

Klasifikasi *Image* Jenis Kayu pada Furnitur dengan *Convolutional Neural Network*

Edward Christopher R.^{#1}, Evasaria Magdalena Sipayung^{#2}

[#]*Program Studi Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia
Jl. Jalur Sutera Kav. 7-9, Alam Sutera, Tangerang, Indonesia*

¹*echristopherr@gmail.com*

²*evasaria.sipayung@gmail.com*

Abstract— *Different types of wood have unique patterns and colors. A classification process is necessary to identify the type of wood using the Convolutional Neural Network (CNN) method. This method enables feature extraction, which becomes the data used to classify wood species. The wood image data collected from augmentation data consists of 120 images, including teak, mahogany, oak, and pine wood types. The four wood species classes have a 70% training data and a 30% test data ratio. Each class uses four convolutional layers with filters of 32, 32, 64, and 64, and a pool size of 2x2 with 512 neurons in the hidden layer. Testing the image classification website using the confusion matrix method resulted in an accuracy of 80.5%.*

Keywords— *wood, classification, Convolutional Neural Network, image classification, confusion matrix*

Abstrak— Kayu memiliki berbagai macam jenis. Setiap jenis kayu memiliki pola-pola tertentu dengan warna yang beragam. Dalam menentukan jenis kayu maka perlu dilakukan sebuah klasifikasi dengan menggunakan metode *Convolutional Neural Network* (CNN). Dengan penggunaan CNN dapat dilakukan ekstraksi sebuah fitur kemudian fitur-fitur tersebut akan menjadi data dalam menentukan klasifikasi jenis kayu. Data gambar kayu yang dikumpulkan dari data augmentasi adalah sebanyak 120 gambar dengan jenis kayu jati, mahoni, oak, dan pinus. Keempat kelas jenis kayu tersebut memiliki perbandingan data latih 70% dan data uji 30%. Masing-masing kelas menggunakan empat *convolutional layer* dengan filter 32, 32, 64, dan 64 dan menggunakan *pool size* sebesar 2x2 dengan *neuron* (*hidden layer*) sebanyak 512. Pengujian *website image classification* dengan menggunakan metode *confusion matrix* didapatkan akurasi sebesar 80,5% dari pengujian yang dilakukan pada data uji.

Kata Kunci— *kayu, klasifikasi, Convolutional Neural Network, image classification, confusion matrix*

I. PENDAHULUAN

Furnitur merupakan benda yang sangat sering kita jumpai setiap hari. Furnitur dapat membantu hidup menjadi lebih nyaman. Terdapat furnitur yang memiliki bahan dasar yang berasal dari kayu. Kayu memiliki berbagai jenis pola yang dapat membedakan jenisnya. Kayu dapat dibedakan dari pola. Dengan adanya pola tersebut, maka dilakukan klasifikasi dalam membedakan jenis kayu.

Convolutional Neural Network (CNN) merupakan *neural network* yang didesain untuk mengolah data dua dimensi.

CNN termasuk dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada ekspresi manusia dengan akurasi 80% [1]. CNN memiliki akurasi yang tinggi sebesar 99% dan dapat memberikan hasil yang baik dalam mengenali sebuah objek pada sebuah pengenalan citra [2].

CNN memiliki fungsi untuk mengklasifikasikan sebuah citra. Hasil penelitian yang diperoleh untuk klasifikasi kualitas kayu kelapa menunjukkan bahwa arsitektur GoogLeNet memiliki performansi klasifikasi dengan rata-rata akurasi 84,89% pada setiap lapisan [3]. CNN dengan model YOLOv5 menggunakan aktivasi *leaky Relu* (*retrified linear unit*) dan *sigmoid* dengan nilai mAP terbaik pada *dataset* kayu pinus menggunakan model YOLOv5s, tanpa penambahan *image enhancement* sebesar 94,3% [4].

