



## Penerapan *Convolutional Neural Network* pada Citra Rontgen Paru-Paru untuk Deteksi SARS-CoV-2

Bambang Pilu Hartato

Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto  
[bambang.pilu@amikompurwokerto.ac.id](mailto:bambang.pilu@amikompurwokerto.ac.id)

### Abstract

*COVID-19 was officially declared as a pandemic by the WHO on March 11, 2020. For COVID-19, the testing methods commonly used are the Antibody Testing and RT-PCR Testing. Both methods are considered to be the most effective in determining whether a person has been suffered from COVID-19 or not. However, alternative testing methods need to be tried. One of them is using the Convolutional Neural Network. This study aims to measure the performance of CNN in classifying x-ray image of a person's chest to determine whether the person is suffered from COVID-19 or not. The CNN model that was built consists of 1 convolutional 2D layer, 2 activation layers, 1 maxpooling layer, 1 dropout layer, 1 flatten layer, and 1 dense layer. Meanwhile, the chest x-ray image dataset used is the COVID-19 Radiography Database. This dataset consists of 3 classes, i.e. COVID-19 class, NORMAL class, and VIRAL\_PNEUMONIA. The experiments consisted of 4 scenarios and were carried out using Google Colab. Based on the experiments, the CNN model can achieve an accuracy of 98.69%, a sensitivity of 97.71%, and a specificity of 98.90%. Thus, CNN has a very good performance to classify the disease based on a person's chest x-ray.*

*Keywords: convolutional neural network, covid-19, image, chest, x-ray*

### Abstrak

COVID-19 ditetapkan secara resmi sebagai suatu pandemi oleh Badan Kesehatan Dunia pada 11 Maret 2020. Untuk COVID-19, metode pengujian yang umum digunakan adalah *Rapid Test* (Tes Antibodi) dan *Swab Test* (RT-PCR). Kedua metode tersebut dianggap yang paling efektif untuk mengetahui terpapar atau tidaknya seseorang oleh COVID-19. Walaupun demikian, keduanya masih memiliki beberapa kekurangan, di antaranya adalah pada masalah waktu pengujian dan prosedur pengujian yang dilakukan. Sehingga metode pengujian alternatif perlu untuk dicoba. Salah satunya adalah menggunakan algoritma *Convolutional Neural Network* (CNN). Penelitian ini bertujuan untuk mengukur performa akurasi, sensitivitas, dan spesifisitas dari CNN dalam melakukan klasifikasi citra rontgen seseorang untuk mengetahui apakah orang tersebut terpapar COVID-19 atau tidak. Model CNN yang dibangun terdiri dari 1 *convolutional 2D layer*, 2 *activation layer*, 1 *maxpooling layer*, 1 *dropout layer*, 1 *flatten layer*, dan 1 *dense layer*. Sedangkan dataset citra rontgen yang digunakan adalah COVID-19 *Radiography Database*. Dataset ini terdiri dari 3 kelas, yaitu kelas COVID-19 yang terdiri dari 219 citra, kelas NORMAL yang terdiri dari 1341 citra, dan kelas VIRAL\_PNEUMONIA yang terdiri dari 1345 citra. Eksperimen yang dilakukan terdiri dari 4 skenario dan dilakukan menggunakan platform Google Colab. Dari eksperimen yang dilakukan, model yang dibangun dapat mencapai akurasi sebesar 98,69%, sensitivitas sebesar 97,71%, dan spesifisitas sebesar 98,90%. Sehingga dapat disimpulkan bahwa CNN memiliki performa yang sangat baik dalam melakukan klasifikasi citra rontgen seseorang untuk mengetahui apakah orang tersebut terpapar COVID-19 atau tidak.

Kata kunci: citra, *convolutional neural network*, covid-19, paru-paru, rontgen

### 1. Pendahuluan

*Corona Virus Disease 19* (COVID-19) ditetapkan secara resmi sebagai suatu pandemi oleh Badan Kesehatan Dunia (*World Health Organization*) pada 11 Maret 2020 [1]. Jika suatu wabah telah dinyatakan sebagai pandemi, artinya wabah tersebut sudah menyerang hampir seluruh negara di bumi ini dan sudah menjadi ancaman global.

Untuk menanggulangi wabah ini, WHO mengampanyekan gerakan 3T lagi, yaitu *Testing* (Pengujian), *Tracing* (Penelusuran), dan *Treatment* (Pengobatan) [2]. Tiga langkah ini dianggap sukses untuk menekan laju persebaran wabah-wabah sebelumnya. Pengujian dilakukan untuk memeriksa apakah seseorang terpapar COVID-19 atau tidak.

Pengujian dilakukan dengan tahapan-tahapan medis tertentu. Jika seseorang tersebut dinyatakan positif COVID-19 maka akan dilakukan 2 tahapan lainnya, yaitu penelusuran dan pengobatan. Penelusuran dilakukan dengan menelusuri siapa saja yang melakukan kontak erat dengan pasien COVID-19. Sementara pengobatan adalah langkah medis yang dilakukan untuk menekan efek dari COVID-19 pada pasien hingga pasien dinyatakan sembuh [3]. Jika dilihat dari skenarionya, pengujian dianggap menjadi ujung tombak dari 3T.

Untuk COVID-19, metode pengujian yang umum digunakan adalah *Rapid Test* (Tes Antibodi) dan *Swab Test* (RT-PCR). Kedua metode tersebut dianggap yang paling efektif untuk mengetahui terpapar atau tidaknya seseorang oleh COVID-19 [4]. Pengujian RT-PCR dilakukan dengan mengambil sampel *swab* dari rongga *nasofaring* dan cairan reagen. Dari sampel tersebut, nantinya akan diketahui ada atau tidaknya potongan kode genetik yang dimiliki oleh virus SARS-CoV-2. Teknik pengujian ini memiliki akurasi, sensitivitas, dan spesifisitas hingga 100% [4].

Sedangkan pengujian dengan metode tes antibodi dilakukan dengan mengambil sampel darah dari pasien untuk diketahui ada atau tidaknya antibodi yang terbentuk pada tubuh pasien terhadap virus SARS-CoV-2. Jika pada tubuh pasien terdeteksi adanya antibodi yang terbentuk maka dapat diindikasikan bahwa pasien tersebut sudah terpapar COVID-19. Biasanya yang menjadi indikator adalah IgM dan IgG. IgM menunjukkan fase awal antibodi terbentuk sedangkan IgG menunjukkan fase akhir setelah antibodi terbentuk [5].

Pengujian antibodi memang tidak memiliki performa yang lebih baik daripada RT-PCR. Berdasarkan pengujian yang dilakukan oleh [6] pengujian antibodi untuk mendeteksi COVID-19 memiliki akurasi sebesar 88,3%, sensitivitas sebesar 85,6%, dan spesifisitas sebesar 91%.

Walaupun demikian, keduanya masih memiliki beberapa kekurangan, di antaranya adalah pada masalah waktu yang digunakan untuk pengujian dan prosedur pengujian yang dilakukan. Masih berdasarkan sumber [4], pengujian menggunakan Tes Antibodi membutuhkan waktu rata-rata 30 menit sedangkan RT-PCR membutuhkan setidaknya 6 jam untuk satu kali uji. Sedangkan untuk pengambilan sampel, keduanya masih mengharuskan adanya kontak fisik terhadap *suspect* pasien COVID-19. Sehingga metode pengujian alternatif perlu dicari.

Salah satu metode pengujian alternatif yang mungkin dapat digunakan adalah dengan memanfaatkan kecerdasan buatan untuk mengklasifikasikan apakah seseorang terpapar oleh COVID-19 atau tidak

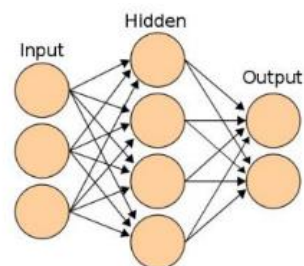
berdasarkan fitur-fitur tertentu. Salah satu fitur yang mungkin dapat digunakan adalah citra rontgen dari paru-paru. Citra rontgen paru-paru biasanya digunakan oleh dokter spesialis gangguan pernafasan untuk membantu menilai apakah seseorang sedang mengidap permasalahan pernafasan atau tidak [7]. Dalam bidang kecerdasan buatan, salah satu algoritma terbaik yang dapat digunakan untuk melakukan klasifikasi citra adalah *Convolutional Neural Network* (CNN) [8].

