

Sistem Rekomendasi *Color Pairing* Makanan dengan Pilihan Piring Menggunakan Teknik Pengolahan Citra

Muhammad Akmal - 13517028
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: akmalmuhammad51@gmail.com

Abstrak—Dewasa ini, sosial media telah mengubah banyak hal dalam kehidupan manusia, terutama dengan fitur berbagi foto lewat berbagai macam cara. Tak terkecuali, foto makanan merupakan salah satu yang sangat menarik perhatian dikarenakan kemampuannya untuk bisa meningkatkan nafsu makan serta selera ketika melihatnya. Namun, bagi orang awam yang ingin melakukannya yang tidak memiliki dasar kemampuan kombinasi dan pemasangan warna (*color pairing*) makanan dengan piring yang dimilikinya, hal ini terasa lebih sulit. Maka dari itu, pada makalah ini akan dibahas sistem rekomendasi *color pairing* makanan dengan pilihan piring pengguna dengan teknik pengolahan citra untuk memudahkan pengguna menghasilkan foto dengan kombinasi warna yang cocok dan menarik.

Kata kunci—makanan, piring, *color pairing*, *food photography*, pengolahan citra, segmentasi citra, *K-Means Clustering*

I. PENDAHULUAN

Dewasa ini, penggunaan sosial media telah banyak mengubah perilaku manusia apalagi dengan adanya pandemi, semua orang menggunakan sosial media untuk bersosialisasi dengan keluarga dan kerabatnya. Salah satu fitur di sosial media yang sangat berpengaruh adalah fitur berbagi foto dengan berbagai macam cara mulai dari *post*, *story* hingga melalui *chat*. Fitur ini merupakan salah satu fitur utama karena dengan foto, orang bisa bersosialisasi dan berbagi ceritanya dengan keluarganya dan kerabatnya. Jenis foto yang dibagikan pun bisa bermacam-macam mulai dari foto studio, foto pemandangan, suasana, dan yang tak kalah menarik adalah foto makanan.

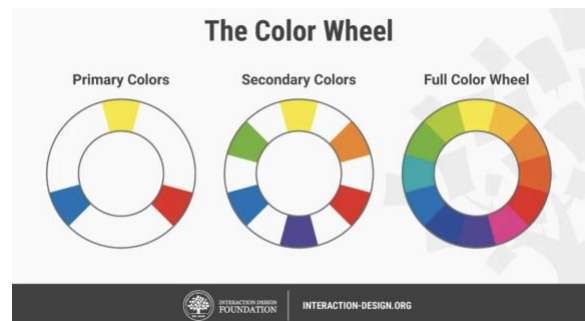
Foto makanan merupakan salah satu foto yang cukup digemari karena pengaruhnya yang bisa menarik perhatian serta meningkatkan nafsu makan yang melihatnya, baik itu untuk membelinya ataupun tidak. Tak heran, banyak orang yang membagikan ceritanya melalui foto makanan. Namun, bagi orang awam yang tidak memiliki pemahaman yang baik akan kombinasi warna di dalam suatu foto, untuk membuat foto makanan menarik cukup sulit dan membutuhkan waktu untuk mencobanya. Pada makalah ini ditawarkan sebuah solusi berbasis pemrosesan citra untuk mempermudah orang awam yang ingin membuat foto makanannya menarik layaknya profesional dari sudut pandang kombinasi warna piring dengan makanan. Pengguna akan diminta untuk mengunggah foto piring yang dimilikinya, dan aplikasi akan memberikan

rekomendasi warna makanan dan komponennya yang cocok untuk ditempatkan pada piring tersebut.

II. DASAR TEORI

A. Teori Warna dan Palet Warna

Teori warna merupakan sekumpulan aturan dan pedoman yang digunakan desainer untuk berkomunikasi dengan penggunaannya melalui skema warna yang menarik di dalam antarmuka visual [1]. Untuk mengambil warna terbaik setiap kalinya, desainer menggunakan *color wheel* (roda warna) dan merujuk pada pengetahuan lainnya tentang kemampuan optik manusia, psikologi, budaya dan lainnya.



