Penerapan Convolutional Neural Network terhadap Citra *Grayscale* dan *Pseudocolor* RGB untuk Klasifikasi *Malware*

Chris Christian[#1], Ventje Jeremias Lewi Engel[#2]

[#]*Program Studi Informatika, Institut Teknologi Harapan Bangsa*
Jalan Dipatiukur No. 80-84, Bandung, Indonesia 40132

[1]cchristian873@gmail.com

[2]ventje@ithb.ac.id

Abstract*— Malware or malicious software is any software intentionally made to harm the function of a computer system. In this internet era, there has been a significant increase in the volume of malware in recent years. Therefore, it is essential to develop a system in order to identify and classify malware samples. This research conducts experiments to classify malware using grayscale and RGB image visualization with Convolutional Neural Network (CNN) method. The dataset is visualized grayscale images which are converted from executable files and consists of 9339 images and 25 families. Whereas, the RGB images are obtained by converting grayscale images into color images using the pseudocolor technique. Before malware images are classified, Principal Component Analysis (PCA) is utilized to handle high dimensional data and improve the classification accuracy by extracting important features. Experimental results show that the Convolutional Neural Network model achieves the highest accuracy 97.591% and the highest recall 95.026%. As well, for the RGB images, the Convolutional Neural Network model obtains the best accuracy 97.591% and the best recall 94.824%. Overall, malware classification with both types of images delivers quite good results.

Keywords*— Cybersecurity, malware, image classification, Convolutional Neural Network (CNN).

Abstrak*— *Malware* atau *malicious software* merupakan perangkat lunak yang sengaja dibuat untuk merusak fungsionalitas sistem komputer. Pada era internet saat ini, terdapat peningkatan yang signifikan pada jumlah *malware* dalam tahun-tahun belakangan. Oleh karena itu, diperlukan sistem yang dapat mengidentifikasi dan mengklasifikasikan sampel *malware*. Penelitian ini melakukan eksperimen untuk melakukan klasifikasi *malware* menggunakan visualisasi citra *grayscale* dan citra RGB dengan metode Convolutional Neural Network (CNN). *Dataset* yang digunakan berupa visualisasi citra *grayscale* yang dikonversi dari *file executable*, dan terdiri dari 9339 citra dan 25 *family*. Sedangkan, citra RGB diperoleh dari konversi citra *grayscale* menjadi citra berwarna dengan teknik *pseudocolor*. Sebelum citra *malware* diklasifikasi, Principal Component Analysis (PCA) digunakan untuk menangani tingginya dimensi data dan meningkatkan akurasi klasifikasi dengan mengekstrak fitur-fitur penting. Hasil penelitian menunjukkan model Convolutional Neural Network mencapai akurasi tertinggi 97,752% dan *recall* tertinggi 95,026% dengan citra *grayscale* malware. Sementara itu, untuk citra RGB malware, model Convolutional Neural Network mendapatkan akurasi terbaik 97,591% dan *recall* terbaik 94,824%. Berdasarkan hasil penelitian, klasifikasi *malware* pada kedua jenis citra memperoleh hasil yang cukup baik.

Kata Kunci*— Cybersecurity, malware, klasifikasi gambar, Convolutional Neural Network (CNN).

I. PENDAHULUAN

Malware (malicious software) adalah perangkat lunak yang dibuat untuk merusak atau menjalankan proses yang dapat membahayakan sistem komputer, contohnya *virus*, *worms*, *trojan*, dan *spyware* [1]. Dengan berkembangnya internet, banyak variasi *malware* baru bermunculan yang seringkali digunakan untuk melakukan penyerangan dan tindakan kejahatan. Sebagai contoh, menurut laporan dari AV Test, dalam periode 2011 – November 2020, melaporkan sebanyak total jumlah *malware* yang terdeteksi sebanyak 1113.73 juta dan sebanyak 267.23 juta varian *malware* baru muncul dalam kurun 2 tahun terakhir [2]. Oleh karena itu, diperlukan sistem deteksi dan klasifikasi *malware* yang efektif untuk mencegah dan mengurangi terjadinya serangan *malware* pada sistem komputer.

Pada saat ini, pendekatan berbasiskan visualisasi citra *grayscale malware* banyak dikembangkan untuk melakukan klasifikasi *malware* [1] [3] [4] [5]. Citra *grayscale malware* dinilai lebih efisien dalam masalah klasifikasi *malware* karena keseluruhan struktur antar *malware* dapat diukur melalui kesamaan dan perbedaan fitur dan tekstur secara visual pada citra *grayscale*. Pada penelitian [3] menerapkan algoritme transformasi *wavelet* yang digunakan untuk metode ekstraksi fitur, sedangkan model *machine learning* yang dikembangkan adalah K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) untuk klasifikasi *malware*. Dalam penelitian ini, *dataset* yang digunakan adalah Malware Image (Malimg) khususnya *malware* yang berjenis *Trojan*. Akurasi terbaik yang dihasilkan model KNN adalah sebesar 89,11%, serta model SVM menghasilkan akurasi sebesar 73,55%. Kelemahan pada penelitian [3] adalah klasifikasi *malware* yang dilakukan belum memakai keseluruhan data di dalam *dataset* Malimg.

Pada penelitian [4], model klasifikasi *machine learning* yang dikembangkan untuk klasifikasi *malware* adalah Random Forest dengan akurasi 97,47%, K-Nearest Neighbor

(KNN) dengan akurasi 96,23%, dan Support Vector Machine (SVM) dengan akurasi 95,23%. Metode *machine learning* konvensional, seperti Random Forest, KNN, dan SVM memiliki kelemahan, yaitu peningkatan performa akan melambat seiring berkembangnya jumlah data masukan sehingga kurang cocok untuk pelatihan dengan volume data yang banyak.

Metode berbasiskan *deep learning* juga banyak dimanfaatkan untuk pengolahan citra *grayscale* untuk melakukan deteksi dan klasifikasi *malware* [1] [5]. Penelitian [1] menerapkan metode Convolution Neural Network (CNN) untuk klasifikasi *malware* terhadap citra *grayscale malware* menghasilkan akurasi 98,52% pada *dataset* Malimg dan 98,99% pada *dataset* Microsoft. Selain CNN, penelitian [5] mengembangkan metode *deep learning* lainnya, yaitu Extreme Learning Machine (ELM) untuk klasifikasi *malware* dengan akurasi 93,9%. Berdasarkan penelitian [1], [3], [4], [5] yang sudah dibahas, metode CNN dinilai mampu menganalisis pola tersembunyi pada data citra *grayscale* untuk klasifikasi *malware* sehingga dapat menghasilkan akurasi yang lebih tinggi dibandingkan metode lainnya.