Pada penelitian ini dilakukan pengklasifikasian *image* tekstur kayu pada furnitur kayu dengan *convolutional neural network*. *Image classification* digunakan untuk menentukan jenis kayu dari sebuah tekstur kayu. Terdapat empat kelas yang dapat diklasifikasikan pada *website image classification*, yaitu kayu jati, kayu mahoni, kayu oak, dan kayu pinus. Informasi yang ditampilkan berupa kelas prediksi dan probabilitas/persen kemiripan dengan kelas yang diprediksi. Dengan memanfaatkan *website image classification* diharapkan dapat membantu pengguna dalam mengenali jenis kayu dari sebuah tekstur kayu.

II. METODOLOGI

CNN merupakan salah satu dari algoritme *deep learning*, perkembangan dari *multilayer perceptron* (MLP) yang digunakan untuk mengolah sebuah data citra dalam bentuk 2 dimensi. Algoritme ini dapat mengklasifikasikan sebuah data yang telah diberikan label. Label tersebut untuk mengklasifikasikan sebuah data baru dengan data yang sudah diuji [5]. Model CNN merupakan sebuah jaringan *neuron multilayer* yang terdiri atas 2 bagian, yaitu ekstraksi fitur dan pelatihan data. Pelatihan data dilakukan untuk masuk ke dalam kelas. Ekstraksi fitur terdiri atas lapisan *feature map* dan bagian pengambilan fitur yang berbeda dari gambar awal, yaitu: konvolusi [6]. CNN terdiri dari terdiri atas *input layer*, *convolution layer*, *activation function*, *pooling layer*, *flatten layer*, dan *fully connected layer* [7].

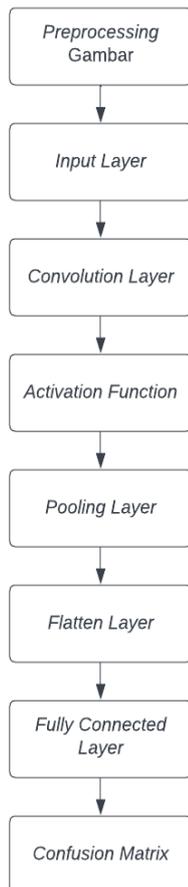
Tahap yang dilalui pada penelitian ini terdiri atas pengumpulan data kemudian dilanjutkan tahap *preprocessing* gambar. Gambar yang sudah melalui tahap *preprocessing*

dikelompokkan dengan menggunakan arsitektur CNN. Arsitektur CNN meniru otak manusia terutama dalam cara kerja korteks dan sistem sarafnya. Adapun tahapan dalam arsitektur CNN terdiri dari *input layer*, *convolution layer*, *activation function*, *pooling layer*, *flatten layer*, dan *fully connected layer*. Metode yang digunakan untuk proses pengujian yaitu *confusion matrix*. Tahapan proses yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 1.

A. *Preprocessing Gambar*

Pengambilan data gambar dari *Google Image* berjumlah 120 gambar berupa *image* tekstur jenis kayu yang dibagi menjadi empat kategori, yaitu: kayu jati, kayu mahoni, kayu oak, dan kayu pinus kemudian dimasukkan ke dalam *dataset*. Dilakukan perbandingan data sebesar 70% untuk data *training*, yaitu pembelajaran data oleh komputer, dan 30% data *test*, yaitu pengujian data yang belum pernah dilihat oleh komputer.

Semua gambar dari data *training* akan dilakukan beberapa proses, seperti *rescale* yang berfungsi untuk mengubah skala *grayscale* (sesuai warna aslinya), rotasi, dan fungsinya untuk melakukan perputaran pada gambar. Gambar awal memiliki warna sesuai dengan jenis kayu dan posisi. *Horizontal flip* dapat membuat gambar yang tadinya ada di posisi kiri berpindah ke posisi kanan/sebaliknya. *Fill mode* dengan *nearest* yang dapat berfungsi untuk mengisi ruang kosong dari setiap piksel dengan mengambil nilai terdekatnya. Untuk validasi data hanya dilakukan proses *rescale* karena komputer



Gambar 1 Proses penelitian

hanya mempelajari data *train*. Agar komputer makin banyak belajar maka gambar diperbanyak dengan menggunakan *preprocessing* (augmentasi) ini.

