

## OPTIMASI AKURASI METODE CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI KUALITAS BUAH APEL HIJAU

Rendi Setya Nugraha <sup>1</sup>, Arief Hermawan <sup>2</sup>

<sup>1,2</sup> Program Studi Informatika Fakultas Sains dan Teknologi Universitas Teknologi Yogyakarta  
*rendi.5200411158@student.uty.ac.id*

### ABSTRAK

Indonesia memiliki potensi dalam pengembangan berbagai jenis buah, termasuk buah apel hijau. Permintaan akan apel hijau semakin meningkat di pasar tradisional, namun seringkali pembeli mengeluh karena kurangnya apel hijau yang berkualitas. Saat ini, pemilihan apel hijau masih dilakukan secara manual, yang memakan banyak waktu terutama di perkebunan atau pertanian yang besar dan rentan terhadap kesalahan manusia. Salah satu solusi adalah menggunakan sistem komputer dengan metode *Convolutional Neural Network* (CNN) untuk mengklasifikasikan apel hijau baik atau busuk. Dalam penelitian ini, dilakukan optimasi metode CNN dengan menambah beberapa *hyperparameter* dan augmentasi data. Hasil penelitian menunjukkan akurasi tertinggi adalah 96,88% pada data *training* dan 98,44% pada data *validation*. Pengujian pada data *testing* menghasilkan akurasi 93,75%, dengan pembagian data pelatihan dan validasi 80%:20% dari total 400 citra apel hijau. Ini menunjukkan bahwa model ini berhasil mengklasifikasikan citra apel hijau dengan kinerja yang optimal.

**Keyword :** *Apel hijau, Citra, Klasifikasi, Convolutional Neural Network, Optimasi*

### 1. PENDAHULUAN

Indonesia merupakan salah satu negara agraris yang memiliki potensi sumber daya dalam bidang pertanian atau perkebunan, termasuk pengembangan berbagai jenis buah. Apel hijau salah satu jenis buah yang dianggap berpotensi, buah ini merupakan buah yang kaya akan manfaatnya. Apel ini memiliki manfaat yang sangat ampuh untuk orang yang sedang diet dan ingin menyehatkan mata dan tubuhnya. Apel hijau biasa juga disebut Apel malang atau apel manalagi dengan nama ilmiahnya *Malus sylvestris*, perbedaan apel hijau dan jenis apel lainnya terletak pada rasa, warna, dan bentuk di mana apel hijau memiliki rasa yang lebih kecut, kulitnya berwarna hijau, dan bentuknya lebih bulat [1].

Saat ini permintaan untuk buah apel meningkat pesat di pasar-pasar tradisional Indonesia, khususnya buah apel hijau yang akan dijadikan bahan pada penelitian ini. Buah apel ini sangat banyak diminati oleh mereka yang menginginkan diet sehat, namun terdapat kendala yang sering dihadapi yaitu kurangnya ketersediaan apel hijau yang berkualitas baik. Dari hasil penelitian yang dilakukan menunjukkan bahwa faktor lingkungan yang membuat apel mudah mengalami penurunan kualitas, faktor tersebut meliputi suhu, kelembaban, dan curah hujan [2]. Proses dalam pemilihan apel hijau yang dilakukan saat ini masih menggunakan cara manual dengan melibatkan manusia sebagai pengambil keputusan apakah apel bagus atau busuk. Tentunya cara ini memiliki beberapa kelemahan seperti membutuhkan waktu yang relatif lama terutama untuk pertanian atau perkebunan besar, dan manusia juga cenderung dapat melakukan kelalaian dalam pengklasifikasian buah apel hijau.

Salah satu cara untuk mengklasifikasikan citra buah apel hijau bagus atau busuk adalah dengan

melalui sistem komputer yang memakai metode *Convolutional Neural Network* (CNN). Atas dasar penelitian yang dilakukan oleh Shafira [3] yang mengklasifikasikan citra tomat menggunakan metode CNN dengan *Keras* dan mendapatkan hasil akurasi yang cukup bagus yaitu 90% dari data uji. CNN bekerja dengan cara menerima gambar sebagai input, kemudian input tersebut akan melalui beberapa lapisan (layer) dalam proses pelatihan, seperti softmax, untuk menghasilkan hasil yang dapat mengidentifikasi objek yang ada dalam gambar yang diinputkan [4]. CNN digunakan dalam konteks pengolahan gambar untuk melakukan tugas seperti pendeteksian objek, analisis visual gambar, dan mengenali objek-objek yang ada di dalamnya [5]. CNN telah banyak dipakai di bidang pertanian maupun perkebunan, seperti yang dijelaskan pada bagian penelitian terdahulu.

Untuk mencapai hasil yang lebih baik lagi peneliti akan melakukan optimasi pada data augmentasi, seperti yang dilakukan oleh [6] yang melakukan Optimasi Klasifikasi Gambar Varietas Jenis Tomat Dengan Data Augmentation dan *Convolutional Neural Network*. Ada juga yang melakukan optimasi dengan menambah beberapa *hyperparameter* [7]. Dengan ini harapannya metode CNN dapat mengklasifikasikan kualitas Buah Apel Hijau berdasarkan tingkat kecerahan warna buah.

