

Sistem Pengenalan Sayur-sayuran dengan Metode *Image Processing*

Nando Rusrin Pratama / 13517148
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: nandopku17@gmail.com

Abstrak—Salah satu dari banyak aplikasi metode *image processing* atau pengolahan citra adalah membantu dalam sistem pengenalan objek. Sistem pengenalan objek dapat membantu dalam mengidentifikasi dan mengklasifikasikan objek-objek ke dalam kelas yang berbeda. Dalam mengenali sebuah objek di antara objek lain akan memudahkan jika terdapat tambahan informasi seperti anotasi dan jumlah objek. Dalam kehidupan sehari-hari, salah satu objek yang sering kita jumpai adalah sayur-sayuran. Dalam mengenali sayur-sayuran, beberapa hal yang menjadi komponen penting adalah warna, bentuk dan ukuran sayur. Salah satu solusi untuk membuat sistem pengenalan sayur-sayuran adalah dengan menggunakan metode *image processing* yaitu segmentasi citra dan metode klasifikasi berbasis *machine learning* yaitu k-Nearest Neighbors (KNN).

Kata kunci—Segmentasi Citra, Deteksi Tepi, Pengolahan Citra, k-Nearest Neighbors, KNN, RGB, HSV, Sayur

I. PENDAHULUAN

Dalam kehidupan sehari-hari, kita sering menjumpai berbagai objek di sekitar kita, mulai dari objek-objek yang ada di dapur hingga yang ada di kamar kita. Dalam mengenali objek-objek yang ada di sekitar kita, akan memudahkan jika terdapat sebuah sistem yang dapat mengenali objek yang kita lihat. Untuk mengenali suatu objek, dapat digunakan *image* atau foto dari objek yang ingin dikenali.

Dalam suatu *image*, dapat terdiri dari beberapa objek-objek yang berbeda. Identifikasi atau pengenalan objek-objek dalam suatu *image* dapat dilakukan dengan melakukan segmentasi citra terhadap objek-objek yang ada di dalam *image* sehingga tiap objek dalam *image* dapat diidentifikasi dan dikenali. Lalu, untuk memudahkan pengenalan objek, dapat dilakukan juga dengan memberi anotasi di samping objek dan juga keterangan mengenai jumlah masing-masing objek.

Salah satu objek yang sering dijumpai atau ditemui dalam kehidupan sehari-hari adalah sayur-sayuran. Pengenalan sayur-sayuran umumnya dapat dilakukan secara visual melalui mata. Namun, pada era digital, pengenalan sayur-sayuran dapat dilakukan dengan lebih cepat secara digital melalui foto dan video digital. Fitur-fitur penting pada sayuran seperti warna, bentuk, dan ukuran sayur dapat digunakan untuk melakukan pengenalan sayur-sayuran.

Untuk mengenali suatu objek dalam sebuah *image*, dapat dilakukan sejumlah *image processing* atau pengolahan citra

sehingga objek yang ingin dikenali atau diidentifikasi dapat diolah dan diklasifikasikan ke dalam kelas objek yang berbeda. Dalam melakukan pengenalan sayur-sayuran, nantinya akan dilakukan identifikasi objek dengan metode *image processing* yaitu segmentasi citra, lalu dilakukan ekstraksi fitur-fitur citra yang tersegmentasi. Fitur-fitur yang sudah diekstraksi akan digunakan untuk melakukan klasifikasi objek berbasis *machine learning* yaitu k-Nearest Neighbors (KNN).

Pada makalah ini akan dibahas bagaimana cara membuat sistem pengenalan sayur-sayuran dengan pendekatan berbasis segmentasi citra dan metode klasifikasi berbasis *machine learning*. Pada penelitian ini, akan dilakukan pengenalan sayur-sayuran untuk kentang dan wortel. Dari penelitian ini, diharapkan dapat membantu dalam pembuatan sistem pengenalan sayur-sayuran serta untuk mengoptimalkan sistem pengenalan objek secara digital.

II. LANDASAN TEORI

Pada bagian ini akan dibahas teori-teori pengolahan citra dan metode klasifikasi berbasis *machine learning* yang dipakai dalam mengimplementasi sistem yang dirancang pada makalah ini.