Data augmentasi adalah sebuah cara yang sering digunakan untuk memperkecil *overfitting* dalam sebuah model. Dengan melakukan augmentasi data, manfaat yang dimiliki dapat menghasilkan sebuah data baru yang menggunakan sebuah perubahan-perubahan yang ada pada data asli. Augmentasi data memungkinkan untuk meningkatkan generalisasi data atau model dalam sebuah citra [8]. Augmentasi dapat menghasilkan data *training* baru dari data aslinya [9].

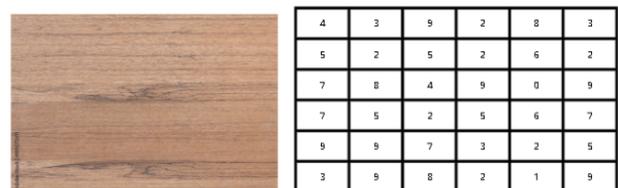
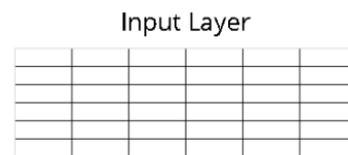
B. *Input Layer*

Layer ini merupakan tempat awal mula gambar dimasukkan sebelum menjalankan sebuah proses klasifikasi gambar. *Layer* ini merupakan *layer* pertama dalam arsitektur CNN. Pada bagian ini nantinya semua gambar dapat dimasukkan dan diproses ke *layer* berikutnya. Mata manusia melihat sebuah gambar, namun sebuah komputer jika melihat sebuah gambar akan direpresentasikan sebagai kumpulan sebuah piksel yang nantinya *input layer* memiliki ukuran gambar sebesar 128 x 128 piksel.

Pada simulasi ini, *input layer* memiliki ukuran 6 x 6 piksel sehingga nantinya seluruh gambar yang masuk ke dalam arsitektur untuk diproses ukuran dikonversi menjadi ukuran 6 x 6 piksel, seperti yang ditunjukkan pada Gambar 2. Semua ukuran dibuat sama sehingga tidak ada data yang memiliki piksel di luar piksel yang ditentukan. Jika terjadi perbedaan ukuran, nantinya nilai piksel Gambar 1 dengan gambar yang lainnya tidak dapat dibandingkan.

C. *Convolution Layer*

Pada *layer* ini dilakukan sebuah proses untuk memperoleh sebuah piksel yang didasarkan nilai piksel sendiri dan nilai piksel tetangga dengan melibatkan sebuah matriks yang biasa dibilang *kernel* dalam menampilkan nilai bobotnya [10]. Dengan adanya *convolution layer* ini, gambar dapat diekstrak fiturunya untuk diambil pola bentuk dan warna. Dengan demikian, komputer dapat membedakan antara kayu jati, kayu mahoni, kayu oak, dan kayu pinus dengan melihat serat dan warna dari tiap kayu pada *layer* sebelumnya.



