

Pengklasifikasian *Grade* Telur Ayam Negeri menggunakan Klasifikasi *K-Nearest Neighbor* berbasis Android

NUR IBRAHIM¹, TASYA FIKRIYAH BACHERAMSYAH², BAMBANG HIDAYAT³,
SJAFRIL DARANA⁴

^{1,2,3}Universitas Telkom

⁴Universitas Padjajaran

Email: nuribrahim@telkomuniversity.ac.id

Received 23 Januari 2018 | Revised 14 Mei 2018 | Accepted 25 Mei 2018

ABSTRAK

Telur ayam negeri yang dibeli oleh masyarakat Indonesia di toko swalayan, pasar, ataupun di peternakan memiliki grade yang berbeda-beda. Pada penelitian sebelumnya, telah dilakukan pembuatan sistem pengklasifikasian telur ayam dengan berbasis Windows, namun belum dapat digunakan secara praktis oleh masyarakat. Penelitian ini dilakukan agar masyarakat dapat menggunakannya di lapangan dalam mengklasifikasikan grade telur ayam negeri, dimana pengklasifikasian grade pada telur ayam negeri ini menggunakan klasifikasi K-Nearest Neighbor (K-NN) yang berbasis android. Berdasarkan hasil pengujian, sistem ini dapat mengklasifikasikan grade telur ayam negeri dengan tingkat akurasi sebesar 80% (dibandingkan menggunakan Haugh Unit Micrometer) menggunakan parameter layer 4 (grayscale), metode penghitungan jarak cosine, dan nilai k=1 dimana jumlah tetangga yang dibandingkan pada algoritma K-NN adalah 1.

Kata kunci: K-NN, telur ayam negeri, android.

ABSTRACT

Chicken eggs purchased by Indonesian people in supermarkets, markets, or farms have different grades. In the previous research, the classification system of chicken eggs has been done in the windows platform, but it cannot be used practically by the people. This research was made so the people can use it on the field to classify chicken eggs grade, using the classification of K-Nearest Neighbor (K-NN) based on android. Based on testing results of this system, can classify eggs grade chicken with an accuracy of 80% (compared with Haugh Unit Micrometer) using layer 4 (grayscale), cosine distance method, and value of k=1 which is the total of compared neighborhood in K-NN algorithm is 1.

Keywords: K-NN, chicken egg, android.

1. PENDAHULUAN

Ada beberapa cara untuk mengetahui kualitas telur yang baik, salah satunya dapat dilihat dari ketebalan putih telur atau *albumen*. Tinggi rendahnya *albumen* adalah patokan untuk mengetahui kesegaran telur. Tinggi rendahnya *albumen* dapat diukur oleh alat *Haugh Unit*. Saat ini warna kuning telur tidak dapat dijadikan patokan utama dalam menentukan kualitas telur ayam, karena perkembangan teknologi memungkinkan kamuflase kuning telur dengan zat kimia atau zat herbal (**Tim Praktikum Produksi Ternak Unggas, 2013**). Oleh karena itu, ketebalan putih telur dapat menjadi alternatif patokan utama dalam menentukan kualitas telur.

Penelitian sebelumnya merancang deteksi kualitas dan kesegaran telur menggunakan berbagai metode ekstraksi ciri, yaitu *Fuzzy Color Histogram* (**Wardani, 2017**), *Histogram Equalization* (**Meirinda, 2016**), dan ekstraksi ciri *Wavelet* (**Basuki, 2016**) dengan metode klasifikasi yang sama yaitu K-NN. Pada klasifikasi *Fuzzy Color Histogram* diperoleh tingkat akurasi 71,87%, sedangkan pada metode *Histogram Equalization* diperoleh akurasi sebesar 76% dan pada metode ekstraksi ciri *Wavelet* diperoleh akurasi sebesar 81,81%. Pada penelitian (**Handayani, 2017**), deteksi kualitas telur ayam menggunakan *Discrete Cosine Transform* (DCT) memiliki tingkat akurasi yang cukup tinggi yaitu sebesar 90.19%, dimana penelitiannya menggunakan *platform windows* dengan *software* Matlab. Kekurangan dari penggunaan *platform windows* adalah tidak mudah digunakan di lapangan karena harus mengambil citra gambar telur ayam dengan menggunakan kamera, kemudian memasukkan citra kamera tersebut ke dalam komputer, dan hasilnya baru bisa diketahui setelahnya. Oleh karena itu, peneliti membuat sistem pengklasifikasian *grade* telur ayam negeri menggunakan klasifikasi *K-Nearest Neighbor* berbasis android agar masyarakat, terutama distributor dan peternak telur ayam negeri dapat lebih mudah melakukan pengukuran kualitas telur di lapangan. Jenis kualitas telur dapat diklasifikasikan menjadi empat kualitas, yaitu : AA (sangat baik), A (Baik), B (Cukup Baik), dan C (Buruk).

1.1 Citra Digital

Citra *digital* merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan *sampling* dan kuantisasi. *Sampling* pada citra menyatakan besar kecilnya ukuran piksel (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin. Dengan kata lain, kuantisasi pada citra menyatakan jumlah warna yang ada pada citra (**Putra, 2010**).

1.1.1 RGB

Suatu citra dalam format RGB terdiri dari tiga bidang citra yang saling lepas, masing-masing terdiri dari warna dasar: merah, hijau dan biru. Suatu warna dispesifikasikan sebagai campuran sejumlah komponen warna dasar. Gambar 1 menunjukkan bentuk geometri dari model warna RGB untuk menspesifikasikan warna menggunakan sistem koordinat cartesian. Spektrum *grayscale* (tingkat keabuan) adalah warna yang dibentuk dari gabungan tiga warna dasar dengan jumlah yang sama, berada pada garis yang menghubungkan titik hitam dan putih (**Kadir & Susanto, 2013**).

Citra dalam format RGB mengandung data dalam bentuk matriks berukuran $r \times g \times b$ yang merepresentasikan warna merah, hijau, dan biru untuk setiap pikselnya. Setiap warna dasar diberikan rentang nilai. Pada layar komputer, nilai rentang paling kecil 0 dan paling besar 255. Pemilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan pada komputer, sehingga total warna yang dapat diperoleh adalah lebih dari 16

juta warna. Warna dari tiap piksel ditentukan oleh kombinasi dari intensitas merah, hijau, dan biru (**Putra, 2010**) seperti pada Gambar 1.



