

MENENTUKAN LUAS OBJEK CITRA DENGAN TEKNIK SEGMENTASI BERDASARKAN WARNA PADA RUANG WARNA HSV

Determining the Image Object Area Using Color-Based Segmentation Technique in HSV Color Space

M. Oni^[1], Bulkis Kanata^[1], Dwi Ratnasari^[1]

^[1] Jurusan Teknik elektro, Fakultas Teknik, Universitas Mataram, Nusa Tenggara Barat, Indonesia
Email : muhammadoni22@gmail.com¹; ujikanata@unram.ac.id²; dwi.ratnasari@unram.ac.id³

ABSTRAK

Teknik segmentasi citra berdasarkan warna merupakan salah satu pendekatan yang digunakan untuk mengekstraksi informasi pada citra digital. Pada penelitian ini, penentuan luas objek citra dilakukan dengan memanfaatkan teknik segmentasi citra berdasarkan warna pada ruang warna HSV. Ada tiga objek citra yang ditentukan luasnya yakni objek segitiga, persegi, dan lingkaran. Luas objek dihitung dengan mengalikan luas asli background citra dengan perbandingan proporsi antara jumlah piksel objek dengan total piksel background. Nilai threshold untuk filter warna HSV ditentukan dengan menggunakan dua pendekatan yang berbeda yakni metode trial-and-error dan metode multi-otsu thresholding. Hasil dari penelitian ini menunjukkan bahwa kinerja sistem penentuan luas objek citra dengan metode trial-and-error diperoleh rata-rata persentase keberhasilan untuk semua objek adalah 98,65%, sedangkan pada metode multi-otsu thresholding adalah 98,69%

Kata kunci : Segmentasi warna, hsv, multi-otsu, luas objek.

ABSTRACT

The color image segmentation technique is one of many approaches used to extract information from an image. In this study, the calculation of the image object area was carried out using the color image segmentation technique in HSV color space. There are three objects that are investigated according of its area i.e triangle, square, and circle. The area of the objects calculated by multiply the image background original area with the proportion difference between the total number of object pixels and the sum of background pixels. The threshold value for the HSV filter was determined using two different approaches, which are the trial-and-error method and the multi-otsu thresholding method. The results of this study shows that the performance of the image object area determination system using the trial-and-error method obtained an average percentage of success for all objects is 98.65%, while the multi-otsu thresholding method is 98.69%.

Keywords: Color segmentation, hsv, multi-otsu, object area.

PENDAHULUAN

Computer vision seringkali dimanfaatkan untuk mempermudah pekerjaan manusia dengan cara mengaplikasikannya ke dalam berbagai macam bentuk terapan dan fungsi. Salah satunya ialah mendeteksi sebuah objek di dalam sebuah citra digital kemudian mengekstrak informasi yang terkandung di dalamnya kemudian diproses sehingga dapat menghasilkan *output* yang bernilai.

Segmentasi objek pada citra digital merupakan bagian paling pokok dalam *computer vision*. Terdapat beberapa metode segmentasi objek, salah satunya adalah

teknik segmentasi objek berdasarkan warna. Teknik segmentasi ini bekerja dengan mendeteksi piksel-piksel yang mengandung warna tertentu, sehingga dari kumpulan piksel-piksel dengan warna yang sama dan memiliki hubungan ketetanggaan terbentuklah sebuah objek yang tersegmentasi.

Citra berwarna memungkinkan segmentasi citra yang lebih andal dibandingkan citra *grayscale* [1]. Segmentasi citra berdasarkan warna dilakukan dengan pendekatan *region-based*. Teknik yang tepat untuk digunakan adalah dengan melihat histogram citra berwarna, dengan begitu

dapat dilihat jumlah kumpulan piksel-piksel yang menyusun sebuah area pada citra.

Terdapat berbagai macam ruang warna berbeda yang banyak digunakan untuk merepresentasikan warna saat ini. Ruang warna L^*A^*B dan HSV merupakan dua ruang warna yang sering dipilih [2]. Dalam studi komparasi kinerja antara ruang warna HSV dengan L^*A^*B menunjukkan bahwa HSV memiliki kinerja yang lebih baik [3]. Studi serupa dilakukan untuk membandingkan kinerja antara ruang warna HSV dengan RGB menunjukkan ruang warna HSV memberikan hasil yang lebih baik [4]. Hal ini karena ruang warna HSV lebih baik terhadap perubahan pencahayaan [5]. RGB tidak cukup baik untuk pemrosesan warna karena RGB sensitif terhadap pencahayaan dan perubahan warna dalam ruang warnanya tidak linier [6].

[7] Melakukan segmentasi warna citra dengan deteksi warna HSV untuk mendeteksi objek. Citra RGB diubah menjadi citra HSV kemudian dilakukan segmentasi dengan cara menerapkan filter warna HSV. Filter warna HSV dibuat dengan menentukan nilai warna acuan dan nilai toleransi pada masing-masing kanal HSV. Piksel-piksel yang tidak melewati filter warna HSV diberikan warna hitam, sedangkan yang melewati filter warna adalah piksel-piksel yang nantinya akan membentuk *blob-blob*. Dari *blob-blob* ini terbentuklah area yang tersegmentasi, selanjutnya dilakukan deteksi objek dan perhitungan jumlah objek yang dideteksi. Dalam penelitiannya juga dilakukan uji coba untuk menentukan kombinasi nilai toleransi pada *hue*, *saturation*, dan *value* yang cocok dan tepat.

