

**ALGORITME INFORMATION GAIN FEATURE SELECTION
PADA SISTEM TEMU KEMBALI CITRA MAKANAN
MENGUNAKAN EKSTRAKSI FITUR WARNA DAN TEKSTUR**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Dyva Agna Fauzan
NIM: 155150200111254



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

PENGESAHAN

ALGORITME INFORMATION GAIN FEATURE SELECTION PADA SISTEM TEMU
KEMBALI CITRA MAKANAN MENGGUNAKAN ESKTRAKSI FITUR WARNA DAN
TEKSTUR

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Dyva Agna Fauzan
NIM: 155150200111254

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Maret 2019

Telah diperiksa dan disetujui oleh:

Pembimbing I

Yuita Arum Sari, S.Kom., M.Kom.
NIK: 2016098807152001

Pembimbing II

Drs. Marji, M.T.
NIP: 19670801 199203 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 26 Maret 2019



Dyva Agna Fauzan

NIM: 155150200111254

PRAKATA

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Algoritme Information Gain Feature Selection Pada Sistem Temu Kembali Citra Makanan Menggunakan Esktraksi Fitur Warna Dan Tekstur” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu Yuita Arum Sari, S.Kom., M.Kom. dan Bapak Drs. Marji, M.T. selaku Pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Ibu Dany Primanita Kartikasari, S.T., M.Kom. selaku dosen Penasihat Akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi,
3. Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi dan selama penyelesaian skripsi ini,
4. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini,
5. Seluruh teman-teman yang telah mengisi hari-hari penulis selama menempuh pendidikan perkuliahan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 26 Maret 2019

Penulis

Email: dyvaagna@student.ub.ac.id

ABSTRAK

Dyva Agna Fauzan, Algoritme Information Gain Feature Selection Pada Sistem Temu Kembali Citra Makanan Menggunakan Ekstraksi Fitur Warna dan Tekstur

Pembimbing: Yuita Arum Sari, S.Kom., M.Kom. dan Drs. Marji, M.T.

Nama makanan yang digunakan sebagai kata kunci atau *query* dalam melakukan pencarian resep makanan pada sistem pencarian memiliki keterbatasan, yaitu pengetahuan akan nama makanan yang ingin dicari resepnya. Sehingga dibutuhkan pendekatan lain untuk melakukan pencarian resep yaitu dengan tampilan atau citra dari makanan. Akan tetapi, dengan banyaknya fitur yang dihasilkan dari citra akan menimbulkan data berdimensi tinggi yang berakibat pada efektivitas sistem pencarian. Untuk itu, diperlukan seleksi fitur untuk menangani data berdimensi tinggi. Penelitian ini melakukan penelitian tentang pengaruh jumlah pengembalian yang dapat memberikan nilai MAP tertinggi serta pengaruh metode seleksi fitur *Information Gain* pada sistem temu kembali citra makanan dengan menggunakan ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* dan fitur warna menggunakan *Color Moments* dan *Color Histogram*. Pada jumlah pengembalian (r) sebanyak 5 mendapatkan nilai MAP mengungguli nilai r lain dengan nilai MAP = 1 pada penggunaan fitur warna saja dan tekstur saja dan nilai MAP = 0,98 pada kombinasi keduanya. Hal ini menunjukkan jumlah pengembalian yang lebih sedikit dapat memberikan nilai MAP yang lebih tinggi. Pengaruh algoritme seleksi fitur *Information Gain* pada sistem adalah dapat memberikan nilai MAP = 1 pada jumlah fitur (n) = 10 pada fitur warna, n = 5 pada fitur tekstur, dan n = 30 pada kombinasi fitur. Hal ini menunjukkan sistem dapat memberikan hasil yang sama baiknya (pada warna dan tekstur) bahkan lebih baik (pada kombinasi fitur) jika dibandingkan dengan tanpa seleksi fitur meski dengan jumlah fitur yang lebih sedikit.

Kata kunci: CBIR, *Information Gain*, *Color Moments*, *Color Histogram*, GLCM, MAP

ABSTRACT

The food name used as a keyword or query in conducting a food recipe search on the search system has limitations, namely the knowledge of the name of the food that the recipe wants to find. So another approach is needed to do recipe searches, namely by the display or the image of food. However, with the many features that are generated from the image it will cause high dimensional data which results in the effectiveness of the search system. For this reason, feature selection is needed to handle high-dimensional data. This research conducted a study of the effect of the number of returns that can provide the highest MAP value and the effect of the Information Gain feature selection method on food image retrieval systems for food recipe searches using texture feature extraction using Gray Level Co-occurrence Matrix and color features using Color Moments and Color Histogram. The number of retrieves (r) of 5 is outperforming other r values with the value of $MAP = 1$ on the use of only color features and textures and the value of $MAP = 0.98$ in the combination of both. This indicates a smaller number of returns can give a higher MAP value. The effect of the Information Gain feature selection algorithm on the system is that it can provide the $MAP = 1$ value on the number of features (n) = 10 on the color feature, $n = 5$ on the texture feature, and $n = 30$ on the combination. This shows that the system with feature selection can provide results that are as good (in color and texture) and even better (in combination of features) with fewer features when compared to without feature selection.

Keywords: CBIR, Information Gain, Color Moments, Color Histogram, GLCM, MAP

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xii
DAFTAR KODE SUMBER	xiii
DAFTAR LAMPIRAN	xiv
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 Kajian Pustaka	5
2.2 Dasar Teori	6
2.2.1 <i>Content Based Image Retrieval (CBIR)</i>	6
2.2.2 <i>Grayscale</i>	6
2.2.3 <i>Color Moments</i>	7
2.2.4 <i>Color Histogram</i>	7
2.2.5 <i>Gray Level Cooccurrence Matrix</i>	9
2.2.6 <i>Information Gain Feature Selection</i>	10
2.2.7 <i>Similarity Measure</i>	10
2.2.8 <i>Mean Average Precision</i>	11
BAB 3 METODOLOGI	12
3.1 Teknik Pengumpulan Data	13



3.2 Peralatan Pendukung	14
3.2.1 Perangkat Keras Pendukung	14
3.2.2 Perangkat Lunak Pendukung	14
BAB 4 PERANCANGAN	15
4.1 Perancangan Algoritme	15
4.1.1 Preprocessing	16
4.1.2 Ekstraksi Fitur <i>Color Moments</i>	17
4.1.3 Ekstraksi Fitur <i>Color Histogram</i>	18
4.1.4 Ekstraksi Fitur GLCM	19
4.1.5 Seleksi Fitur <i>Information Gain</i>	21
4.1.6 <i>Similarity Measure</i>	25
4.2 Perhitungan Manual	27
4.2.1 Ekstraksi Fitur Warna <i>Color Moments</i>	27
4.2.2 Ekstraksi Fitur Warna <i>Color Histogram</i>	29
4.2.3 Ekstraksi Fitur Tekstur <i>Gray Level Cooccurrence Matrix</i>	33
4.2.4 Seleksi Fitur <i>Information Gain</i>	42
4.2.5 <i>Similarity Measure</i>	47
4.2.6 Perancangan Skenario Pengujian	50
BAB 5 IMPLEMENTASI	53
5.1 Batasan Implementasi	53
5.2 Implementasi Sistem	53
5.2.1 Implementasi <i>Preprocessing</i>	53
5.2.2 Implementasi Ekstraksi Fitur	54
5.2.3 Implementasi Seleksi Fitur <i>Information Gain</i>	60
5.2.4 <i>Similarity Measure</i>	62
BAB 6 PENGUJIAN DAN ANALISIS	63
6.1 Pengujian dan Analisis Pengaruh Jumlah Pengembalian	64
6.2 Pengujian dan Analisis Pengaruh Seleksi Fitur terhadap Sistem dengan Fitur Warna	65
6.3 Pengujian dan Analisis Pengaruh Seleksi Fitur terhadap Sistem dengan Fitur Tekstur	67
6.4 Pengujian dan Analisis Pengaruh Seleksi Fitur terhadap Sistem dengan Kombinasi Fitur Warna dan Tekstur	70



BAB 7 PENUTUP..... 73
7.1 Kesimpulan.....73
7.2 Saran.....73
DAFTAR REFERENSI..... 74
LAMPIRAN..... 76



DAFTAR TABEL

Tabel 3.1 Data latih	13
Tabel 4.1 Tabel <i>channel R</i>	27
Tabel 4.2 Tabel <i>channel G</i>	28
Tabel 4.3 Tabel <i>channel B</i>	28
Tabel 4.4 Tabel hasil kuantisasi untuk <i>channel R</i>	30
Tabel 4.5 Tabel hasil kuantisasi untuk <i>channel G</i>	30
Tabel 4.6 Tabel hasil kuantisasi untuk <i>channel B</i>	31
Tabel 4.7 Tabel histogram untuk <i>channel R</i>	31
Tabel 4.8 Tabel histogram untuk <i>channel G</i>	31
Tabel 4.9 Tabel histogram untuk <i>channel B</i>	31
Tabel 4.10 Tabel hasil normalisasi histogram untuk <i>channel R</i>	32
Tabel 4.11 Tabel hasil normalisasi histogram untuk <i>channel G</i>	32
Tabel 4.12 Tabel hasil normalisasi histogram untuk <i>channel B</i>	32
Tabel 4.13 Hasil <i>Gamma Correction</i> pada <i>Channel R</i>	33
Tabel 4.14 Hasil <i>Gamma Correction</i> pada <i>Channel G</i>	33
Tabel 4.15 Hasil <i>Gamma Correction</i> pada <i>Channel B</i>	34
Tabel 4.16 Hasil <i>Grayscale</i>	34
Tabel 4.17 <i>Cooccurrence matrix</i> sudut 0°	36
Tabel 4.18 <i>Cooccurrence matrix</i> sudut 45°	37
Tabel 4.19 <i>Cooccurrence matrix</i> sudut 90°	38
Tabel 4.20 <i>Cooccurrence matrix</i> sudut 135°	38
Tabel 4.21 Hasil normalisasi <i>cooccurrence matrix</i> sudut 0°	39
Tabel 4.22 Hasil normalisasi <i>cooccurrence matrix</i> sudut 45°	39
Tabel 4.23 Hasil normalisasi <i>cooccurrence matrix</i> sudut 90°	40
Tabel 4.24 Hasil normalisasi <i>cooccurrence matrix</i> sudut 135°	40
Tabel 4.25 Tabel interval dari atribut A1 dan kategorinya	42
Tabel 4.26 Data latih	43
Tabel 4.27 Data latih yang telah diubah	45
Tabel 4.28 Hasil <i>Information Gain</i> pada data latih	47
Tabel 4.29 Nilai fitur dari data uji.....	47
Tabel 4.30 Hasil <i>Euclidean distance</i>	48
Tabel 4.31 Hasil pengembalian dari sistem	49
Tabel 4.32 Skenario pengujian pengaruh jumlah pengembalian	50
Tabel 4.33 Skenario pengujian pengaruh seleksi fitur terhadap sistem dengan fitur warna.....	51
Tabel 4.34 Skenario pengujian pengaruh seleksi fitur terhadap sistem dengan fitur tekstur	51
Tabel 4.35 Skenario pengujian pengaruh seleksi fitur terhadap sistem dengan kombinasi fitur warna dan tekstur.....	52
Tabel 6.1 Data uji	63



Tabel 6.2 Hasil pengujian pengaruh jumlah pengembalian.....	64
Tabel 6.3 Hasil pengembalian $r = 5$	65
Tabel 6.4 Hasil pengembalian $r = 10$	65
Tabel 6.5 Hasil pengujian pengaruh seleksi fitur terhadap sistem dengan fitur warna.....	66
Tabel 6.6 Hasil pengembalian $n = 10$	66
Tabel 6.7 Hasil pengembalian $n = 39$	67
Tabel 6.8 Hasil pengujian pengaruh seleksi fitur terhadap sistem dengan fitur tekstur	67
Tabel 6.9 Hasil pengembalian $n = 5$ pada <i>query</i> kelas Donat	68
Tabel 6.10 Hasil pengembalian $n = 16$ pada <i>query</i> kelas Donat	69
Tabel 6.11 Hasil pengembalian $n = 5$ pada <i>query</i> kelas Mie Goreng	69
Tabel 6.12 Hasil pengembalian $n = 16$ pada <i>query</i> kelas Mie Goreng	69
Tabel 6.13 Hasil pengembalian $n = 5$ pada <i>query</i> kelas Fried Chicken	69
Tabel 6.14 Hasil pengujian pengaruh seleksi fitur terhadap sistem dengan kombinasi fitur warna dan tekstur.....	70
Tabel 6.15 Hasil pengembalian $n = 55$ pada <i>query</i> kelas Mie Gepeng	71
Tabel 6.16 Hasil pengembalian $n = 10$ pada <i>query</i> kelas Mie Goreng	72
Tabel 6.17 Hasil pengembalian $n = 55$ pada <i>query</i> kelas Fried Chicken	72



DAFTAR GAMBAR

Gambar 2.1 Sistem CBIR.....	6
Gambar 2.2 Citra dengan histogramnya	8
Gambar 2.3 Citra berdistribusi sama dengan ukuran berbeda	8
Gambar 3.1 Diagram alir metode penelitian	12
Gambar 4.1 Diagram alir sistem.....	15
Gambar 4.2 Diagram alir <i>preprocessing</i>	16
Gambar 4.3 Diagram alir ekstraksi fitur <i>Color Moments</i>	17
Gambar 4.4 Diagram alir ekstraksi fitur <i>Color Histogram</i>	18
Gambar 4.5 Diagram alir GLCM	20
Gambar 4.6 Diagram alir seleksi fitur <i>Information Gain</i>	24
Gambar 4.7 Diagram alir <i>similarity measure</i>	26
Gambar 4.8 Citra yang digunakan pada perhitungan manual.....	27
Gambar 4.9 Citra Hasil <i>Grayscaleing</i>	35
Gambar 4.10 Kejadian piksel yang bernilai 144 dengan 145 bertetangga pada sudut 0°	35
Gambar 4.11 Kejadian piksel yang bernilai 144 dengan 145 bertetangga pada sudut 45°	36
Gambar 4.12 Kejadian piksel yang bernilai 144 dengan 145 bertetangga pada sudut 90°	37
Gambar 4.13 Kejadian piksel yang bernilai 144 dengan 145 bertetangga pada sudut 135°	38
Gambar 4.14 Citra data uji	47
Gambar 6.1 Grafik pengaruh nilai r terhadap nilai MAP	64
Gambar 6.2 Kemiripan antar kelas berdasarkan tekstur	68
Gambar 6.3 Hasil pengembalian	71

DAFTAR KODE SUMBER

Kode sumber 5.1 Implementasi fungsi preprocessing()	53
Kode sumber 5.2 Implementasi fungsi mean().....	54
Kode sumber 5.3 Implementasi fungsi stddev()	55
Kode sumber 5.4 Implementasi fungsi skewness().....	55
Kode sumber 5.5 Implementasi fungsi quantization().....	56
Kode sumber 5.6 Implementasi fungsi histogram().....	56
Kode sumber 5.7 Implementasi fungsi normalize_hist()	57
Kode sumber 5.8 Implementasi fungsi h_mean().....	57
Kode sumber 5.9 Implementasi fungsi h_stddev()	57
Kode sumber 5.10 Implementasi fungsi cooccurance().....	58
Kode sumber 5.11 Implementasi fungsi normalize_matrix()	59
Kode sumber 5.12 Implementasi fungsi energy().....	59
Kode sumber 5.13 Implementasi fungsi contrast().....	59
Kode sumber 5.14 Implementasi fungsi homogeneity()	60
Kode sumber 5.15 Implementasi fungsi entropy()	60
Kode sumber 5.16 Implementasi fungsi gain()	61
Kode sumber 5.17 Implementasi fungsi similarity_measure().....	62

DAFTAR LAMPIRAN

LAMPIRAN A CONTOH DATASET	76
LAMPIRAN B NILAI GAIN	77



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Makanan yang nikmat dapat dibuat dengan mengikuti resep yang tepat. Dengan melimpahnya resep makanan yang ada, akan menyulitkan dalam pencarian resep yang tepat sesuai dengan yang diinginkan. Pencarian secara konvensional seperti membuka buku resep maupun bertanya kepada orang lain, akan memakan waktu yang tidak sebentar dan belum tentu resep yang ada sesuai dengan yang diinginkan. Maka dari itu, sebuah sistem yang dapat melakukan pencarian resep makanan yang dapat mengatasi permasalahan-permasalahan tersebut akan sangat membantu.

Sistem temu kembali informasi atau *information retrieval* adalah sistem yang dapat melakukan pencarian informasi berdasarkan *query* yang diberikan oleh pengguna. *Query* yang diberikan dapat berupa teks, musik, atau citra. Untuk melakukan pencarian resep makanan, pada umumnya nama makanan dijadikan sebagai kata kunci atau *query*. Namun, dalam pencarian resep makanan menggunakan nama makanan terdapat keterbatasan pengetahuan akan nama makanan yang ingin dicari resepnya sehingga dibutuhkan pendekatan lain untuk melakukan pencarian resep. Tampilan merupakan ciri khas suatu makanan sehingga kita dapat menggunakan citra dari makanan yang ingin kita cari resepnya sebagai *query* untuk selanjutnya dilakukan *information retrieval*.

Information retrieval yang memiliki *query* berupa citra dapat disebut juga dengan istilah *Content Based Information Retrieval* (CBIR). CBIR telah banyak diaplikasikan pada berbagai bidang, seperti proses diagnosa penyakit dalam bidang medis dan deteksi sidik jari (Choras, 2007). CBIR bekerja dengan cara mengekstraksi fitur yang terdapat pada citra lalu mengembalikan informasi yang sesuai dengan *query*. CBIR menggunakan warna, bentuk, dan tekstur sebagai fitur yang membedakan antara satu gambar dengan yang lainnya (Goel, et al., 2017).

Fitur warna pada citra direpresentasikan dengan *color model* yang menyimpan informasi warna. *Color model* yang umum digunakan diantaranya adalah RGB (*red, green, blue*), HSV (*hue, saturation, value*) dan $YCbCr$ (*luminance* dan *chrominance*). Distribusi warna yang terdapat pada citra yang direpresentasikan dengan *Color Histogram* dan *Color Moments* telah terbukti efisien dan efektif untuk diaplikasikan pada CBIR. Penelitian Singh & Hemachandran (2012) membuktikan bahwa kombinasi *Color Histogram* dan *Color Moments* dalam CBIR untuk citra makanan memiliki *precision* lebih tinggi dibandingkan dengan hanya menggunakan *Color Histogram* atau *Color Moments* saja.

Retrieval yang hanya menggunakan fitur tunggal dapat menghasilkan hasil yang tidak efisien (Singh & Hemachandran, 2012). Selain warna, tekstur menjadi salah satu fitur yang penting dalam menghitung kesamaan antar citra. *Gray Level Cooccurrence Matrix* (GLCM) dan *Gabor filter* adalah contoh algoritme untuk

melakukan ekstraksi fitur tekstur dari citra. Singh & Hemachandran (2012) melakukan *image retrieval* dengan menggunakan ekstraksi fitur *Gabor filter* mendapatkan *average precision* sebesar 43,6%. Puviarasan, et al. (2014) melakukan *image retrieval* menggunakan GLCM dan mendapatkan *average precision* yang lebih baik, yaitu sebesar 67%. Sehingga kombinasi antara kombinasi antara ekstraksi fitur *Color Moments* dan *Color Histogram* dengan GLCM mampu memberikan hasil yang lebih baik. Hal ini dibuktikan oleh Mehdi, et al. (2017) yang mana dilakukan *image retrieval* dengan kombinasi antara ekstraksi fitur *Color Moments*, *Color Histogram*, dan GLCM dan mendapatkan *average precision* sebesar 84,3%.

Dengan banyaknya fitur yang didapatkan dari proses ekstraksi fitur, maka akan menimbulkan data yang memiliki dimensi yang tinggi. Efektivitas suatu sistem *retrieval* dipengaruhi oleh jumlah fitur yang dikomputasi (Benkrama, et al., 2014). Seleksi fitur merupakan suatu metode yang dapat mengurangi dimensi dengan menyingkirkan fitur yang tidak relevan. Metode seleksi fitur yang umum digunakan adalah algoritme *Relief*, *Chi-squared*, dan *Information Gain*. Berdasarkan penelitian oleh Smita Chormunge dan Sudarson Jena, algoritme *Information Gain* memiliki performa yang lebih baik terhadap data yang berdimensi tinggi dibandingkan algoritme seleksi fitur *Relief* dan *Chi-squared*. Penelitian Baranidharan & Ghosh (2011) membuktikan bahwa seleksi fitur menggunakan *Information Gain* mendapatkan hasil akurasi yang lebih tinggi dibandingkan tanpa menggunakan seleksi fitur.

Berdasarkan permasalahan dan penelitian yang telah dikemukakan sebelumnya, maka pada penelitian ini diusulkan penelitian pengaruh metode seleksi fitur *Information Gain* pada sistem temu kembali citra makanan dengan menggunakan ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* dan fitur warna menggunakan *Color Moments* dan *Color Histogram*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, dirumuskan masalah sebagai berikut:

1. Bagaimana pengaruh dari banyaknya hasil yang dikembalikan oleh sistem pada sistem temu kembali citra makanan?
2. Bagaimana pengaruh dari penerapan metode *Information Gain* pada sistem temu kembali citra makanan berdasarkan fitur warna?
3. Bagaimana pengaruh dari penerapan metode *Information Gain* pada sistem temu kembali citra makanan berdasarkan fitur tekstur?
4. Bagaimana pengaruh dari penerapan metode *Information Gain* pada sistem temu kembali citra makanan berdasarkan fitur warna dan tekstur?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Mengetahui pengaruh dari banyaknya hasil yang dikembalikan oleh sistem pada sistem temu kembali citra makanan.

2. Mengetahui pengaruh dari penerapan metode *Information Gain* pada sistem temu kembali citra makanan dengan menggunakan ekstraksi fitur warna.
3. Mengetahui pengaruh dari penerapan metode *Information Gain* pada sistem temu kembali citra makanan dengan menggunakan ekstraksi fitur tekstur.
4. Mengetahui pengaruh dari penerapan metode *Information Gain* pada sistem temu kembali citra makanan dengan menggunakan ekstraksi fitur warna dan tekstur.

1.4 Manfaat

Manfaat yang ingin dicapai pada penelitian ini adalah dapat memudahkan pencarian suatu makanan dengan *query* berupa citra.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Bahasa pemrograman yang digunakan adalah Python 3.6
2. Dataset yang digunakan pada penelitian ini terdiri dari delapan kelas, antara lain: Donat, Mie Goreng, Mie Gepeng, Telor Ceplok, Telor Dadar, Fried Chicken, Rendang, dan Nasi Kuning.

1.6 Sistematika Pembahasan

Struktur dari skripsi ini terdiri dari enam bab dengan sistematika pembahasan sebagai berikut:

BAB 1 Pendahuluan

Bab ini membahas tentang permasalahan yang ada, penelitian-penelitian sebelumnya yang terkait dengan permasalahan, serta solusi untuk menyelesaikan permasalahan tersebut. Bab ini akan dibagi menjadi beberapa subbab yaitu latar belakang, perumusan masalah, tujuan dari penelitian, manfaat yang akan didapat dari penelitian ini, batasan permasalahan, dan sistematika pembahasan.

BAB 2 Landasan Kepustakaan

Bab ini membahas tentang penelitian-penelitian sebelumnya yang memiliki keterkaitan dengan penelitian ini dan teori-teori yang mendukung penelitian ini.

BAB 3 Metodologi Penelitian

Bab ini membahas tentang langkah-langkah yang akan dilakukan dalam pelaksanaan penelitian ini. Bab ini terdapat penjelasan mengenai metodologi, teknik pengumpulan data, dan peralatan pendukung pada penelitian ini.

BAB 4 Perancangan

Bab ini terdiri dari perancangan algoritme dan perhitungan manual membahas tentang algoritme-algoritme yang diajukan sebagai solusi atas permasalahan yang telah dikemukakan.

