

Pengenalan Jenis Kayu Berdasarkan Citra Makroskopik Menggunakan Metode *Convolutional Neural Network*

<http://dx.doi.org/10.28932/jutisi.v7i2.3706>

Riwayat Artikel

Received: 18 Juni 2021 | Final Revision: 22 Juli 2021 | Accepted: 27 Juli 2021

Eko Yudhi Prastowo✉ #1

¹me@ekoyudhi.id

#Kementerian Keuangan
Jl Gatot Subroto No 40-42 Jakarta

Abstract— Until now, wood has an irreplaceable function. Building materials, shipping, furniture, sports equipment, carvings and handicrafts using wood. Indonesia has more than 4,000 types of wood, so choosing the right wood is a challenge because choosing the wrong type of wood can make the quality of processed products decline and not as expected. In addition, proper identification of timber can also prevent illegal logging, especially on certain types of wood which are now increasingly scarce. Recognition to wood by looking directly is a difficult thing for ordinary people to do and can only be done by a wood expert, so it is necessary to find a method of recognizing wood that can be used by people independently. One method that can be used to identify type of wood is image processing based on characteristics of wood which include color, fiber direction and texture. This paper will describe recognition of wood-based image processing using Convolutional Neural Network (CNN) method. This method is derived from Neural Networks with addition of an extraction layer feature, which can reduce free parameters that are not needed by the system. Wood image data used in this study are four types of wood that are often used as raw materials for making houses and furniture, namely Glugu, Teak, Sengon and Waru. Results of this study were able to recognize four types of wood with an accuracy of 95% in 600 epochs/iteration, so that it can be used as a simple, easy and inexpensive wood recognition system.

Keywords— CNN; computer vision; image processing; wood classification; wood recognition.

I. PENDAHULUAN

Pada era sekarang ini, kayu merupakan bahan bangunan yang belum tergantikan sepenuhnya dan masih menjadi primadona dalam sebuah konstruksi. Kayu masih menjadi salah satu bahan baku utama untuk membangun rumah selain baja ringan dan beton. Kayu sebagai bahan baku juga mendominasi industri manufaktur yang menghasilkan perabot rumah tangga seperti lemari, pintu, jendela, dan kursi. Kualitas kayu berkaitan dengan harga dan kualitas produk akhir. Sehingga penting untuk mengenali jenis kayu yang akan digunakan karena akan menjamin tingkat kualitas dan keawetannya. Identifikasi kayu yang akurat juga diperlukan untuk mencegah pembalakan liar, terutama pada jenis kayu tertentu yang sekarang sudah sangat langka.

Ciri-ciri daun, buah, dan kayu merupakan ciri khas utama yang dapat digunakan untuk menggambarkan jenis pohon [1]. Fitur kayu dapat dibagi menjadi dua kategori, fitur makroskopik dan mikroskopis. Fitur makroskopik menentukan karakteristik kayu berdasarkan warna, tekstur, dan pola. Ciri-ciri tersebut dapat dilihat langsung oleh ahli kayu menggunakan kaca pembesar 10-15x [2][3]. Namun hasil analisis fitur-fitur tersebut bisa saja berbeda berdasarkan pengalaman masing-masing pakar. Ciri-ciri mikroskopis yang sebagian mengandung struktur dan pori-pori bahan kayu dapat diperoleh dari mikroskop cahaya dengan ukuran 100-500x [4]. Karena dibutuhkan alat khusus untuk mengolah bahan kayu, maka ekstraksi dan analisis ciri mikroskopis ini memakan waktu lebih lama [5].

