

ESTIMASI SISA MAKANAN OTOMATIS PADA KOTAK MAKAN MENGUNAKAN SEGMENTASI CITRA BERBASIS CLUSTERING DENGAN ALGORITME K-MEANS

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Haris Bahtiar Asidik

NIM: 175150218113040



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021**

PENGESAHAN

ESTIMASI SISA MAKANAN OTOMATIS PADA KOTAK MAKAN MENGGUNAKAN
SEGMENTASI CITRA BERBASIS CLUSTERING DENGAN ALGORITME K-MEANS

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:

Haris Bahtiar Asidik

NIM: 175150218113040

Skripsi ini telah diuji dan dinyatakan lulus pada
02 Juli 2021

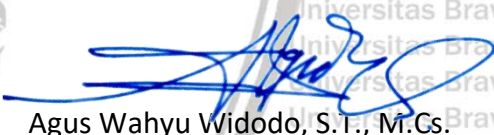
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1

Dosen Pembimbing 2


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

NIK: 2016098807152001


Agus Wahyu Widodo, S.T., M.Cs.

NIP: 197408052001121001

Mengetahui

Ketua Jurusan Teknik Informatika




Achmad Basuki, S.T., M.MG., Ph.D.

NIP: 197411182003121002

PERNYATAAN ORISINALITAS

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

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

Malang, 02 Juli 2021



Haris Bahtiar Asidik
NIM: 175150218113040



PRAKATA

Puji syukur ke hadirat Tuhan Yang Maha Esa yang telah memberikan berkat sehingga skripsi yang berjudul “Estimasi Sisa Makanan Otomatis Pada Kotak Makan Menggunakan Segmentasi Citra Berbasis Clustering Dengan Algoritme K-Means” ini dapat terselesaikan.

Penulis menyadari bahwa dalam penyelesaian dan penyusunan skripsi ini tidak akan terwujud tanpa adanya bantuan dan dorongan dari beberapa pihak. Oleh karena itu, dengan skripsi ini penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu Yuita Arum Sari, S.Kom., M.Kom., selaku dosen pembimbing utama yang telah meluangkan waktu untuk memberikan pengarahan dan masukan kepada penulis.
2. Bapak Agus Wahyu Widodo, S.T., M.Cs., selaku dosen pembimbing kedua yang telah meluangkan waktu untuk memberikan pengarahan dan masukan kepada penulis.
3. Kedua orangtua yang selalu memberikan dukungan, dan juga do’a kepada penulis.
4. Seluruh Dosen Fakultas Ilmu Komputer yang telah mengajarkan ilmu dan pengetahuannya kepada penulis selama perkuliahan.
5. Teman-teman penulis yang memberikan semangat kepada penulis.

Penulis menyadari bahwa dalam masa penyusunan skripsi ini masih banyak kekurangan, dengan itu penulis sangat mengharapkan saran dan kritik yang membangun untuk skripsi ini. Semoga skripsi ini dapat membawa manfaat bagi semua pihak dan digunakan sebagaimana mestinya.

Malang, 02 Juli 2021

Haris Bahtiar Asidik
harisbahtiarasidik@gmail.com

ABSTRAK

Haris Bahtiar Asidik, Estimasi Sisa Makanan Otomatis pada Kotak Makan Menggunakan Segmentasi Citra Berbasis Clustering Dengan Algoritme K-Means
Pembimbing: Yuita Arum Sari, S.Kom., M.Kom. dan Agus Wahyu Widodo, S.T., M.Cs.

Makanan adalah sumber energi bagi makhluk hidup, mengonsumsi makanan dengan porsi yang sesuai standar dapat membantu memenuhi kebutuhan gizi dalam tubuh. Namun pola makan yang berlebihan akan mengakibatkan dampak buruk pada tubuh. Penyelenggaraan makanan di rumah sakit merupakan sistem pendukung dalam mempercepat proses penyembuhan pasien dari penyakit, sisa makanan pasien mencerminkan rendahnya daya terima pasien terhadap makanan yang dapat meningkatkan risiko malnutrisi. Saat ini Volume global pemborosan makanan diperkirakan mencapai 1,6 miliar ton dengan karbon sisa makanan yang dihasilkan diperkirakan mencapai 3,3 miliar ton CO₂ setara dengan gas rumah kaca yang dilepaskan ke atmosfer per tahun. Dibutuhkan alat untuk mengetahui seberapa banyak makanan yang telah dikonsumsi oleh seseorang. Dengan perkembangan teknologi, maka proses penghitungan bobot sisa makanan dapat dilakukan dengan memanfaatkan citra dari makanan sebelum dimakan dan citra makanan setelah dimakan. Citra makanan pada kotak makan disegmentasi untuk didapatkan segmen makanan pada citra melalui metode *clustering* menggunakan algoritme K-means berdasarkan tingkat warna biru pada citra. Diperoleh hasil akurasi *Intersection over Union (IoU)* segmentasi pada citra dengan latar belakang putih mencapai 98,9%. Berdasarkan hasil segmentasi yang didapatkan selanjutnya dilakukan estimasi bobot sisa makanan dengan metode *Elementary Leftover Food Estimation* (EFLE). Dengan menggunakan *Root Mean Square Error* (RMSE), hasil pengujian dengan *error* terkecil mencapai 1,12. Ini menunjukkan bahwa metode yang diusulkan mampu memproyeksikan berat sisa makanan.

Kata kunci: k-means, segmentasi citra, estimasi bobot makanan, pengolahan citra makanan, segmentasi citra makanan

ABSTRACT

Haris Bahtiar Asidik, Automatic Estimation of Food Leftovers in Tray Box Using Clustering-Based Image Segmentation With K-Means Algorithm

Supervisors: Yuita Arum Sari, S.Kom., M.Kom. and Agus Wahyu Widodo, S.T., M.Cs.

Food is a source of energy for living things, consuming food in appropriate portions can help meet the nutritional needs of the body. However, excessive eating patterns will have an impact on the body. Food service in the hospital is a support system in supporting the patient's healing process from illness, food waste reflects the patient's low acceptance of food which can increase the risk of malnutrition. The global volume of food wastage is currently estimated at 1.6 billion tonnes with the resulting food carbon being an estimated 3.3 billion tonnes of CO₂ equivalent of greenhouse gases that are released into the atmosphere per year. It takes a tool to find out how much food a person has consumed. With the development of technology, the calculation of the weight of food waste can be done by utilizing the image of the food before eating and the image of the food after being eaten. The food image in the lunch box is segmented to get the food segment in the image through the clustering method using the K-means algorithm based on the blue color level in the image. The results of the Intersection over Union (IoU) segmentation accuracy on images with a white background reached 98.9%. Based on the segmentation results obtained, the weight of leftover food was predicted using the Elementary Leftover Food Estimation (EFLE) method. By using the Root Mean Square Error (RMSE), test results with the smallest error reached 1.12. This indicates that the proposed method is capable of projecting the weight of the food residue.

Keywords: k-means, image segmentation, food weight prediction, food image processing, food image segmentation

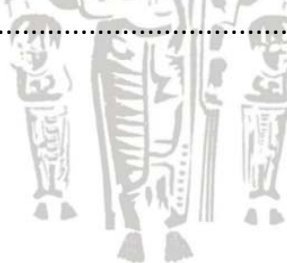
DAFTAR ISI

ESTIMASI SISA MAKANAN OTOMATIS PADA KOTAK MAKAN	
MENGUNAKAN SEGMENTASI CITRA BERBASIS CLUSTERING DENGAN	
ALGORITME K-MEANS.....	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Kajian Pustaka	5
2.2 Sisa Makanan	6
2.3 Estimasi Bobot	7
2.4 Ruang Warna	7
2.4.1 Ruang Warna RGB.....	7
2.4.2 Ruang Warna HSV	7
2.4.3 Ruang Warna YUV / YCbCr	8
2.5 Segmentasi Citra	8
2.6 Clustering	8
2.7 Preprocessing	8
2.7.1 Power-Law Transformation.....	8
2.8 K-Means Clustering	9
2.8.1 Kondisi Berhenti.....	10
2.9 Postprocessing	10
2.9.1 Flood Fill	10
2.9.2 Morfologi - Erosi	10
2.9.3 Morfologi - Dilatasi	10
2.10 Perhitungan Bobot Sisa Makanan	11
2.11 Evaluasi	11
2.11.1 Intersection over Union (IoU)	11
2.11.2 Akurasi Bobot Sisa Makanan	11
2.11.3 Root Mean Square Error (RMSE)	11

BAB 3 METODOLOGI PENELITIAN	12
3.1 Tipe Penelitian	12
3.2 Strategi Penelitian	12
3.3 Metode Pengumpulan Data	12
3.4 Peralatan Pendukung	13
3.4.1 Perangkat Keras	13
3.4.2 Perangkat Lunak	13
3.5 Metode Perancangan Algoritme	14
3.6 Pengujian	14
3.7 Penarikan Kesimpulan dan Saran	14
BAB 4 PERANCANGAN	15
4.1 Perancangan Algoritme	15
4.1.1 <i>Preprocessing</i> Citra	16
4.1.2 Segmentasi K-means	18
4.1.3 <i>Postprocessing</i> Hasil Segmentasi	19
4.1.4 <i>Elementary Leftover Food Estimation (EFLE)</i>	21
4.2 Perhitungan Manual	22
4.2.1 Segmentasi K-means	22
4.2.2 <i>Intersection over Union (IoU)</i>	26
4.2.3 <i>Elementary Food Leftover Estimation (EFLE)</i>	27
4.2.4 <i>RMSE</i> Bobot Estimasi	29
BAB 5 IMPLEMENTASI	30
5.1 Batasan Implementasi	30
5.2 Implementasi Sistem	30
5.2.1 <i>Power Law Transformation</i>	30
5.2.2 <i>Preprocessing</i> Citra	31
5.2.3 Segmentasi Citra Menggunakan Algoritme K-means	32
5.2.4 <i>Postprocessing</i> Hasil Segmentasi	33
5.2.5 Perhitungan Bobot Sisa Makanan	35
BAB 6 PENGUJIAN DAN PEMBAHASAN	37
6.1 Pengujian Hasil Cluster dan Segmentasi Citra	37
6.2 Pengujian Ruang Warna	37
6.3 Pengujian Hasil Segmentasi Citra	39
6.4 Pengujian Hasil Estimasi Bobot Sisa Makanan	41
BAB 7 PENUTUP	46
7.1 Kesimpulan	46
7.2 Saran	46
DAFTAR REFERENSI	47
LAMPIRAN A DATA CITRA MAKANAN PADA KOTAK MAKAN	49
A.1 Data Citra Makanan pada Kotak Makan	49

DAFTAR TABEL

Tabel 4.1 Nilai Citra Masukan.....	22
Tabel 4.2 Nilai Centroid Awal.....	23
Tabel 4.3 Perhitungan Jarak dan Penentuan <i>Cluster</i> pada Iterasi Pertama	23
Tabel 4.4 Pembaruan <i>Centroid</i> pada Iterasi Pertama.....	24
Tabel 4.5 Perhitungan Jarak dan Penentuan Cluster pada Iterasi Kedua.....	25
Tabel 4.6 Pembaruan Centroid pada Iterasi Kedua.....	25
Tabel 5.1 Implementasi Fungsi <i>Power Law</i>	30
Tabel 5.2 Implementasi Fungsi <i>Preprocessing</i>	31
Tabel 5.3 Implementasi Segmentasi dengan Algoritme K-Means	32
Tabel 5.4 Implementasi Fungsi <i>Postprocessing</i>	33
Tabel 6.1 Dataset Citra dengan Warna Kotak Makan Bervariasi.....	37
Tabel 6.2 Dataset Citra dengan Pencahayaan Bervariasi	38
Tabel 6.3 Hasil Segmentasi Menggunakan Channel U pada Ruang Warna YUV ...	38
Tabel 6.4 Akurasi Hasil Segmentasi Terhadap <i>Ground Truth</i>	39
Tabel 6.5 Akurasi Hasil Segmentasi Setiap Kompartemen dan Keseluruhan Citra	40
Tabel 6.6 Perbandingan Hasil Estimasi Bobot Sisa Makanan dengan Target	41
Tabel 6.7 <i>Error</i> Estimasi Bobot Sisa Makanan Terhadap Bobot Asli	42
Tabel 6.8 Perbandingan Hasil Estimasi Bobot Sisa Makanan dengan Target Hasil EFLE dari <i>Ground Truth</i>	43
Tabel 6.9 <i>Error</i> Estimasi Bobot Sisa Makanan Terhadap Bobot Hasil EFLE dari <i>Ground Truth</i>	44



DAFTAR GAMBAR

Gambar 2.1 Representasi citra pada ruang warna berbeda	7
Gambar 2.2 Representasi citra pada ruang warna YUV	8
Gambar 3.1 Strategi Penelitian	12
Gambar 3.2 Contoh Data Citra	13
Gambar 3.3 Diagram Alir Perancangan Sistem	14
Gambar 4.1 Diagram Alir Sistem	15
Gambar 4.2 Diagram Alir <i>Preprocessing</i>	16
Gambar 4.3 Diagram Alir Pembiasan Bayangan	17
Gambar 4.4 Diagram Alir <i>Power Law Transformation</i>	17
Gambar 4.5 Diagram Alir Segmentasi Citra Menggunakan Algoritme K-means	19
Gambar 4.6 Diagram Alir <i>Postprocessing</i>	20
Gambar 4.7 Diagram Alir EFLE	21
Gambar 4.8 Citra Awal	22
Gambar 4.9 Citra Setelah Diperkecil	22





DAFTAR LAMPIRAN

LAMPIRAN A DATA CITRA MAKANAN PADA KOTAK MAKAN.....	49
A.1 Data Citra Makanan pada Kotak Makan	49



Berdasarkan kebutuhan kontrol terhadap konsumsi makanan, maka dibutuhkan alat untuk mengetahui seberapa banyak makanan yang biasa dikonsumsi oleh seseorang. Dengan perkembangan teknologi, maka proses penghitungan bobot sisa makanan dapat dilakukan dengan memanfaatkan citra dari makanan sebelum dimakan dan citra makanan setelah dimakan. Sehingga proses untuk mengetahui jumlah makanan yang telah dikonsumsi dapat dilakukan lebih mudah serta dapat menyesuaikan dengan kondisi pandemi saat ini. Berdasarkan pada penelitian yang dilakukan oleh Sari *et al.*, (2019) yang mengembangkan sebuah alat bernama *Smart Nutrition Box*, alat ini dirancang

sebagai tahapan awal dalam mengembangkan alat untuk estimasi sisa makanan secara otomatis dengan metode segmentasi citra (*image segmentation*). Alat ini merupakan sebuah teknologi yang memanfaatkan sensor kamera untuk mengidentifikasi suatu objek dengan menggunakan metode segmentasi gambar (*image segmentation*).

Pada penelitian sebelumnya yang dilakukan oleh Sari *et al.*, (2019) dengan tujuan untuk melakukan segmentasi pada citra makanan. dimana dalam penelitian ini didapatkan hasil nilai RMSE sebesar 6.73 pada kotak makan berwarna gelap, dan nilai RMSE sebesar 6.85 pada kotak makan berwarna terang. Penelitian tersebut menggunakan metode *Otsu thresholding* untuk melakukan segmentasi pada citra, namun hasil segmentasi dari citra sangat bergantung pada histogram warna dari citra tersebut. Permasalahan terjadi pada segmentasi kompartemen kosong, dimana hasil segmentasi menunjukkan bahwa kompartemen kosong tidak tersegmentasi sebagai kompartemen kosong. Penelitian lain yang dilakukan oleh Sari, Maligan dan Prakoso, (2020) untuk melakukan segmentasi citra dengan menggunakan algoritme *clustering*. Dari hasil penelitian tersebut diperoleh hasil estimasi bobot sisa makanan lebih baik dari segmentasi menggunakan metode *thresholding*. Metode yang digunakan pada penelitian ini memperbaiki penelitian sebelumnya, yaitu masalah segmentasi pada kompartemen kosong. Penelitian lain terkait segmentasi citra dilakukan oleh Andika dan Hafiz (2018), objek pada penelitian ini adalah buah manga, dengan tujuan untuk membandingkan hasil segmentasi menggunakan algoritme K-means dan *Fuzzy C-means (FCM)*. Hasil dari penelitian tersebut menunjukkan hasil segmentasi algoritme K-means lebih baik dibandingkan *FCM*. Penelitian lain yang dilakukan oleh Yao *et al.*, (2013) untuk membandingkan hasil segmentasi menggunakan *Otsu*, *traditional K-means*, dan *improved K-means*. Hasil dari penelitian tersebut menunjukkan metode *clustering* dapat menghasilkan hasil segmentasi objek ikan lebih lengkap daripada metode *thresholding*.

