

APLIKASI PENENTUAN TINGKAT KUALITAS TELUR AYAM BERDASARKAN WARNA DAN TEKSTUR CITRA KERABANG DENGAN METODE *HUE, SATURATION, VALUE*

Cahya Rahmad¹, Erfan Rohadi², Erlangga Adha Widyatama³

^{1,2,3} Jurusan Teknologi Informasi, Politeknik Negeri Malang

¹cahya.rahmad@polinema.ac.id, ²erfanr@polinema.ac.id, ³erlanggaaw@gmail.com

Abstrak

Pengolahan citra digital didalam bidang *computer vision*, dapat menjadi solusi untuk mengetahui kualitas kesegaran dari sebutir telur dengan melakukan klasifikasi yang bertujuan untuk menentukan tingkat kesegaran telur meliputi kualitas internal dan eksternal. Klasifikasi dilakukan dengan menggunakan metode *Hue, Saturation, Value* (HSV) untuk menentukan warna dari kerabang telur. Pengolahan citra digital pada citra telur dilakukan dengan pra pengolahan dengan merubah citra asli telur menjadi citra *Hue, Saturation, Value*. Program dirancang dapat menentukan sebuah telur mempunyai tingkat kesegaran yang bagus atau tidak dengan informasi telur memiliki usia penyimpanan yang berbeda antara satu dengan yang lain.

Kata kunci : Pengolahan citra, kerabang telur, klasifikasi, HSV

1. Pendahuluan

Telur merupakan komoditas pangan yang paling banyak diminati oleh masyarakat Indonesia. Untuk memenuhi kebutuhan konsumen akan konsumsi telur yang terus meningkat, maka industri telur menjadi industri besar di Indonesia. Kebutuhan yang tinggi akan telur harus juga disertai dengan kualitas telur yang bagus pula. Berdasarkan hal tersebut, perusahaan-perusahaan produksi telur bersaing tidak hanya dalam hal menghasilkan telur dalam bentuk yang bagus, namun juga kaya akan nutrisi seperti DHA, Omega3, dan sebagainya.

Dalam produksi untuk penjualan telur, penggolongan atau klasifikasi telur menjadi satu proses penting yang dibutuhkan untuk mengontrol kualitas dari telur yang akan dihasilkan. Dalam kenyataannya, kualitas telur dipengaruhi oleh kualitas internal dan kualitas eksternal. Pada umumnya, kualitas telur yang bagus memiliki permukaan yang lembut, bentuk yang bagus dan tidak pecah. Selain itu kualitas telur dapat dilihat dari adanya titik darah (*bloodspot*), kulit telur yang retak, dan telur yang terindikasi busuk. Oleh karena itu diperlukan sebuah teknik untuk melakukan deteksi terhadap hal tersebut agar diperoleh telur dengan kualitas yang bagus Maimunah (2015).

Pengawasan mutu telur dapat dilakukan terhadap keadaan fisik, kesegaran isi telur, pemeriksaan kerusakan, dan pengukuran komposisi fisik. Keadaan fisik dari telur mencakup hal ukuran (berat, panjang, dan lebar), warna (putih, agak kecoklatan, coklat), kondisi kulit telur (tipis dan tebal), rupa (bulat dan lonjong), dan kebersihan kulit telur. Secara subyektif mutu telur utuh dapat dinilai dengan cara *candling*, yaitu dengan, cara meletakkan telur dalam jalur sorotan sinar (matahari maupun lampu listrik) yang kuat

sehingga memungkinkan pemeriksaan kulit telur dan bagian dalam telur. Pada tingkat pengecer pemeriksaan telur umumnya dilakukan dengan cara peneropongan dengan sumber cahaya matahari maupun lampu pijar yang dilakukan oleh seorang pekerja. Dengan cara ini adanya keretakan pada kulit telur dapat ditemukan, juga posisi dari kuning telur, ukuran dan posisi kantung udara, bintik darah (*bloodspot*), kerusakan oleh mikroorganisme dan pertumbuhan jamur.