Pada dasarnya, CNN merupakan pengembangan dari *Artificial Neural Network* (ANN). Teknik-teknik berbasis *neural network* sangat sering digunakan untuk memecahkan masalah-masalah dengan pendekatan kecerdasan buatan karena performanya yang sangat memuaskan. Beberapa contoh penelitian terbaru yang memanfaatkan teknik-teknik tersebut diulas pada artikel [9] dan [10].

Pada penelitian [9] *neural network* digunakan untuk mendeteksi *Tuberculosis* pada citra rontgen paru-paru. Penelitian tersebut menunjukkan bahwa walaupun *neural network* bekerja dengan *imbalanced class*, *neural network* masih dapat menunjukkan akurasi yang cukup baik, yaitu 85,68%.

Sedangkan pada penelitian [10], *convolutional neural networks* dengan arsitektur ResNet-50 digunakan untuk mengdiagnosa COVID-19 berdasarkan citra rontgen paru-paru. Arsitektur ResNet-50 sendiri merupakan salah satu arsitektur CNN yang cukup kompleks. Dari eksperimen yang dilakukan, model dapat memberikan performa yang sangat baik, yaitu 98,18 untuk akurasi dan 98,19 untuk nilai F1.

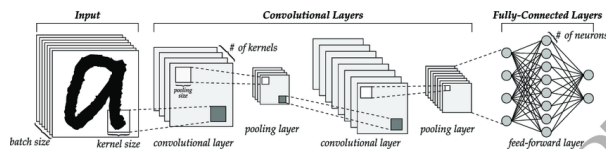
Salah satu kelebihan CNN dibandingkan ANN adalah proses ekstraksi fitur pada CNN tidak perlu menggunakan bantuan manusia, sedangkan ANN sebaliknya. ANN masih membutuhkan bantuan manusia untuk melakukan ekstraksi fitur sebelum fitur tersebut dapat digunakan sebagai input pada ANN [8].



Gambar 1. Arsitektur ANN [11]

Gambar 1 menunjukkan arsitektur dari ANN. ANN umumnya terdiri dari 3 *layer* (lapisan), yaitu *input*, *hidden*, dan *output layer*. Di antara *hidden layer* dan *output layer* juga biasanya ditambahkan fungsi aktivasi. Fungsi aktivasi ini digunakan agar ANN bersifat non-linear sehingga dapat melakukan klasifikasi lebih akurat.

Tidak semua data dapat digunakan sebagai input ANN. Hanya data bersifat numeris saja yang dapat digunakan sebagai input [12]. Hal ini dikarenakan ANN pada dasarnya hanyalah sekumpulan fungsi agregasi yang terdiri dari perkalian dan penjumlahan antara nilai bobot dan input. Untuk dapat melakukan klasifikasi citra, fitur dari citra harus diekstrak terlebih dahulu sebelum dapat digunakan sebagai input ANN.



Gambar 2. Arsitektur CNN [11]

Gambar 2 menunjukkan contoh arsitektur dari CNN. Pada CNN, sebenarnya citra dapat langsung digunakan sebagai *input*. Hal ini dikarenakan CNN memiliki lapisan *convolutional* yang secara otomatis melakukan ekstraksi fitur dari citra yang menjadi *input*. *Output* dari lapisan *convolutional* ini sudah berupa vektor angka. Vektor angka tersebut lah yang akan menjadi *input* dari *fully-connected layer* yang berada pada ujung satunya. Terkadang, *fully-connected layer* yang digunakan sebagai *classifier* dibangun dari setidaknya 2 *layer*. Namun perlu diperhatikan, semakin banyak *layer* maka hasil klasifikasi semakin akurat tetapi tentu beban komputasi juga meningkat.

Untuk data yang berkaitan dengan citra, umumnya CNN digunakan sebagai alternatif algoritma utama. Seperti yang dilakukan pada penelitian [13]. Pada penelitian tersebut CNN digunakan untuk mengidentifikasi ada tidaknya aktivitas manusia pada video *realtime* yang ditangkap oleh CCTV. Pada eksperimen yang dilakukan pada penelitian tersebut, model memiliki akurasi sebesar 81,03%.

Dari sini dapat dinilai bahwa CNN lebih mudah dalam penerapannya jika dibandingkan dengan ANN. Perlu dicatat bahwa yang dimaksud lebih mudah di sini adalah mengenai efisiensi dalam tahapan pra-pengolahan data yang harus dilakukan, khususnya untuk tahapan ekstraksi fitur. Selain itu, CNN memiliki performa yang lebih baik jika dibandingkan dengan ANN. Hal ini dikarenakan CNN dapat memiliki lebih banyak *hidden layer* daripada ANN [14]. Dengan demikian model klasifikasi yang dibangun oleh CNN dapat mendekati model klasifikasi yang dilakukan oleh manusia.

Berdasarkan latar belakang di atas, penelitian ini bertujuan untuk mengukur performa CNN dalam melakukan klasifikasi *suspect* pasien COVID-19 berdasarkan citra rontgen paru-paru dari pasien. Dalam skenarionya, arsitektur CNN yang digunakan adalah arsitektur minimalis. Hal ini dilakukan untuk mengetahui performa dasar dari CNN ketika digunakan untuk melakukan klasifikasi citra. Adapun performa

yang diukur pada penelitian ini adalah akurasi, sensitivitas, dan spesifisitas. Alat-alat ukur performa tersebut merupakan alat ukur yang biasa digunakan pada alat pengujian di bidang kesehatan.

Penelitian ini diharapkan dapat memberikan gambaran objektif terhadap performa CNN ketika digunakan untuk mengklasifikasikan *suspect* pasien COVID-19 berdasarkan citra rontgen paru-paru. Dengan demikian, secara tidak langsung penelitian ini akan berkontribusi dalam memberikan sudut pandang baru dalam hal alternatif pengujian COVID-19 selain menggunakan tes antibodi dan RT-PCR.

Dalam pembahasannya, paper ini dibagi menjadi beberapa bagian. Bagian 1 menjelaskan hal-hal yang melatarbelakangi dilakukannya penelitian ini serta tinjauan literatur singkat terkait penelitian. Bagian 2 menjelaskan metode penelitian yang dilakukan pada penelitian ini. Adapun pembahasan hasil penelitian yang telah dilakukan akan dibahas pada Bagian 3. Sementara kesimpulan dan rencana penelitian selanjutnya akan dijabarkan pada Bagian 4.

## 2. Metode Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini ditunjukkan oleh Gambar 3.



Gambar 3. Metode Penelitian

### 2.1. Studi Literatur

Tahapan penelitian dimulai dari Studi Literatur. Pada tahapan ini, beberapa literatur yang berkaitan dengan masalah yang diangkat serta alternatif-alternatif solusi yang mungkin dapat diambil dikaji dan dipelajari. Pada tahapan ini, juga dilakukan perencanaan penelitian yang meliputi rencana kegiatan, waktu pengerjaan, bagaimana mengerjakannya, skenario-skenario

eksperimen yang akan dilakukan, serta rencana-rencana alternatif jika apa yang direncanakan meleset.

Pada tahapan ini, kami merencanakan eksperimen dilakukan dengan menggunakan *dataset* yang bersifat publik. Artinya *dataset* tersebut dapat diakses secara bebas oleh siapapun. Selain itu, karena salah satu tujuan spesifik penelitian ini adalah untuk menguji model CNN dalam melakukan klasifikasi penyakit yang mirip dengan COVID-19 juga, maka *dataset* yang akan digunakan pun sebaiknya memiliki kelas selain kelas normal dan kelas COVID-19.

Berdasarkan literatur yang didapatkan, kami juga memutuskan untuk menggunakan platform Google Colab sebagai lingkungan eksperimen yang kami gunakan. Hal ini dikarenakan Google Colab menyediakan hampir semua sumber daya komputasi yang dibutuhkan secara gratis. Karena platform yang digunakan adalah Google Colab maka bahasa pemrograman yang digunakan adalah Python 3. Selain itu, kami juga menggunakan layanan Google Drive sebagai media penyimpanan *dataset* yang diolah oleh Google Colab. Dengan demikian, seluruh eksperimen kami berjalan pada *cloud*.