Penelitian [6] melakukan eksperimen klasifikasi *malware* dengan model CNN menggunakan citra RGB yang didapat dari *file malware* pada *platform* Android. Klasifikasi *malware* menggunakan citra RGB dinilai efektif dengan akurasi sebesar 98,77%.

Oleh karena itu, penelitian ini akan menerapkan metode CNN untuk melakukan klasifikasi *malware* menggunakan visualisasi citra *malware* dengan menggunakan teknik *pseudocolor*, serta Principal Component Analysis (PCA) untuk ekstraksi fitur. Berdasarkan penelitian [7], PCA bertujuan untuk melakukan *dimension reduction* atau mengurangi dimensi pada masukan untuk mempercepat proses pembelajaran dan meningkatkan performa klasifikasi dengan menyeleksi dan mengekstrak fitur-fitur penting pada data. Kemudian, pada penelitian ini teknik *pseudocolor* dilakukan pada citra *grayscale malware* dari *dataset* untuk menghasilkan citra berwarna RGB, tujuannya adalah citra RGB memuat 3 *channel* warna, yaitu merah (*red*), hijau (*green*), dan biru (*blue*) sehingga memiliki informasi fitur yang lebih banyak dibandingkan citra *grayscale* yang hanya berisi 1 *channel* warna saja. Penelitian dibuat untuk membandingkan besar nilai akurasi pada klasifikasi terhadap citra *grayscale* dan citra RGB. Pengujian kinerja model klasifikasi CNN menggunakan *confusion matrix* untuk mengukur nilai akurasi, *recall*, dan *F-measure*.

II. METODOLOGI

A. Malware Classification

Malware yang bermunculan saat ini ada banyak jenisnya yang dikelompokkan berdasarkan dengan sifat dan cara menginfeksinya, seperti *virus*, *worm*, *trojan*, *ransomware*, dan *spyware*. Jenis atau kelas *malware* lebih jauh dibagi lagi ke dalam beberapa varian atau disebut *malware family*. Salah satu penyebab tingginya volume sampel *malware* adalah *file malware* yang berbahaya secara terus-menerus dimodifikasi dengan teknik pemrograman, seperti transformasi *polymorphic* dan *metamorphic* dengan tujuan untuk menyamarkan dan menghindari deteksi *malware* [1].

Terdapat 2 jenis analisis *malware* yang umum digunakan pada metode deteksi dan klasifikasi *malware* yaitu, analisis statis dan analisis dinamis. Analisis statis adalah analisis yang dilakukan dengan menganalisis struktur *source code* di dalam program tanpa menjalankan atau mengeksekusi program tersebut untuk mendeteksi *malware*. Sedangkan, pendekatan analisis dinamis dapat dilakukan dengan menjalankan *file* atau program *malware* untuk menganalisis dampaknya terhadap sistem [8].

Klasifikasi *malware* atau *malware classification* adalah proses untuk menetapkan atau melabeli suatu sampel *malware* ke dalam suatu *malware family* yang spesifik. Metode klasifikasi dan deteksi *malware* terdiri dari *signature-based*, *behaviour-based*, dan *heuristic-based* [9].

Signature-based merupakan metode analisis statis yang digunakan untuk mendeteksi *malware* menggunakan *signature* milik *malware* yang dapat diidentifikasi secara unik. Pendekatan ini cepat dan efektif untuk mendeteksi *malware family* yang sudah diketahui sebelumnya, tetapi tidak bisa mendeteksi varian atau *malware family* baru, serta masih memerlukan keterlibatan manusia [9].

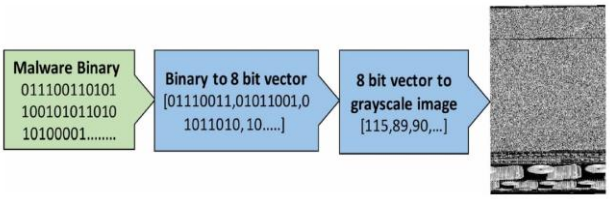
Behaviour-based merupakan pendekatan mendeteksi *malware* dengan melakukan observasi pada perubahan-perubahan yang terjadi pada sistem akibat *malware*. Perubahaan pada sistem dapat dilihat dari pemanggilan API, perubahan *registry*, dan monitoring pada jaringan, apakah perubahan-perubahan tersebut menyebabkan anomali atau kerusakan pada sistem untuk menentukan sampel *file* tersebut merupakan berbahaya atau tidak.

Heuristic-based merupakan metode pendeteksian *malware* dengan sekumpulan *rules* atau aturan untuk menentukan apakah suatu sampel *file* merupakan *malware* atau tidak. Aturan atau *rules* yang dipakai dalam pendekatan *heuristic* ini didapatkan dari hasil analisis baik dalam analisis *source code* dan *behaviour malware*. Setelah aturan ditentukan, sampel *file* akan diuji seberapa besar persentase kesesuaian sample *file* dengan aturan yang sudah dibuat, jika suatu sampel memenuhi persentase tertentu bisa dianggap sebagai *malware*.

B. Visualisasi Malware

Dalam konteks keamanan siber, *malware* terdiri dari kode-kode yang memuat pola-pola berbahaya untuk melakukan akses yang tidak terautentikasi dan penyerangan pada sistem komputer. Pendekatan analisis statis konvensional, seperti deteksi *signature*, bekerja dengan memeriksa kode dan *signature* untuk mengenali pola-pola mencurigakan yang terdapat dalam *malware*. Namun, analisis statis seringkali menemui masalah *code obfuscation.*

Code obfuscation memodifikasi kode *malware* awal menjadi kode yang berbeda, tetapi memiliki kesamaan secara sematik untuk menghindari deteksi *malware* [10]. *Code obfuscation* dibagi menjadi 2 jenis meliputi, teknik *polymorphic* dan *metamorphic.*



Gambar 1 Proses Visualisasi *Malware* [1]

Teknik *polymorphic* merupakan metode enkripsi untuk memodifikasi kode *binary* di dalam *malware*. Teknik *polymorphic* akan terus menghasilkan kode *malware* yang berbeda seiring berubahnya *encryption key*. Sedangkan, teknik *metamorphic* adalah perubahan kode *malware* tanpa penggunaan teknik enkripsi. Teknik *metamorphic* yang dapat dilakukan adalah penyisipan kode yang berulang dan *dead code* yang tidak memengaruhi terhadap fungsi *malware*. Teknik *metamorphic* lainnya adalah subsitusi instruksi untuk mengubah instruksi atau fungsi dengan nama yang berbeda tetapi menghasilkan keluaran yang sama [11].

