

BAB IV ANALISIS DAN PERANCANGAN

Pada bab ini menjelaskan langkah-langkah analisa dan perancangan penelitian. Dengan membuat kerangka konsep penelitian dan menggunakan tahapan sesuai dengan metodologi penelitian.

4.1 Analisa Kebutuhan Sistem

Analisis kebutuhan sistem adalah menentukan dan mengungkapkan kebutuhan sistem. Kebutuhan sistem terbagi menjadi dua yaitu kebutuhan sistem fungsional dan kebutuhan sistem non-fungsional.

4.1.1 Deskripsi Sistem

Aplikasi Identifikasi Tanaman Apel Anna, Manalagi, dan Rome Beauty Menggunakan Metode Deteksi Tepi ACO dan ekstraksi fitur *Moment Invariant* merupakan aplikasi berbasis desktop menggunakan bahasa pemrograman *Python*. Aplikasi ini dapat membaca jenis daun apel anna, manalagi, dan rome beauty.

Sistem aplikasi Identifikasi Tanaman Apel Anna, Manalagi, dan Rome Beauty Menggunakan Metode Deteksi Tepi ACO dan ekstraksi fitur *Moment Invariant* ini terdapat beberapa proses dalam pengembangannya. terbagi menjadi 2 proses utama yaitu, *training* dan *testing*. Pada proses *training* pertama yaitu aplikasi menerima citra daun apel anna, manalagi, dan rome beauty. Kemudian citra tersebut masuk kedalam tahap *preprocessing* yaitu dimana citra akan dilakukan proses antaralain *resize*, kemudian *segmentasi thresholding*, dan perubahan citra ke *grayscale*. Dalam proses *segmentasi* terjadi proses pemisahan antara *background* dengan *foreground*. terdapat beberapa tahap dalam proses segmentasi, yaitu citra dilukan proses *threshold* untuk mengetahui ambang batas dari sebuah citra, lalu citra hasil *threshold* akan di *masking* berdasarkan citra sebelumnya, sehingga didapatkan citra yang tidak memiliki background. Setelah di dapatkan citra yang tersegmentasi. Tahap selanjutnya yaitu merubah citra dengan deteksi tepi menggunakan metode ACO. Kemudian setelah didapatkan hasil citra dari deteksi tepi, dilanjutkan dengan ekstraksi fitur menggunakan *Moment Invariant* dimana akan menghasilkan nilai dari hasil ekstraksi fitur, nilai tersebut disimpan dalam file database (.csv). Untuk proses testing merupakan proses klasifikasi dari hasil proses

training. Perbedaan dari proses *training* dan *testing* terletak di metode KNN, pada proses *training* tidak terdapat proses metode KNN. Untuk proses *preprocessing*nya sendiri sama dengan proses *training*. Pada proses *training* memerlukan dataset fitur citra daun apel anna, manalagi, dan rome beauty dari nilai hasil ekstraksi fitur akan menghasilkan bobot *training*. Selanjutnya bobot tersebut disimpan kedalam flat file database (.csv). Proses *testing* yaitu mencocokkan dengan bobot *training* dengan mengambil bobot terdekat dengan data *training*.

4.1.2 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan-kebutuhan yang memiliki keterkaitan langsung dengan sistem. Kebutuhan fungsional yang harus dipenuhi adalah sebagai berikut:

1. Sistem dapat menerima input citra daun apel dengan jenis tanaman apel Anna, Manalagi, dan Rome Beauty
2. Sistem dapat menampilkan hasil deteksi tepi
3. Sistem dapat menghasilkan nilai dari ekstraksi fitur.
4. Sistem dapat menyimpan nilai ekstraksi fitur.
5. Sistem dapat menampilkan hasil akhir yaitu klasifikasi jenis tanaman apel anna, manalagi, atau rome beauty.

