

Article history

Received Mar 27, 2020

Accepted May 30, 2020

EKSTRAKSI FITUR UNTUK MENGIDENTIFIKASI MARGA TANAMAN MENGUNAKAN ALGORITMA BACKPROPAGATION

Sunu Jatmika¹, Tria Aprilianto², Muhammad Idris³

^{1,2} Fakultas Teknologi dan Desain, Prodi Sistem Komputer, Institut dan Teknologi Bisnis Asia Malang

³ Fakultas Teknologi dan Desain, Prodi Teknik Informatika, Institut dan Teknologi Bisnis Asia Malang
sunu.srg@gmail.com, raptorapril@gmail.com, m.idris995@gmail.com

ABSTRACT

Feature extraction is the beginning to be able to classify and interpret images by linking the characteristics of the leaves into a group of clans according to their type. The algorithm used is backpropagation based on shape and color. The formulation of the problem taken is how the use of the backpropagation algorithm can improve the quality of the community in identifying leaf images. The purpose of this study is to make it easier for the general public to recognize plants, especially the family Azadirachta, Swietenia, and Khaya.

This study uses data collection techniques in the form of observation, interviews, and documentation. Data analysis is done by entering content into the system. Data will be input into the learning machine obtained from feature extraction and processed with the backpropagation method. System design uses backpropagation algorithm to classify plants through leaf features. This system uses Android Studio and SQLite databases.

The results of this study are that of 9 test data there are 8 recognizable data and 1 incorrectly recognized data. The data shows the accuracy of the backpropagation algorithm in facilitating the general public to recognize plants, especially the family Azadirachta, Swietenia, and Khaya is 88.9%. In addition, the results of the questionnaire show that the backpropagation algorithm has 66% application benefits, 76% ease of interaction, and 80% application display. The overall average of the benefits of each aspect is 74.2%.

Keywords : Feature Extraction, Plant Identification, Geometric Feature, Color Feature, Leaf Image, Backpropagation Algorithm

ABSTRAK

Ekstraksi fitur adalah awal untuk dapat mengklasifikasi dan menginterpretasi citra dengan mengaitkan karakteristik daun kedalam sekelompok marga yang sesuai dengan jenisnya. Algoritma yang digunakan adalah backpropagation yang didasarkan pada bentuk dan warna. Rumusan masalah yang diambil adalah bagaimanakah penggunaan algoritma backpropagation dapat meningkatkan kualitas masyarakat dalam mengidentifikasi citra daun. Tujuan penelitian ini adalah mempermudah khalayak umum untuk mengenali tanaman khususnya marga Azadirachta, Swietenia, dan Khaya.

Penelitian ini menggunakan teknik pengumpulan data berupa observasi, wawancara, dan dokumentasi. Analisis data dilakukan dengan memasukkan konten kedalam sistem. Data akan di inputkan ke mesin learning yang didapat dari ekstraksi fitur dan di proses dengan metode backpropagation. Perancangan sistem menggunakan algoritma backpropagation untuk mengklasifikasikan tanaman melalui fitur daun. Pada sistem ini menggunakan Android Studio dan database SQLite.

Hasil dari penelitian ini adalah dari 9 data pengujian terdapat 8 data yang dikenali dan 1 data yang salah dikenali. Data tersebut menunjukkan tingkat keakurasian algoritma backpropagation dalam mempermudah khalayak umum untuk mengenali tanaman khususnya marga *Azadirachta*, *Swietenia*, dan *Khaya* adalah 88,9%. Selain itu, hasil kuesioner menunjukan algoritma backpropagation memiliki kemanfaatan aplikasi 66%, kemudahan interaksi 76%, dan tampilan aplikasi 80%. Rata-rata keseluruhan dari manfaat tiap aspek adalah 74,2%.

Kata Kunci : *Ekstraksi Fitur, Identifikasi Tanaman, Fitur Bentuk, Fitur Warna, Citra Daun, Algoritma Backpropagation*

1. PENDAHULUAN

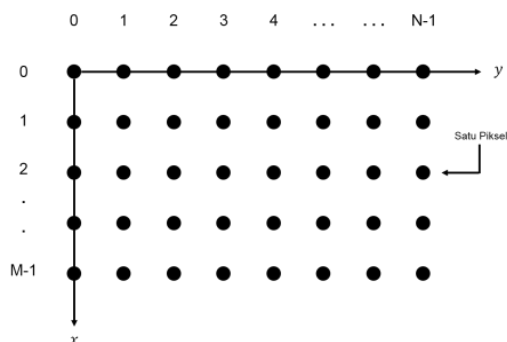
Semakin beragamnya nama citra serta ciri daun tanaman yang terdapat hingga saat ini, menyebabkan masyarakat awam kesulitan untuk mengidentifikasi jenis daun. Ekstraksi fitur menjadi awal untuk dapat melakukan klasifikasi dan interpretasi citra dengan mengaitkan karakteristik daun ke dalam sekelompok marga yang sesuai dengan jenisnya. Algoritma yang digunakan untuk mengidentifikasi marga tanaman menggunakan backpropagation yang didasarkan pada bentuk dan warna daun. Algoritma ini dapat dikembangkan dengan berbagai cara dan dapat mengidentifikasi berbagai macam pola yang dimasukkan ke dalam jaringan syaraf tiruan.

2. LANDASAN TEORI

Pengolahan Citra Digital

Citra adalah suatu gambaran atau kemiripan dari suatu objek. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses oleh komputer secara langsung. Tentu agar bisa diproses di komputer, citra analog harus dikonversi menjadi citra digital. Citra digital adalah citra yang dapat diolah oleh komputer. Sedangkan citra yang dihasilkan dari peralatan digital (citra digital) langsung bisa diolah oleh komputer.

Secara umum, sistem koordinat yang dipergunakan untuk mewakili citra dalam teori pengolahan citra seperti gambar 1 Sebuah citra digital diwakili oleh matriks yang terdiri dari M baris dan N kolom, dimana perpotongan antara baris dan kolom disebut piksel. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah f(x,y), yaitu besar intensitas atau warna dari piksel di titik tersebut.



Gambar 1. Sistem Koordinat Citra

Pre-processing

Preprocessing merupakan serangkaian teknik pengolahan citra yang digunakan untuk menyiapkan citra untuk diproses lebih lanjut dalam ekstraksi / pengambilan nilai tertentu. Adapun langkah – langkah dalam proses ini terdiri dari :

a. Grayscale

Grayscale adalah matriks data yang nilai – nilainya mewakili intensitas setiap piksel berkisar antara 0 sampai dengan 255. Setiap piksel membutuhkan 8 bit memori. Proses perubahan citra berwarna menjadi citra grayscale memiliki rumus umum, yaitu :

$$grayscale = (red + green + blue)/3$$

b. Thresholding

Thresholding merupakan teknik yang sederhana dan efektif untuk segmentasi citra. Proses thresholding sering disebut dengan proses binerisasi. Pada proses ini, citra grayscale akan di ubah menjadi citra biner dengan ambang yang di tentukan. Salah satu pengambilan ambang (threshold) adalah dengan cara mengambil rata-rata dari warna citra grayscale. Adapun rumus perhitungannya sebagai berikut.

$$threshold = \frac{\sum_{i=1}^m \sum_{j=1}^n p_{mn}}{m \times n}$$

c. Thinning

Thinning adalah proses image morphology yang merubah bentuk asli binary image menjadi citra yang menampilkan batas-batas obyek/foreground hanya setebal satu piksel. Algoritma thinning secara iteratif menghapus piksel-piksel pada binary image, dimana transisi dari 0 ke 1 (atau dari 1 ke 0 pada konvensi lain) terjadi sampai dengan terpenuhi suatu keadaan dimana satu himpunan dari lebar per unit (satu piksel) terhubung menjadi suatu garis.

d. Region Filling

Region filling adalah algoritma morfologi dalam pemrosesan citra, yang berkaitan dengan pengisian wilayah dalam gambar dengan warna (obyek). Proses ini dilakukan untuk menandai obyek yang digunakan untuk ekstraksi fitur warna.