Segmentasi citra bertujuan untuk memisahkan antara objek *foreground* (makanan) dengan *background* (kotak makan) dari citra makanan pada kotak makan (Sari, Maligan and Prakoso, 2020). Berdasarkan pemaparan diatas, maka metode *clustering* dengan algoritme K-means dipilih untuk segmentasi makanan pada citra kotak makan. Metode ini dipilih karena telah diketahui jumlah *cluster* dari segmen citra, yaitu *cluster* yang mewakili makanan dan *cluster* yang mewakili kotak makan. Pada dataset yang digunakan memiliki ciri berupa kotak makan yang memiliki warna seragam putih, abu-abu, atau hitam. Biru merupakan warna yang paling jarang ditemui pada hewan maupun tumbuhan (Kupferschmidt, 2019). Hal ini mengakibatkan jarang ditemui warna biru pada makanan tanpa pewarna makanan, sehingga warna biru dapat digunakan sebagai fitur untuk membedakan citra makanan dan kotak makan. Berdasarkan ciri yang terdapat pada dataset, diharapkan metode *clustering* dapat menghasilkan hasil segmentasi dengan tingkat kesalahan kecil dan memiliki akurasi yang baik untuk estimasi sisa makanan pada kotak makan.

1.2 Rumusan Masalah

Dari latar belakang masalah di atas, dirumuskan pokok permasalahan sebagai berikut:

1. Bagaimana tingkat akurasi dari algoritme K-means untuk melakukan segmentasi makanan?
2. Bagaimana tingkat akurasi estimasi sisa makanan berdasarkan hasil segmentasi tersebut?

1.3 Tujuan

Tujuan yang ingin dicapai oleh peneliti dalam penelitian ini antara lain sebagai berikut:

1. Menguji kualitas hasil segmentasi dari algoritme K-means.
2. Menguji tingkat akurasi hasil estimasi bobot makanan berdasarkan hasil segmentasi citra.

1.4 Manfaat

Manfaat yang dapat diperoleh dari penelitian sebagai berikut:

1. Dapat memberikan ruang warna terbaik untuk segmentasi citra makanan pada kotak makan berkompartemen.
2. Dapat memisahkan objek makanan dari citra makanan pada kotak makan berkompartemen.
3. Dapat membantu mengetahui bobot sisa makanan dalam bidang kesehatan.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini sebagai berikut:

1. Penggunaan data berupa data sekunder citra makanan pada kotak makan.
2. Metode segmentasi yang digunakan berbasis *clustering* dengan algoritme K-means.
3. *Cluster* dibagi menjadi 2, yaitu *cluster* kotak makan (*background*) dan *cluster* makanan (*foreground*).
4. Citra berupa makanan pada kotak makan yang memiliki beberapa kompartemen pemisah antar makanan.
5. Warna pada kotak makan seragam (putih, abu-abu, atau hitam).
6. Setiap makanan diletakkan pada kompartemen yang berbeda.
7. Citra merepresentasikan makanan dalam bentuk 2 dimensi.
8. Bobot makanan sebelum dikonsumsi telah diketahui sebelumnya.

1.6 Sistematika Pembahasan

BAB 1 PENDAHULUAN

Bab ini berisi tentang latar belakang, uraian rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan pada penelitian “Estimasi Sisa

Makanan Otomatis di Kotak Makan Menggunakan Segmentasi Citra Berbasis *Clustering* Dengan Algoritme K-Means”.

BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan berisi tentang teori-teori serta penelitian-penelitian sebelumnya yang akan menjadi acuan dalam menyelesaikan penelitian.

BAB 3 METODOLOGI PENELITIAN

Membahas tentang metode yang digunakan dalam penelitian tentang segmentasi makanan pada citra kotak makanan menggunakan algoritme K-means untuk estimasi bobot sisa makanan.

BAB 4 PERANCANGAN

Membahas tentang bagaimana merancang penelitian yang dilakukan berkaitan dengan sebuah komputasi yang akan digunakan dalam Estimasi Sisa Makanan Otomatis di Kotak Makan Menggunakan Segmentasi Citra Berbasis *Clustering* dengan Algoritme K-means.

BAB 5 IMPLEMENTASI

Melakukan implementasi sistem dalam suatu bahasa pemrograman sesuai dengan perancangan system yang telah dibuat sesuai diagram alir. Implementasi tersebut dijelaskan secara rinci pada setiap baris pemrograman.

BAB 6 PENGUJIAN DAN PEMBAHASAN

Melakukan pengujian dan analisis berdasarkan hasil pengujian sesuai dengan skenario pengujian yang ditetapkan pada segmentasi citra menggunakan algoritme K-means dan estimasi sisa bobot makanan.

BAB 7 PENUTUP

Memuat kesimpulan yang diperoleh dari penelitian tentang Estimasi Sisa Makanan Otomatis di Kotak Makan Menggunakan Segmentasi Citra Berbasis *Clustering* dengan Algoritme K-means yang dikembangkan dalam skripsi ini serta saran-saran untuk penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penerapan pengolahan citra digital semakin luas dalam berbagai bidang terutama pada bidang kesehatan salah satunya yakni bertujuan untuk membantu dalam mengetahui jumlah kandungan dari sisa atau makanan yang telah dikonsumsi seseorang. Banyak penelitian telah dilakukan untuk mendapatkan objek makanan dari citra makanan. Penelitian terkait yang menggunakan metode *clustering* untuk melakukan segmentasi citra, salah satunya adalah penelitian yang dilakukan oleh Dhanachandra, Mangleam dan Chanu (2015) dimana dalam penelitian ini bertujuan untuk melakukan segmentasi pada citra medis dengan menggunakan metode K-means *Clustering* dan *Subtractive Clustering* didapatkan hasil nilai RMSE rendah sehingga menghasilkan hasil segmentasi citra yang cukup baik.

Pada penelitian lain yang dilakukan oleh Deeparani dan Sudhakar (2021) untuk melakukan segmentasi citra menggunakan algoritme K-means dengan menambahkan proses *pre* dan *post processing* diperoleh hasil segmentasi yang lebih baik. Kemudian penelitian lain yang berkaitan dengan segmentasi menggunakan algoritme k-means dilakukan oleh Chen, Chen dan Chien (2008) untuk membuat segmentasi citra dengan performa yang cukup cepat menggunakan metode *clustering* pada ruang warna HSV. Penelitian ini menghasilkan rata-rata waktu eksekusi yang dibutuhkan untuk segmentasi citra berukuran 768 x 512 selama 0.29 detik.

Selanjutnya penelitian dari Yao *et al.* (2013) untuk melakukan segmentasi citra menggunakan algoritme K-means dengan optimasi inisialisasi nilai k dan *centroid* awal menghasilkan hasil segmentasi yang lebih stabil dan akurat dengan latar belakang gambar kompleks. Selain itu, penelitian yang dilakukan oleh Sari, Maligan dan Prakoso (2020) untuk melakukan segmentasi citra makanan pada kotak makan menggunakan algoritme K-means pada ruang warna HSV. Penelitian ini merupakan penelitian yang memiliki kesamaan objek yang diteliti yaitu makanan pada kotak makan. Dari penelitian-penelitian di atas, didapatkan bahwa dengan menggunakan metode *clustering* menghasilkan tingkat *error* yang kecil dan performa yang cukup baik. Untuk itu dengan metode *clustering* menggunakan algoritme K-means diharapkan dapat melakukan segmentasi makanan pada kotak makan.

Table 2.2.1. Kajian Pustaka

No.	Judul	Objek	Metode	Hasil
1	<i>Image Segmentation using K-means Clustering and Subtractive</i>	Segmentasi citra medis	K-means <i>Clustering</i> dan	Nilai RMSE rendah dan nilai PSNR di atas 30, menunjukkan

	<i>Clustering Algorithm</i> (Dhanachandra, Manglem and Chanu, 2015)		<i>Subtractive Clustering</i>	hasil segmentasi yang baik
2	<i>Fast Image Segmentation Based on K-Means Clustering with Histograms in HSV Color Space</i> (Chen, Chen and Chien, 2008)	Segmentasi citra	K-means Clustering	Rata-rata waktu eksekusi yang dibutuhkan untuk citra berukuran 768 x 512 adalah 0.29 detik
3	<i>Efficient image segmentation and implementation of K-means clustering</i> (Deeparani and Sudhakar, 2021)	Segmentasi citra	K-means Clustering	<i>Preprocessing</i> dan <i>postprocessing</i> dapat meningkatkan kualitas hasil segmentasi
4	<i>An improved K-means clustering algorithm for fish image segmentation</i> (Yao et al., 2013)	Segmentasi citra ikan	K-means Clustering	Menghasilkan segmentasi yang lebih stabil dan akurat dengan latar belakang gambar kompleks
5	<i>Improving the Elementary Leftover Food Estimation Algorithm by Using Clustering Image Segmentation in Nutrition Intake Problem</i> (Sari, Maligan and Prakoso, 2020)	Segmentasi citra makanan	K-means Clustering, dan EFLE	Menghasilkan prediksi yang baik tentang nutrisi sisa makanan dalam gambar kotak nampan tunggal dengan akurasi segmentasi sebesar 95.81%.

2.2 Sisa Makanan

Limbah makanan mengacu pada penurunan kuantitas makanan yang dihasilkan dari keputusan dan tindakan oleh pengecer, penyedia layanan makanan, dan konsumen. Makanan terbuang dengan berbagai cara (FAO, 2021):

- Produk segar yang menyimpang dari yang dianggap optimal, misalnya dalam hal bentuk, ukuran dan warna, seringkali dikeluarkan dari rantai pasok selama operasi penyortiran.

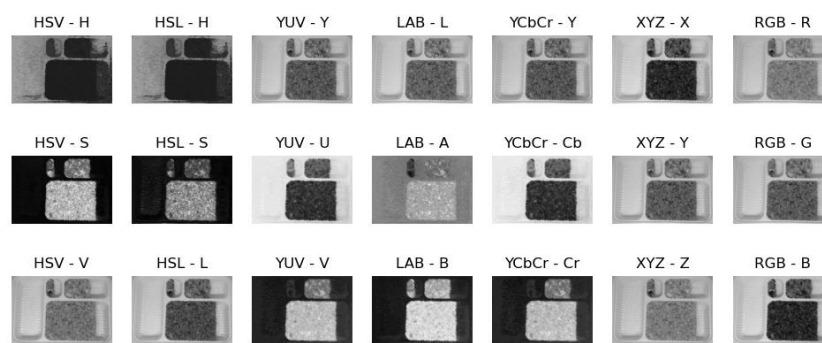
- Makanan yang mendekati, pada, atau melampaui tanggal kadaluarsa sering dibuang oleh pengecer dan konsumen.
- Sejumlah besar makanan sehat yang dapat dimakan seringkali tidak terpakai atau tersisa dan dibuang dari dapur rumah tangga dan tempat makan.

2.3 Estimasi Bobot

Estimasi digunakan untuk menerka sebuah nilai yang belum diketahui. Pada penelitian ini nilai yang akan di terka adalah bobot dari sisa makanan. Estimasi berbeda dengan prediksi, karena prediksi digunakan untuk memperkirakan nilai yang akan datang (belum ada sebelumnya)(Rerung, 2018). Berdasarkan hasil estimasi, dapat diketahui jumlah makanan yang telah dikonsumsi atau jumlah makanan yang terbuang.

2.4 Ruang Warna

Ruang warna, juga dikenal sebagai model warna atau sistem warna, membangun spesifikasi warna berdasarkan beberapa kriteria. Bentuk visual dari model warna adalah sistem koordinat di mana warna dinyatakan sebagai titik koordinat.



Gambar 2.1 Representasi citra pada ruang warna berbeda

2.4.1 Ruang Warna RGB

Pada model warna RGB, setiap warna terdiri dari proporsi tiga komponen warna primer yaitu merah, hijau, dan biru. Model ini digambarkan dalam koordinat kartesius tiga dimensi, di mana setiap warna primer bertindak sebagai sumbu. Setiap warna digambarkan sebagai vector (merah, hijau, dan biru). RGB adalah ruang warna yang paling umum digunakan untuk representasi gambar digital karena sesuai dengan tiga warna primer yang dicampur untuk tampilan pada monitor atau perangkat serupa (Vyas, Yu and Paik, 2018).

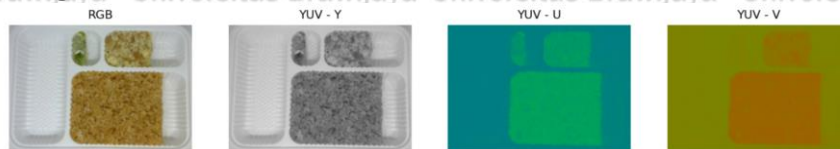
2.4.2 Ruang Warna HSV

Ruang warna perseptual adalah cara alternatif untuk merepresentasikan gambar warna sebenarnya dengan cara yang lebih alami bagi persepsi dan pemahaman manusia tentang warna daripada RGB. Representasi HSV dari gambar 2-D juga sebagai array 3-D yang terdiri dari tiga saluran h; s; v (Vyas, Yu and Paik, 2018).

Masing-masing dari ketiga parameter tersebut dapat diartikan sebagai berikut:

- H (hue) adalah panjang gelombang dominan warna, contoh: merah, biru, hijau.
- S (saturasi) adalah 'kemurnian' warna (dalam arti banyaknya cahaya putih yang bercampur dengannya).
- V (nilai) adalah kecerahan warna (juga dikenal sebagai *luminance*).

2.4.3 Ruang Warna YUV / YCbCr



Gambar 2.2 Representasi citra pada ruang warna YUV

Dalam ruang warna YUV, informasi hitam putih dipisahkan dari informasi warna, sedangkan Cb (U) dan Cr (V) adalah komponen *chrominance* perbedaan biru dan perbedaan merah (Podpora, Korbaś and Kawala-Janik, 2014). Pada penelitian ini *channel* U pada ruang warna YUV digunakan sebagai bahan untuk melakukan segmentasi citra, karena channel U mewakili tingkat warna hijau dan biru pada suatu citra. Warna biru merupakan warna yang jarang dijumpai pada makanan, oleh karena itu warna ini dipilih sebagai acuan dalam melakukan segmentasi.

2.5 Segmentasi Citra

Segmentasi citra bertujuan untuk memisahkan antara objek *foreground* (makanan) dengan *background* (kotak makan) (Sari *et al.*, 2019). Keluaran hasil segmentasi citra berupa citra biner dengan nilai 0 mewakili objek kotak makan dan nilai 1 mewakili objek makanan.

2.6 Clustering

Clustering merupakan sebuah metode untuk membentuk kelompok – kelompok berdasarkan kemiripan sebuah fitur (Bano and Khan, 2018). Di dalam segmentasi, *clustering* digunakan untuk membentuk daerah (*region*) dari kelompok – kelompok yang dihasilkan. Pada penelitian ini *clustering* digunakan untuk mengelompokkan citra berdasarkan kemiripan warna dari setiap piksel.

2.7 Preprocessing

Merupakan tahapan sebelum proses segmentasi dilakukan dengan tujuan untuk mengolah citra masukan agar didapatkan hasil maksimal dari proses segmentasi. Adapun proses yang dilakukan adalah sebagai berikut:

2.7.1 Power-Law Transformation

Peningkatan citra dapat dilakukan dalam domain spasial, salah satu parameter penting yang harus diperhatikan dalam konteks ini adalah peningkatan kontras.

Dalam domain spasial, metode yang digunakan dapat diklasifikasikan lebih lanjut menjadi: transformasi tingkat abu-abu, pemrosesan istogram dll (Vimal and Thiruvikraman, 2012). Transformasi *power-law* biasanya didefinisikan sebagai:

$$s = p(x,y)^g \quad 2.1$$

Keterangan:

s = hasil transformasi

p = piksel pada citra.

g = nilai *gamma*

Semakin tinggi nilai *gamma* akan menghasilkan citra yang cenderung lebih gelap, jika nilai *gamma* kurang dari 1, maka citra yang dihasilkan akan cenderung lebih terang.

2.8 K-Means Clustering

K-means adalah algoritme iteratif yang digunakan untuk mempartisi dataset menjadi subkelompok berbeda yang ditentukan sejumlah k (*cluster*) di mana setiap titik data hanya dimiliki oleh satu kelompok atau *cluster*. (Dabbura, 2018.)

