

**IMPLEMENTASI METODE *FUZZY K-NEAREST NEIGHBOR*
UNTUK KLASIFIKASI PENYAKIT TANAMAN KOPI
BERDASARKAN CITRA DAUN**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Nurul A'yun Hanifiyah

NIM: 125150201111002

UNIVERSITAS BRAWIJAYA



INFORMATIKA/ ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

IMPLEMENTASI METODE *FUZZY K-NEAREST NEIGHBOR* UNTUK KLASIFIKASI
PENYAKIT TANAMAN KOPI BERDASARKAN CITRA DAUN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Nurul A'yun Hanifiyah

NIM: 125150201111002

Skripsi ini telah diuji dan dinyatakan lulus pada
24 Maret 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si, M.Kom

NIK: 201201 850719 1 001

Drs. Marji, M,T

NIP: 19670801 199203 1 001

Mengetahui

Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M,T

NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 10 Maret 2016



Nurul A'yun Hanifiyah
NIM: 125150201111002

KATA PENGANTAR

Segala puja dan puji syukur atas kehadiran Allah SWT yang senantiasa melimpahkan rahmat dan hidayah-Nya serta telah memberikan kemudahan sehingga penulis mampu menyelesaikan penelitian dengan judul “IMPLEMENTASI METODE *FUZZY K-NEAREST NEIGHBOR* UNTUK KLASIFIKASI PENYAKIT TANAMAN KOPI BERDASARKAN CITRA DAUN”.

Berbagai pihak telah ikut berperan membantu penulis dalam menyelesaikan tugas akhir ini dengan memberikan arahan dan bimbingan serta motivasi. Untuk itu pada kesempatan ini penulis mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Bapak Drs. Marji, M.T, selaku Ketua Program Studi Informatika Universitas Brawijaya Malang sekaligus dosen pembimbing dua.
2. Bapak Imam Cholissodin, S.Si, M.Kom selaku dosen pembimbing satu yang telah memberikan bimbingan, dukungan dan arahan dalam penyelesaian skripsi ini.
3. Bapak Katiman sebagai nara sumber dari penelitian sekaligus ayah tercinta.
4. Ibunda Wagirah yang selalu memberikan do’a dan dukungan yang tiada henti.
5. Teman-teman seperjuangan Prodi Informatika Universitas Brawijaya Malang Angkatan 2012.
6. Seluruh pihak yang telah membantu, baik yang secara langsung atau tidak, sekali lagi saya ucapkan banyak terima kasih.

Demikian laporan ini disusun, dengan harapan dapat memberikan manfaat serta pengetahuan bagi pembaca.

Malang, 3 Maret 2016

Penulis

ayunhanifiyah@gmail.com

ABSTRAK

Pada perkembangan teknologi saat ini, penerapan teknologi informasi dan komunikasi dilakukan terhadap semua bidang, termasuk bidang pertanian dan perkebunan untuk tanaman kopi. Rendahnya tingkat produksi petani kopi hanya difokuskan pada pengolahan akhir kopi sedangkan penyebab utama rendahnya produktivitas adalah pada proses pembudidayaan dan proses tanam termasuk penanggulangan penyakit. Padahal penyakit kopi akan mudah diatasi jika dideteksi lebih cepat dari gejala-gejala yang terlihat termasuk yang terlihat pada bagian daun. Salah satu penerapan teknologi informasi dan komunikasi pada bidang pertanian dan perkebunan ialah melakukan komputasi untuk mengidentifikasi dan klasifikasi penyakit kopi berdasarkan gejala yang ditampakan pada bagian daun. Identifikasi penyakit menggunakan teknik *Data Mining* dengan algoritma *Fuzzy K-Nearest Neighbor*. Citra terlebih dahulu dilakukan proses *Thresholding* untuk memperbaiki kualitas citra untuk proses klasifikasi dengan metode *Otsu*. Dataset yang digunakan berupa citra daun kopi berpenyakit yang meliputi penyakit bercak daun kopi dan penyakit karat daun kopi. Metode yang digunakan adalah *Fuzzy K-Nearest Neighbor* yang melakukan prediksi data uji menggunakan nilai keanggotaan pada data uji tiap kelas, kemudian diambil kelas dengan nilai keanggotaan terbesar dari data uji sebagai hasil kelas prediksi. Pengujian dilakukan terhadap empat variabel pengujian yaitu pengujian terhadap perbandingan data latih dan data uji dengan akurasi tertinggi yaitu 100% menggunakan perbandingan 90:10 dengan komposisi 54 data latih dan 6 data uji. Kemudian pengujian kedua menggunakan variabel k dengan nilai antara rentang 3 sampai dengan 7 dengan akurasi 94,4%. Dan pengujian terhadap nilai m dengan nilai 2 menghasilkan akurasi 90,8%, serta pengujian terhadap nilai peubah *Threshold* dengan nilai 10 menghasilkan akurasi 90,8%.

Kata kunci : kopi, *Thresholding*, *Otsu*, klasifikasi, *Fuzzy K-Nearest Neighbor*.

ABSTRACT

In the current technological developments, the application of information and communication technology is made to all fields, including agriculture and plantations for coffee plants. The low level of production of coffee farmers are only focused on the final processing of coffee while the main causes of low productivity is in the process of cultivation and planting processes including disease prevention. Though coffee disease will be easily overcome if detected much faster than the visible symptoms include visible on the leaves. One application of information technology and communication technology in the field of agriculture and farming is to perform computation to identify the disease based on symptoms coffee that is displayed on the leaves. Identification of disease using Data Mining techniques with Fuzzy K-Nearest Neighbor algorithm. First, the system doing image Thresholding to increase image quality for the classification process with Otsu method. Dataset use the image of the diseased coffee leaves include karat daun kopi and bercak daun kopi. The method used is Fuzzy K-Nearest Neighbor who perform predictive data test using data test the value of membership in each class, then taken classes with membership biggest value of the data test as a result of class prediction. Tests conducted on four variables, namely the testing of comparative testing data training and data testing with the highest accuracy of 100% uses the ratio of 90:10 with compositions 54 data training and 6 data testing grades were on the Threshold of achieving the highest accuracy 94.4% in value Threshold 10 with the amount of data training and data testing 36 and 24 tests on the value of k produce accuracy is 94.4% on k between 1 to 5 with the amount of data training and data testing 36 and 24, and value of m produce the highest accuracy 94.4% on the value of m 1 and 2 the amount of data training and data testing 36 and 24.

Keywords: coffee, Thresholding, Otsu, classification, Fuzzy K-Nearest Neighbor.

DAFTAR ISI

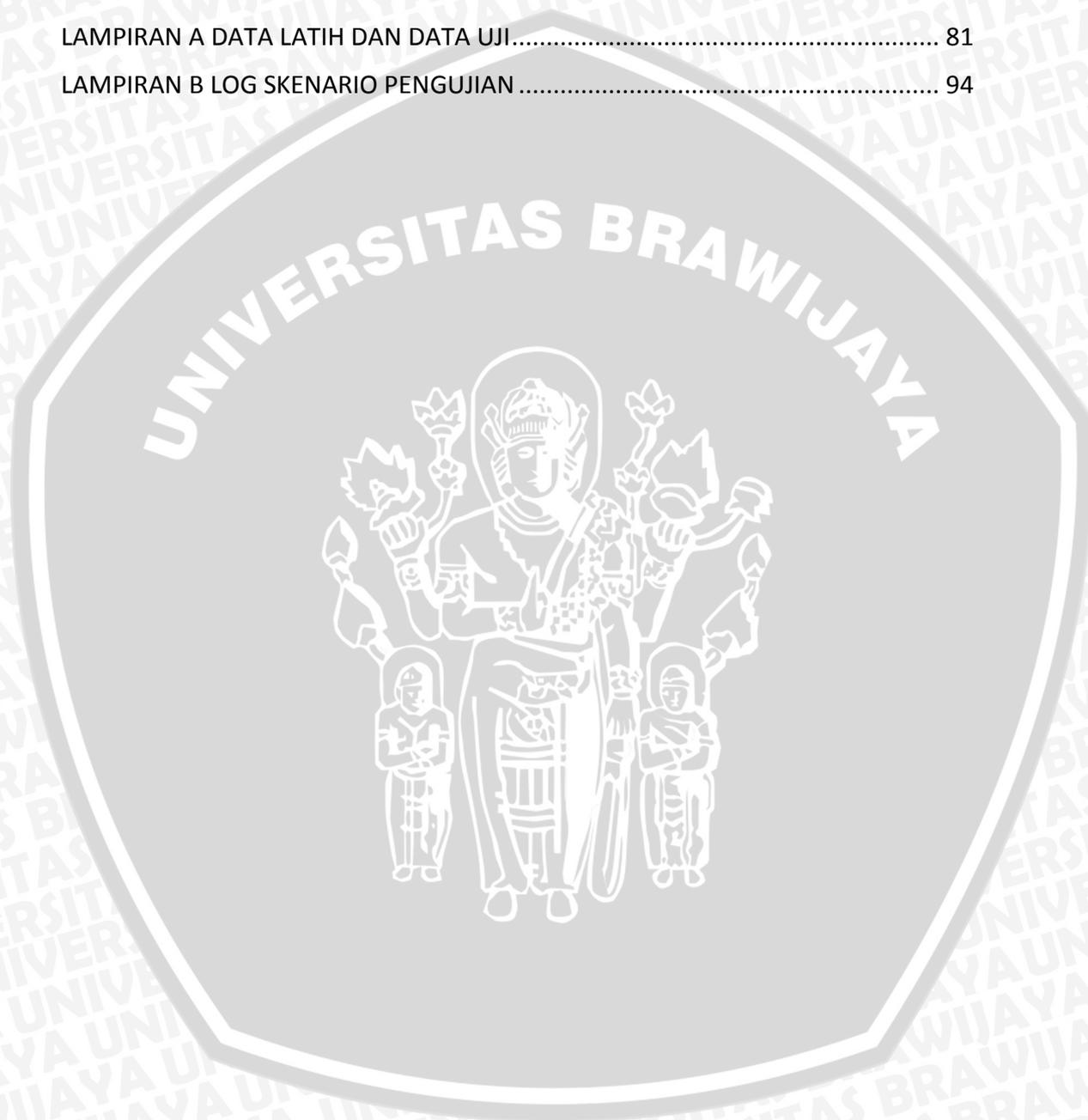
JUDUL	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian pustaka.....	5
2.2 Kopi	7
2.3 Penyakit kopi.....	7
2.3.1 Karat daun kopi.....	7
2.3.2 Bercak daun kopi.....	8
2.3.3 Embun jelaga.....	9
2.4 Definisi citra digital	10
2.4.1 Jenis-jenis citra digital	10
2.4.2 Citra warna (<i>true colour</i>).....	11
2.4.3 <i>Computer vision</i>	11
2.5 Pengolahan citra digital	12
2.5.1 Segmentasi.....	12

2.5.2 Pengembangan (<i>Thresholding</i>)	12
2.5.3 Metode <i>Otsu</i>	12
2.6 Ekstraksi fitur	14
2.7 <i>Data Mining</i>	14
2.7.1 Pengertian <i>Data Mining</i>	14
2.7.2 Klasifikasi	15
2.8 Sistem <i>Fuzzy</i>	15
2.8.1 Pengertian logika <i>Fuzzy</i>	15
2.8.2 Himpunan <i>Fuzzy</i>	15
2.8.3 Himpunan <i>crisp</i>	15
2.9 <i>KNN</i>	16
2.9.1 Pengertian <i>KNN (K-Nearest Neighbor)</i>	16
2.9.2 Algoritma <i>KNN (K-Nearest Neighbor)</i>	16
2.9.3 <i>Euclidean distance</i>	16
2.10 <i>Fuzzy K-Nearest Neighbor</i>	16
2.11 Himpunan keanggotaan	17
2.12 Perhitungan akurasi	17
BAB 3 METODOLOGI PENELITIAN DAN PERANCANGAN	18
3.1 Studi literatur	19
3.2 Pengumpulan data	19
3.3 Analisis kebutuhan	20
3.3.1 Analisis kebutuhan data	20
3.3.2 Analisis kebutuhan sistem	20
3.4 Perancangan sistem	21
3.5 Implementasi sistem	21
3.6. Pengujian	21
BAB 4 PERANCANGAN SISTEM	22
4.1 Diagram alir sistem	22
4.1.1 <i>Preprocessing</i> citra	23
4.1.2 Proses pengujian	30
4.1.3 Proses pelatihan	32
4.2 Perhitungan manual	33

4.2.1 Perhitungan <i>preprocessing</i> menggunakan metode <i>Otsu</i>	34
4.2.2 Perhitungan ekstraksi fitur	47
4.2.3 Perhitungan <i>FK-NN</i>	49
4.2.3 Perhitungan akurasi	55
4.3 Perancangan antarmuka	55
4.3.1 Rancangan antarmuka utama	55
4.3.2 Rancangan antarmuka pelatihan data	56
BAB 5 IMPLEMENTASI	57
5.1 Spesifikasi	57
5.1.1 Spesifikasi perangkat keras	57
5.2 Implementasi kode program	57
5.2.1 Implementasi metode <i>Otsu</i>	58
5.2.2 Implementasi proses pelatihan	62
5.2.3 Implementasi proses pengujian	64
5.3 Implementasi antarmuka	67
5.3.1 Tampilan halaman utama	67
5.3.2 Tampilan halaman pengujian	68
BAB 6 PENGUJIAN DAN ANALISIS	70
6.1 Pengujian terhadap perbandingan data latih dan data uji	70
6.1.1 Skenario pengujian terhadap perbandingan data latih dan data uji	70
6.1.2 Analisis pengujian terhadap perbandingan data latih dan data uji	71
6.2 Pengujian terhadap nilai peubah <i>Threshold (T)</i>	72
6.2.1 Skenario pengujian terhadap nilai peubah <i>Threshold (T)</i>	72
6.2.2 Analisis pengujian terhadap nilai peubah <i>Threshold (T)</i>	73
6.3 Pengujian terhadap nilai <i>k</i>	74
6.3.1 Skenario pengujian terhadap nilai <i>k</i>	74
6.3.2 Analisis pengujian terhadap nilai <i>k</i>	75
6.4 Pengujian terhadap nilai <i>m</i>	75
6.4.1 Skenario pengujian terhadap nilai <i>m</i>	76
6.4.2 Analisis pengujian terhadap nilai <i>m</i>	76



BAB 7 PENUTUP	78
7.1 Kesimpulan.....	78
7.2 Saran	78
DAFTAR PUSTAKA.....	79
LAMPIRAN A DATA LATIH DAN DATA UJI.....	81
LAMPIRAN B LOG SKENARIO PENGUJIAN	94



DAFTAR GAMBAR

Gambar 2.1 Citra daun berpenyakit karat daun	8
Gambar 2.2 Citra daun berpenyakit bercak daun.....	8
Gambar 2.3 Citra daun berpenyakit embun jelaga.....	9
Gambar 3.1 Tahap penelitian.....	18
Gambar 3.2 Alur sistem.....	21
Gambar 4.1 Diagram alir sistem.....	22
Gambar 4.2 Diagram alir <i>preprocessing</i> menggunakan metode otsu	24
Gambar 4.3 Diagram alir mencari nilai y_1 , y_2 , <i>histogram1</i> , dan <i>histogram2</i>	26
Gambar 4.4 Diagram alir mencari nilai <i>Threshold</i>	27
Gambar 4.5 Diagram alir ekstraksi fitur	29
Gambar 4.6 Diagram alir proses pengujian	31
Gambar 4.7 Diagram alir proses pelatihan	32
Gambar 4.8 Rancangan antarmuka halaman utama	55
Gambar 4.9 Rancangan antarmuka pelatihan data	56
Gambar 5.1 Implementasi halaman utama	68
Gambar 5.2 Implementasi halaman pengujian.....	69
Gambar 6.1 Pengujian terhadap perbandingan	71
Gambar 6.2 Pengujian terhadap nilai peubah <i>Threshold (T)</i>	73
Gambar 6.3 Pengujian terhadap nilai k	75
Gambar 6.4 Pengujian terhadap nilai m	77

DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian penulis dengan penelitian sebelumnya.....	5
Tabel 2.2 Format piksel 8 bit.....	11
Tabel 4.1 Nilai piksel citra kopi berpenyakit.....	34
Tabel 4.2 Hasil perhitungan y_1	35
Tabel 4.3 Hasil perhitungan y_2	35
Tabel 4.4 Hasil perhitungan nilai ni_1 (<i>histogram1</i>).....	36
Tabel 4.5 Hasil perhitungan nilai ni_2 (<i>histogram2</i>).....	37
Tabel 4.6 Hasil perhitungan Total Rata-rata dan Momen Kumulatif ke-1 berdasarkan hasil perhitungan ni_1	39
Tabel 4.7 Hasil perhitungan Total Rata-rata dan Momen Kumulatif ke-1 berdasarkan hasil perhitungan ni_2	40
Tabel 4.8 Hasil perhitungan <i>Momen Kumulatif</i> ke-0 berdasarkan hasil perhitungan ni_1	42
Tabel 4.9 Hasil perhitungan <i>Momen Kumulatif</i> ke-0 berdasarkan hasil perhitungan ni_2	43
Tabel 4.10 Hasil perhitungan varian, maksimum varian, dan <i>Threshold1</i> (t_1).....	45
Tabel 4.11 Hasil perhitungan varian, maksimum varian, dan <i>Threshold2</i> (t_2).....	46
Tabel 4.12 Hasil akhir pengolahan citra.....	47
Tabel 4.13 Nilai piksel citra <i>Otsu</i>	48
Tabel 4.14 Nilai data latih secara terpisah.....	49
Tabel 4.15 Nilai data uji.....	49
Tabel 4.16 Hasil perhitungan jarak data uji ke-1	50
Tabel 4.17 Hasil perhitungan jarak data uji ke-2	50
Tabel 4.18 Hasil pengurutan perhitungan jarak data uji ke-1	51
Tabel 4.19 Hasil pengurutan perhitungan jarak data uji ke-2	51
Tabel 4.20 Nilai keanggotaan kelas 1 data uji ke-1.....	51
Tabel 4.21 Nilai keanggotaan kelas 2 data uji ke-1.....	52
Tabel 4.22 Nilai keanggotaan kelas 1 data uji ke-2.....	53
Tabel 4.23 Nilai keanggotaan kelas 2 data uji ke-2.....	54
Tabel 5.1 Spesifikasi perangkat keras	57
Tabel 5.2 Spesifikasi perangkat lunak.....	57

Tabel 6.1 Hasil uji coba akurasi terhadap perbandingan data latih dan data uji .	70
Tabel 6.2 Hasil uji coba akurasi terhadap nilai peubah <i>Threshold</i> (<i>T</i>)	72
Tabel 6.3 Hasil uji coba akurasi terhadap nilai <i>k</i>	74
Tabel 6.4 Hasil uji coba akurasi terhadap nilai <i>m</i>	76



DAFTAR LAMPIRAN

Lampiran A data latih dan data uji.....	80
Lampiran B log skenario pengujian.....	93



BAB 1 PENDAHULUAN

1.1 Latar belakang

Kopi merupakan salah satu jenis tanaman perkebunan yang banyak dibudidayakan oleh petani. Kopi mulai diperkenalkan di Indonesia pada tahun 1696, yang dibawa oleh *Vereegnigde Oostindische Compagnie (VOC)*. Awalnya, kopi hanya ditanam di pulau Jawa dan hanya bersifat coba-coba, akan tetapi karena hasilnya memuaskan dan dipandang oleh *VOC* cukup menguntungkan sebagai komoditi perdagangan, maka *VOC* menyebarkannya ke berbagai daerah agar para penduduk menanamnya (Najiyanti dan Danarti, 2004). Kopi (*Coffea* spp. L) merupakan salah satu hasil komoditi perkebunan yang memiliki nilai ekonomis yang cukup tinggi di antara tanaman perkebunan lainnya dan berperan penting sebagai sumber devisa negara. Kopi tidak hanya berperan penting sebagai sumber devisa melainkan juga merupakan sumber penghasilan bagi tidak kurang dari satu setengah juta jiwa petani kopi di Indonesia (Rahardjo, 2012). Mutu dan kualitas hasil kopi sangat dipengaruhi oleh proses budidaya kopi. Teknologi budidaya dan pengolahan kopi meliputi pemilihan bahan tanam kopi unggul, pemeliharaan, pemangkasan tanaman dan pemberian penaung, pengendalian hama dan gulma, pemupukan yang seimbang, pemanenan, serta pengolahan kopi pasca panen. Pengolahan kopi sangat berperan penting dalam menentukan kualitas dan cita rasa kopi (Rahardjo, 2012). Akan tetapi produksi kopi Indonesia belum dapat memenuhi permintaan dan pemesanan akan kopi berkualitas baik. Rendahnya kualitas kopi yang dihasilkan pada tahap usaha tani terus berdampak sampai ke tahap pengolahan hasil kopi. Oleh karena produksi kopi di Indonesia yang belum ditangani dengan serius mengakibatkan rendahnya kualitas hasil kopi di Indonesia, terutama untuk jenis kopi robusta (CBI, 2013). Oleh karena itu, ketidak konsistenan kuantitas yang berkaitan dengan kualitas produksi kopi menyebabkan rendahnya posisi tawar kopi Indonesia di dunia.

Rendahnya kualitas hasil kopi di Indonesia merupakan masalah utama petani Indonesia. Selain pengolahan kopi, pembudidayaan dan proses tanam kopi juga merupakan tantangan bagi petani kopi karena kopi merupakan tanaman yang banyak disukai oleh jenis serangga dan hama, sampai saat ini tercatat lebih dari 900 jenis serangga hama pada tanaman kopi yang tersebar diseluruh dunia. Di Indonesia terdapat beberapa jenis hama yang merupakan hama utama tanaman kopi, yaitu hama penggerek buah kopi (PBKo) *Hypothenemus hampei*, penggerek cabang hitam *Xyloborus compactus*, penggerek cabang coklat *Xylosandrus morigerus*, kutu hijau *Coccus viridis*, dan penggerek batang merah *Zeuzera coffea* (Danarti, 2004). Hama dan penyakit dalam budidaya tanaman kopi dapat menurunkan produksi cukup besar yaitu sampai 40% (Darwis, 2013). Penyakit pada tanaman kopi bisa dideteksi melalui gejala yang ditampakkan pada daun seperti penyakit karat daun kopi, bercak daun kopi, jamur upas, dan jelaga.

Untuk identifikasi penyakit pada tanaman kopi, sebagian besar petani kopi hanya mengamati pada tampilan luar tanaman. Oleh karena pengamatan

dilakukan dengan mata telanjang, maka penyakit tanaman kopi sering kali tidak teridentifikasi dengan akurat, sehingga menyebabkan penyakit yang lebih serius pada tanaman. Identifikasi yang benar perlu dilakukan untuk tindakan pencegahan dan pengobatan pada penyakit (Katiman, 2015). Dengan menerapkan teknologi informasi dan komunikasi di bidang pertanian, seharusnya masalah seperti ini dapat ditangani.

Salah satu penerapan teknologi informasi dan komunikasi di bidang pertanian yaitu dapat digunakan sebagai alat bantu klasifikasi penyakit tanaman kopi. Klasifikasi merupakan suatu proses penggabungan atau pengelompokan dua atau lebih data sesuai dengan kesamaan suatu kriteria tertentu. Metode klasifikasi *Data Mining* yang biasa digunakan antara lain metode *K-Nearest Neighbor*, *Naive Bayes*, ID3, C4.5 dan metode lainnya. *K-Nearest Neighbor (KNN)* merupakan metode klasifikasi termasuk kelompok *Instance-Based Learning*. Algoritma ini juga merupakan salah satu teknik *Lazy Learning*. *KNN* dilakukan dengan mencari kelompok *k* objek dalam data training yang paling dekat (mirip) dengan objek pada data baru atau data testing (Ricky, et al. 2014). Algoritma *K-Nearest Neighbor (K-NN atau KNN)* dilakukan dengan melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Salah satu penerapan penelitian yang menggunakan metode *K-Nearest Neighbor (KNN)* adalah penggunaan metode *K-Nearest Neighbor* untuk pengklasifikasian penyakit tanaman jeruk keprok dengan citra daun. Dalam penelitian ini, metode *K-Nearest Neighbor (KNN)* dipakai untuk klasifikasi data uji yaitu daun jeruk sebagai objeknya, dan hasil dari penelitian tersebut adalah identifikasi penyakit tanaman jeruk dengan akurasi mencapai 96,67% (Apladosi, 2014).

Modifikasi penggunaan metode *KNN* perlu ditingkatkan untuk mengetahui tingkat akurasi metode tersebut apabila digabungkan dengan metode lain. Salah satu cara untuk menemukan keakurasian tersebut adalah dengan menggabungkannya dengan metode *Fuzzy*. Contoh penerapan *Fuzzy K-NN* adalah pada sebuah penelitian dengan penerapan metode *Fuzzy K-NN* pada klasifikasi jenis mangga berdasarkan tekstur daun yang mengambil objek tekstur tanaman buah mangga dan diekstraksi fitur citranya dan menghasilkan akurasi 81,1574% (Achmad, 2015).

Oleh karena itu, pada skripsi ini penulis mencoba mengimplementasikan algoritma *Fuzzy K-Nearest Neighbor (FK-NN)* untuk klasifikasi penyakit tanaman kopi. Dengan penerapan metode ini diharapkan akan menghasilkan akurasi yang lebih tinggi daripada penelitian yang sudah ada. Berdasarkan latar belakang yang telah dijelaskan di atas, maka penulis mengambil judul untuk skripsi ini adalah "Implementasi Metode *Fuzzy K-Nearest Neighbor* Untuk Klasifikasi Penyakit Tanaman Kopi Berdasarkan Citra Daun".

1.2 Rumusan masalah

Berdasarkan ulasan latar belakang diatas, maka rumusan masalah yang didapat adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode *Fuzzy K-Nearest Neighbor (FK-NN)* untuk klasifikasi penyakit tanaman kopi berdasarkan citra daun?
2. Bagaimana akurasi yang diperoleh dari klasifikasi menggunakan metode *Fuzzy K-Nearest Neighbor (FK-NN)* pada tanaman kopi berdasarkan citra daun?

1.3 Tujuan

Sesuai dengan rumusan masalah yang telah dijelaskan diatas, tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan metode *Fuzzy K-Nearest Neighbor (FK-NN)* untuk klasifikasi penyakit tanaman kopi berdasarkan citra daun.
2. Mengetahui akurasi yang diperoleh dari klasifikasi menggunakan metode *Fuzzy K-Nearest Neighbor (FK-NN)* pada tanaman kopi berdasarkan citra daun.

1.4 Manfaat

Hasil penelitian ini diharapkan bermanfaat bagi:

1. Bagi petani kopi
Sebagai masukan dan solusi dalam menentukan jenis penyakit pada kopi sehingga klasifikasi penyakit kopi tidak dilakukan secara manual dan menambah tingkat akurasi.
2. Bagi peneliti
Penelitian ini diharapkan dapat menambah wawasan ilmu pengetahuan dengan terjun langsung ke lapangan dalam pengambilan kebutuhan data sehingga menambah pengetahuan yang lebih mendalam terutama pada bidang yang dikaji yaitu implementasi metode *Fuzzy K-Nearest Neighbor* dengan objek daun pada tanaman kopi.

1.5 Batasan masalah

1. Penyakit yang dikenali terbatas pada penyakit yang menampakkan gejala daun, yaitu karat daun kopi, bercak daun kopi.
2. Citra daun yang dikenali hanya pada bagian penyakit.
3. Citra daun hanya bisa mengenali satu penyakit tiap daun.
4. Daun kopi yang digunakan sebagai data merupakan kopi jenis robusta.
5. Pengambilan gambar dengan pencahayaan yang sama untuk semua daun.

1.6 Sistematika pembahasan

Sistematika pembahasan dalam penelitian ini terbagi kedalam tujuh bab, sebagai berikut:

BAB 1 : PENDAHULUAN

Bab pendahuluan terdiri dari latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB 2 : KAJIAN PUSTAKA DAN DASAR TEORI

Terdiri dari kajian pustaka dan dasar teori yang meliputi kopi, penyakit kopi, pengertian pengolahan citra digital, pengertian metode *Otsu*, *Data Mining*, sistem *Fuzzy*, *K-Nearest Neighbor*, *Fuzzy K-Nearest Neighbor (FK-NN)*.

BAB 3 : METODOLOGI PENELITIAN DAN PERANCANGAN

Bab metodologi penelitian dan perancangan berisi tentang langkah-langkah yang digunakan dalam pembuatan sistem. Meliputi studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, dan pengujian.

BAB 4 : PERANCANGAN SISTEM

Bab perancangan sistem membahas tentang perancangan yang akan dilakukan terhadap pembuatan sistem. Terdiri dari diagram alir sistem, perhitungan manual dan perancangan antarmuka.

BAB 5 : IMPLEMENTASI

Bab implementasi membahas penerapan metode *Fuzzy K-NN* untuk klasifikasi penyakit tanaman kopi berdasarkan citra daun. Terdiri dari spesifikasi sistem, implementasi kode, dan implementasi antarmuka.

BAB 6 : PENGUJIAN

Bab pengujian memuat proses dan hasil pengujian terhadap sistem yang telah dibuat. Terdiri dari pengujian terhadap empat variabel yaitu perbandingan data latih dan data uji, nilai peubah *Threshold*, nilai *k*, dan nilai *m*

BAB 7 : PENUTUP

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang telah dibuat dalam penelitian serta memuat saran-saran untuk pengembangan selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini merupakan landasan kepustakaan yang berisi kajian pustaka dan dasar teori. Kajian pustaka memberikan informasi mengenai beberapa penelitian yang sudah ada. Dasar teori memberikan informasi mengenai teori-teori yang dipakai dalam penelitian dan penyusunan laporan.

2.1 Kajian pustaka

Berdasarkan judul dan topik skripsi yang dibahas, penulis mencari dan menemukan beberapa hasil penelitian yang relevan untuk mendukung penelitian dalam skripsi yang berjudul “Implementasi Metode *Fuzzy K-Nearest Neighbor* Untuk Klasifikasi Penyakit Tanaman Kopi Berdasarkan Citra Daun” ini. Pada Tabel 2.1 menjelaskan tentang penelitian yang pernah diadakan sebelumnya.

