



Implementasi Metode CNN *Computer Vision* Dalam Identifikasi Tipe Kerusakan Pohon Berbasis FHM

Zuhri Nopriyanto¹, Rico Andrian², Rahmat Safei^{3*},
Kurnia Muludi⁴

^{1,2,4}Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung,

³Program Studi Magister Kehutanan, Fakultas Pertanian, Universitas Lampung,

Jl. Prof. Dr. Ir. Soemantri Brojonegoro No.1, Bandar Lampung,

Provinsi Lampung, Indonesia 35145

*Email Penulis Koresponden: rahmat.safei@fp.unila.ac.id

Abstrak:

Identifikasi tipe kerusakan pohon pada *Forest Health Monitoring* hingga saat ini masih bersifat manual, yaitu menggunakan penglihatan manusia dalam penerapannya. Teknologi Informasi yang kini berkembang pesat dapat dirasakan hingga ke berbagai media penerapan ilmu pengetahuan, dengan demikian terciptalah salah satu solusi dalam memecahkan masalah penelitian kasus identifikasi tipe kerusakan pohon yaitu dengan penggunaan metode *computer vision* dalam upaya memudahkan pekerjaan dalam ilmu kehutanan. Tujuan penelitian ini adalah untuk menerapkan *computer vision* dalam mengidentifikasi tipe kerusakan pohon berbasis *Forest Health Monitoring*. Tahapan penelitian yang dilakukan dalam penelitian ini adalah pengumpulan *dataset*, proses *preprocessing*, pembagian *dataset*, pelatihan model, pengujian model dan terakhir adalah evaluasi model. Hasil penelitian ini berupa model (*prototype*) identifikasi tipe kerusakan pohon dalam 4 kategori yaitu, *LeNet-5 Colab*, *LeNet-5 Tesla*, *MobileNet Colab*, dan *MobileNet Tesla*. Persentase identifikasi model bervariasi, dimana pada kelas tertentu model dapat mengidentifikasi dengan benar dan dikelas lainnya masih ada beberapa identifikasi model yang kurang optimal, disebabkan oleh kemiripan beberapa bentuk *dataset* dalam segi visual komputer. Penelitian penerapan *computer vision* dalam identifikasi kerusakan pohon berbasis *Forest Health Monitoring* berhasil dilakukan dengan menghasilkan dua model (*prototype*) dalam identifikasi kerusakan pohon yang nilai akurasi mencapai angka 89.99% pada model *LeNet-5* dan 99.06% pada model *MobileNet* dengan *tools* yang digunakan adalah *Jupyter notebook* pada *Nvidia Tesla K20 (offline)* dan *Google Colab (online)*.

Keywords:

Forest Health Monitoring;
Computer Vision;
Kerusakan Pohon;
Convolutional Neural Network;

Riwayat Artikel:

Diserahkan Juli 16 2022
Direvisi 17 Agustus 2022
Diterima 07 September 2022
Dipublikasi 30 April 2023

DOI:

10.22441/incomtech.v13i1.16022

1. PENDAHULUAN

Pohon merupakan tumbuhan menjulang keatas dengan tubuh utama berkayu dan menghasilkan oksigen yang merupakan bagian dari ekosistem. Fungsi pohon sendiri beragam selain menghasilkan oksigen pohon berperan dalam menanggulangi bencana alam seperti banjir, habitat satwa dan lainnya. Pohon dikatakan penting dikarenakan keberadaannya yang menopang dalam penjagaan siklus global, dengan demikian kesehatan pohon perlu dijaga dan kerusakannya pula penting untuk dipahami.

Kerusakan pada pohon adalah tolak ukur untuk mengetahui pohon dikatakan sehat atau rusak, selain menghilangkan fungsional penting pohon bagi kehidupan, kerusakan pohon dapat merugikan sektor perekonomian [1]. Pohon yang rusak biasanya tergambarkan oleh kondisi fisiknya dan dapat diamati secara visual. Kondisi-kondisi ini dapat ditemukan sesuai lokasi yang ada di tubuh pohon dari akar hingga ke atas pada bagian tajuk [2]. Kondisi-kondisi tersebut menunjukkan bahwa pohon telah dalam kondisi rusak atau tidak sehat, dengan demikian diperlukan terapan dalam mengetahui kerusakan yang dialami pohon dalam tingkat akurasi yang tinggi agar hasil deteksi akurat dan tepat yang membuat *computer vision* cocok dalam penyelesaian masalah pada penelitian ini.

