

Klasifikasi *Grade* Telur Ayam Negeri secara *non-Invasive* menggunakan *Convolutional Neural Network*

NUR IBRAHIM¹, SOFIA SA'IDAH², BAMBANG HIDAYAT³, SJAFRIL DARANA⁴

^{1,2,3}Universitas Telkom, Indonesia

⁴Universitas Padjajaran, Indonesia

Email: nuribrahim@telkomuniversity.ac.id

Received 9 Agustus 2021 | *Revised* 14 September 2021 | *Accepted* 15 Oktober 2021

ABSTRAK

Telur ayam negeri merupakan salah satu sumber protein yang banyak dikonsumsi masyarakat Indonesia. Untuk menjaga kualitas telur ayam negeri yang beredar di Indonesia, diperlukan sistem yang mampu mengidentifikasi grade telur ayam dan mudah digunakan oleh masyarakat. Penelitian sebelumnya telah mengembangkan sistem pengklasifikasian grade telur ayam negeri secara invasive dengan tingkat akurasi 80%, namun sistem ini membutuhkan sampel telur yang dipecahkan sehingga setiap sampel telur tersebut tidak dapat disimpan dalam waktu lama. Oleh karena itu, penelitian ini mengembangkan sistem klasifikasi grade telur ayam tanpa perlu memecahkan sampel telur ayam (non-invasive). Dengan menggunakan metode Convolutional Neural Network (CNN), sistem mampu mengidentifikasi grade telur ayam negeri pada tingkat akurasi 85,86% dengan arsitektur LeNet-5, optimizer Adam, learning rate 0,001, dan epoch 50.

Kata kunci: telur ayam negeri, non-invasive, convolutional neural network, LeNet-5

ABSTRACT

Local Chicken egg are one of the sources of protein that is widely consumed by the people of Indonesia. To maintain the quality of local chicken egg in the market, a system that can identified chicken egg's grade and easy to use is needed. Previous research has developed an invasive chicken egg's grade classification system with 80% accuracy. However, the system required egg sample to be cracked so the egg sample can't be stored for too long. This research develop a non-invasive chicken egg's grade classification system, which doesn't require egg sample to be cracked. By using Convolutional Neural Network (CNN), system can identified chicken egg's grade at 85,86% accuracy with LeNet-5 architecture, Adam optimizer, learning rate 0,001, and epoch 50.

Keywords: local chicken egg, non-invasive, convolutional neural network, LeNet-5

1. PENDAHULUAN

Kualitas telur dapat ditinjau melalui dua bagian telur, yaitu bagian dalam telur yang terdiri dari kantong udara, putih telur dan kuning telur. Sedangkan kualitas telur bagian luar meliputi tekstur, warna kulit, bentuk, berat dan kebersihan telur **(Sholihin & Rohman, 2018)**. Kualitas telur dapat digolongkan kedalam empat jenis kualitas, diantaranya adalah kualitas AA (sangat baik), A (Baik), B (Cukup Baik), dan C (Buruk), dimana klasifikasi ini dikategorikan berdasarkan ketebalan putih telur **(Sholihin & Rohman, 2018) (Ibrahim, dkk, 2018)**. Proses klasifikasi telur ayam ini perlu dilakukan, mengingat bahwa tingginya tingkat konsumsi telur ayam di kalangan masyarakat Indonesia, sehingga penting untuk dilakukan pemantauan dan penjaminan mutu kualitas telur ayam yang beredar di pasar. Seiring berkembangnya teknologi *computer vision* dan perangkat kamera, maka proses identifikasi kualitas telur secara *non-invasive* memiliki peluang untuk menghasilkan tingkat akurasi yang sama baiknya dengan proses identifikasi kualitas telur secara *invasive*.