Tabel 2.1 Perbandingan penelitian penulis dengan penelitian sebelumnya

No	Judul	Objek	Metode	Parameter	Hasil Penelitian
1	“Analisis Dan Perancangan Aplikasi Sistem Pakar Dengan Metode <i>Backward Chaining</i> Untuk Mendiagnosis Penyakit Tanaman Kopi”	Tanaman Kopi	<i>Backward Chaining</i>	Gejala pada kopi	Diagnosa penyakit kopi berdasarkan gejala yang ditampakan oleh kopi dengan metode <i>Backward Chaining</i>
2	<i>Application Of support Vector Machine for Rice Diseases using shape and color</i>	Citra daun padi	Pengolahan citra dan Support Vending Machine (SVM)	<ol style="list-style-type: none"> 1. <i>Contrast</i> 2. <i>Uniformity</i> 3. <i>Entropy</i> 4. <i>Inverse Difference</i> 5. <i>Linearity correlation</i> 6. <i>Rectangularity</i> 7. <i>Compactness</i> 8. <i>Elongation</i> 9. <i>roundness</i> 	Hasil klasifikasi pada daun padi (bercak coklat dan blas)
3	<i>A Fuzzy K-NN Approach For Cancer Diagnosis With Microarray Gene Expression Data</i>	Gen DNA	Metode <i>Fuzzy K-NN</i>	<ol style="list-style-type: none"> 1. <i>EWS</i> 2. <i>RMS</i> 3. <i>BL</i> 4. <i>NB tumors.</i> 	Diagnosa Kanker dan Tumor
4	Implementasi Metode <i>Fuzzy K-Nearest Neighbor</i> Untuk Klasifikasi	Citra daun kopi dengan penyakit bercak	Metode <i>Fuzzy FK-NN</i>	<ol style="list-style-type: none"> 1. Rata-Rata <i>Red</i> 2. Rata-Rata <i>Green</i> 3. Rata-Rata <i>Blue</i> 	Hasil klasifikasi penyakit kopi untuk penyakit

Penyakit Tanaman Kopi Berdasarkan Citra Daun	daun, karat daun, dan embun jelaga			bercak daun, karat daun, dan embun jelaga.
--	------------------------------------	--	--	--

Tabel 2.1 menunjukkan perbandingan dalam penelitian sebelumnya yang berkaitan dengan penelitian penulis. Salah satu penelitian yaitu penelitian yang dilakukan oleh Abragus Sabra dengan tujuan merancang aplikasi sistem pakar untuk mendiagnosa penyakit tanaman kopi. Dalam menentukan penyakit tanaman kopi, penelitian ini menggunakan metode *Backward Chaining*. Aplikasi yang dirancang dalam penelitian ini dalam bentuk website interaktif dengan meminta masukan dari *user* dan akan menampilkan output yaitu jenis penyakit dari kopi.

Penelitian kedua berjudul "*Application Of support Vector Machine for Rice Diseases using shape and color*" yang dilakukan oleh Qing Yao pada tahun 2009 yaitu pembuatan perangkat lunak untuk identifikasi penyakit padi berdasarkan citra daun padi yang telah terinfeksi penyakit. Objek yang digunakan adalah daun dari padi yang memiliki penyakit hawar, hawar pada pelepah padi, dan blas. *Preprocessing* citra yang digunakan *Image Acquisition, Preprocessing Image, Segmentasi Image, Feature Extraction*, klasifikasi dengan *Support Vector Machine*. Hasil dari penelitian ini adalah adalah klasifikasi penyakit bakteri hawar, hawar pada pelepah padi, dan blas dengan tingkat akurasi 97,2%.

Metode *Fuzzy K-NN* juga digunakan Çiğdem Beyan, Hasan Oğul dengan penelitian berjudul "*A Fuzzy K-NN Approach For Cancer Diagnosis With Microarray Gene Expression Data*". Dataset yang kita digunakan dalam penelitian ini adalah dataset yang diproduksi oleh teknologi *Oligonucleotidebased Microarray*. Menggunakan 6 dataset yang memiliki 2.9 kategori yang berbeda, 50-102 sampel dan 2308- 10.509 gen. Tujuan dari penelitian ini adalah untuk mendiagnosa tumor dan kanker. *Fuzzy K-NN* digunakan untuk memecahkan masalah tersebut. Penelitian ini mencapai tingkat akurasi 94.0% pada objek *SRBCT*.

Dari beberapa referensi tersebut, maka pada penelitian ini penulis mengusulkan judul "*Implementasi Metode Fuzzy K-Nearest Neighbor Untuk Klasifikasi Penyakit Tanaman Kopi Berdasarkan Citra Daun*". Objek yang digunakan dalam penelitian ini yaitu daun kopi yang terinfeksi penyakit dan diambil citranya dengan kamera digital. *Preprocessing* citra yang digunakan terdiri dari proses memisahkan bagian daun yang terkena penyakit dan tidak dengan menggunakan metode *Otsu*, lalu bagian yang berpenyakit diekstraksi fitur sehingga didapat nilai rata-rata *red*, rata-rata *green*, dan rata-rata *blue* dari setiap bagian daun yang berpenyakit. Setelah dilakukan *preprocessing*, nilai yang telah diekstraksi fitur dilakukan pengklasifikasian dengan menggunakan metode *Fuzzy K-Nearest Neighbor*.

2.2 Kopi

Tanaman kopi termasuk dalam Kingdom Plantae, Sub kingdom *Tracheobionta*, Super divisi *Spermatophyta*, Divisi *Magnoliophyta*, Class *Magnoliopsida/Dicotyledons*, Sub class *Asteridae*, Ordo *Rubiales*, Famili *Rubiaceae*, Genus *Coffea*, Spesies *Coffea arabica* L (USDA, 2002). Tanaman kopi termasuk dalam famili Rubiaceae dan terdiri atas banyak jenis, yakni *Coffea arabica*, *Coffea robusta* dan *Coffea liberica* (USDA, 2002).

Kopi (*Coffea spp.* L) merupakan salah satu hasil komoditi perkebunan yang memiliki nilai ekonomis yang cukup tinggi diantara tanaman perkebunan lainnya dan berperan penting sebagai sumber devisa negara. Kopi tidak hanya berperan penting sebagai sumber devisa melainkan juga merupakan sumber penghasilan bagi tidak kurang dari satu setengah juta jiwa petani kopi di Indonesia (Rahardjo, 2012). Mutu dan kualitas hasil kopi sangat dipengaruhi oleh proses budidaya kopi. Teknologi budidaya dan pengolahan kopi meliputi pemilihan bahan tanam kopi unggul, pemeliharaan, pemangkasan tanaman dan pemberian penaung, pengendalian hama dan gulma, pemupukan yang seimbang, pemanenan, serta pengolahan kopi pasca panen. Pengolahan kopi sangat berperan penting dalam menentukan kualitas dan cita rasa kopi (Rahardjo, 2012).

2.3 Penyakit kopi

2.3.1 Karat daun kopi

Penyakit karat daun kopi (*coffee leaf rust*) yang disebabkan oleh jamur *Hemileia vastatrix* B. et Br. adalah penyakit kopi paling penting dan paling berpengaruh di seluruh dunia, hal ini juga berlaku untuk tanaman kopi arabika di Indonesia. Penyakit ini dapat menyebabkan penurunan produksi mencapai 50% (Zambolim *et al.*, 1997 *cit.* Haddad *et al.*, 2009). Gejala dari penyakit karat daun adalah sisi bawah daun yang terserang karat menunjukkan adanya bercak-bercak yang semula berwarna kuning muda yang akhirnya akan menjadi kuning tua dan coklat. Pada bercak terbentuk tepung berwarna jingga cerah (*bright orange*) yang terdiri atas *urediospora* jamur. Bercak tua berwarna coklat tua sampai hitam dan mengering, daun akhirnya gugur sehingga pohon menjadi gundul (Semangun, 2000).

Gangguan penyakit ini tidak hanya mempengaruhi pertumbuhan tanaman, tapi juga menurunkan hasil produksi biji kopi. Meluasnya bercak pada daun sebagai tanda berkembangnya penyakit, menyebabkan area fotosintesis berkurang secara signifikan yang berdampak pada menurunnya pertumbuhan tanaman. Banyaknya daun yang gugur sebagai gejala lanjut dari penyakit ini menyebabkan jumlah bunga yang terbentuk berkurang, yang berdampak pada turunnya jumlah biji kopi yang dihasilkan (Dina, 2013). Dengan kata lain, penyakit karat daun pada kopi menyerang dengan mengurangi daerah fotosintesis pada daun. Citra daun kopi yang terinfeksi penyakit karat ditunjukkan pada Gambar 2.1.



Gambar 2.1 Citra daun berpenyakit karat daun

Gambar 2.1 merupakan contoh citra daun kopi yang mempunyai penyakit karat. Sisi bawah daun yang terserang karat menunjukkan adanya bercak-bercak yang semula berwarna kuning muda yang akhirnya akan menjadi coklat.

2.3.2 Bercak daun kopi

Penyakit ini disebabkan oleh jamur *C. coffeicola* yang dapat muncul di pembibitan sampai tanaman dewasa serta menyerang buah kopi. Daun yang sakit timbul bercak berwarna kuning yang tepinya dikelilingi halo (lingkaran) berwarna kuning. Penyakit ini umumnya dijumpai di pertanaman yang kurang mendapat pemeliharaan. Penyebaran penyakit dibantu oleh keadaan lingkungan yang lembab dan pola tanam yang kurang baik. Penyebaran penyakit melalui spora yang terbawa angin dan aliran air hujan serta alat-alat pertanian. Pengendalian penyakit dengan sanitasi kebun dan membuang bagian-bagian yang sakit, kemudian membenamkannya di dalam tanah. Mengurangi kelembaban kebun dengan pemangkasan, pengaturan naungan dan membuat parit *drainase*. Melakukan pemupukan dan hindari penggunaan bibit yang telah terserang penyakit ini (Departemen pertanian, 2002).



Gambar 2.2 Citra daun berpenyakit bercak daun

Gambar 2.2 merupakan contoh citra daun kopi yang mempunyai penyakit bercak daun. Daun yang sakit timbul bercak berwarna kuning yang tepinya dikelilingi halo (lingkaran) berwarna kuning.

2.3.3 Embun jelaga

Jamur embun jelaga dalam bahasa asing disebutkan *sooty mold*. Meskipun penyakit embun jelaga bukan penyakit penting tanaman kopi tetapi keberadaannya mengganggu proses fotosintesis. Jamur ini tampak berwarna hitam. Jamur embun jelaga disebabkan oleh *Capnodium coffeae*, masuk dalam kelas *Ascomycetes* dan termasuk dalam jamur tidak sempurna (*imperfecti*). *C. coffeae* merupakan jamur *saprophyt* (non parasit) yang mempunyai sifat tidak mengambil makanan dari tanaman inang tetapi mengganggu tanaman inang dalam hal proses fotosintesis. *Miselium* jamur ini pada daun kopi ditandai dengan warna hitam. Warna hitam dari embun jelaga disebabkan karena adanya pigmen melanoid pada dinding sel hifa yang membentuk koloni (*miselium*). *Miselium* tumbuh di permukaan daun sehingga menutupi *stomata* (mulut daun) dan masuk ke dalam jaringan daun. *Miselium* terbentuk dari hifa yang menjalin dan menenun.

Jamur embun jelaga memiliki dinding sel *mucilaginous* yang berfungsi menyerap kelembapan untuk pertumbuhan *C. coffeae*. Hasil ekskresi dari serangga penghisap berupa madu dan kotorannya dimanfaatkan sebagai media tumbuh *C. coffeae*. Kotoran dari serangga penghisap mengandung gula, asam amino, protein, mineral dan vitamin. Kandungan tersebut sangat diperlukan untuk pertumbuhan *C. coffeae*. *C. coffeae* juga dapat tumbuh pada hasil eksudat yang diproduksi oleh kelenjar trikoma (rambut daun). Apabila kondisi udara kering *miselium* dapat lepas dari daun dan pecah menjadi bagian-bagian kecil yang terhembus angin. *C. coffeae* akan berkembang biak cepat pada musim kemarau dan berkurang pada musim hujan. Perkembangan *C. coffeae* dipengaruhi oleh jarak tanaman yang terlalu rapat, naungan yang terlalu banyak, kondisi hangat, suhu tinggi dan kering.



Gambar 2.3 Citra daun berpenyakit embun jelaga

Gambar 2.3 merupakan contoh citra daun yang memiliki penyakit embun jelaga. Dinding sel pada daun jelaga berwarna hitam. Pada bagian yang berwarna hitam, jelaga akan menutupi daun sehingga mencegah proses fotosintesis dari daun.

2.4 Definisi citra digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar X, hasil *CT Scan* dll. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer (T,Sutoyo et al. 2009).

Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel (piksel = *picture element*), yaitu elemen terkecil dari sebuah citra. 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 itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks berikut.

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & \dots & \dots & f(1, M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{matrix} \quad (2.1)$$

Berdasarkan gambaran tersebut, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas $f(x,y)$, dimana harga x (baris) dan y (kolom) merupakan koordinat posisi dan $f(x,y)$ adalah nilai fungsi pada setiap titik (x,y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Pada proses digitalisasi (*sampling* dan kuantitas) diperoleh besar baris M dan kolom N hingga citra membentuk matriks $M \times N$ dan jumlah tingkat keabuan piksel (T, Sutoyo *et al.*2009).

2.4.1 Jenis-jenis citra digital

Jangkauan piksel suatu citra berbeda-beda tergantung warna citranya, tapi secara umum rentang piksel citra adalah 0-255. Berikut adalah jenis citra digital berdasarkan nilai pikselnya

- Citra Biner atau monokrom adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner juga disebut sebagai citra *B&W (black and white)* atau citra monokrom. Untuk mewakili nilai setiap piksel dari citra biner hanya dibutuhkan 1 bit. Citra biner sering kali muncul sebagai hasil dari proses segmentasi, pengembangan, morfologi, ataupun dithering. Citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner juga disebut sebagai citra *B&W (black and white)* atau citra monokrom. Untuk mewakili nilai setiap piksel dari citra biner hanya dibutuhkan 1 bit. Citra biner sering kali muncul sebagai hasil dari proses segmentasi, pengembangan, morfologi, ataupun dithering (darma putra, 2010).
- Citra *Grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian *RED=GREEN=BLUE*. Nilai tersebut digunakan untuk menunjukkan tingkat

intensitas. Warna yang dimiliki adalah warna hitam, keabuan, dan putih (darma putra, 2010).

- Citra Warna (8 bit) merupakan citra digital yang memiliki setiap piksel dari citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Ada 2 jenis citra warna 8 bit (darma putra, 2010).
 1. citra warna dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu.
 2. setiap piksel memiliki format 8 bit sebagai yang ditunjukkan pada Tabel.

Tabel 2.2 Format piksel 8 bit

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	G	G	G	B	B

2.4.2 Citra warna (*true colour*)

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti mempunyai gradasi sebanyak 255. Penyimpanan citra *true color* didalam memori berbeda dengan citra *grayscale*. Setiap piksel dari citra *grayscale* 256 gradasi warna diwakili oleh 1 byte. Sedangkan 1 piksel citra *true color* diwakili oleh 3 byte yang masing- masing byte merepresentasikan warna merah (*Red*), hijau (*Green*), biru (*Blue*) (T, Sutoyo et al.2009).

2.4.3 Computer vision

Computer vision adalah ilmu dan metode aplikasi dalam menggunakan komputer untuk memahami isi citra (*image content*). Area permasalahan dalam *computer vision* adalah pengukuran dan pemrosesan, yang dapat dilakukan dengan berbagai metode. Beberapa area permasalahan *computer vision* adalah sebagai berikut:

1. *Recognition*
Pengenalan bertujuan mengenali objek data citra, aplikasinya seperti *Content Based Image Retrieval* (CBIR), *Optical Character Recognition* (OCR).
2. *Motion*
Gerakan bertujuan mengenali data citra bergerak. Aplikasinya seperti *egomotion* yang membagi gerakan 3D dari kamera, *tracking* yang memperkirakan satu atau beberapa objek dalam citra bergerak
3. Restorasi citra
Bertujuan untuk mendapatkan data citra, citra bergerak atau objek 3D tanpa noise.



2.5 Pengolahan citra digital

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau *reduksi* data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. Input dari pengolahan citra adalah citra, sedangkan outputnya adalah citra hasil pengolahan (T, Sutoyo et al.2009).

2.5.1 Segmentasi

Segmentasi adalah cara untuk membagi citra menjadi beberapa daerah bagian (*region*) dimana setiap daerah memiliki kemiripan atribut. Salah satu teknik segmentasi ialah pengambangan (*Thresholding*) (darma putra, 2010).

2.5.2 Pengambangan (*Thresholding*)

Proses pengambangan akan menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih. Secara umum proses pengambangan citra *grayscale* untuk menghasilkan citra biner adalah sebagai berikut.

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2.2)$$

Dengan $g(x,y)$ adalah citra biner dari citra *grayscale* $f(x,y)$, dan T menyatakan nilai ambang. Nilai T memegang peranan yang sangat penting dalam proses pengambangan. Kualitas hasil citra biner sangat tergantung nilai T yang digunakan (darma putra, 2010).

Terdapat dua jenis pengambangan, yaitu pengambangan global (*global Thresholding*) dan pengambangan secara lokal adaptif (*locally adaptive Thresholding*). Pada pengambangan global, seluruh piksel pada citra dikonversikan menjadi hitam dan putih dengan satu nilai ambang T . Kemungkinan besar pada pengambangan global akan banyak informasi hilang karena hanya menggunakan satu nilai T untuk keseluruhan piksel. Untuk mengatasi masalah ini dapat digunakan pengambangan secara lokal adaptif. Pada pengambangan lokal, suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan pengambangan lokal pada setiap blok dengan nilai T yang berbeda (darma putra, 2010).

2.5.3 Metode *Otsu*

Tujuan dari metode *Otsu* adalah membagi *histogram* citra *gray level* kedalam dua daerah yang berbeda secara otomatis tanpa membutuhkan bantuan user untuk memasukkan nilai ambang.

Pendekatan yang dilakukan oleh metode *Otsu* adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis diskriminan akan memaksimalkan variabel tersebut agar dapat membagi objek latar depan (*foreground*) dan latar belakang (*background*).

Formulasi dari metode *Otsu* adalah dengan melibatkan nilai ambang yang akan dicari dari suatu citra *gray level* dinyatakan dengan k . Nilai k berkisar antara 0 sampai dengan L , dengan nilai $L = 255$. Probabilitas setiap piksel pada level ke- i dapat dinyatakan pada Persamaan (2.3) (Otsu, 1979).

$$P_i = n_i / N \tag{2.3}$$

Dimana: P_i adalah probabilitas piksel ke- i
 n_i menyatakan jumlah piksel pada level ke- i
 N menyatakan total jumlah piksel pada citra.

Nilai *Zeroth Cumulative Moment*, *First Cumulative Moment*, dan total nilai mean berturut-turut dapat dinyatakan dengan rumus berikut.

$$\omega(k) = \sum_{i=1}^k P_i \tag{2.4}$$

$$\mu(k) = \sum_{i=1}^k i \cdot P_i \tag{2.5}$$

$$\mu_t = \sum_{i=1}^L i \cdot P_i \tag{2.6}$$

Dimana: $\omega(k)$ adalah Momen Kumulatif ke-0
 $\mu(k)$ adalah Momen Kumulatif ke-1
 μ_t adalah nilai rata-rata

Nilai ambang k dapat diperoleh dengan menggunakan Persamaan

$$\sigma_B^2(k) = \frac{[\mu_t \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \tag{2.7}$$

Ekstraksi fitur yang digunakan dalam penelitian ini adalah rata-rata *RED*, rata-rata *GREEN*, rata-rata *BLUE*. Untuk mendapatkan fitur tersebut, maka citra harus disegmentasi terlebih dahulu sehingga hanya bagian citra daun yang berpenyakit yang akan diproses dan menghasilkan akurasi yang diinginkan.

Untuk mencari nilai *Red*, *Green*, *Blue* dari sebuah citra dengan metode *Otsu*, hal yang pertama dilakukan adalah yaitu mencari tingkat keabuan ke- i yaitu y_1 dan y_2 yang difungsikan di Persamaan (2.8) dan Persamaan (2.9).

$$y_1 = 2G - R - B \tag{2.8}$$

$$y_2 = 2R - G - B \tag{2.9}$$

Dari nilai y_1 dan y_2 yang diterapkan pada metode *Otsu*, maka didapatkan dua nilai *Threshold* (T_1 dan T_2). Dengan tujuan untuk meminimalkan *noise* yang terdapat pada citra, maka proses segmentasi juga perlu dimodifikasi dengan fungsi sebagai berikut (Yao, et al., 2009).



$$P_s = \begin{cases} (0,0,0) & y_1 > T_1 + 10 \text{ dan } y_2 < T_2 - 10 \\ P_r & y_1 < T_1 + 10 \text{ dan } y_2 > T_2 - 10 \end{cases} \quad (2.10)$$

Dimana P_r adalah nilai *Red*, *Green*, dan *Blue* (RGB) disetiap piksel asli pada citra, dan 10 adalah peubah nilai *Threshold* (T_1 dan T_2).

2.6 Ekstraksi fitur

Ekstraksi fitur adalah proses dimana setiap citra dari daun kopi akan diambil fitur nilai warna *red*, *green*, *blue* nya. Karena penyakit yang akan diklasifikasikan dapat diamati dari perbedaan warna, maka sistem akan mengambil ciri setiap daun dari rata-rata warna merah, rata-rata warna hijau, rata-rata warna biru. Formulasi untuk pengambilan rata-rata warna ditunjukkan pada Persamaan (2.11), Persamaan (2.12), Persamaan (2.13).

$$\text{rata-rata Red} = \frac{\text{total nilai red}}{\text{total pixel}} \quad (2.11)$$

$$\text{rata-rata Green} = \frac{\text{total nilai green}}{\text{total pixel}} \quad (2.12)$$

$$\text{rata-rata Blue} = \frac{\text{total nilai blue}}{\text{total pixel}} \quad (2.13)$$

2.7 Data Mining

2.7.1 Pengertian Data Mining

Data Mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam database. *Data Mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan dan *Machine Learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terakit dari berbagai database besar (Turban et al. 2005).

Menurut Gatner *Data Mining* adalah suatu proses menemukan hubungan yang berarti, pola dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan, dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika (Larose, 2005).

Kemajuan luar biasa yang terus berlanjut dalam bidang *Data Mining* didorong oleh beberapa faktor antara lain (Larose, 2005) :

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam data *warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam database yang handal.
3. Adanya peningkatan akses data melalui navigasi web dan intranet.
4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk *Data Mining* (ketersediaan teknologi).
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

2.7.2 Klasifikasi

Klasifikasi merupakan pendekatan untuk melakukan fungsi pengelompokan dalam *Data Mining* yaitu untuk menggolongkan data. Teknik klasifikasi ini dapat pula digunakan untuk melakukan prediksi atas informasi yang belum diketahui sebelumnya. Beberapa algoritma yang dapat digunakan antara lain adalah algoritma *Decision Tree C.45*, *Artificial Neural Networks (ANN)*, *K-Nearest Neighbor (KNN)*, algoritma *Naive Bayes*, *Neural Network* serta algoritma lainnya.

2.8 Sistem Fuzzy

2.8.1 Pengertian logika Fuzzy

Logika *Fuzzy* adalah logika yang berdasar pada teori himpunan *Fuzzy*. Teori ini menyatakan bahwa derajat keanggotaan dari suatu elemen himpunan bukanlah hanya terdiri dari 0 dan 1 (bukan anggota himpunan dan anggota himpunan), namun dalam rentang $[0,1]$, sehingga seolah-olah ada daerah “abu-abu” daerah yang berlogika antara 0 dan 1 (Riandy Rahman Nugrah, 2013).

2.8.2 Himpunan Fuzzy

Himpunan *Fuzzy* adalah himpunan yang derajat keanggotaan tiap elemennya berada dalam rentang $[0,1]$. Hal ini berbeda dengan himpunan biasa (disebut juga himpunan tegas) yang derajat keanggotaan tiap elemennya hanya bernilai 0 atau 1. Derajat keanggotaan suatu elemen dalam suatu himpunan X dilambangkan dengan $\mu[x]$. Pada himpunan *Fuzzy* derajat keanggotaan suatu elemen berada pada rentang $[0,1]$. Artinya bisa saja nilai keanggotaan suatu elemen pada himpunan *Fuzzy* bernilai 0.1, 0.5, 0.75, dll. Intinya derajat keanggotaan suatu elemen pada himpunan *Fuzzy* tidak harus hanya 0 dan 1 (Riandy Rahman Nugrah, 2013).

Himpunan *Fuzzy* memiliki 2 atribut yaitu :

1. *Linguistik*, yaitu penamaan suatu kelompok yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : kecil, sedang, besar.
2. *Numerik*, yaitu suatu nilai atau angka yang menunjukkan ukuran dari suatu variabel seperti: 5, 10, 15, dsb.

2.8.3 Himpunan *crisp*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu A[x]$, memiliki 2 kemungkinan, yaitu:

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

2.9 KNN

2.9.1 Pengertian KNN (K-Nearest Neighbor)

K-Nearest Neighbor (KNN) termasuk kelompok *Instance-Based Learning*. Algoritma ini juga merupakan salah satu teknik *Lazy Learning*. KNN dilakukan dengan mencari kelompok k objek dalam data training yang paling dekat (mirip) dengan objek pada data baru atau data testing (Chang, C, 2009). Algoritma *K-Nearest Neighbor* adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. *Nearest Neighbor* adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dan kasus lama yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada (Kusrini, Emha T. Luthfi, 2009).

2.9.2 Algoritma KNN (K-Nearest Neighbor)

Langkah-langkah untuk menghitung metode Algoritma K-Nearest Neighbor (Ricky et al, 2014):

- Menentukan parameter K (Jumlah tetangga paling dekat).
- Menghitung kuadrat jarak *Euclid (Query Instance)* masing-masing objek terhadap data sampel yang diberikan.
- Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak *Euclid* terkecil.
- Mengumpulkan kategori Y (Klasifikasi *Nearest Neighbor*)
- Dengan menggunakan kategori *Nearest Neighbor* yang paling mayoritas maka dapat diprediksi nilai *query instance* yang telah dihitung

2.9.3 Euclidean distance

Euclidean Distance merupakan metrik yang sering digunakan untuk mendefinisikan jarak antara dua titik yaitu titik pada data training (x) dan titik pada data testing (y).

$$D(x,y)=\sqrt{\sum_{k=1}^n(x_i - y_i)^2} \quad (2.14)$$

Dengan D adalah jarak antara titik pada data training x dan titik data testing y yang akan diklasifikasi, dimana $x=x_1,x_2,\dots,x_i$ dan $y=y_1,y_2,\dots,y_i$ dan merepresentasikan nilai atribut serta n merupakan dimensi atribut (Ricky, et al. 2014).

2.10 Fuzzy K-Nearest Neighbor

Metode *Fuzzy K-Nearest Neighbor (FK-NN)* diperkenalkan oleh Keller et al (1985) dengan mengembangkan *K-NN* yang digabungkan dengan teori *Fuzzy* dalam menyampaikan definisi pemberian label kelas pada data uji yang diprediksi. Seperti halnya pada teori *Fuzzy*, sebuah data mempunyai nilai

keanggotaan pada setiap kelas, yang artinya sebuah data bisa dimiliki oleh kelas yang berbeda dengan nilai derajat keanggotaan dalam interval [0,1]. Teori *Fuzzy K-Nearest Neighbor* melakukan klasifikasi dengan mendefinisikan nilai keanggotaan sebuah data pada masing-masing kelas.

2.11 Himpunan keanggotaan

Seperti halnya pada teori *Fuzzy*, sebuah data mempunyai nilai keanggotaan pada setiap kelas yang artinya sebuah data bisa dimiliki oleh kelas yang berbeda dengan nilai derajat keanggotaan dalam interval [0,1] (Prasetyo, 2012).

Rumus yang digunakan adalah dalam mencari nilai keanggotaan :

$$u(x, c_i) = \frac{\sum_{k=1}^K u(x_k, c_i) * d(x, y)^{\frac{-2}{(m-1)}}}{\sum_{k=1}^K d(x, y)^{\frac{-2}{(m-1)}}} \quad (2.15)$$

Dimana :

$u(x, c_i)$: nilai keanggotaan data x ke kelas c_i .

K : jumlah tetangga terdekat yang digunakan.

$u(x_k, c_i)$: nilai keanggotaan data tetangga dalam K tetangga pada kelas c_i , nilainya 1 jika data latih x_k milik kelas c_i atau 0 jika bukan milik kelas c_i .

$d(x, y)$: jarak dari data x ke data y dalam K tetangga terdekat.

m : bobot pangkat (*weight exponent*), yang besarnya : $m > 1$.

Dalam metode *FK-NN* perhitungan jarak antara dua data disesuaikan dengan tipe data, dimana setiap tipe datanya memiliki rumus masing-masing (Prasetyo, 2012).

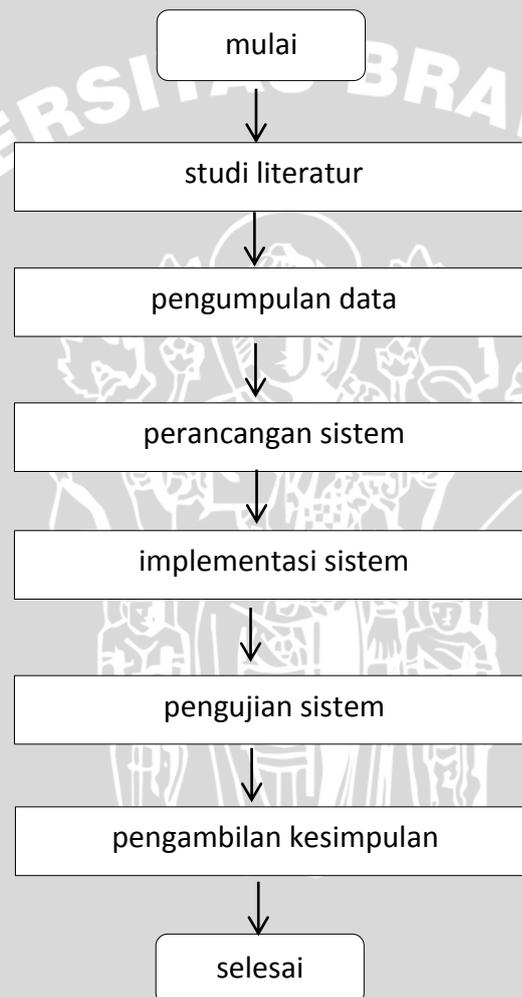
2.12 Perhitungan akurasi

Perhitungan akurasi adalah proses untuk mengetahui keberhasilan sistem. Tujuan dari perhitungan akurasi adalah untuk mengetahui presentase jumlah keberhasilan sistem dalam mengklasifikasikan penyakit pada citra daun tanaman kopi. Nilai akurasi didapatkan dengan perbandingan antara jumlah citra daun yang berhasil dan benar dikenali dengan jumlah seluruh citra daun kopi yang diuji. Formulasi tingkat akurasi ditunjukkan pada Persamaan (2.16).

$$Akurasi = \frac{\text{jumlah data benar}}{\text{jumlah Seluruh data}} \times 100\% \quad (2.16)$$

BAB 3 METODOLOGI PENELITIAN DAN PERANCANGAN

Bab ini menjelaskan tentang metodologi penelitian dan perancangan yang akan dilakukan untuk mengimplementasikan metode *Fuzzy K-Nearest Neighbor* dalam kasus klasifikasi penyakit pada citra daun tanaman kopi. Penelitian ini dilakukan dengan beberapa tahapan proses, yaitu identifikasi masalah, studi literatur, pengumpulan data, *preprocessing* yang meliputi pengambilan citra yang terfokus pada penyakit, ekstraksi ciri tekstur pada citra daun, klasifikasi data *FK-NN* dan penghitungan tingkat akurasi. Tahap-tahap yang dilakukan pada penelitian ini diilustrasikan pada Gambar 3.1.