BAB 5 Implementasi

Bab ini terdiri dari batasan implementasi dan implementasi serta pembahasan kode sumber dari sistem berdasarkan perancangan sistem yang telah dibuat.

BAB 6 Pengujian dan Analisis

Bab ini terdiri dari pengujian yang dilakukan untuk menjawab permasalahan dari penelitian beserta analisis dari hasil pengujian yang didapatkan yang digunakan untuk penarikan kesimpulan.

BAB 7 Penutup

Bab ini terdiri dari kesimpulan yang ditarik berdasarkan hasil pengujian dan analisis yang dilakukan serta saran yang diberikan oleh penulis sebagai bahan perbaikan untuk penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Pada Bab 2 terdiri dari subbab-subbab yang berisikan penjelasan kajian pustaka, dasar teori, konsep, dan metode dari literatur penelitian lainnya yang memiliki keterkaitan dengan penelitian ini.

2.1 Kajian Pustaka

Pada subbab ini terdapat penelitian-penelitian sebelumnya yang memiliki keterkaitan metode dengan penelitian ini. Penelitian pertama adalah penelitian Singh & Hemachandran (2012) yang menggunakan kombinasi antara fitur *Color Histogram* dan *Color Moment* untuk melakukan temu kembali citra. Objek yang digunakan pada penelitian ini adalah citra dari *Corel Image Database*. Dari metode yang digunakan, didapatkan hasil rata-rata *precision* sebesar 68% untuk citra makanan.

Penelitian selanjutnya oleh Anusha, et al., (2014) yang melakukan temu kembali citra dengan menggunakan gabungan metode *color moments* untuk ekstraksi fitur warna dan GLCM untuk ekstraksi fitur tekstur. Selanjutnya dihitung kesamaan antara citra *query* dengan citra pada basis data dengan menggunakan *euclidean distance*. Objek yang digunakan terdiri dari bermacam-macam warna dan tekstur. Dengan metode yang diajukan, penelitian ini memiliki *precision* sebesar 98,4%.

Penelitian selanjutnya oleh El Mehdi, et al., (2017) yang mengombinasikan *Color Moments*, *Color Histogram*, dan *Gray Level Cooccurrence Matrix* sebagai metode yang digunakan. Objek yang digunakan pada penelitian ini adalah Wang Database yang berisikan citra yang dipilih dari Corel Database. Hasil yang didapatkan memiliki *precision* sebesar 85% untuk citra makanan dan memiliki rata-rata sebesar 84% untuk citra secara keseluruhan. Penelitian ini menunjukkan bahwa kombinasi metode tersebut efektif untuk melakukan temu kembali citra.

Chormunge & Jena (2016) membandingkan algoritme *Information Gain*, *Relief*, dan *Chi-squared* sebagai seleksi fitur untuk data berdimensi tinggi. Objek yang digunakan pada penelitian ini adalah *dataset* Colon Cancer yang memiliki 2000 fitur, SRBCT yang memiliki 2308 fitur, DBWorld_bodies_Stemmed yang memiliki 3722 fitur, Lymphoma yang memiliki 4026 fitur, DBWorld_bodies yang memiliki 4703 fitur, Leukemia yang memiliki 7129 fitur, dan CNS yang memiliki 7129 fitur. Masing-masing dataset tersebut diambil sampelnya untuk digunakan pada penelitian tersebut. Hasilnya, seleksi fitur *Information Gain* memiliki performa yang lebih baik dibandingkan algoritme *Relief* dan *Chi-squared*.

Penelitian selanjutnya adalah penelitian oleh Aini, et al., (2018) yang menggunakan metode seleksi fitur *Information Gain* dengan kombinasi KNN dan *Naïve Bayes* untuk melakukan klasifikasi penyakit jantung. Objek yang digunakan pada penelitian ini adalah *dataset statlog* penyakit jantung dari UCI *Machine Learning Repository*. Hasil dari penelitian tersebut menunjukkan bahwa akurasi dengan menggunakan algoritme *Information Gain* memiliki akurasi yang lebih

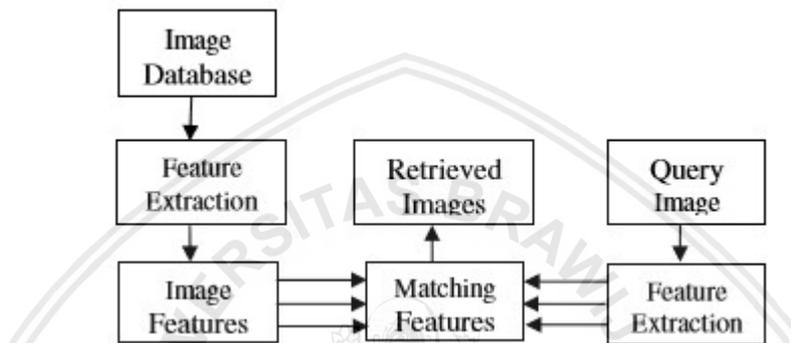
tinggi dibandingkan dengan tidak menggunakan seleksi fitur. Tingkat akurasi tertinggi yang didapat dengan menggunakan seleksi fitur sebesar 92,31%.

2.2 Dasar Teori

Pada subbab ini dijelaskan teori-teori, konsep, dan metode yang mendukung penelitian ini.

2.2.1 Content Based Image Retrieval (CBIR)

CBIR adalah sebuah teknik yang menggunakan konten visual seperti warna, tekstur, dan bentuk untuk mencari informasi yang diinginkan dari basis data citra (Long, et al., 2003). Contoh dari sistem CBIR terdapat pada Gambar 2.1.



Gambar 2.1 Sistem CBIR

Sumber: (Singh & Hemachandran, 2012)

2.2.2 Grayscale

Citra *grayscale* adalah citra yang hanya terdiri dari tingkat warna abu-abu. Citra *grayscale* terdiri dari 8 bit integer yang dapat memberikan 256 kemungkinan warna abu-abu, dari putih dengan nilai 255 sampai hitam dengan nilai 0. Citra *grayscale* sering digunakan karena dapat menyederhanakan algoritme, mengurangi kebutuhan komputasional, dan menyingkirkan informasi warna yang tidak penting (Kanan & Cottrell, 2012). Citra dengan *channel* RGB diubah menjadi *grayscale* dengan menggunakan Persamaan 2.1.

$$grayscale = \frac{1}{3}(R' + G' + B') \quad (2.1)$$

yang mana R' , G' , dan B' adalah *channel* R, G, dan B yang telah melalui proses *gamma correction* dengan menggunakan Persamaan 2.2.

$$x' = 255 \times \frac{x}{255}^{\frac{1}{2.2}} \quad (2.2)$$

Keterangan:

- R' = nilai citra *channel* R yang telah diproses *gamma correction*
- G' = nilai citra *channel* G yang telah diproses *gamma correction*
- B' = nilai citra *channel* B yang telah diproses *gamma correction*
- x = nilai piksel
- x' = nilai piksel yang telah diproses *gamma correction*

2.2.3 Color Moments

Color moments adalah metode pengukuran distribusi warna yang dapat digunakan untuk membedakan citra. Terdapat tiga perhitungan distribusi (*moments*) yang digunakan dalam *color moments* yaitu *mean*, standar deviasi, dan *skewness*. *Moments* dihitung pada tiap kanal warna yang ada. Jika terdapat tiga kanal warna pada citra, maka akan terdapat sembilan *moments* pada suatu citra (Keen, 2005). Ketiga *color moments* didefinisikan dengan Persamaan 2.3, Persamaan 2.4, dan Persamaan 2.5. Jumlah fitur yang didapatkan dari ekstraksi fitur ini sebanyak 9 fitur.

1. *Mean*

$$E_i = \frac{1}{N} \sum_{j=1}^N p_{ij} \quad (2.3)$$

2. Deviasi standar

$$\sigma_i = \sqrt{\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2} \quad (2.4)$$

3. *Skewness*

$$s_i = \sqrt[3]{\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3} \quad (2.5)$$

Keterangan:

- N = jumlah piksel pada citra
- p_{ij} = nilai dari piksel ke- j pada *channel* ke- i

2.2.4 Color Histogram

Color histogram merupakan representasi dari distribusi warna pada suatu citra, yang dibangun oleh kumpulan *bin* yang masing-masing merepresentasikan warna tertentu (Singh & Hemachandran, 2012). *Color histogram* suatu citra didefinisikan dalam vektor seperti pada Persamaan 2.6.

$$H = \{H[0], H[1], H[2], H[3], \dots H[i], \dots H[n]\} \quad (2.6)$$

Keterangan:

- i = nilai warna
- $H[i]$ = jumlah piksel dengan warna i pada citra
- n = total *bin* pada histogram

yang mana i merepresentasikan indeks *bin* warna pada *color histogram* dan $H[i]$ merepresentasikan jumlah piksel dari warna tertentu pada citra, dan n adalah total *bin* yang terdapat pada *color histogram*, dapat disimpulkan bahwa nilai setiap *bin* adalah jumlah piksel yang memiliki warna yang sama. *Bin* dapat terdiri dari satu distribusi atas semua *channel* dan dapat terdiri dari tiga distribusi yang terpisah antar masing-masing *channel*.



Gambar 2.2 Citra dengan histogramnya

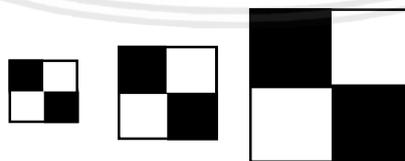
Sumber: (Zhang, 2002)

Pada *color model* RGB, terdapat 16.777.216 warna yang mungkin terjadi, warna-warna tersebut terdiri dari kombinasi masing-masing *channel* yang dapat memiliki 256 nilai. Menggunakan jumlah warna yang banyak tersebut tidak praktis dan tidak penting (Zhang, 2002). Kuantisasi warna adalah cara untuk mengurangi warna yang mungkin terjadi, sehingga dari 16.777.216 warna dapat dikurangi menjadi hanya n^3 warna saja. Kuantisasi dilakukan pada citra sebelum *color histogram* dibentuk, maka dari histogram awal yang berjumlah 256 bin dapat berkurang menjadi hanya 8 bin pada masing-masing *channel*. Kuantisasi dirumuskan dengan Persamaan 2.7.

$$\begin{pmatrix} r' \\ g' \\ b' \end{pmatrix} = \begin{pmatrix} r \times \frac{n}{256} \\ g \times \frac{n}{256} \\ b \times \frac{n}{256} \end{pmatrix} \quad (2.7)$$

Keterangan:

- n = jumlah *bin* yang diinginkan
- r = nilai dari piksel pada *channel* R
- g = nilai dari piksel pada *channel* G
- b = nilai dari piksel pada *channel* B
- r' = nilai dari piksel pada *channel* R setelah dikuantisasi
- g' = nilai dari piksel pada *channel* G setelah dikuantisasi
- b' = nilai dari piksel pada *channel* B setelah dikuantisasi



Gambar 2.3 Citra berdistribusi sama dengan ukuran berbeda

Pada Gambar 2.3, terdapat citra dengan distribusi yang sama dengan ukuran yang berbeda-beda. Perbedaan ukuran pada citra menandakan perbedaan jumlah piksel yang menyebabkan citra-citra memiliki histogram yang berbeda meskipun memiliki distribusi yang sama. Maka dari itu, normalisasi perlu dilakukan. *Color histogram* yang telah dinormalisasi dapat dinotasikan dengan H' yang didefinisikan pada Persamaan 2.8 dan Persamaan 2.9.



$$H' = \{H'[0], H'[1], H'[2], H'[3], \dots H'[i], \dots H'[n]\} \quad (2.8)$$

yang mana

$$H'[i] = \frac{H[i]}{p} \quad (2.9)$$

Keterangan:

- $H[i]$ = jumlah piksel dengan warna i pada citra
- p = jumlah piksel pada citra

Kodituwaku & Selvarajah (2010) menyebutkan bahwa *mean* dan standar deviasi dari histogram dapat dihitung dengan menggunakan Persamaan 2.10 dan Persamaan 2.11.

$$mean = \sum_{i=0}^{255} i * H[i] / \sum_{i=0}^{255} H[i] \quad (2.10)$$

$$deviasi\ standar = \sqrt{\frac{\sum_{i=0}^{255} H[i] * (i - mean)^2}{\sum_{i=0}^{255} H[i]}} \quad (2.11)$$

Keterangan:

- $H[i]$ = jumlah piksel dengan warna i pada citra

Jumlah fitur yang didapatkan dari ekstraksi fitur ini sebanyak 30 fitur.

2.2.5 Gray Level Cooccurrence Matrix

GLCM adalah salah satu metode ekstraksi fitur tekstur pada citra yang umum digunakan. GLCM memperhitungkan terjadinya tingkat keabuan pada pasangan antar piksel dalam beberapa arah atau sudut seperti horizontal, vertikal, dan diagonal (Puviarasan, et al., 2014). Jika diberikan jarak (d) dan sudut (θ) dengan *gray scale pixel* i, j diekspresikan sebagai jumlah elemen *co-occurrence matrix* dengan Persamaan 2.12.

$$P(i, j|d, \theta) = \frac{P(i, j|d, \theta)}{\sum_i \sum_j P(i, j|d, \theta)} \quad (2.12)$$

Keterangan:

- $P(i, j|d, \theta)$ = nilai dari *cooccurrence matrix*.
- i = piksel dengan nilai i
- j = piksel dengan nilai j
- d = jarak tetangga yang dihitung kejadiannya
- θ = sudut letak tetangga yang dihitung kejadiannya

Haralick, et al., (1973) menjelaskan bahwa terdapat 14 fitur yang didapatkan dari hasil perhitungan GLCM, namun pada penelitian ini hanya menggunakan sebanyak empat fitur yang umum digunakan, yaitu *energy*, *entropy*, *contrast*, dan *homogeneity*. Keempat fitur tersebut dirumuskan dengan Persamaan 2.13, Persamaan 2.14, Persamaan 2.15, dan Persamaan 2.16.

1. Energy

$$Energy = \sum_i \sum_j P(i, j)^2 \quad (2.13)$$



2. Entropy

$$Entropy = -\sum_i \sum_j P(i, j) \log(P(i, j)) \quad (2.14)$$

3. Contrast

$$Contrast = \sum_i \sum_j (i - j)^2 P(i, j) \quad (2.15)$$

4. Homogeneity

$$Homogeneity = \sum_i \sum_j \frac{P(i, j)}{1 + (i - j)^2} \quad (2.16)$$

Keterangan:

- $P(i, j)$ = nilai *cooccurrence matrix* piksel dengan nilai i dan j .

Jumlah fitur yang didapatkan dari ekstraksi fitur ini sebanyak 16 fitur.

2.2.6 Information Gain Feature Selection

Information Gain Feature Selection (IGFS) adalah metode pemeringkatan atribut paling sederhana dan banyak digunakan pada analisis data citra (Chormunge & Jena, 2016). IGFS dapat mendeteksi fitur yang memiliki informasi paling banyak yang didapatkan dari perhitungan *entropy* (Shaltout, et al., 2014). *Entropy* adalah ukuran ketidakpastian suatu kelas menggunakan probabilitas dari sebuah atribut yang dirumuskan dengan Persamaan 2.17 dan Persamaan 2.18.

$$H(C) = -\sum_{c \in C} p(c) \log_2 p(c) \quad (2.17)$$

$$H(C|A) = -\sum_{a \in A} p(a) \sum_{c \in C} p(c|a) \log_2 p(c|a) \quad (2.18)$$

Jumlah *entropy* yang berkurang dari suatu kelas menandakan informasi tambahan tentang suatu kelas. Setiap atribut A_i diberikan nilai berdasarkan *Information Gain* antara dirinya sendiri dengan kelas yang dirumuskan dengan Persamaan 2.19.

$$IG_i = H(C) - H(C|A_i) \quad (2.19)$$

Keterangan:

- $p(c)$ = probabilitas atribut c
- $p(c|a)$ = probabilitas kelas c bersyarat atribut a
- $p(a)$ = probabilitas atribut a
- $H(C)$ = nilai *entropy* kelas c
- $H(C|A_i)$ = nilai *entropy* kelas c bersyarat atribut a

2.2.7 Similarity Measure

Untuk melakukan *retrieval* digunakan *similarity measure* atau menghitung kesamaan antara *query* dengan data yang ada pada basis data. Salah satu metode *similarity measure* adalah *Euclidean distance*. *Euclidean distance* dirumuskan pada Persamaan 2.20.

$$d(Q, T) = \left(\sum_{i=0}^{2n-1} (Q_i - T_i)^2 \right)^{\frac{1}{2}} \quad (2.20)$$

Keterangan:

- Q = nilai dari *query*
- T = nilai dari dataset
- n = jumlah atribut
- i = indeks atribut

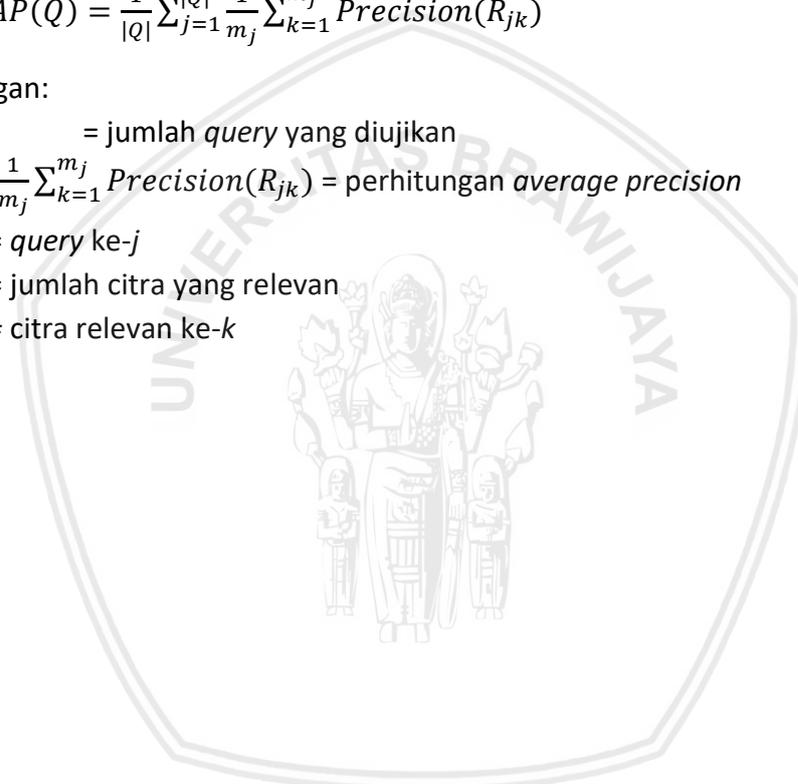
2.2.8 Mean Average Precision

Mean Average Precision (MAP) adalah metode evaluasi untuk *ranked retrieval*. MAP menghitung rata-rata dari rata-rata *precision* seluruh informasi yang dibutuhkan. Persamaan dari MAP terdapat pada Persamaan 2.21.

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \quad (2.21)$$

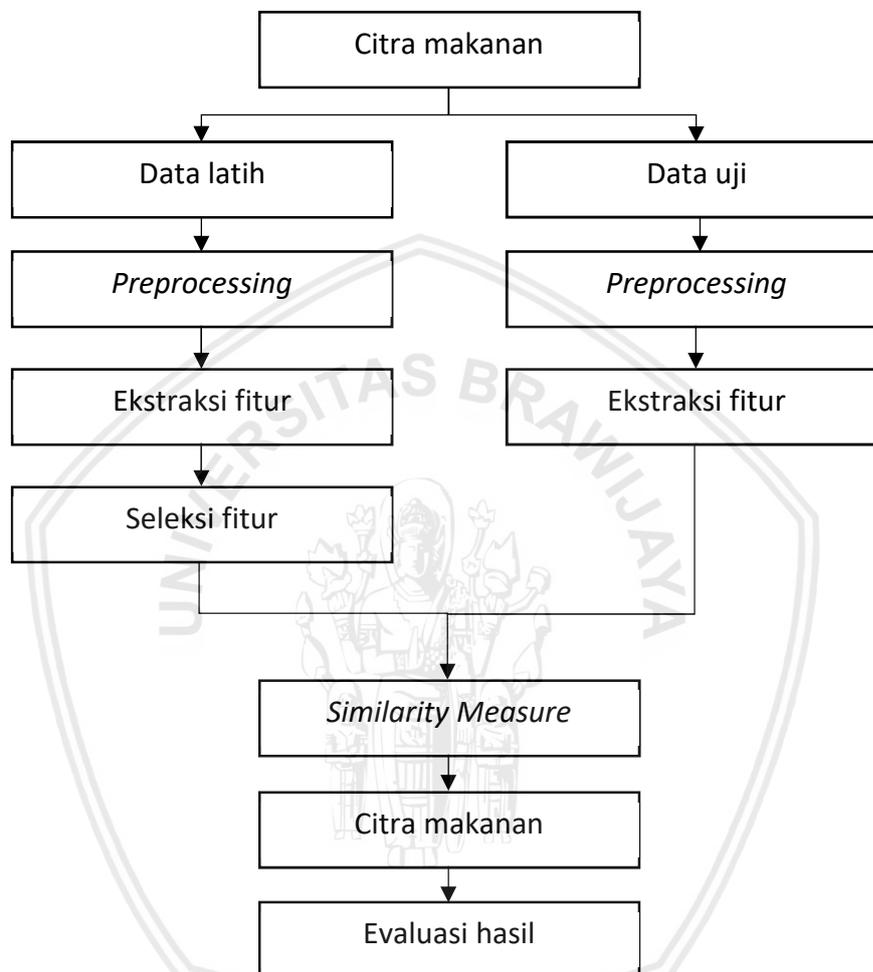
Keterangan:

- Q = jumlah *query* yang diujikan
- $\sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk})$ = perhitungan *average precision*
- j = *query* ke- j
- m = jumlah citra yang relevan
- k = citra relevan ke- k



BAB 3 METODOLOGI

Pada Bab 3 dijelaskan langkah-langkah yang akan dilakukan pada penelitian untuk mencapai tujuan penelitian. Jenis penelitian yang dilakukan yaitu penelitian non-implementatif analitik. Metode secara umum yang akan dilakukan pada penelitian ini diilustrasikan dengan Gambar 3.1.



Gambar 3.1 Diagram alir metode penelitian

Langkah pertama dari penelitian ini adalah mengumpulkan *dataset* citra makanan untuk digunakan sebagai masukan. Dari *dataset* yang telah dikumpulkan, data dibagi menjadi dua bagian yaitu data untuk pelatihan dan data untuk pengujian. *Preprocessing* dan ekstraksi fitur dilakukan pada data latih maupun data uji. Ekstraksi fitur yang digunakan pada penelitian ini adalah *Color Moments* yang menghasilkan sebanyak 9 fitur, *Color Histogram* yang menghasilkan sebanyak 30 fitur, dan *Gray Level Cooccurrence Matrix* yang menghasilkan sebanyak 16 fitur. Jika ditotalkan, maka akan didapatkan 39 fitur dari ekstraksi fitur warna, 16 fitur dari ekstraksi fitur tekstur, dan 55 fitur dari kombinasi fitur tekstur dan warna. Seleksi fitur dengan menggunakan *Information Gain* dilakukan pada data latih untuk mengetahui fitur apa yang paling

berpengaruh terhadap hasil dari sistem. Selanjutnya, fitur-fitur tersebut digunakan pada tahap *similarity measure* untuk mengetahui data latih yang paling mirip dengan data uji dan memberikan keluaran berupa citra makanan. Kemudian dilakukan evaluasi dengan menggunakan MAP terhadap hasil yang diberikan oleh sistem.

3.1 Teknik Pengumpulan Data

Penelitian ini menggunakan data primer berupa citra makanan dengan pengambilan gambar dengan sudut dan tingkat elevasi yang berbeda-beda dan ditentukan secara acak. Pengambilan gambar dilakukan dengan menggunakan kamera *smartphone* Samsung Galaxy A3 2015 dan diambil pada tanggal 28 Agustus 2018 pada pukul 10.00-13.00 di bawah pencahayaan matahari di depan ruang F9.3 gedung F FILKOM lantai 9.

