

DETEKSI COVID-19 IMAGE CHEST X-RAY DENGAN CONVOLUTION NEURAL NETWORK EFFICIENT NET-B7

Adhitio Satyo Bayangkari Karno¹, Dodi Arif², dan Indra Sari²
Kusuma Wardhana³

⁽¹⁾Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok, Jawa Barat 16424

⁽²⁾ STMIK Jakarta STI&K

Jl. BRI No.17, Radio Dalam, Kebayoran Baru, Jakarta Selatan 12140

adh1t10.2@gmail.com

ABSTRAK

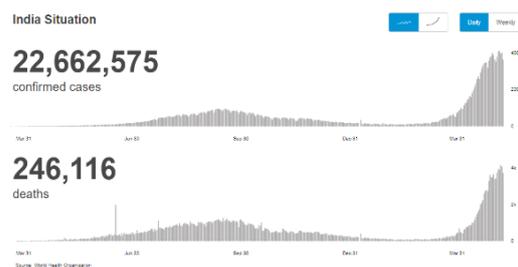
Di era pandemi keberadaan para medis dan rumah sakit terhadap jumlah pasien covid sangat tidak seimbang, lembaga kesehatan memerlukan alat bantu untuk tetap dapat memberikan pelayanan kesehatan. Kecerdasan buatan mampu memprediksi data image chest x-ray terhadap pasien penderita covid dan penyakit lainnya. Penelitian ini bermaksud untuk dapat mendeteksi covid-19 dari data image chest x-ray menggunakan Convolution Neural Network (CNN). Operasi yang ringan dengan kualitas akurasi sangat baik dari arsitektur EfficientNet-B7 dapat dipergunakan oleh komputer performa tanpa Graphics Processing Unit (GPU). Dataset yang dipergunakan berbentuk image chest x-ray berjumlah 4.000 image, terdiri dari 4 klasifikasi yaitu covid, normal, lung opacity dan viral pneumonia masing-masing dengan jumlah data 1.000 image. Hasil penelitian dengan training 50 epoch diperoleh nilai akurasi training 95,5% , akurasi validasi 91,8% dan akurasi testing 96%. Untuk tiap kelas hasil testing covid (96%) , normal (95%), lung opacity (93%) dan viral pneumonia (98%)

Kata Kunci : EfficientNet-B7, Deep Learning, Convolution Neural Network, COVID-19, Chest-Xray

PENDAHULUAN

Januari 2021, evoria terjadi di India yang merasa telah mampu mengatasi pandemi Covid. Perayaan keagamaan massal, demonstrasi politik pemilihan umum, dan perayaan Kumbh Mela, semua dilaksanakan secara besar-besaran dengan mengabaikan protokol kesehatan. Tidak lama setelah masa evoria, sesuatu yang sangat mengerikan mulai terjadi pada bulan Maret 2021. Tsunami Covid gelombang ke dua melanda India, penularan kasus baru setiap hari mencapai ratusan ribu orang. Banyak warga India meninggal karena kesulitan memperoleh pelayanan di unit-unit medis (rumah sakit). Ketersediaan tempat tidur di rumah sakit tidak mampu menampung jumlah pasien (over load). Lembaga kremasi tidak mampu menangani jumlah kematian yang tinggi. Mayat bergelimpangan dimana-mana, sungai dipenuhi oleh bangkai manusia yang tidak dikremasi karena kayu bakar habis. Penambahan kasus corona terbanyak di dunia mencapai pecah rekor di India. Sebanyak 401.993 orang di India terkonfirmasi positif dalam waktu sehari

[1]. "Tsunami Covid" di India merupakan kejadian penambahan jumlah penderita covid-19 (gambar 1) yang sangat luar biasa. Terbatasnya para medis dan rumah sakit tidak mampu menangani tingginya jumlah pasien penderita covid. Kurangnya ketersediaan alat bantu medis mempersulit penanganan gelombang besar pasien.



Gambar 1. Data Covid di India [2]

Penelitian ini bermaksud membantu para medis menyediakan alat bantu untuk dapat mendiagnosa dengan cepat. Alat bantu ini menggunakan metoda Convolution Neural Network (CNN) dengan cara mendeteksi image yang dihasilkan dari peralatan radiologi Computer Tomography Scanner (CT scan). CNN merupakan salah

satu cabang *artificial intelligence* (AI) yang mampu menganalisa image.