4.1.3 Kebutuhan Non-fungsional

Kebutuhan non-fungsional adalah kebutuhan yang tidak secara langsung terkait dengan fitur tertentu di dalam sistem. Kebutuhan non-fungsional terbagi menjadi dua yaitu kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*Software*) sebagai berikut :

1. Kebutuhan Perangkat Keras (*Hardware*)

Adapun perangkat keras yang dibutuhkan untuk mendukung pengoprasian sistem yang dapat memenuhi spesifikasi minimal dari kebutuhan *hardware* sistem. Seperti pada tabel 4.1 di bawah ini:

Tabel 4.1 Kebutuhan Perangkat Keras

Komponen	Spesifikasi
Processor	Intel Core i5-8250U
RAM	8 GB
Harddisk	1 TB
Kamera	Resolusi minimum 8MP

2. Kebutuhan Perangkat Lunak (*Software*)

Adapun perangkat lunak yang dibutuhkan agar sistem dapat berdaun dengan baik serta mampu mendukung pengoprasian sistem. Seperti pada tabel 4.2 dibawah ini:

Tabel 4.2 Spesifikasi *Software*

Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows 10
Editor Pemrograman	JetBrain Pycharm Community Edition 2019
Dokumentasi dan Penyimpanan Data	Microsoft Office Word dan Excel

4.2 Perancangan Sistem

Perancangan sistem merupakan gambaran proses secara rinci sebuah sistem berjalan dengan tujuan untuk menghasilkan produk perangkat lunak sesuai dengan kebutuhan user. Ada beberapa tahap proses untuk pengolahan citra dalam membangun aplikasi ini. Antara lain

4.2.1 Tahap *Training*

Training dilakukan agar komputer dapat membaca dan menyimpan kriteria citra yang menjadi penentu kriteria daun apel yang akan diuji. Data yang dijadikan data latih adalah citra daun yang sudah ditentukan kriterianya. Di dalam data latih terdapat 3 kriteria, yaitu daun apel anna, manalagi, dan rome beauty. Berikut penjelasan dari tiap alur *training*:



Gambar 4. 1 *Flowchart Tahap Training*

1. Masukan Citra Data Latih

Merupakan langkah pengambilan data cira pada media masukan ke dalam sistem. Citra masukan berupa file gambar daun apel yang diambil menggunakan kamera *mirrorless* bermerek Canon EOS M10, dengan ukuran pixel 5184 x 2912 Pixel.

2. *Preprocessing*

Pada tahap *preprocessing* terdapat beberapa proses antara lain :

a) *Resize Citra*

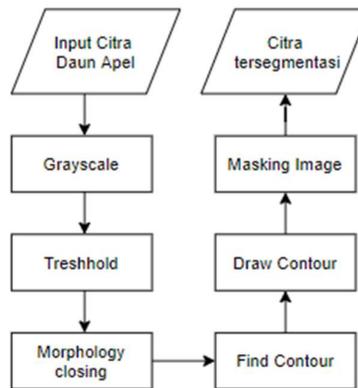
Pada proses *resize*, citra yang dimasukkan akan diresize menjadi 250 x 250 Pixel. *Resize* adalah tahap pertama dari *preprocessing*. *Resize* dilakukan untuk mempercepat dan memudahkan proses perhitungan (Yulianto Sugandi, 2015). Berikut alur proses dari *resize* :



Gambar 4. 2 Diagram proses *resize*

3. Segmentasi

Segmentasi merupakan proses untuk memisahkan *background* dengan *foreground*. Adapaun langkah langkah dari segmentasi dapat dilihat pada Gambar(4.3).



Gambar 4. 3 Diagram proses Segmentasi

Citra masukan dirubah menjadi citra *grayscale*, citra *grayscale* selanjutnya digunakan untuk mencari nilai ambang batas menggunakan metode *threshold*. Setelah didapat *image threshold*. Selanjutnya dilakukan proses *morphology closing* dengan pixel kernel *Rectangle (5x5)* bernilai 1. Tahap selanjutnya adalah mencari nilai kontur dari image *closing*. setelah didapat nilai kontur tahap selanjutnya adalah menggambar kontur berdasar nilai masking dan nilai kontur. Setelah itu dilakukan closing pada gambar kontur. Kemudian citra asli di masking dengan gambar kontur

sehingga di dapat citra tersegmentasi. Pada tahap segmentasi ini penulis menggunakan *library OpenCV* dalam mengerjakan tiap prosesnya.