## 2.2. Pengumpulan Dataset

Pada tahapan ini, kami mencari dan mengumpulkan *dataset* sesuai dengan kriteria yang telah direncanakan. *Dataset* yang digunakan pada penelitian ini bersifat terbuka dan dapat diakses oleh siapapun pada situs [www.kaggle.com](http://www.kaggle.com) [15]. *Dataset* ini juga digunakan pada penelitian [16].

*Dataset* yang digunakan pada penelitian ini berupa citra rontgen paru-paru manusia yang dihimpun oleh *Italian Society of Medical Interventional Radiology (SIRM)* sejak akhir 2019 hingga awal 2020. *Dataset* terdiri dari 2905 citra, yang pembagiannya secara singkat ditunjukkan pada Tabel 1.

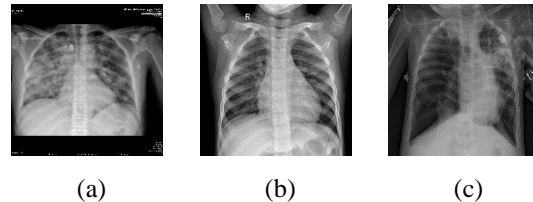
Tabel 1. Pembagian Jumlah Citra

Kelas	Jumlah
COVID	219
NORMAL	1341
VIRAL_PNEUMONIA	1345

Berdasarkan data yang ditampilkan pada Tabel 1, terdapat 219 citra berasal dari orang-orang yang dinyatakan positif COVID-19 (kelas COVID), 1341 citra berasal dari individu-individu yang tidak terpapar penyakit pernafasan apapun (kelas NORMAL), dan sisanya sebanyak 1345 citra berasal dari orang-orang yang dinyatakan mengidap *Viral Pneumonia* namun negatif COVID-19 (kelas VIRAL\_PNEUMONIA).

Citra yang digunakan sebagai *dataset* memiliki format *Portable Network Graphics (PNG)* dan beresolusi 1024 x 1024 pixel. Gambar 4 menunjukkan beberapa sampel citra rontgen yang digunakan pada penelitian ini.

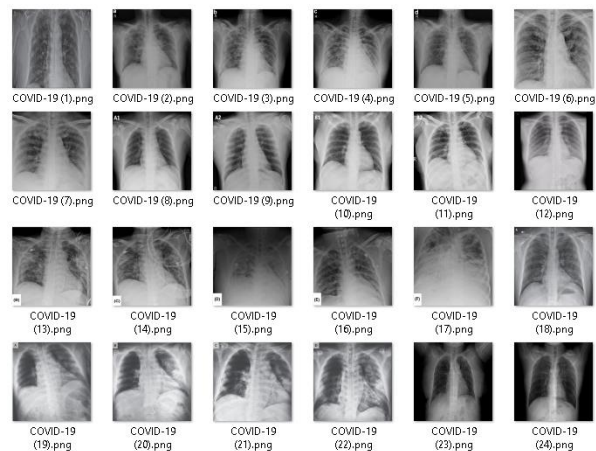
Gambar 4 (a) berasal dari seseorang yang dinyatakan positif COVID-19. Gambar 4 (b) adalah citra rontgen dari orang normal. Gambar 4 (c) berasal dari seseorang yang menderita *Viral Pneumonia*. Bagi orang awam, secara kasatmata ketiganya sulit untuk dibedakan, namun bagi para ahli mungkin ketiganya dapat dibedakan dengan mudah.



Gambar 4. Sampel Citra

## 2.3. Pra-pengolahan Dataset

Tahapan pra-pengolahan *dataset* digunakan untuk menyiapkan *dataset* agar dapat digunakan sesuai dengan skenario-skenario eksperimen yang dilakukan. Pra-pengolahan meliputi pengelompokan citra dalam masing-masing *folder* yang disediakan. Citra dari orang-orang yang dinyatakan positif COVID-19 akan ditempatkan dalam *folder* COVID dan diberi nama sesuai urutannya. Sedangkan citra dari orang-orang normal disimpan pada *folder* NORMAL, dan citra dari orang-orang yang mengidap *viral pneumonia* disimpan pada *folder* VIRAL\_PNEUMONIA.



Gambar 5. Contoh Penamaan Citra

Gambar 5 menunjukkan contoh penamaan citra yang ditempatkan dalam *folder* sesuai dengan kelasnya.

Setelah setiap citra dimasukkan ke dalam masing-masing *folder* maka langkah berikutnya adalah mengunggah *dataset* ke Google Drive sesuai dengan skenario yang akan digunakan. Terdapat 4 skenario eksperimen yang dilakukan dalam penelitian ini.

Tabel 2 menunjukkan jumlah citra untuk masing-masing skenario. Skenario-skenario tersebut sengaja dirancang untuk mengetahui performa model CNN pada dua kelas



dan tiga kelas serta pada komposisi data yang seimbang (*balanced class*) maupun yang tidak seimbang (*imbalanced class*).

Tabel 2. Jumlah Citra pada Masing-Masing Skenario

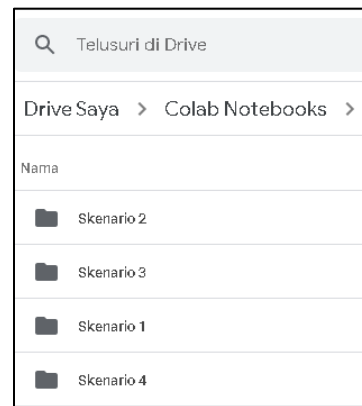
Skenario	Jumlah Citra	Keterangan
Skenario 1	1560	219 citra COVID dan 1341 citra NORMAL
Skenario 2	2906	219 citra COVID, 1341 citra NORMAL, dan 1345 citra VIRAL_PNEUMONIA
Skenario 3	438	219 citra COVID dan 219 citra NORMAL
Skenario 4	657	219 citra COVID, 219 citra NORMAL, dan 219 citra VIRAL_PNEUMONIA

Gambar 6 menunjukkan susunan *folder* pada Google Drive untuk masing-masing skenario. Di mana komposisinya seperti yang dijelaskan oleh Tabel 2.

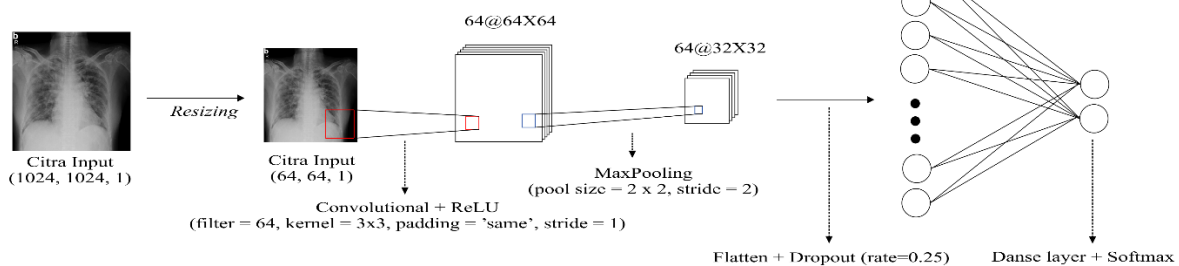
## 2.4 Membangun Model CNN

Setelah semua data siap dan sudah ditempatkan pada masing-masing *folder* skenario, langkah berikutnya

adalah membangun model CNN. Dalam eksperimen yang dilakukan, model CNN dibangun pada *platform* Google Colab. Adapun pemrosesan yang dilakukan tidak menggunakan GPU (*Graphical Processing Unit*) maupun TPU (*Tensor Processing Unit*) melainkan menggunakan CPU (*Central Processing Unit*)



Gambar 6. Folder Skenario Penelitian



Gambar 7. Arsitektur Model CNN yang Digunakan pada Penelitian

Gambar 7 menunjukkan arsitektur dari model CNN yang digunakan di dalam penelitian ini. Model diawali dengan *input* berupa citra berukuran 1024 x 1024 pixel dengan satu kanal warna saja, yaitu *greyscale*. Berikutnya, untuk meminimalisir beban komputasi pada tahapan-tahapan berikutnya, dilakukan *resizing* terhadap citra *input* sehingga citra *input* menjadi berukuran 64 x 64 pixel.

