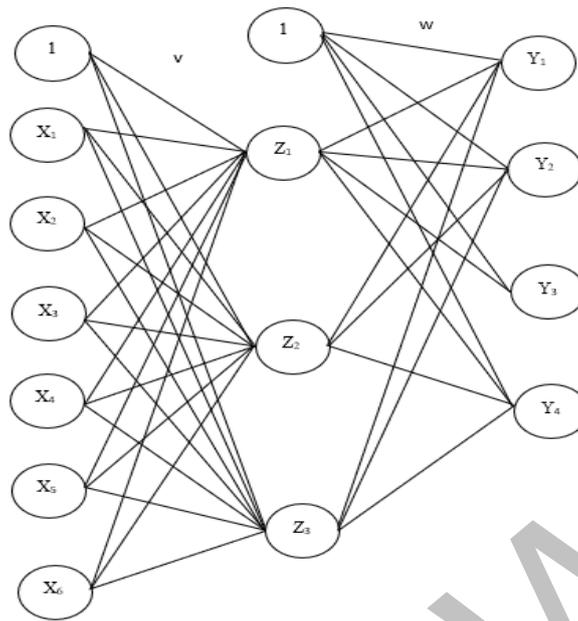
sehingga memiliki prinsip kerja yang sama dengan otak manusia. *Neural network* memiliki sifat seperti otak manusia dalam dua macam bentuk, yaitu *knowledge* diperoleh dari jaringan setelah melalui proses pembelajaran (*learning process*) dan hubungan antar-neuron yang juga dikenal sebagai *synaptic weight* digunakan untuk menyimpan *knowledge* yang sudah diperoleh sebelumnya.

Pada kasus ini, salah satu metode pada jaringan syaraf tiruan yg digunakan adalah *backpropagation*.

2.2.12. Backpropagation

Backpropagation adalah metode penurunan gradien untuk meminimalkan kuadrat error keluaran. Ada tiga tahap yang harus dilakukan dalam pelatihan jaringan, yaitu tahap perambatan maju (*forward propagation*), tahap perambatan balik, dan tahap perubahan bobot dan bias. *Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi (*hidden layer*). Gambar 2.11 adalah arsitektur *Backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah layer tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah untuk unit keluaran.

(Pramunendar & Supriyanto, 2014) menyatakan bahwa BPNN merupakan salah satu algoritma neural network yang banyak digunakan. BPNN terdiri dari dua tahapan yaitu tahapan training dan testing. Dalam proses training, BPNN menghasilkan bobot optimal yang mampu menghasilkan tingkat kesalahan dalam prediksi atau klasifikasi. BPNN terdiri dari beberapa layer yaitu layer input, layer tersembunyi dan layer output. Minimal ada 2 node pada layer input dan 1 node pada layer output. Node-node pada tiap layer dihubungkan dengan bobot yang akan terus berubah nilainya selama tahapan training. Gambar 2.14 merupakan gambar model arsitektur jaringan *backpropagation*.



Gambar 2.14 Arsitektur *backpropagation*

Algoritma pelatihan untuk jaringan dengan satu layer tersembunyi (dengan fungsi aktivasi sigmoid biner) menurut (Siang, 2009) adalah sebagai berikut:

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9

Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8

Fase I : Propagasi maju

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

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

$$z_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$[2.9] z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \quad [2.10]$$

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

$$y_net_k = w_{ko} + \sum_{j=1}^p z_j w_{kj}$$

$$[2.11] y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad [2.12]$$

Fase II : Propagasi mundur

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

$$\delta_k = (t_k - y_k) f'(y_net_k) = (t_k - y_k) y_k (1 - y_k) \quad [2.13]$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar di bawahnya (langkah 7)

Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α

$$\Delta w_{kj} = \alpha \delta_k z_j ; \quad k = 1, 2, \dots, m ; \quad j = 0, 1, \dots, p \quad [2.14]$$

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

$$\delta_net_j = \sum_{k=1}^m \delta_k w_{kj} \quad [2.15]$$

Faktor δ unit tersembunyi :

$$\delta_j = \delta_net_j f'(z_net_j) = \delta_net_j z_j (1 - z_j) \quad [2.16]$$

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

$$\Delta v_{ji} = \alpha \delta_j x_i ; \quad j = 1, 2, \dots, p ; \quad ij = 0, 1, \dots, n \quad [2.17]$$

Fase III : Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot
Perubahan bobot garis yang menuju ke unit keluaran :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (k = 1, 2, \dots, m \ ; \ j = 0, 1, \dots, p) \quad [2.18]$$

Perubahan bobot garis yang menuju ke unit tersembunyi :

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (j = 1, 2, \dots, p \ ; \ i = 0, 1, \dots, n) \quad [2.19]$$

Keterangan:

z_{net} = Hidden jaringan neuron

v = bobot antara neuron input dan neuron output

x = neuron input

z = neuron hidden

f = fungsi

y_{net} = output jaringan

e = eksponensial

w = bobot antara hidden dan target.

y = neuron output

δ = delta

t = target

Δw = perubahan bobot w

α = alpha

δ_{net} = perubahan jaringan

Δv = perubahan bobot v

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Kebutuhan

Pada penelitian ini terdapat beberapa kebutuhan yang terdiri dari perangkat lunak (*software*) dan perangkat keras (*hardware*). Kebutuhan tersebut digunakan untuk membangun dan melakukan pengujian terhadap sistem.

3.1.1. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang akan digunakan penelitian ini adalah meliputi:

- a. Sistem Operasi : Windows 10
- b. Matlab R2013a untuk membangun sistem.

3.1.2. Analisis Kebutuhan Perangkat Keras

Perangkat keras yang akan digunakan dalam penelitian ini adalah meliputi:

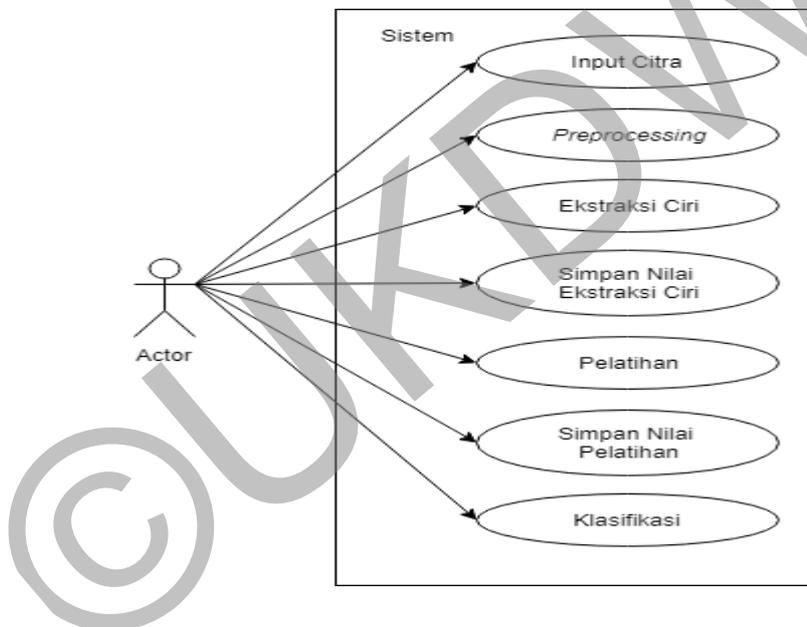
- a. Laptop
 - Processor : Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz
2.00GHz
 - Memory : 4 GB
 - Hard Disk : 500 GB
- b. Kamera
 - HP Vivo 1601(13 MP)

3.2. Perancangan Sistem

Pada penelitian ini akan dibangun sebuah sistem berbasis *desktop* yang dapat mengenali citra buah *pear*. Beberapa perancangan sistem yang akan digunakan pada penelitian ini meliputi:

3.2.1. Diagram Use Case

Pada sub bab ini akan menjelaskan mengenai urutan penggunaan sistem yang harus dilakukan oleh pengguna yang digambarkan melalui Diagram *Use Case*.



Gambar 3.1 Diagram *Use Case* Sistem

Diagram *Use Case* sistem klasifikasi buah *pear* dapat dilihat pada Gambar 3.1. Pengguna yang akan menggunakan sistem perlu melewati beberapa tahapan. Pertama-tama pengguna harus memasukkan alamat folder citra yang akan digunakan sebagai citra uji. Pengguna kemudian melakukan *preprocessing* dimana pada proses ini citra yang awalnya dalam bentuk RGB akan *resize* kemudian diubah ke dalam bentuk *grayscale* lalu dibinerisasi melalui proses

thresholding kemudian dilakukan proses *area open* dan terakhir meng*complement* citra.

Selanjutnya pengguna melakukan ekstraksi ciri. Proses ekstraksi ciri yang dilakukan adalah bentuk menggunakan *eccentricity* dan *compactness*, tekstur menggunakan GLCM dengan fitur teksturnya *contrast*, *correlation*, *energy*, *homogeneity* dan warna menggunakan HSV (*Hue*, *Saturation*, *Value*). Setelah itu nilai hasil ekstraksi ciri disimpan, kemudian dilakukan tahap pelatihan dan nilai hasil pelatihan disimpan. Kemudian tahap akhir yang dilakukan adalah klasifikasi menggunakan metode *backpropagation* dimana buah *pear* dapat dikenali berdasarkan jenisnya.

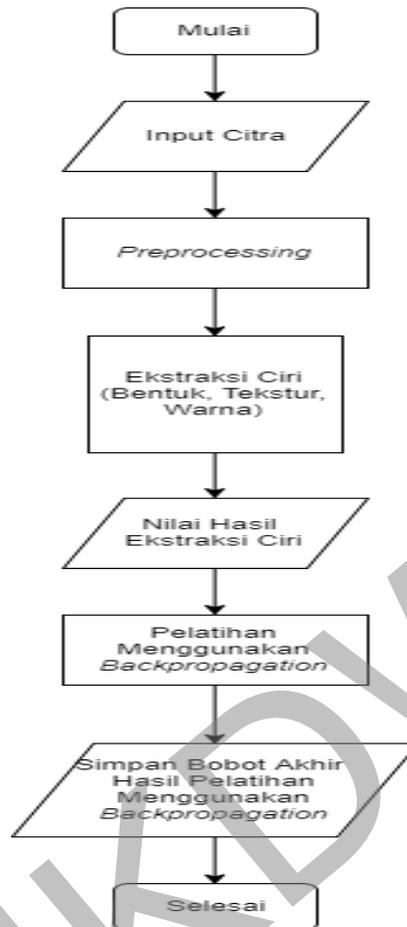
3.2.2. Algoritma dan Diagram Alir

Pada sub bab ini akan menjelaskan mengenai urutan dari proses yang akan dilakukan di dalam sistem yang digambarkan melalui diagram alir sistem. Berikut akan dijelaskan diagram alir sistem Klasifikasi Jenis Buah *Pear*.

3.2.2.1. Sistem (Pelatihan)

Diagram alir sistem untuk pelatihan pada Gambar 3.2 melalui proses sebagai berikut:

1. Sistem menerima inputan berupa citra berwarna.
2. Selanjutnya citra berwarna melakukan *preprocessing*.
3. Citra kemudian masuk dalam proses ekstraksi ciri bentuk, tekstur dan warna.
4. Hasil nilai ekstraksi ciri akan digunakan sebagai *neuron input* untuk proses pelatihan menggunakan algoritma *Backpropagation*.
5. Simpan bobot akhir hasil pelatihan.



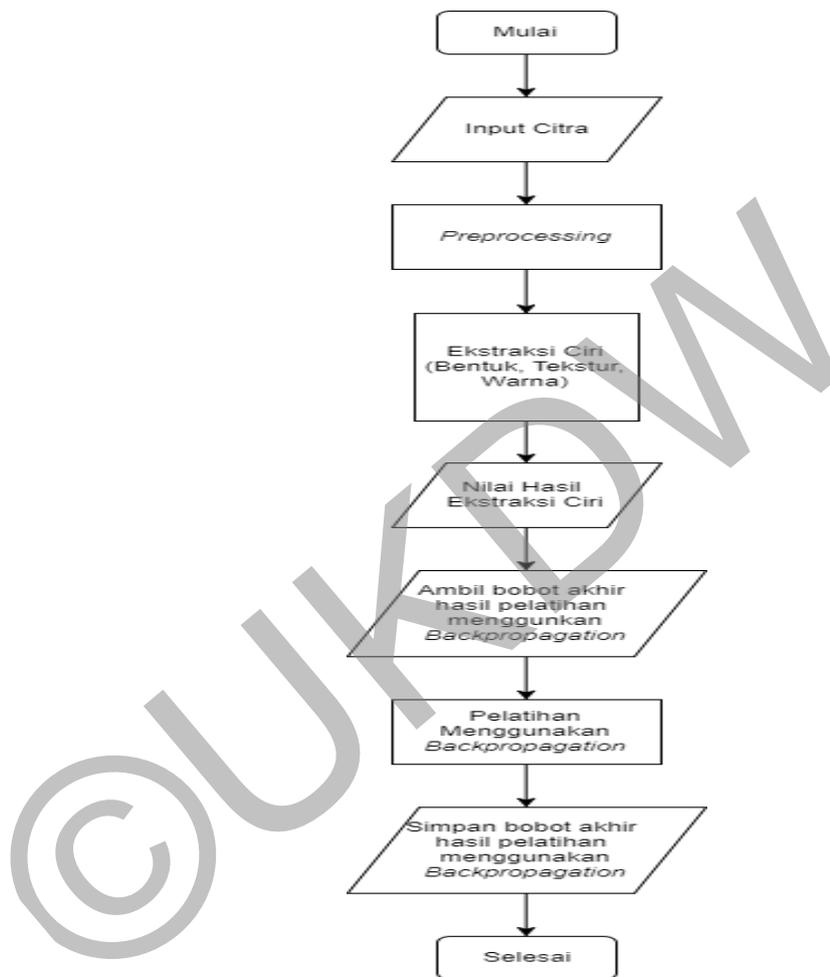
Gambar 3.2 Diagram alir proses pelatihan

3.2.2.2. Sistem (Pengenalan)

Diagram alir sistem untuk pengenalan pada Gambar 3.3 melalui proses sebagai berikut:

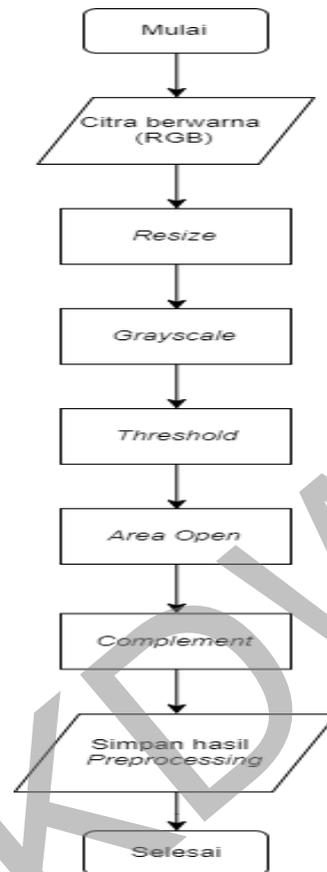
1. Sistem menerima inputan berupa citra berwarna.
2. Selanjutnya citra berwarna melakukan *preprocessing*.
3. Citra kemudian masuk dalam proses ekstraksi ciri bentuk, tekstur dan warna.
4. Mengambil bobot akhir hasil pelatihan yang menggunakan algoritma *Backpropagation*.