Oleh karena itu diperlukan sebuah sistem yang dapat membantu pihak-pihak tersebut dalam memantau kualitas dari telur ayam yang dihasilkan, salah satunya dengan menggunakan pengolahan citra digital. Teknik pengolahan citra digital telah digunakan secara luas dalam berbagai aplikasi dalam *computer vision*, berbagai teknik pengolahan citra digital misalnya digunakan dalam *robotic*, pengklasifikasian objek, sistem *biometry*, medical visualization, perbaikan dan pemugaran citra, industrial *inspection* dan *human computer interface* R. Ibrahim, *et al* (2012).

Melalui kelebihan tersebut maka, penulis mencoba mengangkat suatu ide yang memerlukan metode untuk menganalisa jumlah objek pada citra untuk menentukan volume suatu objek pada sebuah citra. Sistem tersebut dapat melakukan input citra, dimana citra tersebut akan diolah dengan suatu metode sehingga dapat menghasilkan sebuah citra yang memperlihatkan warna kerabang dalam sebuah telur sehingga dari hasil tersebut dapat menunjukkan tingkat kualitas sebuah telur. Dari sistem ini nantinya diharapkan sistem ini dapat digunakan pada instansi-instansi yang memang membutuhkan, dan juga dapat digunakan oleh masyarakat luas.

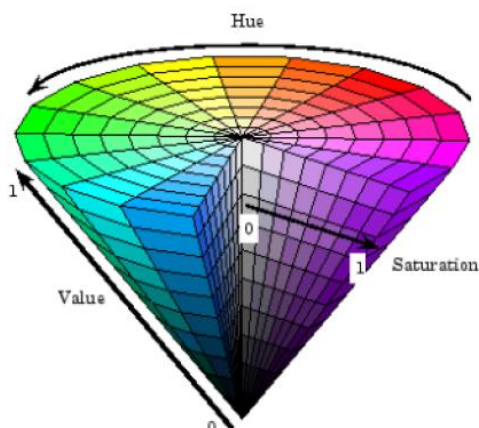
2. Dasar Teori

2.1 Pengolahan Citra

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Secara matematis, citra merupakan fungsi kontinu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Representasi dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*) atau elemen terkecil dari sebuah citra.

Citra diperoleh dari suatu pengambilan atau pengolahan data pada objek dengan menggunakan alat perekam, data yang didapatkan melalui kegiatan interpretasi citra secara manula dengan alat bantu kamera digital. Citra berwarna (*true color*) pada dasarnya terdiri dari unsur warna merah, hijau, dan biru (*red, green, dan blue*, atau sering disingkat RGB). Ketiga komponen warna ini menentukan warna suatu objek. Dalam penelitian ini penulis menggunakan satuan *Hue, Saturation, Value* (HSV) dalam pengembangannya, dimana dalam pendefinisian *Hue* menyatakan warna sebenarnya, seperti merah, violet, dan kuning. *Saturation* menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. *Value* adalah atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warna R. D. Kusumanto, *et al* (2011).

Karena model warna HSV (*Hue Saturation Value*) ini merupakan model warna yang diturunkan dari model warna RGB (*Red Green Blue*) maka untuk mendapatkan hasil warna HSV harus dilakukan proses konversi warna dari RGB ke HSV, dengan menggunakan ruang warna sebagai acuan untuk pendefinisian warna tersebut.



Gambar 1. Ruang Warna HSV

$$H = \begin{cases} 0 \frac{G - B}{\Delta} \text{ mod } 6 & Cmax = R \\ 60 \frac{B' - R'}{\Delta} + 2 & Cmax = G \\ 60 \frac{R' - G'}{\Delta} + 4 & Cmax = B \\ 0 & \Delta = 0 \end{cases} \quad (1)$$

$$H = \begin{cases} 0 & Cmax = 0 \\ \frac{\Delta}{MAX} & Cmax \neq 0 \end{cases}$$

$$V = Cmaxx$$

Keterangan :