Jumlah citra yang diambil sebanyak 249 citra yang terbagi menjadi 8 kelas, yaitu Donat, Mie Goreng, Mie Gepeng, Telor Ceplok, Telor Dadar, Fried Chicken, Rendang, dan Nasi Kuning. Dari 249 citra makanan, 241 data digunakan sebagai data latih dan 8 data digunakan sebagai data uji, yang mana data uji terdiri dari 1 citra yang diambil secara acak dari masing-masing kelas. Data latih yang digunakan terdapat pada Tabel 3.1 dan data uji terdapat pada Tabel 6.1. Contoh lain dari data yang digunakan terdapat pada LAMPIRAN A.

Tabel 3.1 Data latih

Kelas	Jumlah	Contoh citra
Donat	59	
Mie Goreng	15	
Mie Gepeng	15	
Telor Ceplok	58	
Telor Dadar	15	
Fried Chicken	12	

Tabel 3.1 Data latih (lanjutan)

Kelas	Jumlah	Contoh citra
Rendang	48	
Nasi Kuning	17	

3.2 Peralatan Pendukung

Penjelasan mengenai peralatan pendukung yang digunakan dibagi menjadi dua, yaitu spesifikasi perangkat keras dan perangkat lunak.

3.2.1 Perangkat Keras Pendukung

Spesifikasi dari perangkat keras yang digunakan sebagai pendukung dalam implementasi dari sistem adalah sebagai berikut.

1. Prosesor: Intel® Core™ i5-4200U CPU @ 1.60GHz 2.3GHz
2. RAM: 4 GB
3. GPU: NVIDIA® GEFORCE® GT 840M 2GB
4. Harddisk 1TB
5. Monitor 14 *inch*
6. *Keyboard*
7. *Mouse*
8. *Smartphone* Samsung Galaxy A3 2015

3.2.2 Perangkat Lunak Pendukung

Spesifikasi dari perangkat keras lunak yang digunakan sebagai pendukung dalam implementasi dari sistem adalah sebagai berikut.

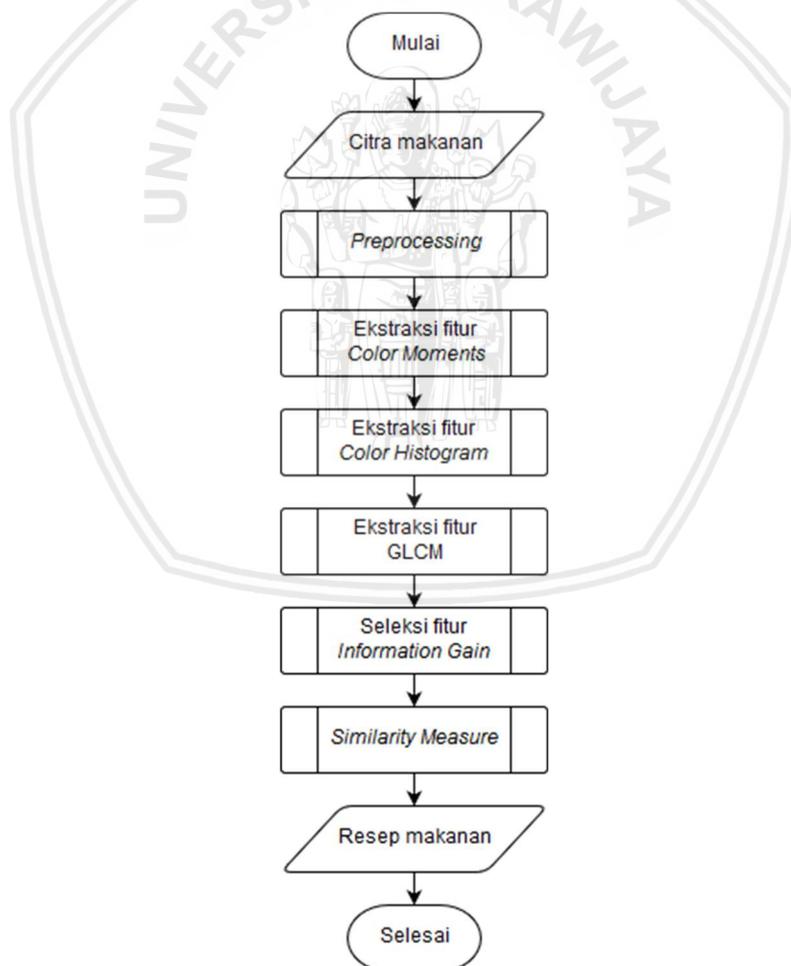
1. Sistem operasi: Windows 10 Pro
2. Bahasa pemrograman: Python 3.7
3. IDE: JetBrains PyCharm

BAB 4 PERANCANGAN

Pada bab ini dilakukan perancangan dan perhitungan manual yang digunakan sebagai acuan untuk mengimplementasikan algoritme Information Gain Feature Selection pada sistem temu kembali citra makanan menggunakan ekstraksi fitur warna dan tekstur.

4.1 Perancangan Algoritme

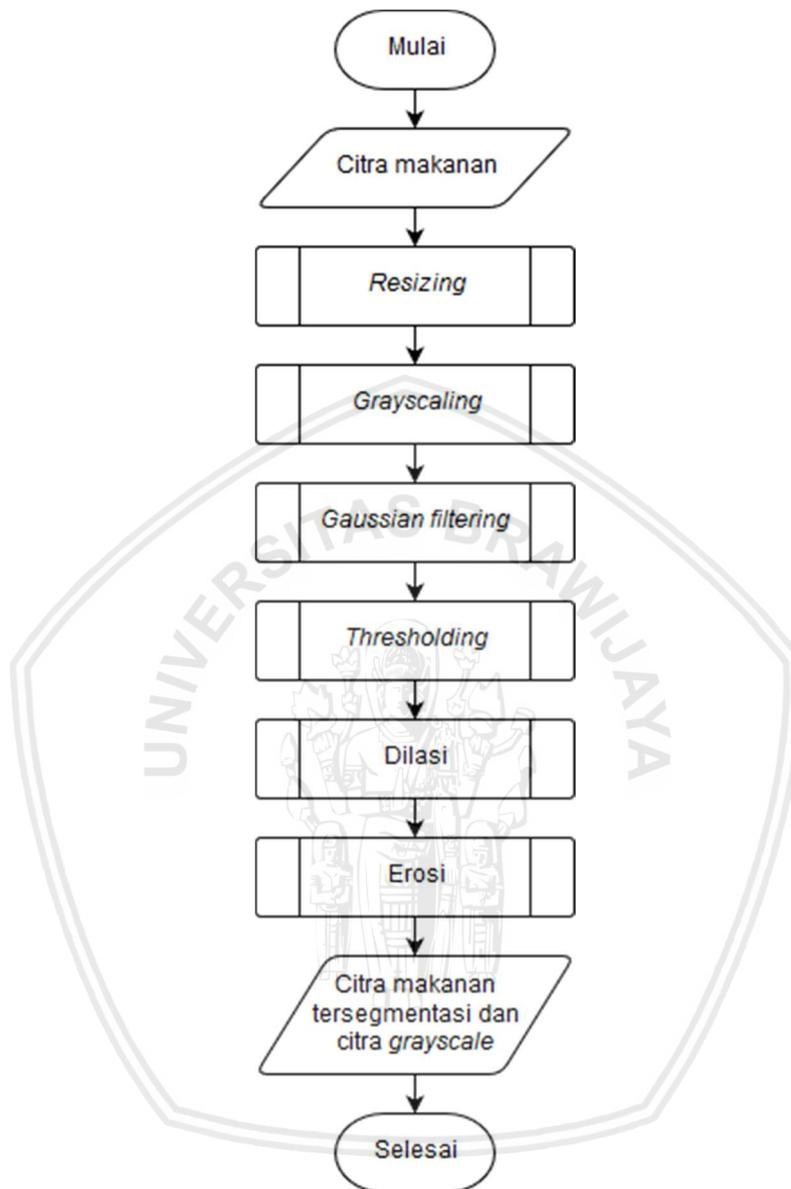
Perancangan algoritme yang digunakan pada sistem digambarkan dengan diagram alir terdapat pada Gambar 4.1 Diagram alir sistem. Sistem menerima masukan berupa citra makanan, lalu dilakukan *preprocessing* dan dilanjutkan dengan melakukan ekstraksi fitur *Color Moments*, *Color Histogram*, dan GLCM, lalu dilakukan seleksi fitur *Information Gain*, dan melakukan pengukuran *similarity measure* menggunakan *Euclidean distance*. Sistem memiliki keluaran berupa citra makanan pada basis data yang memiliki jarak kesamaan terdekat dengan citra yang dijadikan masukan.



Gambar 4.1 Diagram alir sistem

4.1.1 Preprocessing

Diagram alir dari proses *preprocessing* ditunjukkan oleh Gambar 4.2.



Gambar 4.2 Diagram alir *preprocessing*

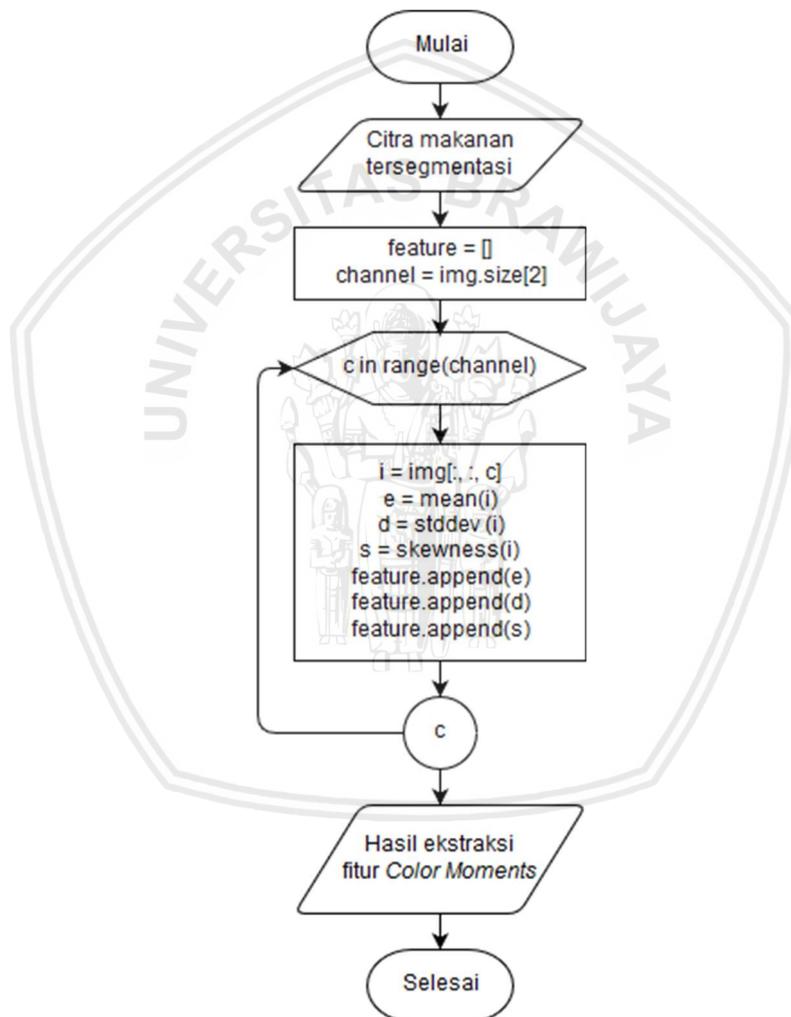
Berikut adalah penjelasan dari Gambar 4.2.

1. Proses *preprocessing* memiliki masukan berupa citra makanan
2. Tahap pertama dari *preprocessing* adalah tahap *resizing* untuk mengubah ukuran dari citra
3. Selanjutnya melakukan proses *grayscale* untuk mendapatkan citra *grayscale*
4. Melakukan proses *Gaussian filtering* untuk mengurangi *noise*.
5. Selanjutnya melakukan *Thresholding* untuk memisahkan antara objek yang ingin diproses dengan background.

- Melakukan dilasi dan erosi untuk memperbaiki hasil dari *thresholding* yang masih terdapat *noise* dan menambal bagian objek yang terbangun pada proses *thresholding*.
- Keluaran berupa citra yang tersegmentasi berwarna yang akan digunakan pada tahap ekstraksi fitur warna dan citra tersegmentasi dalam *grayscale* yang akan digunakan pada tahap ekstraksi fitur tekstur.

4.1.2 Ekstraksi Fitur *Color Moments*

Diagram alir dari proses *preprocessing* ditunjukkan oleh Gambar 4.3.



Gambar 4.3 Diagram alir ekstraksi fitur *Color Moments*

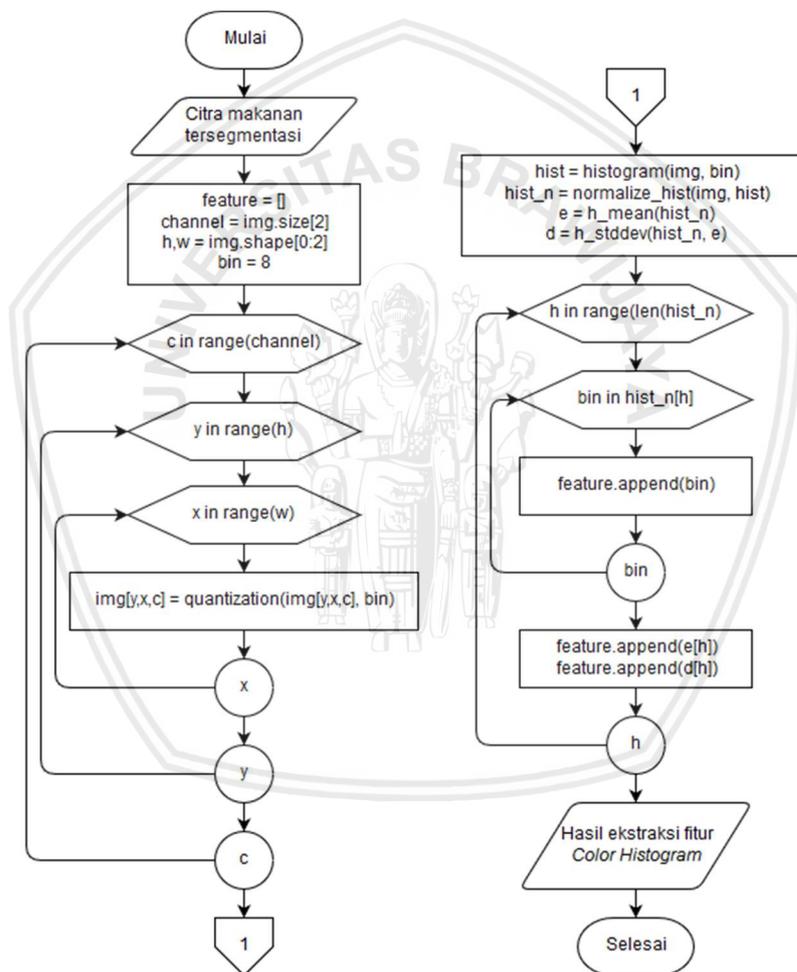
Berikut adalah penjelasan dari Gambar 4.3.

- Pada tahap ekstraksi fitur *Color Moments* memiliki masukan berupa citra makanan yang tersegmentasi.

2. Melakukan inisialisasi variabel *feature* yang digunakan untuk menampung hasil dari ekstraksi fitur dan inisialisasi variabel *channel* yang berisi nilai dari jumlah *channel* warna yang terdapat pada citra.
3. Selanjutnya melakukan perulangan untuk menghitung nilai *mean*, deviasi standar, dan *skewness* dari masing-masing *channel*. Masing-masing hasil dari perhitungan tersebut dimasukkan ke dalam variabel *feature*.
4. Keluaran dari tahap ini adalah variabel *feature* yang berisi nilai *mean*, deviasi standar, dan *skewness* dari masing-masing *channel*.

4.1.3 Ekstraksi Fitur *Color Histogram*

Diagram alir dari proses *preprocessing* ditunjukkan oleh Gambar 4.4.



Gambar 4.4 Diagram alir ekstraksi fitur *Color Histogram*

Berikut adalah penjelasan dari Gambar 4.4.

1. Tahap ekstraksi fitur *Color Histogram* memiliki masukan berupa citra makanan yang tersegmentasi.
2. Melakukan inisialisasi variabel *feature* yang akan digunakan untuk menampung hasil dari ekstraksi fitur, inisialisasi variabel *channel* yang berisi

jumlah *channel* dari citra, inialisasi variabel *h* dan *w* yang berisi ukuran panjang dan lebar dari citra, dan inialisasi variabel *bin* yang berisi jumlah *bin* yang kita inginkan untuk hasil dari proses kuantisasi.

3. Melakukan perulangan pada setiap piksel pada masing-masing *channel* untuk dilakukan proses kuantisasi.
4. Melakukan pembentukan histogram yang disimpan dalam variabel *hist* dengan fungsi *histogram()*, lalu histogram yang telah dibentuk dinormalisasikan dengan fungsi *normalize_hist()* dan disimpan pada variabel *hist_n*. Selanjutnya melakukan perhitungan *mean* dan deviasi standar terhadap histogram yang sudah dinormalisasi dengan menggunakan fungsi *h_mean()* dan *h_stddev()* yang disimpan pada variabel *e* untuk nilai *mean* dan variabel *d* untuk nilai deviasi standar.
5. Melakukan perulangan untuk menyimpan nilai dari tiap *bin* dari histogram pada masing-masing *channel* ke dalam variabel *feature*. Selanjutnya menyimpan nilai variabel *e* dan *d* ke dalam variabel *feature*.
6. Keluaran dari tahap *Color Histogram* adalah histogram dari citra dan hasil perhitungan *mean* dan deviasi standar dari histogram yang terdapat pada variabel *feature*.

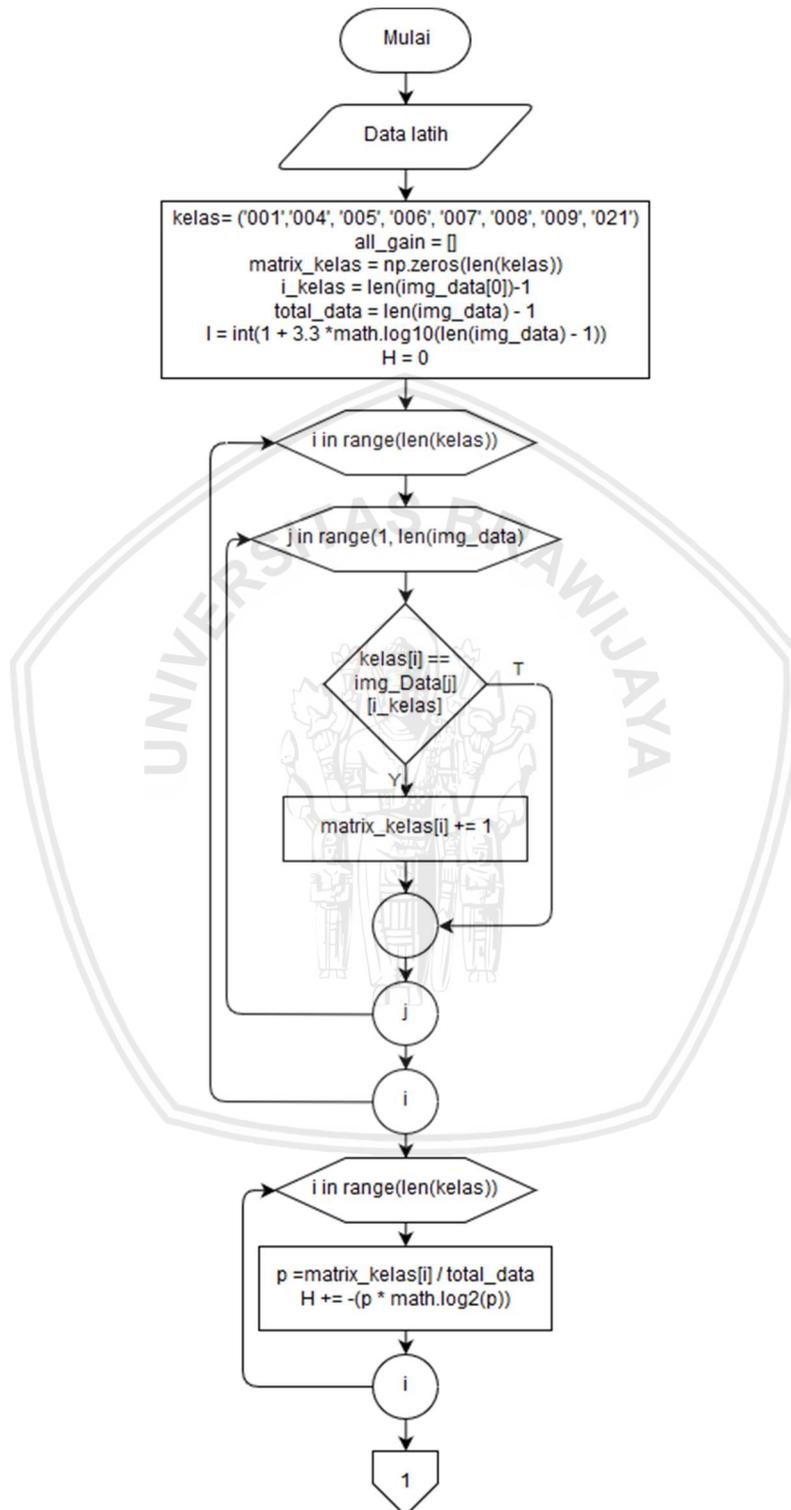
4.1.4 Ekstraksi Fitur GLCM

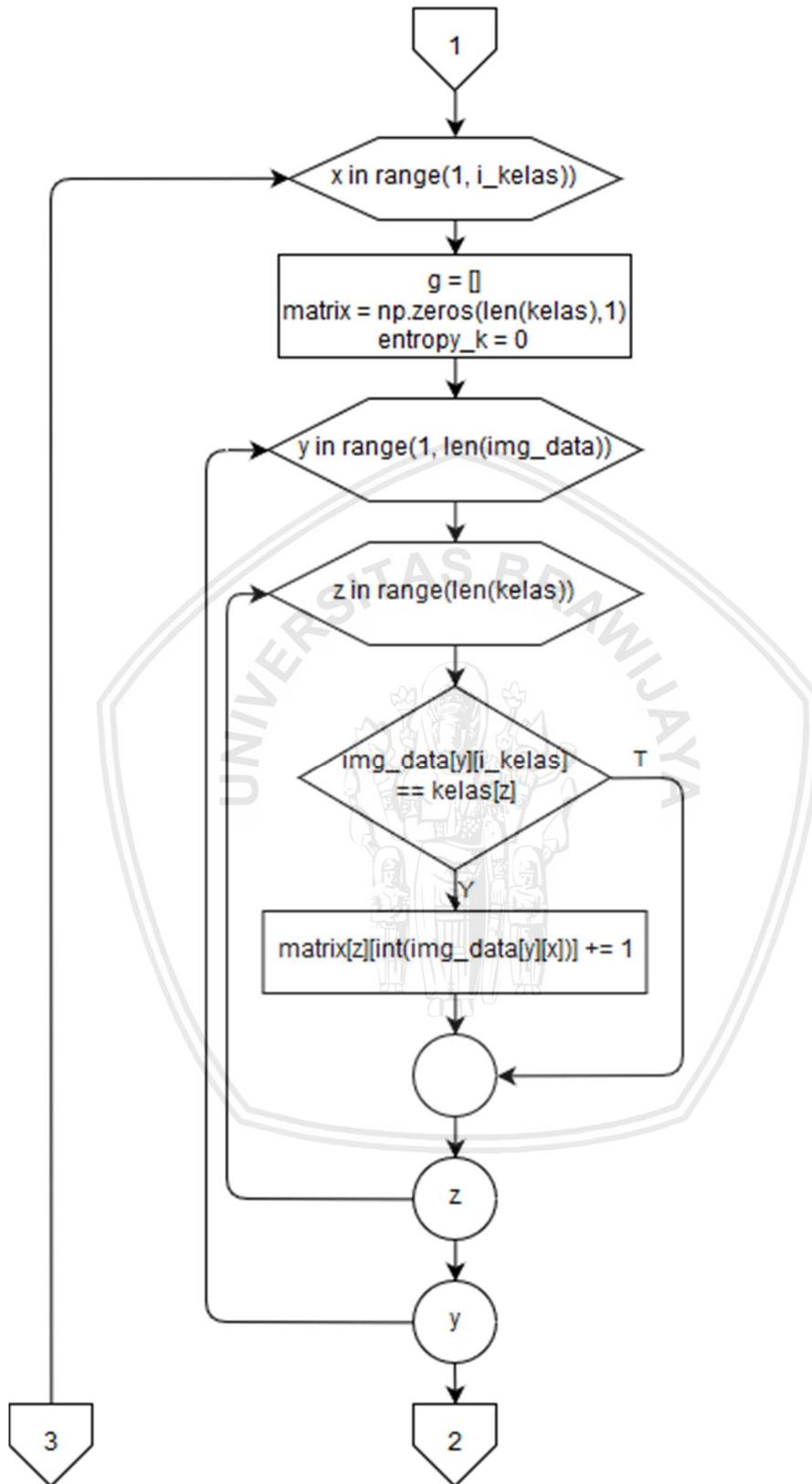
Diagram alir dari proses *preprocessing* ditunjukkan oleh Gambar 4.5. Penjelasan dari Gambar 4.5 adalah sebagai berikut.