5. Hasil nilai ekstraksi ciri akan digunakan sebagai *neuron input* untuk proses pelatihan menggunakan algoritma *Backpropagation*.
6. Hasil dari pengenalan dengan algoritma *Backpropagation* ditampilkan bahwa sistem mengenali citra sebagai salah satu dari jenis *pear* yang dilatih atau tidak dikenali.



Gambar 3.3 Diagram alir proses pengenalan

3.2.2.3. *Preprocessing*



Gambar 3.4 Diagram alir proses *preprocessing*

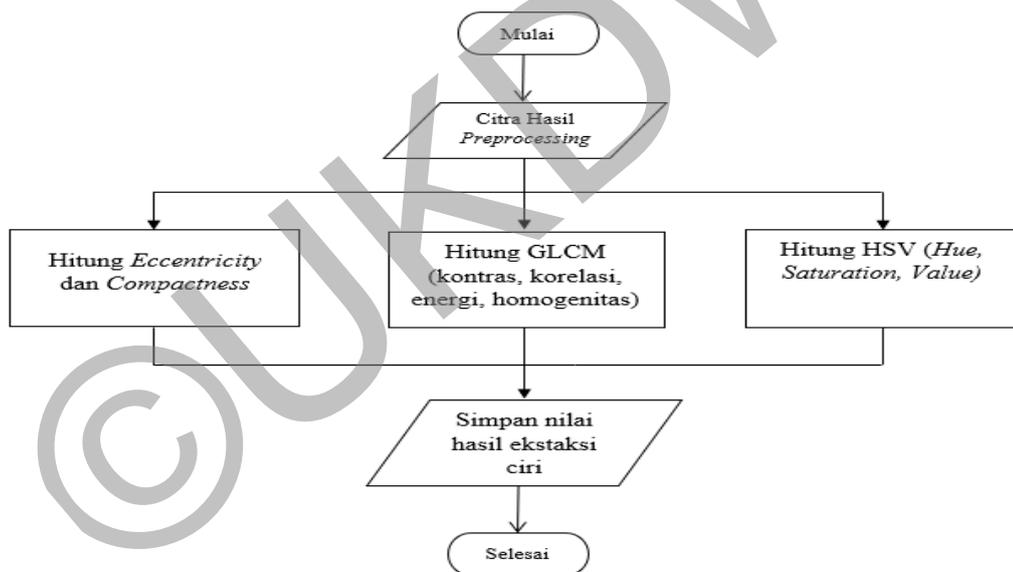
Diagram alir sistem untuk proses *preprocessing* pada Gambar 3.4 terdapat beberapa tahapan yang harus dilakukan melalui proses sebagai berikut:

1. Sistem menerima inputan berupa citra yang masih berwarna (RGB).
2. *Resize* citra menjadi 300 x 300 piksel.
3. Mengubah citra yang telah *resize* menjadi warna abu-abu yang dikenal sebagai proses *grayscale*.
4. Setelah melakukan proses *grayscale* nilai yang didapat akan digunakan pada proses *crop* dan proses *threshold*. *Crop* bertujuan untuk mendapatkan bagian pada citra tanpa adanya *background*, dimana hasil dari proses *crop* nantinya akan digunakan dalam mencari nilai dari

ekstraksi ciri tekstur. Sedangkan pada *threshold* bertujuan untuk mengubah citra *grayscale* menjadi citra hitam putih.

5. Hasil dari proses *threshold* kemudian akan melakukan proses *area open* untuk menghilangkan objek kecil pada matriks citra yang dihasilkan pada tahap sebelumnya dengan luas daerah (piksel) yang sudah ditentukan.
6. Melakukan proses *complement* untuk menginvers citra, agar *background* menjadi warna hitam dan citra menjadi warna putih.
7. Simpan hasil *preprocessing*.

3.2.2.4. Ekstraksi Ciri



Gambar 3.5 Diagram alir proses ekstraksi ciri

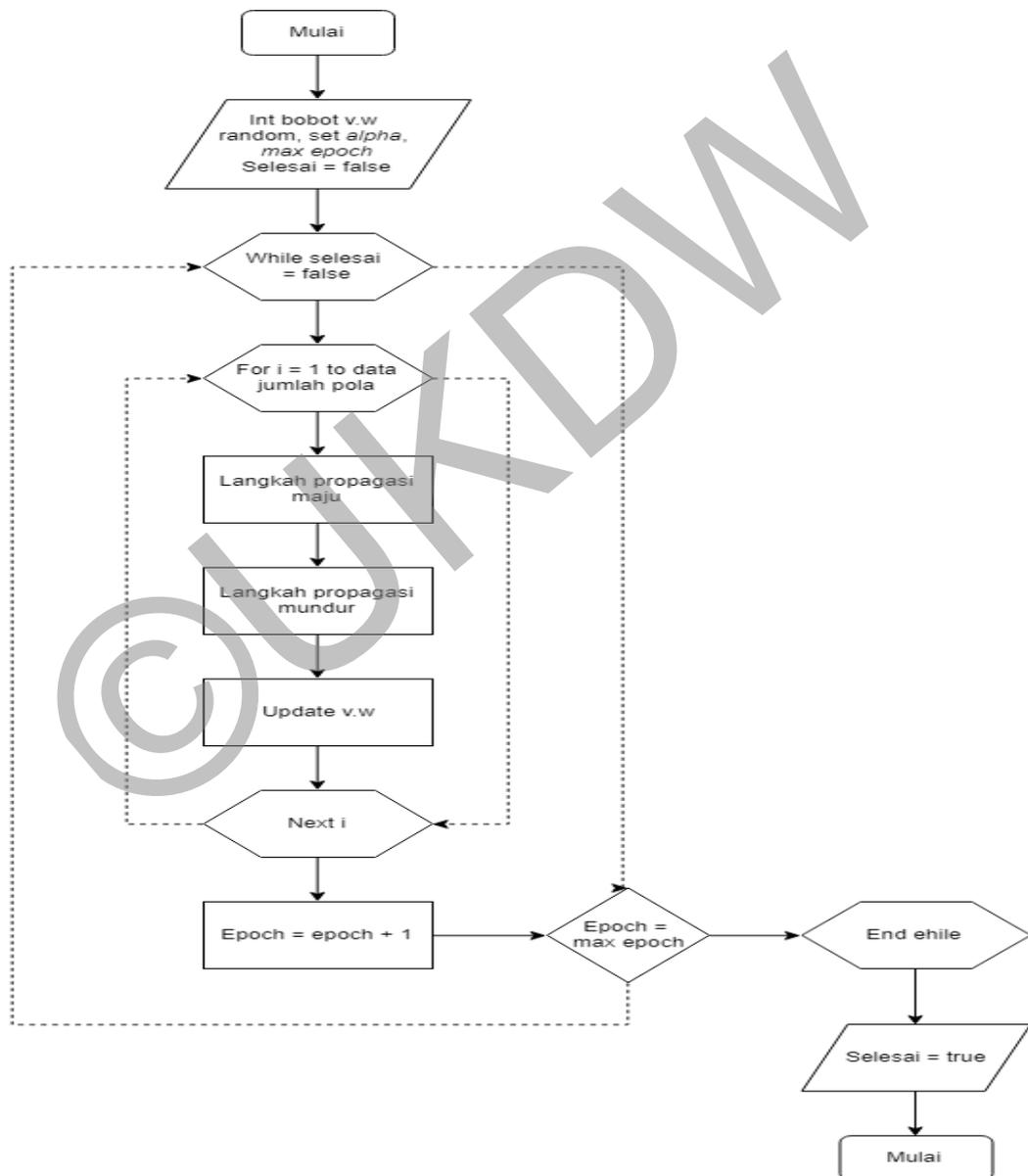
Diagram alir sistem untuk proses ekstraksi citra pada Gambar 3.5 melalui proses sebagai berikut:

1. Sistem menerima inputan berupa citra hasil *preprocessing*.
2. Citra yang telah diinput kemudian melakukan proses penghitungan bentuk, tekstur dan warna. Pada proses perhitungan bentuk

menggunakan *eccentricity* dan *compactness*. Sedangkan pada proses hitung tekstur menggunakan GLCM dengan fitur kontras, korelasi, energi dan homogenitas.

3. Hasil hitung yang didapat kemudian di simpan kedalam *database*.

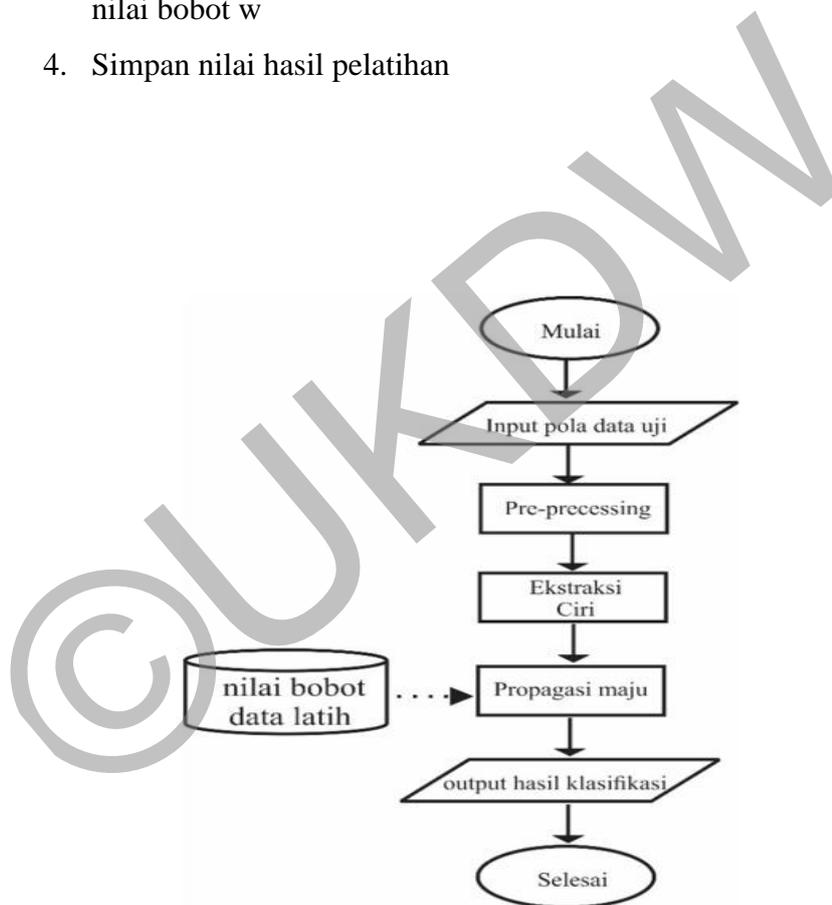
3.2.2.5. *Backpropagation* (Pelatihan)



Gambar 3.6 Diagram alir *backpropagation* (pelatihan)

Diagram alir sistem untuk proses *backpropagation* pelatihan pada Gambar 3.6 melalui proses sebagai berikut:

1. Inisialisasi nilai v random , inisialisasi nilai w random, inisialisasi nilai α , inisialisasi dan nilai max epoch.
2. Setelah melakukan inisialisasi nilai kemudian memulai langkah propagasi maju sampai dengan nilai max epoch terpenuhi.
3. Setelah melakukan propagasi maju selanjutnya adalah dengan melakukan propagasi mundur dengan mengambil nilai bobot v dan nilai bobot w
4. Simpan nilai hasil pelatihan



3.2.2.6. *Backpropagation* (Pengenalan)

Gambar 3.7 Diagram Alir proses *backpropagation* (pengenalan)

Diagram alir sistem untuk proses *backpropagation* pelatihan pada Gambar 3.7 melalui proses sebagai berikut:

1. Sistem menerima inputan data yang akan diuji berupa citra
2. Melakukan tahap *preprocessing*
3. Melakukan ekstraksi ciri
4. Melakukan propagasi maju dan membandingkan nilai yang ada dengan nilai yang tersimpan pada *data base*
5. Menampilkan hasil klasifikasi berupa jenis *pear* yang memenuhi nilai perbandingan pada tahap sebelumnya.

