**IDENTIFIKASI TINGKAT KEMATANGAN BUAH KELAPA SAWIT BERDASARKAN *IMAGE FEATURE* MENGGUNAKAN ALGORITMA *CONVOLUTIONAL***

***NEURAL NETWORK***

# Muksin Alfalah\*1, Dedy Setiawan2, Pradita Eko Prasetyo Utomo3

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi,

Jambi Indonesia

**e-mail:** [1muksinalfalah02@gmail.com,](mailto:1muksinalfalah02@gmail.com) [2dedy\_doel@yahoo.com,](mailto:2dedy_doel@yahoo.com) [3pradita.eko@unja.ac.id](mailto:3pradita.eko@unja.ac.id)

# Abstrak

Kelapa sawit merupakan salah satu tanaman perkebunan yang memiliki peran penting bagi perekonomian nasional, terutama sebagai penyedia lapangan kerja, sumber pendapatan dan devisa negara. Peningkatan akan kebutuhan bahan baku produk-produk minyak kelapa sawit dalam bentuk makanan maupun non makanan tidak hanya dirasakan oleh masyarakat Indonesia tetapi juga negara lain. Hal ini memicu para petani kelapa sawit atau pengelola untuk lebih memaksimalkan hasil produksi minyak kelapa sawit dalam bentuk *Crude Palm Oil* (CPO). Salah satu kendalanya adalah tingkat kematangan dari buah kelapa sawit tersebut, sehingga perlu dilakukan sortasi untuk mendapatkan buah kelapa sawit yang berkualitas dengan tingkat kematangan yang sesuai. Berdasarkan hasil wawancara dengan Manajer PT. Rigunas Agri Utama (2020) yang mana PT. Rigunas Agri Utama merupakan salah satu pabrik pengelola buah kelapa sawit di Kabupaten Tebo, PT. Rigunas Agri Utama selalu melakukan proses sortasi terhadap buah kelapa sawit yang akan diolah menjadi minyak mentah. Proses dalam pemilihan buah kelapa sawit yang dilakukan berbagai perusahaan maupun petani kelapa sawit hanya melibatkan manusia sebagai pengambil keputusan, proses pengidentifikasian seperti itu memiliki kelemahan. Oleh sebab itu, teknologi pengenalan citra dapat menjadi salah satu solusi yang bisa digunakan untuk mengatasi masalah tersebut. Salah satu teknologi dalam pengenalan citra adalah algoritma CNN, Berdasarkan penelitian didapatkan tingkat akurasi sebesar 85% pada proses *training* dan 75% pada proses *testing*. Kemudian penelitian ini

menggunakan data baru untuk menguji model yang telah dibuat dan menghasilkan nilai akurasi yang baik dalam mengenali buah kelapa sawit. Sehingga dapat disimpulkan bahwa model dan *Website* yang telah mampu melakukan identifikasi buah kelapa sawit.

**Kata Kunci:** CNN, Kelapa Sawit, Identifikasi, Akurasi, Citra, *Deep Learning*, model

# PENDAHULUAN

Kelapa sawit merupakan salah satu tanaman perkebunan yang memiliki peran penting bagi perekonomian nasional, terutama sebagai penyedia lapangan kerja, sumber pendapatan dan devisa negara. Pada Tahun 2020, pertumbuhan perkebunan kelapa sawit tercatat mencapai 14.824,60 hektar. Sedangkan untuk luas areal perkebunan kelapa sawit di Indonesia pada tahun 2020 seluas

16.381.000 hektar (Menteri Pertanian, 2020). Kelapa sawit merupakan tanaman penghasil minyak sawit dan inti sawit. cBanyak hal yang mempengaruhi kualitas dari minyak kelapa sawit, diantarnya, kandugan air, kandungan kotoran, kandungan asam lemak bebas, dan salah satu yang paling penting yaitu tingkat kematangan dari buah kelapa sawit (Rangkuti dan Syahputra, 2019). Tetapi dalam praktiknya salah satu kendala kendalanya adalah tingkat kematangan dari buah kelapa sawit tersebut, sehingga perlu dilakukan sortasi untuk mendapatkan buah kelapa sawit yang berkualitas dengan tingkat kematangan yang sesuai. Berdasarkan hasil wawancara dengan Manajer PT. Rigunas Agri Utama (2020) yang mana PT. Rigunas Agri Utama merupakan salah satu pabrik pengelola buah kelapa sawit di Kabupaten Tebo, PT. Rigunas Agri Utama selalu melakukan proses sortasi terhadap buah kelapa sawit yang akan diolah menjadi minyak mentah supaya mendapat hasil minyak berkualitas. Untuk jumlah buah kelapa sawit yang di sortasi oleh pabrik, sekitar 5% buah kelapa sawit yang tidak siap untuk diolah selanjutnya. Proses dalam pemilihan buah kelapa sawit yang dilakukan berbagai perusahaan pengolah maupun para petani yang menanam kelapa sawit pada umumnya hanya dilakukan secara manual dan melibatkan manusia sebagai pengambil keputusan. Cara manual yang dilakukan terlalu banyak memakan waktu, sehingga jika

diterapkan pada skala industri besar diperlukan bantuan mesin pada proses tersebut. Oleh karena itu perlu dibuatkan sistem yang dapat digunakan untuk mengenali tingkat buah kematangan buah kelapa sawit sesuai kategori yang telah ditentukan. Dalam penelitian ini diharapkan mampu memberikan kontribusi dan menjadi alternatif bagi pengelola buah kelapa sawit atau petani dalam membantu proses sortasi. Dalam hal ini pengolahan citra merupakan solusi untuk mengatasi masalah tersebut. Salah satu teknologi yang dapat digunakan yaitu menggunakan teknologi deep learning. Teknologi *Deep Learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network*. Karena metode *Convolutional Neural Network* berusaha meniru sistem pengenalan citra pada *visual cortex* manusia sehingga memiliki kemampuan mengolah informasi citra selayaknya manusia (Maulana & Rochmawati, 2019). Tantangan menggunakan metode CNN adalah bagaimana menghasilkan akurasi yang tinggi dengan biaya komputasi yang relatif rendah. Berdasarkan latar belakang diatas penelitian ini menerapkan implementasi dari metode *deep learning* menggunakan CNN untuk membantu mengidentifikasi tingkat kematangan buah kelapa sawit. Penelitian ini diharapkan mampu menghasilkan akurasi yang terbaik dan juga mampu untuk mengklasifikasikan kematangan buah kelapa sawit.

# TINJAUAN PUSTAKA

* 1. Kelapa Sawit

Kelapa Sawit (Elaeis) adalah tumbuhan industri sebagai bahan baku penghasil minyak yang termasuk dalam keluarga Arecaceae yang terdiri dari dua spesies yaitu kelapa sawit Afrika (Elaeis guineensis) dan kelapa sawit Amerika (Elaeis oleifera). Kelapa sawit merupakan salah satu komoditas hasil perkebunan yang mempunyai peran cukup penting dalam kegiatan perekonomian di Indonesia karena kemampuannya menghasilkan minyak nabati yang banyak dibutuhkan oleh sektor industri. Suatu indikator/alat ukur kematangan tandan buah segar yang diukur melalui jumlah buah yang membrondol dan warna dari buah tersebut. Pengetahuan akan fraksi buah kelapa sawit berfungsi untuk memperkirakan hasil panen secara

kuantitas dan kualitas yang akan diperoleh. Idealnya, proses panen TBS dilakukan pada fraksi 3 atau buah matang (Pahan, 2015). Berikut ini merupakan karakteristik kelapa sawit tersebut:

1. Sawit Mentah

Sawit mentah merupakan kategori buah kelapa sawit yang belum siap diolah. Karakter morfologi dari buah tersebut adalah warna kulit masih berwarna hijau kehitaman dan tidak ada buah yang memberondol. Berdasarkan deskripsi tersebut bentuk gambar dapat dilihat pada Gambar berikut:



