

**KLASIFIKASI CITRA MAKANAN MENGGUNAKAN HSV COLOR
MOMENT DAN HARALICK FEATURE EXTRACTION DENGAN
NAIVE BAYES CLASSIFIER**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Gabriel Mulyawan

NIM: 155150201111135



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

KLASIFIKASI CITRA MAKANAN MENGGUNAKAN HSV COLOR MOMENT DAN HARALICK FEATURE EXTRACTION DENGAN NAIVE BAYES CLASSIFIER

SKRIPS

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Gabriel Mulyawan
NIM: 155150201111135

Skripsi ini telah diuji dan dinyatakan lulus pada

17 Juni 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1

Dosen Pembimbing 2

Yuita Arum Sari, S.Kom., M.Kom.

NIK: 201503 890520 2 001

Muhammad Tanzil Furqon, S.Kom., M.CompSc.

NIP: 19670801 199203 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 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 referensi.

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, 16 Mei 2019

Gabriel mulyawan

NIM: 155150201111135

PRAKATA

Segala puji dan syukur bagi Tuhan Yang Maha Esa atas rahmat, berkah, dan petunjuk-Nya kepada peneliti, sehingga peneliti dapat menyusun skripsi dengan judul "" yang menjadi syarat untuk dapat menyelesaikan Program Sarjana (S1) pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.

Dalam penyusunan dan penelitian skripsi ini, peneliti mendapat rintangan dan hambatan namun peneliti dapat melalui dengan baik berkat adanya dukungan dan bantuan dari beberapa pihak, yaitu bantuan moral, spiritual, dan materi. Maka dari itu, peneliti hendak menyampaikan ucapan terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Heru Nurwarsito, Ir., M.Kom. selaku Wakil Ketua I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Bapak Agus Wahyu Widodo, S.T, M.Cs. selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya Malang.
5. Ibu Yuita Arum Sari, S.Kom, M.Kom selaku dosen pembimbing I yang telah memberikan bimbingan dan pengarahan kepada peneliti, sehingga dapat menyelesaikan skripsi ini.
6. Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc. selaku dosen pembimbing II yang telah memberikan bimbingan dan pengarahan kepada peneliti, sehingga dapat menyelesaikan skripsi ini.
7. Kedua orang tua dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian, dan kesabaran memberikan dukungan kepada peneliti, serta selalu berdoa supaya skripsi ini dapat peneliti selesaikan.
8. Seluruh civitas akademika Jurusan Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya, dan untuk teman-teman Program Studi Teknik Informatika Angkatan 2015, yaitu Rilinka, Rizky, Ian, Yosua, Usman, Jimmy, Wafiq yang telah memberi dukungan dan bantuan kepada peneliti selama penyelesaian skripsi ini dan menemani selama menempuh studi di Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
9. Dan kepada seluruh pihak yang tidak dapat disebutkan satu persatu yang telah membantu dan memberi dukungan.

Peneliti mohon maaf apabila dalam penyusunan skripsi ini masih terdapat banyak kekurangan, sehingga peneliti sangat menerima saran dan kritik yang membangun. Peneliti berharap semoga skripsi ini dapat memberikan manfaat

bagi pihak yang membutuhkan untuk dapat melakukan penelitian-penelitian berikutnya.

Malang, 24 April 2019

Gabriel Mulyawan

gabrielmulyawan@student.ub.ac.id



ABSTRAK

Gabriel Mulyawan, Klasifikasi Citra Makanan Menggunakan HSV Color Moment Dan Haralick Feature Extraction Dengan Naive Bayes Classifier

Pembimbing: Yuita Arum Sari, S.Kom, M.Kom dan Muhammad Tanzil Furqon, S.Kom, M.CompSc

Sebagai makhluk hidup, manusia memiliki kebutuhan untuk bertahan hidup. Salah satu kebutuhan manusia, yaitu makanan. Makanan dapat memberikan gizi seperti Karbohidrat, Mineral, Protein, Lemak, dan Vitamin sebagai sumber energi dan daya tahan tubuh.

Pada dasarnya, makanan dapat dengan mudah dikenali secara visual dengan indera penglihatan manusia. Namun berbeda dengan komputer yang membutuhkan pengenalan atau ekstraksi fitur dari objek makanan untuk dilakukan klasifikasi. Ekstraksi fitur yang digunakan adalah *HSV Color Moment* untuk fitur warna dan *Haralick* untuk fitur tekstur. Kemudian hasil dari ekstraksi fitur tersebut akan dilakukan klasifikasi menggunakan metode *Naive Bayes classifier*. Data yang digunakan adalah berupa data primer yang diambil menggunakan kamera *smartphone* yang terdiri dari 276 citra makanan. Dalam penelitian ini dilakukan dua kali pengujian, yaitu pengujian perbandingan jumlah data latih dan data uji dan pengujian fitur yang digunakan.

Berdasarkan hasil pengujian perbandingan jumlah data latih dan data uji menggunakan *K-Fold Cross Validation* dihasilkan hasil rata-rata akurasi terbaik sebesar 61,95% dengan menggunakan 166 citra data latih dan 110 citra data uji. Kemudian dari hasil pengujian fitur, pengujian yang hanya menggunakan fitur *HSV Color Moment* menghasilkan akurasi sebesar 57,66%. Pengujian yang hanya menggunakan fitur *Haralick* menghasilkan akurasi sebesar 36,67%. Pengujian gabungan antara dua fitur *HSV Color Moment* dan *Haralick* menghasilkan akurasi lebih baik dibandingkan hanya menggunakan fitur tekstur dengan akurasi 56,33%. Teknik pemrosesan gambar menggunakan ekstraksi fitur *HSV Color Moment* dan *Haralick* dapat digunakan untuk klasifikasi citra makanan menggunakan metode klasifikasi Naive Bayes.

Kata kunci: Klasifikasi, *HSV Color Moment*, *Haralick*, *Naive Bayes Classifier*

ABSTRACT

Gabriel Mulyawan, Classification of Food Images with HSV Color Moment and Haralick Feature Extraction using Naive Bayes Classifier

Supervisors: Yuita Arum Sari, S.Kom, M.Kom dan Muhammad Tanzil Furqon, S.Kom, M.CompSc

As living things, humans need to survive. One of the basic need human's bodies require to survive is food. Foods provide nutrients that contain carbohydrates, protein, minerals, fats, and vitamins for boosting endurance.

Basically, foods can be easily identified with human's eyes. But it is not like the brain-computer that require the introduction or features extraction from food objects for classification. The features extraction used are HSV Color Moment for color features and Haralick for texture features. Then, the results of the features extraction will be classified using the Naive Bayes classifier method. The data set used are based of the primary data that contains pictures and the pictures were taken by the smartphone camera consist of 276 foods images.. This research uses 2 testing processes, that are the comparison of the amount of the training data and testing data, and the testing of the used features.

Based on the testing of the comparison of the amount of the training data and the training data using K-Fold Cross Validation, it showed that the best accuracy is 61,95% that using 166 training data images and 110 training data images. Then, the accuracy from the features test that was just using the HSV Color Moment feature is about 57,66%. The accuracy from test that using the Haralick feature is 36,67%. The accuracy from the combination of 2 features of the HSV Color Moment and Haralick are better than only using the texture features with the 56,33% accuracy. The image processing technique using HSV Color Moment and Haralick features extraction can be used for foods image classification using the Naive Bayes classifier method.

Keywords: Classification, HSV Color Moment, Haralick, Naive Bayes Classifier

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Pengolahan Citra Digital.....	4
2.3 Ekstraksi Fitur.....	5
2.3.1 Fitur Warna.....	5
2.3.2 Fitur Tekstur	6
2.4 Naive Bayes Classifier.....	8
2.5 Confusion Matrix.....	8
BAB 3 METODOLOGI PENELITIAN	10
3.1 Tipe Penelitian	10
3.2 Metode Penelitian.....	10
3.3 Lokasi Penelitian	12
3.4 Pengumpulan Data.....	12
3.5 Peralatan Pendukung	14
BAB 4 PERANCANGAN	15

4.1.1 <i>Pre-Processing</i>	16
4.1.2 Ekstraksi Fitur <i>HSV Color Moment</i>	26
4.1.3 Ekstraksi Fitur <i>Haralick</i>	29
4.1.4 Klasifikasi <i>Naive Bayes</i>	38
4.2 Perhitungan Manual	40
4.2.1 Perhitungan <i>pre-processing</i>	40
4.2.2 Perhitungan <i>HSV Color Moment</i>	46
4.2.3 Perhitungan <i>Haralick</i>	48
4.2.4 Perhitungan <i>Naive Bayes</i>	64
4.3 Perhitungan Evaluasi Kinerja Sistem.....	71
BAB 5 IMPLEMENTASI	73
5.1 Batasan Implementasi.....	73
5.2 Implementasi Algoritme.....	73
5.2.1 Algoritme <i>Preprocessing</i> Citra	73
5.2.2 Algoritme Ekstraksi Fitur <i>HSV Color Moment</i>	74
5.2.3 Algoritme Ekstraksi Fitur <i>Haralick</i>	77
Penjelasan <i>Source Code</i> :.....	83
5.2.4 Algoritme Klasifikasi <i>Naive Bayes</i>	85
BAB 6 HASIL PENGUJIAN DAN ANALISIS	90
6.1 Pengujian Jumlah Data Latih dan Data Uji Terhadap Akurasi	90
6.2 Pengujian Metode <i>HSV Color Moment</i> , <i>Haralick</i> dan <i>Naive Bayes Classifier</i> Terhadap Akurasi.....	98
BAB 7 PENUTUP	105
7.1 Kesimpulan.....	105
7.2 Saran	105
DAFTAR REFERENSI.....	106
LAMPIRAN A HASIL PENGUJIAN FITUR	107

DAFTAR TABEL

Tabel 4. 1 Nilai <i>Red</i> dari citra makanan yang diresize	40
Tabel 4. 2 Nilai <i>Green</i> dari citra makanan yang diresize.....	40
Tabel 4. 3 Nilai <i>Blue</i> dari citra makanan yang diresize	41
Tabel 4. 4 Nilai <i>Saturation</i> Hasil Perhitungan Ubah ke <i>Saturation</i>	41
Tabel 4. 5 <i>Gaussian Blur Kernel</i>	42
Tabel 4. 6 Perhitungan <i>Gaussian Blurring</i>	42
Tabel 4. 7 Hasil Perhitungan <i>Gaussian Blurring</i>	42
Tabel 4. 8 Hasil Perhitungan <i>Thresholding</i>	43
Tabel 4. 9 <i>Structuring Element</i>	44
Tabel 4. 10 Perhitungan Erosi.....	44
Tabel 4. 11 Hasil Perhitungan Erosi.....	44
Tabel 4. 12 Perhitungan Dilasi.....	45
Tabel 4. 13 Hasil Perhitungan Dilasi	45
Tabel 4. 14 Nilai <i>Hue</i>	46
Tabel 4. 15 Nilai <i>Saturation</i>	47
Tabel 4. 16 Nilai <i>Value</i>	47
Tabel 4. 17 Mean <i>Hue</i> , <i>Saturation</i> , dan <i>Value</i>	48
Tabel 4. 18 Standar Deviasi <i>Hue</i> , <i>Saturation</i> , dan <i>value</i>	48
Tabel 4. 19 <i>Skewness Hue</i> , <i>Saturation</i> , dan <i>value</i>	48
Tabel 4. 20 Perhitungan Pasangan Matriks GLCM dengan arah 0°	48
Tabel 4. 21 Matriks GLCM dengan arah 0°	49
Tabel 4. 22 <i>Transpose</i> matriks GLCM dengan arah 0°	49
Tabel 4. 23 Hasil Jumlah matriks GLCM dan <i>transpose</i> dengan arah 0°	50
Tabel 4. 24 Normalisasi Matriks GLCM dengan arah 0°	50
Tabel 4. 25 Perhitungan Pasangan Matriks GLCM dengan arah 45°	51
Tabel 4. 26 Matriks GLCM dengan arah 45°	51
Tabel 4. 27 <i>Transpose</i> matriks GLCM dengan arah 45°	51
Tabel 4. 28 Hasil Jumlah matriks GLCM dan <i>transpose</i> dengan arah 45°	52
Tabel 4. 29 Normalisasi Matriks GLCM dengan arah 45°	53

Tabel 4. 30 Perhitungan Pasangan Matriks GLCM dengan arah 90°	53
Tabel 4. 31 Matriks GLCM dengan arah 90°	53
Tabel 4. 32 <i>Transpose</i> matriks GLCM dengan arah 90°	54
Tabel 4. 33 Hasil Jumlah matriks GLCM dan <i>transpose</i> dengan arah 90°	54
Tabel 4. 34 Normalisasi Matriks GLCM dengan arah 90°	55
Tabel 4. 35 Perhitungan Pasangan Matriks GLCM dengan arah 135°	56
Tabel 4. 36 Matriks GLCM dengan arah 135°	56
Tabel 4. 37 <i>Transpose</i> matriks GLCM dengan arah 135°	56
Tabel 4. 38 Hasil Jumlah matriks GLCM dan <i>transpose</i> dengan arah 135°	57
Tabel 4. 39 Normalisasi Matriks GLCM dengan arah 135°	57
Tabel 4. 40 Fitur Data Latih	64
Tabel 4. 41 Mean Fitur Data Latih	66
Tabel 4. 42 Standar Deviasi Fitur Data Latih	66
Tabel 4. 43 Fitur Data Uji	67
Tabel 4. 44 <i>Likelihood</i> Data Uji No 1	68
Tabel 4. 45 Posterior Probability Data Uji No 1	70
Tabel 4. 46 Hasil Klasifikasi Citra	70
Tabel 4. 47 Evaluasi <i>Confusion Matrix</i>	71
Tabel 5. 1 <i>Source Code</i> Algoritme <i>Preprocessing</i> Citra	73
Tabel 5. 2 <i>Source Code</i> Algoritme Ekstraksi Fitur HSV <i>Color Moment</i>	75
Tabel 5. 3 <i>Source Code</i> Algoritme Ekstraksi Fitur Haralick	78
Tabel 5. 4 <i>Source Code</i> Algoritme Klasifikasi Naive Bayes	85
Tabel 6. 1 Perbandingan Jumlah Data Latih dan Data Uji	90
Tabel 6. 2 <i>Confusion Matrix</i> 240 Data latih dan 36 Data Uji	90
Tabel 6. 3 <i>Confusion Matrix</i> 228 Data latih dan 48 Data Uji	92
Tabel 6. 4 <i>Confusion Matrix</i> 216 Data latih dan 60 Data Uji	93
Tabel 6. 5 <i>Confusion Matrix</i> 204 Data latih dan 72 Data Uji	94
Tabel 6. 6 <i>Confusion Matrix</i> 192 Data latih dan 84 Data Uji	95
Tabel 6. 7 Tabel Akurasi Perbandingan Data Latih dan Uji	97
Tabel 6. 8 Akurasi Pengujian <i>K-Fold Cross Validation</i>	98
Tabel 6. 9 Hasil Klasifikasi Pengujian HSV <i>Color Moment</i>	99
Tabel 6. 10 Hasil Klasifikasi Pengujian HSV <i>Color Moment</i> (Lanjutan)	100

Tabel 6. 11 Perbandingan Nilai Fitur Pada HSV <i>Color Moment</i>	100
Tabel 6. 12 Hasil Klasifikasi Pengujian <i>Hralick</i>	101
Tabel 6. 13 Hasil Klasifikasi Pengujian <i>Hralick</i> (Lanjutan).....	102
Tabel 6. 14 Perbandingan Nilai Fitur Pada <i>Hralick</i>	102
Tabel 6. 15 Perbandingan Nilai Fitur Pada <i>Hralick</i> (Lanjutan).....	103
Tabel 6. 16 Hasil Akurasi Pengujian Ekstraksi Fitur	104
Tabel 6. 17 Hasil Pengujian Fitur <i>Color Moment</i>	106
Tabel 6. 18 Hasil Pengujian Fitur <i>Hralick</i>	115
Tabel 6. 19 Hasil Penjajian Fitur <i>Color Moment</i> dan <i>Hralick</i>	123



DAFTAR GAMBAR

Gambar 4.1 Diagram Alir Sistem	16
Gambar 4.2 Diagram alir <i>Pre-Processing</i>	17
Gambar 4.3 Diagram alir ubah ke <i>saturation</i>	19
Gambar 4.4 Diagram alir <i>Image Enchancement</i>	19
Gambar 4.5 Diagram alir <i>Gaussian Blur</i>	20
Gambar 4.6 Diagram alir <i>Thresholding</i>	21
Gambar 4.7 Diagram alir Morfologi Citra	22
Gambar 4.8 Diagram alir Erosi	24
Gambar 4.9 Diagram alir Dilasi.....	25
Gambar 4.10 Diagram Alir <i>Masking</i>	26
Gambar 4.11 Diagram alir <i>HSV Color Moment</i>	28
Gambar 4.12 Diagram alir Ekstraksi Fitur <i>Haralick</i>	29
Gambar 4.13 Diagram alir Matriks GLCM 0°	31
Gambar 4.14 Diagram alir Matriks GLCM 45°	33
Gambar 4.15 Diagram alir Matriks GLCM 90°	35
Gambar 4.16 Diagram alir Matriks GLCM 135°	36
Gambar 4.17 Diagram alir Ekstraksi ciri <i>Haralick</i>	38
Gambar 4.18 Diagram Alir <i>Naive Bayes Classifier</i>	39
Gambar 4.19 Citra Asli.....	40
Gambar 4.20 Citra <i>saturation</i>	41
Gambar 4.21 Citra <i>Gaussian Blur</i>	43
Gambar 4.22 Citra <i>Thresholding</i>	43
Gambar 4.23 Citra Erosi	45
Gambar 4.24 Citra Dilasi.....	46
Gambar 4.25 Citra Setelah <i>Pre-processing</i>	46
Gambar 6. 1 Grafik Akurasi Perbandingan Data Latih dan Uji.....	97
Gambar 6. 2 Citra Tidak Terklasifikasi Dengan Baik Pada <i>HSV Color Moment</i> ...	100
Gambar 6. 3 Citra Tidak Terklasifikasi Pada Fitur <i>Haralick</i>	102
Gambar 6. 4 Hasil Segmentasi Citra yang Kurang Baik.....	104
Gambar 6. 5 Grafik Akurasi Pengujian Metode Ekstraksi Fitur	104

DAFTAR LAMPIRAN

LAMPIRAN A HASIL PENGUJIAN FITUR	106
A.1 Fitur Color Moment.....	107
A.2 Fitur Haralick	115
A.3 Fitur Color Moment dan Haralick.....	123



BAB 1 PENDAHULUAN

1.1 Latar belakang

Makanan merupakan bahan yang dapat berasal dari hewan maupun tumbuhan yang dimakan oleh makhluk hidup. Makanan dapat memberikan gizi kepada tubuh sehingga kebutuhan tubuh akan gizi seperti Karbohidrat, Protein, Lemak, dan Vitamin dapat terpenuhi. Manusia sebagai makhluk hidup membutuhkan makan untuk mendapatkan sumber energi, mempertahankan ketahanan tubuh dalam menghadapi serangan penyakit, dan tumbuh kembang. Oleh karena itu, makanan merupakan kebutuhan primer atau kebutuhan yang harus dipenuhi oleh setiap manusia untuk bertahan hidup.

Dalam mengenali jenis makanan, manusia dapat dengan mudah untuk mengenali citra jenis makanan melalui indra. Berbeda dengan komputer yang tidak dapat secara langsung mengenali jenis makanan. Menurut standar makanan di dunia sulit mengenali nama makanan ketika berada di suatu wilayah yang berbeda (Ireland & Moller, 2016), sehingga perlu dilakukan pengenalan terhadap karakteristik makanan melalui pengolahan citra digital dan klasifikasi. Klasifikasi merupakan pengelompokan data sesuai dengan aturan sehingga menghasilkan hubungan antara nilai atribut dengan kumpulan data latih (Hadi, et al., 2018).

Untuk dapat dilakukan klasifikasi dari sebuah citra makanan, diperlukan pengenalan terhadap karakteristik dari sebuah makanan. Beberapa jenis karakteristik tersebut antara lain warna, bentuk, dan tekstur. Fitur warna yang digunakan adalah gabungan dari *Hue*, *Saturation*, dan *Value* (HSV), dimana *Hue* merupakan nilai *pure* dari suatu warna yang direpresentasikan dari 0 sampai 360 derajat. *Saturation* menunjukkan kemurnian dari nilai *pure color*, sedangkan *Value* merupakan ukuran seberapa besar kecerahan suatu warna. *Color Moment* adalah representasi karakteristik citra warna, pembanding citra untuk mendapatkan *Color Similarity* sebuah citra yang disimpan dalam sebuah *database* citra (Halim, et al., 2013).

Dalam pengenalan karakteristik makanan melalui fitur tekstur dapat digunakan *Haralick Features*, *Grey-Level Co-occurrence Matrix* (GLCM) atau *Local Binary Pattern* (LBP). Pada penelitian sebelumnya yang berjudul *SVM and Haralick Features for Classification of High Resolution Satellite Image from Urban Areas* menggunakan *Haralick Features* sebagai pengenal tekstur dari citra sebuah citra satelit karena kelebihannya yaitu menggunakan matriks dalam perhitungan sehingga mudah untuk diinterpretasikan (Bekkari, et al., 2012).

Dalam penelitian sebelumnya yang berjudul *Klasifikasi Daun Herbal Menggunakan Metode Naive Bayes dan K-Nearest neighbor*. Dilakukan pengujian dua metode terhadap 120 citra daun herbal, yang terdiri dari 96 citra yang digunakan untuk data latih dan 24 citra digunakan sebagai data uji. Menghasil tingkat akurasi 75% untuk metode *Naive bayes* dan 70,83% untuk metode *K-Nearest Neighbor*. Untuk penggunaan data latih yang lebih sedikit metode *Naive Bayes* menghasilkan akurasi yang lebih baik. *Naive Bayes* merupakan sebuah

metode yang berasal dari teorema *Bayes* yaitu memprediksi peluang yang di masa depan dengan berdasarkan peluang yang ada pada masa sebelumnya (Liantoni & Nugroho, 2015). Hasil klasifikasi dengan menggunakan kumpulan data yang sedikit disarankan menggunakan Klasifikasi *Naive Bayes* dibandingkan menggunakan *Logistic Regression* (Pundlik, 2016).

Untuk mengenali citra jenis makanan, maka dalam penelitian ini dibuat aplikasi untuk klasifikasi berdasarkan citra makanan menggunakan *HSV Color Moment* untuk ekstraksi fitur berdasarkan warna dan *Haralick* untuk ekstraksi fitur berdasarkan tekstur dengan menggunakan klasifikasi *Naive Bayes*.

1.2 Rumusan masalah

1. Bagaimana tingkat akurasi dalam melakukan klasifikasi citra makanan menggunakan *HSV Color Moment* dan *Haralick Feature Extraction* menggunakan klasifikasi *Naive Bayes* dengan menggunakan perbandingan jumlah data latih dan data uji berbeda?
2. Bagaimana pengaruh metode ekstraksi fitur *HSV Color Moment* dan *Haralick Feature Extraction* terhadap akurasi klasifikasi citra makanan dengan menggunakan klasifikasi *Naive Bayes*?

1.3 Tujuan

1. Mengetahui tingkat akurasi klasifikasi citra makanan *HSV Color Moment* dan *Haralick Feature Extraction* dalam melakukan klasifikasi citra makanan dengan menggunakan klasifikasi *Naive Bayes* dengan jumlah data latih dan data uji yang berbeda.
2. Mengetahui pengaruh metode Ekstraksi fitur *HSV Color Moment* dan *Haralick Feature Extraction* terhadap akurasi, dalam melakukan klasifikasi citra makanan dengan menggunakan klasifikasi *Naive Bayes*.

1.4 Manfaat

Manfaat yang diharapkan dari dilakukannya penelitian ini adalah:

1. Menyediakan sistem yang dapat melakukan klasifikasi terhadap jenis makan berdasarkan fitur warna dan tekstur menggunakan metode *Naive Bayes*.
2. Mendapatkan pemahaman tentang metode *haralick*, *HSV Color moment* dan menggunakan metode klasifikasi *Naive Bayes* yang digunakan pada penelitian ini.

1.5 Batasan masalah

Beberapa batasan masalah yang terjadi pada penelitian ini adalah:

1. Kumpulan data yang digunakan untuk klasifikasi berupa makanan tunggal atau makanan yang sudah dipisah sesuai jenisnya dan berupa makanan padat.

2. Pengambilan kumpulan data menggunakan kamera *smartphone*
3. Kumpulan data yang digunakan berupa 12 jenis kelas makanan.

1.6 Sistematika pembahasan

Sistematika penulisan pada penelitian ini terdiri dari tujuh bab, antara lain:

BAB I	PENDAHULUAN
	Pada bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan penelitian.
BAB II	TINJAUAN PUSTAKA
	Pada bab ini membahas mengenai dasar teori dan referensi yang dibutuhkan untuk pemahaman permasalahan yang dibahas pada penelitian ini. Teori-teori yang dibahas pada bab ini adalah pengolahan citra digital <i>HSV Color Moment</i> , <i>Haralick feature extraction</i> , dan klasifikasi <i>Naive bayes</i> .
BAB III	METODOLOGI PENELITIAN
	Pada bab ini membahas metode yang digunakan dan langkah-langkah yang diambil dalam penelitian ini meliputi kajian pustaka, pengumpulan data, perancangan sistem, implementasi, pengujian dan analisis, dan pengambilan keputusan.
BAB IV	PERANCANGAN
	Pada bab ini membahas mengenai perancangan algoritme dan perhitungan manual dari sistem.
BAB V	IMPLEMENTASI
	Pada bab ini membahas mengenai implementasi program sistem dan penerapan metode yang digunakan.
BAB VI	HASIL PENGUJIAN DAN ANALISIS
	Pada bab ini membahas mengenai hasil pengujian dan analisis terhadap kinerja sistem.
BAB VII	KESIMPULAN
	Pada bab ini membahas mengenai kesimpulan dan saran yang didapatkan selama penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian pertama dilakukan oleh Febri Liantoni dengan judul *Klasifikasi Daun Herbal Menggunakan Metode Naive Bayes Classifier dan K-Nearest Neighbor*. Penelitian ini menggunakan dua ekstraksi fitur yaitu fitur geometri dan fitur *Invariant Moment* untuk dilakukan klasifikasi. Tujuan dari penelitian ini adalah membandingkan metode klasifikasi *Naive Bayes* dan *K-Nearest Neighbor*. Dari hasil pengujian, didapatkan hasil akurasi tertinggi ketika menggunakan metode *Naive Bayes* dengan akurasi sebesar 75% dibandingkan dengan metode *K-Nearest Neighbor* yang hanya menghasilkan tingkat akurasi sebesar 70,83%.

Pada penelitian kedua yang dilakukan oleh Arwin Halim dengan judul *Aplikasi Image Retrieval Menggunakan Kombinasi Metode Color Moment dan Gabor Texture*. Tujuan dari penelitian ini adalah membandingkan algoritma *Color Moment (region)*, *Color Moment (whole)*, *Gabor Texture*, *Color Moment (region)* dengan *Gabor texture*, *Color moment (whole)* dengan *Gabor Texture*. Berdasarkan hasil percobaan, menunjukkan kombinasi dari algoritma *Color Moment (region)* dan *Gabor Texture* mendapatkan hasil pencarian gambar yang lebih tinggi dibandingkan dengan kombinasi algoritma lain dengan rata-rata presisi 51,79% dan rata-rata *recall* 51,60%.

Penelitian ketiga membahas tentang kombinasi antara algoritma klasifikasi *Support Vector Machine* (SVM) dengan *Spectral Information* dan *Haralick Features*. Penelitian ini membandingkan antara fitur ekstraksi tekstur secara *Spectral Information* dan *Spatial Information* maupun gabungan dari kedua fitur. Hasil dari penelitian ini menunjukkan gabungan antara *Spectral Information* dan *Spectral information* menghasilkan akurasi yang paling baik yaitu 93,68%.

Pada penelitian selanjutnya membandingkan dari sensitivitas data pinjaman konsumen yang menggunakan *logistic regression* dengan *gaussian naive bayes*. Dua variabel yang digunakan antara *good borrower* dan *bad borrower*, selain itu digunakan 1000 dan 500000 kumpulan data yang dibagi lagi menjadi data latih dan data validasi. Dari hasil penelitian menunjukkan penggunaan kumpulan data yang lebih sedikit, *gaussian naive bayes* menghasilkan sensitivitas yang lebih baik dibandingkan dengan *logistic regression* dengan nilai 76.08 dan 73.69.

2.2 Pengolahan Citra Digital

Citra Digital adalah gambar dengan intensitas cahaya pada bidang dua dimensi bernama piksel yang direpresentasikan secara numerik (Kusumanto & Tompunu, 2011). Piksel-piksel yang ada tersebut merepresentasikan ruang warna sehingga menghasilkan citra (Wu, et al., 2013). Pengolahan citra digital merupakan ilmu yang mempelajari tentang cara mengolah suatu citra dengan bantuan komputer.

2.3 Ekstraksi Fitur

Ekstraksi fitur adalah proses untuk mengenali dan menghasilkan informasi morfologi yang berasal dari nilai-nilai fitur yang berasal citra biner yang telah dilakukan deteksi tepi (Liantoni & Nugroho, 2015). Terdapat beberapa fitur yang dapat diekstrak dari sebuah citra makanan antara lain fitur warna dan fitur tekstur. Yang nantinya hasil dari nilai ekstraksi akan diklasifikasikan dan dilakukan analisis.

2.3.1 Fitur Warna

Warna merupakan fitur penting dari sebuah citra digital karena warna dapat langsung diterima / dilihat oleh mata manusia (Loesdau, et al., 2014). Nilai warna dari setiap *pixel* nanti akan digunakan sebagai fitur sesuai dengan ruang warna yang digunakan. Beberapa sistem ruang warna yang digunakan untuk merepresentasikan warna antara lain RGB, HSV, HSI, YUV, dan CIELAB.

Hue, Saturation, dan Value (HSV)

Hue, Saturation, dan Value (HSV), dimana *Hue* merupakan nilai *pure* dari suatu warna atau disebut dengan warna primer yang direpresentasikan dari 0 sampai 360 derajat. 0 derajat adalah warna merah, 60 derajat adalah warna kuning, 120 derajat adalah warna hijau, 180 derajat adalah warna cyan, 240 derajat adalah warna biru, dan 300 derajat adalah warna magenta. *Saturation* menunjukkan kemurnian dari nilai *pure color*. *Saturation* dapat benilai 0 untuk menunjukkan warna abu-abu sampai 1 yang menunjukkan nilai *pure* dari warna primer. Sedangkan *Value* merupakan ukuran seberapa besar kecerahan suatu warna (Kryjak, et al., 2018). Cara Konversi nilai RGB ke dalam HSV:

$$H = \begin{cases} 0, & \text{JIKA Max = Min} \\ \frac{\left(60^\circ \times \frac{(G-B)}{(MAX-MIN)} + 0^\circ\right)}{360}, & \text{JIKA Max = R dan } G \geq B \\ \frac{\left(60^\circ \times \frac{(G-B)}{(MAX-MIN)} + 360^\circ\right)}{360}, & \text{JIKA Max = R dan } G < B \\ \frac{\left(60^\circ \times \frac{(B-R)}{(MAX-MIN)} + 120^\circ\right)}{360}, & \text{JIKA Max = G} \\ \frac{\left(60^\circ \times \frac{(R-G)}{(MAX-MIN)} + 240^\circ\right)}{360}, & \text{JIKA Max = B} \end{cases} \quad (2.1)$$

$$S = \begin{cases} 0, & \text{Jika } Max = 0 \\ 1 - \frac{MIN}{MAX}, & \text{Jika tidak} \end{cases} \quad (2.2)$$

$$V = \frac{Max}{255} \quad (2.3)$$

Keterangan:

<i>Max</i>	= Nilai maksimal dari keseluruhan nilai <i>red</i> , <i>green</i> , dan <i>blue</i> .
<i>Min</i>	= Nilai minimal dari keseluruhan nilai <i>red</i> , <i>green</i> , dan <i>blue</i> .
<i>H</i>	= <i>Hue</i> .
<i>S</i>	= <i>Saturation</i> .
<i>V</i>	= <i>value</i>
<i>R</i>	= <i>Red</i> .
<i>G</i>	= <i>Green</i> .
<i>B</i>	= <i>Blue</i> .

Color Moment

Color Moment adalah representasi karakteristik citra warna, pembanding citra untuk mendapatkan *Color Similarity* sebuah citra yang disimpan dalam sebuah *database* citra (Halim, et al., 2013). Terdapat tiga operasi *moment* dalam penyebaran warna dari sebuah citra, yaitu *mean* (μ) yang merupakan rata-rata nilai warna dalam citra, *Deviation* (σ) yang merupakan akar pangkat dari varian atau standar deviasi. Dan yang terakhir adalah *Skewness* (θ) yang menunjukkan ukuran derajat ketidaksimetrisan.

$$\mu = \frac{1}{N} \sum_{j=1}^N P_{ij} \quad (2.4)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{j=1}^N (P_{ij} - \mu)^2} \quad (2.5)$$

$$\theta = \sqrt[3]{\frac{1}{N} \sum_{j=1}^N (P_{ij} - \mu)^3} \quad (2.6)$$

Keterangan:

μ	= rata-rata nilai penyebaran warna dari sebuah citra (<i>mean</i>).
σ	= akar pangkat dari varian (standar deviasi).
θ	= Ukuran derajat ketidaksimetrisan (<i>skewness</i>)
N	= Jumlah total <i>pixel</i> pada citra.
P_{ij}	= Nilai dari komponen warna ke- i pada <i>pixel</i> ke-j

2.3.2 Fitur Tekstur

Tekstur adalah keteraturan pola-pola yang menunjukkan sifat atau karakteristik dari suatu daerah tertentu yang terbentuk dari pixel-pixel citra (Halim, et al., 2013). Terdapat beberapa metode yang dapat digunakan dalam pengambilan fitur tekstur antara lain *Gray-level Co-occurrence Matrix* (GLCM),

Haralick, dan *Local Binary Pattern* (LBP). Hasil dari analisis tekstur dapat digunakan sebagai dasar klasifikasi citra.

Haralick Features

Haralick features menggunakan *co-occurrence matrix* untuk menyimpan informasi perhitungan tekstur. Terdapat 14 *features* yang dapat digunakan untuk mendefinisikan tekstur sebuah citra menurut Haralick (Suryanto, et al., 2018). Tetapi menurut Connors dan Harlow menyatakan bahwa lima fitur cukup untuk memberikan hasil klasifikasi yang baik (Bekkari, et al., 2012). Kelima fitur tersebut antara lain *Energy* (*E*), *Contrast* (*C*), *Correlation* (*Cor*), *Entropy* (*H*), dan *Local Homogeneity* (*LH*). Kemudian hasil *co-occurrence matrix* dihitung dengan arah derajat yang berbeda.

$$E = \sum_i \sum_j (M(i,j))^2 \quad (2.7)$$

$$C = \sum_i \sum_j (i - j)^2 * M(i,j) \quad (2.8)$$

$$Cor = \frac{1}{\sigma_{i\sigma j}} \sum_i \sum_j (i - \mu i)(j - \mu j) M(i,j) \quad (2.9)$$

$$H = \sum_i \sum_j M(i,j) \log(M(i,j)) \quad (2.10)$$

$$LH = \sum_i \sum_j \frac{M(i,j)}{1 + (i-j)^2} \quad (2.11)$$

Untuk perhitungan *correlation* untuk nilai rata-rata dan varian dapat digunakan rumus:

$$\mu i = \sum_i \sum_j i * M(i,j) \quad (2.12)$$

$$\mu j = \sum_i \sum_j j * M(i,j) \quad (2.13)$$

$$\sigma i = \sqrt{\sum_i \sum_j (i - \mu i)^2 * M(i,j)} \quad (2.14)$$

$$\sigma j = \sqrt{\sum_i \sum_j (j - \mu j)^2 * M(i,j)} \quad (2.15)$$

Keterangan:

E	= Energy
M	= Matriks GLCM yang sudah dinormalisasi.
i	= baris ke-i.
j	= kolom ke-j.
C	= <i>Contrast</i> .
Cor	= <i>Correlation</i> .
σ_i	= Varian untuk horizontal / baris.
σ_j	= Varian untuk vertikal / kolom.
μ_i	= Nilai rata-rata untuk horizontal/baris.
μ_j	= Nilai rata-rata untuk vertikal/kolom.
H	= <i>Entropy</i>
LH	= <i>Local Homogeneity</i> .