Gambar II.1 Contoh Color Wheel

Di dalam roda warna, warna diorganisasikan ke dalam 3 grup yaitu: warna primer, warna sekunder dan warna tersier. Roda warna juga merepresentasikan *hue*, *tints*, *tones* dan *shades* dari masing-masing relative warna [2]. Palet warna merupakan kombinasi warna yang digunakan desainer ketika mendesain. Palet warna dapat digunakan dalam mendesain *user interface*, poster, logo dan banyak hal lainnya. Ketika digunakan secara baik dan benar, palet warna membuat desain yang dihasilkannya memiliki pondasi visual, konsistensi serta estetika. Dewasa ini, palet warna sering direpresentasikan sebagai kombinasi kode *hexadecimal* walaupun bisa juga direpresentasikan sebagai warna yang menyusunnya baik itu menggunakan RGB, CMYK maupun yang lainnya.



Gambar II.2 Comtoh Palet Warna dengan Gambar Inspirasinya

B. Fotografi Makanan

Fotografi makanan adalah fotografi yang berfokus pada makanan, atau menggunakan sebagai subjek fotografi sebagai ekspresi dari ide-ide yang tidak berkaitan dengan makanan [3]. Yang paling penting, hal yang membentuk definisi dari fotografi makanan adalah foto memiliki makanan sebagai subjek utama atau sekunder. Fotografi makanan tradisional bisa didefinisikan sebagai *table top style* atau gaya yang mengambil foto dari atas meja seperti yang sering kita lihat pada *cookbook* atau majalah. Gaya ini biasanya mengambil bagian *closeup*, sudut yang sedikit mengangkat ke atas dan sering dilatari dengan bayangan untuk memberik kesan lebih terhadap tekstur makanan dan juga menggunakan properti dapur tradisional. Namun, belakangan juga berkembang fotografi makanan kontemporer yang memiliki definisi lebih luas dan kreatif, dan bisa menggunakan gabungan dari berbagai metode *styling* serta teknik pencahayaan. Apapun bentuknya, tujuan dari fotografi makanan adalah membuat makanan yang ada di dalam foto mengambil perhatian dan meningkatkan nafsu makan.

Salah satu teknik dasar untuk mendapatkan foto makanan yang baik adalah penggunaan warna pada properti yang digunakan. Dikarenakan ini makanan, maka property dasarnya adalah alas yang digunakan menempatkan makanan atau dengan kata lain piring. Penggunaan piring dengan warna yang cocok dikombinasikan dengan makanan yang ingin difoto merupakan salah satu teknik dasar agar mendapatkan foto makanan yang menarik.

C. Segmentasi Citra

Segmentasi citra (*image segmentation*) bertujuan untuk membagi citra menjadi region-region atau objek-objek serta memisahkan objek dengan latar belakang. Yang dilakukan oleh segmentasi citra adalah menemukan bagian citra yang koheren atau objek spesifik. Citra disegmentasi berdasarkan properti yang dipilih seperti kecerahan, warna, tekstur, dan sebagainya. Segmentasi membagi citra menjadi sejumlah region yang terhubung, tiap region bersifat homogen berdasarkan properti yang dipilih.

Umumnya, metode segmentasi citra dikelompokkan berdasarkan 2 (dua) pendekatan:

1. Diskontinuitas
Metode ini mempartisi citra berdasarkan perubahan nilai intensitas *pixel* yang cepat seperti tepi (*edge detection*)
2. Kesamaan

Metode ini mempartisi citra berdasarkan kemiripan area menurut properti yang ditentukan metode segmentasi citra yang termasuk ke dalam pendekatan ini:

- a. Pengambangan (*thresholding*)
Segmentasi citra didasarkan pada nilai intensitas *pixel-pixel* dan nilai ambang T . Salah satu cara untuk mengekstrak objek dari latar belakang adalah dengan memilih ambang T . Setiap *pixel* (x, y) pada citra di mana $f(x, y) > T$ disebut titik objek, jika tidak maka akan disebut latar belakang. Hasil segmentasi adalah berupa citra biner. Teknik pengambangan dibagi menjadi 3 (tiga) yaitu *global thresholding*, *local thresholding* dan *adaptive thresholding*
- b. *Region growing*
Teknik ini mengelompokkan *pixel* atau *sub-region* yang tumbuh menjadi *region* yang lebih besar. Secara garis besar, algoritme dari teknik ini dimulai dengan “umpan (*seed*)” yang berisi himpunan beranggota satu atau lebih *pixel* dari *region* yang potensial, dan dari sini *region* berkembang dengan menambahkan pada umpan *pixel-pixel* tetangga yang memiliki properti yang mirip dengan umpan, lalu berhenti jika *pixel-pixel* tetangga tidak mirip lagi. Keuntungan dari teknik ini adalah memiliki keterhubungan yang bagus antar *pixel* di dalam *region* sedangkan kelemahannya adalah pemilihan umpan harus tepat Dan mengonsumsi waktu yang lama.
- c. *Split and merge*
Menggunakan algoritme *divide and conquer*. Teknik ini membagi citra menjadi sejumlah *region* yang *disjoint* yang setelah itu melakukan penggabungan (*merge*) *region-region* bertetangga yang homogen.
- d. *Clustering*
Teknik ini membuat bagian-bagian yang memiliki kesamaan menjadi satu kelompok atau *cluster*. Algoritme yang biasa digunakan adalah algoritme *k-means clustering*. Tantangan dalam menggunakan teknik ini adalah mencari nilai k yang tepat jika memang belum tahu ingin mendapatkan berapa kelompok bagian gambar.