Penelitian-penelitian sebelumnya terkait klasifikasi kualitas telur ayam adalah dengan menggunakan metode ekstraksi ciri seperti *Fuzzy Color Histogram (Wardani, 2017)*, *Wavelet (Basuki, 2016)*, *Histogram Equalization (Meirinda, 2016)*, dimana penelitian-penelitian tersebut menggunakan K-Nearest Neighbor (K-NN) sebagai metode klasifikasi. Masing-masing dari penelitian tersebut menghasilkan tingkat akurasi sebesar 71,87% **(Wardani, 2017)**, 76% dan 81,81%. Pada penelitian **(Ibrahim, dkk, 2018)**, proses klasifikasi kualitas telur ayam dilakukan secara *invasive* dengan menggunakan metode KNN dan diimplementasikan pada perangkat android dengan tingkat akurasi 80%, dan pada penelitian **(Distya, dkk, 2019)** dilakukan implementasi pengklasifikasian telur ayam berdasarkan berat telur dan rusaknya telur secara otomatis menggunakan microcontroller dengan sensor berat dan sensor cahaya, dimana sistem ini mencapai tingkat akurasi 87,93%. Beberapa kekurangan dari penelitian-penelitian sebelumnya adalah proses pengukuran dilakukan secara *invasive*, yaitu dengan memecahkan telur dan mengambil citra dari isi telur tersebut untuk kemudian diproses ke dalam. Kemudian, penelitian **(Distya, dkk, 2019)** dilakukan secara *non-invasive*, namun belum mampu mengklasifikasikan secara mendetil ke dalam empat tingkatan kualitas yaitu AA, A, B, dan C. Oleh karena itu, pada penelitian ini dibuat sistem pengklasifikasian kualitas telur ayam negeri secara *non-invasive* berbasis menggunakan metode *Convolutional Neural Network (CNN)* yang diharapkan dapat mengkategorikan telur ayam negeri ke dalam kelas AA, A, B, dan C tanpa harus memecahkan telur dengan tingkat akurasi yang baik.

1.1 Pengolahan Citra *Digital*

Pengolahan citra digital didefinisikan sebagai serangkaian tindakan yang dilakukan terhadap suatu citra digital untuk mendapatkan informasi-informasi tertentu yang terkandung pada citra tersebut. Citra *digital* didapatkan dengan sebuah mesin yang umumnya dikenal dengan istilah kamera, dimana kamera ini menangkap cahaya yang dipantulkan dari objek yang diamati dan kemudian direpresentasikan dalam bentuk piksel-piksel.

Setiap piksel memiliki ukuran dan nilai tingkat kecerahan. Kecerahan dari sebuah piksel direpresentasikan dalam citra keabuan (*grayscale*) **(Hermawati, 2013)**. Tiga jenis representasi dari citra digital diantaranya yaitu RGB (*Red Green Blue*), *grayscale*, dan citra biner **(Kadir & Susanto, 2013)**.

Citra RGB adalah citra *digital* dimana setiap pikselnya terdiri dari 3 warna dasar, yaitu *Red*, *Green* dan *Blue*. Setiap warna dasar memiliki interval nilai 0 – 255. Dengan skala 256 ini, maka setiap kanal warna direpresentasikan sebagai 8 digit bilangan biner. Sehingga total kombinasi warna yang dihasilkan yaitu 16.777.216 kombinasi warna. Setiap piksel akan

merepresentasikan warna yang ditentukan berdasarkan kombinasi intensitas nilai merah, hijau, dan biru (**Hermawati, 2013**) yang dapat dilihat pada Gambar 1.



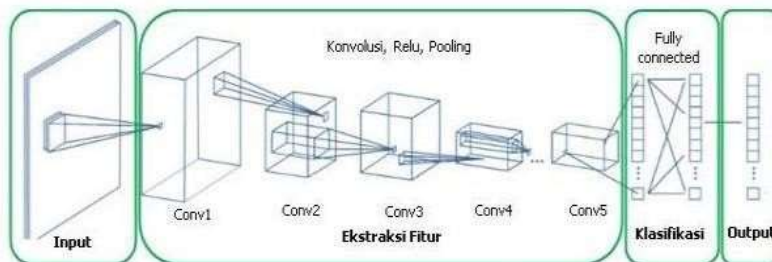
Gambar 1. Komposisi Warna RGB (Hermawati, 2013)

Citra *grayscale* hanya terdiri dari satu nilai kanal warna. Citra *grayscale* direpresentasikan ke dalam 8 bit biner. Nilai intensitas piksel pada citra keabuan ini memiliki rentang dari 0-255. Nilai 0 merepresentasikan warna hitam, 255 warna putih, sementara diantara warna hitam dan putih tersebut merupakan variasi warna abu-abu. Citra *grayscale* dapat dikonversi dari citra RGB atau citra warna dengan menggunakan Persamaan (1):

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

1.2 Convolutional Neural Network (CNN)

CNN (*Convolutional Neural Network*) dikenal sebagai model pembelajaran yang mendalam dan lebih tepat untuk bekerja dengan data gambar. Model ini bertujuan untuk mengklasifikasikan fitur dari gambar dua dimensi secara baik dengan menggunakan metode *supervised learning*, dimana cara kerja ini dengan melatih data masukan sehingga dihasilkan kelompok data yang akan menghasilkan data yang benar. CNN dikenalkan oleh Yann LeCun dengan arsitektur CNN pertama yang bernama LeNet-5 (**Yi, dkk, 2018**). CNN terdiri dari serangkaian lapisan utama yang saling terhubung, seperti pada Gambar 2. Lapisan ini terdiri dari tiga jenis, diantaranya *Convolutional Layer*, *Pooling Layer* dan *Fully-Connected Layer*.