Visualisasi *malware* adalah proses visualisasi dari *malware binary* menjadi citra [8], proses visualisasi *malware* dapat dilihat pada Gambar 1. *Malware binary* merupakan program *executable* yang bisa langsung dijalankan atau biasanya disebut file biner atau *binary file* dengan ekstensi .bin atau .exe, serta bisa direpresentasikan sebagai string yang berisi 1 dan 0 [8].

Setiap string *binary* akan diambil sebagai suatu vektor *8-bit unsigned integer* dan diubah menjadi *array* 2 dimensi. Vektor yang berisi *8-bit unsigned integer* digunakan untuk menentukan nilai keabuan pada setiap *pixel* membentuk citra *grayscale*. Setiap *pixel* memiliki nilai keabuan dengan rentang nilai antara 0-255, nilai 0 merupakan nilai keabuan terendah yang berwarna hitam, sedangkan nilai 255 merupakan nilai keabuan tertinggi yang berwarna putih [12].

Pendekatan visualisasi *malware* merupakan salah satu solusi untuk menangani masalah *polymorphic* dan *metamorphic* karena perubahan-perubahan *malware* dapat terlihat secara visual dalam fitur dan tekstur dalam citra *malware*. Visualisasi *malware* dapat menggambarkan keseluruhan struktur kode *malware* sehingga dapat mengidentifikasi perubahan-perubahan kecil secara visual melalui citra *grayscale*. Perbedaan dan kesamaan tekstur atau fitur dalam citra *grayscale* dapat membantu analisis dalam klasifikasi satu *malware family* dengan *family* lainnya [8].

C. Principal Component Analysis (PCA)

Principal Component Analysis merupakan salah satu metode statistik multivariat yang berfungsi untuk ekstraksi fitur dari *dataset* asli menjadi sekumpulan fitur atau variabel ortogonal yang disebut *principal component* [13]. *Principal component* merupakan ringkasan fitur-fitur penting yang mempunyai varians yang tinggi dari data dalam *dataset*, setiap *principal component* mempunyai tingkat varians yang berbeda. Jumlah informasi yang dipertahankan bergantung pada nilai *threshold* yang ditentukan. Semakin tinggi nilai *threshold*, maka semakin tinggi jumlah *principal component* yang dipertahankan, sedangkan semakin rendah nilai *threshold*, maka semakin rendah *principal component* yang dipertahankan atau semakin tinggi *principal component* yang akan diseleksi. Oleh karena itu, diperlukan pemilihan jumlah *principal component* yang tepat untuk untuk mengurangi dimensi atau ukuran data, sambil mempertahankan fitur dengan informasi atau varians yang signifikan.

D. Pseudocolor

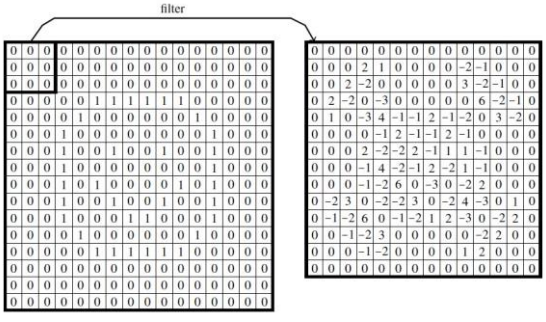
Pseudocolor atau *false color* merupakan pengolahan citra untuk pemberian warna pada nilai keabuan di dalam citra *grayscale* berdasarkan kriteria atau aturan tertentu. Karena manusia lebih mudah memahami dan peka terhadap intensitas warna sehingga penerapan *pseudocolor* dapat membantu manusia dalam melakukan visualisasi dan interpretasi pada bagian keabuan pada suatu atau sekumpulan citra [14].

Salah satu pendekatan *pseudocolor* adalah transformasi warna pada nilai intensitas keabuan. Citra *grayscale* biasanya memiliki *channel* warna tunggal yang bersifat akhromatis atau tingkat kecerahan yang menampilkan informasi, seperti temperatur dan tekstur, lalu dilakukan perhitungan fungsi transformasi pada nilai intensitas dari *pixel* masukan untuk mendapatkan masing-masing nilai *channel* merah, hijau, dan biru sehingga menghasilkan citra berwarna dengan model warna RGB [14].

E. Convolution Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan pengembangan dari *neural network* untuk memproses data matriks 2 dimensi atau lebih sebagai data masukan. Konsep dasar CNN terinspirasi dari cara kerja korteks visual pada hewan yang dirancang secara adaptif dapat mengenali fitur spasial dari fitur yang sederhana (*low-level*) sampai kompleks (*high-level*).

Pada model CNN akan dilakukan operasi konvolusi dengan menggunakan *kernel*. *Kernel* atau *filter* merupakan suatu matriks atau *grid* yang berisi nilai angka, setiap nilai angka dalam *kernel* merupakan *weight* atau bobot. Operasi konvolusi merupakan proses pergerakan *kernel* secara horizontal dan vertikal pada setiap *pixel* di dalam suatu citra sehingga menghasilkan fitur yang disebut *feature map*, seperti yang terlihat pada Gambar 2. Pada saat proses pelatihan, nilai bobot pada *kernel* akan terus diperbaharui atau di-*tuning* sehingga dapat mengenali fitur penting pada citra masukan [15].



Gambar 2 Operasi Konvolusi dengan *Kernel* 3x3 [10]

Arsitektur model CNN secara umum terdiri dari 3 jenis *layer* atau *building block* yaitu, *convolution layer*, *pooling layer*, dan *fully connected layer*. *Convolution* dan *pooling layer* berfungsi untuk melakukan ekstraksi fitur, sedangkan *fully connected layer* digunakan untuk melakukan pemetaan berdasarkan fitur yang diekstrak sebagai keluaran, seperti klasifikasi dan deteksi [16].