4. Deteksi Tepi ACO

Pada penelitian ini menggunakan metode ACO sebagai metode deteksi adapun langkah dari deteksi tepi *ACO* sebagai berikut:

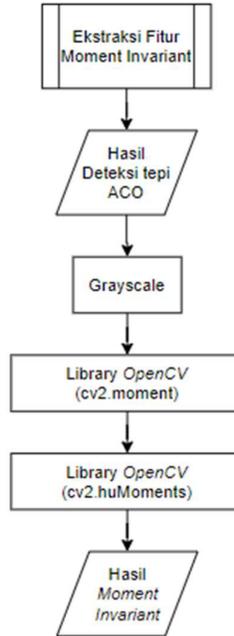


Gambar 4. 4 Diagram proses deteksi tepi ACO

Dalam menentukan deteksi tepi menggunakan metode ACO terdapat beberapa langkah yaitu inisialisasi parameter bertujuan untuk menginisialisasi variable yang dibutuhkan dalam metode ACO, lalu dilakukan perhitungan nilai gradient sebagai metode untuk penempatan semut, setelah dilakukan proses perhitungan gradient dilakukan pemilihan pixel selanjutnya untuk menentukan kemana semut akan berpindah, selanjutnya *update pheromone* dimana dalam proses ini dilakukan pembaruan *pheromone*. Setelah dilakukan *update pheromone* tahap terakhir adalah penentuan tepi berdasarkan nilai *update pheromone* dari setiap *pixel*

5. Ekstraksi Fitur *Moment Invariant*

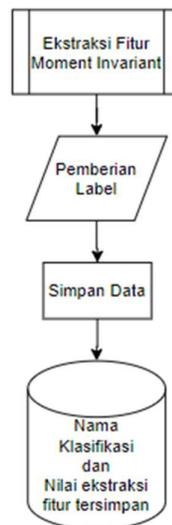
Ekstraksi fitur adalah proses untuk mendapatkan fitur ciri utama yang terdapat pada citra, citra deteksi tepi ACO yang akan digunakan pada tahap ini. Tahap ini akan menghitung beberapa parameter dimana penulis menggunakan *library OpenCV* dalam mengerjakan tiap prosesnya. Berikut diagram proses dari ekstraksi fitur :



Gambar 4. 5 Diagram proses ekstraksi fitur *Moment Invariant*

6. Input Klasifikasi

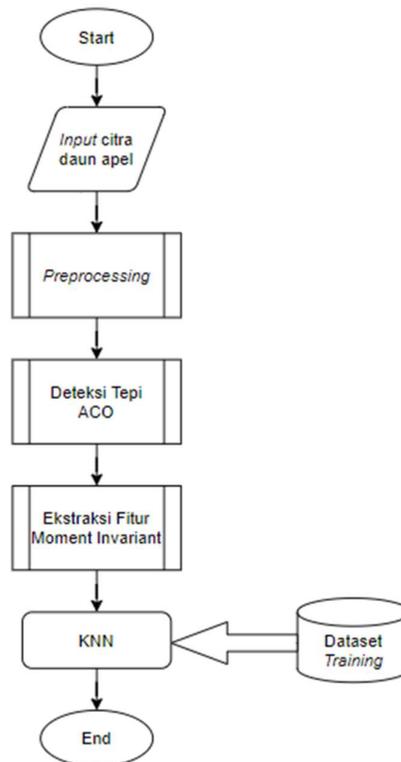
Nama klasifikasi digunakan untuk menentukan nama klasifikasi yang telah didapatkan nilainya per fitur dari ekstraksi fitur *moment invariant*. Data klasifikasi yang menjadi patokan akan disimpan dan dibandingkan dengan data uji. Data yang dilatih meliputi 3 kriteria yaitu daun apel anna, manalagi, dan rome beauty. Berikut diagram proses dari Input Klasifikasi :



Gambar 4. 6 Diagram proses Input Klasifikasi

4.2.2 Tahap Testing

Langkah dalam tahap *testing* atau pengujian merupakan proses klasifikasi dari hasil proses *training*. Perbedaan dari proses *training* dan *testing* terletak di metode KNN, pada proses *training* tidak terdapat proses metode KNN. Untuk proses *preprocessing*nya sendiri sama dengan proses *training* yaitu dilakukan input citra daun apel untuk dilakukan testing kemudian dilakukan proses *preprocessing* setelah itu dilakukan deteksi tepi menggunakan ACO kemudian dilakukan ekstraksi fitur menggunakan metode *Moment Invariant*. Setelah ekstraksi fiturnya sudah dilakukan maka akan dilakukan proses klasifikasi dengan menggunakan metode K-NN yang dimana sistem akan mengambil data dalam *database* terlebih dahulu yang kemudian akan dicocokkan dengan hasil dari ekstraksi fitur pada citra *testing*. Kemudian data uji akan dibandingkan dengan hasil ekstraksi data latih dengan menggunakan KNN *euclidean distance*.