Ekstraksi Fitur

Ekstraksi Fitur merupakan pengambilan nilai tertentu dari suatu citra untuk mendapatkan informasi. Adapun fitur – fitur yang di ambil antara lain :

a. Slimness

Slimness merupakan rasio perbandingan antara panjang daun dan lebar daun Fitur ini memiliki rumus sebagai berikut :

$$slimness = \frac{width}{length}$$

Dimana length adalah panjang maksimum antara titik pada tepi daun (contour leaf). Sedangkan width adalah panjang maksimum antara titik-titik pada tepi daun yang vertikal ke length.

b. Roundness

Roundness merupakan perhitungan untuk menentukan kebulatan obyek. Fitur ini memiliki rumus sebagai berikut :

$$roundness = \frac{4 \pi A}{P^2}$$

Dimana A adalah wilayah / luas area daun dan P adalah keliling daun.

c. Dispersion

Fitur ini untuk menangani obyek yang memiliki bentuk tidak beraturan seperti daun. Fitur ini memiliki rumus sebagai berikut :

$$dispersion = \frac{\max(\sqrt{(x_1 - \bar{x})^2 + (y_1 - \bar{y})^2})}{\min(\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2})}$$

Dimana (\bar{x}, \bar{y}) adalah titik tengah dari obyek daun, dan (x_i, y_i) merupakan titik koordinat piksel pada tepi daun.

d. Mean

$$\mu = \frac{1}{M N} \sum_{i=1}^M \sum_{j=1}^N P_{ij}$$

Dimana M adalah panjang citra, dan N adalah lebar citra. MxN dapat disebut juga dengan A (luas area daun). P_{ij} merupakan nilai piksel pada koordinat i dan j.

e. Standard Deviation

$$\sigma = \sqrt{\frac{1}{M N} \sum_{i=1}^M \sum_{j=1}^N (P_{ij} - \mu)^2}$$

f. Skewness

$$\theta = \frac{\sum_{i=1}^M \sum_{j=1}^N (P_{ij} - \mu)^3}{M N \sigma^3}$$

Backpropagation

Backpropagation merupakan salah satu metode pada Jaringan Syaraf Tiruan.JST dengan layar tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu/beberapa layar tersembunyi diantara layar masukan dan keluaran. Meskipun penggunaan lebih dari satu layar tersembunyi memiliki kelebihan manfaat untuk beberapa kasus, tapi pelatihannya memerlukan waktu yang lama. Maka umumnya orang mulai mencoba dengan sebuah layar tersembunyi terlebih dahulu.

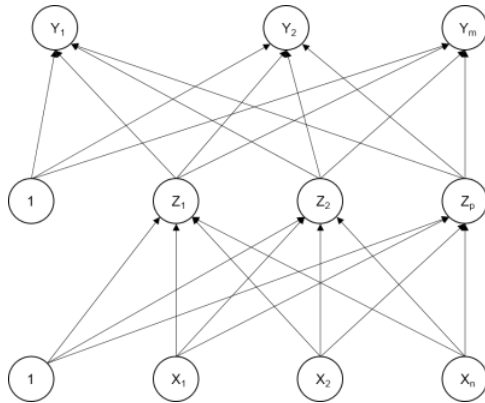
Seperti halnya model JST lain, backpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

a. Arsitektur Backpropagation

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. Gambar 2 adalah arsitektur backpropagation dengan n buah masukan (x) ditambah sebuah bias, sebuah layar tersembunyi (z) yang terdiri dai p unit ditambah sebuah bias, serta m buah unit keluaran (y).

V_{ij} merupakan bobot garis dari unit masukan Xi ke unit layar tersembunyi Zj

(V_j0 merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi Z_j). W_{kj} merupakan bobot dari unit layar tersembunyi Z_j ke unit keluaran Y_k (W_{k0} merupakan bobot bias di layar tersembunyi ke unit keluaran Y_k).

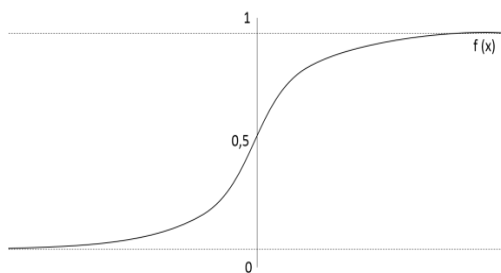


Gambar 2. Arsitektur Backpropagation

b. Fungsi Aktivasi

Dalam backpropagation, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat, yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi sigmoid biner yang memiliki range (0, 1).

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



Gambar 3. Fungsi Aktivasi

Fungsi lain yang sering dipakai adalah fungsi sigmoid bipolar yang bentuk fungsinya mirip dengan fungsi sigmoid biner, tapi dengan range (-1, 1).

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

c. Pelatihan Standar Backpropagation

Pelatihan backpropagation meliputi 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

Android Studio

Android Studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi android yang di kembangkan oleh IntelliJ IDEA. Setiap proyek di android studio berisi satu atau beberapa modul dengan kode sumber dan berkas pendukung. Jenis - jenis modul terdiri dari android app modules, library modules dan google app engine modules.

Basis data pada sistem ini menggunakan fasilitas internal dari aplikasi pengembang android studio, yaitu paket SQLite. Menyimpan data ke database (basis data) sangat ideal untuk sistem data terstruktur atau mengulangi pengambilan nilai di lain waktu, seperti informasi kontak. API (Android Programming Language) yang diperlukan untuk menggunakan basis data di android studio tersedia dalam paket android.database.sqlite.

3. PEMBAHASAN

Analisa Sistem

Analisa sistem didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan yang ada sehingga dapat menemukan perbaikan. Analisa sistem akan melalui beberapa proses yaitu analisa masalah, kebutuhan sistem, perancangan sistem

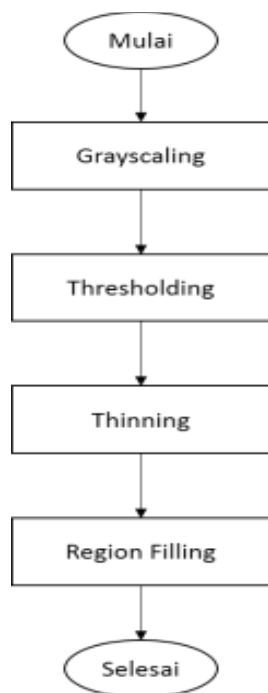
dan penerapan algoritma backpropagation dari hasil ekstraksi fitur pada citra daun.

Penggunaan algoritma backpropagation pada program ekstraksi fitur untuk identifikasi marga tanaman berbasis android memudahkan masyarakat awam untuk mengidentifikasi citra daun berdasarkan bentuk dan warna daun.

Perancangan Sistem

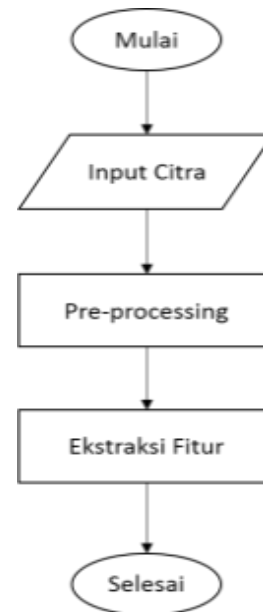
Perancangan sistem sangat diperlukan untuk membuat suatu aplikasi sesuai dengan latar belakang dan dapat membantu efisiensi dalam mengklasifikasikan tanaman. Dalam perancangan ini ada 2 pilihan proses pre-processing untuk mendapatkan fitur daun. Yaitu proses input citra melalui data storage, dimana background tidak memiliki nilai alpha atau nilai alpha bernilai 0 (hasil edit) atau background solid berwarna putih.