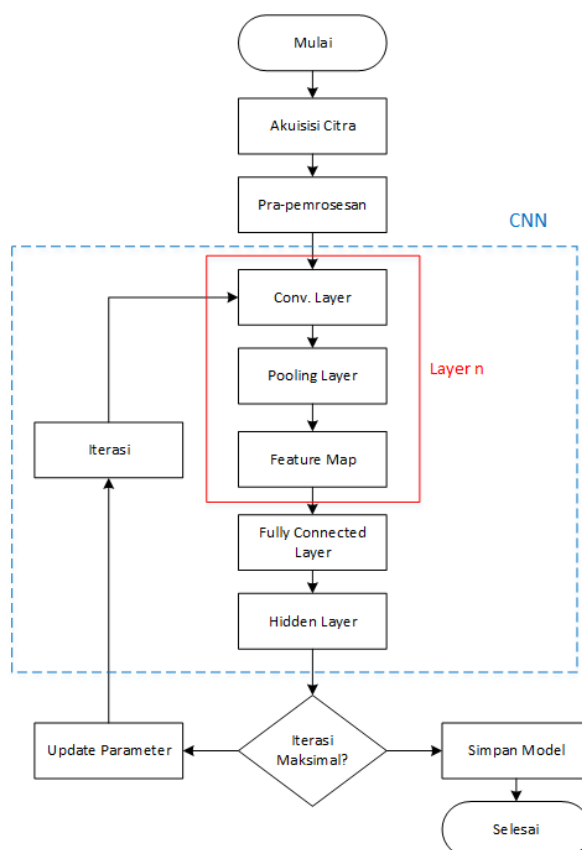
Pengenalan jenis kayu menggunakan fitur citra makroskopik lebih mudah diterapkan, sederhana dan fleksibel [6]. Tantangan utama fitur citra makroskopik adalah kemiripan tekstur antara satu jenis kayu dengan jenis kayu lainnya [7]. Telah banyak penelitian untuk mengidentifikasi jenis kayu menggunakan citra makroskopis dan mikroskopis. Panagiotis dkk melakukan identifikasi 12 jenis kayu menggunakan ekstraksi ciri GLCM dan SVM *Classifier* [2]. Pendekatan mereka mampu mencapai akurasi 91,47%. Sugiarto dkk melakukan pengenalan jenis kayu dari citra makroskopik menggunakan *Histogram*

of *Oriented Gradient* (HOG) dan *SVM Classifier* [7]. Akurasi adalah 70,5% untuk pengujian citra positif dan 77,5% untuk pengujian citra negatif. Yulianti dkk mengembangkan sistem portabel untuk mengidentifikasi jenis kayu berdasarkan pori-pori dan kurva konsentris [8]. Dengan menggunakan *multichannel Gabor Filter* dan *back propagation JST* diperoleh hasil identifikasi diatas 95%. Sedangkan Gunawan dkk menggunakan fitur ekstraksi 2D-PCA dan *SVM Classifier* untuk mengklasifikasikan empat jenis kayu di Indonesia [9]. Hasil akurasi terbaik adalah 95,83% menggunakan *kernel* polinomial. Yusof dan Rosli menggunakan teknik *Filter Gabor* dan metode ekstraksi karakteristik *GLCM* [10], sedangkan untuk klasifikasi menggunakan jaringan syaraf *multilayer* berbasis *popular back-propagation* (MLBP). Akurasi tertinggi yang dihasilkan dari metode tersebut adalah 93,56%.

Dalam makalah ini, penelitian dilakukan untuk mengidentifikasi jenis kayu berdasarkan citra makroskopik kayu menggunakan *Convolutional Neural Network* (CNN). Ini adalah metode terancang yang berasal dari Jaringan Syaraf Tiruan yang memiliki banyak lapisan atau disebut *Multi-Layer Perceptron* (MLP)[11]. Keunggulan CNN dibandingkan varian MLP lainnya adalah mampu mereduksi parameter bebas yang dapat menyebabkan *overfitting* dan mampu memproses citra secara geometris seperti translasi, rotasi dan penskalaan [12]. Adapun kayu yang digunakan dalam penelitian ini ada empat kayu yang sering digunakan untuk bahan baku rumah dan furnitur, yaitu Glugu, Jati, Sengon dan Waru. Hipotesis dari penelitian ini adalah dengan melakukan metode CNN pada citra makroskopik kayu dapat mengidentifikasi jenis kayu dengan akurasi yang lebih baik daripada *machine learning* biasa.