2.4 Naive Bayes Classifier

Naive Bayes merupakan suatu metode klasifikasi yang berasal dari teorema *Bayes* yaitu memprediksi peluang yang di masa depan dengan berdasarkan peluang yang ada pada masa sebelumnya (Liantoni & Nugroho, 2015). *Naive Bayes Classifier* termasuk kategori *supervised learning* karena dapat dilatih berdasarkan contoh yang sudah dikategorikan.

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

(2.16)

Berdasarkan persamaan diatas untuk *prior* peluang terjadinya Y , dapat disederhanakan karena selalu bernilai sama untuk setiap kelas. Sehingga dapat menggunakan persamaan berikut.

$$P(X|Y) = P(Y|X)P(X)$$

(2.17)

Keterangan:

- $P(X|Y)$ = peluang hipotesis berdasarkan kondisi (*Posterior*) .
- $P(Y|X)$ = Peluang berdasarkan kondisi pada hipotesis (*likelihood*).
- $P(X)$ = Peluang hipotesis X tanpa menggunakan *evidence* apapun (*prior*).
- $P(Y)$ = Peluang hipotesis data yang merupakan suatu kelas spesifik (*prior*).

Jika atribut yang digunakan bertipe kontinu, maka solusi *naive bayes* yang dapat digunakan adalah menggunakan distribusi *gaussian* (Tsangaratos & Ilia, 2016). Terdapat 2 parameter yang digunakan dalam distribusi *gaussian* yaitu *mean* dan variansi. Sehingga untuk perhitungan *likelihood* dapat menggunakan persamaan berikut.

$$\mu = \frac{1}{N} \sum_{n=0}^N x_n$$

(2.18)

$$\sigma^2 = \frac{1}{N} \sum_{n=0}^N (x_n - \mu)^2 \quad (2.19)$$

$$P(Y_1, \dots, Y_N | \mu, \sigma^2) = \prod_{Y=1}^{Y_N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_Y - \mu)^2}{2\sigma^2}} \quad (2.20)$$

Keterangan:

n	= data ke- n .
N	= banyak data dalam 1 fitur.
X_n	= nilai data ke- n .
μ	= nilai rata-rata.
σ^2	= varian.
Y_1, \dots, Y_N	= banyak fitur 1 sampai fitur N .
e	= eksponensial

2.5 Confusion Matrix

Dalam melakukan evaluasi hasil klasifikasi menggunakan *confusion matrix*. *Confusion matrix* menerapkan konsep dari *machine learning* yaitu dengan memanfaatkan nilai dari perhitungan antara hasil nilai kelas sebenarnya dan nilai kelas hasil prediksi (Deng, et al., 2016). Tujuan dari perhitungan digunakan *confusion matrix* adalah mendapatkan nilai *accuracy*, *precision*, *recall*, dan *f-score*. *Accuracy* merupakan proporsi dari total jumlah prediksi yang benar. *Precision* merupakan nilai dari akurasi dengan kelas tertentu yang telah diprediksi. *Recall* merupakan ukuran kemampuan prediksi untuk memilih *instance* dari kelas tertentu dari kumpulan data. Dan *f-score* merupakan rata rata *harmonic* dari nilai *precision* dan *recall*.

		Hasil Prediksi		
		A_1	A_j	A_n
Hasil Nilai sebenarnya	A_1	N_{11}	N_{1j}	N_{1n}
	A_i	N_{i1}	N_{ij}	N_{in}
	A_n	N_{n1}	N_{nj}	N_{nn}

$$Accuracy = \frac{\sum_{i=1}^n N_{ii}}{\sum_{i=1}^n \sum_{j=1}^n N_{ij}} \quad (2.21)$$

$$Precision = \frac{N_{ii}}{\sum_{k=1}^n N_{ki}} \quad (2.22)$$

$$Recall = \frac{N_{ii}}{\sum_{k=1}^n N_{ik}} \quad (2.23)$$

$$F - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

(2.24)

Keterangan:

A_n = banyak kelas.

N = merepresentasikan nilai yang diklasifikasikan sebagai nilai sebenarnya dan nilai prediksi.



BAB 3 METODOLOGI PENELITIAN

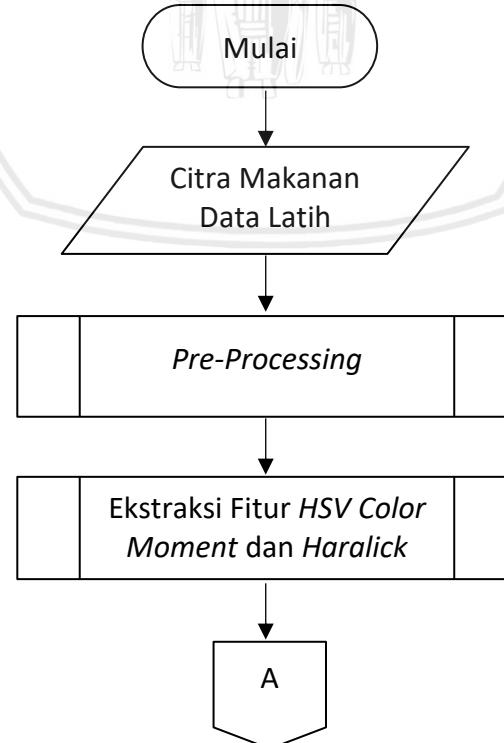
3.1 Tipe Penelitian

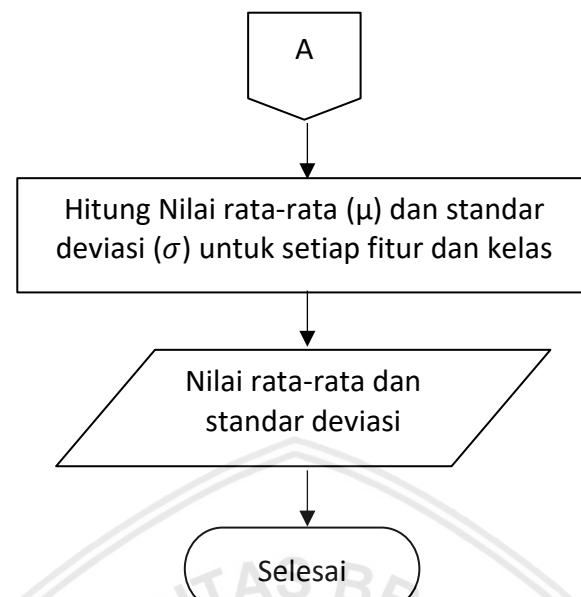
Tipe penelitian yang digunakan pada penelitian ini adalah nonimplementatif, Karena mengimplementasikan suatu metode dalam pengumpulan data. Selain itu tipe penelitian yang dilakukan dibangun berdasarkan studi pustaka yang sudah dilakukan sebelumnya, sehingga penelitian ini termasuk dalam penelitian nonimplementatif analitik.

3.2 Metode Penelitian

Selain itu pada bab ini menjelaskan tentang tahapan-tahapan dan metode yang digunakan dalam proses ekstraksi fitur dan klasifikasi citra makanan. Terdapat 2 tahapan dalam melakukan metode penelitian ini yaitu tahap pelatihan dan tahap pengujian. Pada tahap pelatihan menggunakan data latih berupa citra makanan. Selanjutnya dari citra makanan dilakukan proses *preprocessing* dengan tujuan menghasilkan gambar dengan kualitas yang lebih baik dari sebelumnya dan menghasilkan objek yang terpisah dengan latar belakang.

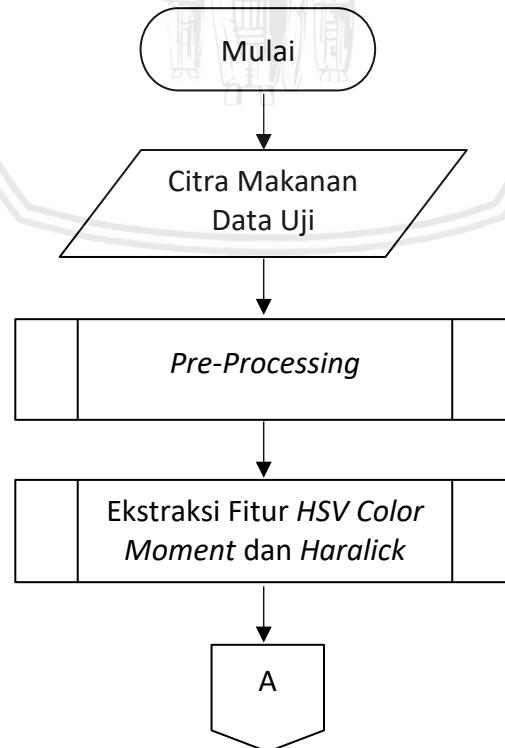
Kemudian dari proses *pre-processing* citra dilakukan proses ekstraksi fitur. Proses ekstraksi fitur yang digunakan adalah proses pengambilan fitur warna menggunakan HSV *Color Moment* dan fitur tekstur menggunakan *Haralick*. Tujuan dilakukan ekstraksi fitur adalah untuk mendapatkan nilai yang akan digunakan untuk proses klasifikasi. Pada akhir tahap pelatihan, dilakukan proses perhitungan nilai rata-rata dan standar deviasi untuk setiap nilai fitur dan kelas. Diagram alir untuk tahap pelatihan dapat dilihat pada Gambar 3.1.

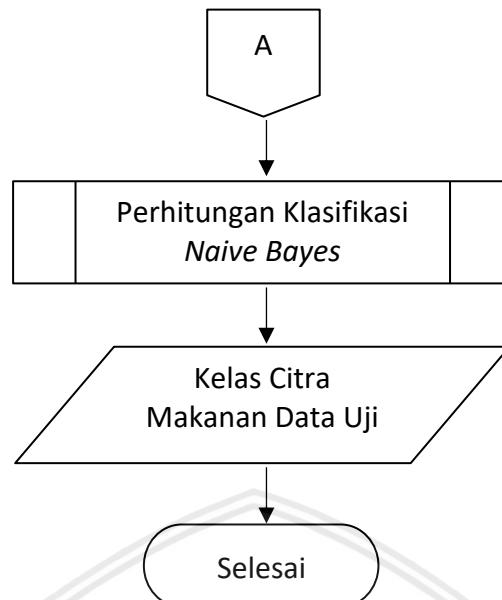




Gambar 3.1 Diagram Alir Tahap Pelatihan

Setelah nilai fitur didapatkan, metode klasifikasi yang digunakan adalah *Naive Bayes*. berbeda dengan tahap pelatihan, pada tahap pengujian nilai rata-rata dan standar deviasi yang telah dihitung pada tahap pelatihan digunakan untuk perhitungan *naive bayes* berdasarkan fitur data uji. Hasil keluaran dari proses klasifikasi adalah kelas citra data uji. Diagram alir tahapan pengujian dapat dilihat pada Gambar 3.2.





Gambar 3.2 Diagram Alir Tahap Pengujian

3.3 Lokasi Penelitian

Lokasi penelitian dilaksanakan di Ruang Grup Riset *Computer Vision* F9.3, Gedung F lantai 9, Fakultas Ilmu Komputer, Universitas Brawijaya, Jalan Veteran, Ketawanggede, Lowokwaru, Kota Malang, Jawa Timur 65145.

3.4 Pengumpulan Data

Penentuan kumpulan data awal yang akan digunakan dalam penelitian ini adalah berupa data primer atau data yang diambil secara manual dengan menggunakan kamera *smartphone*. Tipe *smartphone* yang digunakan adalah tipe *Huawei Nova 3i* dan *Iphone 6s* dengan besar resolusi masing-masing sebesar 12 megapiksel dengan bukaan f/1.9 dan 12 megapiksel dengan bukaan f/2.2.

Besar intensitas cahaya matahari sesuai dengan tempat pelaksanaan pengambilan citra makanan yang dilaksanakan di Ruang Gedung F Lantai 9, Fakultas Ilmu Komputer, Universitas Brawijaya pada Pukul 10.00 sampai 12.30 WIB.

Pengambilan data awal dilakukan dengan melakukan foto makanan sebelum dimakan dan foto makanan sesudah dimakan. Cara pengambilan citra makanan dilakukan berdasarkan kelas citra makanan, dengan jumlah kelas makanan sebanyak 12 kelas. pada setiap kelas jenis makanan dilakukan pengambilan foto sebanyak tiga kali berdasarkan tegak lurus di tiga level ketinggian dan kemiringannya. Kemudian dilakukan pengambilan foto ulang pada makanan yang sudah dimakan sebanyak $\frac{1}{4}$ bagian, $\frac{1}{2}$ bagian, dan $\frac{3}{4}$ bagian pada setiap kelas makanan. Daftar kelas yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.1. Dan daftar kelas data latih dan data uji dapat dilihat pada Tabel 3.2 dan Tabel 3.3.

Tabel 3.1 Daftar Kelas Citra Makanan

Daftar Kelas Citra Makanan				
Gambar				
Kelas (Kode)	Donat (001)	Rendang (009)	Mentimun (010)	Strawberry (016)
Gambar				
Kelas(Kode)	Pisang Ijo (017)	Pisang Kuning (018)	Jeruk Ijo (020)	Beng-Beng (024)
Gambar				
Kelas(Kode)	Tim Tam (026)	Gerry Saluut (028)	Biskuat Coklat (029)	Genji Pie (032)

Tabel 3.2 Daftar Kelas Latih

Daftar Kelas Citra Makanan				
Gambar				
Kelas(Kode)	Donat (001)	Rendang (009)	Mentimun (010)	Strawberry (016)
Gambar				
Kelas(Kode)	Pisang Ijo (017)	Pisang Kuning (018)	Jeruk Ijo (020)	Beng-Beng (024)
Gambar				
Kelas(Kode)	Tim Tam (026)	Gerry Saluut (028)	Biskuat Coklat (029)	Genji Pie (032)

Tabel 3.3 Daftar Kelas Uji

Daftar Kelas Citra Makanan				
Gambar				
Kelas(Kode)	Donat (001)	Rendang (009)	Mentimun (010)	Strawberry (016)
Gambar				
Kelas(Kode)	Pisang Ijo (017)	Pisang Kuning (018)	Jeruk Ijo (020)	Beng-Beng (024)
Gambar				
Kelas(Kode)	Tim Tam (026)	Gerry Saluut (028)	Biskuat Coklat (029)	Genji Pie (032)

3.5 Peralatan Pendukung

Peralatan pendukung yang digunakan pada penelitian ini adalah peralatan yang dipakai selama pembangunan sistem. Peralatan yang digunakan pada penelitian adalah sebagai berikut.

1. Spesifikasi *hardware*

Menggunakan Laptop dengan Prosesor Intel® Core™ i7-6700HQ CPU @2.60GHz (8 CPUs), Memori RAM 16GB DDR4, dengan Ruang penyimpanan sebesar 1TB, dan ukuran layar 15,6 inch.

2. Spesifikasi *software*

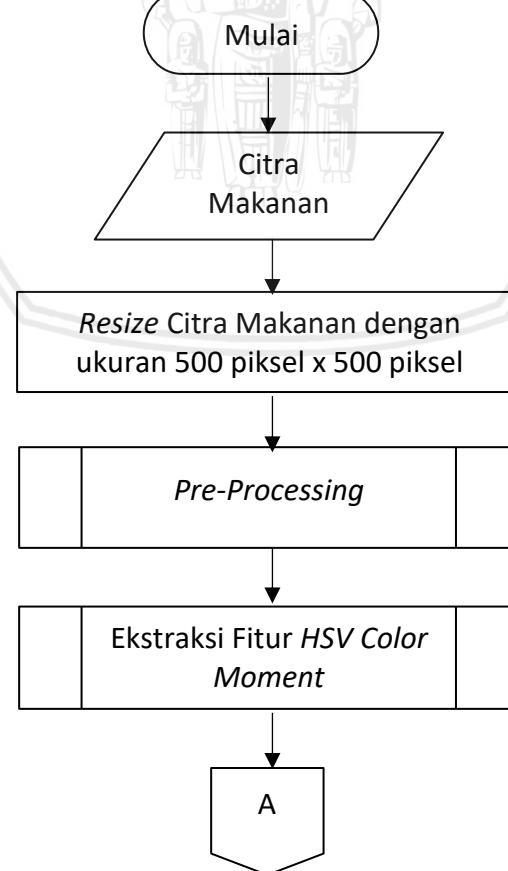
- Microsoft Windows 10.
- Python 3.6.
- PyCharm 2018.1.2 (*Community Edition*)

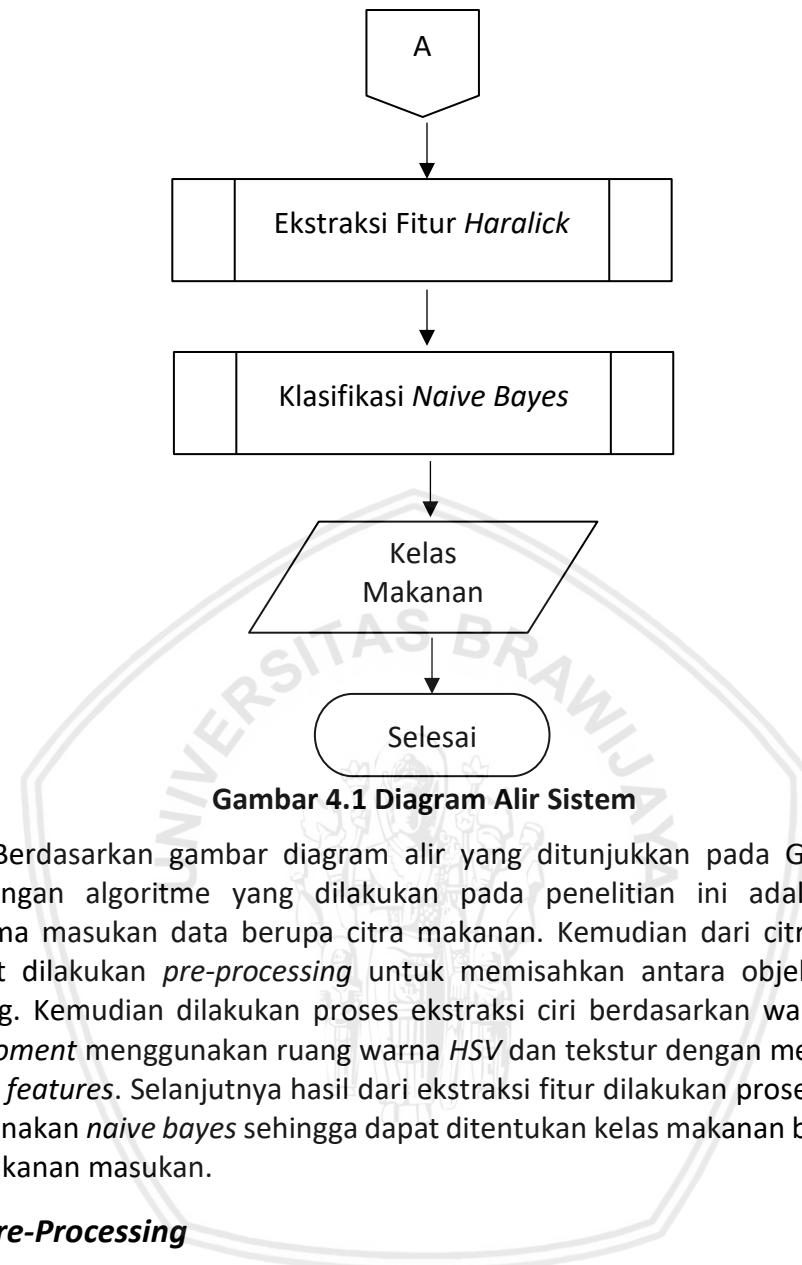
BAB 4 PERANCANGAN

Pada Bab ini akan dijelaskan tentang rancangan algoritme dan perhitungan manual. Tujuan dari dilakukannya perancangan ini adalah sebagai landasan dalam mengimplementasikan sistem klasifikasi jenis makanan menggunakan fitur *Haralick*, *HSV Color Moment*, dan *Simple Morphological Shape Descriptors*.

4.1 Perancangan Algoritme

Dalam melakukan perancangan algoritme ini digambarkan dengan menggunakan *flowchart* atau disebut dengan diagram alir. Diagram alir digunakan untuk menunjukkan alur atau urutan-urutan dari sistem yang akan dibuat. Dan tujuan dari digunakannya diagram alur ini adalah agar alur dari algoritme yang dibuat dapat dengan mudah untuk dipahami dan terurut. Secara umum perancangan algoritme dimulai dengan membaca masukan berupa citra makanan. Kemudian dilakukan proses *pre-processing* pada masukan citra makanan, dengan tujuan mempermudah dalam pengambilan fitur. Tahap selanjutnya adalah pengambilan fitur berdasarkan fitur warna yang menggunakan *HSV color moment*, fitur tekstur yang menggunakan *haralick features*, dan fitur bentuk yang menggunakan *simple morphological shape descriptors*. Kemudian dari fitur-fitur yang sudah didapat dilakukan proses klasifikasi menggunakan *naive bayes* dan hasil keluaran berupa kelas makanan. Perancangan algoritme secara umum dapat dilihat pada Gambar 4.1.



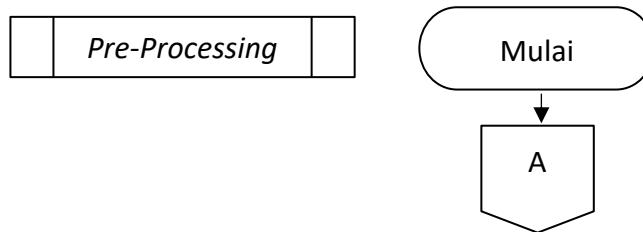


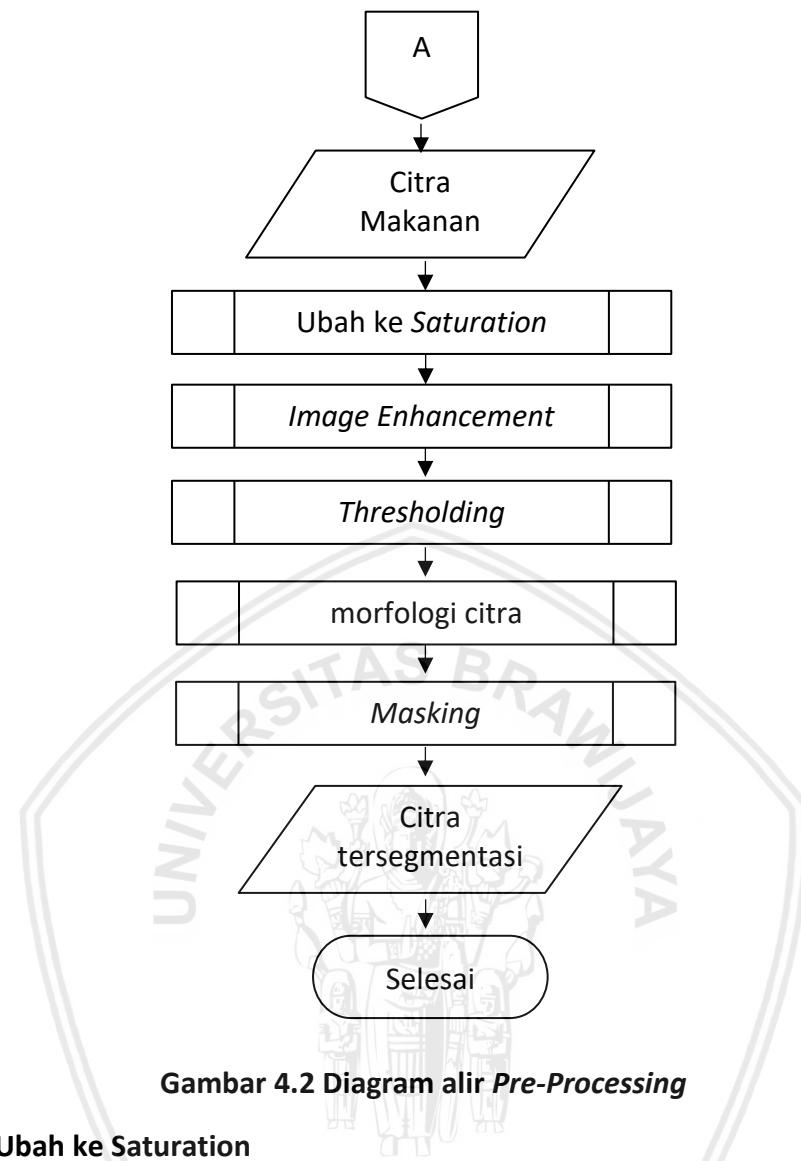
Gambar 4.1 Diagram Alir Sistem

Berdasarkan gambar diagram alir yang ditunjukkan pada Gambar 4.1, perancangan algoritme yang dilakukan pada penelitian ini adalah dengan menerima masukan data berupa citra makanan. Kemudian dari citra makanan tersebut dilakukan *pre-processing* untuk memisahkan antara objek dan latar belakang. Kemudian dilakukan proses ekstraksi ciri berdasarkan warna dengan *color moment* menggunakan ruang warna HSV dan tekstur dengan menggunakan *haralick features*. Selanjutnya hasil dari ekstraksi fitur dilakukan proses klasifikasi menggunakan *naive bayes* sehingga dapat ditentukan kelas makanan berdasarkan citra makanan masukan.

4.1.1 Pre-Processing

Setelah citra makanan diterima sebagai masukan, tahap selanjutnya adalah *Pre-Processing*. *Pre-Processing* merupakan tahap perbaikan dan peningkatan kualitas citra sebelum dilakukan proses pengambilan fitur. Tahapan *pre-processing* yang digunakan adalah Ubah ke *Saturation*, *image enhancement*, *Thresholding*, morfologi citra, dan Masking. Diagram alir untuk *pre-processing* dapat dilihat pada Gambar 4.2.

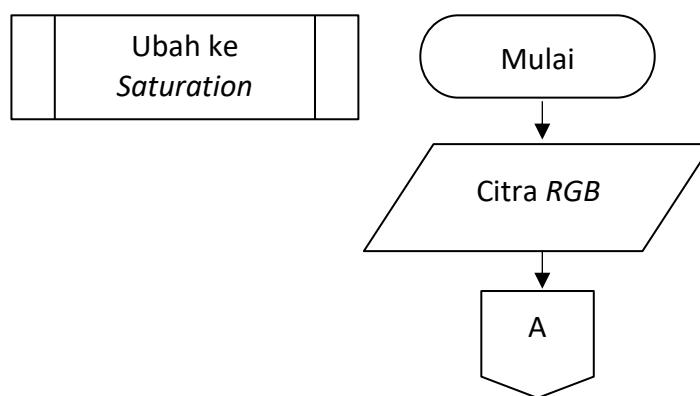


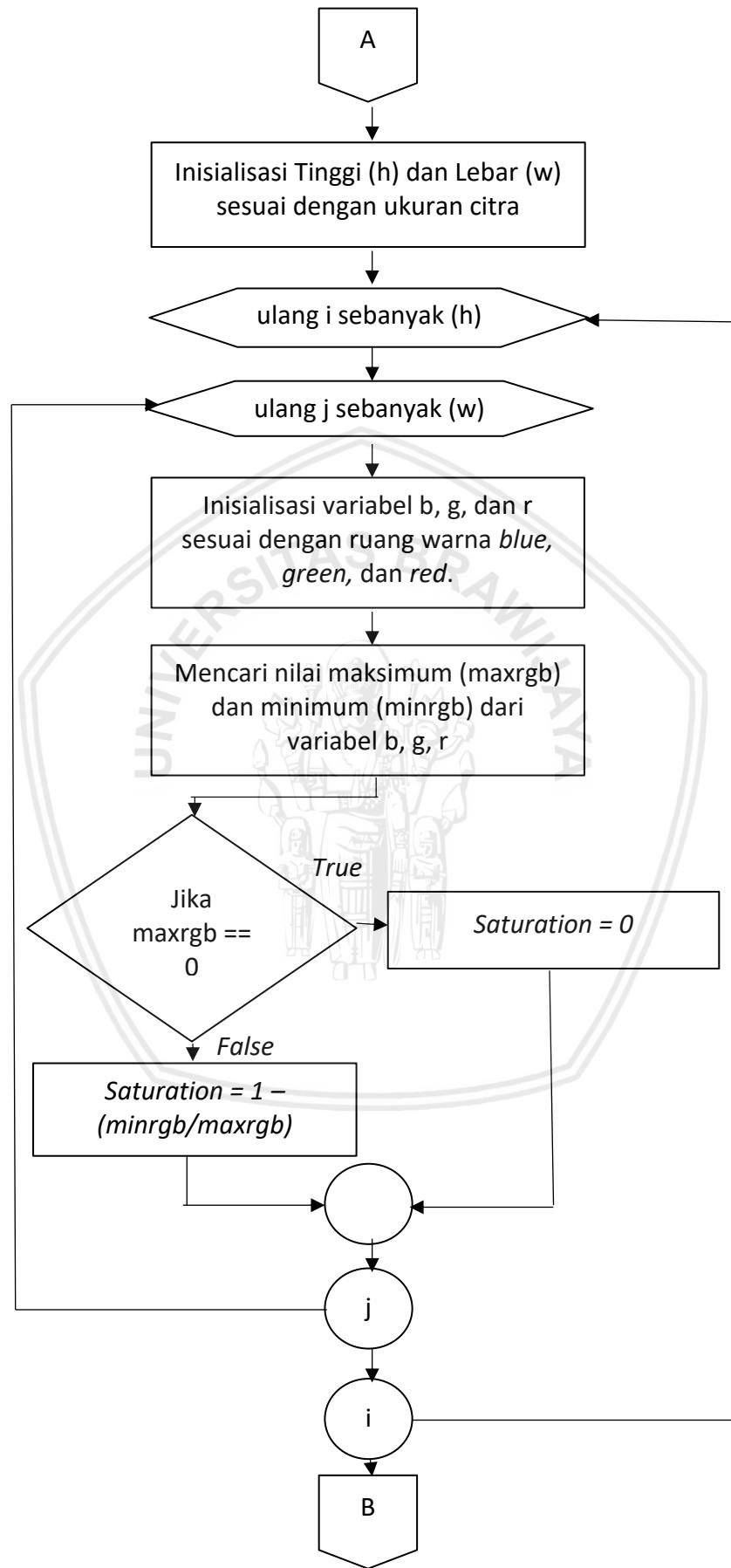


Gambar 4.2 Diagram alir *Pre-Processing*

4.1.1.1 Ubah ke Saturation

Tahap Ubah ke *saturation* adalah tahap mengubah ruang warna citra *red*, *green*, dan *blue* ke dalam ruang warna *hue*, *saturation*, dan *value* tetapi dengan hanya mengambil nilai *saturation* dalam suatu citra makanan dan digunakan pada tahap *pre-processing* berikutnya. Diagram alir untuk Ubah ke *Saturation* dapat dilihat pada Gambar 4.3.





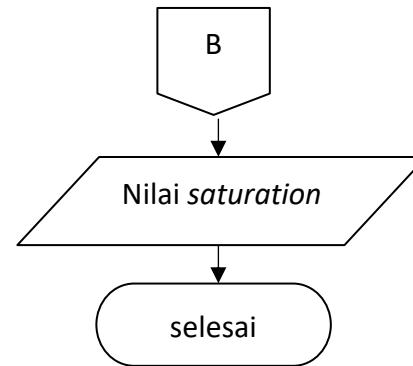
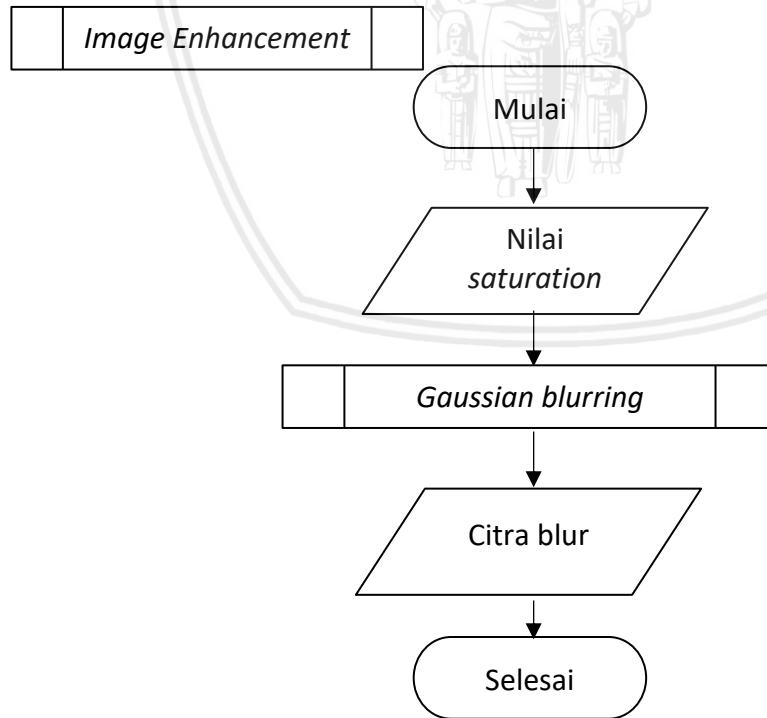
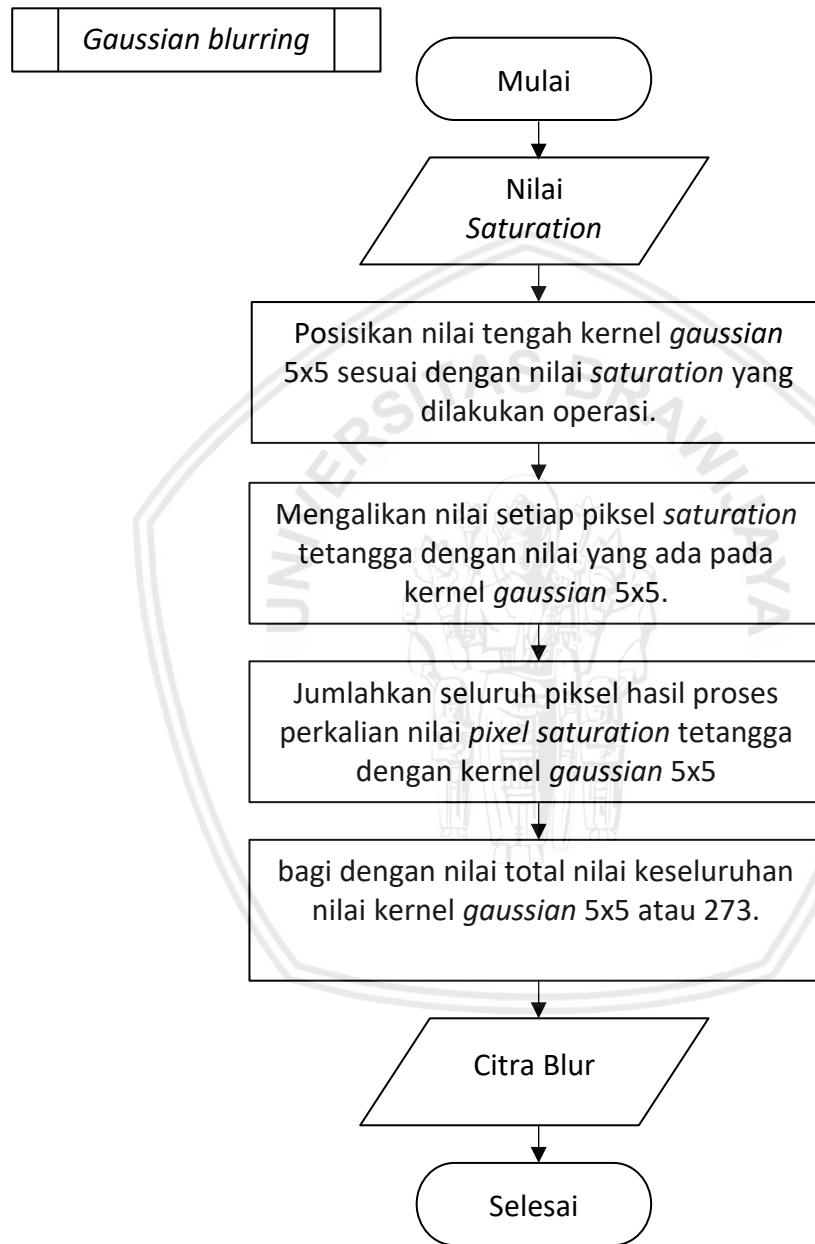
**Gambar 4.3 Diagram alir ubah ke saturation****4.1.1.2 Image Enchancement**

Image Enchancement merupakan tahapan *pre-processing* untuk meningkatkan kualitas citra menjadi lebih baik lagi. Teknik yang digunakan dalam peningkatan kualitas dibagi menjadi dua. Pertama, peningkatan yang dilakukan pada domain spasial yaitu dengan secara langsung memanipulasi pixel. Kedua, peningkatan yang dilakukan pada domain frekuensi. Kemudian pada penelitian ini hanya dilakukan peningkatan kualitas citra secara domain spasial, yaitu *mask processing* dengan *gaussian blurring*. *Point Processing* merupakan meningkatkan kualitas citra yang hanya melibatkan nilai dari satu pixel saja untuk setiap operasi, berbeda dengan *mask processing* yang melibatkan nilai ketetanggaan pada setiap operasi. Diagram alir untuk *Image Enchancement* dapat dilihat pada Gambar 4.4.

