

EKSTRAKSI FITUR RGB *COLOR CHANNEL* DAN *SIMPLE MORPHOLOGICAL SHAPE DESCRIPTORS* DARI CITRA MAKANAN UNTUK PENCARIAN RESEP MAKANAN

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Barbara Sonya Hutagaol

NIM: 155150200111285



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

PENGESAHAN

EKSTRAKSI FITUR RGB *COLOR CHANNEL* DAN *SIMPLE MORPHOLOGICAL SHAPE DESCRIPTORS* DARI CITRA MAKANAN UNTUK PENCARIAN RESEP MAKANAN

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Barbara Sonya Hutagaol
NIM: 155150200111285

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Januari 2019

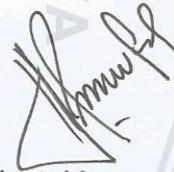
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



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

Dosen Pembimbing II



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

Mengetahui
Ketua Jurusan Teknik Informatika



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 pustaka.

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

Malang, 5 Januari 2019



Barbara Sonya Hutagaol

NIM: 155150200111285



PRAKATA

Puji syukur atas khadirat Allah SWT yang telah memberikan rahmat, taufik dan hidayah-Nya sehingga skripsi yang berjudul “Ekstraksi Fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dari Citra Makanan untuk Pencarian Resep Makanan ” dapat selesai dengan baik.

Penulis menyadari skripsi ini tidak dapat selesai tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan ucapan terimakasih kepada:

1. Ibu Yuita Arum Sari, S.Kom., M.Kom. selaku Dosen Pembimbing I yang telah bersedia memberikan waktunya dan dengan sabar membimbing penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Putra Pandu Adikara, S.Kom., M.Kom. selaku Dosen Pembimbing II yang juga telah bersedia memberi waktunya dan dengan sabar membimbing penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku ketua Program Studi Teknik Informatika.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua Jurusan Teknik Informatika.
5. Sortaliman Hutagaol (ayah), Ir. Sevensi Saragih (ibu), Kak Lolo, Togos, Jordan dan Gomos, serta seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian, dan kesabarannya didalam membesarkan dan mendidik penulis, serta yang senantiasa memberikan semangat untuk menyelesaikan skripsi ini. Skripsi ini penulis persembahkan kepada kalian, keluarga tercinta.
6. Sahabat terdekat saya Pascal Matanari, Yola Napitupulu, dan Chriswanty Marpaung yang selalu memberikan penulis dukungan dan semangat, sehingga penulis tidak merasakan putus asa saat proses pengerjaan skripsi.
7. Kepada sahabat-sahabat saya DCDJ yaitu Gamal, Grace, Abiagael, Dwiki, Oyak, Sahat, Kevin, dan Ferdo yang selalu menjadi teman penulis dan memberikan penulis dukungan dalam banyak hal.
8. Teman-teman penulis Sarah, Vriza, Roy, Riski, Naldo, Sulaiman, Abdan, dan masih banyak lagi, yang sudah tanpa pamrih membantu penulis ketika mengalami kesulitan dalam pengerjaan skripsi ini.
9. Seluruh teman kos yang selalu menemani penulis dalam proses penyusunan skripsi, yang memberikan canda dan tawa sehingga penulis tidak merasa jenuh dalam proses pengerjaan skripsi.
10. PMK Daniel yang selalu menerima kekurangan penulis selama melakukan perkuliahan di Jurusan Teknik Informatika, yang selalu memberikan hangat persaudaraan kepada penulis.

11. PT. Artajasa Pembayaran Elektronik yang memberikan bantuan biaya pendidikan kepada penulis, yang juga memberikan pelatihan mengenai kehidupan pekerjaan yang nantinya akan penulis hadapi.

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

Malang, 5 Januari 2019

Barbara Sonya Hutagaol

Email: barbarash29@gmail.com



ABSTRAK

Barbara Sonya Hutagaol, Ekstraksi Fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dari Citra Makanan untuk Pencarian Resep Makanan

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

Dalam kesehariannya manusia membutuhkan makanan sebagai sumber energi untuk melakukan aktivitasnya. Namun, pada saat ini makanan tidak lagi hanya sekedar makanan pokok. Sebagian masyarakat telah menjadikannya sebagai rekreasi untuk menghibur. Seperti yang dapat dilihat pada media sosial, terdapat sangat banyak foto dari makanan yang menarik hati, sehingga memberikan dorongan untuk memasak dan mencicipinya. Untuk membuat masakan, diperlukan sebuah resep makanan sebagai panduan bagi seseorang untuk mengolah makanan. Secara umum resep makanan dapat ditemukan pada televisi, majalah, tabloid, koran, maupun *website*. Resep biasanya dicari melalui nama dari makanan. Tetapi keterbatasan pengetahuan dari nama makanan menjadi hambatan untuk mencari resep dari makanan tersebut. Berdasarkan masalah tersebut dibutuhkan sebuah sistem yang mampu memberikan hasil pencarian resep dengan *query* dari gambar makanan itu sendiri. Untuk dapat melakukan pencarian resep melalui gambar, maka perlu dilakukan pengolahan citra digital untuk mengetahui ciri atau fitur dari gambar. Warna dan bentuk adalah ciri umum dari sebuah objek pada gambar. Fitur warna yang digunakan adalah RGB *Color Channel* dengan ekstraksi fitur menggunakan *color moment* dan fitur bentuk digunakan *Simple Morphological Shape Descriptors*. Berdasarkan fitur warna dan bentuk, maka dapat dilakukan perhitungan jarak *Euclidean*. Pengujian dilakukan dengan menghitung *Mean Average Precision* (MAP). Hasil yang diperoleh, pencarian terbaik diperoleh ketika nilai *top n* sama dengan 5 dengan MAP 94,1892% yang mana *n* adalah jumlah dokumen teratas yang ditampilkan pada hasil pencarian. Kemudian pengujian untuk membandingkan penggunaan fitur, diperoleh hasil bahwa pemakaian fitur warna dan bentuk cenderung memberikan hasil yang lebih baik dibandingkan penggunaan salah satu fitur saja yaitu pada $n=10$, $n=15$, $n=20$, dan $n=25$.

Kata kunci: resep makanan, *color channel RGB*, *color moment*, *simple morphological shape descriptors*, *Euclidean*

ABSTRACT

Barbara Sonya Hutagaol, RGB Color Channel and Simple Morphological Shape Descriptors Feature Extraction of Food Image for Food Recipe Retrieval

Supervisors: Yuita Arum Sari, S.Kom., M.Kom. and Putra Pandu Adikara, S.Kom., M.Kom.

In daily life, humans need food as source of energy to carry out our activities. But, food is no longer just a basic necessity. Some people have made it as recreation to entertain. As can be seen on social media, there are so many photos of food that attract our attention and gives a boost to cooking and tasting it. To make dishes, we need food recipes as a guide to process the food. In general, food recipes can be found on television, magazines, newspaper, and websites. Recipes are usually searched by the name of the food. But the limitation of knowdlege about the food's name is a barrier to finding the recipes. Based on this problem, we need system that able to provide recipe by the image of the food. To be able finding recipes through the image, we use digital image processing to find out the features of the image. Color and shape are the common characteristics of an object in the image. RGB Color Channel is used as the color features and Simple Morphological Shape Descriptors used as shape feature. Based on that feature, an image can classified using k-Nearest Neighbor. The result get by using Mean Average Precision (MAP). The best way of searching the food recipes, obtained when the top n value is 5 with MAP 94,1892% which n is te top number of document displayed on the search result. Then testing to compare the features, result showed that use the both of color and shape features, give the better result than using only one of them at $n=10$, $n=15$, $n=20$, and $n=25$.

Keywords: food ricepe, color channel RGB, color moment, simple morphological shape descriptos, Euclidean

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xv
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 Resep Makanan.....	6
2.3 Pengolahan Citra Digital	6
2.4 <i>Pre-processing</i>	6
2.4.1 Segmentasi Citra	6
2.4.2 <i>Gaussian Blur</i>	7
2.4.3 Model Warna RGB.....	7
2.4.4 Model Warna HSV	8
2.4.5 Citra <i>Grayscale</i>	9
2.4.6 Citra Biner.....	10
2.4.7 <i>Masking</i>	10
2.5 Ekstraksi Fitur.....	10
2.5.1 Ekstraksi Fitur RGB <i>Color Channel</i>	11



2.5.2	<i>Color Moment</i>	11
2.5.3	Ekstraksi Fitur <i>Simple Morphological Shape Descriptors</i>	12
2.6	Perhitungan Jarak Data	16
2.7	Sistem Temu Kembali Informasi	17
2.8	Pengujian	18
BAB 3	METODOLOGI	19
3.1	Tipe Penelitian	19
3.2	Gambaran Alur Sistem	19
3.3	Lokasi Penelitian	20
3.4	Teknik Pengumpulan Data	20
3.5	Peralatan Pendukung Penelitian	23
BAB 4	PERANCANGAN ALGORITME	24
4.1	Perancangan Algoritme Sistem	24
4.2	Perancangan Algoritme <i>Pre-processing</i> Fitur Warna	25
	25
4.3	Perancangan Algoritme Sistem	26
4.3.1	Perancangan Algoritme Perhitungan <i>Mean</i>	26
4.3.2	Perancangan Algoritme Perhitungan <i>Standard Deviation</i>	28
4.3.3	Perancangan Algoritme Perhitungan <i>Skewness</i>	29
4.4	Perancangan Algoritme <i>Simple Morphological Shape Descriptors</i>	30
4.4.1	Perancangan Algoritme Perhitungan Panjang	32
4.4.2	Perancangan Algoritme Perhitungan Panjang	33
4.4.3	Perancangan Algoritme Perhitungan Lebar	33
4.4.4	Perancangan Algoritme Perhitungan Area	34
4.4.5	Perancangan Algoritme Perhitungan Perimeter	35
4.4.6	Perancangan Algoritme Perhitungan <i>Aspect Ratio</i>	37
4.4.7	Perancangan Algoritme Perhitungan <i>Roundness</i>	37
4.4.8	Perancangan Algoritme Perhitungan <i>Compactness</i>	37
4.4.9	Perancangan Algoritme Perhitungan <i>Rectangulary</i>	38
4.4.10	Perancangan Algoritme Perhitungan Rasio Perimeter <i>L</i>	38
4.4.11	Perancangan Algoritme Perhitungan Rasio Perimeter <i>L</i>	39
4.5	Rancangan Algoritme Perhitungan Jarak	39



4.5.1 Rancangan Algoritme Perhitungan Normalisasi	40
4.5.2 Rancangan Algoritme Perhitungan Jarak <i>Euclidean</i>	41
4.6 Perhitungan Manualisasi	42
4.6.1 Manualisasi RGB <i>Color Channel</i>	42
4.6.2 Manualisasi <i>Simple Morphological Shape Descriptors</i>	46
4.6.3 Manualisasi Perhitungan Jarak <i>Euclidean</i>	50
4.6.4 Manualisasi Pengujian.....	54
BAB 5 IMPLEMENTASI.....	55
5.1 Lingkungan Implementasi.....	55
5.1.1 Lingkungan Perangkat Keras	55
5.1.2 Lingkungan Perangkat Keras	55
5.2 Implementasi Algoritme	55
5.2.1 Impementasi <i>Pre-processing</i> Fitur Warna	55
5.2.2 Impementasi Perhitungan Fitur Warna	56
5.2.3 Impementasi <i>Pre-processing</i> Fitur Bentuk.....	58
5.2.4 Impementasi Perhitungan Fitur Bentuk.....	59
5.2.5 Impementasi Pemanggilan Fitur dan Kelas Citra	62
5.2.6 Impementasi Perhitungan Jarak <i>Euclidean</i>	66
BAB 6 PENGUJIAN DAN ANALISIS	71
6.1 Pengujian dan Analisis pada Variasi Nilai <i>n</i>	71
6.1.1 Skenario Pengujian pada Nilai <i>n</i>	71
6.1.2 Analisis Pengujian pada Nilai <i>n</i>	76
6.2 Pengujian dan Analisis Kombinasi Fitur warna dan Fitur Bentuk.....	77
6.2.1 Skenario Pengujian Kombinasi Fitur warna dan Fitur Bentuk	77
6.2.2 Analisis Pengujian pada Kombinasi Fitur Warna dan Bentuk	78
BAB 7 PENUTUP.....	81
7.1 Kesimpulan.....	81
7.2 Saran.....	81
DAFTAR REFERENSI.....	83
LAMPIRAN A DATA PENELITIAN	86
LAMPIRAN B HASIL EKSTRAKSI fitur	88
LAMPIRAN C RESEP MAKANAN	91





DAFTAR TABEL

Tabel 3.1 <i>Dataset</i> Citra Makanan.....	20
Tabel 4.1 Nilai <i>Red</i> pada Citra Sampel	42
Tabel 4.2 Nilai <i>Green</i> pada Citra Sampel.....	43
Tabel 4.3 Nilai <i>Blue</i> pada Citra Sampel	43
Tabel 4.4 Nilai Fitur RGB <i>Color Channel</i>	46
Tabel 4.5 Nilai Citra Sampel pada Fitur Bentuk	46
Tabel 4.6 Citra Sampel pada Fitur Panjang	47
Tabel 4.7 Citra Sampel pada Fitur Lebar	47
Tabel 4.8 Citra Sampel pada Fitur Area	48
Tabel 4.9 Citra Sampel pada Fitur Perimeter	48
Tabel 4.10 Nilai Fitur <i>Simple Morphological Shape Descriptors</i>	50
Tabel 4.11 Data Manualisasi Fitur Warna	50
Tabel 4.12 Data Manualisasi Fitur Bentuk	51
Tabel 4.13 Hasil Normalisasi pada Fitur Warna	51
Tabel 4.14 Hasil Normalisasi pada Fitur Bentuk	52
Tabel 4.15 Hasil Perhitungan <i>Euclidean</i> Data Uji Nomor 11.....	53
Tabel 4.16 Jarak Terkecil Berdasarkan Nilai n	53
Tabel 4.17 Menghitung <i>Precision</i> pada Data Uji Nomor 11.....	54
Tabel 6.1 Hasil Pencarian dengan n sama dengan 5.....	71
Tabel 6.2 Perhitungan MAP	74
Tabel 6.3 Hasil Pengujian n variasi	76
Tabel 6.4 Hasil Pengujian Kombinasi Fitur	77

DAFTAR GAMBAR

Gambar 2.1 Hasil Segmentasi Citra	7
Gambar 2.2 Citra dengan <i>Gaussian Blur</i>	7
Gambar 2.3 Skema Warna Kubik RGB.....	8
Gambar 2.4 Skema Warna Heksagonal Kerucut HSV.....	8
Gambar 2.5 Citra Hasil Konversi Warna HSV	9
Gambar 2.6 Citra Hasil Konversi <i>Grayscale</i>	10
Gambar 2.7 Citra Hasil Konversi Biner	10
Gambar 2.8 Fitur Bentuk Panjang	12
Gambar 2.9 Fitur Bentuk Lebar	13
Gambar 2.10 Fitur Bentuk Area	13
Gambar 2.11 Fitur Bentuk Perimeter.....	13
Gambar 2.12 Fitur Bentuk <i>Aspect Ratio</i>	14
Gambar 2.13 Fitur Bentuk <i>Roundness</i>	14
Gambar 2.14 Fitur Bentuk <i>Compactness</i>	15
Gambar 2.15 Fitur Bentuk <i>Rectangularity</i>	15
Gambar 2.16 Fitur Bentuk PL	16
Gambar 2.17 Fitur Bentuk <i>PLW</i>	16
Gambar 3.1 Gambaran Alur Sistem	19
Gambar 4.1 Diagram Alir Sistem	24
Gambar 4.2 Diagram Alir <i>Pre-processing</i> Fitur Warna.....	25
Gambar 4.3 Diagram Alir RGB <i>Color Channel</i>	26
Gambar 4.4 Diagram Alir Menghitung <i>Mean</i>	28
Gambar 4.5 Diagram Alir Menghitung <i>Standard deviation</i>	29
Gambar 4.6 Diagram Alir Menghitung <i>Skewness</i>	30
Gambar 4.7 Diagram Alir <i>Simple Morphological Shape Descriptors</i>	32
Gambar 4.8 Diagram Alir <i>Pre-processing</i> Bentuk.....	32
Gambar 4.9 Diagram Alir Menghitung Panjang	33
Gambar 4.10 Diagram Alir Menghitung Lebar	34
Gambar 4.11 Diagram Alir Menghitung Area	35
Gambar 4.12 Diagram Alir Menghitung Area	36

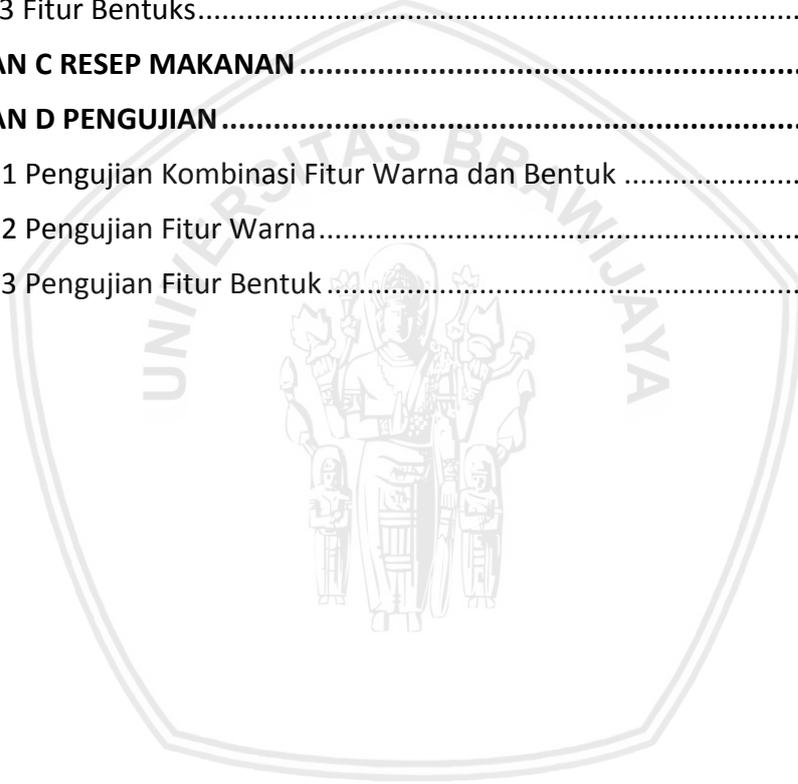


Gambar 4.13 Diagram Alir menghitung <i>Aspect Ratio</i>	37
Gambar 4.14 Diagram Alir menghitung <i>Roundness</i>	37
Gambar 4.15 Diagram Alir Menghitung <i>Compactness</i>	38
Gambar 4.16 Diagram Alir menghitung <i>Rectangulary</i>	38
Gambar 4.17 Diagram Alir Menghitung Rasio Perimeter <i>L</i>	38
Gambar 4.18 Diagram Alir Menghitung Rasio Perimeter <i>L</i> dan <i>W</i>	39
Gambar 4.19 Diagram Alir Perhitungan Jarak	40
Gambar 4.20 Diagram Alir Perhitungan Manualisasi	41
Gambar 4.21 Diagram Alir Perhitungan Jarak <i>Euclidean</i>	42
Gambar 6.1 Citra <i>query</i>	71
Gambar 6.2 Grafik MAP Hasil Pengujian pada <i>n</i> Variasi	77
Gambar 6.3 Grafik MAP pada Hasil Pengujian Kombinasi Fitur	78
Gambar 6.4 Citra yang Memiliki Warna Mirip	79
Gambar 6.5 Citra dengan Intensitas Cahaya yang Berbeda	79
Gambar 6.6 Citra yang Memiliki Bentuk Hampir Sama	80
Gambar 6.7 Citra dengan Kelas Sama tetapi Bentuk Berbeda	80



DAFTAR LAMPIRAN

LAMPIRAN A DATA PENELITIAN	86
A.1 Data Latih.....	86
A.2 Data Uji	87
LAMPIRAN B HASIL EKSTRAKSI FITUR.....	88
B.1 Fitur Warna dan Bentuk	88
B.2 Fitur Warna.....	89
B.3 Fitur Bentuk.....	90
LAMPIRAN C RESEP MAKANAN	91
LAMPIRAN D PENGUJIAN	93
D.1 Pengujian Kombinasi Fitur Warna dan Bentuk	93
D.2 Pengujian Fitur Warna.....	95
D.3 Pengujian Fitur Bentuk	97



BAB 1 PENDAHULUAN

1.1 Latar belakang

Dalam kehidupan sehari-hari manusia membutuhkan makanan demi kelangsungan hidup mereka. Makanan berasal dari tumbuhan dan hewan yang telah diolah untuk dijadikan sebagai sumber energi bagi manusia agar manusia dapat melakukan aktivitasnya (Wijaya, 2016). Saat ini antusiasme masyarakat terhadap makanan atau yang sering disebut kuliner sangat tinggi. Makanan tidak lagi sebagai kebutuhan pokok saja namun sudah menjadi objek rekreasi untuk menghilangkan suntuk karena pekerjaan bagi sebagian masyarakat (Setyahadi, 2014). Seperti pada media sosial sudah sangat banyak gambar makanan yang diunggah dan dengan media fotografi citra makanan tersebut menjadi sangat menarik dan menggurikan, sehingga menimbulkan selera untuk mencicipinya (Arsita, 2017). Untuk mencicipi masakan tersebut dibutuhkan sebuah resep makanan untuk membuatnya. Resep makanan merupakan sebuah panduan bagi seseorang berupa keterangan mengenai bahan yang digunakan dalam mengolah sebuah masakan untuk disajikan (Lestari & Kursini, 2012).

Pada umumnya resep dapat ditemukan melalui buku resep, majalah, televisi maupun koran atau *website* (Yuliana, 2014). Resep dicari berdasarkan nama dari suatu masakan. Namun, keterbatasan pengetahuan mengenai nama dari makanan itu memberikan kesulitan untuk menemukan resep dari makanan yang dicari. Melihat kesulitan menggunakan teknik pencarian menggunakan teks, kemudian dikembangkan teknik *Content Based Image Retrieval* (CBIR). Teknik ini dianggap lebih mampu untuk melakukan pencarian dengan memasukkan *query* berupa citra (Layona, Tunardi, & Tanoto, 2014). Maka dibutuhkan sebuah sistem pencarian informasi mengenai resep makanan melalui *query* berupa citra makanan.

Untuk memecahkan masalah tersebut digunakan teknik pengolahan citra digital. Melalui pengolahan citra digital dapat diperoleh ciri untuk mengidentifikasi objek pada citra (Made & Iswari, 2017). CBIR dapat dilakukan melalui proses ekstraksi fitur pada gambar antara lain seperti warna dan bentuk (Layona Tunardi, & Tanoto, 2014). Citra makanan memiliki warna dan bentuk pada objeknya yang dapat dijadikan ciri untuk membedakannya dengan citra makanan lain. Ciri tersebut kemudian digunakan sebagai atribut pada proses klasifikasi citra makanan.

Sebuah penelitian yang dilakukan oleh (Aryati, Wirayuda, & Dayawati, 2009) menggunakan fitur warna *color moment*. Pada penelitian tersebut dijelaskan *color moment* merupakan ekstraksi fitur berdimensi sedikit yaitu sembilan, sehingga mempercepat proses *retrieved*. Ekstraksi fitur *color moment* juga memiliki akurasi yang konsisten dikarenakan ukuran dari citra tidak mempengaruhi hasil. Hal tersebut disebabkan *color moment* berdasar pada distribusi probabilitas warna.

Penelitian yang telah dilakukan sebelumnya, yaitu penggunaan fitur warna *mean* dan *standard deviation* dan fitur bentuk *morphological shape* menggunakan *aspect ratio*, *roundness*, *rectangulary*, *narrow factor*, rasio perimeter L, dan rasio

perimeter L dan W dalam melakukan pengenalan tanaman berdasarkan gambar daun. Hasil dari penelitian tersebut diperoleh akurasi sebesar 96,32% dengan menggunakan algoritme *Random Forest* (Caglayan, Guclu, & Can, 2013).

Penelitian lain yang dilakukan mengenai klasifikasi citra pada makanan lainnya yaitu penelitian yang mengenai perbandingan algoritme *k*-NN, C4.5, dan *Naive Bayes* dalam pengklasifikasian kesegaran ikan menggunakan media foto. Pada penelitian tersebut dikembangkan sebuah metode untuk mengklasifikasikan kesegaran ikan berdasarkan citra digital ikan. Berdasarkan ketiga metode yang telah disebutkan didapatkan hasil akurasi pada klasifikasi pada Ikan Gurame, Ikan Nila, dan Ikan Kakap dengan rata-rata pada metode *k*-NN mencapai 91,3579, metode C4.5 mencapai 90,1883, dan metode *Naive Bayes* mencapai akurasi sebesar 89,5182. Berdasarkan eksperimen tersebut dapat dilihat bahwa metode *k*-NN memberikan akurasi yang paling tinggi dibandingkan dengan metode C4.5 dan *Naive Bayes* (Wella, Iswari, Ranny, 2017).

Berdasarkan beberapa penelitian yang telah paparkan sebelumnya, maka penelitian ini menggunakan ekstraksi fitur bentuk dan fitur warna dan metode perhitungan jarak *Euclidean* sebagai metode untuk proses klasifikasi karena metode ini merupakan salah satu metode yang baik sebagai metode klasifikasi berdasarkan beberapa penelitian yang telah dilakukan sebelumnya.

1.2 Rumusan masalah

Dengan paparan latar belakang, maka dapat dirumuskan beberapa rumusan masalah pada penelitian ini seperti berikut.

1. Bagaimana merancang pencarian resep makanan menggunakan ekstraksi fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dari resep makanan?
2. Bagaimana pengaruh ekstraksi Fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dari citra makanan untuk pencarian resep makanan?
3. Bagaimana tingkat akurasi pencarian resep makanan melalui ekstraksi Fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dari citra makanan?

1.3 Tujuan

Tujuan penelitian ini ialah seperti berikut.

1. Merancang pencarian resep makanan menggunakan ekstraksi fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dari citra makanan.
2. Mengetahui pengaruh ekstraksi Fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dari citra makanan untuk pencarian resep makanan.
3. Menguji tingkat akurasi pencarian resep makanan melalui ekstraksi Fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dari citra makanan.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini ialah seperti berikut.

1. Mengetahui algoritme ekstraksi fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dalam pencarian resep makanan.
2. Mendapatkan sudut pandang baru mengenai pencarian resep makanan melalui fitur warna dan bentuk dari citra sebuah makanan.
3. Mengetahui kelebihan dan kekurangan dari fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* dalam pencarian resep makanan.

1.5 Batasan masalah

Penelitian ini dibatasi oleh beberapa hal berikut.

1. Dataset yang digunakan terdiri dari 20 citra makanan tunggal yang terdiri dari:
 - Roti: donat, roti tawar, roti gandum
 - Mi Instan: mi goreng keriting, mi goreng gepeng
 - Telur: telur dadar, telur mata sapi
 - Daging: rendang, *fried chicken*
 - Nasi: nasi kuning, nasi merah
 - Makanan Ringan: *Oreo*, *Beng-beng*, *Soba*, *TimTam*, keripik kentang, wafer coklat, biskuit, *Milo*, *Genji*
2. Kamera yang digunakan adalah kamera *iPhone* seri 6 dengan resolusi 8 MP.
3. Gambar di ambil dengan cahaya dari luar ruangan pada siang hari pada pukul 10:00-12:00 WIB.
4. Klasifikasi dilakukan berdasarkan fitur bentuk dan warna.

1.6 Sistematika pembahasan

Sistematika penulisan dalam penelitian ini dijelaskan seperti berikut.

1. BAB I : PENDAHULUAN

Bab ini membahas tentang Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Manfaat Penelitian, Batasan Masalah, dan Sistematika Penulisan.

2. BAB II : LANDASAN KEPUSTAKAAN

Bab ini membahas mengenai referensi teori, konsep, metode, dan literatur ilmiah, sebagai penunjang dalam menyelesaikan penelitian ini.

3. BAB III : METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai metodologi penelitian yang digunakan dalam pengerjaan penelitian yaitu langkah-langkah penyelesaian penelitian ini.

4. BAB IV : PERANCANGAN ALGORITME

Bab ini menjelaskan tentang perancangan sistem menggunakan diaram alir dan perhitungan pada sampel data seperti perhitungan manualisasi dan evaluasi hasil penelitian.

5. BAB IV : IMPLEMENTASI

Bab ini menjelaskan bagaimana sistem diimplementasi dalam bentuk program, perangkat lunak, dan perangkat keras apa saja yang digunakan serta hasil implementasi berupa *sourcecode*.

6. BAB V : PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang bagaimana pengujian yang didapatkan dari penelitian yang telah dilakukan dan menganalisis hasil pengujian dari penelitian yang telah dilakukan.

7. BAB VII : PENUTUP

Bab ini berisi tentang kesimpulan dan saran yang berguna untuk perbaikan dan pengembangan penelitian ke depannya.



BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan masalah dan topik pada penelitian ini.

2.1 Kajian Pustaka

Penelitian ini didasari oleh studi pustaka dari beberapa penelitian yang telah dilakukan peneliti lain sebelumnya dan berkaitan dengan topik dan metode yang digunakan pada penelitian ini.

Penelitian yang dilakukan oleh Caglayan, Guclu, dan Can (2013) yaitu menggunakan fitur bentuk *morphological shape* dan fitur warna untuk melakukan pendekatan pengenalan tanaman pada gambar daun. Pada fitur bentuk menggunakan fitur *aspect ratio*, *roundness*, *rectangulary*, *narrow factor*, *ratio perimeter of L*, dan *ratio perimeter of L and W*. Kemudian pada fitur warna menggunakan fitur warna *mean* dan *standard deviation*. Hasil pengujian terbaik yang diperoleh adalah sebesar 96,32% ketika menggunakan algoritme *Random Forest* pada fitur bentuk dan warna.

Penelitian berikutnya dilakukan oleh Budianita, Jastril, & Handayani (2015) mengenai klasifikasi *k*-NN untuk membangun aplikasi membedakan daging sapi dan daging babi, Ekstraksi fitur yang digunakan ialah histogram menggunakan model warna HSV sebagai fitur warna dan metode orde dua untuk fitur tekstur. Pengujian sistem dilakukan dengan menggunakan data latih dan memberikan nilai *k* yang berbeda. Hasil pengujian mendapatkan akurasi yang tinggi terletak pada pengujian tanpa menggunakan background yaitu sebesar 88,75%. Sedangkan hasil pengujian dengan menggunakan background mencapai akurasi 73,75%.

