

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kopi merupakan salah satu hasil komoditi hasil perkebunan yang memiliki nilai ekonomi yang cukup tinggi dan berperan penting sebagai sumber devisa negara. Selain sebagai sumber devisa negara, kopi juga merupakan sumber penghasilan bagi sebagian besar petani kopi di Indonesia. Kopi rakyat memegang peranan yang penting, karena terdapat nilai ekonomi yang cukup tinggi dan mengingat sebagian besar (93%) produksi kopi merupakan kopi rakyat. Namun demikian kondisi pengelolaan usaha tani pada kopi rakyat relatif masih kurang baik dibanding dengan kondisi perkebunan besar negara. Salah satu permasalahan utama yang diidentifikasi pada perkebunan rakyat yaitu mutu hasil yang kurang memenuhi syarat standarisasi [1]. Indonesia telah menerapkan standar mutu kopi biji berbasis uji fisik atas dasar jumlah nilai cacatnya sejak tahun 1990. Standar mutu ini telah mengalami beberapa kali revisi dan saat ini tertuang dalam Standar Nasional Indonesia (SNI) Nomor 01-2907-2008. Revisi standar mutu dilakukan untuk merespons dinamika tuntutan pasar domestic dan global yang terus berkembang. Sehingga, kriteria mutu dalam SNI harus selalu merujuk pada persyaratan internasional yang dikeluarkan oleh ICO (International coffee Organization). Sesuai keputusan ICO (*International Coffee Organization*) sejak tanggal 1 oktober 1983 sampai saat ini, untuk menetapkan mutu kopi di Indonesia menggunakan nilai cacat (*Defects Value System*). Dengan menggunakan sistem ini, semakin banyak nilai cacatnya maka mutu kopi akan semakin rendah dan semakin sedikit nilai cacatnya maka mutu kopi semakin baik[2].

Salah satu teknik yang paling mudah dalam menentukan cacat biji ialah dengan melihatnya secara langsung. Namun keputusan memilih biji kopi secara manual mengalami banyak kendala akibat adanya sifat subjektif dalam pemilihan biji kopi ataupun kurangnya pemahaman ilmu. Dalam membuat keputusan menentukan biji kopi berkualitas, perlu adanya sistem yang tepat untuk

menganalisa permasalahan tersebut. Metode *Content Based Image Retrieval* (CBIR) merupakan suatu solusi yang efektif bagi permasalahan memilih biji kopi yang sesuai syarat standarisasi. Metode CBIR dapat mendeteksi tekstur yang terdapat pada biji kopi. Hal itu dilakukan dengan melakukan pengambilan nilai fitur tekstur berbasis *Gray Level Co-Occurrence Matrix* (*contrast, correlation, homogeneity, dan energy*) yang diperoleh dari citra biji. Sehingga nantinya dihasilkan biji kopi yang sesuai dengan syarat standarisasi cacat biji.

Beberapa penelitian pernah dilakukan, seperti yang dilakukan [3] menggunakan metode CBIR (*Berbasis Content Based Image Retrieval*) dengan data citra yang digunakan sebanyak 90 data, dengan tingkat *precession* sebesar 55,20%. Jenis biji kopi yang digunakan diambil dari 3 jenis biji kopi yakni, biji kopi robusta, biji kopi arabika dan biji kopi liberika. Kelemahan dalam penelitian ini adalah data citra yang digunakan terlalu sedikit yaitu sekitar 30 data citra dari masing masing jenis biji kopi yang dipakai, sehingga tingkat akurasi yang didapat relatif kecil.

Penelitian yang dilakukan oleh [4] metode yang digunakan AHP (*Analytical Hierarchy Process*), jenis kopi yang dipakai adalah jenis arabika, output berupa kualitas biji kopi dengan input kadar air, cacat biji, dan ketinggian lahan dengan tingkat akurasi sebesar 85%. Kelemahan dari penelitian ini adalah 15% error berupa ketidak akuratan antara data perhitungan manual dengan perhitungan dengan metode AHP dari total data yang dibandingkan.

Penelitian oleh [5] menggunakan metode yang digunakan adalah Jaringan Saraf Tiruan (JST) *Backpropagation* dengan menggunakan data sebanyak 238 dimana 138 sebagai data latih dan 100 sebagai data uji. Mendapatkan tingkat akurasi terbaik dengan jumlah data uji 10 akurasi 80%.

## **1.2 Ruang Lingkup**

- 1.2.1 Biji kopi harus bersih dan tidak ada kulitnya.
- 1.2.2 Citra yang digunakan berupa gambar biji kopi berformat .JPG yang telah melalui proses cropping terlebih dahulu.
- 1.2.3 Sistem bersifat offline.

## **1.3 Rumusan Masalah**

- 1.3.1 Bagaimana mengimplementasikan *Content Based Image Retrieval* dengan metode Segmentasi?
- 1.3.2 Seberapa baik akurasi metode *Content Based Image Retrieval* dalam mengidentifikasi biji kopi jika dikombinasi dengan metode Segmentasi ?
- 1.3.3 Bagaimana merancang suatu sistem agar dapat mengklasifikasikan kualitas biji kopi?

## **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk melihat seberapa akurat ketelitian purwarupa sistem *content based image retrieval* yang dipadukan dengan metode segmentasi

## **1.5 Manfaat Penelitian**

Hasil dari penelitian ini diharapkan dapat memberi kontribusi terhadap perkembangan pengolahan citra digital dan membantu pihak yang berkepentingan dalam mengidentifikasi kualitas biji, khususnya biji kopi.

## BAB II

### KAJIAN TEORI

#### 2.1 Kajian Penelitian Terdahulu

Melakukan studi Pustaka dengan mengkaji jurnal-jurnal ilmiah dan berbagai sumber di internet mengenai penelitian yang berkaitan sebagai acuan untuk melakukan penelitian. Adapun referensi dalam penelitian ini sebagai berikut:

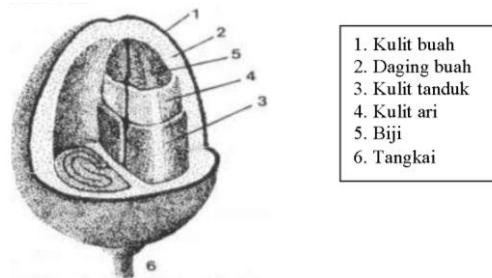
- 2.1.1 Penelitian CBIR pernah dilakukan oleh [6]. Penelitian ini menggunakan 250 citra dataset dari 5 jenis bunga lokal kota Tomohon, Sulawesi Utara. Pengujian dari penelitian ini dihitung dengan menggunakan *recall* dan *precision*, dengan hasil yang didapat yaitu rata-rata *recall* dengan nilai kemiripan mencapai 72.2% dan *precision* juga mencapai 72.2%.
- 2.2.2 Penelitian yang dilakukan oleh [7] menggunakan metode citra digital dikombinasikan dengan metode Jaringan Saraf Tiruan (JST) dan algoritma *Learning Vector Quantization* (LVQ). Data citra yang digunakan sebanyak 570 dengan 70% data training dan 30% data testing dengan tingkat akurasi sebesar 73,7%. Kelemahannya adalah masih terdapat kesalahan dalam pengenalan jenis kopi disebabkan oleh adanya benda lain dalam citra gambar yang diambil serta beberapa kesalahan metode dalam pengambilan gambar.
- 2.2.3 Penelitian oleh [8], data citra yang digunakan berjenis Arabica dengan jumlah data yang diambil secara acak. Menggunakan metode *thresholding*, output data image biji kopi akan diperlihatkan *list grading* dari tiap nilai biji kopi kemudian secara otomatis akan diklasifikasikan berdasarkan nilai dari biner *treshold* berdasarkan data histogram.
- 2.2.4 Penelitian yang menggunakan metode *Region Growing* pernah dilakukan [9]. Jumlah data citra yang digunakan sebanyak 12 data dengan tingkat akurasi yang didapat sebesar 83,33%.

2.2.5 Penelitian tentang *Region Growing* lainnya pernah dilakukan oleh [10]. Penelitian ini menggunakan metode *region growing* dengan parameter nilai tekstur yang didapat dari fitur kontras, homogenitas, entropi, dan energi pada metode *gray level co-occurrence matrix* (GLCM). Mendapatkan tingkat akurasi terbaik sebesar 84,7%.

## 2.2 Kopi

Kopi merupakan salah satu jenis tanaman perkebunan yang sudah lama dibudidayakan dan memiliki nilai ekonomis yang cukup tinggi. Kopi berasal dari Afrika, yaitu daerah pegunungan di Ethiopia. Konsumsi kopi dunia mencapai 70% berasal dari spesies kopi arabika dan 30% berasal dari spesies kopi robusta [11].

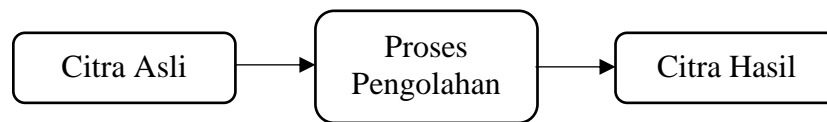
Buah kopi terdiri atas tiga bagian, yaitu : 1. lapisan kulit luar (exocarp) 2. lapisan daging (mesocarp) 3. lapisan kulit tanduk (endoscarp) Adapun susunan buah kopi disajikan pada Gambar 2.1 berikut ini :



**Gambar 2.1 Buah kopi**

## 2.3 Pengolahan Citra Digital

Pengolahan citra adalah suatu metode atau teknik yang dapat digunakan untuk memproses suatu data gambar yang diisikan untuk mendapatkan suatu informasi tertentu mengenai obyek yang diamati [12]. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan perkembangan dunia komputasi yang semakin meningkatnya dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra, maka image processing tidak dapat dilepaskan dari bidang computer vision.



**Gambar 2.2** Proses pengolahan citra

Berdasarkan warna-warna penyusunnya, citra digital dapat dibagi menjadi tiga macam [13] yaitu:

1. Citra biner, yaitu citra yang hanya terdiri atas dua warna, yaitu hitam dan putih. Oleh karena itu, setiap pixel pada citra biner cukup direpresentasikan dengan 1 bit. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengembangan, morfologi, ataupun dithering.
2. Citra grayscale, yaitu citra yang nilai pixel-nya merepresentasikan derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih. Pada umumnya citra grayscale memiliki kedalaman pixel 8 bit (256 derajat keabuan), tetapi ada juga citra grayscale yang kedalaman pixel-nya bukan 8 bit, misalnya 16 bit untuk penggunaan yang memerlukan ketelitian tinggi .
3. Citra berwarna, yaitu citra yang nilai pixel-nya merepresentasikan warna tertentu. Banyaknya warna yang mungkin digunakan bergantung kepada kedalaman pixel citra yang bersangkutan. Citra berwarna direpresentasikan dalam beberapa kanal (channel) yang menyatakan komponen- komponen warna penyusunnya. Banyaknya kanal yang digunakan bergantung pada model warna yang digunakan pada citra tersebut.

#### **2.4 JPEG (*Joint Photographic Experts Group*)**

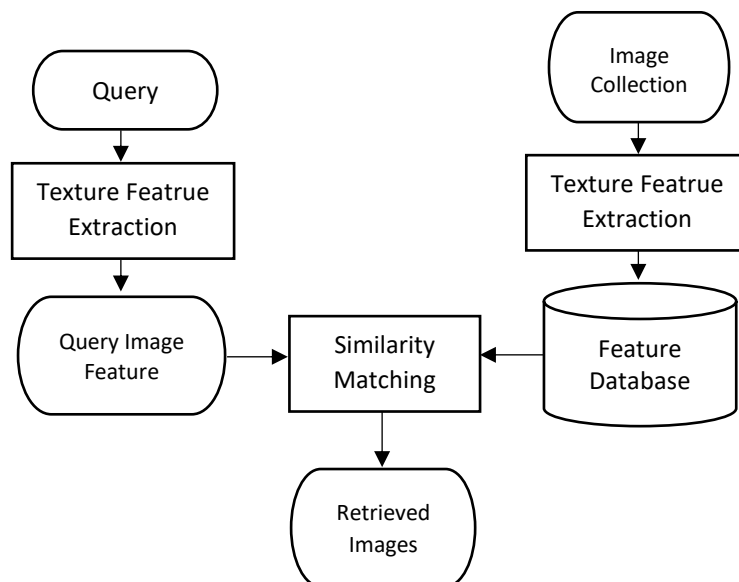
JPEG merupakan sebuah standar untuk format citra yang diperkenalkan oleh *Joint Photographic Experts Group*. JPEG grup merupakan hasil kerja sama antara *International Standardization Organization* (ISO) dan *International Electrotechnical Commission* (IEC). JPEG pertama sekali dibuat pada tahun 1986 dan diperkenalkan pada tahun 1992. Pengembangan dan pembaharuan

dilakukan seiring dengan berjalannya waktu sehingga pada tahun 1994 JPEG grup merilis versi terbaru JPEG. JPEG sering disebut sebagai standar yang cocok untuk algoritma kompresi *file* citra pada komputer dan menjadi salah satu format *file* citra yang didukung oleh *World Wide Web*. JPEG banyak digunakan untuk menyimpan gambar-gambar dengan ukuran *file* lebih kecil.[14]

## 2.5 Content Based Image Retrieval (CBIR)

*Content based image retrieval* (CBIR) adalah sebuah metode pencarian citra dengan melakukan perbandingan antara citra query dengan citra yang ada di database berdasarkan informasi yang ada pada citra tersebut (*Query by Example*).