**Gambar 4.4 Diagram alir Image Enchancement**

4.1.1.2.1 Gaussian Blurring

Gaussian Blurring termasuk dalam *mask processing* yang dimana untuk setiap operasi melibatkan nilai piksel tetangga. Tujuan dari diakukan *gaussian blurring* adalah membuat efek blur pada citra, sehingga akan menghasilkan citra yang lebih baik saat dilakukan proses *thresholding*. Diagram alur untuk *gaussian blurring* dapat dilihat pada Gambar 4.5.

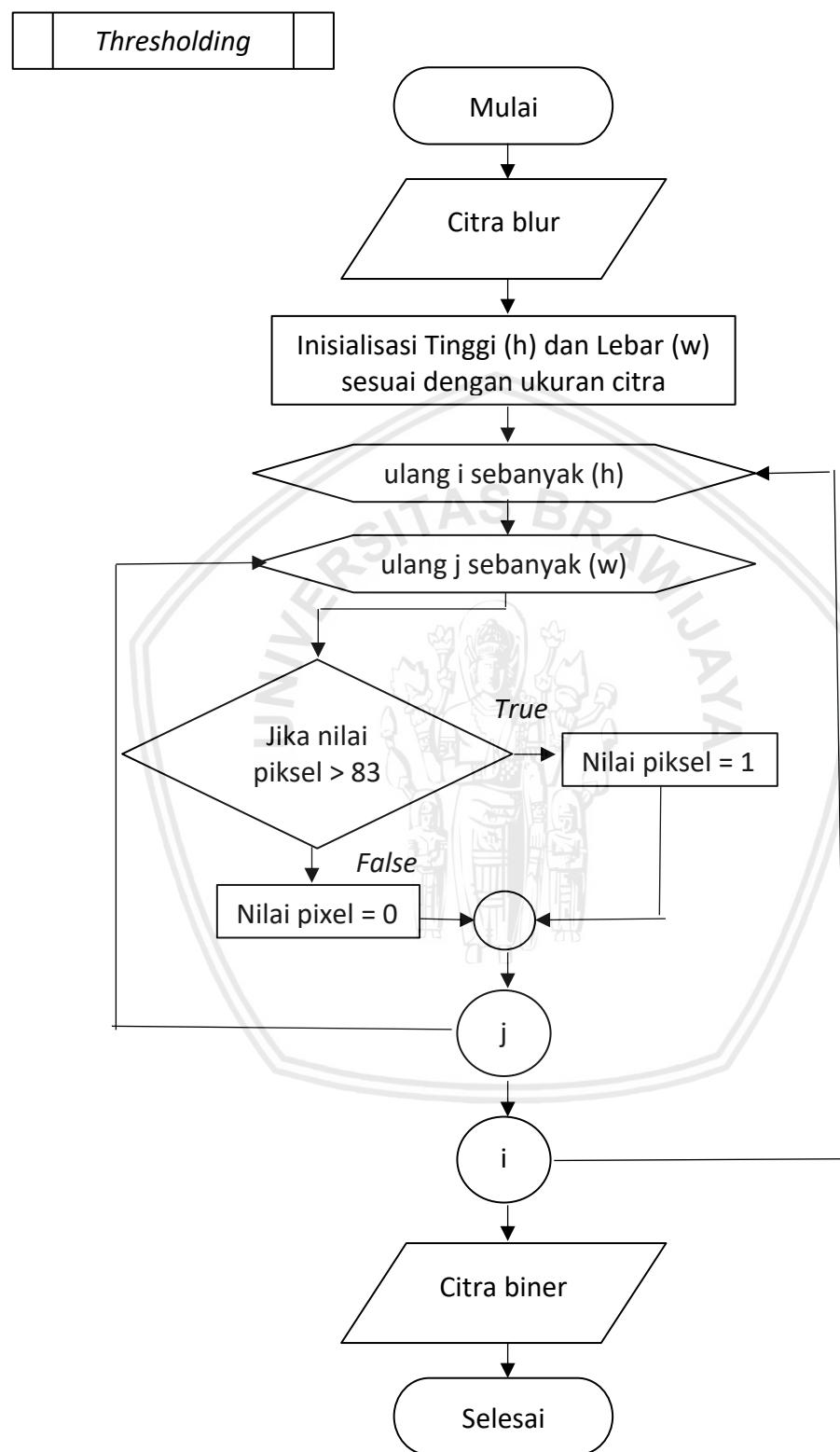


Gambar 4.5 Diagram alir Gaussian Blur

4.1.1.3 Thresholding

Thresholding merupakan proses mengubah citra abu-abu menjadi citra dua nilai atau biner. Tujuan dilakukannya *thresholding* adalah memisahkan objek

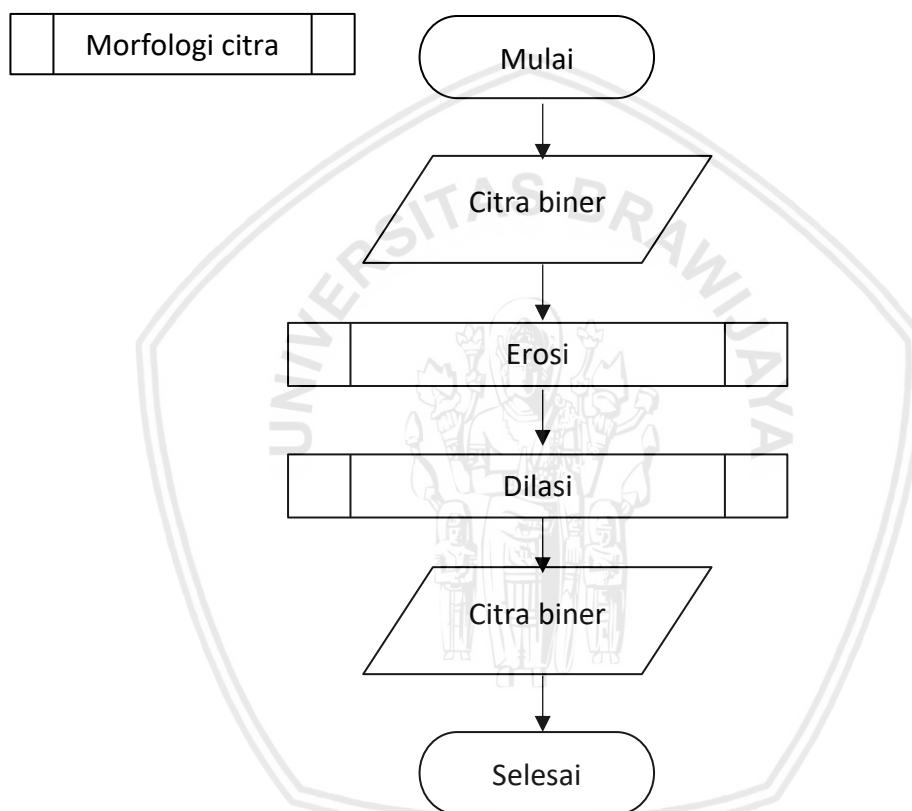
dengan latar belakang. Diagram alir untuk *thresholding* dapat dilihat pada Gambar 4.6.



Gambar 4.6 Diagram alir *Thresholding*

4.1.1.4 Morfologi Citra

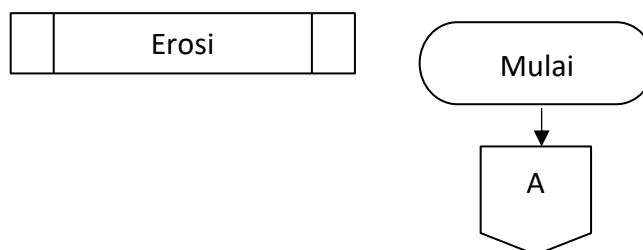
Morfologi citra merupakan teknik yang memanfaatkan konsep logika matematika dalam merepresentasikan dan deskripsi citra biner berdasarkan bentuk. Konsep dalam morfologi adalah dengan menggunakan *structuring element* yang dibuat sesuai keinginan, kemudian diaplikasikan pada citra biner masukan dengan menggunakan operasi logika matematika. Sehingga menghasilkan citra keluaran dengan ukuran yang sama dengan citra masukan. Kegunaan dari morfologi citra adalah menghilangkan *noise* (objek kecil dan *fill holes*) dan memisahkan objek. Terdapat empat operasi dalam morfologi yaitu fit dan hit, erosi, dilasi, dan operasi gabungan. Diagram alir untuk morfologi citra dapat dilihat pada Gambar 4.7.

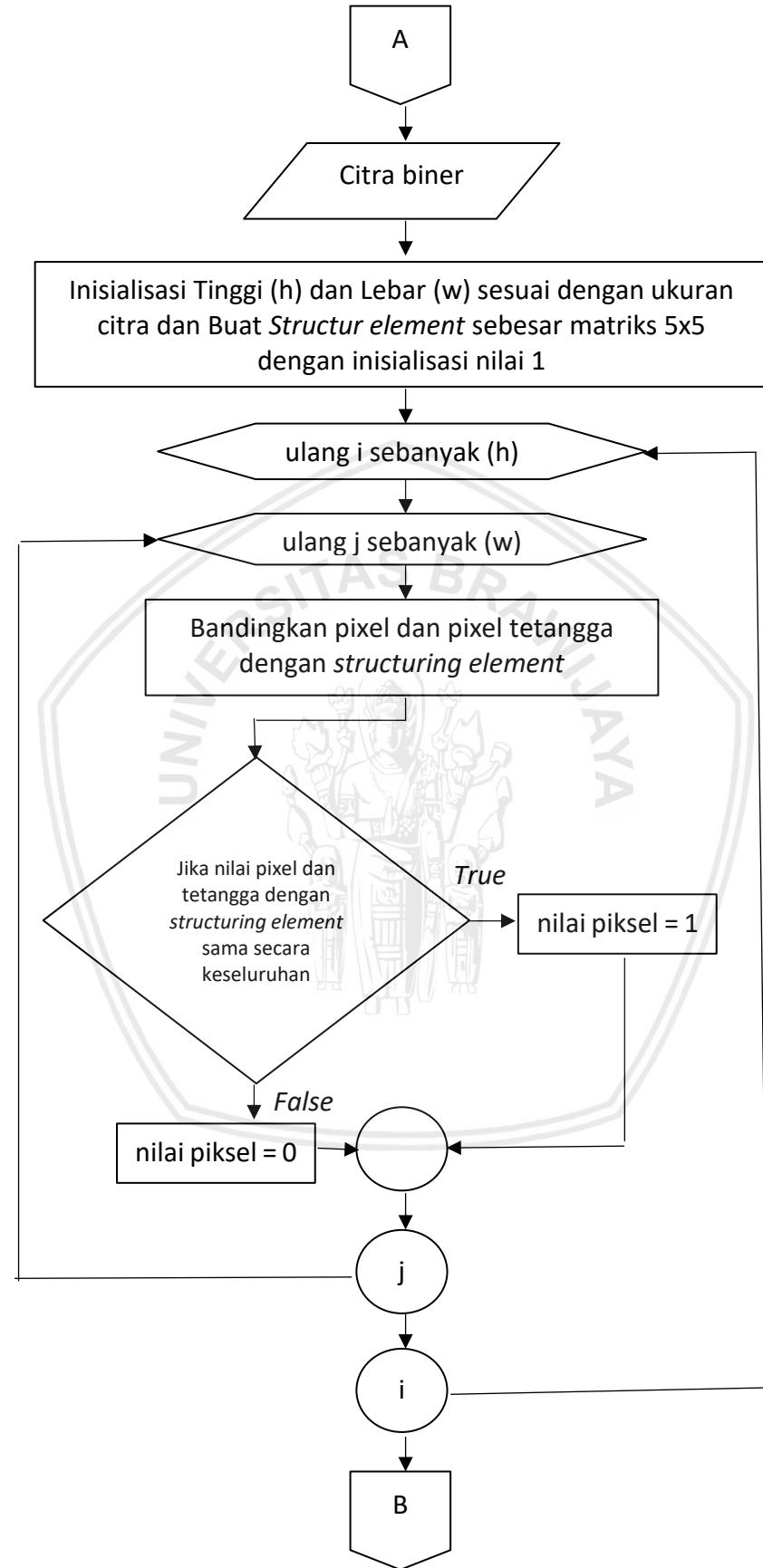


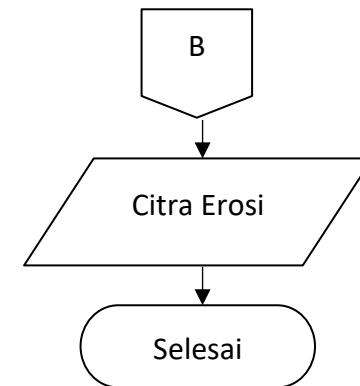
Gambar 4.7 Diagram alir Morfologi Citra

4.1.1.4.1 Erosi

Erosi merupakan tahapan pada morfologi citra dengan melakukan pengurangan nilai pada pinggiran objek, sehingga menghasilkan objek pada citra semakin kecil. Diagram alir untuk erosi dapat dilihat pada Gambar 4.8.



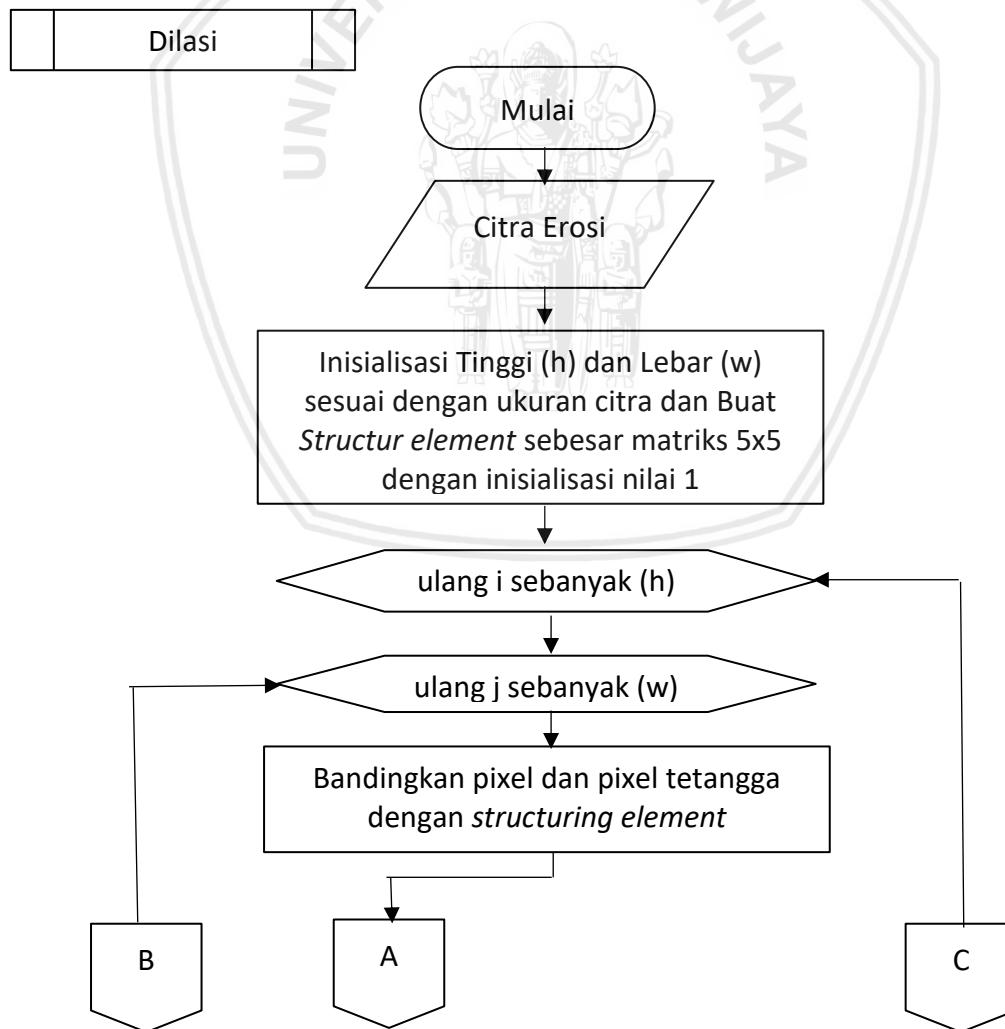


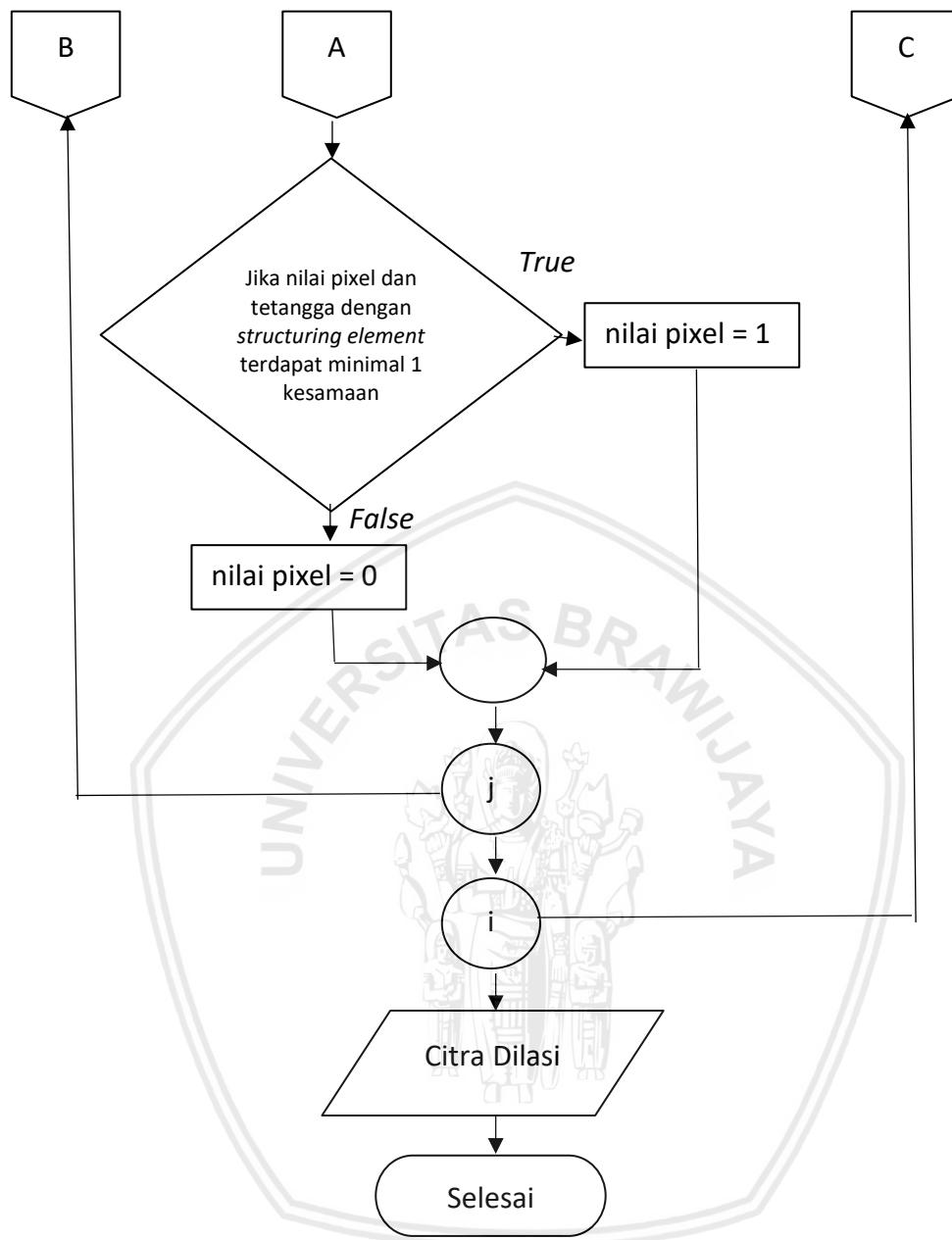


Gambar 4.8 Diagram alir Erosi

4.1.1.4.2 Dilasi

Dilasi merupakan tahapan pada morfologi citra dengan melakukan penambahan nilai pada pinggiran objek, sehingga menghasilkan objek pada citra semakin besar. Diagram alir untuk dilasi dapa dilihat pada Gambar 4.9.

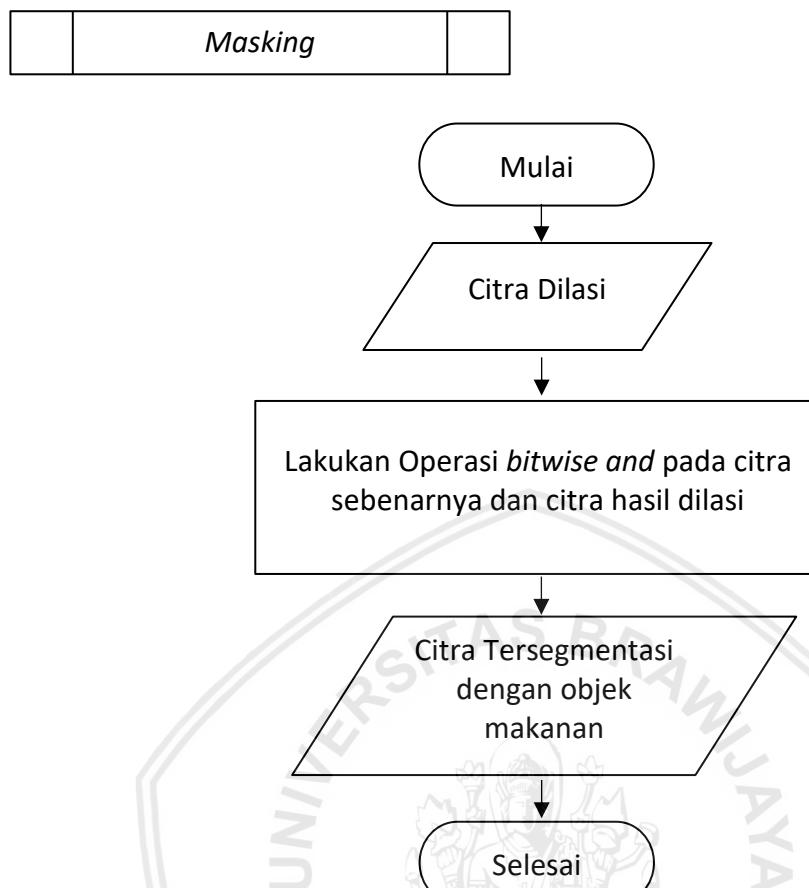




Gambar 4.9 Diagram alir Dilasi

4.1.1.5 *Masking*

Pada tahap *Masking* dilakukan proses menutupi objek latar belakang pada citra sebenarnya dengan menggunakan citra hasil dilasi pada tahap sebelumnya. Hasil dari proses *Masking* berupa citra makanan yang sudah tersegmentasi dengan latar bewarna hitam. Kemudian pengambilan nilai fitur dilakukan hanya pada objek makanan. Diagram alir proses *Masking* dapat dilihat pada Gambar 4.10.

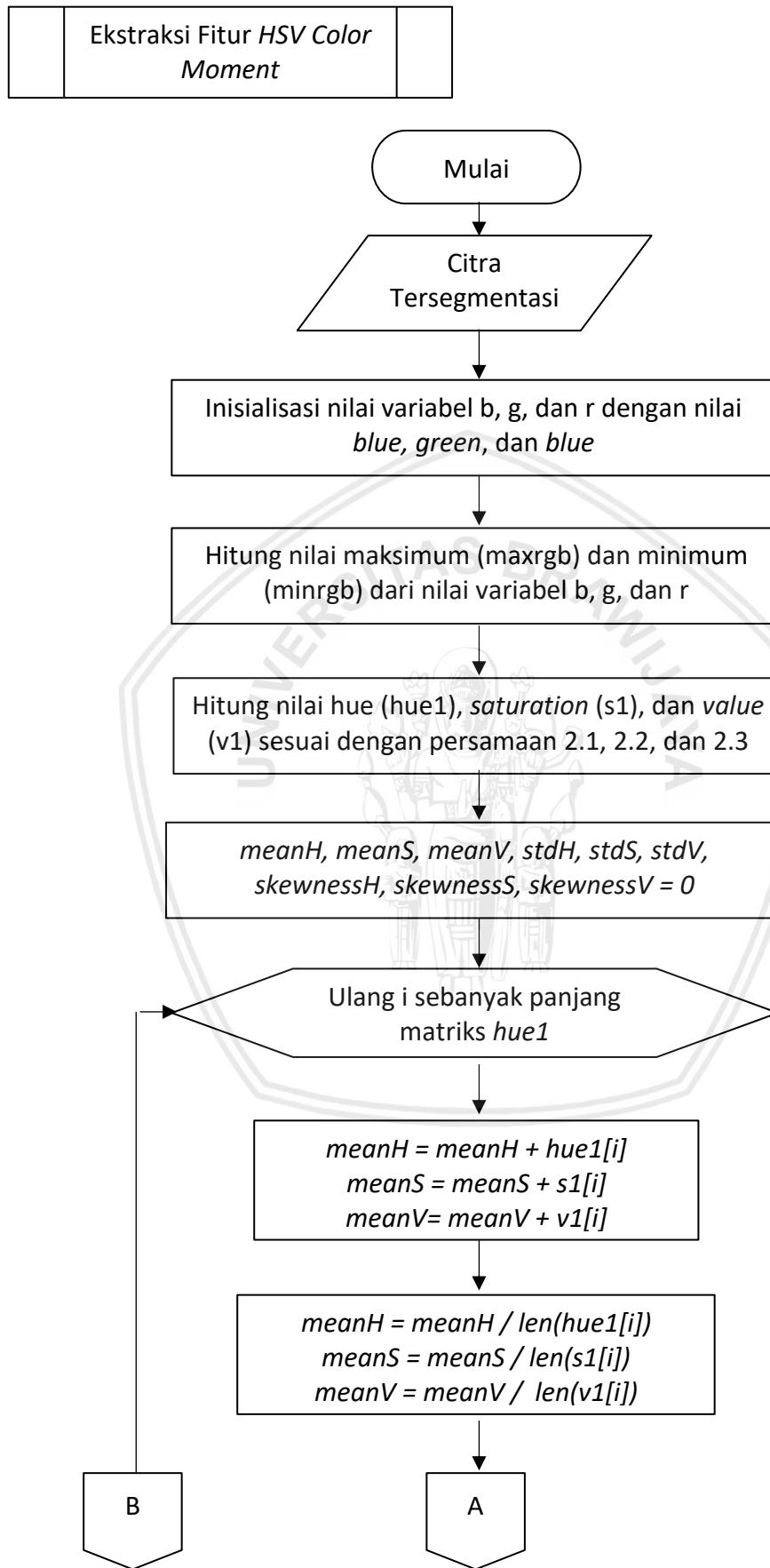


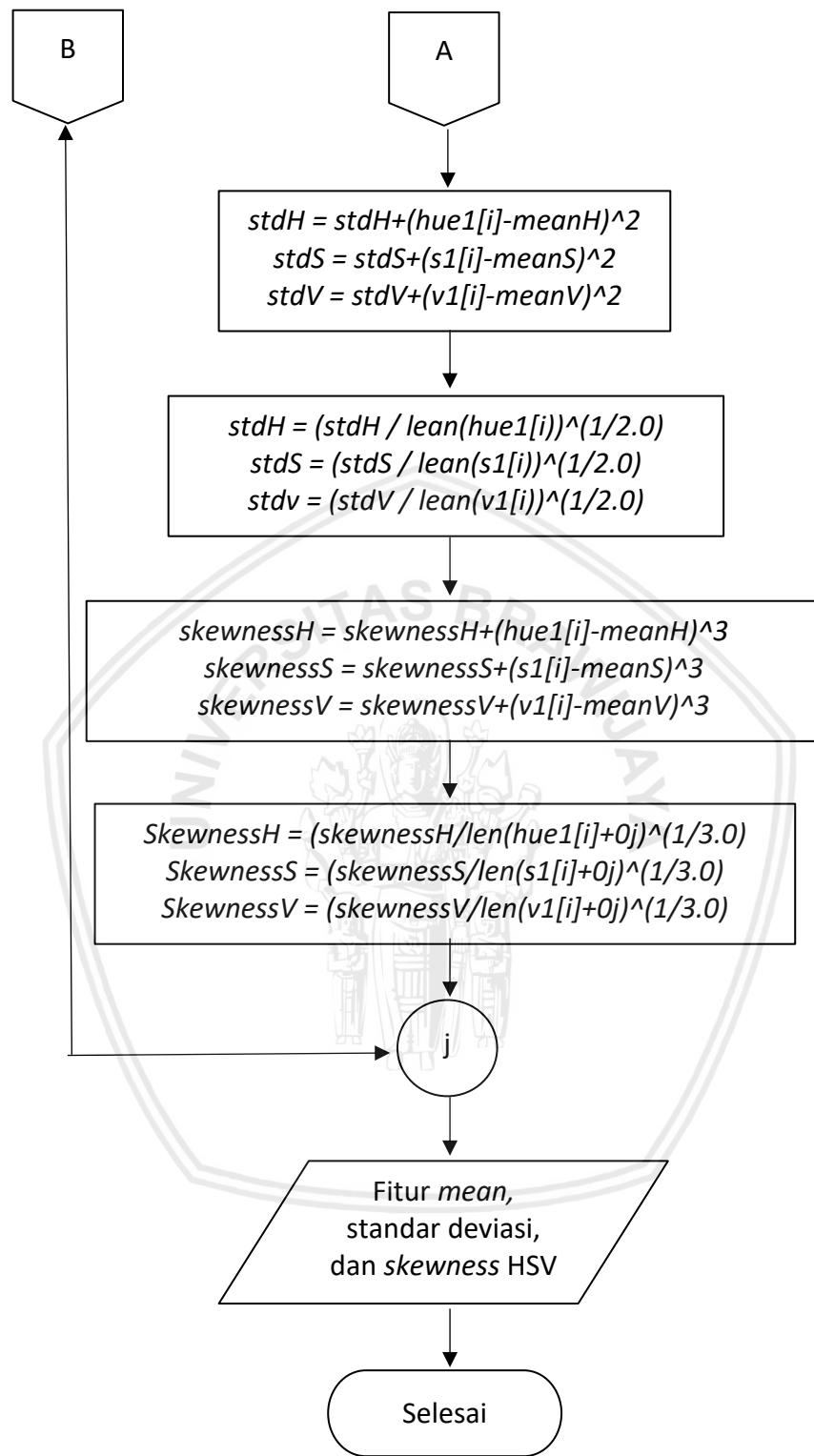
Gambar 4.10 Diagram Alir Masking

Hasil akhir dari tahap proses *pre-processing* berupa citra tersegmentasi antara objek makanan yang sudah dipisahkan dengan objek latar belakang hitam. Kemudian ekstraksi fitur hanya dilakukan dengan menggunakan nilai dari objek makanan.

4.1.2 Ekstraksi Fitur *HSV Color Moment*

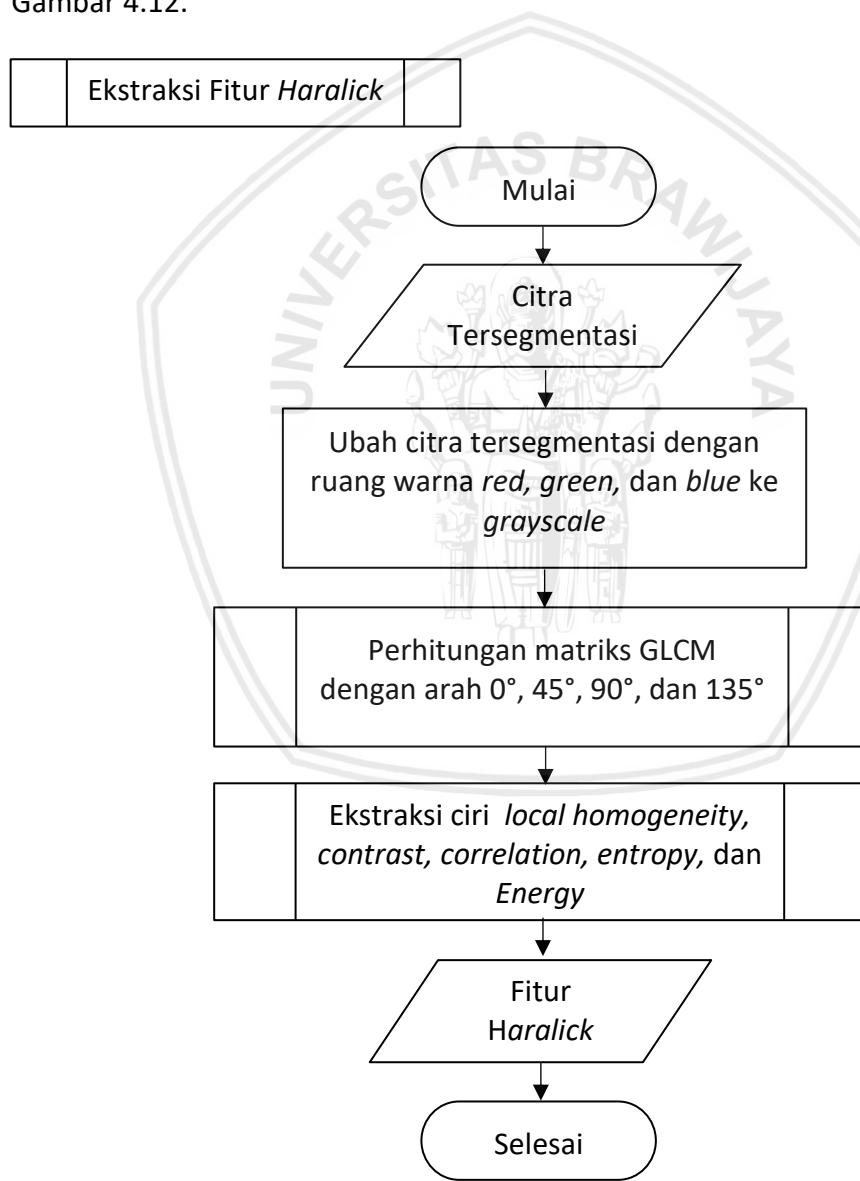
Color Moment merupakan representasi karakteristik citra warna, pembanding citra untuk mendapatkan *Color Similarity* sebuah citra yang disimpan dalam sebuah *database* citra. Pengambilan nilai fitur diambil dari citra makanan yang sudah dilakukan *pre-processing* pada tahap sebelumnya dengan hasil keluaran berupa nilai fitur. Fitur-fitur *color moment* yang digunakan pada penelitian ini yaitu *mean*, *standar deviasi*, dan *skewness*. Kemudian dari setiap fitur dihitung sesuai dengan ruang warna *hue*, *saturation*, dan *value*, sehingga menghasilkan 9 fitur. Diagram alir untuk *HSV color moment* dapat dilihat pada Gambar 4.11.