Metode *computer vision* membutuhkan spesifikasi dalam mewujudkan pengidentifikasian yaitu perolehan data, proses data, analisis serta pemahaman gambar secara umum untuk menghasilkan angka dan simbol informasi dari dunia nyata. Tingkat akurasi yang tinggi dalam identifikasi merupakan salah satu keunggulan metode *computer vision*, selain itu didukung dengan luasnya ruang lingkup dan harga yang terjangkau mendukung dalam mengenali dan mendeteksi objek. Penelitian *computer vision* ini menggunakan algoritma *Convolutional Neural Network* (CNN), karena CNN dikenal sebagai metode yang efisien dan mengalami perkembangan yang pesat dalam beberapa dekade terakhir dalam hal pengenalan pola [3]. CNN biasanya terdiri dari beberapa partisi yaitu *fully-connected layer*, *softmax layer*, dan lapisan ekstraksi fitur yang terdiri dari lapisan *convolutional* dan lapisan *pooling* [4]. Arsitektur yang digunakan dalam penelitian ini adalah *LeNet-5* dan *MobileNet*.

Penelitian terdahulu bermetodekan *Forest Health Monitoring* (FHM) yang diperkenalkan oleh *United States Department of Agriculture (USDA) Forest Service*, dimana metode ini digunakan dalam memonitoring kesehatan hutan dan mengidentifikasinya [5]. Penerapannya metode ini sulit pada hutan tropika, dan membutuhkan banyak sumber daya manusia untuk mendapatkan data status keakuratan yang tinggi. Permasalahan tersebut dapat dipecahkan oleh metode *computer vision* dimana dalam mendapatkan tingkat keakuratan yang tinggi tidak diperlukan sumber daya manusia yang banyak, melainkan karakteristik data akan menentukan data status keakuratan dalam penelitian, selain itu *computer vision* sebagai bagian teknologi dalam pendeteksian memiliki beberapa karakteristik penting yang diantaranya adalah presisi nya yang tinggi [6].

Penelitian terdahulu yang dilakukan adalah penggunaan metode *computer vision* dalam identifikasi jenis tanaman secara otomatis, telah dilakukan dan teruji serta mendapatkan hasil yang cukup baik. Hasil dari penelitian tersebut menggunakan

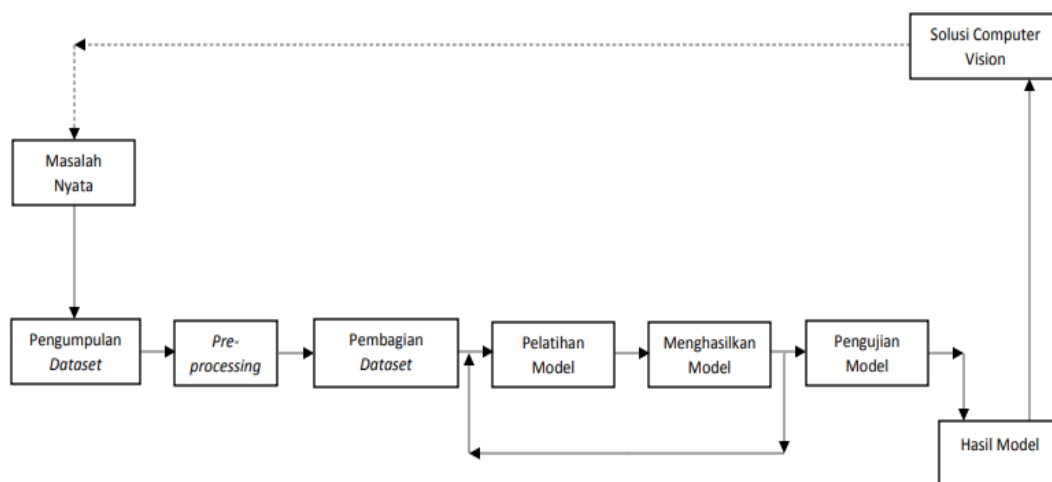
gambar yang berlabel serta *dataset* yang besar dalam pengoperasian sistem, pada tingkatan tersebut menunjukkan tingkat akurasi mencapai 85.1% dalam waktu 5.4 detik per spesies tanaman. Tahapan penelitian yang terjadi adalah *classifying, Segmenting, Extracting, Comparing*. Implementasi sistem sudah berbentuk aplikasi *smartphone* dan telah diuji coba pada sistem operasi iOS dan dilengkapi juga dengan koordinat GPS dan stempel waktu untuk setiap foto [7].