Gambar 1. Citra RGB (Putra, 2010)

1.1.2 *Grayscale*

Citra dalam format *grayscale* terdiri dari satu bidang citra, dimana nilai pada setiap pikselnya merupakan rata-rata dari warna dasar (merah, hijau, dan biru). Nilai-nilai piksel pada citra dalam format *grayscale* menunjukkan intensitas warna, dimana citra jenis ini terdiri dari warna abu-abu. Intensitas tertinggi pada citra ini akan menunjukkan warna putih, dan intensitas terlemah akan menunjukkan warna hitam. Citra dalam format *grayscale* berbeda dengan citra dalam format biner yang hanya terdiri dari warna hitam dan putih. Pada citra dalam format *grayscale*, sangat banyak variasi warna antara hitam dan putih.

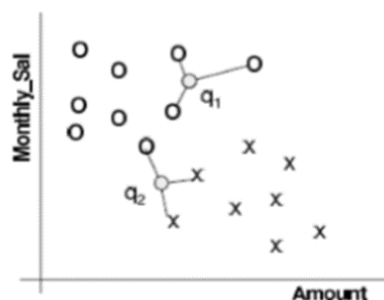
Rentang nilai pada citra dalam format *grayscale* ini sama dengan rentang nilai pada citra dalam format RGB, dimana nilai 0 menunjukkan intensitas terendah (warna hitam) dan nilai 255 menunjukkan intensitas tertinggi (warna putih). Untuk mengubah citra dalam format RGB yang memiliki tiga bidang citra (bidang warna merah, hijau, dan biru) menjadi citra dalam format *grayscale*, maka perlu dilakukan proses konversi nilai piksel citra dengan mencari rata-rata nilai piksel dari bidang warna R (*Red*), G (*Green*) dan B (*Blue*) menjadi nilai G (*Grayscale*), sehingga dapat dituliskan dalam bentuk persamaan sebagai berikut:

$$G = \frac{R + G + B}{3} \quad (1)$$

1.2 *Haugh Unit (HU)*

1.3 *K-Nearest Neighbor (K-NN)*

K-NN (*K-Nearest Neighbor*) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan ciri-ciri data pembelajaran (data latih) yang paling mendekati objek tersebut. Ciri ini direpresentasikan dengan ukuran jarak, sehingga dapat diolah ke dalam hitungan matematis. Persamaan jarak *Euclidean* digunakan untuk mengukur kedekatan jarak (ciri) antara dua obyek, data latih dan data uji. (**Widhiasih et al, 2013**)



Gambar 2. Nilai K=3 Klasifikasi K-Nearest Neighbor (Padraid & Delany, 2007)

Gambar 2 menunjukkan klasifikasi K-NN dengan nilai $K=3$. *Node* q1 akan mendapatkan tiga tetangga terdekatnya, dimana semua tetangga terdekatnya adalah di kelas O, sehingga *node* q1 diklasifikasikan sebagai kelas O. Pada *node* q2, q2 mempunyai 2 tetangga dari kelas X dan satu tetangga dari kelas O. Algoritma K-NN:

1. Tentukan nilai K.
2. Hitung jarak antara data baru ke setiap label data.
3. Tentukan k *labeled data* yang mempunyai jarak yang paling minimal.
4. Klasifikasikan data baru ke dalam label data yang mayoritas K-NN dipilih berdasarkan metrik jarak.

1.3.1 Euclidean Distance

Euclidean distance adalah jarak biasa antara dua titik atau koordinat yang diturunkan dari rumus *pythagoras*. *Euclidean distance* antara titik dan panjang garis yang menghubungkan keduanya adalah ab , dimana ab adalah sisi miring dari garis yang dibentuk pada sumbu x dan sumbu y antara koordinat a dan b.

Sebagai contoh, untuk menghitung jarak antara dua titik X_s dan X_t dengan metode *Euclidean distance*, digunakan rumus (Basuki, 2016):

$$d_{s,t} = \sqrt{\sum_{j=1}^n (s_j - t_j)^2} \quad (3)$$

Keterangan:

s,t : Koordinat titik objek
 n : Jumlah objek

1.3.2 Cityblock Distance

Cityblock distance adalah jarak antara dua titik atau koordinat yang mirip dengan *Euclidean distance*. Namun sedikit berbeda dengan *Euclidean distance* yang mengukur panjang garis yang menghubungkan kedua titik tersebut, *cityblock distance* antara titik a dan b adalah nilai mutlak dari jarak antara a dan b dari sumbu x dan sumbu y.

Untuk menghitung jarak antara dua titik X_s dan X_t dengan metode *cityblock distance*, digunakan rumus: (Basuki, 2016)

$$d_{s,t} = \sum_{i=1}^n |s_j - t_j| \quad (4)$$

Keterangan:

s,t : Koordinat titik objek
 n : Jumlah objek

1.3.3 Cosine Distance

Cosine similarity (cosine) adalah ukuran kesamaan diantara dua vektor dari sebuah *inner product space* yang mengukur kosinus dari sudut diantara dua vektor tersebut. Berbeda dengan dua *distance measure* sebelumnya yang selalu bernilai positif, *cosine similarity distance* memiliki rentang nilai dari -1 hingga 1. Kosinus antara dua vektor dapat dengan mudah diturunkan dari rumus perkalian *dot Euclidean*. Jika vektor a dan b maka nilai kosinus dapat diperoleh dengan melakukan perkalian titik diantara keduanya. (Basuki, 2016)

Dalam *cosine distance*, titik-titik dianggap sebagai vektor, dan dilakukan pengukuran terhadap sudut antara dua vektor tersebut. Untuk memperoleh jarak dua vektor x_s dan x_t , digunakan rumus sebagai berikut: **(Basuki, 2016)**

$$d_{s,t} = 1 - \cos \theta \quad (5)$$

dimana,

$$\cos \theta = \frac{X_s X_t}{|X_s| |X_t|} \quad (6)$$

Keterangan:

s, t : Koordinat titik objek

1.3.4 Correlation Distance

Dalam *correlation distance*, titik-titik dianggap sebagai barisan nilai, jarak antar nilai x_s dan x_t , memakai rumus sebagai berikut: **(Salsabila et al, 2016)**

$$d_{s,t} = 1 - \frac{(x_s - \bar{x}_s)(x_t - \bar{x}_t)}{\sqrt{(x_s - \bar{x}_s)(x_t - \bar{x}_t)} \sqrt{(x_t - \bar{x}_t)(x_t - \bar{x}_t)}} \quad (7)$$

dimana,

$$x_s = \frac{1}{n} + \sum_j x_{sj} \text{ dan } \bar{x}_t = \frac{1}{n} + \sum_j x_{tj} \quad (8)$$