Perbedaannya hanya terdapat pada proses pre-processing-nya saja. Adapun alur pre-processing-nya tetap sama sebagai berikut.



Gambar 4. Flowchart Pre-processing

Proses berikutnya yaitu ekstraksi fitur, dimana dalam proses ini informasi yang akan di ambil masih dalam bentuk citra, sehingga membutuhkan proses pengambilan nilai fitur yang dibutuhkan. Proses ini memiliki alur seperti gambar 5 dibawah ini.



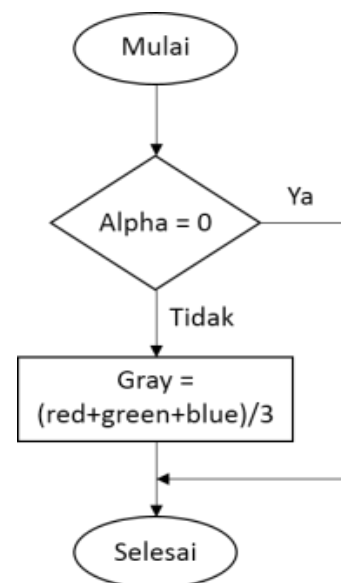
Gambar 5. Flowchart Ekstraksi Fitur

Pre-processing

Sebelum masuk pada proses ekstraksi, citra daun harus melalui tahap pre-processing yang perlu dilalui, yaitu.

a. Grayscale

Pada proses ini, nilai yang akan dihitung adalah nilai warna merah, hijau dan biru. Untuk nilai alpha yang bernilai 0, maka tidak dihitung, alur pada proses grayscale terdapat pada gambar 6.



Gambar 6. Flowchart Grayscale

Terdapat sampel citra berwarna (RGB) pada gambar 7. yang mewakili citra dari sebuah daun.

alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	alpha : 0	R : 106 G : 134 B : 93	R : 128 G : 154 B : 112	alpha : 0	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	R : 147 G : 169 B : 156	R : 135 G : 154 B : 109	R : 132 G : 155 B : 125	R : 104 G : 137 B : 90	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	R : 101 G : 129 B : 88	R : 167 G : 183 B : 162	R : 117 G : 143 B : 91	R : 100 G : 131 B : 90	R : 92 G : 125 B : 86	alpha : 0	alpha : 0
alpha : 0	R : 132 G : 155 B : 141	R : 97 G : 126 B : 94	R : 97 G : 126 B : 82	R : 122 G : 149 B : 92	R : 119 G : 146 B : 100	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	R : 121 G : 146 B : 102	R : 134 G : 157 B : 121	R : 111 G : 135 B : 100	R : 137 G : 157 B : 118	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	R : 107 G : 134 B : 88	R : 105 G : 136 B : 82	R : 113 G : 145 B : 110	alpha : 0	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	alpha : 0	R : 102 G : 129 B : 79	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0

Gambar 7. Sampel Citra RGB

Pada citra RGB, terdapat 4 nilai yang dapat diproses menjadi suatu citra tertentu. Pada variabel alpha yang bernilai 0, maka variabel red, green dan blue tidak tampak, dikarenakan variabel alpha merupakan variabel yang menunjukkan besarnya kesolidan pada warna tersebut. Semakin sedikit nilai alpha, semakin warna RGB tidak terlihat / samar.

Pada proses grayscale, variabel yang di ambil adalah red, green dan blue. Sedangkan variabel alpha, digunakan untuk mendeteksi apakah akan di proses dalam metode grayscale atau

tidak. Matrik perhitungan grayscale pada sampel citra RGB sebagai berikut.

$$grayscale = (red + green + blue)/3$$

Pada tahap ini terdapat 81 piksel . Akan tetapi, tidak semua piksel melakukan proses. Terdapat 24 piksel yang melakukan proses perhitungan. Berikut adalah hasil perhitungan dari piksel – piksel tersebut menggunakan rumus grayscale.

alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	alpha : 0	R : 111 G : 111 B : 111	R : 131 G : 131 B : 131	alpha : 0	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	R : 157 G : 157 B : 157	R : 133 G : 133 B : 133	R : 137 G : 137 B : 137	R : 110 G : 110 B : 110	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	R : 106 G : 106 B : 106	R : 171 G : 171 B : 171	R : 117 G : 117 B : 117	R : 107 G : 107 B : 107	R : 101 G : 101 B : 101	alpha : 0	alpha : 0
alpha : 0	R : 143 G : 143 B : 143	R : 106 G : 106 B : 106	R : 102 G : 102 B : 102	R : 121 G : 121 B : 121	R : 122 G : 122 B : 122	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	R : 123 G : 123 B : 123	R : 137 G : 137 B : 137	R : 115 G : 115 B : 115	R : 137 G : 137 B : 137	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	R : 110 G : 110 B : 110	R : 108 G : 108 B : 108	R : 123 G : 123 B : 123	alpha : 0	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	alpha : 0	R : 103 G : 103 B : 103	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0
alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0	alpha : 0

Gambar 8. Citra Grayscale

b. Thresholding

Proses thresholding bertujuan untuk mendapatkan suatu citra yang bernilai antara 2 kemungkinan (hitam dan putih). Secara umum, warna hitam digunakan untuk nilai obyek, dan putih untuk nilai background.

Pada thresholding akan ditemukan jenis daun yang teridentifikasi oleh software tersebut. Adapun output dari thresholding diatas adalah sebagai berikut.

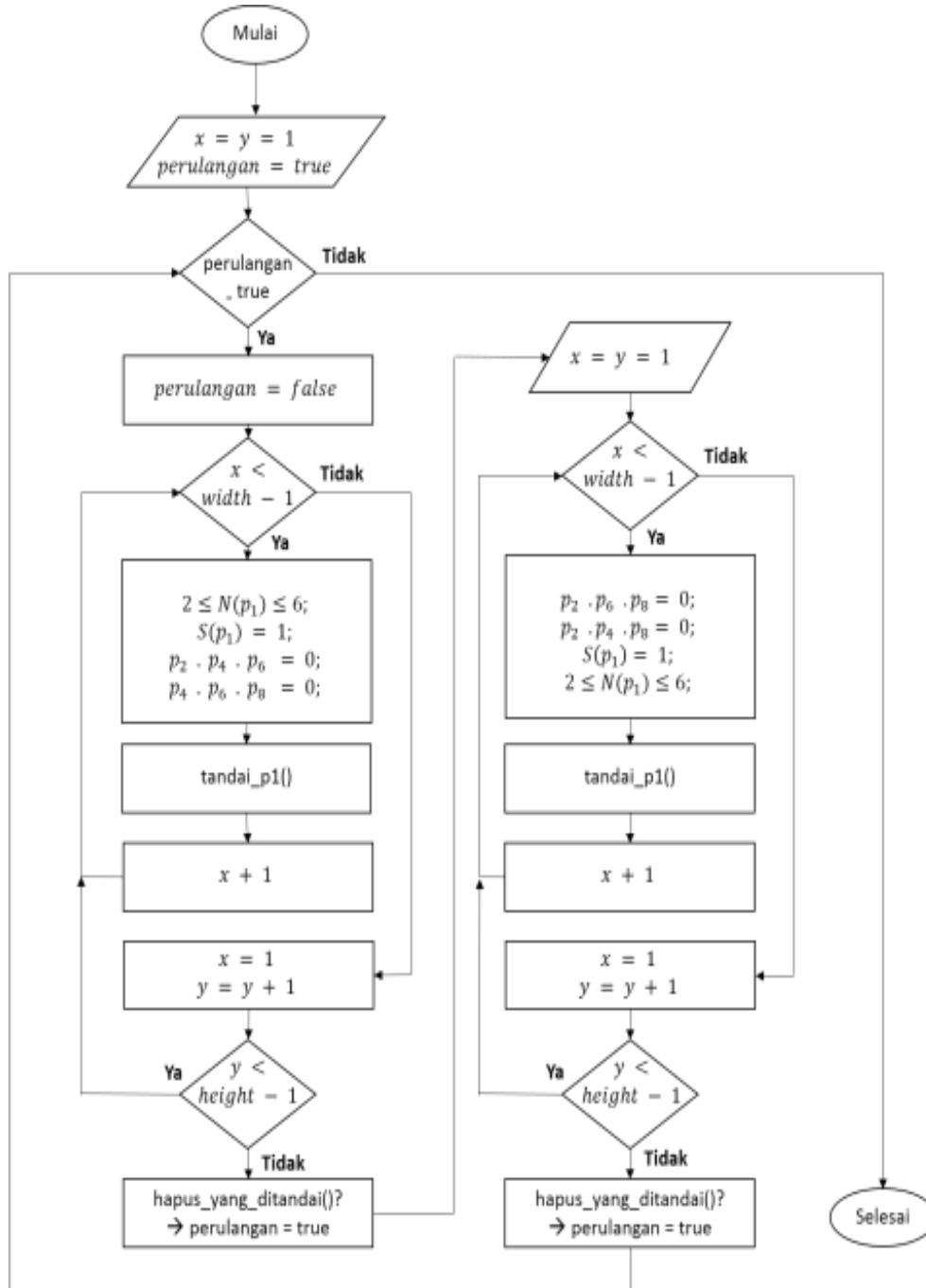
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255