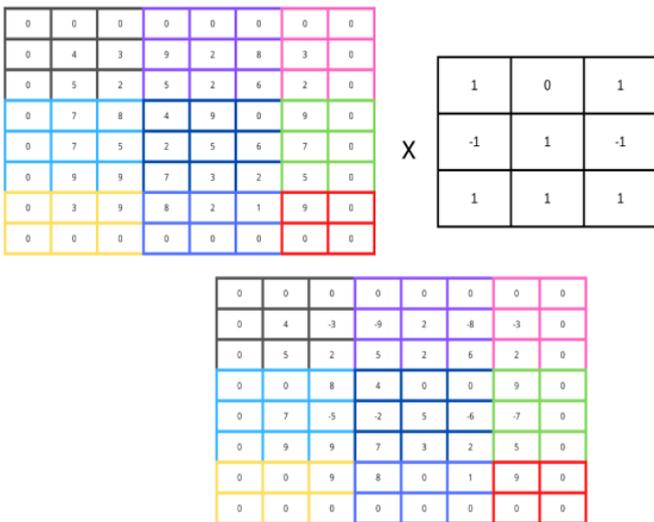
Gambar 2 Simulasi *input layer*

Filter size yang digunakan adalah 3 x 3 karena *input layer* adalah 128 x 128 piksel. Jumlah *convolution layer* terdiri dari 4 buah. Jika menggunakan kurang atau lebih dari empat *layer*, akan mengurangi nilai akurasi untuk jumlah filter baru dengan nilai 32, 32, 64, 64. Pemilihan angka ini didasari dengan pengembangan dari penelitian pendahulu. Jika menggunakan sesuai dengan *layer*, nilai akurasinya menjadi kurang baik.

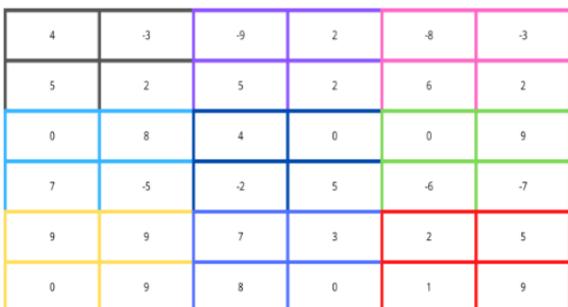
Penggunaan *padding same* dilakukan agar data yang terbuang dikurangi. Angka filter tidak ditentukan sebelumnya. Karena tidak ditentukan di awal, isi dari semua piksel pada *filter* otomatis sesuai dengan fitur yang akan diekstrak pada *fully connected layer*, seperti deteksi tepi dan lain-lain. Contoh simulasi perhitungan *convolution* dengan *layer input* dan *filter* berukuran 3x3 piksel. Karena *padding same*, maka *image input* akan menjadi 8 x 8 piksel dengan *padding 0* dan *image* 6 x 6 piksel menjadi 8 x 8 piksel lalu dikalikan dengan filter baru, seperti yang ditunjukkan pada Gambar 3. Piksel akan kembali ke ukuran semula, yaitu berukuran 6 x 6 piksel, seperti yang ditunjukkan pada Gambar 4.

D. Activation Function

Layer ini merupakan sebuah salah satu dari fungsi aktivasi [11] yang menerapkan *retrified linear unit* (Relu) yang berfungsi untuk membatasi nilai negatif sehingga nantinya diubah menjadi angka nol.



Gambar 3 Simulasi perhitungan filter dengan *input*



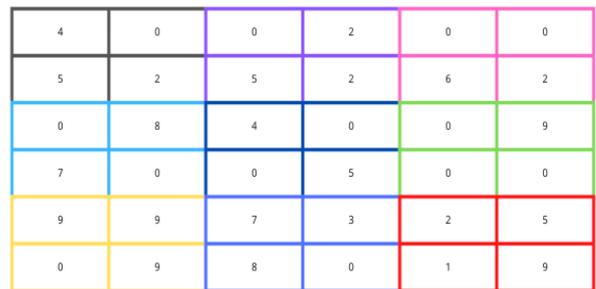
Gambar 4 Simulasi penggunaan *size* semula

Fungsi dari *activation* salah satunya, yaitu untuk membuat semua nilai yang bernilai minus menjadi positif dengan menggunakan aktivasi Relu sehingga nantinya semua nilai yang bernilai minus akan dibuat 0. Angka minus hanya akan memperlambat komputasi komputer dalam melakukan *train* sebuah model karena filter yang bernilai minus tidak dapat diekstrak nilainya sehingga akan dihapus. Fungsi *activation* ini ditunjukkan pada Gambar 5.