### 2. TINJAUAN PUSTAKA

#### 2.1. Penelitian Terdahulu

Beberapa hasil penelitian yang dilakukan oleh peneliti sebelumnya yang memiliki bidang dan tema yang serupa dengan penelitian yang akan dilakukan. Terdapat beberapa penelitian sebelumnya yang menggunakan metode *Convolutional Neural Network* (CNN) yang ditinjau dari berbagai sudut pandang para peneliti, penelitian oleh Maulana, F.

F., dan Rochmawati, N [8] dengan judul Klasifikasi Citra Buah Menggunakan *Convolutional Neural Network*. Penelitian tersebut bermaksud mengembangkan arsitektur CNN yang mampu mengklasifikasikan citra buah-buahan, *dataset* yang diolah adalah *dataset* citra buah-buahan yang berasal dari *dataset Fruit-360*.

Pengklasifikasian pada penelitian tersebut menggunakan metode CNN yang menggunakan perpaduan 3 convolutional layer dan 2 hidden layer yang mampu mengklasifikasikan citra buah-buahan dengan akurasi yang baik. Hasil dari proses learning didapatkan model CNN dengan akurasi 100% dan loss sebesar 0,012. Pada proses pengujian model CNN yang menggunakan 45 sampel citra buah didapatkan akurasi sebesar 91,42%, diharapkan penelitian ini dapat membantu para peneliti di bidang perkebunan dan pertanian dalam mengenali buah-buahan sebagai media pembelajaran.

Penelitian oleh Juliansyah, S., dan Laksito, A. D. [9] yang berjudul Klasifikasi Citra Buah Pir Menggunakan *Convolutional Neural Network*. Penelitian ini bertujuan untuk mengklasifikasikan citra buah pir ke dalam jenis buah pir *Williams* dan *Forelle*, menggunakan *dataset Fruit-360* yang memiliki citra buah pir dengan ukuran sebesar 100x100 pixel untuk citranya. Dengan memakai metode CNN melewati beberapa tahapannya mulai dari menginputkan citra diperoleh *learning rate* 0,001 dengan ukuran filter 3x3, jumlah *epoch* 15, data *training* 640 dan data *validation* 160 data. Diperoleh hasil akurasi *training* sebesar 100% dan *validation* 100%, data baru sebanyak 100 citra digunakan sebagai data *testing* untuk diujikan ke dalam model yang telah dibuat. *Testing* ini menghasilkan tingkat akurasi baru dalam melakukan klasifikasi gambar citra buah pir sebesar 98% antara jenis buah pir *Williams* dan *Forelle*.

Damayanti, S. A., dkk. [10] dalam penelitiannya yang berjudul Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN. Penelitian tersebut bertujuan untuk mendapatkan pendeteksi kematangan pada warna mangga badami dengan menerapkan metode CNN pada aplikasi pengolahan citra digital, sehingga dapat dipastikan buah mangga yang telah melewati masa kematangan untuk dikonsumsi. Penelitian ini mengolah data citra buah mangga badami yang diambil dari repositori Kaggle dengan jumlah 204 citra, citra sebanyak 25 digunakan untuk data uji dan 179 digunakan sebagai data latih. Pengklasifikasian menggunakan metode CNN tentunya melewati beberapa tahapan sebelum memulai pengujian seperti pengumpulan data, pra proses citra, ekstraksi feature, dan klasifikasi serta evaluasi. Hasil akurasi yang didapat sebesar 97,2% dan hasil loss sebesar 0,5% dari hasil pengujian terhadap data uji, hal ini menunjukkan bahwa penggunaan model CNN pada klasifikasi kematangan buah mangga badami cukup baik.

## 2.2. Citra Digital

Citra Merupakan imitasi benda tiga dimensi yang direpresentasikan dalam bentuk dua dimensi melalui kombinasi garis, titik, bentuk, dan warna. Citra digital merupakan data dua dimensi di mana satuan terkecilnya disebut piksel. Informasi dasar tersebut dianalisis dan dikenali komputer untuk mengenali suatu peristiwa atau situasi tertentu [11].

## 2.3. Convolutional Neural Network

Convolutional Neural Network adalah varian yang diilhami secara biologis dari multilayer perceptron (MLP) yang telah terbukti sangat efektif di berbagai bidang seperti pengenalan gambar dan klasifikasi. CNN telah berhasil digunakan untuk mengenali wajah, objek, dan rambu jalan, serta meningkatkan penglihatan robot dan mobil tanpa pengemudi. CNN mengeksplorasi korelasi spasial lokal dengan menerapkan pola konektivitas lokal antar neuron di lapisan yang berdekatan [12].

Hubel dan Wiesel melakukan penelitian pertama tentang korteks visual penglihatan kucing pada tahun 1968 di belakang penemuan CNN. Secara teknis, CNN merupakan salah satu jenis jaringan saraf yang sering digunakan pada data citra. CNN dapat digunakan untuk mendeteksi dan mengidentifikasi objek pada citra. Biasanya, CNN tidak jauh berbeda dengan jaringan saraf normal [13].

## 2.4. Keras

*Tensorflow* memiliki *package* yang bisa digunakan yaitu Keras. Keras mendukung kinerja *Tensorflow* sebagai pengoptimal untuk hasil yang cepat dan bagus. Jadi Keras adalah *wrapper Tensorflow*, membuatnya lebih bertenaga atau lebih sederhana. Meskipun membangun deep neural networks di *tensorflow* jauh lebih mudah daripada melakukannya dari awal, *tensorflow* masih merupakan API tingkat rendah. Keras adalah API tingkat tinggi yang memungkinkan sistem menggunakan *Tensorflow* (atau sebagai alternatif *Theano* atau CNTK Microsoft) untuk membangun deep neural networks dengan cepat [14].