F. Perancangan Sistem

1) Kerangka Pemikiran

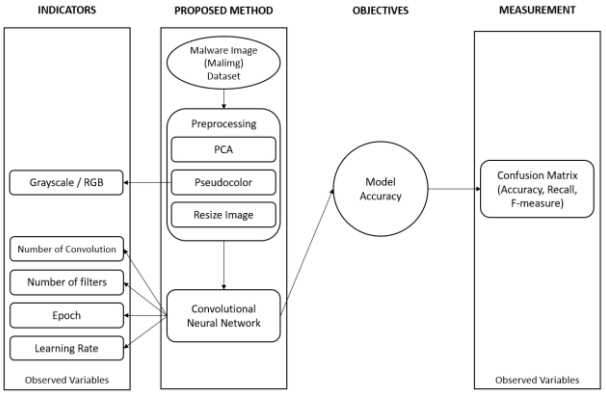
Gambar 3 merupakan kerangka pemikiran dari penelitian yang akan dikembangkan untuk klasifikasi *malware* dengan menggunakan *convolutional neural network*.

Terdapat beberapa indikator yang akan diuji pada penelitian ini, masing-masing indikator dipilih karena dapat memengaruhi hasil dari model klasifikasi yang akan dikembangkan sehingga diharapkan dapat meningkatkan akurasi pada model CNN. Tujuan yang ingin dicapai dari penelitian ini adalah membandingkan nilai akurasi dan *recall* pada klasifikasi *malware* terhadap *citra grayscale* dan citra RGB. Nilai-nilai pengukuran tersebut diperoleh dari *confusion matrix* sehingga menghasilkan nilai akurasi, *recall*, dan *F-measure* untuk evaluasi performa model Convolutional Neural Network dalam mengklasifikasikan *malware family* berdasarkan visualisasi citra *malware*.

Penelitian dimulai dengan data *preprocessing*, indikator pada tahap ini berupa citra masukan yang digunakan untuk klasifikasi, citra *grayscale* dan citra RGB. Citra *grayscale malware* merupakan citra awal yang terdapat dari *dataset*, sedangkan citra RGB merupakan keluaran dari teknik *pseudocolor* pada citra *grayscale.*

Pada model CNN terdapat beberapa indikator *hyperparameter* yang akan diobservasi sebagai berikut.

1. *Number of convolutions* menentukan jumlah *convolution layer* yang digunakan dalam arsitektur Convolutional Neural Network.
2. *Number of filters* merupakan *hyperparameter* pada *convolution layer* untuk menentukan jumlah *filter* atau *kernel* yang digunakan dalam suatu *convolution layer*.
3. *Epoch* merupakan *hyperparameter* untuk menentukan jumlah iterasi pelatihan model CNN dengan seluruh data belajar.
4. *Learning rate* merupakan *hyperparameter* yang berfungsi untuk mengatur seberapa besar pembaharuan bobot pada saat pelatihan.

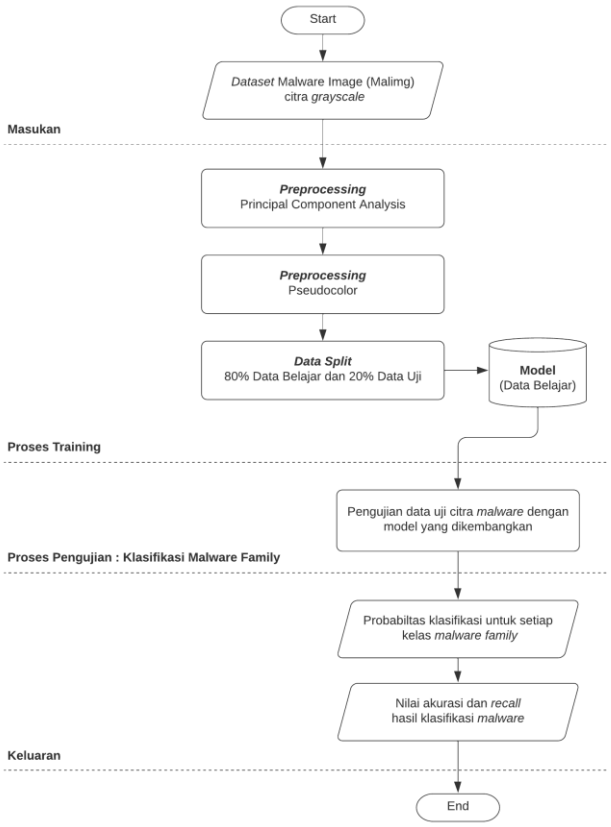


Gambar 3 Kerangka Pemikiran

2) Flowchart Global

Flowchart pada Gambar 4 menunjukkan urutan proses pada penelitian ini sebagai berikut.

1. Pertama, *dataset* yang dipakai adalah Malware Image (Malimg) yang memuat 9339 citra dan 25 *family*.
2. *Dataset* Malimg dilakukan tahap *preprocessing* di mana dilakukan ekstraksi fitur dengan Principal Component Analysis (PCA) untuk menyeleksi fitur-fitur penting, kemudian citra *grayscale malware* diproses dengan teknik *pseudocolor* untuk menghasilkan citra berwarna RGB.
3. *Dataset* baik citra *grayscale* dan citra RGB akan dilakukan pembagian data dengan komposisi 80% data belajar (*training*) dan 20% data uji (*testing*).
4. Data latih atau belajar akan digunakan untuk proses pembelajaran model CNN.
5. Model CNN yang sudah selesai dilatih akan dilakukan pengujian untuk menilai kinerja model CNN dalam mengklasifikasikan citra *malware*.
6. Keluaran yang dihasilkan adalah probabilitas klasifikasi untuk setiap kelas *malware family*. Nilai probabilitas tertinggi yang akan diambil sebagai hasil prediksi *malware family*.