Penelitian tentang metode *computer vision* juga terjadi di Mexico dengan studi kasus pendeteksi kerusakan pada aspal menggunakan berbasis *Deep Neural Network*. Performa *deep learning* telah teruji dan menunjukkan kinerja yang luar biasa dalam hal pemrosesan gambar, dan perihal forensik digital pada gambar [8]. *Dataset* atau sampel yang diambil berasal dari dua negara yaitu Italia dan Mexico, meliputi jalan-jalan aspal di kedua negara tersebut. Model dilatih agar dapat mengidentifikasi dan mengklasifikasikan kerusakan aspal tersebut dengan *Digital Asset Management Tool*, berdasarkan *real-time* video maupun data gambar. Tujuan dalam penelitian ini adalah dalam membantu Lembaga Pemerintahan dalam pemeliharaan infrastruktur yang lebih prediktif [9].

Penelitian ini bertujuan untuk mengimplementasikan *Computer vision* dalam upaya mengidentifikasi tipe kerusakan pohon berbasis *Forest Health Monitoring (FHM)* dengan menerapkan metode *Convolutional Neural Network (CNN)* menggunakan dua arsitektur yang berbeda yaitu *LeNet-5* dan *MobileNet* dengan bahasa pemrograman *python*.

2. METODE PENELITIAN

Berikut merupakan langkah-langkah yang dilakukan dalam identifikasi kerusakan pohon diperlihatkan pada Gambar 1.



Gambar 1. Tahap Penelitian.

2.1. Dataset

Tahapan dalam pelaksanaan penelitian pertama kali ialah mendapatkan *dataset*, tujuannya agar membuat model dapat mengenal informasi yang ada pada gambar. *Dataset* berupa foto yang diambil dari beberapa sampel pohon yang terindikasi mengalami kerusakan, foto untuk dataset diupayakan jelas dan jernih agar ciri-ciri

dari kerusakan pohon tersebut dapat ditemukan untuk pelatihan model. Penerapan *computer vision* selalu identik dengan adanya kamera, dikarenakan *computer vision* sendiri adalah algoritma untuk memperoleh pemahaman dari gambar dan cara ini telah digunakan selama bertahun-tahun. [10]. Pada metode *computer vision*, kualitas keluaran (*output*) ditentukan dari masukkan (*input*) *dataset* yang diterapkan [11].

Dataset yang berhasil terkumpul berjumlah 1600 citra gambar dari 16 kelas berbeda yang tiap-tiap kelas terdiri dari 100 citra gambar. *Over-fitting* dapat terjadi apabila *dataset* yang diolah berukuran besar ataupun banyak [12]. Pengambilan citra *image* untuk dijadikan *dataset* menggunakan kamera, dikarenakan *computer vision* dapat menghasilkan solusi navigasi yang lebih baik apabila digabungkan oleh sensor lain [13]. Kamera *Canon EOS 250D with CMOS* berperan penting dalam pengambilan *dataset* yang dilakukan di Taman Hutan Raya Wan Abdul Rachman (Tahura WAR).