Penelitian selanjutnya membandingkan algoritme *k*-NN, C4.5, dan Naive Bayes dalam pengklasifikasian kesegaran ikan menggunakan media foto. Pada penelitian tersebut dikembangkan sebuah metode untuk mengklasifikasikan kesegaran ikan berdasarkan citra digital pada ikan. Berdasarkan metode yang telah disebutkan diperoleh hasil akurasi klasifikasi pada Ikan Gurame, Ikan Nila, dan Ikan Kakap dengan rata-rata pada metode *K*-NN mencapai 91,35%, metode C4.5 mencapai 90,18%, dan terakhir metode *Naive Bayes* mencapai akurasi sebesar 89,51%. Berdasarkan penelitian tersebut dapat dilihat bahwa metode *k*-NN memberikan akurasi yang paling baik dibandingkan dengan metode C4.5 dan Naive Bayes (Made & Iswari, 2017).

Selanjutnya penelitian mengenai deteksi kualitas keju dengan klasifikasi menggunakan *k*-NN. Penelitian tersebut melakukan klasifikasi pada citra keju berdasarkan fitur warna dan tekstur. Metode yang digunakan adalah *Gabor Wavelet* sebagai metode ekstraksi fitur pada citra. Pada pengujian dilakukan pada 48 citra keju yang terdiri dari 16 citra keju sangat layak konsumsi, 16 citra keju layak konsumsi, dan 16 citra keju tidak layak konsumsi. Hasil akurasi terbaik yang diperoleh adalah sebesar 91,6% dan waktu komputasi 69,6s dengan menggunakan

parameter orde satu (*standard deviation* dan *kurtosis*), $d_1=32$, $d_2=32$, $k=5$, dan menggunakan perhitungan jarak *euclidean* (Ibrahim, Hidayat, & Darana, 2017).

2.2 Resep Makanan

Dalam membuat sebuah masakan diperlukan sebuah resep makanan untuk menghadirkan suatu masakan yang memiliki rasa yang lezat dan penampilan yang menarik. Resep makanan memiliki pengertian sebagai petunjuk tentang penggunaan bahan makanan, takaran bumbu, dan teknik yang digunakan dalam mengolah makanan (Lestari & Kursini, 2012).

2.3 Pengolahan Citra Digital

Menurut Putra (2010) citra atau gambar diartikan sebagai sebuah fungsi dua dimensi $f(x,y)$, dimana variabel x dan y merupakan sebuah koordinat bidang datar, dan nilai fungsi f di setiap pasangan koordinat (x,y) disebut dengan intensitas atau *gray level* dari gambar dititik itu. Jika x , y dan f semuanya berhingga (*finite*) dan nilainya diskrit, maka gambarnya disebut citra digital atau gambar digital.

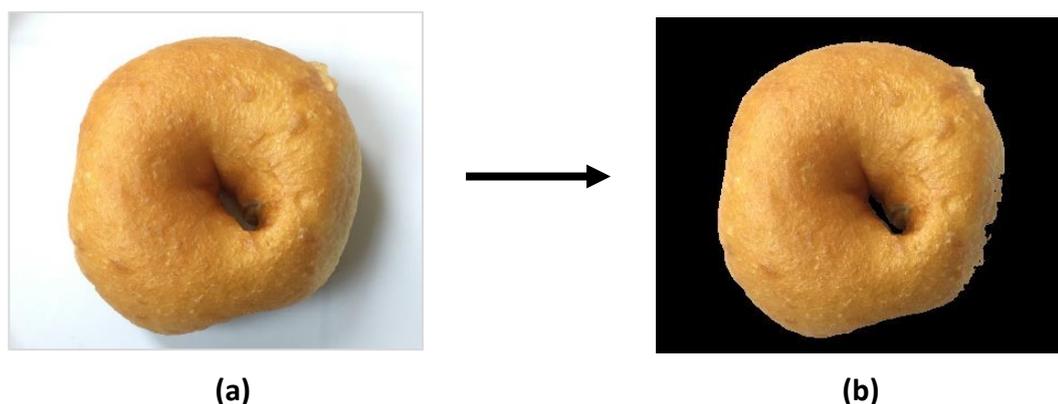
Sebuah citra digital terdiri dari sejumlah elemen yang berhingga dimana masing-masing mempunyai lokasi dan nilai tertentu. Elemen-elemen tersebut disebut juga dengan *picture element*, *image element*, *pels*, atau piksel s .

2.4 Pre-processing

Pre-processing adalah sebuah proses untuk menghilangkan *noise* atau gangguan pada gambar asli yang dapat mempengaruhi ketajaman sebelum gambar tersebut diolah. Tujuan dari *pre-processing* adalah untuk memperoleh citra dengan *background* berwarna hitam untuk fitur warna dan citra biner untuk fitur bentuk. Kemudian hasil *pre-processing* digunakan sebagai citra untuk mendapatkan data yang berisikan nilai-nilai dan digunakan sebagai input dalam melakukan pengklasifikasian. Beberapa kendala dalam tahap *pre-processing* ini mencakup pencahayaan dan perbedaan posisi pengambilan gambar.

2.4.1 Segmentasi Citra

Segmentasi adalah sebuah tahapan *pre-processing* yang dilakukan sebelum melakukan proses ekstraksi fitur. Segmentasi memiliki tujuan untuk membuang bagian yang tidak diperlukan dari citra atau gambar dan mengekstrak *mask* dari objek utama saja (Adinugroho & Sari, 2017). Jenis operasi ini juga bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Tahap-tahap segmentasi untuk ekstraksi fitur warna pada citra yang akan diolah diawali dengan menghilangkan derau dengan *Gaussian Blur*, transformasi RGB ke warna HSV (*hue*, *saturation*, *value*), melakukan *graysaling*, melakukan *thresholding*, *bitwise*, dan *color channel* RGB. Dan untuk ekstraksi fitur bentuk, segmentasi dilakukan sampai tahap erosi. Hasil segmentasi ekstraksi fitur warna ditunjukkan pada Gambar 2.1.



Gambar 2.1 Hasil Segmentasi Citra

2.4.2 Gaussian Blur

Gaussian Blur merupakan sebuah *filter blur* yang memberikan warna transisi secara signifikan pada sebuah citra, kemudian memberikan warna pertengahan dengan tujuan sisi-sisi pada gambar menciptakan efek lembut. *Gaussian Blur* menggunakan persamaan matematika untuk memberikan efek *autofocus* untuk mengurangi ketajaman dan membuat efek kabut pada citra. *Gaussian* merupakan sebuah istilah pada matematika yaitu nama dari seorang ahli matematika yang berasal dari Jerman, *Karl Friedrich Gauss* (Wedianto, Sari, & H, 2016).

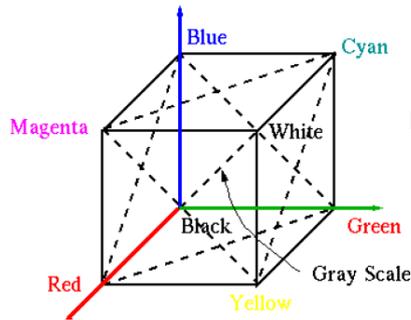
Tujuan dilakukannya *filtering* ialah untuk mereduksi derau pada citra. Pada Gambar 2.2 menunjukkan hasil perubahan dari gambar asli melalui proses *Gaussian Blur*.



Gambar 2.2 Citra dengan *Gaussian Blur*

2.4.3 Model Warna RGB

Dalam model warna RGB (*red, green, blue*) menurut Prasetyo (2011), setiap warna pada gambar menunjukkan komponen *spectral primary* merah, hijau, dan biru yang didasarkan oleh sistem koordinat kartesian. *Sub-space* warna yang dicari yaitu kubus yang mana terdapat nilai RGB pada tiga buah sudut; *cyan*, *magenta*, kuning terdapat pada tiga sudut lainnya, kemudian hitam adalah origin; dan putih merupakan titik yang terjauh dari origin. Lihat pada Gambar 2.3.



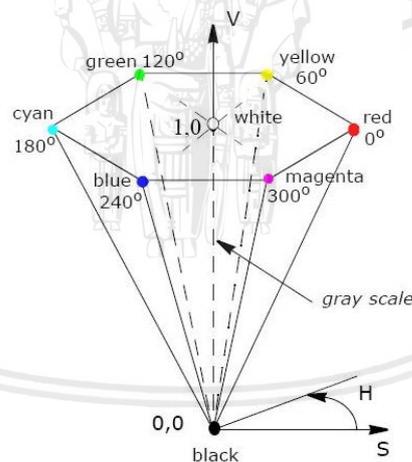
Gambar 2.3 Skema Warna Kubik RGB

Sumber: (Dedy, 2015)

Dalam model warna ini, *grayscale* dilakukan perluasan dari warna hitam ke putih. Perbedaan pada warna ini merupakan titik-titik yang terdapat dalam kubus dan didefinisikan oleh penyebaran vektor dari origin (Prasetyo, 2011).

2.4.4 Model Warna HSV

Warna HSV (*Hue, saturation, value*) merupakan salah satu sistem warna yang dipakai untuk memilih warna dari *color wheel* ataupun *pallette*. Model warna ini dirumuskan oleh pencarian pada kubus warna RGB sepanjang sumbu gabungan dari titik hitam dan putih (keabuan), yang memberikan hasil bentuk heksagonal *color pallette*.



Gambar 2.4 Skema Warna Heksagonal Kerucut HSV

Sumber: scc.ustc.edu.cn

Sepanjang sumbu vertikal (keabuan) seperti yang ditunjukkan pada Gambar 2.4, ukuran pada lapisan *hexagonal* yang tegak lurus terhadap perubahan sumbu. *Hue* merupakan sudut yang berada di sekitar warna *hexagon* dan menggunakan sumbu merah sebagai 0 derajat. *Value* diukur pada sepanjang sumbu kerucut yang mana $V=0$ berada di akhir sumbu yang merupakan warna hitam dan $V=1$ berada pada ujung sumbu pada sisi lainnya merupakan warna putih. Sumbu tersebut terletak pada pusat dari heksagon *full color* sehingga pada sumbu ini

merepresentasikan seluruh bentuk gray. Pengukuran jarak dari sumbu V adalah kemurnian warna (Prasetyo, 2011).

Pada penelitian ini model warna HSV merupakan salah satu tahap untuk proses segmentasi. Warna RGB diubah ke HSV setelah melalui proses *Gaussian Blur*. Dilakukan konversi ke warna HSV ialah untuk mengurangi pengaruh intensitas cahaya dari luar. Dan juga untuk membedakan warna pada objek dan *background* sehingga dapat memberikan hasil segmentasi yang lebih baik. Proses perubahan ke model warna HSV dapat dilihat pada Gambar 2.5.



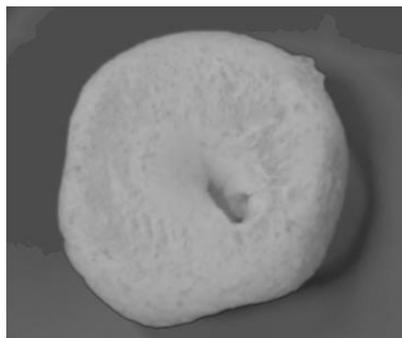
Gambar 2.5 Citra Hasil Konversi Warna HSV

2.4.5 Citra *Grayscale*

Citra dalam bentuk *Grayscale* merupakan sebuah citra digital yang mempunyai sebuah nilai kanal pada pikselnya. Dapat dikatakan pada *channel* merah, hijau, dan biru memiliki nilai yang sama. Nilai tersebut digunakan sebagai petunjuk tingkat intensitas. Warna yang terdapat pada citra *grayscale* ialah warna hitam, warna keabuan, hingga warna putih. Tingkat keabuan yang dimaksud ialah warna abu dengan berbagai tingkatan mulai dari hitam sampai mendekati putih. Nilai piksel pada citra *grayscale* sebesar 8 bit atau sama dengan 256 kombinasi dari warna keabuan (Putra, 2010).

Pada proses segmentasi citra perlu dilakukan sebuah proses *grayscale*. Perubahan citra warna RGB menjadi citra grayscale wajib dilakukan terlebih dahulu sebelum mengubahnya menjadi citra biner (Gonzalez & Woods, 2002). Proses ini dilakukan bertujuan untuk meratakan nilai piksel dari tiga nilai RGB menjadi satu nilai.

Proses ini pada segmentasi dilakukan pada citra hasil konversi HSV. Citra tersebut dikonversi ke citra *grayscale* seperti yang terlihat pada Gambar 2.6. Pada gambar tersebut terlihat berbagai tingkat keabuan pada setiap piksel -nya.



Gambar 2.6 Citra Hasil Konversi *Grayscale*

2.4.6 Citra Biner

Citra Biner atau sering disebut sebagai citra monokrom merupakan citra yang hanya mempunyai dua nilai pada setiap piksel nya yaitu warna hitam dan putih. Disebut sebagai citra biner karena nilai piksel -nya hanya membutuhkan 1 bit yaitu terdiri dari nilai 0 dan 1 (Putra, 2010). Proses mengubah citra menjadi citra biner sering kali digunakan sebagai hasil dari hasil segmentasi, morfologi, pengambangan, dan *dithering*.

Dilakukannya proses ini pada tahap segmentasi bertujuan untuk memberikan warna gelap pada *background* dan warna putih pada objek. Hasil konversi dari citra *grayscale* menjadi citra biner dapat dilihat pada Gambar 2.7.



Gambar 2.7 Citra Hasil Konversi Biner

2.4.7 *Masking*

Penelitian ini menggunakan proses *masking* sebagai tahap terakhir segmentasi yaitu untuk menggabungkan gambar asli dan citra biner. Sehingga dihasilkan gambar objek dengan *background* berwarna hitam atau bernilai 0 seperti yang ditunjukkan pada Gambar 2.1 (b).

2.5 Ekstraksi Fitur

Ekstraksi fitur merupakan suatu pengambilan ciri atau fitur dari suatu citra yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya (Yuristiawan dan Santoso, 2015). Dengan kata lain ekstraksi fitur menggambarkan karakteristik dari sebuah citra. Penelitian ini menggunakan ekstraksi fitur warna

dan bentuk untuk mengklasifikasikan jenis makanan. Ekstraksi fitur warna yang digunakan adalah RGB *Color Channel* dengan pengolahannya menggunakan *Color Moment* untuk penentuan fiturnya. Dan ekstraksi fitur bentuk yang digunakan adalah *Simple Morphological Shape Descriptors*.

2.5.1 Ekstraksi Fitur RGB *Color Channel*

Warna merupakan fitur yang berperan penting karena paling kelihatan saat manusia melihat sebuah gambar pertama kali (Halim, Dewi, & Angkasa, 2013). *Color channel* menyimpan informasi warna untuk salah satu komponen warna utama dari sebuah model warna. Model *color channel* yang dipakai untuk ekstraksi fitur warna untuk penelitian ini adalah RGB yang memiliki tiga *color channel* terpisah yaitu satu warna merah, warna hijau, dan warna biru. Untuk melakukan ekstraksi fitur warna tersebut menggunakan *Color Moment* untuk mendapatkan fitur warna pada gambar.

2.5.2 *Color Moment*

Color moment adalah representasi dari fitur warna dalam memberikan karakteristik warna pada sebuah citra. Perhitungan *moments* bertujuan untuk memperoleh *color similarity* sebuah gambar yang mana nilai dari *similarity* digunakan sebagai perbandingan untuk setiap citra pada *dataset* (Halim et al., 2013).

Pada *color moment* memberikan asumsi distribusi warna dari sebuah citra sebagai distribusi probabilitas. Terdapat tiga *moment* warna yang digunakan sebagai fitur antara lain *mean*, standar deviasi, dan *skewness*.

Menurut Sari, Dewi, & Fatichah (2014) *mean* merupakan nilai rata-rata piksel pada masing-masing *channel* RGB dengan ukuran citra $M \times N$. Untuk mendapatkan nilai fitur *mean*, maka digunakan Persamaan 2.1.

$$\mu = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N P_{ij} \quad (2.1)$$

Keterangan:

μ = Mean

M = Panjang piksel citra

N = Lebar piksel citra

Menurut (Dewi & Ginardi, 2014) untuk mendapatkan nilai fitur *standard deviation* digunakan rumus pada Persamaan 2.2 dan untuk mendapatkan nilai *skewness* digunakan rumus pada Persamaan 2.3.

$$\sigma = \sqrt{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (X_{ij} - \bar{X}_i)^2} \quad (2.2)$$

$$\theta = \frac{1}{M \times N} \frac{\sum_{i=1}^M \sum_{j=1}^N (X_{ij} - \bar{X}_i)^3}{\sigma^3} \quad (2.3)$$

Ketetangan:

σ = Standar Deviasi

θ = *Skewness*

M = Panjang piksel citra

N = Lebar piksel citra

2.5.3 Ekstraksi Fitur *Simple Morphological Shape Descriptors*

Fitur bentuk pada objek sebuah gambar merupakan karakter dari objek. Fitur bentuk merupakan sebuah ciri yang didapatkan berdasarkan bentuk objek dan dinyatakan melalui area, kontur, dan transformasi. Menggunakan fitur bentuk sebagai sebuah ciri digunakan untuk kepentingan indentifikasi objek (Kadir & Susanto, 2013).

Untuk dapat mengetahui area, kontur, dan transformasi pada sebuah objek maka perlu dilakukan *preprocessing* pada citra. Hasil yang didapatkan dari *preprocessing* melalui proses segmentasi sampai tahap *thresholding* yaitu citra biner yang hanya mempunyai dua nilai pada piksel -nya seperti yang ditunjukkan pada Gambar 2.7. Terdapat 13 fitur bentuk yang digunakan pada penelitian ini berupa panjang (L), lebar (W), luas area (A), perimeter (P), *aspect ratio* (AR), *roundness* (R), *compactness* (PA), *rectangularity* (N), rasio perimeter pada panjang (PL), dan rasio perimeter pada panjang dan lebar (PLW).

Menurut Wäldchen & Mäder (2018) fitur bentuk yang telah disebutkan sebelumnya dapat digambarkan seperti berikut.

1. Panjang (L)

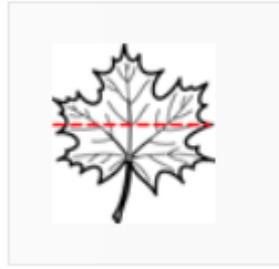
Panjang dapat didefinisikan sebagai jarak pada garis yang menghubungkan pangkal dan ujung objek. Dapat dilihat pada Gambar 2.8.



Gambar 2.8 Fitur Bentuk Panjang

2. Lebar (W)

Lebar didefinisikan sebagai jarak pada garis yang menghubungkan titik paling kiri dan titik paling kanan pada objek yang sumbu tegak lurus dengan panjang. Lebar pada objek ditunjukkan pada Gambar 2.9.



Gambar 2.9 Fitur Bentuk Lebar

3. Area (*A*)

Area merupakan jumlah seluruh piksel yang terdapat pada objek. Luas dari area dapat dilihat pada Gambar 2.10.



Gambar 2.10 Fitur Bentuk Area

4. Perimeter (*P*)

Perimeter atau yang biasa disebut dengan keliling merupakan panjang pada tepi sebuah objek. Luas dari area dapat dilihat pada Gambar 2.11.



Gambar 2.11 Fitur Bentuk Perimeter

5. *Aspect Ratio* (*AR*)

Aspect Ratio (*AR*) merupakan rasio panjang sumbu *L* terhadap panjang sumbu *W* yang menjelaskan mengenai karakteristik objek yang sempit atau lebar. Contoh dari *AR* dapat dilihat pada Gambar 2.12. Rumus untuk mendapatkan nilai *AR* yaitu dengan membagikan *L* dengan *W* seperti yang dapat dilihat pada Persamaan 2.4.

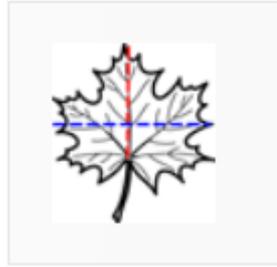
$$AR = \frac{L}{W} \tag{2.4}$$



Keterangan:

L = Panjang

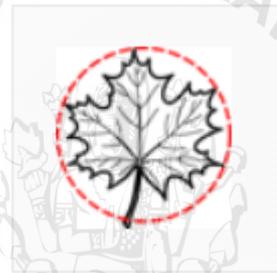
W =Lebar



Gambar 2.12 Fitur Bentuk *Aspect Ratio*

6. *Roundness (R)*

Roundness merupakan fitur yang menggambarkan perbedaan antara objek dan lingkaran. Fitur ini ditunjukkan oleh Gambar 2.13.



Gambar 2.13 Fitur Bentuk *Roundness*

Rumus untuk mendapatkan nilai R dapat menggunakan Persamaan 2.5.

$$R = \frac{4\pi A}{P^2} \tag{2.5}$$

Keterangan:

A = Area

P = Perimeter

7. *Compactness (PA)*

Compactness merupakan sebuah rasio perimeter diatas area objek yang memberikan informasi mengenai kompleksitas umum dan faktor bentuk. Ilustrasi untuk fitur ini dapat dilihat pada Gambar 2.14.





Gambar 2.14 Fitur Bentuk *Compactness*

Rumus yang digunakan untuk mendapatkan nilai *Compactness* menggunakan Persamaan 2.6.

$$PA = \frac{P}{\sqrt{A}} \tag{2.6}$$

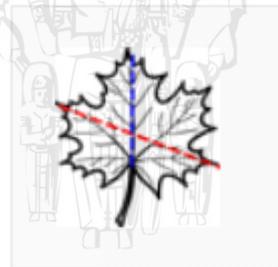
Keterangan:

A = Area

P = Perimeter

8. *Rectangular (N)*

Rectangular merupakan fitur yang merepresentasikan bagaimana bentuk persegi panjang. Gambar untuk fitur ini dapat dilihat pada Gambar 2.15.



Gambar 2.15 Fitur Bentuk *Rectangular*

Rumus untuk mendapatkan nilai pada fitur ini yaitu menggunakan Persamaan 2.7.

$$N = \frac{A}{LW} \tag{2.7}$$

Keterangan:

A = Area

L = Panjang

W = Lebar

9. Rasio Perimeter L (PL)

Rasio perimeter pada panjang (L) ditunjukkan pada Gambar 2.16.



Gambar 2.16 Fitur Bentuk PL

Untuk mendapatkan nilai dari PL dapat menggunakan rumus pada Persamaan 2.18.

$$PL = \frac{P}{L} \tag{2.8}$$

L = Panjang

P = Perimeter

10. Rasio Perimeter L dan W (PLW)

Rasio perimeter L dan W didapatkan dari jumlah dari sumbu L dan W sebagai pembagi. Ilustrasi ditunjukkan pada Gambar 2.17. Rumus yang dipakai untuk mendapatkan nilai fitur PLW dapat menggunakan Persamaan 2.9.

$$PL = \frac{P}{L+W} \tag{2.9}$$



Gambar 2.17 Fitur Bentuk PLW

Keterangan:

L = Panjang

W =Lebar

P = Perimeter

2.6 Perhitungan Jarak Data

Perhitungan jarak menggunakan *Euclidean* yang merupakan salah satu metode klasifikasi yang memiliki tingkat akurasi tinggi, yaitu sebuah metode yang mengkalkulasikan jarak untuk mengetahui tingkat kemiripan antar data. Kemudian data yang memiliki kemiripan terdekat akan dimasukkan di kelompok yang sama. Penentuan nilai k dapat dilakukan dengan dua metode yaitu secara



acak ataupun secara pasti (Made & Iswari, 2017). Untuk dapat menentukan jarak terdekat objek berdasarkan data pembelajaran (Nursalim, Suprapedi, & H.himawan, 2014). Langkah-langkah untuk melakukan perhitungan jarak adalah seperti berikut.

1. Normalisasi

Sebelum melakukan perhitungan jarak dengan *Euclidean* perlu dilakukan normalisasi terhadap data. Tujuan dilakukannya normalisasi ialah untuk mempersempit *range* pada data. Normalisasi yang dipakai dalam penelitian ini adalah *Min-max normalization* (MMN). Hasil dari normalisasi dengan MMN, data akan berada pada *range* 0 sampai 1. Dengan nilai terkecil adalah 0 dan nilai terbesar adalah 1 (Chandrasekhar, Thangavel, & Elayaraja, 2011). Menurut Mutrofin, Kurniawardhani, Izzah, & Masrur, (2014), Rumus untuk melakukan normalisasi dengan MMN, ditunjukkan pada Persamaan 2.10.

$$\text{Normalisasi} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2.10)$$

Keterangan:

x = data latih

$\min(x)$ = nilai x terkecil

$\max(x)$ = nilai x terbesar

2. Menghitung Jarak

Menurut Mustafa & Zhu (2014) terdapat persamaan yang dapat digunakan untuk mengetahui dekat atau jauhnya tetangga, yaitu menghitung jarak dengan *Euclidean* ditunjukkan pada Persamaan 2.11.

$$d_{\text{Euclidean}}(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \quad (2.11)$$

Ketrangan:

d = jarak euclidean

i = variabel data

x_i = data latih

y_i = data uji

2.7 Sistem Temu Kembali Informasi

Sistem temu kembali informasi merupakan sebuah sistem yang dipakai untuk menemukan kembali informasi yang sesuai atau relevan terhadap kebutuhan pengguna pada sebuah kumpulan informasi yang dilakukan secara otomatis (Bunyamin, Negara, Informasi, & Maranatha, 2008)

Pada sistem *information retrieval* ini terdapat dua proses operasi, proses pertama dimulai dari koleksi dokumen yang berisi resep makanan dan proses kedua dimulai dari query yang dimasukkan oleh pengguna berupa citra dari

makanan yang ingin dicari resepnya. Setiap dokumen yang berada pada *document collection (folder file)* akan dibaca oleh sistem. *Output* dari sistem *information retrieval* ini adalah list dokumen tentang resep makanan yang sesuai dengan pencarian yang dimasukan oleh pengguna, daftar yang ditampilkan dimulai dari dokumen yang memiliki bobot tertinggi yang sesuai dengan tingkat kemiripan dengan *query* hingga bobot dokumen yang terendah.

2.8 Pengujian

Pengujian merupakan sebuah langkah untuk mengetahui tingkat keberhasilan sistem dengan metode yang digunakan yaitu sistem temu kembali informasi resep makanan. Pengujian sistem dilakukan dengan melakukan perhitungan *Mean Average Precision (MAP)*. Dalam ranah sistem temu kembali, dokumen dikembalikan sebanyak jumlah k . *Average Precision (AP)* hanya menghitung *precision* dari dokumen yang relevan dan hasilnya dibagi dengan jumlah dari dokumen yang terlibat. *Precision* adalah *Precision* ialah dokumen yang relevan dibandingkan dengan jumlah dokumen yang ditemu balik dalam penelusuran. Pengukuran MAP ialah hasil dari rata-rata dokumen relevan yang di tampil dari setiap *query* yang terlibat yang mana dokumen yang tidak relevan bernilai 0. Menurut Manning, Raghavan, & Schütze (2009) rumus untuk mendapatkan nilai MAP dapat dihitung dengan Persamaan 2.12.

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} P(R_{jk}) \quad (2.12)$$

Keterangan:

Q = jumlah *query*

P = *precision*

R_{jk} = nilai *precision* yang telah dilakukan pemeringkatan

BAB 3 METODOLOGI

Bab Metodologi berisi pembahasan mengenai tipe penelitian, proses dan langkah-langkah penelitian, jadwal penelitian, tempat penelitian, teknik pengumpulan data, peralatan pendukung penelitian, dan teknik analisis data.

3.1 Tipe Penelitian

Berdasarkan panduan skripsi Fakultas Ilmu Komputer, Universitas Brawijaya tahun 2018, penelitian ini menggunakan tipe non-implementatif. Pelaksanaan tipe ini menitikberatkan pada situasi tertentu atau melakukan analisis terhadap hubungan antar suatu fenomena yang dikaji yang kemudian memberikan hasil analisis ilmiah sebagai produk utama. Penelitian nonimplementatif terbagi menjadi dua pendekatan yaitu deskriptif dan analitik. Tipe penelitian ini merupakan nonimplementatif dengan pendekatan analitik.

Analitik atau biasa disebut *explanatory* merupakan sebuah kegiatan penelitian nonimplementatif yang dikerjakan untuk memberikan penjelasan mengenai derajat hubungan antar elemen pada objek penelitian dengan fenomena atau situasi tertentu. Tujuan dari penelitian dengan tipe ini ialah untuk memperoleh hasil analisis.

3.2 Gambaran Alur Sistem

Sistem yang dibuat berupa rekomendasi pencarian resep makanan berdasarkan ekstraksi fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors* menggunakan *Euclidean* sebagai metode perhitungan jarak.



Gambar 3.1 Gambaran Alur Sistem

Secara garis besar masukan, proses, dan keluaran yang diberikan sistem adalah sebagai berikut dan dapat dilihat pada Gambar 3.1.

1. Masukan pada sistem berupa *query* dari citra makanan yang ingin di cari resep makanannya.
2. Kemudian proses yang dilakukan meliputi proses *pre-processing*, ekstraksi fitur RGB *Color Channel* dan *Simple Morphological Shape Descriptors*, kemudian jarak kemiripan antar data dihitung dengan *Euclidean*. Hasil dari perhitungan

jarak kemudian dilakukan proses pencarian melalui proses *information retrieval*.

3. Keluaran yang diberikan sistem adalah sebuah hasil pencarian resep makanan berdasarkan query citra yang dimasukkan. Hasil berupa daftar resep makanan dari dokumen yang memiliki bobot tertinggi sesuai dengan tingkat kemiripan dengan *query input* hingga bobot dokumen terendah.

3.3 Lokasi Penelitian

Penelitian dilakukan di Laboratorium Riset, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

3.4 Teknik Pengumpulan Data

Data yang digunakan pada penelitian ini berupa data primer. Data primer merupakan data yang diambil secara langsung atau merupakan data yang berasal dari sumber pertama (Hasibuan, 2007).

