

**KLASIFIKASI CITRA MAKANAN MENGGUNAKAN *HSV COLOR*
MOMENT DAN *LOCAL BINARY PATTERN* DENGAN
*NAÏVE BAYES CLASSIFIER***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Karunia Ayuningsih

NIM: 155150207111060



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

KLASIFIKASI CITRA MAKANAN MENGGUNAKAN *HSV COLOR MOMENT* DAN
LOCAL BINARY PATTERN DENGAN *NAÏVE BAYES CLASSIFIER*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Karunia Ayuningsih
NIM: 155150207111060

Skripsi ini telah diuji dan dinyatakan lulus pada
27 Desember 2018
Telah diperiksa dan disetujui oleh:

Pembimbing I

Yuita Arum Sari, S.Kom., M.Kom
NIK: 201609 880715 2 001

Pembimbing II



Putra Pandu Adikara, S.Kom., M.Kom
NIP: 19850725 200812 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Pr Astoro Kurniawan, S.T., M.T., Ph.D

NIP: 19710518 200312 1 001

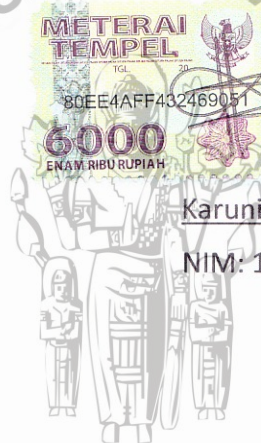


PERNYATAAN ORISINALITAS

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 pustaka.

Apabila ternyata di dalam 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, 07 Januari 2019



Karunia Ayuningsih

NIM: 155150207111060

PRAKATA

Puji syukur penulis ucapkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi ini. Shalawat serta salam tercurahkan pada Nabi Muhammad SAW. Pada kesempatan ini penulis juga menyampaikan rasa terima kasih kepada pihak-pihak yang telah membantu penulis sehingga dapat menyelesaikan skripsi ini, diantaranya:

1. Ibu Yuita Arum Sari, S.Kom., M.Kom selaku dosen pembimbing pertama yang telah memberikan bimbingan, saran, arahan, dan motivasi kepada penulis selama penyusunan skripsi.
2. Bapak Putra Pandu Adikara, S.Kom., M.Kom selaku dosen pembimbing kedua yang telah memberikan bimbingan, saran, dan arahan dalam penulisan kepada penulis selama penyusunan skripsi.
3. Seluruh Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu yang bermanfaat kepada penulis selama menempuh pendidikan.
4. Kedua orang tua dan keluarga penulis yang telah memberikan semangat, do'a, motivasi, dan dukungan kepada penulis selama menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya dan dalam menyelesaikan skripsi.
5. Sahabat terdekat penulis, khususnya Abdullah Abrar, Dita Ayu, Rifka Anisa, Anisa Redhita, dan Cindy Alkadina atas segala do'a, bantuan, dukungan dan motivasi yang diberikan kepada penulis.
6. Sahabat terdekat penulis selama menempuh pendidikan pada perkuliahan, Selma Aulia, Devinta Setyaningtyas, Austenita Pasca, Ismawati Nurjannah, Yugi Trilia, Frisma Yessy, dan Sarah Najla atas doa'a, motivasi, dukungan, dan bantuan yang diberikan selama masa Pendidikan kepada penulis.
7. Semua pihak yang telah membantu terselesainya skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari dalam penyusunan skripsi ini masih banyak kesalahan dan kekurangan baik dalam teknik penyajian materi maupun pembahasan. Demi kemajuan penulis, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga skripsi ini dapat bermanfaat bagi pembaca secara umum dan penulis. Akhir kata penulis ucapkan banyak terima kasih.

Malang, 07 Januari 2019

Karunia Ayuningsih

karuniaaayu@student.ub.ac.id

ABSTRAK

Karunia Ayuningsih, Klasifikasi Citra Makanan Menggunakan *HSV Color Moment* Dan *Local Binary Pattern* Dengan *Naïve Bayes Classifier*

Pembimbing: Yuita Arum Sari, S.Kom., M.Kom dan Putra Pandu Adikara, S.Kom., M.Kom.

Makanan merupakan kebutuhan dasar yang harus dipenuhi bagi kehidupan manusia. Kebiasaan makan dapat menimbulkan kebiasaan yang baik dan buruk. Kebiasaan makan buruk dapat menimbulkan berbagai macam penyakit. Komunikasi, informasi, dan edukasi (KIE) dapat memberikan edukasi terhadap kebiasaan makan. Makanan memiliki jenis beraneka ragam, diperlukan pengenalan jenis makanan agar mempermudah dalam melakukan klasifikasi jenis makanan yang baik. Tujuan dari penelitian ini agar dapat memberikan edukasi mengenali jenis makanan. Proses diawali dengan melakukan klasifikasi citra menggunakan *pre-processing* untuk memisahkan antara objek makanan dan *background*. Tahapan selanjutnya yaitu dengan menggunakan ekstraksi fitur warna *Hue Saturation Value* (HSV) yang terdiri dari fitur *Mean*, *Standard Deviation*, dan *Skewness*. Kemudian, dengan menggunakan ekstraksi fitur tekstur *Local Binary Pattern* (LBP) menghasilkan nilai keabuan dalam histogram. Hasil ekstraksi fitur dari setiap citra kemudian dilakukan klasifikasi menggunakan *Naïve Bayes Classifier*. Pengujian dilakukan dengan membandingkan nilai akurasi dari setiap metode yang digunakan. Adanya pengujian confusion matrix, dapat melihat nilai precision, recall, dan f-measure dari setiap metode. Berdasarkan hasil pengujian jika hanya menggunakan metode HSV menghasilkan nilai akurasi 65%. Selain itu, hasil dari pengujian yang telah dilakukan menggunakan metode HSV menghasilkan akurasi sebesar 65% dan metode LBP menghasilkan akurasi sebesar 60%

Kata kunci: makanan, kebiasaan makan, hue saturation value, local binary pattern, naïve bayes classifier

ABSTRAK

Karunia Ayuningsih, Klasifikasi Citra Makanan Menggunakan *HSV Color Moment* Dan *Local Binary Pattern* Dengan *Naïve Bayes Classifier*

Pembimbing: Yuita Arum Sari, S.Kom., M.Kom dan Putra Pandu Adikara, S.Kom., M.Kom.

Food is a basic need that must be fulfilled in human life. Eating habits can lead to good and bad habits. Bad eating habits can cause various diseases. Komunikasi, informasi, dan edukasi (KIE) can provide education on eating habits. Food has a variety of types, it is necessary to recognize the type of food to make it easier to classification good types of food. The purpose of this study is to be able to provide education to recognize the types of food. The process begins with image classification using pre-processing to separate between food objects and background. On top of that, using the Hue Saturation Value (HSV) color extraction feature consists of the feature Mean, the Standard Deviation, and the Skewness. Then is the use of the Local Binary Pattern (LBP) texture feature extraction produce feature extraction uses gray scales in the histogram. The results of feature extraction from each image are then carried out using the Naïve Bayes Classifier classification. The test result is comparing the value of accuracy from each method that used. Based on the test results, the use of only the HSV method produces a 65% accuracy value. Meanwhile, the use the LBP method, get a 60% accuracy value. In addition, the results of tests that have been carried out using the HSV method produce an accuracy of 65% and the LBP method produces an accuracy of 60%.

Keywords: *food, eating habits, obesity, hue saturation value, local binary pattern, naïve bayes classifiers*

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR ALGORITME	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Pengolahan Citra Digital	6
2.3 <i>Pre-Processing</i> Citra	6
2.4 <i>Hue Saturation Value</i> (HSV).....	7
2.4.1 Ekstraksi Fitur <i>Hue Saturation Value</i> (HSV).....	8
2.5 <i>Local Binary Pattern</i> (LBP).....	8
2.6 Algoritme <i>Naïve Bayes Classifier</i>	9
2.6.1 <i>Bernauli Naïve Bayes Classifier</i>	9
2.6.2 <i>Multinomial Naïve Bayes Classifier</i>	9
2.6.3 <i>Gaussian Naïve Bayes Classifier</i>	10
2.7 Evaluasi Kinerja Sistem	10
BAB 3 METODOLOGI	12
3.1 Tipe Penelitian	12

3.2 Metode Penelitian	12
3.3 Lokasi Penelitian	13
3.4 Metode Pengumpulan Data.....	13
3.5 Peralatan Pendukung.....	15
BAB 4 PERANCANGAN.....	16
4.1 Perancangan Algoritme Proses Implementasi.....	16
4.1.1 Implementasi <i>Pre-Processing</i>	16
4.1.2 Implementasi <i>Hue Saturation Value (HSV)</i>	17
4.1.3 Implementasi <i>Local Binary Pattern (LBP)</i>	18
4.1.4 Implementasi <i>Naïve Bayes Classifier</i>	19
4.2 Perhitungan Manualisasi	20
4.2.1 Perhitungan <i>Pre-Processing</i>	21
4.2.2 Perhitungan <i>Hue Saturation Value (HSV)</i>	26
4.2.3 Perhitungan <i>Local Binary Pattern (LBP)</i>	27
4.2.4 Perhitungan <i>Naïve Bayes Classifier</i>	29
4.2.5 Perhitungan Evaluasi Kinerja Sistem.....	32
BAB 5 IMPLEMENTASI	34
5.1 Batasan Implementasi	34
5.2 Implementasi Algoritme	34
5.2.1 Algoritme <i>Pre-processing</i> Citra	34
5.2.2 Algoritme HSV Citra.....	35
5.2.3 Algoritme Ekstraksi Fitur HSV Citra.....	36
5.2.4 Algoritme LBP Citra	38
5.2.5 Algoritme <i>Naïve Bayes Classifier</i>	39
BAB 6 HASIL PENGUJIAN DAN ANALISIS	42
6.1 Pengujian Metode HSV, LBP dan <i>Naïve Bayes Classifier</i> Terhadap Akurasi	42
BAB 7 PENUTUP	50
7.1 Kesimpulan.....	50
7.2 Saran	50
DAFTAR REFERENSI	51

DAFTAR TABEL

Tabel 2.1 <i>Confusion Matrix</i>	11
Tabel 4.1 Piksel RGB	22
Tabel 4.2 Piksel HSV	22
Tabel 4.3 Nilai Piksel Citra Asli	22
Tabel 4.4 Nilai <i>Grayscale</i>	23
Tabel 4.5 Nilai <i>Gaussian Blur</i>	23
Tabel 4.6 Nilai <i>Otsu Thresholding</i>	24
Tabel 4.7 Nilai <i>Erosion</i>	24
Tabel 4.8 Nilai <i>Dilation</i>	25
Tabel 4.9 Nilai Hasil <i>Masking</i>	25
Tabel 4.10 Nilai V.....	26
Tabel 4.11 Nilai S	26
Tabel 4.12 Nilai H	26
Tabel 4.13 Manualisasi HSV	26
Tabel 4.14 Nilai Mean	27
Tabel 4.15 Nilai <i>Standard Deviation</i>	27
Tabel 4.16 <i>Skewness</i>	27
Tabel 4.17 Nilai <i>Grayscale</i>	27
Tabel 4.18 Nilai Manualisasi LBP	29
Tabel 4.19 Data Latih	29
Tabel 4.20 <i>Mean</i> Data Latih	30
Tabel 4.21 <i>Standard Deviation</i> Data Latih	30
Tabel 4.22 Data Uji	31
Tabel 4.23 <i>Likelihood</i>	31
Tabel 4.24 <i>Posterior Probability</i>	31
Tabel 6.1 Hasil Pengujian Data Uji	42
Tabel 6.2 Hasil Pengujian Metode HSV	44
Tabel 6.3 Hasil Pengujian Metode LBP.....	44
Tabel 6.4 Hasil Pengujian Metode HSV dan LBP	45
Tabel 6.5 Hasil <i>Micro</i> dan <i>Macro Averaging</i>	45



Tabel 6.6 *Standard Deviation* Ekstraksi Fitur HSV..... 47
Tabel 6.7 *Standard Deviation* Ekstraksi Fitur LBP 47
Tabel 6.8 Visualisasi Warna..... 49



DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Metode Pelatihan.....	12
Gambar 3.2 Diagram Alir Metode Pengujian.....	13
Gambar 3.3 Data Latih	14
Gambar 3.4 Data Uji.....	14
Gambar 4.1 Diagram Alir <i>Pre-processing</i>	16
Gambar 4.2 Diagram Alir Ekstraksi Fitur HSV	17
Gambar 4.3 Diagram Alir LBP.....	19
Gambar 4.4 Diagram Alir Klasifikasi <i>Naïve Bayes</i>	20
Gambar 4.5 Citra Asli	21
Gambar 4.6 Citra Hasil Segmentasi.....	21
Gambar 4.7 Citra Asli	23
Gambar 4.8 <i>Grayscale</i>	23
Gambar 6.1 Grafik Hasil <i>Micro dan Macro Averaging</i>	45
Gambar 6.2 Data Latih	46
Gambar 6.3 Grafik Nilai <i>Standard Deviation HSV</i>	48
Gambar 6.4 Grafik Nilai <i>Standard Deviation LBP</i>	48

DAFTAR ALGORITME

Algoritme 1 Pre-processing citra.....	34
Algoritme 2 HSV	35
Algoritme 3 Ekstraksi Fitur HSV.....	36
Algoritme 4 LBP	38
Algoritme 5 <i>Naïve Bayes Classifier</i>	39



BAB 1

PENDAHULUAN

1.1 Latar belakang

Makanan merupakan kebutuhan dasar yang harus dipenuhi bagi kehidupan manusia (Ningsih, 2014). Kebiasaan makan adalah suatu tingkah laku untuk memenuhi kebutuhan hidup dengan asupan makanan. Kebiasaan makan dari memberikan status gizi yang berbeda pada setiap individu sehingga menimbulkan sikap, kepercayaan, dan pemilihan terhadap makanan. Oleh karena itu, kebiasaan makan dapat menimbulkan pengaruh baik dan buruk. Kebiasaan baik dapat menunjang terpenuhinya kebutuhan gizi. Namun, kebiasaan buruk dapat menimbulkan penyakit obesitas, hipertensi, gangguan jantung koroner, dan kencing manis (Kadir, 2016). Berdasarkan data, sebanyak 43,5% masyarakat memiliki kebiasaan makan buruk (Miko & Pratiwi, 2017). Hal tersebut menyebabkan tingginya risiko terkena penyakit.

Komunikasi, informasi, dan edukasi (KIE) adalah upaya perubahan yang dikelola oleh sekelompok orang, memiliki tujuan untuk mengubah dan memberikan gagasan pada sekelompok target yang dituju (Hapsari & Setiawan, 2008). KIE dapat digunakan sebagai upaya dalam menangani kebiasaan buruk (Miko & Pratiwi, 2017). Namun, makanan memiliki jenis yang beranekaragam sehingga mengalami kesulitan dalam mengenali jenis makanan baik yang untuk dikonsumsi. Oleh karena itu, diperlukan pengenalan jenis makanan agar mempermudah dalam melakukan klasifikasi jenis makanan.

Penelitian mengenai pengenalan citra buah telah dilakukan oleh Arivazhagan, *et al* (2010) menggunakan ekstraksi fitur warna dan fitur tekstur. Penelitian tersebut menunjukkan bahwa, ekstraksi fitur warna digabungkan dengan ekstraksi fitur tekstur akan menghasilkan hasil klasifikasi dengan nilai akurasi yang lebih besar dibandingkan hanya menggunakan fitur warna atau fitur tekstur. Hal ini disebabkan oleh fitur warna dapat meningkatkan warna gambar sehingga memperjelas tekstur dari gambar tersebut.

Penelitian yang dilakukan oleh Susilo, Fitriyah, & Setyawan (2018) dalam menentukan kualitas makanan ikan tongkol dapat ditentukan oleh kondisi daging dari segi warna. Penelitian tersebut menggunakan metode HSV untuk fitur warna dan menghasilkan akurasi sebesar 72,727%. Penelitian yang dilakukan oleh Budianita, Jasril, & Handayani (2015) untuk membedakan jenis daging sapi dan daging babi. Penelitian tersebut menghasilkan nilai akurasi sebesar 88,75% tanpa *background* dan 73,375% dengan *background*.

Penelitian mengenai ekstraksi fitur tekstur untuk mengklasifikasi citra daun herbal juga telah dilakukan oleh Aygun & Gunes (2017) menggunakan metode *Local Binary Pattern* (LBP), *Gray Level Co-Occurrence Matrix* (GLCM), dan *Relational Bit Operator* (RBO). Hasil dari penelitian tersebut menunjukkan metode LBP memiliki nilai akurasi yang paling besar diantaranya, yaitu sebesar 99,5%

untuk benih dan 93,5% untuk daun. Metode GLCM menghasilkan nilai akurasi sebesar 68% untuk benih dan 73,5% untuk daun. Selain itu, metode RBO menghasilkan nilai akurasi sebesar 57% untuk benih dan 70% untuk daun. Penelitian yang dilakukan oleh Amynarto, Sari, & Wihandika (2018) dalam mengenali citra emosi berdasarkan ekspresi mikro pada wajah seseorang menggunakan metode LBP menghasilkan nilai akurasi sebesar 70,21%. LBP dapat memudahkan dalam proses pengenalan citra dari segi tekstur (Turiyanto, et al., 2014).

Pada penelitian yang dilakukan oleh Liantoni & Nugroho (2015) menghasilkan sebuah sistem yang dapat mengklasifikasi citra daun herbal. Terdapat dua metode yang digunakan dalam klasifikasi, yaitu *Naïve Bayes Classifier* dan *K-Nearest Neighbor* (KNN). Hasil akurasi dari penelitian menghasilkan sebesar 75% untuk *Naïve Bayes* dan 70,83% untuk KNN. Pada penelitian yang dilakukan oleh Hafsa & Andono (2015) yaitu untuk mendeteksi penyakit kulit secara otomatis menggunakan algoritme *Naïve Bayes*. Penelitian ini menghasilkan nilai akurasi sebesar 73,33% dengan menggunakan pengenalan tekstur berupa histogram serta klasifikasi menggunakan algoritme *Naïve Bayes*. Berdasarkan hasil penelitian menunjukkan bahwa algoritme *Naïve Bayes* dapat digunakan sebagai klasifikasi.