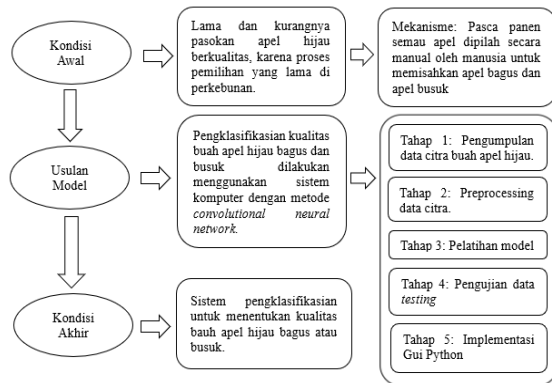
## 2.5. QT Designer

Qt Designer adalah GUI yang penuh dengan widget Qt dan alat lain untuk membuat GUI. Antarmuka *drag* dan *drop* Qt Designer memungkinkan pengguna untuk membuat dan menyesuaikan dialog, jendela, dan widget [15].

## 3. METODE PENELITIAN

Kerangka penelitian yang digunakan pada penelitian ini dapat dilihat pada gambar di bawah ini. Dapat dilihat pada gambar 1, permasalahan yang membuat dibuatnya penelitian ini adalah lama dan kurangnya pasokan apel yang berkualitas di pasar. Ini terjadi karena proses pemilahan apel yang manual sehingga memakan waktu yang cukup lama,

pasca panen semua apel diklasifikasi secara manual oleh manusia untuk memisahkan apel bagus dan apel busuk.



Gambar 1. Kerangka Penelitian

Maka dari itu peneliti mengusulkan sebuah sistem pengklasifikasian untuk mengklasifikasikan kualitas buah apel menggunakan sistem komputer dengan metode *Convolutional Neural Network*, pada sistem ini nantinya proses klasifikasi akan melalui beberapa tahap yang akan dijelaskan hingga bab hasil dan pembahasan.

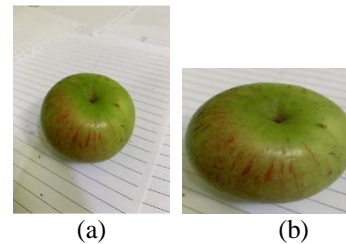
### 3.1. Pengumpulan Data

Dalam tahap pengumpulan data, dilakukan pengambilan data menggunakan kamera Iphone XR. Sebanyak 400 citra apel hijau diambil sebagai sampel, terdiri dari 200 citra apel hijau dalam kondisi baik dan 200 citra apel hijau yang sudah busuk. Penggunaan kamera Iphone XR dipilih karena kualitas gambar yang dihasilkan cukup baik untuk keperluan penelitian ini. Selanjutnya, data yang telah dikumpulkan dibagi menjadi tiga bagian, yaitu data *train* (data latih), data *validation* (data validasi), dan data *testing* (data pengujian).

Data *train* digunakan untuk melatih model atau algoritma yang akan digunakan dalam analisis lebih lanjut. Data *validation* digunakan untuk menguji performa model yang telah dilatih, serta melakukan penyesuaian parameter untuk meningkatkan keakuratan dan kinerja model. Sementara itu, data *testing* digunakan untuk menguji performa akhir model yang telah dioptimalkan, dengan tujuan menguji kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya. Dilakukan pembagian untuk data *testing* sebesar 20%, didapatkan data *testing* sebanyak 80 data citra apel hijau, sehingga sisanya akan digunakan untuk pelatihan yaitu data *train* dan *validation*. Pembagian data menjadi tiga bagian tersebut penting untuk memastikan keandalan dan keakuratan model yang dikembangkan. Selain itu, hal ini juga membantu dalam menghindari *overfitting*, yaitu kondisi di mana model terlalu mempelajari pola khusus dari data latih, sehingga tidak mampu menggeneralisasi dengan baik pada data baru.

### 3.2. Preprocessing dan Augmentasi Data

Alur Preprocessing dilakukan agar kualitas citra meningkat, dan akurasi lebih optimal, sehingga mempermudah kinerja sistem dalam mengenali pola [16]. Pada Preprocessing dataset dilakukan *cropping* terlebih dahulu lalu di *resize* menjadi 200x200px ini karena gambar yang diterima tidak memiliki ukuran piksel yang sama. Jadi jika digunakan langsung dalam program pelatihan, hasilnya akan buruk. Dapat dilihat pada gambar 2 di bawah, terlihat perbedaan ukuran dari kedua gambar yang menunjukkan hasil *cropping* dan *resize*.

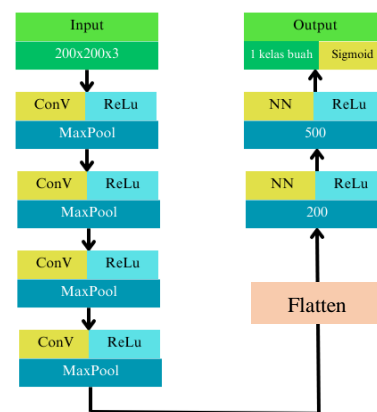


Gambar 2. (a) Sebelum, (b) Hasil *Cropping*

Setelah itu akan dilakukan proses augmentasi pada data citra, augmentasi yang dilakukan yaitu *rotasi*, *horizontal flip*, *vertikal flip*, *zoom*, dll. Ini dilakukan untuk meningkatkan performa *model machine learning*, augmentasi citra juga dapat membantu mengurangi *overfitting*, yaitu ketika model terlalu spesifik terhadap data pelatihan dan tidak dapat menangani data baru dengan baik.

