

Klasifikasi Batik Indonesia Menggunakan Convolutional Neural Network (CNN)

**Maria Misela A. Wona¹, Salsa Aulia Asyifa², Rizka Virgianti³, Muhammad Nabil Hamid⁴,
Irwan Muji Handoko⁵, Ni Wayan Parwati Septiani^{*6}, Mei Lestari⁷**

^{1,2,3,4,5,6,7} Jurusan Teknik Informatika, Universitas Indraprasta PGRI; Jalan Raya Tengah No 80,
Kelurahan Gedong, Pasar Rebo, Jakarta Timur, Jakarta

e-mail: ¹mariawona19@gmail.com, ²salsaaasyifa@gmail.com, ³rizkavgnt30@gmail.com,

⁴mnabilhamid@gmail.com, ⁵irwan.muji77@gmail.com, ^{*6}wayan.parwati@gmail.com,

⁷mei.lestari6@gmail.com

Abstrak

Batik Indonesia merupakan warisan budaya yang kaya akan corak, motif, dan warna, dan telah diakui sebagai warisan budaya dunia oleh UNESCO. Namun, masih terdapat kendala dalam memperkenalkan batik Indonesia kepada masyarakat luas, terutama generasi milenial dan di luar Indonesia. Sistem klasifikasi batik Indonesia berbasis *website* ini menggunakan teknologi *machine learning*, khususnya metode *Convolutional Neural Network (CNN)*. Sistem ini dapat mengidentifikasi dan mengklasifikasikan jenis batik Indonesia berdasarkan ciri-ciri seperti corak, warna, dan bentuk. Hal ini memungkinkan pengguna untuk mempelajari dan mengenali jenis-jenis batik Indonesia secara interaktif melalui *website*, sehingga memperkenalkan kekayaan budaya batik Indonesia kepada masyarakat luas dengan lebih efektif. Melalui pengujian model *deep learning* dengan algoritma CNN, penelitian ini berhasil mencapai tingkat akurasi sebesar 91,24% pada data *testing*. Hasil ini menunjukkan bahwa model yang dikembangkan mampu mengenali dan mengklasifikasikan citra batik dengan baik.

Kata kunci— Batik, Convolutional Neural Network(CNN) , Klasifikasi

1. PENDAHULUAN

Batik adalah salah satu budaya khas Indonesia dan sudah diakui sebagai warisan budaya Internasional oleh UNESCO (*The United Nations Educational, Scientific and Cultural Organization*) pada tanggal 2 Oktober 2009. Batik telah menjadi warisan budaya turun temurun di seluruh Indonesia khususnya di daerah Jawa [1]. Keberagaman corak, motif, dan warna batik menjadikannya sebagai ciri khas yang membanggakan bagi bangsa Indonesia [2].

Namun, masih terdapat kendala dalam memperkenalkan batik Indonesia kepada masyarakat luas, terutama kepada generasi milenial dan di luar Indonesia. Dalam mengatasi kendala tersebut, sistem klasifikasi batik Indonesia berbasis web dengan metode CNN dapat memberikan solusi yang efektif. Sistem ini menyediakan informasi yang mudah diakses dan memperkenalkan kekayaan budaya batik Indonesia kepada masyarakat luas. Dengan menggunakan teknologi *machine learning*, sistem ini dapat mengidentifikasi dan mengklasifikasikan jenis batik Indonesia berdasarkan ciri-ciri tertentu, seperti corak, warna, dan bentuk. Dengan adanya sistem ini, pengguna dapat mempelajari dan mengenali jenis-jenis batik Indonesia secara interaktif melalui *website*.

Penelitian ini dapat melestarikan dan mempromosikan kebudayaan batik Indonesia serta memperluas pengetahuan tentang batik Indonesia di tingkat nasional maupun internasional. Selain itu, teknologi *machine learning* yang diterapkan dalam penelitian ini juga dapat memberikan kontribusi pada pengembangan teknologi informasi di Indonesia. Dengan adanya sistem klasifikasi batik berbasis *website* ini, diharapkan batik Indonesia dapat lebih dikenal dan

diapresiasi oleh masyarakat luas, termasuk generasi milenial, serta dapat menjadi bagian dari upaya pelestarian warisan budaya Indonesia.