3.2.3. Perancangan Antarmuka



Gambar 3.8 Perancangan sistem

Pada bagian ini menjelaskan tentang perancangan antarmuka untuk sistem Klasifikasi Jenis Buah Pear. Berdasarkan perancangan sistem pada Gambar 3.8 terdapat beberapa komponen yang dimiliki oleh sistem yaitu sebagai berikut.

- a. Input Gambar : Tombol untuk menginput citra dalam format .jpg.
- b. Proses (Ekstraksi Warna) : Tombol untuk menampilkan citra hasil HSV yang terdapat pada axes5 dan menampilkan nilai hasil proses HSV pada kolom yang tersedia di ekstraksi warna.

- c. Proses (Ekstraksi Tekstur) : Tombol untuk menampilkan citra hasil *grayscale* yang terdapat pada axes2 dan menampilkan nilai hasil proses GLCM pada kolom yang tersedia di ekstraksi tekstur.
- d. Proses (Ekstraksi Bentuk) : Tombol untuk menampilkan citra hasil *complement* yang terdapat pada axes4 dan menampilkan nilai hasil proses ekstraksi bentuk pada kolom yang tersedia di ekstraksi tekstur.
- e. Proses Klasifikasi : Tombol untuk menampilkan proses hasil klasifikasi *backpropagation* untuk mengetahui jenis *pear* yang telah diinputkan.

©UKDW

BAB 4

IMPLEMENTASI DAN ANALISIS SISTEM

4.1. Implementasi Sistem

Pada sub bab ini akan menjelaskan mengenai tampilan dan urutan proses yang harus dilakukan pengguna agar sistem dapat berjalan sebagaimana mestinya. Penjelasan ini akan dijabarkan dengan bantuan gambar *screenshot* dari penggunaan sistem.



Gambar 4.1 Tampilan Awal Sistem

Gambar 4.1 memperlihatkan tampilan awal sistem untuk menjalankan proses pengenalan. Pada gambar tersebut, terbagi menjadi 5 bagian yaitu bagian pertama adalah untuk menginputkan gambar, bagian kedua adalah proses untuk ekstraksi warna, bagian ketiga adalah proses untuk ekstraksi tekstur, bagian keempat adalah proses untuk ekstraksi bentuk dan bagian kelima untuk proses hasil klasifikasi.

4.1.1. Implementasi *Input* Gambar



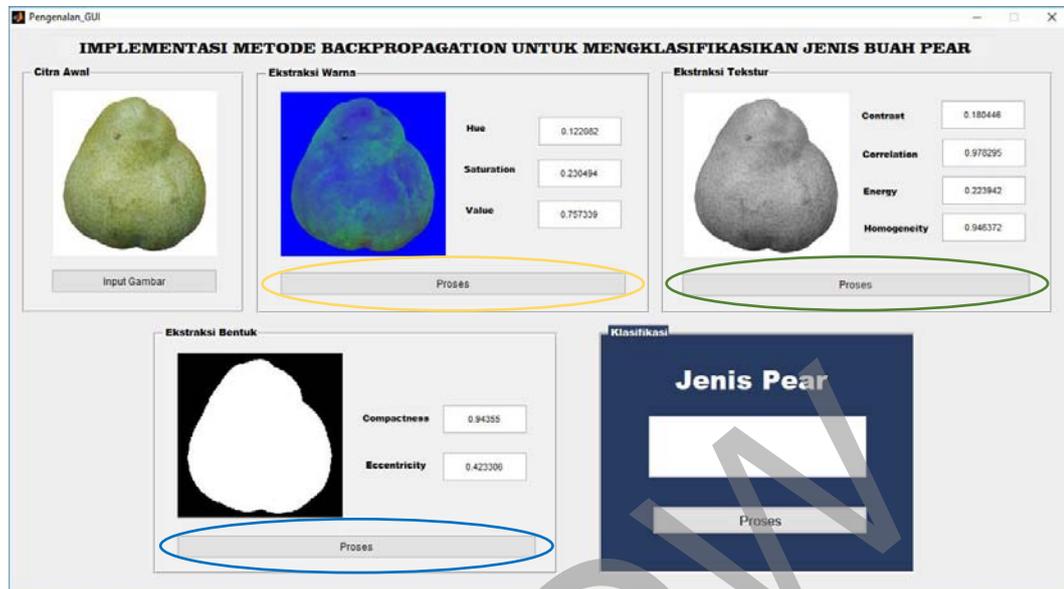
Gambar 4.2 Tampilan hasil *input* gambar

Gambar 4.2 menunjukkan citra awal yang telah ditampilkan setelah melakukan *input*. Pertama pengguna menekan tombol “Input Gambar” seperti yang terdapat pada gambar berlingkaran merah untuk dapat menginputkan gambar, kemudian pengguna dapat memilih gambar *pear* pada komputer yang bersangkutan dalam format (.jpg) yang akan diinputkan.

4.1.2. Implementasi Proses Ekstraksi Ciri

Pada Gambar 4.3 adalah proses menghitung nilai ciri warna, tekstur dan bentuk dan menampilkan citra yang digunakan untuk melakukan proses hitung tersebut. Dalam penelitian ini ekstraksi warna yang digunakan adalah *Hue*, *Saturation* dan *value*. Untuk dapat menampilkan nilai hasil ekstraksi warna, pengguna harus menekan tombol “Proses” seperti yang terdapat pada gambar berlingkaran kuning yang terdapat dalam kolom Ekstraksi Warna. Setelah menekan tombol tersebut, maka akan menampilkan citra hasil HSV dan nilai hasil

ekstraksi. Proses hitung nilai tersebut menggunakan fungsi Matlab diperlihatkan pada Gambar 4.4.



Gambar 4.3 Proses Ekstraksi ciri

```

72 % hsv
73 - imgHSV = rgb2hsv(imgrez);
74 - h = imgHSV(:,:,1); s = imgHSV(:,:,2); v = imgHSV(:,:,3);
75 - meanh = mean(mean(h));
76 - means = mean(mean(s));
77 - meanv = mean(mean(v));

```

Gambar 4.4 Program untuk mencari nilai HSV

Pada kasus ini fitur ekstraksi ciri tekstur yang digunakan pada GLCM adalah *contrast*, *correlation*, *energy* dan *homogeneity*. Untuk dapat melakukan proses ini, pengguna harus menekan tombol “Proses” seperti yang terdapat pada gambar berlingkaran hijau yang terdapat dalam kolom Ekstraksi Tekstur. Setelah menekan tombol tersebut, maka akan menampilkan citra yang digunakan untuk melakukan proses hitung dan nilai hasil ekstraksi. Citra yang ditampilkan merupakan hasil citra yang telah melakukan proses *grayscale*. *Grayscale* adalah proses pengubahan citra warna ke dalam citra abu-abu menggunakan fungsi **rgb2gray()** yang terdapat pada Matlab. Proses hitung nilai tersebut menggunakan fungsi Matlab diperlihatkan pada Gambar 4.5.

```

53 | % 0 derajat
54 - | glcm2 = graycomatrix(glcm, 'NumLevels', 8, 'GrayLimits', [], 'Offset', [0 1]);
55 - | stats = graycoprops(glcm2) ;

```

Gambar 4.5 Program untuk mencari nilai GLCM

Dan pada penelitian ini ekstraksi ciri bentuk yang digunakan adalah *compactness* dan *eccentricity*. Untuk dapat melakukan proses ini, pengguna harus menekan tombol “Proses” seperti yang terdapat pada gambar berlingkaran biru yang terdapat dalam kolom Ekstraksi Bentuk. Setelah menekan tombol tersebut, maka akan menampilkan citra yang digunakan untuk melakukan proses hitung dan nilai hasil ekstraksi. Proses hitung nilai *compactness* yang terdapat pada Matlab adalah pada Gambar 4.6 dan *eccentricity* pada Gambar 4.7.

Citra yang ditampilkan pada kolom Ekstraksi Bentuk telah melalui *preprocessing* sampai menghasilkan citra hasil *complement*. Tahap yang dilalui adalah terusan dari proses citra *grayscale* pada tahap ekstraksi teksur kemudian citra tersebut melakukan proses *thresholding*. Pada tahap ini sistem awalnya akan mencari nilai ambang menggunakan fungsi **graythresh()** yang terdapat pada Matlab. Kemudian nilai ambang yang telah didapat digunakan sebagai acuan untuk pengubahan ke citra biner, jika nilai pixel lebih dari atau sama dengan nilai ambang maka akan menghasilkan nilai ‘1’ dan jika nilai pixel kurang dari nilai ambang maka akan menghasilkan nilai ‘0’. Pengubahan citra abu-abu ke dalam hitam putih selanjutnya dapat dilakukan menggunakan fungsi **im2bw()** yang juga terdapat pada Matlab.

Setelah proses *thresholding* berhasil dilakukan, selanjutnya proses yang dilakukan adalah proses *complement*. Proses ini adalah proses *invert background* agar menjadi warna hitam dan citra menjadi warna putih. Pengubahan dalam proses ini menggunakan fungsi **imcomplement()** yang terdapat pada Matlab.

```

18 | %mencari nilai compacness
19 - | dalamakar= sqrt((4/3.14)*Area);
20 - | compacness= dalamakar/MajorAxisLength;

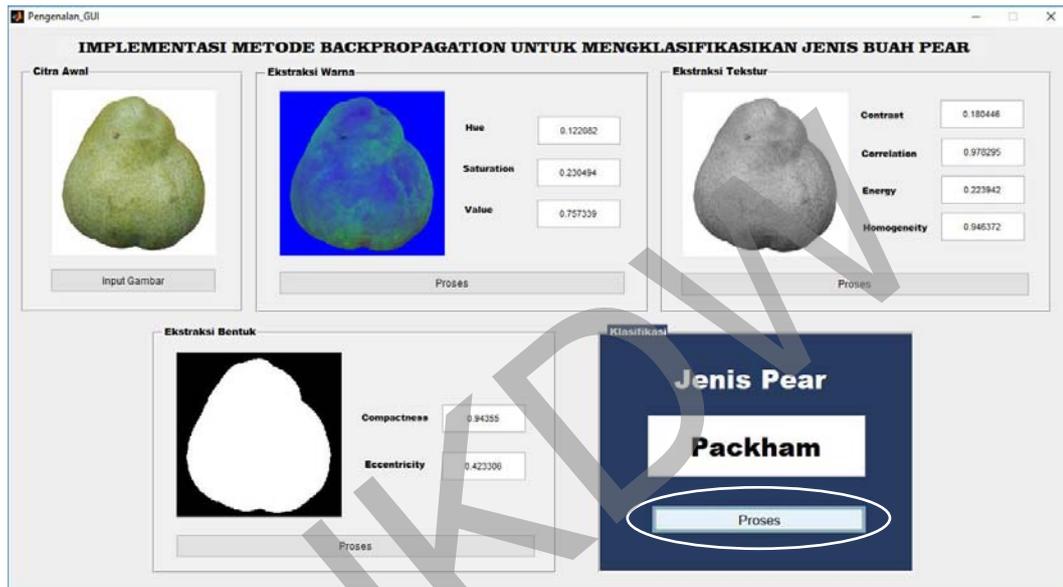
```

Gambar 4.6 Program untuk mencari nilai *compactness*

```
22 | %mencari nilai eccentricity  
23 | - | eccentricity = sqrt(1-(MinorAxisLength^2/MajorAxisLength^2));
```

Gambar 4.7 Program untuk mencari nilai *eccentricity*

4.1.3. Implementasi Proses Klasifikasi



Gambar 4.8 Proses klasifikasi

Gambar 4.8 merupakan hasil dari proses klasifikasi. Proses ini untuk menampilkan *output* kategori jenis *pear* apa yang telah *diinputkan*. Jenis *pear* yang akan ditampilkan dalam klasifikasi ini antara lain Packham/ Singo/ Sweet/ Xianglie. Untuk dapat melakukan proses ini, pengguna harus menekan tombol "Proses" seperti yang terdapat pada gambar berlingkaran putih yang terdapat dalam kolom Klasifikasi Jenis Pear.

4.2. Analisis Sistem

Pada sub bab ini akan diuji pengaruh banyaknya jumlah ciri pada fitur ekstraksi, pengaruh *alpha*, pengaruh *hidden layer*, pengaruh *epoch*, pengaruh

error rate dan pengaruh jumlah data latih. Pengujian yang akan dilakukan menggunakan 120 data uji dengan 30 data uji untuk jenis *Pear Packham*, 30 data uji untuk jenis *Pear Singo*, 30 data uji untuk jenis *Pear Sweet*, dan 30 data uji untuk jenis *Pear Xianglie*.