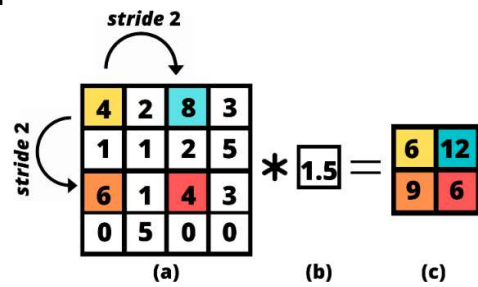
Gambar 2. Arsitektur CNN (Yi, dkk, 2018)

Metode CNN terdiri dari dua tahap, yaitu tahap pelatihan dengan metode *backpropagation* dan tahap klasifikasi citra menggunakan *feedforward*. Parameter pada tahap pelatihan yaitu jumlah *epoch* dan ukuran *batch*. Selanjutnya pada tahap klasifikasi menggunakan metode *feedforward* dengan parameter bobot dan bias.

1.2.1 Convolution Layer

Convolution layer adalah *layer* awal yang menerima sebuah *input* berupa gambar langsung dari arsitektur. *Convolution layer* digunakan sebagai proses ekstraksi fitur input, menerapkan filter untuk menghitung fitur baru terhadap citra. *Convolution layer* dapat dioptimalkan

menggunakan tiga parameter, yaitu *depth*, ukuran *stride*, dan opsi penggunaan *zero padding* (O'Shea & Nash, 2015).



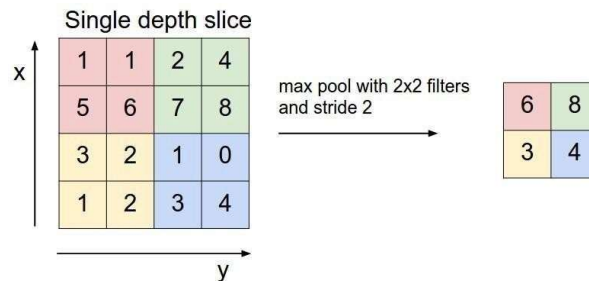
Gambar 3. Contoh Operasi *Convolutional Layer*

Gambar 3 menunjukkan contoh operasi pada *convolutional layer* dengan menggunakan stride 2, yaitu konvolusi akan bergeser sejumlah dua piksel secara horizontal maupun vertikal. Sehingga hasil konvolusi dapat dilihat pada Gambar 3 (c), yaitu perkalian antara nilai piksel citra dengan nilai filter yang kemudian akan menghasilkan nilai fitur baru.

1.2.2 Pooling Layer

Pooling layer merupakan sebuah lapisan dimana fungsi *feature map* digunakan sebagai nilai masukan. *Layer* ini biasanya disisipkan setelah *convolution layer* secara berurutan. *Pooling* berfungsi untuk mereduksi ukuran dimensi *feature map*, serta dapat mempercepat proses komputasi dan dapat mengontrol *overfitting* (Santoso & Ariyanto, 2018).

Pada umumnya, *pooling layer* mempunyai bentuk filter dengan ukuran 2x2, kemudian akan bergeser sebanyak dua kali pada lapisan setiap irisan *input*. Hasil keluaran dari yang sudah dikonvolusi biasanya menggunakan dua algoritma yaitu *Max Pooling* dan *Average Pooling*. Gambar 4 menunjukkan contoh penggunaan *Max Pooling* dengan ukuran filter 2x2 dan *stride* 2, dimana dalam setiap pergeseran filter akan diambil nilai terbesar. Sedangkan untuk *Average Pooling*, digunakan nilai rata-rata.