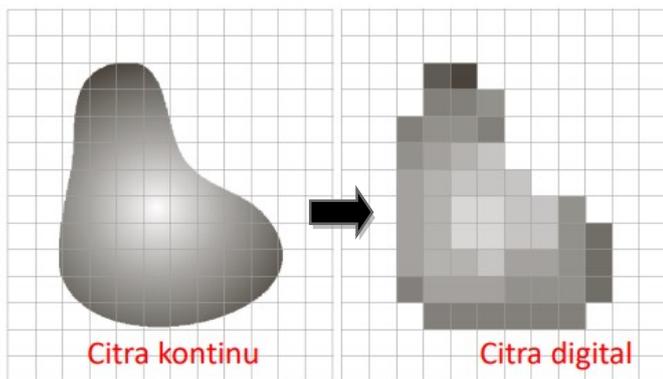
A. Citra Digital

Citra digital atau *digital image* merupakan sekumpulan titik yang disebut sebagai *pixel* yang membentuk informasi gambar pada komputer. Citra digital merupakan representasi dari citra atau gelombang kontinu yang dilakukan digitalisasi sehingga dapat direpresentasikan pada komputer yang datanya berupa diskrit.

Proses digitalisasi citra ada dua tahap yaitu (Rinaldi, 2021):

1. Penerokan (sampling), yaitu proses digitalisasi citra dalam ranah spasial (x,y)
2. Kuantisasi, yaitu proses meng-angka-kan nilai intensitas gelombang $f(x,y)$ menjadi bilangan bulat

Dari 2 tahap pada proses digitalisasi citra di atas, dihasilkan sebuah citra digital yang berbentuk matriks berisi informasi warna. Pada gambar 2.1 dapat dilihat ilustrasi dari proses digitalisasi sebuah citra asli menjadi sebuah citra digital setelah dilakukan penerokan dan kuantisasi.

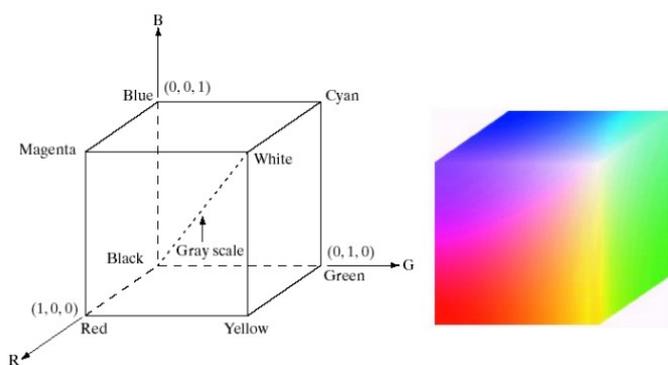


Gambar 2.1 Proses Digitalisasi Citra

B. Model Warna

Pada citra digital, setiap titik atau *pixel* berisi informasi warna yang direpresentasikan dengan model warna tertentu sehingga representasi warna dapat ditampilkan dengan benar di layar komputer. Model warna yang ada merupakan kesepakatan bersama yang telah disepakati oleh lembaga-lembaga yang merancangannya. Model-model warna yang banyak digunakan pada saat ini adalah model warna RGB, CMY, CMYK, YCbCr, HSI, XYZ dan HSV. Masing-masing model memiliki kasus penggunaan yang cocok untuk membantu penyelesaian masalah tertentu baik dari sisi pemampatan ukuran, penyimpanan citra dan teknik pengenalan citra. Pada makalah ini, akan dipakai dua model untuk melakukan eksperimen dalam pemrosesan citra yang dilakukan yaitu model warna RGB dan HSV.

Model warna RGB digunakan untuk *display* atau tampilan pada layar komputer. Model warna RGB merupakan model warna dengan nilai percampuran merah, hijau, dan biru yang dapat direpresentasikan dalam vektor warna 3 dimensi (3D) berbentuk kubus. Model warna RGB dibakukan pada tahun 1931 oleh CIE (Commission International de l'Eclairage) atau *International Lighting Committee*. CIE menstandarkan panjang gelombang warna kontinu untuk RGB adalah R = 700nm; G = 546,1 nm; dan B = 435,8 nm. Pada gambar 2.2 dapat dilihat ilustrasi spektrum model warna RGB dalam bentuk kubus.



Gambar 2.2 Spektrum Model Warna RGB

Pada gambar 2.2, nilai (0,0,1) adalah nilai normalisasi dari nilai (0, 0, 255) pada citra dengan kedalaman 24-bit. Transformasi citra dengan model warna RGB menjadi citra *grayscale* yaitu citra dengan model warna hitam-putih dengan rentang 0-255 juga dapat dilakukan dengan mudah yaitu dengan mengubah nilai setiap *pixel* pada citra dengan rumus sebagai berikut.