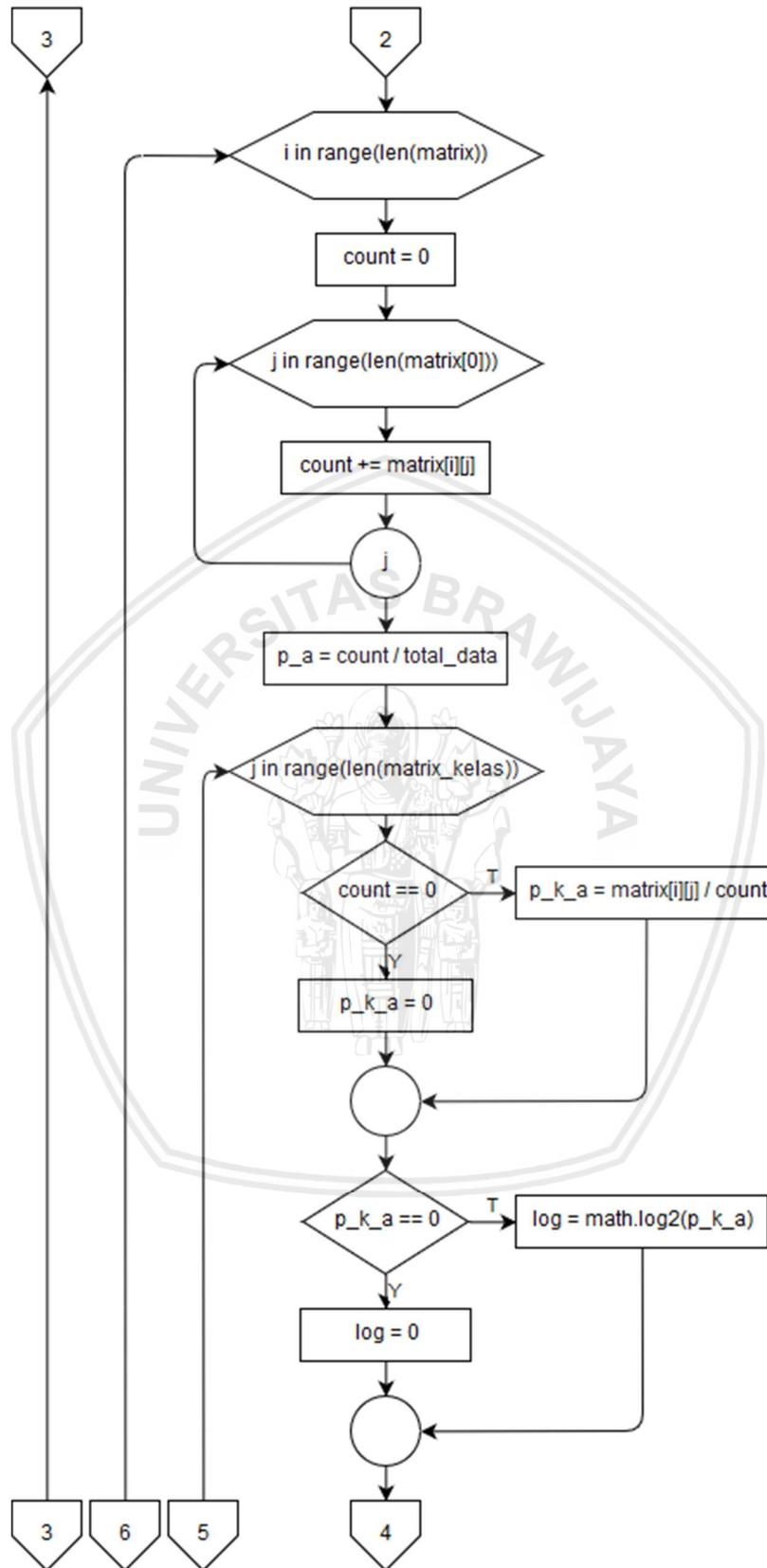
1. Tahap ekstraksi fitur GLCM memiliki masukan berupa citra makanan tersegmentasi dalam bentuk *grayscale*.
2. Melakukan inialisasi variabel *h* dan *w* yang berisi ukuran panjang dan lebar dari citra, variabel *max* dan *min* yang berisi nilai maksimum dan minimum yang dimiliki citra untuk digunakan sebagai ukuran dari matriks kejadian atau *coocurance matrix*, variabel *angle* yang berisi sudut yang akan dihitung kejadiannya, variabel *co* yang menampung nilai dari matriks kejadian, dan variabel *glcm* yang akan menampung hasil dari perhitungan GLCM.
3. Melakukan pembentukan matriks kejadian untuk tiap sudut dengan ukuran yang didapatkan dari pengurangan nilai maksimum dan minimum dari citra. Tahap ini dilakukan dengan melakukan perulangan sebanyak jumlah sudut yang dihitung.
4. Pada tahap ini terdapat perulangan untuk menghitung jumlah kejadian ketetangaan pada citra pada empat sudut yang kemudian disimpan ke dalam variabel *co*.
5. Melakukan normalisasi matriks kejadian dan melakukan perhitungan *energy*, *entropy*, *contrast*, dan *homogeneity* dari setiap sudut dengan menggunakan nilai dari matriks kejadian yang telah dinormalisasi.
6. Tahap ini memiliki keluaran ini adalah nilai *energy*, *entropy*, *contrast*, dan *homogeneity* dari setiap sudut.

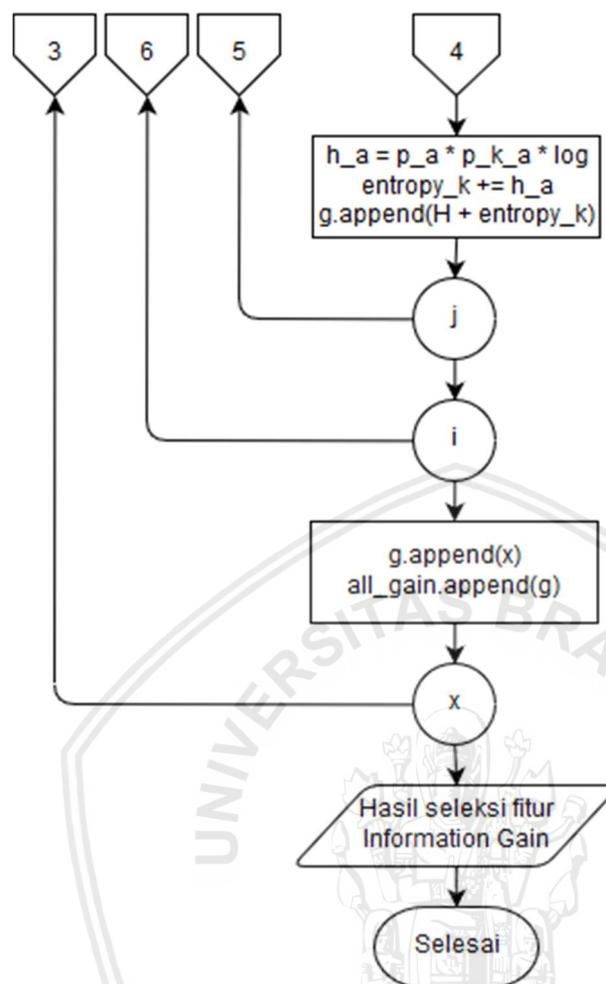
4.1.5 Seleksi Fitur *Information Gain*

Diagram alir dari seleksi fitur *Information Gain* terdapat pada Gambar 4.6.









Gambar 4.6 Diagram alir seleksi fitur *Information Gain*

Berikut adalah penjelasan dari Gambar 4.6.

1. Pada seleksi fitur *Information Gain* memiliki masukan berupa data latih yang telah diubah ke dalam bentuk kategorikal.
2. Melakukan perulangan dengan variabel i sebanyak jumlah kelas yang terdapat pada data latih dan perulangan dengan variabel j sebanyak jumlah data latih untuk menghitung jumlah kejadian dari kelas pada data latih. Terdapat percabangan jika nilai dari variabel $kelas$ sama dengan kelas dari data latih, maka nilai kejadian dari kelas tersebut pada variabel $matrix_kelas$ akan ditambah satu.
3. Melakukan perhitungan *entropy* dari keseluruhan kelas yang disimpan pada variabel H .
4. Melakukan peluang kejadian masing-masing kelas bersyarat atribut tertentu. Tahap ini dilakukan dengan melakukan perulangan dengan variabel x sebanyak jumlah atribut yang ada. Kemudian dilakukan perulangan dengan variabel y sebanyak jumlah data pada data latih dan perulangan dengan variabel z sebanyak jumlah kelas yang ada. Terdapat percabangan jika nilai

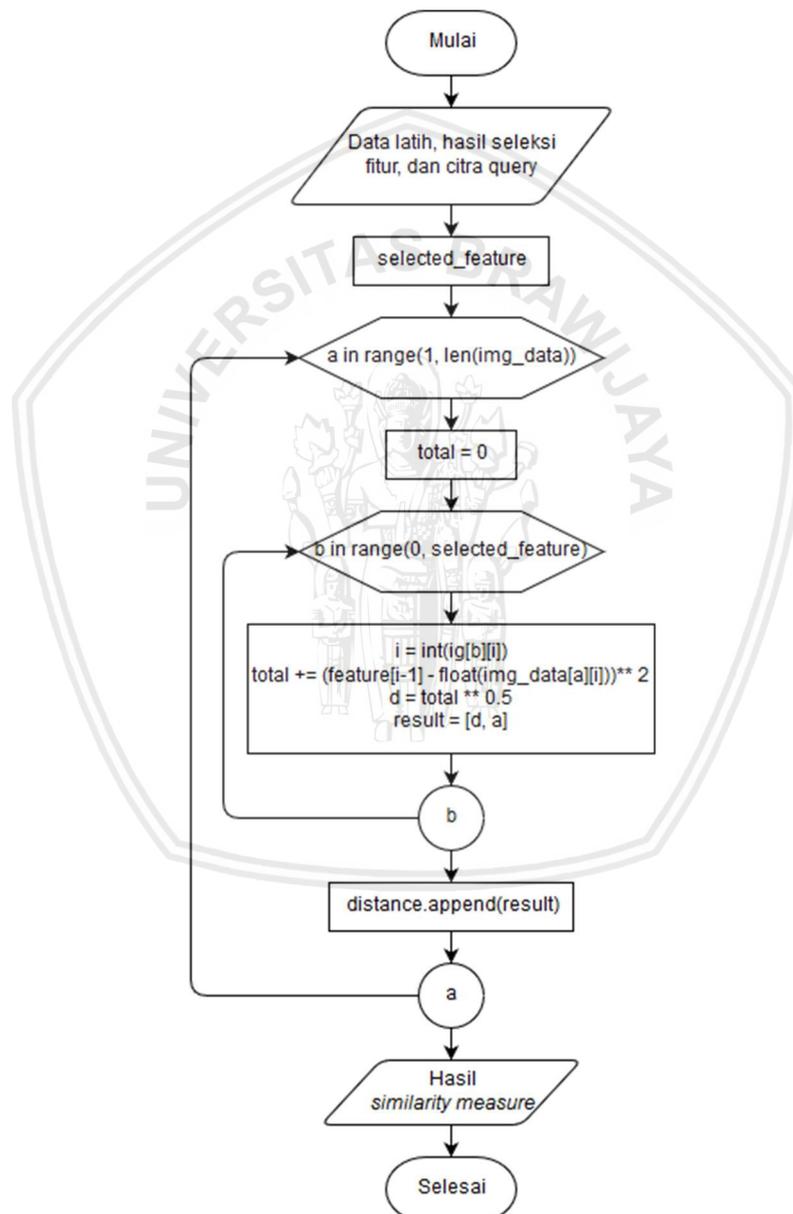
- pada variabel *kelas* sama dengan kelas dari data latih maka nilai pada variabel *matrix* pada indeks z dan atribut x pada data ke- y akan ditambah satu.
5. Selanjutnya melakukan perulangan dengan variabel j sebanyak panjang dari variabel *matrix*. Pada tahap ini dilakukan penjumlahan nilai dari variabel *matrix* yang merupakan jumlah kejadian dari setiap nilai berdasarkan kelasnya. Hasil perhitungan disimpan pada variabel *count*.
 6. Melakukan perhitungan peluang kejadian nilai dari atribut pada data latih.
 7. Melakukan perhitungan peluang kejadian suatu kelas bersyarat atribut. Terdapat percabangan jika variabel *count* bernilai 0 maka nilai variabel p_{k_a} akan bernilai 0. Hal ini menandakan tidak ada kejadian kelas bersyarat atribut. Lalu terdapat percabangan kembali jika variabel p_{k_a} bernilai 0 maka nilai variabel *log* akan bernilai 0.
 8. Melakukan perhitungan *entropy* dari masing-masing kelas bersyarat masing-masing atribut. Kemudian nilai *entropy* tersebut dijumlahkan dan disimpan pada variabel *entropy_k*. Kemudian melakukan penjumlahan variabel H dengan variabel *entropy_k* yang merupakan perhitungan nilai *gain* dari atribut yang sedang dihitung nilai *gain*-nya. Hasil penjumlahan disimpan pada variabel g .
 9. Menambahkan nilai dari variabel x yang merupakan indeks dari atribut yang sedang dihitung nilai *gain*-nya.
 10. Menggabungkan nilai *gain* dari semua atribut dengan disimpan ke variabel *all_gain*.
 11. Keluaran dari tahap ini adalah nilai *gain* dari setiap fitur.

4.1.6 Similarity Measure

Pada tahap *similarity measure* memiliki tiga masukan yaitu data latih, hasil dari seleksi fitur, dan citra yang dijadikan *query*. Pada tahap ini, dilakukan pengukuran jarak dari fitur yang dimiliki oleh citra *query* dengan data yang terdapat pada data latih. Fitur yang digunakan dalam pengukuran jarak dapat diubah sesuai dengan kebutuhan, jika menggunakan sepuluh fitur, maka fitur yang digunakan adalah 10 fitur dengan nilai *gain* tertinggi. Keluaran dari tahap ini adalah nilai jarak antara citra *query* dengan data latih dan diurutkan dari nilai terkecil sampai yang terbesar. Diagram alir dari *similarity measure* terdapat pada Gambar 4.7. Berikut adalah penjelasan dari Gambar 4.7.

1. Tahap ini memiliki masukan berupa data latih, hasil seleksi fitur, dan nilai ekstraksi fitur dari citra *query*.
2. Melakukan inisialisasi variabel *selected_feature* yang berisi jumlah fitur yang akan digunakan dalam proses *similarity measure*.
3. Melakukan perulangan dengan variabel a sebanyak jumlah data latih.
4. Melakukan inisialisasi variabel *total* untuk menyimpan hasil perhitungan *similarity measure*.

5. Melakukan perulangan dengan variabel b sebanyak jumlah fitur yang digunakan untuk perhitungan *similarity measure*.
6. Melakukan proses *similarity measure* dengan perhitungan *Euclidean distance*. Kemudian data latih dan nilai jaraknya dengan *query* disimpan ke dalam variabel *result*.
7. Tahap ini memiliki keluaran berupa jarak antara *query* dengan data pada setiap data latih.



Gambar 4.7 Diagram alir *similarity measure*

4.2 Perhitungan Manual

Pada sub bab ini dilakukan perhitungan manual dari tahap-tahap yang dilakukan pada penelitian ini. Citra yang digunakan pada perhitungan manual ini diambil dari potongan citra makanan rendang dengan ukuran 10 x 10 piksel yang ditampilkan pada Gambar 4.8.



Gambar 4.8 Citra yang digunakan pada perhitungan manual

4.2.1 Ekstraksi Fitur Warna *Color Moments*

Berikut adalah perhitungan manual dari ekstraksi fitur warna *Color Moments*.

1. Memisahkan masing-masing *channel* pada gambar, pada penelitian ini *color model* yang digunakan adalah RGB, sehingga citra akan dipisah menjadi *channel* R, G, dan B. Nilai masing-masing piksel pada tiap *channel* ditunjukkan pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3. Pada tabel, posisi piksel secara horizontal dilambangkan dengan huruf x dan posisi piksel secara vertikal dilambangkan dengan huruf y.

Tabel 4.1 Tabel *channel* R

x,y	0	1	2	3	4	5	6	7	8	9
0	145	140	136	129	144	127	122	119	118	120
1	141	142	127	126	117	126	122	126	120	113
2	134	126	127	117	125	123	124	110	128	109
3	130	127	124	126	122	127	119	114	113	112
4	126	125	130	123	128	121	117	114	112	123
5	131	131	124	118	114	122	116	121	117	122
6	124	125	127	128	113	120	113	121	112	120
7	128	110	115	118	115	114	112	112	114	116
8	123	116	112	112	98	105	103	112	106	109
9	122	117	113	116	112	118	102	98	100	108

Tabel 4.2 Tabel channel G

x,y	0	1	2	3	4	5	6	7	8	9
0	99	94	91	84	99	82	78	75	74	76
1	95	96	82	81	72	81	78	82	76	69
2	89	81	82	72	80	77	80	66	83	64
3	85	82	79	80	76	81	75	70	68	67
4	80	79	86	79	84	76	72	69	67	78
5	85	85	79	73	69	77	71	76	72	77
6	79	80	82	83	68	75	68	76	67	75
7	83	65	70	73	70	69	67	67	69	71
8	80	73	69	69	55	62	60	69	63	66
9	79	74	70	73	69	75	59	55	57	65

Tabel 4.3 Tabel channel B

x,y	0	1	2	3	4	5	6	7	8	9
0	65	60	58	51	70	53	51	48	49	51
1	62	63	49	50	43	52	51	55	51	44
2	58	50	51	43	51	51	55	41	60	41
3	56	53	50	54	50	57	50	45	45	44
4	54	53	59	54	59	53	49	46	46	57
5	62	61	56	50	46	54	48	55	51	56
6	58	57	61	62	47	54	47	57	48	56
7	64	44	51	52	51	50	48	48	50	52
8	63	54	52	50	38	43	41	50	44	47
9	62	57	53	56	52	58	40	36	38	46

2. Selanjutnya dilakukan ekstraksi fitur dengan melakukan perhitungan *mean*, deviasi standar, dan *skewness* masing-masing *channel* menggunakan rumus yang terdapat pada Persamaan 2.3, Persamaan 2.4, dan Persamaan 2.5. Berikut adalah perhitungannya pada masing-masing *channel*:

a. *Channel R*

- $Mean = \frac{145+140+136+1 \dots +108}{100} = \frac{11991}{100} = 119,91$

- Deviasi standar

$$= \sqrt{\frac{(145-119,91)^2+(140-119,91)^2+(136-119,91)^2+\dots+(108-119,91)^2}{100}}$$

$$= \sqrt{\frac{620,5081+403,6081+258,88 \dots +14,848}{100}}$$

$$= \sqrt{\frac{8634,189}{100}} = 9,292$$

- *Skewness*

$$= \sqrt[3]{\frac{(145-119,91)^3+(140-119,91)^3+(136-119,91)^3+\dots+(108-119,91)^3}{100}}$$



$$= \sqrt[3]{\frac{15794,36+8108,487+4165,51+\dots+(-168,41)}{100}}$$

$$= \sqrt[3]{\frac{17484,3042}{100}} = 5,591$$

b. Channel G

- $Mean = \frac{99+94+91+84+\dots+65}{100} = \frac{7534}{100} = 75,34$

- Deviasi standar

$$= \sqrt{\frac{(99-75,34)^2+(94-75,34)^2+(91-75,34)^2+\dots+(65-75,34)^2}{100}}$$

$$= \sqrt{\frac{559,7956+348,1956+245,236+\dots+7,9956}{100}}$$

$$= \sqrt{\frac{7676,439}{100}} = 8,761$$

- Skewness

$$= \sqrt[3]{\frac{(99-75,34)^3+(94-75,34)^3+(91-75,34)^3+\dots+(65-75,34)^3}{100}}$$

$$= \sqrt[3]{\frac{13244,76+6497,33+3840,39+\dots+64,462}{100}}$$

$$= \sqrt[3]{\frac{17946,1}{100}} = 5,640$$

c. Channel B

- $Mean = \frac{65+60+58+51+\dots+46}{100} = \frac{5187}{100} = 51,87$

- Deviasi standar

$$= \sqrt{\frac{(65-51,87)^2+(60-51,87)^2+(58-51,87)^2+\dots+(46-51,87)^2}{100}}$$

$$= \sqrt{\frac{172,3969+66,0969+37,5769+\dots+34,4569}{100}}$$

$$= \sqrt{\frac{4199,31}{100}} = 6,48$$

- Skewness

$$= \sqrt[3]{\frac{(65-51,87)^3+(60-51,87)^3+(58-51,87)^3+\dots+(46-51,87)^3}{100}}$$

$$= \sqrt[3]{\frac{2263,571+537,3678+230,346+\dots+(-202,262)}{100}}$$

$$= \sqrt[3]{\frac{1210,9506}{100}} = 2,296$$

4.2.2 Ekstraksi Fitur Warna *Color Histogram*

Berdasarkan teori yang telah dijelaskan pada dasar teori di Bab 2, dibentuk *Color Histogram* dengan melakukan kuantisasi terlebih dahulu pada masing-



masing *channel* dari Gambar 4.8. Pada penelitian ini, kuantisasi dilakukan untuk mengurangi jumlah *bin* yang berjumlah 256 (8^3) menjadi 8 *bin* (2^3). Contoh perhitungan dari kuantisasi dengan menggunakan nilai piksel 0,0 dari *channel* R yang bernilai 145 adalah sebagai berikut.

$$r = 145 \times \frac{8}{256} = 4$$

Perhitungan kuantisasi dilakukan pada semua piksel pada masing-masing *channel*. Hasil perhitungan kuantisasi secara keseluruhan pada masing-masing *channel* ditunjukkan pada Tabel 4.4, Tabel 4.5, dan Tabel 4.6. Pada tabel, posisi piksel secara horizontal dilambangkan dengan huruf x dan posisi piksel secara vertikal dilambangkan dengan huruf y.