CBIR juga dapat diartikan sebagai teknik untuk mencari gambar yang berhubungan dan mempunyai karakteristik dari suatu kumpulan gambar. CBIR secara umum dibangun dengan melihat karakteristik dari suatu gambar atau dengan kata lain dengan melihat ciri dari gambar tersebut. Ciri merupakan suatu tanda yang khas, yang membedakan antara satu gambar dengan gambar yang lain [15].



**Gambar 2.3** Alur *Content Based Image Retrieval*

## 2.5 Segmentasi Citra

Dalam pengolahan citra, terkadang kita menginginkan pengolahan hanya pada obyek tertentu. Oleh sebab itu, perlu dilakukan proses segmentasi citra yang bertujuan untuk memisahkan antara objek (*foreground*) dengan *background*.

Segmentasi citra adalah proses membagi citra digital menjadi beberapa daerah atau kelompok, dimana masing-masing daerah terdiri dari sekumpulan piksel. Segmentasi citra menyederhanakan dan merubah representasi citra kesesuatu yang lebih bermakna dan lebih mudah untuk dianalisis. Segmentasi citra digunakan untuk mencari obyek yang ingin dicari dan batas-batas bentuk objek seperti garis, kurva dalam citra [16].

## 2.6 Algoritma *Region Growing Segmentation*

*Region growing* adalah sebuah metode segmentasi citra berbasis daerah yang sederhana. Metode ini juga diklasifikasikan sebagai sebuah metode segmentasi citra berbasis piksel karena metode ini mencakup proses seleksi dari titik-titik awal. Pendekatan segmentasi ini memeriksa piksel tetangga dari titik-titik awal dan menentukan apakah piksel tetangga harus digabung ke daerah (*region*). Proses ini bersifat iterasi (perulangan), dengan perilaku yang sama dengan algoritma pengelompokan data (*data clustering*) biasa [17]. Dengan rumus sebagai berikut [18].

$$if |\mu R_{i-1} - I(x,y)| < T$$

Dimana :

T = nilai ambang batas yang ditentukan

I(x,y) = intensitas piksel kandidat

$\mu R$  = intensitas rata-rata piksel di dalam wilayah R

Dapat dijelaskan jika nilai tetangga yang dipilih menghasilkan nilai lebih kecil atau sama dengan nilai *threshold*, maka jadikan nilai tetangga yang dipilih tersebut menjadi *seed*, akan tetapi jika nilai lebih besar maka nilai tetangga yang dikunjungi tidak berubah.



Algoritma yang digunakan dalam metode *Region Growing* adalah sebagai berikut [19] :

1. Menentukan Seed Point Citra masukan yang merupakan hasil dari pre-processing akan di pilih satu piksel yang akan menjadi seed point.
2. Menentukan Tetangga dari Seed Point Seed point yang telah dipilih, lalu akan menentukan piksel tetangga dengan memilih delapan tetangga yang mengelilingi.
3. Memproses Piksel Tetangga Setelah mendapatkan delapan piksel tetangga, maka proses berikutnya adalah menyeleksi piksel. Setiap piksel tetangga akan dicek menurut warna keabuan dengan acuan seed point dan threshold.
4. Menentukan Foreground dan Background
5. Hasil Segmentasi Setelah semua piksel dicek, maka hasil segmentasi adalah nilai dari matrik foreground yang telah digabung dengan matrik background.

## **2.7 Ekstraksi Fitur**

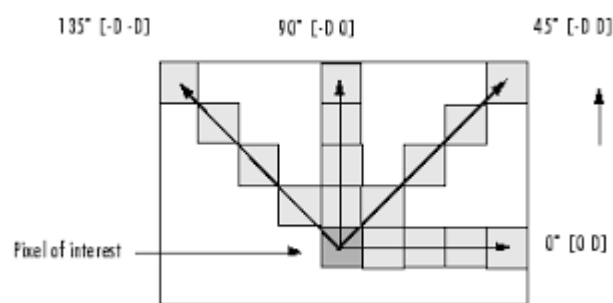
Ekstraksi fitur adalah proses pengambilan ciri sebuah objek yang dapat menggambarkan karakteristik dari objek tersebut [20]. Ekstraksi Fitur merupakan langkah awal dalam melakukan klasifikasi citra, bertujuan untuk memperoleh informasi yang terkandung dalam suatu citra dijadikan sebagai bahan acuan untuk membedakan antara citra yang satu dengan citra yang lain.

Ciri Fitur yang dapat diekstrak dari suatu objek citra antara lain adalah warna, bentuk, ukuran, dan tekstur. Ciri tersebut dapat digunakan sebagai parameter untuk membedakan antara objek-objek lainnya.

## **2.8 Gray Level Co-Occurrence Matrix (GLCM)**

GLCM suatu metode ekstraksi fitur citra yang banyak digunakan untuk klasifikasi citra yang cukup efektif dalam melakukan klasifikasi karena mampu memberikan informasi tekstur suatu citra secara detail. Ekstraksi tekstur dilakukan untuk mengambil informasi pokok dari suatu citra sebelum digunakan ke proses berikutnya. Metode GLCM menggunakan beberapa fitur pendekatan statistik seperti energi, entropi, kontras, dan sebagainya [21].

GLCM memiliki 4 arah sudut dalam ketetangaan, yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ . Sebagai ilustrasi, arah sudut ketetangaan dapat dilihat pada gambar 2.5. untuk sudut  $0^\circ$ , ketetangaan piksel dihitung dengan jarak 1 piksel ke arah kanan. Untuk sudut  $45^\circ$ , ketetangaan piksel dihitung dengan jarak 1 piksel ke kanan atas. Untuk sudut  $90^\circ$ , ketetangaan piksel dihitung dengan jarak 1 piksel ke atas. Dan untuk  $135^\circ$ , ketetangaan piksel dihitung dengan jarak 1 piksel ke kiri atas.



Gambar 2.5 Arah sudut GLCM

Terdapat empat Parameter tekstur yang sering digunakan dalam GLCM, adalah sebagai berikut:

### 1. Contrast

Kontras adalah suatu ukuran intensitas aras keabuan antara piksel dengan piksel lainnya. Kontras memiliki batas nilai dari 0 hingga pangkat 2 dari Panjang matriks GLCM simetris. Untuk mencari ekstraksi fitur kontras dapat digunakan rumus sebagai berikut :

$$\sum_{x,y} (x - y)^2 p(x,y)$$

Dimana :

$x$  dan  $y$  adalah keabuan dari resolusi 2 piksel yang berdekatan.

$p(x, y)$  adalah *co-occurrence* matrix simetris ternormalisasi.

### 2. Correlation

Korelasi dalam GLCM mengukur ketergantungan linear dari aras keabuan dalam ketetangaan piksel citra. Untuk mencari ekstraksi fitur korelasi dapat digunakan rumus sebagai berikut :

$$\sum_{x,y} \frac{(x - \mu_x)(y - \mu_y)p(x,y)}{\sigma_x \sigma_y}$$

### 3. Homogeneity

Homogenitas digunakan untuk mengukur kehomogenan variasi intensitas citra. Nilai homogenitas (H) dapat dihitung dengan rumus sebagai berikut :

$$\sum_{x,y} \frac{p(x,y)}{1 + |x - y|}$$

Dimana :

x dan y adalah sifat keabuan dari resolusi 2 piksel yang berdekatan.

p(x,y) adalah propabilitas kolom (x dan y)

### 4. Energy

Energi menyatakan ukuran konsentrasi pasangan dengan intensitas keabuan tertentu pada matriks. Nilai energi (E) dapat dihitung dengan persamaan berikut :

$$\sum_{x,y} p(x,y)^2$$

Dimana :

x dan y adalah keabuan dari resolusi 2 piksel yang berdekatan.

p(x,y) adalah co-occurrence matrix simetris ternormalisasi.

## 2.9 Euclidean Distance

Euclidean merupakan suatu metode pencarian kedekatan nilai jarak dari 2 buah variable serta mempresentasikan tingkat kemiripan antara dua buah citra. Semakin kecil jarak antara dua buah citra maka akan semakin mirip kedua citra tersebut [22]. Jarak pada euclidean dihitung berdasarkan ciri khusus yang dimiliki oleh suatu citra, misalnya ciri tekstur. Pada tahap ini sistem mengelompokkan hasil ekstraksi ciri sehingga diperoleh suatu identifikasi dari citra tersebut menggunakan metode euclidean dengan merepresentasikan citra ke dalam koordinat 2 dimensi (x, y) menggunakan rumus berikut:

$$d(x - y) = \sqrt{\sum_{j=1}^i (x_j - y_j)^2}$$

- dimana :
- $d$  : jarak data uji ke data pembelajaran
- $x_j$  : data uji ke-j, dengan  $j= 1,2,\dots,n$
- $y_j$  : data pembelajaran ke-j, dengan  $j= 1,2,\dots,n$

### 2.11 Precision dan Recall

*Precision* adalah nilai yang didapat dari jumlah citra relevan yang muncul pada hasil pencarian dibagi dengan jumlah citra yang muncul dari hasil pencarian. Citra relevan disini berarti citra hasil pencarian yang memiliki kelas yang sama dengan citra *query*. *Recall* adalah nilai yang didapat dari jumlah citra relevan yang muncul pada hasil pencarian dibagi dengan jumlah citra relevan keseluruhan yang tersimpan pada basis data [23].

$$Precision = \frac{\text{number of similar image retrieved}}{\text{total number of images retrieved}}$$

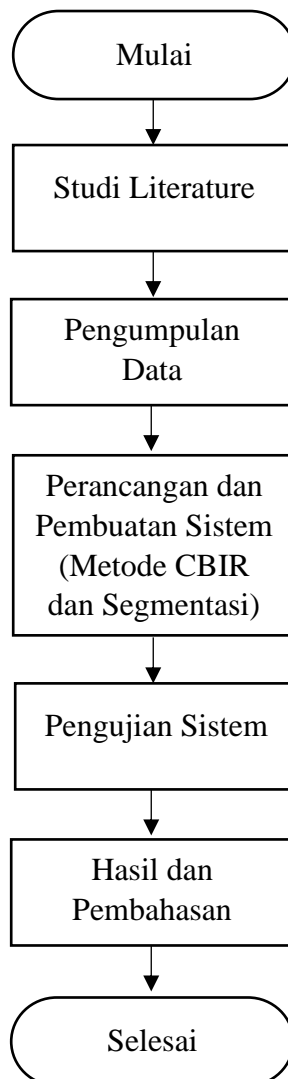
$$Recall = \frac{\text{number of similar image retrieved}}{\text{total number of images retrieved in the database}}$$

### 2.12 MATLAB (*Matrix Laboratory*)

MATLAB (*Matrix Laboratory*) adalah *software/prangkat lunak* yang dikembangkan oleh Mathworks, Inc dengan memanfaatkan matriks dalam penggunaannya. Matrix yang digunakan pada Matlab terbilang sederhana sehingga dapat dengan mudah digunakan. Penelitian ini menggunakan MATLAB sebagai bahasa atau program untuk mengimplementasikan aplikasi sistem yang ingin dibuat. Fitur-fitur dalam MATLAB disebut juga *toolboxes* yang merupakan koleksi atau kumpulan fungsi-fungsi MATLAB yang komprehensif. Beberapa *toolboxes* yang tersedia meliputi penanganan pengolahan sinyal, logika kabur, jaringan saraf tiruan dan pengolahan citra digital serta beragam fungsi lainnya.