Berdasarkan kebutuhan yang telah dijelaskan dalam latar belakang, terdapat tiga metode yang digunakan dalam penelitian ini. Metode fitur warna *Hue Saturation Value* (HSV) dan fitur tekstur *Local Binary Pattern* (LBP) digunakan untuk mendapatkan karakteristik dari setiap citra. Selain itu, tahapan dalam melakukan klasifikasi dengan menggunakan *Naïve Bayes Classifier*.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah yang dapat diangkat antara lain:

1. Bagaimana merancang metode *Local Binary Pattern* (LBP) dan *Hue Saturation Value* (HSV) dengan menggunakan *Naïve Bayes Classifier* dalam melakukan klasifikasi citra makanan?
2. Bagaimana hasil akurasi metode *Local Binary Pattern* (LBP) dan *Hue Saturation Value* (HSV) menggunakan *Naïve Bayes Classifier* dalam melakukan klasifikasi citra makanan?

1.3 Tujuan

Tujuan pada penelitian ini antara lain:

1. Dapat merancang metode *Local Binary Pattern* (LBP) dan *Hue Saturation Value* (HSV) dengan *Naïve Bayes Classifier* dalam melakukan klasifikasi citra makanan.
2. Mengetahui hasil akurasi dari metode *Local Binary Pattern* (LBP) dan *Hue Saturation Value* (HSV) menggunakan *Naïve Bayes Classifier* dalam melakukan klasifikasi citra makanan.

1.4 Manfaat

Manfaat dari penelitian ini antara lain:

1. Membantu dalam melakukan klasifikasi citra makanan berdasarkan perancangan yang digunakan.
2. Memberikan sarana referensi bagi penelitian selanjutnya.

1.5 Batasan masalah

Berdasarkan latar belakang, rumusan masalah, maupun tujuan dari penelitian ini, didapatkan batasan masalah sebagai berikut.

1. Data yang digunakan pada penelitian ini menggunakan data primer yang diambil secara langsung oleh peneliti.
2. Objek penelitian menggunakan jenis makanan yang padat.
3. Kelas yang digunakan dalam klasifikasi jenis makanan berjumlah lima jenis, yaitu donat, mie, nasi kuning, telur dadar dan tomat.

1.6 Sistematika pembahasan

Pada penulisan sistematika pembahasan yang digunakan sebagai penelitian dalam bentuk skripsi ini disusun secara sistematis dan dibagi dalam beberapa bab. Setiap bab memiliki urutan dan penjelasan masing-masing.

BAB 1 PENDAHULUAN

Pada Bab 1 Pendahuluan, menjelaskan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika pembahasan mengenai klasifikasi citra makanan menggunakan *Hue Saturation Value* (HSV) dan *Local Binary Pattern* (LBP) menggunakan metode *Naïve Bayes Classifier*.

BAB 2 LANDASAN KEPUSTAKAAN

Pada Bab 2 Landasan Kepustakaan, menjelaskan tentang teori-teori yang digunakan dalam penelitian ini. Terdiri dari pengolahan citra digital, *Hue Saturation Value* (HSV), Ekstraksi fitur *Hue Saturation Value* (HSV), *Local Binary Pattern* (LBP) dan Metode *Naïve Bayes Classifier*.

BAB 3 METODOLOGI

Pada Bab 3 Metodologi, menjelaskan tentang uraian secara lengkap mengenai metode yang digunakan serta memberikan penjelasan mengenai langkah-langkah yang dilakukan pada penelitian.

BAB 4 ANALISIS DAN PERANCANGAN

Pada Bab 4 Analisis dan Perancangan, menjelaskan mengenai analisis kebutuhan dan perancangan dari sistem.

BAB 5 IMPLEMENTASI

Pada Bab 5 Implementasi, menjelaskan mengenai implementasi yang diterapkan dalam sistem dan penerapan metode yang digunakan.

BAB 6 HASIL DAN PEMBAHASAN

Pada Bab 6 Hasil dan Pembahasan, menjelaskan mengenai hasil yang telah dilakukan pada tahap pengujian oleh sistem.

BAB 7 PENUTUP

Pada Bab 7 Penutup, menjelaskan mengenai kesimpulan serta saran yang didapatkan selama penelitian berlangsung.



BAB 2

LANDASAN KEPUSTAKAAN

Bab 2 berisi mengenai uraian dan pembahasan tentang teori, konsep, dan metode yang digunakan dalam penelitian.

2.1 Tinjauan Pustaka

Susilo, Fitriyah, & Setyawan (2018) melakukan penelitian yang bertujuan untuk menentukan kualitas makanan ikan tongkol berdasarkan kondisi daging dari segi warna. Penelitian tersebut menggunakan metode HSV untuk fitur warna dan menghasilkan akurasi sebesar 72,727%. Penelitian yang dilakukan oleh Budianita, Jasril, & Handayani (2015) untuk membedakan jenis daging sapi dan daging babi. Penelitian ini menghasilkan nilai akurasi sebesar 88,75% tanpa *background* dan 73,375% dengan *background*.

Fitur warna HSV memiliki tiga komponen warna yang terdiri dari *Hue*, *Saturation* dan *Value*. Setiap fitur warna dilakukan ekstraksi fitur untuk mendapatkan karakteristik yang ada pada masing-masing citra. Pada penelitian yang telah dilakukan oleh Singh & Hemachandran (2012), untuk mendapatkan nilai dari fitur warna pada citra digunakan ekstraksi fitur yang terdiri dari *Mean*, *Standard Deviation*, dan *Skewness* menghasilkan nilai akurasi sebesar 58,2%.

Penelitian mengenai ekstraksi fitur tekstur untuk mengklasifikasi citra daun herbal juga telah dilakukan oleh Aygun & Gunes (2017) LBP, GLCM, dan RBO. Hasil dari penelitian tersebut menunjukkan metode LBP memiliki nilai akurasi yang paling besar diantaranya metode GLCM dan RBO. Penelitian yang dilakukan oleh Arynarto, Sari, & Wihandika (2018) dalam mengenali citra emosi berdasarkan ekspresi mikro pada wajah seseorang menggunakan metode LBP menghasilkan nilai akurasi sebesar 70,21%. LBP dapat memudahkan dalam proses pengenalan citra dari segi tekstur (Turiyanto, et al., 2014).

Penelitian untuk mengenali citra makanan telah dilakukan sebelumnya oleh Arivazhagan, et al (2010). Ekstraksi fitur diperlukan untuk mendapatkan ciri dari setiap gambar. Penelitian tersebut menunjukkan bahwa ekstraksi fitur warna digabungkan dengan ekstraksi fitur tekstur akan menghasilkan hasil klasifikasi dengan nilai akurasi yang lebih besar dibandingkan hanya menggunakan fitur warna atau fitur tekstur. Hal ini disebabkan oleh fitur warna dapat meningkatkan warna gambar sehingga memperjelas tekstur dari gambar tersebut.

Pada penelitian yang dilakukan oleh Liantoni & Nugroho (2015) menghasilkan sebuah sistem yang dapat mengklasifikasi citra daun herbal. Terdapat dua metode yang digunakan dalam klasifikasi, yaitu *Naïve Bayes Classifier* dan *K-Nearest Neighbor* (KNN). Hasil akurasi dari penelitian menghasilkan sebesar 75% untuk *Naïve Bayes* dan 70,83% untuk KNN. Pada penelitian yang dilakukan oleh Hafsa & Andono (2015) yaitu untuk mendeteksi penyakit kulit secara otomatis menggunakan algoritme *Naïve Bayes*. Penelitian ini menghasilkan nilai akurasi

sebesar 73,33% dengan menggunakan pengenalan tekstur berupa histogram serta klasifikasi menggunakan algoritme *Naïve Bayes*. Berdasarkan hasil penelitian menunjukkan bahwa algoritme *Naïve Bayes* dapat digunakan sebagai sistem klasifikasi.

Pada penelitian ini menggabungkan fitur warna HSV yang dapat diterapkan dalam melakukan klasifikasi citra dan fitur tekstur LBP yang dapat mengatasi intensitas cahaya yang berbeda pada setiap citra makanan. Selain itu, metode *Naïve Bayes* dapat memberikan asumsi pada setiap fitur dapat berdiri sendiri. Hasil akurasi yang digunakan sesuai dengan jumlah data benar, semakin banyak data benar yang digunakan maka memiliki peluang hasil akurasi yang besar.

2.2 Pengolahan Citra Digital

Sebuah gambar yang terletak dalam bidang dua dimensi merupakan pengertian dari citra. Jika sebuah gambar yang terletak pada bidang dua dimensi namun merupakan suatu fungsi yang berkelanjutan dari intensitas cahaya merupakan pengertian citra secara matematis (Munir, 2004). Alat optik dapat menangkap bayangan dari objek disebabkan oleh bantuan pantulan cahaya. Adanya alat optik memudahkan citra dapat disimpan. Citra yang dihasilkan dalam bentuk analog dan komputer tidak bisa mengolah citra analog. Kemudian, diperlukan proses dalam mengubah citra analog menjadi citra digital (Hafsah & Andono, 2015).

2.3 Pre-Processing Citra

Pada pengolahan citra digital, terdapat tahapan yang disebut *pre-processing*, *pre-processing* adalah sebuah tahapan yang mendasar dalam mengolah citra digital kemudian untuk diidentifikasi ke tahap selanjutnya. Tahapan dalam *pre-processing* meliputi (Sari & Fadlil, 2013):

1. Konversi citra digital ke dalam RGB kemudian dilakukan konversi menjadi *grayscale* (Sari & Fadlil, 2013).
2. *Gaussian Blur* berfungsi untuk menempatkan warna transisi dalam citra sehingga menghasilkan efek lembut pada sisi citra. Digunakan untuk menghilangkan *noise* yang berasal dari pantulan cahaya pada kamera (Wedianto, et al., 2016).
3. *Otsu Thresholding* dilakukan untuk perhitungan persebaran nilai untuk mengetahui nilai batasan yang optimal. Dapat meminimalkan nilai variasi dalam kelas dan memaksimalkan nilai variasi antar kelas (Bangare, et al., 2015).
4. *Erosian* merupakan tahapan yang berguna melakukan pengerosian nilai piksel citra digital, hal ini digunakan untuk mengetahui nilai piksel yang berintensitas rendah (Sari & Fadlil, 2013).
5. *Dilation* merupakan tahapan untuk menambahkan piksel pada sekitar citra (Jawas & Suciati, 2013).

6. Masking merupakan tahapan yang berguna untuk membedakan antara objek digital dengan *background* yang ada di sekitar objek (Sari & Fadlil, 2013).

2.4 Hue Saturation Value (HSV)

Hue Saturation Value (HSV) terdiri dari tiga komponen warna yaitu *Hue*, *Saturation*, dan *Value*. Setiap komponen warna memiliki tiga karakteristik utama yaitu (Junhua & Jing, 2012):

1. *Hue* yaitu merupakan warna sebenarnya yang digunakan untuk menentukan kemerahan, kehijauan, dan lain – lain dalam warna.
2. *Saturation* merupakan warna sebenarnya yang digunakan untuk menentukan kedalaman warna misalnya, kemerahan warna yang dapat dibagi menjadi warna merah terang dan merah gelap yang diukur dalam persentase dari 0% sampai 100%.
3. *Value* merupakan kecerahan dari warna, dengan besaran nilai berkisar 0% – 100%. Apabila bernilai 0% maka berwarna hitam, dan bila menuju angka 100% maka warna akan semakin cerah.

Transformasi warna dari *Red Green Blue* (RGB) ke ruang warna *Hue Saturation Value* (HSV) dapat digunakan pada Persamaan 2.1, Persamaan 2.2, Persamaan 2.3, Persamaan 2.4, dan Persamaan 2.5 (OpenCV, 2018).

$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B)} \quad (2.1)$$

$$V = \max(r, g, b) \quad (2.2)$$

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ \frac{\max(r,g,b) - \min(r,g,b)}{\max(r,g,b)}, & V > 0 \end{cases} \quad (2.3)$$

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ 60 * \left(\frac{(g-b)}{\max(r,g,b) - \min(r,g,b)} \right), & \text{jika } V = r \\ 120 + 60 * \left(\frac{(b-r)}{\max(r,g,b) - \min(r,g,b)} \right), & \text{jika } V = g \\ 240 + 60 * \left(\frac{(r-g)}{\max(r,g,b) - \min(r,g,b)} \right), & \text{jika } V = b \end{cases} \quad (2.4)$$

$$H = H + 360, \text{ jika } H < 0 \quad (2.5)$$

Keterangan:

- R* : Nilai piksel *red*
- G* : Nilai piksel *green*
- B* : Nilai piksel *blue*
- H* : Nilai piksel *hue*
- S* : Nilai piksel *saturation*
- V* : Nilai piksel *value*

2.4.1 Ekstraksi Fitur Hue Saturation Value (HSV)

Penelitian ini menggunakan tiga ekstraksi fitur yang dapat digunakan untuk mendapatkan nilai dari fitur warna. Ekstraksi fitur dalam HSV antara lain (Singh & K., 2012):

1. Mean

Nilai rata-rata atau yang dapat disebut dengan *mean* memiliki nilai yang berbeda-beda. Pada umumnya citra yang memiliki intensitas cahaya yang tinggi akan menghasilkan *mean* yang tinggi, dan nilai yang rendah untuk citra dengan intensitas cahaya yang rendah. Persamaan *mean* dapat dilihat pada Persamaan 2.6.

$$\mu = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N I_{ij} \quad (2.6)$$

Keterangan:

- μ : Mean
- M : Nilai panjang citra
- N : Nilai lebar citra

2. Standard Deviation

Nilai *Standard Deviation* dari sebuah citra dapat dihitung melalui Persamaan 2.7.

$$\sigma = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (I_{ij} - \text{Mean})^2} \quad (2.7)$$

Keterangan:

- σ : Standard deviation
- M : Nilai panjang citra
- N : Nilai lebar citra

3. Skewness

Skewness digunakan untuk mengetahui letak perbedaan *mean* suatu citra. *Skewness* dapat dihitung pada Persamaan 2.8.

$$\text{Skewness} = \sqrt[3]{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (I_{ij} - \text{Mean})^3} \quad (2.8)$$

Keterangan:

- M : Nilai panjang citra
- N : Nilai lebar citra

2.5 Local Binary Pattern (LBP)

Timo Ojala pada (2002) adalah seseorang yang pertama kali memperkenalkan *Local Binary Pattern*. LBP merupakan sebuah metode analisis yang menggunakan model statistika dan struktur, yang mana operator *Local Binary Pattern* digunakan sebagai perbandingan nilai keabuan (*grayscale*) dari masing-masing piksel ketetangaan (Amat, et al., 2017). Terdapat fitur yang bernama histogram, yang

mana fitur tersebut berguna sebagai data statistik yang akan digunakan untuk mengolah data dari nilai piksel ketetangaan (Turiyanto, et al., 2014).

Setiap piksel yang memiliki nilai ketetangaan ke- n dapat disebut sebagai *thresholding*, yang mana menggunakan nilai keabuan (*grayscale*) dari Persamaan 2.9 dan fungsi *thresholding* $s(x)$ pada Persamaan 2.10 (Ojala, et al., 2002). Keuntungan yang didapatkan dengan menggunakan *Local Binary Pattern* yaitu mudah diterapkan serta membutuhkan waktu yang tidak lama untuk ekstraksi fitur dalam menghitung komputasi (Biglari, et al., 2014).

$$LBPP, R(X_c, Y_c) = \sum_{p=0}^{p-1} s(g_p - g_c) 2^p \tag{2.9}$$

$$s(x) \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2.10}$$

Keterangan:

- P : Jumlah piksel tetangga
- R : Nilai radius
- g_c : Nilai piksel x, y
- g_p : Nilai piksel tetangga

2.6 Algoritme *Naïve Bayes Classifier*

Naïve Bayes Classifier merupakan metode yang digunakan untuk mengklasifikasi data statistik yang dapat digunakan untuk memprediksi sebuah peluang dari anggota kelas yang ditentukan. Secara umum, rumus dari metode *Naïve Bayes* dapat dilihat pada Persamaan 2.11 (Saleh, 2015).

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \tag{2.11}$$

Keterangan:

- X : Data dengan *class* yang belum diketahui
- H : Hipotesis data merupakan suatu *class* spesifik
- $P(H|X)$: Probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)
- $P(H)$: Probabilitas hipotesis H (prior probabilitas)
- $P(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis H

2.6.1 *Bernaulli Naïve Bayes Classifier*

Bernaulli Naïve Bayes menggunakan jumlah data yang mengandung pada setiap kelas, bukan frekuensi kemunculan data. Jika *Multinomial* menggunakan jumlah kata maka *Bernaulli* menggunakan jumlah data setiap kelas (Karunia, et al., 2017).

2.6.2 *Multinomial Naïve Bayes Classifier*

Multinomial Naïve Bayes digunakan untuk pengklasifikasian kata atau dokumen. Metode ini memperhitungkan jumlah kemunculan dari setiap kata atau setiap dokumen (Kalokasari, et al., 2017).

2.6.3 Gaussian Naïve Bayes Classifier

Gaussian Naïve Bayes digunakan untuk data yang bersifat kontinyu. Karakteristik dari naïve bayes ini dengan mencari nilai *mean* dan *standard deviation* dari setiap fitur (Saleh, 2015).