Pada suatu citra dengan resolusi $x \times y$, citra tersebut dikelompokkan ke dalam k kelompok. Misalkan $p(x, y)$ adalah piksel *input* yang akan dikelompokkan dan c adalah pusat *cluster* (*centroid*). Langkah-langkah pada metode k-means adalah sebagai berikut (Dhanachandra, Manglem and Chanu, 2015):

1. Inisialisasi jumlah *cluster* dan *centroid*.

$$c_{1i} = \min(p(x, y, i)) \quad 2.3$$

$$c_{2i} = \max(p(x, y, i))$$

Keterangan:

k = 2, untuk membagi citra menjadi dua *cluster* yaitu citra makanan dan non-makanan.

p = piksel pada citra.

i = indeks ruang warna (h, s, v)

2. Untuk setiap piksel pada citra, hitung jarak Euclidian d , antara nilai piksel dengan nilai *centroid* dengan menggunakan rumus dibawah ini.

$$d_k = \| p(x, y) - c_k \| \quad 2.4$$

Keterangan:

k = *cluster*.

p = piksel pada citra.

c_k = *centroid*

d_k = jarak *euclidian* piksel dengan *centroid* c_k

3. Tentukan *cluster* setiap piksel berdasarkan jarak terdekat d dengan *centroid*.

$$c_{xy} = \min(\sum_1^k d_k) \quad 2.5$$

Keterangan:

c_{xy} = *cluster* pada $p(x, y)$

d_k = jarak *euclidian* piksel dengan *centroid* c_k

- Setelah *cluster* setiap piksel telah ditentukan, hitung kembali posisi baru *centroid* dengan menggunakan rumus dibawah ini.

$$c_k = \frac{1}{k} \sum_{y \in c_k} \sum_{x \in c_k} p(x, y) \quad 2.6$$

- Ulangi langkah 2 sampai 4 hingga kondisi berhenti tercapai.

2.8.1 Kondisi Berhenti

Kondisi berhenti merupakan syarat yang telah ditetapkan diawal proses clustering. Pada algoritme K-Means tahapan penghitungan jarak antar setiap piksel terhadap masing-masing *centroid* akan terus dilakukan berulang - ulang dan dapat berhenti apabila kondisi berhenti telah terpenuhi. Terdapat tiga kondisi yang dapat digunakan untuk menghentikan algoritme K-Means:

- Tidak terdapat perubahan *centroid*.
- Setiap data tetap berada pada *cluster* yang sama.
- Jumlah iterasi maksimum telah tercapai.

Pada penelitian ini digunakan kondisi 1 dan 3 sebagai kondisi berhenti pada algoritme K-Means.

2.9 Postprocessing

Merupakan tahapan setelah proses segmentasi dilakukan dengan tujuan untuk meningkatkan kualitas citra hasil segmentasi agar didapatkan hasil estimasi sisa bobot makanan yang lebih optimal.

2.9.1 Flood Fill

Flood fill adalah algoritme yang terutama digunakan untuk menentukan *area* terbatas yang terhubung ke titik tertentu dalam *array* multi-dimensi. Algoritme ini digunakan untuk mendapatkan piksel dari *background* yang saling terhubung (Muthukumar, Poorani and Sindhu, 2016).

2.9.2 Morfologi - Erosi

Merupakan operasi pada citra digital yang berfungsi untuk mengubah ukuran objek pada citra menjadi lebih kecil. Hal ini bertujuan untuk menghilangkan *noise* berupa titik-titik kecil pada citra sesuai dengan ukuran *kernel* yang ditentukan (Raid et al., 2014).

2.9.3 Morfologi - Dilatasi

Merupakan operasi pada citra digital yang berfungsi untuk mengubah ukuran objek pada citra menjadi lebih besar dan menyatukan objek berdekatan yang tidak saling terhubung (Said and Jambek, 2017).

2.10 Perhitungan Bobot Sisa Makanan

Bobot sisa makanan diperoleh melalui jumlah piksel makanan pada citra makanan setelah dimakan kemudian dibagi dengan jumlah piksel makanan pada citra makanan sebelum dimakan lalu dikalikan dengan bobot dari makanan sebelum dimakan. Persamaan yang digunakan untuk menghitung bobot sisa makanan adalah sebagai berikut (Sari, Maligan and Prakoso, 2020):

$$EFLE = Af \times \frac{OBf}{Bf} \quad 2.7$$

Keterangan:

$EFLE$ = Estimasi bobot sisa makanan (dalam gram).

Af = Jumlah piksel makanan pada citra setelah dimakan.

Bf = Jumlah piksel makanan pada citra sebelum dimakan.

OBf = Bobot makanan sebelum dimakan (dalam gram).

2.11 Evaluasi

Evaluasi dalam pengujian ini difungsikan untuk mencari keselarasan antara hasil segmentasi dan bobot sisa makanan yang telah diprediksi dengan nilai target yang sebenarnya.

2.11.1 Intersection over Union (IoU)

Merupakan metode untuk mengukur tingkat akurasi dari hasil segmentasi (prediksi) terhadap *ground truth* (target). Akurasi akan didapatkan melalui kesamaan nilai piksel hasil tumpang tindih antara citra target dan prediksi (*intersection*), kemudian dibagi dengan gabungan antara piksel target dan prediksi (*union*) (Rezatofighi *et al.*, 2019). Berikut rumus akurasi menggunakan *IoU*:

$$\text{Akurasi } IoU = \frac{\text{Citra}_{\text{target}} \cap \text{Citra}_{\text{prediksi}}}{\text{Citra}_{\text{target}} \cup \text{Citra}_{\text{prediksi}}} \times 100\% \quad 2.8$$

2.11.2 Akurasi Bobot Sisa Makanan

Perhitungan presentase akurasi estimasi bobot sisa makanan akan didapatkan dari hasil pengurangan antara estimasi yang diperoleh dengan nilai asli. Berikut adalah rumus akurasi (Sari, Maligan and Prakoso, 2020):

$$\text{Akurasi} = \frac{|W_{\text{sebenarnya}} - W_{\text{prediksi}}|}{|W_{\text{sebenarnya}}|} \times 100\% \quad 2.9$$

2.11.3 Root Mean Square Error (RMSE)

Untuk memperkuat hasil validitas, digunakan RMSE untuk setiap kompartemen pengukuran berat aktual versus *EFLE* (Sari, Maligan and Prakoso, 2020). Persamaan untuk mendapatkan RMSE dinyatakan dalam persamaan 2.10.

$$RMSE = \sqrt{\sum_k^n (W_k - EFLE_k)^2 / n} \quad 2.10$$

Dimana W merupakan berat aktual, dan $EFLE$ merupakan bobot estimasi, k merupakan kompartemen, dan n merupakan jumlah kompartemen.

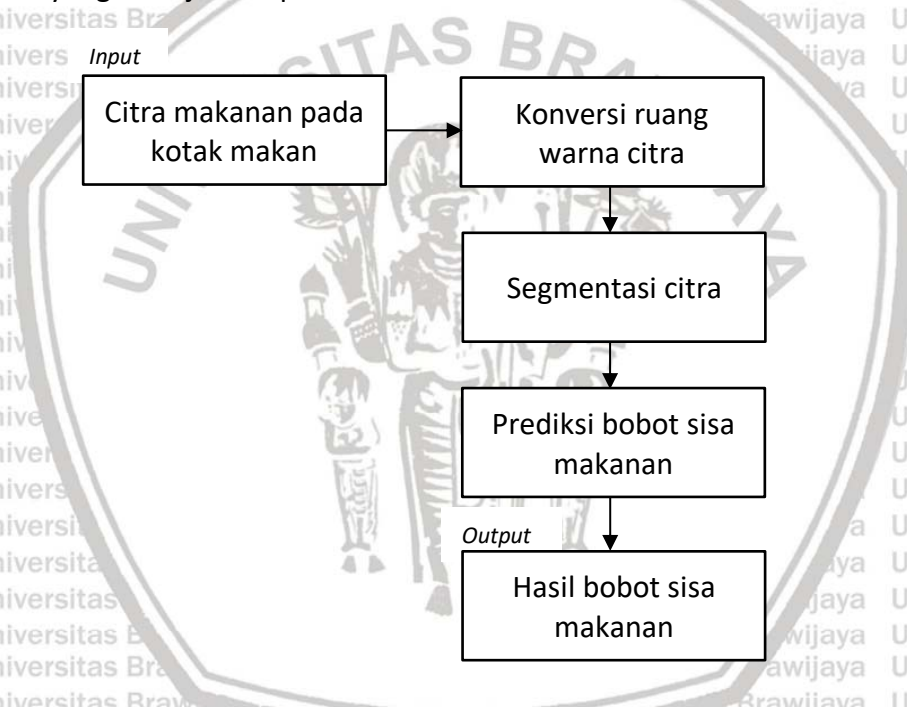
BAB 3 METODOLOGI PENELITIAN

3.1 Tipe Penelitian

Tipe penelitian yang digunakan dalam penelitian ini adalah non-implementatif analitik dengan menerapkan metode yang sudah ada sebelumnya untuk menyelesaikan masalah.

3.2 Strategi Penelitian

Strategi penelitian menggambarkan proses penelitian secara umum, dimulai dari proses memasukkan data, kemudian ke tahap *preprocessing* data, selanjutnya proses segmentasi menggunakan algoritme K-means. Setelah itu dilakukan *postprocessing* terhadap hasil segmentasi. Kemudian data hasil segmentasi digunakan untuk menghitung dan menampilkan keluaran bobot sisa makanan, yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Strategi Penelitian

3.3 Metode Pengumpulan Data

Data yang akan digunakan pada penelitian ini didapat dari penelitian sebelumnya yang dilakukan oleh (Sari, Maligan and Prakoso, 2020). Berikut adalah gambar contoh dari dataset yang akan digunakan dalam penelitian ini.



Gambar 3.2 Contoh Data Citra

3.4 Peralatan Pendukung

3.4.1 Perangkat Keras

Dalam pengembangan dan pengujian, perangkat keras yang digunakan memiliki spesifikasi sebagai berikut,

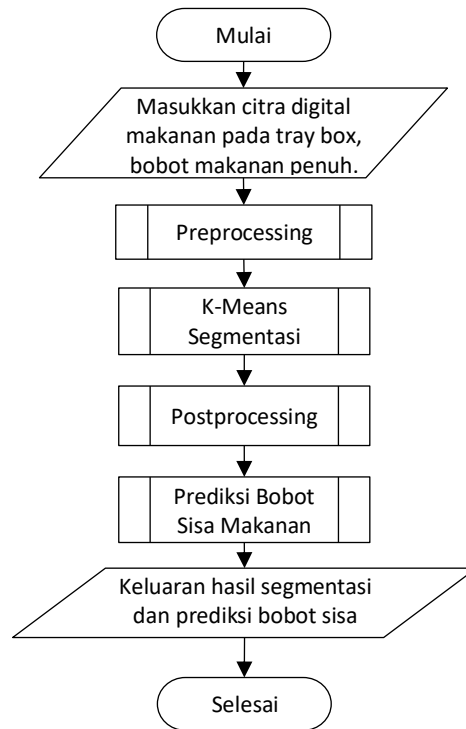
- *Prosesor* : AMD Ryzen 5 3500U APU@ 2.10 GHz
- *GPU* : AMD Vega 8
- *Memory* : 12 GB DDR 4
- *Storage* : SSD berkapasitas 500 GB

3.4.2 Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam proses pengembangan dan pengujian memiliki spesifikasi sebagai berikut,

- Sistem Operasi Windows 10.
- Visual Studio Code.
- Menggunakan bahasa pemrograman Python 3.0 yang memuat *library* OpenCV, Numpy, Pandas, OS.
- Microsoft Office 2019 sebagai media pengerjaan laporan penelitian.

3.5 Metode Perancangan Algoritme



Gambar 3.3 Diagram Alir Perancangan Sistem

3.6 Pengujian

Pengujian dilakukan untuk mendapatkan hasil akurasi dan analisis dari proses implementasi algoritme K-means untuk segmentasi makanan pada kotak makan, dan hasil estimasi bobot sisa makanan yang dihasilkan. Rancangan pengujian yang akan dilakukan adalah sebagai berikut:

1. Pengujian untuk mendapatkan ruang warna terbaik.
2. Pengujian untuk hasil segmentasi.
3. Pengujian untuk hasil estimasi bobot sisa makanan.

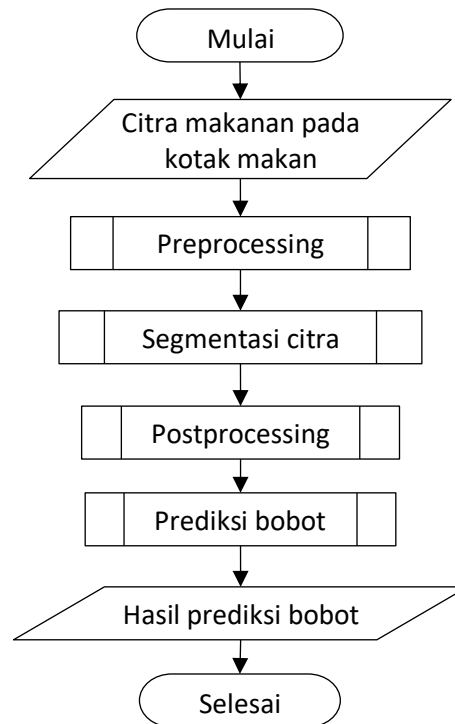
3.7 Penarikan Kesimpulan dan Saran

Penarikan kesimpulan disimpulkan berdasarkan hasil pengujian yang telah dilakukan dan analisis implementasi algoritme K-means untuk segmentasi citra makanan pada kotak makan serta estimasi bobot sisa makanan. Kesimpulan dijelaskan secara ringkas sesuai dengan jawaban atas rumusan masalah yang ada. Pemberian saran bertujuan untuk memperbaiki kekurangan yang ada pada penelitian yang sedang dilakukan dan dapat dikembangkan untuk penelitian selanjutnya.

BAB 4 PERANCANGAN

4.1 Perancangan Algoritme

Perancangan algoritme diimplementasikan melalui diagram alir. Diagram alir yang mengacu pada Gambar 4.1 bertujuan untuk memperlihatkan alur segmentasi citra dari proses awal yaitu inialisasi parameter hingga pada proses terakhir yaitu menyajikan hasil estimasi sisa bobot makanan.



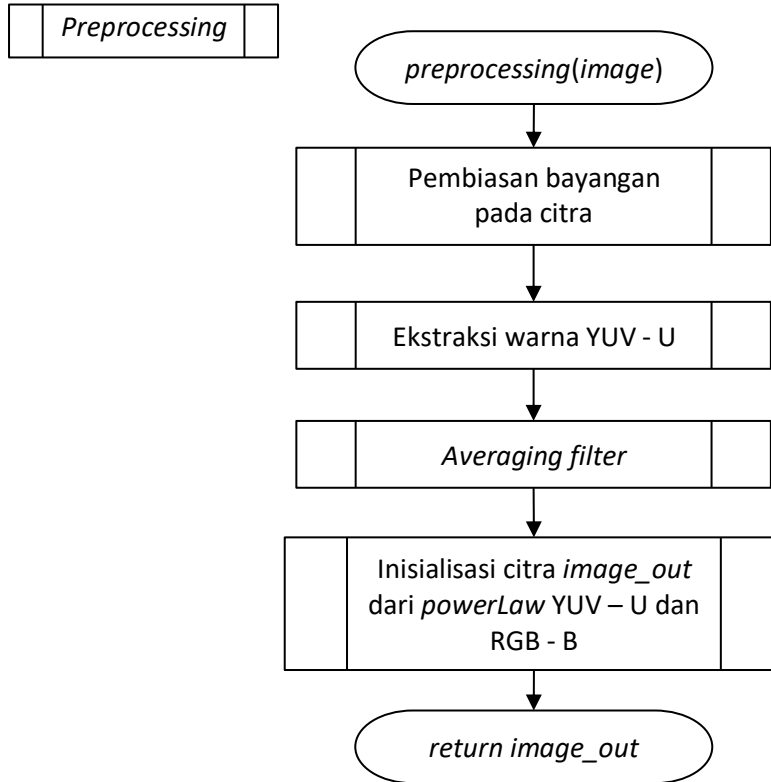
Gambar 4.1 Diagram Alir Sistem

Berdasarkan Gambar 4.1 alur algoritme dimulai dengan memasukkan citra makanan pada kotak makan sebagai data masukan yang akan diolah. Berikutnya adalah tahap *preproceasing*, yaitu tahap mengolah citra sebelum dilakukan proses segmentasi. Selanjutnya citra hasil *preprocessing* digunakan sebagai data masukan untuk proses segmentasi citra menggunakan algoritme K-means.

Kemudian dilakukan proses *postprocessing* terhadap hasil segmentasi untuk meningkatkan kualitas hasil segmentasi dengan menghilangkan noise pada hasil segmentasi menggunakan morfologi dan *flood filling*. Citra hasil segmentasi selanjutnya digunakan untuk menghitung bobot sisa makanan menggunakan metode *EFLE*. Melalui segmentasi dan estimasi bobot sisa makanan tersebut diharapkan dapat diketahui bobot sisa makanan melalui citra makanan pada kotak makan dengan hasil yang baik.