### 3.3. Convolutional Neural Network

Pelatihan model diawali dengan merancang arsitektur *Convolutional Neural Network* (CNN). Dapat dilihat pada gambar 3 di bawah, gambaran arsitektur CNN yang digunakan pada penelitian ini.



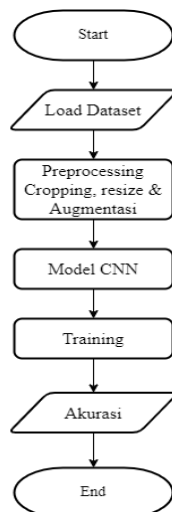
Gambar 3. Arsitektur CNN

Dalam arsitektur CNN yang telah dirancang, terdapat 4-layer konvolusi yang bertujuan untuk mengekstraksi fitur-fitur penting dari data *input*. Setiap layer konvolusi dilanjutkan dengan fungsi aktivasi *ReLU*, penggunaan *ReLU* dapat meningkatkan kecepatan pelatihan model, sehingga cocok untuk mencegah masalah *overfitting* pada

data pelatihan. Setelah setiap layer konvolusi, dilakukan operasi *max pooling* untuk mengurangi dimensi spasial dari fitur yang dihasilkan, sehingga mempertahankan informasi esensial dan mengurangi beban komputasi.

Pada langkah klasifikasi, lapisan pertama melewati langkah perataan *flattening layer*, yang mengubahnya menjadi tensor satu dimensi. Hal ini dilakukan dengan menggunakan operasi reformasi kerangka kerja Tensorflow untuk meratakan output dari lapisan konvolusi sebelumnya dan menghaluskan teksturnya untuk menghasilkan vektor fitur tunggal yang dapat digunakan oleh lapisan yang digabungkan sepenuhnya untuk melakukan klasifikasi. [17]. Lalu ada *Fully Connected Layer* terdapat dua lapisan tersembunyi (*hidden layer*) dengan 500 dan 200-unit masing-masing. Lapisan-lapisan ini bertujuan untuk menggabungkan dan memproses fitur-fitur yang telah diekstraksi sebelumnya melalui konvolusi. Setiap unit dalam *hidden layer* melakukan komputasi linear pada input yang diterima dan mengaplikasikan fungsi aktivasi ReLU hal ini membantu jaringan neural untuk mempelajari pola-pola yang kompleks dan abstrak dari data. Pada akhirnya, output dari jaringan CNN ini menggunakan fungsi aktivasi sigmoid. fungsi ini mentransformasi range nilai dari input  $x$  menjadi antara 0 dan 1 dengan bentuk distribusi fungsi. Sigmoid sangat cocok untuk klasifikasi 2 *class*, sehingga digunakan pada penelitian ini.

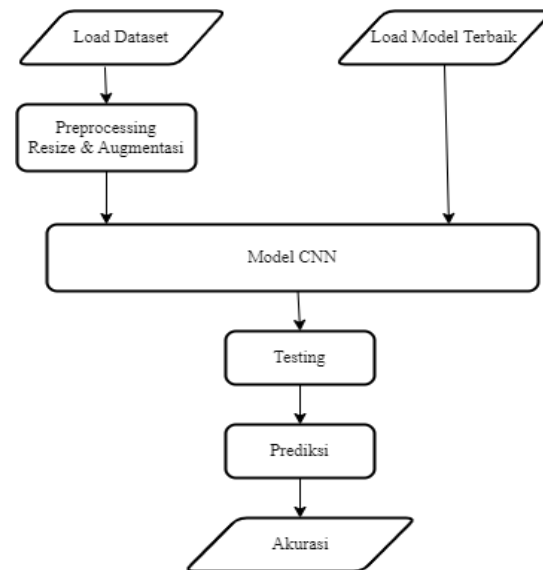
### 3.4. Desain Sistem



Gambar 4. Flowchart *training model*

Pada Gambar 4 di atas, terlihat proses pelatihan sistem ini. Pertama, data berupa dataset gambar apel dimuat untuk digunakan sebagai data pelatihan. Kedua, dilakukan *preprocessing* perubahan ukuran gambar menjadi 200x200 piksel, dan augmentasi. Ketiga, data gambar dimasukkan ke dalam model CNN, yang terdiri dari beberapa lapisan konvolusi seperti *max pooling* dan *hidden layer*. Setelah

dimasukkan ke dalam model, citra akan diubah menjadi fitur yang kemudian digunakan dalam pelatihan. Selama proses pelatihan, program mempelajari pola-pola dalam gambar dan meningkatkan akurasi pembelajaran.



Gambar 5. Flowchart Pengujian

Dari hasil *training*, program dapat mengenali pola gambar yang diberikan dengan memunculkan akurasi dari proses pelatihan. Dari proses pelatihan didapat model terbaik yang akan digunakan pada proses pengujian. Maka proses pengujian akan meload data gambar pengujian, lalu menggunakan model terbaik yang dihasilkan dari pelatihan untuk memprediksi data citra apel hijau.

