

Penerapan Metode HSV-TCA Untuk Mendeteksi Kutu Beras (*Sitophilus Oryzae L*) Secara Real-Time

Uvi Desi Fatmawati^{a1}, Wibby Aldryani Astusi Praditasari^{a2}, Ria Aprilliyani^{a2}

^aFakultas Teknik Militer, Universitas Pertahanan Republik Indonesia
Kawasan IPSC Sentul, Sukahati, Kec. Citeureup, Kabupaten Bogor, Jawa Barat 16810

¹Uvi.fatmawati@idu.ac.id

³Ria.Aprilliyani@idu.ac.id

²Wibby.Aldryani@idu.ac.id

Abstrak

Metode HSV-TCA adalah sebuah penggabungan antara metode *Thresholding* dalam ruang warna HSV yang disempurnakan dengan metode deteksi *Contour Area*. Dalam penelitian ini, metode *thresholding* digunakan untuk memisahkan antara objek yang berupa kutu beras jenis *Sitophilus Oryzae L* dengan *background*-nya yaitu beras. Prinsip *Region Of Interest (ROI)* digunakan untuk meminimalisir kesalahan dalam pendeteksian dari kamera *webcam* dikarenakan ukuran beras dan kutu yang relatif kecil. Nilai *threshold* dan *contour area* (jenis *contour* dan lebar *contour area*) dapat dijadikan input dalam penggambaran *ROI* sehingga dapat dilakukan pengambilan gambar dalam bentuk tertentu seraca *real-time*. Percobaan pada dua kualitas beras telah dilakukan. Beberapa *library OpenCV* digunakan dan berfungsi dengan baik. Hasil penelitian ini menunjukkan bahwa metode ini bisa memisahkan antara objek kutu beras jenis *Sitophilus Oryzae L* dengan *background*-nya yaitu beras dimana keduanya sama-sama berukuran kecil, sekaligus mendeteksi kutu beras jenis *Sitophilus Oryzae L* secara *real-time*.

Kata kunci: thresholding, HSV, opencv, contour area, kutu beras, *sitophilus oryzae l*

Implementation of HSV-TCA Method to detect Rice Weevil (*Sitophilus Oryzae L*) in Real-Time

Abstract

The HSV-TCA method is a combination of the Thresholding method in the HSV color space enhanced with the Contour Area detection method. In this study, the thresholding method was used to separate objects between rice weevil (*Sitophilus Oryzae L*) and the background (rice). The Region Of Interest (ROI) principle was used to minimize errors detection from camera due to the relatively small size of rice weevil and rice. Values of Threshold and contour area (contour type and contour area width) can be used as inputs in representation of ROI, so that real-time shooting can be carried out in certain forms. Experiments on two qualities of rice have been carried out. Several OpenCV libraries were used and run well. The results of this study showed that this method is not only can separated between rice weevil (*Sitophilus Oryzae L*) and their background (rice) which are both small in size, but also can detected the *Sitophilus Oryzae L* in real-time.

Keywords: thresholding, HSV, opencv, contour area, rice weevil, *sitophilus oryzae l*.

I. PENDAHULUAN

Teknologi *computer vision* saat ini berkembang semakin pesat, salah satu penerapan *computer vision* yang sering digunakan dalam penelitian adalah untuk mendeteksi objek. Deteksi objek digunakan untuk mengetahui ada atau tidaknya suatu objek dalam suatu citra. Metode yang sering digunakan dalam deteksi objek dalam suatu citra digital adalah metode *thresholding*. Metode *thresholding* ini bisa diaplikasikan dalam ruang warna HSV. HSV adalah suatu model warna citra digital yang terdiri dari tiga elemen yaitu *Hue*, *Saturation* dan *Value*[1]. Salah satu fitur yang sering digunakan dalam

pendeteksian suatu objek adalah fitur dalam *Library OpenCV (Open Source Computer Vision Library)*. *Library OpenCV* adalah *library computer vision* dan pembelajaran mesin (*Machine Learning*) yang bersifat terbuka (*open source*).

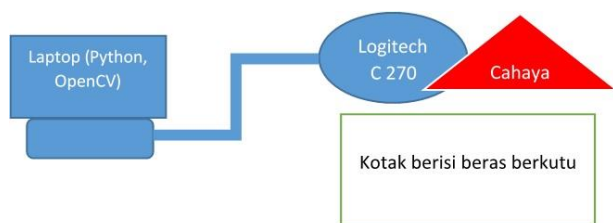
Penggunaan metode *thresholding* dalam ruang warna HSV telah dilakukan oleh banyak peneliti, antara lain oleh[1]. Pada penelitian tersebut telah dilakukan pemisahan antara objek deteksi dengan *background* dari sebuah citra burung yang bertengger pada dahan menggunakan metode HSV (tidak secara *real-time*). Objek deteksi adalah burung beserta dahan yang digunakannya

untuk bertengger (ukuran besar). Kemudian metode *HSV* dibandingkan dengan metode *LAB* dan hasilnya metode *HSV* memberikan hasil yang lebih baik dari metode *LAB*. Penelitian yang dilakukan oleh [2] berhasil mengimplementasikan metode *color filtering HSV* dan *contour area* untuk membedakan antara bola (berukuran sedang) sebagai objek deteksi pada robot sepak bola beroda secara *real-time* dengan tingkat keberhasilan sebesar 94,44%. Penelitian yang dilakukan oleh [3] menunjukkan bahwa metode *HSV* bisa digunakan untuk memisahkan objek deteksi berupa mainan berwarna dengan *background*-nya secara baik. Dalam penelitiannya, penulis menggunakan citra mainan (tidak *real-time*) dan ukuran mainan yang relatif besar. Penelitian lain dilakukan oleh [4], dalam penelitian tersebut dilakukan pemisahan warna dan bentuk untuk mendeteksi objek berupa buku (ukuran sedang-besar) dan dilakukan secara *real-time*. Penelitian yang dilakukan oleh [5] menunjukkan bahwa *thresholding* pada ruang warna *HSV* dapat digunakan untuk mendeteksi iris mata secara *real-time*. Penelitian oleh [6] menunjukkan bahwa metode *thresholding* pada ruang warna *HSV* bisa digunakan untuk melacak warna pada suatu file video apabila dimasukan nilai *thresholding HSV* sesuai dengan warna yang ingin dilacak dalam video tersebut. Kesemua penelitian tersebut [1]–[6] digunakan untuk mendeteksi objek yang berukuran relatif besar (mudah terlihat dan jelas bentuk dan gambarnya) baik secara *real-time* maupun tidak secara *real-time*.