Gambar 3.1 Tahap penelitian

Gambar 3.1 merupakan tahapan-tahapan yang dilakukan dalam penelitian. Berikut penjelasan dari masing-masing tahapan:

1. Studi literatur

Merupakan tahap metode penelitian yang bertujuan untuk mengumpulkan dan mempelajari teori-teori dasar yang selanjutnya akan diterapkan dalam

penelitian. Studi literatur yang diambil meliputi metode *FK-NN*, citra digital, tanaman kopi, dan penyakit pada tanaman kopi.

2. Pengumpulan data

Data yang dikumpulkan pada penelitian ini yaitu citra daun tanaman kopi yang berpenyakit yang diperoleh dari PT.PERKEBUNAN BUMISARI kecamatan Songgon Banyuwangi.

3. Perancangan sistem

Melakukan perancangan sistem yaitu menerapkan metode *Fuzzy K-Nearest Neighbor* untuk klasifikasi penyakit pada citra daun kopi .

4. Implementasi sistem

Implementasi sistem adalah proses pembuatan aplikasi atau perangkat lunak untuk mengklasifikasikan penyakit kopi berdasarkan citra daun dengan berdasarkan perancangan sistem yang telah dibuat.

5. Pengujian sistem

Pengujian sistem digunakan untuk melihat presentase akurasi sistem terhadap citra yang dimasukkan.

6. Pengambilan kesimpulan

Tahap untuk menyimpulkan hasil keseluruhan penelitian dari awal sampai akhir dan berdasarkan tingkat akurasi sistem yang telah dilakukan.

3.1 Studi literatur

Studi literatur bertujuan untuk mempelajari berbagai hal yang berhubungan dengan sistem yang akan dibuat pada penelitian ini, meliputi:

1. Bahasa pemrograman yang digunakan dalam pembuatan sistem yaitu bahasa pemrograman java.
2. Metode yang diimplementasikan yaitu *Fuzzy K-Nearest Neighbor*.
3. Pemrosesan awal citra menggunakan metode *Otsu*.
4. Ekstraksi fitur yang digunakan yaitu data masukan untuk pelatihan dan data masukan untuk pengujian.
5. Penyakit pada kopi yang dapat didiagnosa dari citra daun

Beberapa sumber yang menjadi referensi dan rujukan dari penelitian ini adalah penelitian-penelitian yang telah dilakukan sebelumnya, jurnal ilmiah, buku, dan *e-book* yang didapat melalui internet.

3.2 Pengumpulan data

Dalam penelitian ini, tahap pengumpulan data dilakukan di PT. Perkebunan Bumisari yang berada di daerah Banyuwangi dengan komoditi utama yaitu tanaman kopi. Data yang digunakan adalah gambar atau citra daun kopi yang

terjangkit penyakit. Setelah semua data terkumpul, daun dari tanaman kopi difoto agar menjadi data gambar atau citra digital. Jumlah data yang diambil adalah 60 data, dimana 30 data untuk penyakit karat daun, dan 30 data untuk penyakit bercak daun. Setelah gambar atau citra diperoleh, kemudian akan dilakukan pemotongan citra agar semua data citra mempunyai ukuran yang sama. Pemotongan dilakukan sebesar 200 x 150 piksel.

3.3 Analisis kebutuhan

3.3.1 Analisis kebutuhan data

Data yang terlibat dalam aplikasi klasifikasi penyakit tanaman kopi berdasarkan citra daun ini terdiri dari data masukan dan data keluaran. Data masukan merupakan data yang dimasukkan oleh user kedalam sistem, sedangkan data keluaran adalah data yang akan menjadi output dari sistem sebagai hasil dari pengolahan sistem. Data yang terlibat adalah sebagai berikut:

1. Data masukan

Data latih dan data uji yang merupakan citra daun tanaman kopi yang berpenyakit. Penyakit tanaman kopi yang nantinya akan diklasifikasikan meliputi penyakit karat daun dan bercak daun.

2. Data keluaran

Data keluaran berupa kelas hasil klasifikasi citra daun.

3.3.2 Analisis kebutuhan sistem

Kebutuhan yang akan digunakan dalam pembuatan sistem, antara lain:

1. Kebutuhan hardware meliputi:

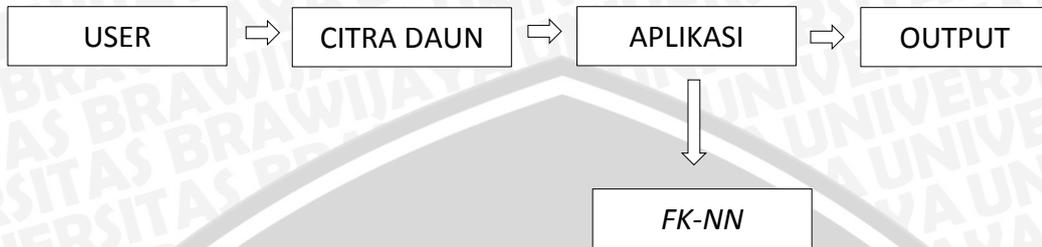
- Kamera Digital dengan spesifikasi minimal sebagai berikut:
 - a. Resolusi 15 MP
 - b. HD Quality
 - c. Canon ixus 145
- Laptop dengan spesifikasi minimal sebagai berikut:
 - a. Sistem operasi Windows 7
 - b. RAM 2 GB
 - c. Prosesor pentium 2,20 GHz

2. Kebutuhan software meliputi:

- Netbeans IDE 7.4 sebagai *tools* dari sistem yang akan dibuat dan dengan menggunakan bahasa pemrograman java.

3.4 Perancangan sistem

Sistem yang dibuat bertujuan untuk mengklasifikasikan penyakit tanaman kopi berdasarkan citra daun dengan menggunakan metode klasifikasi *FK-NN*.



Gambar 3.2 Alur sistem

Gambar 3.2 menjelaskan tentang deskripsi umum sistem. Secara umum, proses dari sistem adalah user menyediakan citra daun kopi, citra daun terdiri dari daun tanaman kopi yang terjangkit penyakit bercak daun kopi, dan karat daun kopi. Citra daun digunakan sebagai data latih dan data uji dari sistem. Kemudian citra daun akan dilakukan proses segmentasi yaitu perbaikan kualitas citra yang kemudian diekstraksi oleh aplikasi dan hasil ekstraksi diklasifikasikan berdasarkan kesamaan ciri dengan menggunakan metode *FK-NN*. Setelah proses selesai aplikasi akan mengeluarkan kelas hasil klasifikasi dari penyakit kopi.

3.5 Implementasi sistem

Pada tahap implementasi sistem, akan dilakukan pembuatan sistem yang merujuk pada perancangan sistem yang dilakukan sebelumnya. Implementasi dilakukan dengan menggunakan bahasa pemrograman Java dan model pemrograman berorientasi objek. *Tools editor* yang digunakan adalah *software* Netbeans IDE 7.4.

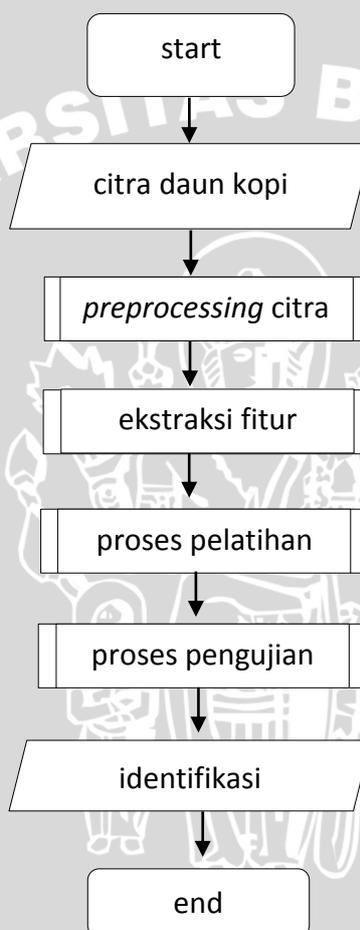
3.6. Pengujian

Pengujian sistem klasifikasi penyakit tanaman kopi ini dilakukan dengan tujuan untuk mengetahui akurasi yang didapatkan berdasarkan variabel-variabel pengujian dan juga untuk mengetahui adanya kesesuaian antara fungsi-fungsi atau layanan-layanan sistem yang diimplementasikan dengan hasil analisa kebutuhan yang sudah ditentukan pada tahap analisis. Selain itu dengan dilakukannya pengujian akan dapat diketahui adanya kesalahan-kesalahan dalam proses coding maupun dalam menghasilkan output program sehingga kesalahan-kesalahan tersebut dapat diperbaiki.

BAB 4 PERANCANGAN SISTEM

4.1 Diagram alir sistem

Diagram alir sistem menunjukkan langkah-langkah atau proses yang akan dikerjakan oleh sistem. Tujuan pembuatan diagram alir sistem adalah untuk mempermudah dalam memahami alur proses yang dilakukan oleh sistem. Dalam sistem klasifikasi penyakit tanaman kopi ini alur yang digunakan yaitu input citra daun kopi, *preprocessing* citra, ekstraksi fitur, *FK-NN*, dan hasil klasifikasi. Secara keseluruhan, diagram alir sistem ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram alir sistem

Pada Gambar 4.1 terlihat bahwa sistem dalam penelitian ini memiliki beberapa tahapan. Tahap pertama, sistem akan diberi data masukan berupa citra atau gambar daun kopi yang berpenyakit sebagai data latih dan data uji.

Tahap kedua, data masukan yang berupa data latih dan data uji akan dilakukan *preprocessing*, yaitu pemrosesan awal citra untuk menghitamkan bagian citra data yang tidak diproses sehingga sistem hanya

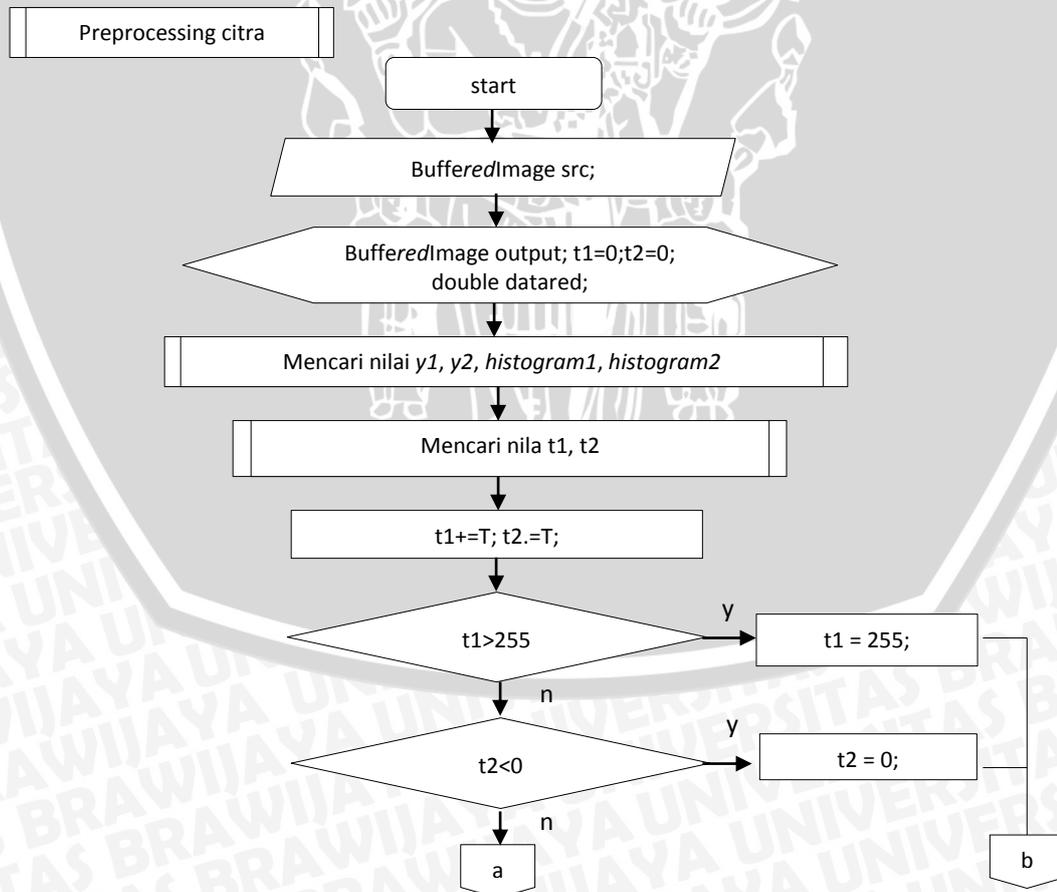
memproses bagian citra daun kopi yang berpenyakit saja. Proses *preprocessing* ini menggunakan metode *Otsu*.

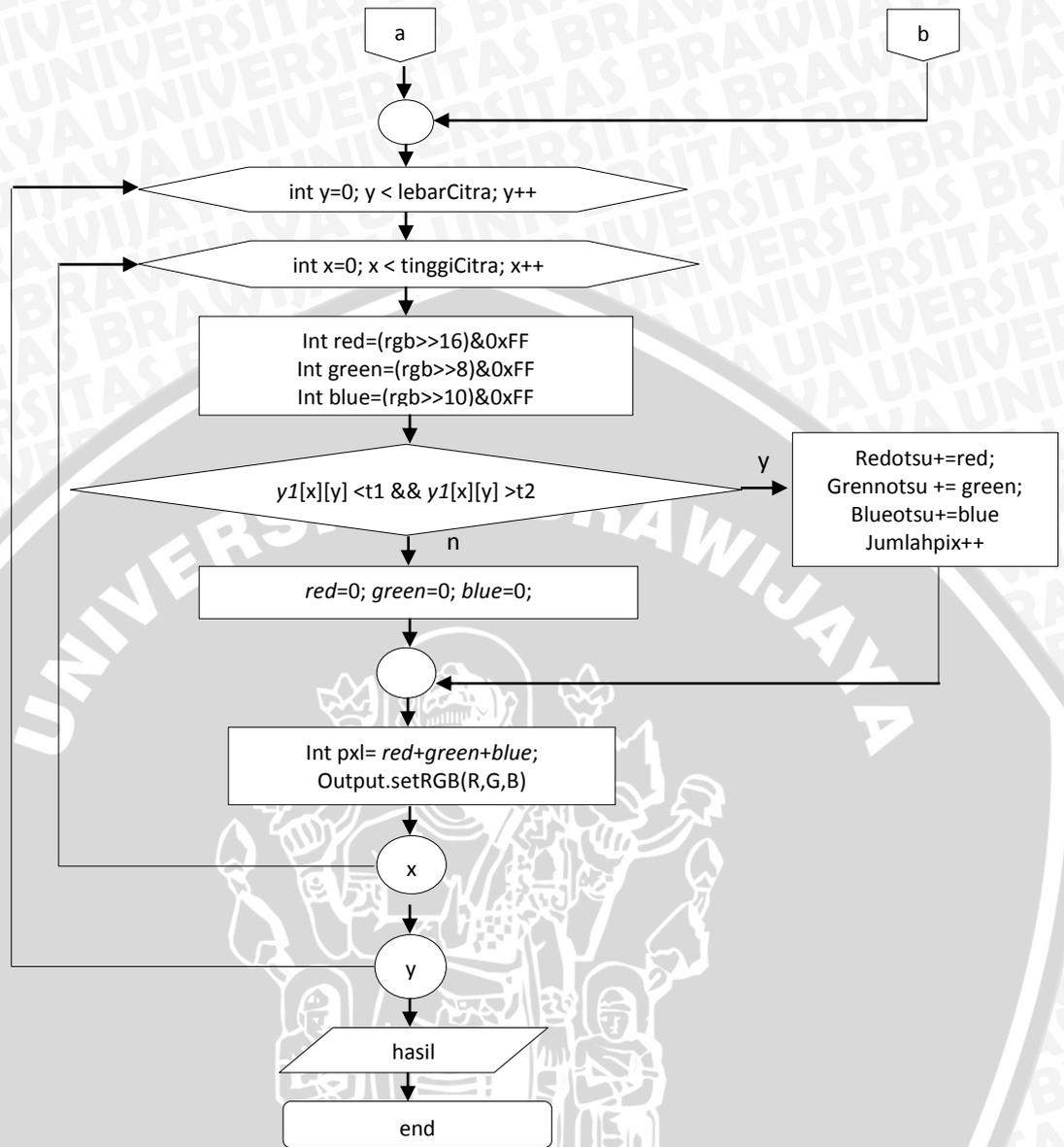
Ketiga, setelah proses segmentasi, citra akan dikonversikkan kedalam bentuk angka pada proses ekstraksi fitur. Fitur yang digunakan dalam penelitian ini adalah rata-rata warna merah (*red*), rata-rata warna hijau (*green*), rata-rata warna biru (*blue*).

Tahap keempat, adalah proses pelatihan pada data latih yang telah berupa data fitur dan tahap kelima adalah proses pengujian sistem (*testing*) pada data uji yang telah berupa data fitur. Dan tahap terakhir adalah penentuan kelas dan perhitungan akurasi, tujuannya adalah untuk mengetahui tingkat keakuratan dari metode *FK-NN* jika diterapkan pada kasus klasifikasi penyakit pada daun kopi. Hasil proses pengujian adalah klasifikasi penyakit kopi yang ada pada citra uji dan tingkat keakuratan dari metode tersebut.

4.1.1 *Preprocessing* citra

Pada proses ini, dilakukan peningkatan kualitas citra. Untuk meningkatkan kualitas citra dilakukan tahap *preprocessing* dimana latar belakang citra dihilangkan sehingga yang tertinggal hanya bagian citra yang berpenyakit. Metode yang digunakan untuk *preprocessing* ini adalah metode *Otsu*. Langkah-langkah yang dilakukan dalam *preprocessing* ditunjukkan pada Gambar 4.2.





Gambar 4.2 Diagram alir *preprocessing* menggunakan metode otsu

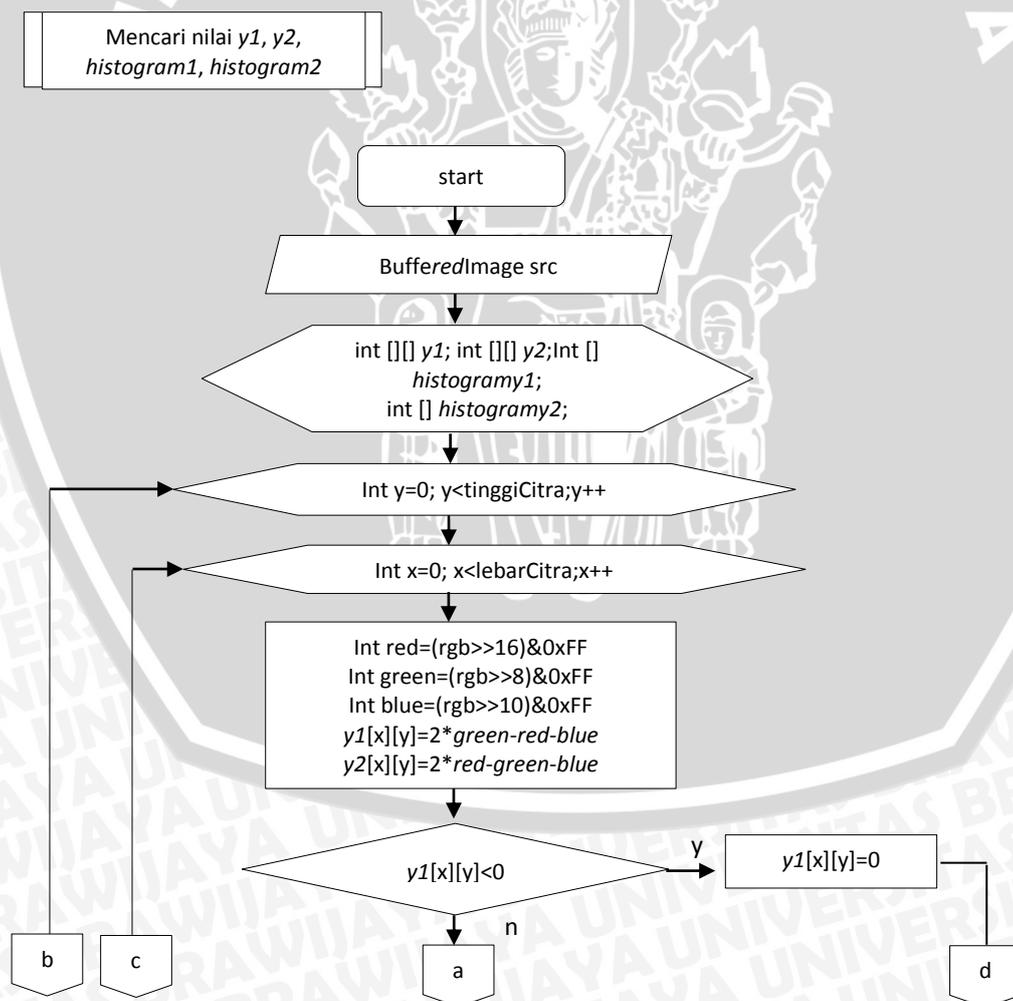
Gambar 4.2 menjelaskan tentang diagram alir dari *preprocessing* citra dengan menggunakan metode *Otsu*. Proses ini bertujuan untuk menghilangkan bagian citra yang tidak terkena penyakit dan hanya memperhitungkan bagian citra yang berpenyakit. Secara rinci, langkah-langkahnya adalah sebagai berikut:

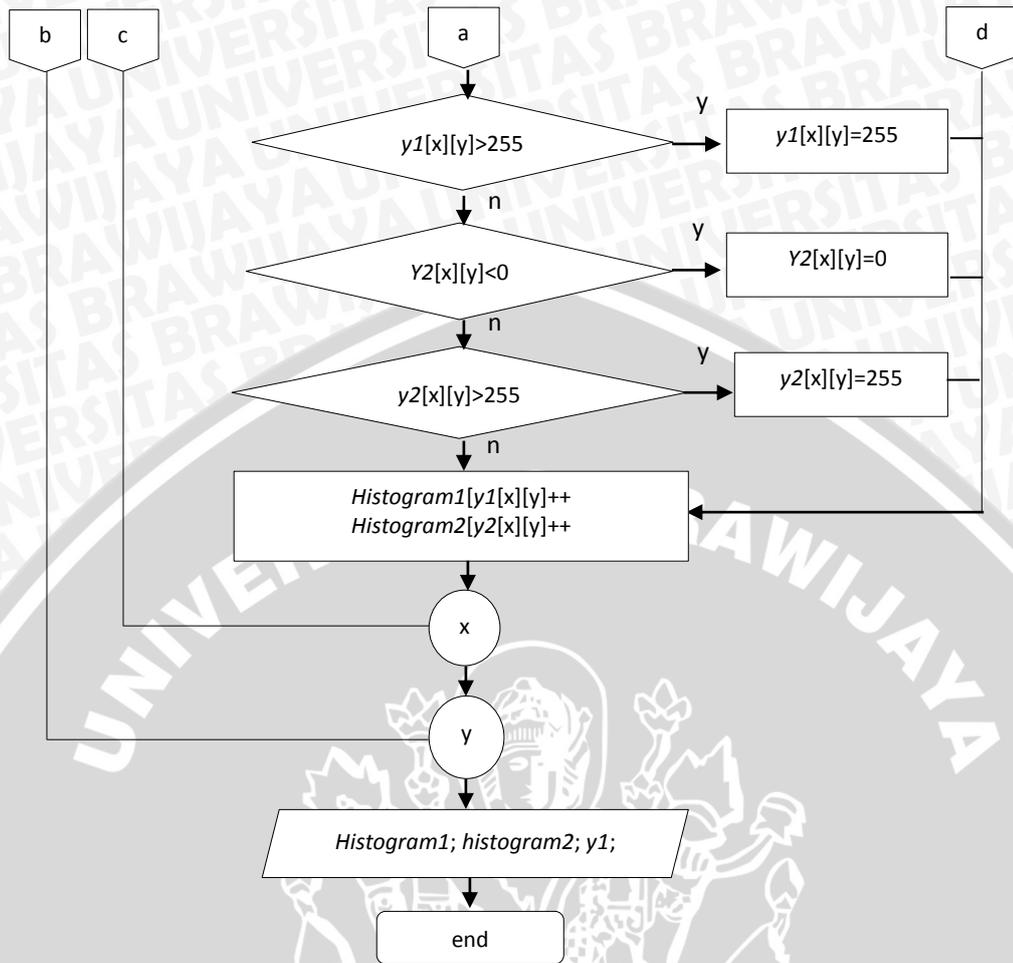
1. Mencari tingkat keabuan citra ke-*i* yaitu y_1 dan y_2 pada setiap piksel daun kopi yang dimasukkan.
2. Menghitung jumlah piksel yang memiliki tingkat keabuan ke-*i* yaitu $ni_1/histogram_1$ dan $ni_2/histogram_2$.
3. Menghitung nilai *Threshold* kesatu (t_1) dengan menggunakan ni_1 dan *Threshold* kedua (t_2) dengan menggunakan ni_2 .

4. Mendapatkan nilai *Threshold* dari metode *Otsu* yaitu dengan menambahkan nilai $t1$ dan mengurangi nilai $t2$ dengan peubah nilai *Threshold* yang telah diinisialisasi sebelumnya.
5. Melakukan normalisasi pada $t1$ dan $t2$ yaitu dengan mengganti nilai $t1$ menjadi 255 jika penambahan $t1$ dengan peubah nilai *Threshold* lebih besar dari 255 dan mengganti nilai $t2$ menjadi 0 jika pengurangan nilai $t2$ dengan peubah nilai *Threshold* lebih kecil dari 0.
6. Membandingkan nilai $y1$ dengan $t1$ dan $y2$ dengan $t2$ disetiap piksel citra daun kopi. Jika nilai $y1$ pada piksel ke- n lebih kecil dari nilai $t1$ dan nilai $y2$ pada piksel ke- n lebih besar dari nilai $t2$ maka warna dari piksel ke- n tersebut tidak berubah, namun jika kondisi tersebut tidak terpenuhi, maka warna piksel ke- n akan diubah menjadi warna hitam.

4.1.1.1 Mencari nilai $y1$, $y2$, *histogram1*, dan *histogram2*

Pada proses ini dilakukan pencarian nilai $y1$, $y2$, *histogram1*, dan *histogram2*. Perhitungan dilakukan pada setiap piksel citra daun kopi. Langkah-langkah yang dilakukan pada proses mencari nilai $y1$, $y2$, *histogram1*, dan *histogram2* adalah sebagai berikut :





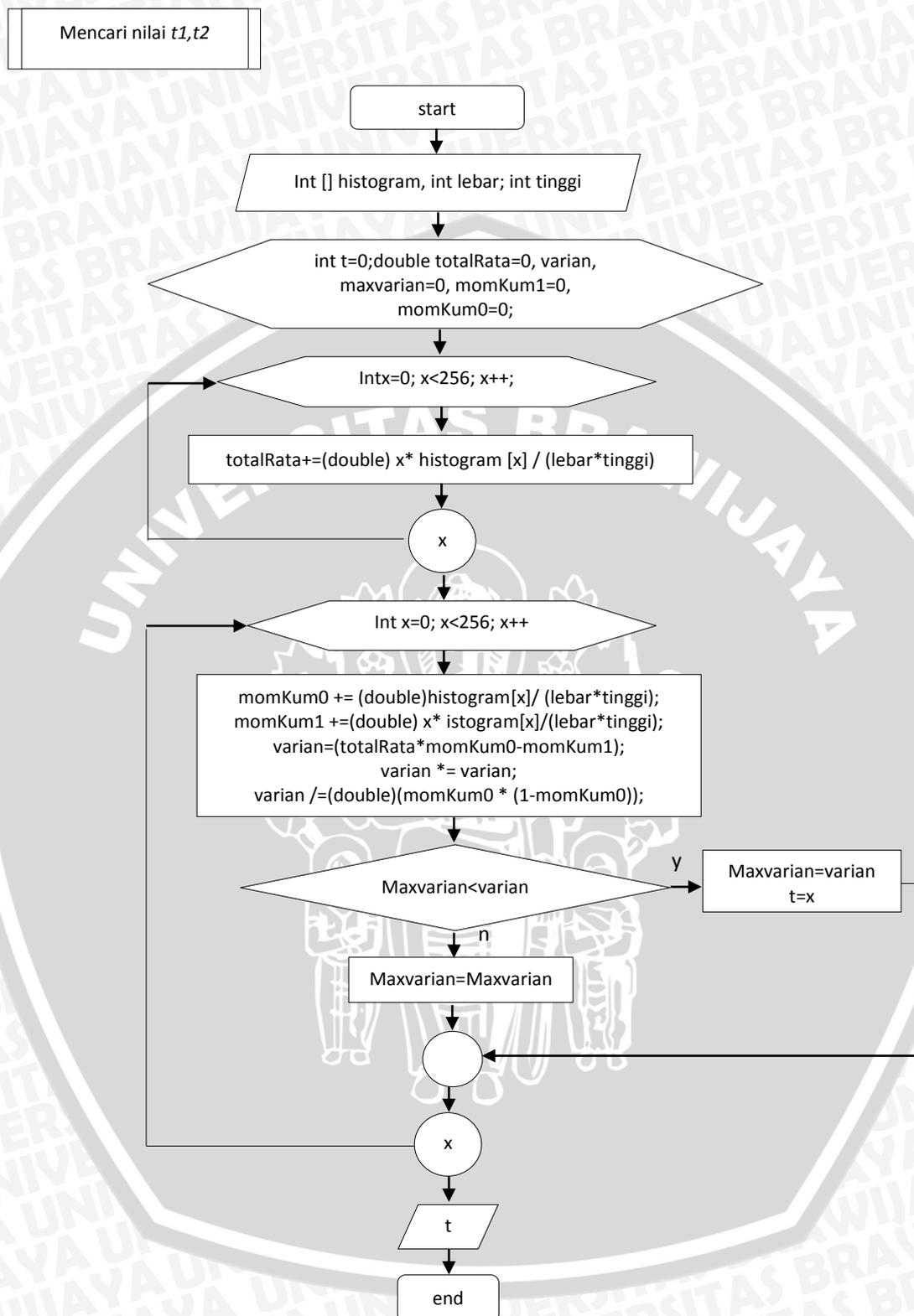
Gambar 4.3 Diagram alir mencari nilai y_1 , y_2 , $histogram_1$, dan $histogram_2$

Gambar 4.3 menjelaskan tentang diagram alir dari mencari nilai y_1 , y_2 , $histogram_1$, dan $histogram_2$. Secara rinci, langkah-langkahnya adalah sebagai berikut:

1. Menginisialisasikan array untuk menyimpan nilai y_1 , y_2 , $histogram_1$, dan $histogram_2$.
2. Melakukan perulangan sesuai dengan jumlah piksel citra.
3. Menghitung nilai y_1 dan y_2 . Nilai y_1 dan y_2 disimpan dalam array pada baris dan kolom yang sesuai dengan citra ke- n saat perulangan.
4. Proses normalisasi pada nilai y_1 dan y_2 . Yaitu dengan mengganti nilai y_1 dan y_2 menjadi 0 jika nilai y_1 dan y_2 kurang dari 0 dan mengganti nilai y_1 dan y_2 menjadi 255 jika lebih besar dari 255.
5. Menambahkan pada $histogram_1$ dan $histogram_2$ sesuai perulangan yang dilakukan.