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$Cmax = \max(R', G', B')$$

$$Cmin = \min(R', G', B')$$

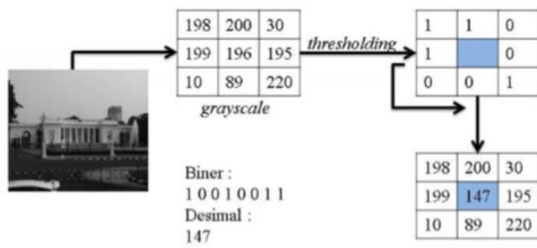
$$\Delta = Cmax - Cmin$$

2.2 Local Binary Patterns (LBP)

Local Binary Pattern merupakan metode yang membandingkan piksel tetangga dengan piksel pusat sehingga diperoleh nilai biner pada matriks, kemudian biner tersebut dihitung dan diubah ke dalam bentuk decimal. *Local Binary Pattern* telah terbukti sebagai *descriptor* tekstur terbaik dalam variasi perubahan tingkat abu-abu monoton dan efisiensi komputasi, sehingga cocok untuk pengolahan gambar perihall keperluan analisis. Operator *Local Binary Pattern* dapat dilihat sebagai pendekatan kesatuan dengan model statistik dan stuktur tradisional yang berbeda dari analisis tekstur. Secara sederhana, *Local Binary Pattern* adalah sebuah kode biner yang menggambarkan pola tekstur local. Hal ini dibangun dengan lingkungan batas nilai abu-abu dari pusatnya T. Ahonen, *et al* (2006).

Metode *local binary pattern* dengan ketetanggaan piksel secara melingkar (*circular*), menggunakan sejumlah koordinat tetangga secara rapat pada sebuah lingkaran dengan menggunakan radius (R) tertentu yang perbusat pada sebuah piksel T. Ahonen, *et al* (2004). Pada LBP dasar, tahap pertama adalah memilih blok berukuran 3x3 menggunakan 8 piksel ketetanggaan in dari sebuah piksel tengah ic. Piksel ketetanggaan ke-n tersebut melalui proses threshold menggunakan nilai keabuan dari piksel tengah seperti yang ditunjukkan pada persamaan (2). Kode *binary* hasil operator LBP piksel ketetanggaan akan digunakan untuk mempresentasikan fitur dari piksel tengah ic (M. Departemen, I. Komputer, and F. Matematika, 2013).

$$LBP(x_c, y_c) = \sum_{n=0}^7 s(i_n - i_c) 2^n \quad (2)$$



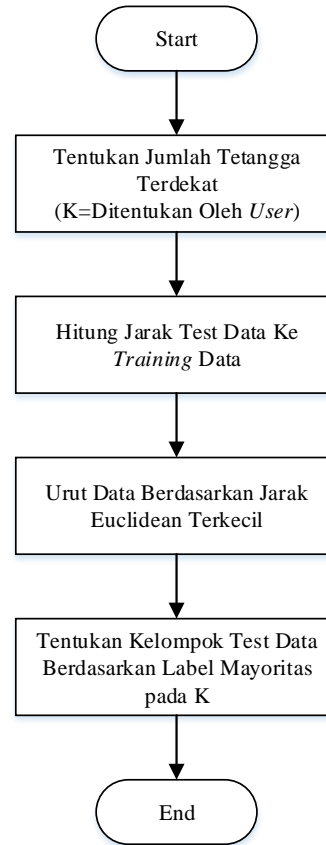
Gambar 2. Ilustrasi Local Binary Pattern

Gambar 2 merupakan ilustrasi dari proses LBP. Langkah pertama adalah melakukan proses *thresholding* pada piksel ketetanggaan yang menggunakan nilai keabuan dari piksel tengah dengan menggunakan persamaan (5) sebagai fungsi *thresholding* jika hasilnya lebih besar atau sama dengan 0 maka diberi nilai 1 dan jika hasilnya kurang dari 0 maka diberi nilai 0. Setelah itu, nilai biner piksel ketetanggaan akan disusun berlawanan arah jarum jam dan 8 bit biner tersebut dikonversikan ke dalam nilai *decimal* untuk menggantikan nilai piksel tengah ic.

2.3 k-Nearest Neighbor

Merupakan sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimendi mempresentasikan fitur diri data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai dengan kelas c, jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat pada titik tersebut. Estimasi kerapatan non-parametrik bisa dihitung dengan dua cara. Cara pertama adalah dengan memilih nilai tetap volume V dan menentukan k dari data. Hal ini dilakukan dalam metoda yang disebut estimasi kerapatan *kernel* (KDE atau *Kernel Density Estimation*). Cara yang kedua yaitu dengan memilih jumlah tetap k dan menentukan volume V yang sesuai dari data.