Gambar 4. Contoh Operasi *Max Pooling*

1.2.3 Fully Connected Layer

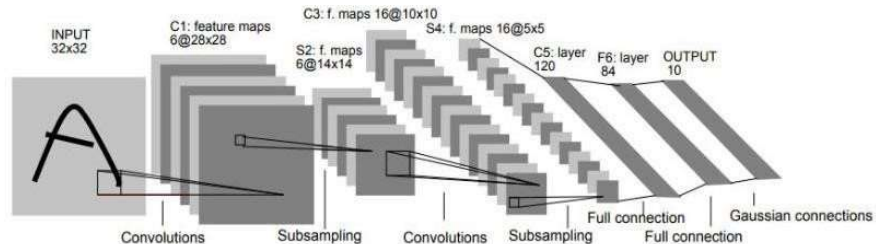
Fully Connected Layer merupakan lapisan yang digunakan sebagai pengklasifikasian berdasarkan fitur yang sudah diperoleh dari proses sebelumnya. *Layer* ini berfungsi sebagai transformator agar data dapat diaplikasikan secara linear. Sebelum masuk ke tahap proses *fully connected layer*, pada *feature map* akan terjadi proses *flatten*. Selanjutnya akan dihasilkan vektor yang menjadi *input* pada *fully connected layer*. *Fully connected layer* ini terdiri dari *hidden layer* dan *output layer* (Nour & Harintaka, 2018).

1.3 Arsitektur LeNet-5

Salah satu arsitektur CNN yang diusulkan untuk pengenalan objek adalah Arsitektur LeNet (Atabay, 2016), yang sekarang lebih mengacu ke LeNet-5. LeNet-5 merupakan

pengembangan dari arsitektur LeNet sebelumnya. Arsitektur LeNet-5 diperkenalkan oleh Yann LeCun yang terdiri dari tujuh lapisan (**Fitriati, 2016**), dimana arsitektur ini merupakan salah satu arsitektur CNN pertama yang digunakan di bank untuk membaca cek dengan *real time*. Meskipun terdapat algoritma lain, seperti *Support Vector Machine* (SVM), tetapi perhitungan dengan arsitektur LeNet-5 lebih unggul karena dapat bekerja secara eksponensial dan lebih cepat daripada algoritma lainnya (**Peryanto, dkk, 2020**).

Arsitektur model LeNet-5 menerima *input* dengan ukuran 32×32, selanjutnya *output* pada *layer* pertama akan dijadikan input pada *layer* berikutnya sehingga pada layer terakhir akan menghasilkan *output* yang digunakan sebagai tahap klasifikasi kualitas telur ayam negeri. Selanjutnya dilakukan pengujian terhadap parameter perfomansi, diantaranya akurasi, loss dan presisi. Gambar 5 menunjukkan arsitektur model LeNet-5.



Gambar 5. Arsitektur LeNet-5

1.4 Kualitas Telur

Pendeteksian kualitas telur yang selama ini dilakukan hanya mengandalkan indera penglihatan manusia. Kualitas telur dideteksi berdasarkan *Haugh Unit* berdasarkan standar *United State Departement of Agriculture* (USDA). Berikut adalah kriteria untuk setiap *grade* telur ditinjau dari ketinggian HU putih telur: (**Sholihin & Rohman, 2018**)

1. Kualitas AA > 72.
2. Kualitas A 60-71.
3. Kualitas B 31-59.
4. Kualitas C < 31.

Haugh Unit adalah satuan untuk mengetahui kesegaran isi telur, ditinjau dari bagian putih telur. Pengukuran dapat dilakukan dengan cara memecah telur terlebih dahulu untuk kemudian diukur ketebalan putih telur menggunakan alat *micrometer*. Semakin tebal putih telur, maka telur tersebut semakin baik kualitasnya. Sehingga dapat disimpulkan bahwa nilai *Haugh Unit* yang semakin tinggi merepresentasikan kualitas telur yang semakin baik. Berikut adalah Persamaan (2) untuk menentukan tinggi HU: (**Sholihin & Rohman, 2018**)

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

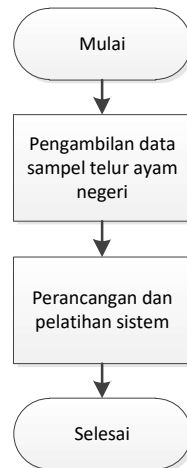
Keterangan:

HU : Nilai *Haugh Unit*
 H : Tinggi putih telur
 W : Berat Telur

Selain berdasarkan *Haugh Unit*, kualitas telur dapat diamati dari tekstur cangkangnya (**Sela & Ihsan, 2017**). Oleh karena itu, pada penelitian ini akan dilakukan pendeteksian kualitas telur berdasarkan citra telur yang belum dipecahkan (*non-invasive*) dan dilakukan validasi berdasarkan nilai *Haugh Unit* setiap sampel telur yang digunakan.