2. METODE PENELITIAN

2.1. Pengumpulan Data

Dataset gambar batik yang dibutuhkan untuk penelitian ini diunduh dari situs *website* <https://www.kaggle.com/datasets/dionisiusdh/indonesianbatik-motifs>. Dataset yang dikumpulkan adalah sebanyak 3.880 data citra batik. Data yang sudah terkumpul ini kemudian dilakukan klasifikasi. Secara keseluruhan, klasifikasi dapat diartikan sebagai langkah pengelompokan, yang pada dasarnya mengidentifikasi dan memisahkan objek yang berbeda [3]. Dataset kemudian dikelompokkan ke dalam 14 jenis batik yaitu batik tambal, batik parang, batik kawung, batik geblek renteng, batik betawi, batik sekar jagad, batik megamendung, batik dayak, batik bali, batik poleng, batik lasem, batik ikat celup, batik cendrawasih. Dataset yang diperoleh kemudian akan dibagi menjadi tiga bagian: dataset training sebanyak 70% (2716 gambar), dataset validation 20% (776 gambar), dan dataset testing 10% (388 gambar).

2.2. Preprocessing atau Persiapan data Citra Batik

Preprocessing adalah langkah yang bertujuan untuk meningkatkan kualitas citra, menghilangkan noise yang mungkin ada, dan menentukan bagian citra yang akan digunakan pada langkah-langkah berikutnya [4]. Data citra batik diolah dan dipersiapkan sebelum dimasukkan ke dalam model CNN. Proses persiapan data meliputi:

- 1) Mengubah ukuran citra menjadi ukuran 150x150 piksel. Hal ini dilakukan karena perbedaan ukuran citra yang terdapat dalam dataset.
- 2) Menormalisasi nilai piksel pada citra, sehingga nilai piksel memiliki rentang antara 0 dan 1.
- 3) Membuat label untuk setiap citra yang digunakan dalam pelatihan dan evaluasi model *convolutional neural network*.

2.3. Membangun Model CNN

Istilah *Deep Learning* pertama kali diperkenalkan oleh Geoffrey Hinton pada tahun 2006 saat Ia memperkenalkan salah satu varian jaringan saraf tiruan (*Artificial Neural Network*) yang disebut *Deep Belief Nets* [5]. Konsep untuk melatih jaringan saraf ini adalah dengan melatih dua lapisan pada awalnya lalu menambahkan satu lapisan di atasnya, kemudian jaringan yang dilatih hanyalah lapisan teratasnya dan begitu seterusnya [2]. *Deep Learning* adalah salah satu cabang ilmu dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset besar. *Deep Learning* dapat memberikan hasil yang lebih akurat karena proses ini seperti meniru cara kerja otak manusia. Salah satu metode dalam *Deep Learning* yang mampu memberikan hasil signifikan mengenai pengenalan objek gambar adalah CNN [6].

CNN merupakan sebuah algoritma yang terdiri dari proses ekstraksi ciri menggunakan konvolusi dan proses klasifikasi yang menggunakan *neural network* [7]. Di dalam CNN terdapat empat lapisan utama anatara lain, lapisan konvolusi (*Convolutional Layer*), lapisan aktivasi (*Activation Layer*), lapisan penggabungan (*Pooling Layer*), dan lapisan terhubung penuh (*Fully-Connected Layer*) [8]. Arsitektur CNN dibangun dengan menentukan jumlah layer dan tipe layer yang akan digunakan. Pada umumnya, arsitektur CNN terdiri dari beberapa layer konvolusi, *pooling*, dan *fully connected layer*. Setiap layer tersebut memiliki fungsi dan parameter yang berbeda untuk memproses input citra. Karena hal tersebut, jumlah layer pada jaringan serta jumlah neuron pada masing-masing layer dianggap sebagai *hyperparameter* dan dioptimasi [9].

