

# KLASIFIKASI DEEP LEARNING UNTUK COVID-19 DENGAN CITRA CXR

Rafi Alvanzah, Syarif Abyan Shahab, Javier Bintoro

Program Studi Matematika, Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan  
Alam, Universitas Indonesia  
Kampus UI Depok, 16424, Indonesia

## Abstrak

Pandemi Coronavirus Disease 2019 (COVID-19) memberikan dampak serius pada kehidupan masyarakat. Untuk mengatasi pandemi COVID-19 diperlukan kemampuan untuk mendeteksi pasien yang terjangkit COVID-19 sedini mungkin agar pasien dapat segera ditangani secara medis dan memutus rantai penyebaran virus. Kebutuhan alat tes COVID-19 yang terus meningkat tidak sebanding dengan supply yang terbatas, hal ini menyebabkan terjadinya kelangkaan alat tes COVID-19. Oleh karena itu, diperlukan cara alternatif untuk mendeteksi COVID-19. Penggunaan X-ray dan CT scan dapat memberikan informasi penting dalam mendeteksi COVID-19 karena COVID-19 adalah virus yang menginfeksi saluran pernapasan. Teknik *data mining* dan *machine learning* terbaru seperti *Convolutional Neural Network* (CNN) dapat diaplikasikan pada citra X-ray dan CT scan paru-paru untuk mendeteksi COVID-19 dengan cepat. Pada studi ini telah dihasilkan model CNN 11-layer yang menerima input berupa citra CXR dan telah diuji menggunakan dataset COVID-19 RADIOGRAPHY DATABASE. Model CNN 11-layer ini mampu melakukan klasifikasi biner antara CXR normal dan terinfeksi COVID-19 dengan akurasi 95.1%, *precision* 95%, dan *recall* 95.5%.

**Keywords:** Classification, Coronavirus, COVID-19, CXR images, CNN, Deep Learning

## 1. Pendahuluan

Coronavirus Disease 2019, atau singkatnya COVID-19, adalah penyakit virus yang menyebabkan pneumonia serius dan berdampak pada berbagai bagian tubuh dari ringan hingga parah tergantung pada sistem imunitas pasien, terdapat beberapa gejala yang biasa ditimbulkan antara lain: demam, batuk, kehilangan rasa serta bau, sakit tenggorokan, bahkan hingga kesulitan bernafas dan nyeri dada. Infeksi ini pertama kali dilaporkan di kota Wuhan China pada Desember 2019, dan kemudian menjadi pandemi global yang menyebar dengan cepat ke seluruh dunia. Dengan di tetapkannya COVID-19 sebagai pandemi oleh WHO maka tertutupnya jalur perdagangan dunia yang mengakibatkan salah satu krisis ekonomi terbesar. Penutupan jalur perdagangan ini juga dipengaruhi oleh penutupan negara-negara yang melakukan *full lockdown* dalam langkah mencoba untuk mengendalikan kenaikan kasus positif di negara tersebut. Virus ini pertama kali terdeteksi masuk ke Indonesia pada awal bulan Maret 2020, dengan terdeteksinya 2 pasien yang positif dengan COVID-19. Tak lama kemudian pada bulan April sudah terdeteksi penyebaran virus ini sampai di semua provinsi di Indonesia.

Oleh karena itu, seiring dengan meningkatnya kasus COVID-19 ini, meningkat pula kebutuhan mendeteksi infeksi dengan cepat dan mudah. Selain meningkatnya kebutuhan untuk mendeteksi COVID-19 dengan cepat dibutuhkan juga yang dapat mendeteksi dengan benar, yang mendapatkan hasil test cepat, mudah dan akurat tanpa banyak *false positive*, dan juga *false negative*, yang sudah terbukti penggunaan rapid test antigen dapat mendeteksi dengan cepat namun masi rawan terjadi *false positive*. Dalam situasi ini, penelitian terbaru menggunakan teknik *radiology imaging* (seperti X-ray dan CT scan) terbukti dapat membantu mendeteksi COVID-19 karena gambar X-ray dan CT scan yang dapat memberikan informasi penting tentang penyakit yang disebabkan COVID-19. Teknik *data mining* dan *machine learning* seperti *Convolutional Neural Network* (CNN) dapat diterapkan bersama dengan gambar sinar-X dan CT scan paru-paru untuk mendeteksi penyakit yang akurat dan cepat, membantu mengurangi masalah kelangkaan pengujian kit.