Keterangan:

s, t : Koordinat titik objek

n : Jumlah objek

1.4 Kualitas Telur

Hal yang akan dibahas pada penelitian ini adalah deteksi kualitas telur ayam negeri yang akan digunakan dalam pendeteksian kualitas telur di peternakan, pasar swalayan untuk mengurangi adanya *human error*. Penentuan kualitas telur berdasarkan *Haugh Unit* menurut standar *United State Departement of Agriculture* (USDA) adalah sebagai berikut: **(Tim Praktikum Produksi Ternak Unggas, 2013)**

1. *Grade AA* memiliki tinggi HU putih telur > 72.
2. *Grade A* memiliki tinggi HU putih telur 60-71.
3. *Grade B* memiliki tinggi HU putih telur 31-59.
4. *Grade C* memiliki tinggi HU putih telur < 31.

Haugh Unit merupakan satuan yang digunakan untuk mengetahui kesegaran isi telur, terutama bagian putih telur. Cara mengukurnya, telur harus dipecah lalu ketebalan putih telur diukur dengan alat *micrometer*. Telur yang segar biasanya memiliki putih telur tebal. Semakin tinggi nilai *Haugh Unit* suatu telur, maka kualitas telur semakin baik. Untuk menentukan tinggi HU, digunakan rumus: **(Tim Praktikum Produksi Ternak Unggas, 2013)**

$$HU = 100 \text{ Log } (H + 7,57 - 1,7W^{0,37}) \quad (2)$$

Keterangan:

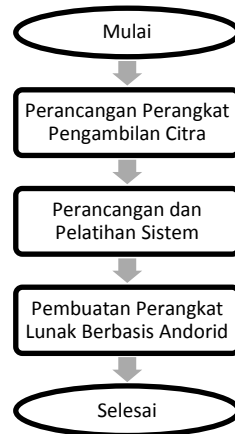
HU : Nilai *Haugh Unit*

H : Tinggi putih telur

W : Berat Telur

2. METODOLOGI PENELITIAN

Tujuan dari penelitian ini adalah untuk memudahkan masyarakat dalam mengukur kualitas telur ayam di lapangan dengan akurat dan efisien, serta mampu mengelompokkannya berdasarkan *grade* telur ayam negeri yaitu *grade* AA (sangat baik), A (baik), B (cukup baik), dan C (buruk). Metodologi yang digunakan pada penelitian ini adalah sebagai berikut:

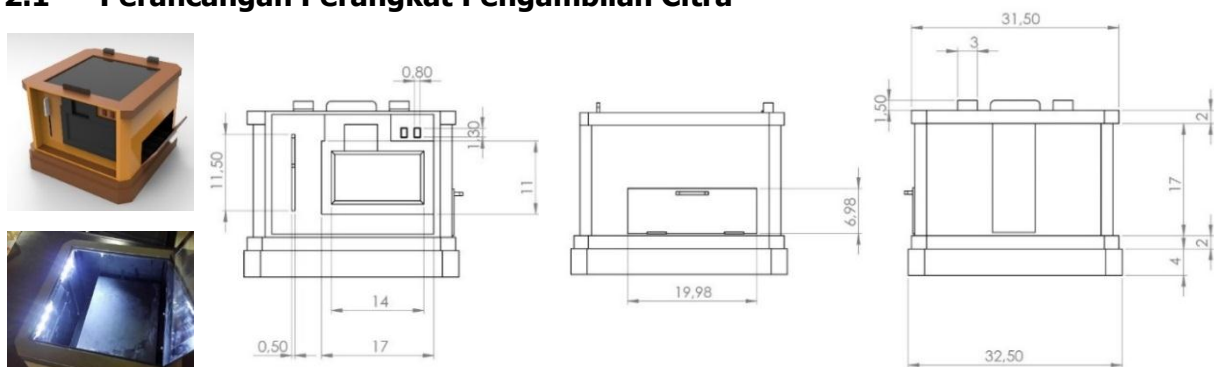


Gambar 3. Flowchart Metodologi Penelitian

Adapun penjelasan mengenai *flowchart* pada Gambar 3 adalah sebagai berikut:

1. Perancangan Perangkat Pengambilan Citra
Perlu dilakukan perancangan perangkat untuk pengambilan citra telur ayam negeri sehingga setiap citra yang didapatkan akan memiliki karakteristik yang sama.
2. Perancangan dan Pelatihan Sistem
Sebanyak 30 telur ayam negeri diambil citranya (10 telur *grade* AA, 10 telur *grade* A, dan 10 telur *grade* B). *grade* C tidak dimasukkan kedalam penelitian ini, karena *grade* C merupakan kualitas telur yang sudah tidak layak konsumsi (busuk). Kemudian, setiap telur yang telah diambil citranya, dihitung nilai HU nya menggunakan HU *micrometer* untuk menentukan *grade* telur tersebut. Data ini kemudian dijadikan sebagai citra latihan dan menjadi *database* yang akan digunakan pada perangkat lunak berbasis android.
3. Pembuatan Perangkat Lunak berbasis Android
Langkah selanjutnya adalah membuat perangkat lunak di android. Perangkat lunak ini akan mengolah citra telur ayam negeri yang akan diuji untuk dibandingkan dengan *database* hasil pelatihan sistem menggunakan klasifikasi K-NN. Hasil klasifikasi citra telur ayam yang diuji, akan ditampilkan pada layar *smartphone* berupa *grading* (AA, A, atau B). Jumlah telur yang diuji sebanyak 55 buah.

2.1 Perancangan Perangkat Pengambilan Citra



Gambar 4. Layout dan Realisasi Media Pengambilan Citra

Gambar 4 menunjukkan desain perangkat dan realisasi perangkat yang digunakan untuk pengambilan citra telur. Telur ayam dipecahkan dan diletakkan di tengah-tengah alas yang berada di dalam perangkat, kemudian diambil citranya menggunakan kamera *handphone* Xiaomi Redmi 4 (dengan spesifikasi kamera 13 *megapixel* dengan *aperture* f/2.2 dan *phase detection autofocus*, serta *platform* OS Android 6.0.1 *Marshmallow*), dimana posisinya berada di depan alas yang jaraknya sudah *fix* sebesar 15,4 cm. Perangkat ini dilengkapi dengan LED sehingga pencahayaan untuk pengambilan setiap citra akan sama.