Algoritme *Naïve Bayes* yang digunakan dalam penelitian ini adalah *Gaussian Naïve Bayes*. Digunakannya *Gaussian Naïve Bayes* karena data yang bersifat kontinyu. Langkah dalam mencari nilai *Naïve Bayes* dengan mencari nilai *mean* dari setiap fitur. Rumus *mean* dapat dilihat pada Persamaan 2.12. Selanjutnya mencari nilai *standard deviation* pada Persamaan 2.13. Mencari peluang *likelihood* menggunakan rumus Persamaan 2.14 (Saleh, 2015).

$$\mu = \frac{\sum_{i=1}^n x_i}{n} \tag{2.12}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n-1}} \tag{2.13}$$

$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \tag{2.14}$$

Keterangan:

- μ : Mean
- σ : Standard deviation
- P : Peluang
- X_i : Atribut ke- i
- x_i : Nilai atribut ke- i
- Y : Kelas yang dicari
- y_j : Sub kelas Y yang dicari
- n : Jumlah data
- π : Nilai π sebesar 3,14
- e : Nilai eksponensial sebesar 2,718

2.7 Evaluasi Kinerja Sistem

Metode evaluasi kinerja sistem yang digunakan dalam *confusion matrix* yaitu terdapat dua metode. Pertama yaitu *macro-averaging*, metode ini digunakan untuk mengetahui kinerja sistem dengan cara menghitung nilai dari semua kelas secara merata. Kedua yaitu *micro-averaging*, metode ini digunakan untuk menghitung nilai dari seluruh data (Sokolova & Lapalme, 2009).

Perhitungan kinerja sistem dengan *macro-averaging* dan *micro-averaging* menggunakan *precision*, *recall*, *f-measure*, dan akurasi pada masing-masing metode. Pertama, *precision* adalah jumlah dokumen yang ditemukan yang mana dokumen tersebut bersifat relevan sesuai dengan kebutuhan pengguna. Kedua, *recall* adalah jumlah dokumen yang ditemukan dari sebuah proses pencarian di sistem (Sokolova & Lapalme, 2009).

Ketiga, *f-measure* adalah pengukuran yang menilai hubungan timba balik antara *precision* dan *recall*. Jika dalam perhitungan nilai *precision* dan *recall*



memiliki perbedaan yang tinggi, dilakukan penyetaraan menggunakan *f-measure*. Terakhir adalah akurasi, yaitu tingkat kedekatan nilai yang sebenarnya dengan nilai prediksi. Tabel 2.1 menjelaskan *confusion matrix* yang digunakan dalam penilaian. Perhitungan *macro-averaging* dapat dilihat pada Persamaan dan *micro-averaging* dapat dilihat pada Persamaan (Sokolova & Lapalme, 2009).

Perhitungan dalam mencari nilai untuk *macro-averaging* dan *micro-averaging* pada setiap komponen *precision*, *recall*, *f-measure* dan akurasi dapat dilihat pada Persamaan 2.15, Persamaan 2.16, Persamaan 2.17 dan Persamaan 2.18 untuk *macro-averaging* dan Persamaan 2.19, Persamaan 2.20, Persamaan 2.21 dan Persamaan 2.22 untuk *micro-averaging*.

Tabel 2.1 Confusion Matrix

Actual Class	Predicted Class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

$$M \text{ Precision} = \frac{\sum_{i=1}^K \frac{tp_i}{tp_i + fp_i}}{K} \quad (2.15)$$

$$M \text{ Recall} = \frac{\sum_{i=1}^K \frac{tp_i}{tp_i + fn_i}}{K} \quad (2.16)$$

$$M \text{ F - Measure} = \frac{2 \times M \text{ Precision} \times M \text{ Recall}}{M \text{ Precision} + M \text{ Recall}} \quad (2.17)$$

$$M \text{ Akurasi} = \frac{\sum_i \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{K} \times 100\% \quad (2.18)$$

$$\mu \text{ Precision} = \frac{\sum_{i=1}^K tp_i}{\sum_{i=1}^K (tp_i + fp_i)} \quad (2.19)$$

$$\mu \text{ Recall} = \frac{\sum_{i=1}^K tp_i}{\sum_{i=1}^K (tp_i + fn_i)} \quad (2.20)$$

$$\mu \text{ F - Measure} = \frac{2 \times \mu \text{ Precision} \times \mu \text{ Recall}}{\mu \text{ Precision} + \mu \text{ Recall}} \quad (2.21)$$

$$\mu \text{ Akurasi} = \sum_i \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i} \times 100\% \quad (2.22)$$

Keterangan:

M : Macro

μ : Micro

K : Jumlah Kelas data

TP : Data positif terdeteksi positif

FP : Data negatif terdeteksi positif

TN : Data negatif terdeteksi negatif

FN : Data positif terdeteksi negatif

BAB 3 METODOLOGI

Pada Bab 3 berisi mengenai metode serta tahapan yang digunakan dalam menyelesaikan penelitian untuk klasifikasi citra makanan.

3.1 Tipe Penelitian

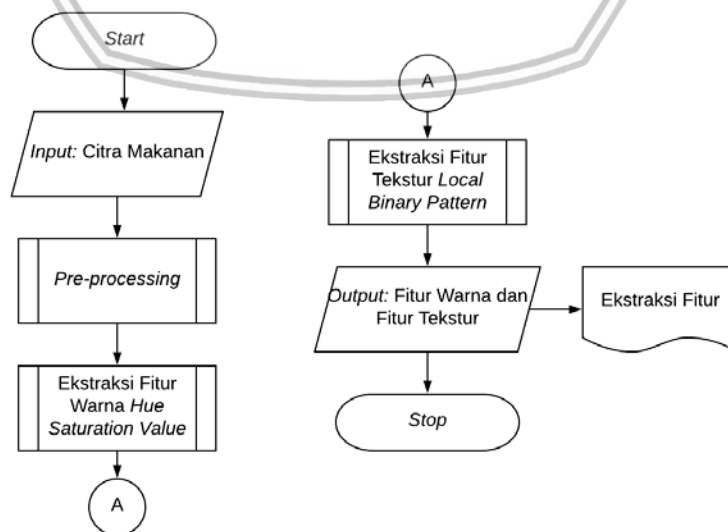
Tipe penelitian pada penelitian ini adalah non-implementatif analitik. Penelitian non-implementatif merupakan penelitian yang menitikberatkan pada kegiatan analisis yang mana menjelaskan hubungan antar elemen dalam objek penelitian.

3.2 Metode Penelitian

Metode penelitian menjelaskan mengenai alur program berjalan yang mana ditunjukkan pada Gambar 3.1. Pada Gambar 3.1 digunakan sebagai alir program untuk metode pelatihan pada data *training*. *Input* yang digunakan berasal dari data citra berupa gambar dengan objek berisi makanan. Selanjutnya citra makanan akan diolah dalam tahapan *pre-processing* agar menghasilkan gambar dengan objek makanan. Tujuan dari *pre-processing* untuk memisahkan antara objek makanan dan *background* dari objek makanan.

Hasil citra setelah melalui tahapan *pre-processing* kemudian akan diolah untuk mendapatkan ekstraksi fitur. Ekstraksi warna menggunakan HSV dan ekstraksi fitur tekstur menggunakan LBP. Setelah mendapatkan masing-masing ekstraksi fitur, masuk ke tahap klasifikasi.

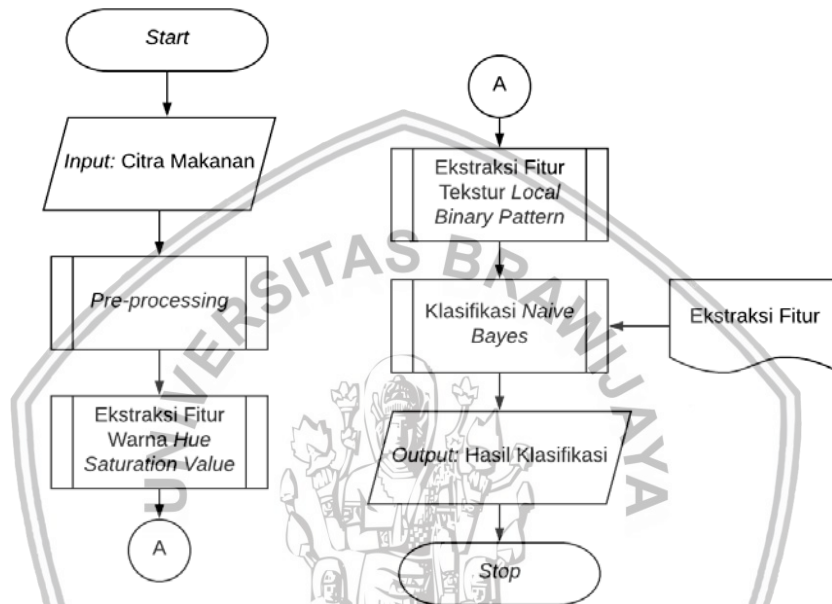
Klasifikasi yang digunakan pada program menggunakan metode *Naïve Bayes*. Hasil keluaran dari program ini akan mendeteksi jenis makanan yang ada di dalam gambar, dengan melabeli masing-masing gambar dengan nama jenis makanan.



Gambar 3.1 Diagram Alir Metode Pelatihan



Pada metode penelitian yang digunakan pada pengujian dapat dilihat pada Gambar 3.2. Tahapan awal yaitu *input* yang digunakan berasal dari citra makanan. Selanjutnya, citra makanan masuk ke dalam tahap *pre-processing*. Tahap selanjutnya yaitu dilakukan ekstraksi fitur warna menggunakan HSV dan ekstraksi fitur tekstur LBP. Setelah mendapatkan masing-masing ekstraksi fitur dari data uji, dilakukan klasifikasi menggunakan *Naïve Bayes* dengan menghitung setiap fitur yang ada pada data latih yang mana sudah dimasukkan ke dalam dokumen berupa excel. Tahapan akhir dari diagram alir yaitu menghasilkan hasil klasifikasi citra makanan.



Gambar 3.2 Diagram Alir Metode Pengujian

3.3 Lokasi Penelitian

Lokasi dalam melakukan penelitian dilakukan di Ruang Grup Riset Computer Vision F9.3 Gedung F Lantai 9, Fakultas Ilmu Komputer, Universitas Brawijaya.

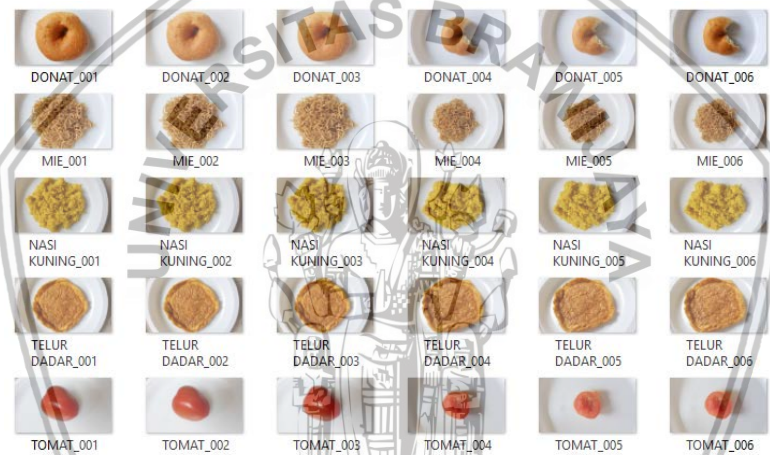
3.4 Metode Pengumpulan Data

Pada tahap pengumpulan data menjelaskan mengenai data yang digunakan dalam penelitian. Data yang digunakan adalah data primer yaitu yang mana pengambilan data di ambil secara manual dengan menggunakan alat berupa *smartphone*. Objek yang digunakan dalam penelitian ini berupa:

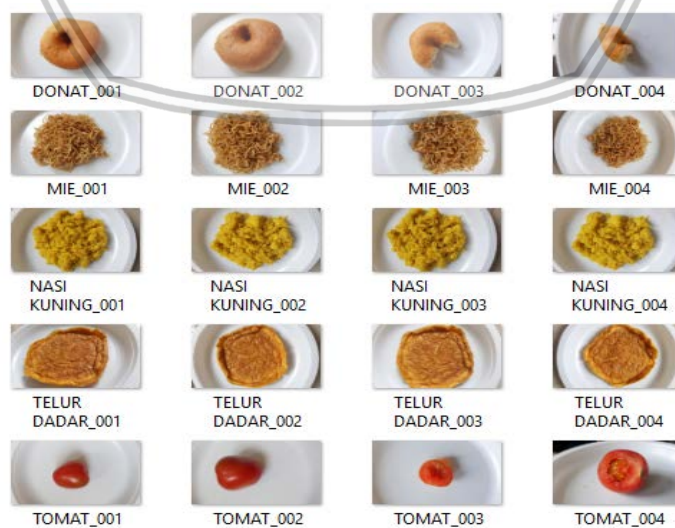
1. Objek yang digunakan pada penelitian ini menggunakan makanan yang berbentuk padat dan hanya terdapat 5 jenis makanan.
2. Intensitas cahaya menggunakan cahaya matahari yang ada pada ruangan Gedung F Lantai 9, Fakultas Ilmu Komputer, Universitas Brawijaya pada pukul 10.00 – 13.00 WIB.
3. Menggunakan *smartphone* betipe Samsung J7 Pro dengan kamera utama 13 megapiksel f/1.9.

4. Kelas yang digunakan dalam klasifikasi jenis makanan berjumlah lima jenis, yaitu donat, mie, nasi kuning, telur dadar, dan tomat.
5. Objek makanan yang mana dalam pengambilan gambar dilakukan secara tegak lurus terhadap objek.
6. Objek makanan donat memiliki 1 bagian, dan $\frac{3}{4}$ bagian.
7. Objek makanan tomat memiliki 1 bagian, dan $\frac{1}{2}$ bagian.
8. Objek makanan mie, nasi kuning, dan telur memiliki 1 bagian yang utuh.

Data yang digunakan selama penelitian ini dapat dilihat pada Gambar 3.3 untuk data latih dan pada Gambar 3.4 untuk data uji. Pada masing-masing kelas yang ada pada data latih, memiliki enam data latih dan memiliki jumlah data yang sebanyak 30 data latih. Sedangkan data yang digunakan pada data uji setiap kelasnya memiliki empat data, sehingga jumlah data uji adalah sebanyak 20 data uji.



Gambar 3.3 Data Latih



Gambar 3.4 Data Uji

3.5 Peralatan Pendukung

Peralatan yang mendukung penelitian ini merupakan kebutuhan yang diperlukan dalam pengembangan sistem. Adapun peralatan dari penelitian ini dapat dilihat sebagai berikut.

1. Spesifikasi *hardware*

Laptop dengan Prosesor Intel® Core™ i3-5005U CPU @ 2.00GHz, Memori 4GB DDR3, Ruang penyimpanan dengan kapasitas 1TB, Ukuran layar 14 inch.

2. Spesifikasi *software*

- Microsoft Windows 10
- Python 3.6



BAB 4 PERANCANGAN

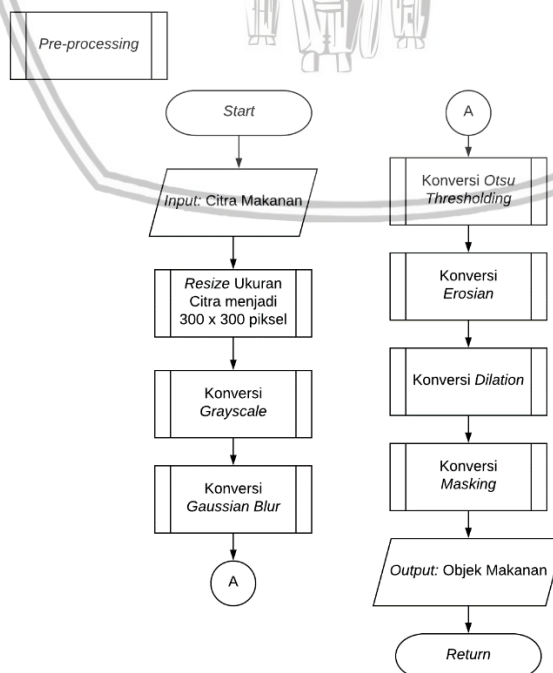
Pada bab 4 berisi mengenai tahapan perancangan dari sistem klasifikasi citra makanan menggunakan metode *Hue Saturation Value* (HSV) dan *Local Binary Pattern* (LBP) sampai menghasilkan hasil klasifikasi menggunakan metode *Naïve Bayes Classifier*.

4.1 Perancangan Algoritme Proses Implementasi

Pada Gambar 3.1 menjelaskan mengenai proses utama yang ada pada sistem klasifikasi citra makanan. Pertama yaitu *pre-processing*, yang mana pada tahap ini citra akan diolah sampai menjadi citra yang siap untuk diolah pada tahapan selanjutnya. Kedua *Hue Saturation Value* (HSV), metode ini digunakan untuk mendapatkan fitur warna dari masing-masing citra hasil *pre-processing*. Ketiga, *Local Binary Pattern* (LBP), metode ini digunakan untuk mendapatkan fitur tekstur dari masing-masing citra hasil *pre-processing*. Terakhir, melakukan klasifikasi menggunakan metode *Naïve Bayes Classifier*.

4.1.1 Implementasi *Pre-Processing*

Pada tahapan *pre-processing*, merupakan tahapan awal pada pengolahan citra yang berfungsi untuk memperbaiki kualitas citra dan memudahkan pada tahapan selanjutnya. *Pre-processing* dilakukan proses *grayscale*, *gaussian blur*, *otsu thresholding*, *dilation*, dan *masking* pada citra data latih dan citra data uji. Masing-masing *pre-processing* memiliki alur sistem yang sama. Alur ditunjukkan pada Gambar 4.1.