### BAB III METODOLOGI PENELITIAN

Guna mendukung berjalannya penelitian, proses penelitian ini dilakukan untuk mengetahui tahapan-tahapan atau alir pembuatan yang akan dilakukan pada penelitian ini. Diagram alir pembuatan dari mulai pengumpulan data hingga pembuatan naskahnya tertuang dalam Gambar 3.1.



Gambar 3.1 Alur Penelitian

Beberapa tahapan yang dilakukan dalam penelitian ini, pertama melakukan pengumpulan data citra biji kopi selanjutnya melakukan studi literatur untuk menambah wawasan penulis mengenai apa yang akan dilakukan. Studi literatur ini

dilakukan melalui referensi-referensi yang berasal dari jurnal, buku, dan penelitian-penelitian yang pernah dilakukan sebelumnya untuk mempelajari cara membangun sesuai dengan metode yang digunakan. Selanjutnya adalah tahap pembangunan metode cbir dan segmentasi sesuai dengan rancangan yang telah dibuat. Tahap pengujian dilakukan untuk menguji apakah sesuai dengan tujuan penelitian yaitu untuk mengetahui seberapa akurat tingkat akurasi dari metode yang digunakan. Setelah berhasil dibangun maka tahap terakhir yakni pembuatan naskah.

### 3.1 Pengumpulan Data

Data berupa file gambar biji kopi robusta diperoleh dari hasil penelitian lapangan yang dilakukan didesa Pekon Balak kec. Balik Bukit Kab. Lampung Barat. Dengan pengambilan data citra melalui kamera *handphone* Oppo A7. Total data yang didapat sebanyak 500 data citra, sebanyak 250 citra biji bagus dan 250 citra sisanya sebagai biji buruk.



Gambar (a) Biji Bagus



Gambar (b) Biji Buruk

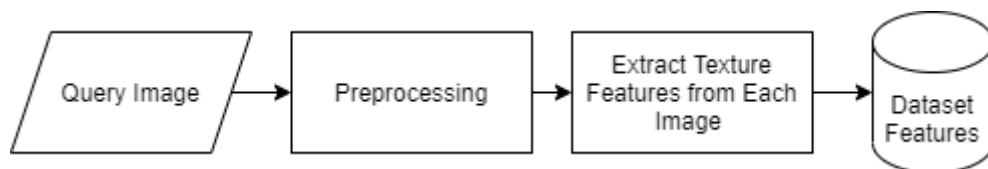
Gambar 3.2 Dataset Biji Kopi, Gambar(a).Bagus, Gambar(b).Buruk

Nilai cacat bisa berupa pecah pada biji, lubang pada biji, dan sebagainya. Dari hasil ekstraksi fitur tekstur untuk biji buruk nilai minimum fitur *Contrast* sebesar 0.1076 sedangkan maksimum sebesar 0.7833, untuk biji bagus nilai minimum sebesar 0.1477 sedangkan nilai maksimum sebesar 1.1335. Untuk fitur *Correlation* nilai minimum biji buruk sebesar 0.7132 sedangkan

maksimum sebesar 0.9547, untuk nilai minimum biji bagus sebesar 0.8592 sedangkan maksimum sebesar 0.9666. Pada fitur *Homogeneity* nilai minimum untuk biji buruk sebesar 0.8766, sedangkan nilai maksimumnya sebesar 0.9944. Biji bagus nilai minimumnya sebesar 0.8799 sedangkan maksimumnya 0.9620. Fitur *Energy* pada biji buruk nilai minimumnya 0.2034 sedangkan nilai maksimumnya 0.9412. untuk biji bagus nilai minimumnya 0.2034 sedangkan maksimumnya 0.5704.

### 3.2 Rancangan Dataset

Dataset merupakan representasi kumpulan citra yang saling berhubungan disimpan secara bersamaan digunakan untuk keperluan penyediaan informasi lebih lanjut.



Gambar 3.3 Alur Dataset

Dimana :

- Query Image : data citra yang terkumpul untuk dijadikan dataset.
- Preprocessing : tahapan merubah ukuran citra semakin kecil dan merubah citra RGB menjadi Grayscale.
- Extract Texture Feature : tahapan untuk mengambil ciri fitur tekstur pada masing masing citra tersebut.
- Dataset Features : berupa hasil ekstraksi fitur tekstur dari masing-masing citra tersimpan sebagai dataset.

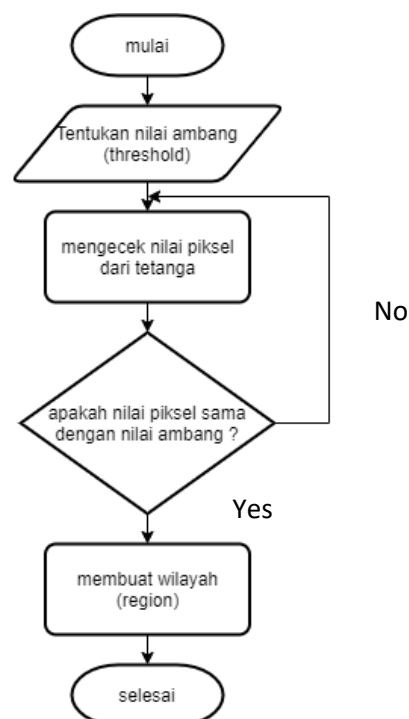
### 3.3 Rancangan Sistem Keseluruhan

Sistem yang akan dibuat bertujuan untuk mencari kemiripan dari biji kopi melalui proses CBIR (*Content Based Image Retrieval*) dan Segmentasi citra. Langkah awalnya citra hasil cropping secara manual akan disegmentasi terlebih dahulu agar terpisah dengan background. Pada tahap *pre-processing* citra akan melalui tahap *resize* dan juga *grayscale*, selanjutnya sistem akan melakukan proses ekstraksi ciri untuk mencari karakteristik citra masukan yang ingin

dibandingkan dengan data yang tersimpan dalam database pelatihan.

Ekstraksi ciri yang digunakan adalah ekstraksi ciri tekstur dengan GLCM (*Gray Level Co-occurrence Matrix*). Setelah dilakukan ekstraksi ciri maka akan didapat vektor-vektor ciri pada *database* yang memiliki nilai kedekatan tertentu untuk diambil kembali (*Retrieved*). Dari situ sistem sudah dapat membedakan antara mana yang biji kopi dalam kategori bagus dan mana biji kopi dengan kategori buruk.

### 3.4 Region Growing Segmentation



Gambar 3.4 *Flowchart Region Growing*

Dimana :

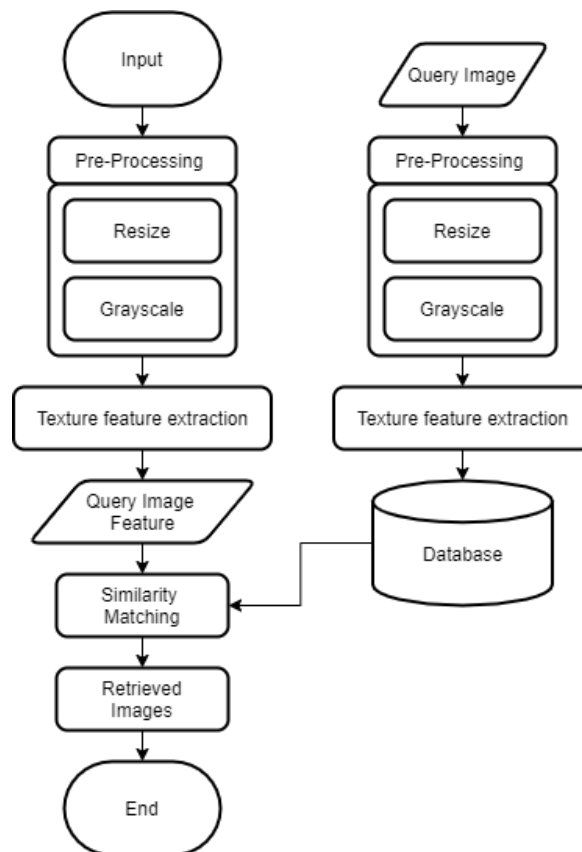
- Tahap pertama menentukan nilai ambang (threshold) yaitu menentukan seed point dari citra sebagai tahap awal pembentukan reion atau ketetangaan.
- Memeriksa setiap wilayah piksel tetangga adakah yang mirip dengan piksel benih.



- c. Jika ada maka akan langsung membuat wilayah (region), jika tidak proses akan kembali untuk mengecek kembali keseluruhan wilayah apakah ada yang memenuhi kriteria dengan wilayah piksel benih.
- d. Hentikan pertumbuhan wilayah jika tidak ada lagi piksel yang memenuhi kriteria penyertaan di wilayah itu, bentuk wilayah region.
- e. Proses selesai.

### 3.5 Perancangan CBIR

Berfungsi untuk menggambarkan secara garis besar dari alur kerja sistem secara keseluruhan. Diagram struktur dapat dilihat pada gambar 3.4.



Gambar 3.5 *Flowchart CBIR*

Dimana :

- Query Image : berisi sekumpulan citra untuk data latih
- Pre-processing : ada 2 rangkaian tahap yang akan dilalui citra latih, yaitu tahap “*resize*” berfungsi untuk merubah ukuran citra agar sama rata, dan tahap “*greyscale*” berfungsi untuk merubah citra RGB menjadi abu-abu.

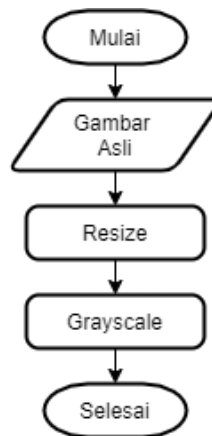
- Texture feature extraction: tahap ini dilakukan pengambilan fitur ciri berdasarkan tekstur biji.
- Database : penyimpanan data hasil ekstraksi fitur tekstur

Pengujian :

- Input : pada tahap ini citra uji yang dipilih akan diinputkan ke dalam program untuk mencari gambar yang mirip.
- Pre-processing : ada 2 rangkaian tahap yang akan dilalui citra latih, yaitu tahap “*resize*” berfungsi untuk merubah ukuran citra agar sama rata, dan tahap “*greyscale*” berfungsi untuk merubah citra RGB menjadi abu-abu.
- Texture feature extraction: tahap ini dilakukan pengambilan fitur ciri berdasarkan tekstur biji.
- Query Image Feature: pada tahap ini hasil dari ekstraksi fitur tekstur tidak disimpan kedalam database melainkan langsung diproses ketahap selanjutnya.
- Similarity Matching: tahap ini mencari citra-citra yang mirip dengan citra inputan dengan cara mencari nilai fitur-fitur tekstur yang ada didalam database.
- Retrieved Images :setelah hasil pencocokan, maka citra – citra yang cocok akan ditampilkan.
- Selesai.

### 3.6 Tahap *Preprocessing*

Tahap *preprocessing* dilakukan untuk mengolah data citra sehingga meningkatkan kualitas citra. Proses pertama yaitu *cropping* yang berguna untuk memilih dan memotong citra biji kopi hasil akuisisi data. Namun pada penelitian ini proses *cropping* dilakukan secara manual. *Resize* dilakukan untuk membuat ukuran citra menjadi sama rata dan akan mempercepat proses perhitungan. Tahap terakhir yaitu *grayscale*, merupakan proses untuk mengubah warna menjadi keabu-abuan. Dengan mengubah nilai RGB setiap piksel gambar menjadi satu nilai yang sama sehingga setiap piksel memiliki nilai yang sama untuk ketiga unsur warna serta didapatkan nilai matriks *grayscale*.