4.1.1.2 Mencari nilai t_1 , t_2

Pada proses ini dilakukan pencarian nilai *Threshold*. Langkah-langkah yang dilakukan pada proses mencari nilai *Threshold* adalah sebagai berikut :



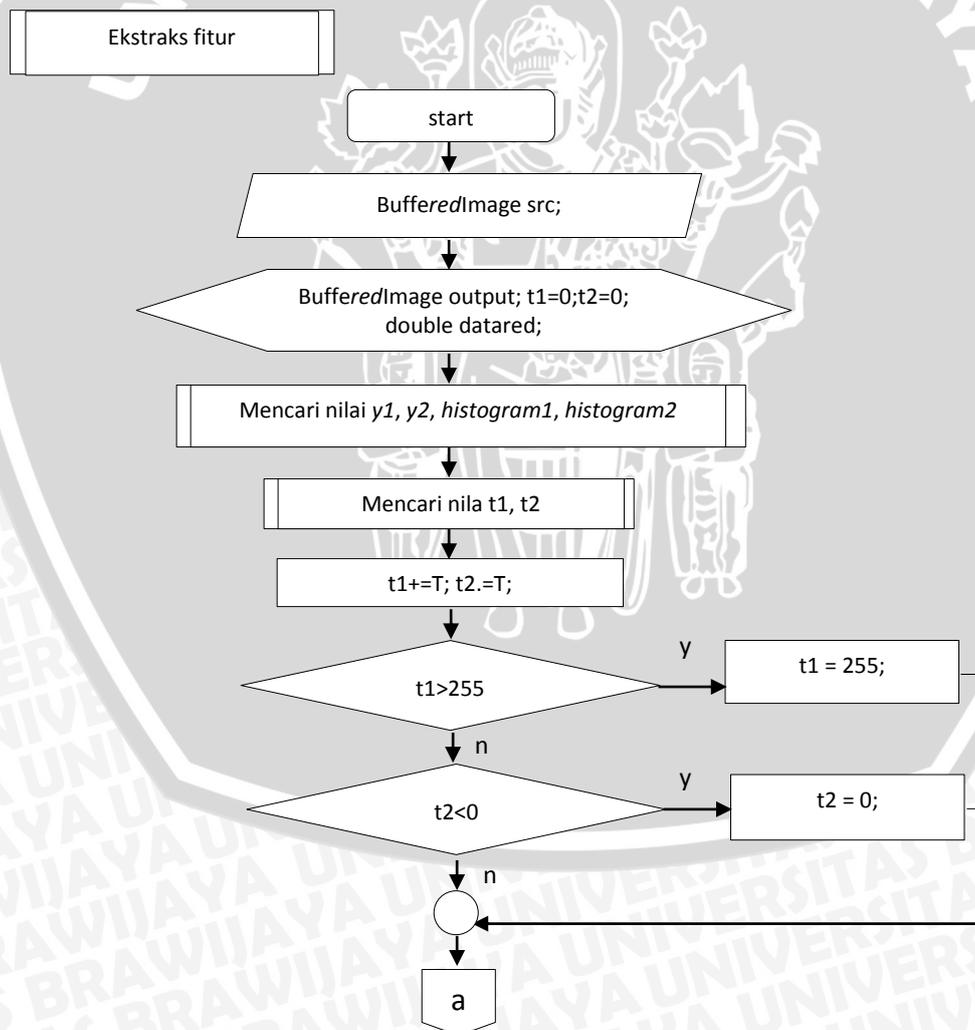
Gambar 4.4 Diagram alir mencari nilai *Threshold*

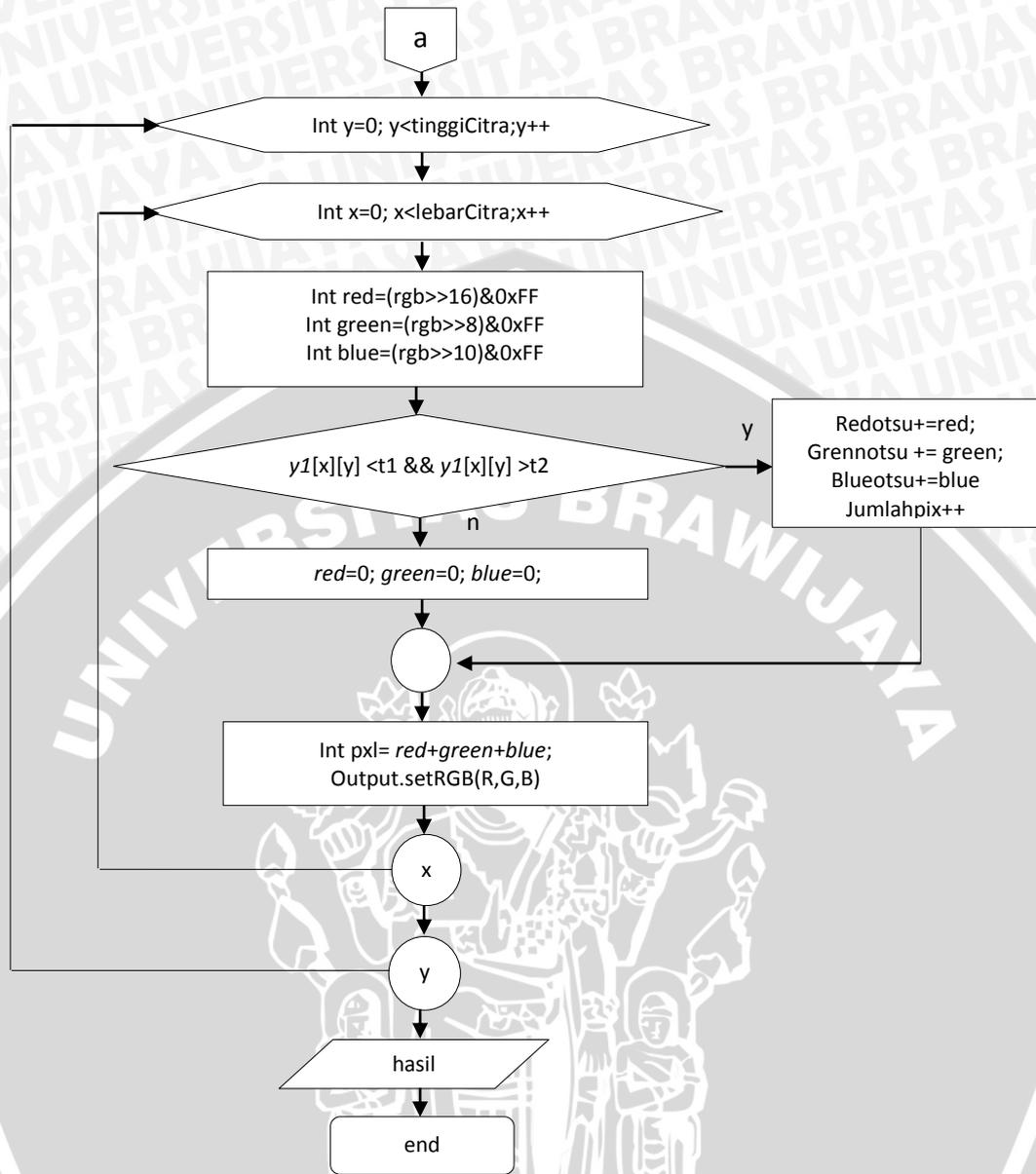
Gambar 4.4 menjelaskan tentang diagram alir dari mencari *Threshold*. Secara rinci, langkah-langkahnya adalah sebagai berikut:

1. Inisialisasi variabel untuk menyimpan nilai Momen Kumulatif ke-0, Momen Kumulatif ke-1, total nilai rata-rata, varian, maksimum varian, dan nilai *Threshold*.
2. Melakukan perulangan sebanyak nilai warna yaitu 255 untuk mencari total nilai rata-rata.
3. Melakukan perulangan sebanyak nilai warna yaitu 255 untuk mencari nilai Momen Kumulatif ke-0 dan nilai Momen Kumulatif ke-1 serta varian.
4. Jika nilai varian lebih besar dari pada nilai maksimal varian maka nilai *Threshold* sama dengan warna ke-*i* jika kondisi tersebut tidak terpenuhi maka perulangan dilanjutkan dan nilai *Threshold* didapatkan saat perulangan selesai.

4.1.1.3 Ekstraksi fitur

Ekstaksi fitur adalah proses untuk mengkonversikan data masukan ke dalam angka agar sistem dapat melakukan identifikasi terhadap citra yang dimasukkan. Citra yang diekstraksi adalah citra yang telah melalui *preprocessing*. Secara garis besar langkah-langkah ekstraksi fitur digambarkan pada Gambar 4.5.





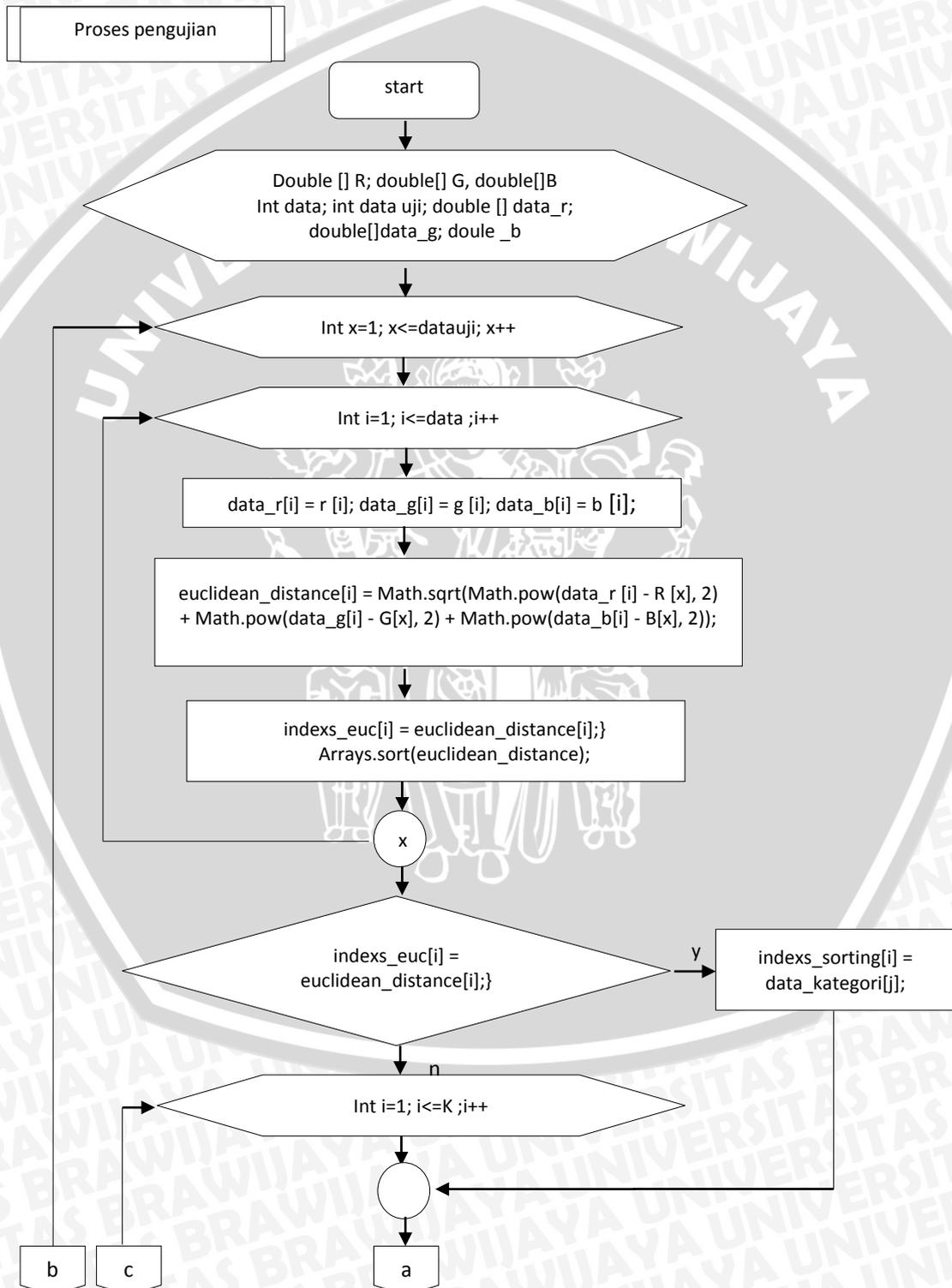
Gambar 4.5 Diagram alir ekstraksi fitur

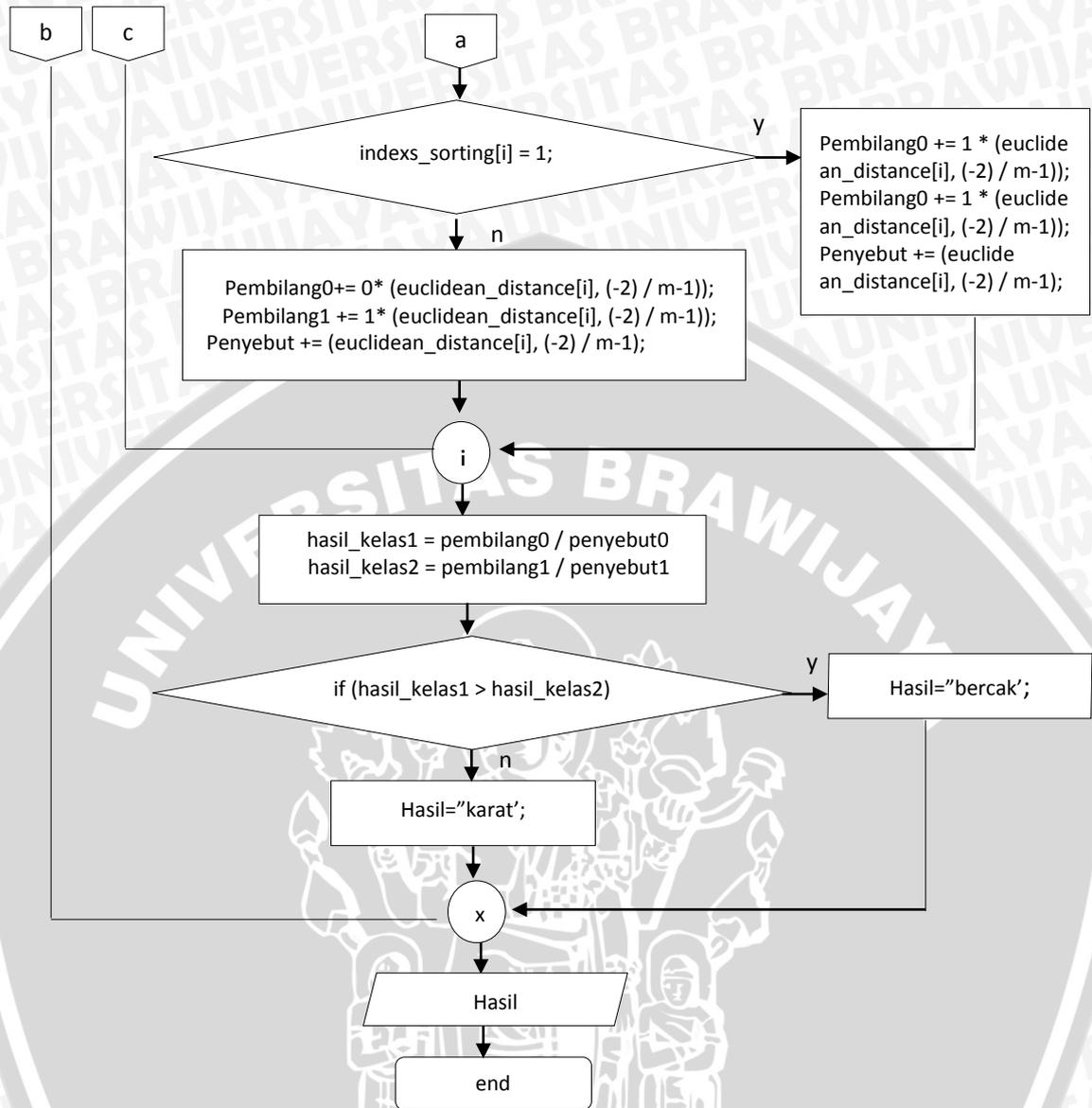
Gambar 4.5 menjelaskan tentang diagram alir dari ekstraksi fitur. Secara rinci, langkah-langkahnya adalah sebagai berikut:

1. Inisialisasi variabel untuk menyimpan nilai *red*, *green*, dan *blue*.
2. Melakukan perulangan sebanyak jumlah piksel citra.
3. Menghitung jumlah piksel citra dan menghitung nilai *red*, nilai *green*, dan nilai *blue*.
4. Setelah nilai *red*, nilai *green*, dan nilai *blue* didapatkan, maka masing-masing nilai tersebut dibagi dengan jumlah piksel citra untuk mendapatkan rata-rata *red*, rata-rata *green*, dan rata-rata *blue*.

4.1.2 Proses pengujian

Pada proses ini, sebelumnya telah ditentukan k kemudian menghitung jarak sampai mendapatkan kelas target dan nilai keanggotaan untuk data testing x . Proses pengujian berfungsi untuk menguji data masukan berupa citra daun untuk diklasifikasikan sesuai penyakit yang dilihat dari fitur. Secara garis besar dapat dilihat di Gambar 4.6





Gambar 4.6 Diagram alir proses pengujian

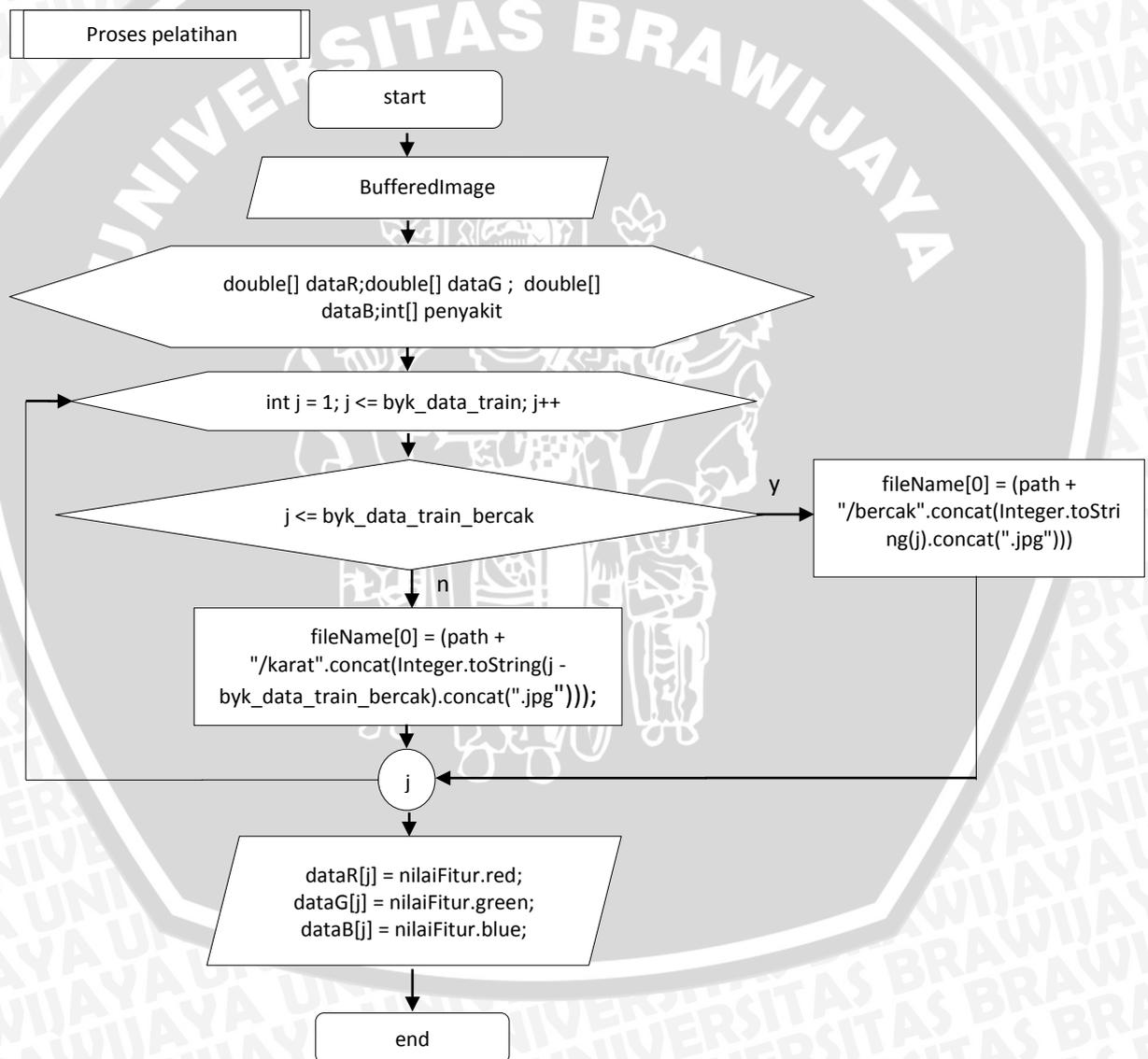
Data yang digunakan adalah hasil data training yang merupakan fitur-fitur sebagai masukan untuk menghitung jarak *Euclidean Distance* yaitu jarak antara data latih dan data uji yang kemudian hasilnya digunakan untuk proses *F-KNN*. Secara rinci langkah-langkahnya adalah sebagai berikut:

1. Mendeklarasikan variabel-variabel yang akan digunakan dalam proses pengujian.
2. Melakukan perulangan sebanyak data uji. Perulangan ini digunakan untuk proses mencari *Euclidean Distance* sebanyak data uji.
3. Melakukan perulangan sebanyak data, yaitu data latih. Perulangan ini berfungsi untuk mencari nilai *Euclidean Distance* dari semua data latih.
4. Menghitung nilai *Euclidean Distance* dengan persamaan yang telah dijelaskan pada Bab 2.



5. Mengurutkan nilai *Euclidean Distance* berdasarkan nilai terkecil.
6. Mengambil data sebanyak k yang telah ditentukan untuk mencari nilai keanggotaan.
7. Menghitung nilai keanggotaan tiap kelas.
8. Apabila nilai keanggotaan kelas 1 lebih besar dari pada kelas 2, maka data uji akan masuk kelas terbesar yaitu kelas 1. Kelas ditentukan berdasarkan nilai keanggotaan terbesar.
9. Apabila nilai keanggotaan kelas 2 lebih besar dari pada kelas 1, maka data uji akan masuk kelas terbesar yaitu kelas 2. Kelas ditentukan berdasarkan nilai keanggotaan terbesar.

4.1.3 Proses pelatihan



Gambar 4.7 Diagram alir proses pelatihan

Gambar 4.7 menggambarkan tentang diagram alir untuk proses pelatihan. Secara rinci dapat dijelaskan sebagai berikut:

1. Masukan berupa citra atau gambar dari penyakit daun kopi. Citra ini yang nantinya akan dihitung fitur *red*, *green*, dan *blue* sebagai hasil dari pelatihan data.
2. Menginisialisasikan variabel-variabel yang akan digunakan dalam proses pelatihan.
3. Melakukan perulangan sebanyak data yang akan dilatih. Perulangan dilakukan untuk mengambil fitur *red*, *green*, dan *blue* dari sejumlah data latih yang tersedia.
4. Membaca file dalam folder yang telah ditentukan, apabila nama file bercak, maka akan masuk ke kelas bercak, dan bila nama file karat maka akan masuk ke kelas karat. Dalam pemberian nama file harus benar dalam folder data latih.
5. Dari masing-masing kelas karat dan bercak untuk semua data latih diambil nilai *red*, *green*, dan *blue* sebagai fitur data latih. Fitur ini yang kemudian akan digunakan sebagai bahan pelatihan dari data yang akan diuji.

4.2 Perhitungan manual

Perhitungan manual adalah langkah-langkah dimana semua data dihitung secara manual. Fungsi dari perhitungan manual adalah sebagai gambaran perancangan perhitungan yang akan diterapkan di sistem. Perhitungan manual dilakukan dengan menggunakan persamaan-persamaan yang telah dijelaskan pada landasan teori. Nilai-nilai yang dihasilkan oleh sistem nantinya harus sama seperti nilai yang dihasilkan pada perhitungan manual. Pada perhitungan manual, langkah-langkah dijelaskan secara detail dengan proses perhitungan yang jelas serta dijelaskan proses pendapatan angka dengan persamaan-persamaan yang digunakan.

Tujuan dari perhitungan manual ini adalah perhitungan yang digunakan dan diterapkan dalam sistem yang nantinya akan dibuat. Semua langkah dalam perhitungan manual harus tercakup dalam sistem sehingga hasil yang didapatkan dari proses perhitungan manual sama dengan hasil yang didapatkan dari perhitungan dengan menggunakan sistem. Sehingga nantinya menghasilkan proses segmentasi citra daun kopi berpenyakit yang bagus dan proses klasifikasi citra daun kopi berpenyakit yang benar dan menghasilkan akurasi yang tinggi.

Perhitungan manual yang dilakukan pada penelitian ini meliputi perhitungan *preprocessing* untuk perbaikan kualitas citra dengan menggunakan metode *Otsu* yang terdiri dari pencarian nilai y_1 dan y_2 , *histogram1* dan *histogram2*, Total Rata-rata dan Momen Kumulatif 1, serta Momen Kumulatif 0, *varian* dan *max varian* untuk mencari nilai *Threshold*, perhitungan ekstraksi fitur, perhitungan metode *FK-NN* dengan mencari nilai *Euclidean Distance* setiap data, dan perhitungan akurasi.

4.2.1 Perhitungan *preprocessing* menggunakan metode *Otsu*

Perhitungan mengambil sampel data citra yang telah dikumpulkan peneliti, yaitu citra daun berukuran 8x8 piksel yang didalamnya terdapat penyakit. Nilai dari setiap piksel citra ditunjukkan pada Tabel 4.1.

Tabel 4.1 Nilai piksel citra kopi berpenyakit

R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
132	159	118	117	145	105	117	141	105	125	140	101	136	148	110	151	164	144	147	168	137	153	169	143
121	145	109	114	146	97	110	137	86	125	133	94	148	156	105	154	161	130	132	150	124	128	155	114
128	147	117	127	150	108	151	162	94	148	111	85	158	97	104	154	145	104	133	152	124	140	154	131
126	146	119	130	144	95	166	138	117	157	72	105	175	74	114	177	104	115	151	160	107	137	147	123
110	136	109	120	131	88	144	78	92	168	83	124	191	106	135	181	99	135	170	155	126	143	159	122
111	143	120	119	146	113	150	98	87	176	82	119	188	100	138	178	90	130	159	132	125	146	172	143
132	154	152	146	165	171	148	148	122	166	116	105	167	106	113	152	118	108	129	139	102	129	153	129
148	157	162	143	154	156	143	155	141	142	159	125	135	159	127	132	153	122	133	148	127	138	152	133

Langkah pertama yang dilakukan adalah mencari tingkat keabuan yaitu y_1 dan y_2 , dan jumlah piksel yang memiliki tingkat keabuan i yaitu ni_1 (*histogram1*) dan ni_2 (*histogram2*). Untuk mencari y_1 dan y_2 , menggunakan Persamaan (2.7) dan Persamaan (2.8) untuk mencari y_1 pada matrik citra Gambar diatas adalah seperti di bawah ini.

Untuk piksel pertama

$$\begin{aligned}
 y_1 &= 2G - R - B \\
 &= 2 \cdot 159 - 132 - 118 \\
 &= 68
 \end{aligned}$$

Untuk piksel ke-1 nilai *Red*=132, nilai *Green*=159, dan Nilai *Blue*=118, maka nilai y_1 adalah= 68.

Untuk keseluruhan perhitungan piksel dari Tabel 4.1 ditunjukkan pada Tabel 4.2.

Tabel 4.2 Hasil perhitungan y_1

68	68	60	54	50	33	52	42
60	81	78	47	59	38	44	68
49	65	79	0	0	32	47	37
47	63	0	0	0	0	62	34
53	54	0	0	0	0	14	53
55	60	0	0	0	0	0	55
24	13	26	0	0	0	47	48
4	9	26	51	56	52	36	33

Tabel 4.2 merupakan hasil dari normalisasi perhitungan y_1 . Untuk y_1 yang bernilai lebih kecil daripada 0, maka dibulatkan menjadi 0, dan untuk y_1 yang bernilai lebih besar dari pada 255 maka dibulatkan menjadi 255. Normalisasi ini juga berlaku untuk perhitungan y_2 .