4.2.1. Pengaruh Jumlah Ciri

Pada proses analisis pengaruh jumlah ciri terhadap tingkat hasil klasifikasi, akan dilakukan pengujian dengan jumlah ciri yang berbeda. Pada proses ini menggunakan *aplha* 0.1, *hidden layer* 1, *epoch* 100 dan *error rate* 0.1. Ada beberapa pengujian yang akan dilakukan terhadap ekstraksi ciri sebagai berikut.

a. Menggunakan masing-masing ciri pada ekstraksi ciri

Tabel 4.1 Hasil pengujian dengan 1 ciri

| Nama Ciri | Dikenali dengan benar | Dikenali dengan salah | Persentase keberhasilan |
|-------------------------|-----------------------|-----------------------|-------------------------|
| <i>Compactness</i> | 30 | 90 | 25.00% |
| <i>Eccentricity</i> | 39 | 81 | 32.50% |
| <i>Contrast</i> | 32 | 88 | 26.60% |
| <i>Correlation</i> | 30 | 90 | 25.00% |
| <i>Energy</i> | 30 | 90 | 25.00% |
| <i>Homogeneity</i> | 30 | 90 | 25.00% |
| H (<i>Hue</i>) | 30 | 90 | 25.00% |
| S (<i>Saturation</i>) | 50 | 70 | 41.67% |
| V (<i>Value</i>) | 30 | 90 | 25.00% |

Tabel 4.1 menunjukkan hasil dari tingkat akurasi sistem jika yang digunakan hanya satu ciri. Hasil yang didapat dari analisis tersebut mencapai nilai tertinggi pada persentase keberhasilan 41.67% dengan data yang dikenali dengan benar sebanyak 50 data dari 120, maka pada analisis selanjutnya akan dicoba dengan mengkombinasi ciri agar didapat hasil yang lebih baik.

b. Menggunakan ekstraksi ciri bentuk, ekstraksi ciri tekstur, ekstraksi ciri warna

Tabel 4.2 Hasil pengujian dengan masing-masing jenis ekstraksi ciri

| Nama Ciri | Dikenali dengan benar | Dikenali dengan salah | Persentase keberhasilan |
|--|------------------------------|------------------------------|--------------------------------|
| Bentuk (<i>Compactness, Eccentricity</i>) | 25 | 95 | 20.83% |
| Tekstur (<i>Contrast, Correlation, Energy, Homogeneity</i>) | 55 | 65 | 45.83% |
| Warna (<i>Hue, Saturation, Value</i>) | 50 | 70 | 41.67% |

Tabel 4.2 menunjukkan hasil analisis dengan menggunakan masing-masing ekstraksi ciri. Hasil tertinggi yang didapat dengan persentase keberhasilan 45.83% adalah pada ekstraksi ciri tekstur dengan jumlah data yang dikenali dengan benar sebanyak 87 dari 120 data. Maka pada analisis berikutnya akan mengkombinasikan dengan ciri yang lain agar didapat hasil yang lebih baik.

c. Menggunakan kombinasi antar ekstraksi ciri

Tabel 4.3 merupakan analisis dari kombinasi antar ekstraksi ciri. Hasil yang didapat dengan persentase tertinggi adalah kombinasi antar ekstraksi ciri tekstur dan warna dengan 68 data yang dikenali dengan benar dan 52 yang tidak dikenali dengan benar dengan persentase keberhasilan 56.57%. Untuk analisis berikutnya akan dicoba dengan mengkombinasikan dengan ciri seluruhnya, dengan tujuan mendapatkan hasil akurasi yang lebih baik

Tabel 4.3 Hasil pengujian dengan kombinasi antar ekstraksi ciri

| Nama Ciri | Dikenali dengan benar | Dikenali dengan salah | Persentase keberhasilan |
|---|------------------------------|------------------------------|--------------------------------|
| Bentuk (<i>Compactness, Eccentricity</i>) + Tekstur (<i>Contrast, Correlation, Energy, Homogeneity</i>) | 53 | 67 | 44.17% |
| Bentuk (<i>Compactness, Eccentricity</i>) + Warna (<i>Hue, Saturation, Value</i>) | 43 | 67 | 44.17% |
| Tekstur (<i>Contrast, Correlation, Energy, Homogeneity</i>) + Warna (<i>Hue, Saturation, Value</i>) | 68 | 52 | 56.67% |

d. Menggunakan kombinasi seluruh ciri

Tabel 4.4 Hasil pengujian dengan kombinasi seluruh ciri

| Nama Ciri | Dikenali | Tidak dikenali | Persentase keberhasilan |
|--|-----------------|-----------------------|--------------------------------|
| <i>Compactness, Eccentricity, Contrast, Correlation, Energy, Homogeneity, Hue, Saturation, Value</i> | 69 | 51 | 57.50% |

Tabel 4.4 merupakan hasil analisis dari 9 ciri dengan persentase keberhasilan mencapai 57.50%. Dari hasil analisis jumlah ciri yang telah dilakukan sebanyak 4 kali, hasil tertinggi adalah pada kombinasi seluruh ciri. Ciri yang digunakan adalah *Compactness, Eccentricity, Contrast, Correlation, Energy, Homogeneity, Hue, Saturation* dan *Value*. Data yang dikenali dengan benar pada analisis ini mencapai 69, sedangkan yang tidak dikenali dengan benar adalah 51 data.

Kesimpulan dari analisis jumlah ciri membuktikan bahwa banyaknya ciri yang digunakan akan mempengaruhi terhadap persentase keberhasilan dalam pengenalan jenis buah *pear*. Penggunaan 9 ciri mempunyai tingkat keberhasilan yang lebih tinggi dibandingkan dengan menggunakan jumlah ciri pada analisis a, b atau c.

4.2.2. Pengaruh *Alpha*

Proses analisis pengaruh *alpha* dilakukan dengan menggunakan struktur dan parameter yang sama, yaitu nilai *hidden layer* 1, epoch 100 dan *error rate* 0.1. Pada percobaan ini akan dicoba nilai *alpha* dengan nilai 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9.

Tabel 4.5 Hasil pengujian pengaruh *alpha*

| <i>Nilai learning rate/alpha</i> | Dikenali dengan benar | Dikenali dengan salah | Persentase keberhasilan |
|----------------------------------|-----------------------|-----------------------|-------------------------|
| 0.1 | 69 | 51 | 57.50% |
| 0.2 | 78 | 42 | 65.00% |
| 0.3 | 78 | 42 | 65.00% |
| 0.4 | 77 | 43 | 64.17% |
| 0.5 | 81 | 39 | 67.50% |
| 0.6 | 8 | 40 | 66.67% |
| 0.7 | 78 | 42 | 65.00% |
| 0.8 | 80 | 40 | 66.67% |
| 0.9 | 75 | 45 | 62.50% |

Tabel 4.5 menunjukkan hasil dari analisis pada nilai *alpha* terhadap tingkat hasil klasifikasi. Dari hasil percobaan yang telah dilakukan nilai *alpha* tertinggi adalah 0.5, dengan jumlah data yang dikenalin dengan benar sebanyak 81 dari 120 data, dengan persentase keberhasilan mencapai 67.50%. Maka untuk analisis selanjutnya nilai *alpha* yang akan digunakan adalah 0.5

4.2.3. Pengaruh *Hidden Layer*

Proses analisis pengaruh *hidden layer* dilakukan dengan menggunakan struktur dan parameter yang sama, yaitu nilai *alpha* 0.5, *epoch* 100 dan *error rate* 0.1. Pada percobaan ini akan dicoba nilai *hidden layer* dengan nilai 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Tabel 4.6 Hasil pengujian pengaruh *hidden layer*

| <i>Hidden Layer</i> | Dikenali dengan benar | Dikenali dengan salah | Persentase keberhasilan |
|---------------------|-----------------------|-----------------------|-------------------------|
| 1 | 81 | 39 | 67.50% |
| 2 | 79 | 41 | 65.83% |
| 3 | 78 | 42 | 65.00% |
| 4 | 79 | 41 | 65.83% |
| 5 | 83 | 37 | 69.17% |
| 6 | 80 | 40 | 66.67% |
| 7 | 82 | 38 | 68.33% |
| 8 | 89 | 31 | 74.17% |
| 9 | 87 | 33 | 72.50% |
| 10 | 90 | 30 | 75.00% |
| 11 | 90 | 30 | 75.00% |

Tabel 4.6 menunjukkan hasil dari analisis nilai *hidden layer*. Dari hasil analisis yang telah dilakukan didapatkan hasil tertinggi pada *hidden layer* dengan nilai 10 dengan persentase keberhasilan mencapai 75.00%. Untuk kasus ini nilai *hidden layer* yang akan digunakan pada analisis selanjutnya adalah 10.

4.2.4. Pengaruh *Epoch*

Proses analisis pengaruh *epoch* dilakukan dengan menggunakan struktur dan parameter yang sama, yaitu nilai *alpha* 0.5, *hidden layer* 10 dan *error rate* 0.1.

Pada percobaan ini akan dicoba nilai *epoch* dengan nilai 100, 200, 500, 700, 1000, 1500, 2000.

Tabel 4.7 Hasil pengujian pengaruh epoch

| <i>Epoch</i> | Dikenali dengan benar | Dikenali dengan salah | Persentase keberhasilan |
|--------------|-----------------------|-----------------------|-------------------------|
| 100 | 90 | 30 | 75.00% |
| 200 | 91 | 29 | 75.83% |
| 500 | 91 | 29 | 75.83% |
| 700 | 87 | 33 | 72.50% |
| 1000 | 91 | 29 | 75.83% |
| 1500 | 87 | 33 | 72.50% |
| 2000 | 86 | 34 | 71.67% |

Tabel 4.7 menunjukkan hasil analisis yang telah dilakukan untuk nilai *epoch*. Dari hasil yang didapat, nilai tertinggi yang dihasilkan didapat pada nilai 200, 500 dan 1000 dengan persentase keberhasilan mencapai 75.83% dengan data yang dikenali secara benar adalah 91 data dari 120 data yang ada. Penulis memilih *epoch* dengan nilai 1000 untuk analisis selanjutnya.

4.2.5. Pengaruh *Error Rate*

Proses analisis pengaruh *error rate* dilakukan dengan menggunakan struktur dan parameter yang sama, yaitu nilai alpha 0.5, hidden layer 10 dan *epoch* 1000. Pada percobaan ini akan dicoba nilai epoch dengan nilai 0.1; 0.01; 0.001; 0.0001; 0.00001.

Tabel 4.8 menunjukkan hasil analisis pada *error rate*. Nilai tertinggi yang dihasilkan adalah pada *error rate* bernilai 0.001, 0.0001, dan 0.00001 dengan persentase keberhasilan 90.00%. Namun pada kasus ini penulis memilih 0.001 dengan alasan waktu yang digunakan untuk menjalankan pelatihan pada program lebih singkat dibandingkan *error rate* dengan nilai 0.0001 atau 0.00001.

Tabel 4.8 Hasil pengujian pengaruh *error rate*

| <i>Error Rate</i> | Dikenali dengan benar | Dikenali dengan salah | Persentase keberhasilan | Waktu |
|-------------------|-----------------------|-----------------------|-------------------------|-------------|
| 0.1 | 91 | 29 | 75.83% | 1 s |
| 0.01 | 105 | 15 | 87.50% | 2 s |
| 0.001 | 108 | 12 | 90.00% | 12 s |
| 0.0001 | 108 | 12 | 90.00% | 146 s |
| 0.00001 | 108 | 12 | 90.00% | 144 s |

4.2.6. Pengaruh Jumlah Data Latih

Pengaruh jumlah data latih terhadap tingkat akurasi adalah dengan melakukan proses uji pada pola data uji, pola data uji adalah pola yang tidak digunakan sebagai data latih. Pengujian menggunakan nilai *alpha* 0.5, *hidden layer* 3 dan *epoch* 3000. jumlah data yang akan dijadikan pola uji adalah 5, 10, 15, dan 20.

Tabel 4.9 Hasil pengujian pengaruh jumlah data latih

| Banyaknya Data Latih (setiap jenis) | Dikenali | Tidak dikenali | Persentase keberhasilan |
|-------------------------------------|------------|----------------|-------------------------|
| 5 | 95 | 25 | 79.17% |
| 10 | 100 | 20 | 83.33% |
| 15 | 106 | 14 | 88.33% |
| 20 | 108 | 12 | 90.00% |

Tabel 4.9 menunjukkan hasil dari analisis pengaruh jumlah data latih. Dari hasil yang didapat nilai tertinggi adalah dengan jumlah data latih 20 perjenis *pear* dengan persentase keberhasilan mencapai 90.00%.

Dari hasil analisis yang telah dilakukan dapat disimpulkan bahwa jumlah data ciri mempunyai pengaruh terhadap persentase keberhasilan dalam pengenalan jenis buah *pear*. Selain itu, nilai *alpha*, *hidden layer*, *epoch*, *error rate* dan banyaknya data pada pengujian mempengaruhi tingkat persentase

keberhasilan. Pada penelitian ini, persentase dengan tingkat akurasi mencapai 90.00% didapat pada pengujian yang dilakukan pada 9 jumlah ciri, dengan *alpha* 0.5, *hidden layer* 10, *epoch* 1000 dan *error rate* 0.001, juga dengan jumlah data pada pelatihan sebanyak 20 buah perjenis *pear*.