Pemanfaatan metode AI dapat membantu memudahkan para medis untuk dapat mendiagnosa pasien terpapar covid-19 dengan cepat dari data image chest x-ray. Umumnya diagnosa dan interpretasi data image chest x-ray, dilakukan oleh orang dengan keahlian dan kemampuan khusus dari para tenaga medis. Terbatasnya jumlah tenaga medis dibandingkan dengan meledaknya jumlah pasien di era pandemi ini, memerlukan waktu 1-2 minggu untuk dapat mengetahui hasil diagnosa yang diinginkan. Keberadaan alat bantu berbentuk perangkat lunak (*software*) menggunakan model matrik hasil training dari metode ini, dengan hanya merekam gambar CT scan maka dalam hitungan detik dapat segera diketahui hasil diagnosa yang diinginkan. Perangkat lunak ini dapat dipergunakan oleh setiap orang (tenaga medis atau non medis), dimanapun (anywhere), kapanpun (anytime), murah (free charge), cepat dan akurasi lebih dari 90%.

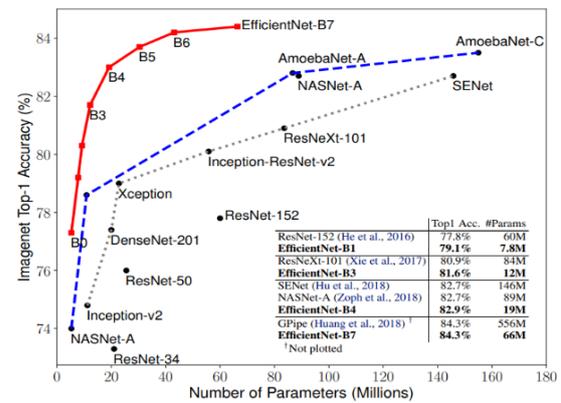
Pengolahan data image dengan menggunakan CNN memerlukan performa komputer yang memadai. Untuk memperoleh kualitas akurasi yang tinggi diperlukan arsitektur dengan layer yang semakin terus bertambah, dan tentunya disesuaikan dengan peningkatan performa komputer menggunakan *Graphics Processing Unit* (GPU). Produk hardware yang dilengkapi dengan GPU relatif lebih mahal, hal tersebut menjadi tantangan bagi penelitian perorangan.

Sampai saat ini masih terus dilakukan penelitian untuk memperoleh jenis arsitektur yang baik (akurasi tinggi dan operasi minimum)[3]. Karena peningkatan performa komputer yang semakin tinggi, untuk meningkatkan akurasi, banyak arsitektur baru lebih mengarah kepertambahan layer yang semakin dalam dari pada design arsitekturnya[4]–[10].

Convolution Neural Network (CNN) adalah metoda untuk mendeteksi objek atau image (gambar) dengan cara mempelajari (training) sejumlah besar data set image secara berulang-ulang. Metoda belajar ini dilakukan hingga memperoleh suatu model

yang diharapkan. Kemudian dengan model ini dapat dipergunakan untuk mendeteksi (klasifikasi) suatu image data testing yang belum diketahui labelnya.

Untuk memperoleh hasil akurasi tinggi dipergunakan arsitektur EfficientNet model B7 dan ukuran input disesuaikan dengan kebutuhan library tensor flow. Dibanding EfficientNet versi sebelumnya (B0-B6), maka EfficientNet B7 mempunyai akurasi lebih tinggi (Gambar 2).

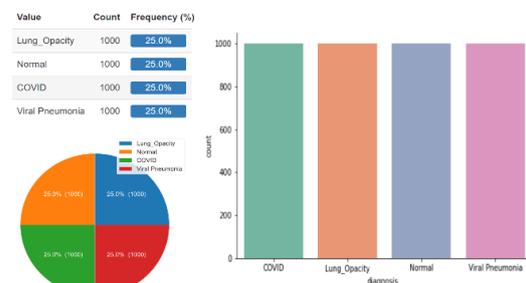


Gambar 2. Komparasi EfficientNet dengan model CNN lainnya[11]

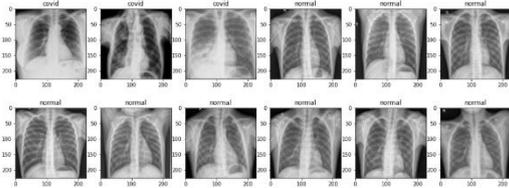
METODE PENELITIAN

Sebanyak 4.000 dataset image [12] dipergunakan dalam penelitian ini. Secara klinis setiap image chest-xray di klasifikasikan ke dalam 4 kelas dengan jumlah dataset tiap kelas sebagai berikut [13] (Gambar 3):