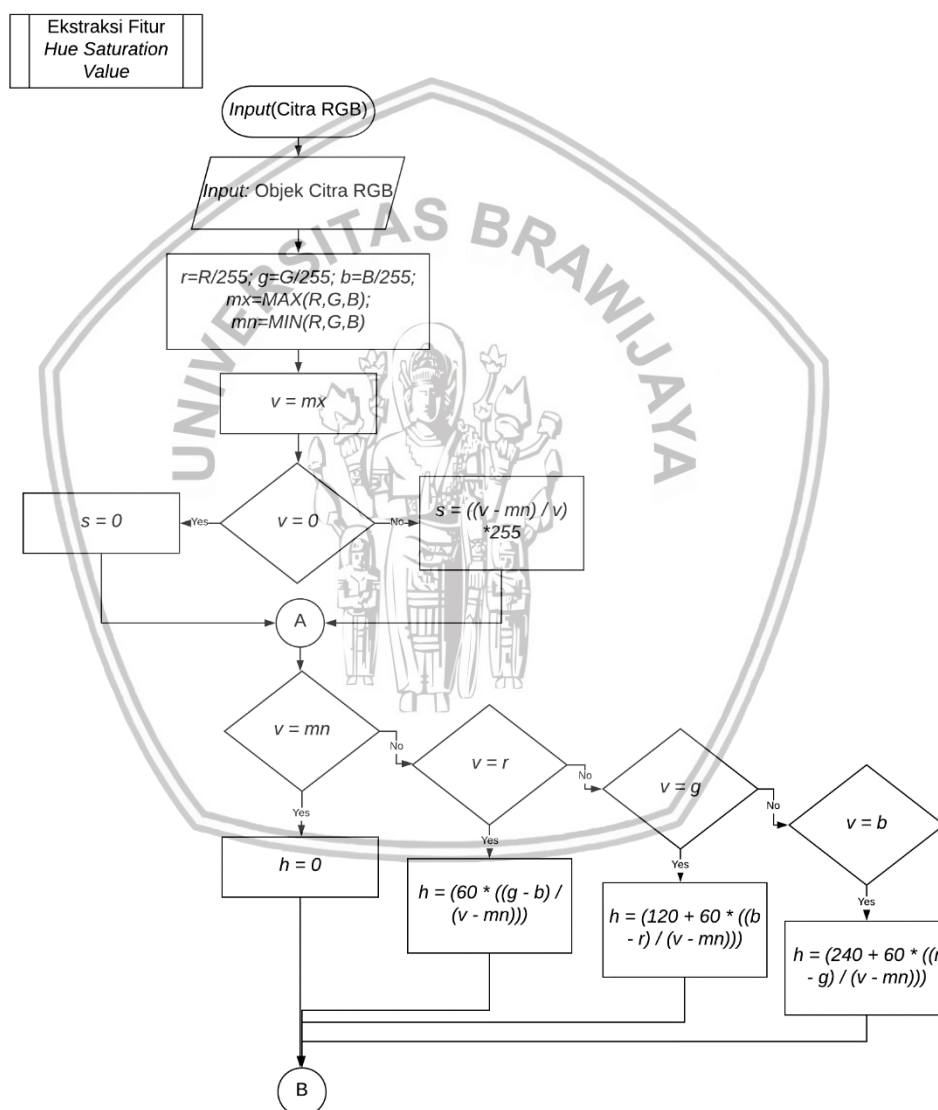


Gambar 4.1 Diagram Alir *Pre-processing*

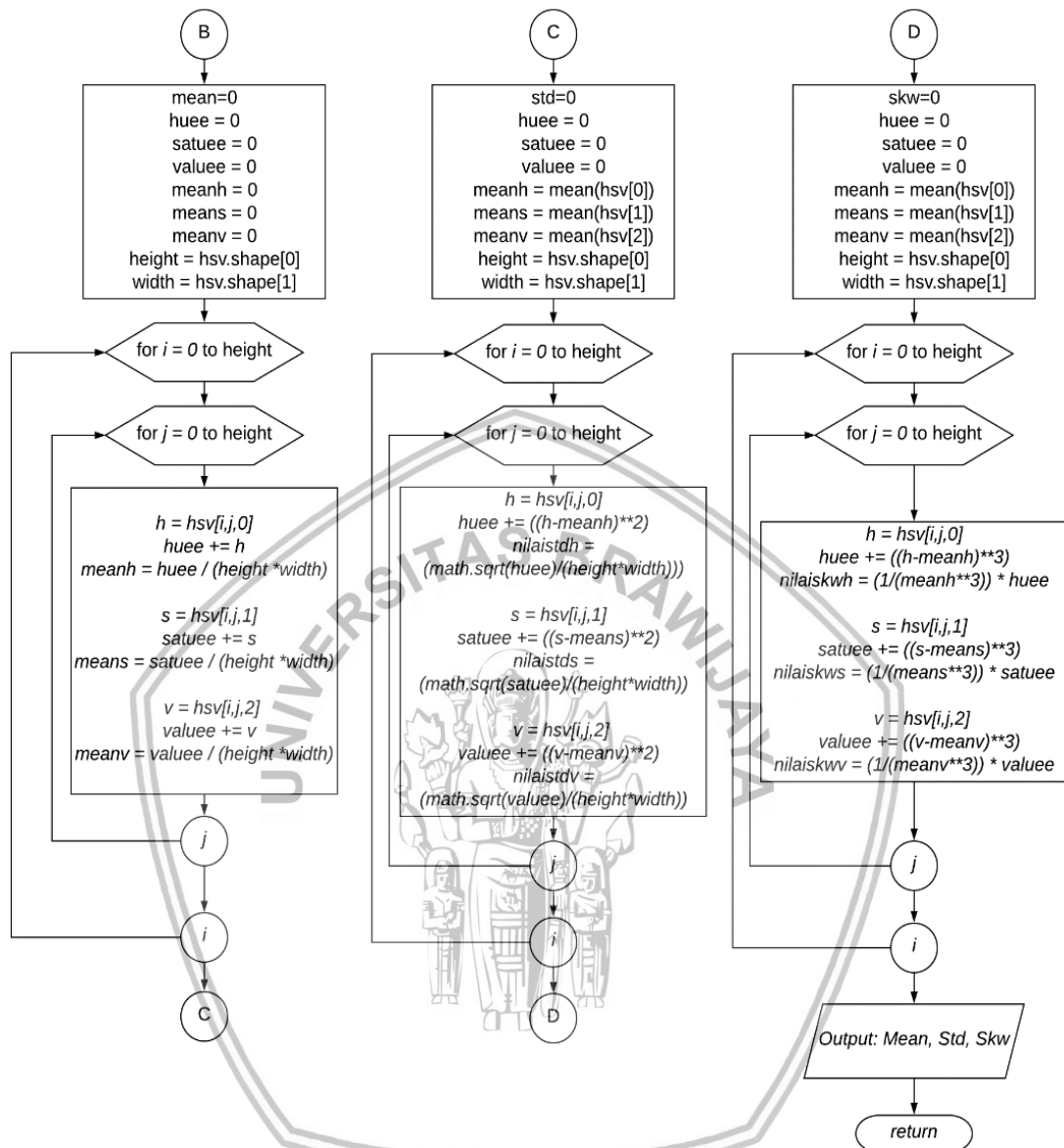


4.1.2 Implementasi Hue Saturation Value (HSV)

Pada tahapan Hue Saturation Value (HSV), merupakan tahapan untuk mendapatkan fitur warna pada citra. Proses ini membutuhkan citra yang telah melalui hasil *pre-processing* dan terdapat nilai Red Green Blue (RGB) dalam citra tersebut. Kemudian Nilai RGB diolah dan dikonversi menjadi Hue Saturation Value (HSV). Setelah mengubah citra ke dalam HSV, setelah itu dilakukan ekstraksi fitur. Ekstraksi fitur menggunakan *mean*, *standard deviation*, dan *skewness*. Alir diagram ekstraksi fitur dapat dilihat pada Gambar 4.2.



Gambar 4.2 Diagram Alir Ekstraksi Fitur HSV

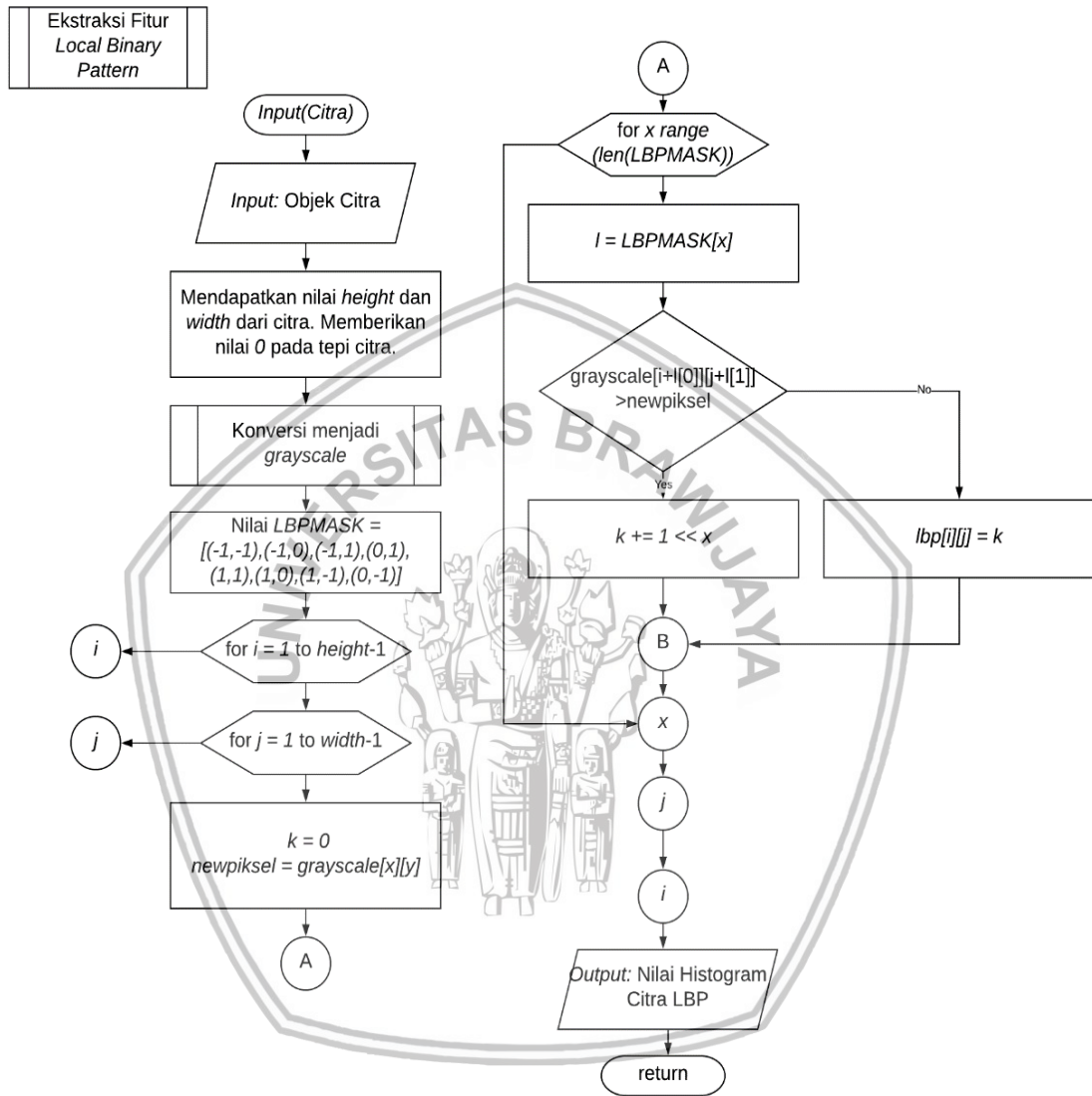


Gambar 4.2 Diagram Alir Ekstraksi Fitur HSV (Lanjutan)

4.1.3 Implementasi *Local Binary Pattern* (LBP)

Pada tahapan *Local Binary Pattern* (LBP), merupakan tahapan untuk mendapatkan fitur tekstur pada citra. Proses ini membutuhkan citra yang telah melalui hasil *pre-processing* kemudian diolah menjadi *grayscale* agar menghasilkan satu *color channel* saja. *Local Binary Pattern* (LBP) merupakan suatu metode yang membandingkan nilai piksel dengan nilai piksel tetangganya. Kemudian dilakukan *thresholding* sehingga menghasilkan bilangan biner, yang diubah menjadi bilangan desimal. Bilangan desimal yang didapat menjadi nilai

piksel yang dipilih. Setelah mendapatkan nilai biner dari piksel terbaru, kemudian nilai-nilai tersebut di letakkan dalam histogram. Alur ditunjukkan pada Gambar 4.4.



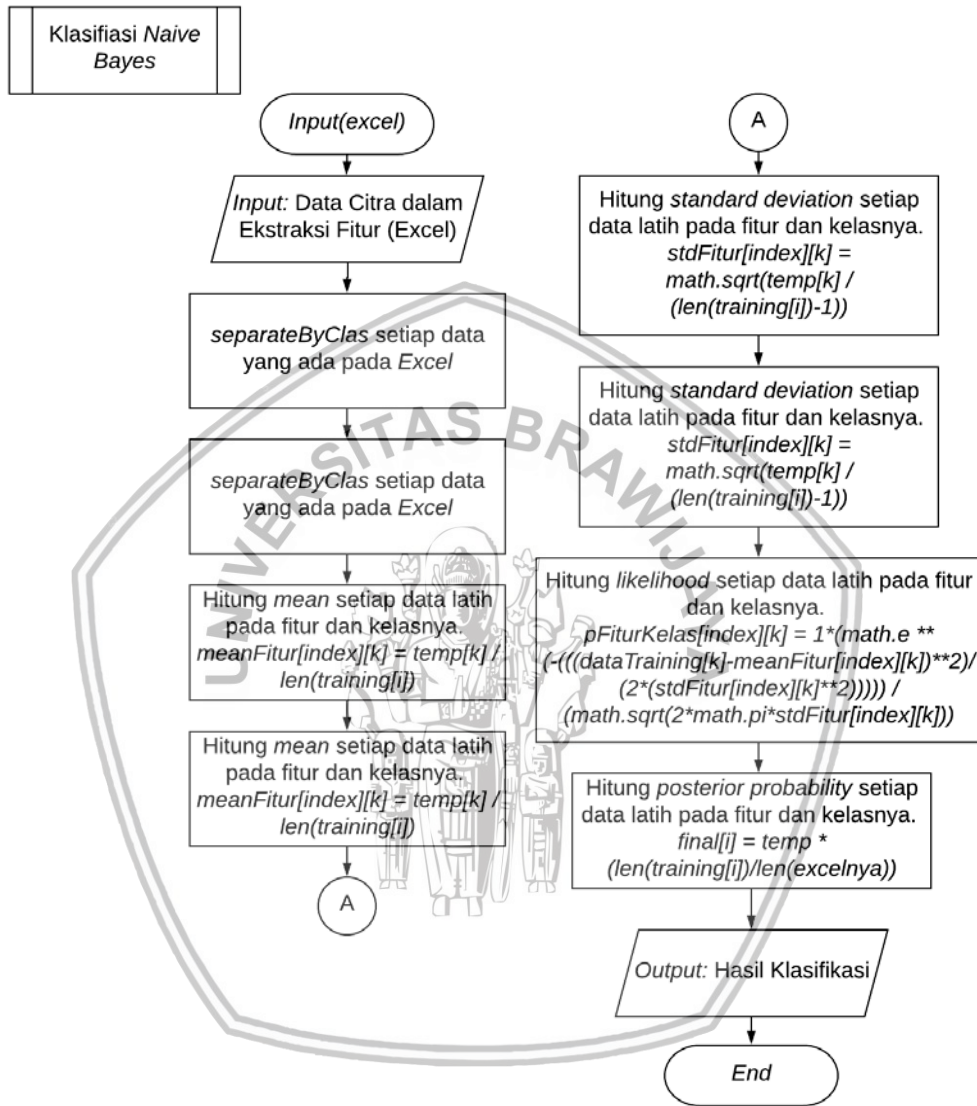
Gambar 4.3 Diagram Alir LBP

4.1.4 Implementasi Naïve Bayes Classifier

Pada tahapan *Naïve Bayes Classifier*, merupakan tahapan yang digunakan untuk mengklasifikasi. Tahapan pertama yang dilakukan yaitu mencari nilai *mean*, *standard deviation* dan *likelihood*, yang mana masing-masing nilai dicari dari setiap data yang ada pada setiap fitur dan ada pada setiap kelas. Setelah mendapatkan masing-masing nilai, kemudian untuk mencari nilai *posterior probability* dapat menggabungkan semua nilai peluang dari *likelihood*, sehingga dapat menghasilkan



nilai yang memiliki peluang tertinggi. Tahapan terakhir yaitu sebuah pengambilan keputusan dalam melakukan klasifikasi pada data uji dapat ditarik sebuah kesimpulan. Diagram alir *Naïve Bayes Classifier* dapat dilihat pada Gambar 4.5.



Gambar 4.4 Diagram Alir Klasifikasi *Naïve Bayes*

4.2 Perhitungan Manualisasi

Perhitungan manualisasi menjelaskan perhitungan yang dimulai dari tahapan *pre-processing* yang terdiri dari HSV, *grayscale*, *gaussian blur*, *otsu thresholding*, *dilation*, dan segmentasi. Kemudian hasil dari *pre-processing* akan diolah menjadi Hue Saturation Value (HSV) dan mencari ekstraksi fitur dari HSV. Kemudian, diolah

menjadi *Local Binary Pattern* (LBP) dan mencari ekstraksi fitur dari LBP. Hasil masing-masing ekstraksi fitur digunakan untuk mengklasifikasi citra.

Data yang digunakan dalam perhitungan manual menggunakan satu sampel dari data latih yang berukuran 300 piksel x 300 piksel. Namun, dalam proses manualisasi digunakan piksel dengan ukuran 3 piksel x 3 piksel, digunakannya citra berukuran 3 piksel x 3 piksel dikarenakan terlalu banyak piksel yang ada pada citra asli. Data citra yang digunakan dalam manualisasi dapat dilihat pada Gambar 4.6. Selain itu, hasil citra dari proses *pre-processing* dapat dilihat pada Gambar 4.7.