2.2 Perancangan dan Pelatihan Sistem

Secara umum, desain sistem dari penelitian ini direpresentasikan sebagai berikut :



Gambar 5. Blok Diagram Umum Sistem

Adapun penjelasan mengenai blok diagram pada Gambar 5 adalah sebagai berikut:

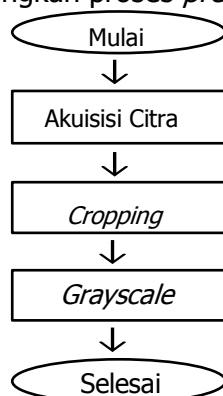
1. Akuisisi citra adalah Pengambilan citra telur menggunakan kamera *handphone* Xiaomi Redmi 4 untuk mendapatkan data gambar.
2. *Preprocessing* adalah proses pengolahan citra *digital*, yang meliputi proses operasi *cropping*, dan *grayscale*.
3. Klasifikasi adalah proses pengelompokkan objek yang membedakan dengan objek lainnya dengan sistem *K-Nearest Neighbor*.

2.2.1 Akuisisi Citra

Akuisisi citra adalah tahap untuk mendapatkan citra *digital*. Citra yang didapat terbagi atas citra latih dan citra uji. Proses pengambilan citra telur ayam negeri adalah dengan memecahkan telur ayam negeri terlebih dahulu diatas permukaan akrilik berwarna hitam *dove*. Kemudian pengambilan citra diambil dari sisi samping telur menggunakan kamera *handphone*. Jumlah data yang diuji sebanyak 55 citra. Pada proses ini akan dimasukkan ke dalam tahap *preprocessing*.

2.2.2 Preprocessing

Gambar 6 menunjukkan langkah-langkah proses *preprocessing*.



Gambar 6. Blok Diagram Tahap Preprocessing

1) Akuisisi Citra

Akuisisi citra adalah tahap untuk mendapatkan citra *digital*. Citra yang didapat terbagi atas citra latih dan citra uji. Pada tahap ini, proses akuisisi citra dilakukan dengan mengambil citra telur ayam negeri menggunakan kamera *handphone* Xiaomi Redmi 4.

2) *Cropping*

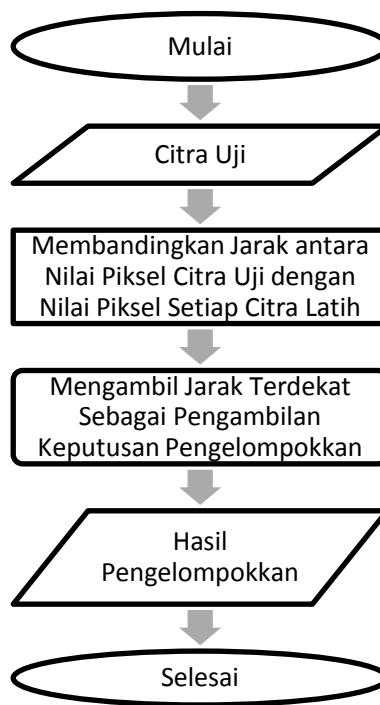
Dalam tahap ini, ukuran citra di-*crop* menyesuaikan ukuran telur, agar lebih mudah untuk dilakukan proses klasifikasi. Ukuran data uji yang digunakan berubah menjadi 2981 x 1018 piksel.

3) *Grayscale*

Pada proses ini dilakukan transformasi warna citra dari RGB menjadi *grayscale* dengan tujuan untuk mereduksi citra tiga dimensi menjadi satu dimensi dengan nilai intensitas yang sama, sehingga lebih mudah di proses dan bisa dijadikan citra masukan pada klasifikasi.

2.2.3 Klasifikasi

Gambar 7 menunjukkan *flowchart* klasifikasi dengan metode K-NN :



Gambar 7. *Flowchart* Klasifikasi dengan Metode K-NN

Adapun penjelasan langkah-langkah dari *flowchart* pada Gambar 7 adalah sebagai berikut:

1. Membandingkan Jarak antara Nilai Piksel Citra Uji dengan Nilai Piksel Setiap Citra Latih
Nilai piksel citra uji akan dibandingkan dengan nilai piksel semua citra latih, dimana akan dicari jarak antara nilai piksel citra uji dengan nilai piksel tiap citra latih. Jarak yang dihitung berdasarkan nilai intensitas keabuan setiap piksel.
2. Mengambil Jarak Terdekat Sebagai Pengambilan Keputusan Pengelompokkan
3. Setelah mendapatkan semua jarak antara nilai piksel citra uji dengan nilai piksel semua citra latih, pengelompokkan citra uji akan dikelompokkan berdasarkan jarak terdekat antara nilai piksel citra uji dengan nilai piksel citra latih.

Pada tahap ini, masukan yang digunakan kedalam proses klasifikasi K-NN adalah hasil dari tahap *preprocessing*. Hal yang dilakukan klasifikasi K-NN terlebih dahulu adalah penentuan nilai K. Penentuan nilai K adalah jumlah tetangga dalam klasifikasi dengan pendekatan K-NN.

Algoritma K-NN berkerja sesuai dengan jarak terpendek dari citra uji ke citra latih. Semakin besar nilai K maka jumlah pembanding akan semakin besar, begitu pula sebaliknya. Analisis yang dilakukan pada penelitian ini adalah pengaruh nilai K yang digunakan terhadap akurasi

sistem yang terbaik dalam mengenali ciri citra telur ayam negeri. Karena banyaknya jumlah citra latih tiap *grade* adalah 10, maka nilai K yang diuji adalah dari 1 sampai 9.