D. Canny Edge Detection

Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak dalam jarak yang singkat. Tepi memiliki arah, dan arah ini berbeda-beda bergantung pada perubahan intensitas. Tepi biasanya terdapat pada batas antara dua daerah yang berbeda intensitas dengan perubahan yang sangat cepat di dalam citra. Tepi memiliki 4 (empat) macam yaitu tepi curam (*step edge*), tepi landai (*ramp edge*), tepi garis (*line edge*), dan tepi atap (*roof edge*). Pendeteksian tepi bertujuan untuk meningkatkan penampakan garis batas atau objek di dalam citra.

Pendeteksian tepi dapat dipahami dengan pendekatan kalkulus diferensial sebab, perubahan intensitas yang besar

dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan yang besar. Kemiringan fungsi dapat dihitung dengan turunan pertama (*gradient*). Operator gradien ini memiliki banyak macam seperti operator Sobel, operator Roberts, operator Prewitt dan operator Canny.

Operator Canny yang biasa juga disebut sebagai *Canny Edge Detection* adalah operator deteksi tepi yang terkenal karena dapat menghasilkan tepi dengan ketebalan 1 *pixel*. Langkah-langkah operator Canny:

1. Haluskan citra I dengan penapis Gaussian G
2. Hitung gradien setiap *pixel* dengan salah satu dari operator yang lain (misalnya operator Sobel)
3. Jika nilai mutlak gradien suatu *pixel* melebihi nilai ambang T, maka *pixel* termasuk *pixel* tepi

E. Deteksi Kontur

Pendeteksi tepi menghasilkan citra tepi yang berupa citra biner. Pixel-pixel tepi berwarna putih, sedangkan pixel bukan-tepi berwarna hitam. Tetapi, tepi belum memberikan informasi yang berguna karena belum ada keterkaitan antara suatu tepi dengan tepi lainnya. Oleh karena itu, citra tepi ini harus diproses lebih lanjut untuk menghasilkan informasi yang lebih berguna yang dapat digunakan dalam mendeteksi bentuk-bentuk sederhana. Bentuk-bentuk sederhana misalnya garis lurus, lingkaran, elips, dan sebagainya. Bentuk-bentuk ini berguna di dalam proses analisis citra. Rangkaian *pixel-pixel* tepi yang membentuk batas daerah (*region boundary*) disebut kontur. Kontur dapat bersifat terbuka atau tertutup.



Gambar II.3 Contoh Kontur Tertutup (kanan) dan Kontur Terbuka (kiri)

Representasi kontur dapat berupa senarai tepi (*edge list*) atau berupa kurva. Senarai tepi merupakan himpunan terurut *pixel-pixel* tepi. Representasi kontur ke dalam kurva merupakan representasi dalam bentuk persamaan. Misalnya, rangkaian *pixel* tepi yang membentuk garis dapat direpresentasikan hanya dengan sebuah persamaan garis lurus. Representasi semacam ini menyederhanakan perhitungan selanjutnya seperti arah dan panjang garis.

F. K Means Clustering

Clustering adalah salah satu teknik analisis data eksplorasi yang paling umum digunakan untuk mendapatkan intuisi tentang struktur data. Ini dapat didefinisikan sebagai tugas mengidentifikasi subkelompok dalam data sedemikian rupa sehingga titik data dalam subkelompok yang sama (*cluster*) sangat mirip sedangkan titik data di cluster yang berbeda sangat berbeda. Dengan kata lain, kita mencoba menemukan subkelompok homogen dalam data sedemikian rupa sehingga titik data di setiap *cluster* semirip mungkin

menurut ukuran kesamaan seperti jarak berbasis euclidean atau jarak berbasis korelasi. Keputusan tentang ukuran kemiripan yang akan digunakan adalah khusus untuk aplikasi. Terdapat banyak sekali algoritme *clustering* salah satunya adalah algoritme *K-Means*.