## 4. HASIL DAN PEMBAHASAN

Penelitian ini melakukan klasifikasi terhadap citra buah apel hijau bagus dan apel hijau busuk. Proses program ini diawali dengan memuat data gambar lalu dilakukan *preprocessing* dengan menggunakan *library* lalu dilakukan *training* menggunakan metode CNN. Proses *training* adalah untuk menghasilkan model dan nilai yang baik, yang nantinya digunakan untuk pengujian. Parameter untuk mengukur tingkat keberhasilan model adalah akurasi, proses *training* menggunakan *packages Keras* pada python dengan *back-end Tensorflow*.

Arsitektur CNN memiliki dua bagian utama yaitu *Feature Learning* dan *Classification*. Pada *features learning* terdapat empat layer konvolusi, dengan empat pooling layer.

$$S(t) = (x * w)(t) \quad (1)$$

Keterangan:

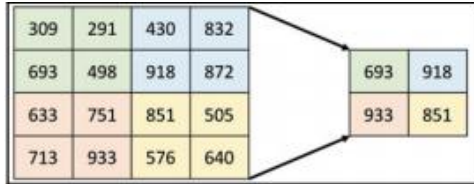
$x$  = Input

$w$  = Kernel atau filter

$t$  = Sebagai piksel

Fungsi  $s(t)$  memberikan *output* tunggal berupa *feature map*. Argumen pertama adalah *input* yang merupakan  $x$  dan argumen kedua  $w$  sebagai kernel atau filter. Apabila dilihat input sebagai citra dua

dimensi, maka bisa dikatakan  $t$  sebagai piksel. Pada konvolusi pertama menggunakan kernel dengan matriks  $3 \times 3$  dan 16 feature map dengan fungsi aktivasi ReLu, kemudian dilakukan proses pooling menggunakan max pooling yaitu mencari nilai maksimal dengan kernel  $2 \times 2$ .



Gambar 6. Operasi Max Pooling

Pada konvolusi kedua menggunakan kernel  $3 \times 3$  dan jumlah filter 32 dengan fungsi aktivasi ReLu.

$$f(x) = \max(0, x) \quad (2)$$

Keterangan:

$X$  = sebagai input, ReLu pada intinya hanya membuat pembatas pada bilangan nol, artinya apabila  $x \leq 0$  maka  $x > 0$  maka  $x = x$ .

Kemudian dilakukan *max pooling* menggunakan kernel  $2 \times 2$ . Pada layer konvolusi ketiga menggunakan kernel  $3 \times 3$  dan jumlah filter 64 dengan fungsi aktivasi ReLu, kemudian dilakukan *max pooling* menggunakan kernel  $2 \times 2$ . Di layer keempat menggunakan kernel  $3 \times 3$  dan jumlah filter 128 dengan fungsi aktivasi ReLu, kemudian dilakukan *max pooling* lagi menggunakan kernel  $2 \times 2$ , dapat dilihat parameternya pada tabel 1 di bawah.

Tabel 1. Layer Convolution

Layer	Kernel	Feature Map Size	Param
Input	0	200x200x3	0
Conv1	3x3	198x198x16	448
Pooling1	2x2	99x99x16	0
Conv2	3x3	97x97x32	4640
Pooling2	2x2	48x48x32	0
Conv3	3x3	46x46x64	18496
Pooling3	2x2	23x23x64	0
Conv4	3x3	21x21x128	73856
Pooling4	2x2	10x10x128	0

Kemudian dilanjutkan dengan proses *flatten* yaitu merubah *output* dari proses *feature learning* yang berupa matriks menjadi sebuah vektor yang kemudian akan dilanjutkan memasuki pada proses klasifikasi dengan menggunakan *Multilayer Perceptron* dengan jumlah *neuron* dan *hidden layer* yang sudah ditentukan. Lalu gambar akan diklasifikasikan berdasarkan nilai dari *neuron* pada lapisan tersembunyi dengan menggunakan aktivasi *Sigmoid* ini dikarenakan klasifikasi ini termasuk dalam kategori *binary classification*. Dapat dilihat parameternya pada tabel 2 di bawah.

Tabel 2. Hidden Layer

Layer	Output Shape	Param
Flatten 1	12800	0
Dense 1	200	2560200
Dropout 1	200	0
Dense 2	500	100500
Dropout 2	500	0
Dense 3	1	501

#### 4.1. Skenario Pelatihan Rasio Pembagian Data Train dan Validation

Skenario pembagian data *train* dan *validation* dalam pembentukan model juga diperhatikan. Jumlah data untuk setiap skenarionya sama, yaitu sebanyak 400 data, namun sudah diambil 80 data untuk data *testing*.

Tabel 3. Skenario pelatihan dengan rasio data

Rasio Train:Validation	Akurasi (%)		
	Train	Validation	Testing
60%:40%	93,75	78,12	93,12
70%:30%	95,83	91,96	92,00
80%:20%	96,88	98,44	93,75

Dapat dilihat pada tabel 3, Skenario pelatihan pertama ini didapatkan hasil terbaik dengan *accuracy training* sebesar 96,88% dan *accuracy validation* mencapai 98,44%. Dengan dilakukan pengujian menggunakan data *testing* didapatkan hasil *accuracy* yaitu 93,75%, ini terjadi pada rasio pembagian data *train* dan *validation* sebesar 80%:20%. Proses pelatihan menggunakan *optimizer* adam dan *learning rate* 0.0001 dengan input gambar sebesar 200x200px. Proses pelatihan menggunakan 10 *epoch*.