4.1.1 Preprocessing Citra

Proses *preprocessing* dilakukan untuk mendapatkan segmen warna pada citra yang akan digunakan untuk proses segmentasi.

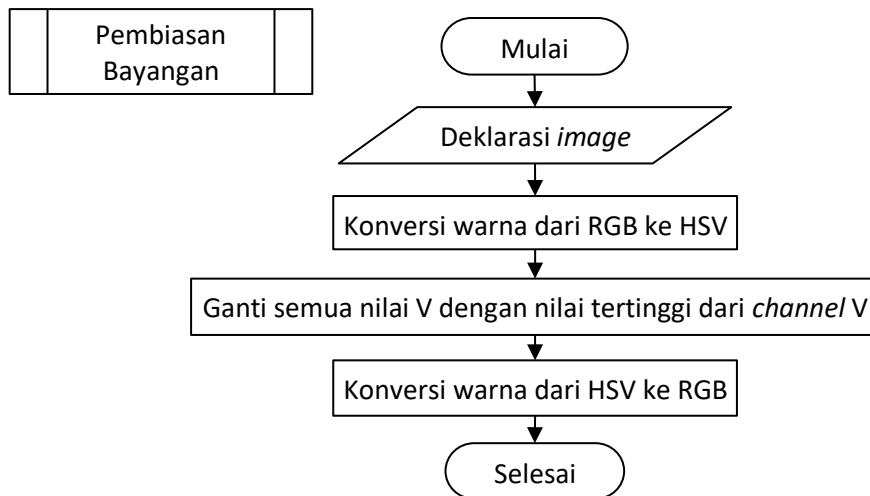


Gambar 4.2 Diagram Alir Preprocessing

Berikut ini adalah langkah-langkah *preprocessing* sesuai diagram alir pada Gambar 4.2:

1. Data masukan berupa citra makanan pada kotak makan dengan ruang warna RGB.
2. Proses mengurangi bayangan atau *shadow* pada citra masukan.
3. Mengubah ruang warna citra menjadi YUV, kemudian diambil *channel* warna U.
4. *Averaging filter* terhadap citra pada ruang warna YUV.
5. Inisialisasi citra keluaran dengan melakukan *power law transformation* terhadap *channel* U pada citra sebelum dan setelah dilakukan *averaging filter*.
6. Menghasilkan citra yang berasal dari *channel* U pada ruang warna YUV dan *channel* B dari ruang warna RGB.

4.1.1.2 Mengurangi Bayangan pada Citra

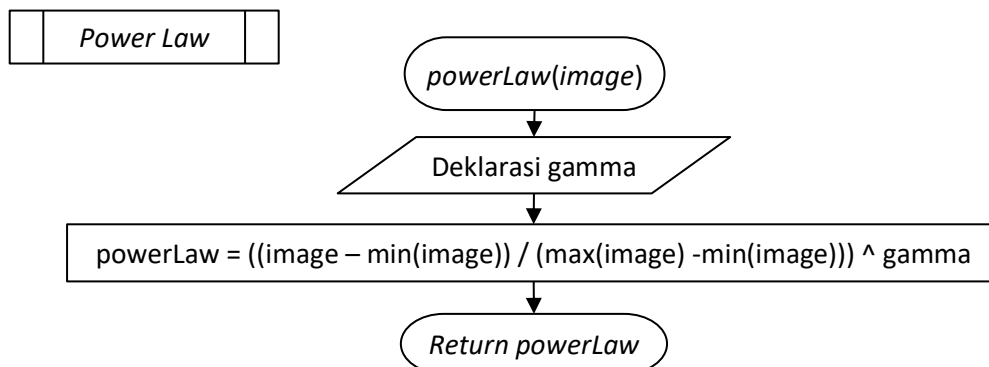


Gambar 4.3 Diagram Alir Pembiasan Bayangan

Berikut ini adalah langkah-langkah pembiasan bayangan sesuai diagram alir pada Gambar 4.3:

1. Data masukan berupa citra dengan ruang warna RGB.
2. Konversi ruang warna citra dari RGB ke HSV menggunakan library OpenCV.
3. Mengganti semua nilai *channel V* dengan nilai tertinggi *V* ruang warna HSV.
4. Kembalian berupa citra yang dikonversi dari ruang warna HSV menjadi RGB menggunakan OpenCV.

4.1.1.3 Power Law Transformation



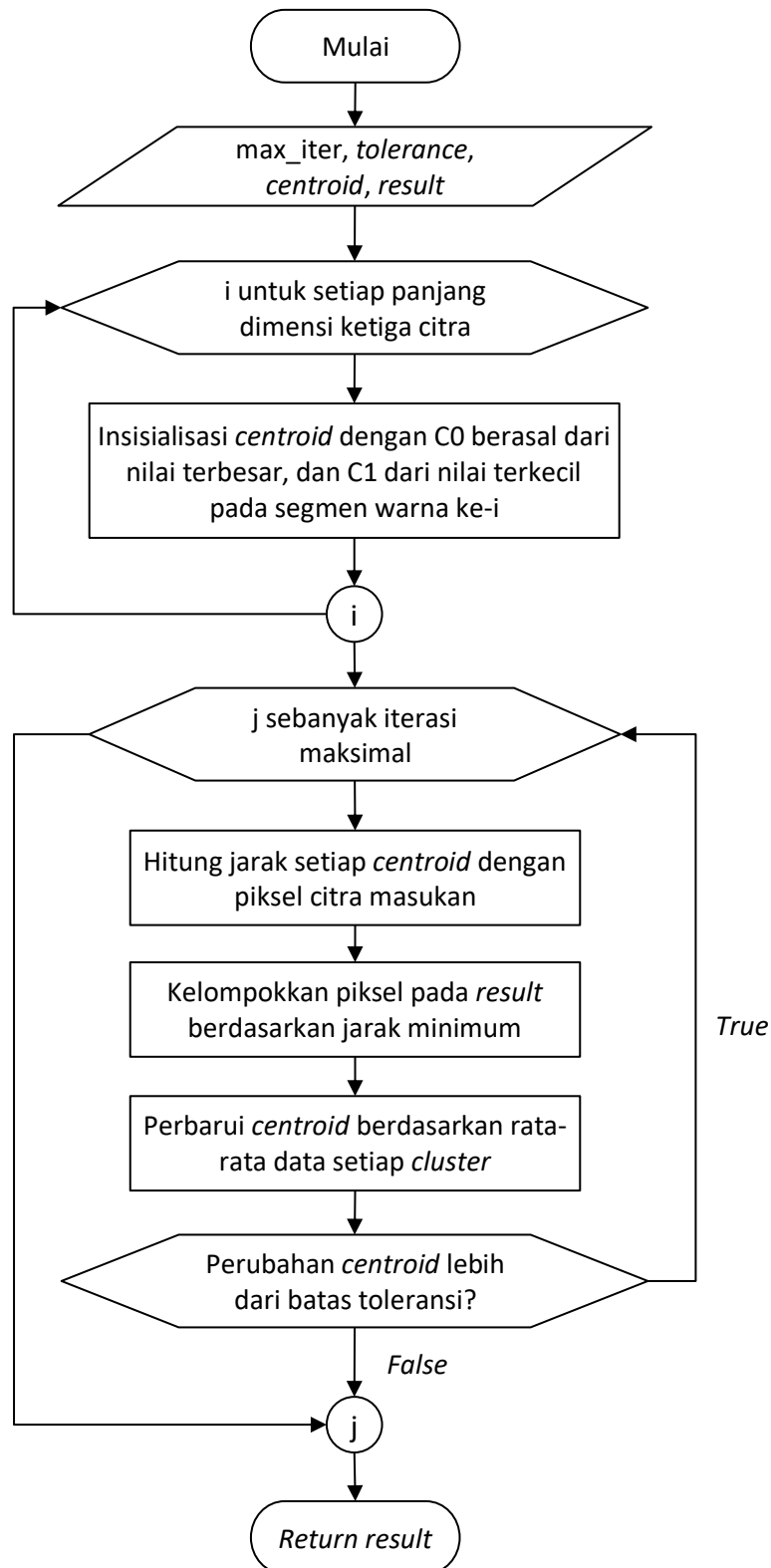
Gambar 4.4 Diagram Alir Power Law Transformation

Berikut ini adalah langkah-langkah *power law transformation* sesuai diagram alir pada Gambar 4.4:

1. Data masukan berupa array 2 dimensi dari channel U pada ruang warna YUV, dan gamma.
2. Proses *power law transformation*.
3. Kembalian berupa array 2 dimensi hasil *power law transformation*.

4.1.2 Segmentasi K-means

Segmentasi Citra

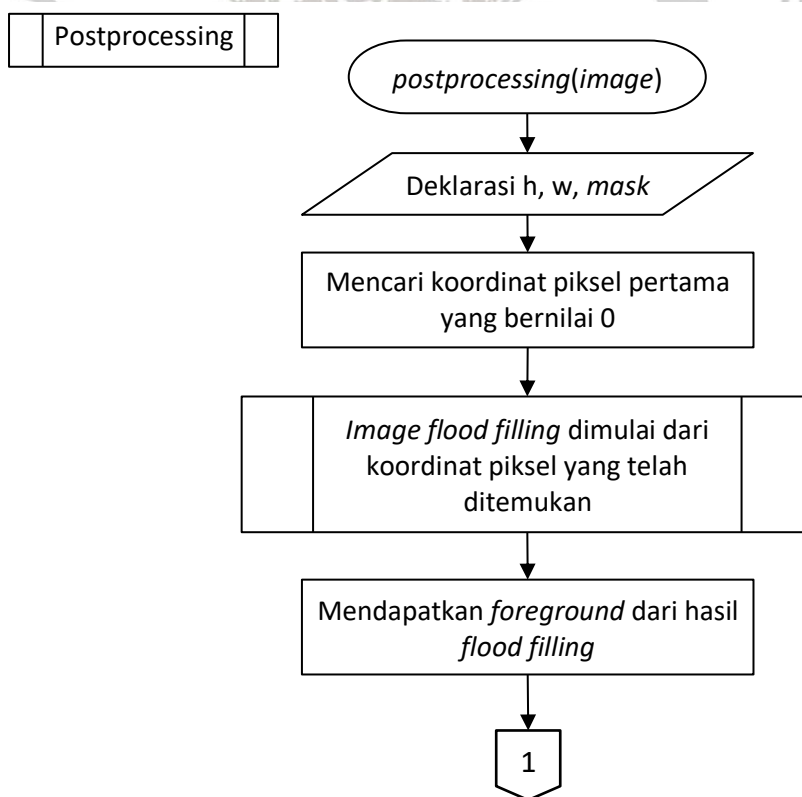


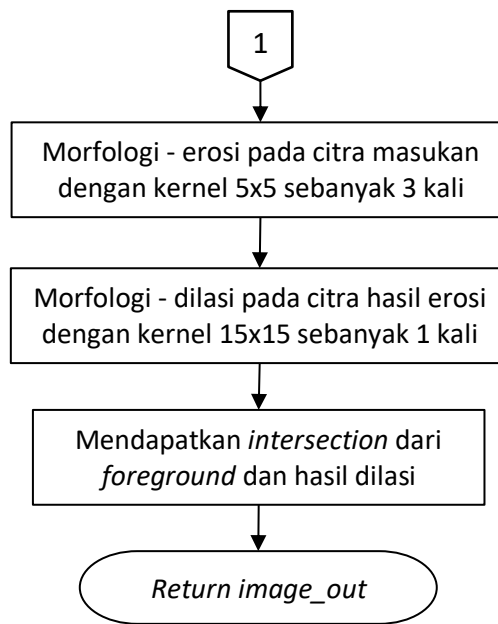
Gambar 4.5 Diagram Alir Segmentasi Citra Menggunakan Algoritme K-means

Berikut ini adalah langkah-langkah segmentasi citra menggunakan algoritme K-means sesuai diagram alir pada Gambar 4.5:

1. Data masukan berupa iterasi maksimal, toleransi perubahan centroid, dan *result* untuk menyimpan hasil *clustering*.
2. Perulangan sebanyak dimensi ketiga citra (sejumlah channel warna pada citra masukan) untuk inialisasi *centroid* dengan nilai terbesar pada setiap channel warna pada citra sebagai C0 (*cluster* untuk *background*) dan nilai terkecil sebagai C1 (*cluster* untuk *foreground*).
3. Melakukan perulangan sebanyak maksimal iterasi untuk menghitung jarak setiap *centroid* dengan citra masukan menggunakan *euclidean distance*. Kemudian piksel citra dikelompokkan sesuai jarak terdekat antara setiap piksel pada citra terhadap C0 dan C1. Selanjutnya perbarui *centroid* dengan nilai rata-rata piksel setiap kategori pada *cluster*. Apabila perubahan nilai *centroid* dibawah batas toleransi atau telah dicapai jumlah iterasi maksimal, maka perulangan dihentikan.
4. Kembalian berupa citra biner (0 mewakili *background*, dan 1 mewakili *foreground*).

4.1.3 Postprocessing Hasil Segmentasi



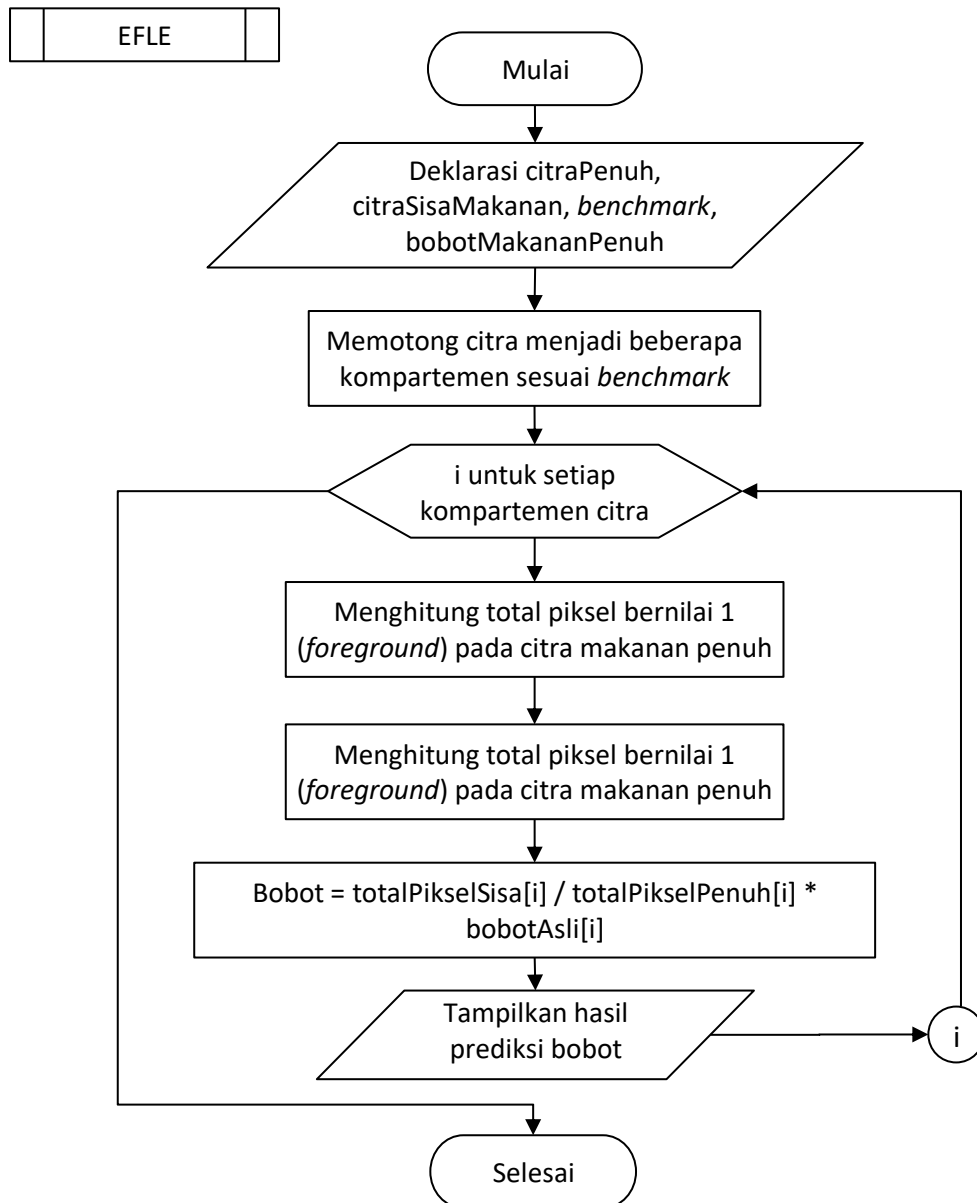


Gambar 4.6 Diagram Alir *Postprocessing*

Berikut ini adalah langkah-langkah *postprocessing* sesuai diagram alir pada Gambar 4.6:

1. Data masukan berupa citra biner hasil segmentasi.
2. Inisialisasi *h* (*height*) dan *w* (*weight*) dari citra masukan, dan *mask* yang akan digunakan untuk proses *flood filling* dengan ukuran $h+2$ dan $w+2$ (tambahan border 1 piksel pada setiap sisi citra).
3. Mencari koordinat piksel dengan nilai 0 (*background*) pertama pada citra masukan, kemudian dilakukan proses *flood filling* menggunakan *library* OpenCV untuk mendapatkan semua koordinat dari piksel yang terhubung oleh piksel awal (untuk mendapatkan *background* dari citra).
4. Mendapatkan *foreground* dari hasil *flood filling* dengan cara melakukan *inverse* hasil *flood filling*.
5. Melakukan operasi morfologi - erosi pada citra hasil segmentasi dengan kernel 5 x 5 sebanyak 3 kali untuk menghilangkan *noise* berupa titik-titik kecil pada citra.
6. Melakukan operasi morfologi - dilasi pada citra hasil erosi dengan kernel 15 x 15 sebanyak 1 kali untuk mengembalikan ukuran objek yang mengecil akibat operasi erosi.
7. Melakukan *intersection* dari hasil *foreground* dan hasil dilasi untuk memperoleh hasil citra dengan *noise* yang telah dihilangkan dan dengan tetap mempertahankan tepi citra.
8. Kembalian berupa citra biner.