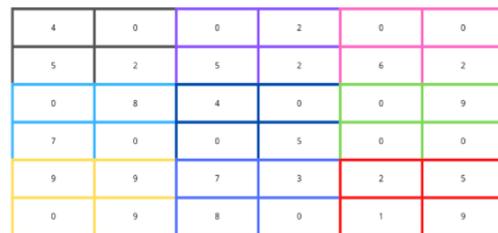
E. Pooling Layer

Pooling layer memiliki *filter* dan *stride* tertentu, yaitu dengan ukuran filter 2x2 dan dengan nilai *stride* berukuran 1. Fungsi dari *pooling layer* ini, yaitu mencari nilai maksimum dari sebuah *kernel* (matriks). Proses *pooling layer* ditunjukkan pada Gambar 6. Penggunaan *pooling layer* sebesar 2x2 karena jika ukuran dimenasi hanya 1x1, akan tetap mempertahankan dimensinya. Namun, jika menggunakan 3x3, akan semakin banyak data yang terbuang [12]. Semakin banyak data yang terbuang, kemungkinan akan terdapat fitur penting yang hilang. Dengan demikian, fitur penting yang didapatkan juga akan semakin sedikit.

Pooling layer memiliki fungsi untuk mengurangi komputasi komputer sehingga hanya mengambil nilai piksel terbesar dari setiap *pool size*. Nilai yang lebih kecil kurang berguna untuk di ekstrak fitur-fiturnya sehingga hanya akan menambah komputasi komputer. *Pool size* sebesar 2 x 2 agar lebih sedikit data yang terbuang. Simulasi *pooling layer* ditunjukkan pada Gambar 6.



Gambar 5 Simulasi *activation function*



Gambar 6 Simulasi *pooling layer*

F. Flatten Layer

Pada layer ini dibentuk suatu *vector* satu dimensi dari nilai matriks pada tahap sebelumnya. Pada *pooling layer* gambar masih memiliki ukuran 2 dimensi. Dengan dilakukan hal ini, maka hasil *flatten layer* ini akan menjadi nilai *input* pada *fully connected layer* untuk mendapatkan hasil klasifikasi [13].

Flatten layer memiliki fungsi menjadi masukan *layer* berikutnya untuk dikumpulkan dan diekstrak fiturnya menjadi satu bagian. Dari bentuk satu dimensi nantinya seluruh nilai yang ada di *flatten layer* akan masuk ke dalam *fully connected layer* yang juga memiliki bentuk satu dimensi. Pada *layer* sebelumnya dari 3x3 piksel akan menjadi satu dimensi yang dilakukan dari kiri atas ke bawah kanan. Simulasi *flatten layer* ditunjukkan pada Gambar 7.

G. Fully Connected Layer

Pada *layer* ini didapatkan nilai masukan (matriks) dari hasil *flatten layer* pada tahap sebelumnya. Setiap masukan akan terhubung dengan *neuron* yang berada di *hidden layer* untuk mendapatkan hasil berdasarkan nilai yang ada pada *hidden layer*. *Fully connected layer* ditunjukkan pada Gambar 8. *Feature map* terbuat dari ekstraksi fitur yang masih berupa *array* multidimensi. *Feature map* perlu diubah menjadi vektor agar dapat digunakan sebagai nilai masukan vektor dari lapisan yang terhubung sepenuhnya.

Fully connected layer merupakan lapisan *neuron* aktif pada lapisan sebelumnya dan terhubung dengan *neuron* pada lapisan berikutnya sehingga menyerupai jaringan syaraf tiruan [14]. *Fully connected layer* berisi kumpulan seluruh fitur dari sebuah gambar mulai dari *edge detection*, pola bentuk dari tiap piksel, warna, dan lain-lain. Setiap *layer* yang ada pada *fully connected layer* akan saling berhubungan satu dengan lainnya sehingga membentuk sebuah jaringan *neuron*. Keluarannya akan menghasilkan sebuah kelas kayu, yaitu kelas jati, mahoni, oak, dan pinus. Simulasi ini ditunjukkan pada Gambar 9.

Bagian kiri berupa piksel warna-warni berisi *flatten* yang merupakan masukan dari *layer* sebelumnya. *Fully connected layer* berisi ekstrak dari sebuah gambar yang telah dipelajari sebelumnya sehingga *flatten layer*, *fully connected layer*, dan



Gambar 7 Simulasi *flatten layer*

output layer akan saling terhubung. Komputer dapat menyimpulkan kelas yang didapatkan dari keterhubungan antar *layer* tersebut.