Tabel 4.4 Tabel hasil kuantisasi untuk *channel* R

x,y	0	1	2	3	4	5	6	7	8	9
0	4	4	4	4	4	3	3	3	3	3
1	4	4	3	3	3	3	3	3	3	3
2	4	3	3	3	3	3	3	3	4	3
3	4	3	3	3	3	3	3	3	3	3
4	3	3	4	3	4	3	3	3	3	3
5	4	4	3	3	3	3	3	3	3	3
6	3	3	3	4	3	3	3	3	3	3
7	4	3	3	3	3	3	3	3	3	3
8	3	3	3	3	3	3	3	3	3	3
9	3	3	3	3	3	3	3	3	3	3

Tabel 4.5 Tabel hasil kuantisasi untuk *channel* G

x,y	0	1	2	3	4	5	6	7	8	9
0	3	2	2	2	3	2	2	2	2	2
1	2	3	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	2	2	2
4	2	2	2	2	2	2	2	2	2	2
5	2	2	2	2	2	2	2	2	2	2
6	2	2	2	2	2	2	2	2	2	2
7	2	2	2	2	2	2	2	2	2	2
8	2	2	2	2	1	1	1	2	1	2
9	2	2	2	2	2	2	1	1	1	2



Tabel 4.6 Tabel hasil kuantisasi untuk *channel B*

x,y	0	1	2	3	4	5	6	7	8	9
0	2	1	1	1	2	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	2	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Selanjutnya dilakukan perhitungan histogram berdasarkan citra yang telah dikuantisasi. Pada penelitian ini, histogram yang dibentuk adalah histogram dari masing-masing *channel*. Histogram dari Gambar 4.8 yang telah dikuantisasi ditunjukkan pada Tabel 4.7, Tabel 4.8, dan Tabel 4.9. Pada tabel, *i* menunjukkan indeks dari *bin* dan $H[i]$ adalah nilai dari *bin* *i*.

Tabel 4.7 Tabel histogram untuk *channel R*

i	0	1	2	3	4	5	6	7
H[i]	0	97	3	0	0	0	0	0

Tabel 4.8 Tabel histogram untuk *channel G*

i	0	1	2	3	4	5	6	7
H[i]	0	7	90	3	0	0	0	0

Tabel 4.9 Tabel histogram untuk *channel B*

i	0	1	2	3	4	5	6	7
H[i]	0	0	0	84	16	0	0	0

Selanjutnya dilakukan normalisasi pada histogram dengan menggunakan Persamaan 2.9. Berikut adalah contoh dari perhitungan normalisasi dengan menggunakan *bin* 1 dari *channel R* yang bernilai 97.

$$H'[97] = \frac{97}{100} = 0,97$$

Hasil normalisasi histogram secara keseluruhan ditunjukkan pada Tabel 4.10, Tabel 4.11, dan Tabel 4.12. Setiap *bin* pada setiap *channel* dijadikan sebagai fitur yang telah dinormalisasi dijadikan sebagai fitur dari citra. Pada tabel, *i* menunjukkan indeks dari *bin* dan $H[i]$ adalah nilai dari *bin* *i*.

Tabel 4.10 Tabel hasil normalisasi histogram untuk *channel R*

i	0	1	2	3	4	5	6	7
H[i]	0	0,97	0,03	0	0	0	0	0

Tabel 4.11 Tabel hasil normalisasi histogram untuk *channel G*

i	0	1	2	3	4	5	6	7
H[i]	0	0,07	0,9	0,03	0	0	0	0

Tabel 4.12 Tabel hasil normalisasi histogram untuk *channel B*

i	0	1	2	3	4	5	6	7
H[i]	0	0	0	0,84	0,16	0	0	0

Selain dari nilai setiap *bin* histogram, dilakukan perhitungan *mean* dan deviasi standar pada masing-masing histogram sebagai fitur dari citra dengan menggunakan persamaan 2.10 dan 2.11.

a. *Channel R*

- $mean = \frac{0*0+1*0,97+2*0,03+3*0+4*0+5*0+6*0+7*0}{0+0,97+0,03+0+0+0+0+0} = \frac{1,06}{1} = 1,06$

- deviasi standar

$$= \sqrt{\frac{0 * (0 - 1,06)^2 + 0,97 * (1 - 1,06)^2 + \dots + 0 * (7 - 1,06)^2}{0 + 0,97 + \dots + 0}} = \sqrt{\frac{0,029}{1}} = 0,171$$

b. *Channel G*

- $mean = \frac{0*0+1*0,07+2*0,9+3*0,03+4*0+5*0+6*0+7*0}{0+0,07+0,9+0,03+0+0+0+0} = \frac{1,96}{1} = 1,96$

- deviasi standar

$$= \sqrt{\frac{0 * (0 - 1,96)^2 + 0,07 * (1 - 1,96)^2 + \dots + 0 * (7 - 1,96)^2}{0 + 0,07 + \dots + 0}} = \sqrt{\frac{0,098}{1}} = 0,314$$

c. *Channel B*

- $mean = \frac{0*0+1*0+2*0+3*0,84+4*0,16+5*0+6*0+7*0}{0+0+0+0,84+0,16+0+0+0} = \frac{3,16}{1} = 3,16$

- deviasi standar

$$= \sqrt{\frac{0 * (0 - 3,16)^2 + 0 * (1 - 3,16)^2 + \dots + 0 * (7 - 3,16)^2}{0 + 0 + \dots + 0}} = \sqrt{\frac{0,134}{1}} = 0,367$$



4.2.3 Ekstraksi Fitur Tekstur *Gray Level Cooccurrence Matrix*

Berikut adalah perhitungan manual dari ekstraksi fitur tekstur dengan menggunakan *Gray Level Cooccurrence Matrix*.

1. Mengubah citra ke dalam bentuk *grayscale*. Berdasarkan Persamaan 2.1 dan Persamaan 2.2, berikut adalah contoh perhitungan *grayscale* dengan menggunakan Gambar 4.8.
 - a. Melakukan *gamma correction*

Berikut adalah contoh perhitungan *gamma correction* pada *channel R* pada piksel [0,0].

$$R'[0,0] = 255 \times \frac{R[0,0]^{\frac{1}{2,2}}}{255} = 255 \times \frac{145^{\frac{1}{2,2}}}{255} = 197$$

2. *Gamma correction* dilakukan pada setiap piksel pada masing-masing *channel*. Untuk hasil perhitungan *Gamma correction* pada *channel R*, G, dan B terdapat pada Tabel 4.13, Tabel 4.14, dan Tabel 4.15. Pada tabel, posisi piksel secara horizontal dilambangkan dengan huruf x dan posisi piksel secara vertikal dilambangkan dengan huruf y.

Tabel 4.13 Hasil *Gamma Correction* pada *Channel R*

x,y	0	1	2	3	4	5	6	7	8	9
0	197	194	192	187	197	186	182	180	180	181
1	195	195	186	185	179	185	182	185	181	176
2	190	185	186	179	184	183	184	174	186	173
3	188	186	184	185	182	186	180	177	176	175
4	185	184	188	183	186	182	179	177	175	183
5	188	188	184	180	177	182	178	182	179	182
6	184	184	186	186	176	181	176	182	175	181
7	186	174	178	180	178	177	175	175	177	178
8	183	178	175	175	165	170	169	175	171	173
9	182	179	176	178	175	180	168	165	167	173

Tabel 4.14 Hasil *Gamma Correction* pada *Channel G*

x,y	0	1	2	3	4	5	6	7	8	9
0	166	162	160	154	166	152	149	146	145	147
1	163	164	152	151	144	151	149	152	147	141
2	158	151	152	144	151	148	151	138	153	136
3	155	152	150	151	147	151	146	142	140	139
4	151	150	156	150	154	147	144	141	139	149
5	155	155	150	144	141	148	143	147	144	148
6	150	151	152	153	140	146	140	147	139	146
7	153	137	142	144	142	141	139	139	141	143
8	151	144	141	141	127	134	132	141	135	138
9	150	145	142	144	141	146	131	127	129	137

Tabel 4.15 Hasil Gamma Correction pada Channel B

x,y	0	1	2	3	4	5	6	7	8	9
0	137	132	130	123	142	125	123	119	120	123
1	134	135	120	122	114	124	123	127	123	115
2	130	122	123	114	123	123	127	111	132	111
3	128	125	122	126	122	129	122	116	116	115
4	126	125	131	126	131	125	120	117	117	129
5	134	133	128	122	117	126	119	127	123	128
6	130	129	133	134	118	126	118	129	119	128
7	136	115	123	124	123	122	119	119	122	124
8	135	126	124	122	107	114	111	122	115	118
9	134	129	125	128	124	130	110	105	107	117

3. Melakukan *grayscale*

Berikut adalah contoh perhitungan *grayscale* untuk piksel [0,0] menggunakan Persamaan 2.1.

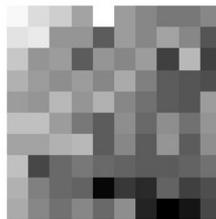
$$grayscale = \frac{1}{3}(197 + 166 + 137) = 166,667 \approx 167$$

Hasil perhitungan *grayscale* secara keseluruhan terdapat pada Tabel 4.16 dan digambarkan dengan Tabel 4.16 dan Gambar 4.9. Pada tabel, posisi piksel secara horizontal dilambangkan dengan huruf x dan posisi piksel secara vertikal dilambangkan dengan huruf y.

Tabel 4.16 Hasil Grayscale

x,y	0	1	2	3	4	5	6	7	8	9
0	167	163	160	155	168	154	151	149	148	150
1	164	165	153	153	145	153	151	155	150	144
2	159	153	154	145	153	151	154	141	157	140
3	157	154	152	154	150	155	149	145	144	143
4	154	153	158	153	157	151	148	145	144	154
5	159	159	154	149	145	152	147	152	148	153
6	155	155	157	158	145	151	145	153	145	152
7	159	142	147	149	147	146	145	145	146	148
8	156	150	147	146	133	139	137	146	140	143
9	155	151	148	150	147	152	136	132	134	142





Gambar 4.9 Citra Hasil Grayscale

4. Menghitung matriks kejadian pada empat sudut (θ), yaitu sudut 0° , 45° , 90° , dan 135° dengan jarak ketetangaan (d) sebesar 1 piksel.
 - a. Sudut 0°

Pada Gambar 4.10 adalah contoh *cooccurrence pixel* atau kejadian ketetangaan pada sudut 0° antara piksel yang bernilai 144 dengan 145 dari Gambar 4.9.

Masing-masing piksel bertetangaan pada citra dihitung jumlah kejadiannya. Nilai *cooccurrence* pada sudut 0° secara keseluruhan disimpan ke dalam bentuk matriks yang ditunjukkan pada Tabel 4.17. Berdasarkan kejadian yang ditunjukkan pada Gambar 4.10, matriks *cooccurrence* pada (144,145) dan (145, 144) akan bernilai 2. Pada tabel, indeks tabel secara horizontal dilambangkan dengan huruf x dan indeks tabel secara vertikal dilambangkan dengan huruf y.

166	162	160	155	167	153	150	148	148	150
163	164	152	152	144	153	150	154	150	143
159	152	153	144	152	150	153	140	157	140
156	153	151	153	150	155	149	144	143	142
153	152	157	152	156	150	147	144	143	153
158	158	153	148	144	151	146	151	147	152
154	154	156	157	144	150	144	152	144	151
158	141	146	148	146	145	144	144	145	147
156	149	146	145	132	139	137	145	140	142
155	150	147	150	146	151	136	131	134	141

Gambar 4.10 Kejadian piksel yang bernilai 144 dengan 145 bertetangaan pada sudut 0°



Tabel 4.17 Cooccurrence matrix sudut 0°

x,y	131	...	144	145	146	...	167
131	0	...	0	0	0	...	0
...
144	0	...	2	2	0	...	0
145	0	...	2	0	2	...	0
146	0	...	0	2	0	...	0
...
167	0	...	0	0	0	...	0

b. Sudut 45°

Pada Gambar 4.11 adalah contoh *cooccurrence pixel* atau kejadian ketetangaan pada sudut 45° antara piksel yang bernilai 144 dengan 145 dari Gambar 4.9.

166	162	160	155	167	153	150	148	148	150
163	164	152	152	144	153	150	154	150	143
159	152	153	144	152	150	153	140	157	140
156	153	151	153	150	155	149	144	143	142
153	152	157	152	156	150	147	144	143	153
158	158	153	148	144	151	146	151	147	152
154	154	156	157	144	150	144	152	144	151
158	141	146	148	146	145	144	144	145	147
156	149	146	145	132	139	137	145	140	142
155	150	147	150	146	151	136	131	134	141

Gambar 4.11 Kejadian piksel yang bernilai 144 dengan 145 bertetangaan pada sudut 45°

Masing-masing piksel bertetangaan pada citra dihitung jumlah kejadiannya. Nilai *cooccurrence* pada sudut 45° secara keseluruhan disimpan ke dalam bentuk matriks yang ditunjukkan pada Tabel 4.18. Berdasarkan kejadian yang ditunjukkan pada Gambar 4.11, matriks *cooccurrence* pada (144,145) dan (145, 144) akan bernilai 1. Pada tabel, indeks tabel secara horizontal dilambangkan dengan huruf x dan indeks tabel secara vertikal dilambangkan dengan huruf y.



Tabel 4.18 Cooccurrence matrix sudut 45°

x,y	131	...	144	145	146	...	167
131	0	...	0	0	0	...	0
...
144	0	...	4	1	1	...	0
145	0	...	1	2	1	...	0
146	0	...	1	1	0	...	0
...
167	0	...	0	0	0	...	0

c. Sudut 90°

Pada Gambar 4.12 adalah contoh *cooccurrence pixel* atau kejadian ketetanggaan pada sudut 90° antara piksel yang bernilai 144 dengan 145 dari Gambar 4.9.

166	162	160	155	167	153	150	148	148	150
163	164	152	152	144	153	150	154	150	143
159	152	153	144	152	150	153	140	157	140
156	153	151	153	150	155	149	144	143	142
153	152	157	152	156	150	147	144	143	153
158	158	153	148	144	151	146	151	147	152
154	154	156	157	144	150	144	152	144	151
158	141	146	148	146	145	144	144	145	147
156	149	146	145	132	139	137	145	140	142
155	150	147	150	146	151	136	131	134	141

Gambar 4.12 Kejadian piksel yang bernilai 144 dengan 145 bertetanggaan pada sudut 90°

Masing-masing piksel bertetanggaan pada citra dihitung jumlah kejadiannya. Nilai *cooccurrence* pada sudut 90° secara keseluruhan disimpan ke dalam bentuk matriks yang ditunjukkan pada Tabel 4.19. Berdasarkan kejadian yang ditunjukkan pada Gambar 4.12, matriks *cooccurrence* pada (144,145) dan (145, 144) akan bernilai 2. Pada tabel, indeks tabel secara horizontal dilambangkan dengan huruf x dan indeks tabel secara vertikal dilambangkan dengan huruf y.



Tabel 4.19 *Cooccurrence matrix* sudut 90°

x,y	131	...	144	145	146	...	167
131	0	...	0	0	0	...	0
...
144	0	...	6	2	2	...	0
145	0	...	2	0	0	...	0
146	0	...	2	0	2	...	0
...
167	0	...	0	0	0	...	0

d. Sudut 135°

Pada Gambar 4.13 adalah contoh *cooccurrence pixel* atau kejadian ketetangaan pada sudut 135° antara piksel yang bernilai 144 dengan 145 dari Gambar 4.9.

166	162	160	155	167	153	150	148	148	150
163	164	152	152	144	153	150	154	150	143
159	152	153	144	152	150	153	140	157	140
156	153	151	153	150	155	149	144	143	142
153	152	157	152	156	150	147	144	143	153
158	158	153	148	144	151	146	151	147	152
154	154	156	157	144	150	144	152	144	151
158	141	146	148	145	145	144	144	145	147
156	149	146	145	132	139	137	145	140	142
155	150	147	150	146	151	136	131	134	141

Gambar 4.13 Kejadian piksel yang bernilai 144 dengan 145 bertetangaan pada sudut 135°

Tabel 4.20 *Cooccurrence matrix* sudut 135°

x,y	131	...	144	145	146	...	167
131	0	...	0	0	0	...	0
...
144	0	...	2	2	0	...	0
145	0	...	2	0	2	...	0
146	0	...	0	2	0	...	0
...
167	0	...	0	0	0	...	0

Masing-masing piksel bertetangaan pada citra dihitung jumlah kejadiannya. Nilai *cooccurrence* secara keseluruhan disimpan ke dalam bentuk matriks yang ditunjukkan pada Tabel 4.20. Berdasarkan kejadian yang ditunjukkan pada Gambar 4.13, matriks *cooccurrence* pada (144,145) dan (145,



144) akan bernilai 2. Pada tabel, indeks tabel secara horizontal dilambangkan dengan huruf x dan indeks tabel secara vertikal dilambangkan dengan huruf y.

- Melakukan normalisasi terhadap *cooccurrence matrix*. Proses normalisasi dilakukan dengan membagi tiap-tiap nilai yang terdapat pada *cooccurrence matrix* dengan total dari semua nilai dalam *cooccurrence matrix*. Berikut adalah contoh perhitungan normalisasi *cooccurrence matrix* dengan menggunakan nilai dari (144,144) pada sudut 0° yang bernilai 2.

$$P_{144,144} = \frac{2}{(0 + \dots + 2 + \dots + 0)} = \frac{2}{180} = 0,012$$

Normalisasi dilakukan terhadap setiap nilai pada setiap sudut. Untuk hasil perhitungan secara menyeluruh terdapat pada Tabel 4.21, Tabel 4.22, Tabel 4.23, dan Tabel 4.24. Pada tabel, indeks tabel secara horizontal dilambangkan dengan huruf x dan indeks tabel secara vertikal dilambangkan dengan huruf y.

Tabel 4.21 Hasil normalisasi *cooccurrence matrix* sudut 0°

x,y	131	...	144	145	146	...	167
131	0	...	0	0	0	...	0
...
144	0	...	0,01	0,01	0	...	0
145	0	...	0,01	0	0,01	...	0
146	0	...	0	0,01	0	...	0
...
167	0	...	0	0	0	...	0

Tabel 4.22 Hasil normalisasi *cooccurrence matrix* sudut 45°

x,y	131	...	144	145	146	...	167
131	0	...	0	0	0	...	0
...
144	0	...	0,02	0,01	0,01	...	0
145	0	...	0,01	0,01	0,01	...	0
146	0	...	0,01	0,01	0	...	0
...
167	0	...	0	0	0	...	0



Tabel 4.23 Hasil normalisasi *cooccurrence matrix* sudut 90°

x,y	131	...	144	145	146	...	167
131	0	...	0	0	0	...	0
...
144	0	...	0,04	0,01	0,01	...	0
145	0	...	0,01	0	0	...	0
146	0	...	0,01	0	0,01	...	0
...
167	0	...	0	0	0	...	0

Tabel 4.24 Hasil normalisasi *cooccurrence matrix* sudut 135°

x,y	131	...	144	145	146	...	167
131	0	...	0	0	0	...	0
...
144	0	...	0,01	0,01	0	...	0
145	0	...	0,01	0	0,01	...	0
146	0	...	0	0,01	0	...	0
...
167	0	...	0	0	0	...	0

6. Melakukan perhitungan *energy*, *entropy*, *contrast*, dan *homogeneity* pada masing-masing *cooccurrence matrix* yang telah dinormalisasi dengan menggunakan Persamaan 2.11, Persamaan 2.12, Persamaan 2.13, dan Persamaan 2.14.

a. Sudut 0°

- $Energy = 0^2 + \dots + 0,011^2 + 0,006^2 + \dots + 0 = 0,009$
- $Entropy = -((0 * 0) + \dots + (0,011 * -2) + (0,006 * -2,2) + \dots + (0 * 0))$
 $= -(0 + \dots + (-0,02) + (-0,02) + \dots + 0)$
 $= 2,059$
- $Contrast = (131 - 131)^2 * 0 + \dots + (144 - 144)^2 * 0,011 + (144 - 145)^2 * 0,006 + \dots + (167 - 167)^2 * 0$
 $= 0 + \dots + 0 + 0,006 + \dots + 0$
 $= 42,589$
- $Homogeneity = \frac{0}{1+(131-1)^2} + \dots + \frac{0,011}{1+(144-144)^2} + \frac{0,006}{1+(144-145)^2} + \dots + \frac{0}{1+(167-167)^2}$
 $= 0 + \dots + 0,011 + 0,006 + \dots + 0$



$$= 0,171$$

b. Sudut 45°

- *Energy* = $0^2 + \dots + 0,024^2 + 0,006^2 + \dots + 0 = 0,011$

- *Entropy*
 $= -((0 * 0) + \dots + (0,024 * -1,7) + (0,006 * -2) + \dots + (0 * 0))$
 $= -(0 + \dots + (-0,039) + (-0,013) + \dots + 0)$
 $= 2,033$

- *Contrast*
 $= (131 - 131)^2 * 0 + \dots + (144 - 144)^2 * 0,024 + (144 - 145)^2 * 0,006 + \dots + (167 - 167)^2 * 0$
 $= 0 + \dots + 0 + 0,006 + \dots + 0$
 $= 45,074$

- *Homogeneity*
 $= \frac{0}{1+(131-131)^2} + \dots + \frac{0,024}{1+(144-144)^2} + \frac{0,006}{1+(144-145)^2} + \dots + \frac{0}{1+(167-167)^2}$
 $= 0 + \dots + 0,024 + 0,003 + \dots + 0$
 $= 0,241$

c. Sudut 90°

- *Energy* = $0^2 + \dots + 0,033^2 + 0,011^2 + \dots + 0 = 0,010$

- *Entropy*
 $= -((0 * 0) + \dots + (0,033 * -1,48) + (0,011 * -1,96) + \dots + (0 * 0))$
 $= -(0 + \dots + (-0,049) + (-0,022) + \dots + 0)$
 $= 2,072$

- *Contrast*
 $= (131 - 131)^2 * 0 + \dots + (144 - 144)^2 * 0,033 + (144 - 145)^2 * 0,011 + \dots + (167 - 167)^2 * 0$
 $= 0 + \dots + 0 + 0,011 + \dots + 0$
 $= 41,989$

- *Homogeneity*
 $= \frac{0}{1 + (131 - 131)^2} + \dots + \frac{0,033}{1 + (144 - 144)^2} + \frac{0,011}{1 + (144 - 145)^2}$
 $+ \dots + \frac{0}{1 + (167 - 167)^2}$
 $= 0 + \dots + 0,033 + 0,005 + \dots + 0$
 $= 0,218$

d. Sudut 135°

- *Energy* = $0^2 + \dots + 0,012^2 + 0,012^2 + \dots + 0 = 0,009$

- *Entropy*
 $= -((0 * 0) + \dots + (0,012 * -1,9) + (0,012 * -1,9) + \dots + (0 * 0))$
 $= -(0 + \dots + (-0,023) + (-0,023) + \dots + 0)$
 $= 2,079$



- *Contrast*

$$= (131 - 131)^2 * 0 + \dots + (144 - 144)^2 * 0,012 + (144 - 145)^2 * 0,012 + \dots + (167 - 167)^2 * 0$$

$$= 0 + \dots + 0 + 0,012 + \dots + 0$$

$$= 47,592$$
- *Homogeneity*

$$= \frac{0}{1 + (131 - 131)^2} + \dots + \frac{0,012}{1 + (144 - 144)^2} + \frac{0,012}{1 + (144 - 145)^2}$$

$$+ \dots + \frac{0}{1 + (167 - 167)^2}$$

$$= 0 + \dots + 0,012 + 0,006 + \dots + 0$$

$$= 0,162$$

4.2.4 Seleksi Fitur *Information Gain*

Berikut adalah perhitungan manual dari seleksi fitur *Information Gain* dengan menggunakan 9 fitur dari 55 fitur yang ada, yaitu *mean*, standar deviasi, dan *skewness* dari masing-masing *channel*. Pada perhitungan manual ini digunakan 30 data latih yang diambil secara acak dari total 248 yang digunakan pada penelitian ini. Pada data latih terdapat delapan kelas, yaitu Donat, Indomie Goreng, Mie Gepeng, Telur Ceplok, Telur Dadar, Ayam Crispy, Rendang, dan Nasi Kuning. Fitur dan data yang digunakan pada perhitungan manual ini terdapat pada Tabel 4.26.

Selanjutnya adalah mengubah data tunggal dari data latih menjadi data distribusi kelompok/kategori agar dapat diproses oleh *Information Gain*. Pada setiap atribut/fitur dilakukan pencarian nilai maksimum dan minimum masing-masing untuk mencari nilai jangkauan. Selanjutnya menghitung banyak kategori yang digunakan yang didapatkan dari perhitungan menggunakan jumlah data secara keseluruhan. Setelah didapatkan nilai jangkauan dan banyaknya kategori, selanjutnya adalah menghitung interval yang merupakan hasil pembagian dari jangkauan dengan banyaknya kategori. Interval antar satu atribut dan yang lainnya berbeda-beda sesuai dengan nilai maksimal dan minimal pada atribut tersebut. Tabel 4.25 menunjukkan contoh interval yang digunakan pada atribut A1 untuk mengubah data tunggal pada A1 menjadi data berkelompok.