Gambar 3.6 *Preprocessing*

### 3.7 Fitur Tekstur

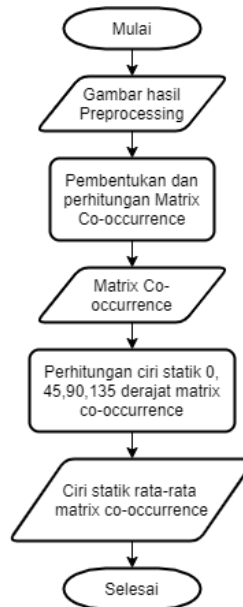
Fitur Tekstur yaitu keteraturan pola tertentu yang terbentuk dari susunan piksel pada citra. Suatu citra dikatakan mempunyai informasi tekstur yaitu apabila memiliki pola atau karakteristik di suatu daerah pada citra yang muncul secara berulang dengan interval jarak dan arah tertentu [24]. Ada beberapa metode dalam ekstraksi fitur tekstur, diantaranya :

- a. Ciri orde pertama didasarkan pada karakteristik histogram citra. Ciri orde pertama umumnya digunakan untuk membedakan tekstur makrostruktur (perulangan pola local secara periodic). Ciri orde pertama antara lain: mean, variance, skewness, kurtosis, dan entropy.
- b. Ciri orde dua didasarkan pada probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Ciri orde dua umumnya digunakan untuk membedakan tekstur mikrostruktur (pola local dan perulangan tidak begitu jelas). Ciri orde dua antara lain: Angular Second Moment, Contrast, Correlation, Variance, Inverse Different Moment, dan Entropy.
- c. Tekstur descriptor lain : laws tektur measure, wavelet, streerable pyramids, local binary patterns (LBP).

### 3.8 Gray Level Co-Occurrence Matrix (GLCM)

Gray Level Co-occurrence Matrix (GLCM) adalah matriks yang menggambarkan jumlah pasangan piksel terhadap frekuensi pada jarak  $d$  dan variasi sudut inklinasi  $\theta$  dengan tujuan menghitung nilai fitur Gray Level Co-

occurrence Matrix (GLCM) [25]. Terdapat 4 nilai statistik yang umum digunakan dalam glcm, yaitu *Contrast*, *Correlation*, *Energy*, *Homogeneity* dengan sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ .



Gambar 3.7 Flowchart GLCM

Dimana :

1. Setelah citra melalui tahap processing, maka selanjutnya ialah ekstraksi ciri tekstur glcm.
2. Tahap pertama citra yang sudah berubah menjadi grayscale akan melalui tahap normalisasi matriks. Adapun tahap dari normalisasi ini diantaranya : Tahap quantitation matriks, Tahap caunt matriks, Tahap tranpose, Tahap caunt + tranpose, dan yang terakhir tahap Normalisasi.
3. Sehingga hasil dari normalisasi matriks tersebut akan digunakan untuk menghitung ciri contrast, correlation, homogeneity, dan energy.
4. Langkah terakhir yaitu merata-rata setiap nilai dan sudut yang didapat sehingga hanya menjadi satu nilai.

### 3.9 Spesifikasi Kebutuhan

Setelah dilakukan perancangan, kemudian dilakukan implementasi . Berikut merupakan spesifikasi dari perangkat keras dan perangkat lunak yang digunakan untuk pembangunan :

1. Spesifikasi perangkat keras yang digunakan :
  - b. Laptop dengan spesifikasi Prosesor: Intel® Core™ i3-2330M CPU @ 2.20GHz, Memory (RAM): 2048MB, Operating Sistem: 64-bit, SSD: 128GB
  - c. Smartphone OPPO A7 dengan spesifikasi Processor: Qualcomm SDM450 Eight core, Memory (RAM): 4GB, Internal Storage: 64GB, Kamera 13/2 MP.
2. Spesifikasi perangkat lunak yang digunakan:
  - a. Matlab R2014a
  - b. Microsoft Office 2016
  - c. Draw.io

## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini akan menjelaskan alur kerja sistem secara keseluruhan, beberapa fungsi atau *method* program yang digunakan dalam penelitian ini, dan pengimplementasian metode kedalam perangkat lunak simulasi yang sudah dijelaskan pada tahap perancangan di bab sebelumnya. Tahap yang dilakukan untuk menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dimengerti oleh komputer serta penerapan perangkat lunak pada keadaan sebenarnya.

#### 4.1 Pengujian Sistem

Tujuan dari pengujian ini adalah untuk mengetahui seberapa besar keberhasilan dan kegagalan dari sistem ini. Serta yang terpenting adalah untuk mengukur keakuratan sistem dalam menganalisis data untuk nantinya dapat ditarik kesimpulan sesuai pengamatan dari pengujian ini. Adapun tahapan dalam pengujian sistem ini sebagai berikut :

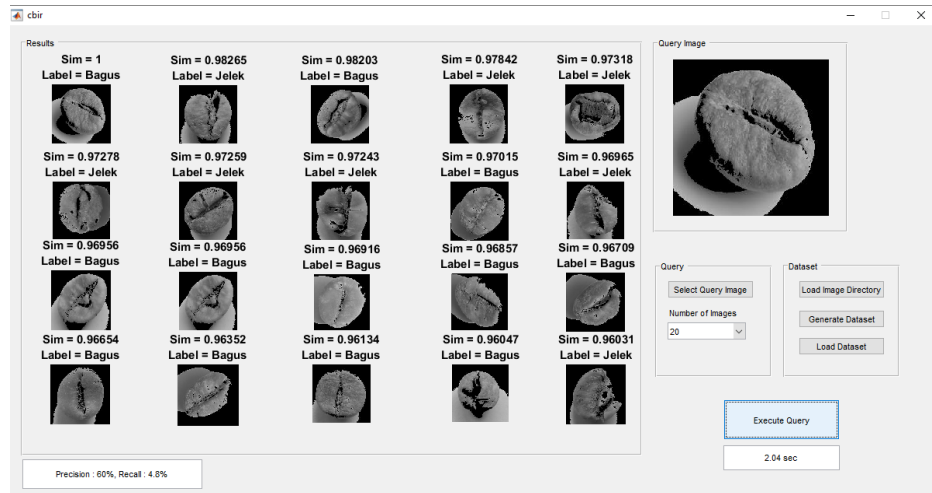
i. Tahap Pelatihan Data

- Pembuatan Dataset

Melakukan proses pelatihan terhadap seluruh citra, dimana pada tahap ini dilakukan pengambilan fitur tekstur menggunakan GLCM. Setelah berhasil mengambil fitur pada masing-masing citra maka disimpan sebagai dataset pelatihan.

ii. Tahap Pengujian Data

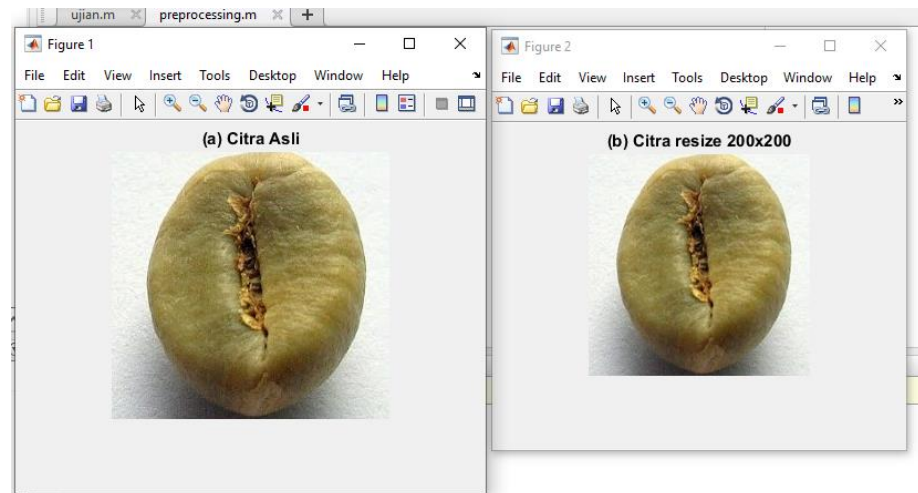
Pada proses mencari kemiripan, citra-citra tersebut akan melalui tahap *resize* dan *greyscale* dengan menggunakan fungsi dari matlab :  $Img\_rsz = imresize(Img,[200,200]);$  dan  $Img\_gray = rgb2gray(Img\_rsz);$ . Citra query akan di ambil fitur tekstur terlebih dahulu, setelah didapatkan fitur tekstur kemudian system melakukan kemiripan citra berdasar fitur tekstur dan Euclidean distance.



Gambar 4.1 Pencarian Citra 20 Query

#### 4.2 Resizing

Untuk proses yang lebih cepat dalam mencari parameter pada data citra, data citra awal akan diubah ukurannya dengan proses reizing dari ukuran asli menjadi ukuran 200 x 200 piksel. Menggunakan fungsi pada matlab yaitu :

$$\text{Img\_rsz} = \text{imresize}(\text{Img}, [200, 200]);$$
Gambar 4.2 Citra *Resizing*, (a) Citra Asli, (b) hasil resize

#### 4.3 Pengambilan Nilai Matrix RGB

Setelah data citra dirisize, kemudian dilakukan pengambilan nilai matrix untuk mempermudah kita dalam melakukan hitungan manual. Dengan mengambil satu contoh citra, Fungsi matlab yang dipakai yaitu :



Gambar 4.3 Citra Sampel

```
red=Img_rsz(:,:,1);
green=Img_rsz(:,:,2);
blue=Img_rsz(:,:,3);
```

berdasarkan citra sampel yang digunakan, nilai matrix R, G, B sudah berhasil diekstrak kedalam bentuk matrix, hasilnya sebagai berikut :

**Tabel 4.1 Nilai R (Red)**

	1	2	3	4	5	6	7	8	9	10	...	200
1	243	239	238	242	245	245	243	242	241	241	...	243
2	239	238	240	243	244	241	238	237	238	238	...	240
3	247	245	242	241	240	237	236	238	241	242	...	242
4	246	244	242	240	239	238	236	237	242	245	...	246
5	237	239	241	240	239	238	237	238	239	240	...	247
6	235	239	241	240	238	239	240	240	239	238	...	245
7	237	240	239	239	242	244	243	241	239	236	...	241
8	234	233	232	234	239	241	240	238	238	238	...	244
9	234	231	233	236	235	238	240	240	242	244	...	246
10	243	240	241	242	239	238	243	245	246	246	...	243
...	...	...	...	...	...	...	...	...	...	...	...	...
200	213	211	204	206	206	207	214	207	208	213	...	231

Pada tabel 4.1 nilai matrix R (Red) dengan ukuran 200x200 piksel. Pada piksel (1,1) nilai  $R = 243$  dan pada piksel (200x200) nilai  $R = 231$ .

**Tabel 4.2 Nilai G (Green)**

	1	2	3	4	5	6	7	8	9	10	...	200
1	247	243	242	246	249	250	248	247	246	246	...	247
2	243	242	244	247	248	246	243	242	243	243	...	244
3	251	249	246	245	244	242	241	243	246	247	...	246
4	250	248	246	244	243	243	241	242	247	250	...	250



5	241	243	245	244	243	243	242	243	244	245	...	251
6	239	243	245	244	243	244	245	245	244	243	...	249
7	241	244	243	243	246	249	248	246	244	241	...	245
8	238	237	236	238	243	246	245	243	243	243	...	248
9	238	235	237	240	239	243	245	245	247	249	...	251
10	247	244	245	246	243	244	248	250	251	251	...	248
...	...	...	...	...	...	...	...	...	...	...	...	...
200	212	211	204	210	210	213	219	213	212	216	...	235

Pada tabel 4.2 nilai matrix G (Green) dengan ukuran 200x200 piksel. Pada piksel (1,1) nilai  $R = 247$  dan pada piksel (200x200) nilai  $R = 235$ .

**Tabel 4.3 Nilai B (Blue)**

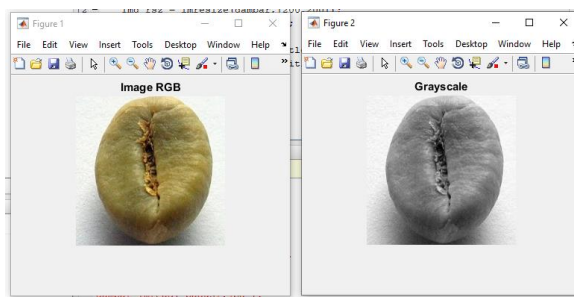
	1	2	3	4	5	6	7	8	9	10	...	200
1	246	242	241	245	248	246	244	243	242	242	...	246
2	242	241	243	246	246	242	239	238	239	239	...	243
3	250	248	245	244	242	238	237	239	242	243	...	245
4	249	247	245	243	242	239	237	238	243	246	...	249
5	240	242	244	243	241	238	238	239	240	241	...	250
6	238	242	244	243	241	240	241	241	240	239	...	248
7	240	243	242	242	244	245	244	242	240	237	...	244
8	237	236	235	237	241	242	241	239	239	239	...	247
9	237	234	236	239	238	239	241	241	243	245	...	248
10	246	243	244	245	241	239	244	246	247	247	...	244
...	...	...	...	...	...	...	...	...	...	...	...	...
200	215	213	206	210	209	211	218	211	211	216	...	233

Pada tabel 4.3 nilai matrix B (Blue) dengan ukuran 200x200 piksel. Pada piksel (1,1) nilai  $R = 246$  dan pada piksel (200x200) nilai  $R = 233$ .