Gambar 4.11 Diagram alir *HSV Color Moment*

4.1.3 Ekstraksi Fitur Haralick

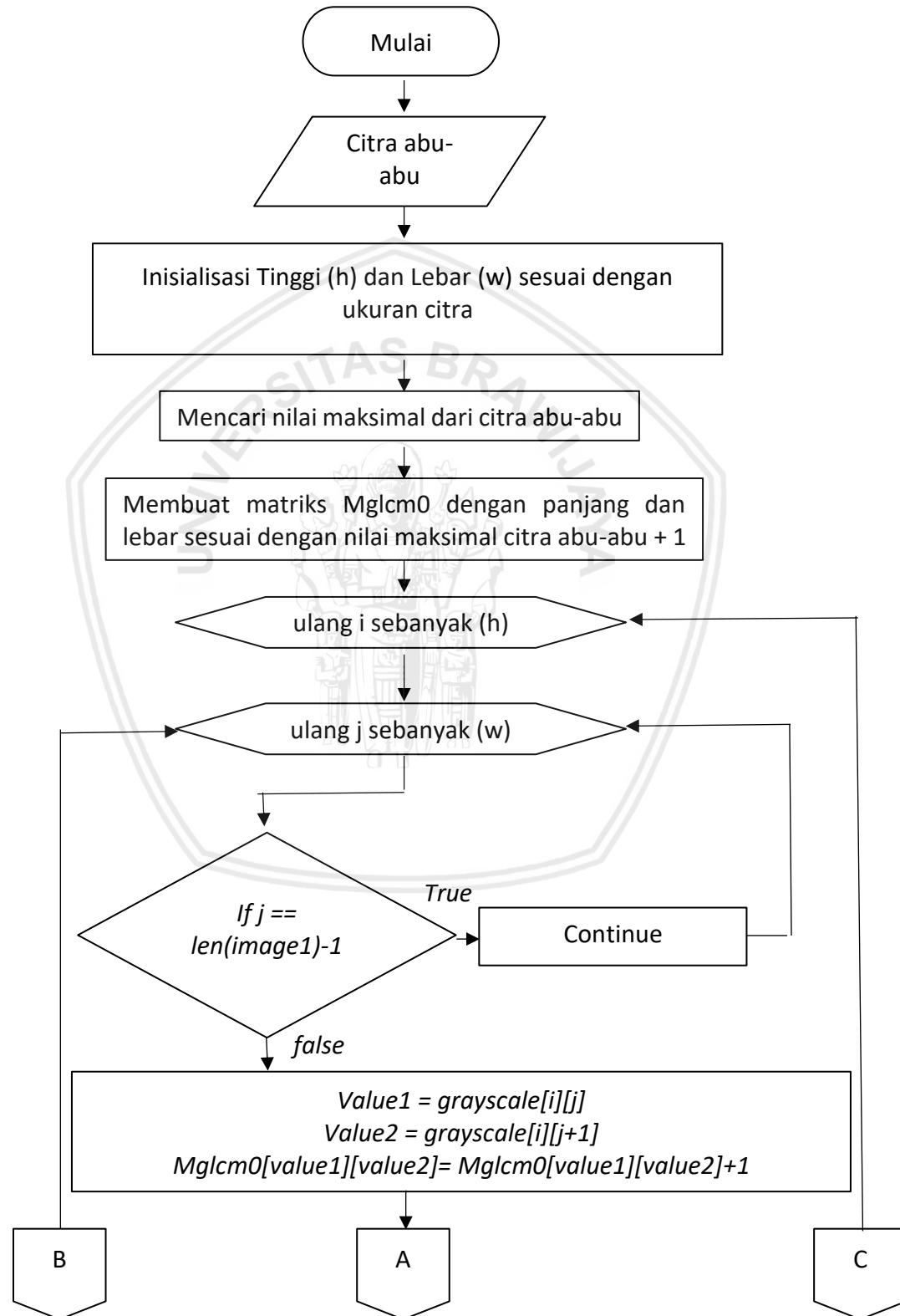
Ekstraksi fitur *haralick* merupakan fitur yang memanfaatkan tekstur dari suatu citra abu-abu. Sama seperti *gray level co-occurrence matrix* (GLCM), proses perhitungan *haralick* juga di awali dengan perhitungan matriks GLCM dengan arah 0° , 45° , 90° , dan 135° . Kemudian setelah matriks GLCM terbentuk, buat matriks baru dengan nilai hasil matriks *transpose* matriks GLCM, dan jumlahkan matriks GLCM dengan matriks *transpose*. Setelah itu, perlu dilakukan normalisasi agar nilai dari seluruh elemen matriks berjumlah 1, dengan membagi nilai dari setiap elemen matriks dengan total nilai keseluruhan matriks. Pada penelitian ini menggunakan 5 fitur ciri, yaitu *local homogeneity*, *contrast*, *correlation*, *entropy*, dan *Energy*. Diagram alir untuk perhitungan fitur *haralick* dapat dilihat pada Gambar 4.12.

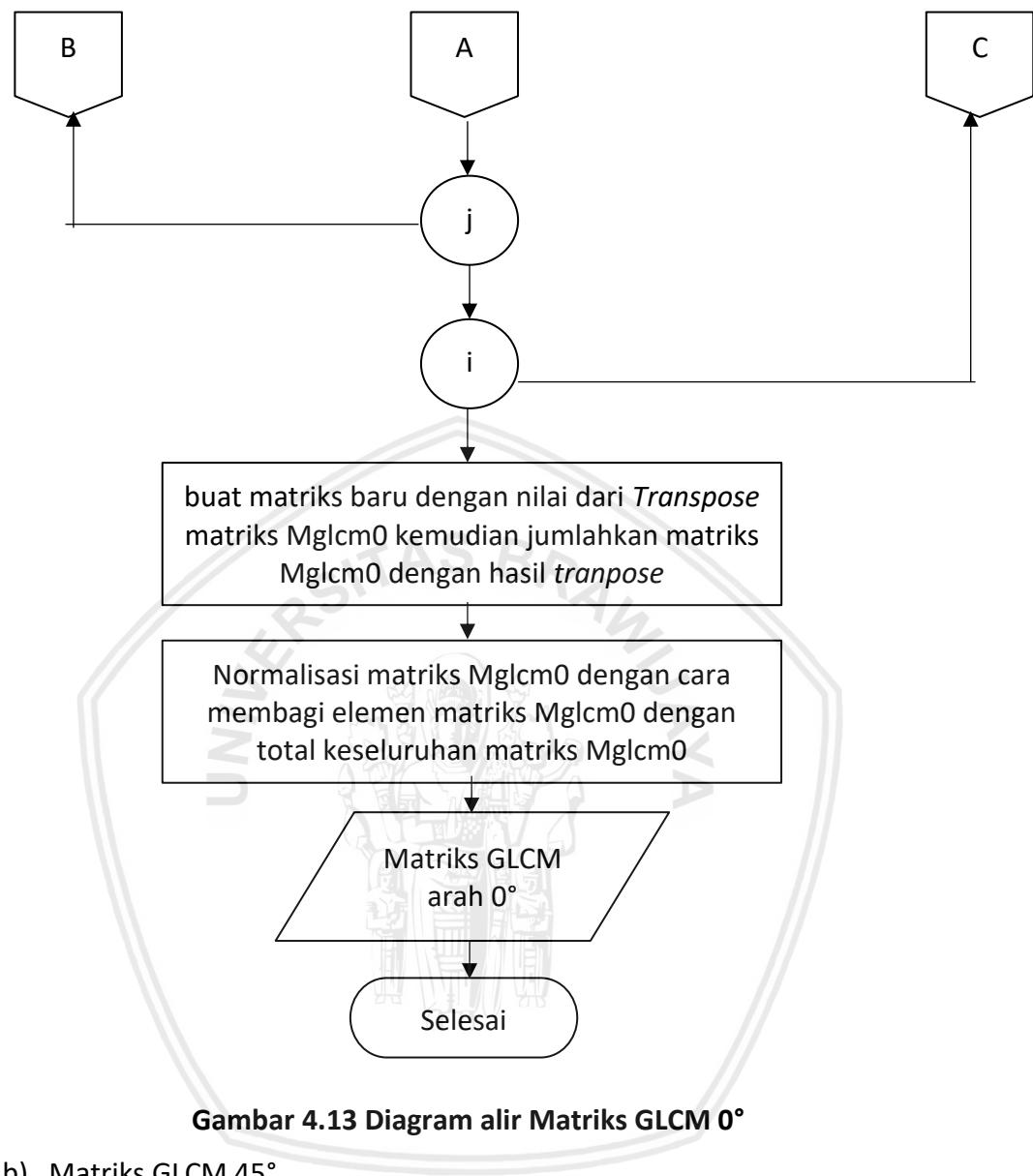


Gambar 4.12 Diagram alir Ekstraksi Fitur Haralick

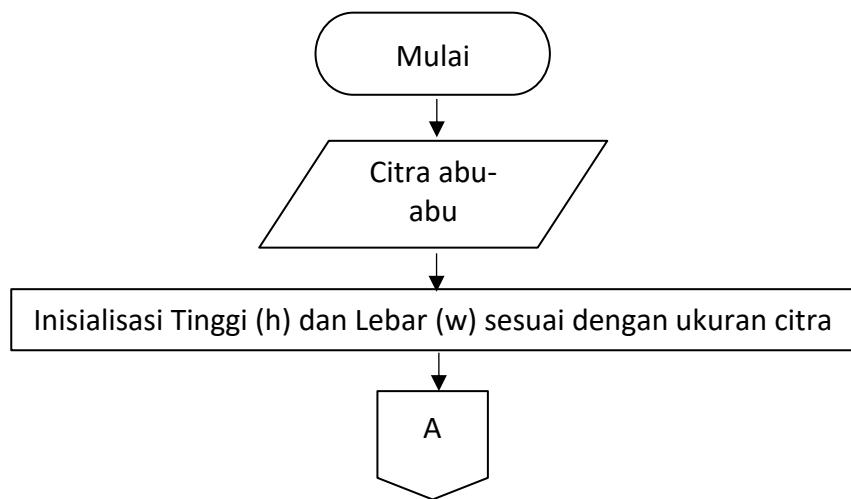
4.1.3.1 Perhitungan matriks GLCM

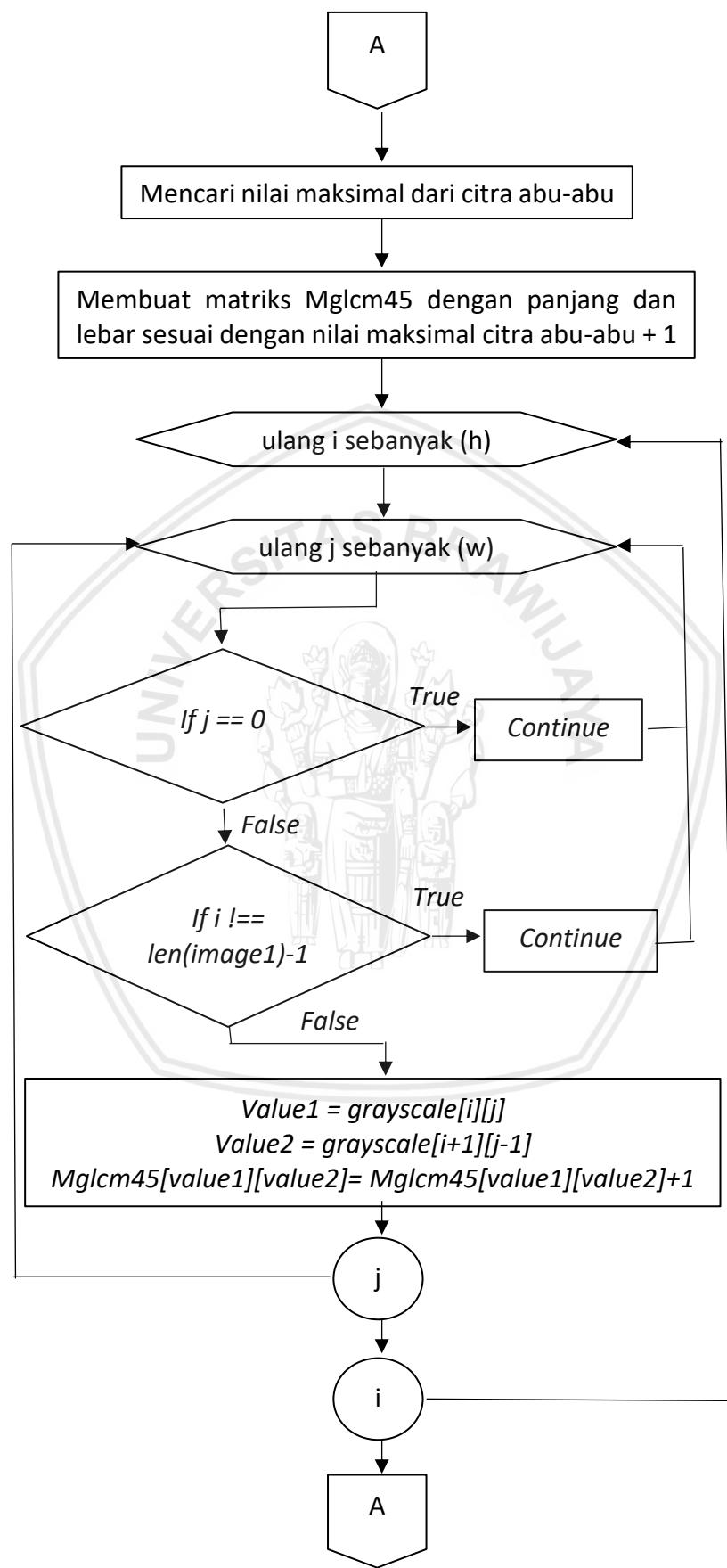
	Perhitungan matriks GLCM dengan arah 0° , 45° , 90° , dan 135°	
a)	Matriks GLCM 0°	

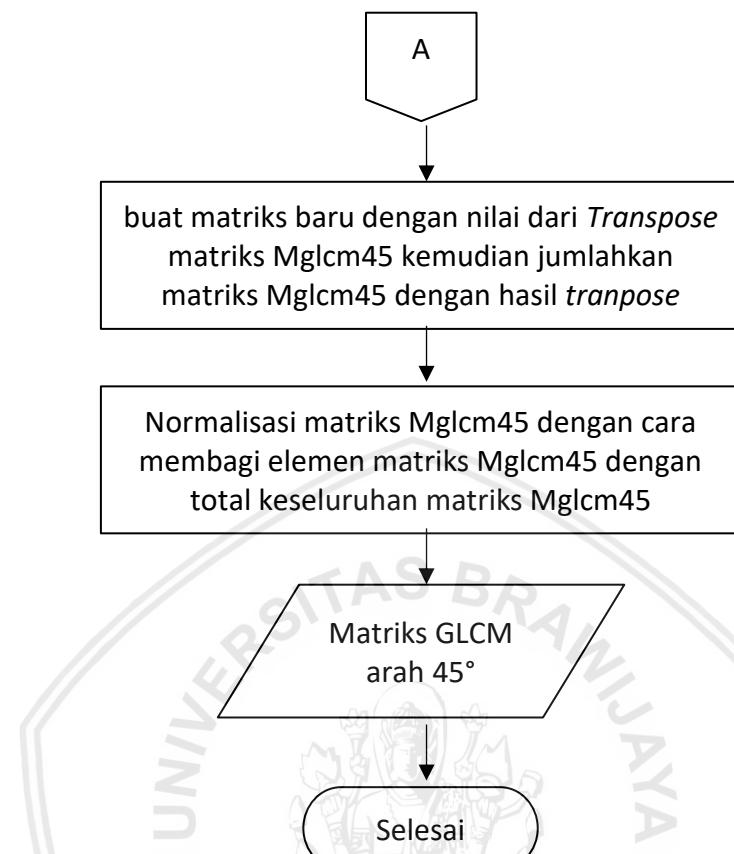




b) Matriks GLCM 45°

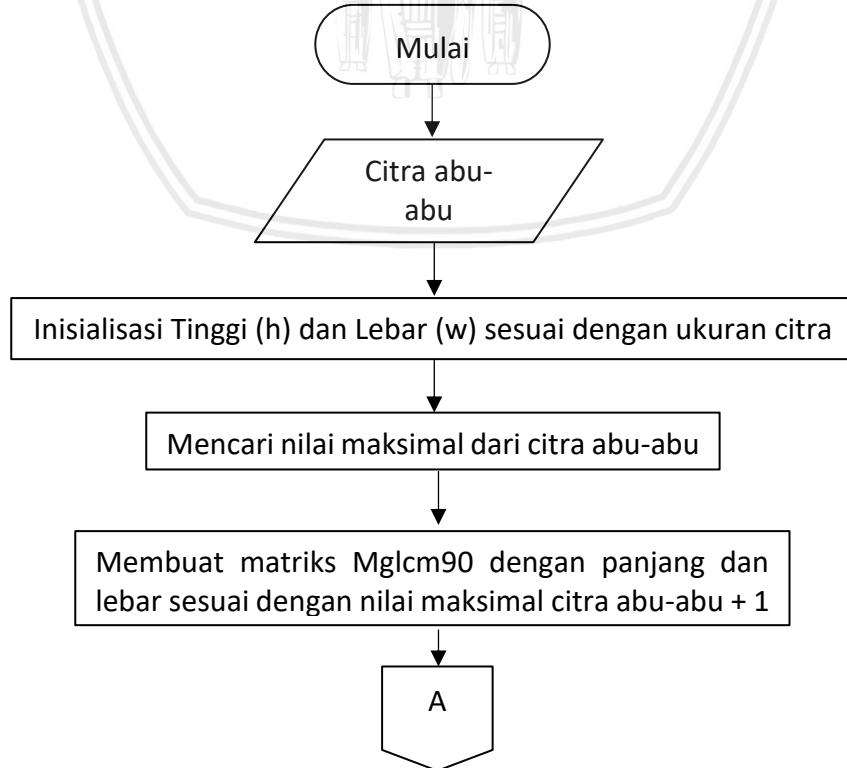


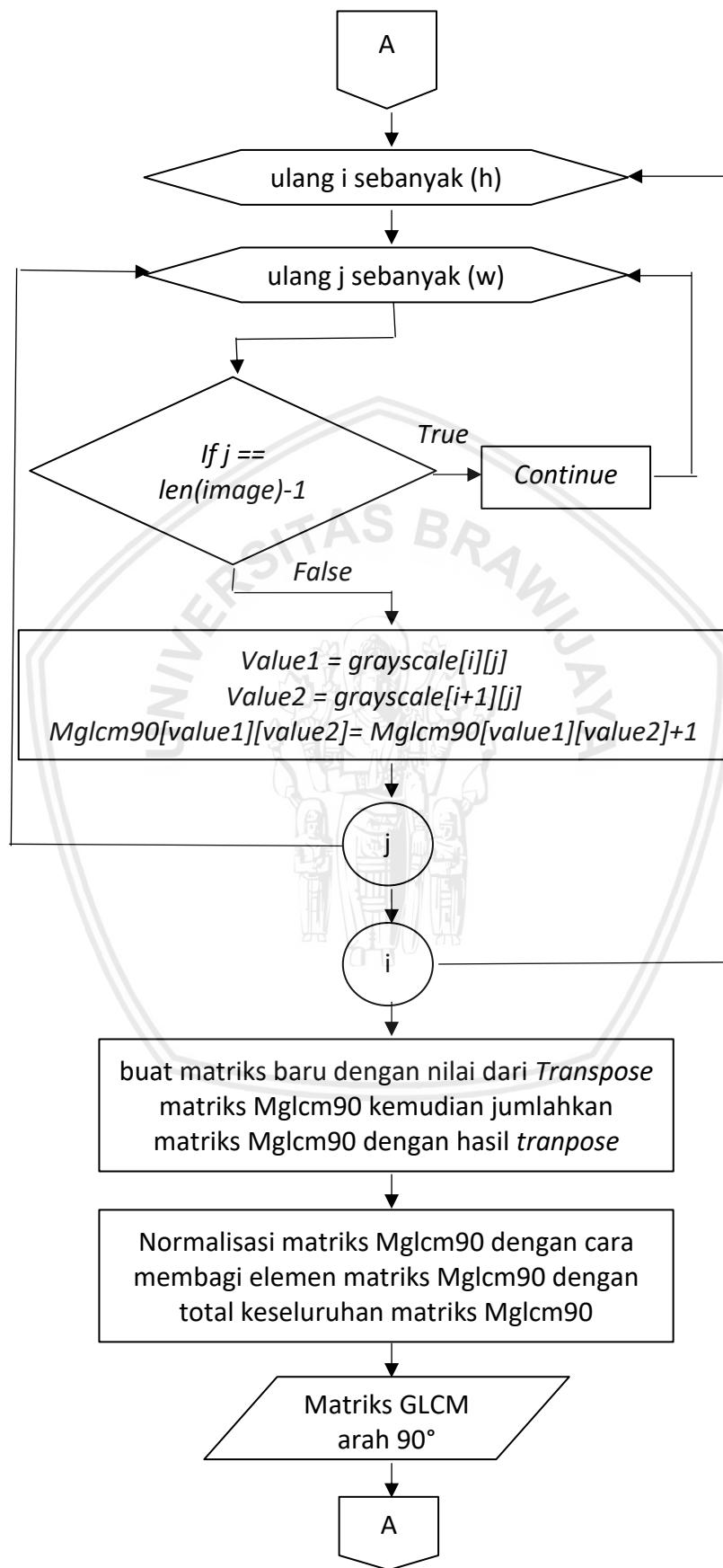


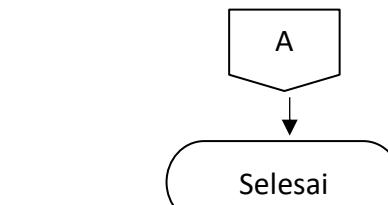


Gambar 4.14 Diagram alir Matriks GLCM 45°

c) Matriks GLCM 90°

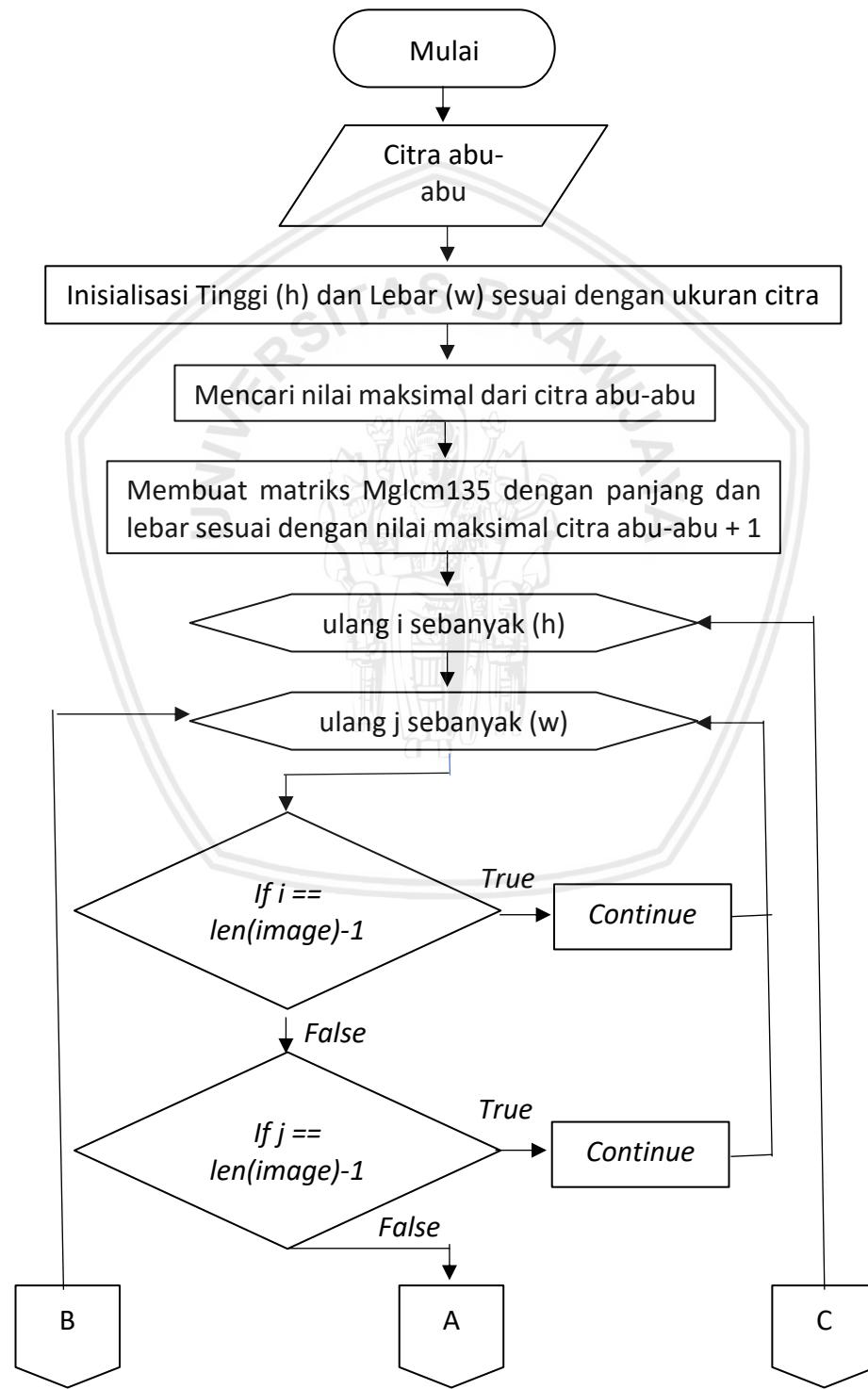


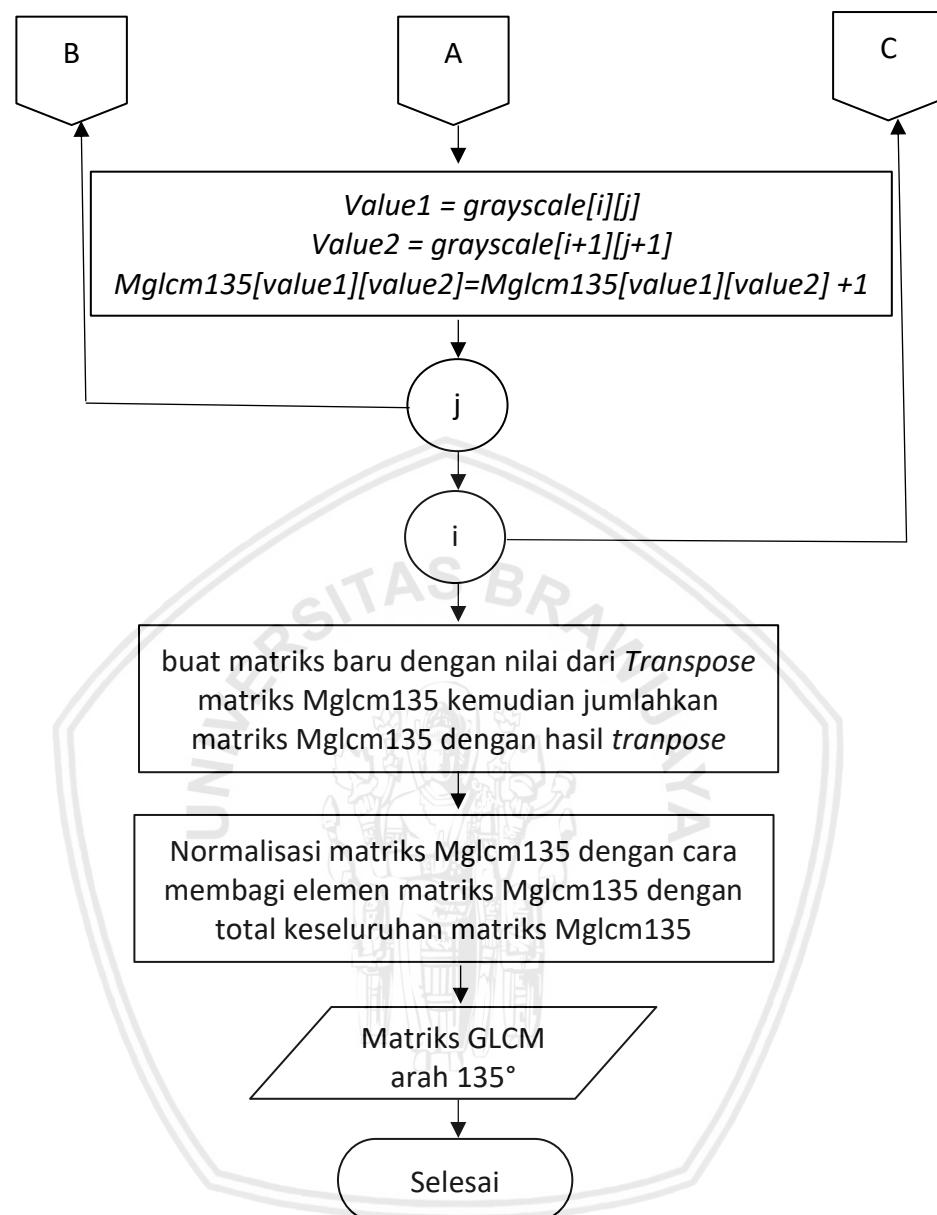




Gambar 4.15 Diagram alir Matriks GLCM 90°

d) Matriks GLCM 135°



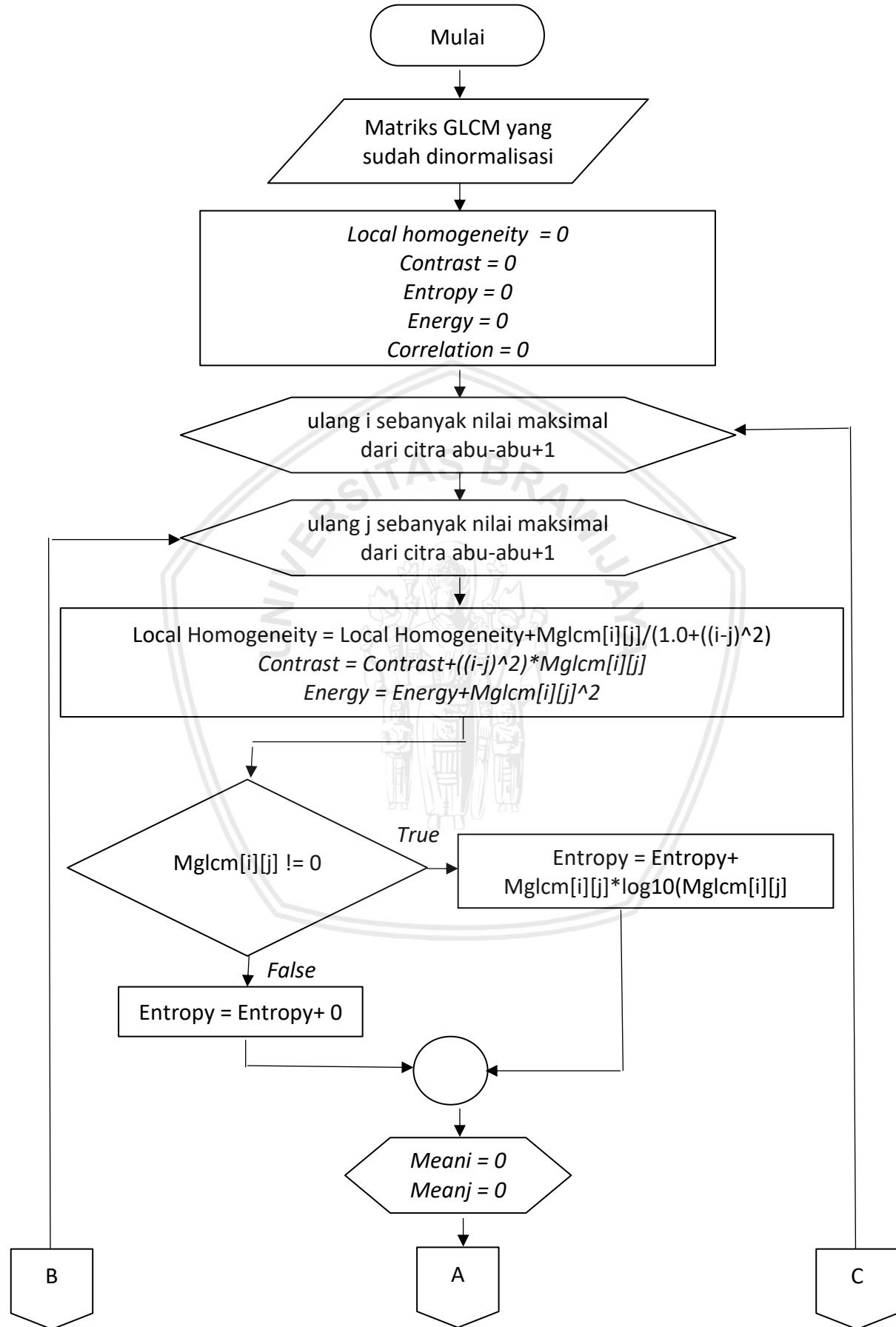


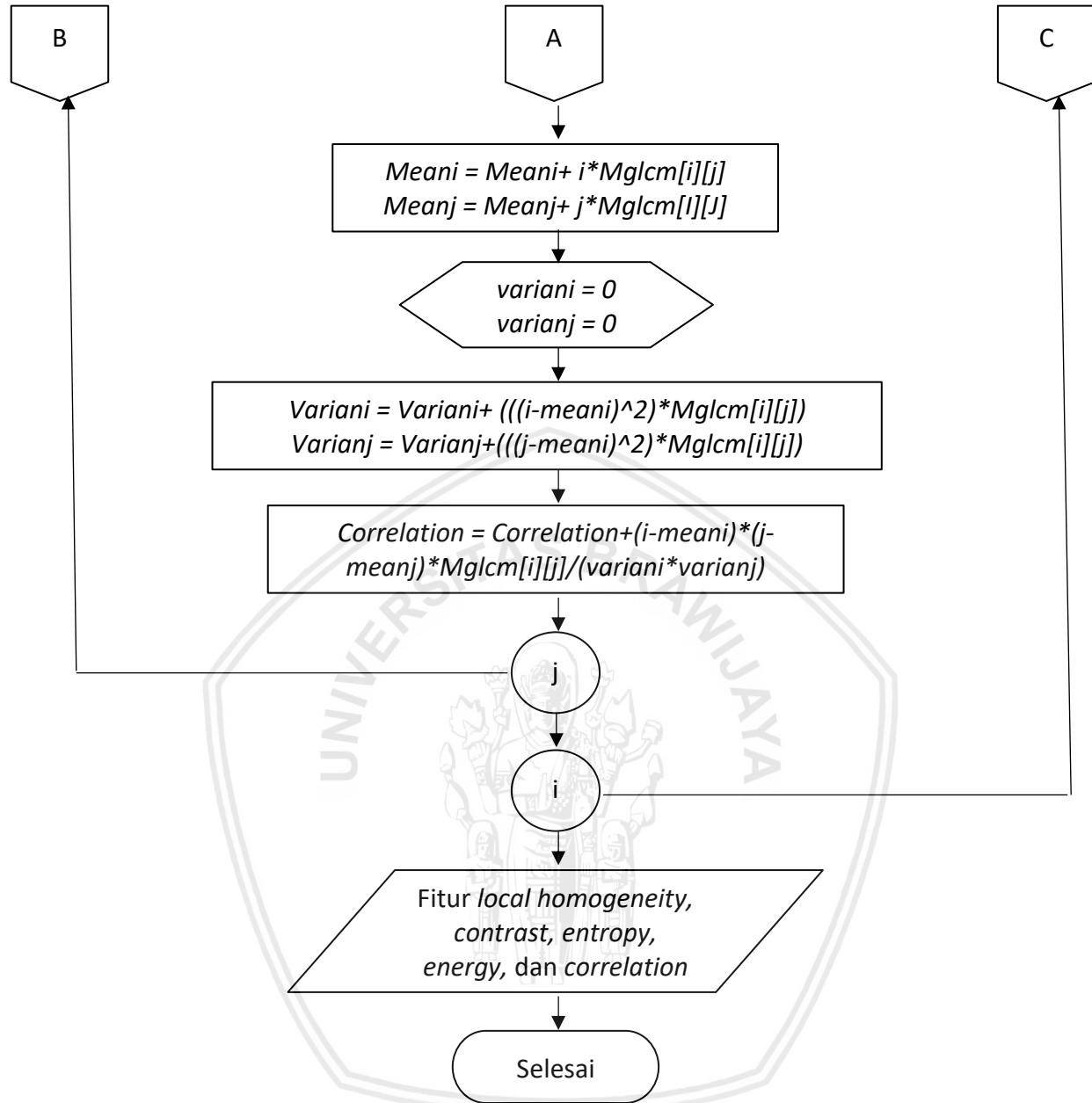
Gambar 4.16 Diagram alir Matriks GLCM 135°

4.1.3.2 Ekstraksi Ciri Haralick

	Ekstraksi ciri <i>local homogeneity</i> , <i>contrast</i> , <i>correlation</i> , <i>entropy</i> , dan <i>Energy</i>	
--	---	--

Ekstraksi ciri matriks GLCM dengan arah 0°, 45°, 90°, dan 135°.