## 2. Dataset

Dataset yang digunakan yaitu COVID-19 RADIOGRAPHY DATABASE. Data yang terlampirkan pada dataset tersebut dikumpulkan dan di-publish oleh sebuah tim peneliti dari Qatar University, Doha, Qatar, dan University of Dhaka, Bangladesh beserta kolaboratornya dari Pakistan dan Malaysia yang telah bekerja sama dengan dokter medis dalam penyediaan database gambar X-ray dada (CXR) untuk kasus positif COVID-19 beserta Gambar Pneumonia Viral dan Normal. Dataset infeksi paru-paru COVID-19, normal, dan lainnya ini berisikan 3.616 kasus positif COVID-19, 10.192 kasus Normal, 6.012 Kasus Paru (Infeksi paru-paru non-COVID-19), dan 1.345 kasus Pneumonia Viral beserta *mask* paru-paru yang sesuai. Format gambar pada dataset Portable Network Graphics (PNG) dengan setiap gambar memiliki resolusi  $299 \times 299$  piksel.

Deteksi COVID-19 dilakukan terhadap beberapa kontrol gambar yang berupa penyakit dada yang terinfeksi dan sehat. Oleh karena itu, dilakukan masalah klasifikasi biner. Yang terbagi atas virus COVID dan normal, dan memiliki perbandingan train test nya 8:2. Data kasus antara COVID-19 dan Normal memiliki data yang *imbalance* dimana terdapat ketimpangan antara 2 kelas (COVID-19 memiliki 3.616 kasus dan normal memiliki 10.192 kasus).

Data	Jumlah Gambar
Normal	10.192
COVID-19	3.616
Lung Opacity	6.012
Pneumonia	1.345

**Tabel 2.1:** Dataset COVID-19 RADIOGRAPHY DATABASE

Karena terjadinya *imbalance* pada dataset yang digunakan maka perlu dilakukan *resampling* data. *Resampling* data dilakukan agar bisa mendapatkan model yang lebih baik dan data menjadi lebih seimbang. *Resampling* data yang dapat dilakukan adalah *undersampling*, *oversampling*, dan *Ensembling method*. Pada kasus ini akan dilakukan SMOTE (*Synthetic Minority Over-sampling Technique*) pada data minoritas dan juga *random undersampling* pada data mayoritas. Proses *resampling* akan dijelaskan lebih lanjut pada bagian Implementasi.

### 3. Metode

Pada bagian ini akan dijelaskan metode yang digunakan dalam studi klasifikasi biner COVID-19 yang terdiri dari 11-layer model CNN untuk mendeteksi COVID-19 menggunakan X-ray dada (CXR). Performa model dievaluasi berdasarkan *accuracy*, *precision*, *recall*, *F1 score*, dan *confusion matrix*.

#### 3.1. Preprocessing

##### 3.1.1. Grayscale

Citra digital berwarna atau RGB triple-channel, diubah menjadi *grayscale image* dengan single-channel. Proses *grayscale* menggunakan metode *luminance* yang dapat dihitung dengan rumus berikut,

$$G_{Luminance} = (0.3 * R) + (0.59 * G) + (0.11 * B)$$

$G_{Luminance}$  adalah citra *grayscale*, R adalah layer pertama citra RGB, G adalah layer kedua citra RGB, dan B adalah layer ketiga citra RGB. Proses *grayscale* yang direpresentasikan dengan 8-bit dilakukan pada citra karena membuat proses komputasi lebih murah dibanding RGB yang direpresentasikan dengan 64-bit. *Luminance* juga yang paling berpengaruh sebagai pembeda dalam citra.

##### 3.1.2. Resize

*Resize* adalah proses pengubahan ukuran citra, baik menjadi lebih besar maupun lebih kecil. Dalam studi ini dilakukan *resize* untuk mengubah ukuran citra menjadi lebih kecil, hal ini dilakukan untuk mempermudah proses komputasi.

##### 3.1.3. Vektorisasi

Vektorisasi dilakukan untuk mengubah citra menjadi matriks/tensor yang nilai-nilainya terdiri dari koefisien keabu-abuan. Vektorisasi dilakukan karena CNN menerima input berupa tensor.