Gambar 1. Sawit Mentah

1. Sawit Matang

Sawit matang merupakan kategori buah kelapa sawit yang siap untuk di olah. Karakter morfologi dari buah tersebut adalah Buah kelapa sawit yang matang mempunyai warna cerah yang didominasi oleh warna kuning kemerahan dan 12- 25% buah memberondol. Berdasarkan deskripsi tersebut bentuk gambar dapat dilihat pada Gambar berikut:



Gambar 2. Sawit Matang

1. Sawit Busuk

Sawit busuk merupakan kategori buah kelapa sawit yang siap diolah atau sebaliknya. Karakter morfologi dari buah tersebut adalah warna kulit coklat kehitaman (Supriadi, 2019). Berdasarkan deskripsi tersebut bentuk gambar dapat dilihat pada Gambar berikut:



Gambar 3. Sawit Busuk

* 1. *Convolutional Neural Network Convolutional Neural Network* atau

dikenal juga dengan sebutan *ConvNets* adalah sebuah metode untuk memproses data dalam bentuk beberapa array, contohnya yaitu gambar berwarna yang terdiri dari tiga array 2D yang mengandung intensitas piksel dalam tiga jenis warna. *Convolutional Neural Network*s (*ConvNets*) merupakan penerapan dari *Artificial Neural Networks* (ANN) yang lebih istimewa dan saat ini diklaim sebagai model terbaik untuk memecahkan masalah pengenalan objek (Goodfellow et al., 2016).

Penelitian awal yang mendasari penemuan *Convolutional Neural Network* pertama kali dilakukan oleh Hubel dan Wiesel (Hubel & Wiesel, T, 1968) mengenai *virual cortex* pada indera penglihatan kucing. Secara teknis, *Convolutional Neural Network* memiliki arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan dan keluaran dari masing-masing tahap adalah beberapa array yang disebut *feature map* atau peta fitur. Contohnya untuk citra *greyscale*, *input* atau masukan adalah berupa *matriks* dua dimensi. *Output* dari masing- masing tahap adalah *feature map* hasil pengolahan dari semua lokasi pada citra masukan. Masing-masing tahap terdiri dari tiga lapisan yaitu *convolution*, aktivasi dan *pooling*. Secara umum, arsitektur dari sebuah *convolution Neural Network* ditunjukkan pada Gambar 4.

2.2.2 Activation Layer

Fungsi aktivasi merupakan operasi matematik yang dikenakan pada sinyal *output*. Fungsi aktivasi berfungsi menentukan apakah suatu *neuron* aktif atau tidak berdasarkan *weighter sum* dari suatu *input*. Beberapa jenis fungsiaktivasi yang sering digunakan pada *Deep Learning* adalah *sigmoid*, *Tanh*, algebraic *sigmoid*, *ReLU*, noisy *ReLU*, Leakly *ReLU*/P*ReLU*, *Randomized Leakly ReLU*, dan *Eksponential Linear Unit* (Khan et al., 2018). Secara garis besar fungsi aktivasi akan di jelaskan sebagai berikut:

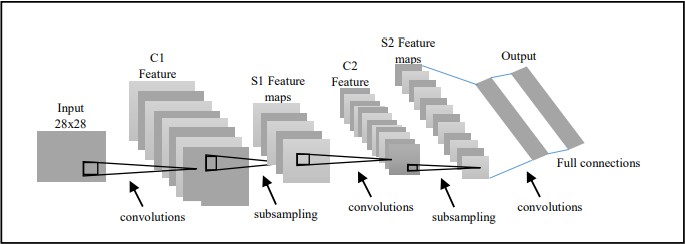
1. ReLU

Fungsi yang digunakan untuk aktivasi pada *ReLU* adalah sebagaimana ditunjukkan pada persamaan (2). Dari persamaan tersebut, maka nilai *output* dari *neuron* bisa dinyatakan sebagai 0 jika *input*nya adalah negatif. Jika nilai *input* dari fungsi aktivasi adalah positif, maka *output* dari *neuron* adalah nilai *input* aktivasi itu sendiri (Krizhevsky et al., 2012).

𝑓(𝑥) = max(0, 𝑥) (2)

1. Sigmoid

Fungsi *sigmoid* adalah fungsi non- *linear* yang mempunyai persamaan matematika sebagaimana ditunjukkan pada Persamaan (3). Masukan untuk fungsi aktivasi tersebut adalah nilai real dan keluaran dari fungsi tersebut adalah nilai antara 0 dan 1. Jika masukannya sangat negatif, maka keluaran yang didapatkan adalah 0, sedangkan jika masukan sangat positif maka nilai keluaran yang didapatkan adalah 1.

𝑓(𝑥) = 1

1+ 𝑒−𝑥

(3)

Gambar 4. Arsitektur CNN

2.2.1 *Convolutions Layer*

Operasi *convolution* merupakan operasi dua fungsi argumen yang bernilai nyata (Goodfellow et al., 2016). Operasi ini menerapkan fungsi keluaran (*output*) sebagai *Feature Maps* dari masukan (*input*) citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen yang mempunyai nilai riil. Operasi untuk *convolution* ke dalam *input* lebih dari satu dimensi dapat ditujukan pada persamaan (1).

* + 1. *Pooling Layer*

*Pooling layer* terletak setelah *convolution layer*. *Pooling layer* terdiri atas sebuah *filter* dengan ukuran dan *stride* tertentu yang secara bergantian bergeser pada seluruh area *feature map*.Jenis *poolinglayer*yang biasa digunakan yaitu *average pooling* dan *max pooling*. *Max Pooling* adalah proses untuk meningkatkan invariansi posisi dari fitur menggunakan operasi *Max-pooling* (Putra et al., 2016).

𝑆(𝑖, 𝑗) = (𝐾 ∗ 𝐼)(𝑖, 𝑗) = ∑𝑚

𝑖=1

𝑛

𝑗=1

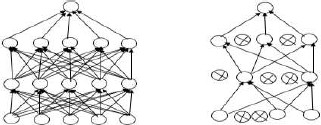
∑

𝐼(𝑖 −

𝑚, 𝑗 − 𝑛)𝐾(𝑚, 𝑛) (1)

* + 1. *Dropout Layer*

*Dropout* yaitu merupakan teknik regularisasi jaringan syaraf dimana beberapa akan dipilih secara *random* dan tidak dipakai selama data latih. Teknik regularisasi adalah teknik yang digunakan untuk melakukan modifikasi pada model neural network yang bertujuan untuk mengurangi generalization error, bukan mengurangi *training* error seperti peran loss function. Kemudian *neuron-neuron* ini dibuang juga secara *random* (Srivastava et al., 2014).



*a. Neural* Biasa b. Setelah dropout Gambar 5*. Operasi Dropout*

* + 1. Fully-Conected Layer

2.2.7 *Cross Entropy Loss Function*

*Loss Function* atau *Cost Function* merupakan fungsi yang menggambarkan kerugian yang terkait dengan semua kemungkinan yang dihasilkan oleh model. *Loss Function* bekerja ketika model pembelajaran memberikan kesalahan yang harus diperhatikan. *Loss Function* yang baik adalah fungsi yang menghasilkan error yang diharapkan paling rendah. Ketika suatu model memiliki kelas yang cukup banyak, perlu adanya cara untuk mengukur perbedaan antara probabilitas hasil hipotesis dan probabilitas kebenaran yang asli, dan selama pelatihan banyak algoritma yang dapat menyesuaikan parameter sehingga perbedaan ini diminimalkan (Zhu et al., (2020). Fungsi yang dilakukan pada proses *Cross Entropy Loss Function* dapat ditunjukkan pada persamaan (4) sebagai berikut:

1

∑ (𝑡𝑎𝑟𝑔𝑒𝑡(𝑥) − 𝑎𝑐𝑡𝑖𝑣𝑎𝑡𝑖𝑜𝑛(𝑥))2

*Fully-Conected Layer* adalah sebuah lapisan dimana semua *neuron* aktivasi dari lapisan sebelumnya terhubung semua dengan *neuron* di lapisan selanjutnya

sama seperti halnya dengan *Neural*

𝑁 𝑥

1

= ∑ (𝑡𝑎𝑟𝑔𝑒𝑡(𝑥)

𝑁 𝑥

|𝑥|

− max (0, ∑ 𝑤𝑖𝑥𝑖 + 𝑏))2

𝑖

*Network* biasa. Pada dasarnya lapisan ini biasanya digunakan pada MLP (Multi- Layer Perceptron) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara *linear* (Lau et al., 2015).