II. METODE PENELITIAN

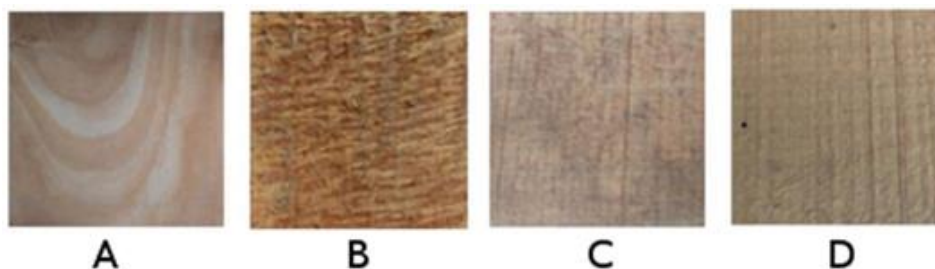
Secara umum alur penelitian diilustrasikan pada Gambar 1. Kali pertama dilakukan adalah proses akuisisi data citra menggunakan kamera *smartphone*. Selanjutnya adalah tahap pra-pemrosesan yaitu dengan cara melakukan normalisasi terhadap semua data citra yang telah diakuisisi. Proses selanjutnya adalah membuat dan mendefinisikan arsitektur CNN. Setelah itu masuk ke dalam tahap pelatihan menggunakan data citra yang telah mengalami pra-pemrosesan dengan sejumlah iterasi tertentu. Pada tahap ini saat melakukan pelatihan terhadap data citra, juga dilakukan validasi model untuk menentukan *loss* dan akurasi model pada setiap iterasi. Proses terakhir adalah melakukan penyimpanan model dan melakukan pengujian terhadap data citra yang sama sekali baru dan tidak diikuti dalam proses pelatihan. Proses ini dilakukan untuk menguji akurasi terhadap model yang telah dihasilkan pada proses pelatihan.



Gambar 1. Alur Penelitian

A. Akuisisi Citra dan Pra-pemrosesan

Dalam proses akuisisi data citra, perangkat yang digunakan berupa kamera *smartphone* untuk mengambil citra dari empat jenis kayu yang telah ditentukan sebelumnya, yaitu Glugu, Jati, Sengon, dan Waru. Contoh untuk masing-masing citra dapat dilihat pada Gambar 2.



Gambar 2. A. Jati, B. Glugu, C. Sengon, D. Waru

Penelitian ini tidak menentukan jumlah pasti citra untuk setiap jenis kayu. Pencahayaan tidak dikondisikan, sehingga citra diambil dalam lingkungan alami. Dari keseluruhan data citra dibagi menjadi data pelatihan, validasi, dan pengujian. Detail masing-masing data dapat dilihat pada Tabel 1.

TABEL 1
TOTAL DATA PELATIHAN, VALIDASI, DAN PENGUJIAN

No	Tipe Kayu	Pelatihan	Validasi	Pengujian	Total
1	Glugu	234	66	33	333
2	Jati	187	53	26	266
3	Sengon	187	53	26	266
4	Waru	125	35	15	175
	Total	733	207	100	1040

Dalam pra-pemrosesan, seluruh citra yang diperoleh pada proses akuisisi dilakukan normalisasi dengan mengubah ukuran semua citra menjadi 224x224 piksel.

B. Convolutional Neural Network (CNN)

CNN merupakan turunan dari Jaringan Syaraf Tiruan yang biasa digunakan untuk mengolah data berdimensi tinggi, seperti citra dan video. Kelahiran CNN diawali dengan model Neocognitron yang dikemukakan oleh Kunihiko Fukushima [12]. Model ini telah mengimplementasikan beberapa *layer* yang secara otomatis mempelajari ekstraksi fitur untuk pengenalan pola. CNN merupakan model yang dapat digunakan baik untuk *supervised learning* maupun *unsupervised learning* [13].

Secara garis besar CNN tidak jauh berbeda dengan *Artificial Neural Network* karena keduanya memiliki bobot, bias dan fungsi aktivasi. Hanya saja, CNN dilengkapi dengan fitur *layer* ekstraksi. Fitur ini melakukan semacam pengkodean suatu citra menjadi fitur berupa angka-angka yang mewakili citra tersebut. Ekstraksi *layer* ini terdiri dari dua bagian, yaitu: *convolutional layer* dan *pooling layer*.

1. *Convolutional Layer*, merupakan komponen terpenting dari CNN yang berhubungan langsung dengan input. *Layer* ini terdiri dari sekumpulan filter yang berfungsi untuk mengekstrak fitur dari input yang dimasukkan. Filter didefinisikan dalam hal berat dan tinggi. Misalnya *layer* pertama disebut *convolutional layer 5x5x3*, artinya lebar 5 piksel, tinggi 5 piksel dan 3 sesuai kanal citra. Ketiga filter/kanal di atas digeser ke seluruh bagian citra. Setiap *shift* dilakukan operasi “titik” antara input dan nilai filter untuk menghasilkan output atau biasa disebut dengan *activation map* atau *feature map*. Ada beberapa parameter yang sangat mempengaruhi ekstraksi fitur pada *layer* ini, yaitu:
 - a. *Stride*, merupakan parameter yang menentukan berapa banyak pergeseran filter. Jika nilai *stride* adalah 1, maka *convolutional* filter bergeser sebanyak 1 piksel secara horizontal kemudian vertikal. Semakin kecil langkahnya, semakin detail informasi yang didapatkan dari sebuah input, tetapi membutuhkan perhitungan yang lebih lama. Perlu dicatat, bahwa langkah kecil tidak menjamin kinerja yang lebih baik.
 - b. *Padding*, merupakan parameter yang menentukan jumlah piksel yang ditambahkan pada setiap sisi input. Ini digunakan untuk tujuan memanipulasi dimensi peta fitur. Dengan *padding*, dimensi output dapat diatur agar tetap sama dengan dimensi input atau setidaknya tidak berkurang drastis. Jadi lapisan *convolutional* dapat digunakan lebih dalam sehingga