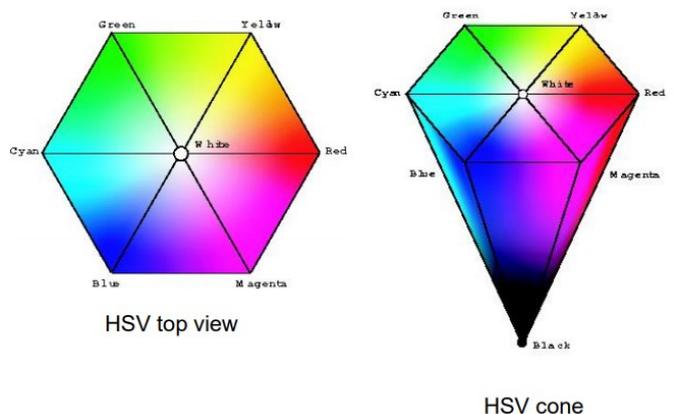
$$Y = 0.299R + 0.587G + 0.144B \quad (1)$$

Citra dengan mode warna *grayscale* nantinya akan dapat diolah untuk dilakukan pendeteksian tepi dan segmentasi citra dalam pengenalan citra.

Model warna HSV merupakan model warna yang terdiri dari komponen *Hue*, *Saturation* dan *Value* yang masing-masing memiliki arti sebagai berikut.

1. *Hue*, jenis warna sebenarnya (merah, ungu, dan sebagainya), dengan rentang nilai 0 sampai 2π
2. *Saturation*, tingkat kemurnian warna, dengan rentang nilai [0,1]
3. *Value*, nilai kecerahan sebuah warna, dengan rentang nilai [0,1]

Representasi dari model warna HSV adalah sebuah vektor warna yang berbentuk prisma segi enam. Pada model warna HSV, terdapat pemisahan yang jelas atas satu warna dengan warna lain. Setiap warna disatukan dalam satu irisan vertikal terhadap prisma yang menghasilkan rentang warna dari terang ke gelap jika dilihat dari atas ke bawah. Pada gambar 2.3 dapat dilihat ilustrasi spektrum model warna HSV dalam bentuk prisma segi enam.



Gambar 2.3 Spektrum Model Warna HSV

C. Deteksi Tepi

Salah satu fitur utama dalam pengenalan citra adalah tepi. Tepi atau *edge* merupakan bagian pinggiran dari sebuah objek dalam citra yang dihasilkan dari perbedaan intensitas warna yang drastis dari satu *pixel* ke *pixel* tetangganya. Deteksi tepi adalah proses pencarian bagian atau posisi dalam citra yang mengalami perubahan intensitas warna (gradien) secara drastis.

Berbagai metode untuk mendeteksi tepi telah banyak ditemukan dan digunakan. Umumnya, seluruh metode ini menggunakan metode konvolusi. Konvolusi merupakan sebuah proses perkalian nilai matriks warna pada citra dengan suatu matriks pengali yang disebut dengan kernel. Masing-masing metode pendeteksi tepi memiliki hasil konvolusi yang berbeda-beda satu sama lain. Perbedaan pada hasil ini diakibatkan oleh perbedaan kernel pada masing-masing metode pendeteksi tepi.

Pada makalah ini, metode deteksi tepi yang akan digunakan adalah metode dengan operator Canny. Nama operator Canny diambil dari penemunya yang bernama John Canny yang mengusulkan pendeteksian tepi dengan langkah-langkah sebagai berikut (Rinaldi, 2021).

1. Haluskan citra I dengan penapis Gaussian (kernel gaussian tidak dijelaskan disini): $G * I$
2. Hitung gradien setiap $pixel$ dengan salah satu dari operator seperti operator Sobel, Roberts, Prewitt, dll.
3. Jika nilai mutlak gradien melebihi nilai ambang T , maka $pixel$ diubah menjadi biner yang menandakan $pixel$ tepi

Metode deteksi tepi dengan operator Canny dipilih karena hasilnya yang berupa $pixel$ biner yang menandakan lokasi tepi dapat langsung dipakai untuk dianalisis dalam pengenalan objek dan segmentasi citra seperti sayur-sayuran. Pada gambar 2.4 dapat dilihat contoh hasil deteksi tepi dengan operator Canny.