[8] Melakukan penelitian untuk menghitung luas objek sebuah citra RGB yang diambil dari *USB webcam* yang terhubung pada PC. Penentuan luas objek dilakukan dengan proses segmentasi yang memanfaatkan metode deteksi tepi Canny. Untuk menentukan luas objek, dilakukan beberapa langkah diantaranya seperti *threshold*, deteksi tepi dan menghilangkan objek-objek yang tidak diinginkan. Selanjutnya menghitung jumlah piksel pada objek yang tersegmentasi dan menghitung luas objek yang sebenarnya dengan rumus perbandingan. Objek yang dihitung luasnya adalah lingkaran, persegi, dan segitiga. Masing-masing objek ditempatkan pada posisi yang berbeda-beda pada setiap sampel gambar. Dari hasil penelitian diperoleh presentase keberhasilan perhitungan luas untuk objek lingkaran 95,5%, persegi 94,98%, dan segitiga sebesar 80,9%.

Dalam penelitian ini dilakukan penentuan luas objek citra dengan memanfaatkan teknik segmentasi warna pada ruang warna HSV. Citra masukan merupakan gambar RGB yang dikonversi ke HSV. Segmentasi warna dilakukan dengan memanfaatkan *threshold* pada masing-masing kanal *Hue*, *Saturation*, dan *Value* sehingga area dengan warna yang termasuk dalam rentang *threshold* adalah area yang tersegmentasi. Pada penelitian ini, area yang akan disegmentasi adalah area *background* pada gambar, maka warna *background* gambar akan ditetapkan pada satu warna tertentu. Area yang tidak termasuk kedalam area yang tersegmentasi merupakan area objek yang akan dihitung ukurannya.

Penentuan ukuran objek dilakukan dengan menghitung jumlah total piksel pada area *background* yang telah tersegmentasi, kemudian mencari jumlah piksel yang tidak tersegmentasi. Dengan rumus perbandingan antara jumlah total piksel objek dengan jumlah total piksel *background*, maka dapat ditemukan ukuran asli dari objek gambar.

KAJIAN PUSTAKA

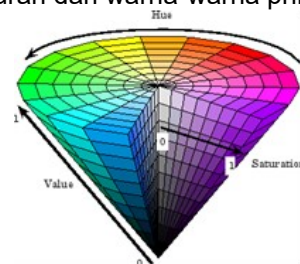
a. Citra Digital

Sebuah citra dapat didefinisikan sebagai sebuah fungsi 2 dimensi, $f(x,y)$, nilai dari f pada setiap pasangan x dan y yang berbeda disebut sebagai intensitas atau level keabuan pada titik tersebut, dengan ketentuan x dan y merupakan koordinat spasial dari sebuah matriks 2 dimensi [9].

Terdapat beberapa jenis citra digital yang sering digunakan dalam penelitian yakni citra warna, citra *grayscale*, dan citra *biner*.

b. Ruang Warna HSV

Ruang warna HSV mendefinisikan warna dalam terminologi *Hue*, *Saturation* dan *Value*. Keuntungan HSV adalah terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia. Sedangkan warna yang dibentuk model lain seperti RGB merupakan hasil campuran dari warna-warna primer.



Gambar 1. Ruang warna HSV [10]

Melalui model Gambar 1, HSV memiliki 3 karakteristik pokok, yaitu *Hue*, *Saturation* dan *Value*.

- *Hue*: menyatakan warna murni, seperti merah, kuning, dan biru yang digunakan untuk menentukan kemerahan (*redness*), kehijauan (*greeness*), dsb.
- *Saturation*: kadang disebut chroma, derajat warna putih yang terkandung dalam suatu warna.
- *Value*: tingkat kecerahan dari suatu warna. Nilainya berkisar antara 0-100 %. Nilai 0 untuk warna tergelap yakni hitam, semakin besar nilai *value* maka warna akan semakin cerah.

c. Segmentasi Citra Digital

Segmentasi citra adalah suatu proses membagi suatu citra menjadi wilayah-wilayah yang homogen [11]. Terdapat dua jenis segmentasi citra berdasarkan cara kerjanya yakni *dividing image space* dan *clustering feature space* [11]. Segmentasi citra berdasarkan warna adalah salah satu segmentasi yang menggunakan pendekatan *region-based*. Pendekatan ini mengidentifikasi setiap piksel kemudian memberikan label yang sama kepada piksel-piksel yang memiliki kesamaan warna. Dari kumpulan piksel-piksel yang memiliki label yang sama, terbentuklah daerah-daerah yang terbagi pada citra yang merupakan hasil dari proses segmentasi.

Berikut merupakan proses segmentasi warna menurut [12]:

- Mengkonversi citra RGB yang akan disegmentasi ke HSV, kemudian menentukan warna HSV acuan dan nilai toleransinya.
- Melakukan filter warna pada citra berdasarkan pada nilai acuan (*T*) dan nilai toleransinya (*tol*). Apabila *x* adalah warna piksel yang ada pada citra, maka warna piksel yang tidak termasuk dalam rentang $T-tol < x < T+tol$ akan diberikan warna hitam.
- Mengkonversi Kembali citra ke RGB, kemudian menampilkan hasil filter.

d. *Multi-Otsu* Thresholding

Metode *multi-otsu thresholding* merupakan sebuah metode yang digunakan untuk memisahkan piksel-piksel pada citra menjadi beberapa kelas yang berbeda, masing-masing kelas memiliki nilai intensitas keabuan yang berbeda pada citra [13]. *Multi-otsu* menentukan beberapa nilai *threshold*, tergantung dari jumlah kelas yang diinginkan.

Apabila jumlah kelas yang diinginkan adalah *n* (integer), maka jumlah *threshold* yang dihasilkan adalah *n-1* nilai *threshold*.

Metode *multi-otsu thresholding* berasal dari *otsu thresholding* yang diperkenalkan pertama kali oleh Nobuyuki Otsu, dalam jurnal ilmiahnya yang berjudul “*A Threshold Selection Method from Gray-Level Histogram*” pada tahun 1979 [14].