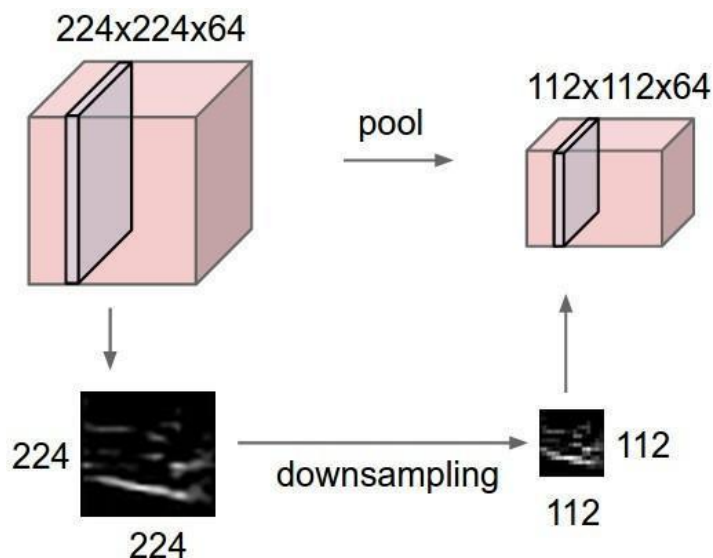
lebih banyak fitur yang berhasil diekstraksi. Ukuran *feature map/output layer* pada CNN secara umum dapat dihitung berdasarkan *padding* dan *stride* dengan formula 1 dan 2 di bawah ini:

$$\text{output width} = \frac{W - F_w - 2P}{S_w} + 1 \quad (1)$$

$$\text{output height} = \frac{W - F_h - 2P}{S_h} + 1 \quad (2)$$

W = width input H = height input
 F_w = filter width F_h = filter height
 S_w = stride width S_h = Stride height
 P = padding

2. *Pooling/subsampling layer*, pada prinsipnya *pooling layer* terdiri dari filter dengan ukuran dan *stride* tertentu yang akan memperbesar seluruh area *feature map*. Jenis *pooling* yang biasa digunakan adalah *max pooling* dan *average pooling*. Penyatuan maksimum pada setiap *shift* filter akan memilih nilai terbesar, sedangkan penyatuan rata-rata akan memilih nilai rata-rata. Tujuan utama dari *pooling layer* adalah untuk memperkecil ukuran *feature map* (*downsampling*) sehingga dapat mempercepat proses komputasi dan mengontrol terjadinya *overfitting* [14], hal ini dijelaskan sesuai Gambar 3 .



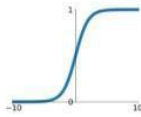
Gambar 3. Cara Kerja *Pooling Layer*[15]

Komponen lain yang penting untuk dipertimbangkan dalam pemodelan *Neural Network* dan turunannya adalah fungsi aktivasi. Tugas fungsi aktivasi adalah untuk membuat jaringan syaraf non-linear. Ada banyak jenis fungsi aktivasi seperti pada Gambar 4, diantaranya yang populer saat ini adalah *Sigmoid*, *Tanh* dan *ReLU* [16]. *ReLU* adalah fungsi aktivasi yang paling populer untuk CNN. Inti dari cara kerja *ReLU* adalah untuk membuat pembagi pada angka nol, jadi jika $x \leq 0$ maka $x = 0$, dan jika $x > 0$ maka $x = x$. Aktivasi *ReLU* memiliki kelebihan dan kekurangan. Kelebihan dari fungsi aktivasi *ReLU* adalah lebih mempercepat proses konvergensi dilakukan dengan *gradient descent stochastic* bila dibandingkan dengan *sigmoid / tanh*. Selain itu, jika dibandingkan dengan *sigmoid / tanh* yang memiliki operasi mahal (eksponensial, dll), *ReLU* dapat diimplementasikan dengan hanya menciptakan ambang batas pada nol. Kelemahannya adalah bahwa unit *ReLU* menjadi rentan selama proses pelatihan dan dapat membuat unit mati. Sebagai contoh, ditemukan bahwa 40% dari jaringan sudah mati (neuron yang tidak akan aktif selama proses pelatihan) jika *learning rate* diinisialisasi terlalu tinggi. Tetapi jika dapat menginisialisasi *learning rate* secara tepat, maka ini jarang menjadi masalah.