2.2.6 Optimizer

*Optimizer* merupakan salah satu parameter yang dibutuhkan untuk membangun model dalam *package Keras. Optimizer* memliki peranan penting untuk meningkatkan akurasi sebuah model. Beberapa *optimizer* yang terdapat dalam package Keras yaitu Adam, Adamax, dan RMSprop. Dalam penelitian ini hanya menggunakan metode Adam. Adam (*Adaptive Moment Estimation)* adalah metode yang menghitung *adaptive learning rate* untuk setiap parameter. Nilai parameter yang direkomendasikan adalah β1 = 0.9, β2 =0.999, dan ϵ = 10-8 dengan β1 = β2 = tingkat penurunan eksponensial dan ϵ = nilai epsilon untuk update parameter (Rawat & Wang., 2017).

2.2.8 Confusion *Matriks*

Penentuan baik atau tidaknya performa suatu model klasifikasi dapat dilihat dari parameter pengukuran performanya, yaitu

tingkat akurasi, recall, dan 4

presisi. Untuk menghitung faktor-

faktor tersebut diperlukan sebuah matrik yang biasa disebut confusion *matriks*. Persamaan-persamaan *matriks* yang digunakan yaitu *Recall, Precision* dan *Accuracy/F1-Score*.

# Recall

Recall atau sensitifitas sistem terhadap suatu kelas juga dapat dilihat. Recall dapat dihitung dengan menggunakan persamaan (5).

𝑇𝑃

𝑆𝑒𝑛𝑠𝑖𝑡𝑖𝑓𝑖𝑡𝑎𝑠 𝑃𝑟𝑒𝑑𝑖𝑘𝑠𝑖 =

𝑇𝑃 + 𝐹𝑁

# Precision

Sementara untuk menghitung 5

tingkat presisi prediksi kejadian dapat digunakan persamaan (6).

𝑇𝑃

𝑃𝑟𝑒𝑠𝑖𝑠𝑖 𝑃𝑟𝑒𝑑𝑖𝑘𝑠𝑖 =

𝑇𝑃 + 𝐹𝑃

# Accuracy

Presisi menggambarkan 6

seberapa tepat suatu model memprediksi kejadian positif 7

dalam serangkaian kegiatan prediksi. Perhitungan presisi biasanya bermanfaat pada pengembangan model prediksi hujan di suatu daerah. Selain presisi dan akurasi, untuk dapat melihat lebih detail lagi kinerja suatau sistem,

𝑇𝑃/𝑇𝑁

terlebih dahulu sebelum masuk jaringan. *Augmentasi* merupakan proses mengubah atau memodifikasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda, namun manusia

masih dapat mengetahui bahwa gambar

𝐴𝑘𝑢𝑟𝑎𝑠𝑖 =

𝑃 + 𝑁

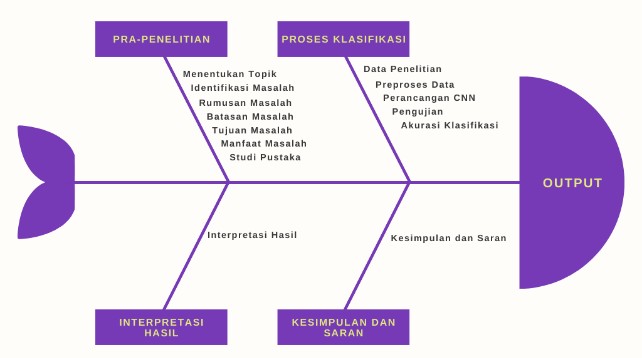
yang diubah tersebut adalah gambar yang

Akurasi digunakan sebagai parameter sebagaimana akurat suatu model melakukan klasifikasi (Fawcett, 2006).

# METODOLOGI PENELITIAN

* 1. Tahapan Penelitian

Proses pada tahapan penelitian ditunjukkan pada Gambar 6 Sebagai berikut:



Gambar 6. Diagram *Fishbone* Penelitian

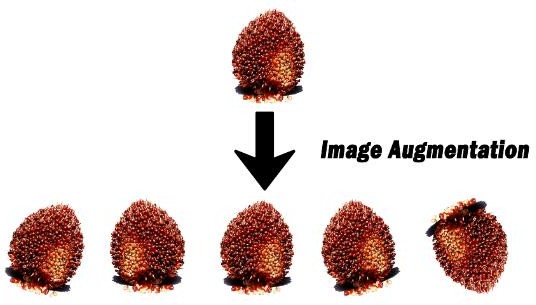
* 1. Data Penelitian

Dalam melakukan penelitian ini, langkah pertama yaitu pengumpulan dataset berupa citra buah kelapa sawit yang dibagi menjadi tiga jenis kelas, yaitu gambar buah kelapa sawit mentah, matang dan busuk. Dataset diambil dari berbagai sumber internet dan juga di foto secara langsung ke lapangan menggunakan kamera handphone sesuai kelas yang telah ditentukan. Dataset berjumlah 1000 gambar yang di bagi menjadi 2 bagian, proses pembagian menggunakan teknik *Train Test Split* yang disediakan oleh *package Sklearn,* selanjutnya data dibagi menjadi 80% data *training* dan 20% data *testing*.

* 1. Praproses Data

Sebelumnya dataset akan di bagi menjadi beberapa bagian, yaitu 80% data latih dan 20% data *testing*. Kemudian setiap gambar buah kelapa sawit akan dirubah dan disimpan berukuran 150x150 pixel dengan *channel* berukuran 3, *input shape* berukuran 150x150x3, *batch size* 16 dan *epoch* 10 untuk kemudian dijadikan sebagai dataset. Gambar tersebut kemudian di augmentasi

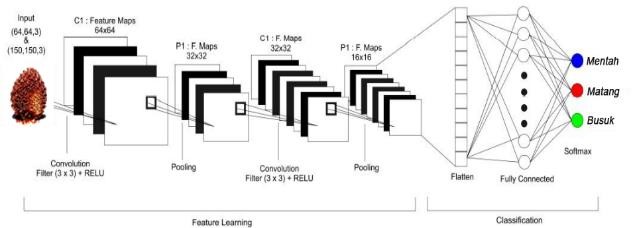
sama. Pada penelitian ini dilakukan proses augmentasi dengan cara *rescaling* menjadi 1/255, *shearing* dengan skala 0.2, *zooming image* dengan *range* 0.2 dan melakukan *horizontal flip,* berikut merupakan ilustrasi gambar dari proses augmentasi.