Tujuan dari penelitian ini adalah memisahkan antara objek kutu beras jenis *Sitophilus Oryzae L* dengan beras sekaligus melakukan pendeteksian kutu beras jenis *Sitophilus Oryzae L* secara *real-time*, dimana pada dasarnya kutu beras tersebut berukuran kecil dan sedikit sulit terlihat, yang berada diatas kumpulan beras dimana beras juga berukuran kecil dan notabene jumlahnya banyak. Kutu beras jenis *Sitophilus Oryzae L* adalah objek yang akan dideteksi sedangkan beras adalah *background* dari objek deteksi

II. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini digambarkan pada Gambar 1 dan Gambar 2. Gambar 1 menjelaskan tata cara pengujian metode yang sudah disiapkan dalam bentuk kode program pada *software python* yang terinstal *library OpenCV* didalamnya. Kemudian Gambar 2 menjelaskan bagaimana metode pendeteksian bekerja.



Gambar 1. Tata Cara Pengujian

Berdasarkan Gambar 1, terdapat sekumpulan beras dimana kondisi beras ini notabene sudah berkontaminasi

dalam jumlah banyak yang mengakibatkan kondisi beras adalah rusak parah dan tidak layak makan maupun beras yang masih bagus dan diberi beberapa kutu beras diatasnya. Beras ini diletakkan dalam sebuah wadah dan diberi cahaya sehingga kondisi wadah tidak gelap, kemudian diatas wadah beras dipasang kamera jenis *webcam* merk *Logitech C270* yang dikoneksikan dengan komputer untuk melakukan *streaming video real-time* pada kutu beras beserta berasnya yang berada dalam wadah beras.



Gambar 2. Alur Pendeteksian Kutu Beras jenis *Sitophilus Oryzae L*

Kemudian untuk metode pendeteksian digambarkan dalam Gambar 2. Berdasarkan Gambar 2, metode *thresholding* dalam ruang warna *HSV* ini diawali dengan mengkonversi suatu citra *RGB* kedalam citra *HSV* dan kemudian menentukan nilai *threshold* suatu citra yang akan dideteksi sehingga terlihat bedanya dengan citra yang tidak akan dideteksi. Citra *HSV* tersebut selanjutnya dikonversi menjadi citra biner (hitam dan putih) yaitu dengan menentukan nilai intensitas dari suatu citra yang akan dideteksi. Nilai intensitas citra yang lebih dari atau sama dengan nilai *threshold* akan diubah menjadi putih (1) sedangkan nilai intensitas citra yang kurang dari nilai *threshold* akan diubah menjadi hitam (0). Dalam hal ini kutu beras jenis *Sitophilus Oryzae L* akan menjadi putih dan *background* yang berupa kumpulan beras menjadi hitam. Untuk menambah efek kejelasan dan kehalusan

pada citra biner yang dihasilkan, digunakan *filtering* citra dan *smoothing* citra. *Filtering* citra digunakan untuk memperjelas citra dengan operasi filter morfologi yaitu *dilasi* dan *opening*. Kemudian *smoothing* citra digunakan untuk menjadikan efek halus (*smoothing*) pada citra sekaligus untuk mempertegas nilai *ROI*. Proses *smoothing* citra ini menggunakan teknik *Gaussian Blur*. Semua operasi-operasi ini ada dalam *library OpenCV*. Dikarenakan sering terjadi jarak antar beras yang dalam citra tersebut bisa terdeteksi sebagai kutu beras jenis *Sitophilus Oryzae L*, karena nilai *threshold*-nya sama dengan nilai *threshold* kutu beras jenis *Sitophilus Oryzae L* dan faktor intensitas cahaya yang berkurang dalam ruang deteksi, maka dalam penelitian ini metode *thresholding* citra *HSV* dikombinasikan dengan pendeteksian *contour area*. Pendeteksian *contour area* ini berupa jenis *contour* dan lebar *contour area* untuk menghasilkan *ROI* yang sesuai dengan ukuran kutu beras jenis *Sitophilus Oryzae L* sebagai objek deteksi sehingga tidak terjadi kesalahan pendeteksian saat mendeteksi suatu objek kecil yaitu kutu beras jenis *Sitophilus Oryzae L* yang berada pada permukaan beras (yang kecil dan banyak) secara *real-time*.

A. Kutu Beras

Salah satu jenis kutu beras yang sering dijumpai dalam beras adalah jenis *Sitophilus Oryzae L*. kutu jenis ini merupakan salah satu jenis serangga gudang yang banyak merusak persediaan beras di tempat penyimpanan. Serangga *Sitophilus Oryzae* menyebabkan butiran beras menjadi berlobang kecil-kecil serta mudah pecah dan remuk bagaikan tepung, sehingga kualitasnya rendah karena rasanya tidak enak dan berbau apek [7]–[9]. Bentuk kutu beras ini jika diperbesar adalah seperti ditunjukkan dalam Gambar 3. Walaupun berbentuk kecil, kutu beras ini bisa dideteksi keberadaannya menggunakan kamera *webcam* yang terintegrasi dengan menggunakan komputer dan implementasi algoritma tertentu.



Gambar 3. Kutu Beras (*Sitophilus Oryzae L*)

B. Webcam Logitech C-270

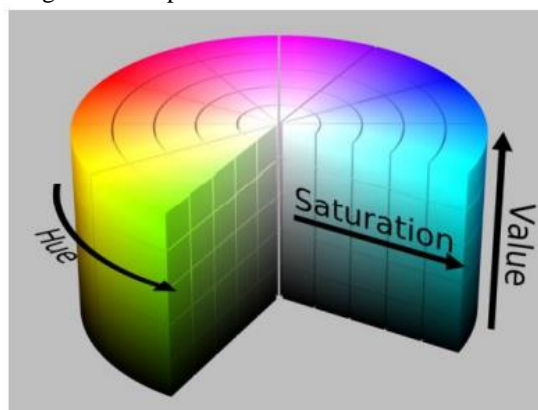
Webcam Logitech C-270 merupakan jenis kamera *webcam* yang mempunyai Resolusi Maksimal 720p/30fps. Kamera ini kompatibel dengan panggilan video *HD 720p/30 fps* yang jelas dengan bidang pandang 55° diagonal dan fitur *auto light correction*. Kamera ini bisa dikoneksikan dengan *USB* dengan jenis fokus *fixed focus* [10].

Karena jenis fokusnya yang *fixed*, sedangkan saat digunakan untuk mendeteksi kutu beras yang kecil kamera harus berada tidak terlalu jauh dari objek deteksi maka perlu dilakukan sedikit modifikasi fokusnya dengan cara membuka casing kamera secara keseluruhan dan

dilakukan pengkoreksian fokus secara manual. Proses ini bisa dilakukan dengan menggunakan obeng, kemudian memasang kembali casing kamera, sehingga kamera bisa digunakan sesuai dengan kebutuhan. Karena jika tidak diatur secara demikian, maka hasil gambar yang ditangkap akan buram dan *blur* atau tidak fokus dan tidak bisa digunakan untuk mendeteksi kutu beras yang berukuran kecil.