Penerapan CNN sebagai inovasi yang signifikan dalam bidang kecerdasan buatan. Beberapa penerapan CNN dalam berbagai aspek kehidupan antara lain, dalam bidang Kesehatan CNN digunakan untuk mendeteksi penyakit alzheimer [10]. dalam bidang pertanian cerdas CNN juga digunakan untuk mendeteksi penyakit pada tanaman [11], [12]. CNN juga digunakan dalam bidang keamanan untuk mendeteksi keberadaan seseorang [13].

TensorFlow adalah pustaka perangkat lunak sumber terbuka dan ngratis untuk pembelajaran mesin. *TensorFlow* dapat digunakan dalam berbagai tugas tetapi memiliki fokus khusus pada pelatihan dan inferensi jaringan neural dalam. *TensorFlow* adalah pustaka matematika simbolis berdasarkan dataflow dan pemrograman. Saat ini, TensorFlow merupakan pustaka pembelajaran mesin paling terkenal di dunia [14].

Python adalah bahasa pemrograman yang bersifat open source, sehingga dapat digunakan secara gratis oleh siapa saja. Python memiliki source code yang sederhana sehingga memudahkan dalam pengembangan aplikasi mulai dari tahap menulis kode, testing, hingga tahap perbaikan jika ada kesalahan. Bahasa pemrograman Python memiliki library yang lengkap sehingga cocok untuk membuat model machine learning. Python code akan lebih sederhana jika dibandingkan dengan kode yang ditulis oleh bahasa pemrograman lainnya seperti Java, C, C# dan lain sebagainya [15].

2.4. Pengujian Model CNN

Model CNN dilatih dengan menggunakan data citra batik yang sudah dipersiapkan dan arsitektur CNN yang sudah dibuat. Proses pelatihan model CNN terdiri dari:

- 1) Propagasi maju (*forward propagation*), yaitu proses dimana input citra diteruskan ke dalam arsitektur CNN dan diolah hingga menghasilkan output kelas batik.
- 2) Perhitungan *loss function*, yaitu selisih antara output yang dihasilkan oleh model CNN dan label yang sebenarnya pada setiap citra.
- 3) Propagasi mundur (*backward propagation*), yaitu proses untuk mengoptimalkan parameter dalam arsitektur CNN agar *loss function* seminimal mungkin.
- 4) Optimisasi parameter, yaitu proses untuk menyesuaikan parameter dalam arsitektur CNN agar model dapat memprediksi kelas batik yang benar pada citra batik yang belum pernah dilihat sebelumnya.

Selanjutnya yaitu mengukur kinerja algoritma yang telah dibuat, dengan membaca hasil akurasi untuk rasio prediksi benar dari keseluruhan data. Dan data *loss* digunakan untuk memaksimalkan kinerja algoritma dalam tahap *training* dan validasi model.

Tahap yang terakhir yaitu penarikan kesimpulan dengan menilai hasil *accuracy* dan *loss* yang diperoleh, jika *accuracy* lebih dari 90% dan *loss* kurang dari 50% maka model *convolutional neural network* dapat mendeteksi motif batik dengan cukup baik.

3. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi batik Indonesia berbasis web menggunakan metode CNN. Dalam pengenalan dan klasifikasi pola batik, penggunaan teknologi *machine learning* dengan algoritma CNN dapat memberikan solusi yang efisien dan otomatis. Namun, terdapat beberapa tantangan yang perlu diatasi, seperti keterbatasan data, segmentasi dan ekstraksi fitur yang kompleks, klasifikasi yang akurat, serta pengembangan sistem berbasis *website* yang responsif dan *user-friendly*.