Tujuan dari algoritma k-NN adalah untuk mengklasifikasikan objek baru berdasarkan atribut dari training samples. Dimana hasil dari sample uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada k-NN. Pada proses pengklasifikasian, algoritma ini tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Algoritma k-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru. Jarak yang digunakan adalah jarak *Euclidean Distance*, jarak ini merupakan jarak yang paling umum digunakan pada data numerik, seperti yang ditunjukkan pada persamaan (3). Dibawah ini adalah alur *flowchart* dari metode k-NN sebagai berikut.



Gambar 3. Flowchart k-NN

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (3)$$

Keterangan:

- $d(x_i, x_j)$ = Jarak *Euclidean* (*Euclidean Distance*)
- (x_i) = record ke-i
- (x_j) = record ke-j
- (a_r) = data ke- r
- i, j = 1,2,3,...n

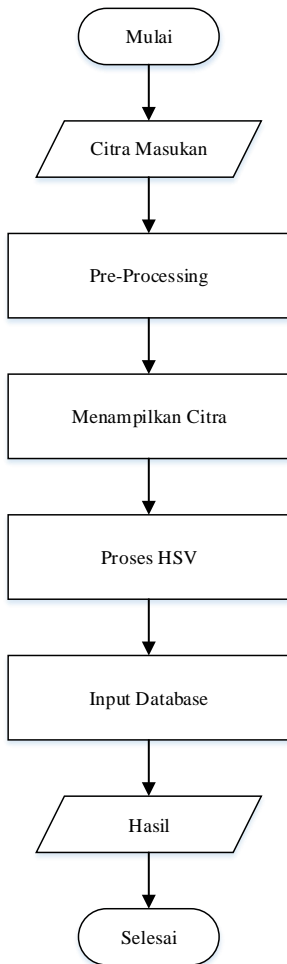
Jika dibandingkan dengan metode klasifikasi yang lain, metode ini memiliki tingkat keakuratan yang cukup tinggi karena data yang masuk akan diklasifikasikan berdasarkan kemiripan ciri yang ada dari data sebelumnya yang sudah diklasifikasikan. Namun, pada algoritma k-NN perlu menentukan nilai dari parameter k (jumlah tetangga terdekat) dan pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik N. Krisandi, *et al* (2013).

3. Implementasi

3.1 Training

Proses pengolahan data *training* dilakukan untuk memperoleh nilai kelas warna kesegaran telur ayam yang akan digunakan sebagai acuan data perbandingan

dengan data uji. Pada proses *training* dimulai dari pengambilan data gambar telur ayam menggunakan kamera Nikon dengan jarak 50 cm yang di letakkan pada bagian atas wadah khusus. Kemudian gambar telur yang telah didapatkan dilakukan proses *cropping* sehingga hasil citra yang dipakai ada adalah citra *cropping* menampilkan kerabang telur ayam.