Confusion matrix merupakan sebuah tabel perhitungan akurasi yang dapat menyatakan keberhasilan uji yang benar atau salah dalam melakukan klasifikasi [15].

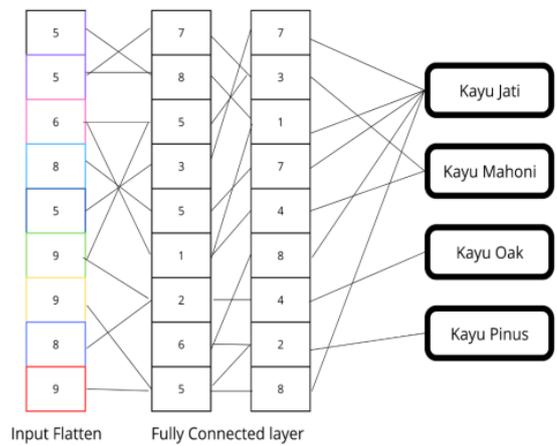
III. HASIL DAN PEMBAHASAN

Hasil Persentase menampilkan gambar yang sebelumnya telah diunggah dan menampilkan hasil prediksi berupa jenis (*class*) kayu (kayu jati, kayu mahoni, kayu oak, dan kayu pinus) dan hasil persentase (*probability*). Pada Hasil Persentase ini juga terdapat tombol *back* untuk kembali ke halaman *home* (beranda). Tampilan Hasil Persentase ditunjukkan pada Gambar 9.

A. Implementasi Convolutional Neural Network

Arsitektur CNN pada *Convolution Layer* 1 terdiri atas 32 *filter* baru dengan *kernel size* sebesar 3x3 dan dengan *input shape* 128, 128, 3 (*pixel width*, *pixel height*, *image channel*).

- *Padding* yang sama ditambahkan *pooling max* (*Pooling Layer* 1) dengan *pool size* sebesar 2x2 pada *Convolution Layer* 2 yang terdiri dari 32 *filter* baru dan ukuran sebesar 3x3.
- *Padding* yang sama dilakukan *pooling max* (*Pooling Layer* 2) yang terdiri dari *pool size* sebesar 2x2 pada



Gambar 8 Simulasi *fully connected layer*



Gambar 9 Hasil persentase kayu jati

- *Convolution Layer* 3 yang terdiri dari 64 *filter* baru dan ukuran sebesar 3x3.
- *Padding* yang sama dilakukan *pooling max* (*Pooling Layer* 3) yang terdiri dari *pool size* sebesar 2x2. Untuk *layer* konvolusi yang terakhir (*Convolution Layer* 4) terdiri dari 64 *filter* baru dan ukuran sebesar 3x3.
- *Padding* yang sama dilakukan *pooling max* (*Pooling Layer* 4) yang terdiri dari *pool size* sebesar 2x2 *flatten* agar menjadi sebuah *vector* yang nantinya digunakan sebagai masukan *fully-connected layer*.

Neuron sebanyak 512 akan menggunakan Relu untuk mengubah semua piksel yang negatif menjadi positif dengan kelas sebanyak 4 buah, yaitu: kayu jati, kayu mahoni, kayu oak, dan kayu pinus.

B. Pengujian Algoritma CNN

Pengujian menggunakan *confusion matrix* dilakukan secara manual. Pengujian dilakukan per satu gambar pada data *test* yang sudah dipisahkan sebelumnya. Sebelum pengujian menggunakan *confusion matrix* dapat dilakukan, harus melewati tahap pelatihan. Tahap pelatihan melibatkan pelatihan data dan pengujian dengan data validasi. *Step per epoch*, yaitu *batch* yang dieksekusi tiap *epoch*, data *train* dibagi *batch size* data *training* dengan *epoch* sebanyak 160 kali. Pada *step validation*, jumlah data *validation* dibagi *batch size* data *validation*. Dilakukan pembagian *batch* ini untuk menghemat waktu pelatihan, dengan catatan *epoch* setidaknya sesuai dengan *batch data train* atau lebih tinggi agar semua data dapat dilatih. Fungsi tahap *train*, yaitu untuk melihat hasil model CNN dari pelatihan data.