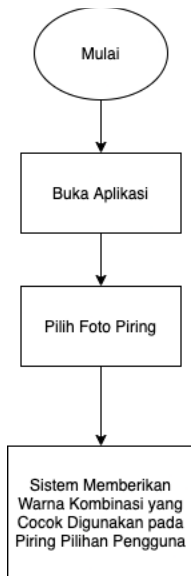
K-Means Clustering adalah algoritma iteratif yang mencoba untuk mempartisi *dataset* menjadi subkelompok *non-overlapping* berbeda yang ditentukan oleh nilai $K(\text{cluster})$ di mana setiap titik data hanya dimiliki oleh satu kelompok. Ini mencoba membuat titik data *intra-cluster* semirip mungkin sambil juga menjaga *cluster* sejauh mungkin. Ini menetapkan poin data ke *cluster* sedemikian rupa sehingga jumlah jarak kuadrat antara titik data dan pusat massa *cluster* (rata-rata aritmatika dari semua titik data yang termasuk dalam cluster itu) minimal. Semakin sedikit variasi yang kita miliki dalam *cluster*, semakin homogen (serupa) titik data dalam *cluster* yang sama.

Cara kerja algoritme *K-Means* adalah sebagai berikut:

1. Tentukan jumlah *cluster* K.
2. Inisialisasi sentroid dengan secara acak memilih titik data K untuk sentroid tanpa penggantian.
3. Terus lakukan iterasi hingga tidak ada perubahan pada sentroid. Yaitu pembagian poin data ke *cluster* tidak berubah. Iterasi yang dimaksud adalah sebagai berikut:
 - a. Hitung jumlah jarak kuadrat antara titik data dan semua sentroid.
 - b. Tetapkan setiap titik data ke cluster terdekat (sentroid).
 - c. Hitung sentroid untuk cluster dengan mengambil rata-rata dari semua titik data yang dimiliki setiap cluster.

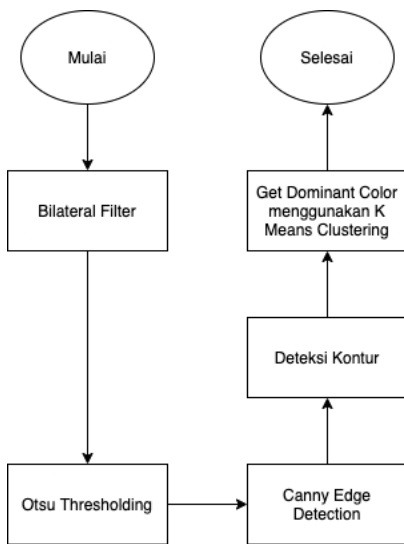
III. IMPLEMENTASI DAN PENGUJIAN SISTEM REKOMENDASI COLOR PAIRING MAKANAN DENGAN PILIHAN PIRING

Sistem rekomendasi ini dikembangkan menggunakan *library* tkinter sebagai *Graphical User Interface* (GUI), pillow sebagai penyimpanan dan pembaca *object Image* serta OpenCV sebagai *library* pemrosesan citra. Secara garis besar, alur penggunaan aplikasi ini dijelaskan di Gambar III.1.



Gambar III.1 Flowchart Penggunaan Aplikasi

Dari segi arsitektur sistem, yang menggunakan OpenCV dan Python sebagai Bahasa pemrograman utama dijelaskan secara garis besar pada Gambar III.2

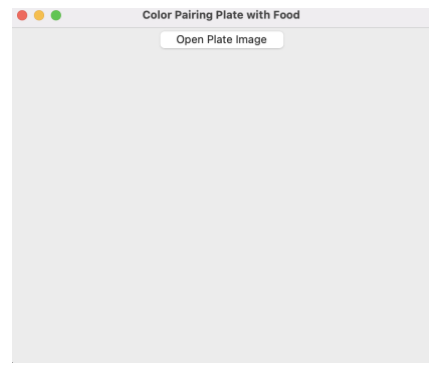


Gambar III.2 Arsitektur Sistem Rekomendasi Color Pairing Makanan dengan Pilihan Piring

Selanjutnya akan dijelaskan masing-masing bagian baik itu pada Gambar III.1 maupun Gambar III.2.