4.1.4 Elementary Leftover Food Estimation (EFLE)



Gambar 4.7 Diagram Alir EFLE

Berikut ini adalah langkah-langkah menghitung bobot sisa makanan sesuai diagram alir pada Gambar 4.7:

1. Data masukan berupa citra biner citra makanan sebelum dimakan (*full*) dan citra makanan setelah dimakan (sisa) hasil segmentasi yang telah melalui proses *postprocessing*.
2. Memotong citra menjadi beberapa bagian, sesuai dengan *benchmark* yang telah dibuat sebelumnya.
3. Perulangan sebanyak potongan citra, kemudian menghitung total piksel *foreground* pada citra makanan sebelum dimakan dan menghitung total piksel

foreground pada citra makanan sesudah dimakan. Selanjutnya dihitung bobot dari sisa makanan.

4. Keluaran yang dihasilkan berupa bobot sisa makanan dalam satuan gram.

4.2 Perhitungan Manual

4.2.1 Segmentasi K-means

Berisi perhitungan manual sebagai contoh manualisasi sesuai metode yang dilakukan, yaitu segmentasi citra makanan pada kotak makan menggunakan algoritme K-means. Citra makanan pada kotak makan berikut merupakan contoh potongan citra berukuran 40×70 dari hasil *preprocessing*. Citra terdiri dari *channel* warna U pada ruang warna YUV yang telah melalui proses *preprocessing* pada Gambar 4.8.



Gambar 4.8 Citra Awal

Kemudian citra awal diperkecil menjadi berukuran 4×7 piksel, citra setelah diperkecil ditunjukkan pada Gambar 4.9.



Gambar 4.9 Citra Setelah Diperkecil

Representasi *value* dari masing-masing piksel pada setiap *channel* citra pada Gambar 4.9 ditunjukkan oleh Tabel 4.1.

Tabel 4.1 Nilai Citra Masukan

Citra input: YUV - Channel U			
No	Metode 1	Metode 2	Metode 3
1	3.72106072	79.057857	2.66501184
2	0.01143984	75.6846815	0.86441254
3	26.9578787	109.595793	11.6417957
4	41.5529697	124.658909	20.5681736
5	42.0927901	126.27837	21.7606264
6	14.5878589	102.41315	8.74627829
7	2.62721895	95.0124922	3.97877723
8	33.9144424	117.696122	18.7286739

9	15.9391191	98.1462201	8.88289939
10	34.6473813	115.905852	14.9307073
11	41.1236031	123.370809	19.1639886
12	42.4844623	127.453387	22.8842083
13	36.165856	120.909139	19.711874
14	11.984484	98.5435903	7.4738755
15	42.3033095	127.770415	23.8368975
16	29.7118125	113.935043	16.082852
17	26.4545257	109.862197	13.706948
18	22.7826431	104.834978	12.0157372
19	27.052115	110.499803	13.9932364
20	30.8897656	113.259698	16.0010689
21	29.9200527	112.782775	15.2537428
22	12.9804077	94.1559505	6.93363447
23	0.00249836	73.9208419	0.56066875
24	0	80.1906645	1.27775892
25	0	79.5287315	1.09659008
26	0	76.0282713	0.7985497
27	0	73.1226828	0.50996555
28	0	73.6900727	0.52458655

Kemudian inisialisasi *centroid* dengan menggunakan *maximum value* untuk C0 dan *minimum value* untuk C1 pada masing-masing channel warna *input*. Hasil *centroid* awal direpresentasikan oleh Tabel 4.2.

Tabel 4.2 Nilai Centroid Awal

C0	42.4844623	127.770415	23.8368975
C1	0	73.1226828	0.50996555

1. Iterasi Pertama

Menghitung jarak antara setiap piksel dengan *centroid* menggunakan *euclidian distance*, D0 merepresentasikan jarak setiap piksel dengan C0, dan D1 merepresentasikan jarak setiap piksel dengan C1. Kemudian pemilihan *cluster* berdasarkan jarak terdekat antara C0 dan C1 terhadap setiap piksel citra masukan. Cluster 0 mewakili C0 (*background*) dan cluster 1 mewakili C1 (*foreground*). Hasil perhitungan jarak dan penentuan *cluster* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Perhitungan Jarak dan Penentuan Cluster pada Iterasi Pertama

No	Nilai Piksel			D0	D1	Cluster
	Metode 1	Metode 2	Metode 3			
1	3.72106072	79.057857	2.66501184	65.75533	7.329175	1
2	0.01143984	75.6846815	0.86441254	71.02546	2.586426	1
3	26.9578787	109.595793	11.6417957	26.83491	46.70046	0
4	41.5529697	124.658909	20.5681736	4.608004	69.17343	0
5	42.0927901	126.27837	21.7606264	2.586601	71.05576	0

6	14.5878589	102.41315	8.74627829	40.60712	33.74276	1
7	2.62721895	95.0124922	3.97877723	55.28134	22.31812	1
8	33.9144424	117.696122	18.7286739	14.17852	58.89739	0
9	15.9391191	98.1462201	8.88289939	42.49553	30.82757	1
10	34.6473813	115.905852	14.9307073	16.7782	56.91044	0
11	41.1236031	123.370809	19.1639886	6.560834	67.55736	0
12	42.4844623	127.453387	22.8842083	1.004054	72.50767	0
13	36.165856	120.909139	19.711874	10.19891	62.93034	0
14	11.984484	98.5435903	7.4738755	45.30126	28.95421	1
15	42.3033095	127.770415	23.8368975	0.181153	72.93895	0
16	29.7118125	113.935043	16.082852	20.36377	52.82949	0
17	26.4545257	109.862197	13.706948	26.08216	47.15712	0
18	22.7826431	104.834978	12.0157372	32.46438	40.70751	0
19	27.052115	110.499803	13.9932364	25.16603	48.06937	0
20	30.8897656	113.259698	16.0010689	20.15932	52.9635	0
21	29.9200527	112.782775	15.2537428	21.358	51.82192	0
22	12.9804077	94.1559505	6.93363447	47.81361	25.53728	1
23	0.00249836	73.9208419	0.56066875	72.43119	0.799772	1
24	0	80.1906645	1.27775892	67.65853	7.109562	1
25	0	79.5287315	1.09659008	68.18586	6.432852	1
26	0	76.0282713	0.7985497	70.80215	2.919884	1
27	0	73.1226828	0.50996555	73.04416	0	1
28	0	73.6900727	0.52458655	72.61595	0.567578	1

Pada iterasi pertama, setelah memperoleh hasil *cluster*. Selanjutnya dilakukan *update centroid* dengan nilai rata-rata data piksel pada setiap *cluster*. Setelah nilai *centroid* baru diperoleh, kemudian dihitung perubahan *centroid* lama dan baru melalui jumlah selisih antara *centroid* baru dan lama yang direpresentasikan oleh Tabel 4.4. Karena nilai jumlah selisih antara *centroid* baru dan lama diatas batas toleransi (0,001) maka iterasi dilanjutkan.

Tabel 4.4 Pembaruan Centroid pada Iterasi Pertama

Centroid Lama			
C1	0	73.1226828	0.50996555
C0	42.4844623	127.770415	23.8368975
Centroid Baru			
C1	4.758007	84.57655	3.408693
C0	33.87024	117.2542	17.35204
Selisih Centroid Baru dan Lama			
C1	4.758007	11.45387	2.898727
C0	8.614222	10.5162	6.484862
Total Selisih Centroid			44.72589

2. Iterasi Kedua

Menghitung jarak antara setiap piksel dengan *centroid* baru menggunakan *Euclidian distance*, D0 merepresentasikan jarak setiap piksel dengan C0, dan D1 merepresentasikan jarak setiap piksel dengan C1. Kemudian pemilihan *cluster* berdasarkan jarak terdekat antara C0 dan C1 terhadap setiap piksel citra masukan. Cluster 0 mewakili C0 (*background*) dan *cluster* 1 mewakili C1 (*foreground*). Hasil perhitungan jarak dan penentuan *cluster* ditunjukkan pada Tabel 4.5.

Tabel 4.5 Perhitungan Jarak dan Penentuan Cluster pada Iterasi Kedua

No	Nilai Piksel			D0	D1	Cluster
	Metode 1	Metode 2	Metode 3			
1	3.72106072	79.057857	2.66501184	50.82956	5.664304	1
2	0.01143984	75.6846815	0.86441254	56.09177	10.39561	1
3	26.9578787	109.595793	11.6417957	11.79148	34.44678	0
4	41.5529697	124.658909	20.5681736	11.14438	57.05184	0
5	42.0927901	126.27837	21.7606264	12.98003	58.90434	0
6	14.5878589	102.41315	8.74627829	25.80943	21.05374	1
7	2.62721895	95.0124922	3.97877723	40.61607	10.66649	1
8	33.9144424	117.696122	18.7286739	1.446501	46.70873	0
9	15.9391191	98.1462201	8.88289939	27.53846	18.41521	1
10	34.6473813	115.905852	14.9307073	2.878345	44.80688	0
11	41.1236031	123.370809	19.1639886	9.659561	55.45881	0
12	42.4844623	127.453387	22.8842083	14.45105	60.34073	0
13	36.165856	120.909139	19.711874	4.919057	50.71788	0
14	11.984484	98.5435903	7.4738755	30.44096	16.24271	1
15	42.3033095	127.770415	23.8368975	14.95862	60.76735	0
16	29.7118125	113.935043	16.082852	5.469943	40.56165	0
17	26.4545257	109.862197	13.706948	11.087	34.87344	0
18	22.7826431	104.834978	12.0157372	17.4828	28.44947	0
19	27.052115	110.499803	13.9932364	10.16811	35.7921	0
20	30.8897656	113.259698	16.0010689	5.163772	40.79411	0
21	29.9200527	112.782775	15.2537428	6.324605	39.61092	0
22	12.9804077	94.1559505	6.93363447	32.83989	13.10717	1
23	0.00249836	73.9208419	0.56066875	57.5044	12.01126	1
24	0	80.1906645	1.27775892	52.7189	6.812895	1
25	0	79.5287315	1.09659008	53.24139	7.311974	1
26	0	76.0282713	0.7985497	55.86403	10.12544	1
27	0	73.1226828	0.50996555	58.12436	12.73705	1
28	0	73.6900727	0.52458655	57.69048	12.22588	1

Tabel 4.6 Pembaruan Centroid pada Iterasi Kedua

Centroid Lama			
C1	4.758007	84.57655	3.408693
C0	33.87024	117.2542	17.35204

Centroid Baru			
C1	4.758007	84.57655	3.408693
C0	33.87024	117.2542	17.35204
Selisih Centroid Baru dan Lama			
C1	0	0	0
C0	0	0	0
Total Selisih Centroid	0		

Pada iterasi kedua, setelah memperoleh hasil *cluster*. Selanjutnya dilakukan update *centroid* dengan nilai rata-rata data piksel pada setiap *cluster*. Setelah nilai centroid baru diperoleh, kemudian dihitung perubahan *centroid* lama dan baru melalui jumlah selisih antara *centroid* baru dan lama yang direpresentasikan oleh Tabel 4.6. Karena nilai jumlah selisih antara *centroid* baru dan lama dibawah batas toleransi (0,001) maka iterasi dihentikan. Hasil segmentasi ditunjukkan oleh Tabel 4.7.

Tabel 4.7 Hasil Segmentasi

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

4.2.2 Intersection over Union (IoU)

Tabel 4.8 Ground Truth

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

Tabel 4.9 Hasil Segmentasi

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

4.2.2.1 Intersection

Merupakan irisan antara nilai-nilai pada setiap piksel hasil segmentasi dengan *ground truth*. Untuk mendapatkan irisan dari hasil segmentasi berupa citra biner, maka digunakan operator *AND* dari kedua citra (citra hasil segmentasi dan *ground truth*) Hasil irisan antar kedua citra ditunjukkan pada Tabel 4.10.

Tabel 4.10 Hasil *Intersection*

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

4.2.2.2 Union

Merupakan gabungan antara nilai-nilai pada setiap piksel hasil segmentasi dengan *ground truth*. Untuk mendapatkan irisan dari hasil segmentasi berupa citra biner, maka digunakan operator *OR* dari kedua citra (citra hasil segmentasi dan *ground truth*). Hasil gabungan antar kedua citra ditunjukkan pada Tabel 4.11.

Tabel 4.11 Hasil *Union*

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

Berdasarkan persamaan 2.8, untuk mendapatkan IoU dari hasil segmentasi adalah dengan membagi jumlah piksel bernilai 1 pada hasil *intersection* (irisn) dengan jumlah piksel bernilai 1 pada hasil *union* (gabungan).

$$IoU = 12 / 13 * 100\%$$

$$IoU = 92,3 \%$$

4.2.3 Elementary Food Leftover Estimation (EFLE)

Untuk memperoleh estimasi bobot sisa makanan, diperlukan masukan berupa bobot dari makanan sebelum dikonsumsi serta citra hasil segmentasi dari makanan sebelum dan sesudah dikonsumsi. Citra hasil segmentasi ditunjukkan oleh Gambar 4.10.



Gambar 4.10 Citra Hasil Segmentasi Sebelum (a) dan Sesudah (b) Dikonsumsi

Estimasi bobot sisa makanan diperoleh melalui luas *area* (jumlah piksel *foreground*) dari citra sebelum dan sesudah dikonsumsi, oleh karena itu dilakukan pemotongan citra sesuai dengan kompartemennya (nasi, blank, telur, timun). Luas

area, bobot dari makanan sebelum dan sesudah dikonsumsi ditunjukkan oleh Tabel 4.12.

Tabel 4.12 Luas Area dan Berat Setiap Kompartemen

Kompartemen	Segmentasi	Luas Area (Jumlah Piksel Segmen Makanan)	Berat (gram)
Makanan Sebelum Dikonsumsi			
Nasi		195724	188
Blank		0	0
Telur		47203	38
Timun		12496	10
Makanan Setelah Dikonsumsi			
Nasi		89002	EFLE?
Blank		0	EFLE?
Telur		23241	EFLE?
Timun		9729	EFLE?

Estimasi bobot sisa makanan diperoleh melalui *EFLE* dengan persamaan 2.7. Hasil estimasi bobot sisa makanan setiap kompartemen ditunjukkan oleh Tabel 4.13.

Tabel 4.13 Estimasi Bobot Sisa Makanan Menggunakan *EFLE*

Kompartemen	Luas Area (Jumlah Pixel Segmen Makanan)		Bobot Makanan	
	Full	Sisa	Full	Estimasi Sisa (<i>EFLE</i>)
Nasi	195724	89002	188	85.48965
Blank	0	0	0	0
Telur	47203	23241	38	18.70979
Timun	12496	9729	10	7.785691

4.2.4 *RMSE* Bobot Estimasi

Untuk mengetahui tingkat kesalahan (*error*) dari hasil estimasi bobot sisa makanan, maka untuk memperoleh *error* terdistribusi digunakan *RMSE* pada persamaan 2.10 untuk mengetahui tingkat kesalahan pada suatu percobaan, hasil dari *RMSE* ditunjukkan oleh Tabel 4.14.