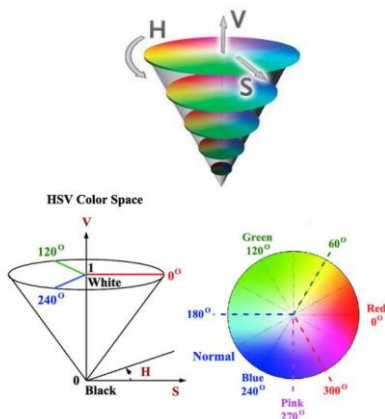
C. Ruang Warna HSV

Cara standar untuk menspesifikasikan suatu warna tertentu adalah menggunakan ruang warna atau model warna (*color model*). *Color model* ini diawali dengan mendefinisikan suatu sistem koordinat 3D, dan suatu ruang bagian yang mengandung semua warna yang dapat dibentuk ke dalam suatu model tertentu [11]. Contoh ruang warna yang sering digunakan adalah *RGB* (*Red Green Blue*), *CMYK* (*Cyan Magenta Yellow Black*), *HSL* (*Hue Saturation Lightness*), *HSV* (*Hue Saturation Value*), *Luminance – Chrominance* dan sebagainya. Pada penelitian ini ruang warna yang digunakan adalah ruang warna *HSV*. *HSV* ini dibentuk dengan menggunakan komposisi warna pokok yang sekaligus memperhitungkan juga kecerahan dan komposisi warnanya. Ruang warna *HSV* digambarkan pada Gambar 4.



Gambar 4. Ruang Warna HSV

Ruang warna *HSV* mempunyai 3 variabel yang berhubungan antara *Hue*, *Saturation* dan *Value*. Ruang warna *HSV* didasarkan pada koordinat polar. *HSV* adalah versi kubus *RGB* dengan transformasi *non-linear* [12]. Berdasarkan Gambar 5, masing-masing komponen dari ruang warna *HSV* bisa dijabarkan sebagai berikut : *Hue* adalah deskripsi untuk derajat warna pada lingkaran yang menunjukkan perbedaan warna, walaupun putaran derajat adalah sebesar 360° , namun nilai *hue* dinormalisasi dalam jarak 0 – 255 dengan nilai awal titik 0 ialah warna merah. *Saturation* adalah representasi kroma/ saturasi pada warna, semakin rendah saturasi maka warna akan memudar dan mendekati putih sedangkan jika semakin tinggi maka warna akan semakin dalam atau jelas (perbedaan semakin mencolok). *Value* adalah representasi dari kecerahan warna, dengan jarak 0 – 255 dimana 0 adalah warna hitam sedangkan 255 adalah warna putih [2].



Gambar 5. Komponen pada Ruang Warna HSV

Format citra yang ditangkap pada penelitian ini adalah dalam ruang warna RGB. Maka untuk mendapatkan nilai *thresholding* pada warna yang diinginkan dalam ruang warna HSV dibutuhkan konversi. Perhitungan konversi dari ruang warna RGB ke ruang warna HSV ditunjukkan pada persamaan (2), setelah sebelumnya dilakukan normalisasi pada warna RGB seperti ditunjukkan pada persamaan (1). Proses normalisasi diperlukan karena pada persamaan (1), nilai H tidak bisa ditentukan jika nilai S = 0 [13]:

$$\begin{aligned}
 r &= \frac{R}{R+G+B} \\
 g &= \frac{G}{R+G+B} \\
 b &= \frac{B}{R+G+B} \dots\dots\dots(1)
 \end{aligned}$$

$$\begin{aligned}
 H &= \tan \left[\frac{3(g-b)}{(r-g)+(r-b)} \right] \\
 S &= 1 - \frac{\min(r,g,b)}{V} \\
 V &= \frac{r+g+b}{3} \dots\dots\dots(2)
 \end{aligned}$$

Cara lain mentransformasikan ruang warna RGB ke ruang warna HSV adalah dengan persamaan (3) :

$$\begin{aligned}
 V &= \max(r, g, b) \\
 S &= \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(r,g,b)}{V}, & \text{jika } V > 0 \end{cases} \\
 H &= \begin{cases} 0 & \text{jika } V = r \\ \frac{60x(g-b)}{S \times V} & \text{jika } V = g \\ 60x \left[2 + \frac{(b-r)}{S \times V} \right] & \text{jika } V = b \\ 60x \left[4 + \frac{(r-g)}{S \times V} \right] & \text{jika } V = b \end{cases} \\
 H &= H + 360 \text{ jika } H < 0 \dots\dots\dots(3)
 \end{aligned}$$

D. Menentukan Nilai Tresholding

Perkembangan teknologi yang semakin canggih, memudahkan pekerjaan manusia dalam segala aspek. Dalam penelitian ini, proses menentukan nilai *threshold*

adalah dengan cara pengambilan citra kudu dalam beras sebagai sample yang sebelumnya citra tersebut diambil dalam kondisi pencahayaan cukup yang sesuai dengan proses deteksi *real-time* nantinya. Kemudian setelah citra didapatkan, digunakan *software colorpicker online* yang banyak dijumpai di laman internet. *Software* ini mempunyai fitur deteksi berbagai nilai ruang warna berdasarkan fitur *pick* gerakan *mouse* pada komputer. Namun nilai ruang warna yang didapatkan tetap harus dikondisikan kembali dengan normalisasi, karena hasil deteksi *software colorpicker online* masih dalam satuan persen dan derajat untuk ruang warna HSV. Proses normalisasi menggunakan persamaan (4) dibawah ini :

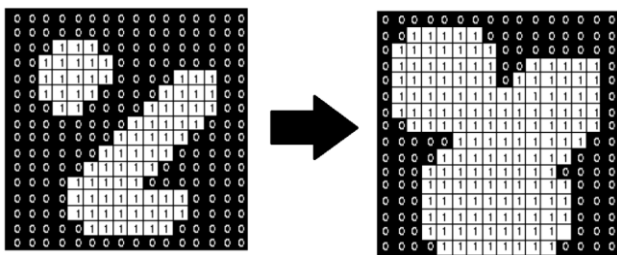
$$\begin{aligned}
 h &= \frac{H}{360} \times 255 \\
 s &= \frac{S}{100} \times 255 \\
 v &= \frac{V}{100} \times 255 \dots\dots\dots(4)
 \end{aligned}$$

Setelah normalisasi nilai HSV ditemukan, selanjutnya menentukan batasan *thresholding* dari nilai HSV normalisasi tersebut, yaitu dengan cara mengurangi +/- 10 poin pada nilai normalisasi HSV untuk mendapatkan nilai *thresholding* yang sesuai dengan kebutuhan. Walaupun sudah ada acuan penuh, penentuan nilai ini tetap menggunakan metode *trial and error* karena masing-masing kudu mempunyai pendekatan warna yang berbeda-beda. Dalam penelitian ini digunakan nilai ruang warna HSV dengan ketentuan sebagai berikut :