Tabel 1. Jumlah *Dataset*

Kode	Tipe Kerusakan	Jumlah Citra
01	Kanker	100
02	Konk	100
03	Luka Terbuka	100
04	Resinosis atau Gumosis	100
05	Batang Pecah	100
06	Sarang Rayap	100
11	Batang atau Akar Patah	100
12	Brum Akar atau Batang	100
13	Akar Patah atau Mati	100
20	Liana	100
21	Mati Pucuk	100
22	Cabang Patah	100
23	Brum Berlebihan	100
24	Daun Pucuk Tunas Rusak	100
25	Daun Berubah Warna	100
31	Dan Lain-lain	100
Total:		1600

2.2. Pengolahan Gambar (*Preprocessing*)

Penelitian ini menerapkan pengolahan gambar terlebih dahulu sebelum *dataset* masuk pada fase *processing*, yaitu *scaling*. Proses *scaling* merupakan proses merubah ukuran citra *dataset* tanpa menghilangkan objek penting ke dalam resolusi yang berbeda [14]. Citra *dataset* yang di-*input*-kan diubah resolusinya menjadi 224x224 *pixel* tanpa menghilangkan ciri-ciri dari objek *dataset* dan dapat memudahkan klasifikasi, proses *scaling* citra *dataset* diperlihatkan pada gambar 2.



Gambar 2. *Scaling (Preprocessing)*

2.3. Pembagian *Dataset*

Dataset dibagi menggunakan *library scikit learn* dengan fungsi *train_test_split*, hal ini diperlukan untuk membuat porsi masing-masing dalam data latihan, data validasi, dan data uji yang sesuai.

- a) Variasi data latih (*Training dataset*)
70% *dataset* dengan jumlah 1120 citra gambar digunakan dalam proses *training* model. *Training* model sendiri merupakan proses melatih model agar dapat mengenali ciri-ciri pada citra gambar.
- b) Variasi data validasi (*Validation dataset*)
10% *dataset* dengan jumlah 160 citra gambar digunakan dalam proses *validation* model. *Validation* model sendiri merupakan proses validasi model dan dapat mencegah *over-fitting*.
- c) Variasi data test (*Testing dataset*)
20% *dataset* dengan jumlah 320 citra gambar digunakan dalam proses *testing* model. *Testing* model sendiri merupakan proses pengujian model yang telah melalui *training* dan validasi.

2.4. Pelatihan Model *LeNet-5* & *MobileNet*

Nilai pada *hyperparameter* harus ditentukan terlebih dahulu sebelum digunakan pada pelatihan dalam menyusun model. Nilai pada *batch size* ditentukan dari nilai data latih yang digunakan, nilai *batch* atau kelompok terdiri dari beberapa data sampel yang isinya merupakan data latih (*training dataset*). Dalam hal ini, terjadi proses iterasi akan membuat model mengingat data dan mempelajari data sampel. Proses iterasi sendiri adalah proses yang terjadi secara *forward* (*input* ke *output*) dan dari arah *backward* (*output* ke *input*) dalam hal pembelajaran model terhadap sampel. Nilai bobot pada kelas tipe kerusakan pohon ditentukan oleh proses pembelajaran iterasi ini, untuk mendapatkan nilai bobot yang tepat dilakukan proses penentuan bobot berulang-ulang dalam menentukan kelas tipe kerusakan pohon.

Arsitektur yang dipergunakan dalam penelitian ini adalah *LeNet-5* dan *MobileNet*. *LeNet-5* diperkenalkan oleh Yan LeCun untuk pertama kalinya, dimana

arsitektur ini berupa jaringan yang mempunyai *layer* (lapisan) berkonsep pada *Convolutional Neural Network* (CNN). Model arsitektur *LeNet-5* terkenal karena kemampuannya untuk melakukan operasi yang lebih cepat daripada model arsitektur lainnya [15]. *MobileNet* merupakan arsitektur model yang didesain untuk perangkat *mobile*, model ini dikembangkan secara kompleksitas dengan *Depthwise Separable Convolution* [16]. Dalam memisahkan konvolusi secara konvensional, arsitektur *MobileNet* menggunakan *depthwise separable* konvolusi sehingga menghasilkan konvolusi *pointwise* dan *depthwise* [17]. Model *MobileNet* mampu mempercepat penelitian berbagai bidang, dikarenakan telah mengalami peningkatan dalam beberapa variasi versinya dan telah teruji pada *tensorflow* [18].