Sebelum melakukan pemasukan pada tabel *confusion matrix*, maka semua data *test* harus diuji satu per satu. Pada kelas kayu jati terdapat 3 kesalahan, pada kelas kayu mahoni terdapat 1 kesalahan, pada kelas kayu oak terdapat 2 kesalahan, dan pada kelas kayu pinus terdapat 1 kesalahan. Semua hasil yang salah ini ditunjukkan pada Tabel I.

Tingkat akurasi dari data *test* pada metode pengujian *confusion matrix* adalah sebesar 80,5% yang didapatkan dari jumlah gambar seluruh kelas benar dibagi seluruh gambar. Kelas yang benar akan berwarna kuning yang disebut dengan *True Positive* (TP) dan angka yang tidak memiliki warna disebut *False Positive* (FP).

$$Accuracy = \frac{TP}{(TP + FP)} \tag{1}$$

Hasil perhitungannya adalah:

$$Accuracy = \frac{6 + 8 + 7 + 8}{6 + 1 + 0 + 8 + 1 + 2 + 2 + 7 + 1 + 0 + 8} = \frac{29}{36} = 0,805$$

IV. SIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan perbandingan terhadap *dataset* sebesar 70% (data *train*) dan 30% (data *test*). Dengan arsitektur CNN yang memiliki kriteria empat *convolution layer*, maka dibuat *convolution layer* pertama dan kedua terdiri atas 32 filter dan

TABEL I
HASIL *CONFUSION MATRIX*

Confusion Matrix	Kelas Prediksi			
	Kayu jati	Kayu mahoni	Kayu oak	Kayu pinus
Kayu jati	6	0	2	0
Kayu mahoni	1	8	0	0
Kayu oak	2	1	7	1
Kayu pinus	0	0	0	8

convolution layer ketiga dan keempat terdiri atas 64 filter. Empat *pooling layer* memiliki *pool size* masing-masing sebesar 2x2 dan memiliki neuron sebanyak 512 untuk 4 kelas jenis kayu, yaitu jati, mahoni, oak, dan pinus. Pelatihan data *train* dan pengujian validasi dilakukan sebanyak 160 kali (*epoch*). Dengan arsitektur ini juga dilakukan pengujian data *test* menggunakan *confusion matrix* terhadap kriteria data *test* dari kelas kayu jati, kayu mahoni, kayu oak, dan kayu pinus. Hasil persentasenya sebesar 80,5 %.