Gambar 7. Ilustrasi Augmentasi Citra

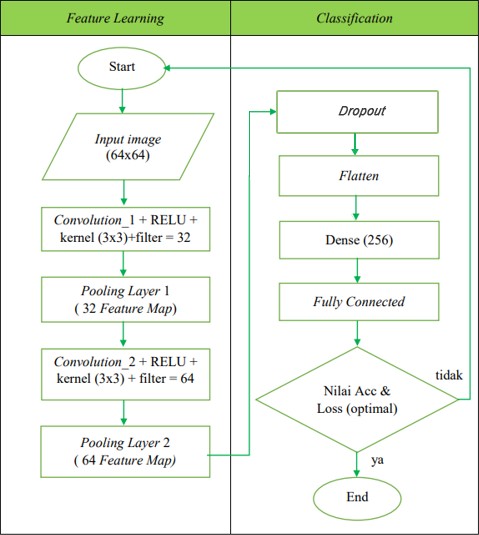
* 1. Rancangan *Convolutional Neural Network* (CNN)

Setelah dilakukan pembuatan data, langkah selanjutnya adalah melakukan pelatihan model CNN. Umumnya dalam CNN memiliki 2 tahapan, yaitu tahap feature learning dan classification. Input gambar pada model CNN menggunakan citra yang berukuran 128x128x3. Angka tiga yang dimaksud adalah sebuah citra yang memiliki 3 channel yaitu Red, Green, dan Blue (RGB) Citra masukan kemudian akan diproses terlebih dahulu melalui proses convolution dan proses pooling pada tahapan feature learning. Jumlah proses convolution pada rancangan ini memiliki dua lapisan convolution. Setiap convolution memiliki jumlah filter dan ukuran kernel yang berbeda. Kemudian dilakukan dropout yang di mulai dengan proses flatten atau proses mengubah feature map hasil pooling layer kedalam bentuk vector. Proses ini biasa disebut dengan tahap fully Connected layer. Berikut adalah rancangan dari arsitektur CNN pada penelitian ini yang dapat dilihat pada Gambar 8. Berdasakan arsitektur pada Gambar 8 dijelaskan terdapat dua tahap dalam arsitektur CNN, yaitu *Feature Learning* dan *classification*. *Feature learning* adalah teknik yang memungkinkan sebuah system berjalan secara otomatis untuk menentukan

representasi dari sebuah *image* menjadi *features* yang berupa angka-angka yang merepresentasikan *image* tersebut. Tahap *Classification* adalah sebuah tahap dimana hasil dari *feature learning* akan digunakan untuk proses klasifikasi berdasarkan *subclass* yang sudah ditentukan. Jika *flowchart* diatas diubah kedalam bentuk gambar, maka dapat dilihat seperti Gambar 9.

Gambar 9. Arsitektur CNN Berdasarkan gambar tersebut pada

proses *convolution* pertama menggunakan jumlah *filter* sebanyak 64 dan kernel dengan *matriks* 3x3. Kemudian dilakukan proses *pooling* menggunakan ukuran *pooling* 2x2 dengan pergeseran *mask* sebanyak dua langkah. Kemudian pada tahapan *convolution* kedua dengan menggunakan jumlah *filter* sebanyak 32 dan *kernel* dengan *matriks* 2x2. Kemudian di lanjutkan dengan *flatten* yaitu merubah *output* dari proses *convolution* yang berupa *matriks* menjadi sebuah *vector* yang selanjutnya akan diteruskan pada proses klasifikasi dengan menggunakan MLP (*Multi-Layer Perceptron*) dengan jumlah *neuron* pada lapisan tersembunyi yang telah ditentukan. Kelas dari citra kemudian diklasifikasikan berdasarkan nilai dari *neuron* pada lapisan tersembunyi dengan menggunakan fungsi aktivasi *softmax*.



Gambar 8. Flowchart CNN

* 1. Rancangan Pengujian Model

Pengujian dilakukan untuk melakukan

evaluasi terhadap model yang dihasilkan oleh CNN. Tahap *Testing* adalah tahap pengujian model yang sudah dilakukan pada tahap pelatihan. Jumlah data uji dalam penelitian ini sebanyak 200 data gambar, dengan jumlah gambar perkelas sebanyak 66 gambar. Pada tahap ini model di uji dengan gambar yang berbeda dengan tujuan menguji apakah model sudah menghasilkan performa yang baik dalam mengklasifikasikan sebuah gambar. Evaluasi model ini dilakukan dengan menggunakan fitur dari *Tensorflow* yaitu *model.evaluate* dan juga menggunakan teknik *confusion matrix* untuk melihat lebih rinci dari evaluasi modelnya. Pada Tabel 1 merupakan rancangan *confusion matriks* penelitian ini.

Tabel 1. *Confusion Matriks*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Matriks* | | Prediksi Kelas | | |
| Ment ah | Mata ng | Bus uk |
| Kelas sebenar nya | Ment ah |  |  |  |
| Mata ng |  |  |  |
| Busu  k |  |  |  |

Setelah mendapatkan nilai prediksi selanjutnya dilakukan perhitungan akurasi dari keseluruhan *matriks* dengan persamaan sebagai berikut:

𝑂𝑣𝑒𝑟𝑎𝑙𝑙 𝐴𝑐𝑐𝑢𝑟𝑎𝑐𝑦 = 𝑇𝑇𝑃𝑎𝑙𝑙

𝑇𝑜𝑡𝑎𝑙 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑇𝑒𝑠𝑡𝑖𝑛𝑔 𝐸𝑛𝑡𝑟𝑖𝑒𝑠

* 1. Rancangan Pengujian Gambar

Pada proses ini dilakukan untuk mengimplementasikan model terhadap data citra. Pada proses ini dilakukan beberapa tahap untuk menghasilkan nilai akurasi dari citra yang di uji, tahap pertama yaitu dengan *input* parameter untuk data citra, lalu definisikan label yang akan dilihat hasilnya, setelah itu lakukan definisi praproses data yang mana data tersebut sebelum dilihat hasil ujinya akan di praproses terlebih dahulu supaya data lebih terstruktur, kemudian load model yang telah di buat sebelumnya, selanjutnya *input*kan gambar yang akan di uji dan yang terakhir yaitu melakukan prediksi citra atau hasil proses pengujian citra, maka akan keluar hasil prediksi

sesuai label yang telah di definisi sebelumnya. Jika dijadikan *flowchart* maka proses pengujian gambar dapat di lihat pada Gambar 10 Sebagai berikut:



Define Label

*Input*

Start



Prapros es Data

*Input*

Predict Image

End

Load

Gambar 10. *Flowchart Testing*

# HASIL DAN PEMBAHASAN

Pada penelitian ini, peneliti melakukan klasifikasi tiga kelas gambar buah kelapa sawit, yaitu sawit matang, sawit mentah dan sawit busuk dengan menggunakan algoritma *Convolutional Neural Network* (CNN). Proses pertama yang dilakukan yaitu mengolah data nya, kemudian proses kedua dilakukan preproses data, akan tetapi proses utama dalam pembuatan model ini diawali dengan proses *training* data. Proses ini bertujuan untuk pembentukan model yang akan digunakan untuk pengujian data *testing*. Parameter untuk mengukur tingkat keberhasilan model adalah nilai akurasi. Nilai akurasi model dapat ditentukan dengan melakukan pengujian menggunakan data *testing*. Proses *training* menggunakan *packages Keras* pada *python* dengan *back-end tensorflow*. *Keras* merupakan salah satu modul yang dibuat oleh *Google* untuk mempermudah dalam *research* mengenai *neural network* dan mampu berjalan diatas *tensorflow, theano, MXNet*.

* 1. Hasil Pengolahan Data Penelitian

Pada tahap ini dilakukan

pengumpulan data terlebih dahulu, proses pengumpulan data dilakukan dengan dua cara yaitu: mengambil citra nya secara langsung menggunakan kamera dan mengunduh gambar yang tersedia di internet sesuai dengan kriteria yang telah di tentukan dan melakukan foto secara langsung menggunakan kamera telepon genggam. Setelah data terkumpul selanjutnya gambar tersebut di olah terlebih dahulu menggunakan aplikasi *photosop* untuk menghilangkan *background* dan merubahnya menjadi warna putih, hal ini dilakukan untuk memaksimalkan kinerja program dalam membaca *pixel* gambar, karena jika

gambar tersebut menggunakan *background* bawaan yang warnanya *random* akan mempengaruhi pembacaan program terhadap *pixel* gambar. Pada Tabel 2 merupakan citra sebelum dan sesudah dilakukan penghapusan *background*:

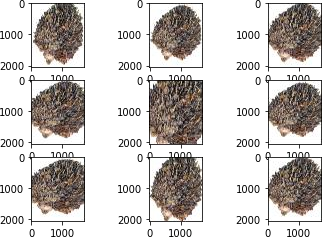
Tabel 2. Proses Penghapusan Background

|  |  |
| --- | --- |
| **Sebelum** | **Sesudah** |
|  |  |

Setelah proses tersebut selanjutnya dilakukan pembagian data dengan teknik *split* data yang disediakan oleh modul *sklearn* dengan membagi data menjadi 80% dan 20% setelah dilakukan proses diatas maka hasilnya yaitu jumlah data yang akan digunakan dalam penelitian ini yang terbagi menjadi 2 bagian, yaitu data *training* yang berjumlah 400 citra dan data *testing* berjumlah 80 citra.