#### 4.2. Skenario Pelatihan dengan Jumlah epoch yang Berbeda

Dalam melakukan skenario pelatihan peneliti mencoba menggunakan jumlah epoch 10, 20, dan 30, menggunakan *optimizer* adam dengan *learning rate* 0,0001. Rasio pembagian data pelatihan menggunakan hasil terbaik 80%:20% Hal ini dilakukan untuk mengetahui era mana yang terbaik dan tidak membebani sistem. Epoch dari proses pelatihan sangat penting karena pembelajaran menunjukkan berapa lama bobot sistem berubah, berapa lama sistem terus mempelajari pola data di setiap epoch untuk menemukan bobot terbaik.

Tabel 4. Skenario perbandingan jumlah epoch

Epoch	Akurasi (%)		
	Train	Validation	Testing
10	96,88	98,44	93,75
20	96,88	99,22	92,43
30	93,23	100	93,75

Dapat dilihat pada tabel 4, hasil terbaik tetap berada pada *epoch* 10, Pada *epoch* 30 *accuracy train* dan *validation* memiliki perbedaan cukup jauh sehingga ini bisa disebut *overfitting*.

#### 4.3. Skenario Pelatihan dengan *Optimizer* yang Berbeda

Selanjutnya peneliti melakukan pelatihan dengan menggunakan algoritma Stochastic Gradient Descent atau biasa disebut dengan SGD dan RMSprop, dengan menggunakan hasil pelatihan terbaik sebelumnya yaitu dengan learning rate 0,0001, rasio data *train* dan *validation* 80:20%, dan *epoch* 10. Hasilnya pada percobaan SGD menurun, jauh berbeda dengan menggunakan Adam dengan learning rate 0,0001. Pada RMSprop hasil pelatihan cukup baik namun masih kalah pada data *testing* dengan *optimizer* Adam.

Tabel 5. Skenario perbandingan dengan *optimizer* berbeda

<i>Optimizer</i>	Akurasi (%)		
	Train	Validation	Testing
Adam	96,88	98,44	93,75
SGD	67,71	55,47	58,74
RMSprop	95,31	100	92,50

#### 4.4. Skenario Pelatihan dengan *Learning rate* Berbeda

Percobaan dilanjutkan dengan melatih model dengan learning rate yang berbeda. Peneliti menggunakan model CNN yang sama dan dari pelatihan terbaik sebelumnya dengan Adam sebagai *optimizer*, 10 *epoch*, dan rasio pembagian data 80%:20%. Namun dengan merubah *learning rate* menjadi 0,01 dan 0,001.

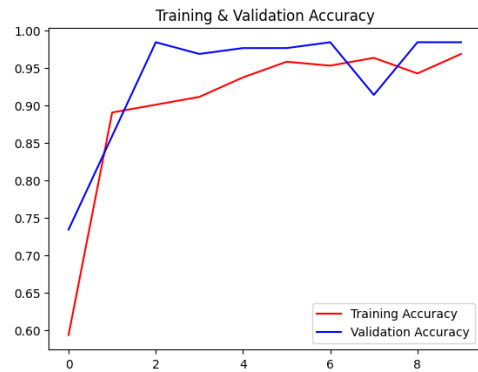
Tabel 6. Skenario dengan *Learning rate* yang berbeda

<i>Learning rate</i>	Akurasi (%)		
	Train	Validation	Testing
0,0001	96,88	98,44	93,75
0,01	95,83	98,44	90,35
0,001	86,46	92,97	93,75

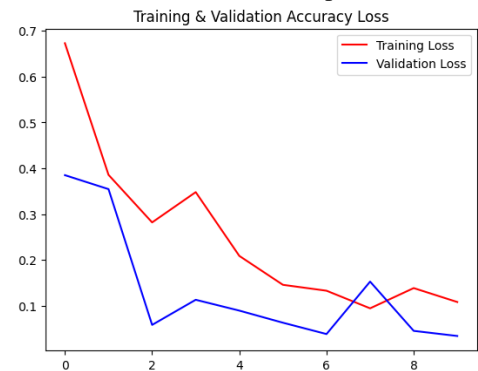
Dapat dilihat pada tabel 5, *accuracy* pada data *testing* ketiganya mendapatkan hasil yang hampir sama khususnya pada *learning rate* 0,0001 dan 0,001. Namun pada proses pelatihan masih bagus dengan *learning rate* 0,0001. Ini tetap termasuk dalam hasil yang cukup bagus.

#### 4.5. Hasil Terbaik

Dari berbagai skenario pelatihan dengan mengubah berbagai parameter didapatkan hasil model CNN terbaik yaitu dengan rasio pembagian data latih 80%:20%, menggunakan *optimizer* Adam, learningrate 0,0001, dan pelatihan dilakukan dengan 10 *epoch*. Dapat dilihat pada gambar 7, didapatkan akurasi terbaik yang tidak *overfitting* yaitu *accuracy training* 94,59% dan *accuracy validation* 94,44%, lalu dilakukan pengujian dengan data *testing* sebanyak 47 citra didapatkan hasil sempurna yaitu 100%.