- h = 20-45
- s = 16-37
- v = 10-80

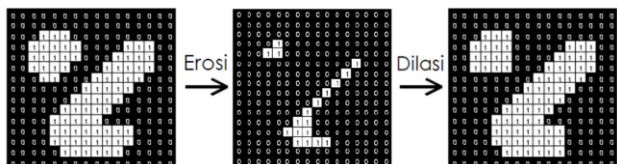
E. Proses Filtering Dan Smoothing Dengan Fitur Pada Library OpenCV

Pada penelitian ini digunakan 2 macam cara untuk menghasilkan deteksi objek yang tajam dan halus yaitu teknik pemfilteran citra dan teknik *smoothing* citra. Filter adalah proses untuk menyaring frekuensi-frekuensi tertentu dari sekumpulan sinyal untuk diloloskan[14]. Sama halnya dengan definisi filter, filter pada citra digital adalah suatu proses dengan mengambil beberapa frekuensi citra tertentu dan membuang frekuensi citra yang tidak diperlukan. Teknik pemfilteran yang digunakan dalam penelitian ini adalah teknik morfologi. Teknik morfologi merupakan teknik pengolahan citra berdasarkan bentuk segmen citra yang bertujuan untuk memperbaiki hasil segmentasi [15], [16]. Teknik morfologi biasanya digunakan pada citra biner atau untuk beberapa kasus juga bisa diterapkan pada citra keabuan (*grayscale*) [17]. Teknik *filtering* morfologi yang digunakan dalam penelitian ini adalah teknik Dilasi dan *Opening*. Dilasi adalah teknik untuk memperbesar atau memperlebar batas-batas objek (citra biner) dengan menambah lapisan disekeliling objek. Teknik dilasi ditunjukkan oleh Gambar 6.



Gambar 6. Filter Morfologi Dilasi

Pada Gambar 6 terlihat titik latar (0) yang bertetangga dengan titik objek (1) menjadi titik objek (1). *Opening* adalah jenis operasi filter morfologi yang menggunakan perpaduan antara dua metode yaitu metode *erosi* yang diikuti metode dilasi. Metode *erosi* adalah kebalikan dari metode dilasi. Proses *opening* dimulai dengan melakukan *erosi* pada citra kemudian pada citra hasil *erosi* tersebut dilakukan dilasi. *Opening* biasanya digunakan untuk menghilangkan objek-objek kecil dan sempit (*noise*) serta dapat membuat tepi citra lebih *smooth* (untuk citra berukuran besar). Proses *opening* digambarkan pada Gambar 7.



Gambar 7. Filter Morfologi *Opening*

Untuk mendapatkan efek halus pada keseluruhan citra sehingga objek yang dideteksi juga menjadi lebih halus sekaligus saat penerapan *ROI*, hasil dari *ROI* menjadi lebih akurat, digunakan teknik *smoothing* citra dengan metode *Gaussian Blurr*. *Gaussian Blurr* adalah sebuah teknik yang sangat efektif untuk menghilangkan *Gaussian noise* pada sebuah citra [18]–[20]. Keseluruhan proses *filtering* dan *smoothing* pada penelitian ini memanfaatkan *library OpenCV* yang terintegrasi dengan *Python*.

F. *Library OpenCV*

Penelitian dibidang *image processing* dan *machine learning* yang semakin meningkat membuat *library OpenCV* menjadi salah satu *library* yang sangat diminati dikalangan para peneliti dan akademisi. *Library OpenCV* (*Open Source Computer Vision Library*) adalah *library* perangkat lunak yang berguna untuk pengolahan citra secara dinamis dan *real-time* [21]. Pada penelitian ini *library OpenCV* ditulis dengan menggunakan Bahasa pemrograman *Python*.

G. Mencari *Contours* Objek Yang Akan Dideteksi

Proses selanjutnya dari pendeteksian objek adalah proses pencarian *contour* objek. Proses ini diawali dengan mencari semua *contour* dalam tangkapan citra yang sudah di filter. Sehingga didapatkan semua *contour* baik untuk citra objek deteksi (kutu beras jenis *Sitophilus Oryzae L*) maupun citra objek-objek yang memiliki warna yang masuk dalam batasan nilai *thresholding*. Selanjutnya masing-masing *contour* dalam citra tersebut dicari titik

tengahnya dan dihitung luasannya. Apabila luasan *contour* sesuai dengan ciri luasan kutu beras jenis *Sitophilus Oryzae L*, maka *contour* tersebut akan dideteksi sebagai kutu beras jenis *Sitophilus Oryzae L*. Proses ini sangat didominasi dengan kualitas beras dan tingkat pencahayaan dalam ruang deteksi. Dikarenakan jika kualitas beras adalah beras yang buruk dan cahaya dalam ruang deteksi sangat minim. Bisa jadi ada objek-objek lain yang bukan kutu beras jenis *Sitophilus Oryzae L* akan masuk dalam nilai *thresholding* dan *contour area* sehingga akan terdeteksi sebagai kutu beras jenis *Sitophilus Oryzae L* dan menghasilkan *error* pendeteksian.

III. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini disajikan dalam Tabel 1, Tabel 2 dan Tabel 3. Tabel 1 menjelaskan hasil dari pendeteksian pada dua kelompok beras berketu. Pengelompokan beras berdasarkan kondisi beras, yaitu kelompok beras pertama adalah kelompok beras dengan kondisi rusak dan tidak layak makan dan kelompok beras kedua adalah kelompok beras dengan kondisi bagus dan layak makan. Penjabaran dari Tabel 1 pada masing-masing kelompok beras terdapat pada Tabel 2 dan Tabel 3 yang akan dijelaskan pada masing-masing sub bab.

TABEL 1. KONDISI BERAS, JUMLAH KUTU BERAS DAN HASIL DETEKSI

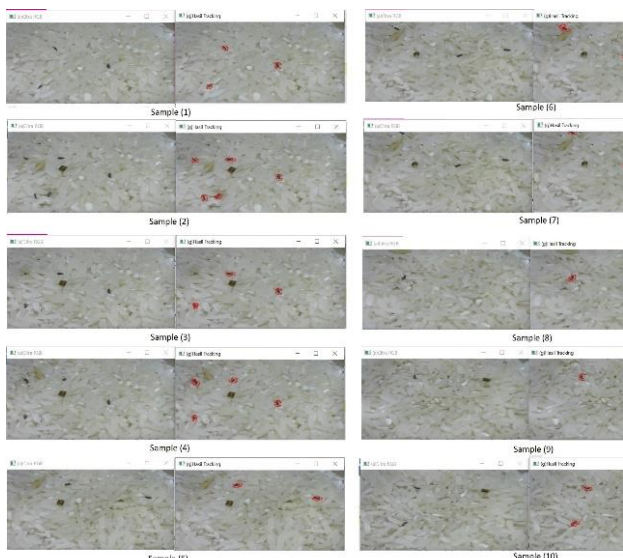
NO. KELOMPOK	KONDISI BERAS	JUMLAH KUTU BERAS	HASIL DETEKSI
1	Rusak dan tidak layak makan	Tidak terhitung	Ditunjukkan dalam Tabel 2
2	Bagus dan layak makan	Diberikan 9 ekor kutu beras jenis <i>Sitophilus Oryzae L</i> didalam beras, namun kutu beras jenis <i>Sitophilus Oryzae L</i> tidak bisa diposisikan stabil tetap diatas beras, terkadang masuk kedalam beras	Muncul beberapa kutu beras jenis <i>Sitophilus Oryzae L</i> ke permukaan beras dan terdeteksi dengan baik, ditunjukkan dalam Tabel 3