#### 4.4 Grayscale

Setelah melalui tahap resize, maka tahap selanjutnya adalah mengubah citra mode RGB menjadi Grayscale.

**Img\_gray = rgb2gray(Img\_rsz);**



Gambar 4.4 *Grayscale*, (a) gambar hasil resize, (b) citra hasil grayscale

Fungsi dari mengubah citra RGB ke grayscale ialah untuk menyederhankan nilai piksel pada sebuah citra. Jika RGB memiliki 3 nilai piksel yaitu: nilai R, nilai G, dan nilai B setelah dirubah menjadi greyscale hanya 1 nilai keabuan saja.

**Tabel 4.4 Nilai RGB**

R = 243	R = 239
G = 247	G = 243
B = 246	B = 242
R = 239	R = 238
G = 243	G = 242
B = 242	B = 241

Pada tabel 4.4 dapat dilihat pada *pixel* (1,1) mempunyai nilai R = 243, G = 247, B = 246, piksel (1,2) mempunyai nilai R = 239, G = 243, B = 242, piksel (2,1) mempunyai nilai R = 239, G = 243, B = 242, dan piksel (2,2) mempunyai nilai R = 238, G = 242, B = 241. Dengan rumus konversi RGB ke *grayscale*, maka nilai dari setiap piksel akan dimasukkan kedalam rumus. Perhitungan menggunakan rumus konversi *grayscale* adalah sebagai berikut [26]:

$$\underline{\underline{Greyscale = 0,299R + 0,587G + 0,114B}}$$

$$Grayscale\ pixel_{(1,1)}$$

$$= (0,299 * 243) + (0,587 * 247) + (0,114 * 246)$$

$$= 72,653 + 144,989 + 28,044$$

$$= 245,686$$

$$= 246$$

$$Grayscale\ pixel_{(1,2)}$$

$$= (0,299 * 239) + (0,587 * 243) + (0,114 * 242)$$

$$= 71,463 + 142,641 + 27,588$$

$$= 241,692$$

$$= 242$$

$$Grayscale\ pixel_{(2,1)}$$

$$= (0,299 * 239) + (0,587 * 243) + (0,114 * 242)$$

$$= 71,461 + 142,641 + 27,588$$

$$= 241,6903$$

$$= 242$$

*Grayscale pixel*<sub>(2,2)</sub>

$$= (0,299 * 238) + (0,587 * 242) + (0,114 * 241)$$

$$= 71,162 + 142,054 + 27,474$$

$$= 240,6903$$

$$= 241$$

Setelah dilakukan terhadap seluruh matriks hasil konversi RGB ke *grayscale* maka, hasilnya dapat dilihat pada tabel 4.5.

**Tabel 4.5 Nilai Konversi RGB ke Grayscale**

	1	2	3	4	5	6	7	8	9	10	...	200
1	246	242	241	245	248	248	246	245	244	244	...	246
2	242	241	243	246	247	244	241	240	241	241	...	243
3	250	248	245	244	243	240	239	241	244	245	...	245
4	249	247	245	243	242	241	239	240	245	248	...	249
5	240	242	244	243	242	241	240	241	242	243	...	250
6	238	242	244	243	241	242	243	243	242	241	...	248
7	240	243	242	242	245	247	246	244	242	239	...	244
8	237	236	235	237	242	244	243	241	241	241	...	247
9	237	234	236	239	238	241	243	243	245	247	...	249
10	246	243	244	245	242	242	246	248	249	249	...	246
...	...	...	...	...	...	...	...	...	...	...	...	...
200	213	211	204	209	209	211	217	211	211	215	...	234

Terlihat pada tabel 4.5, *pixel* (1,1) mempunyai nilai *grayscale* yaitu 246, *pixel* (1,2) mempunyai nilai *grayscale* yaitu 242, *pixel* (2,1) mempunyai nilai *grayscale* yaitu 242 dan *pixel* (2,2) mempunyai nilai *grayscale* yaitu , dan sampai pada *pixel* (200 x 200) mempunyai nilai *grayscale* yaitu 234.

#### 4.5 Segmentasi *Region Growing*

1. yang pertama dilakukan adalah menentukan *seed piksel point*. Nilai *seed point* yang dipilih misal 100. Kemudian menentukan *region membership criterion*, yang dipilih dalam penelitian ini adalah 100-199. Sebagai tabel contoh digunakan matriks 10x10 hasil citra yang sudah di *grayscale*.

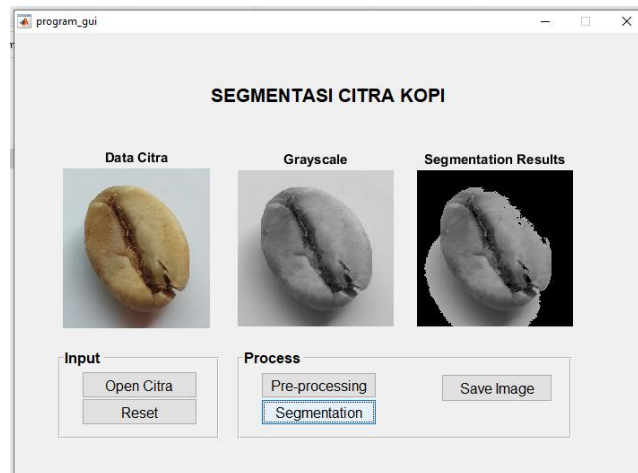
**Tabel 4.6 Matrix Citra**

	1	...	45	46	47	48	49	50	51	52	53	54	...	200
1	246	...	245	242	240	240	240	238	238	240	241	244	...	246
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45	240	...	245	244	246	247	248	249	248	223	187	185	...	247
46	240	...	249	250	250	246	242	247	237	192	182	180	...	246
47	245	...	243	243	249	246	244	248	222	181	176	166	...	245
48	250	...	244	232	240	245	250	235	199	179	171	167	...	245
49	246	...	247	238	243	249	244	198	179	169	165	168	...	243
50	241	...	248	248	248	250	219	174	174	159	157	151	...	244
51	239	...	248	251	247	241	193	172	173	157	143	145	...	246
52	244	...	243	251	249	226	176	165	163	155	141	144	...	247
53	250	...	246	249	245	203	157	152	157	141	143	134	...	246
54	240	...	247	249	225	179	147	150	150	152	149	132	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	213	...	57	58	55	58	67	66	69	64	64	...	...	234

2. Analisa piksel berdampingan berdasarkan kriteria yang ditetapkan. Jika ada piksel berdampingan yang memenuhi kriteria, maka berikan label 1 pada piksel tersebut. Sebaliknya piksel yang tidak memenuhi kriteria diberi label 0.
3. Simpan piksel yang berlabel 1 pada citra keluaran sebagai sebuah wilayah.
4. Pada citra  $f(x,y)$ , temukan piksel yang berlabel 0 dan lakukan analisa seperti pada poin 2 dan 3.
5. Simpan piksel yang berlabel 0 sebagai sebuah wilayah.

**Tabel 4.7 Matrix Segmentasi**

	1	...	45	46	47	48	49	50	51	52	53	54	...	200
1	0	...	0	0	0	0	0	0	0	0	0	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45	0	...	0	0	0	0	0	0	0	0	1	1	...	0
46	0	...	0	0	0	0	0	0	0	1	1	1	...	0
47	0	...	0	0	0	0	0	0	0	1	1	1	...	0
48	0	...	0	0	0	0	0	0	1	1	1	1	...	0
49	0	...	0	0	0	0	0	1	1	1	1	1	...	0
50	0	...	0	0	0	0	0	1	1	1	1	1	...	0
51	0	...	0	0	0	0	1	1	1	1	1	1	...	0
52	0	...	0	0	0	0	1	1	1	1	1	1	...	0
53	0	...	0	0	0	0	1	1	1	1	1	1	...	0
54	0	...	0	0	0	0	1	1	1	1	1	1	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	0	...	0	0	0	0	0	0	0	0	0	...	...	0



Gambar 4.5 Segmentasi Citra

#### 4.6 Ekstraksi Fitur GLCM

Agar memudahkan dalam perhitungan manual, citra contoh dirubah menjadi piksel yang lebih kecil yaitu 3x3 dengan sudut 0°. Pada tahap ini, Langkah pertama yang dilakkan yaitu menghitung matriks citra dan menormalisasikan hasil matriks. Sebelum menormalisasikan matriks dibutuhkan area kerja matriks terlebih dahulu yang didapat dari fungsi :

```
glcm1 = graycomatrix(Img_gray, 'Offset', [0 1]);
```

3x3 uint8				
	1	2	3	4
1	215	129	201	
2	175	108	167	
3	150	80	169	
4				
5				

Gambar 4.6 Matrix citra sampel 3x3 piksel

##### 1. Tahap Quantitation matrix

Dari hasil matrix yang sudah kita dapatkan, akan dilakukan kuantisasi dengan 8 level keabuan dengan range antara 0-255.

0 – 31	= 0	128 – 159	= 4
32 – 63	= 1	160 – 191	= 5
64 – 95	= 2	192 – 223	= 6
96 – 127	= 3	224 – 255	= 7

Setelah kuantisasi dilakukan, maka diperoleh matrix sebagai berikut :

**Tabel 4.8 Hasil Quantitaion Matrix**

6	4	6
5	3	5
4	2	5

## 2. Tahap Caunt Matrix

Perhitungan caunt matrix dilakukan dengan arah Horizontal.

i	j
---	---

Sehingga diperoleh hasil matrix sebagai berikut :

**Tabel 4.9 Hasil Caunt Matrix**

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0
4	0	1	0	0	0	1	0
5	0	0	1	0	0	0	0
6	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0

## 3. Tahap Tranpose

Pada tahap ini merubah baris pada caunt matrix menjadi kolom pada tranpose. Hasil matrix sebagai berikut :

**Tabel 4.10 Hasil Tranpose**

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0
3	0	0	0	0	1	0	0
4	0	0	0	0	0	1	0
5	0	1	1	0	0	0	0
6	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0

## 4. Tahap Caunt + Tranpose

Pada tahap ini dilakukan penjumlahan antara baris dan kolom pada matrix hasil caunt dan tranpose. Hasil matrinya sebagai berikut :

**Tabel 4.11 Hasil Caunt+Tranpose**

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	1	1	0	0
3	0	0	0	0	2	0	0
4	0	1	0	0	0	2	0
5	0	1	2	0	0	0	0
6	0	0	0	2	0	0	0
7	0	0	0	0	0	0	0

## 5. Tahap Normalisasi

Menjumlahkan seluruh angka yang terdapat pada matrix caunt dan tranpose.

$$1 + 1 + 1 + 1 + 2 + 2 + 2 + 2 = 12.$$

**Tabel 4.12 Tabel Hasil Normalisasi**

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	$\frac{1}{12}$	$\frac{1}{12}$	0	0
3	0	0	0	0	$\frac{2}{12}$	0	0
4	0	$\frac{1}{12}$	0	0	0	$\frac{2}{12}$	0
5	0	$\frac{1}{12}$	$\frac{2}{12}$	0	0	0	0
6	0	0	0	$\frac{2}{12}$	0	0	0
7	0	0	0	0	0	0	0

Setelah matrix 3x3 sudah berhasil dinormalisasi, Langkah selanjutnya menghitung ciri.