Pelatihan model menggunakan metode *Convolutional Neural Network* (CNN) dengan *dataset* citra kerusakan pohon beresolusi 224×224 *pixel* dengan total 1600 gambar. Metode dan penerapan yang dipergunakan sama tetapi tools yang digunakan berbeda yaitu penggunaan *tools* yang bersifat *offline* dan *tools* yang bersifat *online*.

Perbedaan dalam metode berada pada penggunaan arsitektur yaitu *LeNet-5* dan *MobileNet*. Model dilatih menggunakan *Jupyter Notebook* dan *Google Colab* sebagai *tools* serta *hyperparameter* dibuat sama. Berikut tabel penjelasan nama dan nilai *hyperparameter* pada model:

Tabel 2. *Hyperparameter*

No	Nama Parameter	Nilai
1	<i>Input sizes</i>	224x224x3
2	<i>Epoch</i>	10,30,50,80
3	<i>Batch size</i>	32
4	<i>Number of Classes</i>	16
5	<i>Optimizer</i>	Adam
6	<i>Learning rate</i>	0,001

2.5. Pengujian Model

Tahapan ini yaitu melakukan pengujian dari perolehan data uji yang ada terhadap model arsitektur *LeNet-5* maupun *MobileNet*.

2.6. Evaluasi Hasil

Tahapan ini yaitu menyajikan hasil dari tahapan sebelumnya yaitu pengujian model, hasil akan dicatat untuk menghitung keberhasilan model. Hasil pengujian model tersebut akan menampilkan persentase akurasi keberhasilan dari pengujian 2 model arsitektur yang berbeda dan menampilkan akurasi tertinggi per tiap model arsitektur.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Model

Penelitian telah berhasil dilakukan, dalam hal ini algoritma yang digunakan adalah *Convolutional Neural Network* (CNN). Proses identifikasi tipe kerusakan pohon dilakukan dengan menggunakan dua arsitektur yaitu *LeNet-5* dan *MobileNet*, dalam percobaan kedua arsitektur tersebut menggunakan dua *tools device* yaitu *Google Colab (online)* dan *Jupyter Notebook (offline)* pada *Nvidia Tesla K20*. Proses identifikasi menghasilkan 4 kategori data yaitu, *LeNet-5 Google Colab*, *LeNet-5 Tesla*, *MobileNet Colab*, dan *MobileNet pada Tesla*.

Hasil akurasi model percobaan menggunakan arsitektur *LeNet-5* dengan *MobileNet* berbeda. Perbedaan ini disebabkan oleh lapisan atau (*layer*) yang dipergunakan pada masing-masing arsitektur. *Layer* pada percobaan *LeNet-5* hanya menggunakan 7 lapis, hal ini membuat proses komputasi tidak banyak memakan waktu seperti proses komputasi yang ada pada *MobileNet*. *Layer* pada percobaan *MobileNet* menggunakan 13 lapisan, tetapi lapisan ini memiliki sub-sub lapisan lagi yang membuat proses komputasi berjalan lebih lama dari percobaan *LeNet-5*. Keuntungan pada penggunaan lapisan yang banyak adalah hasil akhirnya berupa persentase akurasi yang lebih tinggi daripada tingkat akurasi dengan lapisan yang lebih sedikit

Tabel 3. Tingkat Akurasi 4 Kategori Model

Model	Epoch 10		Epoch 30		Epoch 50		Epoch 80	
	Train	Valid	Train	Valid	Train	Valid	Train	Valid
<i>LeNet-5 Colab</i>	1.0000	0.8531	1.0000	0.8813	1.0000	0.8844	1.0000	0.8999
<i>LeNet-5 Tesla</i>	1.0000	0.8125	1.0000	0.8999	1.0000	0.8969	1.0000	0.8969
<i>MobileNet Colab</i>	0.7232	0.6719	0.9098	0.8969	1.0000	0.9837	0.9714	0.9531
<i>MobileNet Tesla</i>	0.8821	0.8094	1.0000	0.9781	1.0000	0.9906	1.0000	0.9844