Gambar 4.5 Citra Asli



Gambar 4.6 Citra Hasil Segmentasi

4.2.1 Perhitungan *Pre-Processing*

Nilai setiap piksel diletakkan dalam bentuk tabel agar memudahkan dalam membaca nilai setiap piksel. Pada Tabel 4.1 untuk data piksel berupa nilai R, G, dan B dan Tabel 4.2 untuk data piksel berupa nilai H, S, dan V.

Tabel 4.1 Piksel RGB

Piksel					
R		R		R	
G	B	G	B	G	B
R		R		R	
G	B	G	B	G	B
R		R		R	
G	B	G	B	G	B

Tabel 4.2 Piksel HSV

Piksel					
H		H		H	
S	V	S	V	S	V
H		H		H	
S	V	S	V	S	V
H		H		H	
S	V	S	V	S	V

Langkah 1:

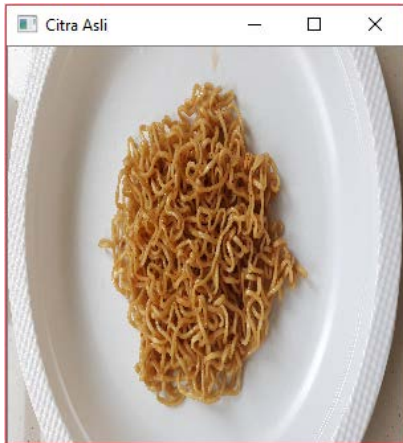
Melakukan konversi citra asli dengan ukuran piksel sebesar 300 piksel x 300 piksel menjadi berukuran 3 piksel x 3 piksel. Nilai piksel dapat dilihat pada Tabel 4.3.

Tabel 4.3 Nilai Piksel Citra Asli

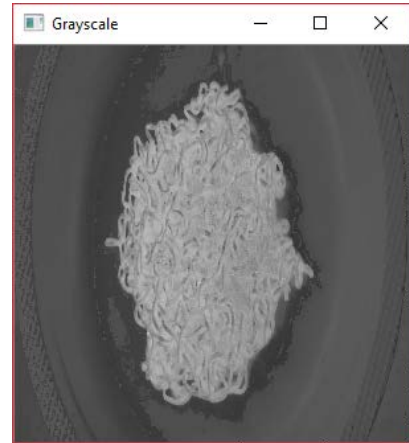
Piksel					
161		167		221	
145	129	107	48	222	227
155		81		228	
145	133	35	14	231	238
168		167		221	
154	138	121	60	224	231

Langkah 2:

Citra asli berukuran 3 piksel x 3 piksel di konversi menjadi *grayscale*. Hasil piksel dapat dilihat pada Tabel 4.4.



Gambar 4.7 Citra Asli



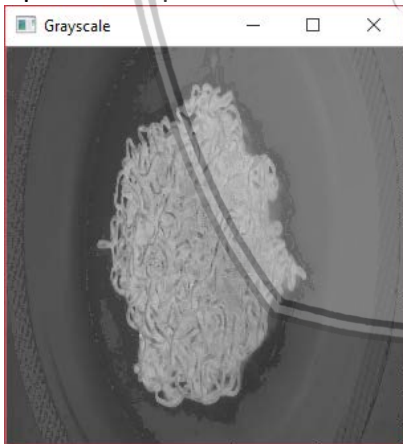
Gambar 4.8 Grayscale

Tabel 4.4 Nilai Grayscale

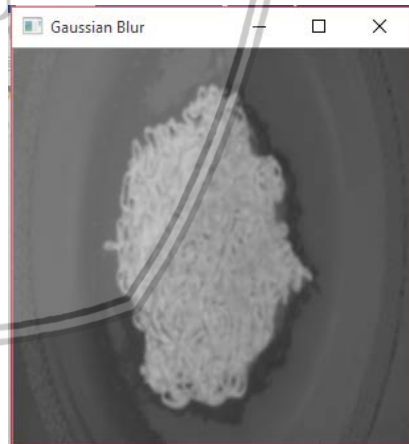
Grayscale		
80	158	85
69	149	90
79	148	88

Langkah 3:

Melakukan konversi citra dari *grayscale* menjadi *gaussian blur*. Hasil piksel dapat dilihat pada Tabel 4.5.



Gambar 4.9 Grayscale



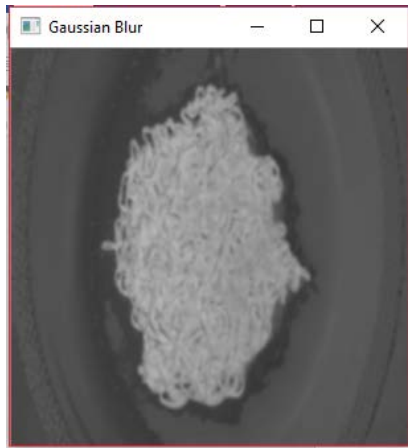
Gambar 4.10 Gaussian Blur

Tabel 4.5 Nilai Gaussian Blur

Gaussian Blur		
115	117	118
114	116	118
114	116	117

Langkah 4:

Melakukan konversi citra dari *gaussian blur* menjadi *otsu thresholding*. Hasil piksel dapat dilihat pada Tabel 4.6.



Gambar 4.10 Gaussian Blur



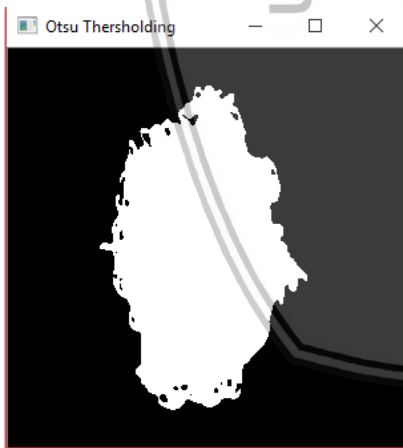
Gambar 4.11 Otsu Thresholding

Tabel 4.6 Nilai Otsu Thresholding

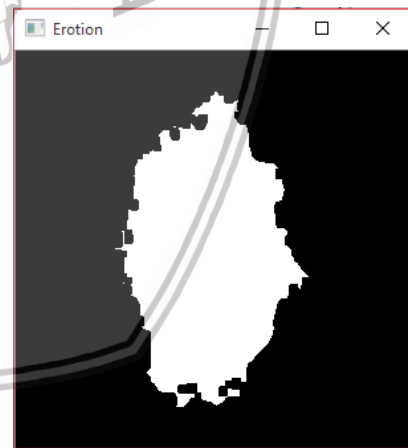
Otsu Thresholding			
0	255	255	
0	255	255	
0	255	255	

Langkah 5:

Melakukan konversi citra dari *otsu thresholding* menjadi *erotion*. Hasil piksel dapat dilihat pada Tabel 4.7.



Gambar 4.11 Otsu Thresholding



Gambar 4.12 Erotion

Tabel 4.7 Nilai Erotion

Erotion		
0	0	0
0	0	0
0	0	0

Langkah 6:

Melakukan konversi citra dari *erotion* menjadi *dilation*. Hasil piksel dapat dilihat pada Tabel 4.8.



Gambar 4.12 Eroton



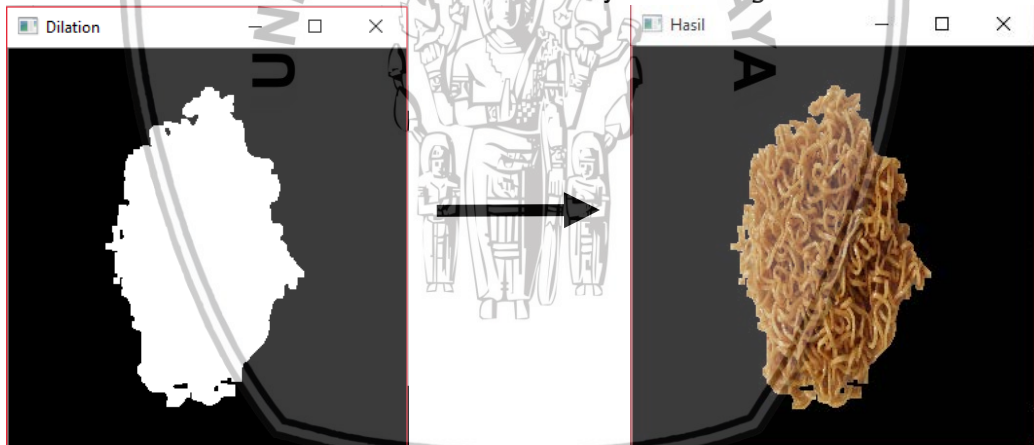
Gambar 4.13 Dilation

Tabel 4.8 Nilai Dilation

Dilation		
0	0	0
0	0	0
0	0	0

Langkah 7:

Melakukan konversi citra dari *dilation* menjadi *masking*.



Gambar 4.13 Dilation

Gambar 4.14 Masking

Sesuai dengan hasil piksel segmentasi, pada Tabel 4.1 menunjukkan format nilai pada masing-masing piksel. Hasil citra segmentasi dapat dilihat pada Tabel 4.9.

Tabel 4.9 Nilai Hasil Masking

Hasil Segmentasi					
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

4.2.2 Perhitungan Hue Saturation Value (HSV).

Langkah 1:

Nilai *Red* (R), *Green* (G), dan *Blue* (B) dari citra berukuran 3 piksel x 3 piksel digunakan untuk mendapatkan nilai dari *Hue* (H), *Saturation* (S), dan *Value* (V). Langkah pertama yang digunakan untuk mendapatkan nilai salah satu dari ketiga *color channel*, yaitu mendapatkan nilai V. Rumus untuk mencari nilai V dapat dilihat pada Persamaan 2.2.

Tabel 4.10 Nilai V

V		
161	167	227
155	81	238
168	167	231

Langkah 2:

Setelah mendapatkan nilai V, kemudian langkah yang dapat dilakukan yaitu mencari nilai S menggunakan rumus pada Persamaan 2.3.

Tabel 4.11 Nilai S

S		
50	182	7
36	211	11
46	163	11

Langkah 3:

Nilai V dan S digunakan untuk mendapatkan nilai H. Rumus untuk mendapatkan nilai H dapat dilihat pada Persamaan 2.4.

Tabel 4.12 Nilai H

H		
15	15	115
16	9	111
16	17	111

Langkah 4:

Langkah selanjutnya yang dilakukan setelah mendapatkan nilai H, S, dan V yaitu melakukan ekstraksi fitur dari fitur warna. Ekstraksi fitur yang digunakan yaitu *Mean*, *Standard Deviation*, dan *Skewness*. Nilai HSV dapat dilihat sebagai berikut. Pada Tabel 4.2 menunjukkan format nilai pada masing-masing piksel. Hasil nilai piksel HSV dapat dilihat pada Tabel 4.13.

Tabel 4.13 Manualisasi HSV

HSV					
15		15		115	
51	161	182	167	7	227
16		9		111	
36	155	211	51	11	238
16		17		11	
46	168	163	167	11	231

Langkah 5:

Ekstraksi fitur pertama yang dilakukan adalah *Mean*. Untuk mendapatkan nilai *mean* dapat dilihat pada Persamaan 2.6. Ekstraksi fitur dilakukan pada masing-masing *color channel*.

Tabel 4.14 Nilai Mean

Mean H	47,22222
Mean S	79,66667
Mean V	177,2222

Langkah 6:

Ekstraksi fitur kedua yang dilakukan adalah *Standard Deviation*. Untuk mendapatkan nilai *Standard Deviation* dapat dilihat pada Persamaan 2.7. Ekstraksi fitur dilakukan pada masing-masing *color channel*.

Tabel 4.15 Nilai Standard Deviation

Standar Deviasi H	46,10321
Standar Deviasi S	76,95309
Standar Deviasi V	46,39631

Langkah 7:

Ekstraksi fitur kedua yang dilakukan adalah *Skewness*. Untuk mendapatkan nilai *Skewness* dapat dilihat pada Persamaan 2.8. Ekstraksi fitur dilakukan pada masing-masing *color channel*.

Tabel 4.16 Skewness

Skewness H	206325,6
Skewness S	912286,4
Skewness V	-135227

4.2.3 Perhitungan *Local Binary Pattern* (LBP)

Langkah 1:

Tentukan jumlah ketetanggaan dan radius dalam menentukan perhitungan yang digunakan dalam manualisasi LBP. Jumlah piksel ketetanggaan dan radius yang digunakan yaitu 8 dan 1. Kemudian, citra hasil proses *pre-processing* diolah menjadi citra berwarna *grayscale*.

Tabel 4.17 Nilai Grayscale

To Grayscale		
80	158	85
69	149	90
79	148	88

Langkah 2:

Pada proses perhitungan dengan menggunakan LBP, nilai tepi dari citra dirubah menjadi nilai 0 agar menghasilkan citra berukuran 3 piksel x 3 piksel.

Piksel				
0	0	0	0	0
0	80	158	85	0
0	69	149	90	0
0	79	148	88	0
0	0	0	0	0

Langkah 3:

Perhitungan dimulai dari piksel yang paling ujung kiri atas dengan ukuran piksel yang digunakan sebesar 3 pikse x 3 piksel agar menghasilkan piksel ketetanggan sebesar 8 dan radius sebesar 1. Setelah mendapatkan piksel yang terpilih, kemudian melakukan *thresholding* dengan ketentuan bahwa jika nilai piksel pusat memiliki nilai yang lebih kecil dari piksel tetangganya, maka nilai piksel tetangga berubah menjadi nilai 1. Jika piksel pusat memiliki nilai yang lebih besar dari piksel tetangganya, maka nilai piksel tetangga berubah menjadi nilai 0. Urutan dalam menghitung nilai piksel tetangga sesuai dengan arah jarum jam dan dimulai dari ujung kiri atas.

Piksel terpilih

0	0	0
0	80	158
0	69	149

Thresholding

0	0	0
0	1	1
0	0	1

Langkah 4:

Piksel yang telah dilakukan *thresholding*, kemudian disusun menjadi bentuk biner sesuai dengan urutan. Nilai yang didapatkan kemudian akan menggantikan nilai pusat dari setiap perputaran setiap piksel.

0	0	0
0		1
0	0	1

1	2	3
8		4
7	6	5

Hasil nilai biner setelah diurutkan.

0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---

Hasil nilai biner yang diubah menjadi nilai desimal.

48



Langkah 5:

Nilai desimal yang telah didapatkan dari masing-masing perhitungan setiap piksel, diletakkan dalam matriks yang baru dengan nama matriks LBP. Nilai desimal diletakkan sesuai dengan koordinat dari setiap perhitungan pada langkah 3. Nilai piksel terbaru dapat dilihat pada Tabel 4.18.

Tabel 4.18 Nilai Manualisasi LBP

0	0	0	0	0
0	48	0	14	0
0	248	128	262	0
0	96	128	386	0
0	0	0	0	0

4.2.4 Perhitungan *Naïve Bayes Classifier*

Langkah 1:

Pada perhitungan *Naïve Bayes Classifier*, setiap data yang digunakan sebagai data latih adalah hasil ekstraksi fitur pada setiap metode yang digunakan. Data latih yang digunakan dapat dilihat pada Tabel 4.19. Setiap fitur dilambangkan *mean* (μ), *standard deviation* (σ), dan *skewness* (*Sk*).

Tabel 4.19 Data Latih

No.	Kelas	μ_H	μ_S	μ_V	σ_H	σ_S	σ_V	<i>SkH</i>	<i>SkS</i>	<i>SkV</i>	LBP 1	...	LBP 8
1	Donat	5,4 42	68,7 27	72,8 78	2125, 504	26804, 152	28271, 322	0,0 02	0,0 02	0,0 02	642 15	...	747 8
...
10	Mie	6,9 17	77,5 34	72,7 28	2479, 345	26842, 167	25325, 073	0,0 08	0,0 01	0,0 01	589 75	...	104 58
...
19	NasiKu ning	9,1 50	97,8 98	72,9 93	3242, 419	33905, 314	25911, 683	0,0 03	0,0 01	0,0 01	600 41	...	944 2
...
28	TelurD adar	6,2 77	84,4 16	73,2 99	2368, 991	31186, 044	27039, 504	0,0 06	0,0 02	0,0 02	618 26	...	930 0
...
37	Tomat	0,7 60	34,3 61	29,1 94	753,2 23	23180, 566	19672, 526	0,1 57	0,0 06	0,0 06	784 27	...	402 6
...

Setiap fitur data yang digunakan, kemudian dicari nilai *mean* setiap fiturnya menggunakan rumus pada Persamaan 2.11. Hasil *mean* setiap fitur dapat dilihat pada Tabel 4.20.



Tabel 4.20 Mean Data Latih

MEAN	μ_H	μ_S	μ_V	σ_H	σ_S	σ_V	SkH	SkS	SkV	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
Donat	3,1 59	40, 501	41, 062	172 7,5 10	222 56, 218	221 84, 716	0,0 06	0,0 06	0,0 06	751 29, 556	227 2,0 00	134 6,0 00	132 0,5 56	170 6,3 33	110 0,8 89	276 7,6 67	435 7,0 00
Mie	4,2 35	49, 994	43, 464	204 1,0 22	236 22, 925	202 40, 290	0,0 14	0,0 04	0,0 05	703 68, 000	324 3,1 11	907 ,77 8	175 2,0 00	205 3,1 11	735 ,11 1	436 0,4 44	658 0,4 44
NasiKuning	8,7 90	93, 134	70, 597	321 7,4 96	336 14, 395	260 33, 992	0,0 02	0,0 01	0,0 02	613 87, 000	444 8,1 11	172 7,7 78	243 0,6 67	286 8,1 11	132 6,8 89	662 3,2 22	918 8,2 22
TelurDadar	6,7 69	89, 972	79, 849	233 8,1 83	308 83, 866	273 41, 693	0,0 02	0,0 01	0,0 01	598 13, 000	459 9,0 00	232 6,7 78	263 5,6 67	333 0,2 22	211 1,3 33	587 5,6 67	930 8,3 33
Tomat	1,0 04	32, 157	29, 918	115 0,8 43	226 48, 407	209 05, 909	0,0 63	0,0 07	0,0 07	795 11, 667	143 1,3 33	980 ,00 0	107 5,1 11	114 3,5 56	612 ,33 3	183 9,2 22	340 6,7 78

Langkah 2:

Setelah mendapatkan nilai *mean* setiap fitur, kemudian mencari nilai *standard deviation* pada setiap fitur menggunakan rumus pada Persamaan 2.12. Nilai *standard deviation* dilambangkan dengan *STD*, dapat dilihat pada Tabel 4.21.