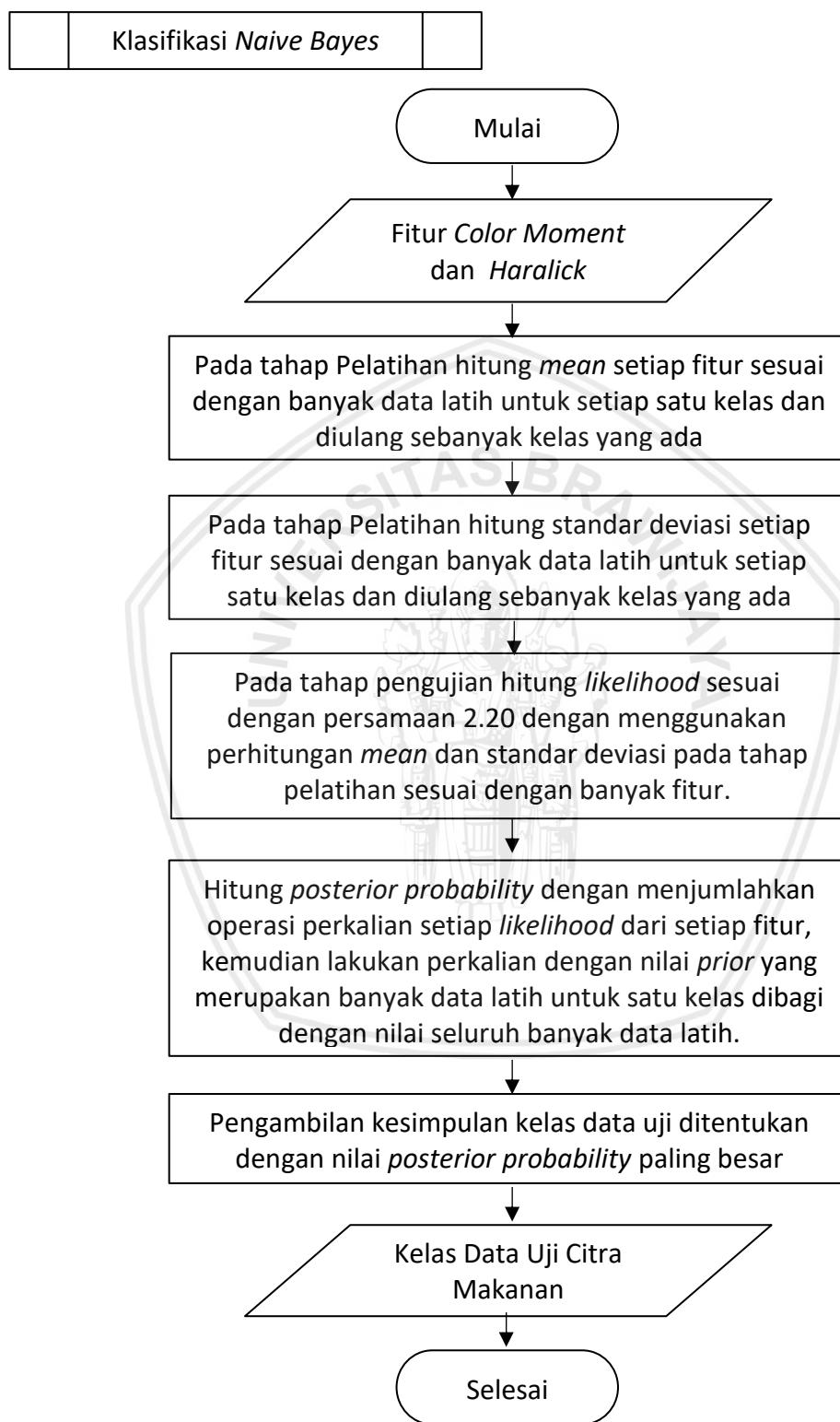


Gambar 4.17 Diagram alir Ekstraksi ciri *Haralick*

4.1.4 Klasifikasi *Naive Bayes*

Pada penelitian ini metode klasifikasi yang digunakan adalah *naive bayes classifier*. Proses klasifikasi menggunakan data fitur warna dan fitur tekstur yang telah dilakukan proses ekstraksi fitur pada tahap sebelumnya. Pada tahap *training* dilakukan proses perhitungan nilai rata-rata dan standar deviasi pada setiap fitur di kelas yang sama. Kemudian pada tahap *testing* dilakukan perhitungan *likelihood* berdasarkan Persamaan 2.19 untuk setiap fitur yang ada. Berdasarkan nilai *likelihood*, dapat ditentukan nilai *posterior probability* atau nilai kemungkinan kelas yang terpilih dengan menggabungkan nilai *likelihood* setiap fitur dengan operasi perkalian. Tahap akhir, berdasarkan nilai tertinggi *posterior probability*

dari setiap kelas dapat dilakukan pengambilan keputusan berdasarkan nilai data uji yang ada. Diagram alir untuk *naive bayes classifier* dapat dilihat pada Gambar 4.18.



Gambar 4.18 Diagram Alir *Naive Bayes Classifier*

4.2 Perhitungan Manual

perhitungan manual digunakan pada citra makanan yang diresize dengan ukuran 3x3. Tahapan-tahapan dalam perhitungan manual meliputi *pre-processing*, ekstraksi ciri *color moment*, *haralick*, *simple morphological shape descriptors*, dan klasifikasi *naive bayes*. Contoh citra asli makanan sebelum dilakukan *pre-processing* dapat dilihat pada Gambar 4.19.



Gambar 4.19 Citra Asli

4.2.1 Perhitungan *pre-processing*

Tahapan-tahapan perhitungan manual *pre-processing* adalah ubah ke *Saturation*, yaitu mengubah ruang warna *red*, *green*, dan *blue* menjadi *saturation*. Tabel untuk *red*, *green*, *blue* dapat dilihat pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3

Tabel 4.1 Nilai *Red* dari citra makanan yang diresize

i/j	1	2	3
1	195	77	211
2	198	192	144
3	195	191	173

Tabel 4.2 Nilai *Green* dari citra makanan yang diresize

i/j	1	2	3
1	194	19	210
2	198	137	133
3	199	190	176

Tabel 4.3 Nilai *Blue* dari citra makanan yang diresize

i/j	1	2	3
1	198	3	214
2	197	19	113
3	203	195	175

Kemudian ambil 1 piksel sebagai contoh untuk perhitungan ubah ke *saturation* pada indeks i = 1 dan j = 1. Maka ditentukan nilai dari maksimal dan minimal dari nilai *red*, *green*, dan *blue*. Rumus untuk ubah ke *saturation* dapat dilihat pada Persamaan 2.2 .

Red = 195, *Green* = 194, *Blue* = 198.

maxRGB = 198(*blue*) dan minRGB = 194 (*green*)

$$S = \left(1 - \frac{194}{198}\right) \times 255 = 0,02 \times 255 = 5$$

Sehingga hasil perhitungan ubah ke *saturation* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Nilai *Saturation* Hasil Perhitungan Ubah ke *Saturation*

S	1	2	3
1	5	245	4
2	1	229	56
3	10	6	4

Gambar contoh hasil perhitungan hasil ubah ke *saturation* dapat dilihat pada Gambar 4.20.

**Gambar 4.20 Citra *saturation***

Tahap *pre-processing* berikutnya adalah *gausian blurring* melakukan konvolusi citra dengan memanfaatkan nilai piksel tetangga. Tabel untuk kernel *Gaussian blur* 3x3 dapat dilihat pada Tabel 4.5.

Tabel 4.5 Gaussian Blur Kernel

1	2	1
2	4	2
1	2	1

Dari Tabel 4.5 digunakan untuk perhitungan untuk piksel baris ke 1 dan kolom ke 1 pada nilai *saturation*. Diperlukan bantuan tambahan piksel di sekeliling citra bernilai 0 untuk membantu perhitungan. Berikut merupakan contoh perhitungan *gaussian blurring* menggunakan kernel 3x3.

Tabel 4.6 Perhitungan Gaussian Blurring

0	0	0	0	0
0	5	245	4	0
0	1	229	56	0
0	10	6	4	0
0	0	0	0	0

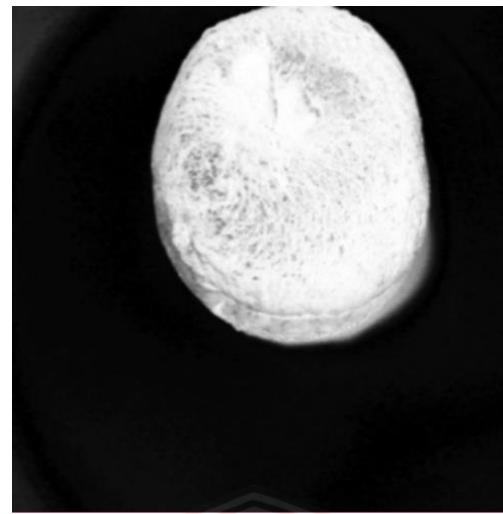
$$\begin{aligned} \text{Blur} &= \frac{(1*0)+(2*0)+(1*0)+(2*0)+(4*5)+(2*245)+(1*0)+(2*1)+(1*229)}{16} \\ &= \frac{0+\dots+20+490+\dots+229}{16} = \frac{741}{16} = 46,3125 \approx 46 \end{aligned}$$

Hasil perhitungan *gaussian blurring* dapat dilihat pada Tabel 4.7 berikut.

Tabel 4.7 Hasil Perhitungan Gaussian Blurring

GB	1	2	3
1	46	95	53
2	46	97	59
3	18	35	23

Gambar contoh hasil perhitungan *gaussian blurring* dapat dilihat pada Gambar 4.21.



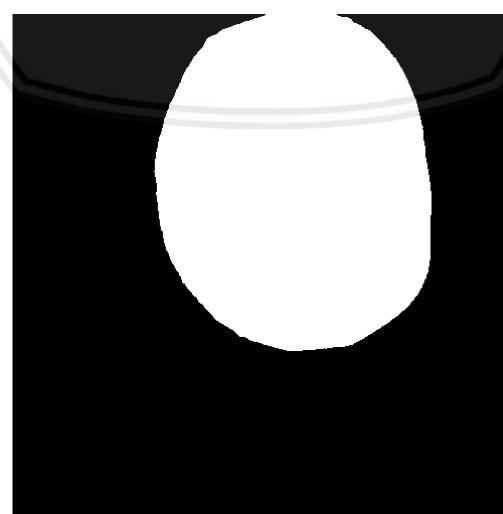
Gambar 4.21 Citra Gaussian Blur

Tahap *Preprocessing* berikutnya adalah segmentasi citra, yaitu dengan melakukan *thresholding* citra, mengubah citra *gaussian blur* menjadi citra biner. *Thershold* yang digunakan untuk perhitungan manual adalah menggunakan nilai 83. Nilai piksel yang kurang dari 83 akan bernilai 0 dan nilai piksel yang bernilai lebih dari atau sama dengan 83 akan bernilai 1. Tabel hasil perhitungan *thresholding* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Perhitungan *Thresholding*

TH	1	2	3
1	0	1	0
2	0	1	0
3	0	0	0

Gambar contoh hasil perhitungan *Thresholding* dapat dilihat pada Gambar 4.22.



Gambar 4.22 Citra *Thresholding*

Tahap *pre-processing* berikutnya adalah erosi dan dilasi. Langkah awal erosi dan dilasi adalah dengan membuat *structuring element (SE)* yang digunakan untuk dibandingkan dengan nilai piksel citra biner. Contoh *structuring element* yang digunakan berupa matriks berukuran 3x3 dan dapat dilihat pada Tabel 4.9.

Tabel 4.9 Structuring Element

SE	1	2	3
1	1	1	1
2	1	1	1
3	1	1	1

Perhitungan erosi adalah dengan membanding nilai piksel citra biner dengan *structuring element (SE)*. Jika nilai piksel sama secara keseluruhan, maka nilai piksel akan bernilai 1. Diperlukan bantuan tambahan piksel di sekeliling citra bernilai 0 untuk membantu perhitungan. Berikut merupakan contoh perhitungan erosi.

Tabel 4.10 Perhitungan Erosi

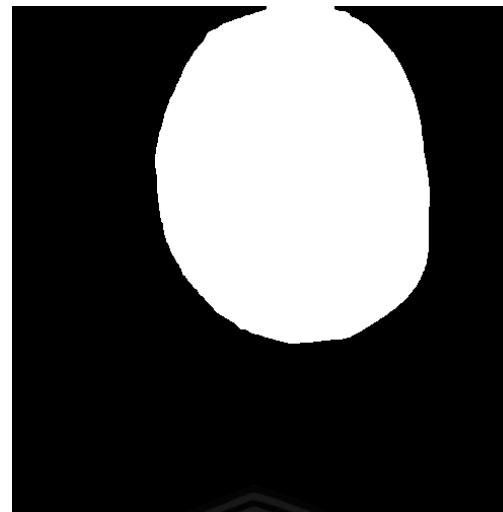
0	0	0	0	0
0	0	1	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Hasil perhitungan Erosi dapat dilihat pada Tabel 4.11 berikut.

Tabel 4.11 Hasil Perhitungan Erosi

E	1	2	3
1	0	0	0
2	0	0	0
3	0	0	0

Gambar contoh hasil perhitungan erosi dapat dilihat pada Gambar 4.23.



Gambar 4.23 Citra Erosi

Perhitungan Dilasi adalah dengan membanding nilai piksel citra biner dengan *structuring element* (SE). Jika terdapat nilai piksel yang sama, maka nilai piksel akan bernilai 1. Diperlukan bantuan tambahan piksel di sekeliling citra bernilai 0 untuk membantu perhitungan. Berikut merupakan contoh perhitungan erosii.

Tabel 4.12 Perhitungan Dilasi

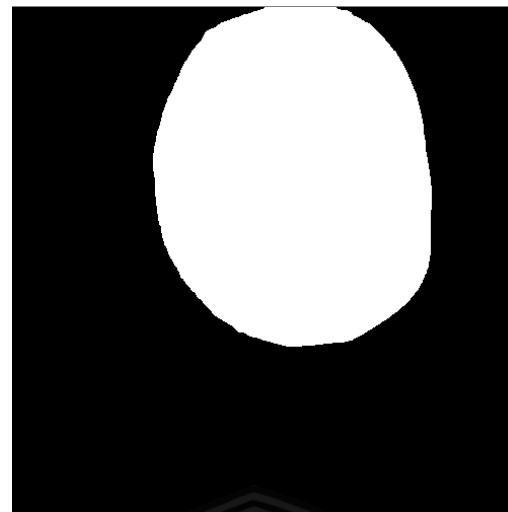
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Hasil perhitungan Dilasi dapat dilihat pada Tabel 4.13 berikut.

Tabel 4.13 Hasil Perhitungan Dilasi

D	1	2	3
1	0	0	0
2	0	0	0
3	0	0	0

Gambar contoh hasil perhitungan dilasi dapat dilihat pada Gambar 4.24.



Gambar 4.24 Citra Dilasi

Langkah terakhir adalah melakukan konversi citra dari dilasi menjadi *masking* menggunakan operator *bitwise and*. Gambar hasil akhir proses *pre-processing* dapat dilihat pada Gambar 4.25.



Gambar 4.25 Citra Setelah *Pre-processing*

4.2.2 Perhitungan HSV Color Moment

Tahap pertama dalam perhitungan HSV *Color Moment* adalah mengubah nilai ruang warna *red*, *green*, dan *blue* ke dalam ruang warna *hue*, *saturation*, dan *value*. Perhitungan untuk mengubah nilai RGB ke dalam *Hue* dapat dilihat pada Persamaan 2.1.

Tabel 4.14 Nilai Hue

Hue	1	2	3
1	0,7083	0,0373	0,7083
2	0,6667	0,1133	0,1075
3	0,5833	0,7333	0,4444

Kemudian untuk perhitungan mengubah nilai RGB ke dalam *saturation* dapat dilihat pada Persamaan 2.2.

Tabel 4.15 Nilai Saturation

S	1	2	3
1	0,02	0,9157	0,0188
2	0,005	0,8883	0,2123
3	0,0396	0,0256	0,0171

Perhitungan untuk mengubah nilai RGB ke dalam *value* dapat dilihat pada Persamaan 2.3.

Tabel 4.16 Nilai Value

V	1	2	3
1	0,7843	0,3255	0,8353
2	0,7843	0,7725	0,5725
3	0,7922	0,7647	0,6863

Ruang warna yang digunakan dalam menghitung *color moment* adalah *hue*, *saturation*, dan *value*. Dimana untuk 3 fitur *color moment* yaitu mean, standar deviasi, dan skewness dilakukan perhitungan pada setiap ruang warna H, S, dan V. Perhitungan *Mean* dapat dilihat pada Persamaan 2.4

$$\text{Mean Hue} = \frac{0,7083 + 0,0373 + \dots + 0,444}{9} = \frac{4,1024}{9} = 0,4558$$

Perhitungan standar deviasi dapat dilihat pada Persamaan 2.5.

Standar deviasi Hue =

$$\sqrt[2]{\frac{((0,7083-0,4558)^2+(0,0373-0,4558)^2+\dots+(0,4444-0,4558)^2)}{9}} = 0,2747$$

Perhitungan *skewness* dapat dilihat pada Persamaan 2.6.

$$\begin{aligned} \text{Skewness Hue} &= \sqrt[3]{\frac{((0,7083-0,4558)^3+(0,0373-0,4558)^3+\dots+(0,4444-0,4558)^3)}{N}} \\ &= -0,216044 \end{aligned}$$

Diatas merupakan contoh perhitungan dari fitur-fitur *color moment*. Hasil perhitungan *mean*, standar deviasi, dan *skewness* untuk ruang warna lengkap untuk *Hue*, *saturation*, dan *value* dapat dilihat pada Tabel 4.8, Tabel 4.9, Tabel 4.10.

Tabel 4.17 Mean Hue, Saturation, dan Value

	Hue	Saturation	Value
Mean	0,4558222	0,23804	0,70196

Tabel 4.18 Standar Deviasi Hue, Saturation, dan value

	Hue	Saturation	Value
Standar Deviasi	0,2747009	0,35993	0,1517

Tabel 4.19 Skewness Hue, Saturation, dan value

	Hue	Saturation	Value
Skewness	-0,216044	0,38767	-0,1779

4.2.3 Perhitungan Haralick

Tahap selanjutnya setelah dilakukan *pre-processing* dan ekstraksi ciri *color moment* adalah menghitung fitur *haralick*. Tedapat 2 tahapan dalam perhitungan fitur *haralick*, yang pertama adalah perhitungan matriks *gray-level occurrence matrix* sampai ternomalisasi. Kedua perhitungan ekstraksi 5 fitur *Haralick* yaitu *local homogeneity, contrast, correlation, entropy, dan energy*.

4.2.3.1 Perhitungan matriks *gray level occurrence matrix*

Perhitungan matriks GLCM dibuat berdasarkan nilai grayscale dan dilakukan dengan penentuan arah 0°, 45°, 90°, dan 135°, selain ditentukan nilai ketetanggaan atau *distance* yaitu sebesar 1. Kemudian besar matriks GLCM ditentukan sesuai dengan nilai maksimal citra *grayscale*.

- a) Matriks GLCM dengan arah 0°

Cara penentuan pasangan *pixel* untuk matriks GLCM dengan arah 0° dapat dilihat pada Tabel 4.20.

Tabel 4.20 Perhitungan Pasangan Matriks GLCM dengan arah 0°

gray	1	2	3
1	193 → 30 → 208		
2	196 → 139 → 133		
3	196 → 189 → 174		

Berdasarkan Tabel 4.20 jumlah pasangan pixel dengan pasangan (193 dan 30) sebanyak 1 pasang. Nilai jumlah pasangan tersebut kemudian dimasukkan kedalam matriks GLCM yang dapat dilihat pada Tabel 4.21

Tabel 4.21 Matriks GLCM dengan arah 0°

i/j	0	...	30	133	139	174	189	193	196	...	208
-----	---	-----	----	-----	-----	-----	-----	-----	-----	-----	-----

0	0	...	0	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	0	0	0	...	1
133	0	...	0	0	0	0	0	0	0	0	...	0
139	0	...	0	1	0	0	0	0	0	0	...	0
174	0	...	0	0	0	0	0	0	0	0	...	0
189	0	...	0	0	0	1	0	0	0	0	...	0
193	0	...	1	0	0	0	0	0	0	0	...	0
196	0	...	0	0	1	0	1	0	0	0	...	0
...	0
208	0	...	0	0	0	0	0	0	0	0	...	0

Langkah selanjutnya membuat matriks transpose dari matriks GLCM, dan menjumlahkan matriks transpose dengan matriks GLCM. Tabel transpose dan Tabel hasil jumlah tranpose matriks GLCM dengan matriks GLCM dilihat pada Tabel 4.22 dan Tabel 4.23.

Tabel 4.22 Transpose matriks GLCM dengan arah 0°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	1	0	...	0
133	0	...	0	0	1	0	0	0	0	...	0
139	0	...	0	0	0	0	0	0	1	...	0
174	0	...	0	0	0	0	1	0	0	...	0
189	0	...	0	0	0	0	0	0	1	...	0
193	0	...	0	0	0	0	0	0	0	...	0
196	0	...	0	0	0	0	0	0	0	...	0
...	0
208	0	...	1	0	0	0	0	0	0	...	0

Tabel 4.23 Hasil Jumlah matriks GLCM dan transpose dengan arah 0°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0

0	0	...	0	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	1	0	...	1	
133	0	...	0	0	1	0	0	0	0	...	0	
139	0	...	0	1	0	0	0	0	1	...	0	
174	0	...	0	0	0	0	1	0	0	...	0	
189	0	...	0	0	0	1	0	0	1	...	0	
193	0	...	1	0	0	0	0	0	0	...	0	
196	0	...	0	0	1	0	1	0	0	...	0	
...	0
208	0	...	1	0	0	0	0	0	0	...	0	

Langkah selanjutnya adalah normalisasi matriks GLCM agar jumlah dari seluruh elemen matriks GLCM adalah 1. Cara untuk normalisasi adalah dengan membagi elemen matriks GLCM dengan nilai total keseluruhan matriks GLCM. Hasil perhitungan normalisasi dapat dilihat pada Tabel 4.24.

Misalkan indeks ke $i = 30$ dan $j = 208$, jumlah keseluruhan matriks = 12

$$M(30, 208) = \frac{1}{12} = 0,083$$

Tabel 4.24 Normalisasi Matriks GLCM dengan arah 0°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	0,083	0	...	0,083
133	0	...	0	0	0,083	0	0	0	0	...	0
139	0	...	0	0,083	0	0	0	0	0,083	...	0
174	0	...	0	0	0	0	0,083	0	0	...	0
189	0	...	0	0	0	0,083	0	0	0,083	...	0
193	0	...	0,083	0	0	0	0	0	0	...	0
196	0	...	0	0	0,083	0	0,083	0	0	...	0
...	0
208	0	...	0,083	0	0	0	0	0	0	...	0

b) Matriks GLCM dengan arah 45°

Cara penentuan pasangan *pixel* untuk matriks GLCM dengan arah 45° dapat dilihat pada Tabel 4.25.

Tabel 4.25 Perhitungan Pasangan Matriks GLCM dengan arah 45°

gray	1	2	3
1	193	30	208
2	196	139	133
3	196	189	174

Berdasarkan Tabel 4.25 jumlah pasangan pixel dengan pasangan (30 dan 196) sebanyak 1 pasang. Nilai jumlah pasangan tersebut kemudian dimasukkan kedalam matriks GLCM yang dapat dilihat pada Tabel 4.26

Tabel 4.26 Matriks GLCM dengan arah 45°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	0	1	...	0
133	0	...	0	0	0	0	1	0	0	...	0
139	0	...	0	0	0	0	0	0	1	...	0
174	0	...	0	0	0	0	0	0	0	...	0
189	0	...	0	0	0	0	0	0	0	...	0
193	0	...	0	0	0	0	0	0	0	...	0
196	0	...	0	0	0	0	0	0	0	...	0
...	0
208	0	...	0	0	1	0	0	0	0	...	0

Langkah selanjutnya membuat matriks *transpose* dari matriks GLCM, dan menjumlahkan matriks *transpose* dengan matriks GLCM. Tabel *transpose* dan Tabel hasil jumlah *tranpose* matriks GLCM dengan matriks GLCM dilihat pada Tabel 4.27 dan Tabel 4.28.

Tabel 4.27 Transpose matriks GLCM dengan arah 45°

Tabel 4.27 Transpose matriks GLCM dengan arah 45° (Lanjutan)

133	0	...	0	0	0	0	0	0	0	0	...	0
139	0	...	0	0	0	0	0	0	0	0	...	1
174	0	...	0	0	0	0	0	0	0	0	...	0
189	0	...	0	1	0	0	0	0	0	0	...	0
193	0	...	0	0	0	0	0	0	0	0	...	0
196	0	...	1	0	1	0	0	0	0	0	...	0
...	0
208	0	...	0	0	0	0	0	0	0	0	...	0

Tabel 4.28 Hasil Jumlah matriks GLCM dan transpose dengan arah 45°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	0	1	...	0
133	0	...	0	0	0	0	1	0	0	...	0
139	0	...	0	0	0	0	0	0	1	...	1
174	0	...	0	0	0	0	0	0	0	...	0
189	0	...	0	1	0	0	0	0	0	...	0
193	0	...	0	0	0	0	0	0	0	...	0
196	0	...	1	0	1	0	0	0	0	...	0
...	0
208	0	...	0	0	1	0	0	0	0	...	0

Langkah selanjutnya adalah normalisasi matriks GLCM agar jumlah dari seluruh elemen matriks GLCM adalah 1. Cara untuk normalisasi adalah dengan membagi elemen matriks GLCM dengan nilai total keseluruhan matriks GLCM. Hasil perhitungan normalisasi dapat dilihat pada Tabel 4.29.

Misalkan indeks ke i = 30 dan j = 196, jumlah keseluruhan matriks = 8

$$M(30, 196) = \frac{1}{8} = 0,125$$

Tabel 4.29 Normalisasi Matriks GLCM dengan arah 45°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	0	0,125	...	0

133	0	...	0	0	0	0	0,125	0	0	...	0
139	0	...	0	0	0	0	0	0	0,125	...	0,125
174	0	...	0	0	0	0	0	0	0	...	0
189	0	...	0	0,125	0	0	0	0	0	...	0
193	0	...	0	0	0	0	0	0	0	...	0
196	0	...	0,125	0	0,125	0	0	0	0	...	0
...	0
208	0	...	0	0	0,125	0	0	0	0	...	0

c) Matriks GLCM dengan arah 90°

Cara penentuan pasangan pixel untuk matriks GLCM dengan arah 90° dapat dilihat pada Tabel 4.30.

Tabel 4.30 Perhitungan Pasangan Matriks GLCM dengan arah 90°

gray	1	2	3
1	193	30	208
2	196	139	133
3	196	189	174

Berdasarkan Tabel 4.30 jumlah pasangan pixel dengan pasangan (193 dan 196) sebanyak 1 pasang. Nilai jumlah pasangan tersebut kemudian dimasukkan kedalam matriks GLCM yang dapat dilihat pada Tabel 4.31.

Tabel 4.31 Matriks GLCM dengan arah 90°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	1	0	0	0	0	...	0
133	0	...	0	0	0	1	0	0	0	...	0
139	0	...	0	0	0	0	1	0	0	...	0

Tabel 4.31 Matriks GLCM dengan arah 135° (Lanjutan)

i/j	0	...	30	133	139	174	189	193	196	...	208
174	0	...	0	0	0	0	0	0	0	...	0
189	0	...	0	0	0	0	0	0	0	...	0
193	0	...	0	0	0	0	0	0	1	...	0
196	0	...	0	0	0	0	0	0	1	...	0

...	0
208	0	...	0	1	0	0	0	0	0	0	...	0

Langkah selanjutnya membuat matriks *transpose* dari matriks GLCM, dan menjumlahkan matriks *transpose* dengan matriks GLCM. Tabel *transpose* dan Tabel hasil jumlah *tranpose* matriks GLCM dengan matriks GLCM dilihat pada Tabel 4.32 dan Tabel 4.33.

Tabel 4.32 Transpose matriks GLCM dengan arah 90°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	0	0	...	0
133	0	...	0	0	0	0	0	0	0	...	1
139	0	...	1	0	0	0	0	0	0	...	0
174	0	...	0	1	0	0	0	0	0	...	0
189	0	...	0	0	1	0	0	0	0	...	0
193	0	...	0	0	0	0	0	0	0	...	0
196	0	...	0	0	0	0	0	1	1	...	0
...	0
208	0	...	0	0	0	0	0	0	0	...	0

Tabel 4.33 Hasil Jumlah matriks GLCM dan transpose dengan arah 90°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	1	0	0	0	0	...	0
133	0	...	0	0	0	1	0	0	0	...	1
139	0	...	1	0	0	0	1	0	0	...	0

Tabel 4.32 Transpose matriks GLCM dengan arah 90° (Lanjutan)

i/j	0	...	30	133	139	174	189	193	196	...	208
174	0	...	0	1	0	0	0	0	0	...	0
189	0	...	0	0	1	0	0	0	0	...	0
193	0	...	0	0	0	0	0	0	1	...	0

196	0	...	0	0	0	0	0	1	2	...	0
...	0
208	0	...	0	1	0	0	0	0	0	...	0

Langkah selanjutnya adalah normalisasi matriks GLCM agar jumlah dari seluruh elemen matriks GLCM adalah 1. Cara untuk normalisasi adalah dengan membagi elemen matriks GLCM dengan nilai total keseluruhan matriks GLCM. Hasil perhitungan normalisasi dapat dilihat pada Tabel 4.34.

Misalkan indeks ke $i = 196$ dan $j = 196$, jumlah keseluruhan matriks = 12

$$M(196, 196) = \frac{2}{12} = 0.167$$

Tabel 4.34 Normalisasi Matriks GLCM dengan arah 90°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0.083	0	0	0	0	...	0
133	0	...	0	0	0	0.083	0	0	0	...	0.083
139	0	...	0.083	0	0	0	0.083	0	0	...	0
174	0	...	0	0.083	0	0	0	0	0	...	0
189	0	...	0	0	0.083	0	0	0	0	...	0
193	0	...	0	0	0	0	0	0	0.083	...	0
196	0	...	0	0	0	0	0	0.083	0.167	...	0
...	0
208	0	...	0	0.083	0	0	0	0	0	...	0

d) Matriks GLCM dengan arah 135°

Cara penentuan pasangan *pixel* untuk matriks GLCM dengan arah 135° dapat dilihat pada Tabel 4.35.

Tabel 4.35 Perhitungan Pasangan Matriks GLCM dengan arah 135°

gray	1	2	3
1	193	30	208
2	196	139	133
3	196	189	174

Berdasarkan Tabel 4.35 jumlah pasangan pixel dengan pasangan (193 dan 139) sebenyak 1 pasang. Nilai jumlah pasangan tersebut kemudian dimasukkan kedalam matriks GLCM yang dapat dilihat pada Tabel 4.36.

Tabel 4.36 Matriks GLCM dengan arah 135°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	1	0	0	0	0	0	...	0
133	0	...	0	0	0	0	0	0	0	...	0
139	0	...	0	0	0	1	0	0	0	...	0
174	0	...	0	0	0	0	0	0	0	...	0
189	0	...	0	0	0	0	0	0	0	...	0
193	0	...	0	0	1	0	0	0	0	...	0
196	0	...	0	0	0	0	1	0	0	...	0
...	0
208	0	...	0	0	0	0	0	0	0	...	0

Langkah selanjutnya membuat matriks *transpose* dari matriks GLCM, dan menjumlahkan matriks *transpose* dengan matriks GLCM. Tabel *transpose* dan Tabel hasil jumlah *tranpose* matriks GLCM dengan matriks GLCM dilihat pada Tabel 4.37 dan Tabel 4.38.

Tabel 4.37 Transpose matriks GLCM dengan arah 135°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0	0	0	0	0	0	...	0
133	0	...	1	0	0	0	0	0	0	...	0
139	0	...	0	0	0	0	0	1	0	...	0
174	0	...	0	0	1	0	0	0	0	...	0

Tabel 4.37 Transpose matriks GLCM dengan arah 135° (Lanjutan)

i/j	0	...	30	133	139	174	189	193	196	...	208
189	0	...	0	0	0	0	0	0	1	...	0
193	0	...	0	0	0	0	0	0	0	...	0
196	0	...	0	0	0	0	0	0	0	...	0

...	0
208	0	...	0	0	0	0	0	0	0	0	...	0

Tabel 4.38 Hasil Jumlah matriks GLCM dan transpose dengan arah 135°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	1	0	0	0	0	0	...	0
133	0	...	1	0	0	0	0	0	0	...	0
139	0	...	0	0	0	1	0	1	0	...	0
174	0	...	0	0	1	0	0	0	0	...	0
189	0	...	0	0	0	0	0	0	1	...	0
193	0	...	0	0	1	0	0	0	0	...	0
196	0	...	0	0	0	0	1	0	0	...	0
...	0
208	0	...	0	0	0	0	0	0	0	...	0

Langkah selanjutnya adalah normalisasi matriks GLCM agar jumlah dari seluruh elemen matriks GLCM adalah 1. Cara untuk normalisasi adalah dengan membagi elemen matriks GLCM dengan nilai total keseluruhan matriks GLCM. Hasil perhitungan normalisasi dapat dilihat pada Tabel 4.39.