Berikutnya, terdapat *Convolutional Layer* yang terdiri dari 64 filter, dengan kernel berukuran 3 x 3, dan nilai *stride* adalah 1. Untuk pengaturan *padding*, penelitian ini menggunakan pengaturan *padding* yang memungkinkan agar ukuran citra *output* dari *convolutional layer* sama dengan ukuran citra *input convolutional layer*. Untuk mengoptimalkan model, *convolutional layer* juga ditambahkan dengan fungsi aktivasi. Fungsi aktivasi yang digunakan adalah ReLU (*Rectified Linear Unit*). Fungsi ini digunakan karena cukup sederhana namun memberikan hasil yang baik pada jaringan saraf tiruan dengan bentuk konvolusional [17].

Lapisan berikutnya adalah *maxpooling Layer*. Lapisan ini memiliki ukuran *pool* sebesar 2 x 2 dengan *stride* sebesar 2. Dengan demikian, *output* dari lapisan ini separuh dari *inputnya*, yaitu 32 x 32. Berikutnya hasil dari *maxpooling layer* ini akan memasuki bagian *fully connected layer* melalui fungsi *flatten*. Untuk menghindari risiko *over fitting* maka dilakukan juga *dropout* dengan rasio 25%. Perlu diketahui, *fully connected layer* yang digunakan hanya terdiri dari 2 lapisan, yaitu *input layer* dan *output layer*.

*Input layer* dari *fully connected layer* adalah hasil fungsi *flatten* terhadap *maxpooling layer*, sehingga lapisan ini memiliki 65536 node. Sedangkan *output layer* adalah lapisan yang akan menentukan hasil klasifikasi model. Jumlah *node* pada lapisan ini tergantung skenario yang digunakan. Untuk skenario yang melibatkan 2 kelas (skenario 1 dan skenario 3), maka jumlah *node* dari *output layer* adalah 2 *node*. Sedangkan untuk skenario yang melibatkan 3 kelas (skenario 2 dan skenario 4) maka jumlah *node output layer*-nya adalah 3 *node*.

Untuk menghasilkan hasil klasifikasi yang bersifat *non-linear decision* maka ditambahkan lapisan fungsi aktivasi di setiap *node output layer*. Adapun fungsi aktivasi yang digunakan pada *output layer* adalah *Softmax*. Lebih lanjut, model yang dibangun memiliki *learning rate* sebesar 0.0001. Alasan penggunaan *learning rate* sebesar 0.0001 adalah karena berdasarkan beberapa percobaan yang telah dilakukan sebelumnya menunjukkan bahwa *learning rate* sebesar 0.0001 dapat memberikan performa klasifikasi yang lebih baik. Namun hal tersebut dengan asumsi bahwa mengabaikan waktu yang dibutuhkan oleh model untuk belajar.

2.5 Melatih dan Menguji Model CNN

Setelah model CNN dibangun maka langkah selanjutnya adalah melatih model. Pada tahapan ini, diberlakukan skenario seperti yang telah dijelaskan sebelumnya. Untuk setiap skenario, *dataset* dibagi menjadi dua bagian. 70% digunakan untuk melatih model CNN (*training set*) sementara sisanya untuk menguji model CNN (*testing set*). Misalnya saja pada Skenario 1, dari 1560 citra, 1092 citra digunakan sebagai *training set* dan 468 citra digunakan sebagai *testing set*.

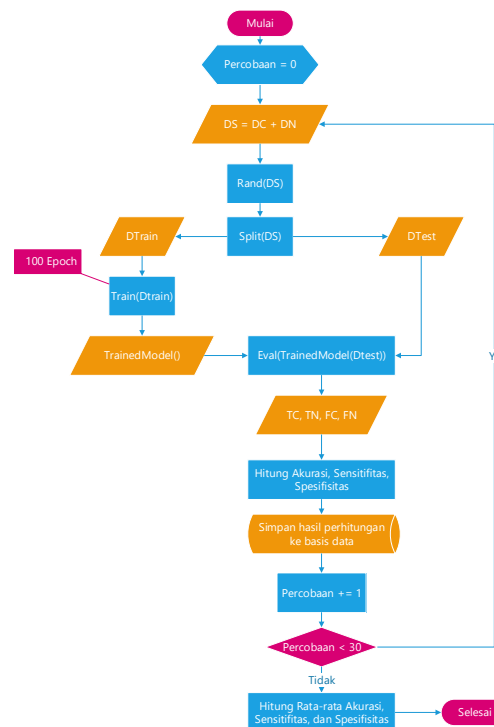
Untuk setiap skenario, dilakukan sebanyak 30 kali percobaan, dan setiap percobaan terdiri dari 100 *epoch*. Performa model dalam melakukan klasifikasi diukur dari performa rata-rata 30 kali percobaan tersebut. Adapun performa yang diukur adalah akurasi, sensitivitas, dan spesifisitas.

Pengujian dilakukan sebanyak sebanyak 30 kali karena sifat CNN yang *non-deterministic* sehingga memberikan hasil performa yang berbeda dari satu percobaan ke percobaan lainnya. Pada setiap pengujian yang dilakukan juga dilakukan pengacakan (*randomization*) terhadap *data set* sebelum dibagi menjadi *training set* dan *testing set*. Jumlah 30 ditentukan karena berdasarkan eksperimen yang dilakukan, perlu setidaknya 30 kali pengujian agar nilai rata-rata performa model tidak mengalami fluktuasi nilai yang signifikan.

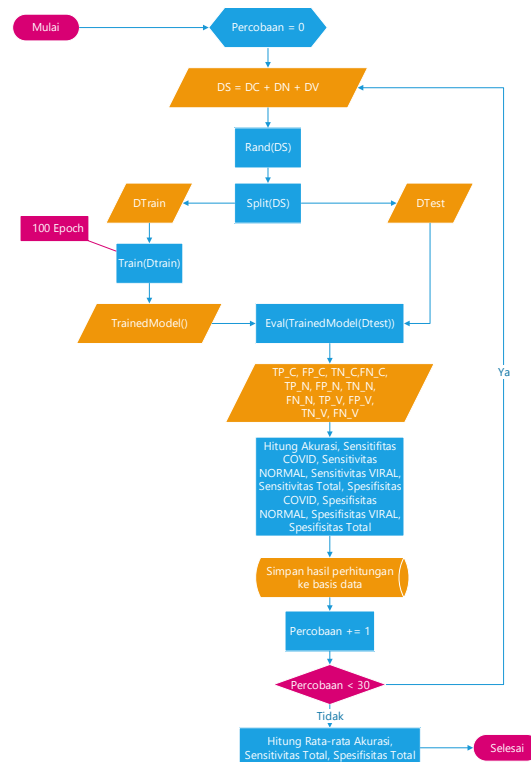
Ilustrasi skema percobaan yang dilakukan ditunjukkan oleh Gambar 8 dan Gambar 9.

Gambar 8 menunjukkan alur eksperimen yang dilakukan dengan 2 kelas (Skenario 1 dan Skenario 3). Eksperimen dimulai dengan mengumpulkan *dataset (DS)*. *DS* terdiri dari *dataset COVID (DC)* dan *dataset NORMAL (DN)*. Berikutnya *DS* diacak dengan memasukkannya ke dalam fungsi *Rand()*. Setelah *DS* berhasil diacak maka *DS* dibagi menjadi dua bagian dengan rasio 70% sebagai *Data Training (DTrain)* dan 30% sebagai *Data Testing (DTest)*. *DTrain* berikutnya digunakan untuk melatih model CNN yang telah dibangun. Model dilatih sebanyak 100 *epoch*. Dari hasil pelatihan tersebut dihasilkan model yang sudah terlatih (*TrainedModel()*).

Berikutnya *TrainedModel()* diuji performanya dengan menggunakan *DTest*. Hasil pengujian ini akan menghasilkan 4 nilai, yaitu *TC, TN, FC, dan FN*.



Gambar 8. Alur Eksperimen Skenario Dua Kelas



Gambar 9. Alur Eksperimen Skenario Tiga Kelas

$TC$  (*True COVID*) adalah jumlah kasus COVID yang diidentifikasi sebagai kelas COVID oleh  $TrainedModel()$ .  $TN$  (*True NORMAL*) adalah jumlah kasus NORMAL yang diidentifikasi sebagai kelas NORMAL oleh  $TrainedModel()$ .  $FC$  (*False COVID*) adalah jumlah kasus yang diidentifikasi sebagai kelas COVID oleh model padahal kasus tersebut bukanlah kasus COVID.  $FN$  (*False NORMAL*) adalah jumlah kasus yang diidentifikasi sebagai kelas NORMAL oleh model padahal kasus tersebut bukanlah kasus NORMAL.