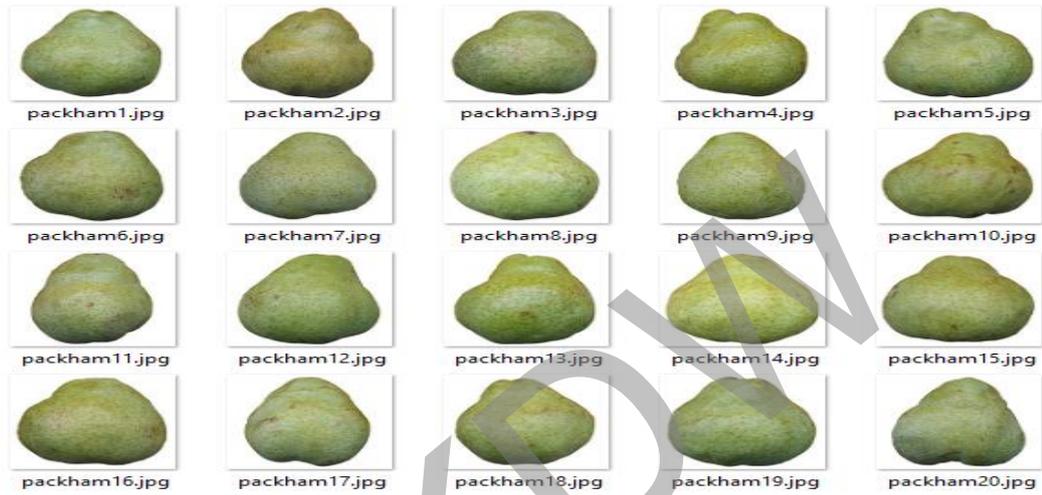
©UKDW

LAMPIRAN A

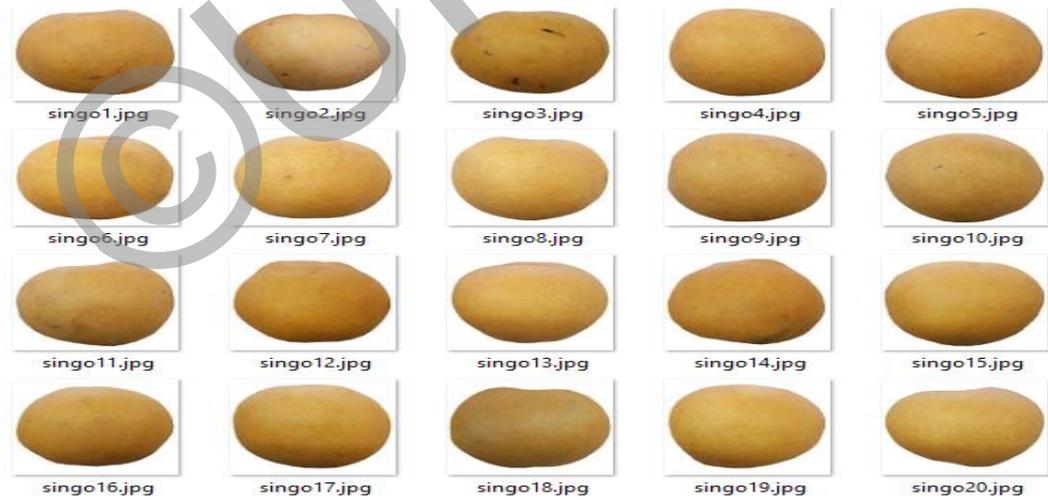
DATA LATIH DAN DATA UJI

- **Data Uji**

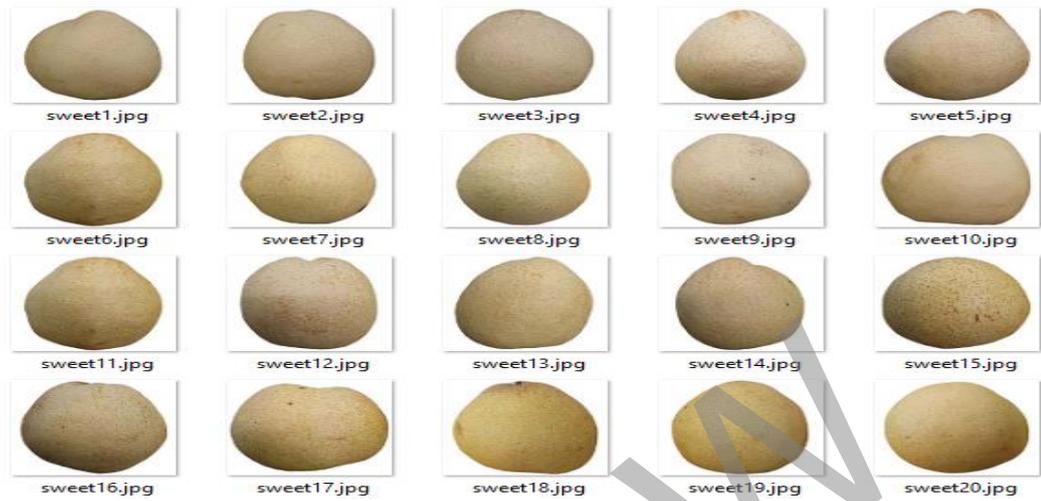
Pear Packham



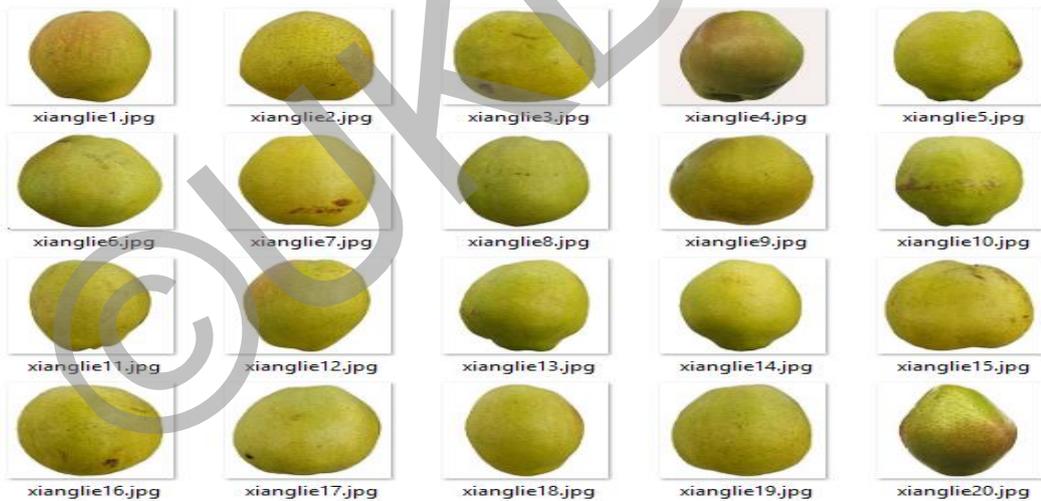
Pear Singo



Pear Sweet

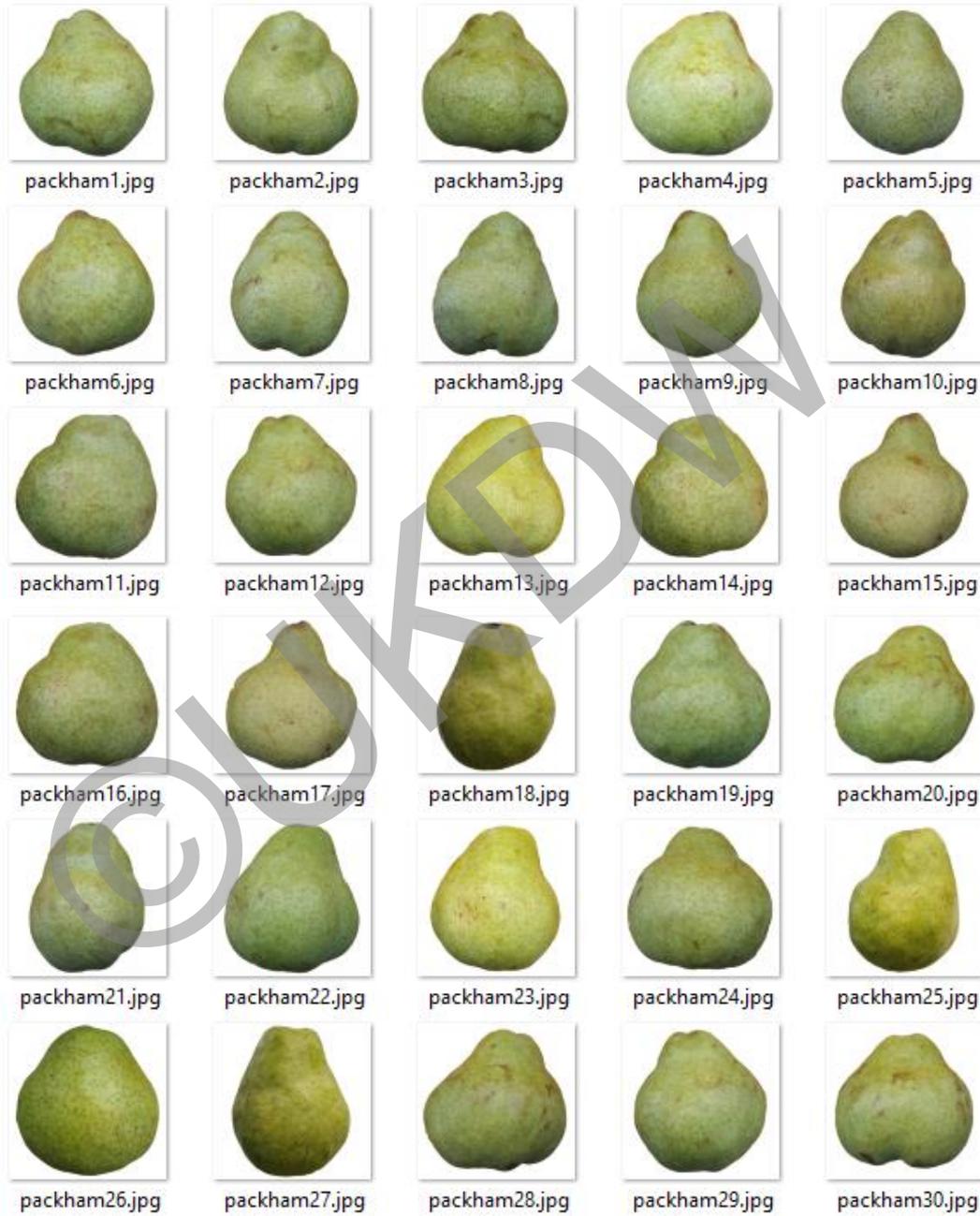


Pear Xiang Lie



- **Data Latih**

Pear Packham



Pear Siong



Pear Sweet



Pear Xiang Lie



LAMPIRAN B
REKAP EKSTRAKSI CIRI DATA PENGUJIAN

- *Pear Packham*

| Nama Buah pada file | Ekstraksi Ciri | | | | | | | | | Dikenali Sebagai |
|---------------------|--------------------|---------------------|-----------------|--------------------|---------------|--------------------|------------|-------------------|--------------|------------------|
| | Bentuk | | Tekstur | | | | Warna | | | |
| | <i>Compactness</i> | <i>Eccentricity</i> | <i>Contrast</i> | <i>Correlation</i> | <i>Energy</i> | <i>Homogeneity</i> | <i>Hue</i> | <i>Saturation</i> | <i>Value</i> | |
| 'Packham1' | 0.9483 | 0.4054 | 0.1734 | 0.9769 | 0.2305 | 0.9442 | 0.1202 | 0.2198 | 0.7738 | 'Packham' |
| 'Packham2' | 0.9435 | 0.4233 | 0.1804 | 0.9783 | 0.2239 | 0.9464 | 0.1221 | 0.2305 | 0.7573 | 'Packham' |
| 'Packham3' | 0.9479 | 0.3746 | 0.1900 | 0.9786 | 0.2069 | 0.9396 | 0.1265 | 0.2836 | 0.7110 | 'Packham' |
| 'Packham4' | 0.9354 | 0.1478 | 0.1688 | 0.9682 | 0.1897 | 0.9359 | 0.1329 | 0.2329 | 0.8150 | 'Packham' |
| 'Packham5' | 0.9048 | 0.5534 | 0.1785 | 0.9800 | 0.2777 | 0.9482 | 0.1083 | 0.1583 | 0.7717 | 'Packham' |
| 'Packham6' | 0.9658 | 0.3154 | 0.1737 | 0.9766 | 0.2252 | 0.9439 | 0.1175 | 0.2309 | 0.7625 | 'Packham' |
| 'Packham7' | 0.8871 | 0.6012 | 0.1807 | 0.9762 | 0.2662 | 0.9487 | 0.1103 | 0.1783 | 0.7860 | 'Packham' |
| 'Packham8' | 0.9027 | 0.5350 | 0.1622 | 0.9798 | 0.2484 | 0.9518 | 0.1231 | 0.1837 | 0.7457 | 'Packham' |
| 'Packham9' | 0.9053 | 0.5417 | 0.1750 | 0.9811 | 0.2736 | 0.9552 | 0.1044 | 0.2186 | 0.7653 | 'Packham' |
| 'Packham10' | 0.8999 | 0.5679 | 0.1955 | 0.9806 | 0.2471 | 0.9456 | 0.0938 | 0.2272 | 0.7383 | 'Packham' |
| 'Packham11' | 0.9568 | 0.3754 | 0.1935 | 0.9781 | 0.2014 | 0.9447 | 0.1304 | 0.2411 | 0.7094 | 'Packham' |
| 'Packham12' | 0.9460 | 0.4241 | 0.1720 | 0.9814 | 0.2488 | 0.9543 | 0.1108 | 0.2385 | 0.7627 | 'Packham' |
| 'Packham13' | 0.9559 | 0.3341 | 0.1559 | 0.9723 | 0.2326 | 0.9488 | 0.1045 | 0.3151 | 0.8181 | 'Packham' |
| 'Packham14' | 0.9594 | 0.3587 | 0.1939 | 0.9767 | 0.2127 | 0.9417 | 0.1097 | 0.2836 | 0.7477 | 'Packham' |
| 'Packham15' | 0.9197 | 0.4857 | 0.1552 | 0.9792 | 0.2828 | 0.9583 | 0.086 | 0.2057 | 0.7795 | 'Packham' |
| 'Packham16' | 0.9746 | 0.2672 | 0.1901 | 0.9792 | 0.2032 | 0.9466 | 0.1192 | 0.2677 | 0.7097 | 'Packham' |
| 'Packham17' | 0.9193 | 0.4857 | 0.1581 | 0.9789 | 0.2765 | 0.9578 | 0.0875 | 0.2093 | 0.7754 | 'Packham' |
| 'Packham18' | 0.8595 | 0.6600 | 0.1309 | 0.9884 | 0.2647 | 0.9635 | 0.0863 | 0.3279 | 0.7197 | 'Xianglie' |
| 'Packham19' | 0.9578 | 0.3687 | 0.1775 | 0.9800 | 0.2077 | 0.9492 | 0.1426 | 0.2338 | 0.7242 | 'Packham' |
| 'Packham20' | 0.9483 | 0.4025 | 0.1446 | 0.9813 | 0.2488 | 0.9583 | 0.1159 | 0.2695 | 0.7611 | 'Packham' |
| 'Packham21' | 0.8515 | 0.6801 | 0.1834 | 0.9792 | 0.2651 | 0.9513 | 0.1082 | 0.1801 | 0.7682 | 'Packham' |
| 'Packham22' | 0.9158 | 0.5155 | 0.1693 | 0.9805 | 0.2260 | 0.9522 | 0.1297 | 0.2522 | 0.7325 | 'Packham' |
| 'Packham23' | 0.9012 | 0.5280 | 0.1564 | 0.9684 | 0.2521 | 0.9480 | 0.0968 | 0.3037 | 0.8386 | 'Sweet' |
| 'Packham24' | 0.9878 | 0.1288 | 0.1718 | 0.9795 | 0.2148 | 0.9508 | 0.1215 | 0.2774 | 0.7236 | 'Packham' |
| 'Packham25' | 0.8447 | 0.6888 | 0.1273 | 0.9857 | 0.2908 | 0.9602 | 0.0828 | 0.3257 | 0.7992 | 'Xianglie' |
| 'Packham26' | 0.9627 | 0.3510 | 0.1875 | 0.9801 | 0.1905 | 0.9469 | 0.1236 | 0.4001 | 0.6972 | 'Packham' |
| 'Packham27' | 0.8500 | 0.6800 | 0.1431 | 0.9878 | 0.2505 | 0.9602 | 0.0913 | 0.3432 | 0.7235 | 'Xianglie' |
| 'Packham28' | 0.9394 | 0.4271 | 0.1593 | 0.9824 | 0.2359 | 0.9578 | 0.1100 | 0.2807 | 0.7152 | 'Packham' |
| 'Packham29' | 0.9460 | 0.4241 | 0.1720 | 0.9814 | 0.2488 | 0.9543 | 0.1108 | 0.2385 | 0.7627 | 'Packham' |
| 'Packham30' | 0.9762 | 0.2335 | 0.1652 | 0.9793 | 0.2454 | 0.9515 | 0.1072 | 0.2647 | 0.7552 | 'Packham' |