Untuk mendapatkan nilai *threshold* dengan metode *otsu thresholding*, ada perhitungan yang harus dilakukan. Langkah awal yang harus dilakukan adalah membuat histogram. Dari histogram dapat diketahui jumlah piksel untuk setiap tingkat keabuan. Tingkat keabuan citra dinyatakan dengan *i* sampai dengan *L*. Level ke *i* dimulai dari 1, yaitu piksel 0. Untuk *L*, maksimal level adalah 256 dengan piksel bernilai 255.

METODE PENELITIAN

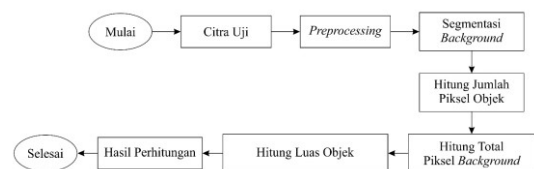
Pada bab ini membahas tentang tahapan penelitian, perancangan dan pembuatan sistem penentuan luas objek pada citra dengan menggunakan teknik deteksi warna.

a. *Studi Literatur*

Studi literatur dilakukan untuk mengumpulkan informasi penunjang untuk mengerjakan dan menyelesaikan penelitian ini. Adapun informasi yang dibutuhkan dalam penelitian ini berkaitan tentang teknik segmentasi berdasarkan warna pada ruang warna HSV.

b. *Perancangan Sistem*

Dalam proses perancangan sistem dilakukan penelitian dan analisis tentang sistem yang akan dibangun. Dalam penelitian ini, yang akan dibangun adalah sistem yang akan digunakan untuk menentukan luas objek pada citra digital menggunakan teknik deteksi warna. Berikut diagram sistem yang akan dibangun dapat dilihat pada Gambar 2.



Gambar 1. Diagram sistem penentuan luas objek pada citra

Citra uji merupakan kumpulan citra-citra yang akan digunakan sebagai bahan uji coba pada sistem penentuan luas objek dalam citra dengan menggunakan deteksi warna. Citra uji diperoleh dengan beberapa tahap yakni

dimulai dengan membuat gambar sintetik melalui perangkat lunak grafis komputer kemudian dicetak, selanjutnya cetakan-cetakan gambar tersebut diambil fotonya menggunakan kamera *smartphone*. Citra-citra yang menjadi bahan uji ini terdiri dari gambar-gambar objek lingkaran, persegi, dan segitiga dengan ukuran resolusi gambar, kombinasi warna, dan tata letak yang berbeda-beda di setiap citra uji. Kombinasi tata letak objek yang dimaksud adalah setiap gambar objek berada di 3 posisi yang berbeda pada setiap sampel, yakni kiri-atas, tengah, dan kanan-bawah dari kertas cetak.

Preprocessing adalah tahapan awal dalam pengolahan citra digital. *Preprocessing* merupakan suatu proses untuk mempersiapkan citra digital agar dapat diproses lebih baik pada tahapan selanjutnya. Pada penelitian ini terdapat beberapa tahapan yang dilakukan dalam *preprocessing*. Tahapan-tahapan yang dilakukan antara lain:

- *Cropping*

Pada tahap ini dilakukan *cropping* yang berfungsi untuk memotong gambar, menghapus bagian yang tidak diperlukan agar pemrosesan gambar lebih optimal.

- *Resizing*

Pada tahap ini citra diubah ukurannya menjadi 3 variasi untuk melihat pengaruh resolusi citra terhadap efektifitas sistem yang akan dibangun.

- Konversi ke HSV

Citra uji yang digunakan merupakan citra RGB, maka untuk kebutuhan sistem dilakukan konversi citra dari RGB ke HSV.

Proses segmentasi berfungsi untuk mengelompokkan piksel-piksel yang serupa pada citra sehingga terbentuk beberapa daerah-daerah pada citra. Dalam penelitian ini, citra disegmentasi berdasarkan warna, dan filter yang digunakan adalah filter yang menggunakan nilai acuan dan nilai toleransi sebagai *threshold*. Bagian *background* diberikan warna hijau/kehijauan, sedangkan objek diberikan warna selain dari warna hijau/kehijauan. Dengan cara seperti ini, memungkinkan untuk membuat filter pada sistem hanya pada satu warna saja, dan warna objek yang bervariasi. Oleh karena itu, bagian awal yang disegmentasi adalah bagian *background* pada citra. Sedangkan area objek adalah bagian-bagian yang selain dari area yang tersegmentasi. Selanjutnya dilakukan *reverse*, yakni area *background* diberikan warna hitam sedangkan area objek diberikan warna putih. Pada hasil segmentasi, akan ada *blob-blob* kecil di area *background* ataupun

area objek, hal ini dikarenakan warna yang tidak sepenuhnya seragam baik dari sisi *hue*, *saturation* dan/atau *value*. Maka untuk keperluan optimalisasi hasil hitung, semua *blob-blob* kecil yang ukurannya kurang dari jumlah piksel minimum akan dihapus dengan menggunakan perintah *bwareaopen* di matlab.

Tahap berikutnya dilakukan perhitungan jumlah piksel pada objek. Piksel-piksel pada area objek adalah yang berwarna putih. Kemudian dilakukan perhitungan total piksel *background*. Total piksel *background* didapatkan dengan menjumlahkan piksel ketiga objek dan piksel pada area *background* yang terlihat. Mengingat sebagian area *background* tertutup oleh ketiga objek, namun jumlah piksel *background* yang tertutup jumlahnya sama dengan total piksel dari ketiga objek.