Gambar 2.4 Hasil Deteksi Tepi dengan Operator Canny

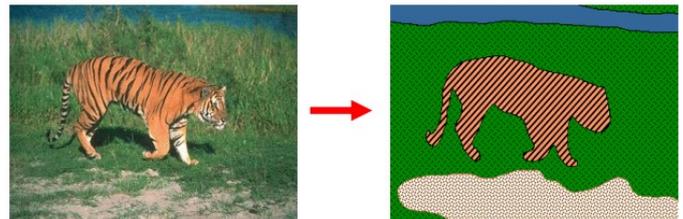
D. Segmentasi Citra

Segmentasi citra adalah proses mempartisi citra digital menjadi beberapa bagian. Tujuan dari segmentasi citra adalah untuk membagi citra menjadi region-region atau objek-objek dan memisahkan objek dari latar belakang. Citra disegmentasi berdasarkan properti yang dipilih seperti kecerahan, warna, tekstur, dan sebagainya. Segmentasi citra membagi citra menjadi sejumlah region yang terhubung, tiap region bersifat homogen berdasarkan properti yang dipilih.

Segmentasi citra merupakan tahapan sebelum melakukan *image/object recognition* atau pengenalan objek. Segmentasi citra berfungsi untuk menyederhanakan dan mengubah

penyajian gambar menjadi sesuatu yang lebih bermakna sehingga lebih mudah untuk dianalisis. Segmentasi citra membagi citra menjadi beberapa bagian berbeda. Bagian-bagian yang berbeda ini menunjukkan kesatuan objek berbeda yang ada dalam suatu citra. Bagian-bagian ini juga dapat menginformasikan karakteristik dari sebuah objek.

Segmentasi citra mengonversikan sebuah citra ke dalam kumpulan region $pixel$ yang direpresentasikan dalam sebuah *mask* atau label. Dengan membagi citra ke dalam segmen-segmen, pemrosesan keseluruhan gambar untuk memproses sebagian gambar tidak lagi diperlukan. Pemrosesan dapat dilakukan pada segmen tertentu saja. Pada gambar 2.5 dapat dilihat contoh segmentasi citra pada citra foto harimau.



Gambar 2.5 Hasil Segmentasi Citra pada Citra Foto Harimau

Segmentasi citra dilakukan dengan cara mencari kedekatan atau kemiripan antar $pixel$. Pendekatan yang umum digunakan dalam mencari kemiripan antar $pixel$ adalah *region growing*, *split and merge*, *clustering*, dan *thresholding*. Dalam makalah ini, pendekatan yang dipakai untuk mencari kemiripan antar $pixel$ adalah *thresholding* atau pengambangan. Teknik segmentasi citra dengan *thresholding* dibagi menjadi 3 yaitu:

1. *Global thresholding*, nilai ambang bergantung pada keseluruhan nilai-nilai $pixel$
2. *Local thresholding*, nilai ambang bergantung pada $pixel$ - $pixel$ bertetangga, hanya untuk sekelompok $pixel$ saja
3. *Adaptive thresholding*, nilai ambang berubah secara dinamis bergantung pada perubahan pencahayaan di dalam citra

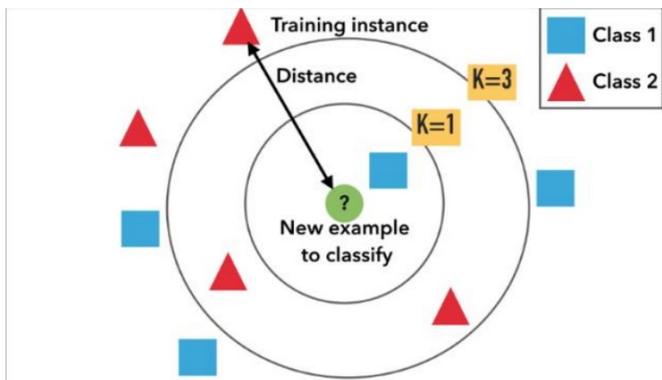
Pada makalah ini, teknik segmentasi citra yang digunakan adalah teknik segmentasi citra dengan pendekatan *global thresholding*, yaitu pengambangan dengan metode Otsu, yang bertujuan untuk menemukan nilai optimal untuk *global threshold*.

E. K-Nearest Neighbors

Algoritma k-nearest neighbors (KNN) adalah algoritma yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (*train data sets*), yang diambil dari k tetangga terdekatnya (*nearest neighbors*). Dengan k merupakan banyaknya tetangga terdekat.

Algoritma k-nearest neighbors melakukan klasifikasi dengan proyeksi data pembelajaran pada ruang berdimensi banyak. Ruang ini dibagi menjadi bagian-bagian yang merepresentasikan kriteria data pembelajaran. Setiap data

pembelajaran direpresentasikan menjadi titik-titik c pada ruang dimensi banyak. Pada gambar 2.6 dapat dilihat ilustrasi klasifikasi dengan algoritma k-nearest neighbors.



Gambar 2.6 Klasifikasi dengan Algoritma k-Nearest Neighbors

Algoritma k-nearest neighbors merupakan sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. Algoritma k-nearest neighbors termasuk dalam kategori *supervised learning*, dimana hasil *query instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam tetangga terdekatnya. Adapun algoritma untuk k-nearest neighbors adalah sebagai berikut.