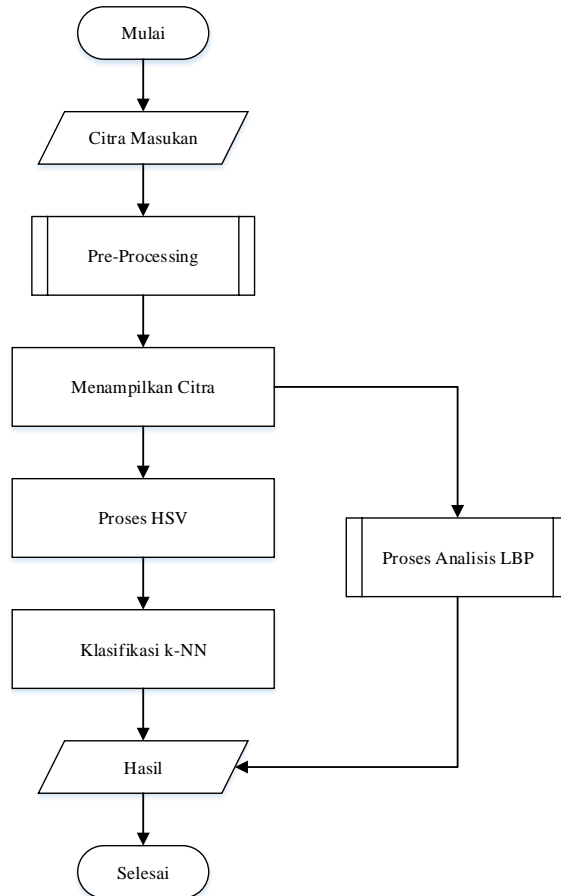
Gambar 4. Flowchart Training

Kemudian mencari nilai rerata dari setiap jumlah *Hue*, jumlah *Saturation*, dan jumlah *Value*, dari citra pada proses ekstraksi fitur dimana proses ini menunjukkan nilai intensitas rerata tiap indeks warna HSV. Kemudian langkah selanjutnya adalah menyimpan nilai dari warna HSV yang telah didapatkan dari citra hasil *crop*, dimana nilai-nilai tersebut nantinya akan digunakan sebagai acuan nilai untuk proses *testing*. Penyimpanan nilai dari data latih disimpan pada file *database* yang telah disiapkan, sehingga pada saat melakukan proses testing data nilai data *training* akan dapat dipanggil oleh *user*.

3.2 Testing

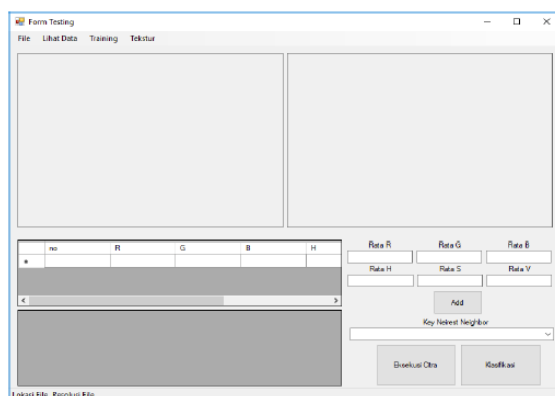
Pada proses *testing* ini dilakukan pengujian aplikasi untuk mendeteksi kesegaran telur ayam. Proses testing ini dilakukan sama seperti proses *training* yaitu

dimulai input citra telur ayam, dan selanjutnya akan dilakukan *preprocessing* dan selanjutnya dilakukan segmentasi menggunakan konversi warna HSV. Kemudian dilakukan klasifikasi dengan menggunakan metode klasifikasi k-NN. Seperti yang ditunjukkan pada *flowchart* dibawah ini.



Gambar 5. Flowchart Testing

3.3 Desain Sistem



Gambar 6. Desain Sistem

Pada desain sistem diatas, terdapat dua *image box* yang berfungsi untuk menampilkan citra telur yang akan di proses, satu *image box* berfungsi sebagai citra awal telur dan satu lagi berfungsi menampilkan citra

telur setelah dilakukan proses pengolahan citra. Selain itu pada bagian atas *image box* terdapat menu yang memiliki fungsi tersendiri, yakni:

- *File*: Berfungsi untuk menginputkan data gambar telur ayam yang akan dilakukan deteksi kesegaran.
- *Lihat Data*: Berfungsi untuk melihat nilai data latih yang telah tersimpan di *file* database.
- *Training*: Digunakan untuk menampilkan *form* baru yang berfungsi untuk proses training data telur yang akan diuji.
- *Tekstur*: Berfungsi untuk menampilkan form perhitungan tekstur telur.

Dalam form *testing* juga terdapat tombol sebuah data *grid*, yang berfungsi untuk menampilkan data citra telur yang telah melalui proses *training* dengan menggunakan metode yang sama pada saat proses *testing*. Hal ini ditujukan agar pada saat citra telur akan melalui proses *testing* kita akan melihat data yang telah tersimpan di dalam *database* dan apabila proses *testing* telah selesai maka akan dicocokkan dengan data yang ada pada data *grid*.