Perhitungan luas objek dilakukan dengan menggunakan rumus perbandingan [8] sebagai berikut:

$$\text{Luas Objek} = \frac{\sum PO}{\sum PC} \times Yb \dots\dots\dots(1)$$

Keterangan:

$\sum PO$ = Jumlah piksel objek

$\sum PC$ = Total piksel *background* (jumlah piksel *background* yang terlihat ditambah total piksel ketiga objek)

Yb = Luas asli *background*

Tingkat efektifitas penentuan luas objek oleh sistem akan dihitung sebagai bahan evaluasi hasil penelitian. Masing-masing objek lingkaran, persegi, dan segitiga, dihitung rata-rata luas yang diperoleh dari ketiga posisi berbeda. Hasil hitung rata-rata luas dari setiap objek kemudian dibandingkan perbedaannya dengan luas objek yang sebenarnya, sehingga didapatkan nilai perbedaan antara luas objek hasil penentuan oleh sistem dengan luas objek yang sebenarnya. Dari nilai perbedaan yang diperoleh, maka selanjutnya adalah menghitung persentase perbedaannya. Berikut adalah persamaan yang digunakan untuk menghitung persentase perbedaan tersebut [8]:

$$\text{Persentase Perbedaan} = \frac{|\bar{x} - Y|}{Y} \times 100\% \dots\dots(2)$$

Keterangan:

\bar{x} = Rata-rata luas objek oleh sistem

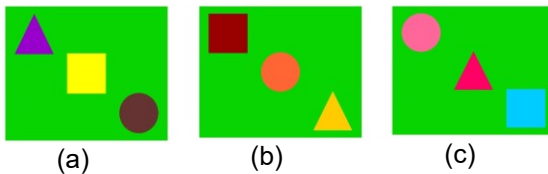
Y = Luas asli objek

HASIL DAN PEMBAHASAN

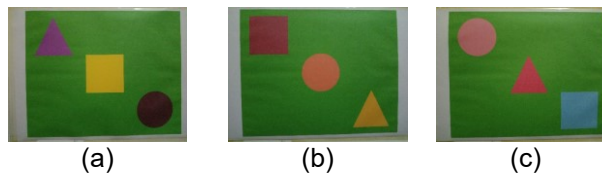
Pada bab ini akan dibahas mengenai hasil penentuan luas objek citra dengan menggunakan teknik deteksi warna.

a. Pengumpulan Citra Uji

Proses pengumpulan citra uji diawali dengan pembuatan sampel citra sintetik dengan kombinasi warna berbeda. Tata letak objek segitiga, persegi, dan lingkaran masing-masing divariasikan pada posisi kiri-atas, tengah dan kanan-bawah, maka dihasilkan tiga buah citra sintetik seperti yang dapat dilihat pada Gambar 3. Langkah selanjutnya adalah mencetak citra sintetik pada lembar kertas A4, baru kemudian dilakukan pengambilan gambar dengan menggunakan kamera *smartphone* dengan resolusi 13 megapiksel. Hasil pengambilan gambar dengan kamera dapat dilihat pada Gambar 4. Ukuran asli *background* pada citra adalah 20 cm x 25 cm. Untuk ukuran asli objek-objek pada citra seperti segitiga memiliki tinggi 6 cm dan panjang alas 6 cm, objek persegi memiliki panjang sisi 6 cm, dan objek lingkaran memiliki diameter 6 cm.



Gambar 3. (a) Citra sintetik A, (b) Citra sintetik B, (c) Citra sintetik C

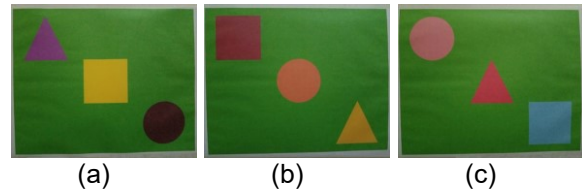


Gambar 4. Hasil foto dengan kamera *smartphone*. (a) Citra A, (b) Citra B, (c) Citra C

Preprocessing

- Cropping dan resizing

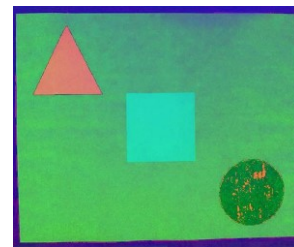
Pada tahap *preprocessing* ini dilakukan pemotongan bagian citra untuk meminimalkan bagian yang tidak diperlukan dan perubahan ukuran citra menjadi 3 variasi ukuran. Dengan begitu total citra uji adalah sebanyak 9 citra. *Cropping* dan *resizing* dilakukan menggunakan bantuan perangkat lunak pengolah gambar dan hasilnya dapat dilihat pada Gambar 5.



Gambar 5. Hasil *cropping* dan *resizing*. (a) Citra A, (b) Citra B, (c) Citra C

- Konversi citra RGB ke HSV

Citra uji yang merupakan citra RGB dikonversi menjadi citra HSV untuk kebutuhan sistem. Untuk dapat melakukan konversi citra RGB ke HSV digunakan fungsi matlab yakni `rgb2hsv()`, citra hsv hasil konversi dapat dilihat pada Gambar 6.



Gambar 6. Hasil konversi citra A ukuran 500 x 417 dari RGB ke HSV

Kode perintah matlab

```
a=imread(filename);
b_hsv=rgb2hsv(a);
```

b. Segmentasi Background

Untuk dapat menemukan *background* pada citra, maka perlu dilakukan segmentasi. Pada penelitian ini, sebuah filter warna HSV dimanfaatkan untuk melakukan segmentasi *background* pada citra yang memiliki warna kehijauan. Untuk membuat filter warna ini diperlukan nilai batas atas dan batas bawah (*threshold*) pada masing-masing kanal *hue*, *saturation*, dan *value* yang merepresentasikan warna pada *background*. Untuk menemukan nilai-nilai *threshold* yang tepat, dilakukan 2 macam pendekatan yakni melalui metode *trial-and-error* dan metode multi-otsu *thresholding*. Kemudian akan dilakukan perbandingan kinerja kedua metode ini terhadap efektifitas sistem yang dibuat.