1. Menentukan parameter k (jumlah tetangga paling dekat)
2. Menghitung jarak antara objek terhadap data training yang diberikan, misalnya dengan kuadrat jarak *euclidean*
3. Mengurutkan hasil no 2 secara berurutan dari nilai tinggi ke rendah
4. Mengumpulkan kategori Y (klasifikasi nearest neighbors berdasarkan nilai k)
5. Dengan menggunakan kategori tetangga terdekat (*nearest neighbors*) yang paling mayoritas maka dapat diprediksikan kategori objek

III. IMPLEMENTASI

Dalam membuat sistem pengenalan sayur-sayuran dengan menggunakan metode *image processing* yaitu segmentasi citra dan metode klasifikasi berbasis *machine learning* yaitu k-Nearest Neighbors (KNN) terdapat beberapa tahapan antara lain:

1. Akuisisi Citra
2. Segmentasi Citra
3. Ekstraksi Fitur
4. Klasifikasi Objek

Dalam implementasi, digunakan kaskas matlab dengan bantuan *toolbox* berupa *image processing toolbox* dan *statistics*

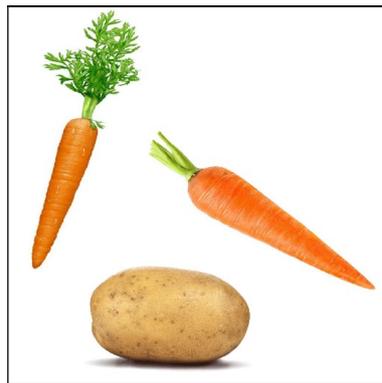
and *machine learning toolbox*. Penggunaan *toolbox* akan dijelaskan pada subbab terkait.

A. Akuisisi Citra

Pada tahap akuisisi citra, citra asli atau masukan yang akan dilakukan pengenalan sudah bersifat statis dan berekstensi jpg atau png. Pada tahap ini, citra masukan sudah siap untuk dibaca dan diolah. Pembacaan citra masukan dibantu dengan fungsi *imread* pada kaskas matlab seperti pada perintah berikut.

```
image = imread(image_sayur);
```

Pada gambar 3.1 dapat dilihat hasil pembacaan citra *image* ketika ditampilkan menggunakan fungsi *imshow* pada kaskas matlab.

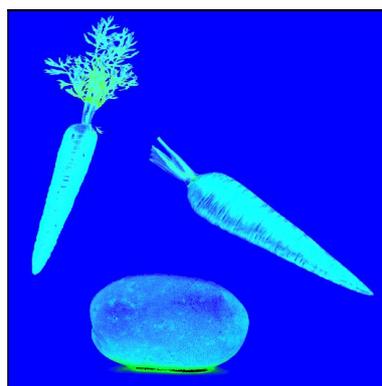


Gambar 3.1 Citra Asli

Pada gambar 3.1, citra asli masih dalam model warna RGB. Citra asli akan dilakukan konversi ke dalam model warna HSV untuk diambil komponen warnanya dan dilakukan segmentasi citra. Konversi citra asli ke model warna RGB dibantu dengan fungsi *rgb2hsv* pada kaskas matlab.

```
hsv_image = rgb2hsv(image);
```

Pada gambar 3.2 dapat dilihat hasil atau tampilan citra *hsv_image* ketika ditampilkan menggunakan fungsi *imshow* pada kaskas matlab.



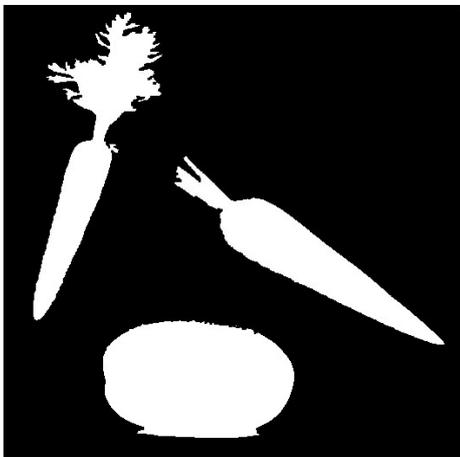
Gambar 3.2 Citra HSV

B. Segmentasi Citra

Pada tahap segmentasi citra, citra akan dilakukan segmentasi berdasarkan nilai-nilai intensitas pixel dan sebuah nilai ambang. Hasil segmentasi citra nantinya adalah sebuah citra biner. Teknik segmentasi citra yang akan digunakan adalah teknik segmentasi citra dengan pendekatan *global thresholding* yaitu pengambangan dengan metode Otsu. Pengambangan dengan metode Otsu bertujuan untuk menemukan nilai optimal untuk *global threshold*. Penentuan nilai *global threshold* atau ambang batas keseluruhan dengan metode Otsu dibantu dengan fungsi *graythresh* dan segmentasi citra dibantu dengan fungsi *im2bw* pada kaskas matlab seperti pada perintah berikut.

```
threshold = graythresh(hsv_image)
bw_image = im2bw(hsv_image, threshold)
```