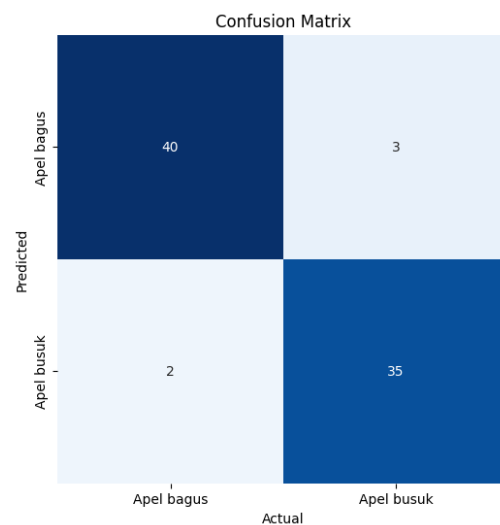


Gambar 7. Grafik akurasi hasil pelatihan terbaik



Gambar 8. Grafik *loss* hasil pelatihan terbaik

Berdasarkan gambar 8, *loss* yang diperoleh dari pelatihan terbaik sebesar 0,1082 pada data *training* dan 0,0575 pada data *validation*. Hal ini bisa dikatakan cukup rendah dan bagus dari hasil yang telah didapat dengan didukung juga oleh tingginya nilai akurasi. Dapat dilihat pada gambar 9 di bawah, *Confusion matrix* hasil pengujian pada data *testing* atau data yang belum dikenali oleh model didapatkan hasil yang cukup bagus. Untuk data *testing*, citra apel bagus yang berhasil diklasifikasi 40 citra, dan citra apel busuk yang berhasil terklasifikasi yaitu 35 citra.

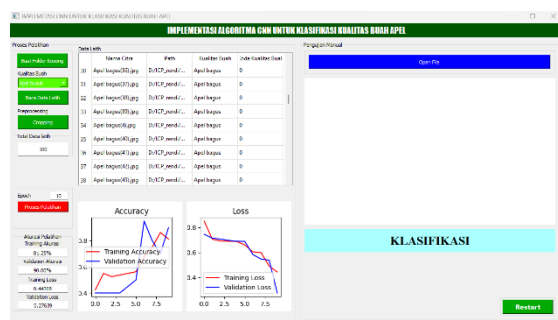


Gambar 9. *Confusion Matrix* data *testing*



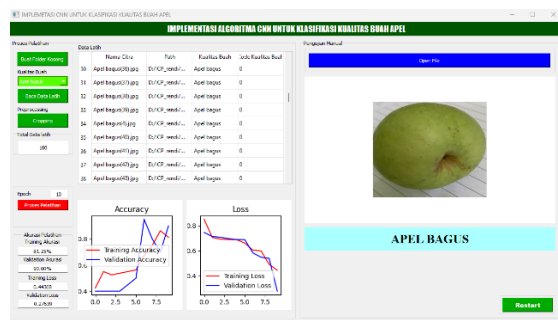
#### 4.6. Implementasi GUI Python

Pada penelitian ini aplikasi diimplementasikan dengan menggunakan GUI Python, dengan tampilan antarmuka pengguna (GUI) yang dibuat menggunakan bantuan QT Designer dan kemudian diintegrasikan ke dalam file Python. Aplikasi dibuat dengan beberapa menu diantaranya ada “Baca data latih”, “Cropping”, “Proses Pelatihan”, “Pengujian Manual”, dan “Restart”. Button “Baca data latih” ini digunakan untuk membaca data yang akan dilatih dan data yang terbaca akan ditampilkan di tabel, lalu button “Cropping” didalamnya terdapat 2 proses preprocessing yaitu Cropping dan resize. Button “Proses pelatihan” ini akan menjalankan pelatihan model dengan arsitektur CNN dan parameter terbaik yang didapat dari skenario uji coba sebelumnya.

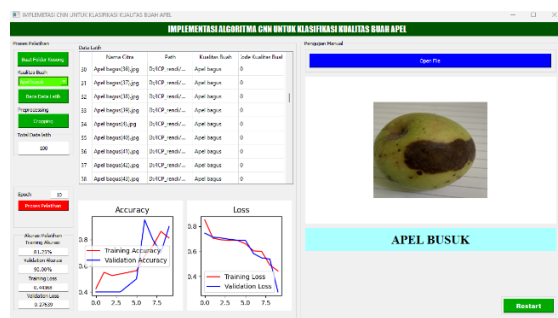


Gambar 10. Tampilan GUI Python – Training

Pada gambar 10, diperlihatkan proses *training* yang dilakukan dan akan menampilkan akurasi dan *loss* dari hasil pelatihan dalam bentuk angka maupun grafik.



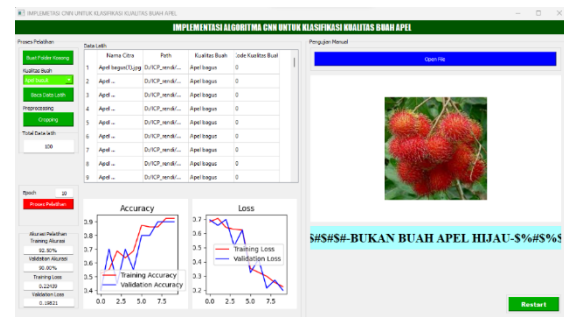
Gambar 11. Tampilan GUI Python – Pengujian 1



Gambar 12. Tampilan GUI Python – Pengujian 2

Gambar 11, memperlihatkan hasil pengujian manual dimana dilakukan pengujian pada buah apel hijau bagus, hasilnya terklasifikasi dengan benar.