Penelitian ini akan melibatkan langkah-langkah sebagai berikut: pertama, pengumpulan dan pengolahan dataset batik Indonesia dengan label kelas yang tepat. Kedua, *preprocessing* data dilakukan dengan melakukan *segmentasi*, *normalisasi*, *resizing*, dan augmentasi data. Ketiga, desain arsitektur CNN akan dirancang dengan mempertimbangkan kompleksitas pola batik dan kemampuan ekstraksi fitur yang diperlukan. Keempat, model CNN akan dilatih dan divalidasi

menggunakan dataset yang disiapkan. Proses pelatihan melibatkan propagasi mundur dan optimisasi parameter. Terakhir, sistem klasifikasi batik akan diimplementasikan dalam bentuk website dengan antarmuka pengguna yang mudah digunakan.

3.1. Pengumpulan Dataset

Dataset yang digunakan pada penelitian ini adalah citra Batik Indonesia seperti yang tersaji pada Gambar 1. Data gambar tersebut diperoleh dengan mengunduh dari situs web yang telah disebut diatas.



Gambar 1. Kumpulan Dataset Citra Batik

3.2. Preprocessing Data

Proses selanjutnya adalah memanggil dataset yang telah di upload ke *google drive*. Fungsi dari pemanggilan data ini untuk dapat mengetahui berapa jumlah data training, data validation, dan data testing yang akan kita proses pada algoritma CNN dan untuk tahap *resizing* citra dengan ukuran 150x150 pixel. Kode memanggil dataset terlihat pada kode berikut:

```
[ ] train_dataset = image_dataset_from_directory(train_dir, # Memanggil Direktori Train
    shuffle=True, # Aktifasi Pengacakan
    batch_size=BATCH_SIZE, # 64 Batch Size
    label_mode="categorical", # Mode Categorical digunakan karena data kami merupakan data multiclass
    color_mode="rgb", # Mode warna RGB
    image_size=(IMG_HEIGHT, IMG_WIDTH)) # Pengaturan image size

validation_dataset = image_dataset_from_directory(validation_dir, # Memanggil Direktori Val
    shuffle=True, # Aktifasi Pengacakan
    batch_size=BATCH_SIZE, # 64 Batch Size
    label_mode="categorical", # Mode Categorical digunakan karena data kami merupakan data multiclass
    color_mode="rgb", # Mode warna RGB
    image_size=(IMG_HEIGHT, IMG_WIDTH)) # Pengaturan image size

test_dataset = image_dataset_from_directory(test_dir, # Memanggil Direktori test
    shuffle=True, # Aktifasi Pengacakan
    batch_size=BATCH_SIZE, # 64 Batch Size
    label_mode="categorical", # Label Mode menggunakan categorical karena multiclass
    color_mode="rgb", # Mode warna RGB
    image_size=(IMG_HEIGHT, IMG_WIDTH)) # Pengaturan image size
```

Gambar 2. Kode Preprocessing Data

Hasil dari baris kode pada Gambar 2 yang telah dijalankan, menghasilkan informasi yang tersaji pada Gambar 3 berikut:

```
Found 2716 files belonging to 14 classes.
Found 776 files belonging to 14 classes.
Found 388 files belonging to 14 classes.
```

Gambar 3. Hasil Preprocessing Data

Dari Gambar 3 di atas diperoleh 14 class dengan banyaknya data training sebanyak 2716 data citra, data validation sebanyak 776 data citra, dan data testing sebanyak 388 data citra.