Untuk menghitung y_2 dengan nilai $RED=36$, $GREEN=10$, dan $BLUE=95$ memiliki nilai $y_2=89$. Dengan formula sebagai berikut.

Untuk piksel pertama

$$\begin{aligned}
 y_2 &= 2R - G - B \\
 &= 2.132 - 159 - 118 \\
 &= -13
 \end{aligned}$$

Untuk keseluruhan perhitungan piksel untuk mencari y_2 dari Tabel 4.1 ditunjukkan pada Tabel 4.3

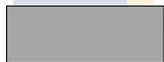
Tabel 4.3 Hasil perhitungan y_2

0	0	0	9	14	0	0	0
0	0	0	23	35	17	0	0
0	0	46	100	115	59	0	0
0	21	77	137	162	135	35	4
0	21	118	129	141	128	59	5
0	0	115	151	138	136	61	0
0	0	26	111	115	78	17	0
0	0	0	0	0	0	0	0

Untuk mencari nilai ni_1 (*histogram1*) yaitu dengan menghitung frekuensi tiap piksel yang digunakan oleh perhitungan y_1 dan untuk mencari ni_2 (*histogram2*) yaitu dengan menghitung jumlah frekuensi piksel yang digunakan di perhitungan y_2 . Untuk hasil perhitungan ni_1 (*histogram1*) ditunjukkan pada Tabel 4.4, dan untuk perhitungan ni_2 (*histogram2*) ditunjukkan pada Tabel 4.5.

Tabel 4.4 Hasil perhitungan nilai *ni1* (histogram1)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
18	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	22
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	
1	0	2	0	0	0	0	0	1	2	1	0	1	1	1	0	0	0	1	0	1	0	0	4	16
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	
1	1	1	1	2	2	2	2	1	0	0	1	3	0	1	1	0	1	0	0	3	0	0	0	23
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	
0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	
																							Total piksel	64



Pixel ke-*i*



Nilai *ni1* (histogram1)

Tabel 4.5 Hasil perhitungan nilai ni_2 (histogram2)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
31	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	2	0	0	0	2	0	1	40
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	
0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	4
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	
0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	3
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	3	0	0	1	0	6
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	0	0	1	0	0	7
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	
																							Total piksel	64

Pixel ke- i

Nilai ni_2 (histogram2)

Tabel 4.4 dan Tabel 4.5 merupakan nilai-nilai *histogram1* dan *histogram2* yang dihasilkan dari perhitungan jumlah nilai y_1 atau y_2 pada tiap piksel citra. Kolom pertama menunjukkan piksel ke- i , dan kolom dibawahnya menunjukkan nilai *histogram1* dan *histogram2*.

Langkah berikutnya adalah mencari nilai Total Rata-rata dengan menggunakan Persamaan, nilai *Momen Kumulatif* ke-0 dengan menggunakan Persamaan (2.4) dan nilai *Momen Kumulatif* ke-1 dengan menggunakan Persamaan (2.5). Nilai varian dengan menggunakan Persamaan (2.6), dan nilai maksimum varian dengan menggunakan Persamaan (2.7) setelah semua diketahui, maka barulah nilai *Threshold1* (t_1) dan *Threshold2* (t_2) dapat dihitung.

Perhitungan pertama adalah menghitung nilai Total Rata-rata, cara perhitungan nilai rata-rata berdasarkan contoh dibawah ini.

$$\begin{aligned} \mu_t &= \sum_{i=1}^L i.P_i \\ &= \sum_{i=1}^{256} 0. \left(\frac{0}{64}\right) \\ &= 0 \end{aligned}$$

Dengan μ_t adalah nilai Total Rata-rata, i adalah nilai piksel pada index tertentu, dan P_i adalah probabilitas nilai piksel pada suatu index. Untuk mencari nilai Total Rata-rata pada piksel ke-1, maka nilai i adalah 1, dan probabilitas dicari dengan membagi nilai piksel ke-1, yaitu 0 dengan jumlah seluruh piksel citra yaitu 64. Dan hasil yang didapatkan adalah 0. Total Rata-rata didapat dengan menjumlahkan sampai L yaitu semua index pada piksel yang berakhir pada piksel ke-255. Untuk contoh perhitungan mencari nilai *Momen Kumulatif* ke-1 adalah sebagai berikut.

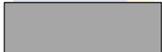
$$\begin{aligned} \mu(k) &= \sum_{i=1}^K i.P_i \\ \mu(1) &= \sum_{i=1}^1 0. \left(\frac{0}{64}\right) \\ &= 0 \end{aligned}$$

Dengan $\mu(k)$ *Momen Kumulatif* ke-1. Untuk mencari *Momen Kumulatif* ke-1 pada piksel ke-1, maka nilai i adalah 1, dan probabilitas dicari dengan membagi nilai piksel ke-1, yaitu 0 dengan jumlah seluruh piksel citra yaitu 64. Dan hasil yang didapatkan adalah 0. Perhitungan *Momen Kumulatif* ke-1 dilakukan sampai K (piksel tertentu).

Untuk keseluruhan perhitungan nilai Total Rata-rata dan Nilai *Momen Kumulatif 1* ditunjukkan pada Tabel 4.7. Sedangkan untuk keseluruhan perhitungan untuk nilai Total Rata-rata dan *Momen Kumulatif* ke-1 ditunjukkan pada Tabel 4.8.

Tabel 4.6 Hasil perhitungan Total Rata-rata dan Momen Kumulatif ke-1 berdasarkan hasil perhitungan *ni1*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,4	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6
24	25	26	27	28	29	29,03	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
1,0	1,0	1,8	1,8	1,8	1,8	1,8	1,8	2,3	3,3	3,9	3,9	4,4	5,0	5,6	5,6	5,6	5,6	6,3	6,3	7,0	7,0	7,0	9,9
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
10,6	11,4	12,2	13,0	14,6	16,3	18,0	19,7	20,5	20,5	20,5	21,5	24,3	24,3	25,3	26,2	26,2	27,3	27,3	27,3	30,4	30,4	30,4	30,4
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
30,4	30,4	30,4	30,4	30,4	30,4	31,7	32,9	32,9	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167
34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216
34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2
217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255									
34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2	34,2									



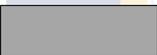
Pixel ke-*i*



rata-rata dan Momen Kumulatif ke-1

Tabel 4.7 Hasil perhitungan Total Rata-rata dan Momen Kumulatif ke-1 berdasarkan hasil perhitungan ni_2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,1	0,1	0,3	0,3	0,3	0,3	0,3	0,5	0,5	0,5	1,0	1,0	1,0	1,0	1,7	1,7	2,0
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
2,0	2,0	2,5	2,5	2,5	2,5	2,5	2,5	2,5	2,5	2,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5	4,3	4,3
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	6,1	6,1	7,1	7,1	7,1	7,1	7,1	7,1	7,1	7,1	7,1	7,1	7,1
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
7,1	7,1	7,1	7,1	7,1	8,3	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5	9,5
96,0	97,0	98,0	99,0	100,0	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
9,5	9,5	9,5	9,5	11,0	11,0	11,0	11,0	11,0	11,0	11,0	11,0	11,0	11,0	11,0	12,8	12,8	12,8	12,8	18,2	18,2	18,2	20,0	20,0
120,0	121,0	122,0	123,0	124,0	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
20,0	20,0	20,0	20,0	20,0	20,0	20,0	20,0	22,0	24,0	24,0	24,0	24,0	24,0	24,0	26,1	28,3	30,4	32,6	32,6	32,6	34,8	34,8	34,8
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167
34,8	34,8	34,8	34,8	34,8	34,8	34,8	37,1	37,1	37,1	37,1	37,1	37,1	37,1	37,1	37,1	37,1	37,1	39,7	39,7	39,7	39,7	39,7	39,7
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216
39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7
217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255									
39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7									



Pixel ke- i



rata-rata dan Momen Kumulatif ke-1

Tabel 4.6 dan Tabel 4.7 merupakan hasil perhitungan Total Rata-rata dan Momen Kumulatif ke-1 berdasarkan hasil perhitungan $ni1$, dan hasil perhitungan Total Rata-rata dan Momen Kumulatif ke-1 berdasarkan hasil perhitungan $ni2$ yang didapat dari Tabel 4.4 dan Tabel 4.6. Dari hasil perhitungan nilai Total Rata-rata dan nilai Momen Kumulatif ke-1 yang ditunjukkan pada Tabel 4.7 dan 4.8, maka didapat nilai Total Rata-rata terhadap hasil perhitungan $ni1$ sebesar 34,2 dan nilai Total Rata-rata terhadap hasil perhitungan $ni2$ sebesar 39,7.

Langkah berikutnya adalah menghitung nilai Momen Kumulatif ke-0 berdasarkan hasil perhitungan $ni1$ pada Tabel 4.5. Sebagai contoh perhitungan ditunjukkan dibawah ini.

$$\begin{aligned}\omega(k) &= \sum_{i=1}^k P_i \\ \omega(1) &= \sum_{i=1}^1 \left(\frac{18}{64}\right) \\ &= 0,3\end{aligned}$$

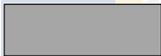
Dimana $\omega(k)$ merupakan nilai Momen Kumulatif ke-0 yang dicari dengan menjumlahkan probabilitas tiap piksel. Pada Tabel 4.4 piksel ke-1 memiliki nilai 18 dengan jumlah seluruh piksel adalah 64. Maka untuk mencari probabilitas adalah 18 dibagi dengan 64 yaitu 0,3. Untuk piksel selanjutnya dicari dengan menjumlahkan probabilitas piksel sebelumnya dengan probabilitas pada piksel itu sendiri. Contohnya pada piksel ke-1 merupakan hasil penjumlahan probabilitas piksel ke-1 yaitu 0,3 dengan probabilitas piksel ke-2 itu sendiri yaitu 0 dan menghasilkan nilai Momen Kumulatif ke-0 pada piksel 2 adalah 0,3. Sedangkan untuk menghitung nilai Momen Kumulatif ke-0 berdasarkan hasil perhitungsn $ni2$ pada Tabel 4.6. Sebagai contoh perhitungan ditunjukkan dibawah ini.

$$\begin{aligned}\omega(k) &= \sum_{i=1}^k P_i \\ \omega(1) &= \sum_{i=1}^1 \left(\frac{31}{64}\right) \\ &= 0,5\end{aligned}$$

Pada Tabel 4.6 piksel pertama atau piksel ke-1 memiliki nilai 31 dengan jumlah seluruh piksel adalah 64. Maka untuk mencari probabilitas adalah 31 dibagi dengan 64 yaitu 0,5. Untuk nilai Momen Kumulatif ke-0 pada piksel kedua yaitu dengan menjumlahkan probabilitas piksel sebelumnya dengan probabilitas piksel kedua yaitu $0,5+0=0,5$, maka nilai Momen Kumulatif ke-0 pada piksel kedua yaitu 0,5. Untuk keseluruhan perhitungan nilai kumulatif ke-0 berdasarkan hasil perhitungsn $ni1$ pada Tabel 4.5 ditunjukkan pada Tabel 4.9. Dan Untuk keseluruhan perhitungan nilai kumulatif ke-0 berdasarkan hasil perhitungsn $ni2$ pada Tabel 4.6 ditunjukkan pada Tabel 4.10.

Tabel 4.8 Hasil perhitungan Momen Kumulatif ke-0 berdasarkan hasil perhitungan $ni1$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,6
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
0,6	0,6	0,6	0,7	0,7	0,7	0,8	0,8	0,8	0,8	0,8	0,8	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	1,0	1,0	1,0	1,0
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255									
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0									



Pixel ke- i



Momen Kumulatif ke-0

Tabel 4.9 Hasil perhitungan Momen Kumulatif ke-0 berdasarkan hasil perhitungan ni^2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,6	0,6	0,6	0,6	0,6	0,6	0,6
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
0,7	0,7	0,7	0,7	0,7	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,9	0,9
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	1,0	1,0	1,0	1,0	1,0	1,0
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255									
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0									

Pixel ke- i

Momen kumulatif ke-0

Tabel 4.8 dan Tabel 4.9 hasil perhitungan Momen Kumulatif ke-0 berdasarkan hasil perhitungan *ni1* dan hasil perhitungan Momen Kumulatif ke-0 berdasarkan hasil perhitungan *ni2*.

Setelah mendapatkan nilai Total Rata-rata, nilai Momen Kumulatif ke-0, dan nilai Momen Kumulatif ke-1, maka perhitungan untuk mencari varian, maksimum varian, dan nilai *Threshold* dapat dilakukan. Berikut contoh perhitungan varian saat warna ke-0 dari hasil nilai Total Rata-rata, nilai Momen Kumulatif ke-1, pada Tabel 4.7 dan perhitungan nilai Momen Kumulatif ke-0 pada Tabel 4.9.

$$\sigma_B^2(k) = \frac{[\mu(t) \cdot \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]}$$

Jika $\omega(k)[1 - \omega(k)] = 0$, maka hasil akhir akan 0. Dari contoh diatas $\sigma_B^2(k)$ adalah nilai ambang yang dicari dengan mengkuadratkan hasil perkalian antara nilai rata-rata dan Momen Kumulatif ke-0 kemudian dikurangi dengan Momen Kumulatif ke-1, kemudian dibagi dengan hasil perkalian antara Kumulatif ke-0 dan 1 dikurangi Kumulatif ke-0. Maka setelah dimasukkan terhadap bilangan yang telah diketahui pada Tabel-Tabel sebelumnya akan menjadi persamaan berikut untuk nilai warna pertama atau piksel ke-0.

$$\begin{aligned} \sigma_B^2(k) &= \frac{[34,2 \cdot 0,3 - 0]^2}{0,3 \cdot [1 - 0,3]} \\ &= 235,84 \end{aligned}$$

Berikut contoh perhitungan varian saat warna ke-1 dari hasil nilai total rata-rata, nilai Momen Kumulatif ke-1, pada Tabel 4.7 dan perhitungan nilai Momen Kumulatif ke-0 pada Tabel 4.9.

$$\begin{aligned} \sigma_B^2(k) &= \frac{[39,77 \cdot 0,5 - 0]^2}{0,5 \cdot [1 - 0,5]} \\ &= 1477 \end{aligned}$$

Dari contoh perhitungan nilai varian tersebut yaitu saat warna ke-0 dan saat warna ke-0 untuk nilai Total Rata-rata, nilai Momen Kumulatif ke-0 menghasilkan nilai varian 235,84 dan 1477 maka nilai maksimum varian adalah 558 dan nilai *Threshold* adalah 26. Untuk keseluruhan perhitungan nilai varian, maksimum varian, dan *Threshold (t1)* dari hasil nilai Total Rata-rata, nilai Momen Kumulatif ke-1, pada Tabel 4.7 dan perhitungan nilai Momen Kumulatif ke-0 pada Tabel 4.9 ditunjukkan pada Tabel 4.11.

Tabel 4.10 Hasil perhitungan varian, maksimum varian, dan *Threshold1(t1)*

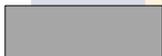
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
236	236	236	236	241	241	241	241	241	241	241	241	241	239	236	236	236	236	236	236	236	548	548	548
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
552	552	558	558	558	558	558	558	554	547	543	543	538	532	526	526	526	526	515	515	503	503	503	448
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
435	422	408	394	366	339	313	288	275	275	275	259	213	213	197	180	180	161	161	161	100	100	100	100
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
100	100	100	100	100	100	68	35	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									

Pixel ke-*i*

Nilai varian, maksimum varian, dan *Threshold1(t1)*

Tabel 4.11 Hasil perhitungan varian, maksimum varian, dan *Threshold2(t2)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1477	1477	1477	1477	1563	1651	1651	1651	1651	1735	1735	1735	1735	1735	1811	1811	1811	1966	1966	1966	1966	2123	2123	2206
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
2206	2206	2288	2288	2288	2288	2288	2288	2288	2288	2288	2420	2420	2420	2420	2420	2420	2420	2420	2420	2420	2420	2462	2462
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
2462	2462	2462	2462	2462	2462	2462	2462	2462	2462	2462	2481	2481	2495	2495	2495	2495	2495	2495	2495	2495	2495	2495	2495
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
176	176	176	176	176	154	133	133	133	133	133	133	133	133	133	133	133	133	133	133	133	133	133	133
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
133	133	133	133	111	111	111	111	111	111	111	111	111	111	111	90	90	90	90	43	43	43	32	32
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
32	32	32	32	32	32	32	32	23	16	16	16	16	16	16	10	6	3	1	1	1	0	0	0
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	236	0	0	0	0
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									



Pixel ke-*i*



Nilai varian. maksimum varian. dan *Threshold2(t2)*

$t1$ dan $t2$ didapat dengan melihat nilai maksimum varian dan piksel ke- i . Nilai maksimum pada Tabel 4.10 akan menjadi $t1$ yaitu piksel ke-26, dan nilai maksimum pada Tabel 4.11 akan menjadi $t2$ yaitu piksel ke-61. Setelah nilai $t1$ dan $t2$ didapatkan, maka nilai $t1$ ditambahkan dan dikurangkan dengan nilai peubah *Threshold* yang pada perhitungan ini menggunakan nilai 10. Hasil $t1$ yaitu 26 akan ditambahkan dengan 10 dan hasil $t1$ yaitu 61 akan dikurangkan dengan 10 sehingga nilai $t1$ 36 dan nilai $t2$ yaitu 51 kemudian akan dibandingkan dengan nilai $y1$ pada Tabel 4.1 dan nilai $y2$ pada Tabel 4.2 untuk mengubah sebagian warna citra daun pada Tabel 4.1. Nilai *Red*, *Green*, *Blue* piksel pertama akan bernilai tetap jika $t1$ lebih besar dari $y1$, dan nilai $t2$ lebih kecil daripada $y2$ pada piksel pertama. jika kondisi tersebut tidak terpenuhi maka nilai *Red*, *Green*, *Blue* akan diubah menjadi 0. Hasil akhir pengolahan citra ditunjukkan pada Gambar 3.11.

Tabel 4.12 Hasil akhir pengolahan citra

R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	148	111	85	158	97	104	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	166	138	117	157	72	105	175	74	114	177	104	115	0	0	0	0	0	0
0	0	0	0	0	0	144	78	92	168	83	124	175	106	135	181	99	135	0	0	0	0	0	0
0	0	0	0	0	0	150	98	87	176	82	119	188	100	138	178	90	130	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	166	116	105	188	106	113	152	118	108	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.2.2 Perhitungan ekstraksi fitur

Ekstraksi fitur adalah proses dimana setiap citra dari daun kopi akan diambil fitur nilai warna *red*, *green*, *blue* nya. Karena penyakit yang akan diklasifikasikan dapat diamati dari perbedaan warna, maka sistem akan mengambil ciri setiap daun dari rata-rata warna merah, rata-rata warna hijau, rata-rata warna biru.

Perhitungan ekstraksi fitur dilakukan setelah hasil *preprocessing* selesai. Perhitungan ekstraksi fitur bertujuan untuk mendapatkan rata-rata nilai *red*, rata-rata *green*, rata-rata *blue* dari citra daun yang nantinya akan digunakan pada proses klasifikasi citra dengan menggunakan metode *Fuzzy K-Nearest Neighbor*. Dimisalkan citra daun yang telah melalui pemrosesan awal menggunakan metode *Otsu* sebesar 8×8 piksel seperti yang ditunjukkan di Tabel 4.13.

Tabel 4.13 Nilai piksel citra *Otsu*

R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	148	111	85	158	97	104	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	166	138	117	157	72	105	175	74	114	177	104	115	0	0	0	0	0	0
0	0	0	0	0	0	144	78	92	168	83	124	175	106	135	181	99	135	0	0	0	0	0	0
0	0	0	0	0	0	150	98	87	176	82	119	188	100	138	178	90	130	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	166	116	105	188	106	113	152	118	108	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Langkah selanjutnya adalah mendapatkan nilai rata-rata *red*, rata-rata *green*, rata-rata *blue* yang akan digunakan sebagai fitur yang akan dihitung menggunakan metode *FK-NN*. Caranya yaitu menjumlahkan semua nilai piksel *red* kemudian dibagi jumlah piksel untuk mencari nilai rata-rata *red*, dan menjumlahkan semua nilai fitur *green* kemudian dibagi dengan jumlah keseluruhan piksel untuk mencari rata-rata *green*, dan menjumlahkan semua nilai *blue* kemudian dibagikan dengan jumlah keseluruhan piksel citra untuk mencari nilai rata-rata *blue*.

$$\text{Rata-rata } R = \frac{(166+144+150+148+157+168+176+166+158+175 + 175+188+188+177+181+178+152)}{17}$$

$$= 166$$

$$\text{Rata-rata } G = \frac{(1138+78+111+72+83+82+116+97+74 + 106+100+106+104+99+90+118)}{17}$$

$$= 105$$

$$\text{Rata-rata } B = \frac{(117+92+87+85+105+124+119+105+104 + 114+135+138+113+115+130+105)}{17}$$

$$= 114$$

4.2.3 Perhitungan FK-NN

Perhitungan *FK-NN* dalam penelitian ini ditentukan $k=5$. Perhitungan *FK-NN* dalam penelitian ini dapat dilakukan jika nilai rata-rata RGB dari data latih dan data uji sudah didapatkan. Dimisalkan terdapat data latih sebanyak 30 data yang memiliki nilai RGB secara terpisah seperti yang ditunjukkan pada Tabel 4.14. dan data uji seperti yang ditunjukkan pada Tabel 4.15.

Tabel 4.14 Nilai data latih secara terpisah

Data Latih	R	G	B	KELAS
1	179	130	60	1
2	210	175	140	1
3	201	141	105	1
4	165	119	38	1
.....
.....
27	161	107	90	2
28	110	69	55	2
29	129	84	74	2
30	210	166	152	2

Tabel 4.14 menunjukkan data latih yang merupakan hasil dari ekstraksi fitur citra melalui proses *preprocessing* dengan fitur *RED (R)*, *GREEN (G)*, *BLUE (B)*.

Tabel 4.15 Nilai data uji

1	166	105	114	?
2	169	154	86	?

Tabel 4.15 merupakan Tabel nilai data yang akan diuji untuk dicari kelas targetnya. Langkah berikutnya adalah menghitung jarak antara data uji terhadap data latih menggunakan persamaan (2.12).

Contoh perhitungan jarak pada data latih pertama adalah sebagai berikut.

$$\text{Jarak} = \sqrt{(166 - 179)^2 + (105 - 130)^2 + (114 - 60)^2}$$

$$= 60,90$$



Untuk keseluruhan hasil perhitungan jarak untuk data uji ke-1 terhadap data latih ditunjukkan pada Tabel 4.16, dan untuk data uji ke-2 terhadap data latih ditunjukkan pada Tabel 4.17.

Tabel 4.16 Hasil perhitungan jarak data uji ke-1

Data Latih	jarak	rank min distance	kelas
1	60,90977	19	1
2	86,67179	24	1
3	51,0098	18	1
4	77,28519	20	1
5	89,82761	16	1
....
27	24,59675	7	2
28	88,95504	28	2
29	58,39521	26	2
30	84,26743	25	2

Tabel 4.16 merupakan nilai jarak data uji ke-1 terhadap data latih. Langkah berikutnya adalah mengurutkan jarak dari nilai terkecil ke nilai terbesar.

Tabel 4.17 Hasil perhitungan jarak data uji ke-2

Data Latih	jarak	rank min distance	kelas
1	36,76955	12	1
2	70,97887	14	1
3	39,42081	18	1
4	59,5399	21	1
5	52,81098	1	1
....
27	47,84349	9	2
28	108,0139	26	2
29	81,51074	28	2
30	78,61934	25	2

Tabel 4.17 merupakan nilai jarak data uji ke-2 terhadap data latih. Langkah berikutnya yaitu mengurutkan jarak dari nilai terkecil ke nilai terbesar. Hasil pengurutan nilai jarak data uji ke-1 ditunjukkan pada Tabel 4.18 dan hasil pengurutan nilai jarak data uji ke-2 ditunjukkan pada Tabel 4.19.

Tabel 4.18 Hasil pengurutan perhitungan jarak data uji ke-1

jarak	rank min distance	kelas
27,38613	1	2
34,42383	2	2
41,82105	3	2
45,71652	4	2
49,49747	5	2
.....
108,9633	27	1
111,1441	28	2
111,4675	29	2
113,1636	30	2

Tabel 4.18 merupakan hasil pengurutan jarak berdasarkan nilai terkecil. Pengurutan ini bertujuan untuk proses selanjutnya yaitu mencari nilai keanggotaan dari kelas uji. Kemudian tentukan K yang akan dipakai, yaitu 15.

Tabel 4.19 Hasil pengurutan perhitungan jarak data uji ke-2

jarak	rank min distance	kelas
28,14249	1	1
33,34666	2	1
35,73514	3	2
36,56501	4	2
36,76955	5	1
.....
83,74963	27	1
102,0735	28	2
108,0139	29	2
111,472	30	2

Tabel 4.19 merupakan hasil pengurutan perhitungan jarak data uji ke-2 terhadap data latih. Langkah Berikutnya adalah menghitung nilai keanggotaan data pada masing-masing kelas. Perhitungan nilai keanggotaan terhadap kelas 1 ditunjukkan pada Tabel 4.20

Tabel 4.20 Nilai keanggotaan kelas 1 data uji ke-1

Jarak	Kelas	u (kelas1)	Nilai Keanggotaan
27,38612788	2	0	0,16
34,42382896	2	0	
41,82104733	2	0	
45,7165178	2	1	



49,49747468	2	0
53,86093204	2	0
54,2678542	2	0
54,85435261	2	0
62,93647591	2	0
63,5767253	1	1
68,8839604	1	1
69,7782201	1	1
71,79136438	1	1
74,47147105	1	1
75,63729239	2	0

Pada Tabel 4.20 diatas nilai u adalah nilai keanggotaan data kedalam kelas, nilainya 1 jika jarak milik kelas 1 dan 0 jika jarak milik kelas 2. Selanjutnya menghitung nilai keanggotaan data terhadap kelas 1 .

$$u(x, c_1) = \frac{0 \cdot 27,38612^{\frac{-2}{(2-1)}} + 0 \cdot 34,423827^{\frac{-2}{(2-1)}} + 0 \cdot 41,8210^{\frac{-2}{(2-1)}} + \dots + 0 \cdot 75,63729^{\frac{-2}{(2-1)}}}{27,38612^{\frac{-2}{(2-1)}} + 34,423827^{\frac{-2}{(2-1)}} + 41,8210^{\frac{-2}{(2-1)}} + \dots + 75,63729^{\frac{-2}{(2-1)}}} = 0,16$$

Nilai keanggotaan data uji ke-1 terhadap kelas 2 ditunjukkan pada Tabel 4.21.

Tabel 4.21 Nilai keanggotaan kelas 2 data uji ke-1

Jarak	Kelas	$u(\text{kelas1})$	Nilai Keanggotaan
27,38612788	2	0	0,83
34,42382896	2	0	
41,82104733	2	0	
45,7165178	2	1	
49,49747468	2	0	
53,86093204	2	0	
54,2678542	2	0	
54,85435261	2	0	
62,93647591	2	0	
63,5767253	1	1	
68,8839604	1	1	
69,7782201	1	1	
71,79136438	1	1	
74,47147105	1	1	
75,63729239	2	0	



Pada Tabel 4.21 diatas nilai u adalah nilai keanggotaan data kedalam kelas, nilainya 1 jika jarak milik kelas 2 dan 0 jika jarak milik kelas 1. Selanjutnya menghitung nilai keanggotaan data terhadap kelas 2.

$$u(x, c_2) = \frac{11 \cdot 27,38612^{\frac{-2}{(2-1)}} + 1 \cdot 34,423827^{\frac{-2}{(2-1)}} + 0 \cdot 41,8210^{\frac{-2}{(2-1)}} + \dots + 1 \cdot 75,63729^{\frac{-2}{(2-1)}}}{27,38612^{\frac{-2}{(2-1)}} + 34,423827^{\frac{-2}{(2-1)}} + 41,8210^{\frac{-2}{(2-1)}} + \dots + 75,63729^{\frac{-2}{(2-1)}}} = 0,83$$

Tabel 4.22 Nilai keanggotaan kelas 1 data uji ke-2

Jarak	Kelas	u (kelas1)	Nilai Keanggotaan
28.14249	1	1	0,68
33.34666	1	1	
35.73514	2	0	
36.56501	2	0	
36.76955	1	1	
36.79674	1	1	
39.42081	1	1	
41.4367	1	1	
42.21374	1	1	
44.51966	2	0	
45.18849	2	0	
47.84349	2	0	
47.86439	1	1	
52.46904	1	1	
52.81098	1	1	

Tabel 4.22 merupakan nilai keanggotaan kelas 1 untuk data uji ke-2, persamaan yang digunakan sama dengan persamaan untuk menghitung nilai keanggotaan kelas 1 data uji ke 1 pada Tabel 4.20 yaitu

$$u(x, c_1) = \frac{1 \cdot 28,1424^{\frac{-2}{(2-1)}} + 1 \cdot 33,346 + 0 \cdot 35,375^{\frac{-2}{(2-1)}} + \dots + 1 \cdot 52,8109^{\frac{-2}{(2-1)}}}{1 \cdot 28,1424^{\frac{-2}{(2-1)}} + 1 \cdot 33,346 + 0 \cdot 35,375^{\frac{-2}{(2-1)}} + \dots + 1 \cdot 52,8109^{\frac{-2}{(2-1)}}} = 0,68$$

Tabel 4.23 Nilai keanggotaan kelas 2 data uji ke-2

Jarak	Kelas	u (kelas1)	Nilai Keanggotaan
28.14249	1	0	0,31
33.34666	1	0	
35.73514	2	2	
36.56501	2	2	
36.76955	1	0	
36.79674	1	0	
39.42081	1	0	
41.4367	1	0	
42.21374	1	0	
44.51966	2	2	
45.18849	2	2	
47.84349	2	2	
47.86439	1	0	
52.46904	1	0	
52.81098	1	0	

Pada Tabel 4.23 diatas nilai u adalah nilai keanggotaan data kedalam kelas, nilainya 1 jika jarak milik kelas 2 dan 0 jika jarak milik kelas 1. Selanjutnya menghitung nilai keanggotaan data terhadap kelas 2.

$$\begin{aligned}
 u(x, c_2) &= \frac{1 \cdot 28,1424 \frac{-2}{(2-1)} + 1 \cdot 33,346 + 0 \cdot 35,375 \frac{-2}{(2-1)} + \dots}{1 \cdot 28,1424 \frac{-2}{(2-1)} + 1 \cdot 33,346 + 0 \cdot 35,375 \frac{-2}{(2-1)} + \dots} \\
 &= \frac{1 \cdot 52,8109 \frac{-2}{(2-1)}}{1 \cdot 52,8109 \frac{-2}{(2-1)}} \\
 &= 0,31
 \end{aligned}$$

Langkah selanjutnya adalah menentukan kelas berdasarkan nilai keanggotaan terbesar. Dari hasil perhitungan metode *Fuzzy K-NN* yang terdapat pada Tabel 4.20 dan 4.21 untuk data uji ke-1 diperoleh nilai keanggotaan pada kelas 1 sebesar 0.18 dan nilai keanggotaan pada kelas 2 sebesar 0.83. Nilai keanggotaan terbesar dipilih sebagai kelas target yaitu kelas 2 atau penyakit karat daun, sehingga data uji masuk kedalam kelas target.