2. METODE PENELITIAN

Penelitian ini bertujuan untuk membantu masyarakat dalam mengetahui kualitas telur ayam dengan mudah, dan akurat. Dengan demikian masyarakat pun dapat mengetahui apakah telur ayam negeri tersebut termasuk kedalam kualitas *grade* AA (sangat baik), A (baik), B (cukup baik), atau C (buruk). Adapun metodologi penelitian yang digunakan dapat dilihat pada Gambar 6:



Gambar 6. Flowchart Metodologi Penelitian

Penjelasan mengenai Gambar 6 adalah:

1. Pengambilan Data Sampel Telur Ayam Negeri
Sebanyak 984 citra telur ayam negeri (328 telur *grade* AA, 328 telur *grade* A, dan 328 telur *grade* B) diambil citranya menggunakan kamera *smartphone* dengan resolusi 12 MegaPixel dengan jarak 10 cm. *Grade* C tidak dimasukkan kedalam penelitian ini, dengan alasan bahwa telur sudah tidak layak konsumsi. Pengambilan citra telur dilakukan pada *background* berwarna putih dengan pencahayaan menggunakan LED untuk mendapatkan citra dengan kondisi lingkungan yang sama. Selanjutnya, citra telur diukur ketinggian HU nya menggunakan HU *micrometer* untuk kemudian dijadikan dasar dalam penentuan *grade* telur tersebut.
2. Perancangan dan Pelatihan Sistem
Sistem dirancang dengan menggunakan Arsitektur LeNet-5 untuk kemudian dilatih menggunakan data sampel citra telur yang telah didapatkan. Sistem kemudian dilatih dengan komposisi data latih sejumlah 80% dari total data citra dan 20% sisanya digunakan sebagai data uji.

2.1 Pengambilan Data Sampel Telur Ayam Negeri

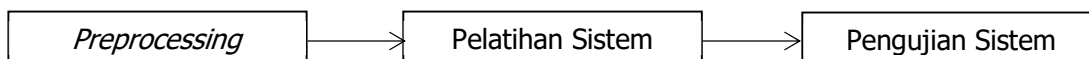
Data yang digunakan adalah berupa citra telur ayam negeri dalam kondisi tidak dipecahkan, seperti pada Gambar 7. Telur ayam yang hendak diambil citranya, diletakkan pada area dengan *background* berwarna putih yang diberikan penerangan menggunakan LED untuk mendapatkan kondisi lingkungan yang sama pada semua citra telur ayam. Setelah diambil citranya, telur ayam tersebut dipecahkan untuk kemudian diukur ketinggian putih telurnya sehingga didapatkan nilai HU. Nilai HU ini kemudian digunakan untuk mengklasifikasi telur ke dalam tingkat kualitas telur (AA, A, B, dan C) sebagai data validasi.



Gambar 7. Sampel Citra Telur Ayam Negeri

2.2 Perancangan dan Pelatihan Sistem

Desain sistem secara umum dari penelitian ini dapat dilihat pada Gambar 8 :



Gambar 8. Blok Diagram Umum Sistem

Berikut adalah penjelasan Gambar 8:

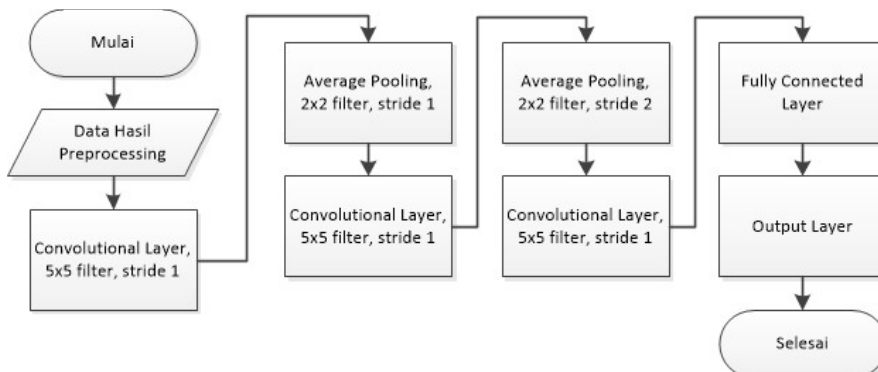
1. *Preprocessing* adalah proses pengolahan dataset citra telur yang telah didapatkan, yaitu dilakukan proses augmentasi dan *resize*.
2. Pelatihan sistem adalah proses pembelajaran sistem untuk dapat mengenali jenis kualitas telur ayam negeri dengan menggunakan dataset citra telur.
3. Klasifikasi merupakan sebuah tahapan untuk mengelompokkan objek berdasarkan kelasnya. Pada penelitian ini digunakan metode klasifikasi dengan sistem *K-Nearest Neighbor*.