- Pear Singo

| Nama Buah pada file | Ekstraksi Ciri | | | | | | | | | Dikenali Sebagai |
|---------------------|--------------------|---------------------|-----------------|--------------------|---------------|--------------------|------------|-------------------|--------------|------------------|
| | Bentuk | | Tekstur | | | | Warna | | | |
| | <i>Compactness</i> | <i>Eccentricity</i> | <i>Contrast</i> | <i>Correlation</i> | <i>Energy</i> | <i>Homogeneity</i> | <i>Hue</i> | <i>Saturation</i> | <i>Value</i> | |
| 'Singo1' | 0.9568 | 0.4021 | 0.1204 | 0.9864 | 0.1971 | 0.9669 | 0.0698 | 0.4522 | 0.8041 | 'Singo' |
| 'Singo2' | 0.9362 | 0.4805 | 0.1159 | 0.9881 | 0.2046 | 0.9658 | 0.0643 | 0.4287 | 0.8084 | 'Singo' |
| 'Singo3' | 0.9692 | 0.3400 | 0.1188 | 0.9852 | 0.1851 | 0.9654 | 0.0698 | 0.4234 | 0.8502 | 'Singo' |
| 'Singo4' | 0.9321 | 0.4933 | 0.1043 | 0.9891 | 0.2153 | 0.9709 | 0.0636 | 0.428 | 0.8112 | 'Singo' |
| 'Singo5' | 0.9454 | 0.4464 | 0.1273 | 0.9877 | 0.1911 | 0.9625 | 0.0633 | 0.446 | 0.8058 | 'Singo' |
| 'Singo6' | 0.9568 | 0.4021 | 0.1204 | 0.9864 | 0.1971 | 0.9669 | 0.0698 | 0.4522 | 0.8041 | 'Singo' |
| 'Singo7' | 0.9532 | 0.4112 | 0.0906 | 0.9884 | 0.1957 | 0.9718 | 0.0689 | 0.3513 | 0.8472 | 'Singo' |
| 'Singo8' | 0.953 | 0.4177 | 0.1264 | 0.9875 | 0.1965 | 0.966 | 0.0733 | 0.393 | 0.7804 | 'Singo' |
| 'Singo9' | 0.9563 | 0.4046 | 0.0913 | 0.9878 | 0.2280 | 0.9735 | 0.0648 | 0.4573 | 0.8081 | 'Singo' |
| 'Singo10' | 0.9366 | 0.4664 | 0.1146 | 0.9857 | 0.2412 | 0.9684 | 0.064 | 0.3769 | 0.8498 | 'Singo' |
| 'Singo11' | 0.9877 | 0.214 | 0.1007 | 0.9852 | 0.2007 | 0.9679 | 0.0685 | 0.4356 | 0.8249 | 'Singo' |
| 'Singo12' | 0.919 | 0.5313 | 0.1058 | 0.9880 | 0.2403 | 0.9723 | 0.0639 | 0.3871 | 0.8575 | 'Singo' |
| 'Singo13' | 0.9302 | 0.4982 | 0.1107 | 0.9863 | 0.2134 | 0.9688 | 0.0686 | 0.4107 | 0.8524 | 'Singo' |
| 'Singo14' | 0.9415 | 0.4571 | 0.1091 | 0.9860 | 0.2215 | 0.9727 | 0.0709 | 0.4143 | 0.8566 | 'Singo' |
| 'Singo15' | 0.9614 | 0.3761 | 0.1097 | 0.9830 | 0.2109 | 0.9691 | 0.0732 | 0.4098 | 0.8670 | 'Singo' |
| 'Singo16' | 0.9563 | 0.4048 | 0.1169 | 0.9878 | 0.2087 | 0.9716 | 0.0731 | 0.3991 | 0.7674 | 'Singo' |
| 'Singo17' | 0.9177 | 0.5340 | 0.1109 | 0.9891 | 0.2249 | 0.9725 | 0.0619 | 0.4235 | 0.8368 | 'Singo' |
| 'Singo18' | 0.9513 | 0.4246 | 0.0891 | 0.986 | 0.2222 | 0.9759 | 0.0689 | 0.4001 | 0.8670 | 'Singo' |
| 'Singo19' | 0.9609 | 0.3845 | 0.1328 | 0.9873 | 0.2069 | 0.9719 | 0.0728 | 0.4156 | 0.7482 | 'Singo' |
| 'Singo20' | 0.9629 | 0.3721 | 0.1222 | 0.9858 | 0.2127 | 0.9636 | 0.0646 | 0.4923 | 0.7908 | 'Singo' |
| 'Singo21' | 0.9488 | 0.4319 | 0.1223 | 0.9861 | 0.2309 | 0.9667 | 0.0650 | 0.3841 | 0.8523 | 'Singo' |
| 'Singo22' | 0.9713 | 0.3201 | 0.1510 | 0.9853 | 0.1784 | 0.9508 | 0.0697 | 0.4200 | 0.8007 | 'Singo' |
| 'Singo23' | 0.9417 | 0.4561 | 0.1080 | 0.9871 | 0.2242 | 0.9711 | 0.0648 | 0.4151 | 0.8490 | 'Singo' |
| 'Singo24' | 0.947 | 0.4421 | 0.1241 | 0.9875 | 0.2037 | 0.9682 | 0.0703 | 0.3537 | 0.7583 | 'Sweet' |
| 'Singo25' | 0.946 | 0.4447 | 0.1372 | 0.9881 | 0.2061 | 0.9627 | 0.0677 | 0.3886 | 0.7713 | 'Singo' |
| 'Singo26' | 0.9265 | 0.5124 | 0.1348 | 0.9876 | 0.2030 | 0.9651 | 0.0715 | 0.3947 | 0.7605 | 'Singo' |
| 'Singo27' | 0.9344 | 0.4834 | 0.1306 | 0.9870 | 0.2138 | 0.962 | 0.0715 | 0.3807 | 0.7866 | 'Singo' |
| 'Singo28' | 0.9448 | 0.4497 | 0.1544 | 0.9875 | 0.1966 | 0.9526 | 0.0626 | 0.4131 | 0.7507 | 'Singo' |
| 'Singo29' | 0.9225 | 0.5213 | 0.1332 | 0.9871 | 0.2112 | 0.9556 | 0.0599 | 0.3977 | 0.7681 | 'Singo' |
| 'Singo30' | 0.9296 | 0.4652 | 0.131 | 0.9828 | 0.2194 | 0.9526 | 0.0796 | 0.4288 | 0.7934 | 'Singo' |

- *Pear Sweet*

| Nama Buah pada file | Ekstraksi Ciri | | | | | | | | | Dikenali Sebagai |
|---------------------|--------------------|---------------------|-----------------|--------------------|---------------|--------------------|------------|-------------------|--------------|------------------|
| | Bentuk | | Tekstur | | | | Warna | | | |
| | <i>Compactness</i> | <i>Eccentricity</i> | <i>Contrast</i> | <i>Correlation</i> | <i>Energy</i> | <i>Homogeneity</i> | <i>Hue</i> | <i>Saturation</i> | <i>Value</i> | |
| 'Sweet1' | 0.9738 | 0.3148 | 0.1453 | 0.9799 | 0.1856 | 0.9570 | 0.0735 | 0.4220 | 0.7502 | 'Singo' |
| 'Sweet2' | 0.9701 | 0.3292 | 0.1698 | 0.9839 | 0.1854 | 0.9618 | 0.0748 | 0.3776 | 0.7767 | 'Sweet' |
| 'Sweet3' | 0.9915 | 0.1729 | 0.1988 | 0.9729 | 0.1625 | 0.9343 | 0.0767 | 0.2540 | 0.7324 | 'Sweet' |
| 'Sweet4' | 0.9537 | 0.4039 | 0.1409 | 0.9734 | 0.1928 | 0.9526 | 0.0741 | 0.3820 | 0.8612 | 'Singo' |
| 'Sweet5' | 0.9768 | 0.2733 | 0.1508 | 0.9801 | 0.1723 | 0.9486 | 0.0772 | 0.4049 | 0.8320 | 'Singo' |
| 'Sweet6' | 0.9609 | 0.3759 | 0.1526 | 0.9799 | 0.1801 | 0.9521 | 0.0637 | 0.2413 | 0.7369 | 'Sweet' |
| 'Sweet7' | 0.9508 | 0.4180 | 0.1570 | 0.9788 | 0.1839 | 0.9476 | 0.0768 | 0.2988 | 0.7857 | 'Sweet' |
| 'Sweet8' | 0.9615 | 0.3770 | 0.1778 | 0.9739 | 0.1947 | 0.9422 | 0.0712 | 0.2231 | 0.8148 | 'Sweet' |
| 'Sweet9' | 0.9396 | 0.4178 | 0.1783 | 0.9741 | 0.1961 | 0.9377 | 0.0953 | 0.3037 | 0.8156 | 'Sweet' |
| 'Sweet10' | 0.6612 | 0.8763 | 0.1315 | 0.9849 | 0.1891 | 0.9551 | 0.4244 | 0.3018 | 0.8038 | 'Xianglie' |
| 'Sweet11' | 0.9421 | 0.4550 | 0.1773 | 0.9742 | 0.1852 | 0.9412 | 0.0672 | 0.1987 | 0.7807 | 'Sweet' |
| 'Sweet12' | 0.9265 | 0.5117 | 0.1505 | 0.9761 | 0.2067 | 0.9505 | 0.0629 | 0.1831 | 0.7857 | 'Sweet' |
| 'Sweet13' | 0.9326 | 0.4876 | 0.1785 | 0.9757 | 0.2046 | 0.9450 | 0.0620 | 0.1839 | 0.7981 | 'Sweet' |
| 'Sweet14' | 0.9783 | 0.2786 | 0.1748 | 0.9788 | 0.1674 | 0.9434 | 0.0769 | 0.3174 | 0.8071 | 'Sweet' |
| 'Sweet15' | 0.9939 | 0.1372 | 0.1743 | 0.9728 | 0.1723 | 0.9356 | 0.0801 | 0.3849 | 0.7818 | 'Singo' |
| 'Sweet16' | 0.9890 | 0.1757 | 0.1658 | 0.9779 | 0.1675 | 0.9454 | 0.0801 | 0.3663 | 0.8022 | 'Singo' |
| 'Sweet17' | 0.9595 | 0.3393 | 0.1926 | 0.9670 | 0.1564 | 0.9289 | 0.0825 | 0.2931 | 0.7319 | 'Sweet' |
| 'Sweet18' | 0.9639 | 0.3372 | 0.2127 | 0.9699 | 0.1628 | 0.9287 | 0.0783 | 0.3092 | 0.7686 | 'Sweet' |
| 'Sweet19' | 0.8559 | 0.4626 | 0.1350 | 0.9762 | 0.2028 | 0.9536 | 0.0693 | 0.2332 | 0.7921 | 'Sweet' |
| 'Sweet20' | 0.9610 | 0.3821 | 0.2121 | 0.9751 | 0.2322 | 0.9325 | 0.0796 | 0.2806 | 0.7470 | 'Sweet' |
| 'Sweet21' | 0.9555 | 0.4055 | 0.1992 | 0.9747 | 0.1529 | 0.9341 | 0.0825 | 0.3104 | 0.7136 | 'Sweet' |
| 'Sweet22' | 0.9620 | 0.3710 | 0.1674 | 0.9754 | 0.1711 | 0.9419 | 0.0743 | 0.2658 | 0.7763 | 'Sweet' |
| 'Sweet23' | 0.9790 | 0.2825 | 0.2003 | 0.9722 | 0.1617 | 0.9280 | 0.0788 | 0.2809 | 0.7309 | 'Sweet' |
| 'Sweet24' | 0.9553 | 0.4049 | 0.1694 | 0.9845 | 0.1860 | 0.9577 | 0.0696 | 0.4037 | 0.7380 | 'Singo' |
| 'Sweet25' | 0.9545 | 0.4108 | 0.1821 | 0.9853 | 0.1742 | 0.9551 | 0.0697 | 0.4428 | 0.7230 | 'Sweet' |
| 'Sweet26' | 0.9783 | 0.2786 | 0.1748 | 0.9788 | 0.1674 | 0.9434 | 0.0769 | 0.3174 | 0.8071 | 'Sweet' |
| 'Sweet27' | 0.9251 | 0.5131 | 0.1514 | 0.9749 | 0.2164 | 0.9507 | 0.0673 | 0.2457 | 0.7784 | 'Sweet' |
| 'Sweet28' | 0.6399 | 0.8677 | 0.1202 | 0.9692 | 0.1833 | 0.9525 | 0.1082 | 0.3030 | 0.8116 | 'Sweet' |
| 'Sweet29' | 0.7671 | 0.6809 | 0.1161 | 0.9737 | 0.1993 | 0.9554 | 0.0994 | 0.2903 | 0.8175 | 'Sweet' |
| 'Sweet30' | 0.9508 | 0.4249 | 0.1744 | 0.9776 | 0.1864 | 0.9433 | 0.0765 | 0.3190 | 0.7364 | 'Sweet' |