##### 3.1.4. Normalisasi

Normalisasi adalah proses mengubah nilai pada tensor agar memiliki nilai pada interval [0,1]. Normalisasi dilakukan untuk mempermudah proses pembelajaran pada model.

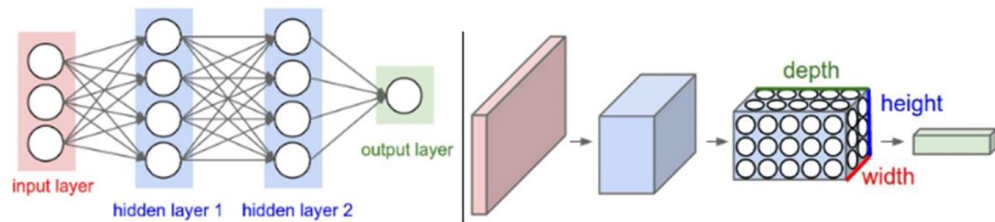
##### 3.1.5. Resampling

*Resampling* dilakukan untuk menyeimbangkan data. Terdapat beberapa teknik *resampling*, yaitu *oversampling*, *undersampling* dan sintesis. Pada kasus ini akan dilakukan SMOTE (*Synthetic Minority Over-sampling Technique*) pada data minoritas dan juga *random undersampling* pada data mayoritas. Kedua metode ini dilakukan berdasarkan rekomendasi dari pencipta metode SMOTE yang menyatakan bahwa performa akan lebih baik dibanding melakukan hanya *undersampling* saja. Metode SMOTE dilakukan untuk menambah jumlah data minoritas dengan cara membangkitkan data sintesis. Selanjutnya, metode *random undersampling* secara acak memilih observasi dari kelas mayoritas untuk dieliminasi hingga dataset menjadi seimbang.

#### 3.2. Convolutional Neural Network (CNN)

Dalam beberapa tahun ke belakang, penggunaan *deep learning* khususnya *Convolutional Neural Network* (CNN) berhasil mengungguli performansi *machine learning* tradisional dalam bidang klasifikasi citra, tidak terkecuali citra medis. Banyak variasi arsitektur jaringan yang dibangun dari CNN, antara lain Inception, VGGNet, MobileNet, Densenet, dsb. CNN pada dasarnya adalah susunan banyak layer yang terdiri dari *convolution layer*, *pooling layer*, dan *fully connected layer*. Berbeda dengan jaringan syaraf tiruan biasa seperti yang terlihat pada Gambar 3.2.1, CNN mengatur neuron sehingga memiliki tiga dimensi (lebar, tinggi

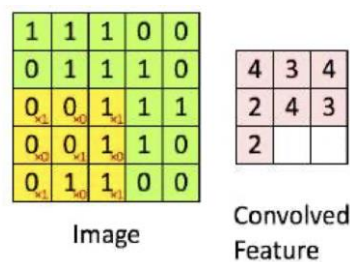
dan kedalaman) yang pada Gambar 3.2.1 didefinisikan sebagai satu layer. Setiap layer pada CNN mentransformasi input 3D menjadi output 3D dari aktivasi neuron.



**Gambar 3.2.1:** Dataset COVID-19 RADIOGRAPHY DATABASE

### 3.3. Convolution Layer

*Convolution Layer* melakukan operasi konvolusi pada output dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra di semua offset yang memungkinkan seperti yang ditunjukkan pada Gambar 3.3.1. Kotak hijau secara keseluruhan adalah citra yang akan di konvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.



**Gambar 3.3.1:** Ilustrasi konvolusi

### 3.4. Pooling Layer

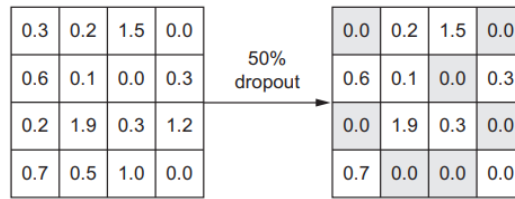
*Pooling layer* digunakan untuk mengurangi ukuran spasial dengan tujuan mengurangi jumlah parameter dan komputasi, selain itu juga untuk menghindari kondisi *overfitting* dimana model sangat akurat memprediksi data latih namun gagal mengenali data di luar data latih. *Feature map* menjadi lebih kecil sehingga mudah dikelola dan mudah mengontrol *overfitting*. Ada beberapa teknik yang dapat digunakan, diantaranya *max pooling*, dan *average pooling*. Pada studi ini digunakan *max pooling*.