Setelah nilai-nilai tersebut didapatkan, maka langkah selanjutnya adalah menghitung akurasi, sensitivitas, dan spesifisitas. Akurasi adalah rasio keberhasilan model dalam menebak benar semua kelas dari setiap kasus yang diberikan. Adapun formula yang digunakan untuk menghitung akurasi ditunjukkan oleh rumus 1.

$$Akurasi = \frac{TC + TN}{TC + TN + FC + FN} \quad (1)$$

Sedangkan sensitivitas adalah rasio keberhasilan model dalam menebak kasus COVID dari semua kasus COVID yang diujikan padanya. Formula yang digunakan untuk menghitung sensitivitas ditunjukkan oleh rumus 2.

$$Sensitifitas = \frac{TC}{TC + FN} \quad (2)$$

Sementara spesifisitas adalah rasio keberhasilan model dalam menebak kasus NORMAL (negatif COVID) dari semua kasus NORMAL yang diujikan padanya. Adapun persamaan yang digunakan untuk menghitung spesifisitas ditunjukkan oleh persamaan 3.

$$Spesifisitas = \frac{TN}{TN + FC} \quad (3)$$

Nilai akurasi, sensitivitas, dan spesifisitas yang diperoleh selanjutnya disimpan pada basis data (*spreadsheet*). Berikutnya proses diulang lagi dari tahapan pengumpulan  $DS$ . Perulangan proses dilakukan sebanyak 30 kali. Jumlah tersebut ditentukan dengan asumsi bahwa model dapat menunjukkan performa sebenarnya setelah setidaknya 30 kali percobaan.

Setelah ketiga puluh percobaan berhasil dilakukan maka langkah selanjutnya adalah menghitung rata-rata akurasi, rata-rata sensitivitas, dan rata-rata spesifisitas. Rata-rata Akurasi dihitung dengan menggunakan persamaan 4, dengan  $Avg\_acc$  adalah nilai akurasi rata-rata dan  $Akurasi_i$  adalah nilai akurasi pada percobaan ke- $i$ .

$$Avg\_acc = \frac{\sum_{i=1}^{30} Akurasi_i}{30} \quad (4)$$

Sedangkan rata-rata sensitivitas dihitung dengan persamaan 5, dengan  $Avg\_sens$  adalah nilai rata-rata

sensitivitas dan  $Sensitifitas_i$  adalah nilai sensitivitas pada percobaan ke- $i$ .

$$Avg\_sens = \frac{\sum_{i=1}^{30} Sensitifitas_i}{30} \quad (5)$$

Sementara rata-rata spesifisitas dihitung dengan persamaan 6, dengan  $Avg\_spes$  adalah nilai rata-rata spesifisitas dan  $Spesifisitas_i$  adalah nilai spesifisitas pada percobaan ke- $i$ .

$$Avg\_spes = \frac{\sum_{i=1}^{30} Spesifisitas_i}{30} \quad (6)$$

Setelah nilai  $Avg\_acc$ ,  $Avg\_sens$ , dan  $Avg\_spes$  diperoleh maka eksperimen telah selesai dilakukan dan dapat ditarik kesimpulan terhadap performa model klasifikasi yang dilakukan.

Gambar 9 menunjukkan alur eksperimen yang menggunakan 3 kelas (Skenario 2 dan Skenario 4). Secara umum eksperimen dengan 2 kelas dan 3 kelas tidak jauh berbeda. Perbedaannya hanya terletak pada komposisi  $DS$ , hasil dari pengujian  $TrainedModel()$ , dan alat ukur performa yang digunakan.

Pada Skenario 2 dan 4,  $DS$  merupakan gabungan antara  $DC$  (*dataset* kelas COVID),  $DN$  (*dataset* kelas NORMAL), dan  $DV$  (*dataset* kelas VIRAL\_PNEUMONIA). Dikarenakan  $DC$  menggunakan tiga kelas maka hasil pengujian  $TrainedModel()$  tidak saja 4 nilai tetapi 12 nilai, yaitu  $TP\_C$ ,  $FP\_C$ ,  $TN\_C$ ,  $FN\_C$ ,  $TP\_N$ ,  $FP\_N$ ,  $TN\_N$ ,  $FN\_N$ ,  $TP\_V$ ,  $FP\_V$ ,  $TN\_V$ , dan  $FN\_V$ .

$TP\_C$  (*True Positive COVID*) adalah jumlah kasus COVID yang diidentifikasi sebagai kelas COVID oleh  $TrainedModel()$ .  $FP\_C$  (*False Positive COVID*) adalah jumlah kasus yang bukan COVID tetapi diidentifikasi sebagai kelas COVID oleh  $TrainedModel()$ .  $TN\_C$  (*True Negative COVID*) adalah jumlah kasus yang bukan COVID dan memang tidak diidentifikasi sebagai kelas COVID oleh  $TrainedModel()$ .  $FN\_C$  (*False Negative COVID*) adalah jumlah kasus COVID yang diidentifikasi sebagai bukan COVID oleh  $TrainedModel()$ .

$TP\_N$  (*True Positive NORMAL*) adalah jumlah kasus NORMAL yang diidentifikasi sebagai kelas NORMAL oleh  $TrainedModel()$ .  $FP\_N$  (*False Positive NORMAL*) adalah jumlah kasus bukan NORMAL tetapi diidentifikasi sebagai kelas NORMAL oleh  $TrainedModel()$ .  $TN\_N$  (*True Negative NORMAL*) adalah jumlah kasus bukan kelas NORMAL yang diidentifikasi bukan kelas NORMAL oleh  $TrainedModel()$ .  $FN\_N$  (*False Negative NORMAL*) adalah jumlah kasus NORMAL yang diidentifikasi sebagai bukan kelas NORMAL oleh  $TrainedModel()$ .

$TP\_V$  (*True Positive VIRAL*) adalah jumlah kasus VIRAL\_PNEUMONIA yang diidentifikasi sebagai

kelas VIRAL\_PNEUMONIA oleh *TrainedModel()*.  $FP_V$  (*False Positive VIRAL*) adalah jumlah kasus bukan VIRAL\_PNEUMONIA tetapi diidentifikasi sebagai kelas VIRAL\_PNEUMONIA oleh *TrainedModel()*.  $TN_V$  (*True Negative VIRAL*) adalah jumlah kasus bukan kelas VIRAL\_PNEUMONIA yang diidentifikasi sebagai bukan VIRAL\_PNEUMONIA oleh *TrainedModel()*.  $FN_V$  (*False Negative VIRAL*) adalah jumlah kasus VIRAL\_PNEUMONIA yang diidentifikasi sebagai bukan kelas VIRAL\_PNEUMONIA oleh *TrainedModel()*.

Tidak seperti skenario yang menggunakan 2 kelas, pada skenario 3 kelas, alat ukur performa yang digunakan lebih kompleks. Alat ukur yang digunakan di antaranya adalah akurasi, sensitivitas COVID, sensitivitas NORMAL, sensitivitas VIRAL, sensitivitas Total, spesifisitas COVID, spesifisitas NORMAL, spesifisitas VIRAL, dan spesifisitas Total.

Akurasi dengan skenario tiga kelas pada dasarnya adalah menghitung rasio kebenaran dalam menebak setiap kelas terhadap semua tebakan yang dilakukan. Adapun formula yang digunakan untuk menghitung akurasi dengan skenario tiga kelas ditunjukkan oleh persamaan 7, dengan  $Acc_3$  adalah nilai akurasi untuk tiga kelas,  $TP_C$  adalah *True Positive* COVID,  $TP_N$  adalah *True Positive* NORMAL,  $TP_V$  adalah *True Positive* VIRAL, dan  $n_{DTest}$  adalah jumlah data yang ada pada *DTest*.

$$Acc_3 = \frac{TP_C + TP_N + TP_V}{n_{DTest}} \quad (7)$$

Sensitivitas COVID mengukur rasio kebenaran model dalam menebak suatu kasus sebagai COVID dari setiap kasus COVID yang ada pada *DTest*. Persamaan yang digunakan untuk mengukur sensitivitas COVID ditunjukkan pada formula 8, dengan  $Sens_{cov}$  adalah nilai sensitivitas COVID,  $TP_C$  adalah *True Positive* COVID, dan  $FN_C$  adalah *False Negative* COVID.

$$Sens_{cov} = \frac{TP_C}{TP_C + FN_C} \quad (8)$$

Sensitivitas NORMAL mengukur rasio kebenaran model dalam menebak suatu kasus sebagai kelas NORMAL dari setiap kasus NORMAL yang ada pada *DTest*. Persamaan yang digunakan untuk mengukur sensitivitas NORMAL ditunjukkan oleh persamaan 9, dengan  $Sens_{norm}$  adalah nilai sensitivitas NORMAL,  $TP_N$  adalah *True Positive* NORMAL dan  $FN_N$  adalah *False Negative* NORMAL.

$$Sens_{norm} = \frac{TP_N}{TP_N + FN_N} \quad (9)$$

Sensitivitas VIRAL mengukur rasio kebenaran model untuk mengklasifikasikan kasus sebagai kelas VIRAL\_PNEUMONIA dari setiap kasus VIRAL\_PNEUMONIA yang terdapat pada *DTest*.