- **Energy**

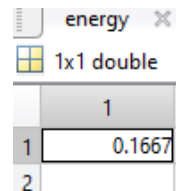
$$\sum_{i,j} p(i,j)^2$$

$$p(i,j)^2 = \left(\frac{1}{12}\right)^2 + \left(\frac{1}{12}\right)^2 + \left(\frac{1}{12}\right)^2 + \left(\frac{1}{12}\right)^2 + \left(\frac{2}{12}\right)^2 + \left(\frac{2}{12}\right)^2 + \left(\frac{2}{12}\right)^2 + \left(\frac{2}{12}\right)^2 +$$

$$= \left(\frac{2 + 2 + 2 + 2 + 4 + 4 + 4 + 4}{144}\right) = \left(\frac{24}{144}\right)$$

$$= 0,1667$$

Hasil matrix energy pada matlab yaitu :



energy	
1x1 double	
	1
1	0.1667
2	

Gambar 4.7 Nilai *energy* pada matlab

- **Contrast**

$$\sum_{i,j} (i-j)^2 p(i,j)$$

$$Con_{(2,4)} = (2-4)^2 \times \frac{1}{12}$$

$$= \frac{4}{12} = 0,3333$$

$$Con_{(2,5)} = (2-5)^2 \times \frac{1}{12}$$

$$= \frac{9}{12} = 0,75$$

$$Con_{(3,5)} = (3-5)^2 \times \frac{2}{12}$$

$$= \frac{8}{12} = 0,6667$$

$$Con_{(4,2)} = (4-2)^2 \times \frac{1}{12}$$

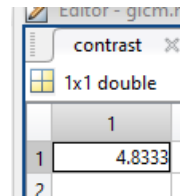
$$= \frac{4}{12} = 0,3333$$

$$Con_{(4,6)} = (4-6)^2 \times \frac{2}{12}$$



$$\begin{aligned}
 &= \frac{8}{12} = 0,6667 \\
 Con_{(5,2)} &= (5 - 2)^2 \times \frac{1}{12} \\
 &= \frac{9}{12} = 0,75 \\
 Con_{(5,3)} &= (5 - 3)^2 \times \frac{2}{12} \\
 &= \frac{8}{12} = 0,6667 \\
 Con_{(6,4)} &= (6 - 4)^2 \times \frac{2}{12} \\
 &= \frac{8}{12} = 0,6667 \\
 Con &= \frac{4 + 9 + 8 + 4 + 8 + 9 + 8 + 8}{12} \\
 &= \frac{58}{12} = \mathbf{4,8333}
 \end{aligned}$$

Hasil ini sesuai dengan hasil kalkulasi pada matlab, yaitu :



The image shows a MATLAB Editor window titled 'Editor - gicm.m'. A variable named 'contrast' is displayed as a '1x1 double' with the value '4.8333'. The variable is shown in a table-like format with a row index '1' and a column index '2'.

1	4.8333
---	--------

**Gambar 4.8 Nilai *contrast* pada matlab**

- **Homogeneity**

$$\begin{aligned}
 &\boxed{\sum_{i,j} \frac{p(i,j)}{1 + |i - j|}} \\
 Hom_{(2,4)} &= \frac{\frac{1}{12}}{1 + |(2 - 4)|} \\
 &= \frac{1}{12 + |(1 + 2)|} = \frac{1}{36} = 0,0277 \\
 Hom_{(2,5)} &= \frac{\frac{1}{12}}{1 + |(2 - 5)|} \\
 &= \frac{1}{12 + |(1 + 3)|} = \frac{1}{48} = 0,0208 \\
 Hom_{(3,5)} &= \frac{\frac{2}{12}}{1 + |(3 - 5)|}
 \end{aligned}$$

$$= \frac{2}{12 + |(1+2)|} = \frac{2}{36} = 0,0556$$

$$\begin{aligned} Hom_{(4,2)} &= \frac{\frac{1}{12}}{1 + |(4-2)|} \\ &= \frac{1}{12 + |(1+2)|} = \frac{1}{36} = 0,0277 \end{aligned}$$

$$\begin{aligned} Hom_{(4,6)} &= \frac{\frac{2}{12}}{1 + |(4-6)|} \\ &= \frac{2}{12 + |(1+2)|} = \frac{2}{36} = 0,0556 \end{aligned}$$

$$\begin{aligned} Hom_{(5,2)} &= \frac{\frac{1}{12}}{1 + |(5-2)|} \\ &= \frac{1}{12 + |(1+3)|} = \frac{1}{48} = 0,0208 \end{aligned}$$

$$\begin{aligned} Hom_{(5,3)} &= \frac{\frac{2}{12}}{1 + |(5-3)|} \\ &= \frac{2}{12 + |(1+2)|} = \frac{2}{36} = 0,0556 \end{aligned}$$

$$\begin{aligned} Hom_{(6,4)} &= \frac{\frac{2}{2}}{1 + |(6-4)|} \\ &= \frac{2}{12 + |(1+2)|} = \frac{2}{36} = 0,0556 \end{aligned}$$

$$\begin{aligned} Hom &= \frac{1 + 1 + 2 + 2 + 2 + 2}{36} + \frac{1 + 1}{48} \\ &= \frac{10}{36} + \frac{2}{48} = \frac{40 + 6}{144} = \mathbf{0,3194} \end{aligned}$$

Hasil kalkulasi secara manual sama dengan hasil kalkulasi pada matlab, sebagai berikut :

The image shows a MATLAB workspace window with two tabs: 'contrast' and 'homogeneity'. Below the tabs, there is a 1x1 double array containing the value 0.3194. The array is displayed in a grid format with columns labeled 1, 2, and 3, and rows labeled 1 and 2. The value 0.3194 is located in the first row and first column.

	1	2	3
1	0.3194		
2			

**Gambar 4.9** Nilai *homogeneity* pada matlab

Maka hasil pengambilan fitur tekstur sebagai berikut :

**Tabel 4.13** Pengambilan Sudut

Fitur	Sudut 0°	Sudut 45°	Sudut 90°	Sudut 135°
Contras	4.8333	5	0.8333	5.7500
Correlation	-0.3843	-0.6000	0.9467	-0.7745
Homogeneity	0.3194	0.3750	0.5833	0.3333
Energy	0.1666	0.2500	0.2222	0.2500

Pada table diatas hasil pengambilan fitur tekstur dengan menggunakan fungsi pada matlab dengan seluruh sudut yang ada pada GLCM. Langkah selanjutnya adalah merata-rata nilai dari semua sudut tersebut dengan menggunakan fungsi pada matlab, supaya didapatkan satu nilai tunggal dari setiap fitur untuk lebih memudahkan dalam tahap pencocokan. Berikut hasilnya :

**Tabel 4.14** Hasil Perataan nilai fitur

	Contras	Correlation	Homogeneity	Energy
contoh1	4.1042	-0.20304	0.40278	0.22222

Dari tabel diatas, data tersebut akan digunakan untuk proses mencari kemiripan citra dengan melakukan perhitungan similarity terhadap seluruh nilai fitur yang ada didalam database.

#### 4.7 Perhitungan Euclidean Distance

Pencocokan citra merupakan salah satu bagian dari pengolahan citra yang dilakukan untuk mencari citra lain yang sejenis atau memiliki kemiripan. Salah

satu parameter yang mempresentasikan tingkat kemiripan antara dua buah citra adalah jarak Euclidean. Semakin kecil jarak Euclidean antara dua buah citra maka akan semakin mirip kedua citra tersebut.

Perhitungan citra menggunakan rumus Euclidean dapat dipresentasikan sebagai berikut :

$$d(x - y) = \sqrt{\sum_{j=1}^i (x_j - y_j)^2}$$

Contoh perhitungan jarak euclidean antara dua citra adalah sebagai berikut :

**Tabel 4.15 Data Sampel**

	<b>Contrast</b>	<b>Correlation</b>	<b>Homogeneity</b>	<b>Energy</b>
Citra 1 ( $x$ )	0.3376	0.9236	0.9282	0.2412
Citra 2 ( $y$ )	0.4000	0.9214	0.9285	0.2713

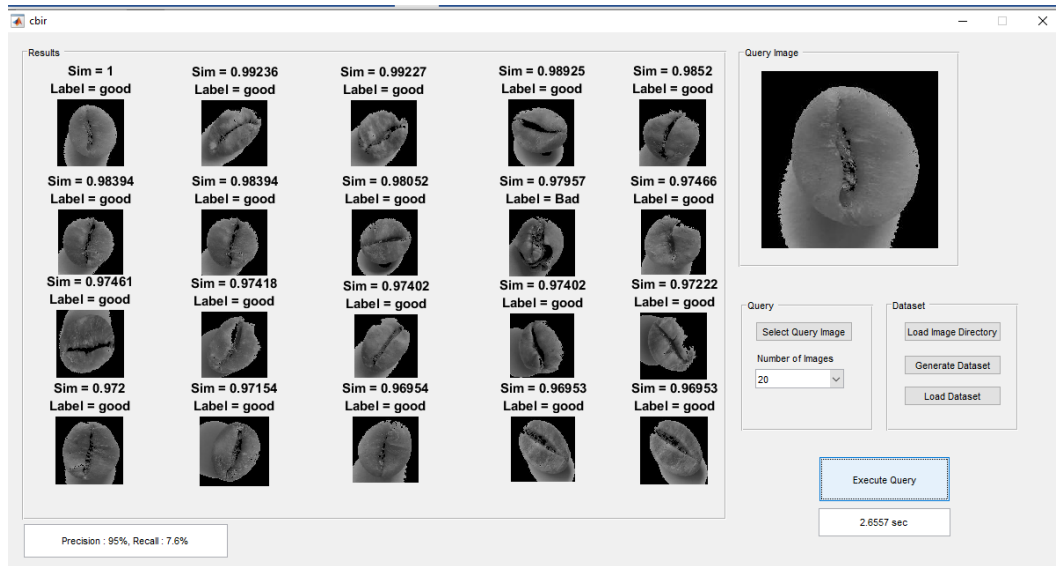
$$\begin{aligned}
 &= \sqrt{(0,3376 - 0,4000)^2 + (0,9236 - 0,9214)^2 + (0,9282 - 0,9285)^2 + (0,2412 - 0,2713)^2} \\
 &= \sqrt{0,00389376 + 0,00000484 + 0,00000009 + 0,00090601} \\
 &= \sqrt{0,0048047} \\
 &= 0,0693
 \end{aligned}$$

Didapat *Euclidean distance* dari sampel citra tersebut adalah 0,0693.

## 4.8 Implementasi Sistem

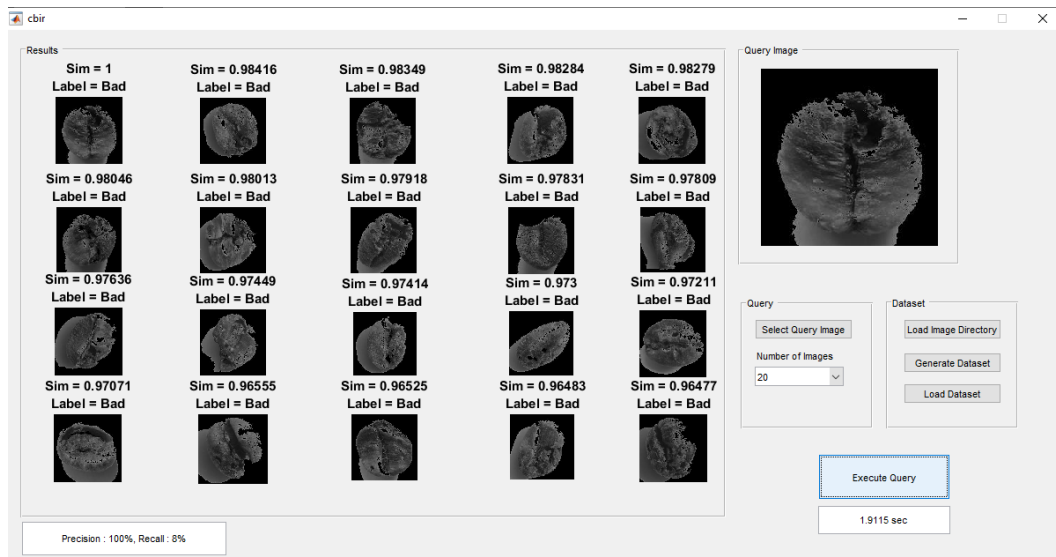
### 4.8.1 Evaluasi CBIR

Untuk mengetahui hasil pengujian dari penelitian ini, maka dilakukan perbandingan dengan penelitian-penelitian sebelumnya yang telah dijadikan bahan referensi.



**Gambar 4.10 Pengujian Biji Bagus**

Citra query pada gambar 4.10 dapat digolongkan sebagai biji bagus karena memiliki nilai *Contrast* 0.2382, nilai *Correlation* 0.9489, nilai *Homogeneity* 0.9569, dan nilai *Energy* 0.2705. Dengan nilai *Precision* sebesar 95% dan nilai *Recall* sebesar 7,6%.