3.3. Pelatihan dan Validasi Model

Proses training terhadap model dilakukan dengan memanggil fungsi fit() dan mengisi parameter pada fungsi fit tersebut. Parameter tersebut terdiri dari iterasi atau epoch dari pelatihan yang nantinya akan ditampilkan untuk melihat pergerakan proses pelatihan. Untuk melakukan training data, digunakan optimizer Adam (Adaptive Moment Estimator) untuk mengoptimalkan model CNN.

```
[ ] # Melakukan Compiling model menggunakan optimizer, Loss dan metrics yang sesuai

adam = tf.keras.optimizers.Adam(learning_rate = 0.0001)
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Gambar 4. Kode Compiling Model

Adam (Adaptive Moment Estimator) adalah teknik pengoptimalan yang memiliki kecepatan pembelajaran adaptif untuk setiap parameter atau bobot. Optimizer digunakan untuk mencari melalui berbagai bobot jaringan dan metrik opsional apapun yang ingin dikumpulkan dan dilaporkan selama training (pelatihan). Dalam hal ini, bisa digunakan cross entropy sebagai argumen loss. Kerugian ini untuk masalah tugas biner dan multiclass. Jenis cross entropy ini membutuhkan label untuk dikodekan sebagai kategori dan didefinisikan dalam Keras sebagai “categorical_crossentropy”.

```
# Melakukan training model

history = model.fit(
    train_data, # training data
    train_dataset, # Masukkan data yang akan di train yaitu train dataset
    validation_data=validation_dataset, # Masukkan data yang akan di validasi yaitu validation dataset
    epochs=30, # Masukkan epoch sebanyak 10 kali
    verbose=1 # Verbose
)
```

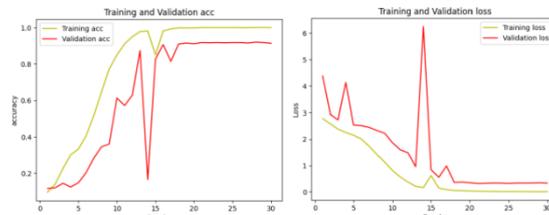
Gambar 5. Kode Training Model

Seperti yang terlihat pada baris kode pada Gambar 7 diatas, pada baris pertama melatih model dengan memanggil fungsi fit() dan ditampung pada variable history. Pada baris kedua train_dataset adalah semua data gambar dari data latih. pada baris ketiga, parameter validation_data, diisi dengan data validasi yang akan memvalidasi data latih saat proses pelatihan. Pada baris keempat, parameter epochs merupakan banyaknya iterasi pelatihan yang akan dilakukan. Pada baris terakhir yaitu verbose merupakan parameter untuk melihat proses kemajuan per epoch saat pelatihan.