**Gambar 3.4.1:** Ilustrasi *Pooling Layer*

### 3.5. Dropout Layer

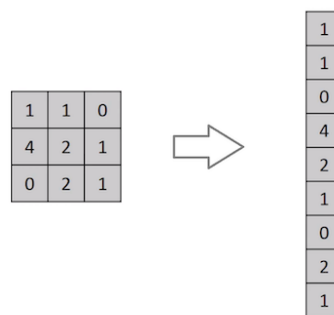
Salah satu cara paling efektif dalam mencegah *overfitting* adalah dengan menambah *dropout layer*, ia berfungsi sebagai penambah *noise* pada *feature map* agar model tidak mempelajari pola-pola yang dapat menciptakan *overfitting*. Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses learning. Dropout mengacu kepada menghilangkan neuron yang berupa *hidden layer* maupun yang *visible* di dalam *network*. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara neuron dari *network* yang ada.



**Gambar 3.5.1:** Ilustrasi *Dropout Layer*

### 3.6. Flatten Layer

*Flatten layer* berfungsi untuk mengubah *feature map* menjadi 1D agar bisa menjadi input untuk *fully connected layer (dense layer)*. Dalam pengertian yang lebih sederhana proses *Flatten Layer* merupakan proses simplifikasi layer dari awalnya 3x3 (3 kolom, 3 baris) menjadi satu kolom. Fungsinya adalah untuk memudahkan proses analisis.



**Gambar 3.6.1:** Ilustrasi *Flatten Layer*

### 3.7. Fungsi Aktivasi

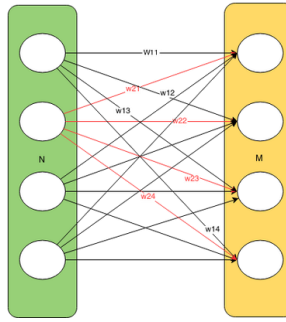
Fungsi Aktivasi secara khusus digunakan dalam jaringan saraf tiruan untuk mengubah sinyal input menjadi sinyal output yang pada gilirannya diumpankan sebagai input ke lapisan berikutnya. Pada studi ini digunakan fungsi aktivasi sigmoid dan ReLU.

Fungsi aktivasi sigmoid disebut juga fungsi logistik. Ini adalah fungsi yang sama digunakan dalam algoritma klasifikasi biner. Fungsi mengambil nilai riil apa pun sebagai nilai input dan output dalam rentang 0 hingga 1. Dirumuskan sebagai,  $f(x) = \frac{1}{1+e^{-x}}$ .

ReLU (*Rectified Linear Unit*) merupakan *nonsaturating activation function* yang dapat meningkatkan nonlinearitas *decision function* dan *network* secara keseluruhan, tanpa harus mempengaruhi bidang-bidang reseptif pada *convolution layer*. ReLU sangat efektif untuk menghapus nilai negatif pada *feature/activation map* dan dijadikan 0. Dirumuskan sebagai,  $f(x) = \max(0, x)$ .

### 3.8. Fully Connected Layer

Layer *Fully-connected* adalah lapisan yang semua neuron aktif lapisan sebelumnya terhubung dengan neuron lapisan berikutnya sehingga dapat diimplementasikan oleh jaringan syaraf tiruan. Setiap aktivitas dari lapisan sebelumnya harus diubah menjadi data 1D sebelum dapat ditautkan ke semua neuron di lapisan *Fully-Connected*. Layer ini adalah layer yang biasanya digunakan dalam penerapan *multi layer perceptron* dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.



**Gambar 3.8.1:** Ilustrasi *Fully Connected Layer*

### 3.9. Optimizer (Adam)

Algoritma optimasi Adam (*Adaptive Moment Estimation*) mampu meningkatkan kecepatan konvergensi dan menemukan minimum yang lebih baik untuk fungsi loss dibandingkan dengan algoritma aslinya. Solusi yang diusulkan meminjam beberapa ide dari pengoptimal berbasis momentum dan teknik peluruhan eksponensial.