Pada gambar 3.3 dapat dilihat hasil atau tampilan citra *bw_image* ketika ditampilkan menggunakan fungsi *imshow* pada kaskas matlab.



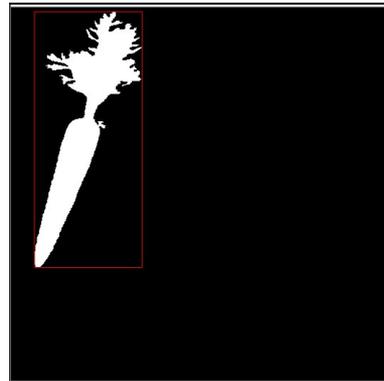
Gambar 3.3 Citra Hasil Segmentasi

C. Ekstraksi Fitur

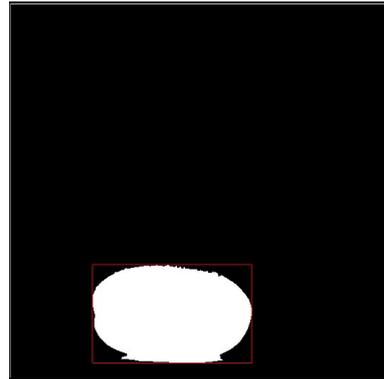
Pada tahap ekstraksi citra, tiap objek dalam citra yang sudah disegmentasi akan diambil dan dilakukan ekstraksi untuk fitur-fiturnya. Hasil ekstraksi fitur nantinya akan digunakan untuk tahapan klasifikasi objek. Sebelum melakukan ekstraksi fitur pada objek sebelumnya perlu dilakukan pemisahan masing-masing objek dari citra awal agar tiap objek dapat dilakukan ekstraksi fitur masing-masing. Untuk melakukan pemisahan masing-masing objek pada citra dibantu dengan fungsi *bwconncomp* pada kaskas matlab seperti pada perintah berikut.

```
extract_image = bwconncomp(bw_image)
```

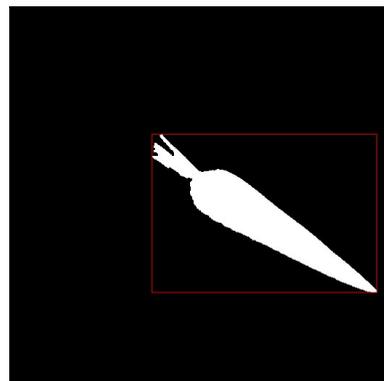
Pada gambar 3.4, 3.5, dan 3.6 dapat dilihat hasil pemisahan sejumlah objek dari citra awal yaitu *bw_image* ketika ditampilkan menggunakan fungsi *imshow* pada kaskas matlab.



Gambar 3.4 Citra Pemisahan Objek 1



Gambar 3.5 Citra Pemisahan Objek 2



Gambar 3.6 Citra Pemisahan Objek 3

Setelah tiap objek pada citra awal telah dipisahkan, maka ekstraksi fitur pada masing-masing objek dapat dilakukan. Pada ekstraksi fitur terdapat 4 fitur utama yang dipilih yaitu warna, bentuk, dan ukuran (panjang, lebar) sayur-sayuran (objek). Untuk melakukan ekstraksi fitur untuk masing-masing objek pada citra dibantu dengan fungsi *bwconncomp* pada kaskas matlab seperti pada perintah berikut.