A. *Kondisi Beras Rusak Dan Tidak Layak Makan*

Kelompok beras pertama adalah kelompok beras dengan kondisi rusak dan tidak layak makan seperti pada Gambar 8. Ciri-ciri beras ini adalah berwarna sedikit kecoklatan dan tekstur yang sedikit menepung. Kerusakan beras ini sendiri terjadi akibat kutu beras yang berkembang biak didalam ruang penyimpanan beras. Selain kutu beras jenis *Sitophilus Oryzae*, dalam beras rusak ini juga terdapat serangga lain dengan bentuk yang hampir mirip dengan kutu beras jenis *Sitophilus Oryzae L*, namun warnanya sedikit lebih cerah. Pendeteksian kutu beras jenis *Sitophilus Oryzae L* pada kelompok beras pertama ini terlihat pada Gambar 9.



Gambar 8. Beras Rusak Dan Tidak Layak Makan



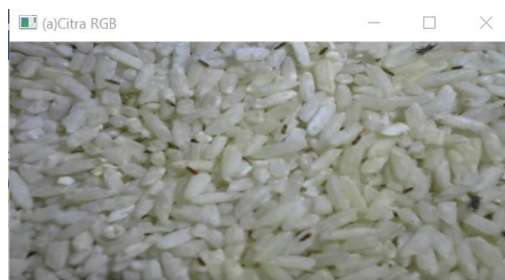
Gambar 9. Pendeteksian Kutu Pada Beras Rusak Dan Tidak Layak Makan

Hasil dari beras kelompok 1 disajikan dalam Tabel 2. Dimana Tabel 2 ini diambil dari Gambar 9.

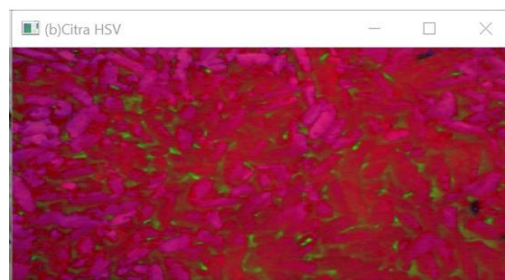
TABEL 2. BERAS RUSAK DAN TIDAK LAYAK MAKAN

Nama Sampel	Jumlah Kutu Beras	Hasil Deteksi Kutu Beras	Contour Blok Beras Yang terdeteksi Sebagai Kutu Beras
Sampel 1	Terlihat 3 ekor	Terdeteksi 3 ekor	Tidak Ada
Sampel 2	Terlihat 1 ekor	Terdeteksi 1 ekor	Tidak Ada
Sampel 3	Terlihat 5 ekor	Terdeteksi 4 ekor	3 Contour Blok
Sampel 4	Terlihat 4 ekor	Terdeteksi 2 ekor	2 Contour Blok
Sampel 5	Terlihat 5 ekor	Terdeteksi 4 ekor	Tidak Ada
Sampel 6	Terlihat 4 ekor	Terdeteksi 3 ekor	Tidak Ada
Sampel 7	Terlihat 4 ekor	Terdeteksi 4 ekor	1 Contour Blok
Sampel 8	Terlihat 4 ekor	Terdeteksi 4 ekor	1 Contour Blok
Sampel 9	Terlihat 5 ekor	Terdeteksi 5 ekor	1 Contour Blok
Sampel 10	Terlihat 4 ekor	Terdeteksi 4 ekor	Tidak Ada
Rata-rata deteksi = 88,5 %		Rata-rata error deteksi = 11,5 %	

Dari Tabel 2 bisa diketahui bahwa presentase rata-rata efektivitas pendeteksian kutu beras pada beras rusak dan tidak layak makan sebesar 88,5 % dengan error pendeteksian sebesar 11,5%. Pada sample nomor 3,4,7,8,9 beras rusak tidak layak makan, ada beberapa *contour* blok beras terdeteksi sebagai kutu beras, sedangkan pada sampel yang lain tidak terjadi demikian. Berdasarkan data pada Tabel 2, diambil sampel nomor 3 untuk dijabarkan gambarnya. Gambar-gambar yang disajikan dalam karya ini merupakan citra hasil tangkapan video *real-time* yang di *capture* saat proses pengujian berlangsung.



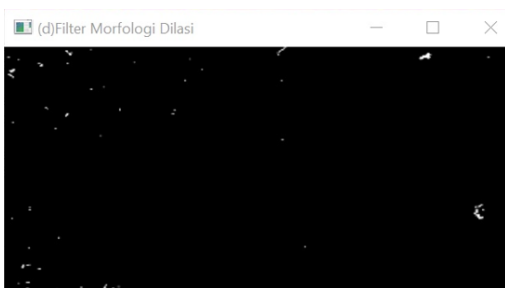
Gambar 10. Sample Nomor 3 Kondisi Beras Rusak Dan Tidak Layak Makan (Citra RGB)



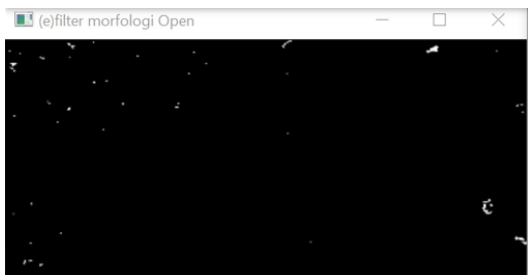
Gambar 11. Sample Nomor 3 Kondisi Beras Rusak Dan Tidak Layak Makan (Citra HSV)



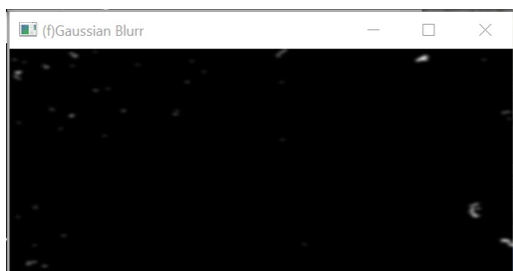
Gambar 12. Sample Nomor 3 Kondisi Beras Rusak Dan Tidak Layak Makan (Citra Biner)