Tabel 4.21 Standard Deviation Data Latih

STD	μ_H	μ_S	μ_V	σ_H	σ_S	σ_V	SkH	SkS	SkV	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
Donat	1,7 19	21, 370	24, 009	298 ,79 3	351 8,1 84	460 8,2 21	0,0 03	0,0 03	0,0 03	825 2,5 12	131 3,5 44	740 ,63 6	696 ,63 9	986 ,46 8	604 ,05 7	156 3,7 34	235 2,7 47
Mie	2,1 27	22, 241	23, 198	346 ,01 1	312 5,7 58	439 8,8 14	0,0 09	0,0 03	0,0 03	910 3,5 68	152 5,8 05	285 ,41 2	869 ,81 2	100 5,5 31	205 ,00 9	213 4,0 80	308 4,9 97
NasiKuning	0,3 13	3,5 11	2,6 97	42, 585	270 ,94 1	530 ,53 3	0,0 01	0,0 00	0,0 00	108 5,3 18	136 ,37 1	196 51, 024	196 ,83 5	281 ,87 0	473 46, 129	501 ,02 7	501 ,91 9
TelurDadar	0,5 94	7,4 59	7,0 99	14, 508	469 ,09 2	218 ,15 6	0,0 02	0,0 00	0,0 01	313 9,2 98	594 ,53 3	101 ,80 8	381 ,91 5	577 ,76 8	128 ,82 7	139 5,2 04	561 ,04 4
Tomat	0,3 08	6,9 23	7,3 06	537 ,16 8	234 4,0 12	289 5,7 66	0,0 64	0,0 01	0,0 01	203 2,3 87	454 ,08 4	256 ,17 9	322 ,08 9	167 ,91 7	124 ,11 5	266 ,24 9	819 ,45 5

Langkah 3:

Nilai *mean* dan *standard deviation* dari setiap fitur yang telah didapatkan, kemudian digunakan untuk mencari nilai *likelihood* setiap fitur data uji pada setiap



fitur pada data latih menggunakan rumus pada Persamaan 2.13. Data uji yang akan dilakukan klasifikasi dapat dilihat pada Tabel 4.22.

Tabel 4.22 Data Uji

	μH	μS	μV	σH	σS	σV	SkH	SkS	SkV	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
Data uji	6,1 31	71, 156	61, 664	237 3,4 95	269 15, 177	235 73, 753	0,0 06	0,0 02	0,0 02	618 72	450 8	132 4	246 5	305 3	999	638 5	939 4

Hasil nilai *likelihood* dapat dilihat pada Tabel 4.23.

Tabel 4.23 Likelihood

STD	μH	μS	μV	σH	σS	σV	SkH	SkS	SkV	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
Donat	0,0 68	0,0 31	0,0 56	0,0 02	0,0 03	0,0 06	7,2 60	3,0 61	3,3 69	0,0 01	0,0 03	0,0 15	0,0 04	0,0 05	0,0 16	0,0 01	0,0 01
Mie	0,1 84	0,0 54	0,0 61	0,0 14	0,0 04	0,0 05	3,0 59	5,1 29	5,0 62	0,0 03	0,0 07	0,0 08	0,0 10	0,0 08	0,0 12	0,0 06	0,0 05
NasiKuning	0,0 00	0,0 00	0,0 01	0,0 00	0,0 00	0,0 00	0,0 00	0,5 86	4,9 84	0,0 11	0,0 31	0,0 00	0,0 28	0,0 19	0,0 00	0,0 16	0,0 16
TelurDadar	0,2 91	0,0 06	0,0 06	0,0 05	0,0 00	0,0 00	0,4 86	4,5 60	3,9 73	0,0 06	0,0 16	0,0 00	0,0 18	0,0 15	0,0 00	0,0 10	0,0 17
Tomat	0,0 00	0,0 00	0,0 00	0,0 01	0,0 02	0,0 05	1,0 60	0,0 03	0,0 08	0,0 00	0,0 00	0,0 10	0,0 00	0,0 00	0,0 00	0,0 00	0,0 00

Langkah 4:

Langkah terakhir yaitu mencari nilai *posterior probability* menggunakan rumus pada Persamaan 2.14. Data uji dihitung sehingga menghasilkan nilai *posterior probability*. Hasil dari klasifikasi pada data uji dapat dilihat pada Tabel 4.24.

Tabel 4.24 Posterior Probability

Donat	5,42473E-31
Mie	9,02868E-27
NasiKuning	2,547E-306
TelurDadar	7,346E-143
Tomat	3,7124E-236

Langkah 5:

Pada Tabel menunjukkan bahwa hasil *posterior probability* pada kelas mie memiliki nilai yang lebih besar dari pada kelas lainnya. Sehingga, data uji termasuk ke dalam kelas mie.

4.2.5 Perhitungan Evaluasi Kinerja Sistem

Berdasarkan hasil klasifikasi data, dilakukan evaluasi terhadap kinerja sistem dalam mengklasifikasi data menggunakan metode *Naïve Bayes Classifier*. Hasil evaluasi kinerja sistem dapat dilihat pada Tabel 4.25. Menghitung nilai evaluasi kinerja sistem pada setiap komponen yang ada pada *macro-averaging* dan *micro-averaging* dapat dilihat pada Persamaan 2.15, Persamaan 2.16, Persamaan 2.17 dan Persamaan 2.18 untuk *macro-averaging* dan Persamaan 2.19, Persamaan 2.20, Persamaan 2.21 dan Persamaan 2.22 untuk *micro-averaging*.

Langkah 1:

Mengelompokkan data sesuai pada setiap kelasnya. Setiap kelas yang terpisah digunakan untuk menghitung nilai *macro-averaging* serta pada kelas yang bernama *micro-averaging* digunakan untuk menghitung *micro-averaging*.

Kelas Donat

Actual Class	Predicted Class	
	Positive	Negative
Positive	3	1
Negative	0	0

Kelas Mie

Actual Class	Predicted Class	
	Positive	Negative
Positive	3	1
Negative	0	0

Kelas Nasi Kuning

Actual Class	Predicted Class	
	Positive	Negative
Positive	3	1
Negative	0	0

Kelas Telur Dadar

Actual Class	Predicted Class	
	Positive	Negative
Positive	2	2
Negative	0	0

Kelas Tomat

Actual Class	Predicted Class	
	Positive	Negative
Positive	2	2
Negative	0	0

Micro-averaging

Actual Class	Predicted Class	
	Positive	Negative
Positive	13	7
Negative	0	0

Langkah 2:

Melakukan perhitungan *macro-averaging*.

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

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

$$Recall\ Donat = \frac{3}{3+1} = 0,75$$

$$Recall\ Mie = \frac{3}{3+1} = 0,75$$

$$\begin{aligned}
 \text{Precision Nasi Kuning} &= \frac{3}{3+0} = 1 & \text{Recall Nasi Kuning} &= \frac{3}{3+1} = 0,75 \\
 \text{Precision Telur Dadar} &= \frac{2}{2+0} = 1 & \text{Recall Telur Dadar} &= \frac{2}{2+2} = 0,5 \\
 \text{Precision Tomat} &= \frac{2}{2+0} = 1 & \text{Recall Tomat} &= \frac{2}{2+2} = 0,5 \\
 \text{Macro Precision} &= \frac{1+1+1+1+1}{5} = 1 \\
 \text{Macro Recall} &= \frac{0,75+0,75+0,75+0,5+0,5}{5} = \frac{3,25}{5} = 0,65 \\
 \text{Macro F - Measure} &= \frac{2 \times 1 \times 0,65}{1 + 0,65} = \frac{1,3}{1,65} = 0,7878 \\
 \text{Macro Akurasi} &= \frac{0,75+0,75+0,75+0,5+0,5}{5} \times 100\% = \frac{3,25}{5} \times 100\% = 65\%
 \end{aligned}$$

Langkah 3:

Melakukan perhitungan *micro-averaging*.

$$\text{Micro Precision} = \frac{13}{13+0} = 1$$

$$\text{Micro Recall} = \frac{13}{13+7} = 0,65$$

$$\text{Micro F - Measure} = \frac{2 \times 1 \times 0,65}{1 + 0,65} = 0,78789$$

$$\text{Macro Akurasi} = \frac{13}{13+7} \times 100\% = \frac{13}{20} \times 100\% = 65\%$$

Langkah 4:

Mendapatkan nilai pada perhitungan *macro-averaging* dan *micro-averaging*.

<i>Micro Averaging</i>				<i>Macro Averaging</i>			
<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>	Akurasi	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>	Akurasi
1	0,65	0,823	65%	1	0,65	0,823	65%

BAB 5 IMPLEMENTASI

5.1 Batasan Implementasi

Batasan implementasi pada penelitian ini diperlukan dalam memberikan batasan dalam melakukan implementasi. Sehingga dapat memperjelas dalam perancangan yang telah dibuat. Batasan implementasi pada penelitian ini digunakan untuk klasifikasi citra makanan menggunakan HSV dan LBP dengan *Naïve Bayes Classifier*.

5.2 Implementasi Algoritme

5.2.1 Algoritme *Pre-processing* Citra

Implementasi algoritme *pre-processing* citra pada penelitian ini bertujuan untuk menghasilkan citra yang telah di segmentasi. Tahapan yang ada pada *pre-processing* ini meliputi tahapan melakukan *resize*, konversi *grayscale*, konversi *gaussian blur*, konversi *Otsu thresholding*, konversi *erotion*, konversi *dilation*, konversi *masking*, sehingga menghasilkan citra yang telah di segmentasi. Tahapan *pre-processing* dapat dilihat pada Algoritme 1.

Algoritme 1: *Pre-processing* citra

```

1 import cv2
2 def preprocessing(image):
3     img = cv2.resize(image, (300,300))
4     imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
5     #Konversi ke Grayscale
6     imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
7     #Konversi ke Gaussian Blur
8     blur = cv2.GaussianBlur(imgGray, (5,5),0)
9     #Konversi ke Otsu Thresholding
10    ret3,th4=cv2.threshold(blur,120,255,cv2.THRESH_BINARY+cv2.T
    HRESH_OTSU)
11    #Konversi ke Erotion
12    kernel = np.ones((5,5), np.uint8)
13    img_erosion = cv2.erode(th4, kernel, iterations=1)
14    #Konversi ke Dilation
15    img_dilation = cv2.dilate(img_erosion, kernel, iterations=1)
16    #Konversi ke Masking
17    imgMask = cv2.bitwise_and(img, img, mask = img_dilation)
18
19    return imgMask

```

Penjelasan:

1. Baris 2 merupakan method bernama *preprocessing*.
2. Baris 3 digunakan untuk *resize* ukuran citra menjadi ukuran 300 piksel x 300 piksel.
3. Baris 4 digunakan untuk konversi citra menjadi HSV.

4. Baris 6 digunakan untuk konversi citra menjadi *grayscale*.
5. Baris 8 digunakan untuk konversi citra menjadi *gaussian blur*.
6. Baris 10 digunakan untuk konversi citra menjadi *otsu thresholding*.
7. Baris 12-13 digunakan untuk konversi citra menjadi *erotion*.
8. Baris 15 digunakan untuk konversi citra menjadi *dilation*.
9. Baris 17 digunakan untuk konversi citra menjadi *masking*.
10. Baris 19 digunakan untuk mengembalikan nilai pada variabel.

5.2.2 Algoritme HSV Citra

Algoritme 2: HSV

```

1 def rgbtohsv(r, g, b):
2     r = r / 255.0
3     g = g / 255.0
4     b = b / 255.0
5     mx = max(r, g, b)
6     mn = min(r, g, b)
7     v = mx
8
9     if v == 0:
10        s = 0
11    else:
12        s = ((v - mn) / v) * 255
13    if v == mn:
14        h = 0
15    elif v == r:
16        h = (60 * ((g - b) / (v - mn)))
17    elif v == g:
18        h = (120 + 60 * ((b - r) / (v - mn)))
19    elif v == b:
20        h = (240 + 60 * ((r - g) / (v - mn)))
21
22    v = v * 255
23    h = h / 2
24    return h, s, v

```

Penjelasan:

1. Baris 1 merupakan method bernama *rgbtohsv*, method ini digunakan untuk merubah citra warna RBB menjadi citra HSV.
2. Baris 2-3 digunakan untuk melakukan normalisasi nilai pada setiap piksel.
3. Baris 5 digunakan untuk mencari nilai maksimal dari citra yang telah di normalisasi.
4. Baris 6 digunakan untuk mencari nilai minimal dari citra yang telah di normalisasi.
5. Baris 7 merupakan nilai dari *value (v)*.
6. Baris 9 merupakan kondisi *if* jika nilai $v = 0$, maka nilai dari *saturation (s)* adalah 0.
7. Baris 10 merupakan kondisi *else* jika nilai $v \neq 0$, maka nilai dari *saturation (s)* dicari menggunakan rumus *saturation*.

8. Baris 13 merupakan kondisi *if* jika nilai $v = \text{minimal}$ dari citra maka nilai dari *hue* (h) adalah 0.
9. Baris 15-16 merupakan kondisi *else* jika nilai $v = r$, maka nilai *hue* (h) dicari menggunakan rumus *hue* sesuai dengan kondisi.
10. Baris 17-18 merupakan kondisi *else* jika nilai $v = g$, maka nilai *hue* (h) dicari menggunakan rumus *hue* sesuai dengan kondisi.
11. Baris 19-20 merupakan kondisi *else* jika nilai $v = b$, maka nilai *hue* (h) dicari menggunakan rumus *hue* sesuai dengan kondisi.
12. Baris 22-23 untuk mengalikan nilai dengan 255 dan h dibagi dengan nilai 2.
13. Baris 24 digunakan untuk mengembalikan nilai pada variabel.

5.2.3 Algoritme Ekstraksi Fitur HSV Citra

Algoritme 3: Ekstraksi Fitur HSV

```

1 #Mean
2 def meanh(h):
3     huee = 0
4     for i in range(0, height):
5         for j in range(0, width):
6             h = hsv3[i,j,0]
7             huee += h
8             meanh = huee / (height *width)
9     return meanh
10
11 def means(s):
12     satuee = 0
13     for i in range(0, height):
14         for j in range(0, width):
15             s = hsv3[i,j,1]
16             satuee += s
17             means = satuee / (height *width)
18     return means
19
20 def meanv(v):
21     valuee = 0
22     for i in range(0, height):
23         for j in range(0, width):
24             v = hsv3[i,j,2]
25             valuee += v
26             meanv = valuee / (height *width)
27     return meanv
28
29 #Standard deviation
30 def stdh(h, meanh):
31     huee = 0
32     for i in range(0, height):
33         for j in range(0, width):
34             h = hsv3[i,j,0]
35             huee += ((h-meanh)**2)
36             nilaistdh = math.sqrt(huee)
37     return nilaistdh
38
39 def stds(s, means):
40     satuee = 0

```

Algoritme 4: Ekstraksi Fitur HSV

```

41     for i in range(0, height):
42         for j in range(0, width):
43             s = hsv3[i,j,1]
44             satuee += ((s-means)**2)
45             nilaistds = math.sqrt(satuee)
46     return nilaistds
47
48 def stdv(v, meanv):
49     valuee = 0
50     for i in range(0, height):
51         for j in range(0, width):
52             v = hsv3[i,j,2]
53             valuee += ((v-meanv)**2)
54             nilaistdv = math.sqrt(valuee)
55     return nilaistdv
56
57 #Skewness
58 def skwh(h, meanh, stdh):
59     huee = 0
60     for i in range(0, height):
61         for j in range(0, width):
62             h = hsv3[i,j,0]
63             huee += ((h-meanh)**3)
64             nilaiskwh = (1/(stdh**3)) * huee
65     return nilaiskwh
66
67 def skws(s, means, stds):
68     satuee = 0
69     for i in range(0, height):
70         for j in range(0, width):
71             s = hsv3[i,j,1]
72             satuee += ((s-means)**3)
73             nilaiskws = (1/(stds**3)) * satuee
74     return nilaiskws
75
76 def skwv(v, meanv, stdv):
77     valuee = 0
78     for i in range(0, height):
79         for j in range(0, width):
80             v = hsv3[i,j,2]
81             valuee += ((v-meanv)**3)
82             nilaiskwv = (1/(stdv**3)) * valuee
83     return nilaiskwv

```

Penjelasan:

1. Baris 2 merupakan method bernama *meanh* memiliki nilai parameter *h*, method ini digunakan untuk mencari nilai *mean* dari citra HSV.
2. Baris 3-8 digunakan untuk mencari nilai *mean* pada piksel *hue* (*h*) dengan melakukan kondisi perulangan *for*.
3. Baris 9 digunakan untuk mengembalikan nilai pada variabel.
4. Baris 11 merupakan method bernama *means* memiliki nilai parameter *s*, method ini digunakan untuk mencari nilai *mean* dari citra HSV.
5. Baris 12-17 digunakan untuk mencari nilai *mean* pada piksel *saturation* (*s*) dengan melakukan kondisi perulangan *for*.