Formula yang digunakan untuk menghitung sensitivitas VIRAL ditunjukkan oleh persamaan 10, dengan  $Sens_{vir}$  adalah nilai sensitivitas VIRAL,  $TP_V$  adalah nilai dari *True Positive* VIRAL, dan  $FN_V$  adalah nilai *False Negative* VIRAL.

$$Sens_{vir} = \frac{TP_V}{TP_V + FN_V} \quad (10)$$

Sensitivitas Total adalah nilai rata-rata dari  $Sens_{cov}$ ,  $Sens_{norm}$ , dan  $Sens_{vir}$ , sehingga nilai sensitivitas Total dapat dihitung dengan persamaan 11, dengan  $Sens_{tot}$  adalah nilai sensitivitas Total,  $Sens_{cov}$  adalah nilai Sensitivitas COVID,  $Sens_{norm}$  adalah nilai Sensitivitas NORMAL, dan  $Sens_{vir}$  adalah nilai Sensitivitas VIRAL.

$$Sens_{tot} = \frac{Sens_{cov} + Sens_{norm} + Sens_{vir}}{3} \quad (11)$$

Spesifisitas COVID mengukur rasio kebenaran model dalam menebak suatu kasus sebagai bukan kasus COVID dari setiap kasus bukan COVID yang ada pada *DTest*. Spesifisitas COVID dihitung dengan menggunakan persamaan 12, dengan  $Spe_{cov}$  adalah nilai Spesifisitas COVID,  $TN_C$  adalah nilai *True Negative* COVID, dan  $FP_C$  adalah nilai *False Positive* COVID.

$$Spe_{cov} = \frac{TN_C}{TN_C + FP_C} \quad (12)$$

Spesifisitas NORMAL menghitung rasio kebenaran model dalam mengklasifikasikan suatu kasus sebagai bukan kelas NORMAL dari setiap kasus bukan NORMAL yang ada pada *DTest*. Persamaan yang digunakan untuk menghitung spesifisitas NORMAL ditunjukkan oleh persamaan 13, dengan  $Spe_{norm}$  adalah nilai spesifisitas NORMAL,  $TN_N$  adalah nilai *True Negative* NORMAL, dan  $FP_N$  adalah nilai *False Positive* NORMAL.

$$Spe_{norm} = \frac{TN_N}{TN_N + FP_N} \quad (13)$$

Spesifisitas VIRAL adalah nilai rasio kebenaran model dalam menebak suatu kasus sebagai bukan kelas VIRAL\_PNEUMONIA dari setiap kasus bukan VIRAL\_PNEUMONIA yang terdapat pada *DTest*. Rumus yang digunakan untuk menghitung Spesifisitas VIRAL ditunjukkan oleh persamaan 14, dengan  $Spe_{vir}$  adalah nilai spesifisitas VIRAL,  $TN_V$  adalah nilai *True Negative* VIRAL, dan  $FP_V$  adalah nilai *False Positive* VIRAL.

$$Spe_{vir} = \frac{TN_V}{TN_V + FP_V} \quad (14)$$

Spesifisitas Total adalah nilai rata-rata dari  $Spe_{cov}$ ,  $Spe_{norm}$ , dan  $Spe_{vir}$ . Dengan demikian nilai Spesifisitas Total dapat dihitung dengan persamaan 15, dengan  $Spe_{tot}$  adalah nilai spesifisitas Total,  $Spe_{cov}$  adalah nilai



spesifisitas COVID,  $Spes_{norm}$  adalah nilai spesifisitas NORMAL, dan  $Spes_{vir}$  adalah nilai spesifisitas VIRAL.

$$Spes_{tot} = \frac{Spes_{cov} + Spes_{norm} + Spes_{vir}}{3} \quad (15)$$

Nilai akurasi 3 kelas, sensitivitas Total, dan spesifisitas Total yang diperoleh selanjutnya disimpan pada basis data (*spreadsheet*). Berikutnya proses diulang kembali dari tahapan pengumpulan *DS*. Percobaan dilakukan sebanyak 30 kali.

Setelah ketiga puluh percobaan berhasil dilakukan maka langkah selanjutnya adalah menghitung rata-rata akurasi 3 kelas, rata-rata sensitivitas Total, dan rata-rata spesifisitas Total. Rata-rata akurasi 3 kelas dihitung dengan persamaan 16, dengan  $Avg\_acc\_3$  adalah nilai rata-rata akurasi tiga kelas dan  $Acc\_3_i$  adalah nilai akurasi tiga kelas pada percobaan ke-*i*.

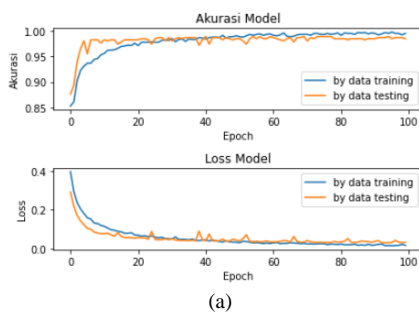
$$Avg\_acc\_3 = \frac{\sum_{i=1}^{30} Acc\_3_i}{30} \quad (16)$$

Sedangkan nilai rata-rata sensitivitas Total dihitung dengan formula 17, dengan  $Avg\_sens_{tot}$  adalah nilai rata-rata sensitivitas Total dan  $Sens_{tot_i}$  adalah nilai sensitivitas Total pada percobaan ke-*i*.

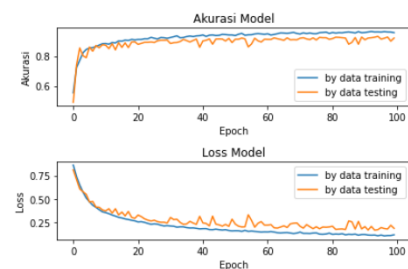
$$Avg\_sens_{tot} = \frac{\sum_{i=1}^{30} Sens_{tot_i}}{30} \quad (17)$$

Sementara nilai rata-rata spesifisitas Total dihitung dengan persamaan 18, dengan  $Avg\_spes_{tot}$  adalah nilai rata-rata spesifisitas Total dan  $Spes_{tot_i}$  adalah nilai spesifisitas Total pada percobaan ke-*i*.

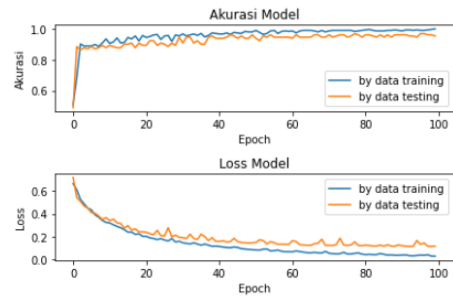
$$Avg\_spes_{tot} = \frac{\sum_{i=1}^{30} Spes_{tot_i}}{30} \quad (18)$$



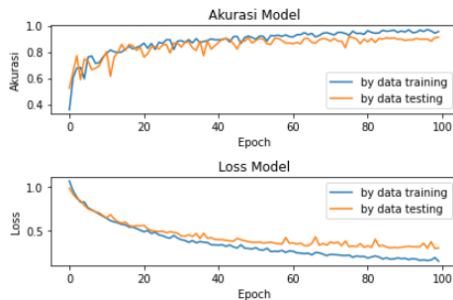
(a)



(b)



(c)



(d)

Gambar 10. Grafik Hasil Pelatihan untuk Setiap Epoch

Gambar 10 (a) – (d) menunjukkan sampel hasil pelatihan untuk setiap *epoch* pada setiap skenarionya. Gambar 10 (a) adalah grafik pelatihan untuk Skenario 1. Gambar 10 (b) menunjukkan hasil pelatihan pada Skenario 2. Gambar 10 (c) menunjukkan grafik pelatihan untuk Skenario 3. Sedangkan Gambar 10 (d) adalah grafik hasil pelatihan untuk Skenario 4.