Tabel 3.1 Dataset Citra Makanan

No	Kategori Makanan	Nama Makanan	Citra	Jumlah Data Latih
1	Roti	Donat		34
		Roti Gandum		37
		Roti Tawar		37
2	Mi Instan	Mi Goreng Keriting		8

No	Kategori Makanan	Nama Makanan	Citra	Jumlah Data Latih
		Mi Goreng Gepeng		8
3	Telur	Telur Mata Sapi		32
		Telur Dadar		12
4	Daging	Rendang		8
		Fried Chicken		25
5	Nasi	Nasi Kuning		6
		Nasi Merah		7

No	Kategori Makanan	Nama Makanan	Citra	Jumlah Data Latih
		<i>Beng-beng</i>		35
		<i>Soba Mi</i>		8
		<i>Keripik Kentang</i>		8
		<i>Biskuit</i>		24
		<i>Genji Pie</i>		22

Data yang diperlukan berupa 16 kelas citra makanan tunggal yang terdiri dari 311 data latih dan 37 data uji. Data terbagi menjadi beberapa kategori makanan yaitu roti, mi goreng, daging, telur, nasi, dan makanan ringan. Data yang diambil secara lengkap ditunjukkan pada Tabel 3.2. Jumlah data latih pada setiap kelasnya berbeda-beda. Data diambil melalui kamera *smartphone iPhone* seri 6 dengan resolusi sebesar 8 MP dan diambil pada siang hari dengan cahaya dari luar ruangan pada pukul 10:00-12:00 WIB. Citra diambil secara tegak lurus dan pada berbagai tingkat kemiringan.

Data diperoleh secara langsung bersama dengan *Team Food Project* Fakultas Ilmu Komputer tahun 2018 dan pengambilan citra dilakukan di Gedung F, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

3.5 Peralatan Pendukung Penelitian

Untuk menunjang penelitian ini, digunakan beberapa piranti keras dan lunak antara lain:

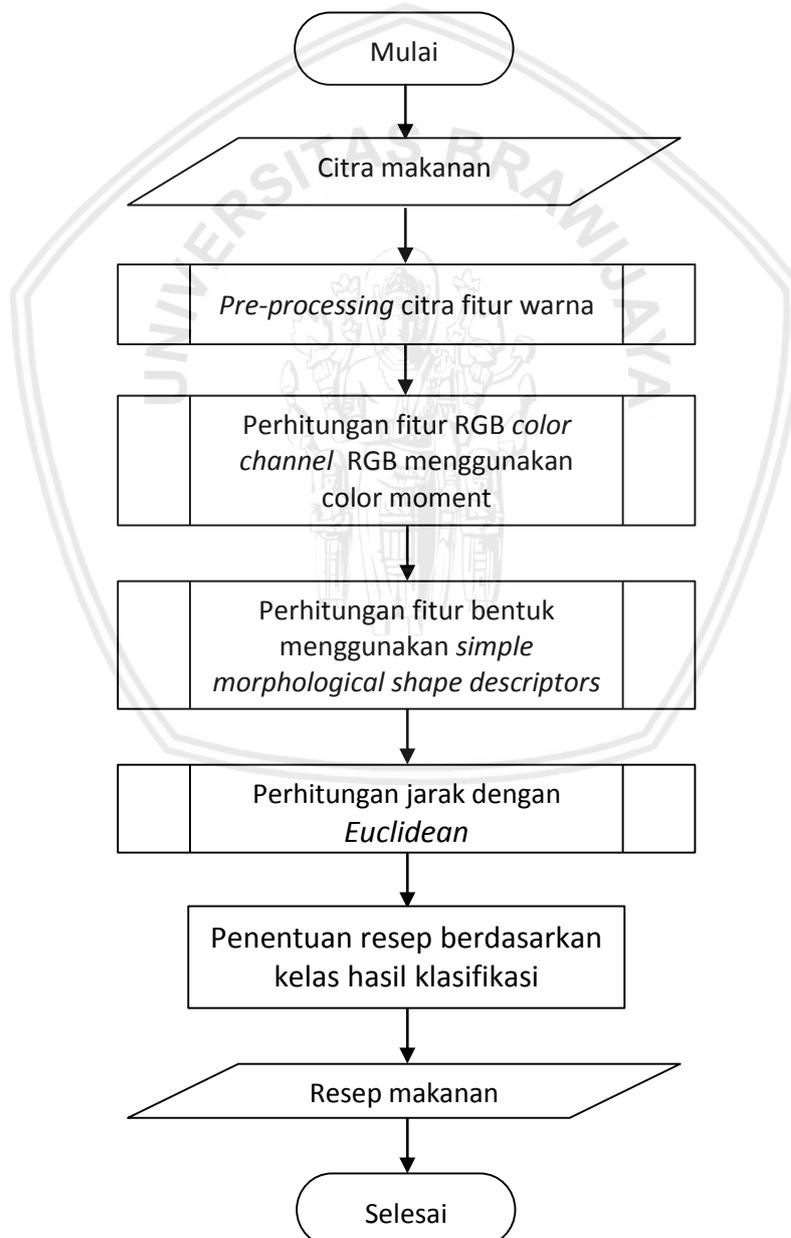
1. Smartphone Iphone Seri 6 32 GB, Kamera 8 MP
2. Laptop Asus Seri A455L, Intel Core i3 CPU @ 1.7 GHz, RAM 6 GB.
3. Sistem Operasi Windows 8.1 Pro.
4. Editor Sublime Text 3.
5. *Library* Python OpenCV.



BAB 4 PERANCANGAN ALGORITME

4.1 Perancangan Algoritme Sistem

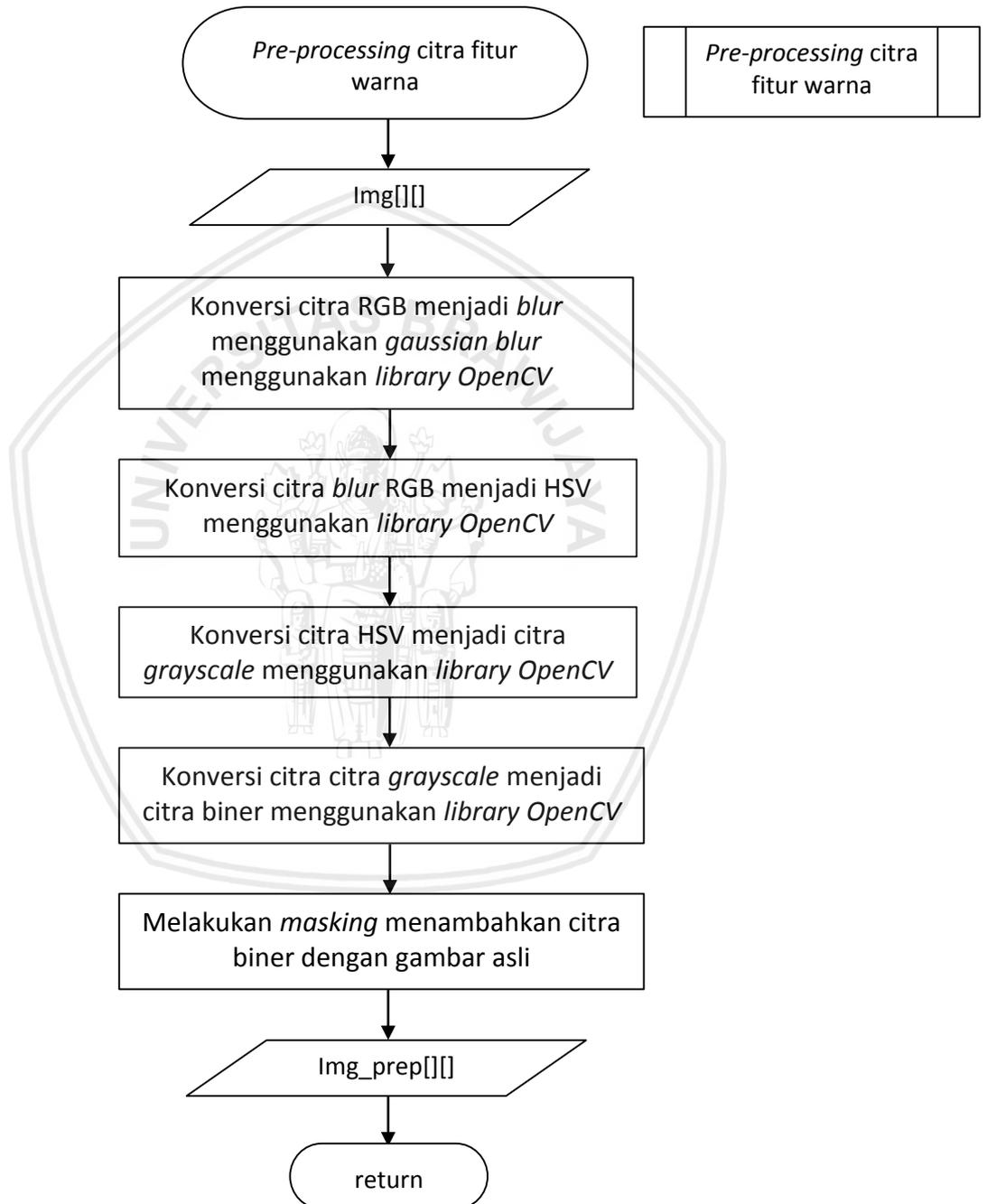
Rancangan algoritme sistem yang akan dibuat diawali dengan memasukkan *query* yaitu sebuah citra makanan tunggal. Kemudian pada citra dilakukan *pre-processing* dengan hasil citra memiliki *background* hitam. Dari *query* hasil segmentasi tersebut dilakukan pengambilan fitur citra yaitu warna dengan tiga fitur menggunakan ekstraksi fitur *color moment* dan bentuk dengan 13 fitur menggunakan ekstraksi fitur *simple morphological shape descriptors*. Kemudian fitur yang telah didapatkan dihitung kemiripan data menggunakan *Euclidean*. Diagram alir sistem secara keseluruhan ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram Alir Sistem

4.2 Perancangan Algoritme *Pre-processing* Fitur Warna

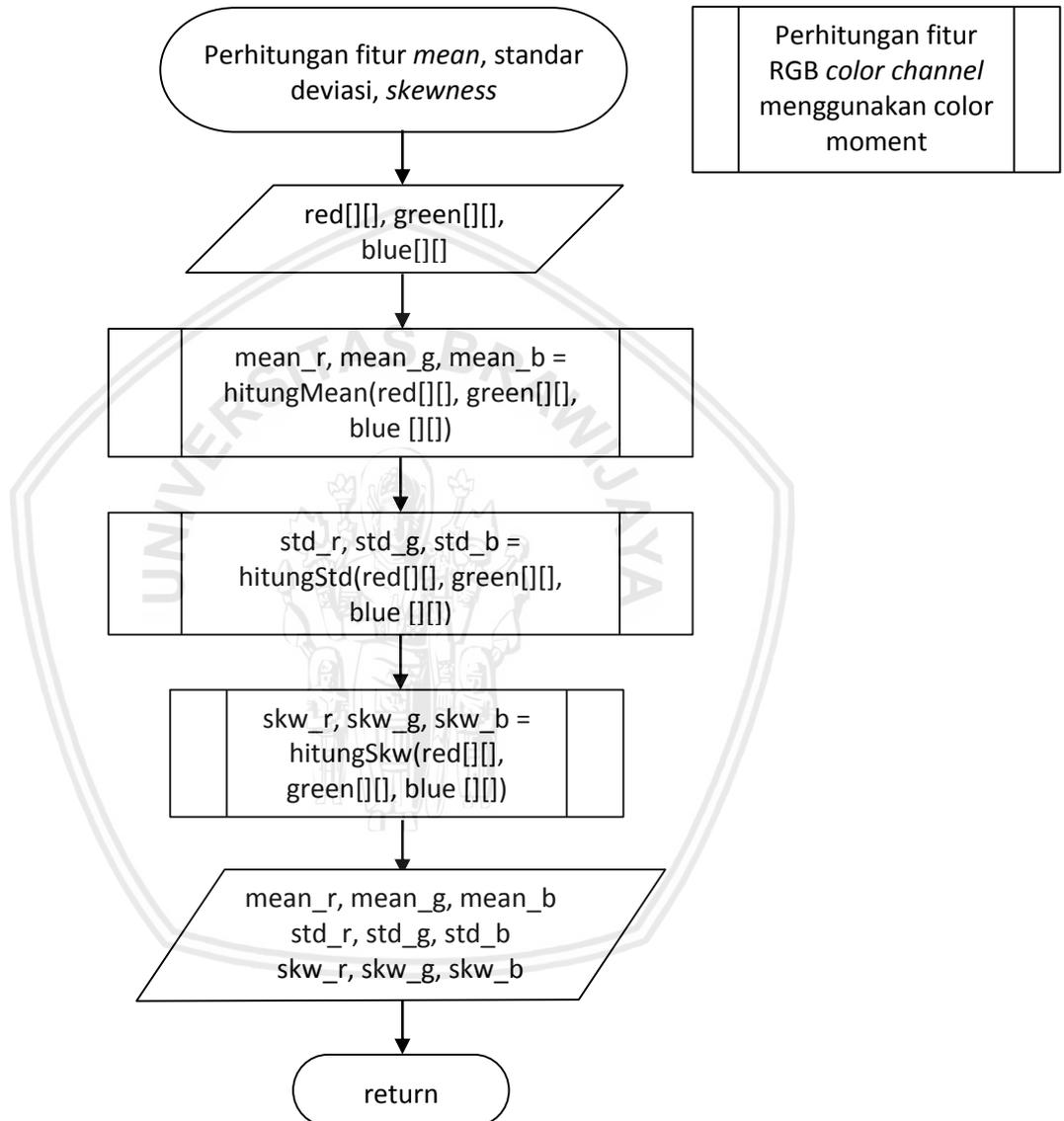
Pada *pre-processing* untuk ekstraksi fitur warna, dilakukan proses untuk menghasilkan citra dengan objek yang memiliki *background* berwarna hitam seperti yang ditunjukkan pada Gambar 2.1 (b). Hal tersebut dilakukan agar perhitungan warna hanya berfokus pada objek saja. Diagram alir untuk melakukan *pre-processing* fitur warna dapat dilihat pada Gambar 4.2.



Gambar 4.2 Diagram Alir *Pre-processing* Fitur Warna

4.3 Perancangan Algoritme Sistem

Perhitungan fitur pada RGB *color channel* menggunakan *color moment*. Proses diawali dengan mengambil nilai piksel pada setiap *channel*-nya. Kemudian dilakukan perhitungan *mean*, *standard deviation*, dan *skewness*. Nilai yang didapatkan akan dikembalikan untuk dijadikan sebagai sebuah fitur. Perancangan algoritme untuk proses ini ditunjukkan pada Gambar 4.3.

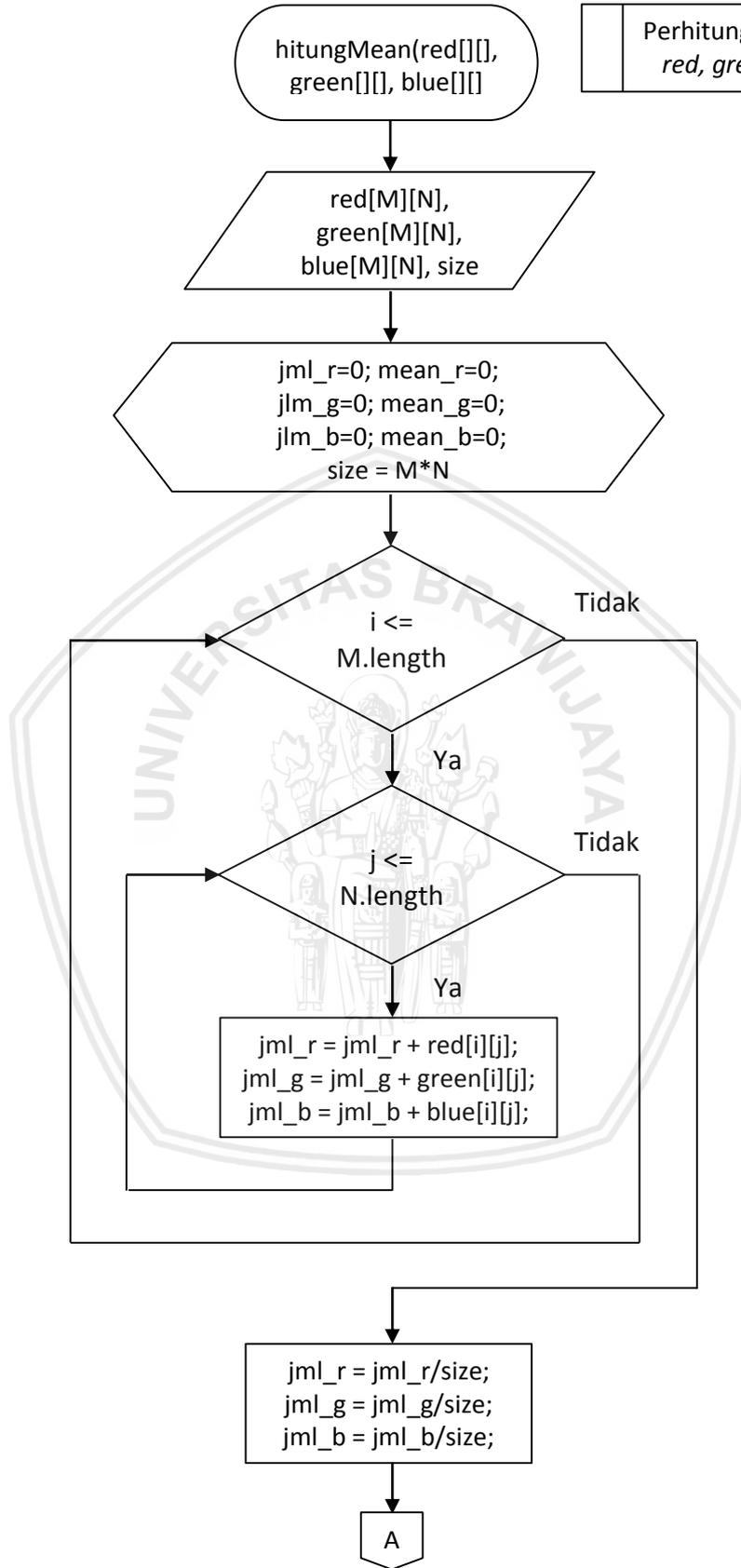


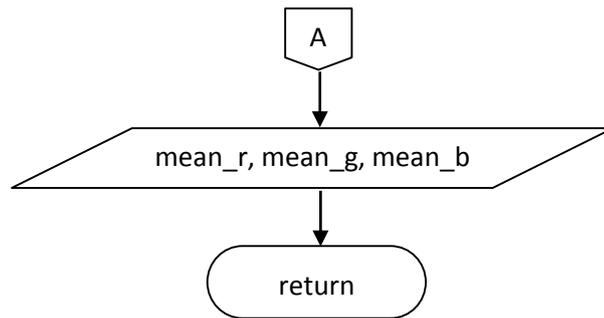
Gambar 4.3 Diagram Alir RGB Color Channel

4.3.1 Perancangan Algoritme Perhitungan Mean

Fitur warna pada *color moment* yang pertama adalah *mean* RGB. Perhitungan diawali dengan menjumlahkan seluruh nilai piksel pada masing-masing *channel* kemudian dibagi dengan ukuran piksel. Diagram alir untuk perhitungan *mean* atau rata-rata setiap *channel* warna dari citra makanan ditunjukkan pada Gambar 4.4.

Perhitungan <i>mean</i> <i>red, green blue</i>

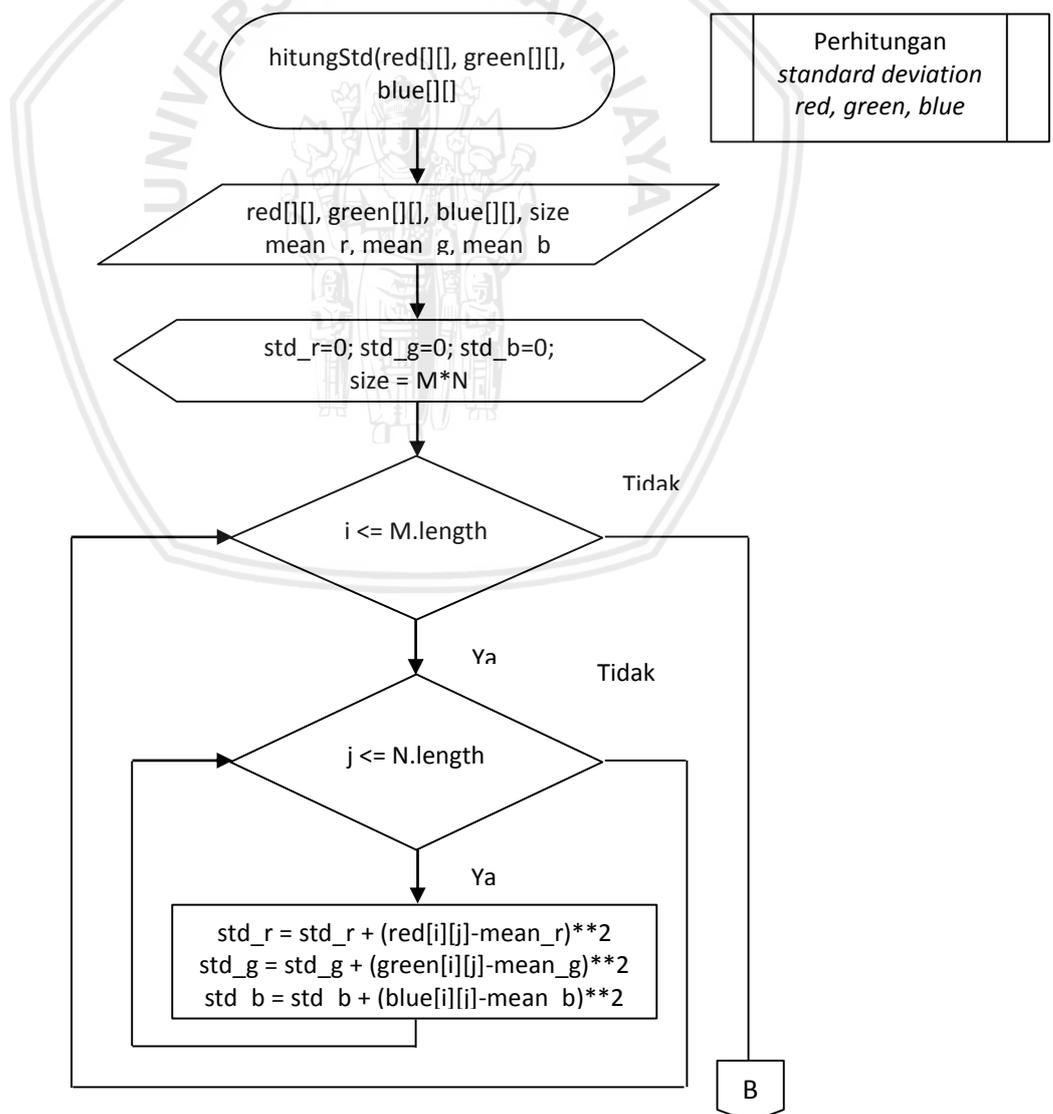


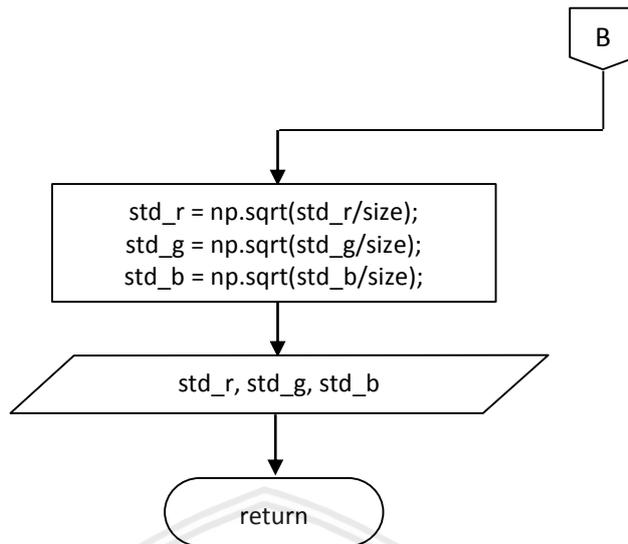


Gambar 4.4 Diagram Alir Menghitung Mean

4.3.2 Perancangan Algoritme Perhitungan Standard Deviation

Fitur warna pada *color moment* kedua adalah *standard deviation* setiap *channel* RGB. Perhitunganawali dengan menjumlahkan nilai kuadrat dari setiap piksel dikurangi *mean* dan kemudian membagikannya dengan ukuran citra. Hasil dari proses tersebut diakar kuadrat sehingga didapatkan nilai dari *standard deviation*. Diagram alir perhitungan ini dapat dilihat pada Gambar 4.5.

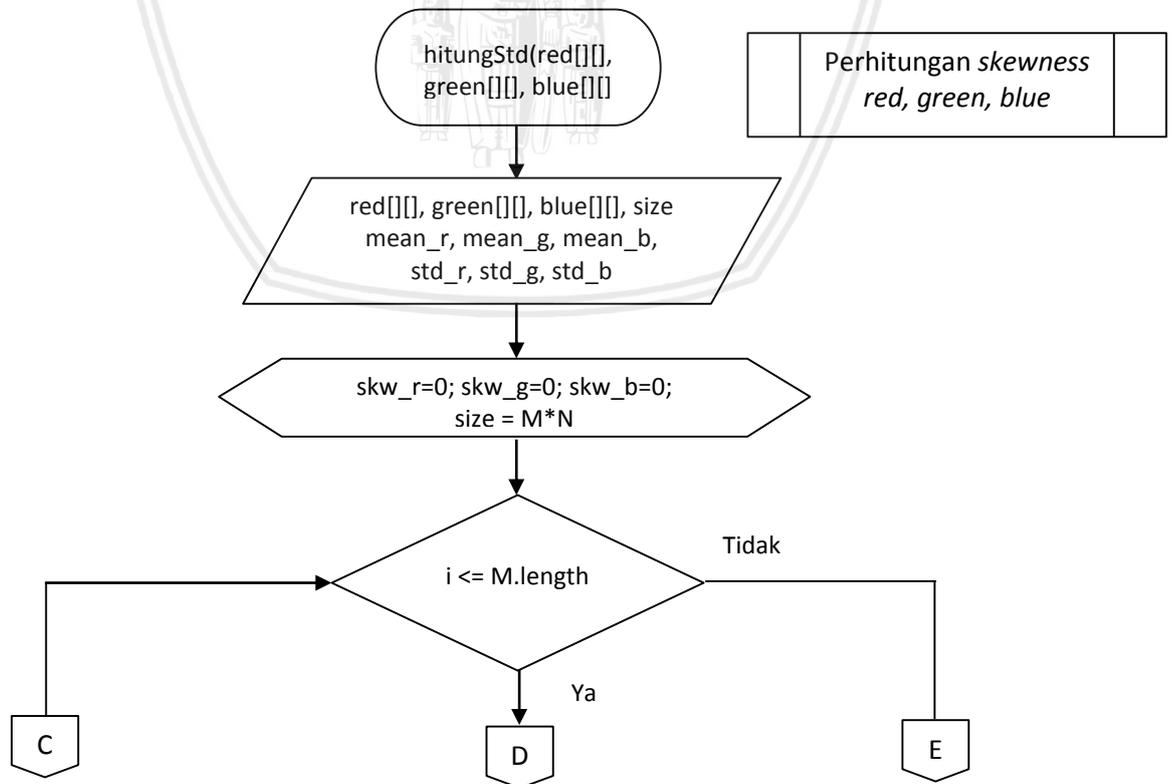


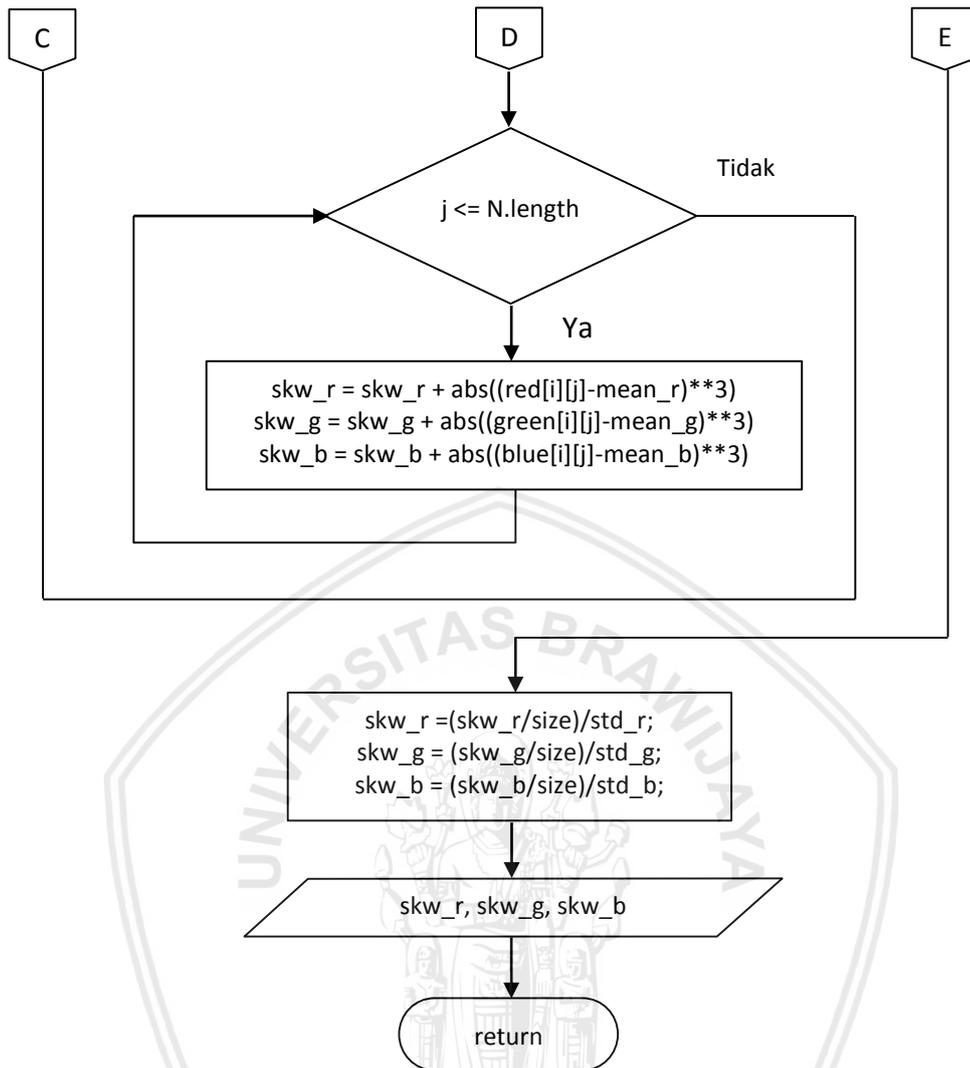


Gambar 4.5 Diagram Alir Menghitung *Standard deviation*

4.3.3 Perancangan Algoritme Perhitungan *Skewness*

Fitur warna *color moment* yang ketiga adalah *skewness* RGB. Perhitungan dilakukan dengan memberi pangkat tiga pada setiap nilai piksel dikurangi *mean* dan membagikannya dengan ukuran citra. Hasil dari perhitungan tersebut kemudian dibagikan dengan nilai *standard deviation* yang telah diperoleh sebelumnya. Diagram alir untuk proses ini ditunjukkan pada Gambar 4.6.