Misalkan indeks ke i = 30 dan j = 133, jumlah keseluruhan matriks = 8

$$M(196, 196) = \frac{1}{8} = 0,125$$

Tabel 4.39 Normalisasi Matriks GLCM dengan arah 135°

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0,125	0	0	0	0	0	...	0
133	0	...	0,125	0	0	0	0	0	0	...	0
139	0	...	0	0	0	0,125	0	0,125	0	...	0

Tabel 4.39 Normalisasi Matriks GLCM dengan arah 135° (Lanjutan)

i/j	0	...	30	133	139	174	189	193	196	...	208
0	0	...	0	0	0	0	0	0	0	...	0
...
30	0	...	0	0,125	0	0	0	0	0	...	0

133	0	...	0,125	0	0	0	0	0	0	0	...	0	
139	0	...	0	0	0	0,125	0	0,125	0	0,125	0	...	0
174	0	...	0	0	0,125	0	0	0	0	0	...	0	
189	0	...	0	0	0	0	0	0	0	0,125	...	0	
193	0	...	0	0	0,125	0	0	0	0	0	...	0	
196	0	...	0	0	0	0	0	0,125	0	0	0	...	0
...	0	
208	0	...	0	0	0	0	0	0	0	0	0	0	

4.2.3.2 Perhitungan Ekstraksi Fitur Haralick

Pengambilan fitur *Haralick* diambil setelah matriks GLCM dengan arah 0° , 45° , 90° , dan 135° dilakukan normalisasi. Fitur-fitur yang pakai antara lain *local homogeneity*, *contrast*, *correlation*, *entropy*, dan *energy*. Perhitungan dilakukan sesuai dengan Persamaan 2.7 sampai Persamaan 2.11 untuk setiap fitur. Berikut merupakan contoh perhitungan ekstraksi fitur *haralick*.

- a) Perhitungan ekstraksi fitur *haralick* dengan sudut 0°

$$\begin{aligned} \text{Local Homogeneity} &= \frac{0}{(1+(0-0)^2)} + \dots + \frac{0,08333}{(1+(133-139)^2)} \\ &\quad + \frac{0,08333}{(1+(139-133)^2)} + \frac{0,08333}{(1+(139-196)^2)} \\ &\quad + \dots + \frac{0}{(1+(208-208)^2)} \\ &= 0 + \dots + 0,00225 + 0,00225 + \dots + 0 \\ &= 0,00864 \end{aligned}$$

$$\begin{aligned} \text{Contrast} &= (0 - 0)^2 * 0 + \dots + (133 - 139)^2 * 0,08333 \\ &\quad + (139 - 133)^2 * 0,08333 + \dots + (208 - 208)^2 * 0 \\ &= 0 + \dots + 2,99988 + 2,99988 + \dots + 0 \\ &= 10.301,58792 \end{aligned}$$

Untuk *Correlation*, perlu dilakukan perhitungan *mean* dan *varian* untuk baris dan kolom terlebih dahulu.

$$\begin{aligned} \mu_i &= 0 * 0 + \dots + 0,08333 * 30 + 0,08333 * 30 + 0,0833 * 133 + \dots \\ &\quad + 0 * 208 \end{aligned}$$

$$\begin{aligned} &= 0 + \dots + 2,4999 + 2,4999 + 11,08289 + \dots + 0 \\ &= 151,3273 \end{aligned}$$

$$\begin{aligned} \mu_j &= 0 * 0 + \dots + 0,08333 * 193 + 0,08333 * 208 + 0,0833 * 139 + \dots \\ &\quad + 0 * 208 \end{aligned}$$

$$= 0 + 2,4999 + 2,4999 + 11,08289 + \dots + 10 \\ = 151,3273$$

$$\sigma_i = \sqrt{(0 - 151,3273)^2 * 0 + \dots + (139 - 151,3273)^2 * 0,08333 + \\ (139 - 151,3273)^2 * 0,08333 + \dots + (208 - 151,3273)^2 * 0} \\ = \sqrt{0 + \dots + 12,66302 + 12,66302 + \dots + 0} \\ = 59,4215$$

$$\sigma_j = \sqrt{(0 - 151,3273)^2 * 0 + \dots + (139 - 151,3273)^2 * 0,08333 + \\ (133 - 151,3273)^2 * 0,08333 + \dots + (208 - 151,3273)^2 * 0} \\ = \sqrt{0 + \dots + 12,66302 + 27,98971 + \dots + 0} \\ = 59,4215$$

$$\text{Correlation} = \frac{(0-151,3273)*(0-151,3273)*0}{59,4215*59,4215} + \dots \\ + \frac{(133-151,3273)*(139-151,3273)*0,08333}{59,4215*59,4215} \\ + \frac{(139-151,3273)*(133-151,3273)*0,08333}{59,4215*59,4215} + \dots \\ + \frac{(208-151,3273)*(208-151,3273)*0}{59,4215*59,4215} \\ = 0 + \dots + 0,00533 + 0,00533 + \dots + 0 \\ = -0,45877$$

$$\text{Entropy} = 0,08333 * \log(0,08333) + 0,08333 * \log(0,08333) \\ + 0,08333 * \log(0,08333) + \dots + 0,08333 * \log(0,08333) \\ = (-0,08992) + (-0,08992) + (-0,08992) + \dots \\ + (-0,08992) \\ = (-1,07916)$$

$$\text{Energy} = (0)^2 + (0,08333)^2 + (0,08333)^2 + (0,08333)^2 + \dots + (0)^2 \\ = 0 + 0,00694 + 0,00694 + 0,00694 + \dots + 0 \\ = 0,08333$$

b) Perhitungan ekstraksi fitur *haralick* dengan sudut 45°

$$\text{Local Homogeneity} = \frac{0}{(1+(0-0)^2)} + \dots + \frac{0,125}{(1+(30-196)^2)} \\ + \frac{0,125}{(1+(133-189)^2)} + \frac{0,125}{(1+(139-196)^2)} \\ + \dots + \frac{0}{(1+(208-208)^2)} \\ = 0,00022$$

$$\text{Contrast} = (0 - 0)^2 * 0 + \dots + (30 - 196)^2 * 0,125$$



$$\begin{aligned}
 & + (133 - 189)^2 * 0,125 + \dots + (208 - 208)^2 * 0 \\
 & = 0 + \dots + 3.444,5 + 392 + \dots + 0 \\
 & = 9.675,5
 \end{aligned}$$

Untuk *Correlation*, perlu dilakukan perhitungan *mean* dan *varian* untuk baris dan kolom terlebih dahulu.

$$\begin{aligned}
 \mu_i &= 0 * 0 + \dots + 0,125 * 30 + 0,125 * 133 + 0,125 * 139 + \dots \\
 &\quad + 0 * 208
 \end{aligned}$$

$$\begin{aligned}
 &= 0 + \dots + 3,75 + 16,625 + 17,375 + \dots + 0 \\
 &= 153,75
 \end{aligned}$$

$$\begin{aligned}
 \mu_j &= 0 * 0 + \dots + 0,125 * 196 + 0,125 * 189 + 0,125 * 196 + \dots \\
 &\quad + 0 * 208
 \end{aligned}$$

$$\begin{aligned}
 &= 0 + 24,5 + 23,625 + 24,5 + \dots + 0 \\
 &= 153,75
 \end{aligned}$$

$$\begin{aligned}
 \sigma_i &= \sqrt{(0 - 153,75)^2 * 0 + \dots + (30 - 153,75)^2 * 0,125 + \\
 &\quad (133 - 153,75)^2 * 0,125 + \dots + (208 - 153,75)^2 * 0} \\
 &= \sqrt{0 + \dots + 1.914,25781 + 53,82031 + \dots + 0} \\
 &= 54,69861
 \end{aligned}$$

$$\begin{aligned}
 \sigma_j &= \sqrt{(0 - 153,75)^2 * 0 + \dots + (196 - 153,75)^2 * 0,125 + \\
 &\quad (189 - 153,75)^2 * 0,125 + \dots + (208 - 153,75)^2 * 0} \\
 &= \sqrt{0 + \dots + 223,13281 + 155,32031 + \dots + 0} \\
 &= 54,69861
 \end{aligned}$$

$$\begin{aligned}
 \text{Correlation} &= \frac{(0-153,75)*(0-153,75)*0}{54,69861*54,69861} + \dots \\
 &\quad + \frac{(30-153,75)*(196-153,75)*0,125}{54,69861*54,69861} \\
 &\quad + \frac{(133-153,75)*(189-153,75)*0,125}{54,69861*54,69861} + \dots \\
 &\quad + \frac{(208-153,75)*(208-153,75)*0}{54,69861*54,69861} \\
 &= 0 + \dots + (-0,21844) + (-0,03056) + \dots + 0 \\
 &= -0,61693
 \end{aligned}$$

$$\begin{aligned}
 \text{Entropy} &= 0,125 * \log(0,125) + 0,125 * \log(0,125) \\
 &\quad + 0,125 * \log(0,125) + \dots + 0,125 * \log(0,125) \\
 &= (-0,11289) + (-0,11289) + (-0,11289) + \dots \\
 &\quad + (-0,11289)
 \end{aligned}$$

$$= -0,90309$$

$$\begin{aligned} Energy &= (0)^2 + (0,125)^2 + (0,125)^2 + (0,125)^2 + \dots + (0)^2 \\ &= 0 + 0,01562 + 0,01562 + 0,01562 + \dots + 0 \\ &= 0,125 \end{aligned}$$

c) Perhitungan ekstraksi fitur *haralick* dengan sudut 90°

$$\begin{aligned} Local Homogeneity &= \frac{0}{(1+(0-0)^2)} + \dots + \frac{0,08333}{(1+(30-139)^2)} \\ &\quad + \frac{0,08333}{(1+(133-174)^2)} + \frac{0,08333}{(1+(133-208)^2)} \\ &\quad + \dots + \frac{0}{(1+(208-208)^2)} \\ &= 0,18388 \end{aligned}$$

$$\begin{aligned} Contrast &= (0 - 0)^2 * 0 + \dots + (30 - 139)^2 * 0,08333 \\ &\quad + (133 - 174)^2 * 0,08333 + \dots + (208 - 208)^2 * 0 \\ &= 0 + \dots + 990,04373 + 140,07773 + \dots + 0 \\ &= 5787,352 \end{aligned}$$

Untuk *Correlation*, perlu dilakukan perhitungan *mean* dan *varian* untuk baris dan kolom terlebih dahulu.

$$\begin{aligned} \mu_i &= 0 * 0 + \dots + 0,08333 * 30 + 0,08333 * 133 + 0,08333 * 133 + \dots \\ &\quad + 0 * 208 \\ &= 0 + \dots + 2,4999 + 11,08289 + 11,08289 + \dots + 0 \\ &= 160,5602 \end{aligned}$$

$$\begin{aligned} \mu_j &= 0 * 0 + \dots + 0,08333 * 139 + 0,08333 * 174 + 0,08333 * 208 \\ &\quad + \dots + 0 * 208 \\ &= 0 + 11,58287 + 14,49942 + 17,33264 + \dots + 0 \\ &= 160,5602 \end{aligned}$$

$$\begin{aligned} \sigma_i &= \sqrt{(0 - 160,5602)^2 * 0 + \dots + (30 - 160,5602)^2 * 0,08333 + \\ &\quad (133 - 160,5602)^2 * 0,08333 + \dots + (208 - 160,5602)^2 * 0} \\ &= \sqrt{0 + \dots + 1.420,44033 + 63,29452 + \dots + 0} \\ &= 47,92274 \end{aligned}$$

$$\begin{aligned} \sigma_j &= \sqrt{(0 - 160,5602)^2 * 0 + \dots + (139 - 160,5602)^2 * 0,08333 + \\ &\quad (174 - 160,5602)^2 * 0,08333 + \dots + (208 - 160,5602)^2 * 0} \\ &= \sqrt{0 + \dots + 38,7353 + 15,05175 + \dots + 0} \\ &= 47,92274 \end{aligned}$$

$$\begin{aligned}
 Correlation &= \frac{(0-160,5602)*(0-160,5602)*0}{47,92274*47,92274} + \dots \\
 &\quad + \frac{(30-160,5602)*(139-160,5602)*0,08333}{47,92274*47,92274} \\
 &\quad + \frac{(133-160,5602)*(174-160,5602)*0,08333}{47,92274*47,92274} + \dots \\
 &\quad + \frac{(208-160,5602)*(208-160,5602)*0}{47,92274*47,92274} \\
 &= 0 + \dots + 0,102137 + (-0,01344) + \dots + 0 \\
 &= 0,212777
 \end{aligned}$$

$$\begin{aligned}
 Entropy &= 0,08333 * \log(0,08333) + 0,08333 * \log(0,08333) \\
 &\quad + 0,08333 * \log(0,08333) + \dots + 0,08333 * \log(0,08333) \\
 &= (-0,08992) + (-0,08992) + (-0,08992) + \dots \\
 &\quad + (-0,08992) \\
 &= -1,0291
 \end{aligned}$$

$$\begin{aligned}
 Energy &= (0)^2 + (0,08333)^2 + (0,08333)^2 + (0,08333)^2 + \dots + (0)^2 \\
 &= 0 + 0,00694 + 0,00694 + 0,00694 + \dots + 0 \\
 &= 0,097328
 \end{aligned}$$

d) Perhitungan ekstraksi fitur *haralick* dengan sudut 135°

$$\begin{aligned}
 Local\ Homogeneity &= \frac{0}{(1+(0-0)^2)} + \dots + \frac{0,125}{(1+(30-133)^2)} \\
 &\quad + \frac{0,125}{(1+(133-30)^2)} + \frac{0,125}{(1+(139-174)^2)} \\
 &\quad + \dots + \frac{0}{(1+(208-208)^2)} \\
 &= 0,005313
 \end{aligned}$$

$$\begin{aligned}
 Contrast &= (0-0)^2 * 0 + \dots + (30-133)^2 * 0,125 \\
 &\quad + (133-30)^2 * 0,125 + \dots + (208-208)^2 * 0 \\
 &= 0 + \dots + 1.326,125 + 1.326,125 + \dots + 0 \\
 &= 3.699,75
 \end{aligned}$$

Untuk *Correlation*, perlu dilakukan perhitungan *mean* dan *varian* untuk baris dan kolom terlebih dahulu.

$$\begin{aligned}
 \mu_i &= 0 * 0 + \dots + 0,125 * 30 + 0,125 * 133 + 0,125 * 139 + \dots \\
 &\quad + 0 * 208 \\
 &= 0 + \dots + 3,75 + 16,625 + 17,375 + \dots + 0 \\
 &= 149,125
 \end{aligned}$$

$$\mu_j = 0 * 0 + \dots + 0,125 * 133 + 0,125 * 30 + 0,125 * 174 + \dots$$

$$\begin{aligned}
 & +0 * 208 \\
 & = 0 + 16,625 + 3,75 + 21,75 + \dots + 0 \\
 & = 149,125 \\
 \sigma_i & = \sqrt{(0 - 149,125)^2 * 0 + \dots + (30 - 149,125)^2 * 0,125 +} \\
 & \quad (133 - 149,125)^2 * 0,125 + \dots + (208 - 149,125)^2 * 0 \\
 & = \sqrt{0 + \dots + 1.773,846 + 32,50195 + \dots + 0} \\
 & = 51,21874 \\
 \sigma_j & = \sqrt{(0 - 149,125)^2 * 0 + \dots + (133 - 149,125)^2 * 0,125 +} \\
 & \quad (30 - 149,125)^2 * 0,125 + \dots + (208 - 149,125)^2 * 0 \\
 & = \sqrt{0 + \dots + 32,50195 + 1.773,846 + \dots + 0} \\
 & = 51,21874 \\
 \text{Correlation} & = \frac{(0-149,125)*(0-149,125)*0}{51,21874*51,21874} + \dots \\
 & + \frac{(30-149,125)*(133-149,125)*0,125}{51,21874*51,21874} \\
 & + \frac{(133-149,125)*(30-149,125)*0,125}{51,21874*51,21874} + \dots \\
 & + \frac{(208-149,125)*(208-149,125)*0}{51,21874*51,21874} \\
 & = 0 + \dots + 0,09153 + 0,09153 + \dots + 0 \\
 & = 0,29485
 \end{aligned}$$

$$\begin{aligned}
 \text{Entropy} & = 0,125 * \log(0,125) + 0,125 * \log(0,125) \\
 & \quad + 0,125 * \log(0,125) + \dots + 0,125 * \log(0,125) \\
 & = (-0,11289) + (-0,11289) + (-0,11289) + \dots \\
 & \quad + (-0,11289) \\
 & = -0,90309
 \end{aligned}$$

$$\begin{aligned}
 \text{Energy} & = (0)^2 + (0,125)^2 + (0,125)^2 + (0,125)^2 + \dots + (0)^2 \\
 & = 0 + 0,01562 + 0,01562 + 0,01562 + \dots + 0 \\
 & = 0,125
 \end{aligned}$$

4.2.4 Perhitungan *Naive Bayes*

Terdapat dua tahap dalam perhitungan *naive bayes* yaitu tahap pelatihan dan tahap pengujian. Pada tahap pelatihan digunakan hasil ekstraksi fitur dari data latih untuk setiap fitur yang digunakan. Hasil ekstraksi fitur untuk data latih dapat dilihat pada Tabel 4.40.

Kode	Kelas	μH	μS	μV	σH	σS	σV	SkH	..	Corr135
001	Donat	0,114	0,715	0,661	0,118	0,145	0,256	0,192	..	0,986
...
009	Rendang	0,155	0,746	0,391	0,179	0,203	0,226	0,246	..	0,947
...
010	mentimun	0,255	0,569	0,450	0,117	0,194	0,231	0,166	..	0,976
...
016	strawberry	0,477	0,679	0,338	0,407	0,239	0,215	0,267	..	0,944
...
017	Pisang Ijo	0,148	0,492	0,447	0,129	0,213	0,168	0,199	..	0,984
...
018	Pisang Kuning	0,138	0,727	0,531	0,129	0,128	0,208	0,204	..	0,987
...
020	Jeruk Ijo	0,128	0,806	0,539	0,169	0,192	0,266	0,239	..	0,956
...

Tabel 4.40 Fitur Data Latih (Lanjutan)

Kode	Kelas	μH	μS	μV	σH	σS	σV	SkH	..	Corr135
024	Beng Beng	0,119	0,572	0,495	0,193	0,246	0,299	0,266	..	0,947

...
026	Tim Tam	0,09 2	0,56 9	0,19 5	0,16 9	0,23 4	0,11 6	0,25 4	0,939	
...
028	Gerry Saluut	0,09 3	0,54 9	0,17 0	0,17 1	0,22 4	0,13 2	0,25 6	0,930	
...
029	Biskuat Coklat	0,13 9	0,66 0	0,47 5	0,17 5	0,18 7	0,26 7	0,24 1	0,978	
...
032	Genji Pie	0,16 1	0,67 5	0,45 7	0,19 4	0,22 4	0,27 8	0,25 3	0,965	
...

Dari hasil nilai ekstraksi fitur, ditentukan nilai rata-rata dan standar deviasi seluruh fitur setiap kelas sesuai dengan Persamaan 2.18. Contoh perhitungan nilai rata-rata dan standar deviasi untuk fitur *mean H* pada kelas Donat sebagai berikut.

$$\mu_{meanH} = \frac{(0,114 + 0,114 + 0,123 + \dots + 0,125)}{20} = 0,123$$

$$\sigma_{meanH} = \sqrt{\frac{(0,114 - 0,123)^2 + (0,114 - 0,123)^2 + (0,123 - 0,123)^2 + \dots + (0,125 - 0,123)^2}{20}} \\ = 0,009$$

Tabel hasil perhitungan nilai rata-rata dan standar deviasi seluruh fitur setiap kelas dapat dilihat pada Tabel 4.41 dan Tabel 4.42.

Tabel 4.41 Mean Fitur Data Latih

Kelas	μ_H	μ_S	μ_V	σ_H	σ_S	σ_V	SkH	SkS	Corr13 5
Donat	0,12 3	0,80 1	0,61 7	0,12 9	0,14 2	0,23 9	0,20 4	0,07 4	0,982
Rendang	0,13 1	0,73 9	0,39 5	0,15 6	0,16 5	0,18 2	0,22 8	0,09 5	0,938

Mentimun	0,23 6	0,64 5	0,43 7	0,11 9	0,18 6	0,22 4	0,17 0	0,13 6	0,977
Strawberry	0,56 0	0,75 8	0,32 6	0,42 6	0,20 2	0,19 3	0,17 3	0,08 8	0,913
Pisang Ijo	0,18 8	0,64 4	0,42 5	0,13 2	0,17 3	0,19 4	0,19 3	0,10 9	0,986
Pisang Kuning	0,15 2	0,74 7	0,59 8	0,13 5	0,14 4	0,26 6	0,20 5	0,06 6	0,989
Jeruk Ijo	0,13 7	0,78 6	0,53 3	0,14 3	0,18 8	0,24 2	0,21 4	0,09 3	0,968
Beng Beng	0,11 3	0,61 1	0,27 8	0,20 1	0,23 0	0,18 4	0,28 1	0,17 7	0,932
Tim Tam	0,18 4	0,61 9	0,22 3	0,24 6	0,24 4	0,16 6	0,28 2	0,14 3	0,930
Gerry Saluut	0,12 1	0,65 4	0,18 5	0,19 5	0,24 3	0,12 9	0,26 1	0,14 3	0,892
Biskuat Coklat	0,13 7	0,67 5	0,55 7	0,15 9	0,17 8	0,27 1	0,22 8	0,12 0	0,981
Genji Pie	0,14 1	0,71 3	0,54 9	0,15 9	0,19 7	0,26 6	0,23 0	0,07 8	0,969

Tabel 4.42 Standar Deviasi Fitur Data Latih

Kelas	μ_H	μ_S	μ_V	σ_H	σ_S	σ_V	SkH	SkS	Corr13 5
Donat	0,00 9	0,88 6	0,06 1	0,01 3	0,02 0	0,01 9	0,01 3	0,02 4	0,008
Rendang	0,01 3	0,08 4	0,10 7	0,02 4	0,02 6	0,03 1	0,02 1	0,04 0	0,015
Mentimun	0,02 9	0,08 1	0,08 1	0,02 6	0,03 4	0,02 3	0,02 1	0,07 0	0,010

Tabel 4.42 Standar Deviasi Fitur Data Latih (Lanjutan)

Kelas	μ_H	μ_S	μ_V	σ_H	σ_S	σ_V	SkH	SkS	Corr13 5
Strawberry	0,06 6	0,07 8	0,10 0	0,02 0	0,02 4	0,04 8	0,06 6	0,01 6	0,034
Pisang Ijo	0,01 7	0,06 0	0,03 8	0,01 1	0,02 1	0,02 5	0,00 9	0,05 1	0,002

Pisang Kuning	0,01 6	0,06 7	0,07 3	0,01 1	0,01 7	0,02 8	0,01 0	0,02 1	.. .	0,004
Jeruk Ijo	0,02 0	0,04 8	0,10 5	0,01 9	0,01 9	0,04 3	0,01 9	0,01 7	.. .	0,010
Beng Beng	0,01 4	0,02 9	0,07 2	0,02 6	0,01 7	0,03 6	0,02 6	0,04 4	.. .	0,010
Tim Tam	0,15 2	0,04 9	0,06 1	0,09 3	0,02 9	0,04 0	0,06 4	0,07 9	.. .	0,031
Gerry Saluut	0,05 4	0,08 4	0,07 5	0,04 7	0,03 9	0,02 3	0,02 3	0,05 7	.. .	0,054
Biskuat Coklat	0,01 1	0,03 9	0,05 4	0,01 6	0,01 7	0,02 3	0,01 2	0,05 4	.. .	0,005
Genji Pie	0,01 4	0,05 0	0,06 1	0,02 1	0,01 7	0,01 9	0,01 5	0,03 7	.. .	0,008

Kemudian langkah pertama pada tahap Pengujian adalah dengan mencari *likelihood* seluruh fitur dari setiap data uji dengan menggunakan Persamaan 2.20. Hasil ekstraksi fitur untuk data uji dapat dilihat pada Tabel 4.43

Tabel 4.43 Fitur Data Uji

No	μH	μS	μV	σH	σS	σV	SkH	...	Corr135
1	0,122	0,676	0,608	0,132	0,189	0,264	0,206	...	0,972
...
4	0,144	0,735	0,409	0,167	0,204	0,220	0,239	...	0,944
...
7	0,226	0,566	0,309	0,113	0,201	0,184	0,167	...	0,976
...
10	0,489	0,680	0,341	0,406	0,239	0,218	0,251	...	0,942
...

Tabel 4.43 Fitur Data Uji (Lanjutan)

No	μH	μS	μV	σH	σS	σV	SkH	...	Corr135
13	0,187	0,601	0,465	0,144	0,183	0,237	0,203	...	0,988
...
16	0,136	0,706	0,590	0,121	0,148	0,266	0,193	...	0,989
...

19	0,131	0,827	0,631	0,118	0,177	0,266	0,192	...	0,979
...
22	0,113	0,597	0,292	0,188	0,228	0,189	0,266	...	0,935
...
25	0,114	0,550	0,330	0,213	0,256	0,210	0,294	...	0,942
...
28	0,103	0,573	0,174	0,167	0,222	0,115	0,244	...	0,931
...
31	0,132	0,643	0,587	0,156	0,165	0,258	0,228	...	0,985
...
34	0,147	0,659	0,475	0,179	0,202	0,266	0,244	...	0,972
...

Perhitungan contoh *likelihood* dapat dilihat sebagai berikut.

$$P(\mu H | \text{Donat}) = \frac{1}{\sqrt{2 * 3,14 * 0,009}} * (2,71)^{\frac{(0,122 - 0,123)}{2 * (0,009)^2}} = 0,427$$

$$P(\mu S | \text{Donat}) = \frac{1}{\sqrt{2 * 3,14 * 0,886}} * (2,71)^{\frac{(0,676 - 0,123)}{2 * (0,886)^2}} = 0,124$$

Contoh hasil *likelihood* untuk data uji no 1 dapat dilihat pada Tabel

Tabel 4.44 Likelihood Data Uji No 1

Kelas	μH	μS	μV	σH	σS	σV	SkH	SkS	..	Corr13 5
Donat	0,42 7	0,12 4	0,21 6	0,36 0	0,09 7	0,18 1	0,36 7	0,16 4	0,145
Rendang	0,31 1	0,16 9	0,09 7	0,21 2	0,21 7	0,08 9	0,21 7	0,14 9	0,105

Tabel 4.44 Likelihood Data Uji No 1 (Lanjutan)

Kelas	μH	μS	μV	σH	σS	σV	SkH	SkS	..	Corr13 5
Donat	0,42 7	0,12 4	0,21 6	0,36 0	0,09 7	0,18 1	0,36 7	0,16 4	0,145
Rendang	0,31 1	0,16 9	0,09 7	0,21 2	0,21 7	0,08 9	0,21 7	0,14 9	0,105

Mentimun	0,08 3	0,19 1	0,09 6	0,26 4	0,26 2	0,13 4	0,13 8	0,13 2	..	0,373
Strawberry	0,08 3	0,15 1	0,08 5	0,08 3	0,26 7	0,13 4	0,19 6	0,08 7	..	0,125
Pisang Ijo	0,08 3	0,20 0	0,08 3	0,39 8	0,25 5	0,08 7	0,22 1	0,14 3	..	0,083
Pisang Kuning	0,13 4	0,15 6	0,20 4	0,38 0	0,09 3	0,28 1	0,41 2	0,19 1	..	0,083
Jeruk Ijo	0,26 3	0,09 4	0,16 3	0,28 7	0,31 9	0,22 2	0,29 9	0,08 5	..	0,380
Beng Beng	0,30 4	0,10 0	0,08 3	0,09 0	0,10 1	0,09 8	0,08 7	0,08 4	..	0,083
Tim Tam	0,16 1	0,16 0	0,08 3	0,13 5	0,11 8	0,09 1	0,14 9	0,13 5	..	0,157
Gerry Saluut	0,22 5	0,19 4	0,08 3	0,14 6	0,15 0	0,08 3	0,09 8	0,11 2	..	0,131
Biskuat Coklat	0,22 0	0,25 0	0,17 5	0,14 7	0,28 7	0,29 2	0,14 0	0,13 3	..	0,232
Genji Pie	0,20 5	0,19 5	0,16 8	0,18 3	0,31 3	0,32 1	0,16 2	0,18 6	..	0,412

Langkah Terakhir adalah mencari *posterior probability* dari setiap kelas dengan melakukan perkalian *likelihood* seluruh fitur dalam satu kelas. Kemudian lakukan perkalian dengan nilai *prior* sesuai dengan Persamaan 2.17. berikut merupakan contoh perhitungan *posterior probability*.

$$P(\text{Donat}|\text{datauji1}) = (0,427 * 0,124 * \dots * 0,145) * \left(\frac{20}{240}\right)$$

$$= 6,001623669103715e - 25$$

$$P(\text{Rendang}|\text{datauji1}) = (0,311 * 0,169 * \dots * 0,105) * \left(\frac{20}{240}\right)$$

$$= 8,003757790852583e - 28$$

Tabel hasil perhitungan *posterior probability* dapat dilihat pada Tabel 4.45.

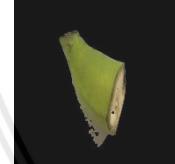
Tabel 4. 45 Posterior Probability Data Uji No 1

Kelas	Posterior Probability
Donat	6,001623669103715e-25
Rendang	8,003757790852583e-28
Mentimun	5,872890585000557e-27
Strawberry	1,7530011266730376e-30

Pisang Ijo	5,5455450246058374e-30
Pisang Kuning	6,113019921515777e-29
Jeruk Ijo	2,4570009711853253e-24
Beng Beng	7,561509392827921e-32
Tim Tam	5,1999466661765975e-30
Gerry Saluut	2,1830548743184202e-30
Biskuat Coklat	4,4359256108821945e-28
Genji Pie	6,772439749974858e-26

Dari hasil *Posterior Probability* dapat diambil kesimpulan untuk data uji nomor 1 merupakan kelas Jeruk Ijo karena memiliki nilai *posterior probability* paling besar dibanding dengan kelas lain. Hasil dari klasifikasi dapat dilihat pada Tabel 4.46.

Tabel 4.46 Hasil Klasifikasi Citra

Hasil Klasifikasi Citra				
Gambar				
Hasil	Jeruk Ijo (020)	Rendang (009)	Mentimun (010)	Rendang (009)
Gambar				
Hasil	Pisang Ijo (017)	Pisang Kuning (018)	Donat (001)	Beng-Beng (024)

Tabel 4.46 Hasil Klasifikasi Citra (Lanjutan)

Hasil Klasifikasi Citra				
Gambar				

Hasil	Beng-Beng (024)	Gerry Saluut (028)	Biskuat Coklat (029)	Biskuat Coklat (029)
-------	--------------------	-----------------------	-------------------------	-------------------------

4.3 Perhitungan Evaluasi Kinerja Sistem

Perhitungan evaluasi kinerja sistem dilakukan dengan melihat hasil dari sistem dalam melakukan klasifikasi citra. Metode evaluasi yang digunakan adalah dalam penelitian ini adalah dengan menggunakan *Confusion matrix*. Hasil dari evaluasi kinerja sistem dalam melakukan klasifikasi dapat dilihat pada Tabel 4.47.

Tabel 4.47 Evaluasi Confusion Matrix

		Hasil Prediksi											
		001	009	010	016	017	018	020	024	026	028	029	032
Hasil Nilai seben arnya	001	2	0	0	0	0	0	1	0	0	0	0	0
	009	0	2	0	0	0	0	0	0	1	0	0	0
	010	0	0	2	0	0	0	0	0	0	0	1	0
	016	0	1	0	0	0	0	1	0	0	1	0	0
	017	0	0	0	0	3	0	0	0	0	0	0	0
	018	0	0	0	0	0	3	0	0	0	0	0	0
	020	1	0	1	0	0	0	1	0	0	0	0	0
	024	0	0	0	0	0	0	0	2	0	1	0	0
	026	0	0	0	0	0	0	0	2	1	0	0	0
	028	0	0	0	1	0	0	0	0	1	1	0	0
	029	1	0	0	0	0	0	0	0	0	0	2	0
	032	0	0	0	0	0	0	1	0	0	0	1	1

Perhitungan evaluasi sistem dapat dilihat sebagai berikut.

$$Accuracy = \frac{2 + 2 + 2 + 0 + 3 + 3 + 1 + 2 + 1 + 1 + 2 + 1}{2 + 1 + 2 + 1 + 2 + 1 + \dots + 1} = \frac{20}{36} = 55,56$$

$$Precision\ 001 = \frac{2}{2+1} = 0,67$$

$$Recall\ 001 = \frac{2}{2+1+1} = 0,5$$

$$Precision\ 009 = \frac{2}{2+1} = 0,67$$

$$Recall\ 009 = \frac{2}{2+1} = 0,67$$

$$Precision\ 010 = \frac{2}{2+1} = 0,67$$

$$Recall\ 010 = \frac{2}{2+1} = 0,67$$

$$Precision\ 016 = \frac{0}{1+1+1} = 0$$

$$Recall\ 016 = \frac{0}{0+1} = 0$$

$$Precision\ 017 = \frac{3}{3+0} = 1$$

$$Recall\ 017 = \frac{3}{3+0} = 1$$

$$Precision\ 018 = \frac{3}{3+0} = 1$$

$$Recall\ 018 = \frac{3}{3+0} = 1$$

$$Precision\ 020 = \frac{1}{1+1+1} = 0,33$$

$$Recall\ 020 = \frac{1}{1+1+1+1} = 0,25$$

$$Precision\ 024 = \frac{2}{2+1} = 0,67$$

$$Recall\ 024 = \frac{2}{2+2} = 0,5$$

$$Precision\ 026 = \frac{1}{2+1} = 0,33$$

$$Recall\ 026 = \frac{1}{1+1+1} = 0,33$$

$$Precision\ 028 = \frac{1}{1+1+1} = 0,33$$

$$Recall\ 028 = \frac{1}{1+1+1} = 0,33$$

$$Precision\ 029 = \frac{2}{1+2} = 0,67$$

$$Recall\ 029 = \frac{2}{1+2+1} = 0,5$$

$$Precision\ 032 = \frac{1}{1+1+1} = 0,33$$

$$Recall\ 032 = \frac{1}{1} = 1$$

$$F-score = \frac{2 * 0,67 * 0,67 * 0,67 * 0 * \dots * 1}{0,67 + 0,67 + 0,67 + \dots + 1} = 0$$



BAB 5 IMPLEMENTASI

5.1 Batasan Implementasi

Batasan yang digunakan pada penilitian ini diperlukan dengan tujuan memperjelas dalam melakukan perancangan sistem yang telah dibuat. Batasan tersebut adalah klasifikasi citra makanan dengan menggunakan HSV *color moment* dan fitur *haralick* dengan menggunakan klasifikasi *naive bayes*.