Gambar 9. Citra Hasil Thresholding

c. Thinning

Proses thinning bertujuan untuk mendapatkan contour (garis luar) dari obyek daun. Sehingga dari citra hasil thinning dapat digunakan untuk proses ekstraksi fitur geometrik dan warna. Pada proses ini

diharapkan citra sudah tidak memiliki noise sehingga proses thinning tidak terganggu oleh noise dan citra sudah dalam bentuk hitam putih (hasil thresholding). Adapun diagram alur pada thinning terdapat pada gambar 10 sebagai berikut.



Gambar 10. Flowchart Thinning

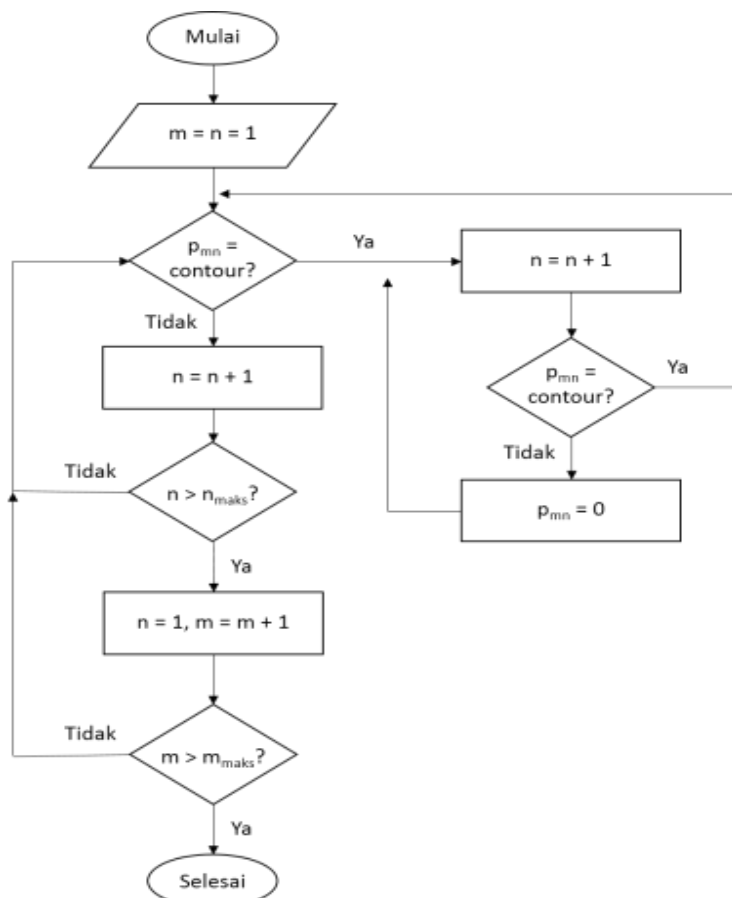
Flowchart pada gambar 10 melakukan penandaan kemudian penghapusan (pemberian warna background) pada piksel yang ditandai hingga dalam satu iterasi tidak terdapat perubahan piksel.

R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255

Gambar 11. Citra Hasil Thinning

d. Region Filling

Pada proses ini, citra hasil thinning di proses untuk mendapatkan obyek yang akan ditandai, yaitu dengan memberikan nilai 0 (obyek) di antara piksel – piksel contour.



Gambar 12. Flowchart Region Filling

Flowchart diatas menunjukkan alur region filling yang terjadi pada software identifikasi daun. Region filling akan menyeleksi daun menggunakan software tersebut. Adapun output dari region filling diatas adalah sebagai berikut.

R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 0 G : 0 B : 0	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255
R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255	R : 255 G : 255 B : 255