```
Epoch 16/30 [-----] 181s 4s/step - loss: 0.1472 - accuracy: 0.9812 - val_loss: 0.5561 - val_accuracy: 0.9859
Epoch 17/30 [-----] 182s 4s/step - loss: 0.0897 - accuracy: 0.9923 - val_loss: 0.9849 - val_accuracy: 0.8164
Epoch 18/30 [-----] 184s 4s/step - loss: 0.0528 - accuracy: 0.9982 - val_loss: 0.3687 - val_accuracy: 0.9098
Epoch 19/30 [-----] 181s 4s/step - loss: 0.0457 - accuracy: 0.9974 - val_loss: 0.3696 - val_accuracy: 0.9149
Epoch 20/30 [-----] 183s 4s/step - loss: 0.0368 - accuracy: 0.9978 - val_loss: 0.3482 - val_accuracy: 0.9111
Epoch 21/30 [-----] 181s 4s/step - loss: 0.0258 - accuracy: 0.9996 - val_loss: 0.3188 - val_accuracy: 0.9175
Epoch 22/30 [-----] 171s 4s/step - loss: 0.0203 - accuracy: 0.9996 - val_loss: 0.3253 - val_accuracy: 0.9162
Epoch 23/30 [-----] 179s 4s/step - loss: 0.0183 - accuracy: 0.9996 - val_loss: 0.3161 - val_accuracy: 0.9175
Epoch 24/30 [-----] 188s 4s/step - loss: 0.0144 - accuracy: 0.9996 - val_loss: 0.3328 - val_accuracy: 0.9162
Epoch 25/30 [-----] 179s 4s/step - loss: 0.0146 - accuracy: 0.9996 - val_loss: 0.3211 - val_accuracy: 0.9175
Epoch 26/30 [-----] 183s 4s/step - loss: 0.0129 - accuracy: 0.9993 - val_loss: 0.3344 - val_accuracy: 0.9175
Epoch 27/30 [-----] 195s 5s/step - loss: 0.0127 - accuracy: 0.9993 - val_loss: 0.3354 - val_accuracy: 0.9149
Epoch 28/30 [-----] 277s 6s/step - loss: 0.0097 - accuracy: 1.0000 - val_loss: 0.3359 - val_accuracy: 0.9281
Epoch 29/30 [-----] 258s 6s/step - loss: 0.0075 - accuracy: 1.0000 - val_loss: 0.3458 - val_accuracy: 0.9175
Epoch 30/30 [-----] 241s 6s/step - loss: 0.0095 - accuracy: 0.9996 - val_loss: 0.3325 - val_accuracy: 0.9137
```

Gambar 6. Hasil Training Model CNN

Terlihat pada baris kode pada Gambar 8 bahwa iterasi/epoch yang digunakan yaitu sebanyak 30. Penjelasan nilai loss, accuracy, val_loss, dan val_accuracy ditampilkan pada Gambar 9 berikut:



Gambar 7. (Training dan Validation) Accuracy dan Loss

3.4. Tampilan User Interface

3.4.1. Tampilan Home

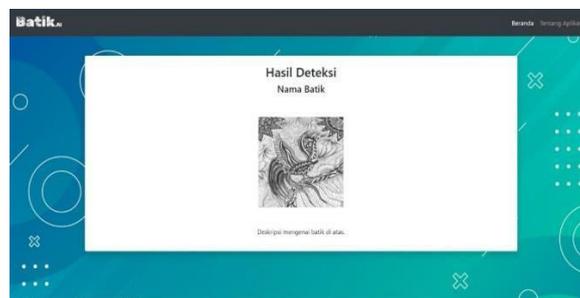
Tampilan pada Gambar 10 merupakan tampilan utama pada website berisi tentang kolom Unggah gambar atau foto motif batik yang ingin dideteksi, dan tombol deteksi untuk melanjutkan proses deteksi setelah berhasil mengunggah motif batik yang ingin dicari.



Gambar 8. Tampilan Home

3.4.2. Tampilan Hasil

Tampilan pada Gambar 11 merupakan tampilan yang menampilkan hasil deteksi oleh Artificial Intelligence yang sudah dilatih untuk mengklasifikasikan motif motif batik yang ada di Indonesia, pada tampilan ini berisi tentang Nama batik yang berhasil dideteksi, Gambar motif batik tersebut maupun deskripsi mengenai motif batik tersebut.



Gambar 9. Tampilan Hasil

3.4.3. Tampilan Tentang

Pada tampilan pada Gambar 12 ditampilkan hal hal tentang website Batik.AI ini dan juga siapa saja yang terlibat dalam pembuatan website Batik.Ai ini



Gambar 10. Tampilan Tentang

4. KESIMPULAN

Berdasarkan analisis dan hasil pembuatan serta pengujian model deep learning dengan algoritma CNN dalam penelitian ini, dapat disimpulkan bahwa penelitian ini berhasil menghasilkan model deep learning yang mengimplementasikan metode CNN untuk mengklasifikasikan 14 jenis citra batik. Arsitektur CNN terbaik yang ditemukan melalui perbandingan parameter- parameter tertentu mencakup ukuran citra 150 x 150 piksel, batch size 64, ukuran filter atau kernel 3x3, learning rate 0,0001, jenis optimizer Adam, fungsi aktivasi ReLu dan Softmax, epoch 30, dan skenario pembagian dataset 70%: 20%: 10% dari total 3.880 data citra dengan jenis citra RGB (berwarna). Hasil pengujian menunjukkan tingkat akurasi sebesar 91,24% pada data testing menggunakan model arsitektur terbaik tersebut.