5.2 Implementasi Algoritme

5.2.1 Algoritme *Preprocessing* Citra

Tahapan-tahapan dalam algoritme *preprocessing* citra meliputi *resize* atau mengubah ukuran piksel dari citra, mengubah nilai citra dengan ruang warna *red*, *green*, dan *blue* ke dalam nilai *saturation*, konvolusi citra menggunakan *gaussian blur* dengan kernel 5x5 untuk mendapatkan citra blur, *thresholding*, *erotion*, dan *dilation*. Tujuan implementasi algoritme *preprocessing* citra adalah untuk mendapatkan citra yang lebih baik dan sudah di segmentasi. *Source code* untuk algoritme *preprocessing* dapat dilihat pada Tabel 5.1

Tabel 5.1 Source Code Algoritme Preprocessing Citra

No	
1	import cv2
2	import numpy as np
3	import os.path
4	from imutils import paths
5	tempNama = ''
6	for imagePath in
7	paths.list_images("E:/Dataset_Skripsi/Data_fix_pakai_2/026/"):
8	if (tempNama !=
9	str(os.path.split(os.path.dirname(imagePath))[-1])):
10	tempNama =
11	str(os.path.split(os.path.dirname(imagePath))[-1])
12	jumlah = 0
13	jumlah += 1
14	print (imagePath)
15	image = cv2.imread(imagePath)
16	image1 = cv2.resize(image, (500, 500))
17	rows,cols,channels = image1.shape
18	roi = image1[0:rows, 0:cols]
19	nilaiS = np.zeros((500,500))
20	for i in range(0, rows):
21	for j in range(0, cols):
22	b = float(image1[i, j, 0])
23	g = float(image1[i, j, 1])
24	r = float(image1[i, j, 2])
25	# print r
26	maxrgb = max(b, g, r)
27	minrgb = min(b, g, r)
28	if(maxrgb == 0):
	s=0
	else:
	s = (1.0 - (minrgb / maxrgb))

IMPLEMENTASI ALGORITMA EKSTRAKSI FITUR HSV COLOR MOMENT PADA CITRA MAKANAN

Algoritma ekstraksi fitur HSV Color Moment pada citra makanan yang dilakukan dengan menggunakan *OpenCV*. Berikut ini merupakan contoh *source code* dalam bahasa Python yang dapat dijalankan pada sistem operasi Windows.

```

29         nilaiS[i][j] = s*255
30     #Konvolusi Citra Gaussian Blurring
31     blurred = cv2.GaussianBlur(nilaiS, (5, 5), 0)
32     #Global Thresholding
33     ret, th = cv2.threshold(blurred, 83 , 255,
34     cv2.THRESH_BINARY)
35     #Erosi dan Dilasi
36     structuringelement = np.ones((5, 5), np.uint8)
37     img_erosion = cv2.erode(th, structuringelement,
38     iterations=1)
39     img_dilation = cv2.dilate(img_erosion, structuringelement,
40     iterations=1)
41     cv2.imwrite('cobacoba.png', img_dilation)
42     mask =cv2.imread('cobacoba.png', 0)
43     #Masking
44     img1_bg = cv2.bitwise_and(roi, roi, mask=mask)
45     rename = "E:/Dataset_Skripsi/Data_fix_pakai_segmentasi_4/"
46     + str(os.path.split(os.path.dirname(imagePath)) [-1]) + "_" +
47     str(jumlah) + ".jpg"
48     cv2.imwrite(rename, img1_bg)

```

Penjelasan *source code*:

- Baris 2-4 merupakan *library* yang digunakan untuk melakukan *preprocessing* citra.
- Baris 5-11 digunakan untuk membuka *file* direktori dari citra yang akan dilakukan *preprocessing* dan variabel tempNama digunakan untuk mengecek perpindahan file dari kelas citra makanan, dan dilakukan set nilai 0 terhadap variabel jumlah jika setiap *file* kelas baru dan digunakan untuk penomoran kelas citra hasil *preprocessing*.
- Baris 12 digunakan untuk membaca *file* gambar dengan menggunakan *library* cv2.
- Baris 13 digunakan untuk merubah ukuran piksel citra dengan ukuran 500 piksel x 500 piksel.
- Baris 14 digunakan untuk mengambil nilai tinggi dan lebar dari citra.
- Baris 15 variabel roi digunakan untuk menyimpan citra asli.
- Baris 16-29 Variabel nilaiS digunakan untuk menyimpan hasil perhitungan nilai *saturation* setiap piksel. Dan dilakukan proses pengambilan nilai *red*, *green*, dan *blue* dari setiap piksel. Kemudian dicari nilai maksimal dan minimal dari nilai RGB untuk perhitungan *saturation* dan disimpan dalam variabel nilaiS.
- Baris 31 proses konvolusi citra menggunakan *gaussian blur* dengan kernel 5x5.
- Baris 33 proses *thresholding* dengan menggunakan *threshold* sebesar 83.
- Baris 34-39 proses pembuatan matriks *structuring element* dengan nilai 1 sebesar matriks 5x5 yang digunakan untuk proses erosi dan dilasi. Dan menampilkan citra hasil dilasi untuk disimpan sebagai *masking*.
- Baris 41 digunakan untuk proses *masking* dari variabel roi dan mask.
- Baris 42-43 digunakan untuk menyimpan gambar hasil *preprocessing*.

5.2.2 Algoritme Ekstraksi Fitur HSV *Color Moment*

Tahapan-tahapan dalam algoritme ekstraksi fitur HSV *Color Moment* adalah menghitung nilai HSV dari setiap piksel. Kemudian menghitung *mean*, standar deviasi, dan *skewness* dari setiap ruang warna *Hue*, *saturation*, dan *value* untuk

dijadikan fitur. *Source code* untuk algoritme ekstraksi fitur HSV *color moment* dapat dilihat pada tabel 5.2.

Tabel 5.2 Source Code Algoritme Ekstraksi Fitur HSV Color Moment

No	
1	import cv2
2	import numpy as np
3	from math import log10
4	import os.path
5	from imutils import paths
6	import pandas as pd
7	meanH = []
8	meanS = []
9	meanV = []
10	stdH = []
11	stdS = []
12	stdV = []
13	skwH = []
14	skwS = []
15	skwV = []
16	
17	tempNama = ''
18	
19	for imagePath in
	paths.list_images("E:/Dataset_Skripsi/Data_fix_pakai_testing/")
20	:
	if (tempNama !=
	str(os.path.split(os.path.dirname(imagePath))[-1])):
21	tempNama =
	str(os.path.split(os.path.dirname(imagePath))[-1])
22	jumlah = 0
23	print (imagePath)
24	image = cv2.imread(imagePath)
25	image1 = cv2.resize(image, (500, 500))
26	h, w = image1.shape[:2]
27	hue1 = []
28	s1 = []
29	v1 = []
30	hsv = np.zeros(image1.shape, dtype=np.float)
31	for i in range(0,h):
32	for j in range(0,w):
33	b = float(image1[i, j, 0])
34	g = float(image1[i, j, 1])
35	r = float(image1[i, j, 2])
36	jumlah = r+g+b
37	if(jumlah == 0):
38	continue
39	else:
40	# print r
41	maxrgb = max(b, g, r)
42	minrgb = min(b, g, r)
43	if ((r > b and r > g and g >= b) or (r == g and r > b and g > b)):
44	hue = 60 * ((g - b) / (maxrgb - minrgb)) / 360
45	hue1.append(hue)
46	# print hue
47	elif ((r > b and r > g and g < b) or (r == b and r > g and g < b)):

```
and r > g and b > g)):  
48             hue = (60 * ((g - b) / (maxrgb - minrgb)) +  
49             360 )/360  
50             hue1.append(hue)  
51             # print hue  
52         elif (g > b and g > r):  
53             hue = (60 * ((b - r) / (maxrgb - minrgb)) +  
54             120)/360  
55             hue1.append(hue)  
56             # print hue  
57         elif (b > r and b > g):  
58             hue = (60 * ((r - g) / (maxrgb - minrgb)) +  
59             240)/360  
60             hue1.append(hue)  
61             # print hue  
62         else:  
63             hue = 0  
64             hue1.append(hue)  
65             # print hue  
66         if (maxrgb == 0):  
67             s = 0  
68         else:  
69             s = (1.0 - (minrgb / maxrgb))  
70             s1.append(s)  
71             v = (maxrgb/255)  
72             v1.append(v)  
73             hsv[i ,j, 0] = hue  
74             hsv[i , j, 1] = s  
75             hsv[i , j, 2] = v  
76     # cv2.imshow('g', hsv)  
77     # cv2.waitKey(0)  
78     # cv2.destroyAllWindows()  
79     #Mean  
80     meanH = 0  
81     meanS = 0  
82     meanV = 0  
83     for i in range(0,len(hue1)):  
84         meanH += hue1[i]  
85         meanS += s1[i]  
86         meanV += v1[i]  
87     meanH = meanH / len(hue1)  
88     _meanH.append(meanH)  
89     meanS = meanS / len(s1)  
90     _meanS.append(means)  
91     meanV = meanV / len(v1)  
92     _meanV.append(meanV)  
93     #Deviation  
94     stdH = 0  
95     stdS = 0  
96     stdV = 0  
97     for i in range(0,len(hue1)):  
98         stdH += (hue1[i]-meanH)**2  
99         stdS += (s1[i] - meanS)**2  
100        stdV += (v1[i] - meanV)**2  
101    stdH = (stdH / len(hue1))**(1 / 2.0)  
102    _stdH.append(stdH)  
103    stdS = (stdS / len(s1))**(1/2.0)  
104    _stdS.append(stdS)  
105    stdV = (stdV / len(v1))**(1 / 2.0)
```

```

103     _stdV.append(stdV)
104     #Skewness
105     skwH = 0
106     skwS = 0
107     skwV = 0
108     for i in range(0, len(hue1)):
109         skwH += ((hue1[i] - meanH)**3)
110         skwS += ((s1[i] - meanS)**3)
111         skwV += ((v1[i] - meanV)**3)
112     skwH = ((skwH/len(hue1))**(1/3.0)).real
113     _skwH.append(skwH)
114     skwS = ((skwS/len(s1))**(1/3.0)).real
115     _skwS.append(skwS)
116     skwV = ((skwV/len(v1))**(1/3.0)).real
117     _skwV.append(skwV)

```

Penjelasan *Source code*:

1. Baris 1-6 merupakan *library* yang digunakan selama proses ekstraksi ciri HSV *color moment*.
2. Baris 7-15 deklarasi dari variabel *mean*, standar deviasi, dan *skewness* yang digunakan untuk menyimpan hasil perhitungan ekstraksi ciri *color moment*.
3. Baris 19-23 digunakan untuk membuka *file* direktori dari citra hasil *preprocessing* dan variabel *tempNama* digunakan untuk mengecek perpindahan file dari kelas citra makanan, dan dilakukan set nilai 0 terhadap variabel jumlah jika setiap *file* kelas baru dan digunakan untuk penomoran kelas citra hasil *color moment*.
4. Baris 26 digunakan untuk mengambil nilai tinggi dan lebar dari citra.
5. Baris 27-30 deklarasi variabel yang digunakan untuk menyimpan perhitungan nilai konversi dari nilai RGB ke dalam HSV.
6. Baris 31-42 proses pengambilan nilai *red*, *green*, dan *blue* dari setiap piksel. Proses seleksi kondisi untuk mengambil piksel objek citra. Kemudian dicari nilai maksimal dan minimal dari nilai RGB yang digunakan untuk perhitungan nilai *hue*, *saturation*, dan *value*.
7. Baris 43-72 proses perhitungan konversi nilai RGB ke HSV dari setiap piksel dan disimpan dalam variabel *hue1*, *s1*, dan *v1*.
8. Baris 77-89 proses perhitungan fitur *mean* untuk setiap ruang warna *hue*, *saturation*, dan *value*. Dan seluruh perhitungan *mean* disimpan dalam variabel *_meanH*, *_meanS*, dan *_meanV*.
9. Baris 91-103 proses perhitungan standar deviasi untuk setiap ruang warna *hue*, *saturation*, dan *value*. Dan seluruh perhitungan standar deviasi disimpan dalam variabel *_stdH*, *_stdS*, dan *_std_V*.
10. Baris 104-117 proses perhitungan *skewness* untuk setiap ruang warna *hue*, *saturaton*, dan *value*. Dan seluruh perhitungan *skewness* disimpan dalam variabel *_skwH*,

5.2.3 Algoritme Ekstraksi Fitur *Haralick*

Tahapan-tahapan dalam algoritme Ekstraksi fitur *haralick* adalah melakukan konversi ke citra grayscale, membuat matriks GLCM dengan nilai arah 0° , 45° , 90° , dan 135° , operasi penjumlahan matriks GLCM dengan matriks *tranpose* GLCM,

normalisasi matriks GLCM, dan tahapan terakhir adalah ekstraksi fitur *haralick*. Source code untuk algoritme ekstraksi fitur *haralick* dapat dilihat pada tabel 5.3.

Tabel 5.3 Source Code Algoritme Ekstraksi Fitur Haralick

No	
1	import cv2
2	import numpy as np
3	from math import log10
4	import os.path
5	from imutils import paths
6	import pandas as pd
7	namaFile = []
8	LH0 = []
9	LH45 = []
10	LH90 = []
11	LH135 = []
12	Cont0 = []
13	Cont45 = []
14	Cont90 = []
15	Cont135 = []
16	En0 = []
17	En45 = []
18	En90 = []
19	En135 = []
20	E0 = []
21	E45 = []
22	E90 = []
23	E135 = []
24	Corr0 = []
25	Corr45 = []
26	Corr90 = []
27	Corr135 = []
28	tempNama = ''
29	
30	for imagePath in paths.list_images("E:/Dataset_Skripsi/Data_fix_pakai_testing/") :
31	if (tempNama != str(os.path.split(os.path.dirname(imagePath))[-1])): tempNama = str(os.path.split(os.path.dirname(imagePath))[-1])
32	jumlah = 0
33	print (imagePath)
34	image2 = cv2.imread(imagePath)
35	image3 = cv2.resize(image2, (500, 500))
36	h, w = image3.shape[:2]
37	grayscale = cv2.cvtColor(image3, cv2.COLOR_BGR2GRAY)
38	# grayscale = [[195,35,211],[198,140,134],[198,191,175]]
39	maxGS = max(max(x) for x in grayscale)
40	#-----
41	-----
42	# NOL DERAJAT
43	Mglcm0 = [[0 for x in range(maxGS+1)] for y in range(maxGS+1)]
44	for i in range(0, h):
45	for j in range(0, w):
46	if(j == len(image1)-1):
47	continue

```
48         else:
49             value1 = grayscale[i][j]
50             value2 = grayscale[i][j+1]
51             Mglcm0[value1][value2] += 1
52     # matris transpose kemudian jumlah matrix glcm dan matrix
53     glcm
54     transpose0 = [list(i) for i in zip(*Mglcm0)]
55     for i in range(0, maxGS+1):
56         for j in range(0, maxGS+1):
57             Mglcm0[i][j] += transpose0[i][j]
58     # normaslisasi
59     jumlahpixel = 0.0
60     for i in range(0, maxGS+1):
61         for j in range(0, maxGS+1):
62             jumlahpixel += Mglcm0[i][j]
63     for i in range(0, maxGS+1):
64         for j in range(0, maxGS+1):
65             Mglcm0[i][j] /= jumlahpixel
66     #Perhitungan Ekstraksi Ciri
67     #Local Homogeneity, Contrast, Entropy, dan Energy
68     LH0 = 0
69     Cont0 = 0
70     En0 = 0
71     E0 = 0
72     for i in range(0, maxGS+1):
73         for j in range(0, maxGS+1):
74             LH0 += Mglcm0[i][j]/(1.0+((i-j)**2))
75             Cont0 += ((i-j)**2)*Mglcm0[i][j]
76             if(Mglcm0[i][j] != 0):
77                 En0 += Mglcm0[i][j]*log10(Mglcm0[i][j])
78             E0 += Mglcm0[i][j]**2
79     #Correlation
80     meani0 = 0.0
81     meanj0 = 0.0
82     for i in range(0, maxGS+1):
83         for j in range(0, maxGS+1):
84             meani0 += i*Mglcm0[i][j]
85             meanj0 += j*Mglcm0[i][j]
86     variani0 = 0.0
87     varianj0 = 0.0
88     for i in range(0, maxGS+1):
89         for j in range(0, maxGS+1):
90             variani0 += (((i - meani0) ** 2) * Mglcm0[i][j])
91             varianj0 += (((j - meanj0) ** 2) * Mglcm0[i][j])
92     variani0 = variani0 ** (1.0/2.0)
93     varianj0 = varianj0 ** (1.0/2.0)
94     Corr0 = 0.0
95     for i in range(0, maxGS+1):
96         for j in range(0, maxGS+1):
97             Corr0 += ((i-mani0)*(j-
meanj0)*Mglcm0[i][j])/(variani0*varianj0)
98
99     #-----
-----#
90     # 45 DERAJAT
91     Mglcm45 = [[0 for x in range(maxGS + 1)] for y in
range(maxGS + 1)]
92     for i in range(0, h):
```

```
93     for j in range(0, w):
94         if(j == 0):
95             continue
96         elif(i == len(imagel)-1):
97             continue
98         else:
99             value1 = grayscale[i][j]
100            value2 = grayscale[i+1][j-1]
101            Mg lcm45[value1][value2] += 1
102    # matris transpose kemudian jumlah matrix glcm dan matrix
103    glcm
104    transpose45 = [list(i) for i in zip(*Mg lcm45)]
105    for i in range(0, maxGS + 1):
106        for j in range(0, maxGS + 1):
107            Mg lcm45[i][j] += transpose45[i][j]
108    # normaslisasi
109    jumlahpixel = 0.0
110    for i in range(0, maxGS + 1):
111        for j in range(0, maxGS + 1):
112            jumlahpixel += Mg lcm45[i][j]
113    for i in range(0, maxGS + 1):
114        for j in range(0, maxGS + 1):
115            Mg lcm45[i][j] /= jumlahpixel
116    # Perhitungan Ekstraksi Ciri
117    # Local Homogeneity, Contrast, Entropy, dan Energy
118    LH45 = 0
119    Cont45 = 0
120    En45 = 0
121    E45 = 0
122    for i in range(0, maxGS + 1):
123        for j in range(0, maxGS + 1):
124            LH45 += Mg lcm45[i][j] / (1.0 + ((i - j) ** 2))
125            Cont45 += ((i - j) ** 2) * Mg lcm45[i][j]
126            if(Mg lcm45[i][j] != 0):
127                En45 += Mg lcm45[i][j]*log10(Mg lcm45[i][j])
128            E45 += Mg lcm45[i][j]**2
129    # Correlation
130    meani45 = 0.0
131    meanj45 = 0.0
132    for i in range(0, maxGS + 1):
133        for j in range(0, maxGS + 1):
134            meani45 += i * Mg lcm45[i][j]
135            meanj45 += j * Mg lcm45[i][j]
136    variani45 = 0.0
137    varianj45 = 0.0
138    for i in range(0, maxGS + 1):
139        for j in range(0, maxGS + 1):
140            variani45 += (((i - meani45) ** 2) * Mg lcm45[i][j])
141            varianj45 += (((j - meanj45) ** 2) * Mg lcm45[i][j])
142    variani45 = variani45 ** (1.0 / 2.0)
143    varianj45 = varianj45 ** (1.0 / 2.0)
144    Corr45 = 0.0
145    for i in range(0, maxGS + 1):
146        for j in range(0, maxGS + 1):
147            Corr45 += ((i - meani45) * (j - meanj45) *
148 Mg lcm45[i][j]) / (variani45 * varianj45)
149    #-----
```

```
150      # 90 DERAJAT
151      Mglcm90 = [[0 for x in range(maxGS + 1)] for y in
152          range(maxGS + 1)]
153          for i in range(0, h):
154              for j in range(0, w):
155                  if(i == len(image1)-1):
156                      continue
157                  else:
158                      value1 = grayscale[i][j]
159                      value2 = grayscale[i+1][j]
160                      Mglcm90[value1][value2] += 1
161      # matris transpose kemudian jumlah matrix glcm dan matrix
162      # glcm
163      transpose90 = [list(i) for i in zip(*Mglcm90)]
164      for i in range(0, maxGS + 1):
165          for j in range(0, maxGS + 1):
166              Mglcm90[i][j] += transpose90[i][j]
167      # normaslisasi
168      jumlahpixel = 0.0
169      for i in range(0, maxGS + 1):
170          for j in range(0, maxGS + 1):
171              jumlahpixel += Mglcm90[i][j]
172      for i in range(0, maxGS + 1):
173          for j in range(0, maxGS + 1):
174              Mglcm90[i][j] /= jumlahpixel
175      # Perhitungan Ekstraksi Ciri
176      # Local Homogeneity, Contrast, Entropy, dan Energy
177      LH90 = 0
178      Cont90 = 0
179      En90 = 0
180      E90 = 0
181      for i in range(0, maxGS + 1):
182          for j in range(0, maxGS + 1):
183              LH90 += Mglcm90[i][j] / (1.0 + ((i - j) ** 2))
184              Cont90 += ((i - j) ** 2) * Mglcm90[i][j]
185              if(Mglcm90[i][j] != 0):
186                  En90 += Mglcm90[i][j]*log10(Mglcm90[i][j])
187              E90 += Mglcm90[i][j]**2
188      # Correlation
189      meani90 = 0.0
190      meanj90 = 0.0
191      for i in range(0, maxGS + 1):
192          for j in range(0, maxGS + 1):
193              meani90 += i * Mglcm90[i][j]
194              meanj90 += j * Mglcm90[i][j]
195      variani90 = 0.0
196      varianj90 = 0.0
197      for i in range(0, maxGS + 1):
198          for j in range(0, maxGS + 1):
199              variani90 += (((i - meani90) ** 2) * Mglcm90[i][j])
200              varianj90 += (((j - meanj90) ** 2) * Mglcm90[i][j])
201      variani90 = variani90 ** (1.0 / 2.0)
202      varianj90 = varianj90 ** (1.0 / 2.0)
203      Corr90 = 0.0
204      for i in range(0, maxGS + 1):
205          for j in range(0, maxGS + 1):
206              Corr90 += ((i - meani90) * (j - meanj90) *
Mglcm90[i][j]) / (variani90 * varianj90)
```

```
205
206     #-----
207     # 135 DERAJAT
208     Mg lcm135 = [[0 for x in range(maxGS + 1)] for y in
209     range(maxGS + 1)]
210     for i in range(0, h):
211         for j in range(0, w):
212             if(j == len(image1)-1):
213                 continue
214             elif(i == len(image1)-1):
215                 continue
216             else:
217                 value1 = grayscale[i][j]
218                 value2 = grayscale[i+1][j+1]
219                 Mg lcm135[value1][value2] += 1
220     # matris transpose kemudian jumlah matrix glcm dan matrix
221     # glcm
222     transpose135 = [list(i) for i in zip(*Mg lcm135)]
223     for i in range(0, maxGS + 1):
224         for j in range(0, maxGS + 1):
225             Mg lcm135[i][j] += transpose135[i][j]
226     # normasisasi
227     jumlahpixel = 0.0
228     for i in range(0, maxGS + 1):
229         for j in range(0, maxGS + 1):
230             jumlahpixel += Mg lcm135[i][j]
231     for i in range(0, maxGS + 1):
232         for j in range(0, maxGS + 1):
233             Mg lcm135[i][j] /= jumlahpixel
234     # Perhitungan Ekstraksi Ciri
235     # Local Homogeneity, Contrast, Entropy, dan Energy
236     LH135 = 0
237     Cont135 = 0
238     En135 = 0
239     E135 = 0
240     for i in range(0, maxGS + 1):
241         for j in range(0, maxGS + 1):
242             LH135 += Mg lcm135[i][j] / (1.0 + ((i - j) ** 2))
243             Cont135 += ((i - j) ** 2) * Mg lcm135[i][j]
244             if(Mg lcm135[i][j] != 0):
245                 En135 += Mg lcm135[i][j]*log10(Mg lcm135[i][j])
246             E135 += Mg lcm135[i][j]**2
247     # Correlation
248     meani135 = 0.0
249     meanj135 = 0.0
250     for i in range(0, maxGS + 1):
251         for j in range(0, maxGS + 1):
252             meani135 += i * Mg lcm135[i][j]
253             meanj135 += j * Mg lcm135[i][j]
254     variani135 = 0.0
255     varianj135 = 0.0
256     for i in range(0, maxGS + 1):
257         for j in range(0, maxGS + 1):
258             variani135 += (((i - meani135) ** 2) *  
Mg lcm135[i][j])  
             varianj135 += (((j - meanj135) ** 2) *  
Mg lcm135[i][j])  
             variani135 = variani135 ** (1.0 / 2.0)
```

```
259 varianj135 = varianj135 ** (1.0 / 2.0)
260 Corr135 = 0.0
261 for i in range(0, maxGS + 1):
262     for j in range(0, maxGS + 1):
263         Corr135 += ((i - meani135) * (j - meanj135) *
Mglcm135[i][j]) / (variani135 * varianj135)
264     _LH0.append(LH0)
265     _LH45.append(LH45)
266     _LH90.append(LH90)
267     _LH135.append(LH135)
268     _Cont0.append(Cont0)
269     _Cont45.append(Cont45)
270     _Cont90.append(Cont90)
271     _Cont135.append(Cont135)
272     _En0.append(En0)
273     _En45.append(En45)
274     _En90.append(En90)
275     _En135.append(En135)
276     _E0.append(E0)
277     _E45.append(E45)
278     _E90.append(E90)
279     _E135.append(E135)
280     _Corr0.append(Corr0)
281     _Corr45.append(Corr45)
282     _Corr90.append(Corr90)
283     _Corr135.append(Corr135)
```

Penjelasan Source Code:

1. Baris 1-6 merupakan *library* yang digunakan selama proses ekstraksi ciri *haralick*.
2. Baris 7-28 variabel yang digunakan untuk menyimpan seluruh fitur dari semua citra beserta kelas.
3. Baris 30-33 digunakan untuk membuka *file* direktori dari citra hasil *preprocessing* dan variabel *tempNama* digunakan untuk mengecek perpindahan file dari kelas citra makanan, dan dilakukan set nilai 0 terhadap variabel jumlah jika setiap *file* kelas baru dan digunakan untuk penomoran kelas citra hasil ekstraksi *haralick*.
4. Baris 38 Proses konversi ke citra *grayscale*.
5. Baris 40 inisialisasi variabel *maxGS* untuk menyimpan nilai maksimal dari piksel *grayscale*.
6. Baris 43 Deklarasi matriks GLCM arah 0° dengan besar sesuai dengan nilai maksimal piksel *grayscale* (*maxGS*).
7. Baris 44-51 Proses inisialisasi dengan melakukan operasi penjumlahan nilai 1 pada matriks GLCM sesuai dengan arah piksel 0° .
8. Baris 53-56 Proses operasi penjumlahan matriks GLCM 0° dengan matriks *transpose* GLCM 0° .
9. Baris 58-64 Proses Normalisasi Matriks GLCM 0° dengan melakukan operasi pembagian nilai matriks dengan nilai total keseluruhan matriks.
10. Baris 67-70 inisialisasi variabel fitur *haralick* untuk menyimpan hasil perhitungan fitur *haralick* dengan arah 0° setiap citra.
11. Baris 73 proses perhitungan fitur *local homogeneity* matriks GLCM dengan arah 0° .

12. Baris 74 proses perhitungan fitur *Contrast* matriks GLCM dengan arah 0° .
13. Baris 76 proses perhitungan fitur *Entropy* matriks GLCM dengan arah 0° .
14. Baris 77 proses perhitungan fitur *Energy* matriks GLCM dengan arah 0° .
15. Baris 79-92 proses mencari nilai rata-rata dan standar deviasi yang digunakan dalam perhitungan fitur *correlation*.
16. Baris 96 proses perhitungan fitur *Energy* matriks GLCM dengan arah 0° .
17. Baris 91 Deklarasi matriks GLCM arah 45° dengan besar sesuai dengan nilai maksimal piksel *grayscale* (maxGS).
18. Baris 92-101 Proses inisialisasi dengan melakukan operasi penjumlahan nilai 1 pada matriks GLCM sesuai dengan arah piksel 45° .
19. Baris 103-106 Proses operasi penjumlahan matriks GLCM 45° dengan matriks *transpose* GLCM 45° .
20. Baris 108-114 Proses Normalisasi Matriks GLCM 45° dengan melakukan operasi pembagian nilai matriks dengan nilai total keseluruhan matriks.
21. Baris 117-120 inisialisasi variabel fitur *haralick* untuk menyimpan hasil perhitungan fitur *haralick* dengan arah 45° setiap citra.
22. Baris 123 proses perhitungan fitur *local homogeneity* matriks GLCM dengan arah 45° .
23. Baris 124 proses perhitungan fitur *Contrast* matriks GLCM dengan arah 45° .
24. Baris 126 proses perhitungan fitur *Entropy* matriks GLCM dengan arah 45° .
25. Baris 127 proses perhitungan fitur *Energy* matriks GLCM dengan arah 45° .
26. Baris 130-143 proses mencari nilai rata-rata dan standar deviasi yang digunakan dalam perhitungan fitur *correlation*.
27. Baris 147 proses perhitungan fitur *Energy* matriks GLCM dengan arah 45° .
28. Baris 151 Deklarasi matriks GLCM arah 90° dengan besar sesuai dengan nilai maksimal piksel *grayscale* (maxGS).
29. Baris 152-159 Proses inisialisasi dengan melakukan operasi penjumlahan nilai 1 pada matriks GLCM sesuai dengan arah piksel 90° .
30. Baris 161-164 Proses operasi penjumlahan matriks GLCM 90° dengan matriks *transpose* GLCM 90° .
31. Baris 166-172 Proses Normalisasi Matriks GLCM 90° dengan melakukan operasi pembagian nilai matriks dengan nilai total keseluruhan matriks.
32. Baris 175-178 inisialisasi variabel fitur *haralick* untuk menyimpan hasil perhitungan fitur *haralick* dengan arah 90° setiap citra.
33. Baris 181 proses perhitungan fitur *local homogeneity* matriks GLCM dengan arah 90° .
34. Baris 182 proses perhitungan fitur *Contrast* matriks GLCM dengan arah 90° .
35. Baris 184 proses perhitungan fitur *Entropy* matriks GLCM dengan arah 90° .
36. Baris 185 proses perhitungan fitur *Energy* matriks GLCM dengan arah 90° .
37. Baris 187-200 proses mencari nilai rata-rata dan standar deviasi yang digunakan dalam perhitungan fitur *correlation*.
38. Baris 204 proses perhitungan fitur *Energy* matriks GLCM dengan arah 90° .
39. Baris 208 Deklarasi matriks GLCM arah 135° dengan besar sesuai dengan nilai maksimal piksel *grayscale* (maxGS).
40. Baris 209-218 Proses inisialisasi dengan melakukan operasi penjumlahan nilai 1 pada matriks GLCM sesuai dengan arah piksel 135° .

41. Baris 220-223 Proses operasi penjumlahan matriks GLCM 135° dengan matriks transpose GLCM 135° .
42. Baris 225-231 Proses Normalisasi Matriks GLCM 135° dengan melakukan operasi pembagian nilai matriks dengan nilai total keseluruhan matriks.
43. Baris 234-237 inisialisasi variabel fitur *haralick* untuk menyimpan hasil perhitungan fitur *haralick* dengan arah 135° setiap citra.
44. Baris 240 proses perhitungan fitur *local homogeneity* matriks GLCM dengan arah 135° .
45. Baris 241 proses perhitungan fitur *Contrast* matriks GLCM dengan arah 135° .
46. Baris 243 proses perhitungan fitur *Entropy* matriks GLCM dengan arah 135° .
47. Baris 244 proses perhitungan fitur *Energy* matriks GLCM dengan arah 135° .
48. Baris 246-259 proses mencari nilai rata-rata dan standar deviasi yang digunakan dalam perhitungan fitur *correlation*.
49. Baris 263 proses perhitungan fitur *Energy* matriks GLCM dengan arah 135° .
50. Baris 264-283 proses meyimpan seluruh nilai fitur *haralick* dari semua citra.

5.2.4 Algoritme Klasifikasi Naive Bayes

Tahapan-tahapan dalam algoritme klasifikasi Naive Bayes adalah melakukan pelatihan pada data latih yaitu dengan mencari nilai rata-rata dan standar deviasi dari seluruh fitur. Pada pengujian dilakukan perhitungan *posterior probability* setiap kelas dan tahap terakhir adalah pengambilan kesimpulan dengan sesuai dengan nilai maksimal dari *posterior probability*. *Source code* untuk algoritme klasifikasi *naive bayes* dapat dilihat pada Tabel 5.4.

Tabel 5.4 Source Code Algoritme Klasifikasi Naive Bayes

No	
1	import math
2	import pandas as pd
3	
4	exceltesting =
	pd.read_excel("E:/Dataset_Skripsi/fiturdatatesting.xlsx",
	"Sheet1")
5	def separateByTesting(dataset):
6	separated = {}
7	for i in range(len(dataset)):
8	vector = [dataset['Mean H'][i],
9	dataset['Mean S'][i],
10	dataset['Mean V'][i],
11	dataset['Standar Deviasi H'][i],
12	dataset['Standar Deviasi S'][i],
13	dataset['Standar Deviasi V'][i],
14	dataset['Skewness H'][i],
15	dataset['Skewness S'][i],
16	dataset['Skewness V'][i],
17	dataset['Local Homogeneity 0'][i],
18	dataset['Local Homogeneity 45'][i],
19	dataset['Local Homogeneity 90'][i],
20	dataset['Local Homogeneity 135'][i],
21	dataset['Contrast 0'][i],
22	dataset['Contrast 45'][i],
23	dataset['Contrast 90'][i],

```
24             dataset['Contrast 135'][i],
25             dataset['Entropy 0'][i],
26             dataset['Entropy 45'][i],
27             dataset['Entropy 90'][i],
28             dataset['Entropy 135'][i],
29             dataset['Energy 0'][i],
30             dataset['Energy 45'][i],
31             dataset['Energy 90'][i],
32             dataset['Energy 135'][i],
33             dataset['Correlation 0'][i],
34             dataset['Correlation 45'][i],
35             dataset['Correlation 90'][i],
36             dataset['Correlation 135'][i])
37     if (dataset['Kelas'][i] not in separated):
38         separated[dataset['Kelas'][i]] = []
39     separated[dataset['Kelas'][i]].append(vector)
40 return separated
41 dataTesting = separateByTesting(exceltesting)
42 print("---- DATA Testing")
43 print(dataTesting)
44
45 excel = pd.read_excel("E:/Dataset_Skripsi/fitur.xlsx",
46 "Sheet1")
47 def separateByClass(dataset):
48     separated = {}
49     for i in range(len(dataset)):
50         vector = [dataset['Mean H'][i],
51                   dataset['Mean S'][i],
52                   dataset['Mean V'][i],
53                   dataset['Standar Deviasi H'][i],
54                   dataset['Standar Deviasi S'][i],
55                   dataset['Standar Deviasi V'][i],
56                   dataset['Skewness H'][i],
57                   dataset['Skewness S'][i],
58                   dataset['Skewness V'][i],
59                   dataset['Local Homogeneity 0'][i],
60                   dataset['Local Homogeneity 45'][i],
61                   dataset['Local Homogeneity 90'][i],
62                   dataset['Local Homogeneity 135'][i],
63                   dataset['Contrast 0'][i],
64                   dataset['Contrast 45'][i],
65                   dataset['Contrast 90'][i],
66                   dataset['Contrast 135'][i],
67                   dataset['Entropy 0'][i],
68                   dataset['Entropy 45'][i],
69                   dataset['Entropy 90'][i],
70                   dataset['Entropy 135'][i],
71                   dataset['Energy 0'][i],
72                   dataset['Energy 45'][i],
73                   dataset['Energy 90'][i],
74                   dataset['Energy 135'][i],
75                   dataset['Correlation 0'][i],
76                   dataset['Correlation 45'][i],
77                   dataset['Correlation 90'][i],
78                   dataset['Correlation 135'][i])
79     if (dataset['Kelas'][i] not in separated):
80         separated[dataset['Kelas'][i]] = []
81     separated[dataset['Kelas'][i]].append(vector)
82 return separated
```

```
82 print("---- Training")
83 training = separateByClass(excel)
84 _meanFitur = [[0 for i in range(len(excel.columns) - 1)] for j
85 in range(len(training))] #buat array sepanjang banyak fitur, dan
86 banyak kelas
87 #Hitung Mean
88 index = 0
89 for i in training:
90     mean = [0 for i in range(len(excel.columns) - 1)]
91     for j in range(len(training[i])):
92         for k in range(len(training[i][j])):
93             mean[k] += training[i][j][k]
94     for k in range(len(training[i][0])):
95         _meanFitur[index][k] = mean[k] / len(training[i])
96     index += 1
97 print("Mean")
98 print (_meanFitur)
99 #Hitung Standar Deviasi
100 index = 0
101 _stdFitur = [[0 for i in range(len(excel.columns) - 1)] for j
102 in range(len(training))] #buat array sepanjang banyak fitur, dan
103 banyak kelas
104 for i in training:
105     std = [0 for i in range(len(excel.columns) - 1)]
106     for j in range(len(training[i])):
107         for k in range(len(training[i][j])):
108             std[k] += (training[i][j][k] -
109             _meanFitur[index][k]) ** 2
110     for k in range(len(training[i][0])):
111         _stdFitur[index][k] = (std[k] / (len(training[i]) - 1))
112     index += 1
113 print("Standar Deviasi")
114 print (_stdFitur)
115 #Testing
116 print("---- Testing")
117 likelihood = [[[0 for i in range(len(excel.columns) - 1)] for
118 j in range(len(training))] for z in range(len(exceltesting))]
119 index2 = 0
120 for z in dataTesting:
121     for x in range(len(dataTesting[z])):
122         index = 0
123         for i in training:
124             for j in range(len(training[i][0])):
125                 # MENGGUNAKAN LIKELIHOOD
126                 akar = (2*math.pi*_stdFitur[index][j])** (1/2.0)
127                 eksponen = math.e**(-((dataTesting[z][x][j]-
128                 _meanFitur[index][j])**2)/(2*(_stdFitur[index]
129                 [j]**2)))
130                 # print(akar)
131                 final = 1/akar*eksponen
132                 # LAPLACIAN SMOOTHING
133                 likelihood[index2][index][j] = (final +
134                 1)/len(training)
135             index+=1
136         index2+=1
137 print("Likelihood")
```

```
125 print(_likelihood[0][11])
126 _final = [[0 for i in training]for j in
127 range(len(exceltesting))]
128 index2 = 0
129 for z in dataTesting:
130     for x in range(len(dataTesting[z])):
131         index = 0
132         for i in training:
133             # print (i)
134             temp = 1
135             for j in range(len(training[i][0])):
136                 # print(j)
137                 temp *= _likelihood[index2][index][j]
138             _final[index2][index]=temp *
139             (len(training[i])/len(excel))
140             index +=1
141             index2+=1
142 index2 = 0
143 for z in dataTesting:
144     for x in range(len(dataTesting[z])):
145         index = 0
146         kelas = []
147         print ("Data Testing ke-"+str(index2+1))
148         maksimal = max(_final[index2])
149         for i in training:
150             print(str(i)+". "+str(_final[index2][index]))
151             if(_final[index2][index] == maksimal):
152                 kelas.append(i)
153             index+=1
154         print("Nilai maksimalnya adalah "+str(maksimal)+""
155             adalah kelas dengan kode "+str(kelas))
156             index2+=1
```

Penjelasan *Source code*:

1. Baris 1-2 merupakan *library* yang digunakan selama proses klasifikasi *naive bayes*.
2. Baris 4 membaca *file excel* dari fitur data testing pada *sheet 1*.
3. Baris 5 method yang digunakan untuk mengambil variabel nilai fitur dari *file excel*.
4. Baris 6 inisialisasi variabel *separated* yang digunakan untuk meyimpan nilai seluruh fitur.
5. Baris 8-36 proses pengambilan fitur sesuai dengan judul kolom fitur.
6. Baris 37-39 proses pemisahan nilai fitur sesuai dengan kelas.
7. Baris 41 proses inisialisasi variabel *dataTesting* sesuai dengan nilai kembalian method *separatedByTesting* yaitu nilai fitur.
8. Baris 45 membaca *file excel* dari fitur data latih pada *sheet 1*.
9. Baris 46 method yang digunakan untuk mengambil variabel nilai fitur dari *file excel*.
10. Baris 47 inisialisasi variabel *separated* yang digunakan untuk meyimpan nilai seluruh fitur.
11. Baris 48-77 proses pengambilan fitur sesuai dengan judul kolom fitur.
12. Baris 78-80 proses pemisahan nilai fitur sesuai dengan kelas.