A. Buka Aplikasi

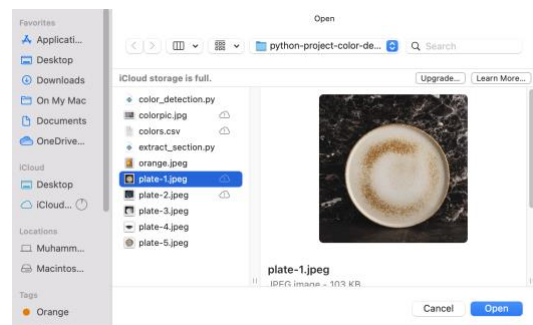
Saat pengguna membuka aplikasi system rekomendasi *color pairing* makanan dengan pilihan piring, pengguna akan ditampilkan tombol untuk melakukan pembukaan pada *file* foto piring yang ingin dimasukkan. *Graphical User Interface* (GUI) pada tahapan ini ditunjukkan di Gambar III.3.



Gambar III.3 Graphical User Interface (GUI) Tahap Membuka Aplikasi

B. Pilih Foto Piring

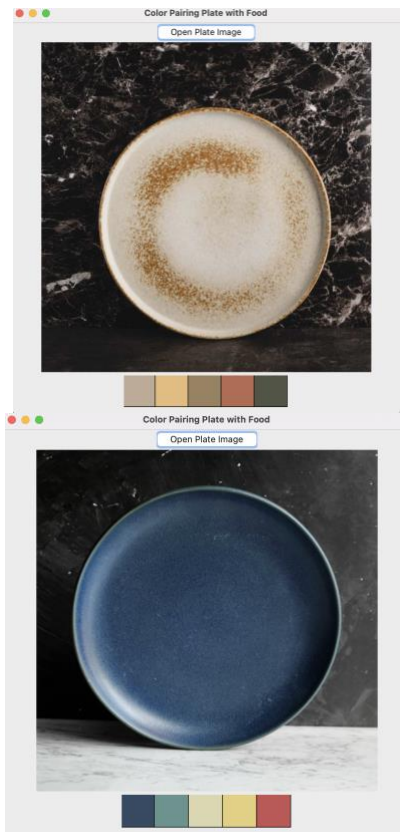
Pada tahapan ini, ketika pengguna menekan tombol *Open Plate Image*, maka akan muncul *file dialog* yang mengarahkan pengguna untuk memasukkan gambar piring yang ingin digunakan oleh pengguna. GUI *file dialog* ditunjukkan pada Gambar III.4.



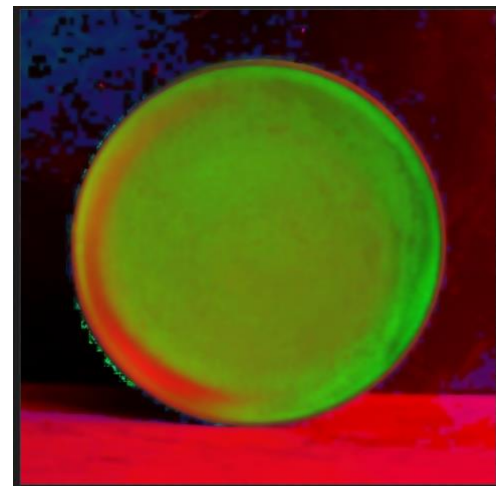
Gambar III.4 Graphical User Interface (GUI) File Dialog

C. Sistem Memberikan Warna Kombinasi yang Cocok Digunakan pada Piring Pilihan Pengguna

Pada tahapan ini, setelah pengguna memilih foto piring yang ingin digunakan, antarmuka akan menampilkan foto piring tersebut pada aplikasi disertai di bawahnya terdapat rekomendasi warna makanan yang dapat digunakan dalam piring tersebut. *Graphical User Interface* (GUI) pada tahapan ini ditunjukkan pada Gambar III.5



Gambar III.5 Graphical User Interface (GUI) Tahap Sistem Memberikan Rekomendasi Warna Makanan



Gambar III.6 Hasil Penapisan Bilateral / Bilateral Filtering

D. Bilateral Filtering

Bilateral filter merupakan jenis filter yang mengurangi *noise* atau derau untuk membuat gambar lebih *smooth*. Tipe dari *filter* ini adalah *non-linear* yang mengubah sebuah citra dengan rata-rata *filter* dari citra tersebut namun bobot dari *pixel* tidak hanya bergantung pada jarak *Euclidean* namun juga perbedaan *radiometric*[6].