2.2.1 Preprocessing

Pada tahap ini, setiap citra yang ada di dataset dilakukan proses augmentasi (*flip, rotate vertical, rotate horizontal*) dan *resize*. Proses augmentasi dilakukan secara random, dan ukuran citra hasil *resize* yang digunakan adalah 32x32, 64x64, 128x128, dan 224x224 piksel. Hal ini dilakukan untuk mengetahui ukuran citra masukan yang optimal yang dapat diolah oleh sistem.

2.2.2 Pelatihan Sistem

Gambar 9 menunjukkan langkah-langkah proses pelatihan sistem.



Gambar 9. Blok Diagram Tahap Pelatihan Sistem

Pada tahap ini, citra yang telah dilakukan *preprocessing* akan dimasukkan ke dalam *layer* pertama yaitu *convolutional layer* dengan filter berukuran 5x5 dan *stride* 1, kemudian keluaran dari layer ini akan dimasukkan ke dalam layer berikutnya yaitu *average pooling layer* dengan

filter 2x2 dan stride 1, dan akan dilanjutkan hingga layer ke tujuh yaitu *output layer*. Proses ini akan dilakukan untuk setiap citra yang dialokasikan sebagai data latih, yaitu sejumlah 80% dari total dataset citra. Setiap citra yang telah dimasukkan ke dalam proses pelatihan ini akan meng-*update* nilai bobot dan bias pada setiap layer, dalam hal ini adalah nilai-nilai filter pada setiap layer. Bobot yang telah di-*update* ini akan digunakan oleh sistem pada proses pengujian.

2.2.3 Pengujian Sistem

Pada proses pengujian sistem, citra yang digunakan sebagai data uji adalah 20% dari total dataset citra, dimana citra-citra ini bukan citra yang digunakan pada proses pelatihan. Tahapan-tahapan pada proses pengujian sistem ini sama dengan tahapan-tahapan pada proses pelatihan sistem seperti yang ditunjukkan pada Gambar 9. Yang membedakan antara pelatihan sistem dengan pengujian sistem adalah pada pengujian sistem, nilai-nilai bobotnya tidak akan berubah seperti yang terjadi pada proses pelatihan sistem. Keluaran dari pengujian sistem ini adalah hasil identifikasi sistem terhadap citra telur yang dibaca. Pada tahap ini akan dilakukan pengukuran performansi berdasarkan nilai akurasi, *loss*, dan presisi.

Akurasi didefinisikan sebagai tingkat kebenaran dalam suatu penelitian yang menunjukkan hasil yang akurat, *loss* didefinisikan sebagai hasil perhitungan yang menunjukkan ketidaksesuaian sistem dalam mengenali objek, dan presisi didefinisikan sebagai tingkat ketepatan data dengan membandingkan jumlah yang diprediksi dengan jumlah total data keseluruhan. Berikut adalah persamaan matematis dari akurasi, *loss*, dan presisi:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$\text{Loss} = \frac{P + FN}{TP + TN + FP + FN} \quad (4)$$

$$\text{Presisi} = \frac{TP + TN}{TP + FP} \quad (5)$$

Dimana:

- TP (*True Positive*) : jumlah data positif yang terdeteksi benar
- TN (*True Negative*) : jumlah data negatif yang terdeteksi benar
- FP (*False Positive*) : jumlah data positif yang terdeteksi salah
- FN (*False Negative*) : jumlah data negatif yang terdeteksi salah

3. HASIL DAN DISKUSI

Tahap pengujian dilakukan pada beberapa parameter, yaitu parameter *size* citra, *optimizer*, *learning rate* dan *epoch*. Ukuran *size citra* yang diujikan sebesar 32x32, 64x64, 128x128 dan 224x224 piksel. Setelah didapatkan *size citra* yang optimal, maka dilanjutkan dengan menguji beberapa *optimizer*, yaitu Adam, SGD dan RMSprop. Untuk nilai *learning rate* yang diujikan yaitu sebesar 0.01, 0.001 dan 0.0001. Nilai *epoch* yang diujikan sebesar 50, 100 dan 150. Selain itu, *batch size* yang digunakan sebesar 32. Dalam skenario perbandingan data *training* dan data *testing* yaitu 80:20, 80% untuk data *training* dan 20% untuk data *testing*.

3.1 Pengujian dan Analisis Pengaruh *Size* Citra Masukan

Pengujian pertama dengan mencari nilai ukuran *size* citra terbaik. Untuk menentukan *size* citra maka dilakukan dengan menggunakan ukuran *size* citra yang berbeda. *Size citra* yang diujikan adalah 32x32, 64x64, 128x128 dan 224x224 piksel. Selain itu, dibantu dengan beberapa parameter yaitu dengan *optimizer* Adam, menggunakan *learning rate* 0.0001, *epoch* 50 dan *batch size* 32. Hasil pengujian dapat dilihat pada Tabel 1.