Tabel 4. Waktu *Training* 4 Kategori Model

Model	Epoch 10	Epoch 30	Epoch 50	Epoch 80
	Waktu (s)	Waktu (s)	Waktu (s)	Waktu (s)
<i>LeNet-5 Tesla</i>	32	91	151	242
<i>MobileNet Tesla</i>	133	392	654	1042
<i>LeNet-5 Colab</i>	641	1924	3205	5122
<i>MobileNet Colab</i>	2209	7152	11570	18722

Berdasarkan Tabel 3, pemodelan dengan menggunakan arsitektur *MobileNet* mempunyai hasil akurasi lebih tinggi dibandingkan model arsitektur *LeNet-5* selain itu berdasarkan Tabel 4, terdapat perbandingan waktu *training* atau lamanya pelatihan model pada *Tesla* dan *Google Colab* yang dapat dilihat bahwa *Google Colab* membutuhkan proses pelatihan model yang lebih lama dibandingkan *Tesla*.

Data yang tertera pada Tabel 3, menunjukkan bahwa pada percobaan *MobileNet* nilai akurasi pada *epoch 50* lebih besar dibandingkan *epoch 80*. Hal tersebut membuktikan bahwa makin tinggi nilai *epoch* tidak menjamin model yang dihasilkan semakin baik, dalam kasus ini dibuktikan bahwa nilai *epoch* tidak berbanding lurus dengan nilai tingkat akurasi yang dihasilkan. Proses komputasi

yang tidak sempurna menyebabkan hal dapat ini terjadi, salah satu solusinya yaitu dengan mengubah kombinasi *hyperparameter* yang dipergunakan dalam percobaan pelatihan model.

3.2. Pembahasan Model

Hasil penelitian ini berupa model pada 2 arsitektur yang digunakan beserta *tools* nya yaitu arsitektur *LeNet-5 Google Colab* dengan akurasi 89.99% dan arsitektur *MobileNet Jupyter Notebook* dengan akurasi 99.06%.

Tabel 5. Hasil Akurasi Tertinggi (*Testing*)

Model	Epoch	Akurasi
<i>LeNet-5 Colab</i>	80	89.99%
<i>MobileNet Tesla</i>	50	99.06%

Berdasarkan Tabel 6, diperlihatkan nilai *precision*, *recall* dan *accuracy* serta *F1 score* yang ada dalam model arsitektur *LeNet-5 Google Colab* dengan akurasi 89.99% di *epoch* 80 pada percobaan CNN, dimana data tabel diperlihatkan sebagai berikut.