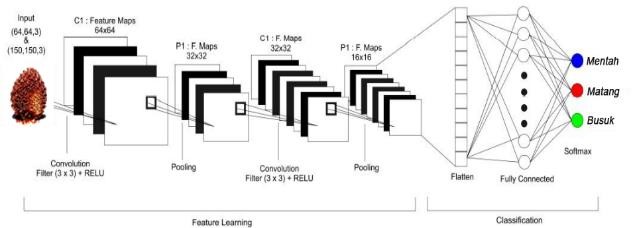
* 1. Hasil Praproses Data

Praproses dilakukan dilakukan dengan melakukan *segmentasi* dan perubahan ukuran pada citra. Pada penelitian ini praproses data dilakukan dengan teknik *augmentasi* supaya komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda. Proses *augmentasi* yang dilakukan yaitu berupa *rotation range* menjadi 15, *rescaling* menjadi 1/255, *shearing* dengan skala 0.2, *zooming image* dengan *range* 0.2, melakukan *horizontal flip* kemudian melakukan *width* dan *height shift* dengan *range* 0.1. setelah dilakukan proses tersebut variable yang telah di buat sebelumnya kemudian akan di panggil lagi untuk proses implementasinya terhadap dataset, bagian *train\_generator* begitu juga untuk data yang lainnya. Serta untuk hasil dari praproses dapat di lihat pada Gambar 11.



Gambar 11. Hasil Augmentasi

* 1. Hasil Rancangan *Convolutional Neural Network*



Gambar 12. Arsitektur CNN Gambar 12 merupakan arsitektur

jaringan pada proses *training* untuk menghasilkan model yang optimal. Penelitian ini menggunakan *input* gambar dengan ukuran 128x128x3, tujuannya adalah untuk membandingkan nilai akurasi berdasarkan ukuran gambarnya. Arsitektur diatas dapat dijelaskan sebagai berikut:

1. Proses *Convolution* pertama digunakan kernel berukuran 3x3 dan jumlah *filter* sebanyak 128 *filter*, proses *convolution* ini adalah proses kombinasi antara dua buah *matriks* yang berbeda untuk menghasilkan suatu nilai *matriks* yang baru. Setelah proses *convolution*, maka ditambahkan sebuah aktivasi fungsi yaitu RELU (*Retrified Linear Unit*). Fungsi aktivasi ini bertujuan untuk mengubah nilai negative menjadi nol (menghilangkan nilai negatif dalam sebuah *matriks* hasil *convolution*). Hasil *convolution* ini memiliki ukuran yang sama yakni 128x128 karena pada saat proses *convolution* digunakan nilai *padding* 0.
2. Proses *pooling* pada dasarnya terdiri dari sebuah *filter* dengan ukuran dan tertentu yang akan secara bergantian bergeser pada seluruh area feature map. Penelitian ini menggunakan *max- pooling* untuk mendapatkan nilai *matriks* yang baru hasil dari proses *pooling*. Berdasarkan hasil *pooling* menghasilkan *matriks* baru berukuran 63x63 dengan menggunakan kernel *pooling* 2x2. Cara kerja *max*-*pooling*adalah mengambil nilai paling maksimum berdasarkan pergeseran kernelnya sebanyak nilai stridenya yaitu 2.
3. Proses *Convolution* kedua yaitu meneruskan hasil dari proses *pooling* pertama yakni dengan *input matriks* gambar sebesar 32x32 dengan jumlah *filter* sebanyak 64 *filter* dan dengan ukuran kernel 3x3. Proses *convolution* kedua ini sama-sama menggunakan fungsi aktivasi RELU.
4. Proses selanjutnya masuk ke proses *pooling* yang kedua, proses ini hampir sama dengan proses *pooling* yang pertama, namun ada perbedaan pada nilai *output* akhir dari matriksnya. *Output* yang dihasilkan memiliki ukuran gambar 30x30.
5. Proses *Convolution* ketiga yaitu meneruskan hasil dari proses *pooling* pertama yakni dengan *input matriks* gambar sebesar 32x32 dengan jumlah *filter* sebanyak 32 *filter* dan dengan ukuran kernel 3x3. Proses *convolution* kedua ini sama-sama menggunakan fungsi aktivasi RELU.
6. Proses selanjutnya masuk ke proses *pooling* yang ketiga, proses ini hampir sama dengan proses *pooling* hang pertama, namun ada perbedaan pada nilai *output* akhir dari *matriks*nya. *Output* yang dihasilkan memiliki ukuran gambar 14x14.
7. Selanjutnya *Flatten* atau *fully- connected*. Pada tahap ini digunakan hanya satu *hidden layer* pada jaringan MLP (*Multi-Layer Perceptron*). *Flatten* disini mengubah *output pooling layer* menjadi sebuah *vector*. Sebelum melakukan proses klasifikasi atau memprediksi gambar, pada proses ini digunakan nilai *Dropout. Dropout* adalah sebuah teknik regulasi jaringan syaraf dengan tujuan memilih beberapa *neuron* secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain *neuron-neuron* tersebut dibuang secara acak. Tujuan dari proses ini yaitu mengurangi *overfitting* pada saat proses *training*. Jadi pada proses *dropout* data yang sebelumnya terbagi menjadi beberapa *neuron* kemudian diambil beberapa *neuron* saja yang akan digunakan dalam proses *training* hal ini supaya dapat mengurangi overfitting dan menghemat waktu pada proses pembuatan model.

Berdasarkan uraian penjelasan dari arsitektur jaringan diatas, arsitektur tersebut digunakan untuk proses *training*. Sehingga dari proses *training* didapatkan model dari arsitektur tersebut. Berikut model yang terbentuk:

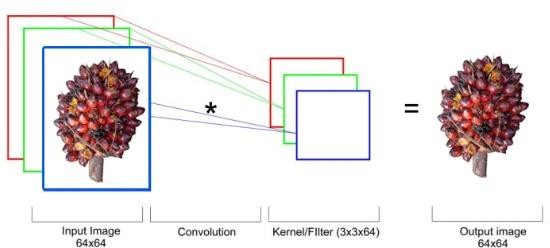
Tabel 3. Model CNN

|  |  |  |
| --- | --- | --- |
| *Nama* | *Size* | *Parame ter* |
| *input* | 128\*128\*3 | 0 |
| *conv2d\_2* | 3\*3\*3+1\*1  28 | 896 |
| *batch\_normaliza tion\_1* | 256+256 | 512 |
| *MaxPool\_1* | 63\*63\*128 | 0 |
| *dropout\_1* | 63\*63\*128 | 0 |
| *conv2d\_2* | 3\*3\*128+1  \*64 | 73792 |
| *batch\_normaliza tion\_2* | 128+128 | 256 |
| *MaxPool\_2* | 30\*30\*64 | 0 |
| *dropout\_2* | 61\*61\*64 | 0 |
| *conv2d\_3* | 3\*3\*64+1\*  32 | 18464 |
| *MaxPool\_3* | 14\*14\*32 | 0 |
| *dropout\_3* | 14\*14\*32 | 0 |
| *flatten* | 6272 | 0 |
| *dense* | 6272\*512+  512 | 321177  6 |
| *Dense output* | 512+1\*3 | 1539 |
| *Total* | | 3.312.0  99 |

Tabel diatas merupakan model yang terbentuk dari hasil *training*. Untuk menghitung *input* kedalam konvo digunakan rumus “*input*\_*size* + 2\**padding*

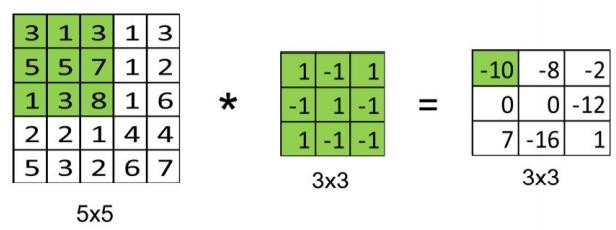
- (*filter*\_*size* -1)”. Total parameter yang terbentuk dari model sebanya 3.312.099 *neuron*.