Tabel 4.14 *Error RMSE* Estimasi Bobot Sisa Makanan Menggunakan *EFLE*

	Kompartemen			
	Nasi	Blank	Telur	Timun
Bobot Asli	68	0	11	6
Bobot Estimasi	85.48965	0	18.70979	7.785691
Selisih (<i>error</i>)	17.48965	0	7.70979	1.785691
<i>RMSE</i>	9.598404			

BAB 5 IMPLEMENTASI

5.1 Batasan Implementasi

Batasan implementasi memuat batasan yang diterapkan selama implementasi sistem dalam estimasi sisa makanan otomatis di kotak makan menggunakan segmentasi citra berbasis *clustering* dengan algoritme K-means.

1. Sistem menggunakan library OpenCV, Numpy, Pandas, dan OS.
2. Metode segmentasi yang digunakan adalah algoritme K-means dengan nilai $K = 2$, dan toleransi perubahan centroid 0.000001.
3. Kategori *cluster* yang digunakan adalah *cluster* makanan (1) dan kotak makan (0).

5.2 Implementasi Sistem

Implementasi sistem berisi kode program berdasarkan perancangan sistem berupa diagram alir yang telah dibuat pada bab sebelumnya.

5.2.1 Power Law Transformation

Dalam pelaksanaan ini, data-data citra makanan pada kotak makan akan dilakukan proses segmentasi untuk mendapatkan citra makanan. Untuk kode program segmentasi citra makanan ditunjukkan oleh Tabel 5.1.

Tabel 5.1 Implementasi Fungsi Power Law

No	Algoritme 1: Power Law
1	<code>def powerLaw(img, gamma):</code>
2	<code> maxVal = np.max(img)</code>
3	<code> limit = round(maxVal * 0.9)</code>
4	<code> img[img > limit] = limit</code>
5	<code> return np.array(maxVal * (img / maxVal) ** gamma, dtype =</code>
	<code>'uint8')</code>

Penjelasan Tabel 5.1:

1. Baris 1 merupakan inisialisasi fungsi `powerLaw` dengan parameter masukan `img` (data array citra) dan `gamma` (nilai `gamma`).
2. Baris 2 digunakan untuk mendapatkan nilai tertinggi dari citra.
3. Baris 3 untuk menghitung batas nilai (90% dari nilai tertinggi).
4. Baris 4 untuk mengganti semua nilai yang lebih tinggi dari batas nilai (*limit*) dengan nilai batas (*limit*).
5. Baris 5 untuk mendeklarasikan nilai kembalian dari fungsi, yaitu hasil dari operasi pemangkatan dari citra yang dinormalisasi dalam rentang nilai 0 hingga 1 dengan nilai `gamma`.

5.2.2 Preprocessing Citra

Tabel 5.2 Implementasi Fungsi *Preprocessing*

No	Algoritme 2: Preprocessing
1	def preprocessing(image):
2	HSV= cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
3	HSV[:, :, 2] = np.amax(HSV[:, :, 2])
4	image = cv2.cvtColor(HSV, cv2.COLOR_HSV2BGR)
5	RGB= cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
6	YUV = cv2.cvtColor(bilateral, cv2.COLOR_BGR2YUV)
7	img_out = np.zeros((YUV.shape[0], YUV.shape[1], 3),
8	dtype=float)
9	img_out[:, :, 0] = powerLaw(YUV[:, :, 1], 7)
10	img_out[:, :, 1] = powerLaw(RGB[:, :, 1], 5)
11	blur = cv2.blur(YUV, (5, 5))
12	img_out[:, :, 2] = powerLaw(blur[:, :, 1], 4)
	return img_out

Penjelasan Tabel 5.2:

1. Baris 1 merupakan inisialisasi fungsi *preprocessing* dengan parameter masukan img (data array citra).
2. Baris 2 digunakan untuk mendapatkan citra dalam ruang warna HSV.
3. Baris 3 bertujuan untuk mengubah nilai setiap piksel pada *channel* warna V citra dalam ruang warna HSV menjadi nilai tertinggi dari channel warna V. Ini bertujuan untuk meminimalisir shadow pada citra.
4. Baris 4 digunakan untuk mendapatkan nilai citra dalam ruang warna BGR dari citra HSV.
5. Baris 5 digunakan untuk mendapatkan citra dalam ruang warna RGB.
6. Baris 6 untuk mendapatkan nilai citra dalam ruang warna YUV.
7. Baris 7 untuk inisialisasi variabel *output* berupa array 3 dimensi dengan dimensi pertama dan kedua sama seperti dimensi citra masukan, sedangkan dimensi ketiganya adalah 3.
8. Baris 8 untuk mengisi kolom pertama dari variabel keluaran citra dengan hasil kembalian dari fungsi *powerLaw()* dengan parameter masukan berupa citra (*channel* warna U pada citra YUV) dan nilai *gamma* (7).
9. Baris 9 untuk mengisi kolom kedua dari variabel keluaran citra dengan hasil kembalian dari fungsi *powerLaw()* dengan parameter masukan berupa citra (*channel* warna B pada citra RGB) dan nilai *gamma* (5).
10. Baris 10 untuk memanggil fungsi *cv2.blur()* dengan parameter masukan yaitu data citra YUV, dan kernel berukuran 5x5.

11. Baris 11 untuk mengisi kolom ketiga dari variabel keluaran citra dengan hasil kembalian dari fungsi `powerLaw()` dengan parameter masukan berupa citra (*channel* warna U pada citra YUV setelah dilakukan *averaging filter*) dan nilai *gamma* (4).

12. Baris 12 mendeklarasikan citra kembalian berupa *array* 3 dimensi.

5.2.3 Segmentasi Citra Menggunakan Algoritme K-means

Tabel 5.3 Implementasi Segmentasi dengan Algoritme K-Means

No	Algoritme 3: K-Means Segmentation
1	<code>def K-meansSegmentation(images):</code>
2	<code> max_iter = 1000</code>
3	<code> tol = 0.001</code>
4	<code> K = 2</code>
5	<code> classifications = {}</code>
6	<code> result = np.zeros((images.shape[0], images.shape[1]),</code> <code>dtype=np.int)</code>
7	<code> centroids = np.zeros((K, len(images[0, 0])), dtype=np.int)</code>
8	<code> for i in range(len(images[0, 0])):</code>
9	<code> centroids[0][i] = np.amin(images[:, :, i])</code>
10	<code> centroids[1][i] = np.amax(images[:, :, i])</code>
11	
12	<code> for iteration in range(max_iter):</code>
13	<code> for k in range(K):</code>
14	<code> classifications[k] = np.sqrt(np.sum(np.power((images -</code> <code>centroids[k]), 2), axis=-1))</code>
15	<code> marker = classifications[0] > classifications[1]</code>
16	
17	<code> result[marker] = 1</code>
18	<code> prev_centroids = centroids.copy()</code>
19	<code> centroids[0] = np.average(images[np.invert(marker)],</code> <code>axis=0)</code>
20	<code> centroids[1] = np.average(images[marker], axis=0)</code>
21	
22	<code> accuracy = np.sum(abs(centroids -</code> <code>prev_centroids))/np.sum(prev_centroids) * 100.0</code>
23	<code> if accuracy < tol:</code>
24	<code> result[marker] = 1</code>
25	<code> break</code>
26	<code> return result</code>

Penjelasan Tabel 5.3:

1. Baris 1 merupakan inisialisasi *function* atau *method* untuk menjalankan proses segmentasi menggunakan algoritme K-Means dengan parameter *input* variabel *images* yang merupakan variabel citra yang akan disegmentasi.
2. Baris 2 inisialisasi jumlah iterasi maksimal, baris 3 inisialisasi toleransi perubahan *centroid*.
3. Baris 4 inisialisasi jumlah *cluster* ($K=2$), inisialisasi jumlah *cluster* = 2 karena tujuan dari algoritme K-Means adalah untuk mengelompokkan data *piksel* menjadi 2 *cluster*, yaitu *cluster* makanan dan *cluster* *tray box*.
4. Baris ke 5 inisialisasi variabel *classifications* berupa *empty dictionary* untuk menyimpan piksel hasil *clustering*.

5. Baris ke 6 inialisasi *variabel result* berupa 2D *array* dengan dimensi sesuai dimensi citra *input* dengan *initial value* 0.
6. Baris ke 7 inialisasi *variabel centroids* berupa 2D *array* dengan dimensi K baris dan kolom sejumlah *channel* warna citra *input* dengan *initial value* 0.
7. Baris ke 8, perulangan sebanyak *channel* warna pada piksel citra *input*.
 - a. Baris ke 9, set variabel *centroids*[0][i] (variabel *centroid* pada baris pertama, kolom ke-i) dengan nilai minimum *value* dari citra pada *channel* warna ke-i. bertujuan untuk inialisasi *centroid* awal ke-1 dengan *minimum value* dari setiap *channel* warna citra *input*.
 - b. Baris ke 10, set variabel *centroids*[1][i] (variabel *centroid* pada baris kedua, kolom ke-i) dengan nilai *maximum value* dari citra pada *channel* warna ke-i. bertujuan untuk inialisasi *centroid* awal ke-2 dengan *maximum value* dari setiap *channel* warna citra *input*.
8. Baris ke 12, perulangan iterasi sebanyak *max_iter* (iterasi maksimal).
9. Baris ke 13, perulangan k sebanyak K (jumlah *cluster*).
10. Baris ke 14 merupakan perhitungan jarak *Euclidian* antara *centroid* dengan setiap piksel, Set variabel *classifications*[k] (pada *index* ke-k) dengan nilai *Euclidian* terhadap *centroid* ke-k.
11. Baris ke 15, set variabel *marker* dengan nilai dari *boolean masking* perbandingan antara perhitungan jarak *Euclidian centroid* pertama dan kedua untuk mendapatkan jarak terdekat antara kedua *centroid*. *Value index* pada *marker* akan bernilai *True* apabila jarak ke *centroid* pertama lebih besar dan akan bernilai *False* sebaliknya.
12. Baris ke 16, bertujuan untuk mengelompokkan data berdasarkan jarak terdekat terhadap *centroid*. Set variabel *result*[*marker*] dengan nilai 1. Maka variabel *result* yang awalnya berupa 2D *array* dengan nilai 0 akan diubah menjadi 1 pada *index* dari *marker* yang bernilai *True*.
13. Baris 18, salin *value* dari *centroids* lama ke variabel *prev_centroids*.
14. Baris 19, hitung *centroid* baru (C1) dengan rata-rata data pada cluster 0.
15. Baris 20, hitung *centroid* baru (C2) dengan rata-rata data pada cluster 1.
16. Baris 22, menghitung *accuracy* dari *centroid* (nilai perubahan *centroid*).
17. Baris 23, jika nilai perubahan < *tol* (toleransi), maka array *result* pada *index cluster 1* (*cluster* makanan) yang memiliki jarak lebih kecil dari *cluster 0* (*cluster* kotak makan) diubah nilainya menjadi 1, kemudian iterasi dihentikan, dan hasil segmentasi akan dikembalikan sebagai *output* pada baris ke 26.

5.2.4 Postprocessing Hasil Segmentasi

Tabel 5.4 Implementasi Fungsi Postprocessing

No	Algoritme 4: Postprocessing
----	-----------------------------


```

1 def postprocessing(images):
2     im_floodfill = images.copy()
3     h, w = images.shape
4     mask = np.zeros((h+2, w+2), np.uint8)
5     point = np.where(im_floodfill == 0)
6     cv2.floodFill(im_floodfill, mask, (point[1][0], point[0][0]),
7                 255)
8     im_floodfill_inv = cv2.bitwise_not(im_floodfill)
9     img_out = images | im_floodfill_inv
10    kernel = np.ones((5, 5), np.uint8)
11    erosion = cv2.erode(img_out, kernel, iterations = 3)
12    kernel = np.ones((15, 15), np.uint8)
13    dilation = cv2.dilate(erosion, kernel, iterations = 1)
14    img_out = img_out & dilation
15    img_out[img_out > 0] = 1
16    return img_out

```

Penjelasan Tabel 5.4:

1. Baris 1 merupakan inisialisasi fungsi untuk menjalankan proses *postprocessing* dengan parameter *input* citra biner hasil segmentasi.
2. Baris 2 digunakan untuk menyalin citra masukan ke variabel *im_floodfill*.
3. Baris 3 untuk deklarasi *h* (*height*) dan *w* (*width*) dari citra masukan.
4. Baris 4 untuk deklarasi *masking* yang akan digunakan pada proses *flood filling* dengan ukuran $h + 2$, dan $w + 2$.
5. Baris 5 untuk mendapatkan index dari titik yang bernilai 0 pada citra masukan (*background*).
6. Baris 6 untuk menjalankan fungsi *floodFill* dari *library* OpenCV dengan parameter masukan berupa citra, *masking*, titik awal, dan nilai untuk mengganti bagian yang telah melalui proses *flood filling*.
7. Baris 7 digunakan untuk mendapatkan inverse dari hasil *flood filling*.
8. Baris 8 bertujuan untuk menggabungkan citra masukan dengan citra *inverse* hasil *flood filling* untuk mendapatkan *masking* dari *foreground* dengan operasi OR.
9. Baris 9 untuk inisialisasi kernel dengan ukuran 5×5 .
10. Baris 10 untuk menjalankan operasi morfologi - erosi pada citra masukan sebanyak 3 kali dengan ukuran kernel 5×5 .
11. Baris 11 untuk inisialisasi kernel dengan ukuran 15×15 .
12. Baris 12 untuk menjalankan operasi morfologi - dilasi pada citra hasil erosi sebanyak 1 kali dengan ukuran kernel 15×15 .
13. Baris 13 digunakan untuk menggabungkan citra hasil *flood filling* dengan citra hasil dilasi dengan operasi AND.
14. Baris 14 berfungsi untuk mengubah semua nilai yang lebih dari 0 pada citra keluaran, menjadi bernilai 1. Hal ini bertujuan untuk menghasilkan keluaran berupa citra biner.

15. Keluaran berupa citra biner.

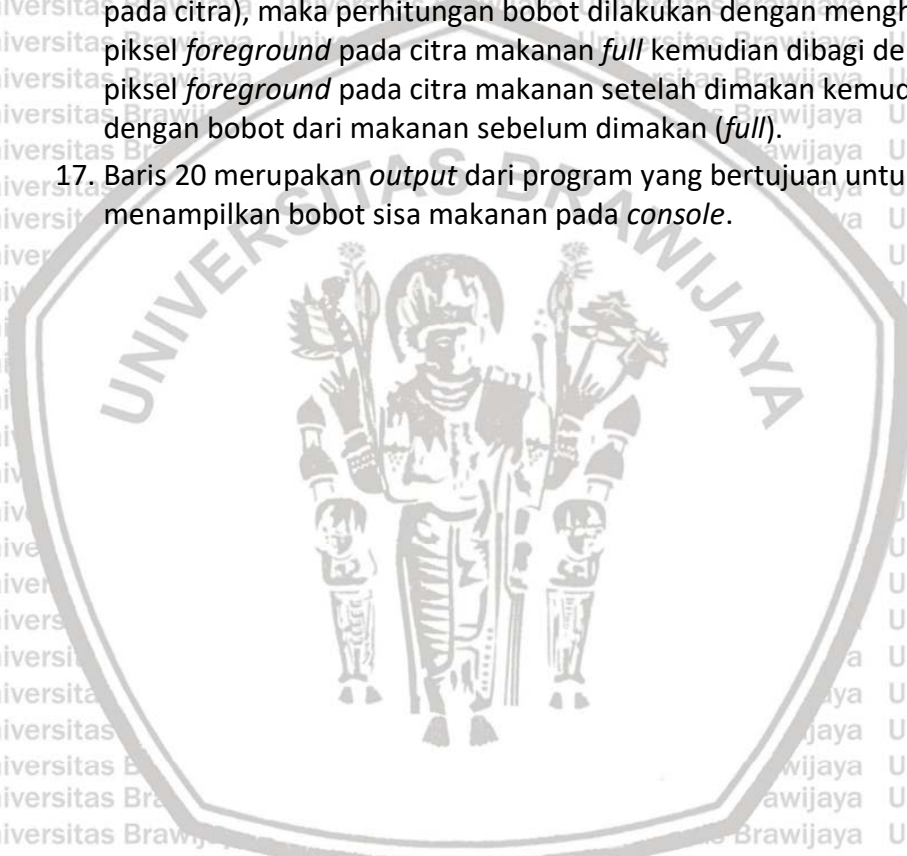
5.2.5 Perhitungan Bobot Sisa Makanan

No	Algoritme 5: EFLE
1	weight = pd.read_excel("weight.xlsx").values
2	image = cv2.imread("dataset/contour/ct_full.jpg")
3	gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
4	edges = cv2.Canny(gray, 50, 200)
5	contours, hierarchy = cv2.findContours(edges.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
6	sorted_contours = sorted(contours, key=cv2.contourArea, reverse=True)
7	for (i,c) in enumerate(sorted_contours):
8	x,y,w,h = cv2.boundingRect(c)
9	cropSegmentation = segmentation[y:y+h, x:x+w]
10	shapeSeg = cropSegmentation.shape
11	shapeFull = fullPixel[i].shape
12	row, col = shapeFull
13	if shapeFull[0] > shapeSeg[0]:
14	row = shapeSeg[0]
15	if shapeFull[1] > shapeSeg[1]:
16	col = shapeSeg[1]
17	weightPrediction = 0
18	if np.sum(fullPixel[i][:row, :col]) > 0:
19	weightPrediction = np.sum(cropSegmentation) / pixelFull[i]
20	* weightFull[i] print("Bobot sisa makanan :", weightPrediction)

Penjelasan Tabel 5.3:

1. Baris 1 digunakan untuk mendapatkan bobot makanan dari citra sebelum dimakan (*full*).
2. Baris 2 untuk membaca *manual benchmarking* yang akan digunakan untuk membagi citra menjadi beberapa kompartemen.
3. Baris 3 bertujuan untuk mengubah citra kedalam format ruang warna grayscale dengan library OpenCV.
4. Baris 4 digunakan untuk mendapatkan garis dari citra dengan metode *Canny*.
5. Baris 5 untuk mendapatkan *contours* atau pembagian *region* dari setiap kompartemen.
6. Baris 6 berfungsi untuk mengurutkan potongan citra berdasarkan luas yang dimiliki.
7. Baris 7 merupakan perulangan untuk setiap kompartemen.
8. Baris 8 digunakan untuk mendapatkan koordinat dari potongan kompartemen.
9. Baris 9 digunakan untuk mendapatkan potongan hasil segmentasi pada koordinat dari kompartemen saat ini.
10. Baris 10 bertujuan untuk mendapatkan dimensi dari potongan hasil segmentasi citra setelah dimakan (*sis*sa).
11. Baris 11 bertujuan untuk mendapatkan dimensi dari potongan hasil segmentasi citra sebelum dimakan (*full*).