- 0 – Normal 1.000 (25%)
- 1 – Viral Pneumonia 1.000 (25%)
- 2 – Lung Opacity 1.000 (25%)
- 3 – Covid 1.000 (25%)

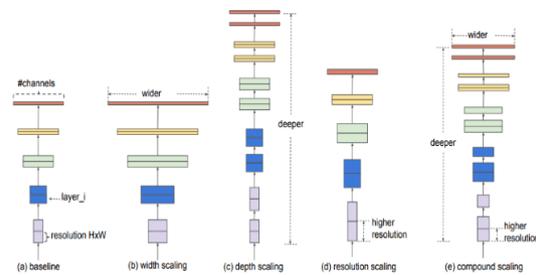


Gambar 3. Pie dan histogram dataset



Gambar 4. Dataset Image Chest X-Ray

Telah menjadi hal yang umum untuk meningkatkan akurasi adalah dengan menambah sumberdaya yang ada. Namun EfficientNet adalah model yang sistematis, mengidentifikasi secara berhati-hati untuk mempertimbangkan kedalaman, lebar jaringan dan resolusi yang dapat menuju performansi lebih baik. (Gambar 5). Tan dan Lee [14] adalah orang yang telah memperkenalkan model yang sangat efisien ini. Representasi kombinasi efisien dan akurasi yang baik dengan tingkat skala yang bervariasi, arsitektur ini melahirkan model dari B0 sampai B7.



Gambar 5. Model scaling EfficientNet [14]

Pengembangan arsitektur didefinisikan dengan :

$$\mathcal{N} = \bigodot_{i=1 \dots s} \mathcal{F}_i^{L_i} (X_{(H_i, W_i, C_i)})$$

$$\max_{d, w, r} \text{Accuracy}(\mathcal{N}(d, w, r))$$

$$s.t. \quad \mathcal{N}(d, w, r) = \bigodot_{i=1 \dots s} \hat{\mathcal{F}}_i^{d \cdot L_i} (X_{(r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i)})$$

$$\text{Memory}(\mathcal{N}) \leq \text{target_memory}$$

$$\text{FLOPS}(\mathcal{N}) \leq \text{target_flops}$$

$\mathcal{F}_i^{L_i}$ adalah layer F ke i diulang sebanyak L sesuai tahap i, (H_i, W_i, C_i) menunjukkan input tensor X dari lapisan i. Model skala akan berusaha untuk memperluas panjang (Li), Lebar (Wi) dan resolusi (Hi, Wi) tidak mempengaruhi besar Fi yang telah ditetapkan sebelumnya [15]. (Gambar 6)

Layer (type)	Output Shape	Param #
efficientnetb7 (Functional)	(None, 7, 7, 2560)	64097687
dropout_6 (Dropout)	(None, 7, 7, 2560)	0
flatten_2 (Flatten)	(None, 125440)	0
batch_normalization_8 (Batch Normalization)	(None, 125440)	501760
dense_8 (Dense)	(None, 256)	32112896
batch_normalization_9 (Batch Normalization)	(None, 256)	1024
activation_6 (Activation)	(None, 256)	0
dropout_7 (Dropout)	(None, 256)	0
dense_9 (Dense)	(None, 128)	32896
batch_normalization_10 (Batch Normalization)	(None, 128)	512
activation_7 (Activation)	(None, 128)	0
dropout_8 (Dropout)	(None, 128)	0
dense_10 (Dense)	(None, 32)	4128
batch_normalization_11 (Batch Normalization)	(None, 32)	128
activation_8 (Activation)	(None, 32)	0
dense_11 (Dense)	(None, 4)	132
Total params: 96,751,163		
Trainable params: 37,736,964		
Non-trainable params: 59,014,199		

Gambar 6. Layer EfficientNet B7 [14]

Pengembangan tiap model sistematis ini menghasilkan akurasi yang semakin baik (Gambar 7). Dengan mempertimbangkan nilai input yang digunakan dalam tiap model. Maka dalam penelitian ini digunakan model EfficientNet B7 dengan ukuran input (224x224) yang dapat digunakan untuk library tensor flow.