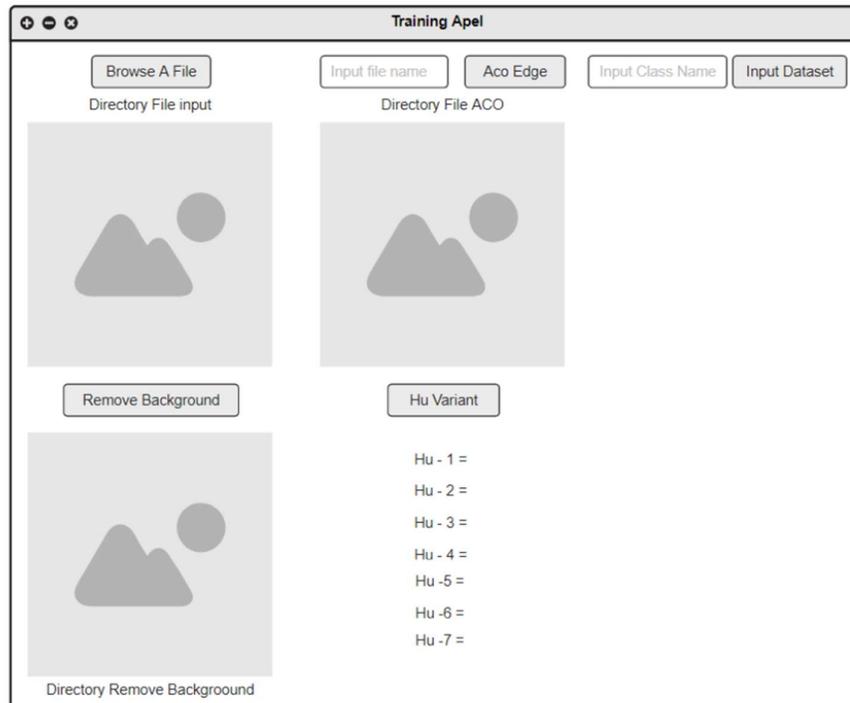


Gambar 4. 7 Flowchart Tahap Testing

Pada tahap testing, hasil ekstraksi dari data uji akan dibandingkan dengan ekstraksi data latih. Pada data latih terdapat 3 kriteria, jadi beberapa data uji yang diinputkan akan dicari nilai terkecil dengan kriteria- kriteria dari data latihnya

4.3 Perancangan *User Interface*

Perancangan *User Interface* bertujuan untuk memudahkan pengguna dalam mengoperasikan sistem dan mendapatkan informasi yang dibutuhkan dari sistem tersebut. Mockup sistem pelatihan dan pengujian dapat dilihat pada Gambar 4.5 dan Gambar 4.6.



Gambar 4. 8 *Mockup Form Sistem Training*

Pada gambar 4.5 terdapat tombol *Browse File* untuk memasukkan citra daun, label *Directory File* untuk menampilkan informasi dimana file citra berada. tombol *Remove Background* digunakan untuk segmentasi citra *Text Input file name* digunakan untuk memberi nama pada citra ACO yang telah di *input*. tombol ACO Edge digunakan untuk mengolah deteksi tepi ACO pada citra. Tombol Hu Variant digunakan untuk melakukan perhitungan nilai fitur dari *Moment Invariant*. Tombol *Input Dataset* digunakan untuk menyimpan nilai hasil ekstraksi dan *Text Input Class Name* digunakan untuk memberi nama sesuai kelas kedalam *database*. Kemudian terdapat juga picturebox untuk menampilkan daun yang akan diolah, label untuk menampilkan nilai hasil ekstraksi fitur dari *Moment Invariant*



Gambar 4. 9 *Mockup Form Sistem Testing*

Pada gambar 4.6 terdapat tombol *Browse File* untuk memasukkan citra daun, label *Directory File* untuk menampilkan informasi dimana file citra berada, tombol *Remove Background* digunakan untuk segmentasi citra, *Text Input file name* digunakan untuk memberi nama pada citra ACO yang telah di *input*. tombol ACO Edge digunakan untuk mengolah deteksi tepi ACO pada citra. Tombol *Browse Image ACO* digunakan untuk memasukkan citra deteksi tepi untuk mempercepat proses *Testing*, Tombol *Hu Variant* digunakan untuk melakukan perhitungan nilai fitur dari *Moment Invariant*. Tombol *KNN* digunakan untuk melakukan proses kalsifikasi daun apel anna, manalagi, rome beauty. Kemudian terdapat juga picturebox untuk menampilkan daun yang akan diolah, label hasil knn untuk menampilkan hasil dari klasifikasi.