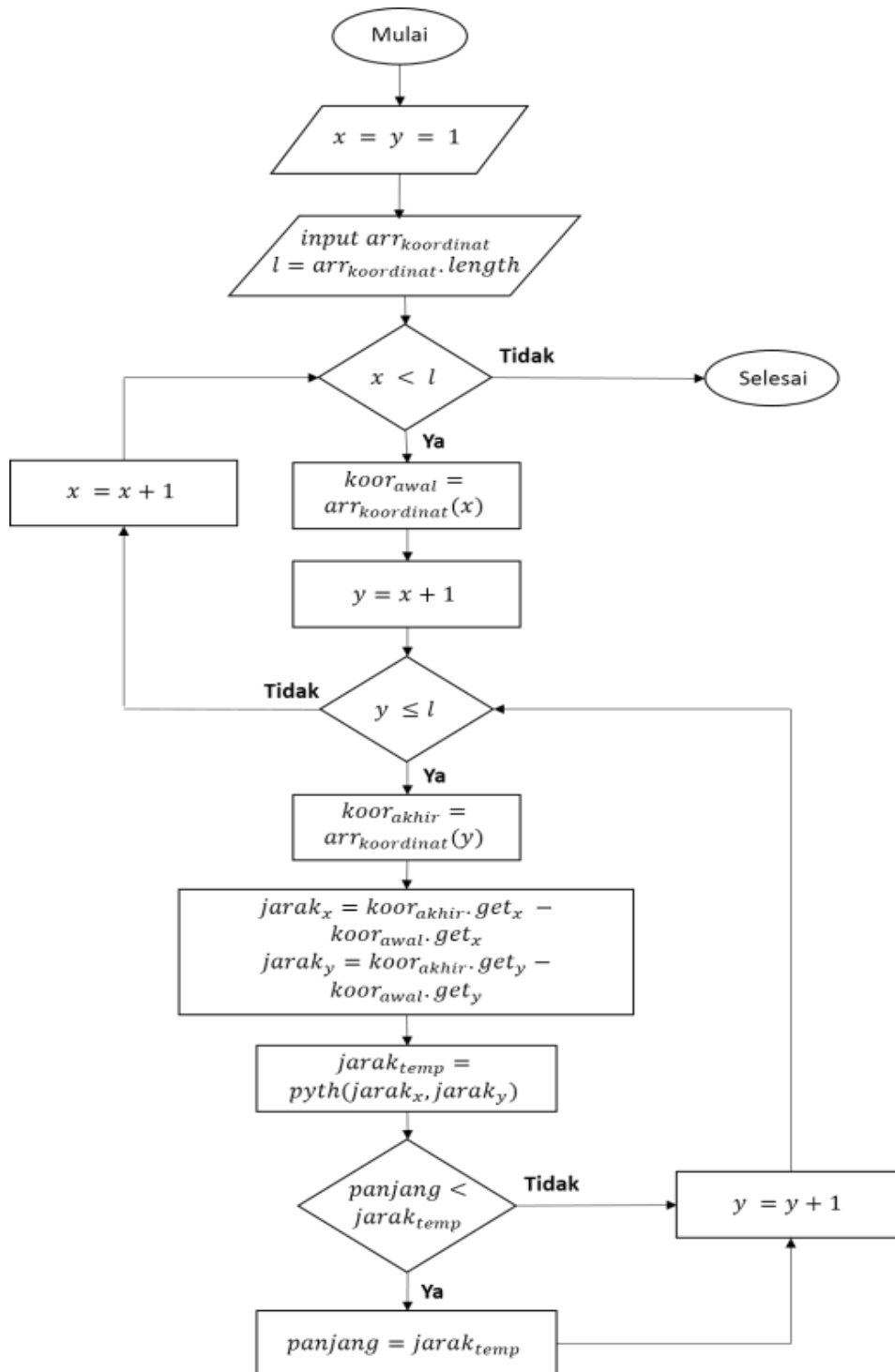
Gambar 13. Citra Hasil Region Filling

Ekstraksi Fitur

Ekstraksi fitur pada citra daun, melibatkan 3 fitur geometrik (bentuk) dan 3 fitur warna.

a. Slimness

Fitur slimness membutuhkan nilai panjang dan lebar. Alur untuk mendapatkan nilai panjang, terdapat pada flowchart 1.14 dibawah ini.

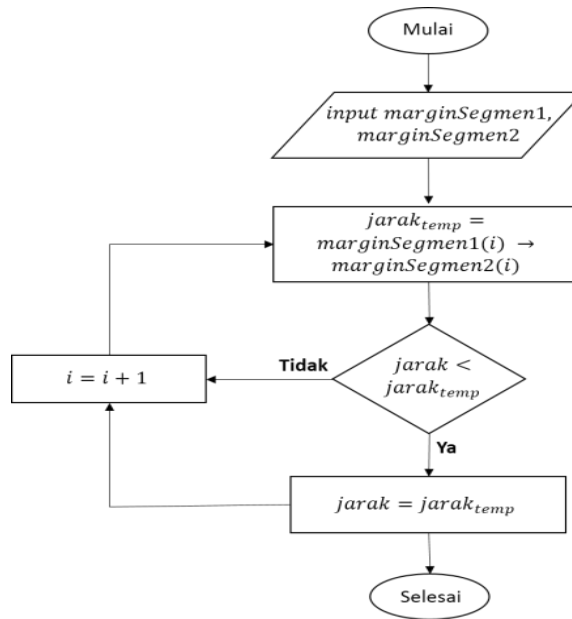


Gambar 14. Flowchart Panjang (*Length*)

Flowchart diatas menunjukkan proses pencarian panjang daun. Berikut adalah keterangan pencarian panjang. Sistem mendapatkan data koordinat dari obyek thinning, kemudian semua titik dihitung jarak koordinat x dan y serta dihitung jarak dengan

rumus pitagoras untuk mendapatkan panjang kemiringan dari titik ke titik yang dihitung.

Pada slimness selain mencari panjang, juga harus mengetahui lebar daun. Berikut adalah flowchart mencari lebar daun.



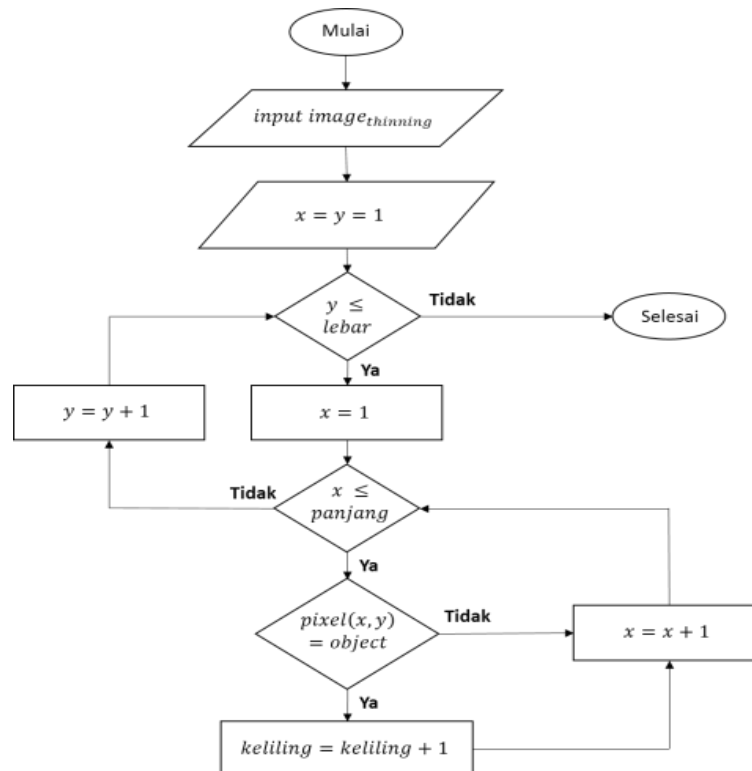
Gambar 15. Flowchart Lebar (*Width*)

Setelah mengetahui panjang dan lebar daun, kemudian dilakukan proses ekstraksi fitur slimness dengan rumus berikut.

$$slimness = \frac{length}{width}$$

b. Roundness

Fitur roundness membutuhkan nilai keliling dan luas daun. Alur untuk mendapatkan nilai keliling, terdapat pada flowchart 16 dibawah ini.

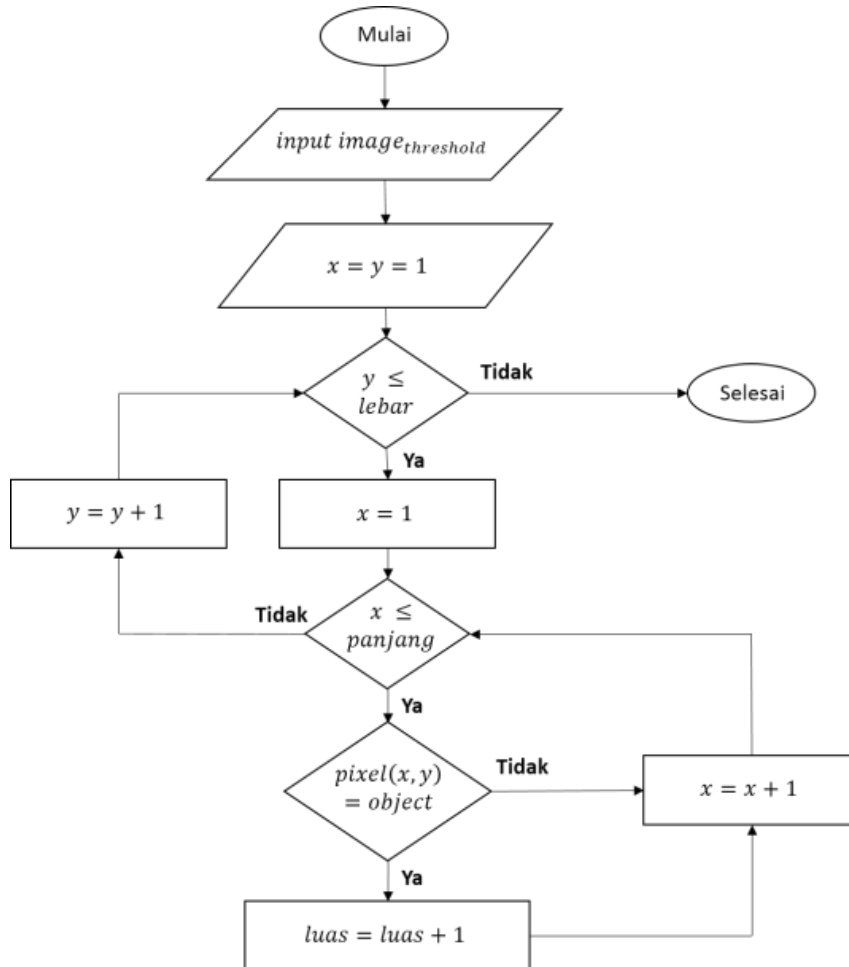


Gambar 16. Flowchart Keliling

Flowchart pada gambar 16 menunjukkan proses pencarian keliling daun. Proses tersebut membaca semua piksel dari citra hasil thinning, apabila piksel tersebut merupakan

obyek (warna bernilai nol) maka di jumlahkan hingga semua piksel di baca.