Tabel 4.25 Tabel interval dari atribut A1 dan kategorinya

Mean dari channel B	Kategori
0 - 7,466851	0
7,466852 - 12,86377	1
12,86378 - 18,26069	2
18,26070 - 23,65761	3
23,65762 - 29,05453	4
29,05454 - 34,45145	5
34,45146 - 39,84837	6
39,84838 - 45,2453	7

Tabel 4.26 Data latih

NO	A1	A2	A3	A4	A5	A6	A7	A8	A9	KELAS
1	9,333507	19,25044	24,70482	20,92144	40,34326	44,54858	36,03679	68,50475	73,44862	1
2	14,53354	27,65132	32,01583	26,67825	47,81073	50,49856	40,66281	72,1141	74,87453	1
3	18,45797	28,20822	32,45068	34,53611	47,27588	43,00002	56,98443	76,12269	64,56971	1
4	7,331493	16,35073	22,30206	22,70101	42,14576	45,56603	38,13875	69,5423	73,17805	1
5	9,18776	20,71988	27,54683	18,54356	38,76069	45,33565	31,47155	64,5146	72,81619	1
6	23,30655	32,94284	37,66499	57,95833	63,08899	45,4501	81,89201	84,0247	44,28495	4
7	21,47877	31,99184	36,77808	53,46573	62,14483	48,609	75,36108	83,14338	53,41165	4
8	17,88175	32,52653	42,63812	42,80453	58,47388	53,88798	60,56627	78,282	64,01571	4
9	15,60564	28,73311	37,1143	39,07385	56,3739	53,61483	56,17134	77,32243	67,25104	4
10	17,80559	26,92878	33,68449	56,59927	60,66006	43,27433	86,59236	86,7102	40,60831	5
11	13,90486	27,12196	38,45613	39,03663	53,83061	50,66274	60,96385	78,81539	64,61657	5
12	17,43181	29,0219	38,14906	49,14854	58,11199	47,98156	76,72819	84,4778	55,19335	5
13	18,28306	39,76102	49,59476	23,89026	46,93676	54,17811	29,61618	56,46251	63,05287	6
14	11,03859	29,44583	39,52297	18,93578	42,34144	50,24653	25,68476	57,3901	67,78013	6
15	4,450955	20,24475	33,47087	8,702639	31,57838	46,00785	12,04328	43,8765	63,86872	6

16	12,5729	33,00406	44,26316	18,03451	42,91365	53,75411	23,49589	54,49167	66,47259	6
17	2,693663	16,13721	30,56936	6,127795	26,63315	42,35499	8,965191	38,23021	59,05561	6
18	42,732	49,19422	49,61336	76,77432	66,25916	30,004	102,1124	82,513	-47,9769	7
19	26,11806	32,41961	36,27248	69,19156	64,60719	31,35533	97,96092	87,07574	-41,7627	7
20	17,73179	28,65892	37,9246	52,80451	61,57459	48,247	77,65403	86,74636	56,42049	7
21	28,58667	36,52953	36,17179	53,2576	60,72094	49,08688	80,74875	85,26677	55,29346	8
22	16,44453	28,1543	31,68383	32,25835	49,2712	49,88682	52,56439	74,88737	68,84645	8
23	11,34703	24,64839	31,23443	21,13132	40,80947	46,98918	34,60457	62,41687	67,37635	8
24	20,01696	30,63218	31,62719	26,72837	39,75918	38,18121	39,21276	57,18302	51,81339	9
25	8,443802	19,53077	26,26027	11,97163	26,57019	32,79811	18,40295	39,84208	46,5496	9
26	4,144688	14,99544	25,46974	6,073438	20,15022	30,7459	9,344566	29,26551	41,06769	9
27	4,948559	15,42573	24,64975	7,489219	21,33912	30,75176	11,99172	32,01943	42,3341	9
28	18,0062	35,20186	44,50332	81,01986	69,49924	-32,9063	95,89436	80,17243	-46,0097	21
29	10,79415	24,18318	35,46711	75,47299	66,96489	19,16263	95,29561	81,10733	-42,5754	21
30	11,74983	24,35882	39,01673	111,6011	61,9555	-55,8601	134,5451	69,27706	-69,3694	21

Keterangan dari Tabel 4.26:

A1 = Mean dari channel B

A6 = Skewness dari channel G

A2 = Standar deviasi dari channel B

A7 = Mean dari channel R

A3 = Skewness dari channel B

A8 = Standar deviasi dari channel R

A4 = Mean dari channel G

A9 = Skewness dari channel R

A5 = Standar deviasi dari channel G

KELAS = Kelas dari citra data latih

Pengubahan data tunggal menjadi data kelompok ini dilakukan pada setiap atribut. Hasil dari pengubahan data tunggal pada atribut menjadi data berkelompok secara keseluruhan pada data latih ditunjukkan pada Tabel 4.27.

Tabel 4.27 Data latih yang telah diubah

NO	A1	A2	A3	A4	A5	A6	A7	A8	A9	KELAS
1	1	1	1	1	3	7	1	5	7	1
2	2	3	3	1	4	7	2	6	7	1
3	3	3	3	2	4	7	3	6	7	1
4	0	1	0	1	3	7	1	5	7	1
5	1	2	2	1	3	7	1	5	7	1
6	3	4	4	3	6	7	4	7	6	4
7	3	4	4	3	6	7	4	7	6	4
8	2	4	6	2	6	7	3	6	7	4
9	2	3	4	2	5	7	2	6	7	4
10	2	3	3	3	6	7	4	7	6	5
11	2	3	5	2	5	7	3	6	7	5
12	2	3	4	3	6	7	4	7	6	5
13	3	6	7	1	4	7	1	4	7	6
14	1	3	5	1	3	7	1	4	7	6
15	0	2	3	0	2	7	0	2	7	6
16	1	4	6	1	4	7	1	4	7	6
17	0	1	2	0	1	7	0	2	7	6
18	7	7	7	5	7	6	5	7	1	7
19	4	4	4	4	7	6	5	7	1	7
20	2	3	4	3	6	7	4	7	6	7
21	4	5	4	3	6	7	4	7	6	8
22	2	3	3	2	4	7	2	6	7	8
23	1	2	3	1	3	7	1	4	7	8



NO	A1	A2	A3	A4	A5	A6	A7	A8	A9	KELAS
24	3	4	3	1	3	6	2	4	6	9
25	1	1	1	0	1	6	0	2	6	9
26	0	0	1	0	0	6	0	1	6	9
27	0	1	1	0	1	6	0	1	6	9
28	2	5	6	5	7	1	5	7	1	21
29	1	2	4	4	7	5	5	7	1	21
30	1	2	5	7	6	0	7	5	0	21

Setelah data diubah menjadi data berkelompok, maka perhitungan *Information Gain* dapat dilakukan. Langkah pertama yang dilakukan adalah menghitung *entropy* dari kelas yang terdapat pada data latih dengan menggunakan Persamaan 2.15. Berikut adalah perhitungan dari *entropy* kelas.

$$\begin{aligned}
 H(C) &= -(P(1) \log_2 P(1) + P(4) \log_2 P(4) + P(5) \log_2 P(5) + \dots + \\
 &\quad P(21) \log_2 P(21)) \\
 &= -\left(\frac{5}{30} \log_2 \frac{5}{30} + \frac{4}{30} \log_2 \frac{4}{30} + \frac{3}{30} \log_2 \frac{3}{30} + \dots + \frac{3}{30} \log_2 \frac{3}{30}\right) \\
 &= -((-0,431) + (-0,388) + (-0,332) + \dots + (-0,332)) \\
 &= 2,9656
 \end{aligned}$$

Selanjutnya adalah menghitung *entropy* dari masing-masing kategori pada atribut dengan menggunakan Persamaan 2.16. Berikut adalah contoh perhitungan *entropy* dengan menggunakan atribut A1.

$$\begin{aligned}
 H(C|A1) &= -(P(0) * P(1|0) * \log_2 P(1|0) + P(0) * P(4|0) * \\
 &\quad \log_2 P(4|0) + \dots + P(4) * P(7|4) * \log_2 P(7|4) + \dots + P(7) * \\
 &\quad P(21|7) * \log_2 P(21|7)) \\
 &= -\left(\frac{5}{30} * \frac{1}{5} * \log_2 \frac{1}{5} + \frac{5}{30} * \frac{0}{5} * \log_2 \frac{0}{5} + \dots + \frac{2}{30} * \frac{1}{2} * \log_2 \frac{1}{2} + \dots + \frac{1}{30} * \right. \\
 &\quad \left. \frac{0}{1} * \log_2 \frac{0}{1}\right) \\
 &= -(-0,0074 + 0 + \dots + (-0,033) + \dots + 0) \\
 &= 1,966
 \end{aligned}$$

Setelah mendapatkan *entropy* dari atribut, dilakukan perhitungan *Information Gain* dengan mengurangi *entropy* dari kelas dengan *entropy* dari atribut seperti pada Persamaan 2.17. Berikut adalah contoh perhitungan *Information Gain* dari atribut A1.

$$IG_{A1} = H(C) - H(C|A1) = 2,9656 - 1,966 = 0,999$$

Hasil keseluruhan *Information Gain* pada atribut-atribut yang digunakan pada data latih terdapat pada Tabel 4.28.



Langkah selanjutnya adalah memilih n atribut yang terbesar sebagai fitur yang akan digunakan pada tahap *similarity measure*. Pada perhitungan manual ini, lima atribut akan digunakan pada tahap *similarity measure*. Fitur yang akan digunakan adalah lima atribut dengan *gain* terbesar, yaitu A7, A5, A4, A8, dan A3.

Tabel 4.28 Hasil *Information Gain* pada data latih

Atribut	Gain
A1	0,999
A2	0,124
A3	1,393
A4	1,441
A5	1,486
A6	1,065
A7	1,523
A8	1,405
A9	1,079

4.2.5 *Similarity Measure*

Pada tahap ini dilakukan perhitungan kesamaan antara data uji dengan data latih dengan menggunakan *Euclidean distance* untuk mencari data yang paling relevan dengan data uji. Fitur yang digunakan pada perhitungan manual *similarity measure* ini adalah fitur yang didapatkan dari seleksi fitur *Information Gain*, antara lain fitur A7 (*Mean* dari *channel R*), A5 (*Standar deviasi* dari *channel G*), A4 (*Mean* dari *channel G*), A8 (*deviasi standar* dari *channel R*), dan A3 (*Skewness* dari *channel B*).



Gambar 4.14 Citra data uji

Citra dari data uji terdapat pada Gambar 4.14 Citra data uji dan nilai dari fitur pada data uji yang digunakan pada perhitungan manual ini terdapat pada Tabel 4.29.

Tabel 4.29 Nilai fitur dari data uji

A3	A4	A5	A7	A8
37,379	57,756	62,859	81,571	83,902

Berikut adalah contoh perhitungan *similarity measure* dengan menggunakan *Euclidean distance* berdasarkan Persamaan 2.18. Contoh

perhitungan dilakukan antara data uji (x) dengan data latih nomor enam yang terdapat pada Tabel 4.26.

$$d(x, 6) = \sqrt{(37,379 - 37,665)^2 + (57,756 - 57,958)^2 + (62,859 - 63,089)^2 + (81,571 - 81,892)^2 + (83,902 - 84,043)^2}$$

$$= \sqrt{0,082 + 0,041 + 0,053 + 0,103 + 0,02} = \sqrt{0,298} = 0,546$$

Hasil dari perhitungan *Euclidean distance* antara data uji dengan data latih secara keseluruhan dapat dilihat pada Tabel 4.30.

Tabel 4.30 Hasil *Euclidean distance*

Data Latih ke- n	Jarak
1	65,83933
2	55,07699
3	38,35756
4	63,06953
5	71,42018
6	0,541969
7	7,64305
8	27,26201
9	32,857
10	7,274806
11	29,72566
12	10,99999
13	70,72383
14	75,88629
15	99,17725
16	79,12453
17	106,6849
18	30,77039
19	20,34096
20	7,063829
21	5,366833
22	42,3065
23	67,32578
24	63,54151
25	97,30463
26	113,3083
27	109,2967
28	29,24251
29	23,03433
30	76,96068

Langkah selanjutnya adalah mengembalikan r data yang memiliki nilai jarak terkecil sebagai hasil dari sistem. Variabel r adalah jumlah hasil pengembalian dari sistem. Pada perhitungan manual ini, nilai r yang digunakan adalah $r = 10$. Hasil pengembalian dari sistem terdapat pada Tabel 4.31. Berdasarkan Tabel 4.31, data yang dikembalikan oleh sistem adalah data latih ke-6, 21, 20, 7, 12, 19, 29, 8, dan 28.

Tabel 4.31 Hasil pengembalian dari sistem

Peringkat	Data Latih ke- n	Jarak	Citra
1	6	0,541969	
2	21	5,366833	
3	20	7,063829	
4	10	7,274806	
5	7	7,64305	
6	12	10,99999	
7	19	20,34096	
8	29	23,03433	

Tabel 4.31 Hasil pengembalian dari sistem

Peringkat	Data Latih ke- n	Jarak	Citra
9	8	27,26201	
10	28	29,24251	

4.2.6 Perancangan Skenario Pengujian

Untuk mencapai tujuan dari penelitian ini dilakukan pengujian terhadap algoritme yang digunakan. Perhitungan evaluasi dari hasil pengujian dilakukan dengan menggunakan *Mean Average Precision* (MAP) karena hasil dari sistem ini berupa pengembalian yang memiliki peringkat (*ranked retrieval*). Pada penelitian ini terdapat empat pengujian yang dilakukan.

Skenario pengujian pertama yang dilakukan pada penelitian ini adalah pengujian pengaruh jumlah pengembalian (r) terhadap nilai MAP. Pada pengujian ini dilakukan perbandingan nilai MAP yang didapatkan dari sistem dengan jumlah pengembalian (r) sebanyak 5, 10, 15, dan 20. Pengujian ini dilakukan untuk mengetahui jumlah pengembalian yang dapat memberikan nilai MAP yang lebih tinggi yang artinya sistem memberikan hasil yang lebih baik. Selain itu, pengujian ini juga dilakukan untuk memfokuskan pengujian pada subbab-subbab berikutnya pada jumlah pengembalian atau nilai r yang terbaik. Perancangan pengujian ini terdapat pada Tabel 4.32.

Tabel 4.32 Skenario pengujian pengaruh jumlah pengembalian

		r			
		5	10	15	20
Fitur	Warna				
	Tekstur				
	Kombinasi				

Skenario pengujian kedua yang dilakukan adalah pengujian pengaruh seleksi fitur terhadap sistem yang menggunakan fitur warna saja. Pada pengujian ini dilakukan perbandingan antara nilai MAP yang didapatkan dari sistem dengan seleksi fitur dan dengan tanpa seleksi fitur. Fitur warna memiliki jumlah fitur sebanyak 39. Nilai MAP dari jumlah fitur tersebut dibandingkan dengan fitur yang

didapatkan dari proses seleksi fitur, yaitu 5, 10, 15, 20, 25, 30, dan 35 fitur yang memiliki nilai *gain* tertinggi. Jumlah fitur yang digunakan diberi nama variabel n . Pengujian dilakukan dengan jumlah pengembalian atau nilai r yang memberikan nilai MAP terbaik yang didapatkan dari pengujian nilai r . Perancangan pengujian ini terdapat pada Tabel 4.33.

Tabel 4.33 Skenario pengujian pengaruh seleksi fitur terhadap sistem dengan fitur warna

n	MAP
5	
10	
15	
20	
25	
30	
35	
39	

Skenario pengujian ketiga yang dilakukan pada penelitian ini adalah pengujian pengaruh seleksi fitur terhadap sistem dengan fitur tekstur. Pada pengujian ini dilakukan perbandingan antara nilai MAP yang didapatkan dari sistem dengan seleksi fitur dan dengan tanpa seleksi fitur. Fitur tekstur memiliki jumlah fitur sebanyak 16. Nilai MAP dari jumlah fitur tersebut dibandingkan dengan fitur yang didapatkan dari proses seleksi fitur, yaitu 5 dan 10 fitur yang memiliki nilai *gain* tertinggi. Jumlah fitur yang digunakan diberi nama variabel n . Pengujian dilakukan dengan jumlah pengembalian atau nilai r yang memberikan nilai MAP terbaik yang didapatkan dari pengujian nilai r . Perancangan pengujian ini terdapat pada Tabel 4.34.

Tabel 4.34 Skenario pengujian pengaruh seleksi fitur terhadap sistem dengan fitur tekstur

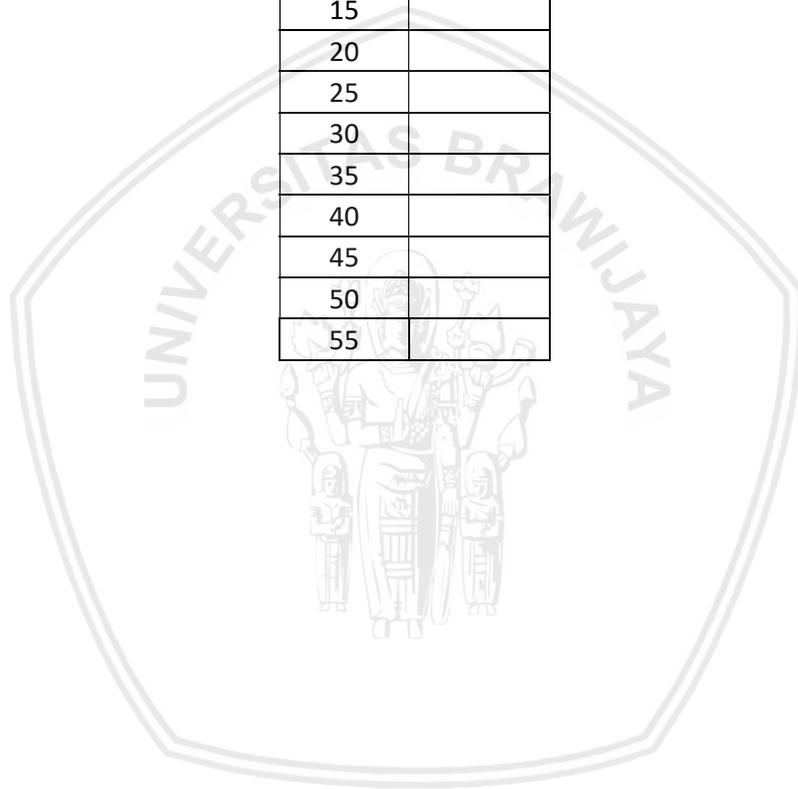
n	MAP
5	
10	
16	

Skenario pengujian keempat yang dilakukan pada penelitian ini adalah pengujian pengaruh seleksi fitur terhadap sistem dengan kombinasi fitur warna dan tekstur. Pada pengujian ini dilakukan perbandingan antara nilai MAP yang didapatkan dari sistem dengan seleksi fitur dan dengan tanpa seleksi fitur. Kombinasi antara fitur warna dan tekstur menghasilkan jumlah fitur sebanyak 35.

Nilai MAP dari jumlah fitur tersebut dibandingkan dengan fitur yang didapatkan dari proses seleksi fitur, yaitu 5, 10, 15, 20, 25, 30, 35, 40, 45, dan 50 fitur yang memiliki nilai *gain* tertinggi. Jumlah fitur yang digunakan diberi nama variabel n . Pengujian dilakukan dengan jumlah pengembalian atau nilai r yang memberikan nilai MAP terbaik yang didapatkan dari pengujian nilai r .

Tabel 4.35 Skenario pengujian pengaruh seleksi fitur terhadap sistem dengan kombinasi fitur warna dan tekstur

n	MAP
5	
10	
15	
20	
25	
30	
35	
40	
45	
50	
55	



BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi dari sistem berdasarkan perancangan yang telah dibuat sebelumnya. Penjelasan yang terdapat pada bab ini adalah batasan implementasi dan kode sumber dari program yang telah dibuat.

5.1 Batasan Implementasi

Batasan dari implementasi sistem ini antara lain:

1. Metode *preprocessing* yang digunakan adalah *resizing*, *grayscale*, *Gaussian filtering*, *thresholding*, dan *closing*.
2. Metode ekstraksi fitur yang digunakan adalah *color moments*, *color histogram*, dan GLCM. Proses ekstraksi fitur menghasilkan fitur sebanyak 55 fitur.
3. Metode seleksi fitur yang digunakan adalah *Information Gain*.
4. Metode *similarity measure* yang digunakan adalah *Euclidean distance*.

5.2 Implementasi Sistem

Pada subbab ini terdapat kode sumber dari masing-masing metode *preprocessing*, ekstraksi fitur, seleksi fitur dan *similarity measure* dari sistem serta penjelasannya.

5.2.1 Implementasi *Preprocessing*

Implementasi dari tahap *preprocessing* terdapat pada Kode sumber 5.1.

```

1 def preprocessing(image):
2     image = cv2.resize(image, (240, 240))
3     gs = Preprocessing.grays(image)
4     blur = cv2.GaussianBlur(gs, (5, 5), 0)
5     dst, th = cv2.threshold(blur, 180, 255, cv2.THRESH_BINARY_INV)
6     kernel = np.ones((3, 3), np.uint8)
7     th = cv2.dilate(th, kernel, iterations=10)
8     th = cv2.erode(th, kernel, iterations=10)
9     mask = cv2.merge([th, th, th])
10    result = cv2.bitwise_and(image, mask)
11    gs = Preprocessing.grays(result)
12    return result, gs

```

Kode sumber 5.1 Implementasi fungsi *preprocessing()*

Berikut adalah penjelasan untuk Kode sumber 5.1.

1. Baris 1 adalah deklarasi fungsi *preprocessing()* dengan parameter variabel *image* yang merupakan sebuah citra.
2. Baris 2 adalah proses untuk melakukan *resizing* untuk menyamakan ukuran citra sebelum diproses. Citra dari variabel *image* akan diubah ke ukuran 240 × 240 piksel.
3. Baris 3 adalah proses untuk mengubah citra menjadi *grayscale*.
4. Baris 4 adalah proses *blurring* dengan menggunakan *Gaussian Blur*.
5. Baris 5 adalah proses *thresholding* untuk memisahkan bagian yang akan dihitung pada proses ekstraksi fitur.

6. Baris 6 adalah proses inialisasi *kernel* yang akan digunakan pada tahap dilasi dan erosi.
7. Baris 7 adalah proses dilasi untuk melebarkan bagian *foreground* dari hasil *thresholding*.
8. Baris 8 adalah proses erosi untuk menghilangkan *noise*.
9. Baris 9 adalah proses untuk menggabungkan hasil *thresholding* ke dalam bentuk 3 dimensi untuk diterapkan pada 3 *channel*.
10. Baris 10 adalah tahap untuk menerapkan hasil *thresholding* pada citra untuk menciptakan citra tersegmentasi.
11. Baris 11 adalah proses *grayscale* pada citra tersegmentasi.
12. Baris 12 adalah proses untuk mengembalikan hasil dari fungsi *preprocessing()* yaitu berupa citra tersegmentasi dan citra tersegmentasi dalam bentuk *grayscale*.

5.2.2 Implementasi Ekstraksi Fitur

Pada implementasi untuk ekstraksi fitur terbagi menjadi tiga bagian, yaitu implementasi untuk *Color Moments*, implementasi untuk *Color Histogram*, dan implementasi untuk *Gray Level Coocurance Matrix*.

5.2.2.1 Implementasi *Color Moments*

Pada implementasi *Color Moments* terbagi menjadi tiga, yaitu implementasi untuk fungsi *mean()* terdapat pada Kode sumber 5.2, implementasi untuk fungsi *stddev()* terdapat pada Kode sumber 5.3, dan implementasi untuk fungsi *skewness()* terdapat pada Kode sumber 5.4.

```

1 def mean(i):
2     h,w = i.shape
3     total = 0
4     for y in range(h):
5         for x in range(w):
6             total += i[y][x]
7     divide = i.size
8     result = total / divide
9     return result

```

Kode sumber 5.2 Implementasi fungsi *mean()*

Berikut adalah penjelasan untuk Kode sumber 5.2.