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	77.1%	93.3%	5.3M	1x	0.39B	1x
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	79.1%	94.4%	7.8M	1x	0.70B	1x
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	80.1%	94.9%	9.2M	1x	1.0B	1x
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	81.6%	95.7%	12M	1x	1.8B	1x
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	82.9%	96.4%	19M	1x	4.2B	1x
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	83.6%	96.7%	30M	1x	9.9B	1x
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	84.0%	96.8%	43M	1x	19B	1x
EfficientNet-B7	84.3%	97.0%	66M	1x	37B	1x
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

We unit ensemble and multi-crop models (Hu et al., 2018), or models pretrained on 3.5B Instagram images (Majumdar et al., 2018).

Gambar 7. Hasil kemampuan tiap model EfficientNet [16]

Untuk menghindari overfitting atau underfitting karena pemakaian epoch yang berlebih atau terlalu kecil, adalah masalah yang sering dihadapi dalam memilih nilai

training epoch yang akan digunakan. Dalam program yang dibuat dalam penelitian ini menggunakan metoda early stopping[17]. Metode ini untuk menghindari pemakaian epoch yang besar dan akan berhenti melakukan training ketika kemampuan model sudah tidak dapat ditingkatkan lagi. Untuk menghindari epoch yang besar nilai learning rate akan berkurang sesuai dengan nilai faktor yang ditentukan (0-1) dan proses training akan berhenti setelah mencapai nilai minimum learning rate yang ditentukan. Karena dengan nilai learning rate yang sudah sangat kecil kemampuan model sudah tidak dapat berubah atau tidak meningkat lagi.

```

steps_per_epoch = train_generator.samples // train_generator.batch_size
validation_steps = valid_generator.samples // valid_generator.batch_size
hist = model.fit(
    train_generator,
    epochs=epochs, steps_per_epoch=steps_per_epoch,
    validation_data=valid_generator,
    validation_steps=validation_steps).history

```

```

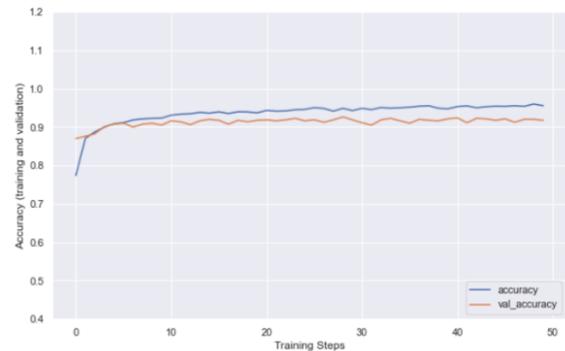
0.9225
Epoch 45/50
100/100 [=====] - 240s 2s/step - loss: 0.3771 - accuracy: 0.9
0.9262
Epoch 46/50
100/100 [=====] - 246s 2s/step - loss: 0.3805 - accuracy: 0.9
0.9225
Epoch 47/50
100/100 [=====] - 244s 2s/step - loss: 0.3771 - accuracy: 0.9
0.9200
Epoch 48/50
100/100 [=====] - 241s 2s/step - loss: 0.3750 - accuracy: 0.9
0.9258
Epoch 49/50
100/100 [=====] - 241s 2s/step - loss: 0.3805 - accuracy: 0.9
0.9237
Epoch 50/50
100/100 [=====] - 245s 2s/step - loss: 0.3769 - accuracy: 0.9
0.9175

```

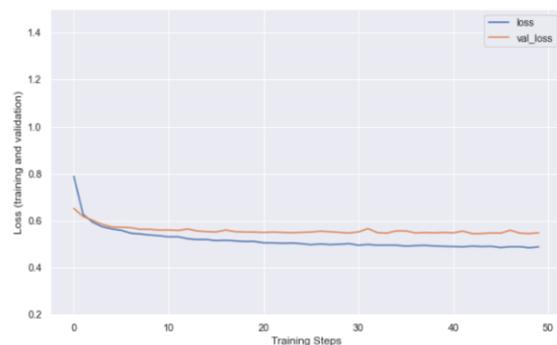
Gambar 8. Akurasi training 99,5% dan validasi 91,8%

Dengan mempertimbangkan keterbatasan infrastruktur dan durasi waktu training, maka penelitian menggunakan jumlah data 4.000 image ini hanya dilakukan 50 epoch dengan waktu training \pm 4 jam (50 epoch x 245 s). Pada saat akhir training (epoch 50) diperoleh nilai akurasi training dan validasi sangat baik yaitu masing 99,5% dan 91,8%. (Gambar 8).