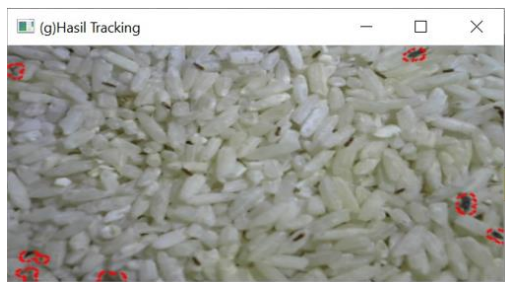
Gambar 13. Sample Nomor 3 Kondisi Beras Rusak Dan Tidak Layak Makan (Citra Dilasi)



Gambar 14. Sample Nomor 3 Kondisi Beras Rusak Dan Tidak Layak Makan (Citra *Opening*)



Gambar 15. Sample Nomor 3 Kondisi Beras Rusak Dan Tidak Layak Makan (Citra *Gaussian Blurr*)



Gambar 16. Sample Nomor 3 Kondisi Beras Rusak Dan Tidak Layak Makan (Citra hasil deteksi)

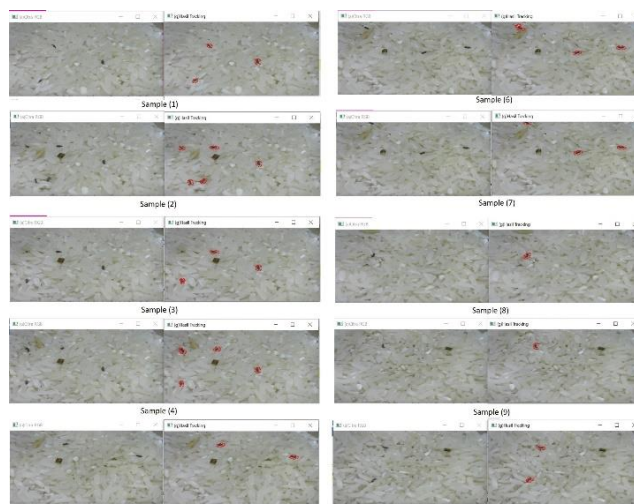
Berdasarkan Gambar 10, citra RGB yang diambil dari sampel nomer 3 beras rusak dan tidak layak makan, dapat dilihat bahwa terdapat 5 buah kutu beras jenis *Sitophilus Oryzae L* dan beberapa serangga lain didalam beras, dengan metode *HSV-TCA* bisa terdeteksi 4 ekor kutu beras jenis *Sitophilus Oryzae L* dan 1 ekor kutu beras jenis *Sitophilus Oryzae L* tidak terdeteksi. Hasil olah *HSV* pada Gambar 11 menunjukkan hasil konversi citra RGB (Gambar 10) ke *HSV*. Dari gambar terlihat bahwa kutu beras terlihat berwarna hitam sedangkan beras terlihat berwarna merah dan celah-celah antar beras terlihat berwarna hijau menyala. Hasil konversi *RGB* ke *HSV* tersebut dikonversi lagi menjadi biner seperti terlihat pada Gambar 12, hasil binarisasi memperlihatkan bahwa kutu beras jenis *Sitophilus Oryzae L* yang masuk dalam nilai *thresholding* sejumlah 4 buah diubah menjadi putih (1) sedangkan beras, 1 kutu beras jenis *Sitophilus Oryzae L* dan serangga lain dalam wadah diubah menjadi hitam (0), namun beberapa celah-celah antar beras tersebut masuk terkonsversi menjadi warna putih yang kemudian dianggap sebagai *noise*. Hasil *filtering* dengan *dilasi* (Gambar 13) menunjukkan bahwa *noise* tidak dapat dihilangkan dengan sempurna namun citra kutu beras yang semula sangat kecil menjadi melebar dan sedikit lebih jelas, begitu juga saat digunakan filter *opening* (Gambar 14). Proses *erosi* dan *dilasi* pada filter *opening* tetap tidak bisa

menghilangkan *noise* yang ada secara keseluruhan. Kemudian saat proses *smoothing* dengan *Gaussian Blurr* (Gambar 15), citra yang dihasilkan terlihat halus, namun beberapa *noise* masih terlihat dengan kentara walaupun beberapa *noise* bisa hilang. Sehingga pada citra pendeteksian *contour area* (Gambar 16), terlihat 3 *contour* blok area beras yang ditandai sebagai kutu beras jenis *Sitophilus Oryzae L*. 3 *contour* blok beras tersebut berasal dari *noise* dimana setelah proses *thresholding*, *filtering* dan *smoothing*, area celah-celah beras tersebut tetap masuk dalam *contour area* kutu beras jenis *Sitophilus Oryzae L*. hasil akhir pendeteksian menunjukkan hanya 4 ekor dari 5 ekor kutu jenis *Sitophilus Oryzae L* yang bisa terdeteksi dengan sempurna.

B. Kondisi Beras Bagus Dan Layak Makan



Gambar 17. Beras Bagus Dan Layak Makan



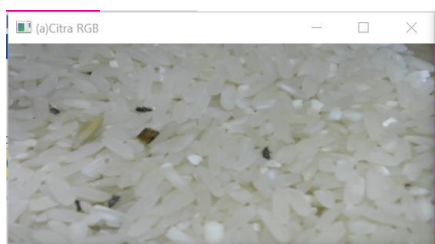
Gambar 18. Pendeteksian Kutu Pada Beras Baik Dan Layak Makan

Hasil dari beras kelompok 2 disajikan dalam Tabel 3. Dimana Tabel 3 ini diambil dari Gambar 18.

TABEL 3. BERAS BAGUS DAN LAYAK MAKAN

Nama Sampel	Jumlah Kutu Beras	Hasil Deteksi Kutu Beras	Beras Yang Terdeteksi Sebagai Kutu
Sampel 1	Terlihat 3 ekor	Terdeteksi 3 ekor	Tidak Ada
Sampel 2	Terlihat 5 ekor	Terdeteksi 5 ekor	Tidak Ada
Sampel 3	Terlihat 3 ekor	Terdeteksi 3 ekor	Tidak Ada
Sampel 4	Terlihat 4 ekor	Terdeteksi 4 ekor	Tidak Ada
Sampel 5	Terlihat 2 ekor	Terdeteksi 2 ekor	Tidak Ada
Sampel 6	Terlihat 3 ekor	Terdeteksi 3 ekor	Tidak Ada
Sampel 7	Terlihat 2 ekor	Terdeteksi 2 ekor	Tidak Ada
Sampel 8	Terlihat 1 ekor	Terdeteksi 1 ekor	Tidak Ada
Sampel 9	Terlihat 1 ekor	Terdeteksi 1 ekor	Tidak Ada
Sampel 10	Terlihat 2 ekor	Terdeteksi 2 ekor	Tidak Ada
Rata-rata deteksi = 100%		Rata-rata error deteksi = 0%	