Pada roundness selain mencari keliling, juga harus mengetahui luas daun. Berikut adalah flowchart mencari luas daun.



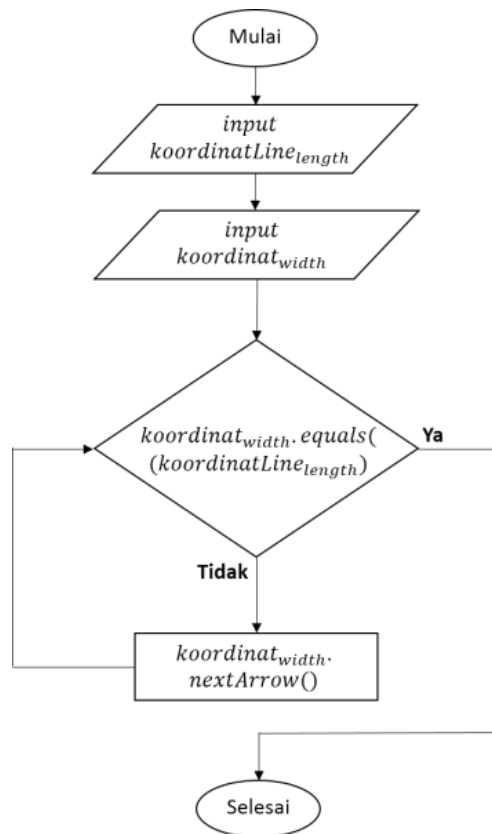
Gambar 17. Flowchart Luas

Berdasarkan flowchart pencarian luas pada gambar 17, sistem akan membaca semua piksel dari citra hasil threshold untuk di jumlahkan menjadi nilai luas. Apabila pada pembacaan piksel didapatkan nilai warna (merah, hijau atau biru), maka akan dihitung satu dan dijumlahkan dengan piksel obyek yang lain. Setelah mengetahui keliling dan luas daun, kemudian dilakukan proses ekstraksi fitur roundness dengan rumus berikut.

$$roundness = \frac{4 \pi A}{p^2}$$

c. Dispersion

Fitur dispersion membutuhkan nilai titik tengah. Alur untuk mendapatkan nilai titik tengah, terdapat pada flowchart 1.18 dibawah ini.



Gambar 18. Flowchart Titik Tengah

Flowchart diatas menunjukkan proses pencarian titik tengah dari citra daun. Proses tersebut membutuhkan data koordinat dari garis panjang dan koordinat dari titik width (lebar). Koordinat awal dari titik lebar ditambah/dikurangi dengan arah untuk menuju koordinat dari titik lebar yang kedua. Setiap titik koordinat berubah menuju titik kedua akan di cek kesamaannya dengan semua koordinat dari garis panjang hingga menemukan salah satu titik dari garis panjang. Setelah mengetahui titik tengah daun, kemudian dilakukan proses ekstraksi fitur dispersion dengan rumus berikut.

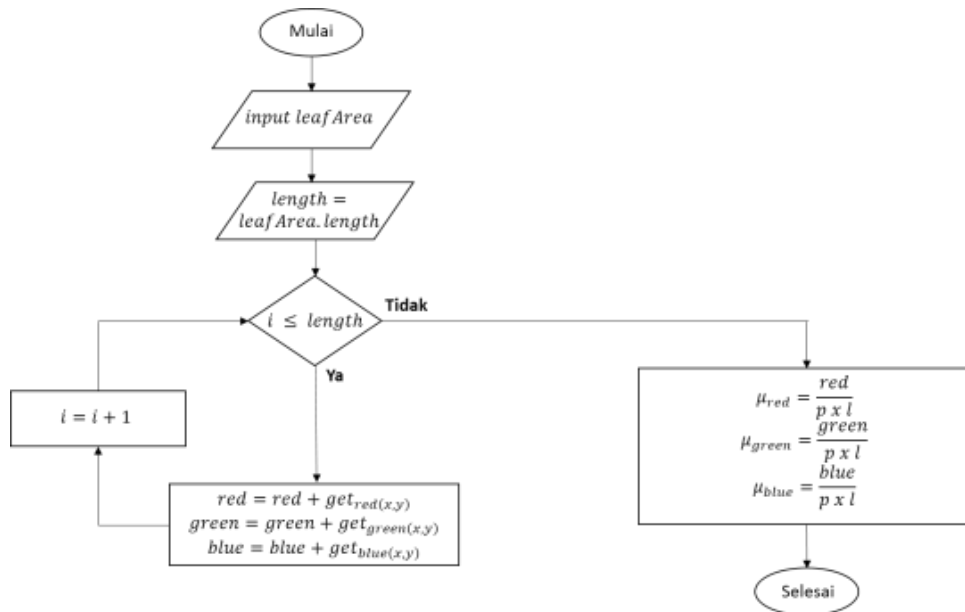
$$\begin{aligned}
 & dispersion \\
 & = \frac{\max(\sqrt{(x_1 - \bar{x})^2 + (y_1 - \bar{y})^2})}{\min(\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2})}
 \end{aligned}$$

(\bar{x}, \bar{y}) merupakan koordinat titik tengah, dan (x_i, y_i) merupakan koordinat setiap garis luar dari obyek daun.

Data yang telah diperoleh pada proses thinning akan diambil untuk mencari panjang dan lebar daun. Data ini kemudian akan dihitung menggunakan rumus dispersion untuk mengetahui fitur bentuk dispersion dari daun tersebut.

d. Mean

Fitur mean akan menangkap semua warna pada daun yang sudah terseleksi pada region filling. Setelah menangkap warna pada daun, kemudian warna tersebut akan dihitung rata – ratanya. Berikut adalah flowchart dari fitur mean.



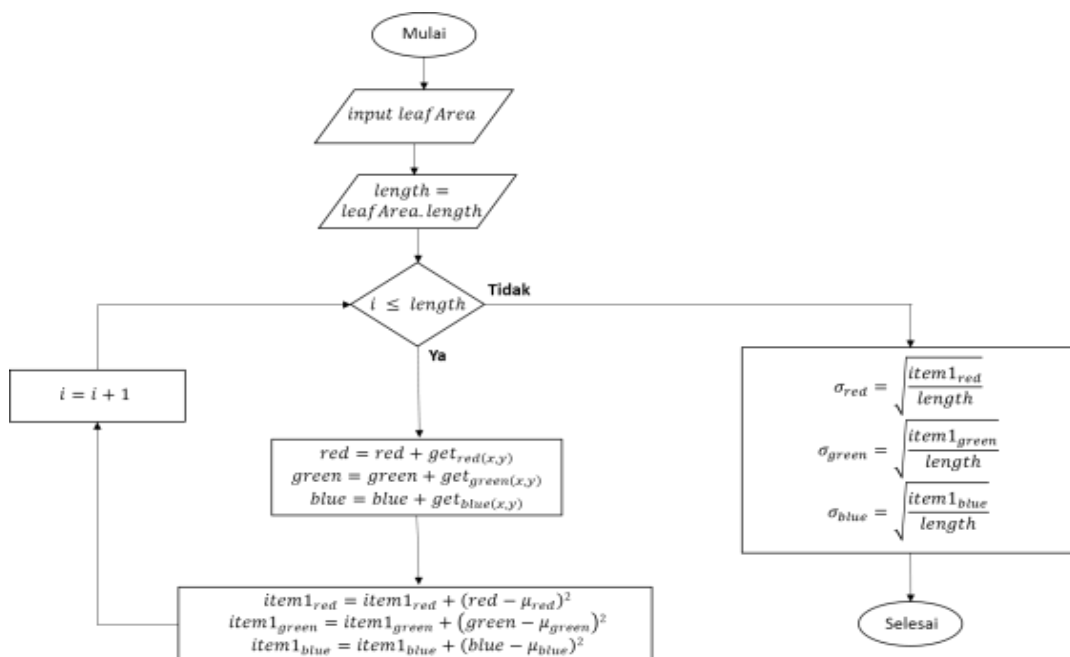
Gambar 19. Flowchart Mean

Flowchart pada gambar 19 menunjukkan proses pencarian mean daun. Proses tersebut akan menghitung semua piksel untuk mendapatkan nilai (setiap warna) rata-rata dari piksel yang dianggap obyek. Untuk mengetahui mean daun, dilakukan proses ekstraksi fitur mean dengan rumus berikut.