### 3.10. Loss Function (BCE)

Binary Cross Entropy atau yang biasa disebut dengan log loss adalah rata-rata negatif dari log dari prediksi probabilitas yang dikoreksi, BCE membandingkan masing-masing probabilitas yang diprediksi dengan output artinya seberapa dekat atau jauh dari nilai sebenarnya. BCE dapat didefinisikan dengan,  $Log\ loss = \frac{1}{N} \sum_{i=1}^N -(y_i * \ln p_i + (1 - y_i) * \ln 1 - p_i)$ .

### 3.11. Confusion Matrix

*Confusion matrix* juga sering disebut *error matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui. Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN).

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

**Gambar 3.11.1:** Ilustrasi *Confusion Matrix*

### 3.12. Precision

*Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Nilai *precision* dapat diperoleh dengan cara,  $Precision = \frac{TP}{TP+FP}$ .

### 3.13. Recall

*Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Recall dapat diperoleh dengan cara,  $Recall = \frac{TP}{TP+FN}$ .

### 3.14. F1 Score

*F1 Score* digunakan untuk mengukur keseimbangan antara Precision – Recall. *F1 score* dapat diperoleh dengan cara,  $F_1 = \frac{TP}{TP + \frac{1}{2}(FP+FN)}$ .

## 4. Simulasi Hasil dan Analisa Pembahasan

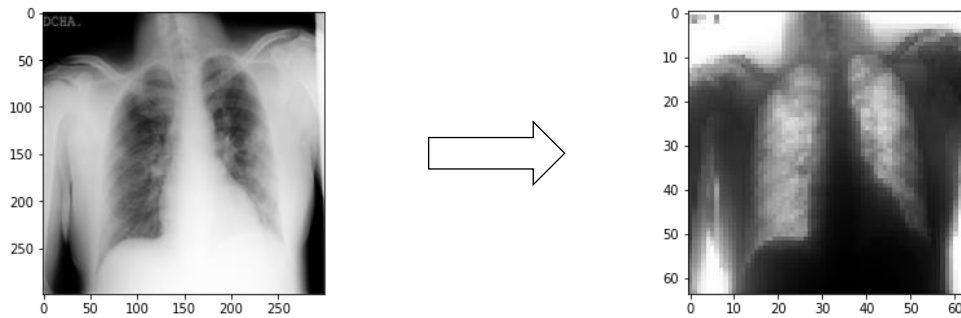
### 4.1. Preprocessing

#### 4.1.1. Grayscale

Proses *grayscale*, citra/gambar CXR digital berwarna atau RGB *triple-layered*, diubah menjadi *grayscale image*. Library PIL yang digunakan untuk mengubah citra menjadi *grayscale* menggunakan metode yang disebut *Luminance* dan diimplementasikan pada modul `ImageOps.grayscale`.

#### 4.1.2. Resize

Pada proses *resize* gambar dari yang semula beresolusi 299x299 piksel menjadi 64x64 piksel. Library PIL juga digunakan pada proses ini yang diimplementasi dengan modul `ImageOps.resize`.



**Figur 4.1.1:** Citra CXR setelah proses *grayscale* dan *resize*

#### 4.1.3. Vektorisasi

Pada proses vektorisasi citra diubah menjadi tensor 3-dimensi (64, 64, 1). Vektorisasi dilakukan menggunakan Library Numpy dengan modul `array`.

#### 4.1.4. Normalisasi

Normalisasi dilakukan dengan cara membagi setiap nilai pada tensor dengan nilai maksimum dari koefisien keabu-abuan, yaitu 255. Lalu setiap nilai pada tensor memiliki nilai antara interval [0, 1].

#### 4.1.5. Data Splitting

*Data splitting* dilakukan dua kali, pertama untuk membagi keseluruhan data menjadi data latih dan data tes, kedua untuk membagi data latih menjadi data latih dan data validasi. Pada pembagian pertama pembagian data menggunakan rasio 8:2, dengan  $\frac{8}{10}$  data untuk data latih dan  $\frac{2}{10}$  data untuk data tes. Pembagian kedua menggunakan rasio 9:1, dengan  $\frac{9}{10}$  data tes untuk data tes dan  $\frac{1}{10}$  data test menjadi data validasi. *Data splitting* dilakukan dengan Library scikit-learn dengan modul `model_selection.train_test_split`.