Tabel 1 menunjukkan ilustrasi klasifikasi dengan metode K-NN:

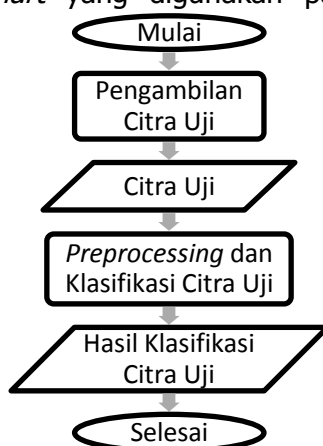
Tabel 1. Ilustrasi Klasifikasi Metode K-NN

Citra latih (<i>Grade</i>)	Jarak dengan citra uji	Penentuan jarak terdekat			Pengambilan keputusan		
		K = 1	K = 3	K = 5	K = 1	K = 3	K = 5
C01 (AA)	0,2				Masuk <i>grade</i> B	B jarak terdekat, sehingga masuk <i>grade</i> B	B jarak terdekat, sehingga masuk <i>grade</i> B
C02 (AA)	0,3						
C03 (AA)	0,6						
C04 (A)	0,5						
C05 (A)	0,1						
C06 (A)	0,35						
C07 (B)	0,25						
C08 (B)	0,4						
C09 (B)	0,01						

Pada Tabel 1, diilustrasikan bahwa sebuah citra uji dibandingkan dengan 9 citra latih, kemudian dihitung jarak antara nilai piksel citra uji dengan nilai piksel masing-masing citra latih. Ketika menggunakan parameter $K = 1$, maka jarak terdekat yang dipilih adalah 1 buah, yaitu dengan citra latih C09 yang masuk dalam *grade* B sehingga citra uji diklasifikasikan kedalam *grade* B. Ketika menggunakan parameter $K = 3$, maka jarak terdekat yang dipilih adalah 3 buah, yaitu dengan citra latih C01 (*grade* AA), C05 (*grade* A), C09 (*grade* B). Karena peluang jumlah jarak terdekat dengan tiap *grade* sama yaitu 1, maka pengambilan keputusan dilihat dari nilai jarak yang paling kecil, sehingga citra uji diklasifikasikan kedalam *grade* B.

2.3 Pembuatan Perangkat Lunak berbasis Android

Gambar 8 menunjukkan *flowchart* yang digunakan pada pembuatan perangkat lunak berbasis android:



Gambar 8. Blok Diagram Tahap *Preprocessing*

1. Pengambilan Citra Uji

Telur yang akan diuji, dipecahkan terlebih dahulu dan diletakkan ditengah-tengah penampang didalam perangkat pengambilan citra, kemudian difoto menggunakan kamera *smartphone* yang sudah terintegrasi pada perangkat pengambilan citra.

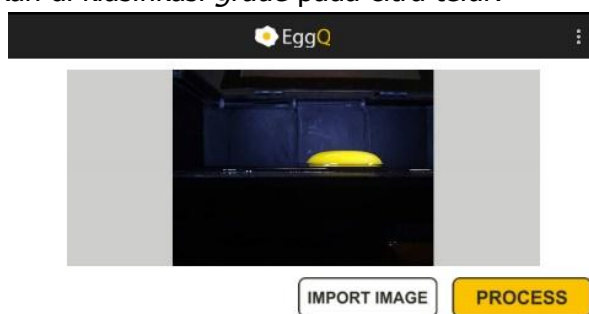
2. *Preprocessing* dan Klasifikasi Citra Uji

Selanjutnya, citra akan di proses melewati tahap *preprocessing* yaitu *cropping* dan *grayscale*, kemudian akan diproses lebih lanjut kedalam tahap klasifikasi menggunakan metode KNN, dimana akan dilakukan pengelompokkan kedalam kategori AA, A, atau B, dengan membandingkan jarak antara nilai *pixel* hasil *preprocessing* citra uji dengan *database* nilai *pixel* hasil pelatihan. Hasil pengelompokkan akan ditampilkan dilayar *smartphone* yang sudah terintegrasi pada perangkat pengambilan citra.

3. HASIL DAN DISKUSI

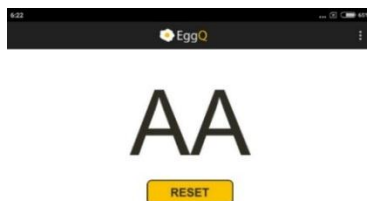
Pada proses pengujian ini, digunakan beberapa parameter uji, yaitu parameter *layer 1= Red, layer 2= Green, layer 3= Blue, layer 4= Grayscale*, dan metode K-NN dengan jenis *distance* yang digunakan yaitu *Euclidean, cityblock, cosine, dan correlation*. Pengujian ini dilakukan menggunakan *software* Matlab, karena pada aplikasi *android*, tidak ada *tools* untuk membantu mengukur beberapa parameter seperti waktu komputasi, beberapa fungsi untuk memudahkan proses pengujian. Metode penghitungan jarak yang digunakan dalam perangkat lunak berbasis android hanya *cosine distance*. Metode penghitungan jarak di K-NN pada *library* OpenCV adalah *cosine distance*, dan tidak bisa diubah dengan jenis *distance* lainnya, sehingga jarak yang digunakan hanya bisa jenis *cosine distance* (Puttemans, 2015). Oleh karena itu, pengujian yang dapat dilakukan pada aplikasi *android* adalah mengukur parameter akurasi dengan cara menguji tiap citra dan membandingkan hasil klasifikasi di android dengan hasil klasifikasi di Matlab dan hasil perhitungan menggunakan *Micrometer Haugh Unit*. Data yang digunakan untuk data latih adalah sebanyak 30 buah citra dan data yang diuji adalah sebanyak 55 buah citra.

Pada Gambar 9 disajikan tampilan aplikasi android yang akan digunakan untuk mengklasifikasi *grade* pada telur ayam negeri. Tombol *import image* digunakan untuk mengambil citra yang akan diuji. Pengambilan citra bisa langsung diambil dari kamera atau mengambil dari galeri. Setelah pengambilan citra, tombol *process* digunakan untuk memproses citra yang akan di klasifikasi *grade* pada citra telur.



Gambar 9. Tampilan Aplikasi Setelah Pengambilan Citra

Gambar 10 menunjukkan tampilan aplikasi android setelah di klasifikasi menggunakan metode K-NN, Hasil dari pengklasifikasian *grade* telur ayam negeri diatas menghasilkan *grade* AA yang memiliki arti sangat baik.