5. SARAN

Untuk penelitian selanjutnya, disarankan untuk menggabungkan teknik pembelajaran *machine learning* lainnya dengan model CNN untuk dapat meningkatkan akurasi klasifikasi batik Indonesia. Penelitian dapat berfokus pada *transfer learning* atau model *hybrid* dan meningkatkan jumlah dataset batik.

DAFTAR PUSTAKA

- [1] F. Hasyim, K. Malik, F. Rizal, and Yudistira, "Implementasi algoritma convolutional neural networks (cnn) untuk klasifikasi batik," *COREAI: Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi*, vol. 2, no. 2, pp. 40–47, 2022, doi: 10.33650/coreai.v2i2.3365.
- [2] L. Hakim, H. R. Rahmanto, S. P. Kristanto, and D. Yusuf, "Klasifikasi citra motif batik banyuwangi menggunakan convolutional neural network," *Jurnal Teknoinfo*, vol. 17, no. 1, p. 203, 2023, doi: 10.33365/jti.v17i1.2342.
- [3] F. N. Cahya, N. Hardi, D. Riana, and S. Hadianti, "Klasifikasi penyakit mata menggunakan convolutional neural network (cnn)," vol. 10, pp. 618–626, 2021.
- [4] N. P. Sutramiani, I. K. G. D. Putra, and M. Sudarma, "Local adaptive thresholding pada preprocessing citra lontar aksara bali," vol. 14, no. 1, pp. 27–30, 2015.
- [5] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. doi: 10.1038/nature14539.
- [6] F. U. Larasati, N. Aini, and A. H. S. Irianti, "Proses pembuatan batik tulis remekan di kecamatan ngantang," *Prosiding Pendidikan Tata Boga Busana*, p. 8, 2021.
- [7] Y. Heryadi and T. Wahyono, *Dasar-dasar deep learning dan implementasi*. Yogyakarta: GAVA MEDIA, 2021.

- [8] A. Hibatullah and I. Maliki, "Penerapan metode convolutional neural network pada pengenalan pola citra sandi rumput," *Journal of Informatics and Computer Science*, vol. 1, no. 02, pp. 1–8, 2019.
- [9] I. W. Suartika E. P, A. Y. Wijaya, and R. Soelaiman, "Klasifikasi citra menggunakan convolutional neural network (cnn) pada caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, p. 76, 2016.
- [10] J. Wen *et al.*, "Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation," *Med Image Anal*, vol. 63, 2020, doi: 10.1016/j.media.2020.101694.
- [11] G. Shrestha, Deepsikha, M. Das, and N. Dey, "Plant Disease Detection Using CNN," *Proceedings of 2020 IEEE Applied Signal Processing Conference, ASPCON 2020*, pp. 109–113, 2020, doi: 10.1109/ASPCON49795.2020.9276722.
- [12] P. Deepalakshmi, K. T. Prudhvi, C. S. Siri, K. Lavanya, and P. N. Srinivasu, "Plant leaf disease detection using CNN algorithm," *International Journal of Information System Modeling and Design*, vol. 12, no. 1, pp. 1–21, 2021, doi: 10.4018/IJISMD.2021010101.
- [13] J. Park, J. Chen, Y. K. Cho, D. Y. Kang, and B. J. Son, "CNN-based person detection using infrared images for night-time intrusion warning systems," *Sensors (Switzerland)*, vol. 20, no. 1, 2020, doi: 10.3390/s20010034.
- [14] L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek deteksi makanan khas Palembang menggunakan algoritma yolo (you only look once)," *Jurnal Nasional Ilmu Komputer*, vol. 2, no. 3, pp. 213–232, 2021, doi: 10.47747/jurnalnik.v2i3.534.
- [15] E. Retnoningsih and R. Pramudita, "Mengenal machine learning dengan teknik supervised dan unsupervised learning menggunakan python," *BINA INSANI ICT JOURNAL*, vol. 7, no. 2, pp. 156–165, 2020.