6. Baris 18 digunakan untuk mengembalikan nilai pada variabel.
7. Baris 20 merupakan method bernama *meanv* memiliki nilai parameter *v*, method ini digunakan untuk mencari nilai *mean* dari citra HSV.
8. Baris 21-26 digunakan untuk mencari nilai *mean* pada piksel *value (v)* dengan melakukan kondisi perulangan *for*.
9. Baris 27 digunakan untuk mengembalikan nilai pada variabel.
10. Baris 29 merupakan method bernama *stdh* memiliki nilai parameter *h*, *meanh*, method ini digunakan untuk mencari nilai *standard deviation* dari citra HSV.
11. Baris 31-36 digunakan untuk mencari nilai *standard deviation* pada piksel *hue (h)* dengan melakukan kondisi perulangan *for*.
12. Baris 37 digunakan untuk mengembalikan nilai pada variabel.
13. Baris 39 merupakan method bernama *stds* memiliki nilai parameter *s*, *means*, method ini digunakan untuk mencari nilai *standard deviation* dari citra HSV.
14. Baris 40-45 digunakan untuk mencari nilai *standard deviation* pada piksel *saturation (s)* dengan melakukan kondisi perulangan *for*.
15. Baris 46 digunakan untuk mengembalikan nilai pada variabel.
16. Baris 48 merupakan method bernama *stdh* memiliki nilai parameter *v*, *meanv*, method ini digunakan untuk mencari nilai *standard deviation* dari citra HSV.
17. Baris 49-54 digunakan untuk mencari nilai *standard deviation* pada piksel *value (v)* dengan melakukan kondisi perulangan *for*.
18. Baris 55 digunakan untuk mengembalikan nilai pada variabel.
19. Baris 58 merupakan method bernama *skwh* memiliki nilai parameter *h*, *meanh*, *stdh* method ini digunakan untuk mencari nilai *skewness* dari citra HSV.
20. Baris 59-64 digunakan untuk mencari nilai *skewness* pada piksel *hue (h)* dengan melakukan kondisi perulangan *for*.
21. Baris 65 digunakan untuk mengembalikan nilai pada variabel.
22. Baris 67 merupakan method bernama *skws* memiliki nilai parameter *s*, *means*, *stds* method ini digunakan untuk mencari nilai *skewness* dari citra HSV.
23. Baris 59-64 digunakan untuk mencari nilai *skewness* pada piksel *saturation (s)* dengan melakukan kondisi perulangan *for*.
24. Baris 74 digunakan untuk mengembalikan nilai pada variabel.
25. Baris 76 merupakan method bernama *skwh* memiliki nilai parameter *v*, *meanv*, *stdv* method ini digunakan untuk mencari nilai *skewness* dari citra HSV.
26. Baris 77-82 digunakan untuk mencari nilai *skewness* pada piksel *value (v)* dengan melakukan kondisi perulangan *for*.
27. Baris 83 digunakan untuk mengembalikan nilai pada variabel.

5.2.4 Algoritme LBP Citra

Algoritme 5: LBP

1	Import cv2
2	def lbp(img):
3	height = img.shape[0]
4	width = img.shape[1]
5	lbp = np.zeros((height,width), np.uint8)

Algoritme 6: LBP

```

6
7
8     grayscale = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
9
10    LBPMASK    =    [(-1,-1),(-1,0),(-1,1),(0,1),(1,1),(1,0),(1,-
11), (0,-1)]
11    for i in range(1, height-1):
12        for j in range(1, width-1):
13            k = 0
14            newpiksel = grayscale[i][j]
15            for x in range(len(LBPMASK)):
16                l = LBPMASK[x]
17                if grayscale[i+1[0]][j+1[1]]>newpiksel:
18                    k += 1 << x
19            k = lbp[i][j]
20    hist = plt.hist(lbp.ravel(),8,[0,256]);
21    print ("Histogram : ",hist[0])
22    return hist[0]

```

Penjelasan:

1. Baris 2 merupakan method bernama *lbp* memiliki nilai parameter *img*, method ini digunakan untuk mencari ekstraksi fitur tekstur menggunakan *Local Binary Pattern* dari citra.
2. Baris 3-4 digunakan untuk mencari panjang dan lebar dari citra.
3. Baris 5 digunakan untuk memberikan nilai 0 pada tepian gambar citra.
4. Baris 7 digunakan untuk konversi citra menjadi *grayscale*.
5. Baris 9 digunakan untuk *masking* nilai citra.
6. Baris 10-19 digunakan untuk mengubah nilai piksel ketetanggan yang ada pada nilai piksel citra.
7. Baris 20-21 digunakan untuk menggabungkan nilai piksel yang baru ke dalam bentuk histogram dan menampilkan nilai dalam bentuk histogram.
8. Baris 22 digunakan untuk mengembalikan nilai pada variabel.

5.2.5 Algoritme Naïve Bayes Classifier

Algoritme 7: Naïve Bayes Classifier

```

1  dataTraining = [meanofh, meanofs, meanofv, stdofh, stdofs,
2                  stdofv,          skwofh,          skwofs,
3                  skwofv,lbpo[0],lbpo[1],lbpo[2],lbpo[3],lbpo[4],lbpo[5],lbpo[6],
4                  lbpo[7]]
5
6  excelnya = pd.read_excel("excel/training.xlsx", "Sheet1")
7  def separateByClass(dataset):
8      separated = {}
9      for i in range(len(dataset)):
10         vector = [dataset['Skewness H'][i],dataset['Skewness
11 S'][i],dataset['Skewness V'][i],dataset['LBP 1'][i],dataset['LBP
12 2'][i],dataset['LBP 3'][i],dataset['LBP 4'][i],
13                    dataset['LBP 5'][i],dataset['LBP
14 6'][i],dataset['LBP 7'][i],dataset['LBP 8'][i]]
15         if (dataset['File'][i] not in separated):
16             separated[dataset['File'][i]] = []

```

Algoritme 8: Naïve Bayes Classifier

```

10         separated[dataset['File'][i]].append(vector)
11     return separated
12
13
14 training = separateByClass(excelnya)
15 meanFitur = [[0 for i in range(len(excelnya.columns)-1)] for j
in range(len(training))]
16 index = 0
17 for i in training:
18     temp= [0 for i in range(len(excelnya.columns)-1)]
19     for j in range(len(training[i])):
20         for k in range(len(training[i][j])):
21             temp[k]+=training[i][j][k]
22     for k in range(len(training[i][0])):
23         meanFitur[index][k] = temp[k] / len(training[i])
24     index+=1
25
26 index = 0
27 stdFitur = [[0 for i in range(len(excelnya.columns)-1)] for j in
range(len(training))]
28 for i in training:
29     temp= [0 for i in range(len(excelnya.columns)-1)]
30     for j in range(len(training[i])):
31         for k in range(len(training[i][j])):
32             temp[k]+=(training[i][j][k]-
meanFitur[index][k])**2
33     for k in range(len(training[i][0])):
34         stdFitur[index][k] = math.sqrt(temp[k] / (len(training[i])-
1))
35     index+=1
36
37 pFiturKelas = [[0 for i in range(len(excelnya.columns)-1)] for j
in range(len(training))]
38 index = 0
39 for i in training:
40     for k in range(len(training[i][0])):
41         exAtas = (dataTraining[k]-meanFitur[index][k])**2
42         exBawah = 2*(stdFitur[index][k]**2)
43         exBagi = exAtas/exBawah
44         fBawah = math.sqrt(2*math.pi*stdFitur[index][k])
45         pFiturKelas[index][k] = 1*(math.e **
(-
(exAtas/exBawah))) / fBawah
46     index+=1
47
48 final = {}
49 index=0
50 for i in training:
51     temp = 1
52     for j in range(len(training[i][0])):
53         temp *= pFiturKelas[index][j]
54     final[i] = temp * (len(training[i])/len(excelnya))
55     index+=1
56
57 print("Mean")
58 for i in meanFitur:
59     print(i)
60 print("Standard Deviation")
61 for i in stdFitur:

```

Algoritme 9: <i>Naïve Bayes Classifier</i>	
--	--

62	print(i)
63	print("Likelihood")
64	for i in pFiturKelas:
65	print(i)
66	print("Hasil")
67	for i in final:
68	print(i, final[i])

Penjelasan:

1. Baris 1 merupakan *array* yang menampung isi data uji yang akan dilakukan klasifikasi.
2. Baris 3 digunakan untuk membaca data latih dalam bentuk excel.
3. Baris 4 merupakan method bernama *separatedByClass* yang memiliki parameter *dataset*.
4. Baris 5 merupakan *array* yang digunakan untuk menampung data.
5. Baris 6-7 merupakan kondisi perulangan pada *dataset* yang digunakan untuk memisah setiap fitur data latih.
6. Baris 8-10 kondisi *if* jika *dataset* tidak dapat dipisahkan.
7. Baris 11 digunakan untuk mengembalikan nilai pada variabel.
8. Baris 14-24 digunakan untuk mencari nilai *mean* dari setiap fitur pada setiap kelas pada data latih.
9. Baris 26-35 digunakan untuk mencari nilai *standard deviation* dari setiap fitur pada setiap kelas pada data latih.
10. baris 37-46 digunakan untuk mencari nilai *likelihood* dari setiap fitur pada setiap kelas pada data latih dan data uji, serta nilai *mean* dan *standard deviation* yang telah dihitung sebelumnya.
11. Baris 48-55 digunakan untuk mencari nilai *posterior probability* pada data uji.
12. Baris 57-68 digunakan untuk menampilkan nilai *mean*, *standard deviation*, *likelihood*, dan *posterior probability*.

SS

BAB 6

HASIL PENGUJIAN DAN ANALISIS







Bab ini menjelaskan mengenai hasil pengujian dan analisis terhadap algoritme yang telah diimplementasikan pada bab sebelumnya. Pada pengujian, dilakukan pada masing-masing metode ekstraksi fitur maupun kedua metode yang digunakan yaitu HSV dan LBP. Terdapat lima kelas yang digunakan sebagai data latih dan data uji yaitu donat, mie, nasi kuning, telur dadar dan tomat.

6.1 Pengujian Metode HSV, LBP dan *Naïve Bayes Classifier* Terhadap Akurasi













Pada Tabel 6.1 dapat dilihat perbandingan dari hasil kelas sebenarnya dan kelas prediksi. Hasil pengujian data uji pada Tabel 6.1 menggunakan metode HSV dan LBP. Sebanyak 20 data uji yang digunakan terdapat 7 data yang tidak sesuai dengan kelas sebenarnya. Oleh karena itu, diperlukan sebuah analisis mengenai hasil pengujian sistem.

Metode yang digunakan dalam melakukan pengujian menggunakan *confusion matrix*. Selain itu, untuk membandingkan kinerja sistem maka dilakukan pengujian terhadap setiap metode. Skenario dalam melakukan pengujian terbagi menjadi tiga. Pertama yaitu jika menggunakan metode HSV. Kedua yaitu menggunakan metode LBP. Ketiga yaitu menggunakan metode HSV dan LBP.


Tabel 6.1 Hasil Pengujian Data Uji

No	Data Uji	Kelas Sebenarnya	Kelas Prediksi
1		Donat	Donat
2		Donat	Mie
3		Donat	Donat
4		Donat	Donat
5		Mie	Mie
6		Mie	Mie

Tabel 6.1 Hasil Pengujian Data Uji (Lanjutan)

No	Data Uji	Kelas Sebenarnya	Kelas Prediksi
7		Mie	Mie
8		Mie	Donat
9		Telur Dadar	Donat
10		Telur Dadar	Telur Dadar
11		Telur Dadar	Donat
12		Telur Dadar	Telur Dadar
13		Nasi Kuning	Donat
14		Nasi Kuning	Nasi Kuning
15		Nasi Kuning	Nasi Kuning
16		Nasi Kuning	Nasi Kuning
17		Tomat	Donat
18		Tomat	Tomat

Tabel 6.1 Hasil Pengujian Data Uji (Lanjutan)

No	Data Uji	Kelas Sebenarnya	Kelas Prediksi
19		Tomat	Tomat
20		Tomat	Mie

Pengujian menggunakan *confusion matrix* memiliki dua cara untuk mengetahui kinerja sistem. *Macro-averaging* dan *micro-averaging* adalah metode yang dipakai dalam pengujian. Hasil pengujian jika diubah ke dalam bentuk *confusion matrix* maka dapat dilihat pada Tabel 6.2, Tabel 6.3, Tabel 6.4.

Setiap metode HSV dan LBP yang digunakan, menghasilkan nilai pengujian yang berbeda-beda. Nilai yang digunakan sebagai pertimbangan dalam menganalisis *confusion matrix* yaitu nilai *precision*, *recall*, *f-measure*, dan akurasi.

Tabel 6.2 Hasil Pengujian Metode HSV

<i>Actual Class</i>	<i>Predicted Class</i>				
	Donat	Mie	Nasi Kuning	Telur Dadar	Tomat
Donat	3	1	0	0	0
Mie	1	3	0	0	0
Nasi Kuning	1	0	3	0	0
Telur Dadar	2	0	0	2	0
Tomat	1	1	0	0	2

Tabel 6.3 Hasil Pengujian Metode LBP

<i>Actual Class</i>	<i>Predicted Class</i>				
	Donat	Mie	Nasi Kuning	Telur Dadar	Tomat
Donat	3	1	0	0	0
Mie	0	3	0	0	1
Nasi Kuning	1	0	3	0	0
Telur Dadar	1	0	0	3	0
Tomat	4	0	0	0	0

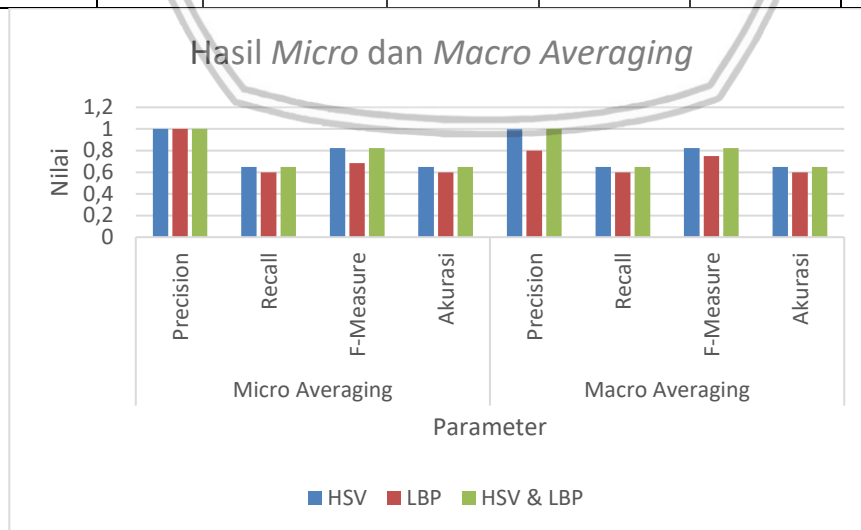
Tabel 6.4 Hasil Pengujian Metode HSV dan LBP

Actual Class	Predicted Class				
	Donat	Mie	Nasi Kuning	Telur Dadar	Tomat
Donat	3	1	0	0	0
Mie	1	3	0	0	0
Nasi Kuning	1	0	3	0	0
Telur Dadar	2	0	0	2	0
Tomat	1	1	0	0	2

Pada Tabel 6.5, menunjukkan hasil *macro-averaging* dan *micro-averaging* pada setiap scenario yang digunakan. Perhitungan pengujian dapat dilihat pada perhitungan evaluasi sistem. Analisis hasil pada perhitungan *macro-averaging* berdasarkan jumlah kelas. Berkaitan dengan hal itu, jika terdapat nilai kelas yang salah maka tidak dapat mengetahui dimana letak kesalahan. Selain itu, jika menggunakan *micro-averaging* perhitungan yang digunakan berdasarkan jumlah seluruh data yang ada pada data latih.

Tabel 6.5 Hasil Micro dan Macro Averaging

Metode	Micro Averaging				Macro Averaging			
	Precision	Recall	F-Measure	Akurasi	Precision	Recall	F-Measure	Akurasi
HSV	1	0,65	0,823	65%	1	0,65	0,823	65%
LBP	1	0,6	0,685	60%	0,8	0,6	0,75	60%
HSV & LBP	1	0,65	0,823	65%	1	0,65	0,823	65%

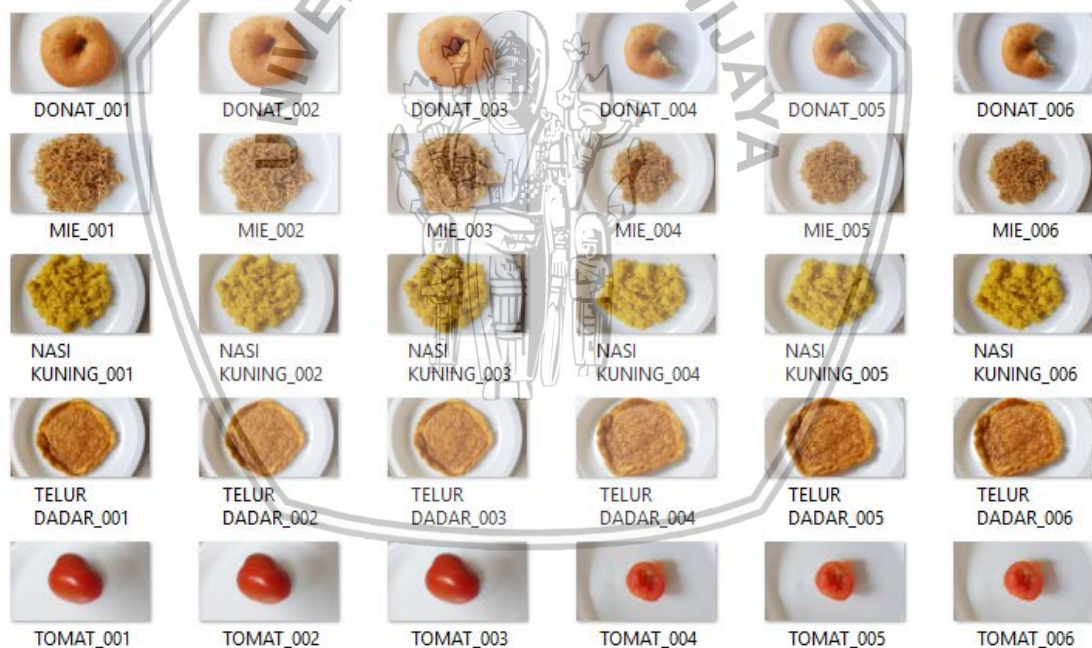


Gambar 6.1 Grafik Hasil Micro dan Macro Averaging

Analisis hasil Tabel 6.5 dengan melihat nilai *precision*, *recall*, *f-measure*, dan akurasi. Metode HSV dan LBP menghasilkan nilai yang sama dengan metode HSV. Hal ini membuktikan bahwa metode HSV yang digunakan dalam klasifikasi citra menghasilkan hasil yang baik. Namun, jika melihat pada Gambar 6.1 terlihat perbedaan pada grafik yang dihasilkan. Perbedaan dilihat pada nilai *precision* metode LBP. Hal ini disebabkan karena *macro-averaging* hanya melihat jumlah kelas yang digunakan.