Tabel 6. *Precision*, *Recall*, *Accuracy*, dan *F1 Score*

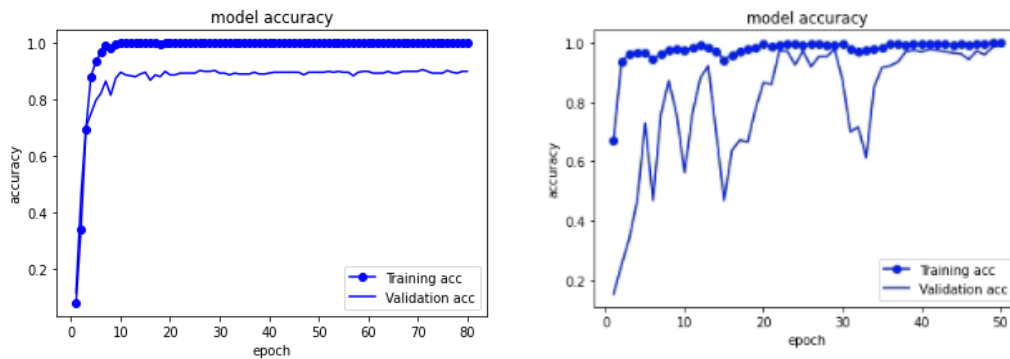
Kode	Tipe Kerusakan	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Support</i>
01	Kanker	94%	94%	94%	18
02	Konk	89%	89%	89%	19
03	Luka Terbuka	83%	83%	83%	24
04	Resinosis atau Gumosis	100%	85%	92%	26
05	Batang Pecah	83%	83%	83%	18
06	Sarang Rayap	82%	82%	82%	22
11	Batang atau Akar Patah	92%	96%	94%	23
12	Brum Akar atau Batang	71%	94%	81%	16
13	Akar Patah atau Mati	100%	100%	100%	21
20	Liana	75%	64%	69%	14
21	Mati Pucuk	100%	95%	97%	19
22	Cabang Patah	100%	100%	100%	15
23	Brum Berlebihan	95%	83%	88%	23
24	Daun Pucuk Tunas Rusak	88%	100%	94%	23
25	Daun Berubah Warna	87%	91%	89%	22
31	Dan Lain-lain	100%	100%	100%	17
<i>Accuracy</i>				90%	320

Berdasarkan Tabel 7, diperlihatkan nilai *precision*, *recall* dan *accuracy* serta *F1 score* yang ada dalam model arsitektur *MobileNet Nvidia Tesla K20* dengan akurasi 99.06% di *epoch* 50 pada percobaan CNN, dimana data tabel diperlihatkan sebagai berikut.

Tabel 7. *Precision*, *Recall*, *Accuracy*, dan *F1 Score*

Kode	Tipe Kerusakan	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Support</i>
01	Kanker	100%	100%	100%	18
02	Konk	100%	100%	100%	19
03	Luka Terbuka	100%	100%	100%	24
04	Resinosis atau Gumosis	100%	100%	100%	26
05	Batang Pecah	100%	100%	100%	18
06	Sarang Rayap	96%	100%	98%	22
11	Batang atau Akar Patah	100%	96%	98%	23
12	Brum Akar atau Batang	100%	100%	100%	16
13	Akar Patah atau Mati	100%	100%	100%	21
20	Liana	100%	95%	97%	14
21	Mati Pucuk	100%	100%	100%	19
22	Cabang Patah	96%	96%	96%	15
23	Brum Berlebihan	96%	100%	98%	23
24	Daun Pucuk Tunas Rusak	100%	100%	100%	23
25	Daun Berubah Warna	100%	100%	100%	22
31	Dan Lain-lain	100%	100%	100%	17
<i>Accuracy</i>				99%	320

Berdasarkan Gambar 3, diperlihatkan perbandingan kinerja model antara arsitektur *LeNet-5 Google Colab* dengan akurasi 89.99% (kiri) dan arsitektur *MobileNet* dengan akurasi 99.06% (kanan), dimana data gambar diperlihatkan sebagai berikut.



Gambar 3. Perbandingan Kinerja Model

4. KESIMPULAN

Penelitian berhasil dilakukan, dapat ditarik kesimpulan bahwa algoritma *Convolutional Neural Network* (CNN) mampu sebagai bagian *computer vision* untuk mengidentifikasi tipe kerusakan pohon. Persentase tingkat akurasi *MobileNet* mencapai 99.06% diikuti dengan persentase tingkat akurasi *LeNet-5* yang mencapai 89.99%. Waktu yang dibutuhkan dalam *training dataset* pada *Nvidia Tesla K20 (offline)* lebih efisien dibandingkan penggunaan *Google Colab (online)* yang dijabarkan urutannya sebagai berikut: *LeNet-5 Tesla*, *MobileNet Tesla*, *LeNet-5 Google Colab*, dan *MobileNet Google Colab*.

ACKNOWLEDGEMENTS

The experiment in this research used the NVIDIA Tesla K80 and Tesla K20 provided by Department of Computer Science, University of Lampung.