Gambar 10. Tampilan Aplikasi Setelah di Klasifikasi

Tabel 2 dibawah ini menunjukkan data hasil pelatihan:

Tabel 2. Data Pelatihan

No	Berat (gr)	Tinggi Putih	HU	Grade	No	Berat (gr)	Tinggi Putih	HU	Grade
1	55	5,5	74,68	AA	16	62,3	5,17	69	A
2	65,5	5,7	72,28	AA	17	58,75	4,76	66,81	A
3	59	5,4	72,3	AA	18	60,4	4,98	68,1	A
4	58	5,46	73,18	AA	19	57,4	4,4	63,97	A
5	57,8	5,7	75,15	AA	20	59	4,35	62,68	A
6	64,8	5,68	72,37	AA	21	57,8	2,5	38,79	B
7	58	5,45	73,1	AA	22	59	2,7	41,24	B
8	55,1	5,68	76,02	AA	23	58,9	3,72	55,74	B
9	56	5,78	76,43	AA	24	64,8	3,25	45,69	B
10	60	5,55	73,13	AA	25	58	3,1	48,19	B
11	60,9	5	68,07	A	26	63,45	3,33	47,79	B
12	66,3	4,47	60,38	A	27	59,6	3,87	57,12	B
13	63,6	4,5	61,99	A	28	58,6	3,25	49,89	B
14	59	4,5	64,19	A	29	60,1	3,85	56,61	B
15	62,35	4,78	65,38	A	30	57,5	2,9	45,6	B

Data nilai HU pada Tabel 2 didapatkan dengan menggunakan Persamaan 2, dan pengklasifikasian *grade* mengacu pada **(Tim Praktikum Produksi Ternak Unggas, 2013)**. Untuk memenuhi persyaratan Persamaan 2, maka digunakan timbangan dan *micrometer* HU untuk mendapatkan nilai HU. Hasil klasifikasi *grade* ini akan digunakan sebagai ciri dari tiap citra latih dan akan digunakan sebagai *database* untuk citra uji.

Tabel 3 menunjukkan data hasil pengujian pada Matlab dan pada aplikasi android dengan parameter layer 4, nilai K=1, dan metode penghitungan jarak *cosine*.

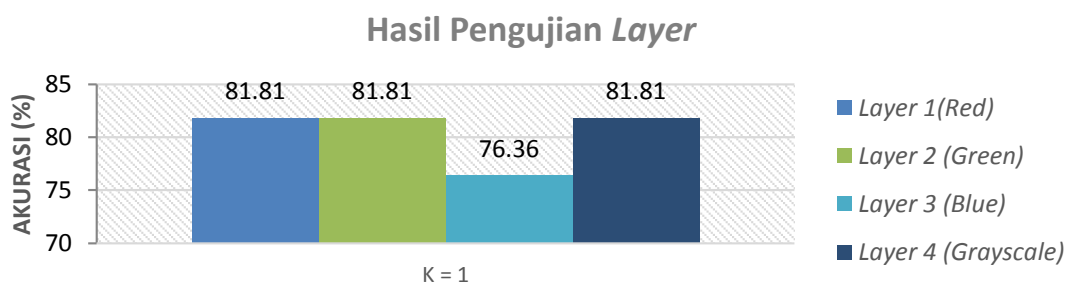
Tabel 3. Data Hasil Pengujian

No	Berat (gr)	Tinggi Putih	HU Rumus	Grade		
				Micrometer HU	Android	Matlab
1	64,4	7,02	82,29	AA	AA	AA
2	64	6,88	81,49	AA	AA	AA
3	68	10,44	99,61	AA	AA	AA
4	72,9	8,03	86,27	AA	A	A
5	66,2	9,19	94,15	AA	AA	AA
6	69	7,87	86,31	AA	AA	AA
7	64	5,49	71,09	A	A	A
8	54,1	6,59	82,72	AA	AA	AA
9	62,7	7,65	86,69	AA	AA	AA
10	55,8	4,13	62,03	A	A	A
11	71,8	3,64	46,92	B	B	B
12	55,9	2,66	43,08	B	B	B
13	58	4,6	65,64	A	A	A
14	68,9	3,74	50,11	B	AA	AA
15	73,4	6,23	73,78	AA	AA	AA
16	67,7	4,12	55,67	B	B	B
17	67,1	4,05	55,14	B	B	B
18	63,2	2,49	33,78	B	B	B
19	70,8	6,94	79,85	AA	AA	AA
20	71,7	6,31	74,97	AA	B	B
21	63,9	4,78	64,68	A	A	A
22	65,4	3,41	47,66	B	B	B
23	66,9	4,78	63,34	A	A	A
24	65,6	5,7	72,24	AA	B	B
25	67,2	3,81	52,06	B	B	B
26	59,9	3,19	48,17	B	B	B
27	70,2	3,33	43,20	B	A	A
28	62,3	3,93	56,33	B	B	B
29	65,4	3,92	54,48	B	B	B
30	63,4	2,47	33,19	B	A	A
31	66,3	4,48	60,48	A	A	A
32	67,6	5,25	67,56	A	A	A
33	67,4	5,76	72,07	AA	AA	AA
34	74,9	3,92	49,07	B	B	B
35	67,3	6,69	79,18	AA	AA	AA
36	62,1	4,4	61,68	A	A	A
37	62,7	5,69	73,24	AA	B	B
38	66,3	4,54	61,13	A	A	A
39	62,3	4,21	59,53	B	B	B
40	72,2	4,66	59,64	B	B	B
41	61,4	4,44	62,43	A	A	A
42	62,1	7	82,85	AA	A	A
43	61,7	6,19	77,42	AA	AA	AA
44	58,1	6,01	77,37	AA	AA	AA
45	61,5	5,13	68,98	A	A	A
46	64,9	4,3	59,20	B	B	B
47	58,9	5,8	75,51	AA	A	A
48	67,4	2,95	38,86	B	A	A
49	62,1	4,06	57,95	B	B	B
50	55,25	3,7	57,62	B	B	B
51	65,5	2,56	33,07	B	B	B
52	68,8	3,2	42,08	B	A	A
53	62,1	2,38	32,59	B	B	B
54	54,45	4,5	66,36	A	A	A
55	56,75	2,95	46,89	B	B	B

Tabel 3 menunjukkan bahwa klasifikasi menggunakan metode K-NN pada aplikasi *android* dan Matlab menghasilkan nilai akurasi yang sama yaitu sebesar 80%, dengan menggunakan hasil klasifikasi berdasarkan *micrometer* HU dan Persamaan 2 sebagai standar acuan dalam menentukan tingkat akurasi sistem. Kesalahan pengklasifikasian yang terjadi pada data Tabel 3 disebabkan oleh faktor jumlah citra latih, dimana hal ini dapat menyebabkan kekeliruan sistem dalam mendeteksi jarak terpendek antara nilai piksel citra uji dengan *database* nilai piksel citra latih. Semakin banyak jumlah citra latih, kemungkinan akurasi klasifikasi akan semakin tinggi. Data nilai HU Rumus pada Tabel 3 didapatkan dengan menggunakan Persamaan 2, dan pengklasifikasian *grade* mengacu pada **(Tim Praktikum Produksi Ternak Unggas, 2013)**.