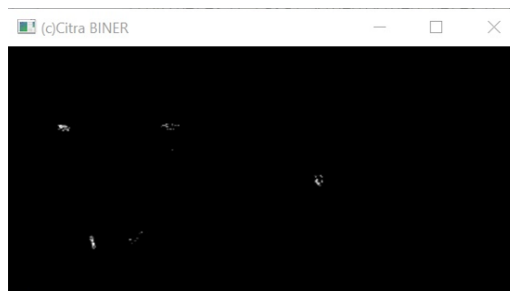
Dari Tabel 3 bisa diketahui bahwa presentase rata-rata efektivitas pendeteksian kutu pada beras kualitas bagus dan layak makan sebesar 100 % dengan *error* pendeteksian sebesar 0 %. Pada sampel beras bagus dan layak makan ini, terdapat 1 butir beras yang warnanya sedikit kecoklatan, 1 butir batu kecil yang warnanya kehitaman dan juga terdapat 1 butir gabah tertangkap kamera, namun ketiganya tidak terdeteksi sebagai kutu. Hasil yang signifikan ini bisa diperoleh dengan kualitas pencahayaan yang baik dalam ruang deteksi. Berdasarkan data pada Tabel 3, diambil sampel nomor 2 untuk dijabarkan gambarnya. Gambar-gambar yang disajikan dalam karya ini merupakan citra hasil tangkapan video *real-time* yang di *capture* saat proses pengujian berlangsung.



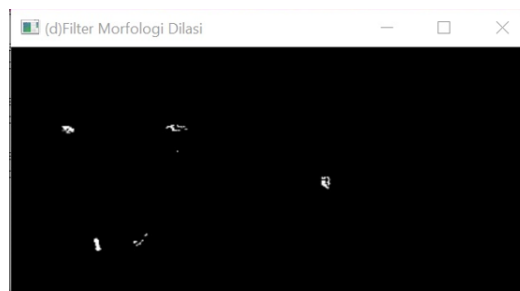
Gambar 19. Sample Nomor 2 Kondisi Beras Baik Dan Layak Makan (Citra RGB)



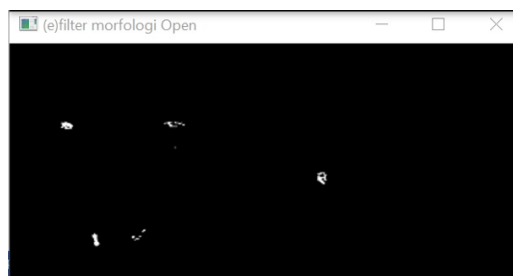
Gambar 20. Sample Nomor 2 Kondisi Beras Baik Dan Layak Makan (Citra HSV)



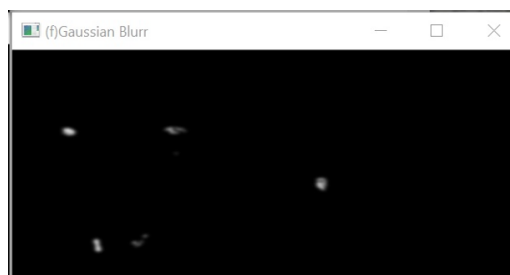
Gambar 21. Sample Nomor 2 Kondisi Beras Baik Dan Layak Makan (Citra Biner)



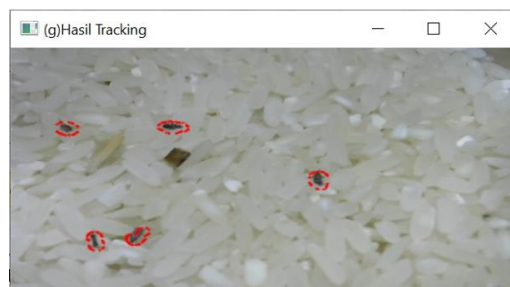
Gambar 22. Sample Nomor 2 Kondisi Beras Baik Dan Layak Makan (Citra Dilasi)



Gambar 23. Sample Nomor 2 Kondisi Beras Baik Dan Layak Makan (Citra Opening)



Gambar 24. Sample Nomor 2 Kondisi Beras Baik Dan Layak Makan (Citra Gaussian Blurr)



Gambar 25. Sample Nomor 2 Kondisi Beras Baik Dan Layak Makan (Citra Hasil Deteksi)

Berdasarkan Gambar 19, citra *RGB* yang diambil dari sampel nomer 3 beras bagus dan layak makan, dapat dilihat bahwa terdapat 5 buah kutu beras jenis *Sitophylus Oryzae L* didalamnya, dengan metode *HSV-TCA* bisa terdeteksi semua kutu beras jenis *Sitophylus Oryzae L*. Hasil olah *HSV* pada Gambar 20 terlihat bahwa semua kutu beras masuk dalam nilai *thresholding*nya dan beras diubah menjadi berwarna merah sedangkan kutu beras jenis *Sitophylus Oryzae L* tetap terlihat berwarna hitam, kemudahan celah-celah antar beras dan juga batu yang terdapat didalam ruang deteksi terlihat berwarna hijau menyala. Gabah yang terdapat pada ruang deteksi juga terlihat berwarna merah sama seperti beras. Hasil konversi *RGB* ke *HSV* tersebut dikonversi lagi menjadi biner seperti terlihat pada Gambar 21, hasil binarisasi memperlihatkan bahwa kutu beras jenis *Sitophylus Oryzae L* diubah menjadi putih (1) sedangkan beras diubah menjadi hitam (0), namun batu, gabah dan celah-celah antar beras tidak terlihat lagi dan dianggap sebagai *background*. Hasil *filtering* dengan *dilasi* (Gambar 22) menunjukkan kutu beras jenis *Sitophylus Oryzae L* terlihat lebih melebar dan tebal, namun terlihat 1 titik *noise* dibawah kutu bagian atas. Saat menggunakan filter *opening* (Gambar 23). Proses *erosi* dan *dilasi* pada filter *opening* bisa menghilangkan *noise* yang ada secara keseluruhan sekaligus menebalkan citra kutu beras jenis *Sitophylus Oryzae L*. Kemudian saat proses *smoothing* dengan *Gaussian Blurr* (Gambar 24), citra yang dihasilkan terlihat halus merata. Sehingga pada citra pendeteksian *contour area* (Gambar 25), terlihat semua kutu beras jenis *Sitophylus Oryzae L* bisa terdeteksi dengan sempurna.

Pemilihan jenis filter morfologi dilasi dan *opening* dalam percobaan ini dilatar belakangi oleh kondisi objek deteksi yang sangat kecil dan kemungkinan *noise-noise* yang timbul. Sehingga setelah proses binarisasi objek deteksi yang kecil harus diperbesar terlebih dahulu dengan dilasi, selanjutnya untuk menghilangkan *noise* digunakan *opening*. Proses *opening* yang pertama yaitu *erosi* akan membuat *noise-noise* kecil menjadi hilang namun tidak menutup kemungkinan proses ini juga akan mengikis objek yang dideteksi yaitu kutu beras jenis *Sitophylus Oryzae L*, sehingga objek deteksi menjadi kecil Kembali. Kemudian proses *opening* kedua yaitu dilasi, menjadikan objek deteksi yang sempat terkikis oleh proses *opening* pertama menjadi lebar kembali.