Pada tahapan ini digunakan kode menggunakan *library* OpenCV yaitu:

```
img = cv2.bilateralFilter(img, 9, 105, 105)
```

9 merepresentasikan diameter dari setiap *pixel neighborhood* yang digunakan untuk penapisan/*filtering*. 105 pertama merepresentasikan *sigmaColor*, yaitu penapis sigma di ruang warna dan 105 kedua merepresentasikan *sigmaSpace*, yaitu penapis sigma di ruang koordinat. Hasil dari penapisan bilateral ini ditunjukkan pada Gambar III.6. Dapat terlihat bahwa hasil penapisan memberikan warna yang lebih jelas pada piring yang dipilih pengguna dibandingkan latar belakangnya.

E. Otsu Thresholding

Pengambangan atau biasa disebut *thresholding* merupakan suatu teknik pemrosesan citra untuk melakukan binerisasi citra berdasarkan intensitas *pixel*. Terdapat banyak sekali macam teknik pengambangan, salah satunya adalah teknik pengambangan Otsu atau yang kerap disebut *Otsu Thresholding*. Teknik ini diberi nama Otsu sesuai dengan orang yang menemukannya yaitu Nobuyuki Otsu. Pada tahapan ini digunakan *library* OpenCV yaitu:

```
ret, thresh_image = cv2.threshold(equalize, 0, 255, cv2.THRESH_OTSU + cv2.THRESH_BINARY)
```

equalize merupakan variable yang menyimpan citra yang sudah dilakukan *histogram equalization*, 0 adalah nilai *pixel* rendah dan 255 adalah nilai *pixel* tinggi. Dapat dilihat bahwa terjadi pembuatan ambang batas, di dalam piring terlihat menjadi putih semua sedangkan pada latar, karena latarnya juga memiliki variasi warna dan memiliki warna yang mirip dengan warna piring, maka terdapat bagian yang berwarna putih dan berwarna hitam.

Hasil dari tahapan ini dapat dilihat pada Gambar III.7



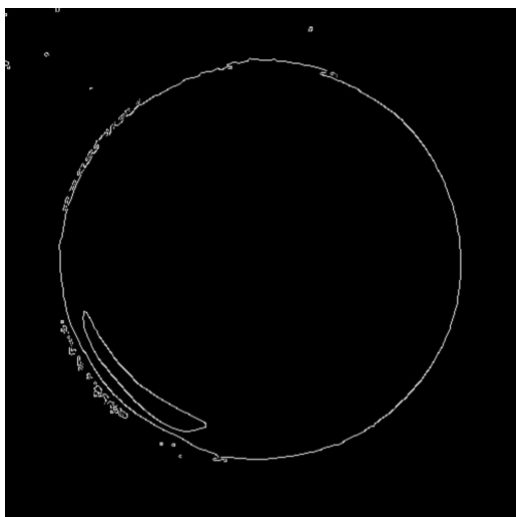
Gambar III.7 Hasil Tahapan Pengambangan Otsu / Otsu Thresholding

F. Canny Edge Detection

Pada tahapan ini digunakan *library* OpenCV yaitu:

```
canny_image = cv2.Canny(equalize,250,255)
canny_image = cv2.convertScaleAbs(canny_image)
```

equalize merepresentasikan citra yang telah dilakukan *histogram equalization*, 250 merepresentasikan *threshold1* yaitu nilai *threshold* tinggi sedangkan 255 merepresentasikan *threshold2* yaitu nilai *threshold* rendah. Hasil dari tahapan ini ditunjukkan pada



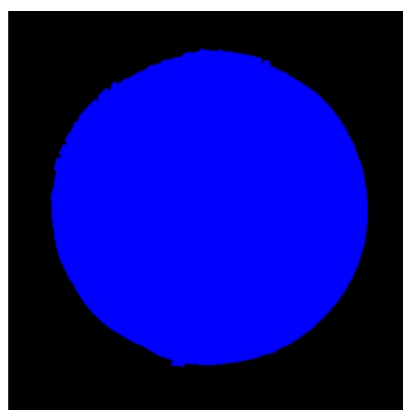
Gambar III.8 Hasil Canny Edge Detection

G. Contour Detection

Pada tahapan ini dilakukan pencarian kontur agar mendapatkan bagian dari piring yang dimasukkan oleh pengguna. Pada tahapan ini menggunakan *library* OpenCV yaitu:

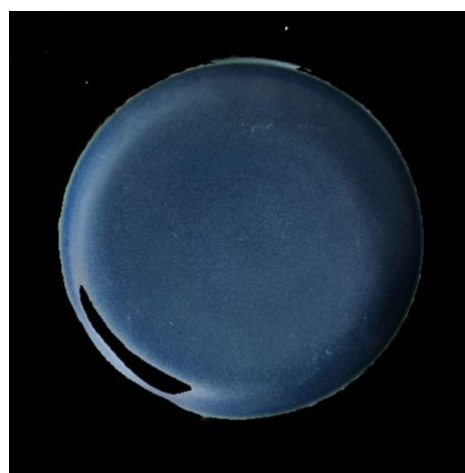
```
dilated_image = cv2.dilate(canny_image, kernel, iterations=1)
contours, hierarchy = cv2.findContours(dilated_image, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
c=contours[0]
mask = np.zeros(img_rgb.shape, np.uint8)
self.extracted_image = cv2.drawContours(mask, [c], 0, 255, -1)
self.extracted_image = cv2.bitwise_and(img_rgb, img_rgb, mask = equalize)
```

Pada bagian ini dilakukan dilatasi terhadap citra terlebih dahulu. Setelah itu dilakukan pencarian kontur dengan *cv2.findContours*. Setelah itu dilakukan *masking*, dan dilakukan penggambaran kontur dengan *mask* yang dibuat sebelumnya. Hasil dari tahapan ini ditunjukkan pada



Gambar III.9 Hasil Tahap Contour Detection

Setelah dilakukan pendeteksian kontur, dilakukan operasi *bitwise and* untuk mendapatkan gambar piring yang lebih jelas dan sudah terfokus tanpa adanya gangguan dari latar belakang lagi. Digunakan *library* OpenCV yaitu:



Gambar III.10 Hasil Bitwise And (extracted_image)

H. Dominant Color Menggunakan K-Means Clustering

Di tahap terakhir, dilakukan pencarian warna dominan dari piring yang dipilih pengguna. Setelah dilakukan berbagai macam teknik pemrosesan citra, digunakan *K-Means Clustering* dengan nilai $k=2$ karena pada gambar seharusnya akan memiliki 2 (dua) warna dominan yaitu warna hitam pada latar belakang serta warna dominan dari piring yang dipilih pengguna. Sebelumnya dilakukan ruang warna dari RGB ke BGR, lalu mengubah bentuk citra menjadi list agar bisa dijadikan fitur untuk dilakukan *K-Means Clustering*. Bagian ini dijelaskan pada kode di bawah.

```
img = cv2.cvtColor(self.extracted_image,
cv2.COLOR_BGR2RGB)
img = img.reshape((img.shape[0] *
img.shape[1], 3))
```

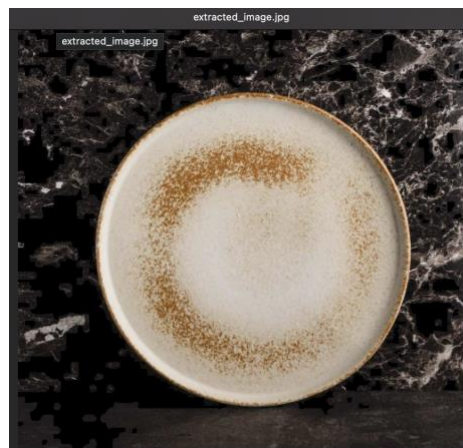
Setelah itu, dilakukan *K-Means Clustering* terhadap citra yang telah diubah bentuknya. Bagian ini dijelaskan pada kode di bawah.

```
kmeans = KMeans(n_clusters =
self.CLUSTERS)
kmeans.fit(img)
colorInt = self.COLORS.astype(int)
```

Variabel `colorInt` menyimpan 2 buah warna dominan, dan diambil warna dengan jumlah komponen-komponen penyusunnya yang lebih besar karena yang lebih kecil seharusnya adalah warna dominan dari latar belakang yaitu hitam. Setelah mendapatkan warna dominan, dilakukan pencarian *color palette* yang sesuai dengan menggunakan API dari `Colormind`. API ini mengembalikan 5 buah warna yang bisa dijadikan *color palette* berdasarkan warna yang diberikan oleh pengguna. Setelah mendapat *color palette* tersebut, ditampilkan ke dalam *Graphical User Interface* (GUI) dengan menggambarkan kotak dan mengisi sesuai dengan warna yang pada *color palette* yang diberikan. Warna dominan yang ditemukan oleh tahapan ini akan ditampilkan di bagian paling kiri dari *color palette* yang ditampilkan di GUI.