DAFTAR REFERENSI

- [1] P. A. Nugroho, I. Fenriana, dan R. Arijanto, "Implementasi *deep learning* menggunakan *convolutional neural network* (CNN) pada ekspresi manusia," *Jurnal Algor*, vol. 2, no. 1, hlm. 12–21, 2020.
- [2] M. R. Efrin dan U. Latifa, "Image recognition berbasis *convolutional neural network* (CNN) untuk mendeteksi penyakit kulit pada manusia," *Jurnal Polektro: Jurnal Power Elektronik*, vol. 11, no. 1, hlm. 276–282, 2022, DOI: [10.30591/polektro.v12i1.3874](https://doi.org/10.30591/polektro.v12i1.3874).
- [3] N. F. Mustamin, Y. Sari, dan H. Khatimi, "Klasifikasi kualitas kayu kelapa menggunakan arsitektur CNN," *Kumpulan jurnal Ilmu Komputer (KLIK)*, vol. 8, no. 1, hlm. 49–59, 2021, DOI: <http://dx.doi.org/10.20527/klik.v8i1.370>.
- [4] T. Riantiarni, "Deteksi Cacat pada Permukaan Kayu Menggunakan Model *Convolutional Neural Network* Yolov5," Skripsi, Telkom University, Bandung, 2022.
- [5] S. Ilaheiyah dan A. Nilogiri, "Implementasi *deep learning* pada identifikasi jenis tumbuhan berdasarkan citra daun menggunakan *convolutional neural network*," *Justindo (Jurnal Sistem & Teknologi Informasi Indonesia)*, vol. 3, no. 2, hlm. 49–56, 2018.
- [6] U. N. Oktaviana, R. Hendrawan, A. D. K. Annas, dan G. W. Wicaksono, "Klasifikasi penyakit padi berdasarkan citra daun menggunakan model terlatih Resnet101," *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, hlm. 1216–1222, 2021, DOI: <https://doi.org/10.29207/resti.v5i6.3607>.
- [7] E. Rasywir, R. Sinaga, dan Y. Pratama, "Analisis dan implementasi diagnosis penyakit sawit dengan metode *convolutional neural network* (CNN)," *Paradigma – Jurnal Informatika dan Komputer*, vol. 22, no. 2, hlm. 117–123, 2020, DOI: <https://doi.org/10.31294/p.v22i2.8907>.
- [8] L. H. Ganda dan H. Bunyamin, "Penggunaan augmentasi data pada klasifikasi jenis kanker payudara dengan model Resnet-34," *Jurnal Strategi*, vol. 3, no. 1, hlm. 187–193, 2021.
- [9] L. Farokhah, "Implementasi *convolutional neural network* untuk klasifikasi variasi intensitas emosi pada *dynamic image sequence*," *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 6, hlm. 1070–1076, 2020, DOI: <https://doi.org/10.29207/resti.v4i6.2644>.
- [10] S. Fauzi, P. Eosina, dan G. F. Laxmi, "Implementasi *convolutional neural network* untuk identifikasi ikan air tawar," dalam *Prosiding Seminar Teknologi Informasi (Semnati)*, vol. 2, 2019, hlm. 163–167.
- [11] R. A. Pangestu, B. Rahmat, dan F. T. Anggraeny, "Implementasi algoritme CNN untuk klasifikasi citra lahan dan perhitungan luas,"

- Jurnal Informatika dan Sistem Informasi (JIFoSI) Vol*, vol. 1, no. 1, hlm. 166–174, 2020.
- [12] M. S. Maheshan dan B. S. H. N. Nagadarshan, “A convolution neural network engine for sclera recognition,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, 2018, DOI: [10.9781/ijimai.2019.03.006](https://doi.org/10.9781/ijimai.2019.03.006).
- [13] G. Wicaksono, S. Andryana, dan Benrahman, “Aplikasi pendeteksi penyakit pada daun tanaman apel dengan metode *convolutional neural network*,” *Jointecs (Journal of Information Technology and Computer Science)*, vol. 7, no. 1, hlm. 9–16, 2020, DOI: <https://doi.org/10.31328/jointecs.v5i1.1221>.
- [14] A. Kholik, “Klasifikasi menggunakan *convolutional neural network* (CNN) pada tangkapan layar halaman Instagram,” *Jurnal Data Mining dan Sistem Informasi (JDMSI)*, vol. 2, no. 2, hlm. 10–20, 2021, DOI: <https://doi.org/10.33365/jdmsi.v2i2.1345>.
- [15] D. Normawati dan S. A. Prayogi, “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 5, no. 2, hlm. 697–711, Sep. 2021, DOI: <http://dx.doi.org/10.30645/j-sakti.v5i2.369>

Edward Christopher R, alumnus Prodi Informatika Universitas Bunda Mulia angkatan 2018.

Evasaria Magdalena Sipayung, menerima Sarjana Teknik dari Sekolah Tinggi Teknologi Telkom jurusan Teknik Informatika tahun 2003 dan gelar Magister Teknik dari ITB jurusan Teknik Elektro bidang Teknologi Informasi pada tahun 2007. Saat ini aktif sebagai dosen tetap di Prodi Informatika Universitas Bunda Mulia Serpong.