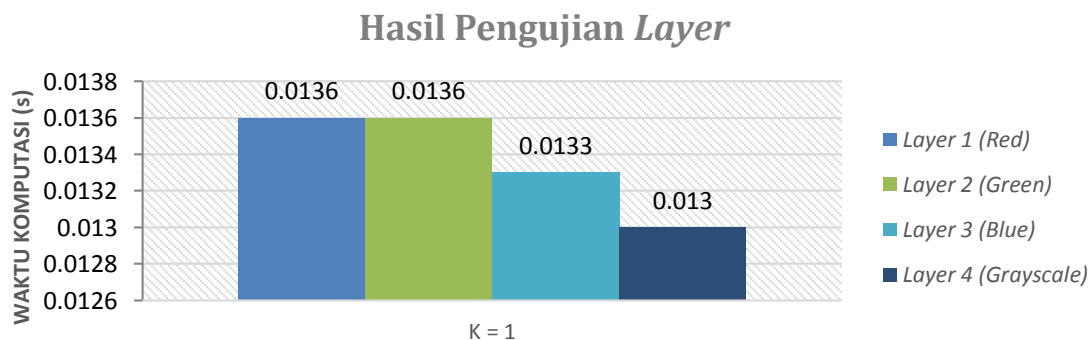
3.1 Pengujian dan analisis pengaruh *layer*

Berikut ini adalah data hasil pengujian parameter *layer*, dimana *layer* yang digunakan pada pengujian ini adalah 1=*Red*, 2=*Green*, 3=*Blue*, 4=*Grayscale* **(Putra, 2010)**. Jenis *distance* yang digunakan adalah *Euclidean* dengan $K=1$. Dari hasil pengujian terhadap 55 buah citra uji, tingkat akurasi tertinggi terdapat pada *layer* 4 dan $K=1$ sebesar 81.81% dan waktu komputasi sebesar 0.0130 s dengan jumlah data yang benar sebanyak 45 buah. Akurasi yang didapatkan pada penelitian ini lebih kecil dibandingkan penelitian **(Handayani, 2017)** yang menghasilkan tingkat akurasi 90,19%. Hal ini disebabkan pada penelitian **(Handayani, 2017)** digunakan kamera *mirrorless* dengan spesifikasi kamera FUJIFILM X-E1 dengan resolusi 16,3 MP, yang menghasilkan citra latih dan citra uji dengan nilai piksel yang lebih banyak. Hal ini akan mempengaruhi tingkat keakuratan sistem dalam pengklasifikasian.



Gambar 11. Grafik Hasil Pengujian Akurasi Pada Layer

Pada Gambar 11 dapat dilihat bahwa pada *layer* 1, 2, dan 4 akurasi yang didapatkan memiliki nilai akurasi yang sama yaitu 81.81% sedangkan pada *layer* 3, didapatkan nilai akurasi yang lebih rendah dibandingkan dengan *layer* yang lainnya. Hal ini disebabkan karena nilai warna kuning pada telur lebih mendekati warna merah (*layer* 1) dan warna hijau (*layer* 2). Pada *layer* 4, proses *grayscale* menyebabkan nilai piksel terdistribusi merata sehingga warna kuning pada telur dapat di deteksi dengan baik.