Proses pencarian *thresholding* dengan kedua metode ini hanya dilakukan pada kanal *hue*, karena pada kanal inilah yang mewakili nilai warna dari sebuah piksel. Jika pada kanal *saturation* dan *value*, yang perlu dihindari adalah nilainya yang rendah. Misal warna kehijauan yang memiliki nilai *saturation* yang

rendah akan menjadi warna abu hingga putih, sedangkan apabila memiliki nilai *value* yang rendah akan menjadi warna gelap hingga hitam sepenuhnya. Beda halnya dengan kanal *hue*, setiap rentang nilai tertentu mewakili sebuah warna seperti kemerahan, kehijauan, kebiruan, dan lainnya. Pada sistem yang dibuat, nilai *saturation* dan *value* ditetapkan pada rentang 40 sampai 100, karena pada rentang ini diasumsikan sebuah piksel masih termasuk sebagai anggota dari warna *hue* yang dipilih.

Perlu diingat bahwa pada laporan penelitian ini, skala yang ditampilkan untuk penyebutan nilai *hue*, *saturation*, dan *value* adalah masing-masing dari rentang 0-360 untuk nilai *hue*, 0-100 untuk nilai *saturation*, serta 0-100 untuk nilai *value*. Namun pada penerapannya di matlab, nilai *hue*, *saturation*, dan *value* semuanya pada skala 0 - 1. Maka apabila disebutkan misalnya nilai *threshold hue* adalah 90 sampai 180, maka nilai yang digunakan di matlab yang sebenarnya adalah dengan membaginya 360, yakni

$$\frac{90}{360} = 0,25 \text{ dan } \frac{180}{360} = 0,5. \text{ Dengan begitu}$$

nilai *threshold hue* yang digunakan pada matlab adalah 0,25 sampai 0,5.

1. Metode trial-and-error.

Metode *trial-and-error* adalah metode yang dilakukan dengan cara coba-coba kemudian melakukan koreksi berdasarkan hasil percobaan yang didapatkan sebelumnya, sampai mendapatkan hasil yang dinilai pas. Dalam hal menemukan *thresholding* yang pas, mula-mula dilihat histogram dari kanal *hue* pada citra uji yang berupa citra HSV. Kemudian berdasarkan asumsi dilakukan penentuan nilai-nilai *threshold background* citra. Setelah itu dilakukan pengujian dengan menerapkan nilai-nilai *threshold* pada filter warna dan dilihat hasilnya. Hasil yang dimaksud adalah hasil penentuan luas objek pada citra. Apabila ternyata hasilnya dirasa kurang baik maka proses diulangi lagi dari penentuan nilai-nilai *threshold* berdasarkan asumsi, hingga didapatkan hasil yang cukup baik. Berikut adalah langkah-langkah penentuan nilai-nilai *threshold* dengan metode *trial-and-error*.

Adapun setelah melihat histogram semua citra uji dan ditentukan asumsi nilai *threshold* batas bawah dan batas atas pada tiap-tiap citra, maka dari sana ditentukan satu pasang nilai *threshold* untuk digunakan pada semua citra uji. Penetapan satu pasang *threshold* untuk semua citra uji ini adalah untuk efisiensi

sehingga tidak harus memasukan *threshold* setiap kali pengujian, melainkan hanya perlu menerapkan satu pasang *threshold* saja pada sistem karena memang *background* yang akan disegmentasi memiliki warna yang satu yakni warna kehijauan. Walaupun ada perubahan warna yang disebabkan dari proses pengambilan gambar oleh kamera, perubahan tersebut relatif kecil. Penetapan satu pasang *threshold* ini ditentukan dengan melihat batas terendah dan batas tertinggi yang ditemukan pada proses sebelumnya. Dari Tabel 1 diperoleh batas terendah sebagai batas bawah yakni 75, dan batas tertinggi sebagai batas atas yakni 141.

Tabel 1. Hasil penentuan *threshold* batas bawah dan batas atas citra uji.

No	Nama Citra	Batas Bawah	Batas Atas	Nilai Acuan	Toleransi
1.	Citra A 500	75	140	107,5	32,5
2.	Citra A 1000	75	140	107,5	32,5
3.	Citra A 1200	75	140	107,5	32,5
4.	Citra B 500	75	140	107,5	32,5
5.	Citra B 1000	75	140	107,5	32,5
6.	Citra B 1200	75	140	107,5	32,5
7.	Citra C 500	75	141	108	33
8.	Citra C 1000	75	141	108	33
9.	Citra C 1200	75	141	108	33

Pada akhiran nama citra dalam Tabel 1 terdapat angka 500, 1000, dan 1200, ini mengindikasikan ukuran dari citra uji tersebut. Misalkan 500 untuk ukuran 500 x 417, 1000 untuk ukuran 1000 x 833, dan 1200 untuk ukuran 1200 x 1000. Nilai acuan (T) merupakan nilai tengah dari *threshold* batas bawah (bb) dan batas atas (ba) yang dapat dicari menggunakan persamaan:

$$T = bb + \left(\frac{ba-bb}{2}\right) \text{ atau } T = ba - \left(\frac{ba-bb}{2}\right).$$

Adapun nilai toleransi (tol) merupakan jarak dari nilai *threshold* batas bawah ke nilai acuan atau jarak dari nilai acuan ke *threshold* batas atas yang dapat dicari menggunakan persamaan:

$$tol = ba - T \quad \text{atau} \quad tol = T - bb$$

2. Metode multi-otsu thresholding.

Metode ini berasal dari metode otsu, jika metode otsu mampu secara otomatis menentukan *threshold* yang membagi wilayah *background* dan objek, maka multi-otsu mampu membagi wilayah-wilayah pada citra sebanyak yang ditetapkan sehingga *threshold*

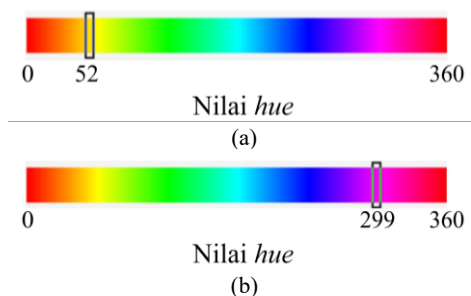
yang didapatkan sebanyak $n-1$, apabila n merupakan jumlah pembagian wilayah pada citra.