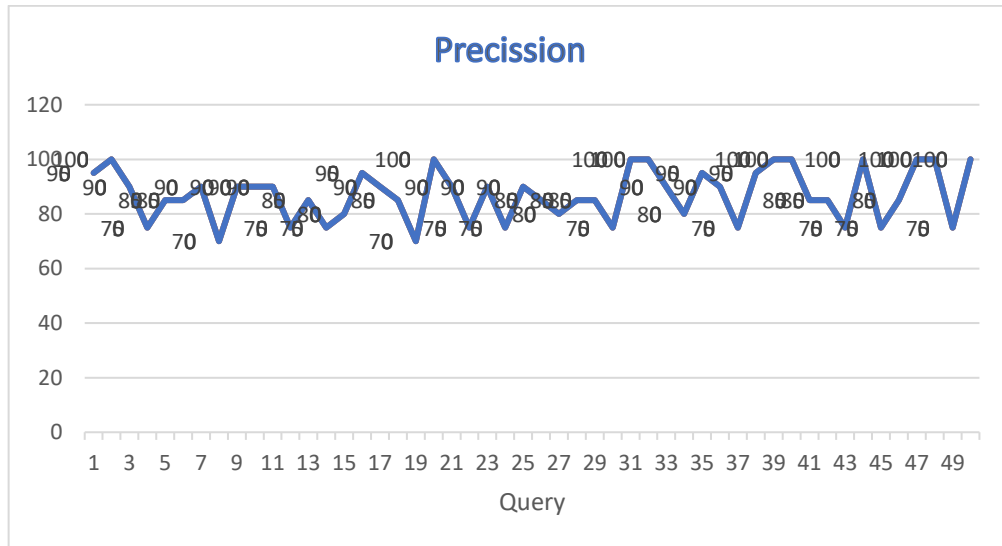
**Gambar 4.11 Pengujian Biji Jelek**

Citra query pada gambar 4.11 dapat digolongkan sebagai biji jelek karena memiliki nilai *Contrast* 0.3239, nilai *Correlation* 0.8421, nilai *Homogeneity* 0.9160, dan nilai *Energy* 0.2856. Dengan nilai *Precision* sebesar 100% dan nilai *Recall* sebesar 8%.

Tabel 4.16 Rekapitulasi Hasil Pengujian

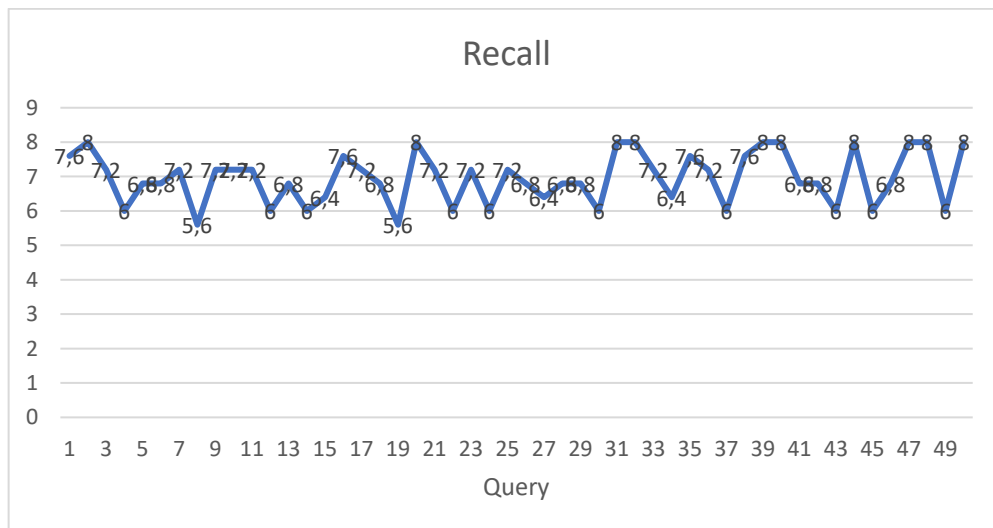
No	Data Uji	Precision	Recall	Time
1	Biji Bagus 1	95	7,6	2.3472 sec
2	Biji Bagus 2	100	8	1.9626 sec
3	Biji Bagus 3	90	7,2	1.9105 sec
4	Biji Bagus 4	75	6	1.8557 sec
5	Biji Bagus 5	85	6,8	2.1994 sec
6	Biji Bagus 6	85	6,8	1.8884 sec
7	Biji Bagus 7	90	7,2	2.2412 sec
8	Biji Bagus 8	70	5,6	2.1393 sec
9	Biji Bagus 9	90	7,2	2.0344 sec
10	Biji Bagus 10	90	7,2	2.6994 sec
11	Biji Bagus 11	90	7,2	1.9198 sec
12	Biji Bagus 12	75	6	1.8955 sec
13	Biji Bagus 13	85	6,8	1.9064 sec
14	Biji Bagus 14	75	6	1.9121 sec
15	Biji Bagus 15	80	6,4	1.8893 sec
16	Biji Bagus 16	95	7,6	1.9663 sec
17	Biji Bagus 17	90	7,2	2.0874 sec
18	Biji Bagus 18	85	6,8	1.8921 sec
19	Biji Bagus 19	70	5,6	1.8991 sec
20	Biji Bagus 20	100	8	2.1364 sec
21	Biji Bagus 21	90	7,2	1.8925 sec
22	Biji Bagus 22	75	6	1.9006 sec
23	Biji Bagus 23	90	7,2	1.8994 sec
24	Biji Bagus 24	75	6	2.1928 sec
25	Biji Bagus 25	90	7,2	1.9521 sec
26	Biji Buruk 26	85	6,8	2.0792 sec
27	Biji Buruk 27	80	6,4	1.909 sec
28	Biji Buruk 28	85	6,8	2.1212 sec
29	Biji Buruk 29	85	6,8	1.9268 sec
30	Biji Buruk 30	75	6	1.9105 sec
31	Biji Buruk 31	100	8	2.2121 sec
32	Biji Buruk 32	100	8	1.893 sec
33	Biji Buruk 33	90	7,2	1.9819 sec
34	Biji Buruk 34	80	6,4	1.902 sec
35	Biji Buruk 35	95	7,6	1.9172 sec
36	Biji Buruk 36	90	7,2	1.8974 sec
37	Biji Buruk 37	75	6	2.1812 sec
38	Biji Buruk 38	95	7,6	1.959 sec
39	Biji Buruk 39	100	8	2.212 sec
40	Biji Buruk 40	100	8	1.9934 sec
41	Biji Buruk 41	85	6,8	2.0591 sec
42	Biji Buruk 42	85	6,8	2.1403 sec
43	Biji Buruk 43	75	6	1.9217 sec
44	Biji Buruk 44	100	8	1.889 sec
45	Biji Buruk 45	75	6	2.1918 sec
46	Biji Buruk 46	85	6,8	1.916 sec
47	Biji Buruk 47	100	8	1.9094 sec
48	Biji Buruk 48	100	8	2.155 sec

49	Biji Buruk 49	75	6	1.9578 sec
50	Biji Buruk 50	100	8	2.1616 sec
<b>Rata - rata</b>		<b>4380/50 = 87</b>	<b>7,096/50 = 7,096</b>	<b>101.0075/50 = 2.0202</b>



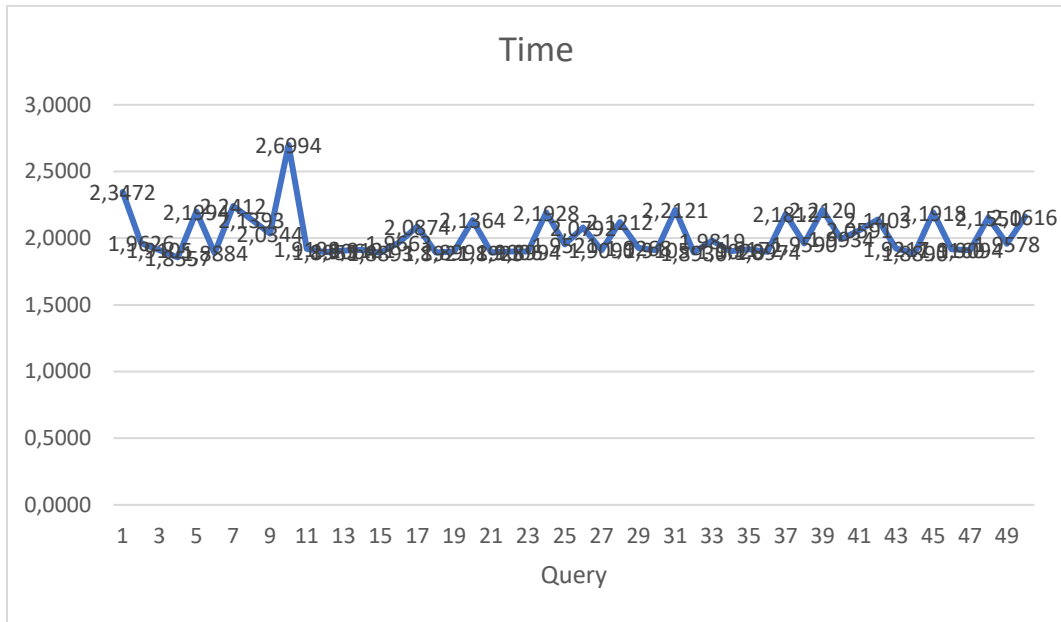
**Grafik 4.1 Persentase Citra Sesuai**

Pada grafik chart 4.1 diatas merupakan grafik dengan citra yang sesuai dengan pengujian CBIR menggunakan perhitungan *Euclidean Distance*.



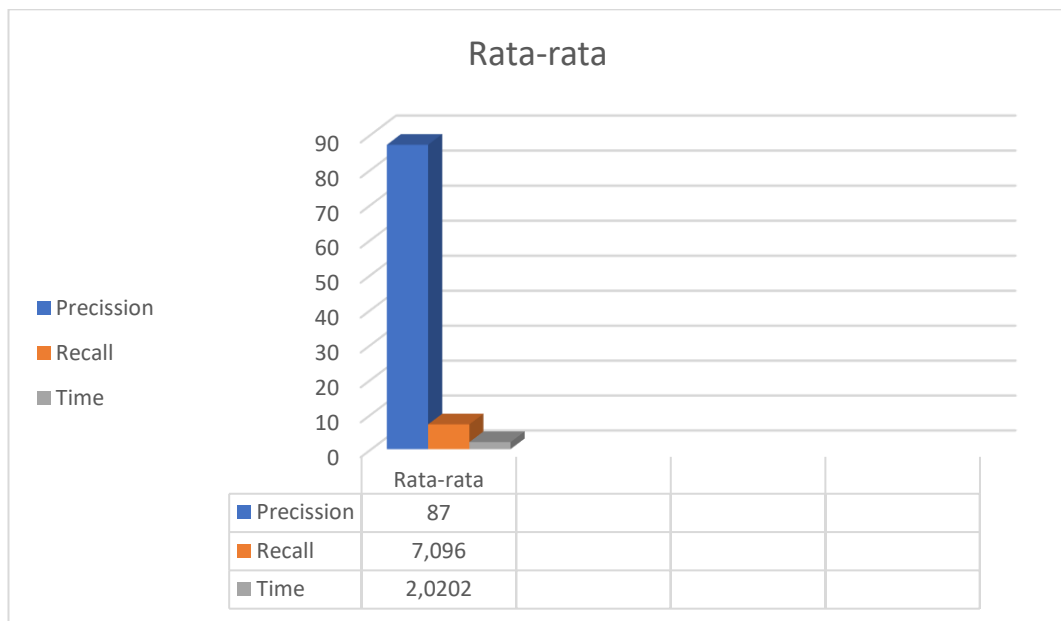
**Grafik 4.2 Hasil Kepresisian Query Citra**

Pada grafik 4.2 diatas merupakan grafik kepresisian citra yang sesuai dengan pengujian CBIR menggunakan perhitungan *Euclidean Distance*.



**Grafik 4.3 Waktu Proses CBIR**

Pada grafik 4.3 diatas merupakan grafik waktu proses citra yang sesuai dengan pengujian CBIR menggunakan perhitungan *Euclidean Distance*.



**Chart 4.4 Rata-rata Precision, Recall, And Time**

Pada grafik 4.4 diatas merupakan grafik rata-rata hasil pengujian sistem dengan *query* sebanyak 50 didapatkan nilai rata-rata *precision* sebesar 87%, nilai *Recall* sebesar 7,096 dan *time* 2,0202 sec.