12. Baris 12 merupakan inisialisasi variabel row dan col yang berisi jumlah baris dan kolom pada kompartemen citra makanan sebelum dimakan.
13. Baris 13 merupakan kondisi apabila jumlah baris pada citra *full* lebih banyak, maka pada baris 14, row diubah menjadi jumlah baris pada citra makanan setelah dimakan (sisa).
14. Baris 15 merupakan kondisi apabila jumlah kolom pada citra *full* lebih banyak, maka pada baris 16, col diubah menjadi jumlah kolom pada citra makanan setelah dimakan (sisa).
15. Baris 17 merupakan inisialisasi nilai awal dari bobot, yaitu 0.
16. Baris 18 merupakan kondisi apabila total piksel *foreground* pada potongan makanan *full* lebih dari 0 (berarti terdapat *foreground* atau segmen makanan pada citra), maka perhitungan bobot dilakukan dengan menghitung jumlah piksel *foreground* pada citra makanan *full* kemudian dibagi dengan jumlah piksel *foreground* pada citra makanan setelah dimakan kemudian dikalikan dengan bobot dari makanan sebelum dimakan (*full*).
17. Baris 20 merupakan *output* dari program yang bertujuan untuk menampilkan bobot sisa makanan pada *console*.



BAB 6 PENGUJIAN DAN PEMBAHASAN

6.1 Pengujian Hasil Cluster dan Segmentasi Citra

Bagian ini memuat hasil pengujian segmentasi citra dan estimasi bobot sisa makanan yang disajikan secara terpisah, dengan skenario pengujian sebagai berikut:

1. Pengujian untuk mendapatkan ruang warna terbaik berdasarkan kualitas hasil segmentasi.
2. Pengujian segmentasi dilakukan pada citra penuh, dan citra yang telah dipotong setiap kompartemen.
3. Pengujian untuk mendapatkan skenario terbaik hasil segmentasi terhadap akurasi *IoU* berdasarkan ruang warna terbaik.
4. Pengujian untuk mendapatkan akurasi estimasi bobot sisa makanan berdasarkan hasil segmentasi citra makanan.

6.2 Pengujian Ruang Warna

Pengujian ruang warna dilakukan melalui keterkaitan antara ruang warna dengan akurasi *IoU* hasil segmentasi. Pengujian pada ruang warna dilakukan dengan menggunakan ruang warna yang bervariasi. Berdasarkan ruang warna tersebut, maka akan dihasilkan akurasi hasil segmentasi yang berbeda-beda. Hasil pengujian ruang warna adalah untuk mendapatkan ruang warna terbaik pada tingkat akurasi segmentasi makanan pada citra kotak makan.

Tabel 6.1 Dataset Citra dengan Warna Kotak Makan Bervariasi

Background Putih		Background Hitam		Background Abu-Abu	
Foreground Selain Putih					
Citra	Nama	Citra	Nama	Citra	Nama
	0_W_1.jpg g		1_B_1.jpg g		1_G_1.jpg g
	0_W_11.jpg eg		1_B_2.jpg g		1_G_2.jpg g
Foreground Putih					
Citra	Nama	Citra	Nama	Citra	Nama
	0_W_2.jpg		2_B_1.jpg g		3_G_1.jpg g
	0_W_22.jpg		2_B_2.jpg g		3_G_2.jpg g

Tabel 6.2 Dataset Citra dengan Pencahayaan Bervariasi

Background Putih					
Pencahayaan Rendah		Pencahayaan Sedang		Pencahayaan Tinggi	
Citra	Nama	Citra	Nama	Citra	Nama
	A1.jpg		1.jpg		full.jpg
	1sisa.jpg		A1sisa.jpg		sisa028.jpg

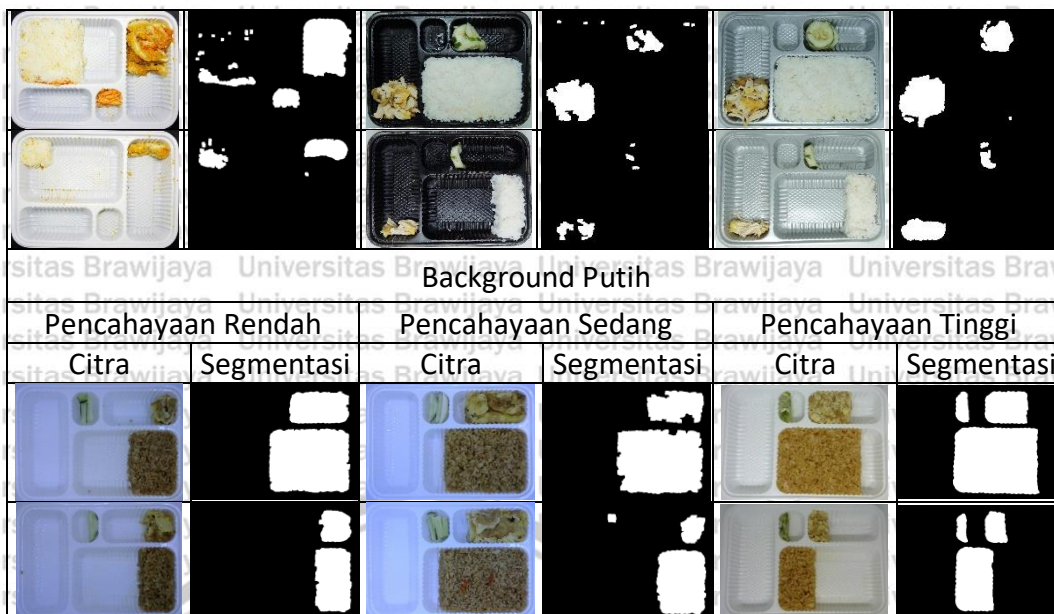
Table 6.1 Rata - Rata Akurasi Segmentasi dan Nilai Silhouette Coefficient

Ruang Warna	Rata - Rata IoU				
	Kotak 1	Kotak 2	Kotak 3	Kotak 4	Keseluruhan
HSV	0.65304	0.3758	0.3666	0.46	0.54351667
HSL	0.64441	0.3449	0.3926	0.4364	0.54797222
RGB	0.68109	0.3005	0.4671	0.47	0.57016667
LAB	0.68982	0.2374	0.3221	0.3951	0.50885
YCbCr	0.69238	0.2554	0.4252	0.4525	0.54567222
YUV	0.6913	0.2498	0.421	0.4221	0.54115
XYZ	0.68288	0.2804	0.4602	0.4594	0.56336667
YUV - U, RGB - B	0.71171	0.4588	0.53	0.547	0.62882222

Berdasarkan Tabel 6.3 diatas hasil pengujian pada pengujian ruang warna menunjukkan hasil terbaik pada ruang warna kebiruan (YUV – U, RGB – B) yang telah melalui tahap *preprocessing* dengan rata-rata akurasi *IoU* setiap kompartemen maupun *IoU* keseluruhan citra terbaik.

Tabel 6.3 Hasil Segmentasi Menggunakan Channel U pada Ruang Warna YUV

Background Putih		Background Hitam		Background Abu-Abu	
Foreground Selain Putih					
Citra	Segmentasi	Citra	Segmentasi	Citra	Segmentasi
					
					
Foreground Putih					
Citra	Segmentasi	Citra	Segmentasi	Citra	Segmentasi



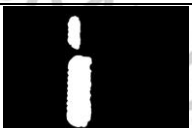
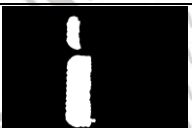
Berdasarkan hasil segmentasi dengan rata-rata *IoU* terbaik dari skenario pengujian ruang warna yang ditunjukkan pada Tabel 6.3, hasil segmentasi terbaik ditunjukkan oleh citra dengan *background* putih dan *foreground* berwarna selain putih. Berdasarkan hasil pengujian tersebut, maka citra dengan *background* putih dan *foreground* berwarna selain putih digunakan untuk menguji akurasi dari estimasi bobot sisa makanan pada pengujian selanjutnya.

6.3 Pengujian Hasil Segmentasi Citra

Pengujian hasil segmentasi bertujuan untuk mengetahui akurasi dari hasil segmentasi terhadap *Ground Truth*. Untuk mengetahui akurasi hasil segmentasi dihitung melalui *Intersection over Union* (*IoU*).

Tabel 6.4 Akurasi Hasil Segmentasi Terhadap *Ground Truth*

Nama Citra	Citra	<i>Ground Truth</i>	Segmentasi	<i>IoU</i>
full.jpg				0.9894
sisa001.jpg				0.9821
sisa002.jpg				0.9662
sisa003.jpg				0.9842

sisa004.jpg				0.9721
sisa005.jpg				0.9815
sisa006.jpg				0.9877
sisa007.jpg				0.9853
sisa008.jpg				0.9478
sisa009.jpg				0.9312
...	...	...	...	...
sisa030.jpg				0.9823

Tabel 6.5 Akurasi Hasil Segmentasi Setiap Kompartemen dan Keseluruhan Citra

Nama Citra	IoU Setiap Kompartemen				IoU Keseluruhan Citra
	Nasi	Blank	Telur	Timun	
full.jpg	0.9938	1	0.9867	0.9325	0.9894
sisa001.jpg	0.9809	1	0.9856	1	0.9821
sisa002.jpg	0.9663	1	0.9817	0.916	0.9662
sisa003.jpg	0.9876	1	1	0.9421	0.9842
sisa004.jpg	0.9759	1	0.9568	0.9661	0.9721
sisa005.jpg	0.9818	1	1	0.9788	0.9815
sisa006.jpg	0.9878	1	1	1	0.9877
sisa007.jpg	0.9863	1	1	0.9712	0.9853
sisa008.jpg	0.9464	1	1	0.9587	0.9478
sisa009.jpg	0.9291	1	1	0.9405	0.9312
sisa010.jpg	1	1	1	0.9302	0.9296
sisa011.jpg	0.9673	1	1	0.9576	0.9659
sisa012.jpg	0.9787	1	1	0.9328	0.974
sisa013.jpg	0.9581	1	1	1	0.9581
sisa014.jpg	0.9649	1	1	0.9181	0.9612

sisa015.jpg	0.9695	1	1	0.9362	0.9665
sisa016.jpg	0.8945	1	1	0.9231	0.8999
sisa017.jpg	0.9263	1	1	0.9364	0.9283
sisa018.jpg	0.9721	1	1	0.9339	0.9673
sisa019.jpg	0.8755	1	1	0.8837	0.8772
sisa020.jpg	0.8731	1	0.9829	0.9343	0.9102
sisa021.jpg	1	1	1	0.9214	0.9214
sisa022.jpg	0.9762	1	1	0.9308	0.9711
sisa023.jpg	0.9865	1	1	0.9396	0.9823
sisa024.jpg	0.9876	1	0.9784	0.9391	0.983
sisa025.jpg	0.9877	1	0.9704	0.9244	0.982
sisa026.jpg	0.94	1	1	0.9134	0.9353
sisa027.jpg	0.9826	1	1	0.917	0.9746
sisa028.jpg	0.9837	1	0.986	0.9293	0.9797
sisa029.jpg	0.9882	1	0.9789	0.9285	0.9822
sisa030.jpg	0.9823	1	1	1	0.9823
Rata-rata	0.965506	1	0.993787	0.943087	0.959987097

Melalui pengamatan visual pada hasil segmentasi citra dengan *background* berwarna putih dan *foreground* selain warna putih yang ditunjukkan pada Tabel 6.4 memiliki bentuk yang hampir menyerupai *ground truth*. Ini menunjukkan bahwa K-means dapat menghasilkan hasil segmentasi pada citra dengan *background* berwarna putih dan *foreground* selain warna putih dengan kualitas baik. Hal ini didukung melalui rata-rata akurasi IoU dari hasil segmentasi sebesar 95.6% yang ditunjukkan pada Tabel 6.5. Dengan akurasi *IoU* yang baik, maka diharapkan pada pengujian selanjutnya diperoleh hasil estimasi bobot sisa makanan dengan akurasi yang baik.

6.4 Pengujian Hasil Estimasi Bobot Sisa Makanan

Tabel 6.6 Perbandingan Hasil Estimasi Bobot Sisa Makanan dengan Target

Nama Citra	Bobot Setiap Kompartemen (gram)							
	Nasi		Blank		Telur		Timun	
	Target	Estimasi	Target	Estimasi	Target	Estimasi	Target	Estimasi
sisa001.jpg	61	77.892	0	0	20	22.411	0	0.000
sisa002.jpg	52	67.528	0	0	20	22.376	5	6.597
sisa003.jpg	91	106.272	0	0	0	0.000	5	6.968
sisa004.jpg	70	81.852	0	0	10	12.899	8	8.908
sisa005.jpg	84	104.807	0	0	0	0.000	7	9.140
sisa006.jpg	118	129.710	0	0	0	0.000	0	0.000
sisa007.jpg	141	151.200	0	0	0	0.000	8	9.171
sisa008.jpg	74	86.134	0	0	0	0.000	7	8.782
sisa009.jpg	31	48.708	0	0	0	0.000	7	9.031
sisa010.jpg	0	0.000	0	0	0	0.000	7	9.234
sisa011.jpg	47	62.585	0	0	0	0.000	7	9.073

sisa012.jpg	82	96.973	0	0	0	0.000	7	9.079
sisa013.jpg	64	77.139	0	0	0	0.000	0	0.000
sisa014.jpg	73	86.026	0	0	0	0.000	4	5.689
sisa015.jpg	86	96.704	0	0	0	0.000	6	8.023
sisa016.jpg	28	43.098	0	0	0	0.000	6	8.383
sisa017.jpg	43	41.351	0	0	0	0.000	6	8.138
sisa018.jpg	59	71.832	0	0	0	0.000	7	8.440
sisa019.jpg	25	40.062	0	0	0	0.000	6	8.528
sisa020.jpg	25	40.259	0	0	19	14.102	6	8.443
sisa021.jpg	0	0.000	0	0	0	0.000	6	8.493
sisa022.jpg	66	83.367	0	0	0	0.000	6	8.410
sisa023.jpg	83	98.968	0	0	0	0.000	6	7.839
sisa024.jpg	99	112.010	0	0	9	15.127	6	7.786
sisa025.jpg	113	122.391	0	0	6	11.006	6	7.714
sisa026.jpg	25	45.413	0	0	0	0.000	6	7.810
sisa027.jpg	51	69.620	0	0	0	0.000	6	7.749
sisa028.jpg	68	85.490	0	0	11	18.710	6	7.786
sisa029.jpg	89	103.510	0	0	17	24.709	6	7.618
sisa030.jpg	57	73.931	0	0	0	0.000	0	0.000