Untuk data uji ke-2, hasil perhitungan metode *Fuzzy K-NN* yang terdapat pada Tabel 4.22 dan 4.23 untuk data uji ke-1 diperoleh nilai keanggotaan pada kelas 1 sebesar 0.68 dan nilai keanggotaan pada kelas 2 sebesar 0.31. Nilai keanggotaan terbesar dipilih sebagai kelas target yaitu kelas 1 atau penyakit bercak daun, sehingga data uji masuk kedalam kelas target.



4.2.3 Perhitungan akurasi

Perhitungan akurasi adalah proses untuk mengetahui keberhasilan sistem. Tujuan dari perhitungan akurasi adalah untuk mengetahui presentase jumlah keberhasilan sistem dalam mengklasifikasikan penyakit pada citra daun tanaman kopi. Untuk pengujian 2 citra kopi maka hasil akurasinya ditunjukkan di bawah ini

$$Akurasi = \frac{\text{jumlah data benar}}{\text{Jumlah Seluruh data}} \times 100\%$$

$$Akurasi = \frac{2}{2} \times 100\%$$

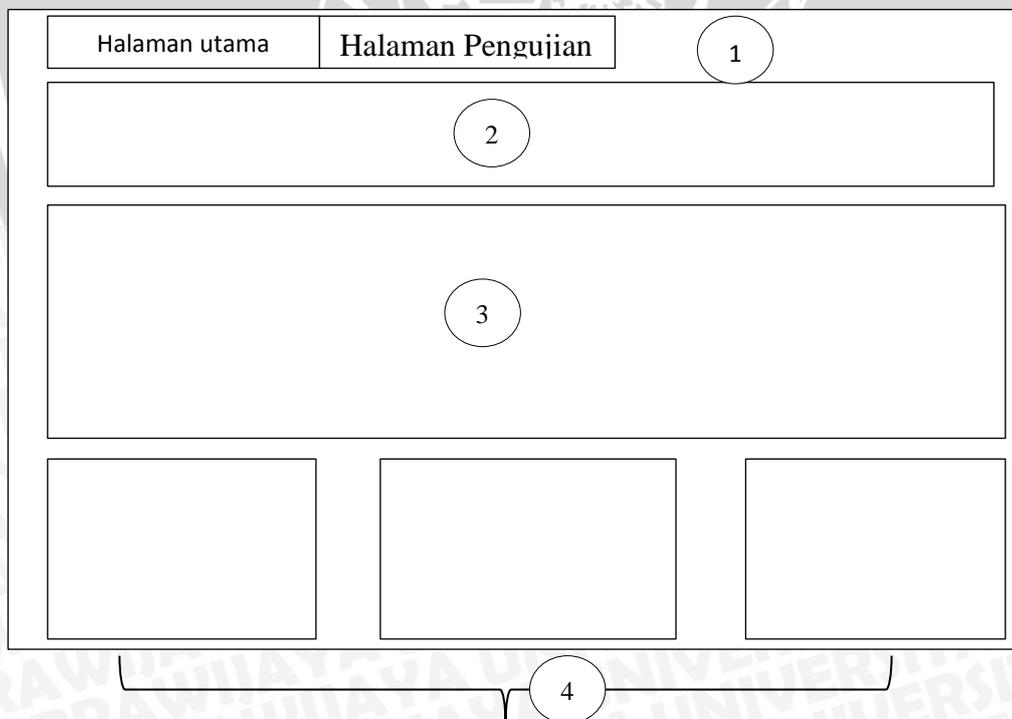
Maka hasil akurasi dari proses manualisasi menggunakan metode *F-KNN* ini adalah 100%.

4.3 Perancangan antarmuka

Perancangan antarmuka merupakan pembuatan desain rancangan user interface yang digunakan user untuk berkomunikasi dengan aplikasi klasifikasi penyakit kopi berdasarkan citra daun. Perancangan antarmuka terdiri dari rancangan antarmuka utama, rancangan antarmuka pelatihan data, rancangan antarmuka pengujian data.

4.3.1 Rancangan antarmuka utama

Antarmuka utama merupakan tampilan antarmuka bagi pengguna ketika baru menjalankan program.



Gambar 4.8 Rancangan antarmuka halaman utama

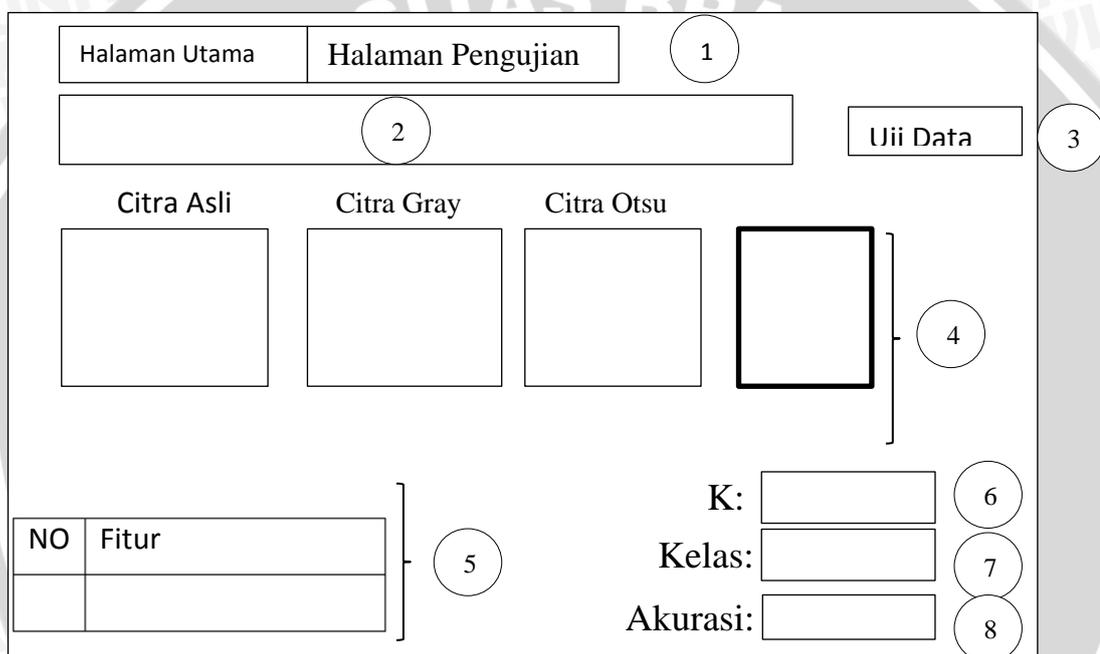


Gambar 4.8 Tampilan utama berisi judul dari aplikasi dan pengantar. Keterangan Gambar:

1. Tombol menu yang terdiri dari halaman utama dan halaman pengujian.
2. Judul Aplikasi
3. Penjelasan Penyakit Kopi
4. Nama Penulis

4.3.2 Rancangan antarmuka pelatihan data

Antarmuka pelatihan data merupakan antarmuka dimana user melakukan pembelajaran sistem terhadap data latih. Tujuannya adalah sistem dapat mengidentifikasi data uji sesuai dengan pelatihan yang dilakukan. Adapun antarmuka pelatihan data ditunjukkan pada Gambar 4.9



Gambar 4.9 Rancangan antarmuka pelatihan data

Gambar 4.9 merupakan rancangan antarmuka pelatihan data. Keterangan Gambar:

1. Tombol menu untuk berpindah ke halaman, terdapat dua menu yang tersedia yaitu halaman utama dan halaman pengujian.
2. Judul aplikasi
3. Tombol untuk memulai pelatihan data
4. Output citra yang terdiri dari gambar asli citra, citra *gray*, dan citra hasil *Otsu*.
5. Data hasil ekstraksi fitur yang terdiri dari rata-rata *red*, rata-rata *green*, dan rata-rata *blue*.
6. Nilai *K* yang sebelumnya telah ditentukan
7. Hasil kelas target
8. Hasil akurasi

BAB 5 IMPLEMENTASI

Bab implementasi berisi pembahasan mengenai implementasi dari perancangan yang telah dilakukan pada bab sebelumnya. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, implementasi algoritma, dan implementasi antarmuka.

5.1 Spesifikasi

Pada sub bab spesifikasi, yang akan dibahas adalah spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam aplikasi klasifikasi penyakit kopi berdasarkan citra daun ini.

5.1.1 Spesifikasi perangkat keras

Perangkat keras yang digunakan dalam penelitian ini seperti yang ditunjukkan oleh Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

Nama Perangkat Keras	Spesifikasi
Prosesor	Intel (R) Pentium (R) CPU B960 @ 2.20 Ghz
Memori (RAM)	2 GB
Hardisk	500 GB

5.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini ditunjukkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat lunak

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 7 Enterprise (32.bit)
Bahasa Pemrograman	Java JDK 6
Tools pemrograman	NetBeans IDE 7.4

5.2 Implementasi kode program

Pada tahap implementasi kode program, akan dijelaskan fungsi-fungsi dari kode yang diterapkan saat pembuatan aplikasi. Terdapat beberapa bagian penting kode yang akan dibahas dalam sub bab ini antara lain implementasi metode *Otsu*, implementasi kode untuk mencari nilai y_1 , y_2 , ni_1 , ni_2 , implementasi ekstraksi fitur, dan implementasi metode *FK-NN*.

5.2.1 Implementasi metode *Otsu*

Metode *Otsu* adalah metode yang digunakan sistem untuk melakukan pemrosesan awal pada citra daun kopi. Proses ini bertujuan untuk menghitamkan bagian citra daun kopi yang tidak berpenyakit sehingga nantinya hanya bagian daun yang berpenyakit yang akan diekstraksi untuk kemudian dihitung nilai *FK-NN*nya untuk diklasifikasikan berdasarkan penyakit kopi. Implementasi metode *Otsu* dibagi menjadi tiga bagian. Bagian pertama yaitu mencari nilai tingkat keabuan yaitu y_1 dan y_2 serta mencari jumlah piksel yang memiliki tingkat keabuan ke- i yaitu ni_1 (*histogram1*) dan ni_2 (*histogram2*) yang ditunjukkan pada *Source code* 5.1. Bagian kedua yaitu implementasi pencarian nilai *Threshold1* t_1 dan *Threshold2* t_2 , yang ditunjukkan pada *Source code* 5.2. dan bagian ketiga yaitu implementasi proses citra menjadi bentuk *Otsu* sehingga bagian yang tidak berpenyakit menjadi hitam ditunjukkan pada *Source code* 5.3

```

1 public void nilaikeabuan(BufferedImage src){
2     y1 = new int[src.getWidth()][src.getHeight()];
3     y2 = new int[src.getWidth()][src.getHeight()];
4     histogramy1=new int [256];
5     histogramy2=new int [256];
6     for (int y=0; y < src.getHeight();y++){
7         for (int x=0; x < src.getWidth();x++){
8             int rgb= src.getRGB(x, y);
9             int red=(rgb>>16)&0xFF;
10            int green=(rgb>>8)&0xFF;
11            int blue=(rgb>>0)&0xFF;
12            y1[x][y]=2*green-red-blue;
13            y2[x][y]=2*red-green-blue;
14            if (y1[x][y]<0){ y1[x][y]=0;}
15            if (y2[x][y]<0){ y2[x][y]=0;}
16            if (y1[x][y]>255){ y1[x][y]=255;}
17            if (y2[x][y]>255){ y2[x][y]=255;}
18            histogramy1[y1[x][y]]++;
19            histogramy2[y2[x][y]]++;
20        }
21    }
22 }

```

Source code 5.1 Implementasi mencari nilai y_1 , y_2 , ni_1 (*histogram1*), dan ni_2 (*histogram2*)

Penjelasan *Source code* 5.1 adalah sebagai berikut.

1. Baris 1 menunjukkan nama fungsi dari proses yaitu nilai keabuan yang memiliki paramater src dengan tipe *bufferedImage*.

2. Baris 2-3 adalah inialisasi array *y1* dan *y2* yang digunakan untuk menyimpan nilai *y1* dan *y2* pada setiap citra. Jumlah baris dan kolom sebanyak jumlah piksel citra.
3. Baris 4-5 adalah inialisasi array *histogram1* dan *histogram2* yang digunakan untuk menyimpan nilai *n1* (*histogram1*) yaitu jumlah piksel yang memiliki tingkat keabuan ke-*i* (*y1*) dan *n2* (*histogram2*) yaitu jumlah piksel yang memiliki tingkat keabuan ke-*i* (*y2*) pada setiap citra. Jumlah slot yang digunakan adalah sebanyak 256.
4. Baris 6-7 adalah proses perulangan sebanyak jumlah piksel citra.
5. Baris 8-11 adalah proses mengambil nilai warna merah, hijau, dan biru dari citra.
6. Baris 12-13 adalah proses mendapatkan nilai *y1* dan *y2*.
7. Baris 14-17 adalah proses normalisasi nilai *y1* dan *y2* yang telah didapatkan pada langkah sebelumnya.
8. Baris 18-19 adalah proses mendapatkan nilai *ni1* (*histogram1*) dan *ni2* (*histogram2*).

```

1 private int Threshold (int[] histogram, int lebar, int
2 tinggi){
3     int t=0;
4     double totalRata=0, varian, maxvarian=0, momKum1=0,
5     momKum0=0;
6     for (int x=0; x<256; x++){
7         totalRata+=
8 (double)x*histogram[x]/(lebar*tinggi);
9     }
10    for (int x=0; x<256; x++){
11        momKum0 += (double)histogram[x]/(lebar*tinggi);
12        momKum1 +=(double) x*
13 histogram[x]/(lebar*tinggi);
14        varian=(totalRata*momKum0-momKum1);
15        varian *= varian;
16        varian /=(double)(momKum0 * (1-momKum0));
17        if (maxvarian<varian){
18            maxvarian=varian;
19            t=x;
20        }
21    }
22    return t;
23 }

```

Source code 5.2 Implementasi mencari nilai *t1* (Threshold1) dan *t2* (Threshold2)

Penjelasan dari *Source code* 4.2 adalah sebagai berikut.

1. Pada baris 1 menunjukkan bahwa fungsi dengan nama *Threshold* bertipe integer memiliki tiga parameter yaitu array *histogram*, lebar dan tinggi citra.
2. Pada Baris 2-4 adalah inisialisasi variabel-variabel yang digunakan. *T* untuk menyimpan nilai *Threshold*, *totalRata* untuk menyimpan nilai total rata-rata, *varian* untuk menyimpan hasil varian, *maxvarian* untuk menyimpan nilai maksimal varian, *momKum1* untuk menyimpan nilai Momen Kumulatif ke-1, dan *momKum0* untuk menyimpan nilai Momen Kumulatif ke-0.
3. Pada baris 5 adalah proses untuk perulangan sebanyak 256 kali.
4. Pada baris 6 adalah proses mendapatkan nilai total rata-rata.
5. Pada baris 7 adalah proses untuk perulangan sebanyak 256 kali.
6. Pada baris 9 adalah proses mendapatkan nilai Momen Kumulatif ke-0.
7. Pada baris 10 adalah proses mendapatkan nilai Momen Kumulatif ke-1.
8. Pada baris 11-12 adalah proses mendapatkan nilai varian.
9. Pada baris 13 proses membandingkan nilai varian dengan nilai maksimum varian.
10. Pada baris ke 14-15 adalah proses perubahan nilai maksimum varian dan proses mendapatkan nilai *Threshold*, dimana nilai maksimum varian sama dengan nilai varian dan nilai *Threshold* sama dengan warna ke-*k*, jika kondisi pada baris 13 terpenuhi.
11. Pada baris 16 adalah menentukan nilai $t=x$.
12. Pada baris 19 adalah proses pengembalian nilai *t*.

```

1 public void imageToOtsu(BufferedImage src) {
2     BufferedImage output = new BufferedImage(lebarCitra,
3     tinggiCitra,
4         BufferedImage.TYPE_INT_RGB);
5     double ratared_ = 0;
6     double ratagreen_ = 0;
7     double ratablue_ = 0;
8     int jumlahpix = 0;
9     int t1, t2;
10    int redOtsu = 0;
11    int greenOtsu = 0;
12    int blueOtsu = 0;
13    nilaikeabuan(src);
14    t1 = Threshold(histogramy1, src.getWidth(),
15    src.getHeight());
16    t2 = Threshold(histogramy2, src.getWidth(),
17    src.getHeight());
18    t1 += T;

```

```

16         t2 -= T;
17         if (t2 < 0) {             t2 = 0; }
18         if (t1 > 255) {         t1 = 255; }
19         for (int y = 0; y < src.getHeight(); y++) {
20             for (int x = 0; x < src.getWidth(); x++) {
21                 int rgb = src.getRGB(x, y);
22                 int red = (rgb >> 16) & 0xFF;
23                 int green = (rgb >> 8) & 0xFF;
24                 int blue = (rgb >> 0) & 0xFF;
25                 if (y1[x][y] < t1 && y2[x][y] > t2) {
26                     redOtsu += red;
27                     greenOtsu += green;
28                     blueOtsu += blue;
29                     jumlahpix++;
30                 } else {
31                     red = 0;
32                     green = 0;
33                     blue = 0;
34                 }
35                 int pxl = blue + (green << 8) + (red << 16);
36                 output.setRGB(x, y, pxl);
37             }
38         }
39         ratared_ = redOtsu / jumlahpix;
40         ratagreen_ = greenOtsu / jumlahpix;
41         ratablue_ = blueOtsu / jumlahpix;
42         this.ratared = ratared_;
43         this.ratagreen = ratagreen_;
44         this.ratablue = ratablue_;
45         this.OtsuImage = output;
46     }

```

Source code 5.3 Implementasi proses menghitamkan bagian citra tidak berpenyakit dan ekstraksi fitur

Penjelasan dari *Source code 4.3* adalah sebagai berikut.

1. Baris 1 menunjukkan nama fungsi dari proses yaitu *imagerOtsu* yang memiliki paramater *src* dengan tipe *bufferedImage*.
2. Pada baris 2-3 adalah proses membuat objek *output* dari kelas *bufferedImage* yaitu kelas yang telah disediakan oleh *java*.
3. Pada baris 4-11 adalah proses inialisasi variabel-variabel yang digunakan dalam fungsi *imagerOtsu*. *Ratared* digunakan untuk menyimpan nilai rata-rata warna merah, *ratagreen* digunakan untuk menyimpan nilai rata-rata warna hijau, *ratablue* digunakan untuk menyimpan nilai rata-rata warna biru, *jumlahpix* digunakan untuk menyimpan nilai jumlah piksel, *t1* dan *t2*

digunakan untuk menyimpan nilai *Threshold1* dan *Threshold2*, *redOtsu* digunakan untuk menyimpan nilai warna merah, *greenOtsu* digunakan untuk menyimpan nilai warna hijau, *blueOtsu* digunakan untuk menyimpan nilai warna biru.

4. Pada baris 12 adalah proses mencari nilai y_1 , y_2 , ni_1 (*histogram1*), ni_2 (*histogram2*) dengan memanggil fungsi nilai keabuan dengan parameter *src*.
5. Pada baris 13 adalah proses mendapatkan nilai t_1 dengan memanggil dan memasukkan array *histogram1*, lebar dan tinggi citra.
6. Pada baris 14 adalah proses mendapatkan nilai t_2 dengan memanggil dan memasukkan array *histogram2*, lebar dan tinggi citra.
7. Pada baris 15 adalah proses menambahkan nilai t_1 dengan peubah nilai *Threshold* (T).
8. Pada baris 16 adalah proses mengurangi nilai t_2 dengan peubah nilai *Threshold* (T).
9. Pada baris 17-18 adalah proses normalisasi nilai t_1 dan t_2 .
10. Pada baris 19-20 adalah proses perulangan sebanyak jumlah piksel citra
11. Pada baris 21-24 adalah proses mengekstrak nilai warna merah, hijau, dan biru pada citra.
12. Pada baris 25 adalah proses untuk mengecek kondisi apakah nilai y_1 lebih kecil dari nilai t_1 dan nilai y_2 lebih besar dari nilai t_2 atau tidak.
13. Pada baris 26-29 adalah proses mengambil warna merah, hijau, biru apabila kondisi pada baris 35 terpenuhi.
14. Pada baris 30-33 adalah proses menghitamkan bagian citra yang tidak memenuhi kondisi pada baris 25.
15. Pada baris 35-36 adalah proses pembentukan nilai warna piksel pada variabel output.
16. Pada baris 39-41 adalah proses mengambil rata-rata merah, rata-rata hijau, dan rata-rata biru sebagai langkah ekstraksi fitur.
17. Pada baris 42-45 adalah proses membentuk rata-rata merah, rata-rata hijau, rata-rata biru dan hasil output.

5.2.2 Implementasi proses pelatihan

Proses pelatihan sistem terhadap data latih bertujuan untuk mendapatkan kelas yang nantinya digunakan untuk mengidentifikasi penyakit pada citra uji. Proses pelatihan diawali dengan melihat nama file data latih yang selanjutnya nama file tersebut akan menjadi kelas dari data setiap penyakit. Jadi nama file dan penyakit harus sesuai sehingga pelatihan data yang dilakukan benar. Proses pelatihan sistem terhadap data latih ditunjukkan pada *Source code* 5.4.

```
1 Final String path = "./CitraKopi\\Data Training";
2     BufferedImage imageInput;
3     ImageProcessing imagePro;
4     ArrayDataTiapFitur nilaiFitur;
5     int byk_data_train = 24;
6     int byk_data_train_bercak = 10;
7     int byk_data_train_karat = 14;
8     double[] dataR = new double[byk_data_train + 1];
9     double[] dataG = new double[byk_data_train + 1];
10    double[] dataB = new double[byk_data_train + 1];
11    int[] penyakit = new int[byk_data_train + 1];
12    final File folder = new File(path);
13    for (int j = 1; j <= byk_data_train; j++) {
14        try {
15            String[] fileName = {" "};
16            String[] kelasFile = {" "};
17            if (j <= byk_data_train_bercak) {
18                fileName[0] = (path +
19                "/bercak".concat(Integer.toString(j).concat(".jpg")));
20                imageInput = ImageIO.read(new File(path +
21                "/bercak".concat(Integer.toString(j).concat(".jpg"))));
22                kelasFile[0] = "bercak";
23                penyakit[j] = 1;
24            } else {
25                fileName[0] = (path + "/karat".concat(Integer.toString(j -
26                byk_data_train_bercak).concat(".jpg")));
27                imageInput = ImageIO.read(new File(path +
28                "/karat".concat(Integer.toString(j -
29                byk_data_train_bercak).concat(".jpg"))));
30                kelasFile[0] = "karat";
31                penyakit[j] = 2;
32            }
33            imagePro =
34            dataPro.dataTraining(imageInput, kelasFile[0]);
35            nilaiFitur = dataPro.getDataFitur();
36            dataR[j] = nilaiFitur.red;
37            dataG[j] = nilaiFitur.green;
38            dataB[j] = nilaiFitur.blue;
39        } catch (IOException e) {
40            System.out.println("Data Tidak
41            Ditemukan");
42        }
43    }
44 }
```

Source code 5.4 Implementasi proses pelatihan data latih

Penjelasan dari *Source code* 5.4 adalah sebagai berikut.

1. Pada baris 1 adalah proses mengambil data dari sebuah folder yang selanjutnya data tersebut akan dilakukan pelatihan
2. Pada baris 2-4 membuat objek baru dari kelas lain.
3. Pada baris 5-7 adalah proses inialisasi jumlah data training, jumlah data yang berpenyakit bercak, dan jumlah data yang berpenyakit karat.
4. Pada baris 8-10 adalah proses inialisasi array untuk menyimpan nilai warna merah, hijau, dan biru.
5. Pada baris 11 adalah proses inialisasi array untuk penyakit dari setiap data.
6. Pada baris 12 adalah proses membaca data dari folder.
7. Pada baris 13 adalah proses perulangan sebanyak data latih.
8. Pada baris 14 adalah proses try untuk mencari kelas citra.
9. Pada baris 15-16 adalah proses inialisasi string untuk nama file dan kelas file.
10. Pada baris 17-22 adalah proses pengecekan data, apabila nama file data terdiri dari kata bercak dan berformat jpg, maka data file tersebut akan terlatih sebagai penyakit kopi berkelas bercak.
11. Pada baris 23-29 adalah proses pengecekan data, apabila nama file data terdiri dari kata karat dan berformat jpg, maka data file tersebut akan terlatih sebagai penyakit kopi berkelas karat.
12. Pada baris 33-35 adalah proses mengambil nilai fitur dari data yang telah di *load*. Fitur yang diambil terdiri dari nilai warna merah, hijau, dan biru.
13. Pada baris 37-38 adalah proses apabila keadaan pada baris 14 tidak terpenuhi pada baris 34-35 adalah proses apabila keadaan pada baris 14 tidak terpenuhi.

5.2.3 Implementasi proses pengujian

Implementasi proses pengujian adalah proses menguji data dengan menggunakan pertimbangan hasil pelatihan data yang telah dilakukan pada proses sebelumnya. Proses pengujian dilakukan dengan mencari jarak antara data uji dan data latih yang kemudian akan diurutkan mulai dari yang terkecil. Proses selanjutnya adalah mencari nilai keanggotaan *Fuzzy* berdasarkan nilai *k* yang ditentukan. *Source code* dari proses pengujian ditunjukkan pada *Source code* 5.5.

```

1 public void uji_fKNN (double [] R,double[] G,double []B) {
2     int data = 55;
3     int datauji=6;
4     double[] data_r = new double [data];
5     double[] data_g = new double [data];

```

```
6 double[] data_b = new double [data];
7 int[] data_kategori = new int [data];
8 hasilpenyakituji = new String [datauji+1];
9 data_kategori=penyakit;
10 double[] euclidean_distance = new double[data];
11 double[] indexs_euc = new double[data];
12 double[] indexs_sorting = new double[data];
13 for (int x=1; x<=datauji; x++){
14     for (int i = 1; i < data; i++) {
15         data_r[i] = r [i];
16         data_g[i] = g [i];
17         data_b[i] = b [i];
18         data_kategori[i] = penyakit [i];
19         euclidean_distance[i] = Math.sqrt(Math.pow(data_r[i] -
20 R [x], 2) + Math.pow(data_g[i] - G[x], 2) + Math.pow(data_b[i] -
21 B[x], 2));
22         indexs_euc[i] = euclidean_distance[i];}
23 Arrays.sort(euclidean_distance);
24 for (int i = 1; i < data; i++) {
25     for (int j = 1; j < data; j++) {
26         if (euclidean_distance[i] == indexs_euc[j]) {
27             indexs_sorting[i] = data_kategori[j];
28             break;
29         }}}
30 double hasil_1;
31 double hasil_2;
32 double pembilang1 = 0;
33 double penyebut1 = 0;
34 double pembilang2 = 0;
35 double penyebut2 = 0;
36 for (int i = 1; i <= K; i++) {
37     if (indexs_sorting[i] == 1) {
38         pembilang1 += 1 * (Math.pow(euclidean_distance[i], (-2) / m-1));
39         penyebut1 += Math.pow(euclidean_distance[i], (-2) / m-1);
40     } else {
41         pembilang1 += 0 * (Math.pow(euclidean_distance[i], (-2) / m-1));
42         penyebut1 += Math.pow(euclidean_distance[i], (-2) / m-1);
43     }}
44     hasil_1 = pembilang1 / penyebut1;
45     for (int i = 1; i <= K; i++) {
46         if (indexs_sorting[i] == 2) {
```

```

46   pembilang2 += 1 * (Math.pow(euclidean_distance[i], (-2) / m-1));
47   penyebut2 += Math.pow(euclidean_distance[i], (-2) / 1);
48       }else {
49   pembilang2 += 0 * (Math.pow(euclidean_distance[i], (-2) / m-1));
50   penyebut2 += Math.pow(euclidean_distance[i], (-2) / m-1);
51       }}
52   hasil_2 = pembilang2 / penyebut2;
53   if (hasil_1 > hasil_2) {
54       hasilpenyakituji [x]= "bercak";
55   System.out.println("Hasil = " + hasil_1 +hasilpenyakituji [x]);
56   } else {
57       hasilpenyakituji [x]="karat";
58   System.out.println("Hasil = " + hasil_2 + hasilpenyakituji [x]);
59   }}}

```

Source code 5.5 Implementasi proses pengujian

Penjelasan dari *Source code* 5.4 adalah sebagai berikut.

1. Pada baris 1 menunjukkan nama fungsi yaitu *uji_fKNN* yang memiliki tiga parameter bertipe double, yaitu *R (red)*, *G (green)* dan *B (blue)*.
2. Pada baris 2-12 adalah proses inialisasi variabel-variabel yang digunakan dalam fungsi *uji_fKNN*, yang meliputi *data_r*, *data_g*, *data_b*, *euclidean_distance*, *index_eu*, *index_sorting* yang memiliki type array double, *data_kategori* yang bertipe integer array, hasil penyakit uji, dan *data_kategori*.
3. Pada baris 13 adalah perulangan yang dilakukan sebanyak data uji.
4. Pada baris 14 adalah perulangan yang dilakukan sebanyak data latih.
5. Pada baris 15-18 adalah proses mengambil nilai warna merah, hijau, dan biru dari citra serta mengambil nilai penyakit untuk diproses.
6. Pada baris 19-20 adalah proses untuk menghitung *Euclidean Distance*.
7. Pada baris 21 adalah proses untuk memberikan index terhadap jarak yang diketahui.
8. Pada baris 22 adalah proses untuk mengurutkan hasil dari *Euclidean Distance* yang berbentuk array.
9. Pada baris 23-24 melakukan perulangan sebanyak jumlah data.
10. Pada baris 25 dilakukan pengecekan kondisi apabila urutan *Euclidean Distance* sama dengan hasil pengurutan maka indek sama dengan data kategori.
11. Pada baris 26 adalah proses memanggil nilai *Euclidean Distance* yang sudah diurutkan.