Activation Functions

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



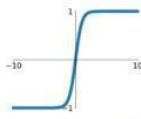
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

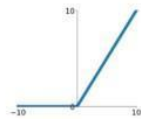


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

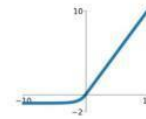
ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



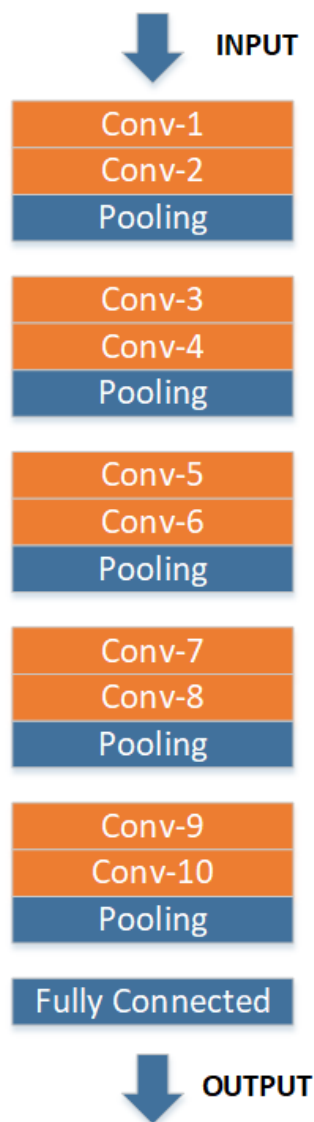
Gambar 4. Fungsi Aktivasi pada *Neural Network*[17]

Dalam pembelajaran mesin, regularisasi adalah cara untuk mencegah *overfitting*. Regularisasi mengurangi *overfitting* dengan menambahkan penalti ke *loss function*. Dengan menambahkan penalti ini, model dilatih sedemikian rupa sehingga tidak mempelajari set bobot fitur yang saling bergantung. Salah satu metode yang terkenal ini disebut *dropout*. *Dropout* adalah pendekatan untuk regularisasi dalam jaringan saraf yang membantu mengurangi pembelajaran yang saling bergantung di antara neuron.

Penelitian ini mengadopsi model arsitektur CNN VGG-16 [18] dengan beberapa penyesuaian agar komputasi menjadi lebih ringan. Penyesuaian antara lain menurunkan ukuran filter pada *convolutional layer* pertama dan kedua dari 64 ke 8, menurunkan ukuran filter pada *convolutional layer* ketiga dan keempat dari 128 ke 16, menurunkan ukuran filter pada *convolutional layer* kelima dan keenam dari 256 ke 32, menurunkan ukuran filter pada *convolutional layer* ketujuh dan kedelapan dari 512 ke 64, menurunkan ukuran filter pada *convolutional layer* kesembilan dan kesepuluh dari 512 ke 128 sebagaimana tertampil pada Gambar 5. Komposisi model dasar untuk setiap lapisan arsitektur CNN yang akan digunakan dengan input awal pada kanal 224x224x3 sebagaimana ditampikan pada Tabel 2.

TABEL 2
LAPISAN ARSITEKTUR CNN

Layer	Output
8x3x3 conv.layer, padding 'same', activation relu	224x224x8
8x3x3 conv. layer, padding 'same', activation relu	224x224x8
2x2 maxpool, stride 2x2	112x112x8
16x3x3 conv. layer, padding 'same', activation relu	112x112x16
16x3x3 conv. layer, padding 'same', activation relu	112x112x16
2x2 maxpool, stride 2x2	56x56x16
32x3x3 conv. layer, padding 'same', activation relu	56x56x32
32x3x3 conv. layer, padding 'same', activation relu	56x56x32
2x2 maxpool, stride 2x2	28x28x32
64x3x3 conv. layer, padding 'same', activation relu	28x28x64
64x3x3 conv. layer, padding 'same', activation relu	28x28x64
2x2 maxpool, stride 2x2	14x14x64
128x3x3 conv. layer, padding 'same', activation relu	14x14x128
128x3x3 conv. layer, padding 'same', activation relu	14x14x128
2x2 maxpool, stride 2x2	7x7x128
Fully-connected (MLP)/ Flatten	6272