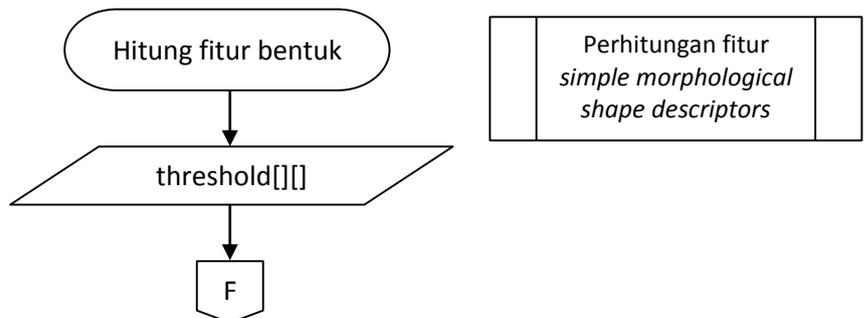


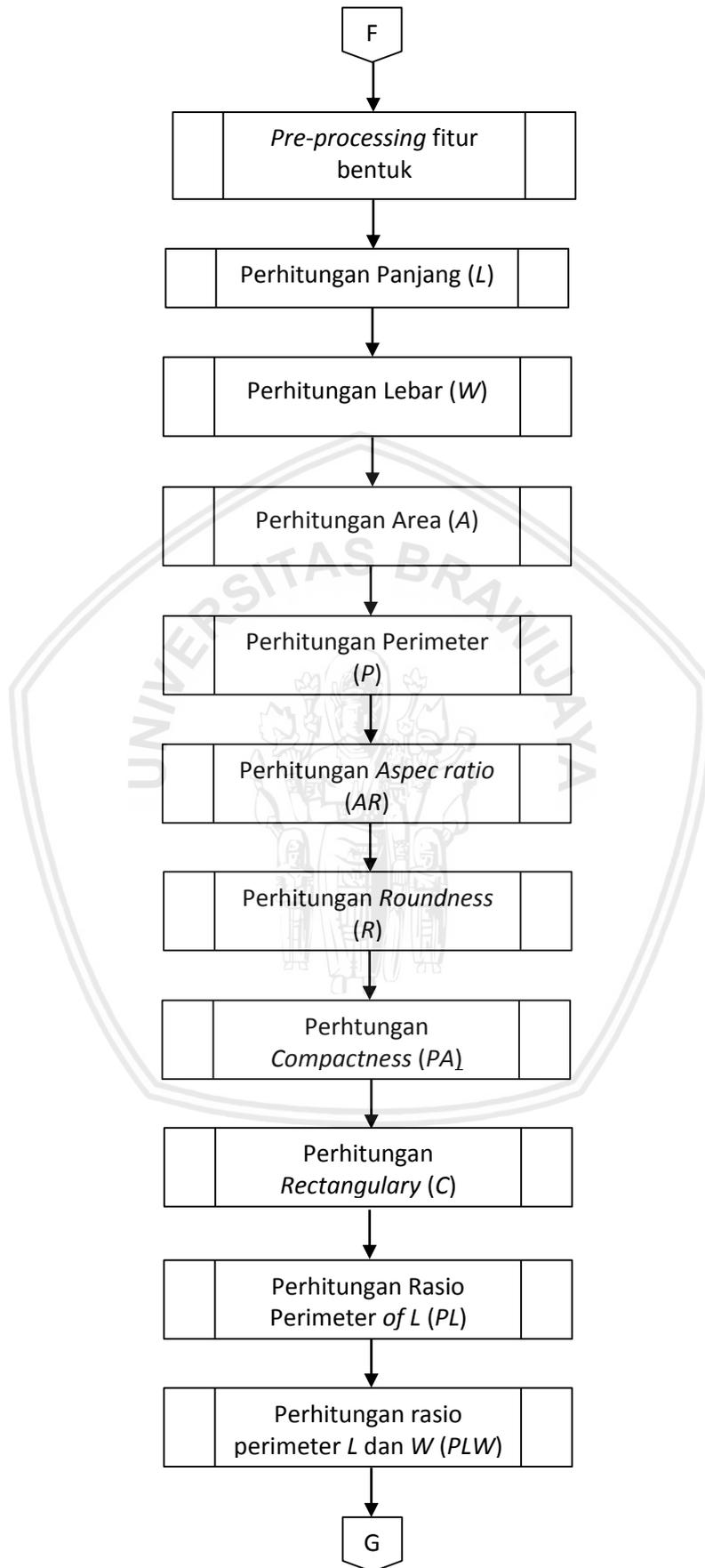


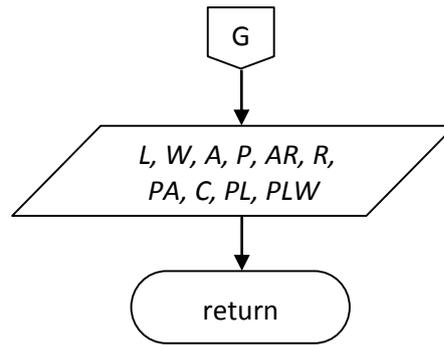
Gambar 4.6 Diagram Alir Menghitung *Skewness*

4.4 Perancangan Algoritme *Simple Morphological Shape Descriptors*

Diagram alir untuk perhitungan fitur bentuk menggunakan ekstraksi fitur *simple morphological shape descriptors* ditunjukkan pada Gambar 4.7. Pada proses ini dilakukan perhitungan untuk mendapatkan nilai pada setiap fiturnya.



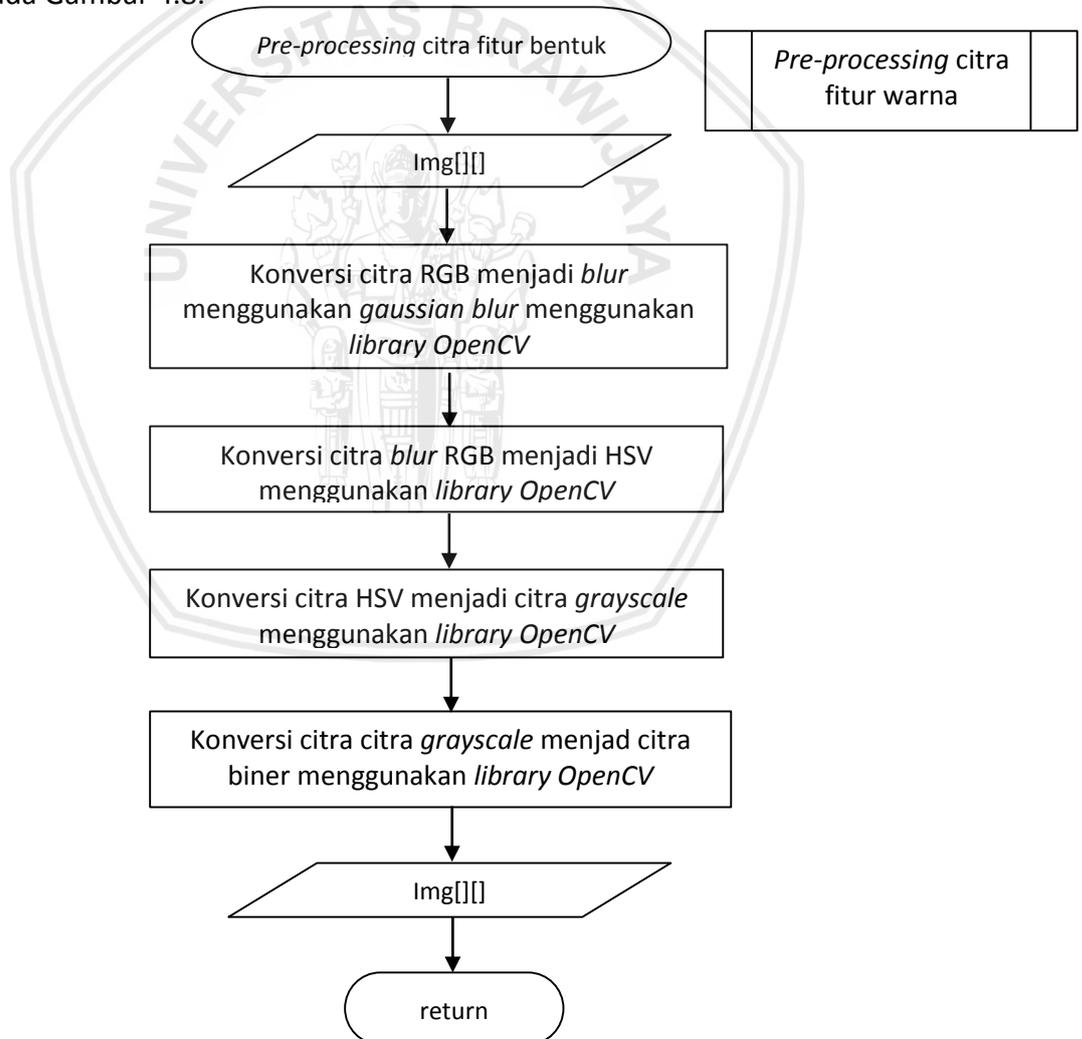




Gambar 4.7 Diagram Alir *Simple Morphological Shape Descriptors*

4.4.1 Perancangan Algoritme Perhitungan Panjang

Perbedaan *pre-processing* pada fitur warna dan bentuk ialah pada fitur bentuk proses berhenti sampai konversi citra biner. Diagram alir proses ini dapat dilihat pada Gambar 4.8.

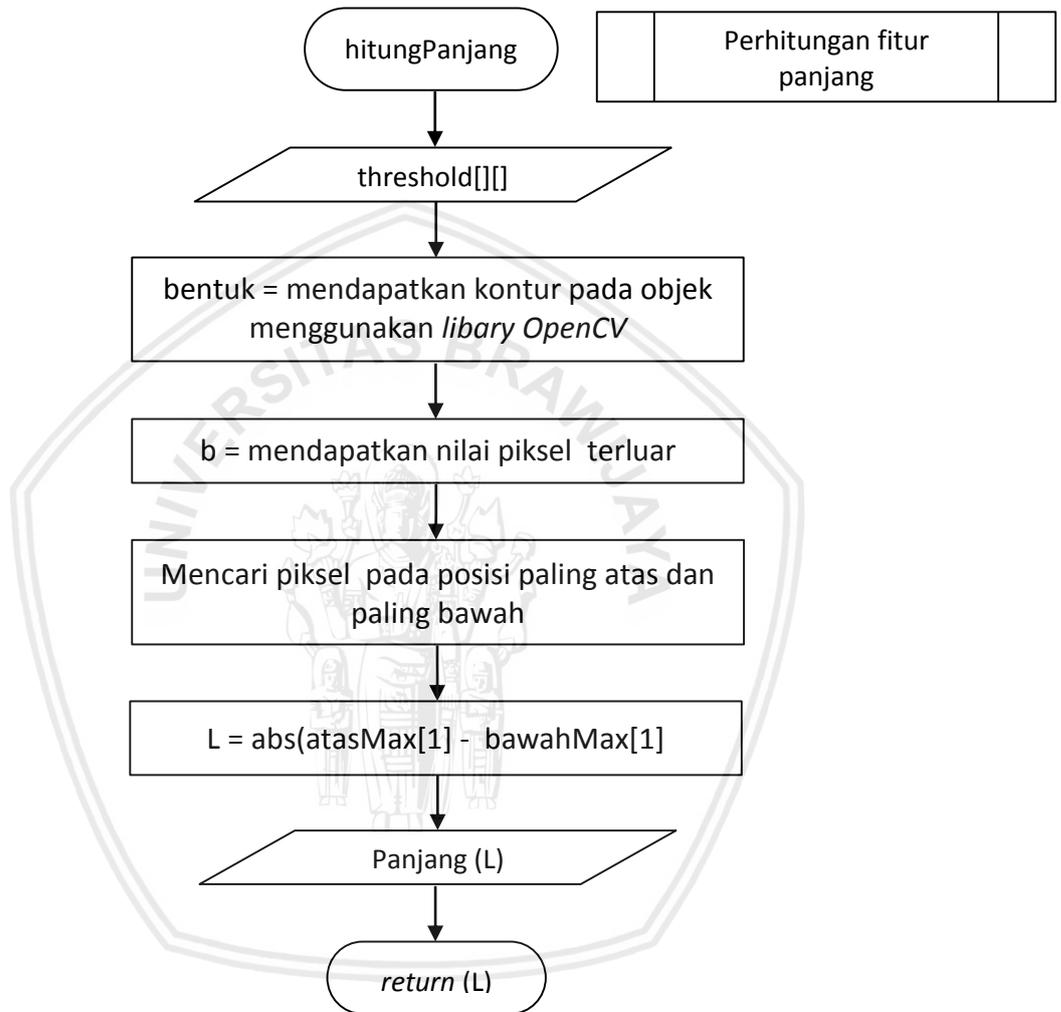


Gambar 4.8 Diagram Alir *Pre-processing Bentuk*



4.4.2 Perancangan Algoritme Perhitungan Panjang

Fitur bentuk yang pertama dihitung ialah panjang objek dari citra. Panjang objek merupakan nilai piksel paling atas dikurangi dengan nilai piksel paling bawah. Untuk mendapatkan nilai terluar dapat digunakan *source code* pada *Library OpenCV*. Diagram alir untuk mendapatkan fitur panjang ditunjukkan pada Gambar 4.9.

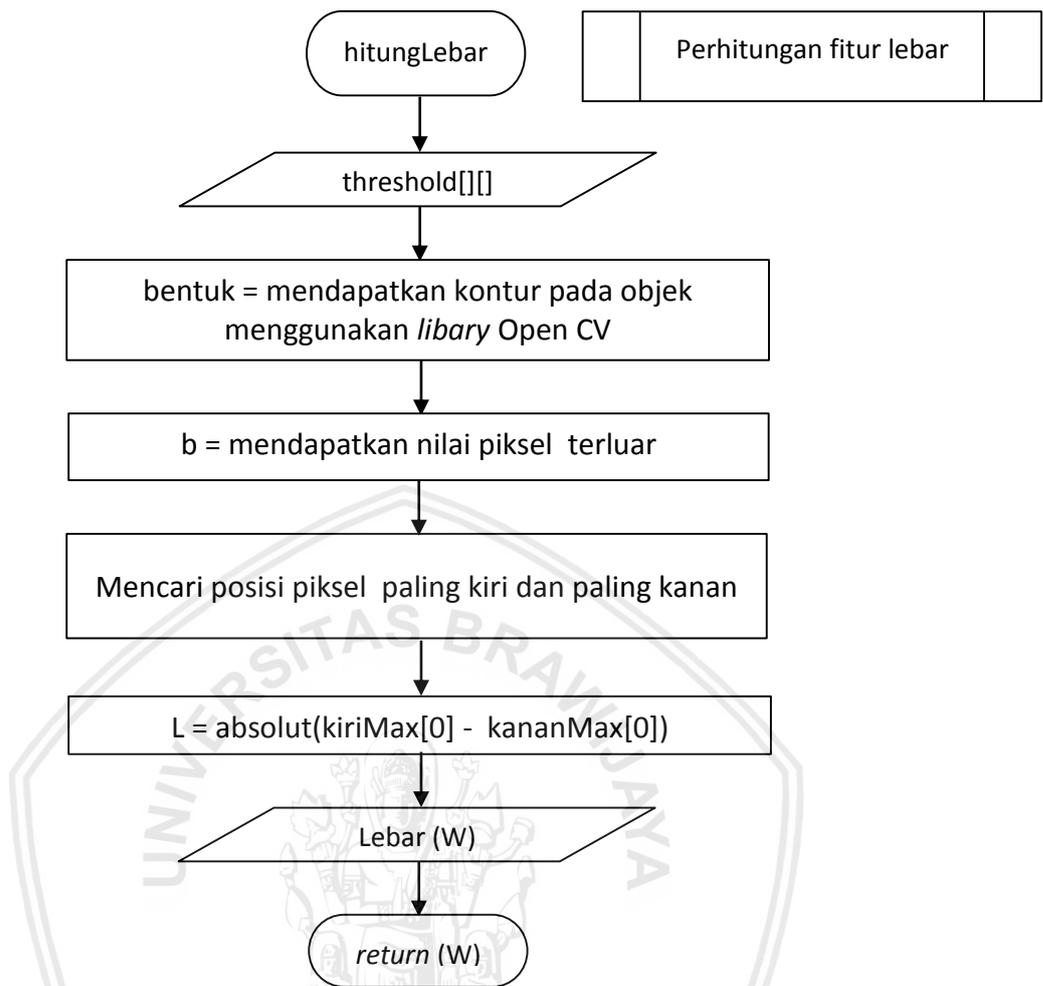


Gambar 4.9 Diagram Alir Menghitung Panjang

4.4.3 Perancangan Algoritme Perhitungan Lebar

Fitur lebar merupakan selisih dari nilai piksel paling kiri dikurangi dengan nilai piksel paling kanan. Untuk mendapatkan kontur dan nilai piksel pada pinggiran objek digunakan *Library OpenCV*. Diagram alir untuk mendapatkan fitur lebar pada objek dapat dilihat pada Gambar 4.10.

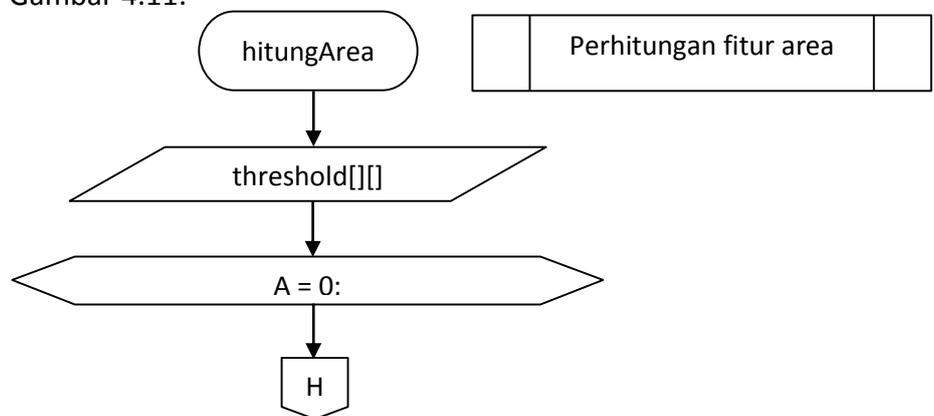


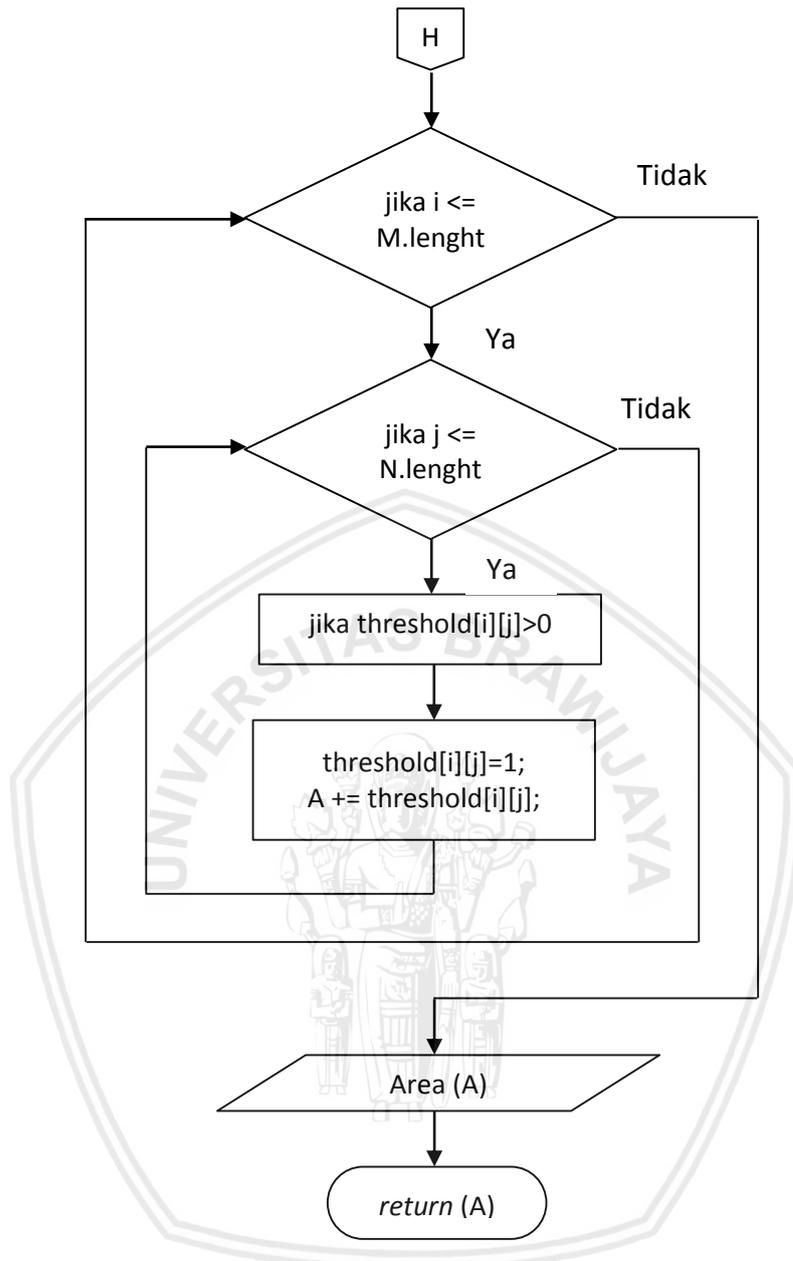


Gambar 4.10 Diagram Alir Menghitung Lebar

4.4.4 Perancangan Algoritme Perhitungan Area

Fitur selanjutnya adalah menghitung nilai area pada objek citra yang mana area berupa seluruh nilai piksel pada objek. Nilai area didapatkan dengan menjumlahkan nilai piksel pada citra yang sebelumnya dilakukan perubahan pada nilai lebih dari 0 menjadi 1. Diagram alir untuk mendapatkan area pada objek citra dapat dilihat pada Gambar 4.11.

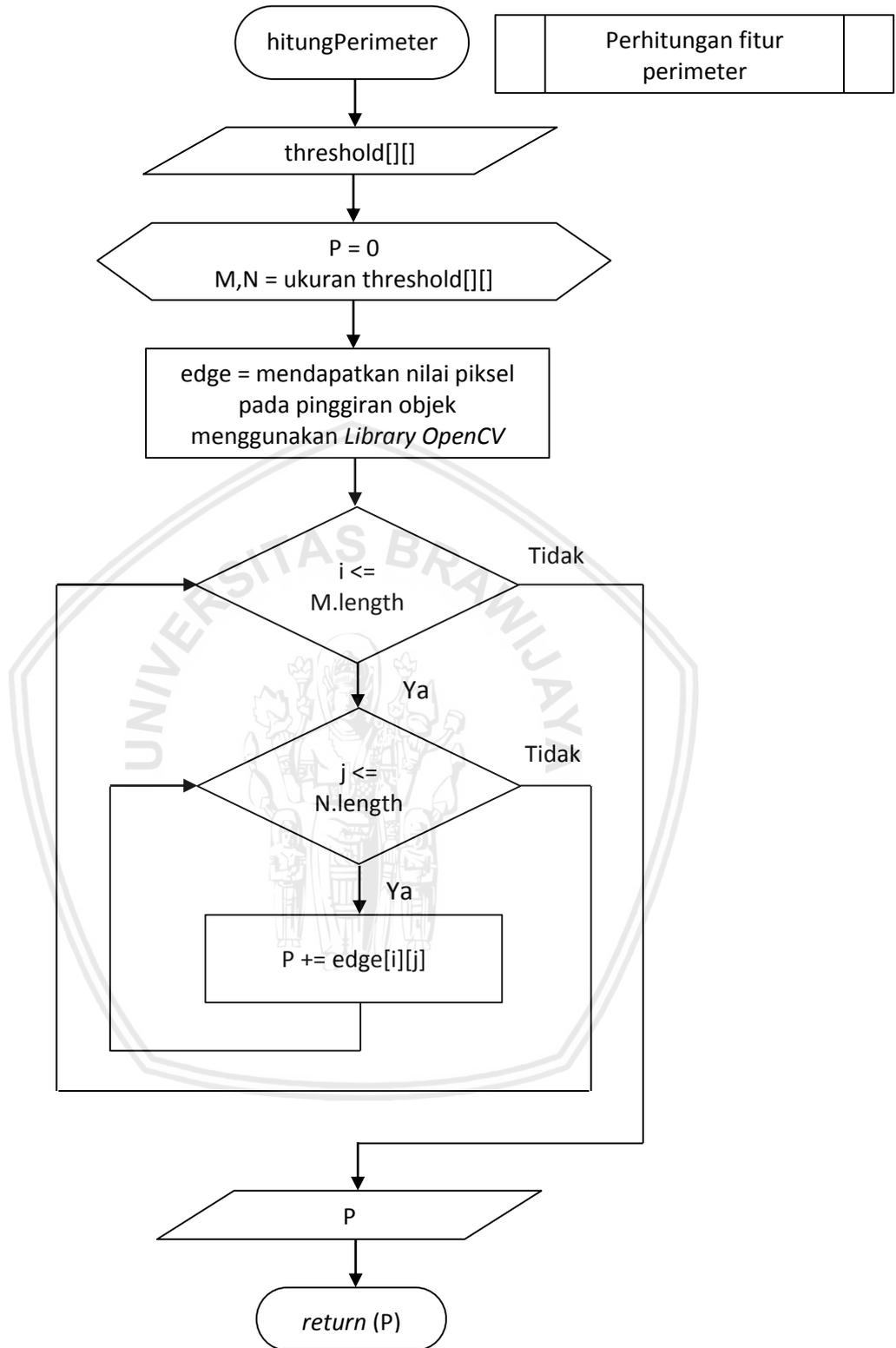




Gambar 4.11 Diagram Alir Menghitung Area

4.4.5 Perancangan Algoritme Perhitungan Perimeter

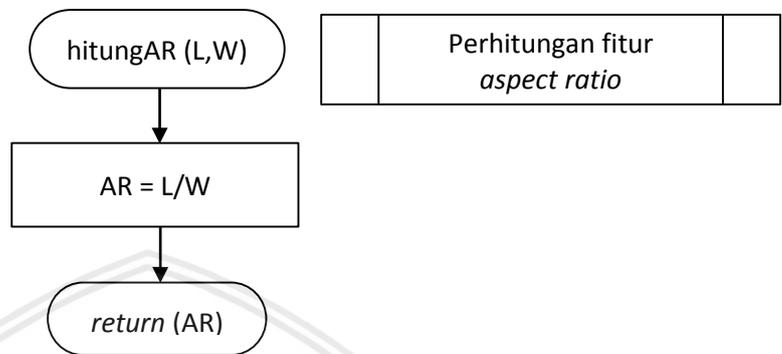
Berikutnya ialah fitur perimeter yang didapatkan dengan mengambil nilai pada pinggiran objek menggunakan *Library OpenCV*. Kemudian nilai yang telah didapatkan dijumlahkan dan nilainya dikembalikan sebagai fitur perimeter. Diagram alir untuk menghitung nilai perimeter dari objek pada citra dapat dilihat pada Gambar 4.12.



Gambar 4.12 Diagram Alir Menghitung Area

4.4.6 Perancangan Algoritme Perhitungan *Aspect Ratio*

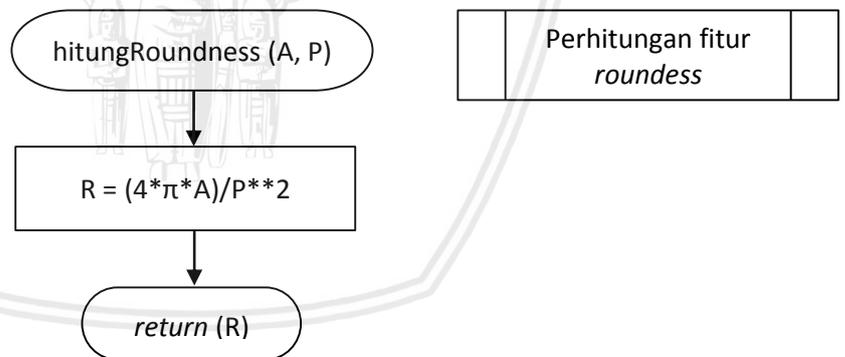
Selanjutnya untuk mendapatkan nilai *aspect ratio* dilakukan perhitungan dengan mengambil nilai L dan W . Nilai pada fitur ini diperoleh dengan membagikan nilai L dengan W . Diagram alir untuk mendapatkan nilai *aspect ratio* dapat dilihat pada Gambar 4.13.



Gambar 4.13 Diagram Alir menghitung *Aspect Ratio*

4.4.7 Perancangan Algoritme Perhitungan *Roundness*

Untuk mendapatkan nilai *roundness* dilakukan dengan menggunakan nilai area dan perimeter. Proses perhitungan dapat dilihat pada diagram alir yang ditunjukkan pada Gambar 4.14.