Gambar 12, memperlihatkan hasil pengujian manual dimana dilakukan pengujian pada buah apel hijau busuk, hasilnya terklasifikasi dengan benar.



Gambar 13. Tampilan GUI Python – Pengujian terhadap *image* berbeda

Dapat dilihat pada gambar 13, program disetting untuk hanya bisa melakukan klasifikasi terhadap citra buah apel hijau. Disini peneliti menggunakan pengenalan warna terhadap citra Sehingga program hanya akan mengenali citra buah apel hijau.

## 5. KESIMPULAN

Berdasarkan penelitian yang dilakukan penggunaan convolutional neural network (CNN) untuk pengklasifikasian citra buah apel hijau dapat menjadi pendekatan yang akurat dan efektif. Dengan melalui tahapan pengujian dengan beberapa *hyperparameter* dan dilakukan augmentasi pada data, berhasil mencapai tingkat akurasi yang signifikan, dari 400 data citra didapatkan hasil akurasi terbaik yaitu 96,88% pada data *train* dan 98,44% pada data *validation*. Hasil klasifikasi dari data *testing* sebanyak 80 citra buah apel untuk menguji model yang terbentuk yaitu citra apel bagus yang berhasil diklasifikasi 40 citra, dan citra apel busuk yang berhasil terklasifikasi yaitu 35 citra.

Hasil penelitian ini memberikan kontribusi penting dalam pengembangan teknologi pengklasifikasian buah berdasarkan kualitas secara otomatis. Dengan adanya penelitian ini, diharapkan dapat mempercepat dan mempermudah proses pengklasifikasian buah apel secara efisien. Penelitian ini dapat menjadi dasar untuk pengembangan lebih lanjut dalam aplikasi deteksi kualitas buah, seperti pengklasifikasian kualitas buah secara real-time tentunya dengan jumlah data latih yang lebih banyak, sehingga berpotensi untuk meningkatkan efisiensi dan kualitas industri buah apel.

## DAFTAR PUSTAKA

- [1] Suryana, D., “Manfaat Buah: Manfaat Buah-buahan”, Bandung: Dayat Suryana Independent, 2018.

- [2] Anggara, T., Shinta, D., Suryanto, A., & Ainurrasjid, A., “Kendala Produksi Apel (*Malus Sylvestris Mill*) Var. *Manalagi* Di Desa *Poncokusumo Kabupaten Malang*”, Jurnal Produksi Tanaman, Vol. 5 No.2, 198-207. 2017.
- [3] Shafira, T. “Implementasi Convolutional Neural Network untuk Klasifikasi Citra Tomat menggunakan Keras”. S1. Universitas Islam Indonesia (Program studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam). Yogyakarta. 2018.
- [4] Arandjelovic, R. et al., “NetVLAD: CNN architecture for weakly supervised place recognition”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5297–5307, 2016.
- [5] Nugroho, P. A., Fenriana, I., & Arijanto, R., “Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia”, *Algor*, 2(1), 12–20, 2020.
- [6] Hafiez, T. M., Iskandar, D., SK, A. W., & Boangmanalu, R. F., “Optimasi Klasifikasi Gambar Varietas Jenis Tomat dengan Data Augmentation dan Convolutional Neural Network”, *Smart Comp: Jurnalnya Orang Pintar Komputer*, 11(2), 175–186, 2022.
- [7] Fuadah, Y. N. et al., “Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma.”, *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(3), 728, 2022.
- [8] Maulana, F. F., & Rochmawati, N. Klasifikasi Citra Buah Menggunakan Convolutional Neural Network. *Journal of Informatics and Computer Science*. Vol 01(02). 104-108. 2019.
- [9] Juliansyah, S., & Laksito, A. D. Klasifikasi Citra Buah Pir Menggunakan Convolutional Neural Networks. *Jurnal Telekomunikasi Dan Komputer*, Vol 11(1), 65–72. 2021.
- [10] Damayanti, S. A., Arkadia, A., & Prasvita, D. S. Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN. *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*. Vol 2(2). 158-165. (2021).
- [11] Kirana, K. C. Pengolahan Citra Digital: Teori dan Penerapan Pengolahan Citra Digital pada Deteksi Wajah. Malang: AhliMedia Press. 2021.
- [12] Bhardwaj, A., Di, W., & Wei, J. “Deep learning essentials: your hands-on guide to the fundamentals of deep learning and neural network modeling”. Birmingham-mumbai: Packt Publishing. 2018.
- [13] Sena, S. “Pengenalan Deep Learning Part 7: Convolutional Neural Network (CNN)”. Diakses tanggal 18 oktober 2022, dari <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>. 2017.
- [14] Bernico, M. “Deep Learning Quick Reference: Useful hacks for training and optimizing deep neural networks with TensorFlow and Keras”. Birmingham-mumbai: Packt Publishing. 2018.
- [15] Willman, J. M. “Beginning PyQt”. Apress Berkeley, CA. 2020.
- [16] Miranda, N. D., Novamizanti, L., & Rizal, S. “Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50”. *Jurnal Teknik Informatika (Jutif)*, 1(2), 61–68. 2020
- [17] Ode, L., & Sagala, A. S., “Klasifikasi Cats dan Dogs”. Diakses tanggal 25 Mei 2023, dari <https://www.researchgate.net/publication/363924413>. 2022