$$\mu = \frac{1}{M N} \sum_{i=1}^M \sum_{j=1}^N P_{ij}$$

e. Standard Deviation

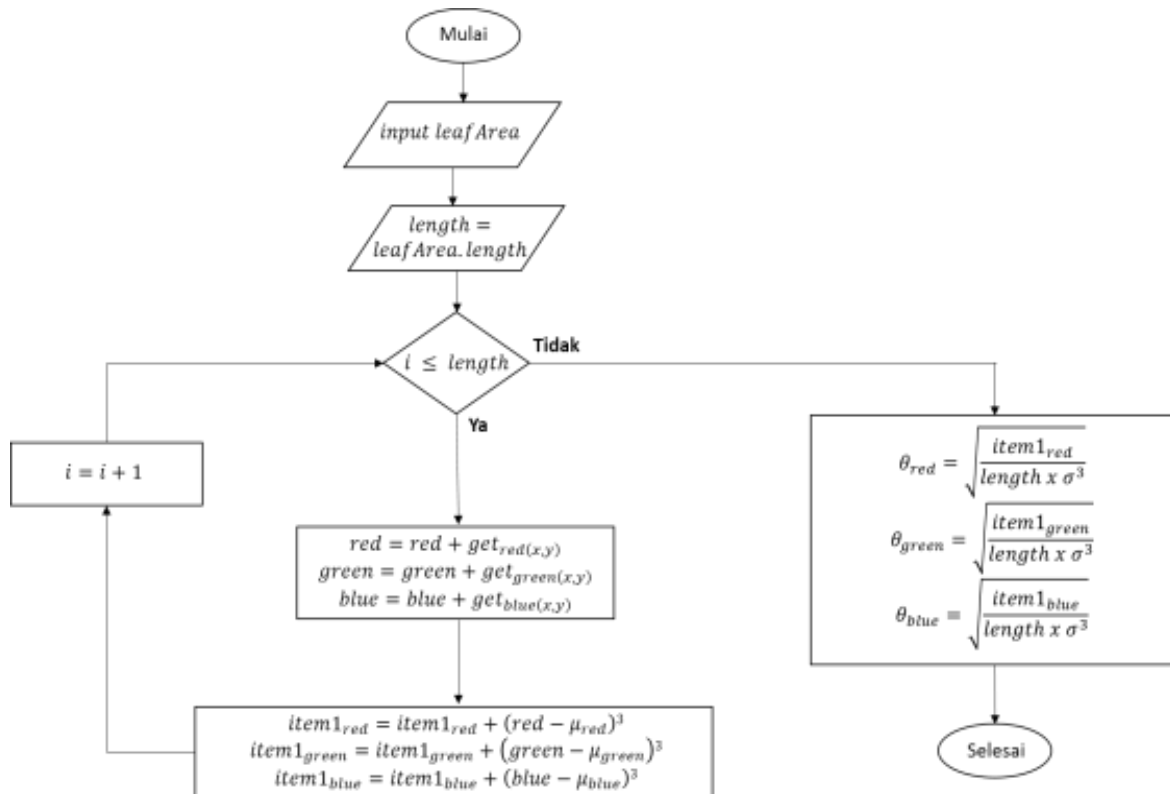
Fitur standard deviation akan menangkap semua warna pada daun yang sudah terseleksi pada region filling. Setelah menangkap warna pada daun, kemudian warna tersebut akan dihitung standard deviasinya. Berikut adalah flowchart dari fitur standard deviasi.



Gambar 20. Flowchart SD

Untuk mengetahui standard deviasi daun, dilakukan proses ekstraksi fitur standard deviasi dengan rumus berikut.

$$\sigma = \sqrt{\frac{1}{M N} \sum_{i=1}^M \sum_{j=1}^N (P_{ij} - \mu)^2}$$



Gambar 21. Flowchart Skewness

Untuk mengetahui skewnes daun, dilakukan proses ekstraksi fitur skewness dengan rumus berikut.

$$\theta = \frac{\sum_{i=1}^M \sum_{j=1}^N (P_{ij} - \mu)^3}{M N \sigma^3}$$

Pelatihan Sistem

Perancangan proses pelatihan menggunakan algoritma backpropagation menggunakan inputan dengan jumlah fitur yang digunakan, yaitu 12 fitur, yaitu slimness, roundness, dispersion, mean red, mean green, mean blue, standard deviation red, standard deviation green, standard deviation blue, skewness red, skewness green, dan skewness blue.

f. Skewness

Fitur skewness akan menangkap semua warna pada daun yang sudah terseleksi pada region filling. Setelah menangkap warna pada daun, kemudian warna tersebut akan dihitung skewnessnya. Berikut adalah flowchart dari fitur skewnes.

Sedangkan hidden layer yang digunakan dalam sistem identifikasi tanaman melalui fitur daun menggunakan 12 hidden layer. Serta menggunakan 2 output layer dengan kode target sebagai berikut.

Tabel 1. Kode Target Output Layer

No	Nama Tanaman	Kode
1	Azadirachta Indica	0 1
2	Swietenia Mahagoni	1 0
3	Khaya Senegalensis	1 1

5. IMPLEMENTASI

Kebutuhan Hardware dan Software

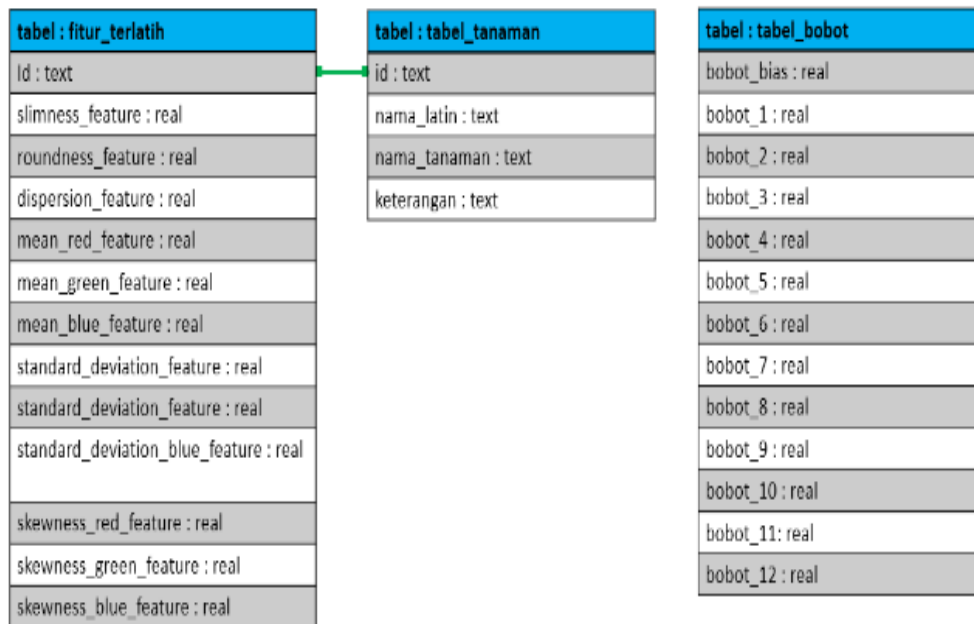
Untuk mengimplementasikan setiap algoritma menjadi satu kesatuan sistem, maka diperlukan hardware maupun software sebagai berikut.