* + 1. Proses *Convolution Layer*

Berdasarkan hasil dari arsitektur jaringan mengenai proses *convolution* dapat ditunjukkan pada Gambar 13.

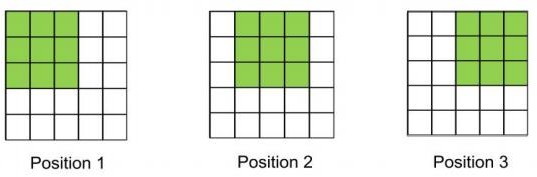
Gambar 13**.** Proses *Convolution Convolution* merupakan proses

mengkombinasi dua buah deret angka yang menghasilkan deret angka yang ketiga. Jika di implementasikan angka pada *convolution* ini adalah berbentuk *matriks array*. Pada *input* citra memiliki ukuran piksel 128x128x3, ini menunjukan bahwa tinggi dan lebar piksel dari gambar sebesar 128 dan citra tersebut memiliki 3 *channel* yaitu *red, green*, dan *blue* atau yang biasa disebut dengan RGB. Setiap *channel* piksel memiliki nilai *matriks* yang berbeda-beda. *Input* akan di *convolution* dengan nilai *filter* yang sudah ditentukan. *Filter* merupakan blok lain atau kubus dengan tinggi dan lebar yang lebih kecil namun kedalaman yang sama yang tersapu di atas citra dasar atau citra asli. *Filter* digunakan untuk menentukan pola apa yang akan dideteksi yang selanjutnya di konvolusi atau dikalikan dengan nilai pada *matriks input*, nilai pada masing-masing kolom dan baris pada *matriks* sangat bergantung pada jenis pola yang akan dideteksi. Jumlah *filter* pada konvolusi ini sebanyak 128 piksel dengan ukuran kernel (3x3), ini artinya gambar yang dihasilkan dari hasil *convolution* akan sebanyak 128 fitur map. Pada penelitian ini digunakan sampel *matriks* pada *input* citra agar lebih mudah dipahami pada proses *convolution.* Karena *input image* memiliki ukuran piksel 128x128, maka pada penelitian ini hanya diambil sebagian nilai *matriks* saja yang dijadikan sampel dalam proses *convolution*.



Gambar 14. Perhitungan *Convolution*

Pada Gambar 14 menunjukan proses *convolution* dengan menggunakan ukuran kernel 3x3, dengan menggunakan *stride* 1. *Stride* adalah jumlah pergeseran *kernel* terhadap *matriks input* berjumlah satu. Hasil *stride* ditunjukkan pada Gambar 15 sebagai berikut:



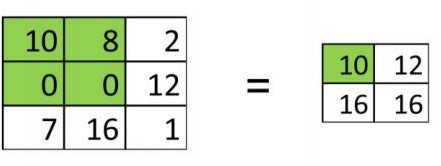
Gambar 15. Posisi *Kernel*

Pada Gambar 25 menunjukan perhitungan *dot product* pada proses *convolution* dimana sebuah *kernel* ukuran 3x3 yang dimulai pada sisi bagian kiri. Proses ini disebut dengan *sliding window*. Namun pada penelitian ini diberikan nilai padding 1, yaitu adanya penambahan nilai

0 disekeliling nilai *matriks input* supaya *input* dan *output* memiliki nilai *matriks* yang sama, sehingga tidak mengurangi informasiinformasi pada gambar. Proses ini dilakukan dari ujung kiri atas sampai ujung kiri bawah. Syarat agar dua buah *matriks* dapat dikalikan adalah *matriks* pertama harus memiliki jumlah kolom yang sama dengan jumlah baris pada *matriks* kedua. Ordo *matriks* hasil perkalian dua buah *matriks* adalah jumlah baris pertama dikali jumlah kolom ke dua. Kemudian sebelum di lanjutkan ke proses *pooling layer*, untuk menghilangkan nilai *negative* pada hasil, pada arsitektur jaringan digunakan aktivasi ReLU (*Rectified Linear Unit*) setelah proses *convolution*. Fungsi dari aktivasi ini adalah melakukan *threshold* atau memberikan nilai batas dari 0 hingga tak terhingga. Nilai yang ada pada hasil *convolution* yang bernilai *negative* akan diubah dengan aktivasi ini menjadi nol dan yang lainnya sampai tak terhingga.

* + 1. Proses Pooling

*Pooling* merupakan pengurangan ukuran *matriks* dengan menggunakan operasi *pooling* (penggabungan). Metode yang digunakan dalam proses *pooling* ini menggunakan *max*-*pooling*. *Max*-*pooling* merupakan salah satu metode umum yang biasa digunakan oleh peneliti yang berkaitan dengan penelitian *deep learning*. Dalam penelitian yang dilakukan oleh Dominik Scherer dkk (Scherer, 2010) menunjukan bahwa penggunaan metode *max*-*pooling* lebih unggul dibanding dengan metode *sub-sampling*. penggunaan metode ini menjadi salah satu metode terbaik dalam proses *pooling*. Berikut ini gambaran dari proses *pooling*:

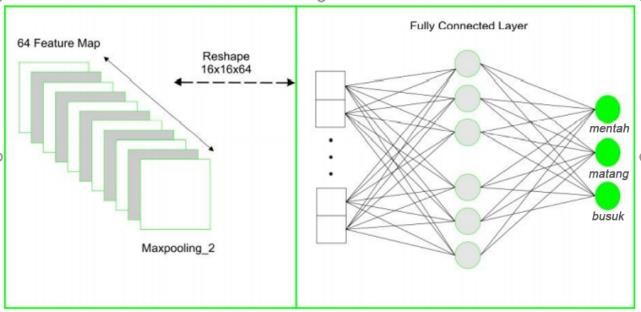


Gambar 16. Proses *Pooling*

Proses *pooling* ini menggunakan ukuran 2x2 dengan *stride* 1 dimana jumlah pergeseran *kernel* terhadap *matriks input* berjumlah satu. Dalam proses *pooling* ini digunakan metode *max- pooling*, dimana *window* akan bergeser sesuai dengan ukuran dan stridennya untuk mendapatkan nilai paling maksimum. Terlihat pada Gambar 16 *output* dari proses ini memiliki nilai yang paling maksimum yang di ambil dari *matriks* fitur map hasil *convolution*. Hasil *max-pooling* tersebut berukuran 2x2.

* + 1. Proses Fully Connected

Selanjutnya adalah *Fully connected Layer*. Proses ini bertujuan untuk untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.

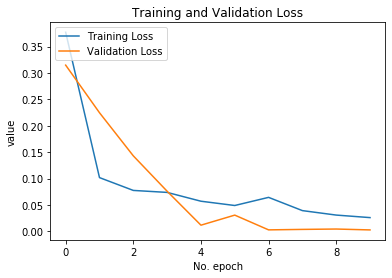


Gambar 17. Proses *Fully Connected*

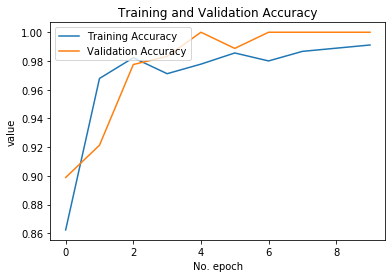
Gambar 17 merupakan proses *converting* hasil dari fitur map *max*-*pooling* menjadi *flatten* atau *vector*, *activation map* yang dihasilkan dari *feature extraction* layer masih berbentuk *multidimensional array*, harus melakukan *reshape activation map* menjadi sebuah vektor agar bisa digunakan sebagai *input* dari *fully- connected layer*. Proses ini sama dengan proses MLP (Multilayer Perceptron). Jaringan ini umumnya menggunakan lapisan yang terhubung sepenuhnya di mana setiap piksel dianggap sebagai *neuron* terpisah. Dalam proses ini biasanya diterapkan metode “*dropout*”. Metode ini bertujuan untuk menonaktifkan beberapa *edge* yang terhubung ke setiap *neuron* untuk menghindari *overfitting*. Setelah itu proses terakhir adalah klasifikasi. Dalam proses ini digunakan aktivasi fungsi *softmax*. Aktivasi ini akan membantu MLP untuk mengklasifikasikan *input* terhadap targetnya, yaitu kedalam 3 kelas, yaitu Mentah, Matang dan Busuk. Setiap *neuron* pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully- connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak *reversibel*, sedangkan *fully-connected layer* hanya dapat diimplementasikan diakhir jaringan.