Proses pencarian *threshold* dengan multi-otsu ini dilakukan setiap kali pengujian citra, maka setiap citra memiliki nilai ambang yang berbeda-beda [15]. Oleh karena itu, pasangan *threshold* yang digunakan untuk mensegmentasi wilayah *background* pada setiap citra uji adalah berbeda-beda rentang nilainya. Berikut adalah pasangan *threshold* untuk semua citra uji.

Tabel 2. Pasangan *threshold* hasil dari metode multi-otsu untuk semua citra uji.

No	Nama Citra	Batas Bawah	Batas Atas	Nilai Acuan	Toleransi
1.	Citra A 500	89,91	199,48	144,69	54,78
2.	Citra A 1000	94,18	224,91	159,54	65,36
3.	Citra A 1200	88,5	299,23	193,86	105,36
4.	Citra B 500	67,6	126,76	97,18	29,58
5.	Citra B 1000	64,79	130,99	97,89	33,1
6.	Citra B 1200	53,51	128,15	90,83	37,32
7.	Citra C 500	53,56	146,6	100,08	46,52
8.	Citra C 1000	71,89	143,78	107,83	35,94
9.	Citra C 1200	52,15	146,6	99,37	47,22

Pada Tabel 2 pasangan nilai *threshold* hasil dari metode multi-otsu berbeda-beda untuk setiap citra dan rentang nilai juga berbeda-beda. Dalam hal penentuan batas wilayah berdasarkan warna, terdapat ketidak-precisian pada metode multi-otsu. Misal saja nilai *threshold* terendah pada Tabel 2 adalah 52,15, semestinya *threshold* hanya mencakup warna kehijauan saja namun nilai hue ini sudah mencapai warna kejingga-jinggaan. Sedangkan untuk nilai *threshold* tertinggi yakni 299,23, nilai hue ini sudah sampai ke warna keunggu-ungguan seperti ditampilkan pada Gambar 7.



Gambar 7. Representasi warna pada kanal *hue* (a) Nilai 52 adalah representasi warna kejingga-jinggaan, dan (b) 299 untuk warna keunggu-ungguan

Namun jika ditinjau dari cara kerja metode otsu [14] bakal dari metode multi-otsu memang wajar nilai-nilai *threshold* yang dihasilkan berkemungkinan bisa melewati batas daerah warna yang diharapkan, namun tetap mampu memberikan hasil segmentasi wilayah yang relatif sesuai dengan yang diinginkan. Hal ini bisa terjadi karena tidak ada atau minimnya jumlah piksel yang memiliki warna sesuai dengan daerah warna lain yang terlewat. Oleh karena itu, *threshold* yang melewati batas warna kehijauan sampai ke daerah warna tertentu hanya dapat digunakan pada citra yang bersangkutan, sehingga perlu dilakukan proses *thresholding* multi-otsu pada setiap citra uji.

c. Hasil Perhitungan oleh Sistem

Perhitungan luas objek oleh sistem dilakukan dengan memanfaatkan rumus perbandingan seperti yang ditampilkan pada persamaan (1).

1. Hasil perhitungan luas objek.

Setelah mendapatkan informasi jumlah piksel objek dan total piksel *background*, maka selanjutnya sistem dapat melakukan perhitungan luas objek pada citra.

Tingkat efektifitas penentuan luas objek pada citra dinilai dari persentase perbedaan antara luas objek hasil perhitungan oleh sistem dengan luas asli objek. Nilai persentase yang kecil atau mendekati 0 mengindikasikan bahwa hasil perhitungan oleh sistem cukup baik karena mendekati luas asli objek. Namun apabila persentase perbedaan ternyata sangat besar, maka ini berarti hasil perhitungan luas objek oleh sistem tidak cukup baik atau buruk. Untuk menghitung persentase perbedaan ini, persamaan yang digunakan adalah persamaan (2).

Untuk hasil perhitungan luas masing-masing objek pada citra dan nilai persentase perbedaan terhadap luas asli objek dapat dilihat pada bagian-bagian di bawah ini. Hasil perhitungan ditampilkan berdasarkan metode *thresholding* yang digunakan.

❖ Luas objek pada citra hasil segmentasi dengan metode *trial-and-error*

Berikut adalah hasil perhitungan luas objek untuk masing-masing objek pada semua citra uji. Luas objek yang ditampilkan pada bagian ini adalah luas objek pada citra yang disegmentasi dengan menggunakan metode *trial-and-error*.

Tabel 3. Luas objek pada kelompok citra uji berukuran 500 x 417.

Objek	Luas Asli (cm ²)	Luas Objek (cm ²)				Persentase Keberhasilan (%)
		Kiri-Atas	Tengah	Kanan-Bawah	Rata-rata (\bar{x})	
Segitiga	18	19,21	18,56	17,07	18,28	98,45
Persegi	36	38,38	36,95	34,62	36,65	98,19
Lingkaran	28,26	30,00	28,98	26,60	28,53	99,06
Rata-rata keberhasilan						98,57

Pada Tabel 3 dapat dilihat hasil perhitungan luas masing-masing objek pada semua citra uji yang berukuran 500 x 417. Adapun setiap objek berada di posisi yang berbeda-beda pada setiap citra uji, posisi-posisi tersebut yakni pada bagian kiri-atas dari *background*, pada bagian tengah, dan pada bagian kanan-bawah. Untuk nilai rata-rata luas (\bar{x}) masing-masing objek segitiga, persegi, dan lingkaran secara berurutan adalah 18,28 cm², 36,65 cm², dan 28,53 cm², sedangkan persentase keberhasilan secara berurutan adalah 98,45 %, 98,19 %, dan 99,06 %.