I. Pengujian Aplikasi

Aplikasi sistem rekomendasi *color pairing* makanan dengan pilihan piring pengguna sudah cukup baik untuk memberikan warna yang cocok digunakan pada piring tersebut. Namun, untuk beberapa kasus perlu ditangani lebih lanjut seperti pada kasus piring pada Gambar III.11 yang ketika dilakukan teknik yang telah disebutkan diatas tidak dapat terlalu presisi dalam menentukan warna dominan dikarenakan terdapat *derau* yang cukup tinggi di bagian *background* atau latar belakangnya.



Gambar III.11 Kasus Tidak Presisi

Pada kasus ini, hasil *extracted image* mirip dengan citra utama, padahal pada kasus piring sebelumnya, *extracted image* menghasilkan Gambar III.10. Jadi diperlukan latar belakang yang tidak memiliki banyak variasi warna dan cukup berbeda dengan warna piring agar sistem dapat menemukan warna dominan secara lebih presisi

IV. KESIMPULAN DAN SARAN

Aplikasi sistem rekomendasi *color pairing* makanan dengan pilihan piring pengguna berjalan sesuai dengan ekspektasi, yaitu bisa memberikan warna rekomendasi yang dapat digunakan untuk digunakan pada piring yang dipilih oleh pengguna. Namun, tidak dapat dipungkiri masih bisa dilakukan perbaikan lagi terutama dengan mencoba berbagai macam teknik penapisan serta deteksi tepi agar bisa mencakup lebih banyak kasus uji dengan karakteristik yang bermacam-macam.

Untuk selanjutnya, aplikasi ini sebenarnya bisa dikembangkan lebih jauh lagi dengan teknik-teknik yang lebih baik dan mutakhir. Warna-warna rekomendasi yang diberikan selanjutnya dapat dicarikan bahan makanan yang sesuai dengan warna tersebut dan sekaligus mempertimbangkan aspek rasa sehingga tidak dapat secara acak memilih bahan yang warnanya sesuai, namun juga memilih sesuai dengan rasa yang akan dihasilkan. Lebih jauh lagi, aplikasi ini bisa saja menjadi aplikasi pemberi rekomendasi resep berdasarkan piring yang dipilih oleh pengguna. Mungkin bisa ditambahkan preferensi pengguna seperti apakah ingin jenis makanan penutup, makanan pembuka atau makanan ini. Bisa juga ditambahkan preferensi makan pengguna seperti apakah ingin yang bebas daging, bebas gluten dan sebagainya

REFERENSI

- [1] <https://www.interaction-design.org/literature/topics/color-theory>, diakses pada 25 Mei 2021
- [2] <https://careerfoundry.com/en/blog/ui-design/introduction-to-color-theory-and-color-palettes/#:~:text=Modern%20color%20theory%20is%20largely,second%20ary%20colors%2C%20and%20tertiary%20colors>, diakses pada 25 Mei 2021
- [3] <https://phoode.com/blog/what-is-the-definition-of-food-photography-and-what-makes-it-so-special/>, diakses pada 25 Mei 2021
- [4] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2020-2021/17-Segmentasi-Citra-2021.pdf>, diakses pada 23 Mei 2021
- [5] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2020-2021/13-Pendeteksian-Tepi-2021.pdf>, diakses pada 23 Mei 2021
- [6] <https://www.codespeedy.com/bilateral-filter-in-opencv-in-python/>, diakses pada 25 Mei 2021
- [7] <https://machinelearningknowledge.ai/bilateral-filtering-in-python-opencv-with-cv2-bilateralfilter/>, diakses pada 25 Mei 2021
- [8] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2020-2021/14-Kontur-2021.pdf>, diakses pada 25 Mei 2021

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 25 Mei 2021



Muhammad Akmal / 13517028