Gambar 5. Arsitektur CNN

III. EKSPERIMEN DAN HASIL

Deskripsi hasil penelitian dibagi menjadi dua tahap, yaitu pelatihan untuk pemodelan menggunakan data pelatihan dan pengujian untuk prediksi menggunakan data pengujian. Pada penelitian ini, proses pelatihan menggunakan *google collaboratory* dengan spesifikasi GPU Tesla T4 16GB, RAM 27 GB, dan storage 68GB. Proses pengujian menggunakan komputer Intel Core i5-10210U @1.6GHz, RAM 8GB, storage 1TB. Penelitian ini juga menggunakan bahasa Python, pustaka Tensorflow dan Keras sebagai penunjang.

A. Pelatihan

Pada proses pelatihan dilakukan 600 kali *epoch*/iterasi terhadap data pelatihan sebanyak 733 citra. Setiap *epoch* membutuhkan waktu sekitar 6-10 detik. Setelah 600 kali *epoch*, *loss* data pelatihan menurun dari 4,3 menjadi 0,4, sebaliknya akurasi meningkat dari awalnya 27% menjadi 88%. Sekilas perkembangan *loss* dan akurasi dalam pelatihan dapat dilihat pada Tabel 3.

TABEL 3
DATA PELATIHAN

No	Epoch	Waktu	Loss	Akurasi
1	1	10s	4.2939	0.2708
2	25	9s	1.7544	0.3333
3	50	9s	1.1806	0.5833
4	75	8s	0.7707	0.6875
5	100	8s	1.0477	0.6458
6	125	9s	1.0116	0.6042
7	150	9s	0.8160	0.6875
8	175	9s	0.9218	0.6458
9	200	9s	0.6696	0.7500
10	225	8s	0.8329	0.7083
11	250	9s	0.5096	0.7292
12	275	9s	0.7248	0.7083
13	300	9s	0.9220	0.7083
14	325	9s	0.4998	0.7917
15	350	9s	0.7171	0.7292
16	375	9s	0.5447	0.7292
17	400	9s	0.7096	0.7083
18	425	9s	0.4335	0.8333
19	450	9s	0.4654	0.7500
20	500	8s	0.4278	0.8455
21	550	9s	0.4490	0.7708
22	600	9s	0.3963	0.8750

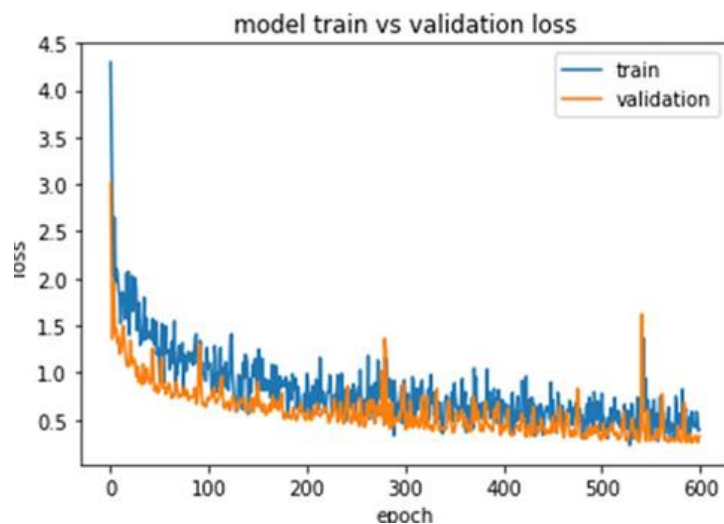
Perbandingan antara data pelatihan dan validasi juga menunjukkan hasil yang baik, tidak ada *overfitting* atau *underfitting*, baik pada *loss* maupun akurasi. *Loss* adalah sebuah fungsi untuk mengukur seberapa baik sebuah prediksi dengan cara mengukur jarak hasil output prediksi dengan nilai target. *Loss* menghasilkan nilai kesalahan atau eror antara output pelatihan dan target, pengukuran jarak biasanya menggunakan jarak *Euclidean* atau *Cross-Entropy*. Akurasi merupakan perbandingan antara data yang diprediksi benar sesuai kelas target dengan keseluruhan data oleh model klasifikasi. Penghitungan akurasi dapat terlihat pada formula 3.