Tabel 4. Luas objek pada kelompok citra uji berukuran 1000 x 833

Objek	Luas Asli (cm ²)	Luas Objek (cm ²)				Persentase Keberhasilan (%)
		Kiri-Atas	Tengah	Kanan-Bawah	Rata-rata (\bar{x})	
Segitiga	18	19,22	18,52	17,03	18,26	98,57
Persegi	36	38,12	36,97	34,64	36,58	98,40
Lingkaran	28,26	29,96	28,91	26,61	28,49	99,17
Rata-rata Keberhasilan						98,71

Pada Tabel 4 ditunjukkan hasil perhitungan luas masing-masing objek pada semua citra uji yang berukuran 1000 x 833. Nilai rata-rata luas masing-masing objek segitiga, persegi, dan lingkaran secara berurutan adalah 18,26 cm², 36,58 cm², dan 28,49 cm². Persentase rata-rata keberhasilan perhitungan luas objek sebesar 98,71 %.

Tabel 5. Luas objek pada kelompok citra uji berukuran 1200 x 1000.

Objek	Luas Asli (cm ²)	Luas Objek (cm ²)				Persentase keberhasilan (%)
		Kiri-Atas	Tengah	Kanan-Bawah	Rata-rata (\bar{x})	
Segitiga	18	19,23	18,54	17,04	18,27	98,52
Persegi	36	38,18	36,95	34,65	36,59	98,35
Lingkaran	28,26	29,97	28,93	26,61	28,50	99,15
Rata-rata keberhasilan						98,67

Pada Tabel 5 dapat dilihat hasil perhitungan luas masing-masing objek pada semua citra uji yang berukuran 1200 x 1000.

Nilai rata-rata luas masing-masing objek segitiga, persegi, dan lingkaran secara berurutan adalah 18,27 cm², 36,59 cm², dan 28,50 cm². Persentase rata-rata keberhasilan perhitungan luas objek sebesar 98,67 %.

❖ Luas objek pada citra hasil segmentasi dengan metode multi-otsu

Berikut adalah hasil perhitungan luas objek untuk masing-masing objek pada semua citra uji dan persentase perbedaannya terhadap luas asli objek. Luas objek dan persentase perbedaan yang ditampilkan pada bagian ini merupakan hasil dari citra yang disegmentasi dengan menggunakan metode multi-otsu *thresholding*.

Tabel 6. Luas objek pada kelompok citra uji berukuran 500 x 417.

Objek	Luas Asli (cm ²)	Luas Objek (cm ²)				Persentase Keberhasilan (%)
		Kiri-Atas	Tengah	Kanan-Bawah	Rata-rata (\bar{x})	
Segitiga	18	19,34	18,43	16,99	18,25	98,61
Persegi	36	38,25	37,36	34,53	36,71	98,02
Lingkaran	28,26	29,85	28,92	26,72	28,50	99,16
Rata-rata keberhasilan						98,60

Pada Tabel 6 dapat dilihat hasil perhitungan luas masing-masing objek pada semua citra uji yang berukuran 500 x 417. Untuk nilai rata-rata luas masing-masing objek segitiga, persegi, dan lingkaran secara berurutan adalah 18,25 cm², 36,71 cm², dan 28,50 cm². Persentase rata-rata keberhasilan perhitungan luas objek sebesar 98,60 %.

Tabel 7. Luas objek pada kelompok citra uji berukuran 1000 x 833.

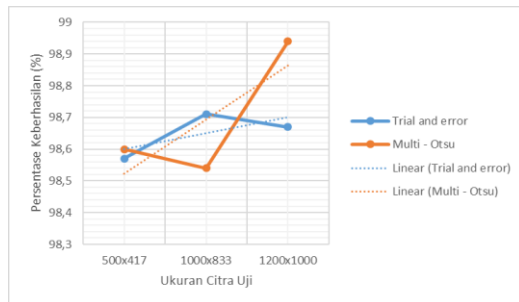
Objek	Luas Asli (cm ²)	Luas Objek (cm ²)				Persentase Keberhasilan (%)
		Kiri-Atas	Tengah	Kanan-Bawah	Rata-rata (\bar{x})	
Segitiga	18	19,48	18,50	16,93	18,30	98,32
Persegi	36	38,07	37,30	34,60	36,66	98,18
Lingkaran	28,26	29,93	28,84	26,74	28,51	99,13
Rata-rata keberhasilan						98,54

Pada Tabel 7 dapat dilihat hasil perhitungan luas masing-masing objek pada semua citra uji yang berukuran 1000 x 833. Untuk nilai rata-rata luas masing-masing objek segitiga, persegi, dan lingkaran secara berurutan adalah 18,30 cm², 36,66 cm², dan 28,51 cm². Persentase rata-rata keberhasilan perhitungan luas objek sebesar 98,54 %.

Tabel 8. Luas objek pada kelompok citra uji berukuran 1200 x 1000.