Gambar 4 *Flowchart* Global

TABEL I
FAMILY DAN JENIS *MALWARE*

No	Family	Jenis	Sampel
1	Adialer.C	Dialer	122
2	Agent.FYI	Backdoor	116
3	Allaple.A	Worm	2949
4	Allaple.L	Worm	1591
5	Alueron.gen!J	Trojan	198
6	Autorun.K	Worm AutoIT	106
7	C2LOP.P	Trojan	146
8	C2LOP.gen!G	Trojan	200
9	Dialplatform.B	Dialer	177
10	Dontovo.A	TDownloader	162
11	Fakerean	Rogue	381
12	Instantaccess	Dialer	431
13	Lolyada.AA1	PWS	213
14	Lolyada.AA2	PWS	184
15	Lolyada.AA3	PWS	123
16	Lolyada.AT	PWS	159
17	Malex.gen!J	Trojan	136
18	Obfuscator.AD	TDownloader	142
19	RBot!gen	Backdoor	158
20	Skintrim.N	Trojan	80
21	Swizzor.gen!E	TDownloader	128
22	Swizzor.gen!I	TDownloader	132
23	VB.AT	Worm	408
24	Wintrim.BX	TDownloader	97
25	Yuner.A	Worm	800

7. Setelah hasil prediksi *malware family* diperoleh maka performa model CNN dapat diukur dengan metode *confusion matrix* sehingga menghasilkan nilai akurasi dan *recall* sebagai hasil pengujian.

3) Dataset

Dataset yang digunakan adalah Malware Image (Malimg) yang bersumber dari penelitian yang dibuat oleh Faculty of Computer Science, Computer Engineering, and Electrical Engineering, University of California [12]. *Dataset* Malimg ini berisi *malware family* yang sering ditemukan pada *file executable*, dan *dataset* sudah diolah sebelumnya menjadi citra *grayscale* dengan format PNG. Jumlah citra *grayscale* di dalam *dataset* adalah 9339 citra yang terdiri dari 7 jenis *malware*, serta dibagi lagi menjadi *malware family* dengan total 25 *family* (lihat Tabel I).

Keseluruhan data akan dibagi dengan komposisi 80% sebagai data belajar dan 20% sebagai data uji. Dalam pembelajaran Convolutional Neural Network, dari 80% data belajar akan dibagi lagi 20% untuk data validasi selama proses pelatihan. Sementara itu, pengujian diperlukan untuk melihat kinerja model saat menganalisis data baru, apakah model klasifikasi sudah mencapai generalisasi dengan baik atau cenderung mengalami *overfit* ataupun *underfit*.

III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian

Pengujian yang dilakukan adalah melakukan perbandingan jenis citra masukan, yaitu citra *grayscale* dan citra RGB *malware*. Sedangkan, pengujian pada model Convolutional Neural Network melibatkan banyak kombinasi *hyperparameter*, berupa jumlah *convolution layer*, jumlah *filter, epoch,* dan *learning rate*. Tabel II menunjukkan nilai-nilai *hyperparameter* yang diuji pada penelitian ini.

Pengujian terhadap model Convolutional Neural Network dibagi menjadi 4 rancangan arsitektur. Setiap rancangan arsitektur dibuat untuk menguji jumlah *convolution layer* dan jumlah *filter*. Gambar 3 hingga Gambar 6 menunjukkan 4 konfigurasi arsitektur untuk melakukan klasifikasi citra *malware*. Untuk skenario pengujian pada suatu jenis citra masukan pada 4 rancangan arsitektur berjumlah 36 pengujian. Jadi, total skenario pengujian pada penelitian ini adalah 72 pengujian.

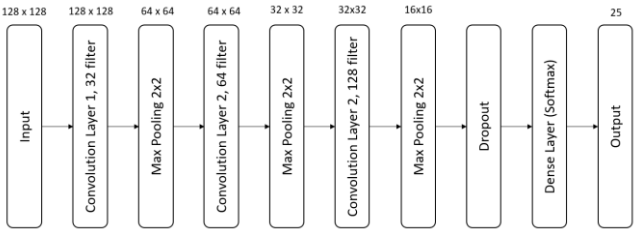
Tabel III hingga Tabel VIII merupakan seluruh hasil pengujian model Convolutional Neural Network pada 4 arsitektur kombinasi pengujian *learning rate* dan *epoch* baik pada citra *grayscale* dan citra RGB *malware*. Hasil pengujian merupakan hasil klasifikasi citra pada data uji yang meliputi nilai akurasi, *precision*, *recall*, dan *F-measure* yang diperoleh dari *confusion matrix*.

Berdasarkan hasil penelitian, klasifikasi dengan citra *grayscale malware* mendapatkan nilai akurasi paling tinggi 97,752% pada model CNN dengan arsitektur 4, *learning rate* 0,0001, dan *epoch* 100 (lihat Tabel VI). Hasil performa tersebut lebih tinggi dibandingkan hasil klasifikasi menggunakan citra RGB *malware* yang memperoleh nilai akurasi tertinggi 97,591% pada model dengan arsitektur 3, learning rate 0,001, epoch 50 (lihat Tabel IX). Tetapi, selisih perbedaan hasil pengujian terbaik antara klasifikasi dengan citra *grayscale* dan citra RGB *malware* tidak terlalu berbeda jauh, serta hasil pengujian terbaik untuk kedua jenis citra *malware* tersebut dihasilkan dari rancangan arsitektur dan nilai *hyperparameter* yang serupa.

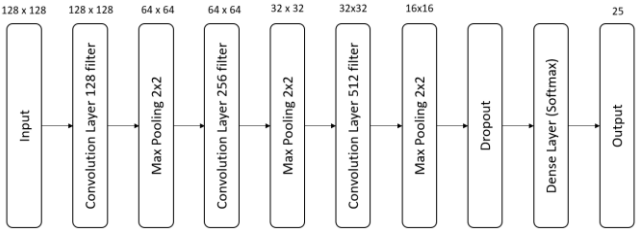
Hyperparameter yang terdapat pada model CNN juga turut memengaruhi kinerja dan performa dalam melakukan klasifikasi. Jumlah *convolution layer* menentukan jumlah konvolusi yang dilakukan pada suatu data citra dalam model Convolution Neural Network. Jumlah *convolution layer* akan memengaruhi seberapa dalam lapisan atau *layer* konfigurasi

TABEL III
NILAI SKENARIO PENGUJIAN

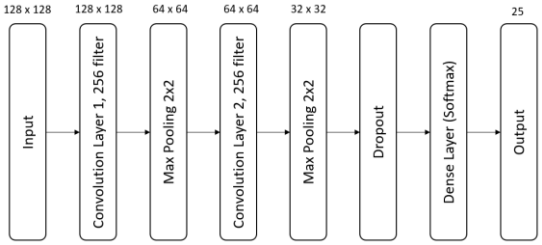
Indikator	*Parameter* Pengujian	Nilai yang Diuji
Jenis Citra Masukan	Model Warna	2 Jenis Citra (*Grayscale* dan RGB)
Convolutional Neural Network (CNN)	*Epoch* *Learning Rate* Rancangan Arsitektur (Jumlah *convolution layer* & *filter*)	10, 50, 100 0.001, 0.005, 0.0001 **Arsitektur 1** – 3 *convolution layer* & 32, 64, 128 *filter* **Arsitektur 2** – 3 *convolution layer* & 128, 256, 512 *filter* **Arsitektur 3** – 2 *convolution layer* & 256, 256 *filter* **Arsitektur 4** – 2 *convolution layer* & 512, 512 *filter*