* + 1. Hasil *Training*

Setelah melalui beberapa proses dalam algoritma *Convolutional Neural Network* (CNN) didapatkan hasil *training* dan *validation*. Proses *train* akan menghasilkan sebuah model yang nantinya akan digunakan dalam proses pengujian, model yang disimpan berisi hasil training yang dilakukan oleh sistem. Secara garis besar isi dari proses *train* model yaitu ditunjukkan pada Tabel 3. berikut grafik hasil *training* menggunakan *tensorflow.keras*:



Gambar 18. Grafik *Training Loss*



Gambar 19. Grafik *Training Accuracy* Berdasarkan Gambar 18 dan 19

terdapat 4 indikator, yaitu *Training Loss*

dan *Accuracy* serta *Validation Loss* dan *Validation Accuracy*. *Training Loss* merupakan hasil *train* yang data nya tidak terbaca, semakin kecil nilai *loss* maka akan semakin baik, sedangkan *Training Accuracy* merupakan nilai akurasi yang di dapat dari proses *training*, semakin tinggi nilai akurasi maka semakin baik model yang telah dibuat. *Validation Loss*

merupakan nilai pembanding dari proses *train*, semakin rendah nilai *validation loss* maka semakin baik modelnya, sedangkan *Validation Accuracy* merupakan nilai pembanding yang diperoleh dari proses *train model*, semakin tinggi nilai akurasi validasi maka semakin baik model yang telah dibuat. Dalam proses *train* kali ini didapat *accuracy* dari *training* model mencapai 0.9911 dengan nilai loss sebesar 0.0261. Proses *training* disini menggunakan *learning rate* 0.001 dengan *input* gambar sebesar 64 x 64 piksel. Waktu pelatihan yang dibutuhkan untuk 10 epoch dalam menjalankan *training* model ini yaitu 30 menit. Semakin Banyak *epoch* maka semakin lama juga waktu yang dibutuhkan untuk *training* model. Kemudian *accuracy* dari data *validation* mencapai 1.000 dengan nilai *loss* sebesar 0.0028.

* 1. Hasil Pengujian Model

Proses *testing* menggunakan data uji sebanyak 200, untuk setiap kelas jenis sawit sebanyak 66 citra. Hasil *confusion matriks* adalah sebagai berikut:

Tabel 4. *Confusion Matriks*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Matriks* | | Prediksi Kelas | | |
| Ment ah | Mata ng | Bus uk |
| Kelas sebenar nya | Ment  ah | 60 | 0 | 6 |
| Mata ng | 1 | 62 | 4 |
| Busu k | 6 | 5 | 55 |

Berdasarkan Tabel 4 hasil hasil prediksi dari model terhadap data *testing* data baru menunjukan hasil yang baik, yaitu prediksi terhadap buah sawit mentah di klasifikasikan benar sebanyak 60 dan *missing* data dari *input* sawit mentah di klasifikasikan sebagai sawit busuk sebanyak 6 data. Prediksi kedua pada Buah sawit matang diklasifikasikan benar sebanyak 62 dan *missing* data dari *input* sawit matang diklasifikasikan sebagai sawit busuk sebanyak 4 data dan sawit mentah sebanyak 1 data. Kemudian yang terakhir adalah prediksi pada buah sawit busuk diklasifikasikan benar sebagai sawit busuk sebanyak 55 dan *missing* data dari *input* sawit busuk diklasifikasikan sebagai sawit mentah sebanyak 6 data dan sawit matang sebanyak 5 data. Perhitungan akurasi dari keseluruhan *matriks* diatas adalah sebagai berikut:

𝑇𝑃/𝑇𝑁

𝐴𝑘𝑢𝑟𝑎𝑠𝑖 =

𝑇𝑜𝑡𝑎𝑙 𝐺𝑎𝑚𝑏𝑎𝑟 𝑌𝑎𝑛𝑔 𝑑𝑖 𝑈𝑗𝑖

177

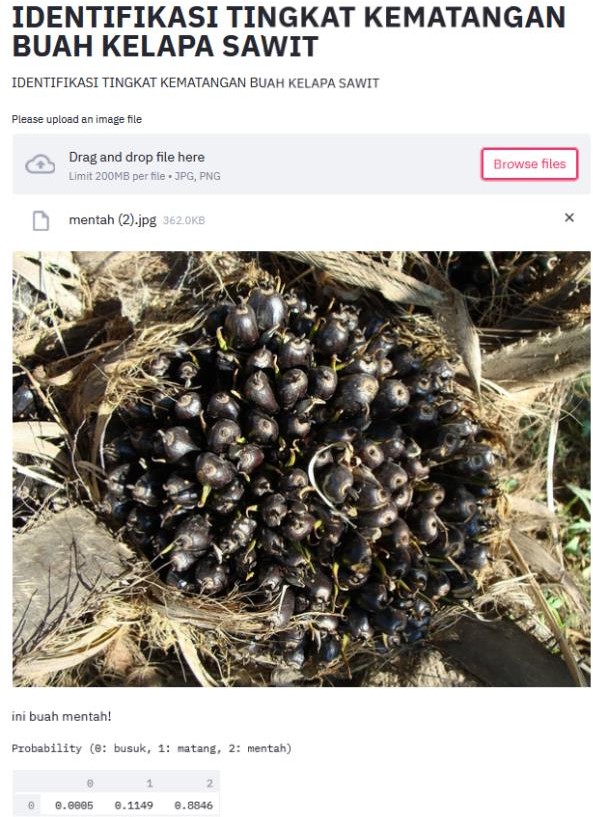
𝐴𝑘𝑢𝑟𝑎𝑠𝑖 = = 88%

200

Jadi akurasi yang dihasilkan oleh model dengan *input* gambar 64x64 piksel dan jumlah sampel *testing* 60 data didapatkan nilai akurasi sebsesar 88%.

* 1. Hasil Klasifikasi Gambar

Pengujian akan dilakukan dengan menggunakan data uji sebanyak 6 gambar yang berbeda. Gambar di bedakan menjadi tiga kelas, yaitu: mentah, matang dan busuk. Pengujian dilakukan menggunakan *website*. Pengujian dilakukan menggunakan *website* yang bertujuan untuk memudahkan dalam proses identifikasi tingkat kematangan buah kelapa sawit. *Website* tersebut dibangun menggunakan bahasa pemograman python dengan bantuan *library streamlit* yang merupakan salah satu *framework website* dari *python* yang masih baru tetapi cukup popular dikalangan *data science* karena simpel dan mudah dipelajari. Hasil yang ditampilkan dalam *website* ini menggunakan model klasifikasi yang telah dibuat sebelumnya. Berikut merupakan tampilan *website* yang telah dibuat:



Gambar 19. Tampilan GUI Berikut merupakan tabel hasil

pengujian gambar:

Tabel 5. Hasil Klasifikasi

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | 0.97 | Matang |
| 2 |  | 0.88 | Mentah |
| 3 |  | 0.73 | Busuk |
| 4 |  | 0.97 | Matang |
| 5 |  | 0.83 | Mentah |
| 6 |  | 0.50 | Busuk |