- Pear Xiang Lie

| Nama Buah pada file | Ekstraksi Ciri | | | | | | | | | Dikenali Sebagai |
|------------------------|--------------------|---------------------|-----------------|--------------------|---------------|--------------------|------------|-------------------|--------------|---------------------|
| | Bentuk | | Tekstur | | | | Warna | | | |
| | <i>Compactness</i> | <i>Eccentricity</i> | <i>Contrast</i> | <i>Correlation</i> | <i>Energy</i> | <i>Homogeneity</i> | <i>Hue</i> | <i>Saturation</i> | <i>Value</i> | |
| 'Xianglie1' | 0.8708 | 0.6493 | 0.1664 | 0.9827 | 0.2510 | 0.9512 | 0.0898 | 0.4280 | 0.7697 | 'Xianglie' |
| 'Xianglie2' | 0.8705 | 0.6518 | 0.1438 | 0.9826 | 0.2776 | 0.9562 | 0.0854 | 0.4124 | 0.7792 | 'Xianglie' |
| 'Xianglie3' | 0.8798 | 0.6306 | 0.1884 | 0.9834 | 0.2431 | 0.9538 | 0.0888 | 0.4620 | 0.7450 | 'Xianglie' |
| 'Xianglie4' | 0.8940 | 0.5985 | 0.1704 | 0.9831 | 0.2300 | 0.9514 | 0.0960 | 0.4761 | 0.7447 | 'Xianglie' |
| 'Xianglie5' | 0.8849 | 0.6188 | 0.1103 | 0.9824 | 0.2641 | 0.9667 | 0.0841 | 0.4197 | 0.7886 | 'Xianglie' |
| 'Xianglie6' | 0.9356 | 0.4834 | 0.1615 | 0.9822 | 0.2207 | 0.9515 | 0.0983 | 0.4550 | 0.7720 | 'Xianglie' |
| 'Xianglie7' | 0.8641 | 0.6629 | 0.1275 | 0.9815 | 0.2684 | 0.9592 | 0.0815 | 0.4103 | 0.7941 | 'Xianglie' |
| 'Xianglie8' | 0.9347 | 0.4820 | 0.1414 | 0.9786 | 0.2106 | 0.9519 | 0.0996 | 0.4682 | 0.7601 | 'Xianglie' |
| 'Xianglie9' | 0.8538 | 0.6815 | 0.1248 | 0.9820 | 0.2876 | 0.9645 | 0.0822 | 0.4106 | 0.8021 | 'Xianglie' |
| 'Xianglie10' | 0.8773 | 0.6361 | 0.1569 | 0.9863 | 0.2496 | 0.9607 | 0.0830 | 0.4344 | 0.7929 | 'Xianglie' |
| 'Xianglie11' | 0.8716 | 0.6477 | 0.1273 | 0.9878 | 0.2423 | 0.9575 | 0.5092 | 0.3965 | 0.7056 | 'Xianglie' |
| 'Xianglie12' | 0.8590 | 0.6741 | 0.1413 | 0.9830 | 0.2643 | 0.9607 | 0.0819 | 0.4128 | 0.8015 | 'Xianglie' |
| 'Xianglie13' | 0.8576 | 0.6771 | 0.1360 | 0.9865 | 0.2771 | 0.9649 | 0.0811 | 0.4038 | 0.8026 | 'Xianglie' |
| 'Xianglie14' | 0.8649 | 0.6613 | 0.1293 | 0.9874 | 0.2671 | 0.9575 | 0.5539 | 0.3775 | 0.7360 | 'Xianglie' |
| 'Xianglie15' | 0.8440 | 0.6967 | 0.0949 | 0.9848 | 0.2942 | 0.9710 | 0.0772 | 0.3866 | 0.8244 | 'Xianglie' |
| 'Xianglie16' | 0.8342 | 0.7178 | 0.1270 | 0.9823 | 0.2833 | 0.9602 | 0.0779 | 0.4064 | 0.8044 | 'Xianglie' |
| 'Xianglie17' | 0.8355 | 0.7154 | 0.1458 | 0.9830 | 0.2948 | 0.9598 | 0.0770 | 0.4110 | 0.8009 | 'Xianglie' |
| 'Xianglie18' | 0.8354 | 0.7153 | 0.1656 | 0.9805 | 0.2766 | 0.9527 | 0.0750 | 0.3829 | 0.8254 | 'Xianglie' |
| 'Xianglie19' | 0.8484 | 0.6925 | 0.1231 | 0.9805 | 0.2876 | 0.9595 | 0.0787 | 0.4053 | 0.8076 | 'Xianglie' |
| 'Xianglie20' | 0.9132 | 0.5491 | 0.1760 | 0.9782 | 0.2152 | 0.9419 | 0.0884 | 0.4605 | 0.8143 | 'Xianglie' |
| 'Xianglie21' | 0.8191 | 0.6412 | 0.1522 | 0.9876 | 0.2812 | 0.9540 | 0.0735 | 0.3406 | 0.7804 | 'Xianglie' |
| 'Xianglie22' | 0.8960 | 0.5917 | 0.1665 | 0.9885 | 0.2392 | 0.9473 | 0.4877 | 0.3913 | 0.7423 | 'Xianglie' |
| 'Xianglie23' | 0.8680 | 0.6551 | 0.1165 | 0.9894 | 0.2529 | 0.9644 | 0.5130 | 0.3635 | 0.7093 | 'Xianglie' |
| 'Xianglie24' | 0.8440 | 0.6268 | 0.1248 | 0.9879 | 0.2350 | 0.9657 | 0.4980 | 0.4339 | 0.7732 | 'Xianglie' |
| 'Xianglie25' | 0.8726 | 0.6429 | 0.1426 | 0.9862 | 0.2826 | 0.9590 | 0.5282 | 0.3961 | 0.7598 | 'Xianglie' |
| 'Xianglie26' | 0.8589 | 0.6689 | 0.1376 | 0.9907 | 0.2997 | 0.9566 | 0.5315 | 0.3336 | 0.7471 | 'Xianglie' |
| 'Xianglie27' | 0.8455 | 0.6825 | 0.1567 | 0.9863 | 0.2724 | 0.9578 | 0.5461 | 0.3825 | 0.7416 | 'Xianglie' |
| 'Xianglie28' | 0.8521 | 0.6551 | 0.1379 | 0.9870 | 0.2505 | 0.9626 | 0.5228 | 0.4119 | 0.7920 | 'Xianglie' |
| 'Xianglie29' | 0.8771 | 0.6327 | 0.1349 | 0.9878 | 0.2420 | 0.9580 | 0.5276 | 0.4122 | 0.7190 | 'Xianglie' |
| 'Xianglie30' | 0.8538 | 0.6237 | 0.1157 | 0.9882 | 0.2585 | 0.9677 | 0.5494 | 0.3833 | 0.7401 | 'Xianglie' |

©UKDW

LAMPIRAN C

LISTING PROGRAM

Fungsi EkstraksiCiri.m (fungsi untuk mendapatkan ekstraksi ciri)

```
function [ciri1,ciri2,ciri3,ciri4,ciri5,ciri6,ciri7,ciri8,ciri9] =
EkstraksiCiri(gambar)

img = imread(gambar);
imgrez = imresize(img,[300 300]);
imggray = rgb2gray(imgrez);
threshold = graythresh(imggray);
bw = im2bw(imggray,threshold);
bw = bwareaopen(bw,50);
imgcomp = imcomplement(bw);

stats =
regionprops(imgcomp,'Area','MajorAxisLength','MinorAxisLength');
MajorAxisLength=stats.MajorAxisLength;
MinorAxisLength=stats.MinorAxisLength;
Area=stats.Area;

% Mencari nilai compacness
dalamakar= sqrt((4/3.14)*Area);
compacness= dalamakar/MajorAxisLength;

% Mencari nilai eccentricity
eccentricity = sqrt(1-(MinorAxisLength^2/MajorAxisLength^2));

ciri1 = compacness
ciri2 = eccentricity

% glcm
glcm = imggray;

% 0 derajat
glcm2 =
graycomatrix(glcm,'NumLevels',8,'GrayLimits',[],'Offset',[0 1]);
stats = graycoprops(glcm2) ;

ciri3 = stats.Contrast
ciri4 = stats.Correlation
ciri5 = stats.Energy
ciri6 = stats.Homogeneity

% hsv
imghsv = rgb2hsv(imgrez);
```

```

h = imghsv(:,:,1); s = imghsv(:,:,2); v = imghsv(:,:,3);
meanh = mean(mean(h));
means = mean(mean(s));
meanv = mean(mean(v));

ciri7 = meanh
ciri8 = means
ciri9 = meanv

end

```

SetupData.m (membuat *database*)

```

clc;clear all;

motif{1} = 'Packham';
motif{2} = 'Singo';
motif{3} = 'Sweet';
motif{4} = 'Xianglie';

jumlah_motif = 4;
conter = 0;
for j = 1:jumlah_motif
for i = 1:20

    namafile = ['C:\skripsipear\masterr\' motif{j} num2str(i)
'.jpg']
    [ciri1 ciri2 ciri3 ciri4 ciri5 ciri6 ciri7 ciri8 ciri9 ] =
EkstraksiCiri(namafile)

    conter = conter + 1;
    DataPear{conter,1} = namafile;
    DataPear{conter,2} = [motif{j} num2str(i)];
    DataPear{conter,3} = motif{j};
    DataPear{conter,4} = ciri1;
    DataPear{conter,5} = ciri2;
    DataPear{conter,6} = ciri3;
    DataPear{conter,7} = ciri4;
    DataPear{conter,8} = ciri5;
    DataPear{conter,9} = ciri6;
    DataPear{conter,10} = ciri7;
    DataPear{conter,11} = ciri8;
    DataPear{conter,12} = ciri9;

end
end

save DataMaster
['Setup Ekstraksi Data Selesai']

```

Pelatihan.m (proses pelatihan data)

```

clear all; clc;

load DataMaster
motif{1} = 'Packham';
motif{2} = 'Singo';
motif{3} = 'Sweet';
motif{4} = 'Xianglie';

jumlahpola = 4;

a = cell2mat(DataPear(:,4));
maks1 = max(a)
a = cell2mat(DataPear(:,5));
maks2 = max(a)
a = cell2mat(DataPear(:,6));
maks3 = max(a)
a = cell2mat(DataPear(:,7));
maks4 = max(a)
a = cell2mat(DataPear(:,8));
maks5 = max(a)
a = cell2mat(DataPear(:,9));
maks6 = max(a)
a = cell2mat(DataPear(:,10));
maks7 = max(a)
a = cell2mat(DataPear(:,11));
maks8 = max(a)
a = cell2mat(DataPear(:,12));
maks9 = max(a)

conter = 0;
for itarget = 1: jumlahpola
    for ipola = 1:20
        conter = conter + 1;

x(conter,1)=DataPear{conter,4}/maks1 ;
x(conter,2)=DataPear{conter,5}/maks2 ;
x(conter,3)=DataPear{conter,6}/maks3 ;
x(conter,4)=DataPear{conter,7}/maks4 ;
x(conter,5)=DataPear{conter,8}/maks5 ;
x(conter,6)=DataPear{conter,9}/maks6 ;
x(conter,7)=DataPear{conter,10}/maks7 ;
x(conter,8)=DataPear{conter,11}/maks8 ;
x(conter,9)=DataPear{conter,12}/maks9 ;

simpannamafile{conter} = motif{itarget} ;
        end;
    end;

jumlahpmaster = jumlahpola * 20;