Gambar 4.14 Diagram Alir menghitung *Roundness*

4.4.8 Perancangan Algoritme Perhitungan *Compactness*

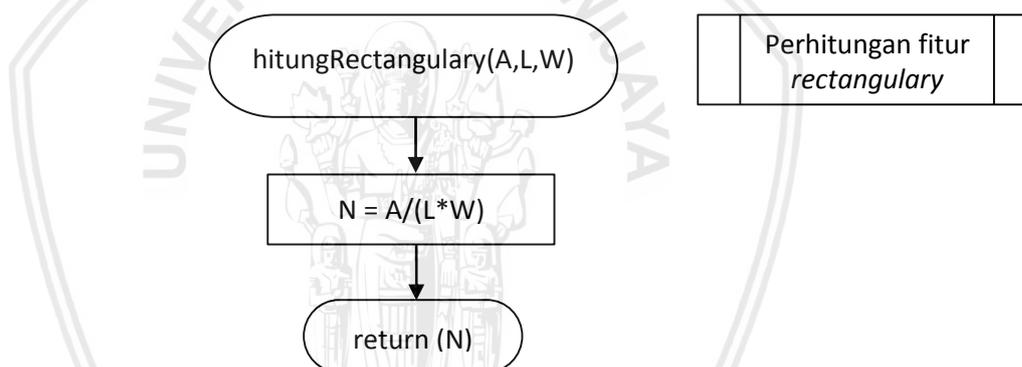
Untuk mendapatkan nilai dari *compactness* dilakukan perhitungan dengan mengambil nilai perimeter dan *area*. Nilai fitur ini didapatkan dengan membagikan perimeter dengan akar dari area. Diagram alir mencari nilai dari fitur *compactness* ditunjukkan pada Gambar 4.15.



Gambar 4.15 Diagram Alir Menghitung *Compactness*

4.4.9 Perancangan Algoritme Perhitungan *Rectangulary*

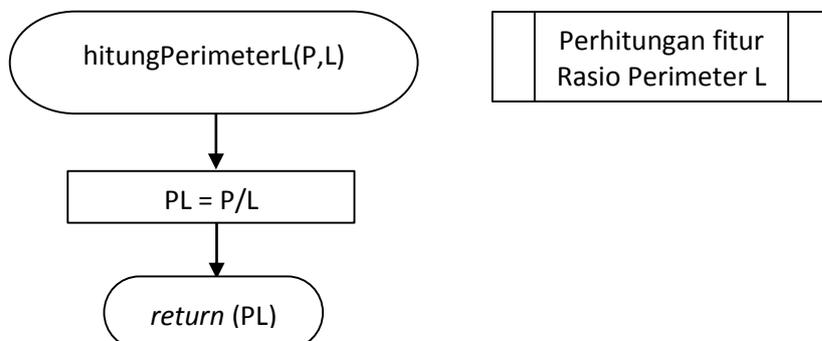
Untuk mendapatkan nilai dari *rectangulary* dilakukan perhitungan dengan menggunakan area, panjang dan lebar sebagai parameter. Diagram alir mencari untuk mendapatkan nilai dari fitur *compactness* ditunjukkan pada Gambar 4.16.



Gambar 4.16 Diagram Alir menghitung *Rectangulary*

4.4.10 Perancangan Algoritme Perhitungan Rasio Perimeter *L*

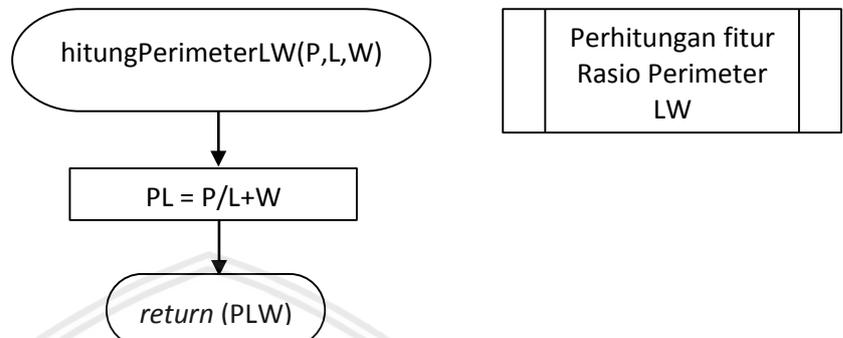
Fitur objek pada citra selanjutnya adalah rasio perimeter *L*. Perhitungan untuk mendapatkan nilai dari rasio perimeter *L* ditunjukkan pada Gambar 4.17.



Gambar 4.17 Diagram Alir Menghitung Rasio Perimeter *L*

4.4.11 Perancangan Algoritme Perhitungan Rasio Perimeter L

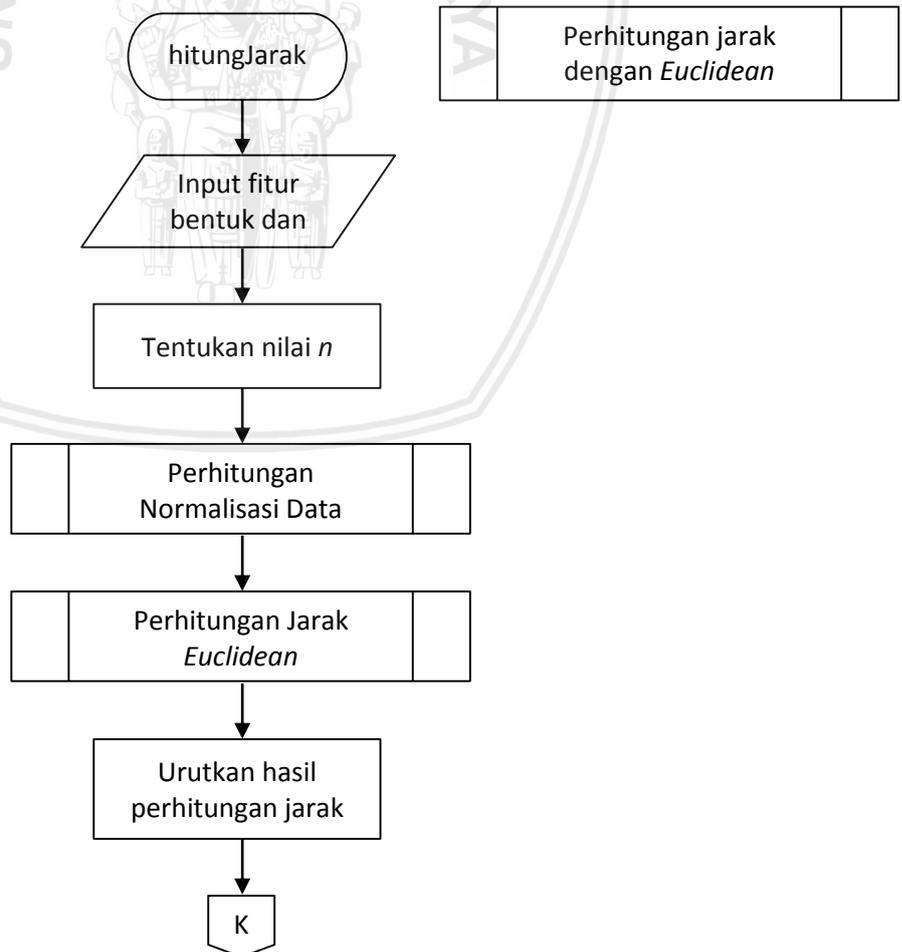
Fitur terakhir objek pada citra adalah rasio perimeter L dan W yaitu dengan membagikan nilai perimeter dengan hasil perkalian panjang dan lebar. Perhitungan untuk mendapatkan nilai dari rasio perimeter L dan W ditunjukkan pada Gambar 4.18.

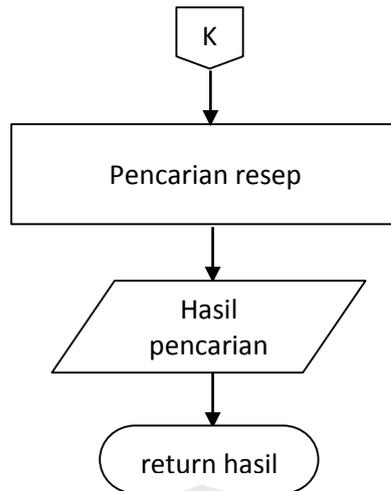


Gambar 4.18 Diagram Alir Menghitung Rasio Perimeter L dan W

4.5 Rancangan Algoritme Perhitungan Jarak

Rancangan algoritme untuk melakukan dengan perhitungan jarak menggunakan *Euclidean* dapat dilihat pada Gambar 4.19.

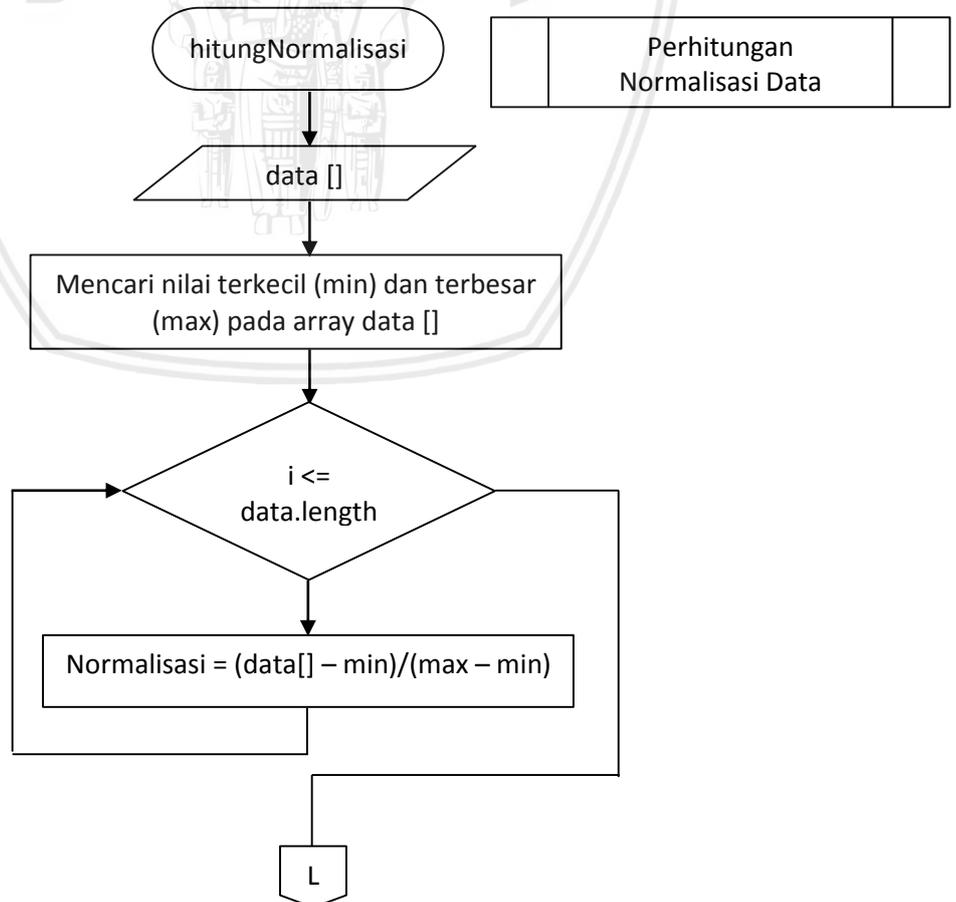




Gambar 4.19 Diagram Alir Perhitungan Jarak

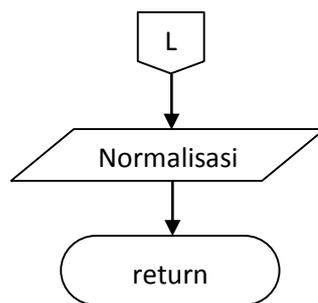
4.5.1 Rancangan Algoritme Perhitungan Normalisasi

Diagram alir untuk menghitung nilai normalisasi pada setiap fitur menggunakan MMN dapat dilihat pada Gambar 4.20. Perhitungan dilakukan dengan terlebih dahulu mencari nilai terbesar dan terkecil pada setiap fitur. Kemudian nilai setiap fitur dikurangi dengan nilai terkecil lalu dibagi dengan nilai selisih terbesar dan terkecil.



	Perhitungan Normalisasi Data	
--	------------------------------	--

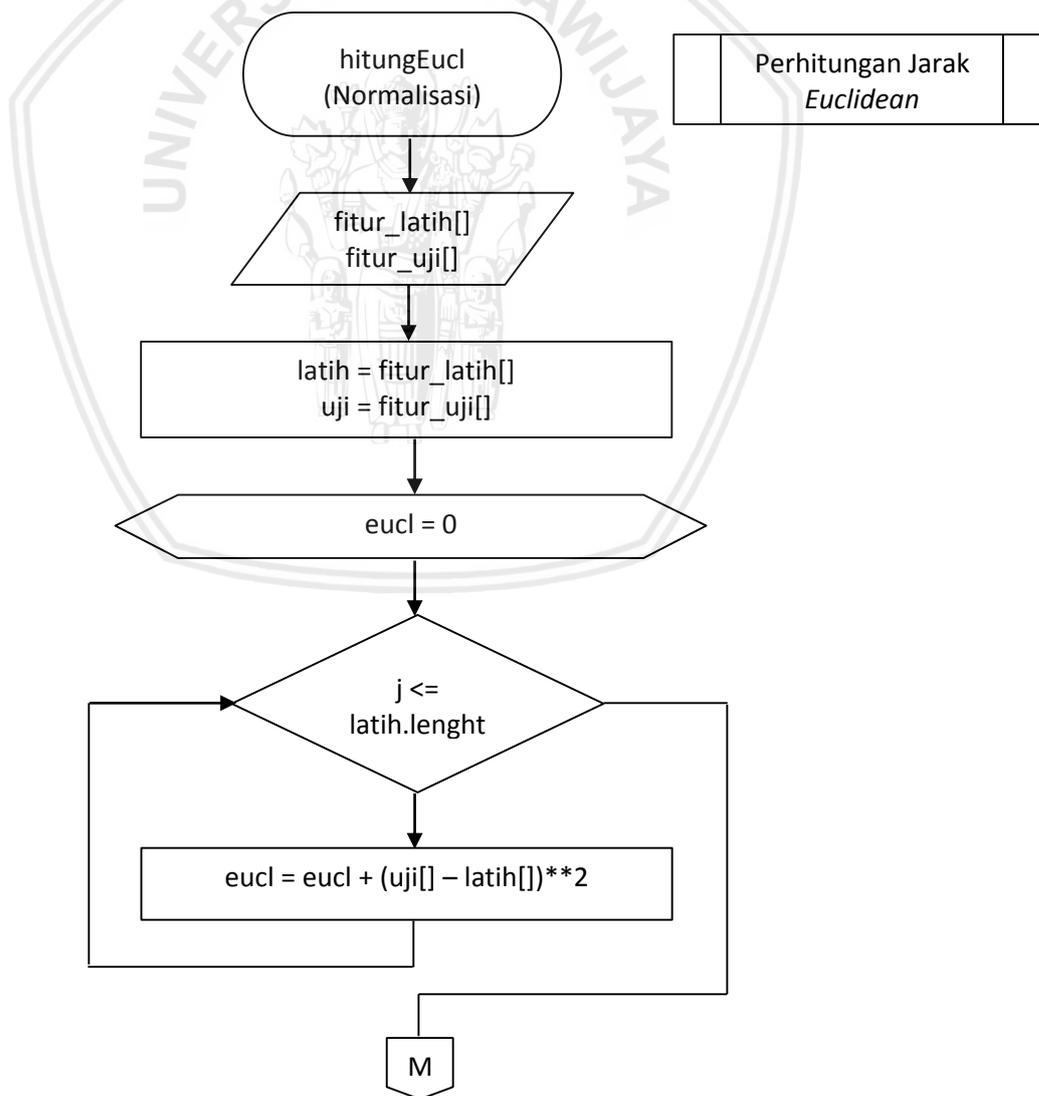


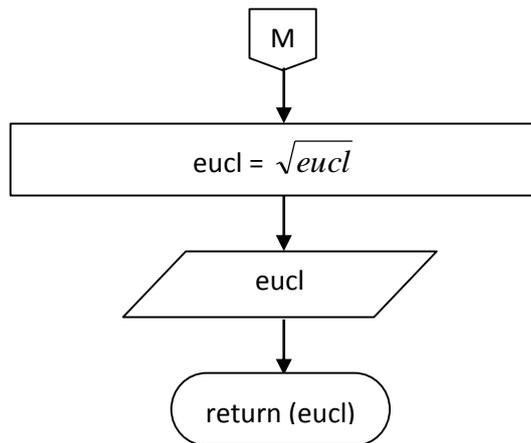


Gambar 4.20 Diagram Alir Perhitungan Manualisasi

4.5.2 Rancangan Algoritme Perhitungan Jarak *Euclidean*

Diagram alir untuk menghitung jarak antara data uji dengan seluruh data latih dapat dilihat pada Gambar 4.21. Data uji diambil berdasarkan hasil normalisasi pada keseluruhan data. Misal terdapat 12 data yang dinormalisasi. Maka data dibagi menjadi data uji dan data latih. Sebagai contoh jika ditentukan 10 data sebagai data latih, maka 10 data sisa merupakan data uji.





Gambar 4.21 Diagram Alir Perhitungan Jarak *Euclidean*

4.6 Perhitungan Manualisasi

Perhitungan manualisasi merupakan sebuah proses untuk memperlihatkan proses perhitungan secara manual pada setiap fitur yang digunakan pada penelitian ini. Citra sampel dipakai ialah citra dengan ukuran lebih kecil dibandingkan dengan gambar aslinya. Data pada proses manualisasi berupa data latih sejumlah 10 dan data uji sejumlah 1.

4.6.1 Manualisasi RGB *Color Channel*

Untuk mendapatkan nilai dari fitur pada RGB *color channel*, dilakukan perhitungan pada fitur *mean*, *standard deviation*, dan *skewness* pada masing-masing *channel* RGB. Setiap fitur dilakukan perhitungan pada masing-masing *channel*. Sehingga akan didapatkan 9 fitur warna untuk masing-masing citra. Sampel yang digunakan untuk melakukan perhitungan manualisasi merupakan citra dengan ukuran piksel 12x9. Nilai piksel dari *channel red*, *green*, *blue* pada citra ditunjukkan dan disajikan secara berturut-turut pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3.

Tabel 4.1 Nilai *Red* pada Citra Sampel

i,j	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	47	53	47	74	61	3	0	0	0
2	0	0	1	38	65	55	65	30	70	61	0	0
3	0	0	33	42	47	40	37	41	42	32	0	0
4	0	0	32	42	41	37	49	64	36	43	3	0
5	0	1	32	48	32	37	39	56	46	32	0	0
6	0	3	24	38	29	39	37	31	36	19	0	0
7	0	2	28	25	33	26	31	21	17	21	0	0
8	0	0	0	10	45	43	34	55	14	12	0	0
9	0	0	0	1	57	29	21	28	0	0	0	0



Tabel 4.2 Nilai *Green* pada Citra Sampel

i,j	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	126	128	110	147	140	1	0	0	0
2	0	0	1	109	123	109	123	75	162	151	0	0
3	0	0	99	99	104	98	91	92	98	91	0	0
4	0	0	91	93	94	81	100	122	92	95	0	0
5	0	0	75	100	81	87	93	111	101	83	0	0
6	0	0	81	88	76	82	95	76	79	78	2	0
7	0	0	83	81	84	73	80	69	61	84	0	0
8	0	0	2	59	110	98	89	132	71	95	0	0
9	0	0	0	0	100	98	99	98	0	0	0	0

Tabel 4.3 Nilai *Blue* pada Citra Sampel

i,j	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	183	186	166	204	192	1	0	0	0
2	0	0	5	167	178	162	178	127	213	209	0	0
3	0	0	158	151	156	150	146	148	155	150	0	0
4	0	1	150	149	145	134	156	174	147	150	0	0
5	0	0	127	153	131	139	146	163	152	136	9	0
6	0	2	132	141	127	133	147	120	122	130	1	0
7	0	0	134	130	134	124	130	121	105	134	0	0
8	0	0	0	109	159	149	140	181	117	146	0	0
9	0	0	0	0	130	150	157	150	0	0	0	0

Berdasarkan nilai pada setiap *channel* RGB, dilakukan ekstraksi fitur pada masing-masing warna menggunakan *Color Moment*. Berikut ini adalah contoh perhitungan untuk memperoleh fitur RGB *color channel* pada citra sampel. Rumus untuk mendapatkan nilai *mean*, *standard deviation*, dan *skewness* dapat dilihat secara berturut-turut pada Persamaan 2.1, Persamaan 2.2, dan Persamaan 2.3.

1. *Mean*

$$\begin{aligned} \mu_{red} &= \frac{1}{12 \times 9} (0 + 0 + 0 + 47 + 53 + \dots + 0) \\ &= \frac{1}{12 \times 9} (2463) \\ &= 22,805 \end{aligned}$$

$$\begin{aligned} \mu_{green} &= \frac{1}{12 \times 9} (0 + 0 + 0 + 126 + 128 + \dots + 0) \\ &= \frac{1}{12 \times 9} (6099) \end{aligned}$$

$$= 56,472$$

$$\mu_{blue} = \frac{1}{12 \times 9} (0 + 0 + 0 + 183 + 186 + \dots + 0)$$

$$= \frac{1}{12 \times 9} (9402)$$

$$= 87,055$$

Keterangan:

μ_{red} = mean red

μ_{green} = mean green

μ_{blue} = mean blue

2. Standard Deviation

$$\sigma_{red} = \sqrt{\frac{1}{12 \times 9} ((0 - 22,805)^2 + (0 - 22,805)^2 + (0 - 22,805)^2 + \dots + (0 - 22,805)^2)}$$

$$= \sqrt{\frac{1}{12 \times 9} ((520,093) + (520,093) + (520,093) + \dots + (520,093))}$$

$$= \sqrt{\frac{1}{12 \times 9} (51710,916)}$$

$$= 21,881$$

$$\sigma_{green} = \sqrt{\frac{1}{12 \times 9} ((0 - 56,472)^2 + (0 - 56,472)^2 + (0 - 56,472)^2 + \dots + (0 - 56,472)^2)}$$

$$= \sqrt{\frac{1}{12 \times 9} ((3189,112) + (3189,112) + (3189,112) + \dots + (3189,112))}$$

$$= \sqrt{\frac{1}{12 \times 9} (272320,916)}$$

$$= 50,214$$

$$\sigma_{blue} = \sqrt{\frac{1}{12 \times 9} ((0 - 87,0556)^2 + (0 - 87,0556)^2 + (0 - 87,0556)^2 + \dots + (0 - 87,0556)^2)}$$

$$= \sqrt{\frac{1}{12 \times 9} ((7578,669) + (7578,669) + (7578,669) + \dots + (7578,669))}$$

$$= \sqrt{\frac{1}{12 \times 9} (611575,667)}$$

$$= 75,251$$



Keterangan:

σ_{red} = standar deviasi *red*

σ_{green} = standar deviasi *green*

σ_{blue} = standar deviasi *blue*

3. Skewness

$$\theta_{red} = \frac{\frac{1}{12 \times 9} (|0 - 22,805|^3 + |0 - 22,805|^3 + |0 - 22,805|^3 + \dots + |0 - 22,805|^3)}{(21,881)^3}$$

$$= \frac{\frac{1}{12 \times 9} ((11861,018) + (11861,018) + (11861,018) + \dots + (11861,018))}{(21,881)^3}$$

$$= \frac{13482,019}{10477,018}$$

$$= 1,286818$$

$$\theta_{green} = \frac{\frac{1}{12 \times 9} (|0 - 56,472|^3 + |0 - 56,472|^3 + |0 - 56,472|^3 + \dots + |0 - 56,472|^3)}{(50,214)^3}$$

$$= \frac{\frac{1}{12 \times 9} ((180096,23) + (180096,23) + (180096,23) + \dots + (180096,23))}{(50,214)^3}$$

$$= \frac{148542,399}{126615,206}$$

$$= 1,173179$$

$$\theta_{blue} = \frac{\frac{1}{12 \times 9} (|0 - 87,0556|^3 + |0 - 87,0556|^3 + |0 - 87,0556|^3 + \dots + |0 - 87,0556|^3)}{(75,251)^3}$$

$$= \frac{\frac{1}{12 \times 9} ((11861,018) + (11861,018) + (11861,018) + \dots + (11861,018))}{(75,251)^3}$$

$$= \frac{470610,31}{426127,599}$$

$$= 1,104388$$

Keterangan:

θ_{red} = skewness *red*



θ_{green} = skewness green

θ_{blue} = skewness blue

Berdasarkan perhitungan yang telah dilakukan, diperoleh nilai pada ekstraksi fitur RGB color channel yang ditunjukkan pada Tabel 4.4.

Tabel 4.4 Nilai Fitur RGB Color Channel

No.	Nama Channel	Fitur	Nilai
1	Red	Mean	22,805
		Standard Deviation	21,881
		Skewness	1,286818
2	Green	Mean	56,472
		Standard Deviation	50,214
		Skewness	1,713179
3	Blue	Mean	87,055
		Standard Deviation	75,251
		Skewness	1,104388

4.6.2 Manualisasi Simple Morphological Shape Descriptors

Fitur yang digunakan untuk fitur bentuk terdapat sebanyak 10 buah fitur. Untuk mendapatkan nilai dari setiap fitur, dilakukan perhitungan pada citra biner dengan sampel berukuran 12x9 piksel . Nilai dari citra sampel ditunjukkan pada Tabel 4.5. Nilai 0 merepresentasikan warna hitam dan nilai 1 merepresentasikan warna putih.

Berdasarkan nilai pada citra sampel tersebut, maka dilakukan perhitungan untuk mendapatkan fitur panjang, lebar, area, perimeter, *aspect ratio*, *roundness*, *compactness*, *rectangulary*, rasio perimeter *L*, dan rasio perimeter *L* dan *W*.

Tabel 4.5 Nilai Citra Sampel pada Fitur Bentuk

i,j	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	1	1	0	0	0
4	0	0	1	1	1	1	1	1	1	1	0	0
5	0	0	1	1	1	1	1	1	1	1	0	0
6	0	0	0	1	1	1	1	1	1	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0

1. Panjang (L)

Nilai pada fitur L didapatkan dengan mencari selisih pada posisi piksel teratas dengan piksel terbawah yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 Citra Sampel pada Fitur Panjang

i,j	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	1	1	0	0	0
4	0	0	1	1	1	1	1	1	1	1	0	0
5	0	0	1	1	1	1	1	1	1	1	0	0
6	0	0	0	1	1	1	1	1	1	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0

Berdasarkan citra sampel maka didapatkan posisi pada piksel teratas pada sumbu vertikal adalah 2 dan yang terbawah adalah 5. Kemudian dihitung selisih dari kedua posisi tersebut.

$$\begin{aligned}
 L &= \text{posisi piksel terbawah} - \text{posisi piksel teratas} \\
 &= 5 - 2 \\
 &= 3
 \end{aligned}$$

2. Lebar (W)

Untuk memperoleh nilai pada fitur W , dilakukan dengan mencari selisih pada posisi piksel yang terletak di paling kiri dan yang terletak di paling kanan. Pada Tabel 4.7 dapat dilihat pada sumbu horizontal posisi piksel paling kiri adalah 2 dan yang paling kanan adalah 9.

Tabel 4.7 Citra Sampel pada Fitur Lebar

i,j	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	1	1	0	0	0
4	0	0	1	1	1	1	1	1	1	1	0	0
5	0	0	1	1	1	1	1	1	1	1	0	0
6	0	0	0	1	1	1	1	1	1	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0



$$\begin{aligned}
 W &= \text{posisi piksel paling kanan} - \text{posisi piksel paling kiri} \\
 &= 9 - 2 \\
 &= 7
 \end{aligned}$$

3. Area (*A*)

Nilai fitur *A* didapatkan dengan menjumlahkan seluruh nilai pada citra biner yang merupakan luas dari objek.

Tabel 4.8 Citra Sampel pada Fitur Area

<i>i,j</i>	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	1	1	0	0	0
4	0	0	1	1	1	1	1	1	1	1	0	0
5	0	0	1	1	1	1	1	1	1	1	0	0
6	0	0	0	1	1	1	1	1	1	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0

Berdasarkan citra sampel pada Tabel 4.8 maka didapatkan nilai dari fitur area adalah 26.

4. Perimeter (*P*)

Tabel 4.9 Citra Sampel pada Fitur Perimeter

<i>i,j</i>	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	1	1	1	1	1	1	0	0
4	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	1	0	0	0	0	0	0	1	0	0
6	0	0	1	1	1	1	1	1	1	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0

Fitur *P* merupakan keliling dari objek yang nilainya didapatkan dengan menjumlahkan garis tepi objek seperti yang terlihat pada Tabel 4.9. Berdasarkan tabel tersebut didapatkan jumlah dari seluruh piksel adalah 19. Maka nilai dari fitur perimeter objek adalah 19.

5. Aspect Ratio (AR)

Untuk mendapatkan nilai *AR* dibutuhkan nilai *input* berupa fitur *L* dan *W*. Lalu digunakan Persamaan 2.4 untuk mendapatkan nilai fitur *AR* seperti berikut.

$$AR = \frac{3}{7} = 0,4285$$

6. Roundness (R)

Nilai fitur *R* diperoleh dengan rumus pada Persamaan 2.5 dengan menggunakan nilai dari fitur *A* dan *P* sebagai nilai *input*. Perhitungan untuk mendapatkan nilai dari fitur ini adalah sebagai berikut.

$$R = \frac{4 \times 3,14285714 \times 26}{19^2} = 0,90542145$$

7. Compactness (PA)

Nilai dari fitur *PA* didapatkan dengan menggunakan fitur *P* dan *A* sebagai data input dan digunakan rumus pada Persamaan 2.6 untuk mendapatkan nilai dari Fitur *PA* dengan perhitungan seperti berikut.

$$PA = \frac{19}{\sqrt{26}} = 3,726207$$

8. Rectangulary (N)

Untuk mendapatkan nilai dari Fitur *C* diperlukan nilai fitur *A*, *L*, dan *W*. Rumus yang digunakan ialah pada Persamaan 2.7 dengan perhitungan sebagai berikut.

$$N = \frac{26}{3 \times 7} = 1,23809524$$

9. Rasio Perimeter L

Nilai rasio perimeter *L* diperoleh dengan menggunakan nilai fitur *P* dan *L* sebagai variabel input. Kemudian dilakukan perhitungan menggunakan persamaan 2.8 untuk memperoleh nilai dari fitur rasio perimeter *L* seperti berikut ini.

$$PL = \frac{19}{3} = 6,333333$$

10. Rasio Perimeter L dan W

Nilai dari fitur rasio perimeter *L* dan *W* didapatkan dengan memakai nilai fitur *P*, *L*, dan *W*. Nilai tersebut dihitung menggunakan Persamaan 2.9 dengan proses sebagai berikut.

$$PLW = \frac{19}{3+7} = 1,9$$

Berdasarkan perhitungan yang telah dilakukan untuk mendapatkan nilai dari semua fitur bentuk dan disajikan pada Tabel 4.10.