Selanjutnya lebar *contour area* yang digunakan dalam percobaan ini adalah 400-1000. Lebar *contour area* diperoleh dengan *trial and error* terlebih dahulu sampai ditemukan komposisi *contour area* yang sesuai

IV. KESIMPULAN

Penerapan metode *HSV-TCA* dalam percobaan ini bisa membedakan objek kutu beras jenis *Sitophylus Oryzae L* yang berukuran kecil dengan *background*nya yaitu kumpulan beras sekaligus bisa mendeteksi keberadaan kutu beras secara *real-time*. Pada percobaan menggunakan beras rusak berketu yang sudah tidak layak makan, hasil pendeteksian mencapai keberhasilan 88,5% sedangkan pada percobaan menggunakan beras baik layak

makan yang diberi kutu pada bagian atasnya, menunjukkan tingkat keberhasilan pendeteksian mencapai 100%. Berdasarkan presentase antara pendeteksian pada beras rusak dan tidak layak makan dengan beras baik layak makan, dapat dilihat bahwa tingkat keberhasilan pendeteksian kutu beras jenis *Sitophylus Oryzae L* dalam ruang deteksi bergantung pada kualitas beras. Karena jika berasnya rusak maka pendeteksian kutu beras menjadi berkurang disebabkan oleh adanya *noise* berupa celah-celah antar beras dan warna beras yang tidak terlalu cerah atau sedikit coklat serta kondisi beras yang sudah menepung. Selanjutnya tingkat keberhasilan dari pengujian ini juga dipengaruhi oleh pencapaian dalam ruang deteksi. Jika cahaya dalam ruang deteksi terang merata, maka pendeteksian akan semakin baik. Karena kamera *webcam* ini ibarat mata manusia. Jika dalam ruang gelap mata akan sulit untuk melihat objek-objek kecil, begitu juga dengan kamera *webcam* yang digunakan. Percobaan ini sekaligus bisa membedakan dengan jelas kutu beras yang di deteksi yaitu jenis *Sitophilus Oryzae L*.

UCAPAN TERIMA KASIH

Penelitian ini didukung sepenuhnya oleh Universitas Pertahanan Republik Indonesia dalam program Penelitian Dosen Internal UNHAN RI TA 2021.

DAFTAR PUSTAKA

- [1] D. J. Bora, A. K. Gupta, and F. A. Khan, "Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Image Segmentation," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 5, no. 2, pp. 192–203, 2015, [Online]. Available: <http://arxiv.org/abs/1506.01472>
- [2] S. Andrea, "Implementasi metode color model filtering hsv untuk mendeteksi bola pada robot sepak bola beroda," Universitas Sumatera Utara, 2018. [Online]. Available: <https://repositori.usu.ac.id/handle/123456789/16379>
- [3] A. Hastawan, R. Septiana, and Y. Windarto, "Perbaikan Hasil Segmentasi HSV Pada Citra Digital Menggunakan Metode Segmentasi RGB Grayscale," *Edu Komputika J.*, vol. 6, no. 1 SE-Articles, Jun. 2019, doi: 10.15294/edukomputika.v6i1.23025.
- [4] A. Sulistyowati, Y. S. Hariyani, and A. Novianti, "Perancangan aplikasi pembaca warna dan bentuk berbasis pengolahan citra untuk daftar katalog perpustakaan," *e-Proceeding Appl. Sci.*, vol. 4, no. 3, pp. 2554–2566, 2018.
- [5] F. Islami, "Journal of Computer Networks , Architecture and High Performance Computing Implementation of HSV- based Thresholding Method for Iris Detection Journal of Computer Networks , Architecture and High Performance Computing," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 3, no. 1, pp. 97–104, 2021.
- [6] A. K. Panggabean, A. Syahfaridzah, and N. A. Ardiningih, "Mendeteksi objek berdasarkan warna dengan segmentasi warna hsv menggunakan aplikasi matlab," *J. Manag. Inform. dan Komputerisasi Akunt.*, vol. 4, no. 2, pp. 94–97, 2020.
- [7] H. Hendrival and F. P. Romadani, "KAJIAN KERENTANAN DAN KERUSAKAN BERAS LOKAL PROVINSI SUMATERA SELATAN TERHADAP HAMA PASCAPANEN *Sitophilus oryzae L.* (Coleoptera: Curculionidae)," *J. Biota*, vol. 4, no. 2, pp. 90–97, 2018, doi: 10.19109/biota.v4i2.2104.
- [8] M. Isnaini, E. Rosa Pane, and S. Wiridianti, "PENGUJIAN BEBERAPA JENIS INSEKTISIDA NABATI TERHADAP KUTU BERAS (*Sitophilus oryzae L.*)," *J. Biota*, vol. 1, no. 1, pp. 1–8, 2015.
- [9] J. Manueke and J. Pelealu, "KETERTARIKAN HAMA *Sitophilus oryzae* PADA BERAS, JAGUNG PIPILAN KACANG TANAH, KACANG KEDELAI, DAN KOPRA," *Eugenia*, vol. 21, no. 2, pp. 70–79, 2015, doi: 10.35791/eug.21.2.2015.9706.

- [10] "C 270 HD Webcam." <https://www.logitech.com/id-id/products/webcams/c270-hd-webcam.960-000584.html> (accessed Oct. 06, 2021).
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [12] G. H. Joblove and D. Greenberg, "Color Spaces for Computer Graphics.," *Comput Graph*, vol. 12, no. 3, pp. 20–25, 1978, doi: 10.1145/965139.807362.
- [13] A. Ford and A. R. Alanrobertsrdbbccouk, *Colour Space Conversions*, vol. 1998. Pynton: Charles, 1998.
- [14] A. R. Smith, "Color Gamut Transform Pairs," *IEEE Comput. Society Press*, 1999.
- [15] X. . Jiang and P.J.Scot, "Advance Metrology," 2020. www.sciencedirect.com
- [16] A. . Bovik, "The Essential Guide to Image Processing," 2009. www.sciencedirect.com
- [17] A. Yulio, "Operasi Morfologi Pada Pengolahan Citra." <https://devtrik.com/opencv/operasi-morfologi-pada-pengolahan-citra> (accessed Oct. 06, 2021).
- [18] Admin, "Image blurring menggunakan opencv," 2021. <https://skillplus.web.id> (accessed Feb. 10, 2022).
- [19] Anonim, "Smoothing Images." https://docs.opencv.org/4.5.3/dc/dd3/tutorial_gaussian_median_blur_bilateral_filter.html
- [20] L. Tan, *Digital signal processing fundamentals and application*. Georgia: Gergia : Elsevier, 2008. [Online]. Available: https://www-elec.inaoep.mx/~jmram/Digital_Signal_Processing__LI_TAN.pdf
- [21] R. Laganiere, *OpenCV 2 Computer Vision Application Programming Cookbook, First*. Brimingham: Packt Publishing, 2011.