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

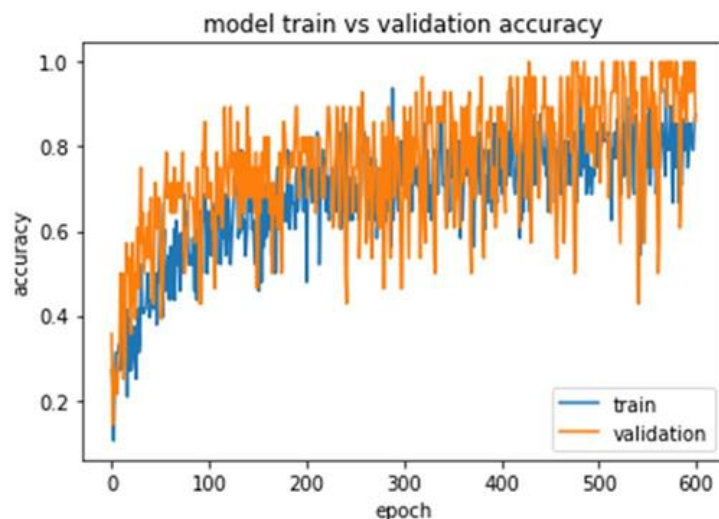
TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

Loss yang diperoleh setiap iterasi pada proses pelatihan cenderung semakin kecil. Pada iterasi ke-1 *loss* yang diperoleh sebesar 4.2939. Pada iterasi ke-100 *loss* yang diperoleh sebesar 1.0477. Pada iterasi ke-101 s.d. ke-600 *loss* semakin kecil dan diperoleh *loss* sebesar 0.3963 pada iterasi ke-600. Bahwa semakin kecil *loss* juga mempengaruhi tingkat akurasi model pada saat proses pelatihan. Pada iterasi ke-1 tingkat akurasi sebesar 27,08%. Semakin lama proses iterasi semakin tinggi pula tingkat akurasi dari model tersebut. Akurasi tertinggi dari model sebesar 87,5% diperoleh pada iterasi ke-600.

Grafik *loss* dan akurasi pada proses pelatihan model dapat dilihat pada Gambar 6 dan Gambar 7.



Gambar 6. Loss pada Model Pelatihan vs Validasi



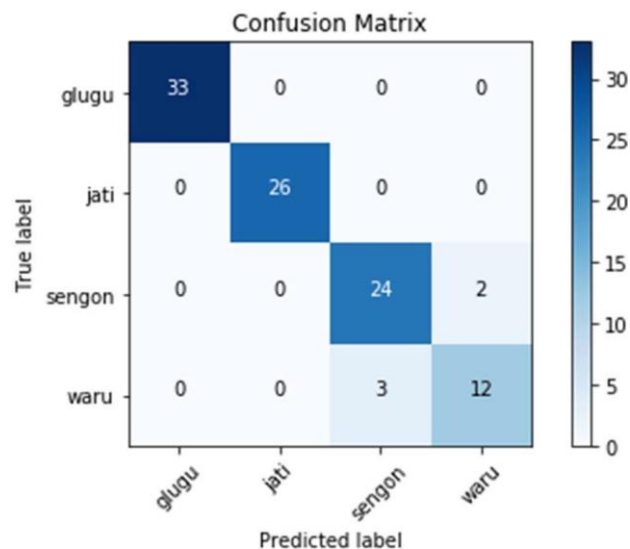
Gambar 7. Akurasi pada Model Pelatihan vs Validasi

B. Pengujian

Sebanyak 100 data dari empat kelas kayu yang berbeda diuji menggunakan model yang telah dibuat sebelumnya. Hasil prediksi benar mencapai 95 data dan terdapat 5 kesalahan identifikasi. Dengan kata lain akurasi prediksi mencapai 95%. Pada pengujian tidak dilakukan *cross validation* karena model memprediksi atas data citra yang sama sekali baru dan juga sebelumnya tidak diikuti dalam proses pelatihan maupun validasi. Hasil selengkapnya dapat dilihat pada Tabel 4 dan *confusion matrix* pada Gambar 8.