Tabel 4.10 Nilai Fitur *Simple Morphological Shape Descriptors*

No.	Nama Fitur	Nilai
1	Panjang	3
2	Lebar	7
3	Area	26
4	Perimeter	19
5	<i>Aspect Ratio</i>	0,4285
6	<i>Roundness</i>	0,90542145
7	<i>Compactness</i>	3,726207
8	<i>Rectangulary</i>	1,23809524
9	Rasio Perimeter L	6,3333
10	Rasio Perimeter L dan W	1,9

4.6.3 Manualisasi Perhitungan Jarak *Euclidean*

Manualisasi pada perhitungan jarak *Euclidean* menggunakan 10 data sebagai data latih dan 2 data sebagai data uji. Data yang digunakan berupa citra makanan yang sudah dilakukan proses segmentasi. Kelas pada data terdapat tiga kelas, yaitu donat, mie gepeng, dan *fried chicken*. Fitur yang digunakan untuk proses klasifikasi yaitu sejumlah 9 fitur warna dan 10 fitur bentuk. Fitur warna terdiri dari *mean*, *standard deviation*, dan *skewness* pada masing-masing channel RGB. Data pada fitur warna dapat dilihat pada Tabel 4.11.

Tabel 4.11 Data Manualisasi Fitur Warna

No	Kode Gambar	Kelas	Fitur Warna								
			<i>Mean</i>			<i>Standard Deviation</i>			<i>Skewness</i>		
			Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
1	001_0001	Donat	27,9744	53,4513	73,2941	35,5499	64,2417	87,7048	1,3205	1,1632	1,1175
2	001_0006	Donat	31,3629	72,2026	107,1239	29,2309	58,8401	84,2016	1,4450	1,2121	1,1772
3	001_0002	Donat	31,5169	63,4701	89,1945	39,4916	75,2492	103,1564	1,3693	1,1722	1,1130
4	005_0048	Mie Gepeng	19,4210	49,0250	72,8877	27,0463	54,6067	75,7800	2,0589	1,3142	1,1764
5	005_0047	Mie Gepeng	19,3811	48,9001	72,6749	26,9864	54,5834	75,7758	2,0466	1,3132	1,176
6	005_0049	Mie Gepeng	24,0431	58,8593	85,5333	30,4399	61,5822	84,7445	1,8996	1,2313	1,1251
7	008_0079	Fried Chicken	21,3589	40,4916	63,0017	31,7385	55,3289	79,4864	1,7198	1,5010	1,3167
8	008_0080	Fried Chicken	23,1493	43,6538	68,036	33,5664	58,3714	84,217	1,6650	1,4574	1,2846
9	008_0077	Fried Chicken	28,5416	45,5149	63,4098	43,7372	66,9753	88,4086	1,6617	1,5140	1,3501
10	008_0078	Fried Chicken	18,9524	36,9218	58,0432	29,0889	51,2530	73,7909	1,8494	1,5629	1,3563
11	001_0009	Donat	32,8208	72,3611	105,4792	31,8869	63,9125	90,2559	1,4811	1,2068	1,1321
Min			18,9524	36,9218	58,0432	26,9864	51,2530	73,7909	1,3205	1,1632	1,113
Max			32,8208	72,3611	107,1239	43,7372	75,2492	103,1564	2,0589	1,5629	1,3563

Kemudian fitur bentuk terdiri dari panjang, lebar, area, perimeter, *aspect ratio*, *roundness*, *compactness*, *rectangulary*, rasio perimeter *L*, dan rasio perimeter *L* dan *W*. Data pada fitur bentuk dapat dilihat pada Tabel 4.12.

Tabel 4.12 Data Manualisasi Fitur Bentuk

No	Kode Gambar	Kelas	Fitur Bentuk									
			L	W	P	A	AR	R	PA	C	PL	PLW
1	001_0001	Donat	8	6	25	41	1,333333	0,824359	3,904344	0,854166	3,125	1,785714286
2	001_0006	Donat	8	9	18	75	0,888888	2,908882	2,07846	1,041667	2,25	1,058823529
3	001_0002	Donat	8	6	20	51	1,333333	1,602212	2,80056	1,0625	2,5	1,428571429
4	005_0048	Mie Gepeng	8	7	22	55	1,142857	1,427996	2,966479	0,982142	2,75	1,466666667
5	005_0047	Mie Gepeng	8	7	22	51	1,142857	1,324142	3,080616	0,910714	2,75	1,466666667
6	005_0049	Mie Gepeng	8	10	20	63	0,8	1,979203	2,519763	0,7875	2,5	1,111111111
7	008_0079	Fried Chicken	4	6	19	29	0,666666	1,009486	3,528211	1,208333	4,75	1,9
8	008_0080	Fried Chicken	8	7	18	61	1,142857	2,36589	2,304663	1,089285	2,25	1,2
9	008_0077	Fried Chicken	3	7	19	26	0,428571	0,905057	3,726206	1,238095	6,33	1,9
10	008_0078	Fried Chicken	4	6	19	30	0,666666	1,044296	3,468909	1,25	4,75	1,9
11	001_0009	Donat	8	10	20	72	0,8	2,261946	2,357022	0,9	2,5	1,111111111
Min			3	6	18	26	0,428571	0,824359	2,07846	0,7875	2,25	1,058823529
Max			8	10	25	75	1,333333	2,908882	3,904344	1,25	6,33	1,9

Setelah semua data didapatkan fiturnya, selanjutnya dilakukan normalisasi menggunakan Persamaan 2.12. Sebagai contoh perhitungan, berikut ini proses normalisasi pada kode gambar 001_0001 fitur *mean* pada *channel R* (red).

$$\begin{aligned}
 \text{Normalisasi} &= \frac{27,9744 - 18,9524}{32,8208 - 18,9524} \\
 &= \frac{9,02196}{13,8683} \\
 &= 0,6505
 \end{aligned}$$

Hasil normalisasi pada fitur warna keseluruhan data dapat dilihat pada Tabel 4.13

Tabel 4.13 Hasil Normalisasi pada Fitur Warna

No	Kode Gambar	Kelas	Fitur Warna								
			Mean			Standard Deviation			Skewness		
			Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
1	001_0001	Donat	0,6505	0,4664	0,3107	0,5112	0,5413	0,4738	0	0	0,1182
2	001_0006	Donat	0,8949	0,9955	1	0,134	0,3162	0,3545	0,1687	0,1223	0,3387
3	001_0002	Donat	0,906	0,7491	0,6347	0,7465	1	1	0,0661	0,0225	0,1015
4	005_0048	Mie Gepeng	0,0338	0,3415	0,3024	0,0036	0,1398	0,0677	1	0,3777	0,3357
5	005_0047	Mie Gepeng	0,0309	0,338	0,2981	0	0,1388	0,0676	0,9834	0,3753	0,3344
6	005_0049	Mie Gepeng	0,3671	0,619	0,5601	0,2062	0,4305	0,373	0,7844	0,1702	0,1462

No	Kode Gambar	Kelas	Fitur Warna								
			Mean			Standard Deviation			Skewness		
			Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
7	008_0079	Fried Chicken	0,1735	0,1007	0,101	0,2837	0,1699	0,194	0,5407	0,845	0,8537
8	008_0080	Fried Chicken	0,3026	0,19	0,2036	0,3928	0,2966	0,355	0,4666	0,7359	0,7354
9	008_0077	Fried Chicken	0,6914	0,2425	0,1093	1	0,6552	0,4978	0,4621	0,8775	0,9771
10	008_0078	Fried Chicken	0	0	0	0,1255	0	0	0,7163	1	1
11	001_0009	Donat	1	1	0,9665	0,2926	0,5276	0,5607	0,2175	0,1089	0,1719

Hasil normalisasi untuk fitur bentuk dapat dilihat pada Tabel 4.14.

Tabel 4.14 Hasil Normalisasi pada Fitur Bentuk

No	Kode Gambar	Kelas	Fitur Bentuk									
			L	W	P	A	AR	R	PA	C	PL	PLW
1	001_0001	Donat	1	0	1	0,3061	1	0	1	0,1441	0,21446	0,8641
2	001_0006	Donat	1	0,75	0	1	0,5087	1	0	0,5495	0	0
3	001_0002	Donat	1	0	0,2857	0,5102	1	0,3731	0,3954	0,5945	0,06127	0,4395
4	005_0048	Mie Gepeng	1	0,25	0,5714	0,5918	0,7894	0,2895	0,4863	0,4208	0,12255	0,4848
5	005_0047	Mie Gepeng	1	0,25	0,5714	0,5102	0,7894	0,2397	0,5488	0,2664	0,12255	0,4848
6	005_0049	Mie Gepeng	1	1	0,2857	0,7551	0,4105	0,5540	0,2416	0	0,06127	0,0621
7	008_0079	Fried Chicken	0,2	0	0,1428	0,0612	0,2631	0,0888	0,794	0,9099	0,61275	1
8	008_0080	Fried Chicken	1	0,25	0	0,7142	0,7894	0,7395	0,1238	0,6525	0	0,1678
9	008_0077	Fried Chicken	0	0,25	0,1428	0	0	0,0387	0,9024	0,9742	1	1
10	008_0078	Fried Chicken	0,2	0	0,1428	0,0816	0,2631	0,1055	0,7615	1	0,61275	1
11	001_0009	Donat	1	1	0,2857	0,9387	0,4105	0,6896	0,1525	0,2432	0,06127	0,0621

Data latih untuk manualisasi adalah 10 data yaitu data pada nomor 1 hingga 10. Kemudian data uji yang digunakan adalah data dengan nomor 11. Perhitungan dilakukan dengan menghitung jarak pada data uji dengan seluruh data latih. Contoh perhitungan untuk menghitung jarak dengan *euclidean* menggunakan data uji nomor 11 (kode gambar 001_0009) dengan data latih nomor 1 (kode gambar 001_0001) pada setiap fitur warna dan bentuk. Jarak dihitung dengan memakai Persamaan 4.14

$$\begin{aligned}
 d_{euclidean} &= \sqrt{(1 - 0,6505)^2 + (1 - 0,4664)^2 + \dots + (0,6127 - 0,21446) + 0 - 0,2321)} \\
 &= \sqrt{4,821534998}
 \end{aligned}$$

$$= 2,19579939$$

Hasil dari perhitungan jarak pada semua fitur menggunakan *euclidean* untuk data uji nomor 11 dengan seluruh data latih disajikan pada Tabel 4.15.

Tabel 4.15 Hasil Perhitungan *Euclidean* Data Uji Nomor 11

No	Kode Gambar	Kelas	Jarak <i>Euclidean</i>
1	001_0001	Donat	2,348270778
2	001_0006	Donat	0,83528749
3	001_0002	Donat	1,549879988
4	005_0048	Mie Gepeng	2,218367291
5	005_0047	Mie Gepeng	2,243188258
6	005_0049	Mie Gepeng	1,264978687
7	008_0079	Fried Chicken	2,985045867
8	008_0080	Fried Chicken	1,990200241
9	008_0077	Fried Chicken	3,082254361
10	008_0078	Fried Chicken	3,245796463

Setelah jarak data uji terhadap seluruh data latih diperoleh, jarak diurutkan dari nilai yang terkecil hingga terbesar. Seperti yang ditunjukkan pada Tabel 4.16 untuk data uji nomor 11 yang mana kelas asli dari data uji tersebut adalah donat. Kemudian data diambil sebanyak nilai n . Nilai n ditentukan secara acak. Untuk proses manualisasi nilai n ditentukan ialah $n = 3$, sehingga didapatkan 3 jarak terkecil seperti yang ditunjukkan pada Tabel 4.17.

Tabel 4.16 Jarak Terkecil Berdasarkan Nilai n

No	Kode Gambar	Kelas	Hasil <i>Euclidean</i>
1	001_0006	Donat	0,83528749
2	005_0049	Mie Goreng Gepeng	1,264978687
3	001_0002	Donat	1,549879988

Pada jarak yang terkecil sejumlah n , kemudian dipilih kelas yang paling banyak sebagai kelas data yang akan dievaluasi. Berdasarkan Tabel 4.17, didapatkan kelas donat yang muncul sebanyak 2 dan kelas mie gepeng sebanyak 1. Sehingga hasil klasifikasi untuk data uji nomor 11 merupakan kelas donat.

Berdasarkan hasil klasifikasi, resep makanan sebagai hasil pencarian disesuaikan dengan kelas yang didapatkan. Apabila data uji no 11 dijadikan sebagai *query input* untuk pencarian resep makanan dan mendapatkan hasil klasifikasi adalah "donat", maka hasil pencarian resep yang didapatkan berupa "resep membuat donat".

4.6.4 Manualisasi Pengujian

Pengujian dilakukan dengan melakukan perhitungan pada *mean average precision* (MAP). Untuk melakukan menghitung nilai MAP maka perlu dilakukan pencarian nilai *precision* bertingkat pada setiap data uji dengan membandingkan kelas hasil klasifikasi dan kelas yang sebenarnya. Untuk manualisasi digunakan nilai $n = 3$. Nilai n berfungsi untuk menunjukkan jumlah maksimal pencarian yang akan ditampilkan. Perhitungan rata-rata untuk data uji nomor 11 disajikan pada Tabel 4.18 yang mana kelas sebenarnya ialah “donat” dan citra sebagai *query* ditunjukkan pada Gambar.

Tabel 4.17 Menghitung Precision pada Data Uji Nomor 11

Peringkat	Kelas	Data Uji Nomor 11	
		Relevansi	Precision
1	Donat	Ya	1
2	Mi Gepeng	Tidak	0
3	Donat	Ya	0,6667

Perhitungan pada nilai dari *precision* dan rata-rata pada data uji nomor 11 dilakukan seperti berikut.

$$\text{Kelas peringkat 1} = \frac{1}{1} = 1$$

$$\text{Kelas peringkat 3} = \frac{2}{3} = 0,6667$$

$$\text{Rata-rata } P \text{ data uji no. 11} = \frac{1 + 0,6667}{2} = 0,5556$$

Maka dapat diperoleh nilai dari MAP melalui perhitungan seperti berikut.

$$\text{MAP} = \frac{0,83335}{1} = 0,83335$$

Hasil dari pengujian yang didapatkan untuk mendapatkan resep makanan berdasarkan fitur warna dan fitur bentuk menggunakan MAP adalah rata-rata *precision* kemudian dibagikan dengan jumlah *query*. Hasil yang diperoleh ialah sebesar 0,83335.

BAB 5 IMPLEMENTASI

Bab implementasi berisi tentang penjelasan *source code* berdasarkan perancangan algoritme yang telah dilakukan pada Bab 4. Setiap *source code* akan diberikan penjelasan pada setiap fungsinya. Terdapat dua lingkungan implementasi yang digunakan pada penelitian ini yaitu lingkungan perangkat keras dan lingkungan perangkat lunak. Implementasi yang dijelaskan antara lain proses *pre-processing*, ekstraksi fitur RGB *color channel* menggunakan *color moment*, ekstraksi bentuk menggunakan *simple morphological shape descriptors*, dan perhitungan jarak menggunakan *Euclidean*.

5.1 Lingkungan Implementasi

5.1.1 Lingkungan Perangkat Keras

Perangkat keras pendukung yang digunakan dalam proses implementasi algoritme, antara lain sebagai berikut.

1. *Processor: Intel® Core™ i3-4005U CPU 1,7 GHz*
2. *RAM: 6,00 GB*
3. *VGA: Nvidia GT930M*
4. *Harddisk: 500 GB*
5. *Ukuran Layar: 14 inch*
6. *Mouse*

5.1.2 Lingkungan Perangkat Lunak

Lingkungan perangkat lunak yang dipakai untuk melakukan implementasi dijelaskan sebagai berikut.

1. *Sistem operasi Microsoft Windows 8.1 Pro*
2. *Python 3.7*

5.2 Implementasi Algoritme

5.2.1 Implementasi *Pre-processing* Fitur Warna

Source code berikut berisi tentang proses *pre-processing* pada fitur warna.

No	File: cMoment.py
1	<code>import cv2</code>
2	<code>import numpy as np</code>
3	<code>class ColorMoment:</code>
4	<code> def prepWarna(img):</code>
5	<code> gaussian = cv2.GaussianBlur(img, (5,5), 0)</code>
6	<code> hsv = cv2.cvtColor(gaussian, cv2.COLOR_BGR2HSV)</code>
7	<code> gray = cv2.cvtColor(hsv, cv2.COLOR_BGR2GRAY)</code>
8	<code> ret, threshold = cv2.threshold(gray, 110, 255, cv2.THRESH_BINARY)</code>

9	hasilthreshold = np.repeat(threshold[:, :, np.newaxis], 3, axis=2)
10	bitwise = cv2.bitwise_and(img, hasilthreshold)
11	return bitwise

Penjelasan:

Baris 1-2 adalah proses untuk meng-*import library* dari *Open Source Computer Vision Library* (OpenCV) sebagai pustaka perangkat lunak pada pengolahan citra dan *package Numpy* untuk komputasi ilmiah pada pemrograman *Python*.

Baris 3 adalah deklarasi kelas *ColorMoment*.

Baris 4 adalah sebuah fungsi dengan nama *prepWarna* untuk melakukan *pre-processing* pada fitur warna. Paramter yang digunakan ialah *img* yaitu citra makanan. Fungsi ini akan dipanggil jika diperlukan hasil dari *pre-processing* fitur warna.

Baris 5 adalah proses mengubah gambar asli menjadi lebih kabur atau disebut dengan *blur*. Proses ini dilakukan menggunakan *library* OpenCV dengan rata-rata *filter kernel* 5x5. Nama variabel untuk menyimpan hasil konversi ialah *gaussian*.

Baris 6 adalah proses konversi memakai *library* OpenCV dari citra pada variabel *gaussian* menjadi citra dengan *channel* HSV. Hasil dari konversi disimpan pada variabel bernama *hsv*.

Baris 7 adalah proses konversi citra dengan variabel *hsv* menjadi citra keabuan. Konversi dilakukan menggunakan *library OpenCV*. Hasil dari konversi ini disimpan dalam variabel bernama *gray*.

Baris 8 adalah proses mengubah citra *grayscale* menjadi citra biner. Piksel dengan nilai 0-110 akan diubah menjadi warna hitam (0) dan piksel dengan nilai 111-255 diubah menjadi warna putih (1). Hasil dari konversi ini disimpan pada variabel bernama *hasilthreshold*.

Baris 9 adalah proses untuk menggabungkan hasil dari *pre-processing* yaitu citra yang tersimpan pada variabel *hasilthreshold* dengan citra asli. Hasil dari proses ini disimpan didalam variabel bernama *bitwise*.

Baris 10 adalah proses untuk mengembalikan nilai variabel *bitwise*.

5.2.2 Impementasi Perhitungan Fitur Warna

Source code berikut berisi mengenai proses perhitungan untuk mendapatkan fitur warna *mean*, *standard deviation*, dan *skewness*.

No	File: cMoment.py
1	def mean (img):
2	M,N = img.shape
3	size=M*N
4	jml_img = 0
5	for i in range(M):
6	for j in range(N):

```

7         jml_img = jml_img + img[i][j];
8
9     mean_img=jml_img/size
10    return mean_img
11
12 def standarDeviasi (img,mean):
13     M,N = img.shape
14     size=M*N
15     jml_img = 0
16
17     for i in range(M):
18         for j in range(N):
19             jml_img = jml_img + (img[i][j]-mean)**2
20
21     std_img=np.sqrt((jml_img)/size)
22     return std_img
23
24 def skewness (img,mean,standarDeviasi):
25     M,N = img.shape
26     size=M*N
27     jml_img = 0
28
29     for i in range(M):
30         for j in range(N):
31             jml_img = jml_img + abs((img[i][j]-
32 mean)**3)
33
34     skw_img=((jml_img)/size)/((standarDeviasi**3)
35     return skw_img

```

Penjelasan:

Baris 1 adalah fungsi bernama *mean* untuk perhitungan *mean* dengan parameter *img* yaitu citra makanan yang telah melalui proses *pre-processing*. Parameter *img* dapat mengambil citra dengan setiap *channel* RGB.

Baris 2-3 adalah proses untuk mengambil nilai panjang dan lebar piksel dari gambar masing-masing pada variabel *M* dan *N*. Kemudian pada variabel *size* disimpan nilai *M x N* yang merupakan jumlah piksel gambar.

Baris 4 adalah inisialisasi variabel *jml_img* sama dengan 0. Proses ini bertujuan untuk memberikan nilai awal pada variabel tersebut.

Baris 5-7 adalah proses untuk menjumlahkan seluruh piksel. Perulangan dengan variabel *i* sejumlah *M* dan perulangan variabel *j* sejumlah *N*. Pada proses ini variabel *jml_img* akan secara berulang dijumlahkan dengan *jml_img* dengan indeks *i* dan *j* secara berurutan sampai akhir dari nilai variabel *i* dan *j*. Berdasarkan proses perulangan tersebut akan diperoleh jumlah seluruh nilai piksel yang disimpan pada variabel *jml_img*.

Baris 8-9 adalah proses memperoleh fitur *mean* yaitu dengan membagikan nilai variabel *jml_img* dengan variabel *size*. Setelah itu nilai fitur *mean* disimpan dan dikembalikan di variabel *mean_img*.

Baris 10 adalah fungsi bernama *standarDeviasi* untuk melakukan perhitungan fitur *standard deviation* dengan parameter *img* dan *mean*.

Baris 11-12 adalah proses untuk mengambil nilai panjang dan lebar citra masing-masing pada variabel *M* dan *N*. Kemudian ukuran dari citra disimpan pada variabel *size* dengan mengalikan nilai pada *M* dan *N*.

Baris 13 merupakan inisialisasi variabel *jml_img* dengan memberikan nilai awal sama dengan 0.

Baris 14-16 dilakukan perulangan indeks *i* sebanyak nilai *M* dan perulangan indeks *j* sebanyak nilai *N*. Perulangan dilakukan untuk mendapatkan nilai dari setiap piksel dikurang *mean* kemudian dipangkatkan dengan dua. Nilai tersebut disimpan dan dijumlahkan sebanyak indeks *i* dan *j* pada variabel *jml_img*.

Baris 17-18 adalah proses memperoleh fitur *standard deviation* yaitu dengan membagikan nilai variabel *jml_img* dengan variabel *size* dan diakarkan dengan menggunakan *numpy*. Kemudian nilai tersebut disimpan dan dikembalikan pada variabel *std_img*.

Baris 19 adalah fungsi dengan nama *skewness* untuk memperoleh fitur *skewness* dengan parameter *img*, *mean*, dan *standarDeviasi*.

Baris 20-21 adalah proses untuk mendapatkan ukuran dari citra yang disimpan pada variabel *M* sebagai panjang dan *N* sebagai lebar. Kemudian pada variabel *seze* disimpan nilai dari *M* dikalikan dengan *N* sebagai ukuran dari citra.

Baris 22-25 adalah proses perulangan indeks *i* sepanjang *M* dan *j* sepanjang *N* untuk mendapatkan nilai mutlak setiap piksel dikurangi dengan *mean* kemudian dipangkatkan dengan tiga. Nilai tersebut dijumlahkan sebanyak indeks *i* dan *j* dan disimpan secara berulang pada variabel *jml_img*.

Baris 26-27 adalah proses untuk memperoleh nilai dari fitur *skewness*. Nilai tersebut diperoleh dengan membagikan nilai pada variabel *jml_img* dengan variabel *size* dan kemudian hasilnya dibagikan dengan variabel *std_img* melalui parameter *standarDeviasi* dipangkatkan dengan 3. Nilai dari *skewness* disimpan dan dikembalikan pada variabel *skw_img*.

5.2.3 Impementasi *Pre-processing* Fitur Bentuk

Berikut merupakan *source code* untuk *pre-processing* citra pada fitur bentuk.

No	File: smsd.py
1	<code>import cv2</code>
2	<code>import numpy as np</code>
3	<code>class SMSD:</code>
4	<code> def prepBentuk(img):</code>
5	<code> gaussian = cv2.GaussianBlur(img, (5, 5), 0)</code>
6	<code> hsv = cv2.cvtColor(gaussian, cv2.COLOR_BGR2HSV)</code>
7	<code> gray = cv2.cvtColor(hsv, cv2.COLOR_BGR2GRAY)</code>



8	<code>ret, threshold =</code>
9	<code>cv2.threshold(gray, 110, 255, cv2.THRESH_BINARY)</code> <code>return threshold</code>

Penjelasan:

Baris 1-2 adalah proses untuk meng-*import library* OpenCV dan *package* Numpy.

Baris 3 adalah proses pendklarasian kelas SMSD.

Baris 4 adalah pembuatan fungsi dengan nama *prepBentuk* untuk melakukan proses *pre-processing* pada citra. Parameter dari fungsi ini berupa *img*.

Baris 5 adalah proses konversi citra asli menjadi citra dengan efek *blur*. Proses ini dilakukan menggunakan *library OpenCV* dengan *filter kernel 5x5*. Citra yang telah dikonversi disimpan dengan nama variabel *gaussian*.

Baris 6 adalah proses konversi citra pada variabel *gaussian* menjadi citra HSV. Konversi dilakukan dengan memakal *library OpenCV* dan hasil konversi disimpan pada variabel bernama *hsv*.

Baris 7 adalah proses untuk mengubah citra pada variabel *HSV* menjadi citra keabuan dengan menggunakan *library OpenCV*. Hasil konversi disimpan pada variabel dengan nama *gray*.

Bari 8 ialah proses untuk mengubah citra pada variabel *gray* menjadi citra biner yang mana nilai piksel 0-110 akan diubah nilainya menjadi 0 dan piksel dengan nilai 111-255 akan diubah menjadi 1. Nilai tersebut disimpan pada variabel bernama *threshold*.

Baris 9 ialah proses untuk mengembalikan nilai citra hasil *pre-processing* pada variabel *threshold*.

5.2.4 Impementasi Perhitungan Fitur Bentuk

Source code berikut merupakan proses perhitungan untuk mendapatkan fitur bentuk panjang, lebar, area, perimeter, *aspect ratio*, *roundness*, *compactness*, *rectangulary*, rasio perimeter *L*, dan rasio perimeter *L* dan *W*.

No	File: smsd.py
1	<code>def panjang(img):</code>
2	<code> bentuk=cv2.findContours(img,</code>
3	<code>cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)</code> <code> bentuk = bentuk[1]</code>
4	<code> b = max(bentuk, key=cv2.contourArea)</code>
5	<code> atasMax = tuple(b[b[:, :, 1].argmin()][0])</code>
6	<code> bawahMax = tuple(b[b[:, :, 1].argmax()][0])</code>
7	<code> L = abs(atasMax[1]-bawahMax[1])</code>
8	<code> return L</code>
9	<code>def lebar(img): bentuk = cv2.findContours(img,</code>
10	<code>cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)</code> <code> bentuk = bentuk[1]</code>



```

11     b = max(bentuk, key=cv2.contourArea)
12     kiriMax = tuple(b[b[:, :, 0].argmin()][0])
13     kananMax = tuple(b[b[:, :, 0].argmax()][0])
14     W = abs(kiriMax[0]-kananMax[0])
15     return W

16 def perimeter(img):
17     P=0
18     M,N = img.shape
19     edge=cv2.Canny(img,0,1)
20     edge[edge>0]=1

21     for i in range(M):
22         for j in range(N):
23             P+=edge[i][j]
24     return P

25 def area(img):
26     A=0
27     M,N = img.shape
28     for i in range(M):
29         for j in range(N):
30             if img[i][j]>0:
31                 img[i][j]=1
32                 A += img[i][j]
33     return A

34 def aspectRatio(img,L,W):
35     AR = L/W
36     return AR

37 def roundness(img,A,P):
38     R = (4*np.pi*A)/P**2
39     return R

40 def rectangulary(img,A,L,W):
41     N = A/(L*W)
42     return N

43 def compactness(img,A,P):
44     PA = P/np.sqrt(A)
45     return PA

46 def rasioPL(img,P,L):
47     PL = P/L
48     return PL

49 def rasioPLW(img,P,L,W):
50     PLW = P/(L+W)
51     return PLW

```

Penjelasan:

Baris 1-8 adalah fungsi *panjang* untuk mendapatkan nilai dari fitur panjang. Pada variabel *bentuk* terdapat fungsi *library OpenCV* yaitu *cv2.findContours* dengan tiga parameter. Parameter yang pertama adalah citra sumber, kedua adalah mode

pengambilan kontur, dan yang ketiga adalah metode perkiraan kontur. Keluaran dari fungsi ini adalah kontur dan hierarki. Kemudian dicari nilai paling luar dari objek. Selanjutnya pada variabel *atasMax* dicari nilai yang berada di titik paling atas pada tepi objek. Dan pada variabel *bawahMax* dicari nilai yang terletak pada titik terendah pada tepi objek. Pada variabel *L* kemudian dicari selisih dari variabel *atasMax* dan *bawahMax*.

Baris 9-15 adalah fungsi *lebar* untuk mendapatkan nilai dari fitur lebar. Sama halnya dengan fitur panjang, pencarian kontur dilakukan pada variabel *bentuk* dan pada variabel *b* mencari nilai tepi objek. Terdapat variabel *kiriMax* yang berguna untuk menemukan titik terluar sebelah kiri dan variabel *kananMax* untuk menemukan titik terluar sebelah kanan. Kemudian dicari selisih dari kedua variabel tersebut dan hasilnya disimpan pada variabel *W* sebagai fitur lebar.

Baris 16-24 adalah fungsi untuk memperoleh fitur perimeter dengan nama fungsi *perimeter* dan parameter *img*. Variabel *edge* digunakan untuk menyimpan nilai piksel pada tepi objek yang didapatkan dari fungsi *cv2.Canny*. Kemudian setiap nilai lebih dari 1 diubah menjadi sama dengan 1. Pada perulangan *i* dan *j* digunakan untuk menyimpan jumlah seluruh piksel. Nilai disimpan dan dikembalikan pada variabel bernama *P*.

Baris 25-33 adalah fungsi untuk mendapatkan area dari objek citra dengan nama fungsi *area* dan parameter *img* yaitu citra yang akan dicari fiturnya. Pada perulangan indeks *i* dan *j* merupakan proses untuk menjumlahkan seluruh nilai piksel yang mana sebelumnya setiap nilai piksel lebih dari 1 akan diubah nilainya menjadi 1. Jumlah seluruh piksel pada citra disimpan dan dikembalikan pada variabel *A*.

Baris 34-36 adalah fungsi untuk memperoleh fitur *aspect ratio*. Parameter yang digunakan pada fungsi ini adalah *img*, *L*, dan *W*. Perhitungan dilakukan dengan membagikan nilai *L* dengan *W* kemudian hasilnya disimpan dan dikembalikan di variabel dengan nama *AR*.