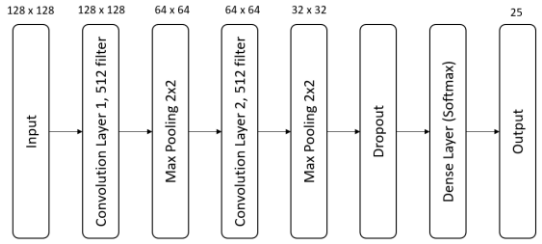
Gambar 3 Rancangan Arsitektur CNN 1



Gambar 4 Rancangan Arsitektur CNN 2



Gambar 5 Rancangan Arsitektur CNN 3



Gambar 6 Rancangan Arsitektur CNN 4

model CNN. Namun, jumlah *convolution layer* yang banyak belum tentu menghasilkan model CNN dengan akurasi yang tinggi. Pada arsitektur 1 dan 2 memakai 3 *convolution layer* dibandingkan arsitektur 3 dan 4 yang memakai 2 *convolution layer*. Tetapi, pada pengujian klasifikasi citra *grayscale*, akurasi tertinggi dicapai dengan arsitektur 4 mencapai akurasi 97,752% dibandingkan arsitektur lainnya (lihat Tabel VI).

Jumlah *filter* menentukan jumlah fitur (*feature map*) yang akan diekstrak dalam suatu *convolution layer*. Semakin banyak *filter* yang digunakan, maka fitur yang diekstrak dari suatu citra masukan juga semakin banyak, demikian juga sebaliknya. Penggunaan jumlah *filter* yang tinggi dapat meningkatkan akurasi model dalam mengklasifikasi *malware*. Tetapi, jumlah *filter* yang terlalu tinggi dapat menyebabkan jumlah fitur yang diekstrak terlalu banyak sehingga model cenderung menghafal fitur pada citra. Hal tersebut dapat meningkatkan kemungkinan terjadinya misklasifikasi pada citra yang belum dikenali sebelumnya dan berdampak pada penurunan performa model. Sebagai contoh, pada arsitektur 3 menggunakan *filter* dengan total 512 dan arsitektur 4 menggunakan filter dengan total 1024 sehingga model dengan arsitektur 4 diharapkan menghasilkan akurasi yang lebih tinggi secara teori. Namun, berdasarkan hasil pengujian pada kasus klasifikasi citra RGB *malware* yang dapat dilihat pada Tabel IX dan Tabel X, arsitektur 3 memperoleh nilai akurasi yang lebih tinggi (97,591%) daripada arsitektur 4 (96,949%). Hal tersebut dikarenakan ada *hyperparameter* lain yang turut memengaruhi, seperti *epoch* dan *learning rate*, sehingga perlu dilakukan pencarian jumlah *filter* dengan dikombinasikan dengan nilai *learning rate* dan *epoch* yang optimal.

Learning rate merupakan salah satu *hyperparameter* terpenting pada konfigurasi model Convolutional Neural Network karena mempengaruhi seberapa cepat model beradaptasi saat proses pelatihan. Jika nilai *learning rate* terlalu besar, maka pembaharuan bobot juga terlalu besar yang memungkinkan model melewati titik konvergensi. Secara keseluruhan, *learning rate* 0,005 tidak cocok digunakan untuk klasifikasi *malware* karena dinilai terlalu tinggi, sedangkan *learning rate* 0,001 dan 0,0001 dapat melatih model dengan cukup baik pada penelitian ini. Contoh pada kasus arsitektur pertama untuk klasifikasi dengan citra RGB *malware* (lihat Tabel VII), penggunaan nilai *learning rate* 0,0001 yang disertai nilai *epoch* 50 merupakan nilai *learning rate* yang paling sesuai karena model memiliki akurasi yang tinggi, yaitu 96,842%. Sedangkan, nilai *learning rate* yang sama belum tentu cocok digunakan pada arsitektur yang berbeda, seperti pada arsitektur kedua dengan *learning rate* 0,0001 (lihat Tabel VIII), nilai akurasi terbaik yang diperoleh lebih rendah sebesar 96,36%. Oleh karena itu, nilai *learning rate* sangat memengaruhi hasil klasifikasi pada suatu model sehingga diperlukan pengujian untuk menemukan nilai *learning rate* yang optimal.

Epoch merupakan *hyperparameter* yang menentukan jumlah iterasi pelatihan yang dilakukan pada model Convolutional Neural Network. *Epoch* dengan nilai 50 dan 100 merupakan nilai *epoch* yang optimal untuk digunakan melatih model CNN melakukan klasifikasi *malware*. Namun, perlu dipertimbangkan juga *epoch* yang relatif tinggi tidak secara pasti dapat meningkatkan performa model klasifikasi, tetapi dapat memengaruhi waktu pelatihan model yang akan berdampak pada meningkatnya beban komputasi yang dibutuhkan untuk proses pelatihan. Sebagai contoh, pada klasifikasi citra *grayscale malware* memakai arsitektur 2 dan *learning rate* 0,0001, *epoch* dengan nilai 100 mengalami penurunan nilai akurasi menjadi 96,627% jika dibandingkan dengan nilai *epoch* 50 yang mencapai akurasi sebesar 97,163% pada arsitektur yang sama (lihat Tabel IV). Hal ini disebabkan model terlalu banyak melakukan pembelajaran pada data belajar sehingga model cenderung mengalami *overfitting*. Walaupun model memiliki akurasi yang tinggi pada proses pelatihan, tetapi terjadi penurunan akurasi saat pengujian dengan nilai *epoch* 100 dibandingkan *epoch* 50.