Tabel 6.7 Error Estimasi Bobot Sisa Makanan Terhadap Bobot Asli

Nama Citra	Error				MSE	RMSE
	Nasi	Blank	Telur	Timun		
sisa001.jpg	16.892	0	2.41133	0	72.787	8.53
sisa002.jpg	15.528	0	2.37591	1.5973	62.326	7.89
sisa003.jpg	15.272	0	0	1.9678	59.275	7.7
sisa004.jpg	11.852	0	2.89905	0.9085	37.426	6.12
sisa005.jpg	20.807	0	0	2.1397	109.38	10.5
sisa006.jpg	11.71	0	0	0	34.28	5.85
sisa007.jpg	10.2	0	0	1.1709	26.352	5.13
sisa008.jpg	12.134	0	0	1.782	37.603	6.13
sisa009.jpg	17.708	0	0	2.0309	79.423	8.91
sisa010.jpg	0	0	0	2.2342	1.2479	1.12
sisa011.jpg	15.585	0	0	2.0733	61.795	7.86
sisa012.jpg	14.973	0	0	2.0789	57.127	7.56
sisa013.jpg	13.139	0	0	0	43.157	6.57
sisa014.jpg	13.026	0	0	1.689	43.13	6.57
sisa015.jpg	10.704	0	0	2.0234	29.667	5.45
sisa016.jpg	15.098	0	0	2.3827	58.409	7.64
sisa017.jpg	1.6489	0	0	2.1378	1.8223	1.35
sisa018.jpg	12.832	0	0	1.4403	41.682	6.46
sisa019.jpg	15.062	0	0	2.5275	58.313	7.64
sisa020.jpg	15.259	0	4.89823	2.4427	65.699	8.11

sisa021.jpg	0	0	0	2.4931	1.5539	1.25
sisa022.jpg	17.367	0	0	2.4099	76.854	8.77
sisa023.jpg	15.968	0	0	1.8385	64.588	8.04
sisa024.jpg	13.01	0	6.12739	1.7865	52.499	7.25
sisa025.jpg	9.3906	0	5.00561	1.7137	29.044	5.39
sisa026.jpg	20.413	0	0	1.8105	104.99	10.2
sisa027.jpg	18.62	0	0	1.7489	87.438	9.35
sisa028.jpg	17.49	0	7.70979	1.7857	92.129	9.6
sisa029.jpg	14.51	0	7.7089	1.6184	68.149	8.26
sisa030.jpg	16.931	0	0	0	71.661	8.47
Error terkecil					1.2479	1.12
Rata-rata					54.327	6.99

Berdasarkan data hasil pengujian yang ditunjukkan pada Tabel 6.7 menunjukkan selisih estimasi bobot sisa makanan yang cukup besar. Meskipun dari pengujian hasil segmentasi sebelumnya menghasilkan akurasi yang cukup besar, namun hasil estimasi yang dihasilkan memiliki *error* yang cukup besar. Berdasarkan rumus yang digunakan pada *EFLE*, seharusnya semakin akurat hasil segmentasi citra makanan sebelum dan sesudah dimakan akan menghasilkan estimasi bobot dengan akurasi lebih baik. Dengan asumsi *ground truth* merupakan hasil segmentasi paling akurat terhadap citra asli, maka apabila rumus *EFLE* digunakan pada citra *ground truth* diperoleh hasil bobot sesuai dengan bobot asli makanan.

Tabel 6.8 Perbandingan Hasil Estimasi Bobot Sisa Makanan dengan Target Hasil *EFLE* dari *Ground Truth*

Nama Citra	Bobot Setiap Kompartemen (gram)							
	Nasi		Blank (Kosong)		Telur		Timun	
	Target	Estimasi	Target	Estimasi	Target	Estimasi	Target	Estimasi
sisa001.jpg	76.534	77.892	0	0	20.613	22.411	0.000	0.000
sisa002.jpg	65.075	67.528	0	0	20.895	22.376	6.530	6.597
sisa003.jpg	103.521	106.272	0	0	0.000	0.000	6.475	6.968
sisa004.jpg	79.785	81.852	0	0	10.790	12.899	7.172	8.908
sisa005.jpg	102.356	104.807	0	0	0.000	0.000	7.965	9.140
sisa006.jpg	126.996	129.710	0	0	0.000	0.000	0.000	0.000
sisa007.jpg	148.835	151.200	0	0	0.000	0.000	7.304	9.171
sisa008.jpg	81.410	86.134	0	0	0.000	0.000	7.832	8.782
sisa009.jpg	44.722	48.708	0	0	0.000	0.000	7.835	9.031
sisa010.jpg	0.000	0.000	0	0	0.000	0.000	7.372	9.234
sisa011.jpg	60.785	62.585	0	0	0.000	0.000	8.246	9.073
sisa012.jpg	93.402	96.973	0	0	0.000	0.000	7.996	9.079
sisa013.jpg	73.308	77.139	0	0	0.000	0.000	0.000	0.000
sisa014.jpg	81.785	86.026	0	0	0.000	0.000	5.222	5.689
sisa015.jpg	92.185	96.704	0	0	0.000	0.000	6.277	8.023
sisa016.jpg	37.821	43.098	0	0	0.000	0.000	6.064	8.383

sisa017.jpg	37.901	41.351	0	0	0.000	0.000	6.333	8.138
sisa018.jpg	69.123	71.832	0	0	0.000	0.000	7.494	8.440
sisa019.jpg	34.854	40.062	0	0	0.000	0.000	6.585	8.528
sisa020.jpg	34.717	40.259	0	0	11.207	14.102	6.096	8.443
sisa021.jpg	0.000	0.000	0	0	0.000	0.000	6.742	8.493
sisa022.jpg	81.478	83.367	0	0	0.000	0.000	7.261	8.410
sisa023.jpg	97.273	98.968	0	0	0.000	0.000	6.783	7.839
sisa024.jpg	109.394	112.010	0	0	14.210	15.127	7.291	7.786
sisa025.jpg	120.041	122.391	0	0	10.669	11.006	7.582	7.714
sisa026.jpg	42.547	45.413	0	0	0.000	0.000	6.649	7.810
sisa027.jpg	67.126	69.620	0	0	0.000	0.000	6.505	7.749
sisa028.jpg	82.671	85.490	0	0	17.752	18.710	7.231	7.786
sisa029.jpg	101.311	103.510	0	0	23.220	24.709	6.923	7.618
sisa030.jpg	71.125	73.931	0	0	0.000	0.000	0.000	0.000

Tabel 6.9 Error Estimasi Bobot Sisa Makanan Terhadap Bobot Hasil EFLE dari *Ground Truth*

Nama Citra	Error (gram)				MSE	RMSE
	Nasi	Blank	Telur	Timun		
sisa001.jpg	1.357511	0	1.79816	0	1.269054	1.126523
sisa002.jpg	2.452479	0	1.481251	0.067305	2.053322	1.432942
sisa003.jpg	2.750503	0	0	0.493183	1.952124	1.397184
sisa004.jpg	2.067471	0	2.109235	1.736596	2.934768	1.713117
sisa005.jpg	2.451481	0	0	1.174385	1.847234	1.35913
sisa006.jpg	2.713399	0	0	0	1.840633	1.356699
sisa007.jpg	2.365073	0	0	1.866698	2.269534	1.506497
sisa008.jpg	4.724376	0	0	0.949854	5.805488	2.409458
sisa009.jpg	3.985864	0	0	1.195524	4.329097	2.080648
sisa010.jpg	0	0	0	1.862525	0.86725	0.931262
sisa011.jpg	1.799263	0	0	0.827155	0.980383	0.990143
sisa012.jpg	3.571199	0	0	1.083077	3.481629	1.865912
sisa013.jpg	3.830655	0	0	0	3.66848	1.915328
sisa014.jpg	4.240426	0	0	0.46678	4.549775	2.13302
sisa015.jpg	4.519192	0	0	1.746089	5.867982	2.422392
sisa016.jpg	5.277495	0	0	2.318819	8.307219	2.882225
sisa017.jpg	3.45018	0	0	1.805166	3.790591	1.946944
sisa018.jpg	2.709169	0	0	0.946719	2.058969	1.434911
sisa019.jpg	5.207843	0	0	1.942163	7.723406	2.779102
sisa020.jpg	5.541792	0	2.894907	2.346745	11.14979	3.33913
sisa021.jpg	0	0	0	1.751301	0.766764	0.875651
sisa022.jpg	1.888552	0	0	1.14898	1.221696	1.105303
sisa023.jpg	1.695366	0	0	1.055774	0.997231	0.998615
sisa024.jpg	2.61634	0	0.917456	0.495895	1.983218	1.408268
sisa025.jpg	2.349319	0	0.336205	0.131833	1.412429	1.188456

sisa026.jpg	2.866476	0	0	1.161751	2.391587	1.546476
sisa027.jpg	2.493769	0	0	1.243745	1.941446	1.393358
sisa028.jpg	2.818519	0	0.957739	0.554465	2.292187	1.513997
sisa029.jpg	2.198955	0	1.488977	0.695299	1.883974	1.372579
sisa030.jpg	2.805436	0	0	0	1.967618	1.402718
Error terkecil					0.766764	0.875651
Rata-rata					3.120163	1.660933

Berdasarkan data hasil pengujian yang ditunjukkan pada Tabel 6.9 menunjukkan selisih estimasi bobot sisa makanan yang jauh lebih kecil. Hal ini menunjukkan bahwa semakin baik akurasi hasil segmentasi akan menghasilkan estimasi bobot dengan lebih akurat. Berdasarkan rumus perhitungan bobot menggunakan *EFLE*, bobot diperoleh melalui luas area citra *foreground* yang diketahui. Jadi bobot diperoleh berdasarkan citra dalam 2 dimensi, sedangkan pada kasus sesungguhnya, makanan merupakan objek 3 dimensi, sehingga terdapat satu dimensi yang tidak dapat diperhitungkan oleh algoritme *EFLE* dalam mengestimasi bobot sisa makanan, yaitu pada dimensi ketiga makanan (ketebalan atau kepadatan makanan).



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil pengujian pada bagian sebelumnya maka dapat dibuat 2 rangkuman sesuai dengan rumusan masalah. Yang pertama, algoritme K-Means dapat digunakan untuk melakukan segmentasi citra makanan dengan baik. Hal tersebut dibuktikan dengan hasil akurasi *IoU* cukup tinggi dengan nilai *error* rendah, dari pengujian diperoleh rata-rata akurasi sebesar 95.6% dengan akurasi tertinggi sebesar 98.9%. Hasil segmentasi tersebut dipengaruhi oleh pemilihan segmen warna, tahapan *preprocessing* dan *postprocessing* pada proses segmentasi. Kedua, hasil estimasi bobot makanan berdasarkan hasil segmentasi kurang akurat, karena terdapat satu dimensi yang tidak dapat diperhitungkan oleh algoritme *EFLE* dalam mengestimasi bobot sisa makanan, yaitu pada dimensi ketiga makanan (ketebalan atau kepadatan makanan), namun apabila ketebalan atau kepadatan makanan sebelum dan sesudah dimakan sama, maka akan menghasilkan estimasi bobot makanan dengan *error* lebih kecil (hasil estimasi lebih akurat). *Error RMSE* terkecil dari hasil estimasi sebesar 1.12.

7.2 Saran

Dalam riset selanjutnya perlu diperhatikan adanya *noise* berupa bayangan dan tekstur pada *dataset*. Segmentasi dilakukan untuk setiap kompartemen untuk mengatasi perbedaan warna cukup tinggi dari makanan pada kompartemen yang berbeda. Selain itu perlu dilakukan pengujian menggunakan fitur gabungan ruang warna yang lebih banyak. Pada penelitian selanjutnya menggunakan data dengan makanan yang lebih bervariasi, serta pengambilan data dari berbagai sudut pandang.

DAFTAR REFERENSI

- Andika, Tahta Herdian and Aliy Hafiz. 2018. "Analisis Perbandingan Segmentasi Citra Menggunakan Metode K-Means Dan Fuzzy C-Means." *Seminar Nasional Teknologi Dan Bisnis 2018* 237–46.
- Bano, Saima and M. N. A. Khan. 2018. "A Survey of Data Clustering Methods." *International Journal of Advanced Science and Technology* 113(April):133–42.
- Chen, Tse Wei, Yi Ling Chen, and Shao Yi Chien. 2008. "Fast Image Segmentation Based on K-Means Clustering with Histograms in HSV Color Space." in *Proceedings of the 2008 IEEE 10th Workshop on Multimedia Signal Processing, MMSP 2008*.
- Deeparani, K. and P. Sudhakar. 2021. "Efficient Image Segmentation and Implementation of K-Means Clustering." *Materials Today: Proceedings* 45:8076–79.
- Dhanachandra, Nameirakpam, Khumanthem Manglem, and Yambem Jina Chanu. 2015. "Image Segmentation Using K-Means Clustering Algorithm and Subtractive Clustering Algorithm." Pp. 764–71 in *Procedia Computer Science*. Vol. 54. Elsevier.
- Kementerian Kesehatan RI. 2020. "Pengelolaan Air Limbah Pengelolaan Limbah Padat Domestik Pengelolaan Limbah B3 Medis Padat." 1–14.
- Kumboyono, Vina. 2013. "Indikator Pemenuhan Kebutuhan Nutrisi Oleh Pasien Rawat Inap Rumah Sakit Tentara Dr. Soepraoen Malang." *Ners* 8(2):183–89.
- Kupferschmidt, Kai. 2019. "Meet the Blue Crew, Scientists Trying to Give Food, Flowers, and More a Color Rarely Found in Nature." *Science*.
- Laksmi. 2018. "Indonesian Journal of Human Nutrition." *Indonesian Journal of Human Nutrition* 1(1):23–34.
- Muthukumar, K., S. Poorani, and S. Sindhu. 2016. "Color Image Segmentation Using Similarity Based Region Merging and Flood Fill Algorithm." 5(06):40–46.
- Nisak, Nova Khairun, Putri Ronitawati, and Khairizka Citra. 2019. "Metode PDAT Dan Comstock Lebih Efisien Dibandingkan Food Weighing Dalam Menilai Sisa Makanan Pasien." *Nutrire Diaita* 11(1):18.
- Podpora, Michal, Grzegorz Paweł Korbaś, and Aleksandra Kawala-Janik. 2014. "YUV vs RGB—Choosing a Color Space for Human-Machine Interaction." *Position Papers of the 2014 Federated Conference on Computer Science and Information Systems* 3(September):29–34.
- Putri, Rani, Reni Sofiyatin, Lalu Abdi, and AASP Chandradewi. 2019. "KESESUAIAN PORSI LAUK DAGING AYAM YANG DITERIMA PADA MAKANAN BIASA." *Jurnal Gizi Prima* 4(1):54.
- Raid, A. ..., W. ... Khedr, M. ... El-dosuky, and Mona Aoud. 2014. "Image Restoration Based on Morphological Operations." *International Journal of Computer Science, Engineering and Information Technology*.
- Rerung, Rintho Rante. 2018. "Penerapan Data Mining Dengan Memanfaatkan Metode Association Rule Untuk Promosi Produk." *Jurnal Teknologi Rekayasa* 3(1):89.
- Rezatofighi, Hamid, Nathan Tsoi, Junyoung Gwak, Amir Sadeghian, Ian Reid, and

- Silvio Savarese. 2019. "Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2019*-June:658–66.
- Said, Khairul Anuar Mat and Asral Bahari Jambek. 2017. "A Study on Image Processing Using Mathematical Morphological." *2016 3rd International Conference on Electronic Design, ICED 2016* (April 2018):507–12.
- Sari, Yuita Arum, Ratih Kartika Dewi, Jaya Mahar Maligan, Anindya Sasri Ananta, and Sigit Adinugroho. 2019. "Automatic Food Leftover Estimation in Tray Box Using Image Segmentation." in *Proceedings of 2019 4th International Conference on Sustainable Information Engineering and Technology, SIET 2019*.
- Sari, Yuita Arum, Jaya Mahar Maligan, and Andriko Fajar Prakoso. 2020. "Improving the Elementary Leftover Food Estimation Algorithm by Using Clustering Image Segmentation in Nutrition Intake Problem." Pp. 435–39 in *CENIM 2020 - Proceeding: International Conference on Computer Engineering, Network, and Intelligent Multimedia 2020*. Institute of Electrical and Electronics Engineers Inc.
- Vimal, S. P. and P. K. Thiruvikraman. 2012. "Automated Image Enhancement Using Power Law Transformations." *Sadhana - Academy Proceedings in Engineering Sciences* 37(6):739–45.
- Vyas, Aparna, Soohwan Yu, and Joonki Paik. 2018. *Fundamentals of Digital Image Processing*.
- Yao, Hong, Qingling Duan, Daoliang Li, and Jianping Wang. 2013. "An Improved K-Means Clustering Algorithm for Fish Image Segmentation." *Mathematical and Computer Modelling* 58(3–4):790–98.
- Yuliana, Y. 2020. "Corona Virus Diseases (Covid-19): Sebuah Tinjauan Literatur." *Wellness And Healthy Magazine* 2(1):187–92.

A.1 Data Citra Makanan pada Kotak Makan