12. Pada baris 29-34 merupakan inisialisasi variabel-variabel yang akan digunakan dalam perhitungan nilai keanggotaan.
13. Pada baris 35 adalah proses perulangan sebanyak k , nilai k telah ditentukan sebelumnya.
14. Pada baris 36 adalah pengecekan saat kondisi index ke- i sama dengan satu.
15. Pada baris 37-38 adalah proses perhitungan nilai keanggotaan bagian pembilang dan penyebut pada bagian pembilang.
16. Pada baris 40-41 adalah proses perhitungan nilai pembilang dan penyebut jika indek tidak sama dengan satu.
17. Pada baris 43 merupakan proses mencari pembilang pada persamaan mencari nilai keanggotaan *Fuzzy*.
18. Pada baris 44 dilakukan proses perulangan sebanyak nilai k .
19. Pada baris 45 adalah kondisi saat index ke- i sama dengan dua.
20. Pada baris 46-47 adalah proses perhitungan nilai keanggotaan bagian pembilang dan penyebut.
21. Pada bari 49-50 adalah proses perhitungan nilai pembilang dan penyebut jika indek tidak sama dengan dua pada bagian penyebut.
22. Pada baris 52 merupakan proses mencari penyebut pada Persamaan mencari nilai keanggotaan *Fuzzy*.
23. Pada baris 53 adalah pengecekan kondisi apabila pembilang lebih besar dari pada penyebut.
24. Pada baris 52-55 merupakan proses penentuan penyakit bercak.
25. Pada baris 56-58 merupakan proses penentuan penyakit karat.

5.3 Implementasi antarmuka

Implementasi antarmuka merupakan penerapan pada perancangan antarmuka pada Bab 4. Implementasi antarmuka dilakukan dengan membuat *user interface* yang nantinya digunakan untuk berinteraksi langsung degan user. Terdapat dua implementasi antarmuka pada sistem ini, halaman utama yaitu halaman awal saat user membuka sistem, dan halaman pengujian yaitu halaman pengujian sistem.

5.3.1 Tampilan halaman utama

Halaman utama adalah halaman yang muncul saat pertama kali user menjalankan sistem. Halaman utama ini digunakan oleh penulis untuk memperkenalkan sistem, mulai dari deskripsi singkat sistem, dan perkenalan penulis. Tampilan halaman utama ditunjukkan pada Gambar 5.1.

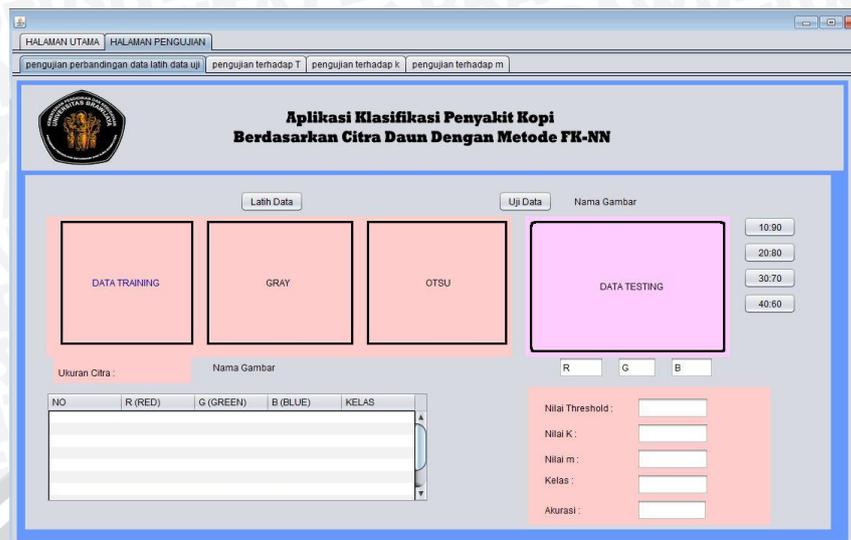


Gambar 5.1 Implementasi halaman utama

Gambar 5.1 menunjukkan halaman utama. Menu pertama yaitu halaman utama berisi tentang halaman home yang terdiri dari judul aplikasi yaitu nama dari aplikasi yang telah dibuat, penulis yaitu perkenalan penulis, deskripsi singkat mengenai tanaman kopi sehingga sebelum memulai untuk menguji data pengguna lebih dulu mengetahui tujuan dari pembuatan aplikasi dan bagian terakhir adalah citra untuk penyakit kopi yaitu bercak daun kopi, dan karat daun kopi. Menu kedua yaitu halaman pengujian dimana user dapat mulai menggunakan sistem.

5.3.2 Tampilan halaman pengujian

Halaman pengujian muncul saat menu yang dipilih oleh *user* adalah halaman pengujian. Tujuan dari halaman pengujian adalah untuk melatih sejumlah data dan sekaligus menguji data yang lain. Bagian dari halaman utama terdiri dari menu bar untuk pilihan dari user, header aplikasi terdiri dari logo Universitas Brawijaya dan judul aplikasi. Pada proses pelatihan data, terdapat tombol latihan data, tujuan dari tombol ini adalah mengambil data yaitu citra daun dari satu folder yang telah ditentukan kemudian menampilkan citra asli pada label pertama, citra *gray* pada label kedua, dan citra hasil *Otsu* pada label ketiga. Bagian bawah label citra terdapat label detail citra, tujuan dari label detail citra adalah menampilkan ukuran dari citra yang ditampilkan oleh sistem. Label detail citra akan terus berulang sampai citra yang ditampilkan selesai dilatih dalam satu folder. Bagian lain dari pelatihan adalah tabel. Tabel ini memiliki lima kolom, kolom no untuk penomoran citra, kolom *R (RED)* untuk penempatan hasil ekstraksi nilai warna merah pada citra, kolom *G (GREEN)* untuk penempatan hasil ekstraksi nilai warna hijau pada citra, kolom *B (BLUE)* untuk penempatan hasil ekstraksi nilai biru pada citra, dan kolom kelas yaitu kolom yang berisi kelas-kelas citra yang telah dilatih oleh sistem. Implementasi antarmuka halaman utama ditunjukkan pada Gambar 5.2.



Gambar 5.2 Implementasi halaman pengujian

Gambar 5.2 menunjukkan implementasi antarmuka halaman pengujian, yang terdiri dari proses pelatihan dan pengujian dalam satu halaman. Pada bagian uji data terdapat tombol uji data, fungsi dari tombol ini adalah untuk mengambil data dari suatu folder kemudian data diuji dan citra asli dari data ditampilkan pada label citra testing. Terdapat juga tiga label dibawah label citra, label ini berfungsi untuk menampilkan hasil ekstraksi warna merah, warna hijau, dan warna biru dari citra. Bagian terakhir dari proses pengujian yaitu panel yang berisi nilai *Threshold*, nilai *k*, nilai *m*, kelas, dan akurasi.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas mengenai pengujian dan analisis klasifikasi penyakit kopi dengan menggunakan metode *FK-NN*. Proses pengujian dilakukan dengan menguji beberapa skenario pengujian. Pada pengujian akurasi akan dilakukan perbandingan antara hasil pelatihan oleh sistem dengan pengenalan secara manual. Proses analisis dilakukan dengan menganalisa hasil pengujian yang telah dilakukan.

Proses pengujian dilakukan untuk mengetahui tingkat akurasi penggunaan metode *FK-NN* dalam pengenalan dan pengklasifikasian penyakit kopi. Terdapat beberapa skenario dalam pengujian, antara lain pengujian terhadap perbandingan data latih dan data uji, pengujian terhadap nilai peubah *Threshold* (T), pengujian terhadap nilai k , dan pengujian terhadap nilai m .

6.1 Pengujian terhadap perbandingan data latih dan data uji

Pengujian pertama yang dilakukan adalah pengujian terhadap perbandingan data latih dan data uji. Pengujian ini bertujuan untuk mengetahui perbandingan data latih dan data uji yang optimal dari data.

6.1.1 Skenario pengujian terhadap perbandingan data latih dan data uji

Pengujian menggunakan lima variasi perbandingan data latih dan data uji, yaitu 90:10, 80:20, 70:30, 60:40 dan 50:50. Pengujian yang dilakukan menggunakan 60 data latih dan 60 data uji dengan lima variasi perbandingan dan pengambilan data latih dan data uji dilakukan secara random. Variabel penentu yang lain disamakan dalam setiap skenario pengujian tiap perbandingan, variabel penentu tersebut antara lain nilai T (nilai peubah *Threshold*) yaitu 10, k yaitu 5, dan m yaitu 2. Pengujian terhadap perbandingan data latih dan data uji dilakukan sebanyak 5 kali untuk tiap perbandingan sehingga menghasilkan akurasi terbaik. Hasil dari pengujian terhadap jumlah data latih dan data uji ditunjukkan pada Tabel 6.2.

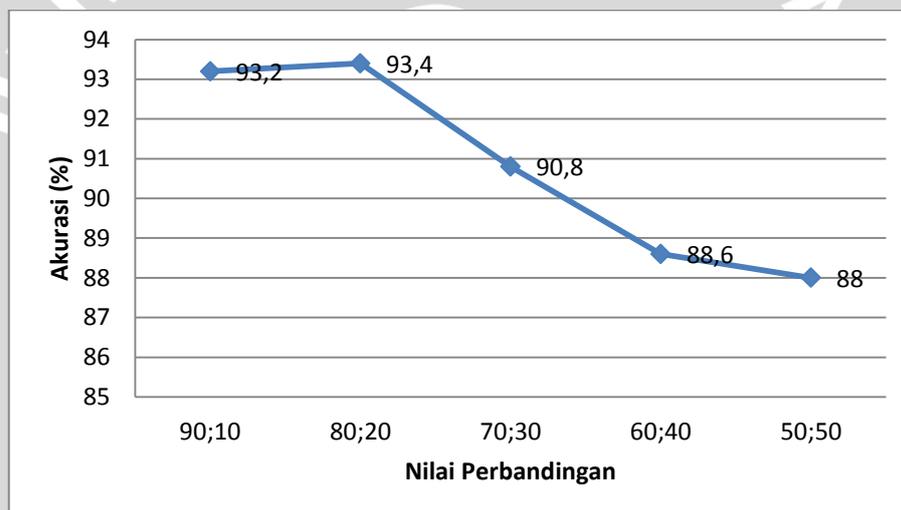
Tabel 6.1 Hasil uji coba akurasi terhadap perbandingan data latih dan data uji

Rasio Perbandingan (%)	Hasil Akurasi (%)					Rata-rata Akurasi (%)
	Percobaan ke- i					
	1	2	3	4	5	
90:10	100	100	100	83,3	83,3	93,2
80:20	100	91,67	91,67	100	83,3	93,4
70:30	94,44	83,33	88,89	94,44	94,44	90,8
60:40	91,67	91,67	91,67	91,67	87,5	88,6
50:50	90	93,33	80	96,67	80	88

Pada Tabel 6.1 diketahui bahwa nilai perbandingan 90:10 dengan lima kali percobaan menghasilkan akurasi tertinggi pada percobaan ke satu, dua dan tiga dengan akurasi sebesar 100%. Pada skenario percobaan kedua dengan perbandingan 80:20 didapatkan akurasi tertinggi sebesar 100% pada percobaan pertama dan keempat. Skenario ketiga dengan perbandingan 70:30 menghasilkan akurasi tertinggi yaitu 94,44% pada tiga kali percobaan yaitu pertama, ketiga, dan keempat. Pada skenario keempat hasil akurasi yang didapat sama dalam setiap percobaan yaitu sebesar 91,67% kecuali pada percobaan kelima sebesar 87,7%. Dan pada skenario percobaan terakhir dengan perbandingan 50:50 menghasilkan akurasi tertinggi pada percobaan keempat dengan akurasi 96,67%.

6.1.2 Analisis pengujian terhadap perbandingan data latih dan data uji

Pengujian yang pertama dilakukan adalah pengujian terhadap perbandingan data latih dan data uji. Hasil pengujian terhadap perbandingan data latih dan data uji ditunjukkan oleh Gambar 6.1.



Gambar 6.1 Pengujian terhadap perbandingan data latih dan data uji

Pada Gambar 6.1 menjelaskan hasil akurasi yang didapatkan dari rata-rata akurasi dalam tiap percobaan dalam satu skenario pengujian. Pengujian terhadap perbandingan data latih dan data uji yang dilakukan dengan lima skenario pengujian perbandingan. Grafik garis semakin menurun yang menghubungkan semua skenario perbandingan data latih dan data uji. Perbandingan pertama yaitu 90:10 yaitu dengan komposisi 54 data latih dan 6 data uji menghasilkan rata-rata akurasi sebesar 93,2%, perbandingan kedua dengan komposisi 48 data latih dan 12 data uji menghasilkan akurasi rata-rata sebesar 93,4%, perbandingan ketiga dengan komposisi 42 data latih dan 18 data uji menghasilkan akurasi rata-rata sebesar 90,8%, perbandingan keempat dengan komposisi 36 data latih dan 24 data uji menghasilkan akurasi rata-rata sebesar 88,6% dan perbandingan skenario pengujian terakhir dengan jumlah komposisi data latih dan data uji masing-masing 50 menghasilkan rata-rata akurasi sebesar 88%. Berdasarkan

gambar didapatkan kesimpulan bahwa akurasi tertinggi didapatkan saat pengujian pertama dan kedua yaitu dengan perbandingan 90:10 dan 80:20.

Berdasarkan Gambar 6.1, perbedaan akurasi yang dihasilkan dipengaruhi oleh perbandingan data latih dan data uji, semakin banyak data latih dan semakin sedikit data uji, maka akurasi yang dihasilkan semakin tinggi. Sebaliknya, semakin sedikit data latih dan semakin banyak data uji maka akurasi yang dihasilkan semakin rendah. Hal ini disebabkan karena proses pelatihan menggunakan banyak data sehingga variasi hasil latih yang didapatkan lebih beragam sehingga memudahkan pada proses pengujian yang menggunakan data lebih sedikit dalam memutuskan klasifikasi data citra sehingga akurasi yang dihasilkan tinggi.

6.2 Pengujian terhadap nilai peubah *Threshold (T)*

Pengujian kedua yang dilakukan adalah pengujian terhadap nilai peubah *Threshold (T)*. Peubah *Threshold (T)* adalah variabel yang digunakan pada *preprocessing* citra dengan metode *Otsu*. Pengujian ini bertujuan untuk mencari nilai peubah *Threshold (T)* yang paling optimal untuk proses *preprocessing* sehingga menghasilkan akurasi terbaik.

6.2.1 Skenario pengujian terhadap nilai peubah *Threshold (T)*

Nilai yang digunakan untuk pengujian terhadap nilai peubah *Threshold (T)* antara lain 10, 20, 40, 60, 80, dan 100. Perhitungan dilakukan dengan metode *Otsu* sebagai metode *preprocessing* dan menggunakan metode *FK-NN* sebagai metode klasifikasi dengan nilai *k* yaitu 5, dan nilai *m* yaitu 2. Data yang digunakan menggunakan hasil perbandingan pada pengujian pertama yaitu perbandingan 70:30 dengan jumlah data latih sebanyak 42 dan data latih sebanyak 18. Hasil dari pengujian terhadap nilai peubah *Threshold (T)* ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil uji coba akurasi terhadap nilai peubah *Threshold (T)*

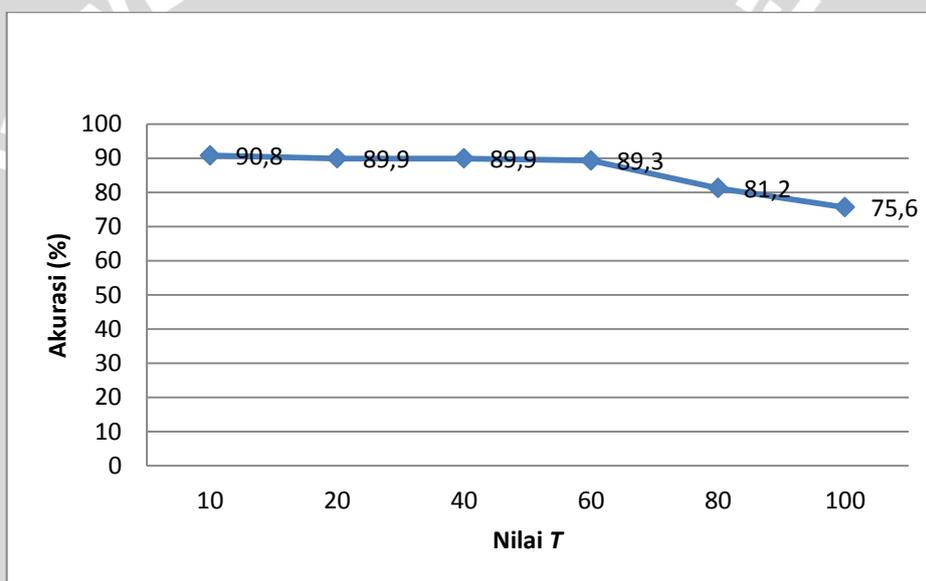
Nilai T	Hasil Akurasi (%)					Rata-rata Akurasi (%)
	Percobaan ke- <i>i</i>					
	1	2	3	4	5	
10	94,44	83,33	88,89	94,44	94,44	90,8
20	88,89	83,33	88,89	94,44	94,44	89,9
40	88,89	83,33	88,89	94,44	94,44	89,9
60	91,6	94,44	88,89	94,4	77,2	89,3
80	72	88,89	88,89	77,77	77,77	81,2
100	66,66	77,77	83,3	72,2	77,77	75,6

Pada Tabel 6.2 diketahui bahwa nilai peubah *Threshold (T)* berpengaruh terhadap akurasi yang dihasilkan pada proses pengujian. Pada skenario pengujian pertama dengan nilai *T* 10 didapat nilai akurasi tertinggi yaitu 94,44% pada percobaan pertama, keempat, dan kelima. Skenario pengujian kedua

mendapatkan nilai akurasi terbesar yaitu 94,44% pada percobaan keempat dan kelima. Skenario pengujian selanjutnya menggunakan nilai T 40 dan didapat nilai akurasi tertinggi pada percobaan keempat dan kelima dengan nilai 94,4%. Skenario pengujian berikutnya yaitu dengan nilai T 60 didapatkan nilai akurasi terbaik pada percobaan kedua dan keempat dengan akurasi 94,44%. Skenario pengujian kelima menggunakan T 80 mendapatkan nilai akurasi terbaik yaitu 88,89 pada percobaan kedua dan ketiga. Dan skenario pengujian terakhir pada perbandingan data adalah T dengan nilai 100 menghasilkan akurasi terbaik pada percobaan ketiga dengan akurasi 83,3%.

6.2.2 Analisis pengujian terhadap nilai peubah *Threshold* (T)

Pengujian kedua yang dilakukan adalah pengujian terhadap nilai peubah *Threshold* (T). Semua skenario pengujian terhadap nilai peubah *Threshold* (T) dilakukan dengan jumlah data latih 42 dan data uji 18. Hasil pengujian terhadap nilai peubah *Threshold* ditunjukkan oleh Gambar 6.2.



Gambar 6.2 Pengujian terhadap nilai peubah *Threshold* (T)

Berdasarkan Gambar 6.1 menunjukkan garis semakin menurun pada tiap pengujian. Skenario pengujian pertama mendapatkan 90,8% sebagai rata-rata akurasi, skenario pengujian kedua mendapatkan rata-rata akurasi sebesar 89,9%, skenario pengujian ketiga memiliki rata-rata akurasi yang sama dengan skenario kedua yaitu 89,9%, kemudian skenario berikutnya dengan nilai T 60 mendapatkan rata-rata akurasi sebesar 89,3%, untuk skenario pengujian dengan nilai T 80 mempunyai rata-rata akurasi sebesar 81,2% dan skenario pengujian terakhir dengan nilai T 100 mendapatkan rata-rata akurasi terendah yaitu 75,6%. Dari Gambar 6.2 dapat disimpulkan bahwa semakin besar T , maka akurasi yang dihasilkan semakin rendah. Hal ini disebabkan semakin besar T , maka nilai pembandingan *histogram* akan semakin besar yang menyebabkan terbentuknya banyak *noise*. Sehingga banyak bagian citra tidak terdeteksi atau terdeteksi salah.

6.3 Pengujian terhadap nilai k

Pengujian selanjutnya yaitu pengujian terhadap nilai k . Nilai variabel k digunakan oleh sistem pada proses klasifikasi citra dengan metode *Fuzzy K-Nearest Neighbor*. Pengujian ini bertujuan untuk mencari nilai k yang paling optimal untuk proses klasifikasi citra sehingga klasifikasi yang dihasilkan memiliki akurasi yang tinggi.

6.3.1 Skenario pengujian terhadap nilai k

Nilai yang digunakan untuk pengujian terhadap nilai k antara lain 3, 5, 7, 10, 15, 20, 25. Perhitungan dilakukan dengan metode *Otsu* sebagai metode *preprocessing* dan menggunakan metode *FK-NN* sebagai klasifikasi dengan nilai peubah *Threshold (T)* yaitu 10, dan nilai m yaitu 2. Data yang digunakan menggunakan hasil perbandingan pada pengujian pertama pada pengujian perbandingan data latih dan data uji yaitu perbandingan 70:30 dengan jumlah data latih sebanyak 42 dan data latih sebanyak 18. Hasil dari pengujian terhadap nilai peubah k ditunjukkan pada Tabel 6.3.

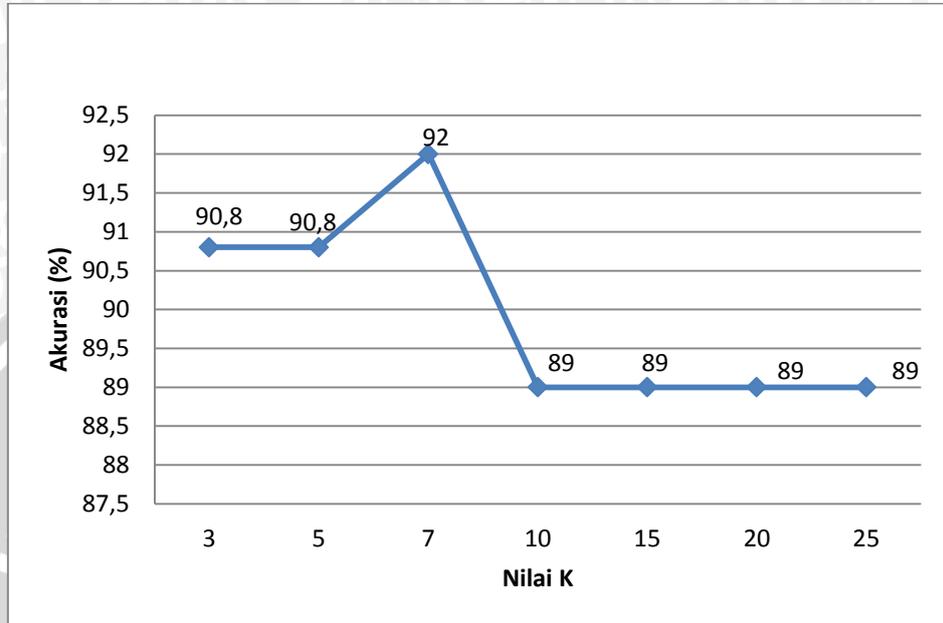
Tabel 6.3 Hasil uji coba akurasi terhadap nilai k

Nilai k	Hasil Akurasi (%)					Rata-rata Akurasi (%)
	Percobaan ke- i					
	1	2	3	4	5	
3	94,44	83,33	88,89	94,44	94,44	91
5	94,44	83,33	88,89	94,44	94,44	91
7	88,89	88,89	83,33	94,44	94,44	92
10	88,89	83,33	88,89	94,44	88,89	89
15	88,89	83,33	88,89	94,44	88,89	89
20	88,89	83,33	88,89	94,44	88,89	89
25	88,89	83,33	88,89	94,44	88,89	89

Tabel 6.3 menunjukkan pengaruh nilai k yaitu nilai tetangga terhadap akurasi klasifikasi yang dihasilkan oleh sistem. Pada skenario pengujian pertama dengan menggunakan nilai k 3 didapat akurasi terbaik yaitu 94,44% pada percobaan pertama, keempat dan kelima. Skenario kedua dengan nilai k 5 memiliki nilai yang sama dengan skenario pengujian pertama. Skenario pengujian ketiga dengan nilai k 7 menghasilkan akurasi terbaik pada percobaan keempat dan kelima dengan nilai 94,44%. Skenario pengujian berikutnya adalah dengan nilai k 10, 15, 20, dan 25 memiliki akurasi terbaik yang sama yaitu 94,44% pada percobaan keempat.

6.3.2 Analisis pengujian terhadap nilai k

Pengujian ketiga yang dilakukan adalah pengujian terhadap k . Semua skenario pengujian terhadap nilai k dilakukan dengan jumlah data latih 42 dan data uji 18. Hasil pengujian terhadap nilai k ditunjukkan oleh Gambar 6.3.



Gambar 6.3 Pengujian terhadap nilai k

Gambar 6.3 menunjukkan grafik lurus pada nilai k 3 dan 5 dengan akurasi 90,8%, kemudian akurasi tertinggi pada nilai k 7 dengan akurasi 92% dan grafik menurun pada k selanjutnya yaitu 10, 15, 20, dan 25 dengan nilai akurasi yang sama yaitu 89%. Hasil pengujian yang didapatkan pada pengujian terhadap nilai k terhadap tingkat akurasi yaitu semakin bertambahnya nilai k , maka akurasi yang didapatkan semakin menurun dan stabil pada nilai k tertentu. Hal ini disebabkan rentang kelas pada k yang semakin banyak memberikan pengaruh besar pada penentuan prediksi klasifikasi dan nilai keanggotaan yang digunakan untuk penentuan kelas target pada penelitian kemudian akurasi yang dihasilkan oleh nilai k yang berdekatan cenderung stagnan, hal ini disebabkan persebaran nilai fitur pada data, untuk data yang berdekatan lebih sering memiliki kelas yang sama sehingga mempengaruhi nilai derajat keanggotaan.

6.4 Pengujian terhadap nilai m

Pengujian selanjutnya adalah pengujian terhadap nilai m . Nilai variabel m digunakan oleh sistem pada proses menentukan derajat keanggotaan tiap kelas pada *Fuzzy K-Nearest Neighbor*. Pengujian ini bertujuan untuk mencari nilai m yang paling optimal untuk proses mencari derajat keanggotaan untuk selanjutnya diproses sebagai klasifikasi citra sehingga klasifikasi yang dihasilkan memiliki akurasi yang tinggi.

6.4.1 Skenario pengujian terhadap nilai m

Nilai yang digunakan untuk pengujian terhadap nilai m antara lain 1, 2, 3, 4, 6, 8, dan 10. Perhitungan dilakukan dengan metode *Otsu* sebagai metode *preprocessing* dan menggunakan metode *FK-NN* sebagai klasifikasi dengan nilai peubah *Threshold (T)* yaitu 10, dan nilai k yaitu 5. Data yang digunakan menggunakan hasil perbandingan pada pengujian pertama pada pengujian perbandingan data latih dan data uji yaitu perbandingan 70:30 dengan jumlah data latih sebanyak 42 dan data latih sebanyak 18. Hasil dari pengujian terhadap nilai peubah m ditunjukkan pada Tabel 6.4.

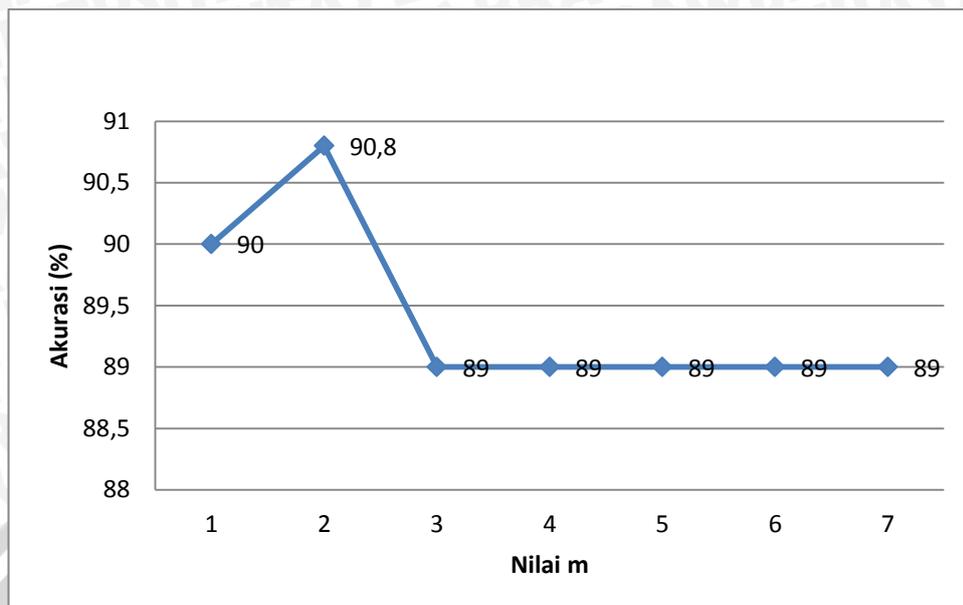
Tabel 6.4 Hasil uji coba akurasi terhadap nilai m

Nilai m	Hasil Akurasi (%)					Rata-rata Akurasi (%)
	Percobaan ke- i					
	1	2	3	4	5	
1	94,44	88,89	88,89	88,89	88,89	90
2	94,44	83,33	88,89	94,44	94,44	90,8
3	88,89	88,89	88,89	94,44	88,89	89
4	88,89	88,89	88,89	94,44	88,89	89
6	88,89	88,89	88,89	94,44	88,89	89
8	88,89	88,89	88,89	94,44	88,89	89
10	88,89	88,89	88,89	94,44	88,89	89

Tabel 6.4 menunjukkan pengaruh nilai m terhadap akurasi klasifikasi yang dihasilkan oleh sistem. Pada skenario pengujian pertama dengan nilai m 1 akurasi tertinggi adalah 94,44% yang didapat pada percobaan pertama. Skenario pengujian kedua dengan nilai m 2 menghasilkan nilai akurasi tertinggi yaitu 94,44% pada percobaan pertama, keempat, dan kelima. Skenario pengujian selanjutnya yaitu dengan nilai m 3, 4, 6, 8, dan 10 menghasilkan akurasi tertinggi yang sama yaitu pada percobaan keempat dengan akurasi 94,44%.