TABEL III
HASIL PENGUJIAN KLASIFIKASI CITRA *GRAYSCALE* DENGAN ARSITEKTUR 1

Learning Rate	Epoch	Accuracy	Precision	Recall	F-Measure
0,005	10	94,004%	91,649%	91,447%	90,879%
	50	93,201%	89,929%	92,534%	90,281%
	100	95,557%	91,993%	92,509%	91,261%
0,001	10	96,306%	95,000%	93,434%	92,152%
	50	97,591%	95,389%	95,026%	95,141%
	100	96,842%	95,121%	93,968%	94,109%
0,0001	10	86,456%	85,701%	88,682%	85,622%
	50	97,537%	94,744%	94,444%	94,568%
	100	97,591%	94,917%	94,502%	94,650%

TABEL IV
HASIL PENGUJIAN KLASIFIKASI CITRA *GRAYSCALE* DENGAN ARSITEKTUR 2

Learning Rate	Epoch	Accuracy	Precision	Recall	F-Measure
0,005	10	94,486%	88,450%	91,787%	89,503%
	50	94,111%	89,307%	91,463%	89,699%
	100	94,540%	92,032%	91,962%	90,935%
0,001	10	97,109%	94,137%	94,287%	94,109%
	50	96,520%	93,582%	93,502%	93,220%
	100	96,895%	94,108%	94,296%	93,970%
0,0001	10	96,253%	93,617%	93,472%	93,135%
	50	97,163%	93,976%	94,455%	94,001%
	100	96,627%	92,979%	93,200%	92,971%

TABEL V
HASIL PENGUJIAN KLASIFIKASI CITRA *GRAYSCALE* DENGAN ARSITEKTUR 3

Learning Rate	Epoch	Accuracy	Precision	Recall	F-Measure
0,005	10	94,540%	90,298%	90,484%	89,657%
	50	95,771%	93,481%	93,328%	92,695%
	100	95,128%	92,595%	92,386%	90,778%
0,001	10	95,610%	93,591%	92,960%	91,823%
	50	95,503%	92,909%	93,591%	92,235%
	100	96,895%	94,200%	94,212%	93,009%
0,0001	10	96,788%	94,033%	93,576%	93,385%
	50	97,109%	93,580%	93,607%	93,331%
	100	97,377%	93,468%	93,723%	93,539%

TABEL VI
HASIL PENGUJIAN KLASIFIKASI CITRA *GRAYSCALE* DENGAN ARSITEKTUR 4

Learning Rate	Epoch	Accuracy	Precision	Recall	F-Measure
0,005	10	93,790%	89,317%	89,132%	88,531%
	50	94,861%	92,254%	91,380%	91,609%
	100	94,272%	90,933%	91,247%	90,853%
0,001	10	96,574%	94,805%	93,751%	93,812%
	50	93,362%	91,631%	92,347%	89,918%
	100	96,146%	94,168%	94,182%	93,338%
0,0001	10	96,306%	92,582%	93,218%	91,959%
	50	97,645%	94,014%	94,385%	93,961%
	100	97,752%	93,899%	94,324%	94,077%

TABEL VII
HASIL PENGUJIAN KLASIFIKASI CITRA RGB DENGAN ARSITEKTUR 1

Learning Rate	Epoch	Accuracy	Precision	Recall	F-Measure
0,005	10	92,238%	87,604%	88,932%	87,827%
	50	94,165%	89,934%	91,634%	90,225%
	100	94,807%	91,277%	92,829%	91,728%
0,001	10	95,021%	92,198%	93,567%	92,709%
	50	96,467%	92,223%	92,997%	92,501%
	100	96,360%	92,353%	93,698%	92,635%
0,0001	10	93,194%	93,194%	94,824%	93,902%
	50	96,842%	94,413%	94,824%	94,568%
	100	96,520%	93,794%	93,823%	93,590%

TABEL VIII
HASIL PENGUJIAN KLASIFIKASI CITRA RGB DENGAN ARSITEKTUR 2

Learning Rate	Epoch	Accuracy	Precision	Recall	F-Measure
0,005	10	94,218%	89,718%	91,790%	89,701%
	50	94,540%	89,486%	91,059%	89,873%
	100	94,807%	90,328%	90,291%	89,446%
0,001	10	95,289%	93,445%	91,392%	90,805%
	50	96,146%	93,274%	94,216%	93,442%
	100	96,681%	92,543%	94,024%	93,097%
0,0001	10	95,771%	92,490%	94,714%	93,397%
	50	95,021%	91,810%	93,896%	92,612%
	100	96,360%	94,027%	94,153%	93,554%

TABEL IX
HASIL PENGUJIAN KLASIFIKASI CITRA RGB DENGAN ARSITEKTUR 3

Learning Rate	Epoch	Accuracy	Precision	Recall	F-Measure
0,005	10	94,379%	91,755%	91,535%	91,138%
	50	94,914%	92,422%	91,627%	91,740%
	100	94,754%	91,212%	91,023%	90,532%
0,001	10	95,396%	91,617%	93,204%	91,140%
	50	97,591%	94,650%	94,548%	94,453%
	100	96,842%	93,090%	92,945%	92,211%
0,0001	10	97,323%	93,459%	93,609%	92,951%
	50	96,199%	94,562%	94,128%	93,661%
	100	97,163%	93,596%	93,563%	93,426%

TABEL X
HASIL PENGUJIAN KLASIFIKASI CITRA RGB DENGAN ARSITEKTUR 4

Learning Rate	Epoch	Accuracy	Precision	Recall	F-Measure
0,005	10	94,593%	89,855%	91,683%	89,922%
	50	96,199%	93,506%	92,714%	92,975%
	100	93,844%	89,592%	89,554%	88,047%
0,001	10	94,165%	89,441%	90,796%	88,652%
	50	96,949%	94,453%	93,495%	93,813%
	100	96,413%	92,794%	93,128%	92,454%
0,0001	10	96,734%	93,129%	93,130%	93,103%
	50	96,788%	94,106%	93,329%	93,524%
	100	96,949%	94,298%	93,609%	93,741%

IV. SIMPULAN

Pada penelitian ini, hasil pengujian dari sistem klasifikasi *malware* lebih menitikberatkan pada nilai akurasi dan *recall* untuk menilai performa model Convolution Neural Network. Akurasi digunakan untuk mengevaluasi jumlah prediksi yang benar terhadap keseluruhan klasifikasi *malware* yang dihasilkan model. Semakin tinggi nilai akurasi, maka semakin banyak prediksi *malware family* yang diklasifikasikan dengan tepat. Sedangkan, *recall* merupakan pengukuran tingkat kesalahan dalam mengklasifikasi suatu *malware family* sebagai *malware family* yang lain sehingga digunakan untuk menekan jumlah *false negative*.