### 3. Hasil dan Pembahasan

Seperti yang telah dijelaskan sebelumnya, penelitian ini menggunakan 4 skenario, dan masing-masing skenario dilakukan sebanyak 30 kali percobaan. Berikut hasil eksperimen untuk masing-masing skenario.

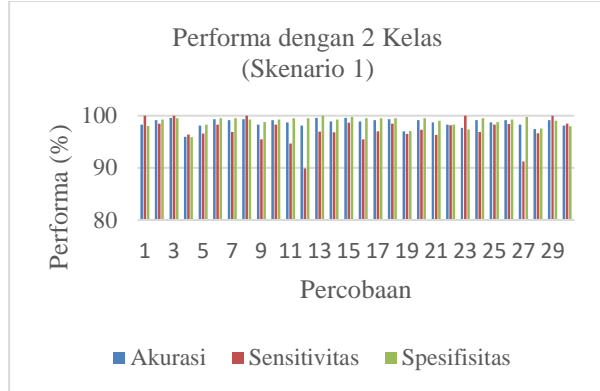
#### 3.1. Skenario 1

Skenario ini menggunakan 1560 buah citra sebagai *DS*, yang terdiri dari 219 citra COVID dan 1341 citra NORMAL. Dari 1560 citra, 1092 citra digunakan sebagai *DTrain* dan 468 citra digunakan sebagai *DTest*.

Tabel 3. Hasil Pengukuran Performa Skenario 1

Percobaan	Akurasi (%)	Sensitivitas (%)	Spesifisitas (%)
1	98,29	100,00	98,04
2	99,15	98,44	99,26
3	99,57	100,00	99,50
...	...	...	...
30	98,08	98,46	98,01
<b>Rata-rata</b>	<b>98,69</b>	<b>97,21</b>	<b>98,90</b>

Tabel 3 menunjukkan hasil pengukuran performa model dalam melakukan klasifikasi dengan Skenario 1. Dari hasil tersebut, terlihat bahwa model memiliki akurasi rata-rata sebesar 98.69%, sensitivitas sebesar 97,21%, dan spesifisitas sebesar 98,90%.



Gambar 11. Performa Akurasi, Sensitivitas, dan Spesifisitas Skenario 1

Gambar 11 menunjukkan perbandingan nilai akurasi, sensitivitas, dan spesifisitas pada percobaan ke-1 hingga percobaan ke-30 untuk Skenario 1. Terlihat dari perbandingan tersebut, fluktuasi paling tajam terjadi pada nilai sensitivitas.

### 3.2. Skenario 2

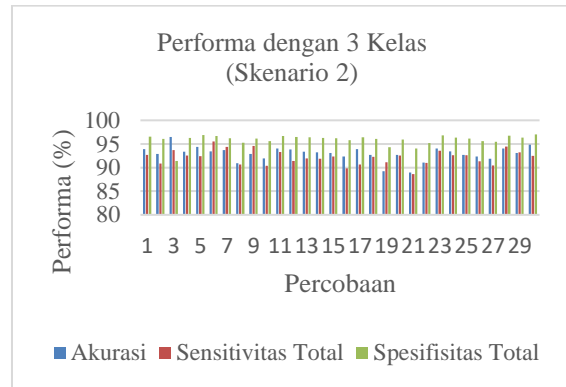
Skenario ini menggunakan 2906 citra sebagai *dataset*-nya. Komposisi *dataset* tersebut terdiri dari 219 citra COVID, 1341 citra NORMAL, dan 1345 citra VIRAL\_PNEUMONIA. *Dataset* dibagi menjadi dua bagian, sebanyak 2034 citra digunakan sebagai *DTrain* dan 872 citra digunakan sebagai *DTest*.

Tabel 4. Hasil Pengukuran Performa Skenario 2

Percobaan	Akurasi (%)	Sensitivitas Total (%)	Spesifisitas Total (%)
1	93,92	92,70	96,53
2	92,89	90,85	96,04
3	96,47	93,69	91,40
...	...	...	...
30	94,84	92,46	97,01
<b>Rata-rata</b>	<b>92,96</b>	<b>92,17</b>	<b>95,91</b>

Tabel 4 menunjukkan hasil performa model dalam melakukan klasifikasi dengan Skenario 2. Hasil menunjukkan bahwa model memiliki akurasi rata-rata sebesar 92,96%, rata-rata sensitivitas total sebesar 92,17%, dan rata-rata spesifisitas total sebesar 95,91%.

Gambar 12 menunjukkan perbandingan nilai akurasi, sensitivitas total, dan spesifisitas total pada percobaan ke-1 hingga percobaan ke-30 untuk Skenario 2. Pada skenario ini, yang mengalami fluktuasi nilai paling tajam justru akurasi dan sensitifitas total.



Gambar 12. Performa Akurasi, SensitivitasTotal, dan SpesifisitasTotal Skenario 2

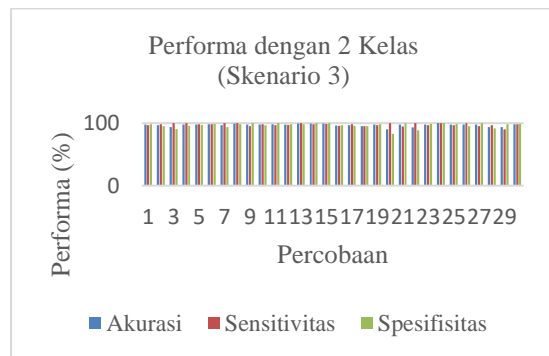
### 3.3. Skenario 3

Skenario 3 menggunakan 438 citra sebagai *DS* yang terdiri dari 219 citra COVID dan 219 citra NORMAL. Dari total 438 citra, 307 citra digunakan sebagai *DTrain* dan 131 citra sebagai *DTest*.

Tabel 5. Hasil Pengukuran Performa Skenario 3

Percobaan	Akurasi (%)	Sensitivitas (%)	Spesifisitas (%)
1	97,71	96,92	98,48
2	96,95	98,48	95,38
3	93,89	100,00	90,36
...	...	...	...
30	98,47	98,36	98,57
<b>Rata-rata</b>	<b>97,17</b>	<b>97,71</b>	<b>96,67</b>

Tabel 5 menunjukkan hasil performa model dalam melakukan klasifikasi pada Skenario 3. Hasil percobaan menunjukkan bahwa model memiliki akurasi rata-rata sebesar 97,17%, sensitivitas rata-rata sebesar 97,71%, dan spesifisitas rata-rata sebesar 96,67%.



Gambar 13. Performa Akurasi, Sensitivitas, dan Spesifisitas Skenario 3

Gambar 13 menunjukkan perbandingan nilai akurasi, sensitivitas, dan spesifisitas untuk percobaan ke-1 hingga percobaan ke-30 pada Skenario 3. Secara umum, nilai akurasi, sensitivitas, dan spesifisitas mengalami kestabilan di atas angka 90%. Hanya saja pada percobaan ke-20 dan ke-22 nilai spesifisitas berada di bawah 90%.

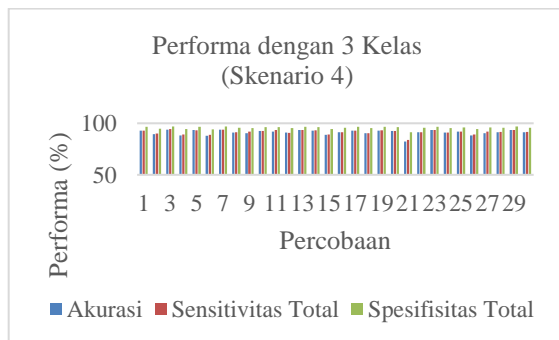
### 3.4. Skenario 4

Skenario ini menggunakan 657 citra sebagai *dataset*-nya. Komposisi *dataset* tersebut terdiri dari 219 citra COVID, 219 citra NORMAL, dan 219 citra VIRAL\_PNEUMONIA. *Dataset* dibagi menjadi dua bagian, sebanyak 460 citra digunakan sebagai *DTrain* dan 197 citra digunakan sebagai *DTest*.