Selain itu juga terdapat dua tombol pada bagian bawah form yakni tombol *start* dan tombol data uji dan kolom *hue*, *saturation*, *value* serta tiga kolom rata-rata. Setiap tombol dan kolom tersebut memiliki fungsinya masing-masing yakni:

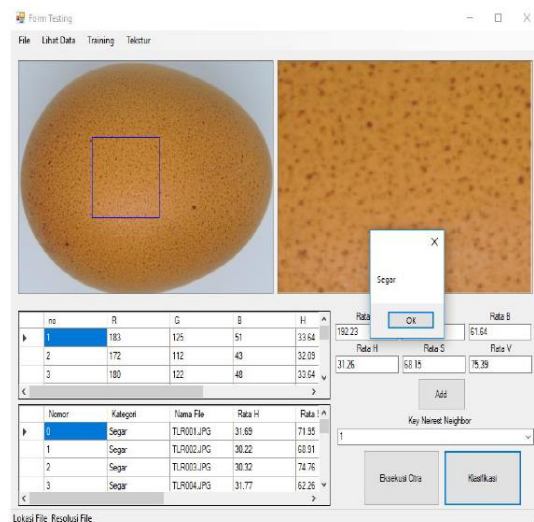
- *Tombol Start*: Berfungsi sebagai tombol eksekusi pada saat citra telur yang telah berada dalam *image box* siap untuk dilakukan pengolahan citra menggunakan metode yang telah diterapkan.
- *Tombol Data uji*: Tombol ini berfungsi untuk menampilkan data uji yang tersimpan dalam database ke data *grid*, sehingga saat pengujian kesegaran telur ayam telah selesai hasil proses tersebut dapat dibandingkan dengan data yang terdapat dalam data *grid*.
- *Kolom Hue*: Berfungsi untuk menampilkan nilai dari citra hasil pengujian menggunakan metode HSV, dalam hal ini nilai yang ditunjukkan adalah nilai dari *hue*.
- *Kolom Saturation*: Berfungsi untuk menampilkan nilai dari citra hasil pengujian menggunakan metode HSV, dalam hal ini nilai yang ditunjukkan adalah nilai dari *saturation*.
- *Kolom Value*: Berfungsi untuk menampilkan nilai dari citra hasil pengujian menggunakan metode HSV, dalam hal ini nilai yang ditunjukkan adalah nilai dari *value*.

3.4 Hasil Sistem

Hasil dari sistem ini nantinya adalah nilai dari citra yang telah diproses berupa nilai hsv. Selain itu juga menampilkan citra telur ayam dalam citra HSV. Seperti yang ditunjukkan pada gambar 7. Juga dapat dilihat hasil dari *preprocessing* citra yang diambil dengan metode *grayscale luminance*.

Dari hasil yang akan diharapkan apabila sistem ini berhasil, terdapat beberapa kemungkinan yang mempengaruhi kualitas dari citra yang dihasilkan oleh sistem ini, seperti:

- *Ukuran piksel kamera*
Ukuran piksel pada kamera yang dipakai mempengaruhi hasil citra yang akan dihasilkan. Semakin tinggi piksel yang digunakan maka hasil citra akan semakin jelas.
- *Sudut pengambilan gambar*
Sudut pengambilan citra telur mempengaruhi dalam menentukan jumlah piksel yang dihasilkan. Misalkan apabila mengambil citra telur di bagian depan maka citra yang dihasilkan menampilkan latar belakang lebih banyak dibandingkan citra telur. Maka dari itu pengambilan telur dilakukan dari atas agar memungkinkan citra telur yang dihasilkan akan lebih baik.
- *Warna latar belakang*
Pemilihan warna sangat mempengaruhi hasil citra yang diambil. Dengan memilih warna *background* yang tepat hasil citra yang dimiliki akan semakin baik. Apabila telur diambil dengan warna latar belakang hitam maka pada *invers* citra akan menghasilkan warna putih. Sebaiknya menggunakan latar belakang berwarna hitam, agar citra telur lebih baik.