Baris 37-39 adalah fungsi mencari nilai fitur *roundness*. Parameter fungsi ini berupa *img*, *A*, dan *P*. Untuk mendapatkan nilai *roundness* dilakukan perhitungan 4 dikali π dikalikan lagi dengan variabel *A*. Hasil dari perkalian tersebut kemudian dibagikan dengan variabel *P* dipangkat 2. Hasil yang diperoleh kemudian disimpan dan dikembalikan di variabel bernama *R*.

Baris 40-42 adalah fungsi untuk mendapatkan nilai dari fitur *rectangulary*. Parameter yang digunakan antara lain *img*, *A*, *L*, dan *W*. Fitur ini diperoleh dengan membagikan variabel *A* dengan *L* dikalikan dengan *W*. Hasil dari perhitungan kemudian disimpan dan dikembalikan dalam variabel *N*.

Baris 43-45 adalah fungsi menghitung nilai fitur *compactness*. Fitur ini diperoleh dengan membagikan nilai *P* dengan akar dari *A*. Hasil dari perhitungan tersebut disimpan dan dikembalikan nilainya pada variabel bernama *PA*.

Baris 46-48 adalah fungsi memperoleh fitur rasio perimeter L . Perhitungan yang dilakukan yaitu dengan membagikan nilai variabel P dengan L . Kemudian hasil yang didapatkan disimpan dan dikembalikan nilainya pada variabel PL .

Baris 49-51 adalah sebuah fungsi untuk menghitung fitur rasio perimeter LW . Proses perhitungan yang dilakukan yaitu dengan membagikan nilai pada variabel P dengan nilai dari L dikalikan dengan W . Hasil dari perhitungan tersebut disimpan dikembalikan nilainya pada variabel PLW .

5.2.5 Impementasi Pemanggilan Fitur dan Kelas Citra

Source code berikut ini ialah proses untuk mengambil nilai fitur dan kelas setiap citra kemudian menyimpannya dalam *file* dengan ekstensi *.csv*.

No	File: ekstraksiFitur.py
1	import cv2
2	import numpy as np
3	import os
4	from os import walk
5	from BENTUK import smsd
6	from WARNA import cMoment
7	import csv
8	def kelas(filename):
9	file = filename.split('_')
10	if(file[1] == '001'):
11	return 'Donat'
12	elif (file[1] == '002'):
13	return 'Roti Gandum'
14	elif (file[1] == '003'):
15	return 'Roti Tawar'
16	elif (file[1] == '004'):
17	return 'Mi Goreng Keriting'
18	elif (file[1] == '005'):
19	return 'Mi Goreng Gepeng'
20	elif (file[1] == '006'):
21	return 'Telur Mata Sapi'
22	elif (file[1] == '007'):
23	return 'Telur Dadar'
24	elif (file[1] == '008'):
25	return 'Fried Chicken'
26	elif (file[1] == '009'):
27	return 'Rendang'
28	elif (file[1] == '021'):
29	return 'Nasi Kuning'
30	elif (file[1] == '022'):
31	return 'Nasi Merah'
32	elif (file[1] == '023'):
33	return 'Oreo'
34	elif (file[1] == '024'):
35	return 'Beng Beng'
36	elif (file[1] == '025'):
37	return 'Soba Mie'
38	elif (file[1] == '026'):
39	return 'Tim Tam'
40	elif (file[1] == '027'):

```
41     return 'Keripik Kentang'
42     elif (file[1] == '028'):
43         return 'Wafer Coklat'
44     elif (file[1] == '029'):
45         return 'Biskuit'
46     elif (file[1] == '030'):
47         return 'Nugget Coklat'
48     elif (file[1] == '031'):
49         return 'Genji Pie'
50     else:
51         return 'None'

52 path = "D:/KULIAH/Semester 7/Skripsi/Original/"
53 file = os.listdir(path)

54 with open('D:/KULIAH/Semester 7/Skripsi/kodingan/in the name
of Jesus/x.csv', 'a') as myfile:
55     wrt = csv.writer(myfile, delimiter=',')
56     for filename in file:

57         img = cv2.imread(str(path+filename), 1)
58         img = cv2.resize(img, (408,306))

59         hasilEkstraksi=[]

60         nilaipergambar=[]
61         nilaipergambar = [filename]

62         kelas = kelas(filename)

63         i = cMoment.ColorMoment.prepWarna(img)
64         R = i[:, :, 0]
65         G = i[:, :, 1]
66         B = i[:, :, 2]

67         mr = cMoment.ColorMoment.mean(R)
68         mg = cMoment.ColorMoment.mean(G)
69         mb = cMoment.ColorMoment.mean(B)

70         sdr = cMoment.ColorMoment.standarDeviasi(R, mr)
71         sdg = cMoment.ColorMoment.standarDeviasi(G, mg)
72         sdb = cMoment.ColorMoment.standarDeviasi(B, mb)

73         sr = cMoment.ColorMoment.skewness(R, mr, sdr)
74         sg = cMoment.ColorMoment.skewness(G, mg, sdg)
75         sb = cMoment.ColorMoment.skewness(B, mb, sdb)

76         nilaipergambar.append(kelas)
77         nilaipergambar.append(mr)
78         nilaipergambar.append(mg)
79         nilaipergambar.append(mb)
80         nilaipergambar.append(sdr)
81         nilaipergambar.append(sdg)
82         nilaipergambar.append(sdb)
83         nilaipergambar.append(sr)
84         nilaipergambar.append(sg)
85         nilaipergambar.append(sb)

86         threshold = smsd.SMSD.prepBentuk(img)
```

```

87     pjpg = smsd.SMSD.panjang(threshold)
88     lbr = smsd.SMSD.lebar(threshold)
89     area = smsd.SMSD.area(threshold)
90     peri = smsd.SMSD.perimeter(threshold)
91     ar = smsd.SMSD.aspectRatio(threshold,pjpg,lbr)
92     roundn = smsd.SMSD.roundness(threshold,area,peri)
93     comp = smsd.SMSD.compactness(threshold,area,pjpg,lbr)
94     rect = smsd.SMSD.rectangulary(threshold,area,peri)
95     rPL = smsd.SMSD.rasioPL(threshold,peri,pjpg)
96     rPLW = smsd.SMSD.rasioPLW(threshold,peri,pjpg,lbr)

97     nilaipergambar.append(pjpg)
98     nilaipergambar.append(lbr)
99     nilaipergambar.append(area)
100    nilaipergambar.append(peri)
101    nilaipergambar.append(ar)
102    nilaipergambar.append(roundn)
103    nilaipergambar.append(comp)
104    nilaipergambar.append(rect)
105    nilaipergambar.append(rPL)
106    nilaipergambar.append(rPLW)

107    wrt.writerow(nilaipergambar)

```

Penjelasan:

Baris 1-2 adalah proses untuk melakukan *import library OpenCV, numpy*.

Baris 3-4 adalah *import os* yang berguna untuk menyediakan cara penggunaan fungsi yang bergantung pada sistem operasi. Kemudian dilakukan *import walk* untuk mendapatkan nama file dalam pohon direktori yang menghasilkan 3 *tuple (dirpath, dirnames, filenames)*.

Baris 5-6 adalah proses untuk meng-*import* fail smsd.py dari folder BENTUK dan fail cMoment.py dari folder WARNA.

Baris 7 dilakukan *import csv* yang berfungsi untuk membuat data dalam bentuk tabel dan disimpan dalam fail csv.

Baris 8-51 adalah sebuah fungsi bernama *kelas* yang digunakan untuk mengambil nama kelas dari setiap citra berdasarkan kode pada nama fail citra. Fungsi ini dilakukan dengan cara membagi nama fail citra dengan memisahkan dengan lambang “_”. Kemudian diambil indeks ke 1 untuk memperoleh kode citra. Apabila kondisi kode tertentu memenuhi, maka akan dikembalikan nilai berupa nama kelas. Sebagai contoh pada baris 10-11 pada kondisi file[1] sama dengan “001” maka akan dikembalikan sebuah nama kelas yaitu “Donat”.

Baris 52-53 adalah proses untuk membaca fail secara berulang pada sebuah folder untuk mengambil citra yang akan melalui proses *pre-processing*, perhitungan fitur warna, dan perhitungan fitur bentuk. Kemudian dilakukan pengambilan daftar yang berisi nama-nama entri dalam direktori yang sudah ditentukan sebelumnya. Daftar tersebut disimpan dalam variabel *file*.

Baris 54-56 adalah proses untuk menyimpan fail .csv pada direktori yang telah ditentukan yang mana sebelumnya disimpan pada variabel *myfile*. Kemudian fail yang disimpan pada direktori diberikan nama *x.csv*.

Baris 57-58 adalah proses untuk membaca citra yang telah diambil sebelumnya. Kemudian ukuran gambar ditetapkan dengan piksel 408x306.

Baris 59-61 adalah proses inialisasi variabel array *hasilEkstraksi* dan variabel array *nilaipergambar* yang diisikan dengan variabel *filename*.

Baris 62 adalah inialisasi variabel *kelas* untuk menyimpan nama kelas setiap citra dari fungsi *kelas*.

Baris 63 adalah inialisasi variabel *i* yang menyimpan hasil dari citra yang telah dilakukan *pre-preprocessing* pada fail *cMoment.py*, kelas *ColorMomen*, fungsi *prepWarna* dengan parameter *img*.

Baris 64-66 pada variabel *R*, *G*, dan *B* secara berturut-turut mengambil nilai piksel pada *channel* merah, hijau, dan biru dari citra pada variabel *i*.

Baris 67-69 adalah proses menyimpan nilai pada variabel *mr*, *mg*, *mb* secara berturut-turut dari citra yang disimpan di *R* untuk *channel* merah, *G* untuk *channel* hijau, dan *B* untuk *channel* biru. Proses dilakukan pada fungsi *mean* yang berada pada kelas *ColorMoment*, fail *cMoment*.

Baris 70-71 adalah proses untuk menyimpan nilai dari proses yang dilakukan pada fungsi *standarDeviasi*, kelas *ColorMoment*, fail *cMoment*. Nilai dari proses tersebut disimpan pada variabel *sdr*, *sdg*, dan *sdb* yang mana secara berturut-turut mengambil nilai *R* dan *mr* sebagai parameter untuk *channel* merah, mengambil nilai *G* dan *mg* sebagai parameter untuk *channel* hijau, dan mengambil nilai *B* dan *mb* sebagai parameter untuk *channel* biru.

Baris 73-75 adalah proses untuk menyimpan nilai pada variabel *sr*, *sg*, *sb* berdasarkan fungsi *skewness* pada kelas *ColorMoment*, fail *cMoment*. Pada variabel tersebut secara berturut-turut mengambil nilai *R*, *mr*, *sdr* sebagai parameter untuk *channel* merah, mengambil nilai *G*, *mg*, *sdg* sebagai parameter untuk *channel* hijau, dan mengambil nilai *B*, *mb*, *sdb* sebagai parameter untuk *channel* biru.

Baris 76-85 adalah proses untuk memasukkan nilai variabel *kelas*, *mr*, *mg*, *mb*, *sdr*, *sdg*, *sdb*, *sr*, *sg*, dan *sb* pada variabel *nilaipergambar*.

Baris 86 adalah proses untuk melakukan *pre-pocessing* pada fungsi *prepBentuk* pada kelas *SMDS*, fail *smsd.py* dengan parameter *img*. Hasil dari proses tersebut disimpan pada variabel bernama *threshold*.

Baris 87 adalah proses untuk memperoleh fitur panjang yang disimpan pada variabel *pjg*. Fitur ini diperoleh dari fungsi *panjang* yang berada pada kelas *SMDS*, fail *smsd.py*.

Baris 88 adalah proses untuk mendapatkan nilai dari fitur lebar pada fungsi *lebar* yang terdapat pada kelas *SMDS*, fail *smsd.py*. Parameter dari fungsi ini adalah citra

pada variabel *threshold*. Hasil dari proses ini kemudian disimpan pada variabel bernama *lbr*.

Baris 89 adalah proses memperoleh fitur area melalui fungsi *area* pada kelas *SMDS*, fail *smsd.py* dengan parameter *threshold*. Hasil proses ini disimpan pada variabel bernama *area*.

Baris 90 adalah proses untuk memperoleh fitur perimeter yang disimpan pada variabel *peri*. Proses ini dijalankan pada fungsi *perimeter*, kelas *SMDS*, fail *smsd.py* dengan parameter *threshold*.

Baris 91 adalah proses mendapatkan fitur *aspect ratio* melalui fungsi *aspectRatio*, yang berada pada kelas *SMDS*, fail *smsd.py*. Parameter fungsi ini ialah *threshold*, *pjpg*, dan *lbr*. Nilai dari fitur tersebut kemudian disimpan pada variabel *ar*.

Baris 92 adalah proses untuk mengambil nilai fitur *roundness* melalui fungsi bernama *roundness*, pada kelas *SMDS*, fail *smsd.py*. Parameter yang digunakan adalah *threshold*, *area*, dan *peri*. Kemudian hasilnya disimpan pada variabel *roundn*.

Baris 93 adalah proses mengambil nilai fitur *compactness* yang tersimpan pada variabel *comp*. Proses ini berjalan berdasarkan fungsi *compactness* yang berada pada kelas *SMDS*, fail *smsd.py*. Parameter yang digunakan antara lain *threshold*, *area*, *pjpg*, dan *lbr*.

Baris 94 adalah proses untuk mendapatkan nilai fitur *rectangulary* berdasarkan fungsi *rectangulary* pada kelas *SMDS*, fail *smsd.py*. Parameter untuk fungsi ini ialah *threshold*, *area*, dan *peri*. Nilai yang diperoleh kemudian disimpan pada variabel bernama *rect*.

Baris 95 adalah proses memperoleh nilai dari fitur rasio perimeter L. Fitur didapatkan melalui fungsi *rasioPL* yang berada di kelas *SMDS*, fail *smsd.py*. Parameter dari fungsi ini ialah *threshold*, *peri*, dan *pjpg*. Hasilnya kemudian akan disimpan pada variabel bernama *rPL*.

Baris 96 adalah proses mendapatkan nilai fitur rasio perimeter L dan W. Fitur yang diperoleh berdasarkan fungsi *rasioPLW* pada kelas *SMDS*, fail *smsd.py*. Parameter fungsi ini terdiri dari *threshold*, *peri*, *pjpg*, dan *lbr*. Hasil yang diperoleh kemudian disimpan pada variabel bernama *rPLW*.

Baris 97-106 adalah proses untuk menambahkan nilai pada variabel *pjpg*, *lbr*, *area*, *peri*, *ar*, *roundn*, *comp*, *rect*, *rPL*, dan *rPLW* ke variabel *nilaipergambar*.

Baris 107 adalah proses untuk menyimpan setiap nilai pada variabel *nilaipergambar* pada fail *.csv*.

5.2.6 Implementasi Perhitungan Jarak *Euclidean*

Source code berikut ini adalah implementasi pada proses perhitungan jarak *Euclidean*. Masukan pada proses ini berupa nilai fitur warna dan bentuk yang telah tersimpan pada fail *.csv*. Keluaran dari proses ini adalah kelas hasil klasifikasi dari data uji yang di-*input* sebagai *query*.

No	File: knn.py
1	import cv2
2	import csv
3	import numpy as np
4	def read_csv(file_name):
5	array_2D = []
6	with open(file_name, 'rt') as csvfile:
7	read = csv.reader(csvfile, delimiter=',')
8	for row in read:
9	if(len(row) != 0):
10	array_2D.append(row)
11	return array_2D
12	def normalisasi(dataset):
13	arr = []
14	for i in range(len(dataset)):
15	arr.append([])
16	for j in range(2, len(dataset[0])):
17	arr[i].append(float(dataset[i][j]))
18	d_min = np.amin(arr, axis=0)
19	d_max = np.amax(arr, axis=0)
20	for i in range(len(dataset)):
21	for j in range(2, len(dataset[0])):
22	dataset[i][j] = (float(dataset[i][j]) - d_min[j-2]) / (d_max[j-2] - d_min[j-2])
23	return dataset
24	def pecahData(dtNormal):
25	dtLatih = []
26	dtUji = []
27	tDtLatih = 0
28	tDtUji = 0
29	# membuat array
30	for i in range(len(dtNormal)):
31	if i < 310:
32	dtLatih.append([])
33	else:
34	dtUji.append([])
35	# memasukkan nilai
36	for j in range(len(dtNormal[0])):
37	if i < 310:
38	dtLatih[tDtLatih].append(dtNormal[i][j])
39	else:
40	dtUji[tDtUji].append(dtNormal[i][j])
41	if i < 310:
42	tDtLatih += 1
43	else:
44	tDtUji += 1
45	return dtLatih, dtUji
46	def euclidean(dtLatih, dtUji):
47	eucl = np.zeros((len(dtUji), len(dtLatih)))

```

46     for h in range(len(dtUji)):
47         for i in range(len(dtLatih)):
48             for j in range(2,len(dtLatih[0])):
49                 eucl[h][i] += (dtUji[h][j] -
dtLatih[i][j])**2
50                 eucl[h][i] = np.sqrt(eucl[h][i])
51     return eucl

52 def namaKelas(dtLatih,dtUji):
53     classname= np.empty((len(dtUji),len(dtLatih)), dtype =
object)
54     for i in range(len(dtUji)):
55         for j in range(len(dtLatih)):
56             classname[i][j] = dtLatih[j][1]
57     return classname

58 def kodeResep(dtLatih, dtUji):
59     resep= np.empty((len(dtUji),len(dtLatih)), dtype = object)
60     for i in range(len(dtUji)):
61         for j in range(len(dtLatih)):
62             kode = dtLatih[j][0].split('_')
63             resep[i][j]=kode[2]
64     return resep

65 data1 = read_csv('D:/KULIAH/Semester 7/Skripsi/kodingan/in
the name of Jesus/x.csv')
66 dtNormal = normalisasi(data1)
67 dtLatih, dtUji = pecahData(dtNormal)
68 eucl = euclidean(dtLatih,dtUji)
69 classname = namaKelas(dtLatih,dtUji)
70 code = kodeResep(dtLatih,dtUji)

71 klasifikasi = []
72 k = 5
73 for i in range (len(eucl)):
74     klasifikasi.append([])
75     for j in range (len(eucl[0])):
76         klasifikasi[i].append([])
77         klasifikasi[i][j].append(classname[i][j])
78         klasifikasi[i][j].append(eucl[i][j])
79         klasifikasi[i][j].append(code[i][j])

80 for i in range(len(klasifikasi)):
81     klasifikasi[i] = sorted(klasifikasi[i], key = lambda x:
x[1])

82 uji = []
83 for i in range (len(klasifikasi)):
84     uji.append([])
85     for j in range (k):
86         uji[i].append(klasifikasi[i][j])

87 resep = []
88 for i in range(len(uji)):
89     for j in range((len(uji[0])):
90         nama = 'D:/KULIAH/Semester 7/Skripsi/kodingan/in the
name of Jesus/resep/'+ str(uji[i][j][2])+'.csv'
91         dok = open(nama,"r")
92         for line in dok:

```

93	<code>print(line)</code>
94	<code>print('=====')</code>

Penjelasan:

Baris 1-3 adalah proses untuk *import library OpenCV, csv, dan numpy* dengan nama *np*.

Baris 4-11 adalah proses untuk membaca fail csv dengan parameter *file_name*. Pada setiap baris, data yang akan dibaca dibatasi oleh tanda “,”. Data pada fail tersebut dibaca secara berulang pada perulangan variabel *row*. Selanjutnya data pada variabel *row* ditambahkan dan disimpan pada variabel *array_2D*.

Baris 12-23 adalah proses melakukan normalisasi pada semua data tanpa memisahkan data uji dan data latih. Variabel *arr* dibuat untuk menyimpan dataset mulai dari kolom indeks yang ke-2 dengan kata lain tidak terdapat string didalamnya. Kemudian pada variabel *d_min* dicari nilai terkecil pada setiap fitur dan variabel *d_max* dicari nilai terbesar pada setiap fitur. Selanjutnya dilakukan perulangan agar semua fitur dapat melakukan normalisasi dengan rumus mengurangi fitur dengan nilai terkecil kemudian dibagi dengan nilai terbesar dikurang nilai terkecil setiap fitur. Seluruh fitur yang telah dinormalisasi disimpan pada variabel *dataset*.

Baris 24-43 adalah fungsi untuk memisahkan data latih dan data uji. Pemisahan dilakukan dengan cara memasukkan data hasil normalisasi pada urutan lebih kecil dari 310 ke variabel *dtLatih* karena jumlah dari data latih adalah 310 dan selain itu dimasukkan ke variabel *dtUji*.

Baris 44-51 adalah fungsi untuk melakukan perhitungan jarak *Euclidean*. Pada variabel *eucl* dilakukan perhitungan jarak data uji terhadap seluruh data latih, sehingga pada satu data uji didapatkan jarak sejumlah data latih. Perhitungan yang dilakukan adalah dengan mencari selisih data uji pada setiap data latih kemudian dipangkatkan dengan dua dan kemudian hasilnya diakarkan menggunakan *numpy package*. Perhitungan jarak yang diperoleh kemudian disimpan pada variabel *eucl*.

Baris 52-57 adalah fungsi untuk mengambil nama kelas pada data latih. Dibuat sebuah variabel *array* untuk menyimpan nama dari setiap data latih bernama *classname*.

Baris 58-64 adalah fungsi untuk mengambil kode kelas pada data latih. Dibuat sebuah variabel *array* bernama kode yaitu pada nama file data latih yang di pisahkan oleh “_” pada index 2.

Baris 65 adalah proses untuk menyimpan data pada fail .csv pada variabel *data1*.

Baris 66 adalah proses untuk menyimpan nilai hasil normalisasi dari fungsi bernama *normalisasi*. Hasil dari fungsi tersebut kemudian disimpan pada variabel bernama *dtNormal*.

Baris 67 adalah proses untuk memisahkan data uji dan data latih melalui fungsi *pecahData*. Parameter yang digunakan pada fungsi ini adalah *dtNormal*. Setelah dilakukan pemisahan, maka data latih akan disimpan pada variabel *dtLatih* dan data uji disimpan pada variabel *dtUji*.

Baris 68 adalah proses untuk menyimpan nilai dari hasil perhitungan jarak *Euclidean* pada fungsi bernama *euclidean*. Parameter dari fungsi ini adalah *dtLatih* dan *dtUji*. Hasil dari fungsi tersebut disimpan pada variabel bernama *eucl*.

Baris 69 adalah proses penyimpanan hasil dari fungsi *namaKelas* pada variabel *classname*. Parameter dari fungsi ini adalah *dtLatih* dan *dtUji*.

Baris 70 adalah proses menyimpan hasil dari fungsi *kodeResep* yang disimpan pada variabel *code*. Parameter fungsi ini merupakan *dtLatih* dan *dtUji*.

Baris 71 adalah inisialisasi variabel *array* bernama *klasifikasi*.

Baris 72 adalah inisialisasi variabel *k* sama dengan 3.

Baris 73-79 adalah perulangan untuk memasukkan nilai dari variabel *classname*, variabel *eucl* dan variabel *code* ke dalam variabel *klasifikasi*.

Baris 80-81 adalah proses untuk mengurutkan nilai pada variabel *klasifikasi* berdasarkan indeks pertama yaitu nilai dari variabel *eucl*.

Baris 82 adalah inisialisasi variabel *array* bernama *kelasGambar*.

Baris 83-86 adalah perulangan untuk mengambil nilai dan kelas pada variabel *klasifikasi* sebanyak nilai *k*. Apabila terdapat nama kelas yang sama akan dijumlahkan dan disimpan nilainya pada variabel *kelasGambar* sesuai dengan indeks masing-masing kelas.

Baris 87-93 adalah perulangan untuk mengambil nilai dari variabel *nama* yaitu variabel yang menyimpan file ekstensi csv yang diambil dari folder sesuai dengan kode yang tersimpan pada data variabel *array* data uji.

Baris 94 adalah mencetak "=====

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini berisi mengenai hasil pengujian dan analisis dari hasil pengujian pada penelitian yang telah dilakukan. Pengujian dilakukan dengan menghitung *mean average precision* (MAP) pada variasi nilai n dan pemakaian fitur warna RGB *color channel*, fitur bentuk *simple morphological shape descriptors*, dan kedua fitur warna dan bentuk pada beberapa nilai *top n*. Nilai n berfungsi sebagai nilai maksimal dokumen teratas yang akan ditampilkan dalam pencarian. Beberapa nilai n yang digunakan terdiri dari $n=5$, $n=10$, $n=15$, $n=20$, dan $n=25$. Nilai n ditentukan dengan kelipatan 5 dikarenakan terdapat *pagination* pada pencarian sehingga pencarian menampilkan beberapa dokumen. Pengujian dilakukan pada data uji sebanyak 37 citra dengan data latih sebanyak 311 citra makanan.

6.1 Pengujian dan Analisis pada Variasi Nilai n

6.1.1 Skenario Pengujian pada Nilai n

Pengujian fitur warna, fitur bentuk, dan fitur warna dan bentuk dengan berbagai nilai n dilakukan untuk mengetahui apakah variasi nilai n berpengaruh pada MAP. Pengujian ini bertujuan untuk mengetahui bagaimana hasil pencarian dengan berbagai tingkat banyaknya dokumen yang di tampilkan. *Query* yang digunakan sebagai data uji yang pertama berupa citra dengan kelas bernama "donat" dan ditunjukkan pada Gambar 6.1.



Gambar 6.1 Citra *query*

Hasil dari pencarian yang diperoleh berdasarkan *query* disajikan pada Tabel 6.1 dengan $n=5$.

Tabel 6.1 Hasil Pencarian dengan n sama dengan 5

Peringkat	Hasil Pencarian		Relevansi
	Resep Makanan	Nama Makanan	
1	<p>Resep Membuat Donat Original</p> <p>Bahan: 500 gr tepung terigu protein tinggi, 100 gr gula pasir, 11 gr atau 1 bungkus ragi instan, 1/4 sdt baking powde, 1/2 sdt garam, 2 kuning telur, 100 gr mentega, 250 ml susu cair</p> <p>Olesen donat: mentega, coklat cair, meses, selai, gula halus</p> <p>Cara Membuat:</p>	Donat	Ya

Peringkat	Hasil Pencarian		Relevansi
	Resep Makanan	Nama Makanan	
	<ol style="list-style-type: none"> 1. Campurkan tepung terigu, ragi instan, gula pasir, <i>baking powder</i>, dan garam kemudian aduk adonan menggunakan tangan. 2. Tambahkan kuning telur. 3. Tuang susu cair sedikit demi sedikit dan aduk adonan sampai merata. 4. Tambah mentega sampai adonan menjadi kalis. 5. Masukkan adonan pada baskom dan diamkan selama 1 jam sampai adonan mengembang. 6. Setelah itu kempeskan adonan sampai semua udara keluar dari adonan. 7. Bagi adonan menjadi 30 bagian dan bentuk bulatan seperti bola. Diamkan lagi selama 10 menit. 8. Kemudian lubangi masing-masing adonan dengan tangan. 9. Goreng donat pada minyak panas dengan api kecil-sedang hingga warna menjadi kuning keemasan. 10. Angkat donat dari penggorengan dan tunggu hingga dingin. 11. Hias donat sesuai selera. <p>(sumber: vemale.com)</p>		
2	<p>Resep Membuat Donat Kentang</p> <p>Bahan: 3 buah kentang (direbus kemudian ditumbuk hingga halus), 250 gr terigu, ½ bungkus ragi, 2 kuning telur, 1 sdm mentega (dicairkan), 1 sdm gula pasir, susu bubuk secukupnya.</p> <p>Cara membuat:</p> <ol style="list-style-type: none"> 1. Campur semua bahan dan diaduk hingga adonan kalis (tidak lengket di tangan). 2. Kemudian di tutupi dengan kain basah hingga mengembang (kira-kira 30 menit sampai 1 jam). 3. Setelah itu dicetak donat menggunakan tangan. 4. Goreng donat pada minyak panas hingga berwarna kuning keemasan. 5. Angkat donat dan dinginkan. 6. Hias donat sesuai selera. <p>(sumber: vemale.com)</p>	Donat	Ya
3	<p>Resep Membuat Donat Isi Selai</p> <p>Bahan: 500 gram tepung terigu protein sedang, 1 butir telur, 5 sdm gula pasir, 1 sdm ragi instant, 2 sdm margarin, 100 gram selai sesuai kesukaan, 50 ml air, gula halus secukupnya untuk taburan.</p> <p>Cara membuat:</p> <ol style="list-style-type: none"> 1. Campur tepung terigu dan ragi instant hingga rata. 2. Tambahkan gula pasir, telur, margarin dan air, aduk semua hingga rata. 	Donat	Ya

Peringkat	Hasil Pencarian		Relevansi
	Resep Makanan	Nama Makanan	
	<p>3. Aduk adonan hingga kalis dan tidak menempel di tangan.</p> <p>4. Tutup adonan dengan serbet bersih dan diamkan selama 1 jam. Tunggu hingga adonan mengembang dua kali ukuran semula.</p> <p>5. Jika sudah mengembang, kempeskan adonan dengan cara menekannya dengan genggaman tangan.</p> <p>6. Ambil adonan kurang lebih 10 gram, bentuk bulat lalu pipihkan. Isi dengan selai, kemudian bulatkan kembali.</p> <p>7. Diamkan adonan selama 15 hingga 20 menit.</p> <p>8. Goreng donat dalam minyak memakai api sedang. Tunggu hingga donat berwarna kuning kecokelatan, angkat, tiriskan.</p> <p>9. Tata donat yang sudah dingin di atas piring saji, beri taburan gula halus dengan bantuan ayakan tepung.</p> <p>(sumber: vemale.com)</p>		
4.	<p>Resep Donat Coklat Empuk</p> <p>Bahan: 600 gram tepung terigu, 60 gram susu bubuk, 90 ml air dingin, 20 gram baking powder, 30 gram ragi instan, 250 gram gula pasir halus, ¼ sdt garam, 2 butir telur ayam, 300 gram mentega</p> <p>Cara membuat:</p> <ol style="list-style-type: none"> 1. Pertama, campurkan semua bahan donat dan diuleni sampai kalis. Diamkan selama 30 menit hingga adonan mengembang 2x lipat. 2. Kemudian, bulatkan adonan sampai habis dan diamkan selama 10 menit. 3. Setelah adonan benar-benar mengembang, pipihkan untuk diberi isian. Bentuk bulat kembali dan diamkan selama 10 menit. 4. Panaskan minyak, goreng donat hingga matang kuning keemasan. Jika sudah dingin, celupkan bagian atas donat ke dalam dark cooking coklat dan selanjutnya beri motif garis dengan white cooking coklat. 5. Donat Coklat siap untuk dihidangkan. <p>(sumber: resepsimbok)</p>	Donat	Ya
5.	<p>Resep Roti Gandum</p> <p>Bahan: 150 gr tepung terigu protein tinggi, 150 gr tepung gandum halus, 5 gr tepung gandum kasar, 22.5 gr susu bubuk, 60 gr gula pasir, 1 1/2 sdt ragi, 270 gr (30 gr) whippy bubuk, 2btr telur ayam kampung, susu cair, 45 gr butter (saya blueband cookies n cream), 3 gr garam, sedikit oat untuk taburan di atas roti.</p> <p>Cara membuat:</p>	Roti Gandum	Tidak

Peringkat	Hasil Pencarian		Relevansi
	Resep Makanan	Nama Makanan	
	<ol style="list-style-type: none"> 1. Campur semua bahan kecuali butter dan garam,ulenin sampai menyatu dan kalis. 2. Masukkan butter dan garam. Ulenin atau mixer sampai kalis elastis. 3. Diamkan dalam wadah tertutup selama 45 menit atau sampai adonan mengembang 2kali lipat. 4. Kempiskan adonan,bagi adonan dan bentuk sesuai selera. (saya bulatkan dan giling memanjang,gulung adonan hingga rapat dan taruh di loyang roti tawar) 5. Diamkan lagi selama 1 jam atau sampai mengembang hingga 2/3 ukuran loyang. Panggang di oven yang telah dipanaskan sebelumnya dengan suhu 180c selama 20 menit atau sampai matang. 		