Dapat dilihat pada Tabel 5 hasil pengujian dengan gambar yang telah dipilih mendapat hasil yang cukup baik ang mana buah kelapa sawit dengan kelas matang mendapat hasil akurasi sebesar 0.9752. Dalam percobaan tidak ada *missing* data dari 6 citra yang di uji. Namun terdapat beberapa pengujian yang mendapat akurasi cukup rendah, yaitu pada pengujian citra buah busuk dengan akurasi sebesar 0.5051, hal tersebut disebabkan karena buah sawit busuk yang diuji memiliki *background* dan warna sedikit berbeda dengan koleksi dataset yang dimiliki, solusi dari masalah tersebut yaitu dengan menambah koleksi data dari masing-masing kelas secara detail dengan ciri-ciri dari kelas tersebut terlebih terhadap data yang buah sawit mentah dan busuk. Proses penghitungan akurasi merupakan proses akhir pada penelitian ini. Akurasi dalam penelitian ini merupakan variabel yang m erepresentasikan kinerja yang digunakan untuk menilai tolak ukur keberhasilan model CNN dalam mengidentifikasi tingkat kematangan buah kelapa sawit. Persamaan yang digunakan untuk menghitung akurasi ditunjukkan pada persamaan berikut ini:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Kategori | Jumlah Data Uji | Salah | Benar |
| 1 | Matang | 2 | 0 | 1 |
| 2 | 0 | 1 |
| 3 | Mentah | 2 | 0 | 1 |
| 4 | 0 | 1 |
| 5 | Busuk | 2 | 0 | 1 |
| 6 | 0 | 1 |
| Total | |  | - | 6 |

Pada Tabel 6 hasil pengujian terhadap dataset yang diuji menunjukan hasil yang baik. Proses perhitungan akurasi dapat ditunjukkan pada perhitungan berikut ini:

6

𝐴𝑘𝑢𝑟𝑎𝑠𝑖 = 𝑥 100% = 100%

6

Akurasi yang didapatkan dari proses pengujian yang menggunakan 6 data uji menunjukkan nilai akurasi 100%.

# 5. KESIMPULAN DAN SARAN

* 1. Kesimpulan

Adapun kesimpulan yang diberikan pada penelitian ini sebagai berikut:

1. Dari hasil penelitian dan pembahasan didapatkan hasil bahwa identifikasi tingkat kematangan buah kelapa sawit menggunakan algoritma *Convolutional Neural Network* (CNN) mendapatkan nilai akurasi yang tinggi yaitu sebesar 99% pada proses *training* dan 88% pada proses *testing*.
2. Kemudian penelitian ini menggunakan data baru acak untuk menguji model yang dibuat dan menghasilkan nilai akurasi yang baik dalam mengidentifikasi buah kelapa sawit. Sehingga dapat disimpulkan bahwa model yang telah dibuat dengan mengimplementasikan metode *deep learning* menggunakan algoritma *Convolutional Neural Network* (CNN) mampu melakukan identifikasi dan klasifikasi buah kelapa sawit.
   1. Saran

Adapun saran yang diberikan pada penelitian ini sebagai berikut:

1. Penelitian selanjutnya diharapkan dapat menambah jumlah dataset buah kelapa sawit sesuai kelas yang telah ditentukan.
2. Menambahkan parameter seperti perbandingan *input* gambar yang lebih besar ukuran pikselnya, variasi jumlah *epoch* dan *learning rate*, serta penggunaan *optimizer* yang lainnya. Sehingga dengan menghasilkan model dengan penggunaan hyperparameter terbaik.
3. Penelitian ini dapat dikembangkan kedalam robotika atau alat yang dapat mengenali gambar tersebut dan dapat di implementasikan ke perusahaan atau petani.

# 5. DAFTAR PUSTAKA

Badan Pusat Statistik Nasional, 2020. Statistik Perkebunan Kelapa Sawit Tahun 2020. Jakarta: Badan Pusat Statistik.

Danukusumo, K.P. (2017). Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Candi Berbasis Gpu.

Deng, L. and Yu, D. (2014) Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing, 7, 197-387.

Direktorat Jenderal Perkebunan. 2020. Statistik Perkebunan Indonesia Kelapa Sawit Indonesia 2018-2020. Direktorat Jenderal Perkebunan. Jakarta.

Dzulqarnain, M.F, Suprapto and Makhrus,

F. 2019. Improvement of Convolutional Neural Network Accuracy on Salak Classification Based Quality on Digital Image. Indonesian Journal of Computing and Cybernetics Systems Vol.13, No.2, April 2019, pp. 189~198 ISSN (print): 1978-1520, ISSN (online): 2460-7258 DOI: 10.22146/ijccs.42036.

Fawcett, Tom. (2006). Introduction to ROC analysis. Pattern Recognition Letters. 27. 861-874.

10.1016/j.patrec.2005.10.010.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Han, J. and Kamber, M. (2006) Data Mining: Concepts and Techniques. 2nd Edition, Morgan Kaufmann Publishers, San Francisco.

Khan, S., Rahmani, H., Ali Shah, S. A., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. Synthesis Lectures on Computer Vision, 8(1).

Krizhevsky, A., Sutskever, I. dan Hinton, G.E., 2012, ImageNet Classification with Deep Convolutional Neural Networks, Advances In Neural Information Processing Systems, pp.1–9.

Kusumadewi, S., 2004, Membangun Jaringan Syaraf Tiruan, Yogyakarta: Graha

Ilmu.

Labach, Alex & Salehinejad, Hojjat & Valaee, Shahrokh. (2019). Survey of Dropout Methods for Deep Neural Networks.

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, 253– 256.

https://doi.org/10.1109/ISCAS.201 0.5537907.

Lau, M.M., Lim, K.H. dan Gopalai, A.A., 2015. Malaysia Traffic Sign Recognition with Convolutional Neural Network. 2015 IEEE International Conference on Digital Signal Processing (DSP), IEEE (978-1- 4799-8058-1/15), 1006–1010.

Maulana, F. F., & Rochmawati, N. (2019). Klasifikasi Citra Buah Menggunakan Convolutional Neural Network. *Journal of Informatics and Computer Science*, *01*, 104–108. jurnalmahasiswa.unesa.ac.id › article.

Minarni, Salumbae, R. & Hasbi, Z. (2018). *Komunikasi Fisika Indonesia*. *15*(01), 36–45.

Nugroho, A. (2019). *Buku Teknologi Agroindustri Kelapa Sawit* (Issue August).

Pahan, I. (2015). *Panduan Teknis Budidaya Kelapa Sawit Untuk Praktisi Perkebunan*, Jakarta:Penebar Swadaya.

Prasetyo, E., 2012. Data Mining Konsep dan Aplikasi Menggunakan Matlab, Andi Offset, Yogyakarta.

Prathivi, R. (2019). *Optimasi Model TL-CNN Untuk Klasifikasi Citra CIFAR-10*. *1*(10), 3–7.

Putra, I. S., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101. JURNAL TEKNIK ITS, 5(1).

Rangkuti, I.U.P. dan A. Syahputra. (2019). Warna Minyak Sawit Mentah dan Stabilitas Warna Berdasarkan Tingkat Kematangan Buah yang Berasal dari Kebun dengan ketinggian 800 mdpl. Jurnal Agro Fabrica vol. 1 (2): 32-37.

Rismiyati, & Azhari, S. N. (2017). Convolutional Neural Network implementation for image-based Salak sortation. *Proceedings - 2016 2nd International Conference on Science and Technology-Computer, ICST 2016*, 77–82.

https://doi.org/10.1109/ICSTC.201 6.7877351.

Sari, N., Shiddiq, M., Fitra, R. H., & Yasmin, N. Z. (2019). Ripeness Classification of Oil Palm Fresh Fruit Bunch Using an Optical Probe. *Journal of Aceh Physics Society*, *8*(3), 72–77.

https://doi.org/10.24815/jacps.v8i3

.14122.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15(56), 1929-

1958.

Zhu, Qiuyu & He, Zikuang & Zhang, Tao & Cui, Wennan. (2020). Improving Classification Performance of Softmax Loss Function Based on Scalable Batch-Normalization. Applied

Sciences. 10. 2950.

10.3390/app10082950.