#### 4.1.6. Resampling

Proses *resampling* dilakukan pada data latih untuk menyeimbangkan kedua kelas (COVID-19 dan normal), pertama dilakukan *oversampling* menggunakan teknik SMOTE pada data minoritas sehingga perbandingan dan dilanjutkan dengan *undersampling* menggunakan teknik *Random-undersampling* pada data mayoritas. *Resampling* dilakukan dengan Library imbalanced-learn dengan modul `over_sampling.SMOTE` untuk proses *oversampling* dan modul `under_sampling.RandomUnderSampler` untuk proses *undersampling*.

Data		Train-Test Split		Data Latih	Data Tes
Normal	10.192	→	Normal	8.135	2.039
COVID-19	3.616		COVID-19	2.893	723

**Figur 4.1.2:** Pembagian data menjadi data latih dan data tes

Data Latih		Data Latih		Data Latih	
Normal	8.135	SMOTE →	Normal	8.135	Random-undersampling →
COVID-19	2.893		COVID-19	5.707	
				Normal	5.707
				COVID-19	5.707

**Figur 4.1.3:** Proses *resampling*

Data Latih		Train-Validation Split	Data Latih		Data Validasi
Normal	5.707	→	Normal	4.615	497
COVID-19	5.707		COVID-19	4.629	531

**Figur 4.1.4:** Pembagian data menjadi data latih dan data validasi

## 4.2. Model Convolutional Neural Network

Model yang digunakan pada studi ini adalah arsitektur CNN yang terdiri dari 11-layer. Rangkuman dari model CNN 11-layer dapat dilihat pada Tabel 4.2.1, dapat dilihat bahwa model terdiri dari tiga layer conv2d dan dua layer dua max\_pooling2d. Ada pula layer dropout untuk meningkatkan *noise* pada model sehingga mengurangi *overfitting*. Untuk proses evaluasi model, digunakan Adam Optimizer (Adaptive Moment Estimation) untuk mengoptimasi beban (*weight*) dan fungsi loss Binary Cross Entropy (BCE). Proses penentuan jumlah layer, fungsi aktivasi, neuron, dan filter pada model dilakukan dengan eksperimenasi secara berulang untuk mendapatkan model dengan hasil paling optimal.

Di awal, model terdiri dari 8-layer dan menghasilkan akurasi 50% dengan menggunakan fungsi aktivasi linear pada output layer, setelah menambah layer conv2d, max\_pooling2d, dan dropout dan menambah pula jumlah filter. Akurasi model meningkat menjadi 90% tetap dengan *running time* selama 46 menit. Selanjut setelah dilakukan beberapa eksperimen dengan mengurangi jumlah filter, unit, dan mengubah fungsi aktivasi pada output layer menjadi sigmoid ditemukan kombinasi layer, fungsi aktivasi, fungsi loss, neuron, filter dengan akurasi 99% pada data latih dan 95% pada data tes.

Layer	Output Shape	Parameter
conv2d (Conv2D)	(62, 62, 32)	320
max_pooling2d (MaxPooling2D)	(31, 31, 32)	0
dropout (Dropout)	(31, 31, 32)	0
conv2d_1 (Conv2D)	(29, 29, 64)	18.496
max_pooling2d_1 (MaxPooling2D)	(14, 14, 64)	0
dropout_1 (Dropout)	(14, 14, 64)	0
conv2d_2 (Conv2D)	(12, 12, 64)	36.928
flatten (Flatten)	9.216	0
dense (Dense)	64	589.888
dropout_2 (Dropout)	64	0
dense_1 (Dense)	1	65
Parameter Total: 645,697		

**Tabel 4.2.1:** Rangkuman arsitektur CNN 11-layer

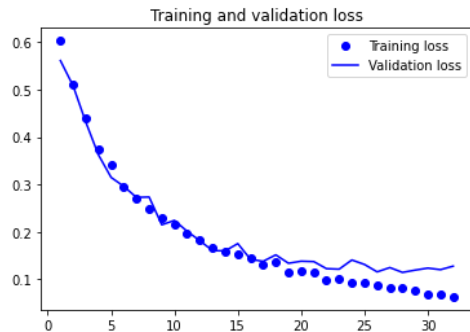
## 4.3. Train Model

Proses *training* model menggunakan data latih dan data validasi untuk proses evaluasi yang ada pada Figur 4.1.4. Model diimplementasikan menggunakan *library* Keras dan Tensorflow. Seluruh proses eksperimen dan *training* dilakukan menggunakan Server Kaggle yang menggunakan GPU NVidia K80. *Training* model dilakukan dengan *learning rate*: 0.001, ukuran *batch*: 256, dan epoch: 50.