13. Baris 83 proses inisialisasi variabel training sesuai dengan nilai kembalian method separatedByClass yaitu nilai fitur.
14. Baris 84-85 deklarasi variabel _meanFitur yang digunakan untuk menyimpan nilai rata-rata fitur setiap kelas.
15. Baris 89-96 proses perhitungan mencari nilai rata-rata fitur setiap kelas.
16. Baris 91 Deklarasi variabel _stdFitur yang digunakan untuk menyimpan nilai standar deviasi fitur setiap kelas.
17. Baris 94-100 Proses perhitungan mencari nilai standar deviasi fitur setiap kelas.
18. Baris 107-108 deklarasi variabel _likelihood yang digunakan untuk menyimpan nilai *likelihood* fitur setiap kelas.
19. Baris 110-119 proses perhitungan nilai *likelihood* fitur setiap kelas dengan menggunakan nilai fitur data uji dan hasil perhitungan nilai rata-rata dan standar deviasi fitur.
20. Baris 121 proses *laplacian smoothing* untuk mendapatkan nilai *likelihood* tidak sama dengan 0.
21. Baris 126 deklarasi variabel _final untuk menyimpan nilai hasil perhitungan *posterior probability*.
22. Baris 129-140 Proses perhitungan *posterior probability*.
Baris 143-155 Proses pengambilan kesimpulan klasifikasi berdasarkan nilai maksimum *posterior probability*.

BAB 6 HASIL PENGUJIAN DAN ANALISIS

Bab ini akan menjelaskan pengujian yang dilakukan pada penelitian ini, yaitu pengujian terhadap perbandingan jumlah data latih dan data uji, dan pengujian pada masing-masing metode ekstraksi fitur terhadap akurasi. Pengujian dilakukan pada 12 kelas citra makanan dan bertujuan untuk menunjukkan hasil kinerja sistem.

6.1 Pengujian Jumlah Data Latih dan Data Uji Terhadap Akurasi

Pengujian pertama yang dilakukan adalah pengujian pengaruh jumlah data latih dan data uji terhadap akurasi. Banyak kumpulan data citra adalah 276, dengan banyak data 23 citra untuk setiap kelas dan terdiri dari 12 kelas, selain itu pengujian dilakukan sebanyak 5 kali dengan data uji yang berbeda. Banyak data latih dan data uji yang akan dilakukan pengujian dapat dilihat pada Tabel 6.1.

Tabel 6.1 Perbandingan Jumlah Data Latih dan Data Uji

	Data Latih	Data Uji
	240	36
	228	48
	216	60
	204	72
	192	84

Hasil pengujian *Confusion matrix* yang menggunakan data latih sebanyak 240 citra dan data uji sebanyak 36 citra dapat dilihat pada tabel 6.2.

Tabel 6.2 Confusion Matrix 240 Data latih dan 36 Data Uji

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Srawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
Hasil Nilai sebenarnya	Donat	7	0	1	0	0	1	4	0	0	0	2	0
	Rendang	0	4	0	2	0	0	0	2	6	1	0	0
	Mentimun	0	0	9	0	0	0	0	0	3	0	3	0

Tabel 6.2 Confusion Matrix 240 Data latih dan 36 Data Uji (Lanjutan)

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Strawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
Hasil Nilai sebenarnya	strawberry	0	4	0	9	0	0	1	0	0	1	0	0
	Pisang Ijo	1	0	3	0	11	0	0	0	0	0	0	0
	Pisang Kuning	1	0	0	0	2	12	0	0	0	0	0	0
	Jeruk Ijo	1	2	1	0	0	0	9	0	0	0	0	2
	Beng Beng	0	0	0	0	0	0	0	9	2	4	0	0
	Tim Tam	0	0	0	0	0	0	0	10	3	2	0	0
	Gerry Salut	0	0	0	1	0	0	0	0	5	9	0	0
	Biskuat Coklat	1	0	0	0	0	0	0	0	0	0	13	1
	Genji Pie	0	0	0	0	0	0	2	0	0	3	4	6

Hasil pengujian *Confusion matrix* yang menggunakan data latih sebanyak 228 citra dan data uji sebanyak 48 citra dapat dilihat pada tabel 6.3.

Tabel 6.3 Confusion Matrix 228 Data latih dan 48 Data Uji

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Srawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
Hasil Nilai sebenarnya	Donat	10	0	0	0	0	3	4	0	0	0	3	0
	Rendang	1	7	0	4	0	0	0	2	4	2	0	0
	Mentimun	3	0	10	0	0	0	0	0	3	0	4	0
	srawberry	0	4	0	12	0	0	1	0	0	3	0	0
	Pisang Ijo	0	0	4	0	15	0	1	0	0	0	0	0
	Pisang Kuning	2	0	0	0	0	18	0	0	0	0	0	0
	Jeruk Ijo	5	3	1	0	0	0	11	0	0	0	0	0
	Beng Beng	0	2	0	0	0	0	0	12	3	3	0	0
	Tim Tam	0	0	0	0	0	0	0	10	4	6	0	0
	Gerry Salut	0	3	0	2	0	0	0	0	6	9	0	0
	Biskuat Coklat	4	0	0	0	0	0	0	0	0	0	14	2

Tabel 6.3 Confusion Matrix 19 Data latih dan 4 Data Uji (Lanjutan)

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Srawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
	Genji Pie	1	0	0	0	0	0	4	0	0	3	5	7

Hasil pengujian *Confusion matrix* yang menggunakan data latih sebanyak 216 citra dan data uji sebanyak 60 citra dapat dilihat pada tabel 6.4.

Tabel 6.4 Confusion Matrix 216 Data latih dan 60 Data Uji

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Srawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
Hasil Nilai sebenarnya	Donat	15	0	1	0	0	3	3	0	0	0	3	0
	Rendang	0	9	1	4	0	0	0	4	4	3	0	0
	Mentimun	2	0	14	0	0	2	0	0	4	0	3	0
	srawberry	0	3	0	15	0	0	1	0	0	6	0	0
	Pisang Ijo	0	0	4	0	20	0	1	0	0	0	0	0
	Pisang Kuning	3	0	0	0	1	20	0	0	0	0	1	0
	Jeruk Ijo	6	3	2	0	0	0	11	0	0	0	0	3
	Beng Beng	0	0	0	0	0	0	0	18	7	0	0	0

Tabel 6.4 Confusion Matrix 216 Data latih dan 60 Data Uji (Lanjutan)

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Strawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
Hasil Nilai sebenarnya	Tim Tam	0	0	0	0	0	0	0	10	8	7	0	0
	Gerry Salut	0	4	0	3	0	0	0	0	7	11	0	0
	Biskuat Coklat	4	0	0	0	0	0	0	0	0	0	18	3
	Genji Pie	0	0	0	0	0	0	8	0	1	2	4	10

Hasil pengujian *Confusion matrix* yang menggunakan data latih sebanyak 204 citra dan data uji sebanyak 72 citra dapat dilihat pada tabel 6.5.

Tabel 6.5 Confusion Matrix 204 Data latih dan 72 Data Uji

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Strawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
Hasil Nilai sebenarnya	Donat	17	0	1	0	1	4	4	0	0	0	3	0
	Rendang	1	12	0	5	0	0	1	3	4	3	1	0
	Mentimun	4	0	15	0	0	2	0	0	4	0	5	0
	strawberry	0	5	0	14	0	0	1	0	0	7	0	3

Tabel 6.5 Confusion Matrix 204 Data latih dan 72 Data Uji (Lanjutan)

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Strawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
Hasil Nilai sebenarnya	Pisang Ijo	0	0	8	0	21	1	0	0	0	0	0	0
	Pisang Kuning	3	0	1	0	2	22	0	0	0	0	2	0
	Jeruk Ijo	6	3	5	0	0	0	11	0	0	0	0	5
	Beng Beng	0	2	0	0	0	0	0	19	6	3	0	0
	Tim Tam	0	0	0	0	0	0	0	11	8	11	0	0
	Gerry Salut	0	3	0	5	0	0	0	1	5	16	0	0
	Biskuat Coklat	4	0	0	0	0	0	0	0	0	0	22	4
	Genji Pie	1	0	0	0	0	0	6	0	4	0	3	16

Hasil pengujian *Confusion matrix* yang menggunakan data latih sebanyak 192 citra dan data uji sebanyak 84 citra dapat dilihat pada tabel 6.6.

Tabel 6.6 Confusion Matrix 192 Data latih dan 84 Data Uji

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Strawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Salut	Biskuat Coklat	Genji Pie
	Donat	20	0	2	0	1	2	4	0	0	1	5	0

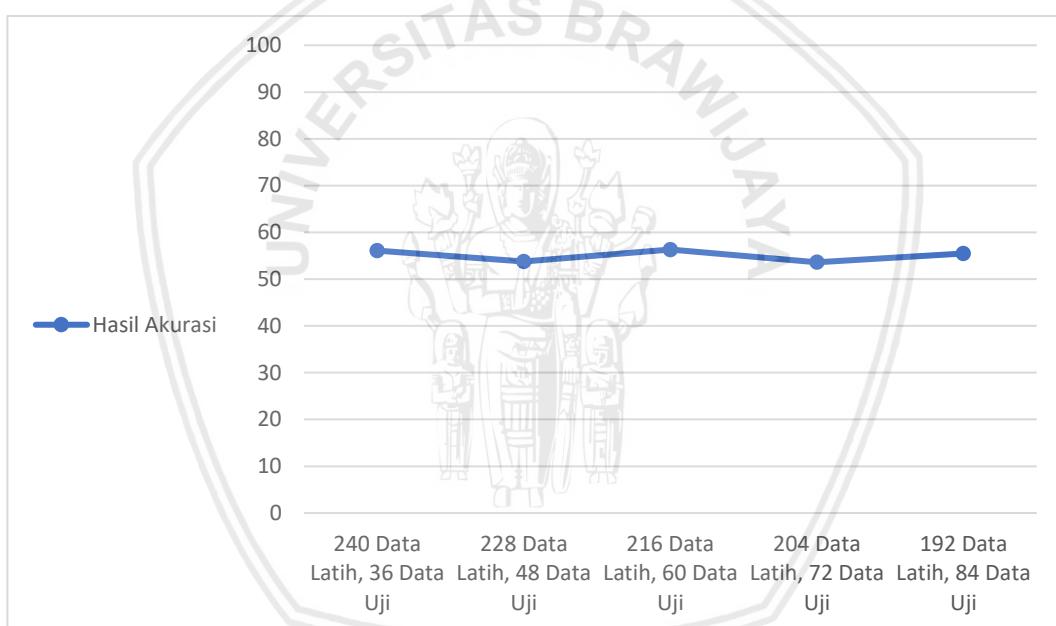
Tabel 6.6 Confusion Matrix 192 Data latih dan 84 Data Uji (Lanjutan)

		Hasil Prediksi											
		Donat	Rendang	Mentimun	Srawberry	Pisang Ijo	Pisang Kuning	Jeruk Ijo	Beng Beng	Tim Tam	Gerry Saluut	Biskuat Coklat	Genji Pie
	Ren dan g	0	13	1	3	0	0	1	4	8	5	0	0
	Mentimun	2	0	20	0	2	0	0	0	4	0	7	0
	srawberry	0	5	0	19	0	0	1	0	0	7	0	3
	Pisang Ijo	0	0	11	0	24	0	0	0	0	0	0	0
	Pisang Kuning	4	0	0	0	3	26	0	0	0	0	2	0
	Jeruk Ijo	5	3	5	0	0	0	18	0	0	0	0	4
	Beng Beng	0	3	0	0	0	0	0	19	8	5	0	0
	Tim Tam	0	0	0	0	0	0	0	15	8	12	0	0
	Gerry Saluut	0	4	0	1	0	0	0	0	13	17	0	0
	Biskuat Coklat	4	0	0	0	0	0	0	0	0	0	28	3
	Genji Pie	1	0	0	0	0	0	8	0	4	0	1	21

Dari hasil perhitungan *confusion matrix* dengan perbandingan data latih dan uji untuk nilai rata-rata *recall*, rata-rata *precision*, *f-score*, dan akurasi dapat dilihat pada Tabel berikut

Tabel 6.7 Tabel Akurasi Perbandingan Data Latih dan Uji

Data Latih	Data Uji	Rata-rata <i>Recall</i>	Rata-rata <i>Precision</i>	<i>F-Score</i>	Akurasi
240	36	0,56	0,58	2,75882E-08	56,11%
228	48	0,54	0,57	1,59192E-08	53,75%
216	60	0,56	0,59	8,26585E-08	56,33%
204	72	0,54	0,55	2,28455E-08	53,61%
192	84	0,55	0,58	4,72549E-08	55,47%



Gambar 6.1 Grafik Akurasi Perbandingan Data Latih dan Uji

Analisis hasil Tabel 6.7 menunjukkan hasil nilai *F-Score* tidak ada yang bernilai 0, sehingga menunjukkan semua kelas data uji dapat terklasifikasi. Kemudian analisis berikutnya terhadap nilai rata rata *recall* dan *precision* yang digunakan untuk menghitung akurasi sistem menunjukkan nilai yang stabil dengan akurasi diantara 50-60%. Sehingga untuk menghasilkan akurasi terbaik tidak dapat ditentukan berdasarkan banyak data latih dan data uji.

Berdasarkan grafik akurasi perbandingan data latih dan data uji yang ditunjukkan pada Gambar 6.1, akurasi terbaik pada pengujian yang menggunakan 216 data latih dan 60 data uji dengan nilai akurasi 56,33%. Dari 5 kali hasil pengujian perbandingan jumlah data latih dan data uji, menunjukkan nilai akurasi

naik dan turun, sehingga pada penelitian ini akurasi terbaik tidak dapat ditentukan berdasarkan banyak jumlah data latih dan data uji.

Pengujian berikutnya menggunakan *K-Fold Cross Validation* dengan nilai K sebesar 5. Total seluruh kumpulan data dibagi menjadi 5 bagian dengan total citra sebanyak 55 citra untuk setiap bagian. Kemudian secara bergantian untuk setiap bagian dilakukan proses pengujian. Hasil akurasi Pengujian *K-Fold Cross Validation* dapat dilihat pada Tabel 6.8 berikut.

Tabel 6.8 Akurasi Pengujian *K-Fold Cross Validation*

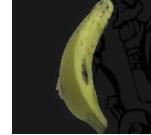
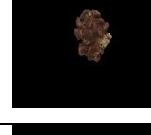
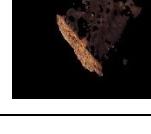
Data Latih	Data Uji	K Fold Cross Validation					Rata-rata Akurasi
		K = 1	K = 2	K = 3	K = 4	K = 5	
221	55	61,82%	60%	61,81%	65,45%	58,93%	61,60%
166	110	60%	60,91%	62,73%	63,96%	62,16%	61,95%
111	165	60,61%	59,39%	63,61%	60,24%	57,23%	58,22%
56	220	55,91%	46,15%	46,61%	58,82%	50,23%	51,54%

Hasil analisis pengujian *K-Fold Cross Validation* dengan menggunakan kumpulan data sebanyak 276 citra dan K sebesar 5 menghasilkan kinerja sistem dengan rata-rata akurasi terbaik sebesar 61,95% dengan menggunakan data latih dan data uji sebanyak 166 dan 110 citra. Kemudian analisis terhadap 4 kali pengujian menunjukkan nilai rata-rata akurasi yang stabil, sehingga pada penelitian ini untuk menghasilkan akurasi terbaik tidak dapat ditentukan berdasarkan banyak data latih dan data uji.

6.2 Pengujian Metode HSV *Color Moment*, *Haralick* dan *Naive Bayes Classifier* Terhadap Akurasi

Pengujian dilakukan pada 3 skenario yang berbeda, yaitu menggunakan metode ekstraksi fitur HSV *color moment*, metode ekstraksi fitur *haralick*, dan terakhir pada kedua metode HSV *color moment* dan *haralick*. Pengujian dilakukan dengan menggunakan data latih dan data uji dengan akurasi terbaik pada pengujian sebelumnya dengan menggunakan 18 data latih dan 5 data uji. Hasil Pengujian fitur ekstraksi fitur dapat dilihat pada Tabel 6.8, Tabel 6.9, dan Tabel 6.10. Analisis hasil Tabel 6.8 pada fitur yang hanya menggunakan HSV *color moment* terdapat citra makanan yang tidak terkласifikasi dengan benar. Jika dilihat dari visualisasi Hal ini disebabkan karena citra makanan memiliki kemiripan warna yang sama, dapat dilihat pada Gambar 6.2. Namun jika dilihat dari nilai fitur, citra yang tidak terkласifikasi memiliki perbedaan nilai fitur yang sedikit dengan hasil klasifikasi. Hasil klasifikasi untuk pengujian fitur *Color moment* dapat dilihat pada Lampiran A.1.

Tabel 6.9 Hasil Klasifikasi Pengujian HSV *Color Moment*

No	Data Uji	Kelas Sebenarnya	Kelas Prediksi
1		Donat	Donat
2		Rendang	Rendang
3		Mentimun	Mentimun
4		Strawberry	Strawberry
5		Pisang Ijo	Mentimun
6		Pisang Kuning	Pisang Kuning
7		Jeruk Ijo	Donat
8		Beng Beng	Beng Beng
9		Tim Tam	Beng Beng

Tabel 6.10 Hasil Klasifikasi Pengujian HSV *Color Moment* (Lanjutan)

No	Data Uji	Kelas Sebenarnya	Kelas Prediksi
10		Gerry Saluut	Gerry Saluut
11		Biskuat Coklat	Biskuat Coklat
12		Genji Pie	Biskuat Coklat

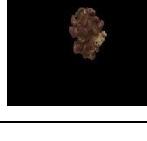
**Gambar 6.2 Citra Tidak Terkласifikasi Dengan Baik Pada HSV *Color Moment*****Tabel 6.11 Perbandingan Nilai Fitur Pada HSV *Color Moment***

Nilai Fitur Citra Data Uji								
μH	μS	μV	σH	σS	σV	skH	skS	skV
0,158	0,558	0,442	0,111	0,161	0,164	0,174	0,159	0,083
Nilai Rata-Rata Fitur Kelas Data Latih Hasil Klasifikasi								
μH	μS	μV	σH	σS	σV	skH	skS	skV
0,232	0,647	0,439	0,118	0,184	0,222	0,169	0,132	0,096
Nilai Standar Deviasi Ekstraksi Fitur <i>Color Moment</i>								
μH	μS	μV	σH	σS	σV	skH	skS	skV
0,027	0,081	0,079	0,028	0,033	0,023	0,023	0,071	0,017

Analisis hasil Tabel 6.8 terhadap nilai standar deviasi ke-sembilan fitur *Color moment* yang digunakan, memiliki besaran nilai yang mendekati nilai 0 pada ke-sembilan fitur, menunjukkan sebaran data antara nilai satu fitur dengan nilai rata-rata fitur memiliki jarak yang dekat dan menimbulkan kemiripan nilai fitur dan menyebabkan citra tidak terkласifikasi dengan baik seperti pada contoh Gambar 6.2. jika dilihat secara visualisasi citra makanan tidak terkласifikasi

memiliki kemiripan warna hijau pada kelas mentimun, pisang ijo, jeruk ijo, dan warna coklat pada kelas donat, rendang, beng-beng, tim-tam, gerry saluut, biskuat coklat, dan genji.

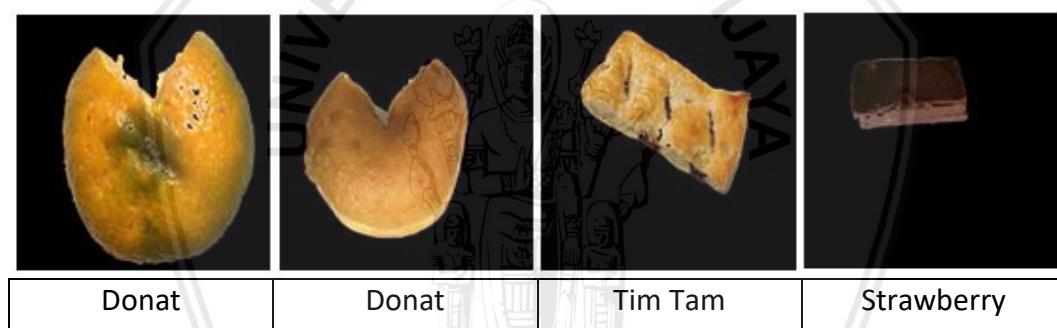
Tabel 6.12 Hasil Klasifikasi Pengujian Haralick

No	Data Uji	Kelas Sebenarnya	Kelas Prediksi
1		Donat	Donat
2		Rendang	Beng Beng
3		Mentimun	Biskut Coklat
4		Strawberry	Rendang
5		Pisang Ijo	Donat
6		Pisang Kuning	Pisang Kuning
7		Jeruk Ijo	Donat
8		Beng Beng	Beng Beng

Tabel 6.13 Hasil Klasifikasi Pengujian Haralick (Lanjutan)

No	Data Uji	Kelas Sebenarnya	Kelas Prediksi

9		Tim Tam	Strawberry
10		Gerry Saluut	Tim Tam
11		Biskuat Coklat	Pisang Ijo
12		Genji Pie	Tim Tam



Gambar 6.3 Citra Tidak Terkласifikasi Pada Fitur Haralick

Tabel 6.14 Perbandingan Nilai Fitur Pada Haralick

Fitur Haralick	Nilai Fitur Citra Data Uji	Nilai Rata-Rata Fitur Kelas Data Latih Hasil Klasifikasi	Nilai Standar Deviasi Ekstraksi Fitur Haralick
<i>Local Homogeneity 0°</i>	0,629	0,707	0,108
<i>Local Homogeneity 45°</i>	0,598	0,697	0,111
<i>Local Homogeneity 90°</i>	0,618	0,710	0,108
<i>Local Homogeneity 135°</i>	0,599	0,697	0,111
<i>Contrast 0°</i>	78,669	105,846	43,443

Tabel 6.15 Perbandingan Nilai Fitur Pada Haralick (Lanjutan)

Fitur Haralick	Nilai Fitur Citra Data Uji	Nilai Rata-Rata Fitur Kelas Data Latih Hasil Klasifikasi	Nilai Standar Deviasi Ekstraksi Fitur Haralick
<i>Contrast 45°</i>	123,4949	137,566	57,260
<i>Contrast 90°</i>	101,054	94,184	36,633
<i>Contrast 135°</i>	123,776	135,784	58,204
<i>Entropy 0°</i>	-2,122	-1,583	0,513
<i>Entropy 45°</i>	-2,188	-1,607	0,522
<i>Entropy 90°</i>	-2,146	-1,575	0,513
<i>Entropy 135°</i>	-2,188	-1,608	0,523
<i>Energy 0°</i>	0,219	0,405	0,164
<i>Energy 45°</i>	0,216	0,402	0,165
<i>Energy 90°</i>	0,218	0,405	0,164
<i>Energy 135°</i>	0,216	0,402	0,165
<i>Correlation 0°</i>	0,989	0,986	0,003
<i>Correlation 45°</i>	0,983	0,981	0,004
<i>Correlation 90°</i>	0,986	0,987	0,003
<i>Correlation 135°</i>	0,983	0,982	0,004

Kemudian pengujian selanjutnya dilakukan skenario dengan hanya menggunakan fitur *Haralick*. Dari Analisis hasil Tabel 6.9 berdasarkan nilai standar deviasi, fitur *contrast* memiliki nilai standar deviasi yang besar menunjukkan nilai sebaran nilai piksel ketetanggaan yang bervariasi, sehingga fitur *contrast* baik digunakan untuk mengenali citra makanan. Selain itu fitur *correlation* yang memiliki nilai standar deviasi yang kecil mendekati 0, menunjukkan tingkat korelasi antara sebuah piksel dan piksel tetangga yang rendah. Kemudian fitur *local homogeneity* memiliki nilai standar deviasi yang kecil, menunjukkan tingkat kehomogenan yang rendah. Selanjutnya fitur *Energy* memiliki nilai standar deviasi yang kecil, menunjukkan tidak terjadi keseragaman nilai piksel ketetanggaan atau bersifat acak. Terakhir fitur *Entropy* memiliki nilai standar deviasi yang kecil, menunjukkan kompleksitas piksel ketetanggaan yang rendah.

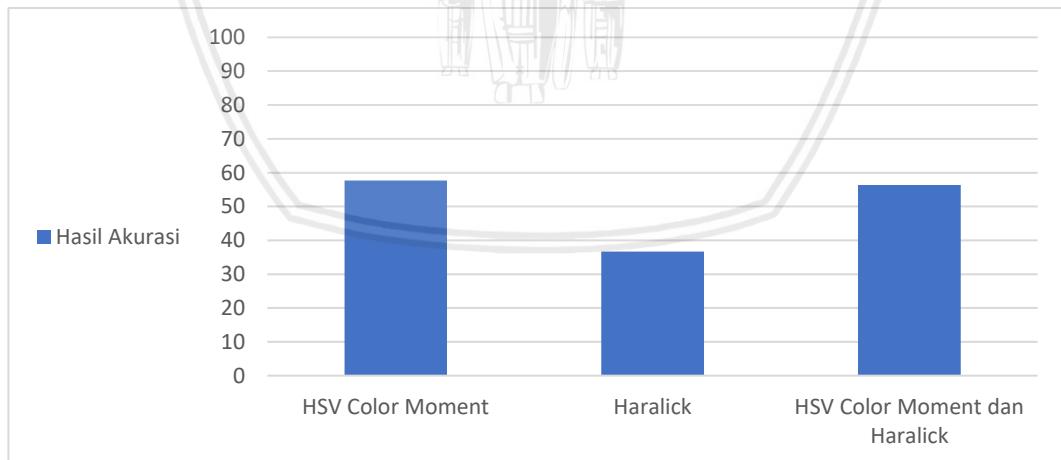
Dari fitur *haralick*, 16 dari 20 fitur yang digunakan memiliki nilai standar deviasi yang rendah, menghasilkan kemiripan nilai fitur sehingga menyebabkan citra makanan tidak terkласifikasi dengan baik seperti pada Gambar 6.3. Hal ini disebabkan karena terdapat variasi tekstur citra seperti, menggunakan $\frac{1}{4}$ bagian, $\frac{1}{2}$ bagian, dan $\frac{3}{4}$ bagian citra makanan setelah dimakan. Selain itu hasil segmentasi yang kurang baik pada tahap *pre-processing* juga membuat tekstur makanan menjadi tidak dikenali. Hasil klasifikasi untuk pengujian fitur *Haralick* dapat dilihat pada Lampiran A.2.



Gambar 6.4 Hasil Segmentasi Citra yang Kurang Baik

Tabel 6.16 Hasil Akurasi Pengujian Ekstraksi Fitur

No	Metode	Akurasi
1	<i>HSV Color Moment</i>	57,66 %
2	<i>Haralick</i>	36,67 %
3	<i>HSV Color Moment</i> dan <i>Haralick</i>	56,33 %



Gambar 6.5 Grafik Akurasi Pengujian Metode Ekstraksi Fitur

Berdasarkan Hasil pengujian penggunaan fitur warna dapat memberikan hasil lebih baik dibandingkan dengan penggunaan fitur tekstur, dilihat dari akurasi yang dihasilkan *HSV color moment* lebih besar dibandingkan dengan *haralick* dengan nilai 57.66% dan 36.67%. Kemudian penggunaan kedua fitur mampu memperbaiki hasil akurasi dari fitur tekstur dengan menghasilkan nilai akurasi 56.33%.

BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan yang didapat selama melakukan penelitian ini adalah sebagai berikut.

1. Dalam melakukan klasifikasi citra makanan menggunakan metode *Naive Bayes Classifier*, Perbandingan data latih dan data uji menghasilkan nilai akurasi naik dan turun, sehingga penggunaan perbandingan data latih dan data uji tidak dapat digunakan untuk menghasilkan nilai akurasi yang lebih baik, dengan menghasilkan nilai akurasi terbaik yaitu 56,33% dengan data latih dan data uji sebanyak 216 dan 60 citra. Kemudian hasil pengujian dengan *5-Fold Cross Validation* menghasilkan rata-rata akurasi terbaik yaitu 61,95% dengan data latih dan data uji sebanyak 166 dan 110 citra.
2. Dari penelitian ini dilakukan perbandingan metode ekstraksi fitur yang digunakan dalam melakukan klasifikasi citra makanan menggunakan *Naive Bayes Classifier*. Metode ekstraksi fitur yang digunakan antara lain, fitur warna atau HSV *Color Moment* mendapatkan nilai akurasi sebesar 57,66% dan fitur tekstur atau *Haralick* mendapatkan nilai akurasi sebesar 36,67%. Kemudian gabungan antara dua fitur HSV *Color moment* dan *Haralick* mampu menghasilkan nilai akurasi lebih baik dibandingkan hanya menggunakan fitur tekstur yaitu dengan akurasi 56,33%.

7.2 Saran

Saran yang didapat dari hasil penelitian ini dengan tujuan dapat diimplementasikan pada penelitian berikutnya sebagai berikut.

1. Diperlukan pemilihan metode *pre-processing* yang tepat guna menghasilkan nilai ekstraksi fitur yang lebih baik.
2. Dari pengujian warna dan tekstur, beberapa fitur memiliki standar deviasi yang rendah yang berarti memiliki kemiripan nilai fitur. Diperlukan penambahan kumpulan data latih untuk menambah variansi data.
3. Diperlukan pengujian terhadap jumlah *distance* pada fitur tekstur untuk melihat pengaruh piksel ketetanggaan yang lebih besar.

Diperlukan pemilihan fitur yang tepat dan penambahan fitur bentuk dengan tujuan dalam melakukan klasifikasi dapat memberikan hasil citra yang lebih spesifik

DAFTAR REFERENSI

- Bekkari, A., Idbraim, S., Elhassouny, A. & Mammass, D., 2012. SVM and Haralick Features for Classification of High Resolution Satellite Images from Urban Areas. p. 2.
- Deng, X., Liu, Q., Deng, Y. & Mahadevan, S., 2016. An Improved Method to Construct Basic Probability Assignment Based on the Confusion Matrix for Classification Problem. *Information Sciences*, Volume 340-341, pp. 250-261.
- Hadi, W., Al-Radaideh, Q. A. & Alhawari, S., 2018. Integrating Associative Rule-based Classification with Naive Bayes for Text Classification. *Applied Soft Computing*, Volume 69, pp. 344-356.
- Halim, A., Dewi, C. & Angkasa, S., 2013. Aplikasi Image Retrieval Menggunakan Kombinasi Metode Color Moment dan Gabor Texture. Volume 14, p. 2.
- Ireland, J. & Moller, A., 2016. Food Classification and Description. *Encyclopedia of Food and Health*, p. 1.
- Kryjak, T., Komorkiewicz, M. & Gorgon, M., 2018. Real-time Hardware-software Embedded Vision System for ITS Smart Camera Implemented in Zynq SoC. pp. 123-159.
- Kusumanto, R. & Tompunu, A. N., 2011. Pengolahan Citra Digital untuk Mendeteksi Objek Menggunakan Pengolahan Warna Model Normalisasi RGB. p. 1.
- Liantoni, F. & Nugroho, H., 2015. Klasifikasi Daun Herbal Menggunakan Metode Naive Bayes Classifier dan K-Nearest Neighbor. *Jurnal Simantec*, Volume 5, p. 4.
- Loesdau, M., Chabrier, S. & Gabillon, A., 2014. Hue and Saturation in the RGB Color Space. pp. 203-212.
- Pundlik, R., 2016. Comparison of Sensitivity for Consumer Load Data Using Gaussian Naive Bayes (GNB) and Logistic Regression (LR). *7th International Conference on Intelligent Systems*, p. 5.
- Suryanto, E. D., Suherman & Sihombing, P., 2015. Ekstraksi Fitur Haralick pada Citra Mikroskop Digital Trinocular untuk Identifikasi Cacing Penyakit Kaki Gajah. p. 1.
- Tsangaratos, P. & Ilia, L., 2016. Comparison of a Logistic Regression and Naive Bayes Classifier in Landslide Susceptibility Assessments: The Influence of Models Complexity and Training Dataset Size. *Catena*, pp. 164-179.
- Wu, J., Peng, B., Huang, Z. & Xie, J., 2013. Research on Computer Vision-Based Object Detection and Classification. *International Federation for Information Processing*, Volume 1, pp. 183-188.