- a. Spesifikasi Hardware
 - Processor A8-6410 2.0 GHz
 - Memory 4.00 GB RAM (3,46 usable)
 - Harddisk 350 GB HDD

- b. Spesifikasi Software
 - OS Windows 7 64-bit
 - Built Gradle 3.1.1
 - SDK Version 26 (Compile) – SDK Version 18 (Minimum)
 - Android Version 4.3 (Jelly Bean)

Perancangan Database

Database menggunakan SQLite, pada Gambar 22. merupakan rancangan database untuk membuat sistem identifikasi tanaman secara keseluruhan.



Gambar 22. Rancangan Database

Implementasi Inteface



Gambar 23. Halaman Utama (Home)

Bagian tombol LEARNING merupakan tombol yang mengarahkan ke halaman fitur citra daun terlatih dan pembuatan serta penyimpanan bobot. Sedangkan tombol TESTING merupakan tombol yang mengarahkan ke halaman ekstraksi fitur dari citra daun yang akan diidentifikasi.

Gambar 24 merupakan interface halaman pelatihan (learning). Pada halaman pelatihan terdapat tombol pada opsi 1, yaitu menyimpan bobot yang telah disediakan dari data latih sebelumnya tanpa melakukan pelatihan. Sedangkan tombol pada opsi 2 terdapat pernyataan-pernyataan algoritma pelatihan dengan bobot yang sudah di sediakan.



Gambar 24. Halaman Pelatihan

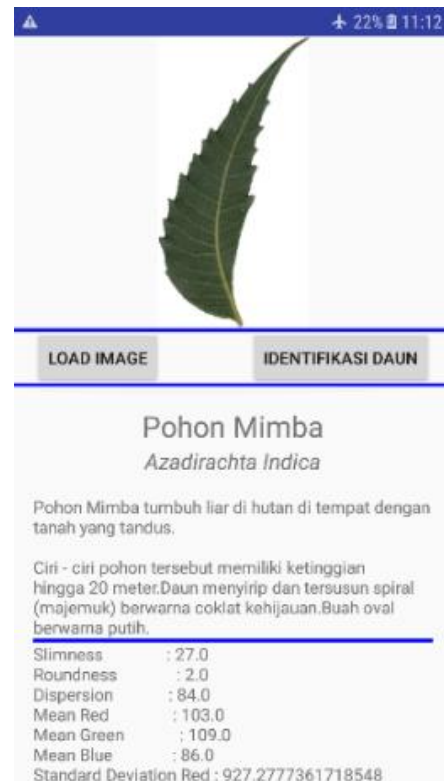
Tombol CITRA 1 untuk menampilkan keterangan yang berada di bawahnya antara lain nama tanaman, kode target dan fitur-fitur pada citra tersebut. Sama halnya dengan 8 tombol citra yang lainnya.

Fitur – fitur tersebut merupakan hasil dari ekstraksi fitur citra daun yang di ambil beberapa angka dibelakang koma. Sebelum proses pelatihan, beberapa fitur tersebut di normalisasikan supaya angka dari inputan tidak memiliki jarak yang terlalu jauh. Adapun fitur yang di normalisasikan sebagai berikut.

Tabel 2. Normalisasi Fitur

No	Nama Fitur	Normalisasi
1	Slimness	$slimness = slimness * 100$
2	Roundness	$roundness = roundness * 100$
3	Skewness Merah	$skewness_{red} = skewness_{red} * 1.000.000$
4	Skewness Hijau	$skewness_{green} = skewness_{green} * 1.000.000$
5	Skewness Biru	$skewness_{blue} = skewness_{blue} * 1.000.000$

Gambar 25. merupakan interface halaman pengujian (testing) yang dapat dilakukan setelah proses pelatihan selesai.



Gambar 25. Halaman Pengujian (Testing)

Proses pengujian dibutuhkan bobot hasil dari pelatihan yang sukses dilakukan. Adapun sampel nilai pada bobot tersebut sebagai berikut.

Tabel 3. Sampel Bobot

No	Bobot	Nilai
1	v_1	0.5884538691130501
2	v_2	0.054344192735858665
3	v_3	0.01858059314117011
4	v_4	0.01858059314117011
5	v_5	-0.23328758570292574
6	v_6	0.9356032905052036
7	v_7	-0.04760615476935459
8	v_8	-0.12877635229164375
9	v_9	-0.028701452224643043
10	v_{10}	0.1580305763741613
11	v_{11}	-0.6467310623746154
12	v_{12}	0.07455285847337123

Pengujian Sistem

Sistem identifikasi fitur daun yang dibuat kemudian diuji untuk mengukur tingkat keakurasiannya serta kemanfaatannya bagi pengguna dengan cara sebagai berikut.

- a. Terdapat 9 data pelatihan dan 9 data pengujian. Dari 9 data pengujian, terdapat 8 data yang dikenali dan 1 data yang salah dikenali. Sehingga dapat dihitung keakurasiannya dengan rumus :

$$akurasi = \frac{data_dikenali}{\sum data} \times 100\%$$

Tingkat keakurasiannya adalah $(8 / 9) * 100\% = 88,9\%$.

- b. hasil kuesioner menunjukkan algoritma backpropagation memiliki kemanfaatan aplikasi 66%, kemudahan interaksi 76%, dan tampilan aplikasi 80%. Rata-rata keseluruhan dari manfaat tiap aspek adalah 74,2%

6. KESIMPULAN

Berdasarkan penelitian yang telah dilaksanakan dapat ditarik beberapa kesimpulan. Ekstraksi fitur citra daun membutuhkan hasil dari pre-processing yang berbeda-beda dan akan berpengaruh apabila citra daun memiliki noise. Untuk output dalam sistem pengenalan obyek membutuhkan normalisasi nilai untuk menuju ke target (output layer).

Selain itu pelatihan sistem menggunakan algoritma backpropagation untuk mendapatkan bobot dengan bobot awal (random) membutuhkan beberapa kali percobaan dengan percobaan yang tidak dapat di tentukan.

Pengujian yang dilakukan terhadap sistem menggunakan algoritma backpropagation membutuhkan fitur citra daun yang mirip dengan fitur pada pelatihan sistem untuk mendapatkan persentase pengenalan yang lebih tinggi.

Untuk tingkat keakurasi aplikasi identifikasi tanaman melalui fitur citra daun dengan data latih sebanyak 9 citra, dan data uji sebanyak 9 citra. Didapatkan keakurasi mencapai 88,9%.

Hasil pengujian tingkat kemanfaatan terhadap 10 pengguna menunjukkan bahwa kemanfaatan aplikasi mencapai 66%,

kemudahan interaksi aplikasi mencapai 76%, dan tampilan aplikasi mencapai 80%.

7. DAFTAR PUSTAKA

- Ladjamudin, A. B. B. (2005). Analisis dan desain sistem informasi. Yogyakarta: Graha Ilmu, 1, 1-6.
- George, J., & Raj, S. G. (2017, August). Leaf recognition using multi-layer perceptron. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 2216-2221). IEEE.
- Purnomo, M. H., & Muntasa, A. (2010). Konsep pengolahan citra digital dan ekstraksi fitur. Yogyakarta: Graha Ilmu.
- Rideng, I. Made. 1989. Taksonomi Tumbuhan Biji.
- Wu, Q., Zhou, C., & Wang, C. (2006). Feature extraction and automatic recognition of plant leaf using artificial neural network. *Advances in Artificial Intelligence*, 3, 5-12.