Berdasarkan Tabel 1 dapat dilihat bahwa citra dengan *size* 32*32 menghasilkan performansi yang paling baik dibandingkan dengan *size* lainnya. Hal ini disebabkan citra dengan ukuran yang besar memiliki jumlah feature map yang lebih banyak dan lebih detail, sehingga sistem membaca tingkat kemiripan yang lebih tinggi antara kelas AA, A, dan B sehingga sistem sulit membedakan antara kelas yang satu dengan kelas yang lainnya.

Tabel 1. Hasil Pengujian Pengaruh *Size* terhadap Performansi Sistem

<i>Size</i> Citra	Akurasi	<i>Loss</i>	Presisi
32*32	85,86%	0,3779	81,82%
64*64	32,46%	1,3561	8,11%
128*128	30,89%	1,3628	7,72%
224*224	34,55%	1,3159	8,63%

3.2 Pengujian dan analisis pengaruh *optimizer*

Pengujian kedua dilanjutkan dengan menentukan *optimizer* terbaik. Dengan *size* citra terbaik yang sudah didapatkan pada skenario pertama, yaitu 32*32. Setelah model prediksi sudah dirancang maka dilanjutkan dengan menentukan *optimizer* untuk menghasilkan akurasi yang terbaik. Beberapa *optimizer* yang akan diujikan pada umumnya yaitu Adam, SGD dan RMSprop. Model parameter yang digunakan sama, yaitu *learning rate* 0.0001, *epoch* 50 dan *batch size* 32. Hasil pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Pengaruh *Optimizer* terhadap Performansi Sistem

<i>Optimizer</i>	Akurasi	<i>Loss</i>	Presisi
Adam	85,86%	0,3779	81,82%
SGD	37,70%	1,2958	29,67%
RMSprop	27,23%	1,3726	6,8%

Berdasarkan Tabel 2 dapat dilihat bahwa *optimizer* Adam mampu menghasilkan akurasi terbaik yaitu sebesar 85,86%. Hal ini disebabkan *optimizer* Adam merupakan penggabungan dari *optimizer* SGD dan RMSprop yang memadukan keunggulan dari *optimizer* SGD dan RMSprop. Meskipun presisinya sedikit lebih rendah dibandingkan dengan *optimizer* RMSprop, secara keseluruhan *optimizer* Adam masih menghasilkan performansi yang lebih unggul dari aspek akurasi dan *loss*. Selain itu, Tabel 2 juga menunjukkan bahwa karakteristik citra telur ayam negeri yang digunakan pada penelitian ini lebih tepat jika disandingkan dengan penggunaan *optimizer* Adam.

3.3 Pengujian dan Analisis Pengaruh *Learning Rate*

Pengujian ketiga dilakukan untuk memperoleh *learning rate* terbaik. Pengujian ini menggunakan parameter yang sudah didapatkan sebelumnya, yaitu nilai *size* citra 32*32 dan *optimizer* Adam. Nilai *learning rate* yang diuji adalah 0.01, 0.001, dan 0.0001. Model parameter yang digunakan sama, yaitu *epoch* 50 dan *batch size* 32. Hasil pengujian dapat dilihat pada Tabel 3.

Berdasarkan Tabel 3, dapat disimpulkan bahwa *learning rate* 0.001 menunjukkan hasil yang terbaik dibandingkan dengan nilai *learning rate* lainnya. Semakin kecil nilai *learning rate*, maka semakin detail sistem dalam mengevaluasi hasil pembelajaran dari proses pelatihan. Namun tidak selalu performansi semakin baik seiring dengan semakin kecilnya nilai *learning rate*. Hal ini dikenal dengan istilah *overfitting*, dimana sistem mempelajari fitur terlalu detail sehingga sistem terlalu tegas dalam mengidentifikasi data uji, yang mengakibatkan performa sistem menjadi turun.