Nilai akurasi yang dihasilkan, jika menggunakan skenario pertama menghasilkan nilai akurasi 65% pada *macro-averaging* dan *micro-averaging*. Jika menggunakan skenario kedua, menghasilkan nilai akurasi 60% pada *macro-averaging* dan *micro-averaging*. Jika menggunakan skenario ketiga, menghasilkan nilai akurasi 65% pada *macro-averaging* dan *micro-averaging*.

Pada Gambar 6.2 jika dilakukan analisis secara visualisasi, terdapat tiga warna yang tampak oleh mata yaitu warna coklat yang berasal dari kelas donat, mie dan telur dadar, warna merah yang berasal dari kelas tomat, dan warna kuning yang berasal dari kelas nasi kuning. Jika dilihat dari sisi tekstur, setiap citra tampak jelas secara visualisasi memiliki tekstur yang berbeda-beda.



Gambar 6.2 Data Latih

Namun pengujian yang dilakukan dengan *confusion matrix* membuktikan bahwa nilai akurasi dari gabungan fitur warna dan fitur tekstur, serta fitur warna saja menghasilkan nilai yang lebih besar dari pada hasil akurasi yang dihasilkan oleh fitur tekstur. Hasil analisis visualisasi jelas menunjukkan bahwa banyak perbedaan jika dilihat dari segi fitur tekstur. Meskipun kelas donat, mie, dan telur dadar memiliki warna yang hampir serupa, fitur warna HSV masih dapat berkerja dengan baik dalam melakukan klasifikasi citra makanan.

Analisis selanjutnya dapat dilihat pada perhitungan nilai *standard deviation* dari hasil fitur warna dan fitur tekstur. *Standard Deviation* merupakan nilai statistik yang digunakan untuk mengetahui sebaran data. Menghitung nilai *standard deviation* diambil dari sebagian sampel dari populasi untuk mewakili seluruh populasi.

Jika nilai *standard deviation* memiliki nilai perbedaan yang semakin kecil pada setiap fiturnya maka variasi nilai data dianggap sama. Jika nilai *standard deviation* memiliki nilai perbedaan yang semakin besar pada setiap fiturnya maka data yang dimiliki semakin bervariasi. Nilai *standard deviation* dapat dilihat pada Tabel 6.6 dan Tabel 6.7.

Tabel 6.6 Standard Deviation Ekstraksi Fitur HSV

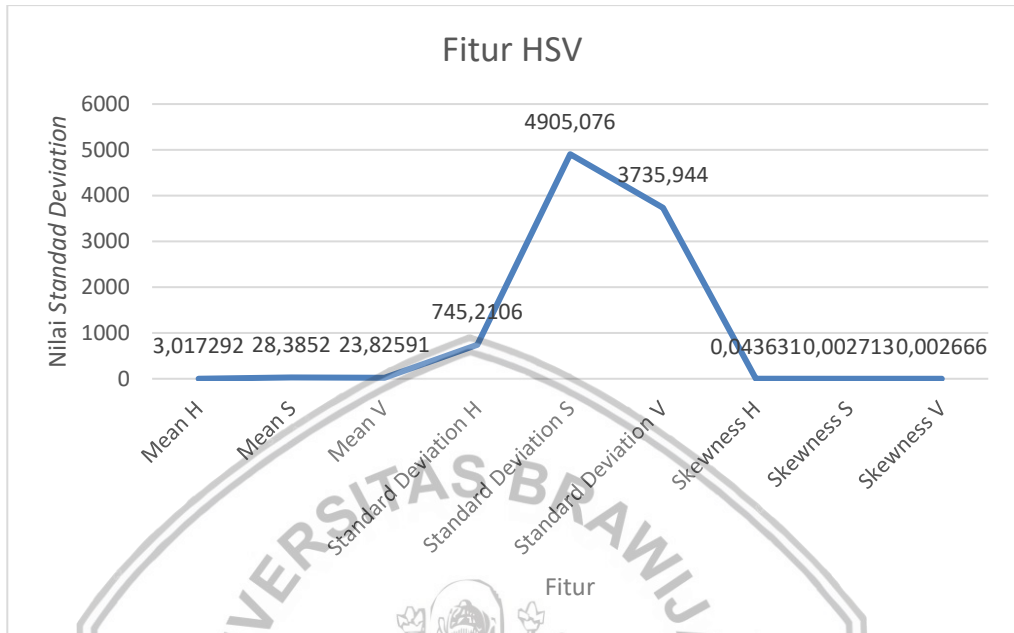
Mean H	3,017292
Mean S	28,3852
Mean V	23,82591
Standard Deviation H	745,2106
Standard Deviation S	4905,076
Standard Deviation V	3735,944
Skewness H	0,043631
Skewness S	0,002713
Skewness V	0,002666

Tabel 6.7 Standard Deviation Ekstraksi Fitur LBP

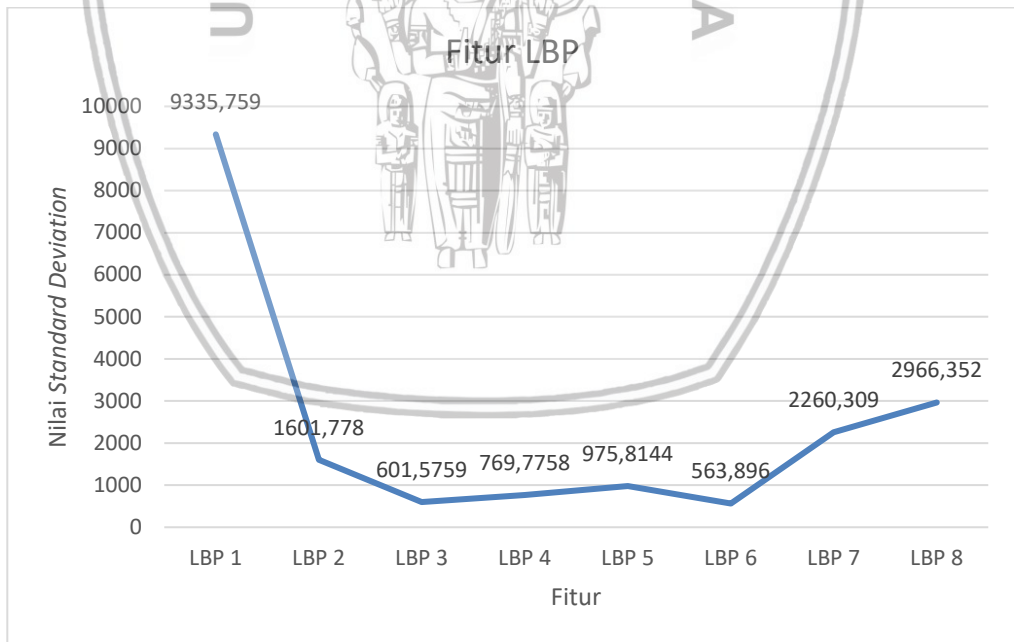
LBP 1	9335,759
LBP 2	1601,778
LBP 3	601,5759
LBP 4	769,7758
LBP 5	975,8144
LBP 6	563,896
LBP 7	2260,309
LBP 8	2966,352

Pada Tabel 6.6 dan Tabel 6.7 didapatkan sebuah analisis sebaran data yang dihasilkan dari nilai *standard deviation*. Pada Tabel 6.6 memiliki nilai sebaran data yang memiliki jarak perbedaan nilai yang besar hal ini dapat dilihat pada Gambar 6.3. Pada gambar ini terlihat jelas bahwa fitur warna memiliki rentan jarak yang

berbeda. Pada Tabel 6.7 memiliki nilai sebaran data memiliki jarak perbedaan yang tidak terlalu besar hal ini dapat dilihat pada Gambar 6.4.



Gambar 6.3 Grafik Nilai Standard Deviation HSV








Gambar 6.4 Grafik Nilai Standard Deviation LBP

Kelemahan pada fitur warna dalam penelitian ini yaitu data yang digunakan memiliki kemiripan warna. Warna yang terlihat secara visualisasi hanya dapat dibedakan sebanyak tiga warna, dapat dilihat pada Tabel 6.8. Perbedaan warna yang tidak bervariasi menyebabkan data latih hanya memiliki sedikit data mengenai warna yang mampu dikenali.



Tabel 6.8 Visualisasi Warna

Warna	Cream - Cokelat	Kuning	Merah Tomat
Gambar	 (Donat)	 (Nasi Kuning)	 (Tomat)
	 (Mie)		
	 (Telur Dadar)		

Kelemahan yang dihasilkan pada fitur tekstur adalah penggunaan data latih yang memiliki bagian yang berbeda-beda. Contohnya dalam kelas donat dan tomat yang memiliki $\frac{3}{4}$ bagian dan $\frac{1}{2}$ bagian. Informasi tekstur dalam data latih tidak memiliki banyak jenis tekstur dalam setiap kelasnya, sehingga fitur tekstur hanya memiliki sedikit informasi. Data yang memiliki bagian yang berbeda-beda dapat berpengaruh pada bentuk, sebaiknya diperlukan fitur untuk mengenali bentuk dari makanan.

Hasil pengujian dan analisis didapatkan bahwa fitur warna dapat menghasilkan nilai akurasi yang lebih baik dari nilai akurasi fitur tekstur. Nilai akurasi yang dihasilkan oleh fitur warna yaitu 65% dan nilai akurasi yang dihasilkan oleh fitur tekstur yaitu 60%.

BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan yang didapat dalam penelitian ini yaitu sebagai berikut.

1. Perancangan dalam menggunakan metode LBP dan HSV dengan *Naïve Bayes Classifier* dapat melakukan klasifikasi citra makanan dapat menghasilkan citra sesuai dengan kelas makanan yang ada. Perancangan yang digunakan terbagi menjadi empat tahapan utama. Pertama, *pre-processing* untuk mendapatkan objek makanan. Kedua, ekstraksi fitur warna HSV yang mana menggunakan fitur *mean, standard deviation, dan skewness*. Ketiga, ekstraksi fitur LBP yang mana menggunakan nilai keabuan pada citra. Terakhir dengan menggunakan semua fitur yang didapat kemudian dilakukan klasifikasi dengan *Naïve Bayes Classifier* untuk menentukan kelas makanan.
2. Metode yang digunakan dalam melakukan klasifikasi citra makanan menggunakan metode LBP dan HSV dengan *Naïve Bayes Classifier* menghasilkan akurasi sebesar 65%. Selain itu, hasil dari pengujian yang telah dilakukan menggunakan metode HSV menghasilkan akurasi sebesar 65% dan metode LBP menghasilkan akurasi sebesar 60%.

7.2 Saran

Saran yang didapatkan dari hasil penelitian ini yaitu sebagai berikut.

1. Diperlukan tambahan jenis makanan agar dapat mengenali lebih banyak jenis makanan.
2. Diperlukan pengembangan dalam melakukan klasifikasi citra makanan, agar dapat mendeteksi makanan yang tidak berbentuk padat.
3. Jika menggunakan fitur warna dalam melakukan penelitian, sebaiknya data yang digunakan memiliki variasi warna.
4. Diperlukan pengenalan bentuk dalam melakukan klasifikasi citra jika memiliki data yang tidak utuh, seperti memiliki $\frac{1}{2}$ bagian dan $\frac{3}{4}$ bagian pada makanan.

DAFTAR REFERENSI

- Amat, R., Sari, J. Y. & Ningrum, I. P., 2017. Implementasi Metode Local Binary Patterns Untuk Pengenalan Pola Huruf Hiragana dan Katakana pada Smartphone. *Jurnal Ilmiah Teknologi Informasi*, pp. 162 - 172.
- Amyarto, N., Sari, Y. A. & Wihandika, R. C., 2018. Pengenalan Emosi Berdasarkan Ekspresi Mikro Menggunakan Metode Local Binary Pattern. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(10), pp. 3230-3238.
- Arivazhagan, S., Shebiah, R. N., Nidhyandhan, S. S. & Ganesan, L., 2010. Fruit Recognition Using Color and Texture Features. *Journal of Emerging Trends in Computing and Information Sciences*, 1(2), pp. 90-94.
- Aygun, S. & Gunes, E. O., 2017. *A Benchmarking: Feature Extraction and Classification of Agricultural Textures Using LBP, GLCM, RBO, Neural Networks, k-NN, and Random Forest*. s.l., s.n.
- Bangare, S. L., Dubal, A., Bangare, P. S. & Patil, S. T., 2015. Reviewing Otsu's Method For Image Thresholding. *International Journal of Applied Engineering Research*, 10(9), pp. 21777-21783.
- Biglari, M., Mirzaei, F. & Neycharan, J. G., 2014. Persian/Arabic handwritten digit recognition using local binary patterns. *International Journal of Digital Information and Wireless Communications (IJDWC)*, Volume 4, pp. 486 - 492.
- Budianita, E. J. & Handayani, L., 2015. Implementasi Pengolahan Citra dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi. *Jurnal Sains, Teknologi dan Industri*, 12(2), pp. 242-247.
- Hafsah, I. S. & Andono, P. N., 2015. Deteksi Otomatis Penyakit Kulit Menggunakan Algoritma Naive Bayes.
- Hapsari, H. & Setiawan, I., 2008. Kajian Model Komunikasi, Informasi, Edukasi (KIE) Ketahanan Pangan Keluarga Miskin di Kabupaten Bandung Propinsi Jawa Barat. *Jurnal Kependudukan Padjadjaran*, 10(1), pp. 12-22.
- Jawas, N. & Suciati, N., 2013. Image Inpainting using Erosion and Dilation Operation. *International Journal of Advanced Science and Technology*, Volume 51, pp. 127-134.
- Junhua, C. & Jing, L., 2012. Research on Color Image Classification Based on HSV Color Space. *Second International Conference on Instrumentation & Measurement, Computer, Communication and Control*.
- Kadir, A., 2016. Kebiasaan Makan dan Gangguan Pola Makan serta Pengaruhnya terhadap Status Gizi Remaja. *Jurnal Publikasi Pendidikan*, 6(1).

- Kalokasari, D. H., Shofi, I. M. & Setyaningrum, A. H., 2017. Implementasi Algoritma Multinomial Naive Bayes Classifier. *Jurnal Teknik Informatika*, 10(2), pp. 109-118.
- Karunia, S. A., Saptono, R. & Anggrainingsih, R., 2017. Online News Classification Using Naive Bayes Classifier with Mutual Information for Feature Selection. *Jurnal Ilmiah Teknologi dan Informasi*, 6(1), pp. 11-15.
- Liantoni, F. & Nugroho, H., 2015. Klasifikasi Daun Herbal Menggunakan Metode Naive Bayes Classifier dan K-Nearest Neighbor. *Jurnal Simatec*, 5(1).
- Miko, A. & Pratiwi, M., 2017. Hubungan Pola Makan Dan Aktivitas Fisik Dengan Kejadian Obesitas Mahasiswa Politeknik Kesehatan Kemenkes Aceh. *Aceh Nutrition Jurnal*, pp. 1-5.
- Munir, R., 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.
- Ningsih, R., 2014. Penyuluhan Hygiene Sanitasi Makanan Dan Minuman, Serta Kualitas Makanan Yang Dijajakan Pedagang Di Lingkungan Sdn Kota Samarinda. *Jurnal Kesehatan Masyarakat*, 10(1), pp. 64-72.
- Ojala, T. et al., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Ieee Transactions On Pattern Analysis And Machine Intelligence*, 24(7), pp. 971 - 987.
- OpenCV, 2018. *Color Conversions*. [Online] Available at: https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html [Accessed 4 Oktober 2018].
- Saleh, A., 2015. Implementasi Metode Klasifikasi Naive Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga. *Citec Journal*, 2(3), pp. 207-217.
- Sari, D. P. & Fadlil, A., 2013. Sistem Identifikasi Citra Jenis Kunyit (Curcuma Domestica Val.) Menggunakan Metode Klasifikasi Minkowski Distance Family. *Jurnal Sarjana Teknik Informatika*, 1(2), pp. 399-408.
- Singh, S. M. & K., H., 2012. Content-Based Image Retrieval using Color Moment and Gabor Texture Feature. *IJCSI International Journal of Computer Science Issues*, 9(5), pp. 299-309.
- Sokolova, M. & Lapalme, G., 2009. A Systematic Analysis of Performance Measures For Classification Tasks. *Information Processing and Management*, 45(4), pp. 427-437.
- Susilo, F. A., Fitriyah, H. & Setyawan, G. E., 2018. Sistem Klasifikasi Kualitas Ikan Tongkol Beku Berdasarkan Fitur Nilai Warna HSV Menggunakan Metode Naive Bayes. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(1), pp. 753-760.

Turiyanto, M. D., Purwanto, D. & Dikairono, R., 2014. Penerapan Teknik Pengenalan Wajah Berbasis Fitur Local Binary Pattern pada Robot Pengantar Makanan. pp. 1- 6.

Wedianto, A., Sari, H. L. & Suzanti, Y., 2016. Analisa Perbandingan Metode Filter Gaussian , Mean Dan Median Terhadap Reduksi Noise. *Jurnal Media Infotama*, 12(1), pp. 21-30.