6.4.2 Analisis pengujian terhadap nilai m

Pengujian keempat yang dilakukan adalah pengujian terhadap nilai m . Semua skenario pengujian terhadap nilai m dilakukan dengan jumlah data latih 42 dan data uji 18. Perhitungan dilakukan dengan metode *Otsu* sebagai metode *preprocessing* dan menggunakan metode *FK-NN* sebagai klasifikasi dengan nilai peubah *Threshold (T)* yaitu 10, dan nilai k yaitu 5. Hasil pengujian terhadap terhadap nilai m ditunjukkan oleh Gambar 6.4.



Gambar 6.4 Pengujian terhadap nilai m

Pada Gambar 6.4 diketahui bahwa skenario pengujian kesatu dan kedua memberikan nilai rata-rata akurasi yang tinggi yaitu berturut-turut 90% dan 90,8%. Dengan nilai m yaitu 1 dan 2. Kemudian pada skenario pengujian ketiga sampai skenario ketujuh, grafik bergerak lurus dengan akurasi yang sama yaitu 89%. Pada pengujian terhadap nilai m , nilai m pada proses pengklasifikasian menggunakan metode *Fuzzy K-Nearest Neighbor* tidak berpengaruh langsung terhadap nilai akurasi, namun berpengaruh pada nilai derajat keanggotaan tiap-tiap data uji terhadap masing-masing kelas. Variabel m merupakan bobot pangkat yang digunakan untuk mengetahui seberapa besar jarak antar tetangga ketika menghitung pengaruh tetangga pada nilai keanggotaan. Nilai m yang semakin besar akan membuat nilai keanggotaan semakin rendah, sehingga berpengaruh pada penentuan hasil kelas klasifikasi penyakit kopi berdasarkan citra daun dimana hasil klasifikasi tersebut yang mempengaruhi hasil klasifikasi.

BAB 7 PENUTUP

7.1 Kesimpulan

Dari hasil uji dan analisa yang telah dilakukan terhadap klasifikasi penyakit tanaman kopi menggunakan metode *Fuzzy K-Nearest Neighbor*, dapat diambil kesimpulan diantaranya ialah:

1. Algoritma *Fuzzy K-Nearest Neighbor (FK-NN)* dapat diterapkan pada data yang berupa citra daun kopi. Terdapat dua proses utama yang dilakukan dalam penentuan klasifikasi tanaman kopi berdasarkan daun, yaitu *preprocessing* citra dan proses klasifikasi. Proses *preprocessing* bertujuan untuk memperbaiki kualitas citra sehingga akurasi yang dihasilkan pada proses klasifikasi bisa optimal. Selanjutnya proses klasifikasi melibatkan beberapa tahapan yaitu menentukan k sebagai tetangga terdekat yang kemudian dicari jarak antar data uji dan data latih, kemudian dicari nilai keanggotaan tiap-tiap kelas sehingga didapat kelas terdekat yang merupakan kelas target dari data uji yang baru.
2. Tingkat akurasi pada *Fuzzy K-Nearest Neighbor (FK-NN)* dipengaruhi oleh beberapa faktor antara lain perbandingan antara jumlah data latih dan data uji, nilai *Threshold*, nilai k , dan nilai m . Pada perbandingan data latih dan data uji, akurasi yang dihasilkan mencapai 100% dengan jumlah komposisi data latih lebih banyak dari pada data uji. Untuk pengujian terhadap nilai *Threshold*, dari beberapa percobaan yang dilakukan didapat akurasi tertinggi pada nilai *Threshold* 10 dengan akurasi mencapai 90,8%. Pada pengujian lain menggunakan nilai k , k terbaik adalah pada nilai 7 dengan akurasi yang dihasilkan adalah 82%. Dan pengujian terakhir adalah nilai pangkat m yang menghasilkan akurasi tertinggi pada m 2 yaitu mencapai 88,89% .

7.2 Saran

1. Salah satu penerapan teknologi informasi dan komunikasi di bidang pertanian yaitu dapat digunakan sebagai alat bantu klasifikasi penyakit tanaman kopi dan tidak hanya di lingkup teknologi tetapi juga terhadap lingkungan.
2. Untuk penelitian selanjutnya perlu penambah data dengan kondisi daun normal dan variasi data yang lebih banyak dengan menggunakan nilai fitur yang kompleks untuk objek terkait sehingga menghasilkan akurasi yang lebih tinggi.
3. Menggunakan metode segmentasi lain yang lebih baik untuk memperbaiki kualitas citra sehingga didapatkan hasil klasifikasi yang lebih baik.

DAFTAR PUSTAKA

- Achmad Z.M., Ridok Achmad. 2013. *Penerapan Metode Fuzzy K-NN Pada Klasifikasi Jenis Mangga Berdasarkan Tekstur Daun*. Program Teknologi Informasi Dan Komunikasi. Universitas Brawijaya. Malang.
- Beyan Cigdem, Ogul Hasan. *A Fuzzy K-NN Approach For Cancer Diagnosis With Microarray Gene Expression Data*. Dataset.
- Cahyaningtyas Yunita, Ridok Achmad. 2013. *Penerapan Fuzzy K-Nearest Neighbor Untuk Menentukan Status Evaluasi Kinerja Karyawan*. Program Teknologi Informasi Dan Komunikasi. Universitas Brawijaya. Malang.
- CBI. 2013. *Tailored Market Intelligence: Market Insights For Sustainably-Sourced Coffee From Indonesia*.
- Danang Juniarwoko. 2014. *Pengembangan Industri Pengolahan Kopi*. Tersedia Di <http://www.Academia.Edu/9712453/Pengembangan_Industri_Pengolahan_Kopi> [Diakses 18 September 2015].
- Darwis Safitri. 2014. *Ayo Kenali Penyakit Mati Ujung Pada Kopi*. Tersedia Di <<http://Ditjenbun.Pertanian.Go.Id/Bbpptpmedan/Tinymcpuk/Gambar/File/Ayokenalimatiujung.Pdf>>. [Diakses 18 September 2015].
- Direktorat Jenderal Bina Produksi Perkebunan Departemen Pertanian. 2002. *Musuh Alami, Hama dan Penyakit Tanaman Kopi*. Jakarta.
- Ernawati Dina, Yuliyanto Yudi. 2013. *Fluktuatif Serangan Xyloborus Morigerus Wilayah Kerja Balai Besar Perbenihan Dan Proteksi Tanaman Perkebunan (Bbpptp) Surabaya Pada Triwulan Ii 2013*. Tersedia Di: <<http://Ditjenbun.Pertanian.Go.Id/Bbpptpsurabaya/Tinymcpuk/Gambar/File/6.%20x.%20morigerus%20-%20yudi%20kiki.Pdf>> [Diakses 20 April 2015].
- Katiman. 2015. Wawancara " *Penyakit Tanaman Kopi*" di perkebunan Bumisari. Banyuwangi.
- Kusrini dan Luthfi, E. T., 2009. *Definisi Data Mining*. Jurusan Informatika. STMIK AMIKOM Yogyakarta
- Larose , Daniel T, 2005. *Discovering Knowledge in Data: An Introduction to Data Mining*, John Willey & Sons. Inc.
- Mubarok Achmad, Ridok Achmad, Tyroni Yusi. 2015. *Penerapan Metode Fuzzy K-NN pada Klasifikasi Jenis Mangga Berdasarkan Tekstur Daun*. Universitas Brawijaya. Malang.
- Ndaumanu R.I., Kusrini, Arief M.R. 2014. *Analisis Prediksi Tingkat Pengunduran Diri Mahasiswa dengan Metode K-Nearest Neighbor*. Vol. 1 No. 1.
- Nugraha R.R., 2012. *Penerapan Logika Fuzzy untuk Menghitung Uang Saku Perhari*. Tersedia di <<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2012.2013/Makalah2012/Makalah-IF2091-2012-047.pdf>>. [Diakses 23 september 2015].
- Otsu Nubuyoki. 1979. *A Threshold Selection Method From Gray-Level Histograms*. IEEE transacnons ON SYSTEMS, MAN, AND cybernencs, VOL SMC-9, NO. 1.

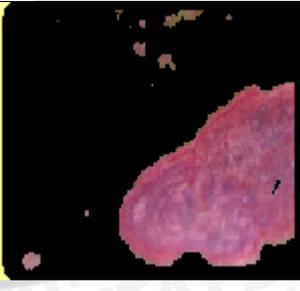
- Prasetyo Eko, 2012. *Fuzzy K-Nearest Neighbor In Every Class Untuk Klasifikasi*.
- Priambodo Apiladosi, Dewi Candra, Triwiratno Anang. 2014. *Implementasi metode k-nearest neighbour untuk identifikasi penyakit tanaman jeruk keprok berdasarkan citra daun*. Universitas Brawijaya. Malang.
- Putra darama. 2010. *Pengolahan citra digital*. Jakarta.
- Rahardjo, Pudji. 2012. *Panduan Budidaya dan Pengolahan Kopi Arabika dan Robusta*. Penebar Swadaya. Jakarta.
- Sabra Abragus. 2011. *Analisis Dan Perancangan Aplikasi Sistem Pakar Dengan Metode Backward Chaining Untuk Mendiagnosis Penyakit Tanaman Kopi*. Universitas Sumatera Utara. Medan.
- Sukama Alfian, Et Al., 2014. *K-Nearest Neighbor Information Retrieval*. Universitas Airlangga Surabaya.
- Sutoyo, T, dkk. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta
- Qing Yao. Guan Zexin. Zhou Yingfeng. 2009. *Application Of support Vector Machine for Rice Diseases using shape and color*. China.
- Yuliyanto Yudhi, Rejeki Tri. 2013. *Fluktuatif serangan xylovorus merlgerus wilayah kerja balai besar perbenihan dan proteksi tanaman perkebunan (BBPPTP) surabaya pada triwulan 11*. 2013. Surabaya.

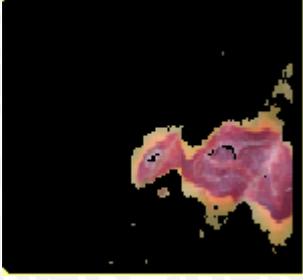
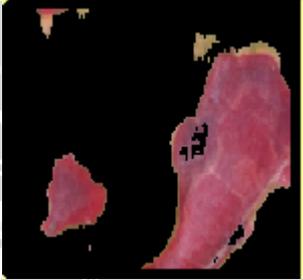
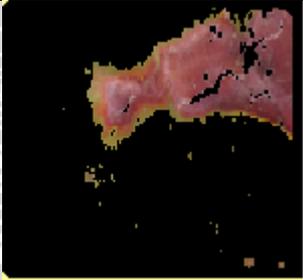


LAMPIRAN A DATA LATIH DAN DATA UJI

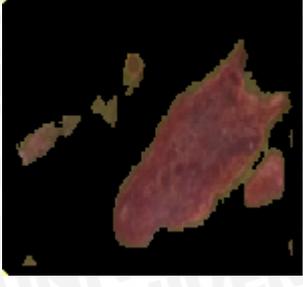
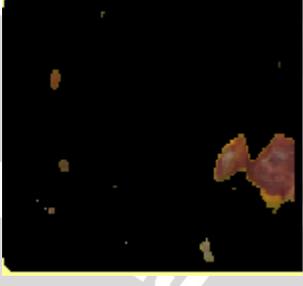
Hasil *Otsu* menggunakan perbandingan data latih dan data uji 90:10

Data Latih

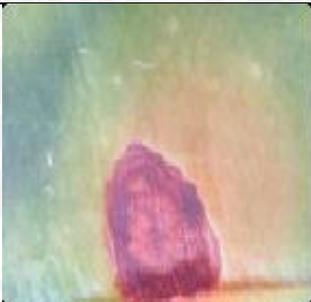
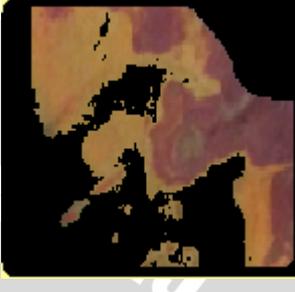
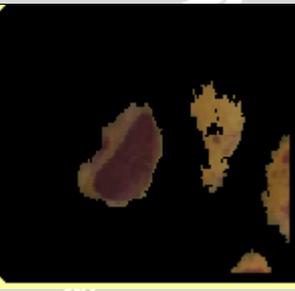
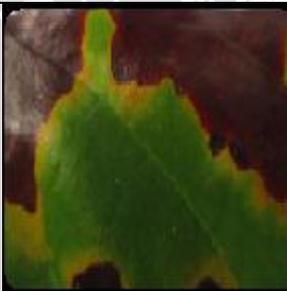
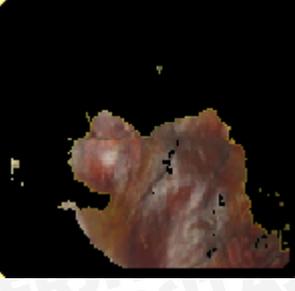
No	Nama Penyakit	Citra Asli	Citra Hasil <i>Otsu</i> +10
1	Karat		
2	Karat		
3	Karat		
4	Karat		

5	Karat		
6	Karat		
7	Karat		
8	Karat		
9	Karat		

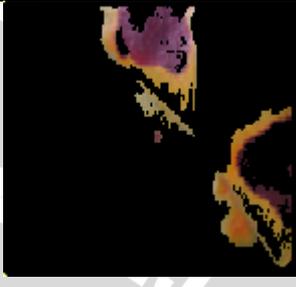
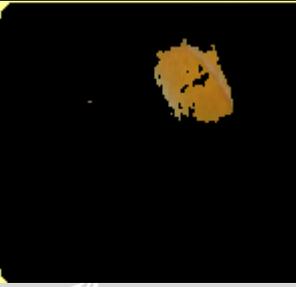
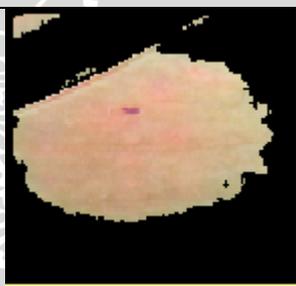


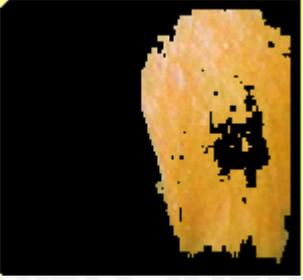
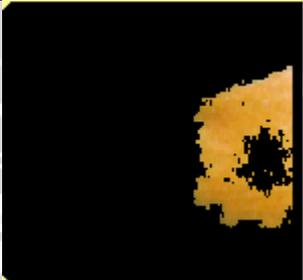
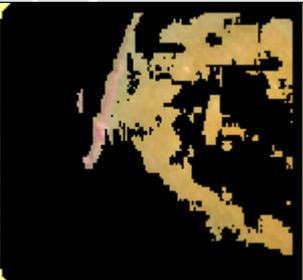
10	Karat		
11	Karat		
12	Karat		
13	Karat		
14	Karat		



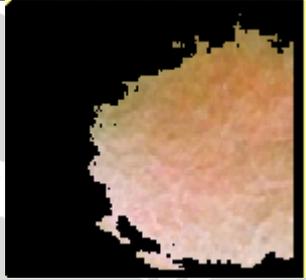
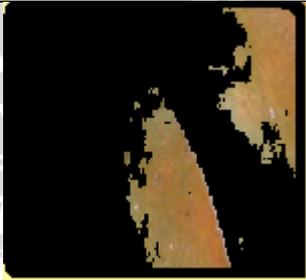
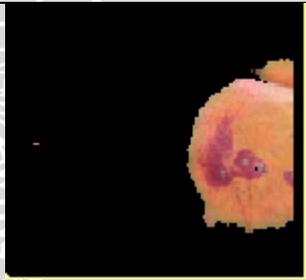
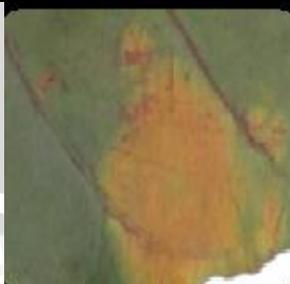
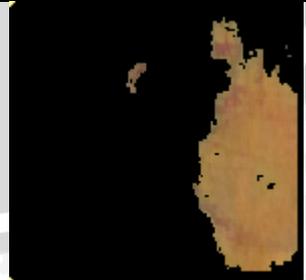
15	Karat		
16	Karat		
17	Karat		
18	Karat		
19	Karat		

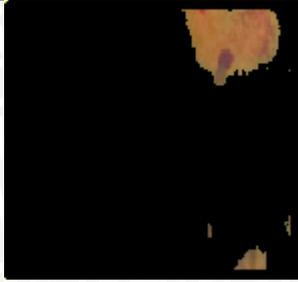
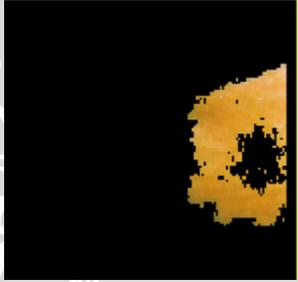


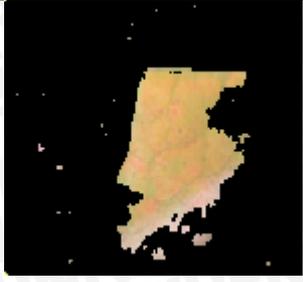
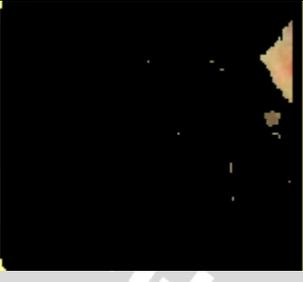
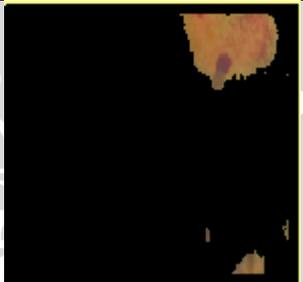
20	Karat		
21	Karat		
22	Bercak		
23	Bercak		
24	Bercak		

25	Bercak		
26	Bercak		
27	Bercak		
28	Bercak		
29	Bercak		



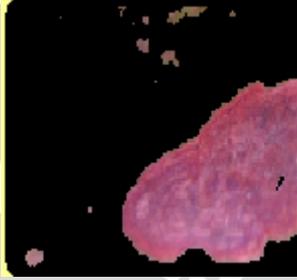
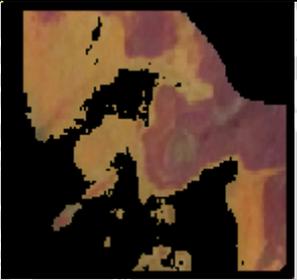
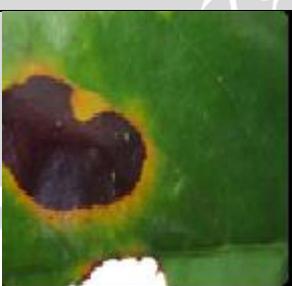
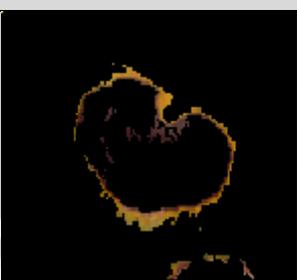
30	Bercak		
31	Bercak		
32	Bercak		
33	Bercak		
34	Bercak		

35	Bercak		
36	Bercak		
37	Bercak		
38	Bercak		
39	Bercak		

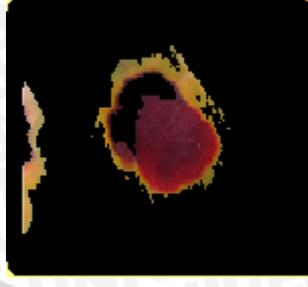
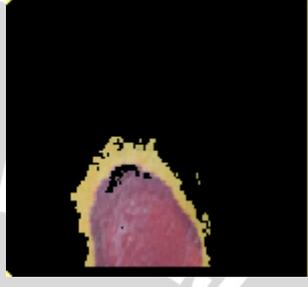
40	Bercak		
41	Bercak		
42	Bercak		

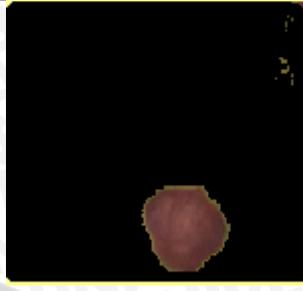
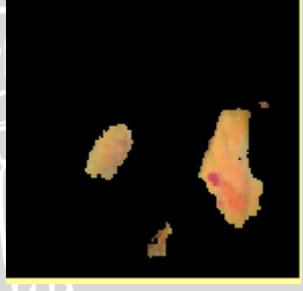
Data Uji

No	Nama Penyakit	Citra Asli	Citra Hasil <i>Otsu +10</i>
1	Karat		
2	Karat		

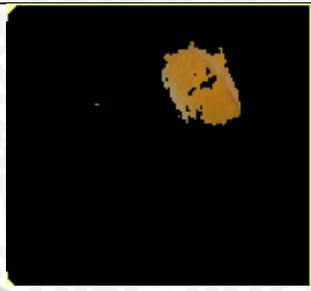
3	Karat		
4	Karat		
5	Karat		
6	Karat		
7	Karat		



8	Karat		
9	Karat		
10	Karat		
11	Bercak		
12	Bercak		

13	Bercak		
14	Bercak		
15	Bercak		
16	Bercak		
17	Bercak		

18 Bercak



LAMPIRAN B LOG SKENARIO PENGUJIAN

1. Pengujian Terhadap Perbandingan Data Latih dan Data Uji

- Perbandingan : 90% : 10%
- Jumlah Data Latih : 54
- Jumlah Data Uji : 6
- Nilai K : 5
- Nilai m : 2

Data Latih

Keterangan :

Target 1 : Penyakit Bercak daun

Target 2 : Penyakit Karat daun

NO	R	G	B	Euclidean Distance	Target
1	179	130	60	123	1
2	210	175	140	26	1
3	201	141	105	75	1
4	188	192	150	35	1
5	218	162	68	95	1
6	210	153	96	73	1
7	220	164	70	92	1
8	202	165	98	68	1
9	227	183	139	21	1
10	208	154	100	70	1
11	191	168	89	79	1
12	169	154	86	96	1
13	210	133	99	82	1
14	165	124	72	122	1
15	160	114	67	133	1
16	177	151	90	90	1
17	220	164	70	92	1

NO	R	G	B	Euclidean Distance	Target
18	232	160	53	110	1
19	207	160	99	68	1
20	207	170	115	50	1
21	188	152	104	74	1
22	122	104	70	158	1
23	199	146	85	88	1
24	151	123	86	121	1
25	179	141	41	134	1
26	210	175	140	26	1
27	218	162	68	95	1
28	151	101	97	128	2
29	136	99	89	142	2
30	174	128	110	91	2
31	171	95	119	113	2
32	162	112	99	114	2
33	191	131	104	85	2
34	161	83	94	138	2
35	138	85	71	159	2
36	188	110	118	94	2
37	108	66	55	197	2
38	116	75	49	190	2
39	161	107	90	123	2
40	110	69	55	194	2
41	129	84	74	163	2
42	210	166	152	26	2
43	137	93	66	158	2
44	109	74	47	196	2
45	79	52	38	231	2
46	106	74	57	192	2
47	116	64	54	194	2



NO	R	G	B	Euclidean Distance	Target
48	136	94	63	160	2
49	137	96	55	163	2
50	151	94	66	151	2
51	166	115	113	103	2
52	71	44	25	248	2
53	171	95	119	113	2
54	129	84	74	163	2

Data Uji beserta Hasil

R	G	B	Target	Hasil
222	188	159	1.00	Benar
158	123	82	1.00	Benar
195	186	133	1.00	Benar
158	123	82	2.00	Benar
129	84	74	2.00	Benar
161	107	90	2.00	Benar

2. Pengujian Terhadap Nilai Peubah Threshold (T)

- Perbandingan : 90% : 10%
- Jumlah Data Latih : 54
- Jumlah Data Uji : 6
- Nilai K : 5
- Nilai m : 2
- Nilai T : 60

Data Latih

Keterangan :

Target 1 : Penyakit Bercak daun

Target 2 : Penyakit Karat daun

NO	R	G	B	Euclidean Distance	Target
1	179	130	60	102	1
2	210	175	140	11	1
3	201	141	105	85	1
4	188	192	150	30	1
5	218	162	68	90	1
6	210	153	96	58	1
7	220	164	70	88	1
8	202	165	98	59	1
9	227	183	139	23	1
10	208	154	100	58	1
11	191	168	89	111	1
12	169	154	86	100	1
13	210	133	99	70	1
14	165	124	72	116	1
15	160	114	67	128	1
16	177	151	90	113	1
17	220	164	70	88	1

NO	R	G	B	Euclidean Distance	Target
18	232	160	53	90	1
19	207	160	99	56	1
20	207	170	115	35	1
21	188	152	104	58	1
22	122	104	70	116	1
23	199	146	85	76	1
24	151	123	86	73	1
25	179	141	41	120	1
26	210	175	140	11	1
27	218	162	68	90	1
28	151	101	97	92	2
29	136	99	89	100	2
30	174	128	110	76	2
31	171	95	119	90	2
32	162	112	99	78	2
33	191	131	104	78	2
34	161	83	94	98	2
35	138	85	71	122	2
36	188	110	118	90	2
37	108	66	55	169	2
38	116	75	49	158	2
39	161	107	90	123	2
40	110	69	55	171	2
41	129	84	74	132	2
42	210	166	152	6	2
43	137	93	66	124	2
44	109	74	47	162	2
45	79	52	38	200	2
46	106	74	57	129	2
47	116	64	54	143	2



NO	R	G	B	Euclidean Distance	Target
48	136	94	63	149	2
49	137	96	55	166	2
50	151	94	66	126	2
51	166	115	113	55	2
52	71	44	25	203	2
53	171	95	119	90	2
54	129	84	74	132	2

Data Uji beserta Hasil

R	G	B	Target	Hasil
222	188	159	2.00	Salah
158	123	82	2.00	Salah
195	186	133	1.00	Benar
158	123	82	2.00	Benar
129	84	74	2.00	Benar
161	107	90	2.00	Benar

3. Pengujian Terhadap Nilai Peubah k

- Perbandingan : 90% : 10%
- Jumlah Data Latih : 54
- Jumlah Data Uji : 6
- Nilai m : 2
- Nilai T : 10
- Nilai K : 10

Data Latih

Keterangan :

Target 1 : Penyakit Bercak daun

Target 2 : Penyakit Karat daun

NO	R	G	B	Euclidean Distance	Target
1	179	130	60	123	1
2	210	175	140	26	1
3	201	141	105	75	1
4	188	192	150	35	1
5	218	162	68	95	1
6	210	153	96	73	1
7	220	164	70	92	1
8	202	165	98	68	1
9	227	183	139	21	1
10	208	154	100	70	1
11	191	168	89	79	1
12	169	154	86	96	1
13	210	133	99	82	1
14	165	124	72	122	1
15	160	114	67	133	1
16	177	151	90	90	1
17	220	164	70	92	1

NO	R	G	B	Euclidean Distance	Target
18	232	160	53	110	1
19	207	160	99	68	1
20	207	170	115	50	1
21	188	152	104	74	1
22	122	104	70	158	1
23	199	146	85	88	1
24	151	123	86	121	1
25	179	141	41	134	1
26	210	175	140	26	1
27	218	162	68	95	1
28	151	101	97	128	2
29	136	99	89	142	2
30	174	128	110	91	2
31	171	95	119	113	2
32	162	112	99	114	2
33	191	131	104	85	2
34	161	83	94	138	2
35	138	85	71	159	2
36	188	110	118	94	2
37	108	66	55	197	2
38	116	75	49	190	2
39	161	107	90	123	2
40	110	69	55	194	2
41	129	84	74	163	2
42	210	166	152	26	2
43	137	93	66	158	2
44	109	74	47	196	2
45	79	52	38	231	2
46	106	74	57	192	2
47	116	64	54	194	2



NO	R	G	B	Euclidean Distance	Target
48	136	94	63	160	2
49	137	96	55	163	2
50	151	94	66	151	2
51	166	115	113	103	2
52	71	44	25	248	2
53	171	95	119	113	2
54	129	84	74	163	2

Data Uji beserta Hasil

R	G	B	Target	Hasil
222	188	159	1.00	Benar
158	123	82	2.00	Salah
195	186	133	2.00	Salah
158	123	82	2.00	Benar
129	84	74	2.00	Benar
161	107	90	2.00	Benar

4. Pengujian Terhadap Nilai m

- Perbandingan : 90% : 10%
- Jumlah Data Latih : 54
- Jumlah Data Uji : 6
- Nilai T : 10
- Nilai K : 5
- Nilai m : 6

Data Latih

Keterangan :

Target 1 : Penyakit Bercak daun

Target 2 : Penyakit Karat daun

NO	R	G	B	Euclidean Distance	Target
1	179	130	60	123	1
2	210	175	140	26	1
3	201	141	105	75	1
4	188	192	150	35	1
5	218	162	68	95	1
6	210	153	96	73	1
7	220	164	70	92	1
8	202	165	98	68	1
9	227	183	139	21	1
10	208	154	100	70	1
11	191	168	89	79	1
12	169	154	86	96	1
13	210	133	99	82	1
14	165	124	72	122	1
15	160	114	67	133	1
16	177	151	90	90	1
17	220	164	70	92	1

NO	R	G	B	Euclidean Distance	Target
18	232	160	53	110	1
19	207	160	99	68	1
20	207	170	115	50	1
21	188	152	104	74	1
22	122	104	70	158	1
23	199	146	85	88	1
24	151	123	86	121	1
25	179	141	41	134	1
26	210	175	140	26	1
27	218	162	68	95	1
28	151	101	97	128	2
29	136	99	89	142	2
30	174	128	110	91	2
31	171	95	119	113	2
32	162	112	99	114	2
33	191	131	104	85	2
34	161	83	94	138	2
35	138	85	71	159	2
36	188	110	118	94	2
37	108	66	55	197	2
38	116	75	49	190	2
39	161	107	90	123	2
40	110	69	55	194	2
41	129	84	74	163	2
42	210	166	152	26	2
43	137	93	66	158	2
44	109	74	47	196	2
45	79	52	38	231	2
46	106	74	57	192	2
47	116	64	54	194	2

NO	R	G	B	Euclidean Distance	Target
48	136	94	63	160	2
49	137	96	55	163	2
50	151	94	66	151	2
51	166	115	113	103	2
52	71	44	25	248	2
53	171	95	119	113	2
54	129	84	74	163	2

Data Uji beserta Hasil

R	G	B	Target	Hasil
222	188	159	1.00	Benar
158	123	82	2.00	Salah
195	186	133	2.00	Salah
158	123	82	2.00	Benar
129	84	74	2.00	Benar
161	107	90	2.00	Benar