REFERENSI

- [1] B Abimanyu, R. S, "Application of Forest Health Monitoring Method in Assessing Tree Damage in Metro Urban Forests", Bandar Lampung: *Jurnal Sylvia Lestari*, 2019.
- [2] Safe'i, R, "Kajian Kesehatan Hutan dalam Pengelolaan Hutan Rakyat di Provinsi Lampung", Bogor: Institut Pertanian Bogor, 2015.
- [3] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and W. Wilson, "CNN architectures for large-scale audio classification," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Mar. 2017, pp.131–135.
- [4] Y. Wang, J. Yan, Q. Sun, Z. Yang "A MobileNets Convolutional Neural Network for GIS Partial Discharge Pattern Recognition in the Ubiquitous Power Internet of Things Context: Optimization, Comparison, and Application" in *IEEE Digital Object Identifier*, Vol 7, Oct 2019, doi: 10.1109/ACCESS.2019.2946662.
- [5] S. A. Alexander, "Forest Health Monitoring: Field Methods Guide", Las Vegas (US): Environmental Monitoring Systems, 1995.
- [6] Y. Feng, T. Yang and Y. Niu, "Subpixel Computer Vision Detection Based on Wavelet Transform," in *IEEE Digital Object Identifier*, Vol 8, May 2020, doi: 10.1109/ACCESS.2020.2991846.
- [7] K. Neeraj, P. N., "Leafsnap: A Computer Vision System for Automatic Plant Species Identification", Berlin: Springer-Verlag, 2012.
- [8] K. B. Meena and V. Tyagi, "A deep learning-based method for image splicing detection," in *Proc. J. Phys., Conf.*, vol. 1714, no. 1, Art. no. 012038, 2021.
- [9] A. Angulo, V. F., "Road Damage Detection Acquisition System Based on Deep Neural Network for Physical Asset Management", Mexico: MICAI, 2019.
- [10] L. Ruotsalainen, A. M. M. Makela, J. Rantanen, and N. Sokolova, "Improving Computer Vision-Based Perception for Collaborative Indoor Navigation," in *IEEE SENSORS JOURNAL VOL 22 NO 6*, 15 March, 2022.
- [11] Babatunde, Hezekiah, O., Armstrong, L., Leng, J., & Diepeveen, D., "A Survey of Computer-Based Vision Systems for Automatic Identification of Plant Species". *Journal of Agricultural Informatics*, 61-71. 2015
- [12] Md. R. Islam, A. Matin, "Detection of COVID 19 from CT image by the novel Lenet-5 CNN Architecture", *23rd ICCIT (International Conference on Computer and Information Technology)*, pp-19-21, Dec 2020.
- [13] N. Kronenwett and G. F. Trommer, "Multi sensor pedestrian navigation system for indoor and outdoor environments," in *Proc. DGON Inertial Sensors Syst. (ISS)*, pp. 1–21, Sep. 2019, doi:10.1109/ISS46986.2019.8943692.]
- [14] Z. Arham, & W, N. I. "Pembangunan Virtual Mirror Eyeglasses Menggunakan Teknologi

- Augmented Reality”. *Komputa: Jurnal Ilmiah Komputer Dan Informatika*, 1(2), 79–84, 2012.
- [15] R.A. Hazarika, A. Abraham, D. Kandar and A.K. Maji, “An Improved LeNet-Deep Neural Network Model for Alzheimer’s Disease Classification Using Brain Magnetic Resonance Images,” in *IEEE Digital Object Identifier*, Vol 8, Nov 2021, doi: 10.1109/ACCESS.2021.3131741.
- [16] M. Alwanda, R. Ramadhan, and D. Alamsyah, “Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle, *Algoritme vol 1*, 2020.
- [17] K. Kadam, S. Ahirrao, K.Kotecha, S. Sahu “Detection and Localization of Multiple Image Splicing Using MobileNet V1”, in *IEEE Digital Object Identifier*, Vol 9, Nov 2021.
- [18] El Fatimi E, Eryigit R, Elfatimi L, “Beans Leaf Diseases Classification Using MobileNets Models”, *IEEE Digital Object Identifier*, Vol 10, Jan 2022.