TABEL 4
 HASIL PENGUJIAN

No	Tipe Kayu	Data Pengujian	Benar	Salah
1	Glugu	33	33	0
2	Jati	26	26	0
3	Sengon	26	24	2
4	Waru	15	12	3
Total		100	95	5



Gambar 8. Confusion Matrix Hasil Pengujian

IV. SIMPULAN

Dalam penelitian ini telah diimplementasikan suatu metode untuk mengidentifikasi jenis kayu dari citra dengan *Convolutional Neural Network* (CNN). Hasil identifikasi pada data pengujian menunjukkan bahwa model mampu mengidentifikasi jenis kayu dengan akurasi 95%. Pada penelitian selanjutnya, akan menambahkan lebih banyak kelas jenis kayu dan kondisi tekstur kayu yang berbeda seperti cacat kayu, pelapukan, dan lain-lain.

DAFTAR PUSTAKA

- [1] I. Ibrahim, A. S. M. Khairuddin, M. S. Abu Talip, H. Arof, & R. Yusof, *Tree species recognition system based on macroscopic image analysis*, *Wood Sci Technol*, vol. 51, no. 2, pp. 431–444, Mar. 2017.
- [2] P. Barmpoutis, K. Dimitropoulos, I. Barboutis, N. Grammalidis, & P. Lefakis, *Wood species recognition through multidimensional texture analysis*, *Computers and Electronics in Agriculture*, vol. 144, pp. 241–248, Jan. 2018.
- [3] J. L. Bowyer, R. Shmulsky, & J. G. Haygreen, *Forest products and wood science: an introduction*, 5th ed. Ames, Iowa: Blackwell Pub, 2007.
- [4] I. Gunawan, J. Halomoan, & R. D. Atmaja, *Perancangan Sistem Identifikasi Kualitas Kayu Untuk Quality Kontrol Berbasis Pengolahan Citra Digital*, *eProceedings of Engineering*, Telkom University, vol. 2, no. 1, pp. 1–8, 2015.
- [5] A. Harjoko & Gasim, *Comparison of some features extraction method in wood identification*, in 2010 International Conference on Distributed Frameworks for Multimedia Applications, 2010, pp. 1–6.
- [6] S. Hu, K. Li, & X. Bao, *Wood Species Recognition Based on SIFT Keypoint Histogram*, arXiv, 2015.
- [7] B. Sugiarto, E. Prakasa, R. Wardoyo, R. Damayanti, Krisdianto, L. M. Dewi, H. F. Pardede, & Y. Rianto, *Wood identification based on histogram of oriented gradient (HOG) feature and support vector machine (SVM) classifier*, in 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, 2017, pp. 337–341.
- [8] E. Yuliasuti, Suprijanto, & S. R. Sasi, *Compact computer vision system for tropical wood species recognition based on pores and concentric curve*, in 2013 3rd International Conference on Instrumentation Control and Automation (ICA), Ungasan, 2013, pp. 198–202.
- [9] A. A. G. R. Gunawan, S. Nurdiani, & Y. Arkeman, *Identifikasi Jenis Kayu Menggunakan Support Vector Machine Berbasis Data Citra*, *JIKA*, vol. 3, no. 1, p. 1, Jan. 2017.
- [10] R. Yusof & N. R. Rosli, *Tropical Wood Species Recognition System Based on Gabor Filter as Image Multiplier*, in 2013 International Conference on Signal-Image Technology & Internet-Based Systems, Kyoto, Japan, 2013, pp. 737–743.
- [11] M. Xin & Y. Wang, *Research on image classification model based on deep convolution neural network*, *J Image Video Proc.*, vol. 2019, no. 1, p. 40, Dec. 2019.
- [12] K. Fukushima, S. Miyake, & T. Ito, *Neocognitron: A neural network model for a mechanism of visual pattern recognition*, *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 5, pp. 826–834, Sep. 1983.
- [13] S. Khan, H. Rahmani, S. A. A. Shah, & M. Bennamoun, *A Guide to Convolutional Neural Networks for Computer Vision*, *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, Feb. 2018.
- [14] M. Zufar & B. Setiyono, *Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-time*, *Jurnal Sains dan Seni ITS*, vol. 5, no. 2, pp. 72–77, 2016.
- [15] *CS231n Convolutional Neural Networks for Visual Recognition*. [Online]. Tersedia: <https://cs231n.github.io/convolutional-networks>.
- [16] Y. Hong & J. Kim, *Art Painting Identification using Convolutional Neural Network*, *International Journal of Applied Engineering Research*, vol. 12, no. 4, pp. 532–539, 2017.
- [17] *[Deep Learning] Deep Learning Terminology*. [Online]. Tersedia: <https://byunghyun23.tistory.com/41>.
- [18] K. Simonyan & A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv, Apr. 2015.