Gambar 7. Hasil *Testing*

Tabel 1 menunjukkan hasil dari proses pengolahan citra yang dilakukan dengan metode HSV dan metode GLCM dengan klasifikasi k-NN. Dapat dilihat nilai *hue* pada citra awal hingga hasil citra menunjukkan peningkatan nilai, hal ini diakibatkan karena penurunan kecerahan dari warna kerabang telur ayam yang sebanding dengan lama penyimpanan telur ayam tersebut. Semakin lama telur ayam disimpan, maka akan semakin turun tingkat kecerahan atau semakin redup warna dari kerabang telur ayam tersebut.

Tabel 1. Hasil Proses Pengujian

No	Nilai Citra Awal RGB	Nilai Citra Hue	Klasifikasi	Hasil
1	(121,127,55)	31.94511908	Segar	Sesuai
2	(184,130,60)	34.36058276	Segar	Sesuai
3	(158,112,52)	31.54540707	Segar	Sesuai
4	(198,148,79)	35.2667537	Segar	Tidak Sesuai
5	(188,129,61)	32.34292633	Kurang Segar	Sesuai
6	(184,143,81)	36.71351762	Tidak Segar	Sesuai

4. Kesimpulan

Penggunaan metode *hue, saturation, value* yang diterapkan untuk mengidentifikasi kesegaran telur dinilai dapat mendapatkan hasil yang sesuai dengan harapan. Dari 6 data yang di uji citranya mendapatkan 5 data yang sesuai dan 1 data yang tidak sesuai, sehingga total hasil yang didapatkan berkisar 80%, hal ini juga sesuai dengan lama penyimpanan telur ayam.

Beberapa komponen yang mempengaruhi identifikasi kesegaran telur ayam antara lain, jumlah piksel kamera, jarak pengambilan citra, dan pencahayaan. Apabila ketiga komponen itu terpenuhi maka hasil citra yang didapatkan sebelum melalui proses akan memiliki kualitas yang baik. Begitu pula dengan hasil citra setelah melalui pengolahan citra, akan didapatkan hasil yang sesuai dengan kondisi citra tersebut.

Pada studi selanjutnya, diperlukan eksplorasi yang lebih mendalam tentang penurunan kesegaran telur ayam dengan dilihat dari kerabang telur ayam tersebut dan metode yang lebih sesuai. Perlu diteliti lebih lanjut mengenai kualitas mutu telur ditinjau dari aspek lain misalkan bentuk, keretakan kerabang telur, warna atau kualitas internal telur seperti kuning telur, putih telur sehingga diperoleh kulit telur yang lebih bagus ditinjau dari kualitas eksternal dan internal telur.

Daftar Pustaka :

- Maimunah. (2015): *Deteksi kebersihan kerabang telur ayam berdasarkan pengolahan citra digital*, vol. 3, no. 1, pp. 41–49.
- M. Departemen, I. Komputer, and F. Matematika. (2013): *Aplikasi Mobile Untuk Identifikasi Tumbuhan Obat Menggunakan Local Binary Pattern Dengan Klasifikasi Probabilistic Neural Network obat, Indonesia memiliki lebih dari 38 000 spesies tumbuhan obat (Bappenas 2003). yang terdapat pada buku atau melakukannya,*” no. Bappenas, pp. 1–15.
- N. Krisandi, Helmi, and B. Prihandono. (2013): *Algoritma k-Nearest Neighbor Dalam Klasifikasi Data Hasil Produksi Kelapa Sawit Pada Pt. Minamas Kecamatan Parindu*, Proc. 2014 Int. Symp. Inf. Technol. ISIT 2014.
- R. D. Kusumanto, A. N. Tompunu, and S. Pambudi. (2011): *Klasifikasi Warna Menggunakan*

Pengolahan Model Warna HSV, J. Ilm. Tek. Elektro.

- R. Ibrahim, Z. M. Zin, N. Nadzri, M. Z. Shamsudin, and M. Z. Zainudin. (2012): *Egg's Grade Classification and Dirt Inspection Using Image Processing Techniques*, Proc. World Congr. Eng.
- T. Ahonen, A. Hadid, and M. Pietikäinen. (2004): *Face Recognition with Local Binary Patterns*.
- T. Ahonen, A. Hadid, and M. Pietikäinen. (2006): *Face description with local binary patterns: Application to face recognition*, IEEE Trans. Pattern Anal. Mach. Intell.