History tiap epoch dan perubahan nilai loss, accuracy training dan loss, accuracy validasi dapat dilihat dalam bentuk grafik. Dari grafik history terlihat dalam setiap penambahan epoch, akurasi training dan validasi cenderung menuju ke nilai lebih dari 90% (Gambar 9), dan loss training dan validasi menuju ke nilai kurang dari 60% (Gambar 10).



Gambar 9. Grafik history akurasi training dan testing



Gambar 10. Grafik history Loss Value training dan testing

HASIL DAN PEMBAHASAN

Setelah proses training selesai, tahap selanjutnya adalah melakukan pengujian menggunakan 4.000 data image chest-xray, terdiri dari 4 kelas masing-masing berjumlah 1000 image.

Hasil pengujian dengan menggunakan data testing, mesin learning berhasil memprediksi dengan baik data dari image chest xray dengan penderita

- COVID adalah 952 dari 990, sehingga presisinya sebesar 96%.
- Lung Opacity adalah 969 dari 1.043, sehingga presisinya sebesar 93%.
- Normal adalah 943 dari 988, sehingga presisinya sebesar 95%.
- Pneumonia , 960 dari 979, sehingga presisinya sebesar 98%.

Detail perhitungan untuk mengukur presisi lainnya ditunjukkan dalam bentuk tabel 1, sebagai berikut :

Tabel 1. Penjelasan report klasifikasi

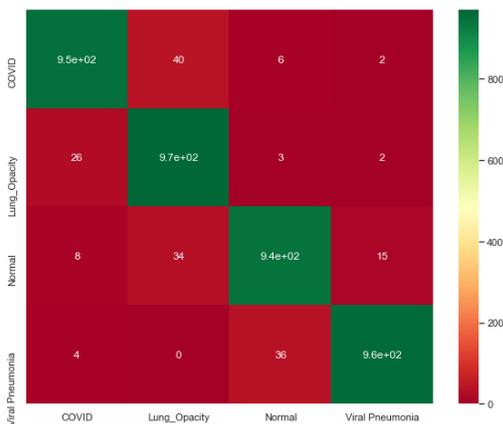
COVID	A
Lung Opacity	B
Normal	C
Viral Pneumonia	D

		Sebenarnya				Support
		A	B	C	D	
Pred	A	952	40	6	2	1.000
	B	26	969	3	2	1.000
	C	8	34	943	15	1.000
	D	4	0	36	960	1.000
		990	1.043	988	979	4.000

	Precision	Recall	f1-score
A	$952/990=0.96$	$952/1000=0.95$	$2*952/(990+1000)=0.96$
B	$969/1043=0.93$	$969/1000=0.97$	$2*969/(1043+1000)=0.95$
C	$943/988=0.95$	$943/1000=0.94$	$2*943/(988+1000)=0.95$
D	$960/979=0.98$	$960/1000=0.96$	$2*960/(979+1000)=0.97$

Accuracy = $(952+969+943+960)/4000 = 0.96$

```
from sklearn.metrics import confusion_matrix, accuracy_score
import seaborn as sn
print(accuracy_score(y_test_class, y_pred))
confusion_matrix(y_test_class, y_pred)
0.956
array([[952, 40, 6, 2],
       [26, 969, 3, 2],
       [8, 34, 943, 15],
       [4, 0, 36, 960]], dtype=int64)
```



Gambar 10. Matrik Confusion

Modul sklearn python dapat dipergunakan untuk mengukur akurasi dan memvisualisasikan matrik confusion seperti yang ditampilkan dalam Gambar 10 dan 11.

	precision	recall	f1-score	support
COVID	0.96	0.95	0.96	1000
Lung Opacity	0.93	0.97	0.95	1000
Normal	0.95	0.94	0.95	1000
Viral Pneumonia	0.98	0.96	0.97	1000
accuracy	0.96			4000