1. Baris 1 adalah deklarasi dari fungsi *mean()* dengan parameter nilai dari citra yang disimpan dalam variabel *i*.
2. Baris 2 adalah inialisasi variabel untuk menampung nilai panjang dan lebar dari citra.
3. Baris 3 adalah inialisasi variabel *total* untuk menampung hasil penjumlahan dari setiap nilai dari variabel *i*.
4. Baris 4-8 adalah proses perhitungan *mean*.
5. Baris 9 adalah proses untuk mengembalikan hasil dari fungsi *mean()*

```

1 def stddev(i,e):
2     h, w = i.shape
3     total = 0
4     for y in range(h):
5         for x in range(w):
6             total += ((i[y][x] - e) ** 2)
7     divide = i.size
8     result = total / divide
9     return result ** 0.5

```

Kode sumber 5.3 Implementasi fungsi stddev()

Berikut adalah penjelasan untuk Kode sumber 5.3.

1. Baris 1 adalah deklarasi dari fungsi stddev() dengan parameter nilai dari citra yang disimpan dalam variabel *i* dan nilai *mean* dari citra yang disimpan dalam variabel *e*.
2. Baris 2 adalah inisialisasi variabel untuk menampung nilai panjang dan lebar dari citra.
3. Baris 3 adalah inisialisasi variabel total untuk menampung hasil penjumlahan dari setiap nilai dari variabel *i*.
4. Baris 4-8 adalah proses perhitungan deviasi standar.
5. Baris 9 adalah proses untuk mengembalikan hasil dari fungsi stddev()

```

1 def skewness(i,e):
2     h, w = i.shape
3     total = 0
4     for y in range(h):
5         for x in range(w):
6             total += ((i[y][x] - e) ** 3)
7
8     divide = i.size
9     result = (total) / divide
10    if result < 0 :
11        result = -result
12    return -result ** 0.33
13    return result ** 0.33

```

Kode sumber 5.4 Implementasi fungsi skewness()

Berikut adalah penjelasan untuk Kode sumber 5.4.

1. Baris 1 adalah deklarasi dari fungsi skewness() dengan parameter nilai dari citra yang disimpan dalam variabel *i* dan nilai *mean* dari citra yang disimpan dalam variabel *e*.
2. Baris 2 adalah inisialisasi variabel untuk menampung nilai panjang dan lebar dari variabel *i*.
3. Baris 3 adalah inisialisasi variabel total untuk menampung hasil penjumlahan dari setiap nilai dari variabel *i*.
4. Baris 4-9 adalah proses perhitungan *skewness*.
5. Baris 10-12 adalah proses yang dilakukan pada kondisi jika variabel *result* bernilai negatif.
6. Baris 13 adalah proses untuk mengembalikan hasil dari fungsi skewness()

5.2.2.2 Implementasi *Color Histogram*

Pada implementasi dari *Color Histogram* terbagi menjadi lima, yaitu implementasi fungsi `quantization()` yang terdapat pada Kode sumber 5.5, implementasi fungsi `histogram()` yang terdapat pada Kode sumber 5.6, implementasi fungsi `normalize_hist()` yang terdapat pada Kode sumber 5.7, implementasi fungsi `h_mean()` yang terdapat pada Kode sumber 5.8, dan implementasi fungsi `h_stddev()` yang terdapat pada Kode sumber 5.9.

```

1 def quantization(h, bin):
2     d = 256
3     result = int(h * bin / d)
4     return result

```

Kode sumber 5.5 Implementasi fungsi `quantization()`

Berikut adalah penjelasan untuk Kode sumber 5.5.

1. Baris 1 adalah deklarasi fungsi `quantization()` dengan parameter nilai dari piksel dari citra yang disimpan dalam variabel *h* dan jumlah *bin* yang diinginkan yang disimpan dalam variabel *bin*.
2. Baris 2 adalah inisialisasi variabel *d* yang berisi nilai dari jumlah *bin* semula.
3. Baris 3 adalah proses kuantisasi.
4. Baris 4 adalah proses untuk mengembalikan hasil dari fungsi `quantization()`.

```

1 def histogram(img, n):
2     hist = []
3     h, w = img.shape[0:2]
4     channel = img.shape[2]
5     for c in range(channel):
6         bin = np.zeros(n, dtype=int)
7         i = img[:, :, c]
8         for y in range(h):
9             for x in range(w):
10                bin[i[y, x]] += 1
11        hist.append(bin)
12    return hist

```

Kode sumber 5.6 Implementasi fungsi `histogram()`

Berikut adalah penjelasan untuk Kode sumber 5.6.

1. Baris 1 adalah deklarasi dari fungsi `histogram()` dengan parameter berupa nilai dari citra yang disimpan dalam variabel *img* dan jumlah *bin* yang digunakan yang disimpan dalam variabel *n*.
2. Baris 2 adalah inisialisasi variabel untuk menampung histogram.
3. Baris 3 adalah inisialisasi variabel untuk menampung nilai panjang dan lebar dari variabel *img*.
4. Baris 4 adalah inisialisasi variabel untuk menampung jumlah *channel* dari variabel *img*.
5. Baris 5-11 adalah proses pembentukan histogram dari masing-masing *channel*.
6. Baris 12 adalah proses untuk mengembalikan hasil dari fungsi `histogram()`

```

1 def normalize_hist(img, hist):
2     h, w = img.shape[0:2]
3     size = h * w
4     result = []
5     for i in range(len(hist)):
6         result.append(hist[i] / size)
7     return result

```

Kode sumber 5.7 Implementasi fungsi normalize_hist()

Berikut adalah penjelasan untuk Kode sumber 5.7.

1. Baris 1 adalah deklarasi dari fungsi `normalize_hist()` dengan parameter berupa nilai dari citra yang disimpan dalam variabel `img` dan histogram dari citra yang disimpan dalam variabel `hist`.
2. Baris 2 adalah inisialisasi variabel untuk menampung nilai panjang dan lebar dari citra.
3. Baris 3 adalah inisialisasi variabel untuk menampung jumlah piksel yang terdapat pada citra.
4. Baris 4-6 adalah proses normalisasi histogram dari masing-masing *channel*.
5. Baris 7 adalah proses untuk mengembalikan hasil dari fungsi `normalize_hist()`.

```

1 def h_mean(hist):
2     mean = []
3     for i in range(len(hist)):
4         total = 0
5         div = 0
6         for j in range(len(hist[i])):
7             total += hist[i][j] * j
8             div += hist[i][j]
9         result = total / div
10        mean.append(result)
11    return mean

```

Kode sumber 5.8 Implementasi fungsi h_mean()

Berikut adalah penjelasan dari Kode sumber 5.8.

1. Baris 1 adalah deklarasi dari fungsi `h_mean()` dengan parameter histogram dari citra yang disimpan dalam variabel `hist`.
2. Baris 2-10 adalah proses perhitungan *mean* dari masing-masing *channel* pada histogram.
3. Baris 11 adalah proses untuk mengembalikan hasil dari fungsi `h_mean()`

```

1 def h_stddev(hist, e):
2     std = []
3     for i in range(len(hist)):
4         total = 0
5         div = 0
6         for j in range(len(hist[i])):
7             total += (hist[i][j] * ((j - e[i]) ** 2))
8             div += hist[i][j]
9         result = (total / div) ** 0.5
10        std.append(result)
11    return std

```

Kode sumber 5.9 Implementasi fungsi h_stddev()

Berikut adalah penjelasan dari Kode sumber 5.9.

1. Baris 1 adalah deklarasi dari fungsi `h_stddev()` dengan parameter histogram dari citra yang disimpan dalam variabel `hist` dan nilai rata-rata dari histogram yang disimpan dalam variabel `e`.
2. Baris 2-10 adalah proses perhitungan deviasi standar dari masing-masing `channel` pada histogram.
3. Baris 11 adalah proses untuk mengembalikan hasil dari fungsi `h_stddev()`

5.2.2.3 Implementasi *Gray Level Cooccurrence Matrix (GLCM)*

Implementasi GLCM terbagi menjadi enam bagian, yaitu implementasi fungsi `cooccurrence` yang terdapat pada Kode sumber 5.10, implementasi fungsi `normalize_matrix()` yang terdapat pada Kode sumber 5.11, implementasi fungsi `asm()` yang terdapat pada Kode sumber 5.12, implementasi fungsi `contast()` yang terdapat pada Kode sumber 5.13, implementasi fungsi `idm()` yang terdapat pada Kode sumber 5.14, dan implementasi fungsi `entropy()` yang terdapat pada Kode sumber 5.15.

```

1  img = query
2  h, w = img.shape
3  max = img.max()
4  min = img.min()
5  size = max - min + 1
6
7  angle = [0, 45, 90, 135]
8  co = []
9
10 for i in range(len(angle)):
11     co.append(np.zeros(shape=(size, size), dtype=np.uint8))
12
13 for y in range(h):
14     for x in range(w):
15         if x < w - 1:
16             co[0][(img[y][x] - min), (img[y][x + 1] - min)] += 1
17             co[0][(img[y][x + 1] - min), (img[y][x] - min)] += 1
18
19         if y < h - 1:
20             co[2][(img[y][x] - min), (img[y + 1][x] - min)] += 1
21             co[2][(img[y + 1][x] - min), (img[y][x] - min)] += 1
22
23         if x < w - 1 and y < h - 1:
24             co[1][(img[y][x + 1] - min), (img[y + 1][x] - min)] +=
25 1
26             co[1][(img[y + 1][x] - min), (img[y][x + 1] - min)] +=
27 1
28             co[3][(img[y][x] - min), (img[y + 1][x + 1] - min)] +=
29 1
30             co[3][(img[y + 1][x + 1] - min), (img[y][x] - min)] +=
31 1

```

Kode sumber 5.10 Implementasi fungsi `cooccurrence()`

Berikut adalah penjelasan dari Kode sumber 5.10.

1. Baris 1 adalah inisialisasi variabel `img` yang berisi nilai dari citra.
2. Baris 2 adalah inisialisasi variabel untuk menampung nilai panjang dan lebar dari citra.

- Baris 3-5 adalah perhitungan untuk menentukan ukuran dari matriks *cooccurrence* yang diberi nama variabel *co*.
- Baris 7-27 adalah proses untuk menghitung jumlah kejadian nilai pada citra bertetangga di setiap sudut-sudutnya. Jumlah kejadian tersebut disimpan ke dalam variabel *co*.

```

1 def normalize(matrix):
2     total = 0
3     h, w = matrix.shape
4     result = np.zeros((h, w))
5     for x in range(w):
6         for y in range(h):
7             total += matrix[x][y]
8     for x in range(w):
9         for y in range(h):
10            result[x][y] = float(matrix[x][y]) / total
11    return result

```

Kode sumber 5.11 Implementasi fungsi `normalize_matrix()`

Berikut adalah penjelasan dari Kode sumber 5.11.

- Baris 1 adalah deklarasi dari fungsi `normalize_hist()` yang memiliki parameter matriks *cooccurrence* yang disimpan dalam variabel *matrix*.
- Baris 2-10 adalah proses perhitungan normalisasi matriks *cooccurrence*.
- Baris 11 adalah proses mengembalikan hasil dari fungsi `normalize_matrix()`, yaitu berupa matriks yang telah dinormalisasi.

```

1 def energy(p):
2     h, w = p.shape
3     result = 0
4     for x in range(w):
5         for y in range(h):
6             result += (p[x][y] ** 2)
7     return result

```

Kode sumber 5.12 Implementasi fungsi `energy()`

Berikut adalah penjelasan dari Kode sumber 5.12.

- Baris 1 adalah deklarasi dari fungsi `energy()` yang memiliki parameter berupa matriks *cooccurrence* yang disimpan dalam variabel *p*.
- Baris 3-6 adalah proses perhitungan *energy*.
- Baris 7 adalah proses untuk mengembalikan hasil dari fungsi `energy()`.

```

1 def contrast(p):
2     h, w = p.shape
3     result = 0
4     for x in range(w):
5         for y in range(h):
6             n = x - y
7             if (n < 0): n = -n
8             result += (n ** 2) * p[x][y]
9     return result

```

Kode sumber 5.13 Implementasi fungsi `contrast()`

Berikut adalah penjelasan dari Kode sumber 5.13.

1. Baris 1 adalah deklarasi dari fungsi `contrast()` yang memiliki parameter berupa matriks *cooccurrence* yang disimpan dalam variabel *p*.
2. Baris 3-8 adalah proses perhitungan *contrast*.
3. Baris 9 adalah proses untuk mengembalikan hasil dari fungsi `contrast()`.

```

1 def homogeneity(p):
2     h, w = p.shape
3     result = 0
4     for x in range(w):
5         for y in range(h):
6             result += (1 / (1 + (x - y) ** 2)) * p[x][y]
7     return result

```

Kode sumber 5.14 Implementasi fungsi `homogeneity()`

Berikut adalah penjelasan dari Kode sumber 5.14.

1. Baris 1 adalah deklarasi dari fungsi `homogeneity()` yang memiliki parameter berupa matriks *cooccurrence* yang disimpan dalam variabel *p*.
2. Baris 3-6 adalah proses perhitungan *homogeneity*.
3. Baris 7 adalah proses untuk mengembalikan hasil dari fungsi `homogeneity()`.

```

1 def entropy(p):
2     h, w = p.shape
3     result = 0
4     sum = 0
5     for x in range(w):
6         for y in range(h):
7             if p[x][y] == 0:
8                 result += 0
9             else:
10                result += -(p[x][y] * math.log10(p[x][y]))
11    return result

```

Kode sumber 5.15 Implementasi fungsi `entropy()`

Berikut adalah penjelasan dari Kode sumber 5.15.

1. Baris 1 adalah deklarasi dari fungsi `entropy()` yang memiliki parameter berupa matriks *cooccurrence* yang disimpan dalam variabel *p*.
2. Baris 2-10 adalah proses perhitungan *entropy*.
3. Baris 11 adalah proses untuk mengembalikan hasil dari fungsi `entropy()`.

5.2.3 Implementasi Seleksi Fitur *Information Gain*

Berikut adalah implementasi dari seleksi fitur *Information Gain*.

```

1 def gain(img_data):
2     kelas = ('001', '004', '005', '006', '007', '008', '009',
3             '021')
4     all_gain = []
5     matrix_kelas = np.zeros(len(kelas))
6     i_kelas = len(img_data[0]) - 1
7     total_data = len(img_data) - 1
8     l = int(1 + 3.3 * math.log10(len(img_data) - 1))
9     for i in range(len(kelas)):
10        for j in range(1, len(img_data)):
11            if kelas[i] == img_data[j][i_kelas]:
12                matrix_kelas[i] += 1

```

```

13
14     H = 0
15     for i in range(len(kelas)):
16         p = matrix_kelas[i] / total_data
17         H += -(p * math.log2(p))
18
19     for x in range(1, i_kelas):
20         g = []
21         matrix = np.zeros((len(kelas), 1))
22         for y in range(1, len(img_data)):
23             for z in range(len(kelas)):
24                 if img_data[y][i_kelas] == kelas[z]:
25                     matrix[z][int(img_data[y][x])] += 1
26         entropy_k = 0
27         for i in range(len(matrix)):
28             count = 0
29             for j in range(len(matrix[0])):
30                 count += matrix[i][j]
31             p_a = count / total_data
32             for j in range(len(matrix_kelas)):
33                 if count == 0 : p_k_a = 0
34                 else : p_k_a = matrix[i][j] / count
35                 if p_k_a == 0:
36                     log = 0
37                 else:
38                     log = math.log2(p_k_a)
39                 h_a = p_a * p_k_a * log
40                 entropy_k += h_a
41             g.append(H + entropy_k)
42         g.append(x)
43         all_gain.append(g)
44
         return sorted(all_gain, reverse=True)

```

Kode sumber 5.16 Implementasi fungsi gain()

Berikut adalah penjelasan dari Kode sumber 5.16.

1. Baris 1 adalah deklarasi dari fungsi `gain()` yang memiliki parameter variabel `img_data` yang berisi nilai-nilai dari fitur yang dimiliki oleh citra yang digunakan sebagai data latih.
2. Baris 2-7 adalah inisialisasi variabel-variabel yang digunakan pada fungsi `gain()`. Variabel `kelas` berisi kelas-kelas yang digunakan, variabel `all_gain` adalah variabel untuk menyimpan hasil perhitungan gain secara keseluruhan, variabel `matrix_kelas` untuk menyimpan jumlah kejadian kelas pada data latih, `i_kelas` adalah variabel yang menyimpan indeks dari kolom kelas pada data latih, `l` adalah variabel yang menyimpan jumlah kategori yang terdapat pada data latih.
3. Baris 8-11 adalah proses untuk menghitung jumlah kejadian seluruh kelas pada data latih.
4. Baris 13-16 adalah proses perhitungan *entropy* dari tiap-tiap kelas, menjumlahkan seluruh *entropy* tiap kelas dan disimpan ke dalam variabel `H`.
5. Baris 18-21 adalah proses untuk menghitung kejadian nilai dalam fitur pada tiap-tiap kelas.
6. Baris 25-39 adalah proses perhitungan *entropy* dari tiap-tiap fitur.

7. Baris 40 adalah proses perhitungan *gain* tiap fitur. Nilai *gain* disimpan ke dalam variabel *g*.
8. Baris 41 adalah proses menambahkan indeks dari fitur ke dalam variabel *g*. Sehingga isi dari variabel *g* adalah nilai *gain* dari fitur dan indeksnya.
9. Baris 42 adalah proses menggabungkan variabel *g* dari semua fitur.
10. Baris 44 adalah proses untuk mengembalikan hasil dari fungsi *gain()* yang mana hasil yang dikembalikan berupa *list* yang berisi nilai *gain* yang telah diurutkan dari nilai *gain* terbesar sampai yang terkecil.

5.2.4 Similarity Measure

Berikut adalah implementasi dari proses *similarity measure* dengan menggunakan *Euclidean distance*.

```

1  img_data = []
2  file = 'img_data.csv'
3  with open(file, 'r') as f:
4      reader = csv.reader(f)
5      for row in reader:
6          img_data.append(row)
7
8  ig = []
9  file = 'information_gain.csv'
10 with open(file, 'r') as f:
11     reader = csv.reader(f)
12     for row in reader:
13         ig.append(row)
14
15 selected_feature = 10
16
17 distance = []
18 for a in range(1, len(img_data)):
19     total = 0
20     for b in range(0, selected_feature):
21         i = int(ig[b][1])
22         total += (feature[i-1] - float(img_data[a][i])) ** 2
23         d = total ** 0.5
24         result = [d, a]
25     distance.append(result)
26
27 distance = sorted(distance)

```

Kode sumber 5.17 Implementasi fungsi *similarity_measure()*

Berikut adalah penjelasan dari Kode sumber 5.17.

1. Baris 1-6 adalah proses untuk mengambil data latih lalu disimpan ke dalam variabel *img_data*.
2. Baris 8-13 adalah proses untuk mengambil hasil dari perhitungan *Information Gain* lalu disimpan ke dalam variabel *ig*.
3. Baris 15 adalah inisialisasi variabel *selected_feature* yang berisi jumlah variabel yang ingin digunakan dalam perhitungan *similarity measure*.
4. Baris 17-25 adalah proses perhitungan *similarity measure* dengan menggunakan *Euclidean distance*.
5. Baris 27 adalah proses pengurutan hasil perhitungan *similarity measure* dari nilai terkecil sampai nilai terbesar.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini dijelaskan mengenai hasil beserta analisis dari hasil yang didapatkan dari pengujian. Pengujian yang dilakukan sesuai dengan skenario pengujian yang telah dijelaskan sebelumnya yang terdapat pada Bab 4.

Pengujian dilakukan dengan menggunakan satu data uji yang diambil secara acak dari masing-masing kelas sehingga total data uji yang digunakan untuk pengujian ini berjumlah 8 dan bersifat tetap yang artinya pada semua pengujian yang dilakukan menggunakan data uji yang sama. Data uji ini dijadikan sebagai *query* sebagai masukan untuk sistem. Citra yang digunakan pada pengujian terdapat pada Tabel 6.1. Data latih yang digunakan berjumlah 241 data dan bersifat tetap yang mana pada semua pengujian yang dilakukan menggunakan data latih yang sama. Data latih beserta data uji yang digunakan pada penelitian ini dapat diunduh pada pranala yang terdapat pada LAMPIRAN A.

Tabel 6.1 Data uji

No.	Citra	Nama File	Kelas
1		001_0001_DAF.jpg	Donat
2		004_0001_DAF.jpg	Mie Goreng
3		005_0001_DAF.jpg	Mie Gepeng
4		006_0001_DAF.jpg	Telor Ceplok
5		007_0001_DAF.jpg	Telor Dadar
6		008_0001_DAF.jpg	Fried Chicken
7		009_0001_DAF.jpg	Rendang
8		021_0001_DAF.jpg	Nasi Kuning

6.1 Pengujian dan Analisis Pengaruh Jumlah Pengembalian

Pada pengujian ini dilakukan perbandingan nilai MAP yang didapatkan dari sistem dengan jumlah pengembalian (r) sebanyak 5, 10, 15, dan 20. Pengujian ini dilakukan untuk mengetahui jumlah pengembalian yang dapat memberikan nilai MAP yang lebih tinggi yang artinya sistem memberikan hasil yang lebih baik. Selain itu, pengujian ini juga dilakukan untuk memfokuskan pengujian pada subbab-subbab berikutnya pada jumlah pengembalian atau nilai r yang memberikan nilai MAP terbaik.

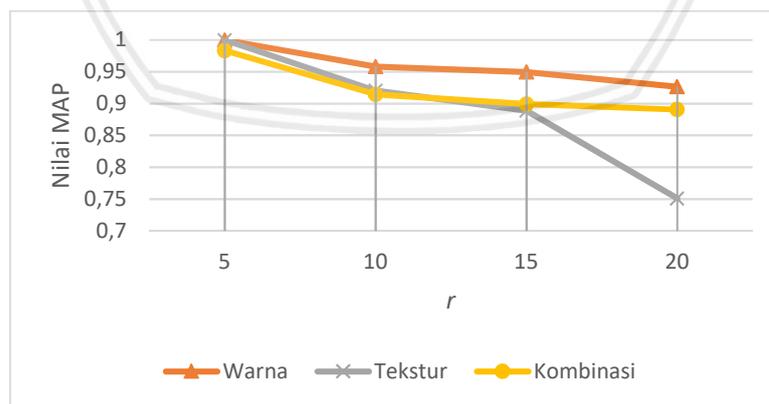
Tabel 6.2 Hasil pengujian pengaruh jumlah pengembalian

		r			
		5	10	15	20
Fitur	Warna	1	0,958333	0,949543	0,926436
	Tekstur	1	0,920366	0,888759	0,750926
	Kombinasi	0,983333	0,914261	0,899351	0,89068

Keterangan:

- r adalah jumlah pengembalian yang diberikan oleh sistem.
- Fitur adalah fitur yang digunakan pada sistem, yaitu fitur warna, tekstur, dan kombinasi (warna dan tekstur).

Dari Tabel 6.2 dapat diketahui bahwa nilai r yang lebih kecil akan mendapatkan nilai MAP yang lebih baik. Nilai MAP pada penggunaan fitur warna, tekstur, dan kombinasi pada $r = 5$ lebih tinggi dibandingkan nilai r yang lainnya. Semakin banyak hasil yang dikembalikan maka akan semakin kecil nilai MAP dari sistem. Hal ini dapat dilihat pada penyajian data dalam bentuk grafik terdapat pada Gambar 6.1.