Objek	Luas Asli (cm ²)	Luas Objek (cm ²)				Persentase Keberhasilan (%)
		Kiri-Atas	Tengah	Kanan-Bawah	Rata-rata (\bar{x})	
Segitiga	18	19,30	18,38	16,80	18,16	99,11
Persegi	36	38,05	37,19	34,60	36,61	98,30
Lingkaran	28,26	29,81	28,78	26,69	28,42	99,42
Rata-rata keberhasilan						98,94

Pada Tabel 8 dapat dilihat hasil perhitungan luas masing-masing objek pada semua citra uji yang berukuran 1200 x 1000. Untuk nilai rata-rata luas masing-masing objek segitiga, persegi, dan lingkaran secara berurutan adalah 18,16 cm², 36,61 cm², dan 28,42 cm². Persentase rata-rata keberhasilan perhitungan luas objek sebesar 98,94%



Gambar 8. Pengaruh ukuran citra terhadap nilai persentase keberhasilan metode trial-and-error dan multi-otsu

Pada Gambar 8 dapat dilihat pengaruh perubahan ukuran citra terhadap persentase keberhasilan metode *trial-and-error* dengan multi-otsu pada penerapan teknik segmentasi berdasarkan warna pada ruang warna HSV. Dari Grafik perbedaan keberhasilan tidak signifikan, namun garis tren linear pada metode multi-otsu cenderung lebih menaik dibanding trial-and-error.

d. Evaluasi

Pada bagian ini dilakukan evaluasi kinerja sistem yang dibuat dengan meninjau nilai persentase keberhasilan sebagai indikasi tingkat efektifitasnya.

Perbandingan hasil evaluasi antara kedua metode thresholding yakni metode *trial-and-error* dengan metode multi-otsu.

Tabel 9. Rata-rata luas objek secara keseluruhan pada kedua metode.

Objek	Luas Asli (cm ²)	Rata-rata Luas Objek Keseluruhan (cm ²)		Persentase Keberhasilan (%)	
		Trial-and-error	Multi-Otsu	Trial-and-error	Multi-Otsu
Segitiga	18	18,27	18,24	98,51	98,68
Persegi	36	36,61	36,66	98,31	98,17
Lingkaran	28,26	28,51	28,48	99,13	99,23
Rata-rata		27,42	27,8	98,65	98,69

Dari Tabel 9 dapat dilihat rata-rata persentase keberhasilan perhitungan luas untuk semua objek dari kedua metode yang digunakan yakni 98,65% dengan metode *trial-and-error* dan 98,69% metode multi-otsu. Secara keseluruhan rata-rata persentase keberhasilan untuk semua objek yang dihasilkan dari kedua metode *thresholding* memiliki perbedaan yang tidak signifikan, sehingga penentuan luas objek dengan kedua metode dapat digunakan. Namun disarankan menggunakan multi-otsu karena lebih mudah dalam implementasi.

KESIMPULAN

Variasi ukuran citra tidak signifikan mempengaruhi persentase keberhasilan perhitungan luas objek citra menggunakan metode segmentasi *trial-and-error* dan multi-otsu. Persentase keberhasilan perhitungan luas pada penerapan teknik segmentasi berdasarkan warna pada ruang warna HSV untuk semua objek yang dihasilkan pada metode *trial-and-error* adalah 98,65% sedangkan metode multi-otsu adalah 98,69%. Persentase keberhasilan dari metode *trial-and-error* dan multi-otsu memiliki perbedaan yang tidak signifikan.

DAFTAR PUSTAKA

[1] Vartak, A. P., & Mankar, V. (2013). Colour Image Segmentation - A Survey. *International Journal of Emerging Technology and Advanced Engineering*, 3(2), 681-688.

[2] Bora, D. J., & Gupta, A. K. (2014). A New Approach Toward Clustering based Color Image Segmentation. *International Journal of Computer Applications*, 107(12), 23-30.

[3] Bora, D. J., Gupta, A. K., & Khan, F. A. (2015). Comparing the Performance of L*A*B and HSV Color Spaces with

- Respect to Color Image Segmentation. *International Journal of Emerging Technology and Advanced Engineering*, 5(2).
- [4] Mohd Ali, N., Md Rashid, N. K. A., & Mustafah, Y. M. (2013). Performance Comparison between RGB and HSV Color Segmentations for Road Signs Detection. *Applied Mechanics and Materials*, 393, 550–555. Doi: 10.4028/www.scientific.net/amm.393.550
- [5] Ong, P. M. B., & Punzalan, E. R. (2014). *Comparative Analysis of RGB and HSV Color Models in Extracting Color Features of Green Dye Solutions*. DLSU Research Congress 2014, De La Salle University, Manila, Philippines.
- [6] Garcia-Lamont, F., Cervantes, J., Lopez, A., & Rodriguez, L. (2018). Segmentation of images by color features: a survey. *Neurocomputing*, 292, 1–27. Doi: 10.1016/j.neucom.2018.01.091
- [7] Putranto, B. Y. B., Hapsari, W., & Wijana, K. (2010). Segmentasi warna citra dengan deteksi warna hsv untuk mendeteksi objek. *Jurnal Informatika*, 6(2), 1-15.
- [8] Utami, R. Z., Suksmadana, I. M. B., & Kanata, B. (2015). Menentukan luas objek citra dengan teknik deteksi tepi. *Dielektrika*, 2(1), 11-17.
- [9] Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O. D., & Wijanarto. (2009). *Teori pengolahan citra digital*. Jogjakarta: Andi.
- [10] Datumizer. (2015). HSV color solid cone. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:HSV_color_solid_cone.png
- [11] Jain, A. K. (1989). *Fundamental of digital image processing*. New Jersey: Prentice-Hall.
- [12] Giannakopoulos, T. (2008). Color Detection Using HSV Color Space (Training and Testing). Diakses dari <https://www.mathworks.com/matlabcentral/fileexchange/18440-color-detection-using-hsv-color-space-training-and-testing>
- [13] Liao, P. S., Chen, T. S., & Chung, P-C. (2001). A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, 17(5), 713-727.
- [14] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9, 62-66.
- [15] Syafi'i, I. S., Wahyuningrum, R. T., & Muntasa, A. (2015). Segmentasi obyek pada citra digital menggunakan metode otsu thresholding. *Jurnal Informatika*, 13(1).