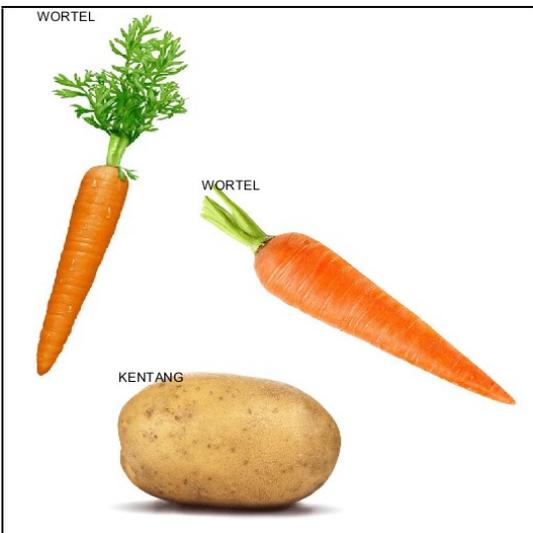
```
object = regionprops(extract_image,
    'Area', 'Perimeter', 'BoundingBox',
    'Eccentricity', 'Centroid', 'FilledImage')
feature = [object.color, object.shape,
    object.length, object.width]
```

D. Klasifikasi Objek

Pada tahap klasifikasi objek, tiap objek dalam citra yang sudah diekstrak fitur-fiturnya akan diolah dan dilakukan klasifikasi berbasis *machine learning* untuk pengenalan objek. Hasil klasifikasi nantinya akan digunakan untuk menentukan objek termasuk kedalam kategori atau kelas objek tertentu. Pada makalah ini, klasifikasi yang digunakan adalah klasifikasi dengan algoritma k-nearest neighbors. Untuk melakukan klasifikasi dengan algoritma k-nearest neighbors pada tiap objek dalam citra dibantu dengan fungsi *bwconncomp* pada kaskas matlab seperti pada perintah berikut.

```
load('featureVectors.mat');
train = ClassificationKNN.fit(
    vector,Y,'NumNeighbors',10);
[label,POSTERIOR, score] = predict(train,feature);
```

Pada gambar 3.7 dapat dilihat hasil klasifikasi sejumlah objek dari citra awal. Pada citra di gambar 3.7, telah diberi label yang menunjukkan kategori atau klasifikasi objek.



Gambar 3.7 Citra Hasil Klasifikasi Objek

Lalu, untuk menampilkan jumlah masing-masing objek dalam citra masukan digunakan perintah sebagai berikut.

```
fprintf('Total Wortel = %i\n',jumlahWortel)
fprintf('Total Kentang = %i\n',jumlahKentang)
```

Pada gambar 3.8 dapat dilihat tampilan yang menampilkan jumlah masing-masing objek dalam citra.

```
Command Window
Total Wortel = 2
Total Kentang = 1
```

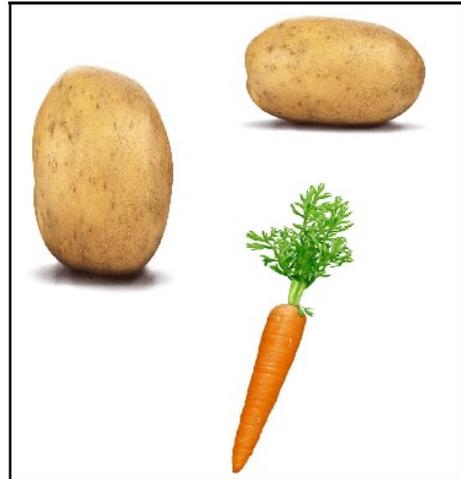
Gambar 3.7 Jumlah Objek pada Citra

IV. HASIL EKSPERIMEN

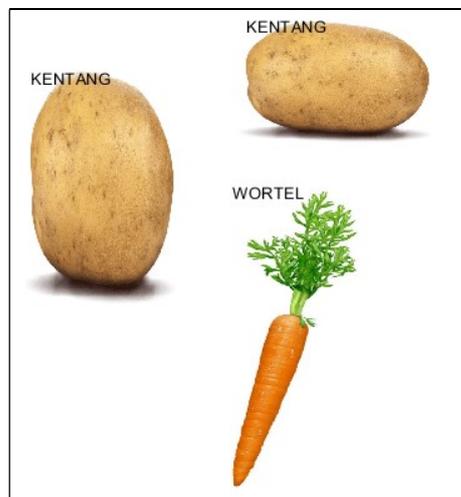
Eksperimen pada makalah ini dilakukan pada dua buah gambar. Hasil eksperimen menunjukkan bahwa implementasi sistem pengenalan sayur-sayuran yang dibangun sudah dapat melakukan pengenalan dan perhitungan jumlah sayur-sayuran seperti kentang dan wortel berdasarkan masukan citra digital. Eksperimen hanya dilakukan pada *image* atau foto yang diam (statis) dan terbatas pada 2 jenis sayuran yaitu kentang dan wortel. Pada subbab dibawah akan dijelaskan lebih lanjut mengenai hasil eksperimen yang sudah dilakukan.