Gambar 6.1 Grafik pengaruh nilai r terhadap nilai MAP

Pada pengembalian yang lebih banyak atau nilai r yang lebih besar, akan semakin banyak hasil yang tidak relevan yang dikembalikan oleh sistem. Hal ini dapat dilihat pada Tabel 6.3 dan Tabel 6.4 yang merupakan hasil pengembalian

dari *query* kelas Fried Chicken. Pada Tabel 6.3, peringkat pertama dan kedua dari hasil pengembalian ditempati dengan hasil yang relevan yaitu Fried Chicken. Namun, pada peringkat ketiga, keempat, dan kelima ditempati oleh Telur Ceplok. Hal ini dapat terjadi karena dari hasil ekstraksi fitur yang dilakukan, kelas Fried Chicken dengan kelas Telur Ceplok dianggap mirip oleh sistem. Analisis lebih dalam mengenai hal ini akan dilakukan pada subbab berikutnya. Meskipun hasil yang tidak relevan menempati 3 dari 5 hasil pengembalian, nilai MAP dari pengembalian ini adalah satu karena peringkat teratas secara berurutan ditempati oleh hasil yang relevan.

Tabel 6.3 Hasil pengembalian $r = 5$

Peringkat	1	2	3	4	5
Citra					

Lain halnya dengan hasil dari $r = 10$ yang terdapat pada Tabel 6.4 yang mana pada peringkat ketujuh terdapat hasil yang relevan. Hal ini menyebabkan nilai MAP menurun karena hasil yang relevan terdapat pada peringkat yang lebih rendah dibandingkan dengan hasil yang tidak relevan. Hal ini berlaku pada pengembalian sebanyak apa pun karena nilai MAP mempertimbangkan peringkat dari hasil yang dikembalikan. Oleh karena itu, untuk mendapatkan nilai MAP yang lebih besar maka pengembalian dilakukan pada jumlah yang lebih sedikit.

Tabel 6.4 Hasil pengembalian $r = 10$

Peringkat	1	2	3	4	5	6	7	8	9	10
Citra										

6.2 Pengujian dan Analisis Pengaruh Seleksi Fitur terhadap Sistem dengan Fitur Warna

Pada pengujian ini dilakukan perbandingan antara nilai MAP yang didapatkan dari sistem dengan seleksi fitur dan dengan tanpa seleksi fitur. Fitur warna memiliki jumlah fitur sebanyak 39 yang didapatkan dari proses ekstraksi fitur *Color Moments* dan *Color Histogram*. Nilai MAP dari jumlah fitur tersebut dibandingkan dengan fitur yang didapatkan dari proses seleksi fitur, yaitu 5, 10, 15, 20, 25, 30, dan 35 fitur warna yang memiliki nilai *gain* tertinggi yang dapat dilihat pada LAMPIRAN B. Jumlah fitur yang digunakan diberi nama variabel n . Pengujian dilakukan dengan jumlah pengembalian atau nilai r yang memberikan nilai MAP

terbaik, yaitu $r = 5$ yang didapatkan dari pengujian pada subbab 6.1. Hasil pengujian ini terdapat pada Tabel 6.5.

Nilai MAP sempurna yaitu sebesar satu didapatkan pada pengujian dengan jumlah fitur 10, 35, dan 39. Hal ini menandakan bahwa sistem dapat memberikan hasil yang sama baiknya dengan tanpa menggunakan seleksi fitur namun dengan kelebihan yaitu menggunakan fitur yang lebih sedikit yang telah melalui tahap seleksi fitur. Pada $n = 10$, fitur-fitur yang digunakan secara berurutan sesuai dengan nilai *gain* tertinggi adalah histogram *channel R bin 5*, histogram *channel R bin 3*, histogram *channel G bin 3*, deviasi standar dari histogram *channel G*, deviasi standar *color moments channel G*, *mean histogram channel R*, *mean color moments* dari *channel G*, *mean histogram channel G*, histogram *channel R bin 0*, dan *mean color moments* dari *channel R*. Selanjutnya, analisis difokuskan terhadap $n = 10$ dan $n = 39$.

Tabel 6.5 Hasil pengujian pengaruh seleksi fitur terhadap sistem dengan fitur warna

n	MAP
5	0,9625
10	1
15	0,9625
20	0,985938
25	0,948438
30	0,985938
35	1
39	1

Keterangan:

- n adalah jumlah fitur yang digunakan
- MAP adalah nilai MAP dari hasil yang dikembalikan oleh sistem
- Pada $n = 39$ merupakan pengujian dengan menggunakan semua fitur / tanpa menggunakan seleksi fitur pada fitur warna

Walaupun sama-sama memiliki nilai MAP sebesar satu, terdapat perbedaan dari hasil yang diberikan oleh $n = 10$ dan $n = 39$. Tabel 6.6 dan Tabel 6.7 adalah hasil pengembalian dari sistem dengan *query* kelas Fried Chicken. Berdasarkan Tabel 6.6, pada pengembalian $n = 10$, terdapat hanya satu hasil yang relevan. Namun, hasil yang relevan tersebut menempati peringkat pertama sehingga nilai MAP yang didapatkan adalah satu.

Tabel 6.6 Hasil pengembalian $n = 10$

Peringkat	1	2	3	4	5
Citra					

Berdasarkan Tabel 6.7, pada $n = 39$ terdapat dua hasil relevan yang menempati dua peringkat teratas sehingga mendapatkan nilai MAP sebesar 1. Perbedaan dari hasil yang didapatkan dari $n = 10$ dan $n = 39$ adalah $n = 10$ memberikan hasil relevan yang lebih sedikit. Selain itu, pada hasil yang tidak relevan, kedua nilai n memberikan hasil yang berbeda yang mana pada $n = 10$, *query* Fried Chicken dianggap mirip dengan Nasi Kuning dan Mie Gepeng. Sedangkan pada $n = 39$, *query* Fried Chicken dianggap mirip dengan Telur Dadar. Hal ini menandakan bahwa berdasarkan tahap ekstraksi fitur, kelas Fried Chicken, Nasi Kuning, Mie Gepeng dan Telur Ceplok memiliki nilai fitur yang mirip.

Tabel 6.7 Hasil pengembalian $n = 39$

Peringkat	1	2	3	4	5
Citra					

6.3 Pengujian dan Analisis Pengaruh Seleksi Fitur terhadap Sistem dengan Fitur Tekstur

Pada pengujian ini dilakukan perbandingan antara nilai MAP yang didapatkan dari sistem dengan seleksi fitur dan dengan tanpa seleksi fitur. Fitur tekstur memiliki jumlah fitur sebanyak 16 yang didapatkan dari ekstraksi fitur menggunakan *Gray Level Cooccurrence Matrix*. Nilai MAP dari jumlah fitur tersebut dibandingkan dengan fitur yang didapatkan dari proses seleksi fitur, yaitu 5 dan 10 fitur tekstur yang memiliki nilai *gain* tertinggi yang dapat dilihat pada LAMPIRAN B. Jumlah fitur yang digunakan diberi nama variabel n . Pengujian dilakukan dengan jumlah pengembalian atau nilai r yang memberikan nilai MAP terbaik, yaitu $r = 5$ yang didapatkan dari pengujian pada subbab 6.1. Hasil pengujian ini terdapat pada Tabel 6.8.

Tabel 6.8 Hasil pengujian pengaruh seleksi fitur terhadap sistem dengan fitur tekstur

n	MAP
5	1
10	1
16	1

Keterangan:

- n adalah jumlah fitur yang digunakan
- MAP adalah nilai MAP dari hasil yang dikembalikan oleh sistem
- Pada $n = 16$ merupakan pengujian dengan menggunakan semua fitur / tanpa menggunakan seleksi fitur pada fitur tekstur.

Pada pengujian ini, berdasarkan Tabel 6.8, pada ketiga jumlah fitur yang diujikan memiliki nilai MAP yang sama yaitu sebesar satu. Hal ini menunjukkan bahwa dengan menggunakan seleksi fitur dapat memberikan nilai MAP yang sama baiknya dengan fitur yang lebih banyak. Pada hasil yang tidak relevan, terdapat kecenderungan kemiripan yang dimiliki antar kelas. Dari kelas yang ada, terbagi menjadi dua berdasarkan kemiripan tekstur seperti pada Gambar 2.1.



Gambar 6.2 Kemiripan antar kelas berdasarkan tekstur

Pada *query* berupa kelas Rendang, sistem mengembalikan citra dari kelas Roti. Pada *query* berupa kelas Roti, sistem mengembalikan citra dari kelas Telur Ceplok. Dari hasil ekstraksi fitur yang dilakukan, ketiga kelas ini dianggap memiliki kemiripan secara tekstur seperti yang ditampilkan pada poin (a) pada Gambar 6.2. Begitu juga dengan kelima kelas lainnya memiliki kemiripan berdasarkan tekstur seperti yang ditampilkan pada poin (b) pada Gambar 6.2. Selanjutnya, analisis akan difokuskan pada nilai n terkecil yang mendapatkan nilai MAP tertinggi yaitu $n = 5$ dan pada nilai n tanpa seleksi fitur yaitu pada $n = 16$.

Tabel 6.9 Hasil pengembalian $n = 5$ pada *query* kelas Donat

Peringkat	1	2	3	4	5
Citra					

Pada *query* dari kelas Donat, pada $n = 5$ dan $n = 16$ sama-sama memiliki nilai *average precision* sebesar satu. Perbedaan hasil dari kedua nilai n dapat dilihat pada Tabel 6.9 dan Tabel 6.10. Pada $n = 5$, sistem mengembalikan dua hasil relevan. Jumlah tersebut lebih sedikit jika dibandingkan dengan $n = 16$ yang mengembalikan empat hasil relevan. Jumlah hasil relevan yang dikembalikan tidak akan berpengaruh pada nilai MAP selama menempati peringkat teratas dan memiliki peringkat lebih tinggi dibandingkan hasil yang tidak relevan. Pada $n = 5$, hasil tidak relevan yang dikembalikan oleh sistem berasal dari kelas Telur Ceplok, sedangkan pada $n = 16$, hasil tidak relevan yang dikembalikan oleh sistem berasal dari kelas Rendang. Hal ini menandakan bahwa kelas Donat, kelas Telur Ceplok, dan kelas Rendang memiliki nilai fitur yang mirip.

Tabel 6.10 Hasil pengembalian $n = 16$ pada *query* kelas Donat

Peringkat	1	2	3	4	5
Citra					

Tabel 6.11 Hasil pengembalian $n = 5$ pada *query* kelas Mie Goreng

Peringkat	1	2	3	4	5
Citra					

Tabel 6.12 Hasil pengembalian $n = 16$ pada *query* kelas Mie Goreng

Peringkat	1	2	3	4	5
Citra					

Tabel 6.13 Hasil pengembalian $n = 5$ pada *query* kelas Fried Chicken

Peringkat	1	2	3	4	5
Citra					

Pada *query* dari kelas Mie Gepeng, pada $n = 5$ dan $n = 16$ sama-sama memiliki nilai *average precision* sebesar satu. Hasil pengembalian dari $n = 5$ terdapat pada Tabel 6.11 dan hasil pengembalian dari $n = 16$ terdapat pada Tabel 6.12. Keduanya memberikan pengembalian hasil relevan sebanyak satu dan menempati peringkat pertama. Perbedaan terjadi pada hasil yang tidak relevan yang mana pada $n = 5$ mengembalikan hasil dari kelas Fried Chicken dan kelas Mie Goreng sedangkan pada $n = 16$ mengembalikan hasil dari kelas Mie Goreng dan kelas Telur Dadar. Hal ini menandakan bahwa kelas Mie Gepeng, kelas Fried Chicken, Mie Goreng, dan Telur Ceplok memiliki nilai yang mirip. Selain itu, kelas Nasi Kuning juga memiliki nilai fitur yang mirip dengan kelas Mie Gepeng, kelas Fried Chicken, dan kelas Mie Goreng. Hal ini dapat dilihat pada hasil pengembalian dengan *query* dari kelas Fried Chicken yang mengembalikan hasil dari kelas Fried Chicken, kelas Mie Gepeng, dan kelas Nasi Kuning. Hasil pengembalian dari *query* kelas Nasi Kuning terdapat pada Tabel 6.13.

6.4 Pengujian dan Analisis Pengaruh Seleksi Fitur terhadap Sistem dengan Kombinasi Fitur Warna dan Tekstur

Pada pengujian ini dilakukan perbandingan antara nilai MAP yang didapatkan dari sistem dengan seleksi fitur dan dengan tanpa seleksi fitur. Kombinasi antara fitur warna dan tekstur menghasilkan jumlah fitur sebanyak 55. Nilai MAP dari jumlah fitur tersebut dibandingkan dengan fitur yang didapatkan dari proses seleksi fitur, yaitu 5, 10, 15, 20, 25, 30, 35, 40, 45, dan 50 fitur warna dan tekstur yang memiliki nilai *gain* tertinggi yang dapat dilihat pada LAMPIRAN B. Jumlah fitur yang digunakan diberi nama variabel n . Pengujian dilakukan dengan jumlah pengembalian atau nilai r yang memberikan nilai MAP terbaik, yaitu $r = 5$ yang didapatkan dari pengujian pada subbab 6.1. Hasil pengujian ini terdapat pada Tabel 6.14.

Tabel 6.14 Hasil pengujian pengaruh seleksi fitur terhadap sistem dengan kombinasi fitur warna dan tekstur

n	MAP
5	0,9625
10	0,985938
15	0,985938
20	0,985938
25	0,985938
30	1
35	1
40	0,96875
45	0,96875
50	0,983333
55	0,983333

Keterangan:

- n adalah jumlah fitur yang digunakan
- MAP adalah nilai MAP dari hasil yang dikembalikan oleh sistem
- Pada $n = 55$ merupakan pengujian dengan menggunakan semua fitur / tanpa menggunakan seleksi fitur pada kombinasi fitur warna dan tekstur

Pada pengujian ini, nilai MAP tertinggi didapatkan pada penggunaan jumlah fitur 30 dan 35 yang mana memiliki nilai MAP sebesar satu. Jika dibandingkan dengan fitur warna saja atau tekstur warna saja, penggunaan kombinasi fitur warna membutuhkan fitur yang lebih banyak untuk mendapatkan nilai MAP sebesar satu yang mana membutuhkan 30 fitur, sedangkan fitur warna membutuhkan 10 fitur dan fitur tekstur hanya membutuhkan 5 fitur saja. Pada penggunaan 30 fitur, sistem dapat mengembalikan hasil yang relevan pada peringkat teratas sehingga mampu mendapatkan nilai MAP yang lebih baik.

Selanjutnya, analisis akan difokuskan untuk mengetahui perbedaan dari hasil pengembalian tanpa seleksi fitur ($n = 55$), fitur paling sedikit yang memberikan nilai MAP tertinggi ($n = 30$), dan $n = 10$ yang merupakan fitur paling sedikit yang memberikan hasil yang lebih baik daripada tanpa menggunakan seleksi fitur.

Pada *query* dari kelas Donat, kelas Telur Ceplok, kelas Nasi Kuning, dan Kelas Rendang, baik $n = 10$, $n = 30$, dan $n = 55$ memberikan lima hasil pengembalian yang relevan dari lima hasil yang dikembalikan. Hal ini menunjukkan bahwa dengan kombinasi fitur dapat memberikan hasil yang baik pada *query* yang berasal dari keempat kelas tersebut. Hasil pengembalian dari *query* kelas Donat, kelas Telur Ceplok, kelas Rendang, dan kelas Nasi Kuning dapat dilihat pada Gambar 6.3.



Gambar 6.3 Hasil pengembalian

Pada *query* dari kelas Mie Gepeng, baik $n = 10$, $n = 30$, dan $n = 55$ memberikan hasil yang serupa yaitu peringkat pertama ditempati kelas Mie Gepeng yang merupakan hasil yang relevan dan peringkat kedua sampai kelima ditempati oleh kelas yang tidak relevan yaitu kelas Mie Goreng pada peringkat kedua dan ketiga dan kelas Telur Dadar pada peringkat keempat dan kelima. Hasil dari *query* kelas Mie Gepeng dapat dilihat pada Tabel 6.15.

Tabel 6.15 Hasil pengembalian $n = 55$ pada *query* kelas Mie Gepeng

Peringkat	1	2	3	4	5
Citra					

Pada *query* dari kelas Mie Goreng, pengembalian dengan menggunakan $n = 30$ dan $n = 55$ memiliki nilai *average precision* sebesar satu yang artinya hasil yang relevan menempati peringkat teratas. Pada $n = 10$ memiliki nilai *average precision*

sebesar 0,8875. Pada $n = 10$, peringkat pertama, kedua, keempat, dan kelima ditempati oleh hasil yang relevan, sedangkan pada peringkat ketiga ditempati oleh hasil yang tidak relevan. Walaupun memberikan hasil yang lebih banyak, . Hal ini menyebabkan menurunnya nilai MAP dari $n = 10$ karena peringkat yang tidak relevan lebih tinggi dibandingkan hasil yang relevan. Hasil pengembalian $n = 10$ terdapat pada Tabel 6.16.

Sedangkan hal yang menyebabkan nilai MAP dari $n = 55$ menurun adalah karena nilai *average precision* dari *query* kelas Fried Chicken bernilai 0,866. Pada peringkat ketiga dan keempat terdapat hasil yang tidak relevan dengan kelas Fried Chicken, namun pada peringkat kelima terdapat hasil yang relevan yang menyebabkan nilai *average precision* turun. Hasil pengembalian dari $n = 10$ pada *query* kelas Fried Chicken terdapat pada Tabel 6.17.

Tabel 6.16 Hasil pengembalian $n = 10$ pada *query* kelas Mie Goreng

Peringkat	1	2	3	4	5
Citra					

Tabel 6.17 Hasil pengembalian $n = 55$ pada *query* kelas Fried Chicken

Peringkat	1	2	3	4	5
Citra					

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan dapat disimpulkan sebagai berikut.

1. Pengaruh dari banyaknya hasil yang dikembalikan oleh sistem pada sistem temu kembali citra makanan berdasarkan fitur warna dan tekstur adalah dengan jumlah pengembalian (r) yang lebih sedikit sistem mampu memberikan nilai MAP yang lebih tinggi. Hal ini ditunjukkan pada penggunaan fitur warna, tekstur, dan kombinasi fitur warna dan tekstur yang memiliki nilai MAP yang lebih tinggi pada $r = 5$. Pada penggunaan fitur warna saja, $r = 5$ memiliki nilai MAP = 1. Pada penggunaan fitur tekstur saja, $r = 5$ memiliki nilai MAP = 1. Pada penggunaan kombinasi fitur warna dan tekstur, $r = 5$ memiliki nilai MAP = 0,983333.
2. Pengaruh Algoritme seleksi fitur *Information Gain* pada sistem temu kembali citra makanan dengan menggunakan ekstraksi fitur warna yaitu dapat memberikan nilai MAP yang sama dengan tanpa menggunakan seleksi fitur namun dengan menggunakan jumlah fitur (n) yang lebih sedikit. Pada $n = 39$ atau tanpa seleksi fitur, sistem memiliki nilai MAP sebesar 1. Nilai MAP yang sama dapat diberikan oleh sistem dengan $n = 10$ setelah melalui tahap seleksi fitur.
3. Pengaruh algoritme seleksi fitur *Information Gain* pada sistem temu kembali citra makanan dengan ekstraksi fitur tekstur adalah dapat memberikan nilai MAP yang sama dengan tanpa menggunakan seleksi fitur namun dengan menggunakan jumlah fitur (n) yang lebih sedikit. Pada $n = 16$ atau tanpa seleksi fitur, sistem memiliki nilai MAP sebesar 1. Nilai MAP yang sama dapat diberikan oleh sistem dengan $n = 5$ setelah melalui tahap seleksi fitur.
4. Pengaruh algoritme seleksi fitur *Information Gain* pada sistem temu kembali citra makanan dengan ekstraksi fitur warna dan tekstur adalah dapat memberikan nilai MAP yang lebih baik dari tanpa menggunakan seleksi fitur dengan kelebihan lainnya yaitu dengan menggunakan jumlah fitur (n) yang lebih sedikit. Pada $n = 55$ atau tanpa seleksi fitur, sistem memiliki nilai MAP sebesar 0,983333. Nilai MAP yang lebih baik yaitu sebesar 1 diberikan oleh sistem dengan $n = 30$ setelah melalui tahap seleksi fitur.

7.2 Saran

Saran yang diberikan untuk penelitian lebih lanjut yaitu:

1. Menambah jumlah *dataset* yang digunakan agar mendapatkan hasil evaluasi yang lebih akurat.
2. Memperbaiki proses *preprocessing* untuk menghasilkan gambar tersegmentasi yang lebih baik.
3. Menggunakan fitur lain pada ekstraksi fitur GLCM karena pada penelitian ini hanya menggunakan 4 dari 14 fitur yang ada pada GLCM.

DAFTAR REFERENSI

- Aini, S., Sari, Y. & Arwan, A., 2018. Seleksi Fitur Information Gain untuk Klasifikasi Penyakit Jantung Menggunakan Kombinasi Metode K-Nearest Neighbor dan Naïve Bayes. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(9), pp. 2546-2554.
- Anusha, V., Reddy, V. U. & Ramashri, T., 2014. Content Based Image Retrieval Using Color Moments and Texture. *International Journal of Engineering Research & Technology*, 3(2), pp. 2812-2815.
- Baranidharan, T. & Ghosh, D. D. K., 2011. Medical Image Classification Using Information Gain for Global Feature Reduction. *International Journal of Computer Science and Information Technologies*, 2(6), pp. 2773-2775.
- Benkrama, S., Zaoui, L. & Charrier, C., 2014. Clustering and dimensionality reduction for image retrieval in high-dimensional spaces. *Journal of Theoretical and Applied Computer Science*, 8(3), pp. 37-50.
- Choras, R. S., 2007. Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics System. *International Journal of Biology and Biomedical Engineering*, 1(1).
- Chormunge, S. & Jena, S., 2016. Efficient Feature Subset Selection Algorithm for High Dimensional Data. *International Journal of Electrical and Computer Engineering*, 6(4), pp. 1880-1888.
- EL Mehdi, E. A., Hassan, S. & Nourddine, E., 2017. A Fast and Efficient Image Indexing and Search System based on Color Texture Features. *Indian Journal of Science and Technology*, Volume 10.
- Goel, R., Kumar, V., Srivastava, S. & Sinha, A. K., 2017. A Review of Feature Extraction Techniques for Image Analysis. *International Journal of Advanced Research in Computer and Communication Engineering*, 6(2).
- Haralick, R. M., Shanmugan, K. & Dinstein, I., 1973. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6), pp. 610-621.
- Kanan, C. & Cottrell, G., 2012. Color-to-Grayscale: Does the Method Matter in Image Recognition?. *PLoS One*, 7(1).
- Keen, N., 2005. *Color Moments*. [Online] Tersedia di: <homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/.../KEEN/av_as2_nkeen.pdf> [Diakses 10 September 2018].
- Kodituwakku, S. & Selvarajah, S., 2010. Comparison of Color Features for Image Retrieval. *Indian Journal of Computer Science and Engineering*, 1(3), pp. 207-211.



- Long, F., Zhang, H. & Feng, D. D., 2003. *Fundamentals of Content-Based Image Retrieval*. s.l.:Springer, Berlin, Heidelberg.
- Oussalah, M., 2008. Content Based Image Retrieval: Review of State of Art and Future Directions. *2008 First Workshops on Image Processing Theory, Tools and Applications*, pp. 1-10.
- Puviarasan, N., Bhavani, R. & Vasanthi, A., 2014. Image Retrieval Using Combination of Texture and Shape Features. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(3), pp. 5873-5877.
- Shaltout, N. A., El-Hefnawi, M., Rafea, A. & Moustafa, A., 2014. *Information Gain as a Feature Selection Method for the Efficient Classification of Influenza Based on Viral Hosts*. London, s.n.
- Singh, S. M. & Hemachandran, K., 2012. Content-Based Image Retrieval using Color Moments and Gabor Texture Feature. *International Journal of Computer Science Issues*, 9(5), pp. 299-309.
- Singh, S. M. & Hemachandran, K., 2012. Image Retrieval based on the Combination of Color Histogram and Color Moment. *International Journal of Computer Applications*, 58(3), pp. 27-34.
- Zhang, Y., 2002. *On the use of CBIR in Image Mosaic Generation*, s.l.: s.n.