Tabel 4.18 Perbandingan Dengan Penelitian Sebelumnya

No	Metode	Hasil Pengujian
1	Metode Segmentasi & CBIR	85,4%
	Metode CBIR [3]	55,20%
	Analisa : jumlah data yang digunakan terlalu sedikit, hanya 90 citra sedangkan terdapat 3 kelas didalam penelitian. jadi masing-masing kelas hanya terdapat 30 data. hal ini kemungkinan sangat mempengaruhi hasil pengujian.	
2	Metode Segmentasi & CBIR	85,4%
	AHP ( <i>Analytical Hierarchy Process</i> )[4]	85%
	Analisa : didapat bahwa hasil pengujian hanya berbeda sedikit, kemungkinan perbedaan penggunaan kriteria seperti kadar air, cacat biji dan ketinggian lahan dapat mempengaruhi hasil dari pengujian.	
3	Metode Segmentasi & CBIR	85,4%
	Jaringan Saraf Tiruan (JST) <i>Backpropagation</i> [5]	80%
	Analisa : dengan jumlah data yang hanya 138 citra. kemungkinan banyak sedikitnya data yang digunakan dalam penelitian sangat berpengaruh dengan hasil pengujian.	
4	Metode Segmentasi & CBIR	85,4%
	Metode CBIR[6]	72,2%
	Analisa : dengan 5 kelas yang terdapat pada penelitian tersebut dan jumlah data hanya 250 citra, sehingga jika di estimasikan perkelas hanya mendapat 50 data saja. Dengan data tersebut kemungkinan informasi yang diperoleh sistem masih kurang mendetail dan sangat mempengaruhi hasil pencocokan citra.	
5	Metode Segmentasi & CBIR	85,4%
	Jaringan Saraf Tiruan (JST) dan algoritma <i>Learning Vector Quantization</i> (LVQ)[7]	73,7%
	Analisa : terdapat kesalahan dalam pengenalan jenis kopi pada penelitian LVQ disebabkan oleh adanya benda lain dalam citra gambar yang diambil serta beberapa kesalahan metode dalam pengambilan gambar.	

Dilihat dari table tersebut, hasil pengujian pada penelitian ini mampu sedikit lebih baik dibandingkan dengan penelitian sebelumnya. Hal yang mungkin mempengaruhi hasil penelitian ialah dari jumlah data yang digunakan, kemudian kriteria-kriteria yang digunakan lalu kemudian metode-metode yang digunakan, ataupun mungkin dari banyaknya jumlah kelas yang ditentukan pun dapat mempengaruhi hasil pengujian.

## BAB V

### KESIMPULAN DAN SARAN

Pada bab ini akan dipaparkan beberapa kesimpulan dari hasil penelitian yang diperoleh serta saran yang dapat digunakan sebagai acuan untuk menyempurnakan sistem pada proses penembangan berikutnya.

#### 5.1 Kesimpulan

1. Berdasarkan hasil penelitian dapat disimpulkan bahwa proses segmentasi citra dengan metode *region growing* dapat memisahkan objek dengan latar belakang. Penerapan algoritma *region growing* dilakukan untuk meningkatkan efisiensi pengambilan fitur tekstur pada biji kopi serta memiliki kelemahan hanya dapat mengetahui persebaran wilayah berdasarkan *Seed* dan *Threshold* yang dipilih.
2. Nilai ambang batas pada *region growing* mempengaruhi tingkat akurasi pada hasil segmentasi. Karena nilai ini yang membatasi antara tekstur biji dengan tekstur latar belakang.
3. Dari hasil pengujian, sistem *content based image retrieval* dapat menemukan citra yang dicari berdasarkan perhitungan *euclidean distance* vector citra *query* dengan vector citra yang ada didalam database, dengan rata-rata persentase *precision* tingkat keberhasilan sistem sebesar 87% dan nilai *Recall* sebesar 7,096% dan waktu komputasi rata-rat 2,0202 detik dengan menggunakan 50 data uji terdiri dari 25 citra bagus dan 25 citra buruk.
4. Kelemahan dari penelitian ini, sistem yang dihasilkan belum begitu sepenuhnya akurat, sehingga terdapat beberapa kesalahan dalam mencari kemiripan citra. Hal ini disebabkan karena pada penelitian ini tidak menggunakan *filter* untuk menghilangkan *noise* pada citra. Noise juga mempengaruhi segmentasi, karena *seed* yang terletak pada noise akan menghasilkan kesalahan segmentasi, dan akan sangat berpengaruh terhadap hasil ekstraksi fiturnya.

## 5.1 Saran

Dalam pembuatan penelitian ini, masih terdapat banyak kekurangan yang dapat diperbaiki untuk pengembangan berikutnya. Beberapa saran yang dapat diberikan yaitu :

1. Data citra yang digunakan dapat ditambah lagi jenis biji kopinya, seperti ditambah jenis arabika, jenis gayo, jenis liberika, atau mungkin bisa dibandingkan dengan biji kopi hasil fermentasi dari hewan luwak/musang.
2. Untuk penelitian selanjutnya dapat dikombinasikan atau dibandingkan dengan metode yang lain seperti, menggunakan algoritma SVM, Jaringan Syaraf Tiruan, dan lain-lain untuk dapat membandingkan tingkat keakuratan yang dihasil dari metode – metode tersebut.

## DAFTAR PUSTAKA

- [1] E. D. Martauli, "April 2018 Volume 01 No 02 ANALYSIS OF COFFEE PRODUCTION IN INDONESIA Elvin Desi Martauli," vol. 01, no. 02, pp. 112–120, 2018.
- [2] B. Raharjo and F. Agustini, "Metode Forward Chaining pada Sistem Pakar Penilaian Kualitas Biji Kopi Berbasis Web," vol. 4, pp. 73–82, 2020.
- [3] Y. Prastyaningsih, A. Noor, and A. Supriyanto, "IDENTIFIKASI JENIS BIJI KOPI MENGGUNAKAN EKSTRAKSI FITUR TEKSTUR BERBASIS CONTENT BASED IMAGE RETRIEVAL Identification types of Coffee Beans Using Texture Feature Extraction Based on Content Based Image Retrieval," vol. 3, no. 2, pp. 105–116, 2020.
- [4] W. M. Kurniawan, "PENENTUAN KUALITAS BIJI KOPI ARABIKA DENGAN MENGGUNAKAN ANALYTICAL HIERARCHY PROCESS ( STUDI KASUS PADA PERKEBUNAN KOPI LERENG GUNUNG KELIR JAMBU SEMARANG )," vol. 8, no. 2, pp. 519–528, 2017.
- [5] M. Olivya, E. Tungadi, and N. B. Rante, "Klasifikasi Kualitas Biji Kopi Ekspor Menggunakan Jaringan Saraf Tiruan Backpropagation," *J. Inf. Sains dan Teknol.*, vol. 3, no. 2, pp. 299–308, 2018.
- [6] T. S. Warongan, S. R. U. A. Sompie, A. Jacobus, T. Elektro, F. Teknik, and U. S. Ratulangi, "Penerapan Metode Content-Based Image Retrieval untuk Pengenalan Jenis Bunga," vol. 13, no. 3, 2018.
- [7] J. Ilmiah, T. Pertanian, and J. S. Tiruan, "Mas'ud Effendi 1 , Ullivia Fatasya 1 , Usman Effendi 1," vol. 2, no. 1, pp. 140–146, 2017.
- [8] M. B. Chaniago, A. Purno, and W. Wibowo, "PENENTUNAN KUALITAS TEKSTUR BIJI KOPI JENIS ARABICA MENGGUNAKAN TEKNIK COMPUTER VISION," pp. 37–42, 2017.
- [9] D. M. Wewo, A. Fanggidae, K. Letelay, J. I. Komputer, and U. N. Cendana, "Aplikasi Penentuan Golongan Darah Manusia Dengan Metode," vol. 6, no. 1, pp. 8–14, 2018.
- [10] R. R. Dea, A. Putri, A. W. Widodo, and M. A. Rahman, "Pemanfaatan Metode Texture- Based Region Growing Untuk Segmentasi Buah Jeruk Keprok ( Citrus Reticulata Blanco )," vol. 3, no. 4, pp. 3201–3207, 2019.
- [11] A. Penciri, S. Biji, K. Hijau, and K. Anggraeni, "Arabika Dan Robusta Menggunakan Libs ( Laser-Induced Breakdown Spectroscopy )," 2016.
- [12] A. R. Putri, "Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 1, no. 01, pp. 1–6, 2016, doi: 10.29100/jipi.v1i01.18.
- [13] H. N. Winata and R. N. Fuad, "Konsep Penyandian File Jpeg Dengan Menggunakan Metode Lsb," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 1, no. 2, pp. 127–132, 2017, doi: 10.30743/infotekjar.v1i2.78.
- [14] M. Oktiana, K. Munadi, and F. Arnia, "Metode Keamanan pada Citra JPEG-Ikhtisar," *Semin. Nas. dan Expo Tek. Elektro*, pp. 38–44, 2015.

- [15] J. Pardede, D. B. Utami, and A. C. Rochman, "Implementasi 'Principal Component Analysis - Scale Invariant Feature Transform' Pada Content Based Image Retrieval," *J. Tek. Inform. dan Sist. Inf.*, vol. 3, no. 3, 2017, doi: 10.28932/jutisi.v3i3.690.
- [16] L. Angriani, "Segmentasi Citra dengan Metode Threshold pada Citra Digital Tanaman Penyelenggara : Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Muslim Indonesia Makassar , Indonesia Editor : Tim Editor SNRIK 2015 Penerbit : Fakultas Ilmu Komputer," no. July, 2018.
- [17] A. M. Simarmata, S. Kom, M. Kom, T. P. Sinaga, and O. William, "DETEKSI PENYAKIT LIMFOMA DENGAN MENGGUNAKAN," no. 2, 2019.
- [18] R. E. Marbun and H. D. A. N. Pembahasan, "Segmentasi Citra Spektrum Menggunakan Algoritma Region Growing TIN: Terapan Informatika Nusantara," vol. 1, no. 9, pp. 424–429, 2021.
- [19] L. Ajeng Wijayanti, D. Bambang Hidayat, D. Suhardjo, and M. SpRKG, "PENGOLAHAN CITRA RADIOGRAF PERIAPIKAL PADA DETEKSI PENYAKIT GRANULOMA MENGGUNAKAN METODE DISCRETE WAVELET TRANSFORM & PRINCIPAL COMPONENT ANALYSIS BERBASIS ANDROID Image Processing of Periapical Radiograph on Granuloma Disease Detection Using Discrete Wav," vol. 4, no. 1, pp. 547–553, 2017.
- [20] D. Satria and M. Mushthofa, "Perbandingan Metode Ekstraksi Ciri Histogram dan PCA untuk Mendeteksi Stoma pada Citra Penampang Daun Freycinetia," *J. Ilmu Komput. dan Agri-Informatika*, vol. 2, no. 1, p. 20, 2013, doi: 10.29244/jika.2.1.20-28.
- [21] M. Ramadhani, "Klasifikasi Jenis Jerawat Berdasarkan Tekstur dengan Menggunakan Metode GLCM," *e-Procending of Engineering*, vol. 5, no. 1, pp. 870–876, 2018.
- [22] I. Riadi and A. Fadlil, "Identifikasi Tulisan Tangan Huruf Katakana Jepang Dengan Metode Euclidean," no. March, 2020, doi: 10.30645/j-sakti.v4i1.184.
- [23] M. Resa, A. Yudianto, and H. Al Fatta, "Wayang Dengan Algoritma Convolutional Neural Network," *J. Teknol. Inf.*, no. 2, pp. 182–190, 2020.
- [24] V. Lusiana *et al.*, "Ekstraksi fitur tekstur menggunakan matriks glcm pada citra dengan variasi arah obyek 1,2,3," pp. 978–979, 2019.
- [25] G. T. Situmorang, A. W. Widodo, and M. A. Rahman, "Penerapan Metode Gray Level Cooccurrence Matrix ( GLCM ) untuk Ekstraksi Ciri pada Telapak Tangan," vol. 3, no. 5, pp. 4710–4716, 2019.
- [26] A. Susanto, "Matematika Citra Digital Untuk Ekstraksi Area Plat Nomor," *J. Pseudocode*, vol. VI, no. 1, pp. 49–57, 2019.