Tabel 3. Hasil Pengujian Pengaruh *Learning Rate* terhadap Performansi Sistem

<i>Learning rate</i>	Akurasi	Loss	Presisi
0,01	34,55%	1,3716	8,63%
0,001	85,86%	0,3779	81,82%
0,0001	51,31%	1,0752	53,75%

3.4 Pengujian dan Analisis Pengaruh *Epoch*

Pengujian keempat dilakukan untuk mencari nilai *epoch* terbaik. Kemudian beberapa nilai *epoch* akan diujikan untuk melihat pengaruh *epoch* terhadap sistem klasifikasi. Beberapa nilai *epoch* yang diujikan adalah 50, 100 dan 150. Pada skenario keempat parameter yang digunakan merupakan hasil terbaik dari pengujian pertama, yaitu size citra 32*32 kemudian pengujian kedua diperoleh optimizer Adam dan pada pengujian diperoleh nilai *learning rate* 0.0001. Parameter pada pengujian ini menggunakan *batch size* 32. Hasil pengujian dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian Pengaruh *Epoch* terhadap Performansi Sistem

<i>Epoch</i>	Akurasi	Loss	Presisi
50	85,86%	0,3779	81,82%
100	83,25%	0,3627	83,39%
150	80,10%	0,4505	80,09%

Berdasarkan Tabel 4, dapat disimpulkan bahwa penggunaan *epoch* 50 mampu memberikan akurasi yang lebih baik yaitu 85,86%. Semakin besar nilai *epoch* mengindikasikan sistem lebih banyak melakukan pengulangan proses pembelajaran. Tidak selalu nilai *epoch* yang semakin besar akan menghasilkan performansi yang lebih baik, karena setiap citra memiliki karakteristik yang berbeda sehingga ada kemungkinan terjadinya *overfitting* akibat sistem terlalu sering mempelajari fitur dan mencapai titik jenuh. Hal ini justru akan membuat performa sistem menjadi menurun.

4. KESIMPULAN

Hasil pengujian menunjukkan bahwa sistem klasifikasi *grade* telur ayam negeri secara *non-invasive* menggunakan *Convolutional Neural Network* dengan arsitektur LeNet-5 mampu mengklasifikasikan telur ayam negeri ke dalam *grade* AA, A, dan B dengan performansi terbaik sebesar 85,86% tingkat akurasi, 0,3779 loss, dan presisi 81,82%. Hasil ini dicapai pada model

sistem dengan konfigurasi *optimizer Adam*, *learning rate* 0.01, dan *epoch* 50. Hal ini menunjukkan bahwa sistem klasifikasi grade telur ayam negeri secara *non-invasive* mampu menyaingi performansi sistem klasifikasi grade telur ayam negeri secara *invasive*. Sistem ini dapat dikembangkan lagi dengan mengaplikasikannya pada perangkat seperti *smartphone* atau Raspberry pi sehingga secara umum dapat bermanfaat bagi masyarakat luas.

DAFTAR RUJUKAN

- Atabay, H. A. (2016). Binary Shape Classification Using Convolutional Neural Networks. *IIOAB Journal*, 332-336.
- 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 (2355-9365)*, 4(1).
- Distya, Y. D., Sari, Z. L., & Putra, B. C. (2019). "EGG-GRADING" Mesin Klasifikasi Telur Ayam (Berat Telur dan Telur Rusak) Otomatis berbasis Microcontroller. *Prosiding Nasional Rekayasa Teknologi Industri dan Informasi XIV (ReTII)*, (pp. 380-385). Yogyakarta: Institut Teknologi Nasional Yogyakarta.
- Fitriati, D. (2016). Perbandingan Kinerja CNN LeNet 5 dan Extreme Learning Machine Pada Pengenalan Citra Tulisan Tangan Angka. *Jurnal Teknologi Terpadu*, 10-16.
- Hermawati, F. A. (2013). *Pengolahan Citra Digital: Konsep dan Teori*. Yogyakarta: Andi.
- Ibrahim, N., Bacheramsyah, T. F., Hidayat, B., & Darana, S. (2018). Pengklasifikasian Grade Telur Ayam Negeri menggunakan Klasifikasi K-Nearest Neighbor berbasis Android. *Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 6(2), 288-302.
- 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.
- Nour, E., & Harintaka. (2018). Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman pada Citra Resolusi Tinggi. *Teknik Geomatika*, 61-65.
- O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *arXiv preprint*, 1(2), p. 1511.08458v2.
- Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. *Journal Applied Informatics and Computing (JAIC)*, 4(1), 46-51.

- Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Jurnal Emitter*, 18(1), 15-18.
- Sela, E. I., & Ihsan, M. (2017). Deteksi Kualitas Telur menggunakan Analisis Tekstur. *Indonesian Journal of Computing and Cybernetics System (IJCCS)*, 11(2), 199-208.
- Sholihin, M., & Rohman, M. G. (2018). Klasifikasi Kualitas Mutu Telur Ayam Ras Berdasarkan Fitur. *Jurnal Teknik*, X, 1056-1059.
- 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 (2355-9365)*, 4(1).
- Yi, N., Li, C., & Feng, X. (2018). Research and Improvement of Convolutional Neural Network. *IEEE Computer Society*, 637-640.