Tabel XI menunjukkan hasil pengujian terbaik untuk klasifikasi citra *grayscale* dan RGB *malware*. Penerapan Convolutional Neural Network untuk klasifikasi citra *grayscale malware* mendapatkan hasil kinerja model klasifikasi terbaik dengan nilai akurasi tertinggi 97,752% pada arsitektur 4 dengan kombinasi *hyperparameter learning rate* 0,0001 dan *epoch* 100, serta diperoleh nilai *recall* tertinggi 95,026% pada arsitektur 1 dengan kombinasi *hyperparameter learning rate* 0,001 dan *epoch* 50. Hasil kinerja model CNN terbaik untuk klasifikasi memakai citra RGB *malware* mendapatkan nilai akurasi tertinggi 97,591% pada arsitektur 3 dengan kombinasi *hyperparameter learning rate* 0.001 dan *epoch* 50, serta diperoleh nilai *recall* tertinggi 94,824% pada arsitektur 1 dengan kombinasi *hyperparameter learning rate* 0,0001 dan *epoch* 50.

Metode Convolutional Neural Network menghasilkan performa akurasi dan *recall* yang lebih baik pada klasifikasi dengan citra *grayscale malware* dibandingkan citra RGB *malware*. Walaupun begitu, model Convolutional Neural Network yang dibangun sudah dapat bekerja secara maksimal mengklasifikasikan masing-masing citra *grayscale* dan citra RGB *malware*, hal ini dapat dilihat dari selisih hasil klasifikasi pada kedua jenis citra *malware* tidak terlalu berbeda jauh karena konfigurasi arsitektur yang uji semuanya seragam dan sudah menghasilkan performa yang cukup baik.

TABEL XI
HASIL PENGUJIAN TERBAIK
KLASIFIKASI CITRA *GRAYSCALE* DAN RGB *MALWARE*

Jenis Citra	Arsitektur	Learning Rate	Epoch	*Accuracy*	*Recall*
Grayscale	4	0.0001	100	**97.752%**	94.324%
Grayscale	1	0.001	50	97.591%	**95.026%**
RGB	3	0.001	50	**97.591%**	94.548%
RGB	1	0.0001	50	96.824%	**94.824%**

DAFTAR REFERENSI

[1] M. Kalash, M. Rochan, N. Mohammed, N. Bruce, Y. Wang, dan F. Iqbal, "A Deep Learning Framework for Malware Classification," *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 12, no.1, hlm. 90-108, Mar 2020.

[2] B. Yadav dan S. Tokekar, "Recent Innovations and Comparison of Deep Learning Techniques in Malware Classification: A Review," *International Journal of Information Security Science*, vol. 9, no. 4, hlm. 230-247, 2020.

[3] A. Makandar dan A. Patrot, "Trojan Malware Image Pattern Classification," dalam *Proceedings of International Conference on Cognition and Recognition*, Springer Singapore, 2018.

[4] Ad J. Fu, J. Xue, Y. Wang, Z. Liu, dan C. Shan, "Malware Visualization for Fine-Grained Classification," *IEEE Access*, vol. 6, hlm. 14510-14523, 2018.

[5] M. Jain, W. Andreopoulos, dan M. Stamp, "Convolutional neural networks and extreme learning machines for malware classification," *Journal of Computer Virology and Hacking Techniques*, vol. 16, hlm. 229-244, 2020.

[6] A. Darwaish dan F. Na¨ıt-Abdesselam, "RGB-based Android Malware Detection and Classification Using Convolutional Neural Network," dalam *GLOBECOM 2020 - 2020 IEEE Global Communication Conference*, hlm. 1-6, 2020.

[7] Rasheed, A.A. Hameed, C. Djeddi, A. Jamil, dan F. Al-Turjman, "A machine learning-based framework for diagnosis of COVID-19 from chest X-ray images," *Interdisciplinary Sciences: Computational Life Sciences*, vol. 13, hlm. 103-117, 2021.

[8] M. Stamp, M. Alazab, dan A. Shalaginov, *Malware Analysis Using Artificial Intelligence and Deep Learning*, Berlin/Heidelberg, Germany: Springer, 2021.

[9] O. Aslan dan A.A.Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms," *IEEE Access*, vol. 9, hlm. 87936-87951, 2021.

[10] M. Alazab, S. Venkatraman, P. Watters, M. Alazab, dan A. Alazab, "Cybercrime: The Case of Obfuscated Malware," *Global security, safety and sustainability & e-Democracy*, hlm. 204-211, 2011.

[11] J. Singh dan J. Singh, "Challenges of Malware Analysis: Obfuscation Techniques," *International Journal of Information Security Science*, vol. 7, no. 3, hlm. 100-110, 2018.

[12] L. Nataraj, S. Karthikeyan, G. Jacob, dan B.S. Manjunath, "Malware images: visualization and automatic classification," dalam *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, artikel 4, hlm. 1-7, 2011.

[13] H. Abdi dan L.J. Williams, "Principal component analysis," *WIREs Comp Stat*, vol. 2, no.4, hlm. 433-459, 2010.

[14] R.C. Gonzalez dan R.E.Woods, *Digital Image Processing*, 3rd Edition, New Jersey: Pearson Prentice Hall, 2008.

[15] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, dan D. De, "Fundamental Concepts of Convolutional Neural Network," *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, vol. 172, Springer, hlm. 519-567, 2019.

[16] R. Yamashita, M. Nishio1, R.K.G. Do, dan K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, hlm. 611-629, 2018.

Chris Christian, menerima gelar Sarjana Komputer dari Institut Teknologi Harapan Bangsa (ITHB) Bandung pada tahun 2022. Saat ini aktif sebagai Backend Engineer di perusahaan Roketin. Minat penelitian meliputi bidang *Image Processing* dan *Machine Learning*.

Ventje Jeremias Lewi Engel, menerima gelar gelar Sarjana Teknik dari Institut Teknologi Bandung (ITB) pada tahun 2012 dan Magister Teknik dari Institut Teknologi Bandung (ITB) pada tahun 2013. Aktif sebagai dosen di Prodi Informatika ITHB. Minat penelitian pada bidang *Deep Learning, Cybersecurity*, dan *Malware Analysis*.