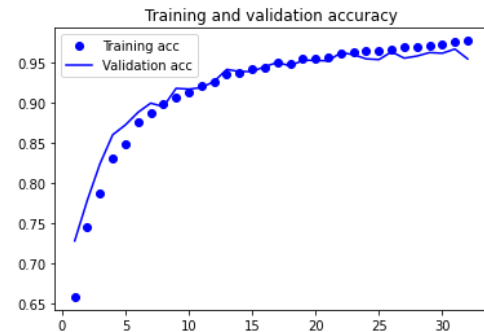
## 4.4. Evaluasi dan Performa Model

Proses evaluasi model CNN 11-layer dapat dilihat dengan membandingkan accuracy dan loss antara data latih dan data validasi yang dapat dilihat pada Gambar 4.4.1 dan Gambar 4.4.2. Dapat dilihat dari gambar bahwa loss yang terjadi pada model tidak terlalu besar (< 0.2) sehingga model dapat dikatakan terlatih

dengan sempurna dan tidak terjadi underfit atau overfit. Performa model lalu diuji mengukur nilai accuracy, precision, recall, F1 score, dan confusion matrix yang dapat dilihat pada Tabel 4.4.3 dan Gambar 4.4.4. Dapat dilihat bahwa performa dari model untuk data latih mendapat nilai hampir sempurna, hal ini dapat diraih karena model terlatih secara sempurna.



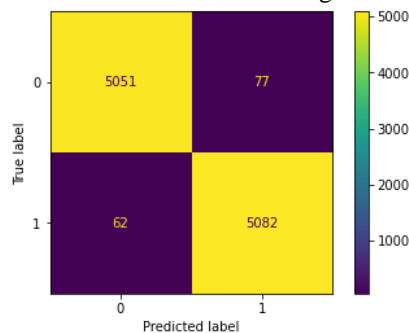
**Gambar 4.4.1:** Evaluasi loss data latih dan data validasi



**Gambar 4.4.2:** Evaluasi *accuracy* data latih dan data validasi

	Data latih		
	Precision	Recall	F1 Score
COVID-19	0.99	0.99	0.99
Normal	0.99	0.99	0.99
Accuracy	0.99		

**Tabel 4.4.3:** Performa model dengan data latih



**Gambar 4.4.4:** *Confusion matrix* dengan data latih

#### 4.5. Prediksi Model dan Pembahasan

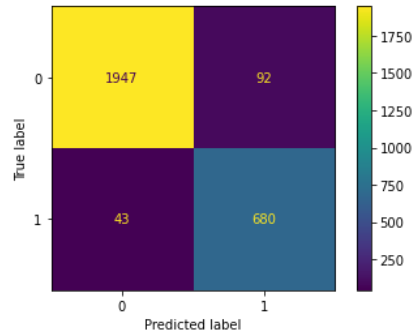
Pada bagian ini akan dilakukan pengujian prediksi model CNN 11-layer yang dilakukan dengan data tes yang diperoleh pada Figur 4.1.2. Performa dari model dengan data tes juga mendapat hasil yang sangat baik dengan melihat Tabel 4.5.1 dan Gambar 4.5.2. Dapat dilihat bahwa model dapat memprediksi dengan baik dengan akurasi 95.1%, *precision* 95%, dan *recall* 95.5%. Hasil prediksi dari model CNN 11-layer dapat mendeteksi COVID-19 lebih akurat dan lebih cepat dibanding tes COVID-19 konvensional, tes PCR memiliki akurasi 95%, tetapi membutuhkan waktu >2 hari dan tes antigen memiliki akurasi 85% dan dengan waktu 15 menit. Meskipun demikian, model yang dihasilkan tidak dilakukan uji klinik sehingga model tidak belum bisa menggantikan tes COVID-19 konvensional, tetapi dapat digunakan untuk membantu pengecekan keakuratan tes COVID-19 konvensional.