Melalui peringkat dokumen pada pencarian resep makanan, maka dapat diperoleh nilai rata-rata *precision* dari setiap kelas hasil pencarian dan MAP pada keseluruhan *query*. Nilai *precision* diperoleh dari relevansi antara kelas pada *query* dan kelas dari hasil pencarian pada setiap peringkat. Proses perhitungan MAP pada seluruh data uji menggunakan fitur warna dan bentuk dengan $n=5$ dapat dilihat pada Tabel 6.2. Pada tabel tersebut dapat dilihat nilai MAP yang diperoleh adalah 94,1892%.

Tabel 6.2 Perhitungan MAP

Data Uji	Kode Gambar	Kelas	Relevansi/ Precision	Peringkat					Rata-rata
				1	2	3	4	5	
1	001_0003	Donat	Relevansi	Ya	Ya2	Ya3	Ya4	Ya5	1
			Precision	1	1	1	1	1	
2	001_0297	Donat	Relevansi	Ya	Ya	Tidak	Tidak	Tidak	1
			Precision	1	1	0	0	0	
3	001_0303	Donat	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
4	001_0614	Donat	Relevansi	Ya	Ya	Ya	Tidak	Tidak	1
			Precision	1	1	1	0	0	
5	002_0017	Roti Gandum	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
6	002_0031	Roti Gandum	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
7	002_0330	Roti Gandum	Relevansi	Ya	Ya	Tidak	Tidak	Tidak	1
			Precision	1	1	0	0	0	
8	002_0626	Roti Gandum	Relevansi	Ya	Ya	Tidak	Tidak	Tidak	1
			Precision	1	1	0	0	0	



Data Uji	Kode Gambar	Kelas	Relevansi/ Precision	Peringkat					Rata-rata
				1	2	3	4	5	
9	003_0031	Roti Tawar	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
10	003_0336	Roti Tawar	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
11	003_0340	Roti Tawar	Relevansi	Ya	Ya	Tidak	Tidak	Tidak	1
			Precision	1	1	0	0	0	
12	003_0636	Roti Tawar	Relevansi	Tidak	Tidak	Tidak	Tidak	Ya	0,2
			Precision	0	0	0	0	0,2	
13	004_0042	Mie Goreng Keriting	Relevansi	Ya	Ya	Tidak	Tidak	Tidak	1
			Precision	1	1	0	0	0	
14	005_0053	Mie Goreng Gepeng	Relevansi	Ya	Ya	Ya	Tidak	Tidak	1
			Precision	1	1	1	0	0	
15	006_0060	Telur Mata Sapi	Relevansi	Ya	Ya	Tidak	Tidak	Tidak	1
			Precision	1	1	0	0	0	
16	006_0351	Telur Mata Sapi	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
17	006_00359	Telur Mata Sapi	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
18	006_0647	Telur Mata Sapi	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
19	007_0072	Telur Dadar	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
20	008_0081	Fried Chicken	Relevansi	Ya	Ya	Ya	Tidak	Tidak	1
			Precision	1	1	1	0	0	
21	009_0091	Rendang	Relevansi	Tidak	Tidak	Tidak	Tidak	Tidak	0
			Precision	0	0	0	0	0	
22	009_0093	Rendang	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
23	009_00371	Rendang	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
24	009_0377	Rendang	Relevansi	Ya	Tidak	Tidak	Tidak	Tidak	1
			Precision	1	0	0	0	0	
25	021_0199	Nasi Kuning	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
26	022_0209	Nasi Merah	Relevansi	Ya	Ya	Ya	Ya	Tidak	1
			Precision	1	1	1	1	0	
27	024_0515	Beng-beng	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
28	024_0525	Beng-beng	Relevansi	Ya	Ya	Ya	Ya	Ya	1
			Precision	1	1	1	1	1	
29	024_0710	Beng-beng	Relevansi	Ya	Ya	Ya	Tidak	Ya	0,95



Data Uji	Kode Gambar	Kelas	Relevansi/ Precision	Peringkat					Rata-rata
				1	2	3	4	5	
30	025_0230	Soba Mi	Precision	1	1	1	0	0,8	1
			Relevansi	Ya	Ya	Ya	Tidak	Tidak	
31	027_0252	Keripik Kentang	Precision	1	1	0	0	0	1
			Relevansi	Ya	Ya	Tidak	Tidak	Tidak	
32	029_0268	Biskuit	Precision	1	1	1	1	1	1
			Relevansi	Ya	Ya	Ya	Ya	Ya	
33	029_0537	Biskuit	Precision	1	1	0	0	0	1
			Relevansi	Ya	Ya	Tidak	Tidak	Tidak	
34	029_0538	Biskuit	Precision	1	1	1	1	1	1
			Relevansi	Ya	Ya	Ya	Ya	Ya	
35	031_0287	Genji Pie	Precision	1	1	1	1	1	1
			Relevansi	Ya	Ya	Ya	Ya	Ya	
36	031_0595	Genji Pie	Precision	1	1	1	1	1	1
			Relevansi	Ya	Ya	Ya	Ya	Ya	
37	031_0605	Genji Pie	Precision	1	0	0	0	0,4	0,7
			Relevansi	Ya	Tidak	Tidak	Tidak	Ya	
MAP								0,941892	

Hasil pengujian MAP pada keseluruhan nilai n pada kombinasi fitur warna dan bentuk disajikan pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian n variasi

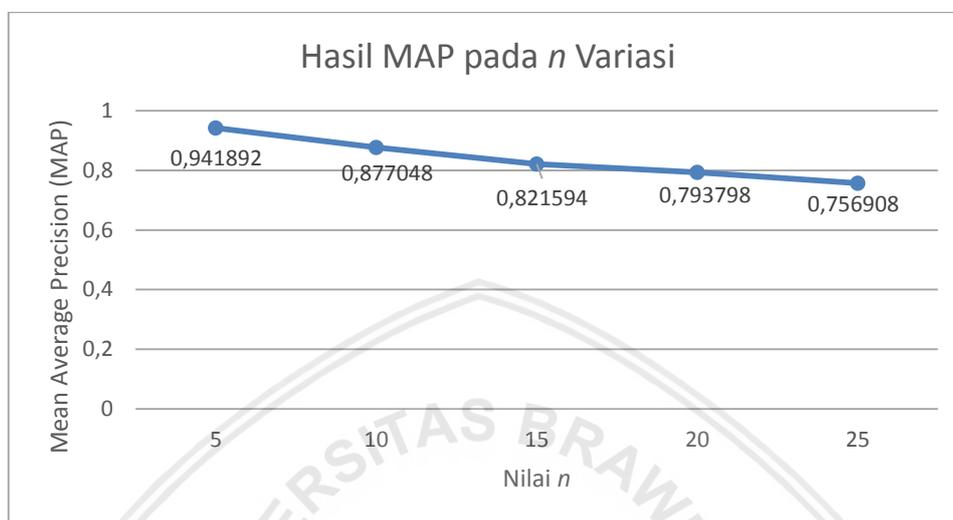
Nilai n	5	10	15	20	25
MAP	0,941892	0,877048	0,821594	0,793798	0,756908

Pada Tabel 6.3 dapat dilihat hasil pengujian menggunakan MAP pada fitur warna dan bentuk dengan nilai n yang bervariasi. Berdasarkan hasil tersebut dapat dibandingkan perolehan MAP terbesar berada pada $n = 5$ yaitu sebesar 0,941892 dan yang terkecil pada $n = 25$ yaitu 0,760299. Perubahan nilai yang paling signifikan terlihat pada $n = 5$ dan 10, yaitu memiliki selisih sebesar 6,7696%. Berdasarkan data tersebut, dapat diketahui bahwa nilai MAP mengalami penurunan ketika nilai n semakin besar.

6.1.2 Analisis Pengujian pada Nilai n

Dari hasil pengujian yang tertera pada Tabel 6.3, disajikan hasil pengujian MAP pada n variasi yang dapat dilihat pada Gambar 6.2. Pada grafik tersebut, terlihat penurunan pada nilai MAP saat nilai n semakin besar. Hal tersebut menunjukkan bahwa nilai n mempengaruhi MAP yang mana nilai MAP akan semakin baik apabila nilai n kecil. Dari pengujian yang dilakukan, dapat dilihat MAP tidak

memperhatikan banyaknya dokumen relevan yang muncul tetapi peringkat dari dokumen yang relevan. Hal tersebut mengakibatkan ketika nilai n semakin besar, maka hasil pencarian akan memberikan dokumen relevan berada pada peringkat yang rendah, sehingga hasil pencarian pada beberapa *query* memberikan nilai *average precision* di bawah 0,5.



Gambar 6.2 Grafik MAP Hasil Pengujian pada n Variasi

6.2 Pengujian dan Analisis Kombinasi Fitur warna dan Fitur Bentuk

Pengujian pada kombinasi fitur warna dan bentuk dilakukan dengan tujuan untuk mengetahui keefektifan fitur warna RGB *color channel* dan fitur bentuk *simple morphological shape descriptor* dalam memberikan hasil pencarian resep makanan.

6.2.1 Skenario Pengujian Kombinasi Fitur warna dan Fitur Bentuk

Pengujian kombinasi fitur warna dan bentuk dilakukan pada n variasi. Pada pengujian ini dilakukan perhitungan MAP pada fitur warna RGB *color channel*, fitur bentuk *simple morphological shape descriptors*, dan kombinasi antara kedua fitur warna dan bentuk tersebut. Hasil dari pengujian yang dilakukan disajikan pada Tabel 6.4 berikut.

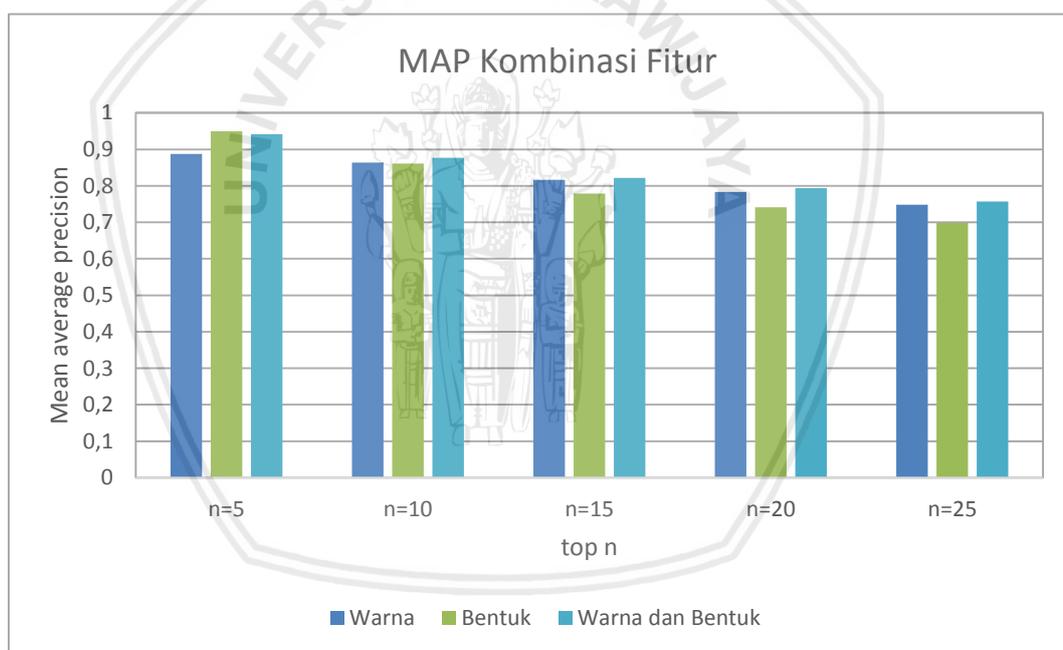
Tabel 6.4 Hasil Pengujian Kombinasi Fitur

No.	Fitur	MAP				
		$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
1	RGB Color Channel	0,887538	0,863817	0,816769	0,782759	0,748129
2	Simple Morphological Shape Descriptors	0,949212	0,861315	0,778565	0,741303	0,698037
3	RGB Color Channel dan Simple Morphological Shape Descriptors	0,941892	0,877048	0,821594	0,793789	0,756908

Berdasarkan hasil pengujian yang telah dilakukan, diperoleh hasil MAP terbaik berada pada kombinasi fitur *Simple Morphological Shape Descriptors* pada nilai $n = 5$ yaitu 94,9212%. Pada kombinasi antara fitur warna dan bentuk diperoleh hasil MAP terbaik pada $n = 10$ sebesar 87,7049%, $n = 15$ sebesar 82,1594%, $n = 20$ sebesar 79,3789% dan $n = 25$ sebesar 75,6908%. Dan pada fitur *color channel* diperoleh nilai MAP terendah pada semua nilai n . Hal tersebut menunjukkan kombinasi antara fitur warna dan bentuk memberikan pengaruh pada nilai MAP.

6.2.2 Analisis Pengujian pada Kombinasi Fitur Warna dan Bentuk

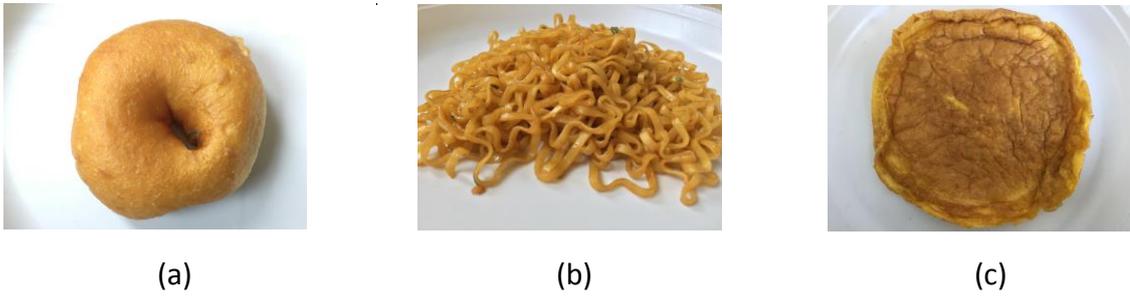
Berdasarkan hasil pengujian pada Tabel 6.4, diketahui bahwa nilai MAP terbesar diperoleh dengan menggunakan fitur bentuk *simple morphological shape descriptors* pada nilai $n = 5$ dan pada kombinasi dari fitur warna RGB *color channel* dan fitur bentuk *simple morphological shape descriptors* pada nilai pada $n = 10$, $n = 15$, $n = 20$, dan $n = 25$. Kemudian nilai MAP fitur warna RGB *color channel* mendapatkan nilai MAP yang terkecil pada semua n . Untuk memudahkan perbandingan nilai MAP pada kombinasi fitur disajikan grafik pada Gambar 6.3.



Gambar 6.3 Grafik MAP pada Hasil Pengujian Kombinasi Fitur

Pengujian pada fitur warna saja mendapatkan hasil pengujian yang paling rendah disebabkan terdapatnya objek citra memiliki warna yang hampir sama dengan objek citra kelas lain. Sebagai contoh yang ditunjukkan pada Gambar 6.4. Pada Gambar 6.4 (a) citra kelas “donat” memiliki warna yang mirip dengan Gambar 6.4 (b) citra dengan kelas “mi goreng gepeng” dan Gambar 6.4 (c) citra dengan kelas “telur dadar” yaitu warna kuning kecoklatan, sehingga fitur warna akan berada pada nilai yang berdekatan. Selain itu terdapatnya perbedaan intensitas cahaya pada objek yang berbeda namun berada pada kelas yang sama seperti contoh ditunjukkan pada Gambar 6.5. Kedua gambar tersebut berada pada

kelas yang sama namun dikarenakan intensitas yang berbeda, memberikan hasil fitur warna yang berbeda pula.

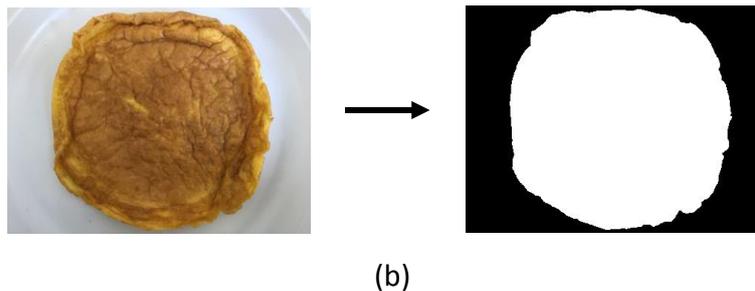
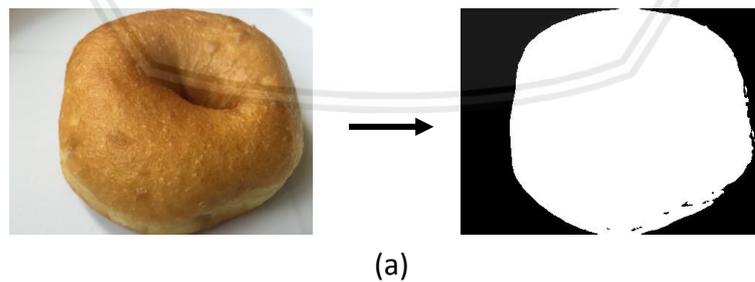


Gambar 6.4 Citra yang Memiliki Warna Mirip



Gambar 6.5 Citra dengan Intensitas Cahaya yang Berbeda

MAP tertinggi diperoleh pada penggunaan fitur bentuk dengan $n = 5$ yaitu 0,949212. Hasil tersebut hanya memiliki selisih 0,732% dengan kombinasi fitur warna dan bentuk pada n yang sama yaitu 94,1892%. Selain pada nilai $n = 5$, penggunaan fitur bentuk saja memperoleh nilai MAP yang lebih rendah dibandingkan dengan penggunaan kedua fitur.



Gambar 6.6 Citra yang Memiliki Bentuk Hampir Sama

Hal tersebut diakibatkan oleh terdapatnya kemiripan bentuk pada objek kelas tertentu dengan kelas lain. Sebagai contoh pada Gambar 6.6 (a) dan (b) merupakan citra dengan kelas yang berbeda. Kelas pada Gambar 6.6 (a) merupakan citra biner dari donat sedangkan (b) merupakan citra biner dari telur dadar. Hal lain yang menyebabkan fitur bentuk tidak memberikan hasil pencarian yang baik yaitu terdapat perbedaan bentuk pada objek kelas yang sama. Seperti yang terlihat pada Gambar 6.7 (a) dan (b), kedua gambar tersebut adalah citra dengan kelas biskuit. Karena diambil dari sisi dan tingkat kemiringan yang berbeda mempengaruhi bentuk dari objek pada citra. Misalnya untuk fitur panjang dan lebar kedua gambar ini akan berbeda karena diambil tidak dengan posisi yang sama.



Gambar 6.7 Citra dengan Kelas Sama tetapi Bentuk Berbeda

Penggunaan kombinasi antara fitur warna dan bentuk cenderung memberikan nilai MAP yang lebih baik. Hal tersebut dibuktikan pada $n = 10$ ketika fitur yang digunakan hanya fitur warna, hasil dari MAP bernilai 86,3817%. Ketika fitur bentuk ditambahkan maka nilai MAP meningkat sebesar 1,3231% menjadi 87,7048%. Seperti contoh pada Gambar 6.4 (a), (b), dan (c). Meskipun ketiga gambar ini memiliki warna yang hampir sama, namun bentuk dari ketiga gambar tersebut berbeda. Selain itu dibuktikan pada $n = 15$ ketika hanya digunakan fitur bentuk saja diperoleh MAP sebesar 77,8565% dan ketika ditambahkan fitur warna akurasi 4,3029% menjadi 82,1594%. Jelas bahwa penggunaan kombinasi kedua fitur warna dan fitur bentuk akan meningkatkan hasil pengujian menggunakan MAP.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan pengujian dan analisis yang sudah dilakukan, maka diperoleh kesimpulan seperti berikut.

1. Berdasarkan penelitian yang dilakukan, dapat ditarik kesimpulan bahwa penggunaan fitur warna RGB *Color Channel* dan fitur bentuk *Simple Morphological Shape Descriptor* mampu memberikan hasil pencarian resep makanan. *Input* yang diberikan pada sistem adalah *query* berupa sebuah citra makanan. Pada citra makanan tersebut kemudian dilakukan *pre-processing* sehingga menghasilkan citra makanan dengan *background* hitam untuk fitur warna dan citra biner untuk fitur bentuk. Selanjutnya dilakukan ekstraksi fitur warna dengan menghitung nilai *mean*, *standard deviation*, dan *skewness* pada dan ekstraksi bentuk dengan mencari nilai panjang (*L*), lebar (*W*), luas area (*A*), perimeter (*P*), *aspect ratio* (*AR*), *roundness* (*R*), *compactness* (*PA*), *rectangularity* (*N*), rasio perimeter pada panjang (*PL*), dan rasio perimeter pada panjang dan lebar (*PLW*) pada objek. Setelah didapatkan fitur dari setiap citra kemudian dilakukan normalisasi menggunakan *Min-max Normalization* untuk merubah nilai fitur berada pada interval 0-1. Selanjutnya dilakukan perhitungan jarak antara data uji terhadap seluruh data latih dan diurutkan berdasarkan jarak terkecil. Kemudian pencarian dilakukan berdasarkan nilai *top n*. Hasil dari proses ini akan memberikan dokumen resep makanan berdasarkan urutan terkecil dari jarak yang telah dihitung.
2. Dari pengujian pada variasi nilai *n* didapatkan nilai MAP tertinggi pada nilai *n*=5 sebesar 94,1892% dan nilai terendah berada pada nilai *n*=25 yaitu 75,6908%. Kesimpulan yang dapat diambil ialah semakin besar nilai *n* akan memberikan hasil MAP yang semakin kecil.
3. Berdasarkan pengujian pada penggunaan fitur RGB *color channel* RGB dan fitur bentuk *simple morphological shape descriptors*, didapatkan hasil terbaik pada penggunaan kombinasi fitur warna dan bentuk pada nilai *n*=10 sebesar 87,7048%, *n*=15 sebesar 82,1594%, *n*=20 sebesar 79,3789%, dan *n*=25 sebesar 75,6908%. Penggunaan fitur warna RGB *color channel* memperoleh nilai MAP yang terkecil pada semua nilai *n*. Selain pada nilai *n*=5, penggunaan salah satu fitur saja memberikan hasil MAP yang lebih kecil dibandingkan penggunaan kedua fitur.

7.2 Saran

Terdapat beberapa saran yang diberikan pada penelitian ini ialah sebagai berikut.

1. Perlu dilakukan *pre-processing* dengan menambahkan erosi dan dilasi untuk megurangi noise pada citra.

2. Perlu dilakukan normalisasi citra dan pengambilan data yang lebih baik dengan memperhatikan intensitas cahaya dan kemiringan posisi pengambilan citra yaitu menyamakan sisi dan tingkat kemiringan pada semua kelas citra.



DAFTAR REFERENSI

- Arsita, A. (2017). Simulakra Baudrillard Dalam Multidimensi Posmodernisme: Kajian Fotografi Makanan Dalam Media Sosial Instagram. *Jurnal Rekam*, 13(2), 85–98. Retrieved from <http://www.imgrum.org/user/redwings>.
- Aryati, K. S., Wirayuda, T. A. B., & Dayawati, R. N. (2009). Pendahuluan.
- Budianita, E., Jasril, J., & Handayani, L. (2015). Implementasi Pengolahan Citra dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi Berbasis Web. *Jurnal Sains Dan Teknologi Industri*, 12(Vol 12, No 2 (2015): Juni 2015), 242–247. Retrieved from <http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/1005>
- Bunyamin, H., Negara, C. P., Informasi, F. T., & Maranatha, U. K. (2008). Aplikasi Information Retrieval (IR) CATA Dengan Metode Generalized Vector Space Model. *Jurnal Informatika*, 4(1), 29–38.
- Caglayan, A., Guclu, O., & Can, A. B. (2013). New Trends in Image Analysis and Processing – ICIAP 2013, 8158(September). <https://doi.org/10.1007/978-3-642-41190-8>
- Chandrasekhar, T., Thangavel, K., & Elayaraja, E. (2011). Effective Clustering Algorithms for Gene Expression Data. *ArXiv Preprint ArXiv: ...*, 32(4), 25–29. Retrieved from <http://arxiv.org/abs/1201.4914>
- Dewi, R. K., & Ginardi, R. V. H. (2014). Identifikasi Penyakit pada Daun Tebu dengan Gray Level Co-occurrence Matrix dan Color Moments, 1(2), 70–77.
- Gonzalez, R., & Woods, R. (2002). *Digital image processing*. Prentice Hall. [https://doi.org/10.1016/0734-189X\(90\)90171-Q](https://doi.org/10.1016/0734-189X(90)90171-Q)
- Halim, A., Dewi, C., Angkasa, S., Studi, P., & Informatika, T. (2013). Aplikasi Image Retrieval Menggunakan Kombinasi Metode Color Moment Dan Gabor Texture, 14(2), 109–117.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. San Francisco, CA, itd: Morgan Kaufmann. <https://doi.org/10.1016/B978-0-12-381479-1.00001-0>
- Hasibuan, Z. A. (2007). Metodologi Penelitian Pada Bidang Ilmu Komputer dan Teknologi Informasi.
- Ibrahim, N., Hidayat, B., & Darana, S. (2017). Deteksi Kualitas Keju Dengan Metoda Discrete Gabor Wavelet Dengan Klasifikasi K-Nearest Neighbor (K-NN) Pada Citra, 4(2), 1710–1717.
- Kadir, A., Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta : ANDI
- Layona, R., Tunardi, Y., & Tanoto, D. F. (2014). Image Retrieval Berdasarkan Fitur Warna, Bentuk, dan Tekstur. *ComTech: Computer, Mathematics and*

- Engineering Applications*, 5(2), 1073.
<https://doi.org/10.21512/comtech.v5i2.2369>
- Leidiyana, H. (2013). Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor. *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, 1(1), 65–76.
- Lestari, S. Y., & Kursini. (2012). Membangun Aplikasi Mobile Resep Masakan Asia (Indonesia,China,Jepang)Berbasis Android. *Jurnal Dasi*, 13(1), 36–41.
- Made, N., & Iswari, S. (2017). Naive Bayes dalam Pengklasifikasian Kesegaran Ikan Menggunakan Media Foto, *IX(2)*, 114–117.
- Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval. Online*. <https://doi.org/>
- Mustafa, R., & Zhu, D. (2014). An Efficient Lip-reading Method Using K-nearest Neighbor Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 13(1). <https://doi.org/10.11591/telkomnika.v13i1.6872>
- Mutrofin, S., Kurniawardhani, A., Izzah, A., & Masrur, M. (2014). Optimasi teknik klasifikasi modified k nearest neighbor menggunakan algoritma genetika. *Jurnal Gamma*, (September), 130–134.
<https://doi.org/10.1006/rwcy.2000.08002.SUMMARY>
- Nursalim, Suprapedi, & H.himawan. (2014). Klasifikasi Bidang Kerja Lulusan Menggunakan Algoritma K-Nearest Neighbor. *Teknologi Informasi*, 10(April), 31–43.
- Prasetyo, E. 2009. *Data Mining Mengolah Data Menjadi Informasi Menggunakan Matlab*. Gresik: Andi Yogyakarta.
- Putra, D., (2010). *Pengolahan Citra Digital*. Yogyakarta: ANDI.
- Sari, Y. A., Dewi, R. K., & Fatichah, C. (2014). Seleksi Fitur Menggunakan Ekstraksi Fitur Bentuk, Warna, Dan Tekstur Dalam Sistem Temu Kembali Citra Daun. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 12(1), 1.
<https://doi.org/10.12962/j24068535.v12i1.a39>
- Setyahadi, P. (2014). Rancang Bangun Aplikasi Resep Masakan Berbasis Mobile Dengan Metode Case-Based Reasoning.
- Wäldchen, J., & Mäder, P. (2018). *Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review. Archives of Computational Methods in Engineering* (Vol. 25). Springer Netherlands.
<https://doi.org/10.1007/s11831-016-9206-z>
- Wedianto, A., Sari, H. L., & H, Y. S. (2016). Analisa Perbandingan Metode Filter Gaussian , Mean Dan Median Terhadap Reduksi Noise. *Jurnal Media Infotama*, 12(1), 21–30.
- Wijaya, W. C. (2016). Sistem Penunjang Keputusan Pemilihan Resep Makanan Berdasarkan Ketersediaan Bahan Menggunakan Metode Forward Chaining. *Journal Of Information and Technology*, 04(01), 47–51.

Yuliana, R. (2014). *Aplijasi Resep Pembuatan Cake 3d Berbasis Multimedia Interaktif*.

Yuristiawan, D., & Santoso, H. A. (2015). Identifikasi Kualitas Daging Sapi Berbasis Android Dengan Ekstraksi Fitur Warna Dan Klasifikasi Knn. *Skripsi, Fakultas Ilmu Komputer*, 1–8. Retrieved from <http://eprints.dinus.ac.id/17000/>