A. Kasus Uji 1

Citra masukan:



Citra keluaran:

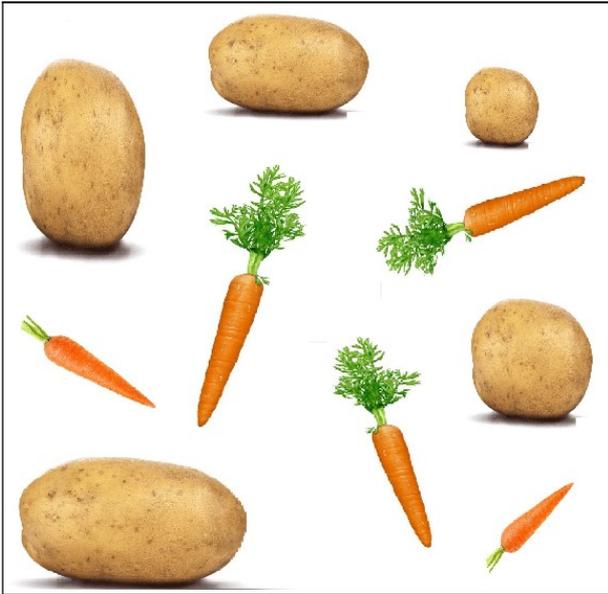


Output program:

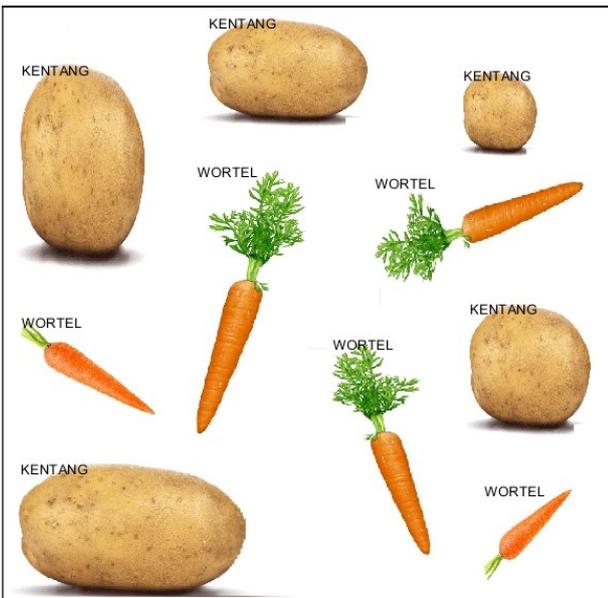
```
Command Window
Total Wortel = 1
Total Kentang = 2
```

B. Kasus Uji 2

Citra masukan:



Citra keluaran:



Output program:

```
Command Window
Total Wortel = 5
Total Kentang = 5
```

KESIMPULAN

Metode segmentasi citra merupakan salah satu metode *image processing* yang dapat dipakai dalam sistem pengenalan sayur-sayuran. Penggunaan segmentasi citra dalam sistem pengenalan sayur-sayuran dengan klasifikasi berbasis *machine learning* seperti algoritma k-nearest neighbors (KNN) dapat melakukan pengenalan objek sayur-sayuran seperti kentang dan wortel serta menghitung jumlah masing-masing jenis sayuran di dalam citra masukan. Penggunaan fitur-fitur seperti warna, bentuk dan ukuran sayuran terbukti cukup optimal. Sistem pengenalan sayur-sayuran juga dapat digunakan dalam melakukan pengenalan untuk lebih banyak jenis sayur-sayuran selain kentang dan wortel. Secara umum, sistem pengenalan sayur-sayuran dapat digunakan untuk mengenali, mengidentifikasi, mengklasifikasikan dan menghitung jumlah sayur-sayuran dalam suatu citra digital.

REFERENCES

- [1] Altman, Naomi S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". *The American Statistician*.
- [2] Azriel Rosenfeld (1969). "Picture Processing by Computer", New York: Academic Press, 1969
- [3] Belongie, Serge, et al. (1998). "Color-and texture-based image segmentation using EM and its application to content-based image retrieval". *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998.
- [4] Cover, Thomas M.; Hart, Peter E. (1967). "Nearest neighbor pattern classification". *IEEE Transactions on Information Theory*.
- [5] Gonzalez, Rafael (2018). "Digital image processing".
- [6] Linda G. Shapiro and George C. Stockman (2001). "Computer Vision".
- [7] Rinaldi Munir (2021). "Bahan Kuliah IF4073 Interpretasi dan Pengolahan Citra Sem. II Tahun Ajaran 2020/2021". Disadur dari <http://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2019-2020/citra19-20.htm>

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, 21 Mei 2021



Nando Rusrin Pratama - 13517148