```



```

    end
end
%masukkan zin ke rumus utk hitung znet
for i=1:jumlahz
    z(ipm,i)=1/(1+exp(-zin(i)));
end
%hitung ynet
for i=1:jumlahciri
    %hitung yin dari y ke-i
    yin(i)=0;
    for j=0:jumlahz
        if j==0
            yin(i)=rand()*0.1;
        else
            yin(i)=yin(i)+z(ipm,j)*w(ipm,i,j);
        end
    end
end
end
%masukkan yin ke rumus utk hitung ynet
for i=1:jumlahciri
    y(ipm,i)=1/(1+exp(-yin(i)));
end
%hitung error
for i=1:jumlahciri
    er(ipm,i)=(x(ipm,i))-y(ipm,i))*y(ipm,i)*(1-y(ipm,i));
end
%hitung delta w
for i=1:jumlahz
    for j=1:jumlahciri
        dw(ipm,i,j)=alpa*er(ipm,j)*z(ipm,i);
    end
end
end
%hitung error inj
for i=1:jumlahz
    einj(i)=0;
    for j=1:jumlahciri
        einj(i)=einj(i)+er(ipm,j)*w(ipm,j,i);
    end
end
end
%hitung error ke-2
for i=1:jumlahz
    er2(ipm,i)=einj(i)*z(ipm,i)*(1-z(ipm,i));
end
end
%hitung delta v
for i=1:jumlahciri
    for j=1:jumlahz
        dv(i,j)=alpa*er2(ipm,j)*(x(ipm,i));
    end
end
end
%update bobot w
for i=1:jumlahciri
    for j=1:jumlahz
        w(ipm,i,j)=w(ipm,i,j)+dw(ipm,j,i);
    end
end
end
%update bobot v

```

```

for i=1:jumlahciri
    for j=1:jumlahz
        v(j,i)=v(j,i)+dv(i,j);
    end
end
%hitung yout
for i=1:jumlahciri
    yout(ipm,i)=0;
    for j=1:jumlahz
        yout(ipm,i)=yout(ipm,i)+z(ipm,j)*w(ipm,i,j);
    end
end

end %next ipm
%hitung mse
mse=0;
for ipm=1:conter
    selisih=0;

    for i=1:jumlahciri
        (x(ipm,i)-y(ipm,i))^2
        selisih = selisih + (x(ipm,i)-y(ipm,i))^2 ;
    end

    if (selisih>0)
        end
        mse=mse+selisih;
    end
    mse=0.5*(mse/conter);

end
selisih
['pelatihan selesai,mse = ' num2str(mse)]

epoch
save Pelatihan

```

Pengujian.m (proses pengujian)

```

clc;
clear all;

namafileuji ='c:\skripsipear\ujii\xianglie30.jpg';

load DataMaster;
img = imread(namafileuji);
imgrez = imresize(img,[300 300]);
imggray = rgb2gray(imgrez);
threshold = graythresh(imggray);

```

```

bw = im2bw(imggray,threshold);
bw = bwareaopen(bw,50);
imgcomp = imcomplement(bw);

stats =
regionprops(imgcomp, 'Area', 'MajorAxisLength', 'MinorAxisLength');
MajorAxisLength=stats.MajorAxisLength;
MinorAxisLength=stats.MinorAxisLength;
Area=stats.Area;

%mencari nilai compacness
dalamakar= sqrt((4/3.14)*Area);
compacness= dalamakar/MajorAxisLength;

%mencari nilai eccentricity
eccentricity = sqrt(1-(MinorAxisLength^2/MajorAxisLength^2));

ciriuji1 = compacness
ciriuji2 = eccentricity

% glcm
glcm = imggray;

% 0 derajat
glcm2 =
graycomatrix(glcm, 'NumLevels',8, 'GrayLimits', [], 'Offset', [0 1]);
stats = graycoprops(glcm2) ;

ciriuji3 = stats.Contrast
ciriuji4 = stats.Correlation
ciriuji5 = stats.Energy
ciriuji6 = stats.Homogeneity

% hsv
imgHSV = rgb2hsv(imgrez);
h = imgHSV(:,:,1); s = imgHSV(:,:,2); v = imgHSV(:,:,3);
meanh = mean(mean(h));
means = mean(mean(s));
meanv = mean(mean(v));

ciriuji7 = meanh
ciriuji8 = means
ciriuji9 = meanv

a = cell2mat(DataPear(:,4))
maks1 = max(a);
a = cell2mat(DataPear(:,5))
maks2 = max(a);
a = cell2mat(DataPear(:,6))
maks3 = max(a) ;
a = cell2mat(DataPear(:,7))

```

```

maks4 = max(a) ;
a = cell2mat(DataPear(:,8))
maks5 = max(a) ;
a = cell2mat(DataPear(:,9))
maks6 = max(a) ;
a = cell2mat(DataPear(:,10))
maks7 = max(a) ;
a = cell2mat(DataPear(:,11))
maks8 = max(a) ;
a = cell2mat(DataPear(:,12))
maks9 = max(a) ;

xuji(1)=ciriuji1/maks1;
xuji(2)=ciriuji2/maks2;
xuji(3)=ciriuji3/maks3;
xuji(4)=ciriuji4/maks4;
xuji(5)=ciriuji5/maks5;
xuji(6)=ciriuji6/maks6;
xuji(7)=ciriuji7/maks7;
xuji(8)=ciriuji8/maks8;
xuji(9)=ciriuji9/maks9;

load Pelatihan
for ipm=1:conter
    %hitung z in
    for j=1:jumlahz
        zin(j)=0
        for i=1:jumlahciri
            zin(j)=zin(j)+xuji(i)*v(j,i);
        end

        %hitung z
        z(ipm,j)=1/(1+exp(-zin(j)));
    end
    %hitung yin
    for k=1:jumlahciri
        yin(k)=0;
        for j=1:jumlahz
            yin(k)=yin(k)+z(ipm,j)*w(ipm,k,j);
        end
        y(ipm,k)=1/(1+exp(-yin(k)));
    end
end

%cek dan bandingkan dengan masternya
for ipm=1:conter
    selisihuji(ipm)=0;
    for i=1:jumlahciri
        selisihuji(ipm)=selisihuji(ipm)+abs(xuji(i)-y(ipm,i));
    end
    selisihuji(ipm)= (selisihuji(ipm)/jumlahciri)*100;
end

```

```
%cari min selisih
min = 19999999
idmin=0

for ipm=1:conter
    if selisihuji(ipm)<min
        min=selisihuji(ipm);
        idmin = ipm;
    end
end
idmin
conter
['dikenali sebagai polamaster = ' simpannamafile{idmin}]
['pengenalan selesai']
```

©UKDW

LAMPIRAN D

KARTU KONSULTASI

|  Kartu Konsultasi Tugas Akhir Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana Yogyakarta Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929 | | | | | | | | | | | | | |
|---|---------------------------|---|---|---|--|--|--|----------|---------------------------|---|--|--|--|
| NIM : ESTER ERNITA WATI SIREGAR Judul : IMPLEMENTASI METODE BACKPROPAGATION UNTUK MENGLASIFIKASIKAN JENIS BUAH PEAR Dosen Pembimbing I : Dra. Widi Hapsari, M.T. | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">1</td> <td>Tanggal: 13 Sept</td> <td>Paraf: </td> </tr> <tr> <td colspan="3"> <ul style="list-style-type: none"> o jurnal o zoning o pendisaan o dll. </td> </tr> </table> | 1 | Tanggal: 13 Sept | Paraf:  | <ul style="list-style-type: none"> o jurnal o zoning o pendisaan o dll. | | | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">2</td> <td>Tanggal: 20 Februari</td> <td>Paraf: </td> </tr> <tr> <td colspan="3"> <ul style="list-style-type: none"> - program - valikits - uji. </td> </tr> </table> | 2 | Tanggal: 20 Februari | Paraf:  | <ul style="list-style-type: none"> - program - valikits - uji. | | |
| 1 | Tanggal: 13 Sept | Paraf:  | | | | | | | | | | | |
| <ul style="list-style-type: none"> o jurnal o zoning o pendisaan o dll. | | | | | | | | | | | | | |
| 2 | Tanggal: 20 Februari | Paraf:  | | | | | | | | | | | |
| <ul style="list-style-type: none"> - program - valikits - uji. | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">3</td> <td>Tanggal: 09 April 2018</td> <td>Paraf: </td> </tr> <tr> <td colspan="3"> <ul style="list-style-type: none"> o Konsultasi program o Konsultasi bab 3 </td> </tr> </table> | 3 | Tanggal: 09 April 2018 | Paraf:  | <ul style="list-style-type: none"> o Konsultasi program o Konsultasi bab 3 | | | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">4</td> <td>Tanggal: 30 April 2018</td> <td>Paraf: </td> </tr> <tr> <td colspan="3"> <ul style="list-style-type: none"> o Konsultasi program o Konsultasi bab 4 </td> </tr> </table> | 4 | Tanggal: 30 April 2018 | Paraf:  | <ul style="list-style-type: none"> o Konsultasi program o Konsultasi bab 4 | | |
| 3 | Tanggal: 09 April 2018 | Paraf:  | | | | | | | | | | | |
| <ul style="list-style-type: none"> o Konsultasi program o Konsultasi bab 3 | | | | | | | | | | | | | |
| 4 | Tanggal: 30 April 2018 | Paraf:  | | | | | | | | | | | |
| <ul style="list-style-type: none"> o Konsultasi program o Konsultasi bab 4 | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">5</td> <td>Tanggal: 7 Mei 2018</td> <td>Paraf: </td> </tr> <tr> <td colspan="3"> <ul style="list-style-type: none"> o Konsultasi bab 1-5 o </td> </tr> </table> | 5 | Tanggal: 7 Mei 2018 | Paraf:  | <ul style="list-style-type: none"> o Konsultasi bab 1-5 o | | | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">6</td> <td>Tanggal: 17 Mei 2018</td> <td>Paraf: </td> </tr> <tr> <td colspan="3" style="text-align: center;">Acc Pendadaran</td> </tr> </table> | 6 | Tanggal: 17 Mei 2018 | Paraf:  | Acc Pendadaran | | |
| 5 | Tanggal: 7 Mei 2018 | Paraf:  | | | | | | | | | | | |
| <ul style="list-style-type: none"> o Konsultasi bab 1-5 o | | | | | | | | | | | | | |
| 6 | Tanggal: 17 Mei 2018 | Paraf:  | | | | | | | | | | | |
| Acc Pendadaran | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">7</td> <td>Tanggal:</td> <td>Paraf:</td> </tr> <tr> <td colspan="3" style="height: 40px;"></td> </tr> </table> | 7 | Tanggal: | Paraf: | | | | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">8</td> <td>Tanggal:</td> <td>Paraf:</td> </tr> <tr> <td colspan="3" style="height: 40px;"></td> </tr> </table> | 8 | Tanggal: | Paraf: | | | |
| 7 | Tanggal: | Paraf: | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| 8 | Tanggal: | Paraf: | | | | | | | | | | | |
| | | | | | | | | | | | | | |



Kartu Konsultasi Tugas Akhir

Program Studi Teknik Informatika Fakultas Teknologi Informasi
 Universitas Kristen Duta Wacana Yogyakarta
 Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929

NIM : ESTER ERNITA WATI SIREGAR
 Judul : IMPLEMENTASI METODE BACKPROPAGATION UNTUK MENGLASIFIKASIKAN JENIS BUAH PEAR
 Dosen Pembimbing II : Dr. Ir. Sri Suwarno, M.Eng.

| | | | | | |
|---------------------------------|---------------------------|------------|---|--|------------|
| 1 | Tanggal: 15 Sept. 2017 | Paraf: | 2 | Tanggal: 13 Okt 2017 | Paraf: |
| Bab I revisi. | | | | Bab II OK | |
| 3 | Tanggal: 25 Okt 2017 | Paraf: | 4 | Tanggal: 29 Jan 18 | Paraf: |
| Bab III OK | | | | Konsultasi konsep program | |
| 5 | Tanggal: 15 Feb 2018 | Paraf: | 6 | Tanggal: 4 Mei 2018 | Paraf: |
| Konsultasi konsep | | | | Bab 4 & 5 OK → revisi tabel & selang tulis. | |
| 7 | Tanggal: 9 Mei 2018 | Paraf: | 8 | Tanggal: 16 Mei 2018 | Paraf: |
| Bab 4 & 5 perlu revisi tulisan. | | | | OK setelah diadakan | |



Program Studi Informatika
Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana Yogyakarta
Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929

FORMULIR PERBAIKAN (REVISI) SKRIPSI
Strata-1 Program Studi Informatika

Yang bertanda tangan di bawah ini:

Nama : ESTER ERNITA WATI SIREGAR
N I M : 71130116
Judul Skripsi : IMPLEMENTASI METODE BACKPROPAGATION UNTUK
MENGKLASIFIKASIKAN JENIS BUAH PEAR
Tanggal Pendadaran : 31 Mei 2018 13:00 WIB

Telah melakukan perbaikan tugas akhir dengan lengkap.

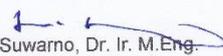
Demikian pernyataan kami agar dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 Juni 2018

Dosen Pembimbing I


Widi Hapsari, Dra. M.T.

Dosen Pembimbing II


Sri Suwarno, Dr. Ir. M.Eng.

Di cetak tanggal: 22 Juni 2018 08:29 WIB

©UKDW