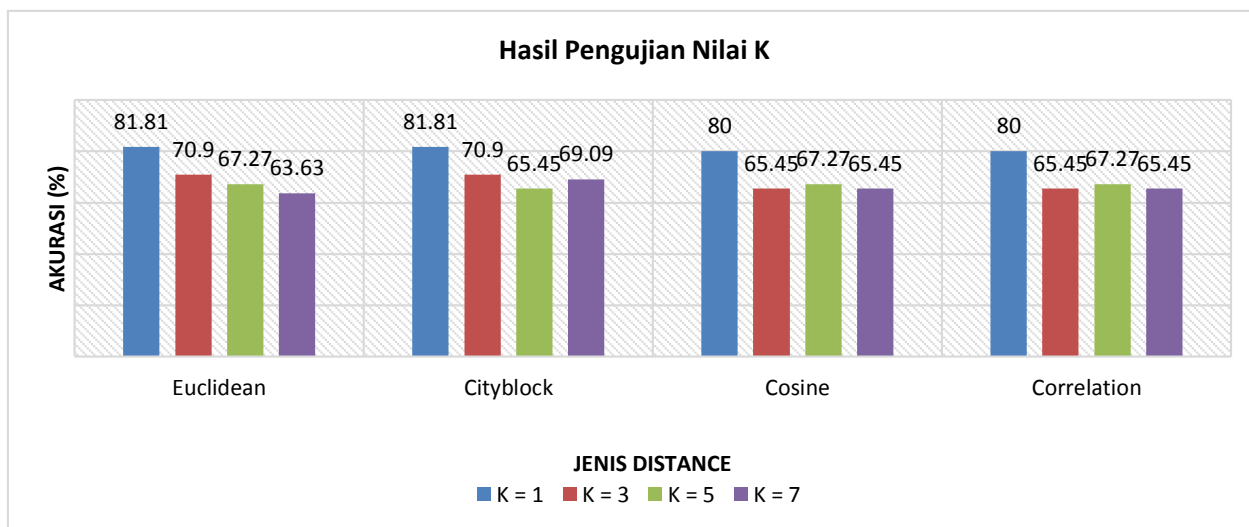


Gambar 12. Grafik Hasil Pengujian Waktu Komputasi Pada Layer

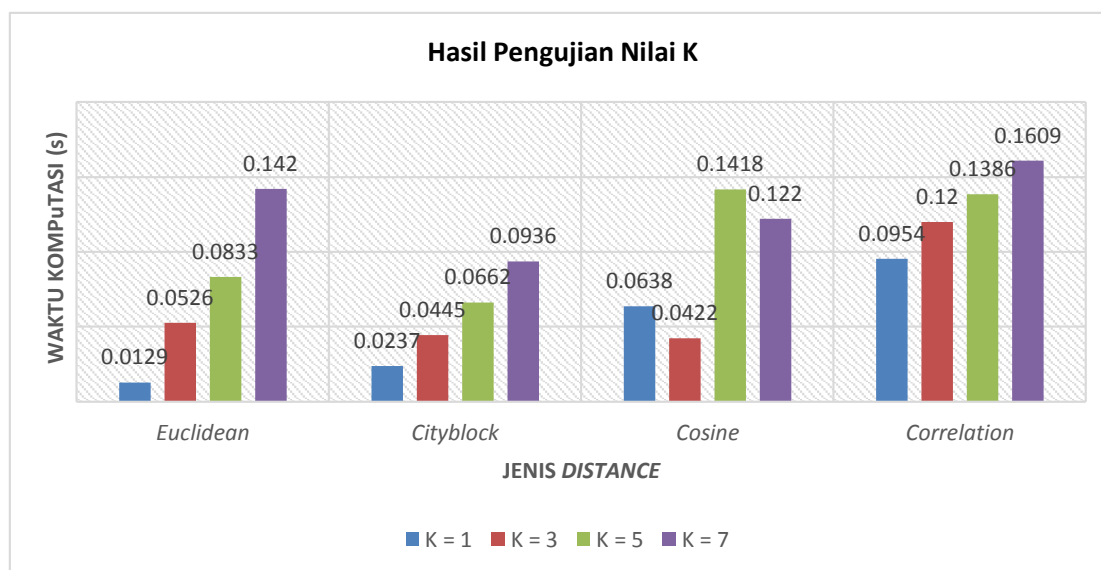
Berdasarkan grafik pada Gambar 12, dapat dilihat bahwa waktu komputasi tercepat yang didapatkan adalah pada *layer 4*, karena pada proses *grayscale* hanya terdapat 1 *layer* saja sehingga mempercepat proses pengklasifikasian pada sistem.

3.2 Pengujian pengaruh nilai K pada klasifikasi K-NN

Berikut ini adalah data hasil pengujian parameter K pada klasifikasi K-NN. Nilai K yang digunakan pada pengujian ini adalah 1, 3, 5, dan 7. Jenis *distance* yang digunakan adalah *Euclidean*, *cityblock*, *cosine*, dan *correlation* dengan *layer 4* (*Grayscale*). Gambar 13 menunjukkan hasil akurasi yang paling baik didapatkan pada saat parameter nilai K=1 dan jenis jarak *Euclidean* yaitu sebesar 81.81%, dengan jumlah data benar sebanyak 45 dari 55 data uji. Gambar 14 menunjukkan waktu komputasi tercepat didapat saat parameter K=1 yaitu 0.0129 s.



Gambar 13. Grafik Hasil Pengujian Akurasi Pada Nilai K



Gambar 14. Grafik Hasil Pengujian Waktu Komputasi Pada Nilai K

Pada pengujian ini, nilai akurasi terbaik didapatkan saat K=1 pada *Euclidean distance* dan *cityblock distance*. Hal ini disebabkan karena pengambilan keputusan (pengklasifikasian)

berdasarkan kepada satu tetangga terdekat sudah cukup mewakili karakteristik dari tiap klasifikasi yang didapat.

4. KESIMPULAN

Hasil dari penelitian ini menunjukkan bahwa sistem pengklasifikasian *grade* telur ayam negeri menggunakan klasifikasi *k-nearest neighbor* telah berhasil diimplementasikan pada perangkat keras berbasis android, dengan parameter *layer 4 (grayscale)*, nilai $K=1$, dan metode penghitungan jarak *cosine*, yang menghasilkan tingkat akurasi rata-rata terbaik sebesar 80% dengan pengukuran menggunakan *micrometer* HU sebagai pembanding. Walaupun hasil terbaik bukanlah dengan metode penghitungan jarak *cosine*, tetapi hasil pengujian menunjukkan perbedaan akurasi dan waktu yang tidak signifikan (selisih akurasi sebesar 1,81% dan selisih waktu sebesar 0,0509 s) dengan metode penghitungan jarak *Euclidean* yang menghasilkan akurasi sebesar 81,81% pada Matlab.

DAFTAR RUJUKAN

- Basuki, A. F. (2016). Deteksi Kualitas dan Kesegaran Telur Berdasarkan Segmentasi Warna dengan Metode Fuzzy Color Histogram dan Wavelet dengan Klasifikasi K-NN. *e-Proceeding of Engineering*, (pp. 4404-4411).
- Tim Praktikum Produksi Ternak Unggas. (2013). *Penuntun Praktikum Produksi Ternak Unggas*. Sumedang: Laboratorium Ternak Unggas Fakultas Peternakan Univeristas Padjadjaran.
- Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: Andi.
- Handayani, N. Y. (2017). Deteksi Kualitas dan Kesegaran Telur Ayam Berbasis Deteksi Objek Transparan dengan Metode Discrete Cosine Transform (DCT) dengan Klasifikasi K-Nearest Neighbor (K-NN). *e-Proceeding of Engineering*, (pp. 1725-1732).
- Kadir, A., & Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi.
- Meirinda, G. (2016). Deteksi Kualitas dan Kesegaran Telur Berdasarkan Segmentasi Warna Menggunakan Metode Fuzzy Color Histogram (FCH) dan Histogram Equalization dengan Klasifikasi K-Nearest Neighbor (K-NN) Pada Citra Digital. *e-Proceeding of Engineering*, (pp. 1603-1610).
- Padraid, C., & Delany, S. J. (2007). K-Nearest Neighbor Classifier. *Technical Report UCD-CSI, 4*, pp. 1-2.
- Puttemans, S. (2015). *What is the distance used by K-nearest neighbour in opencv*. Dipetik pada 18 januari 2018 dari <http://answers.opencv.org/question/66289/what-is-the-distance-used-by-k-nearest-neighbour-in-opencv/>.

- Salsabila, R. K., Hidayat, B., & Darana, S. (2016). Deteksi Kesegaran dan Kualitas Telur Berdasarkan Deteksi Objek Transparan dengan Metode Grey Level Co-Occurrence Matrix (GLCM) dan Klasifikasi K-Nearest Neighbor (KNN). *Teknika*, 1(2), 103-109.
- Wardani, Y. E. (2017). Deteksi Kualitas dan Kesegaran Telur Berdasarkan Segmentasi Warna dengan Metode Fuzzy Color Histogram (FCH) dan Discrete Cosine Transform dengan Klasifikasi K-Nearest Neighbor (K-NN). *e-Proceeding of Engineering*, (pp. 316-324).
- Widhiasih, R. N., Wahani, N. A., & Supriyanto. (2013). Klasifikasi Buah Belimbing Berdasarkan Red-Green-Blue Menggunakan K-NN dan LDA. *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, 1(1), 29-35.