Pada studi kali ini, juga tidak dilakukan analisa Grad-CAM sehingga tidak diketahui area CXR mana yang membuatnya terklasifikasi COVID-19 atau normal. Untuk studi kedepannya perlu dilakukan analisa Grad-CAM dan konsultasi dengan ahli medis untuk mengetahui apakah model mengekstrak fitur CXR untuk COVID-19 dengan tepat.

	Data tes		
	Precision	Recall	F1 Score
COVID-19	0.98	0.97	0.98
Normal	0.92	0.94	0.93
Accuracy	0.95		

**Tabel 4.4.3:** Performa model dengan data tes





**Gambar 4.4.3:** *Confusion Matrix* dengan data tes

## 5. Kesimpulan

COVID-19 memiliki dampak yang serius bagi kehidupan masyarakat, kesehatan, dan ekonomi global, diagnosa COVID-19 menjadi hal yang penting untuk mencegah penyebaran virus. Pada studi ini dapat disimpulkan bahwa penerapan *Deep Learning* dapat digunakan untuk menyelesaikan permasalahan di dunia nyata, yaitu untuk melakukan *image classification*. Model CNN (*Convolutional Neural Network*) 11-layer yang dihasilkan dapat digunakan untuk mendeteksi pasien yang menderita COVID-19 dengan menggunakan citra CXR. Model CNN 11-layer yang dihasilkan dapat mendeteksi COVID-19 dengan performa akurasi 95.1%, *precision* 95%, dan *recall* 95.5%.

Meskipun diraih performa yang sangat baik pada pendeteksian COVID-19 dibanding tes COVID-19 konvensional (PCR dan antigen), model yang dihasilkan pada studi ini tidak di melalui uji klinik sehingga belum bisa menggantikan tes COVID-19 konvensional, tetapi dapat digunakan untuk membantu pengecekan keakuratan tes COVID-19 konvensional.

## Referensi

M. E. H. Chowdhury *et al.*, "Can AI Help in Screening Viral and COVID-19 Pneumonia?," in *IEEE Access*, vol. 8, pp. 132665-132676.(2020). doi: 10.1109/ACCESS.2020.3010287.

Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S.B.A., Islam, M.T., Maadeed, S.A., Zughaier, S.M., Khan, M.S. and Chowdhury, M.E..(2020). Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-ray Images.

N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer.(2011). SMOTE: Synthetic Minority Over-sampling Technique. *Journal Of Artificial Intelligence Research*, 16, 321-357.  
<https://doi.org/10.48550/arXiv.1106.1813>.

Kanan, Christopher & Cottrell, Garrison,(2012). Color-to-Grayscale: Does the Method Matter in Image Recognition?. *PloS one*. 7. e29740. 10.1371/journal.pone.0029740.

Razief M.D.(2022). Klasifikasi Penyakit Paru-paru Berbasis Pengolahan Citra X Ray Menggunakan Convolutional Neural Network (classification Of The Lung Diseases Based On X Ray Image Processing Using Convolutional Neural Network). *Journal of eProceedings of Engineering*, Vol 9, No 2.

Hariyani.Y.S.(2020). Deteksi Penyakit Covid-19 Berdasarkan Citra X-Ray Menggunakan Deep Residual Network. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*. Vol 8, No. 2, 443-453.

Athaiya.A. & Sharma.S.(2020). Activation Functions In Neural Network. *International Journal of Engineering Applied Sciences and Technology*, 2020. Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 310-316.

Ange.T. & Roger.N.(2018). Improving Adam Optimizer.

Kuncahyo.S.N.(2019). Confusion Matrix untuk evaluasi model pada supervised learning.

I Wayan Suartika.E.P., Arya.W.Y., & Rully.S.(2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101. *JURNAL TEKNIK ITS* Vol. 5, No. 1, (2016) ISSN: 2337-3539 (2301-9271).

Sahar.D., Saeed.S.G., & Morteza.S.Z.(2021). Real-Time Activity Recognition and Intention Recognition Using a Vision-Based Embedded System.

Iip.S. & Nur.A.(2022). Identifikasi Nomor Rumah Pada Citra Digital Menggunakan Neural Network. *Jurnal METHODIKA*, Vol. 8 No. 1 MARET 2022 P-ISSN : 2442-7861 E-ISSN :2614-314

Rizkiani.I. & Titik.H.(2017). *Komparasi Teknik Resampling Pada Pemodelan Regresi Logistik Biner*.