Gambar 11. Report Klasifikasi

PENUTUP

- Deteksi image chest xray dengan menggunakan CNN arsitektur EfficientNet-B7 menghasilkan akurasi yang sangat baik yaitu 96%.
- Hasil prediksi 4 kelas yaitu COVID, Lung Opacity, Normal, Viral Pneumonia dengan presisi masing-masing 96%, 93%, 95% dan 98%.
- Hasil akurasi dapat ditingkatkan dengan menambah jumlah data, sehingga mesin menjadi lebih pandai lagi.
- Masih diperlukan penelitian untuk menggunakan arsitektur yang tidak hanya menghasilkan akurasi yang tinggi, tapi juga waktu training minim dan dapat dilakukan dengan performance komputer yang lebih sederhana.
- Model matrik yang dihasilkan dari proses training dapat dipergunakan (*embedding*) di perangkat lunak diagnosa covid. Perangkat lunak dapat menjadi lebih ringan karena ukuran model yang kecil dan waktu operasi cepat karena tidak memerlukan proses training lagi.

DAFTAR PUSTAKA

[1] Rebecca Conway, "Tsunami COVID-19 di India Merangsek Hingga ke Pedalaman," health.detik.com, 2021. [Online]. Available: <https://health.detik.com/fotohealth/d-5582121/tsunami-covid-19-di-india-merangsek-hingga-ke-pedalaman>.

[2] worldometers, "Coronavirus Cases in India," Worldometers, 2021. [Online]. Available: <https://www.worldometers.info/coronavirus/country/india/>.

- [3] L. M. Rasdi Rere, M. I. Fanany, and A. M. Arymurthy, "Metaheuristic Algorithms for Convolution Neural Network," *Comput. Intell. Neurosci.*, vol. 2016, 2016.
- [4] W. Karno, A. S. B., & Hastomo, "Optimalisasi Data Terbatas Prediksi Jangka Panjang Covid-19 Dengan Kombinasi Lstm Dan GRU," *Pros. SeNTIK*, vol. 4, no. September, pp. 181–191, 2020.
- [5] A. S. B. Karno, W. Hastomo, and I. S. K. Wardhana, "Prediksi Jangka Panjang Covid-19 Indonesia Menggunakan Deep Learning Long-Term," *Semin. Nas. Teknol. Inf. dan Komun.*, pp. 483–490, 2020.
- [6] W. Hastomo, A. S. Bayangkari Karno, N. Kalbuana, A. Meiriki, and Sutarno, "Characteristic Parameters of Epoch Deep Learning to Predict Covid-19 Data in Indonesia," *J. Phys. Conf. Ser.*, vol. 1933, no. 1, p. 012050, 2021.
- [7] W. Hastomo and A. Satyo, "Long Short Term Memory Machine Learning Untuk Memprediksi Akurasi Nilai Tukar IDR Terhadap USD," vol. 3, 2019.
- [8] W. Hastomo and A. Satyo, "Kemampuan Long Short Term Memory Machine," vol. 4, no. September, pp. 229–236, 2020.
- [9] A. S. B. Karno, W. Hastomo, & Arif, D., and E. S. Moreta, "Optimasi Portofolio Dan Prediksi Cryptocurrency Menggunakandep Learning Dalam Bahasa Python," vol. 4, no. September, 2020.
- [10] W. Hastomo, "Gesture Recognition For Pencak Silat Tapak Suci Real-Time Animation," vol. 2, pp. 91–102, 2020.
- [11] R. M. Jomaa, H. Mathkour, Y. Bazi, and M. S. Islam, "End-to-end deep learning fusion of fingerprint and electrocardiogram signals for presentation attack detection," *Sensors (Switzerland)*, vol. 20, no. 7, pp. 1–17, 2020.
- [12] S. Sarkar, "COVID 19 XRay and CT Scan Image Dataset," *kaggle.com*, 2020. [Online]. Available: <https://www.kaggle.com/ssarkar445/covid-19-xray-and-ct-scan-image-dataset>.
- [13] D. S. Sisodia, S. Nair, and P. Khobragade, "Diabetic retinal fundus images: Preprocessing and feature extraction for early detection of Diabetic Retinopathy," *Biomed. Pharmacol. J.*, vol. 10, no. 2, pp. 615–626, 2017.
- [14] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.
- [15] L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Simulated Annealing Algorithm for Deep Learning," *Procedia Comput. Sci.*, vol. 72, pp. 137–144, 2015.
- [16] G. Marques, D. Agarwal, I. De, and T. Díez, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," no. January, 2020.
- [17] Jason Brownlee, "Use Early Stopping to Halt the Training of Neural Networks At the Right Time," *machinelearningmastery*, 2020. [Online]. Available: <https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/>.