Tabel 6. Hasil Pengukuran Performa Skenario 4

Percobaan	Akurasi (%)	Sensitivitas Total (%)	Spesifisitas Total (%)
1	92,89	93,00	96,46
2	89,34	90,08	94,76
3	93,91	94,28	97,01
...	...	...	...
30	91,37	91,50	95,66
<b>Rata-rata</b>	<b>91,18</b>	<b>91,57</b>	<b>95,67</b>

Tabel 6 menunjukkan hasil performa model dalam melakukan klasifikasi dengan Skenario 4. Hasil menunjukkan bahwa model memiliki akurasi rata-rata sebesar 91,18%, rata-rata sensitivitas total sebesar 91,57%, dan rata-rata spesifisitas total sebesar 95,67%.



Gambar 14. Performa Akurasi, SensitivitasTotal, dan SpesifisitasTotal Skenario 4

Gambar 14 menunjukkan perbandingan nilai akurasi, sensitivitas total, dan spesifisitas total untuk percobaan ke-1 hingga percobaan ke-30 pada Skenario 4. Serupa dengan Skenario 2, pada skenario ini, akurasi dan sensitivitas total mengalami fluktuasi nilai yang cenderung tajam sedangkan Spesifisitas Total cenderung stabil.

### 3.5. Rangkuman Hasil Pengukuran Performa

Tabel 7 menunjukkan rangkuman hasil pengukuran performa *TrainedModel()* dalam melakukan klasifikasi pada keempat skenario yang telah dijelaskan sebelumnya. Dari hasil tersebut dapat dilihat bahwa akurasi, sensitivitas, dan spesifisitas terendah yang dimiliki oleh *TrainedModel()* adalah masing-masing 91,18%, 91,57%, dan 95,67%. Performa terendah tersebut terjadi pada Skenario 4. Skenario 4 sendiri menggunakan tiga kelas, yaitu kelas COVID, NORMAL, dan VIRAL\_PNEUMONIA. Masing-

masing kelas menggunakan jumlah citra yang sama yaitu 219.

Tabel 7. Rangkuman Hasil Pengukuran Performa untuk 4 Skenario

	Skenario 1 (%)	Skenario 2 (%)	Skenario 3 (%)	Skenario 4 (%)
Rata-rata akurasi	98,69	92,96	97,17	91,18
Rata-rata sensitivitas / sensitivitas total	97,21	92,17	97,71	91,57
Rata-rata spesifisitas / spesifisitas total	98,90	95,91	96,67	95,67

Untuk nilai akurasi dan spesifisitas tertinggi dari *TrainedModel()* adalah 98,69% dan 98,90% yang terjadi pada Skenario 1. Sedangkan untuk nilai sensitivitas tertinggi dari *TrainedModel()* adalah 97,71% pada Skenario 3.

Dari Tabel 7 juga dapat diketahui bahwa ketimpangan jumlah anggota kelas (*imbalanced class*) dapat mempengaruhi performa klasifikasi yang dilakukan. Pada Skenario 1 dan Skenario 3 yang notabene sama-sama mengklasifikasikan dua kelas namun Skenario 1 dengan jumlah anggota kelas yang timpang dan Skenario 3 menggunakan jumlah anggota kelas yang seimbang, terjadi perbedaan nilai performa walaupun perbedaan yang terjadi tidak cukup besar. Hal tersebut juga terlihat dari hubungan antara Skenario 2 dan Skenario 4.

Skenario 2 dan Skenario 4 sama-sama terdiri dari tiga kelas. Skenario 2 menggunakan komposisi data yang tidak seimbang (219 citra kelas COVID, 1341 citra kelas citra kelas NORMAL, dan 1345 citra kelas VIRAL\_PNEUMONIA), sedangkan Skenario 4 menggunakan komposisi data yang seimbang (219 citra kelas COVID, 219 citra kelas citra kelas NORMAL, dan 219 citra kelas VIRAL\_PNEUMONIA).

Dari hasil eksperimen tersebut, terdapat temuan yang cukup menarik. Jika diperhatikan, model memberikan performa yang lebih baik pada skenario yang melibatkan *imbalanced class* dibandingkan pada skenario yang melibatkan kelas dengan jumlah data seimbang (*balanced class*). Hal ini tentu kurang sesuai dengan temuan umum yang biasanya terjadi pada kasus klasifikasi. Di mana, umumnya kinerja *classifier* akan lebih baik pada saat berurusan dengan *balanced class*, seperti yang ditunjukkan pada penelitian [18].

Di sini, dicoba untuk memberikan analisis mengapa model memberikan performa yang lebih baik pada saat berurusan dengan *imbalanced class*. Hal tersebut bisa saja dikarenakan rasio antar kelas yang cukup besar. Jika diperhatikan, perbandingan antara kelas minoritas dan kelas mayoritas kira-kira 1:6. Rasio ini berlaku pada *training set* dan *testing set*.

Jika dianalogikan, ini seperti kasus pengambilan bola pada wadah yang tertutup. Misal terdapat dua buah wadah. Wadah pertama diberi nama Wadah *Training* sedangkan wadah kedua diberi nama Wadah *Testing*. Wadah *Training* berisi 14 buah bola yang terdiri dari 2 buah bola merah dan 12 buah bola biru. Sedangkan Wadah *Testing* berisi 7 buah bola yang terdiri dari 1 bola warna merah dan 6 bola warna biru. Jika seseorang diminta untuk mengambil masing-masing satu bola dari masing-masing wadah beberapa kali maka kemungkinan terjadinya seseorang tersebut akan mengambil bola dengan warna yang sama (dari Wadah *Training* dan Wadah *Testing*) akan sangat besar.

Dari Tabel 7 dapat diketahui juga bahwa jumlah kelas yang digunakan dapat mempengaruhi performa klasifikasi yang dilakukan oleh *TrainedModel()*. Skenario 3 dan Skenario 4 sama-sama menggunakan komposisi data yang seimbang tetapi menggunakan jumlah kelas yang berbeda. Skenario 3 menggunakan dua kelas sedangkan Skenario 4 menggunakan tiga kelas. Dengan penambahan satu kelas tersebut ternyata dapat menurunkan performa klasifikasi yang diujikan mulai dari 1,00 % hingga 6,14%. Hal tersebut juga terjadi pada Skenario 1 dan Skenario 2.

Pada Skenario 1 dan Skenario 2, komposisi data yang digunakan sama-sama tidak seimbang. Namun kedua Skenario tersebut menggunakan jumlah kelas yang berbeda. Skenario 1 terdiri dari dua kelas dan Skenario 2 terdiri dari tiga kelas. Pada kasus ini juga terdapat penurunan performa klasifikasi yang diujikan mulai dari 2,99% hingga 5,73% dengan adanya penambahan satu kelas.

#### 4. Kesimpulan

Dari hasil eksperimen yang dilakukan dengan menggunakan beberapa skenario, didapatkan hasil bahwa walaupun arsitektur CNN yang digunakan cukup sederhana, model masih dapat melakukan klasifikasi citra dengan sangat baik. Model mampu mencapai akurasi sebesar 97,17%, sedangkan untuk sensitivitas dan spesifisitas masing-masing adalah 97,71% dan 96,67%. Hasil eksperimen yang dilakukan juga menunjukkan bahwa ketidakseimbangan komposisi data dapat mempengaruhi performa model dalam melakukan klasifikasi. Selain itu jumlah kelas yang harus diklasifikasikan turut memberikan pengaruh terhadap performa model.

Selanjutnya, penelitian ini akan dikembangkan lagi ke arah pembuatan sistem klasifikasi COVID-19 berdasarkan citra rontgen paru-paru dengan menggunakan model klasifikasi yang dibangun pada penelitian ini. Harapannya sistem yang akan dibangun tersebut dapat bermanfaat dan dapat diakses secara bebas oleh siapapun.

#### Ucapan Terimakasih

Ucapan terima kasih dipersembahkan kepada Universitas Amikom Purwokerto yang telah mendanai penelitian ini.

#### Daftar Rujukan

- [1] R. Sebayang, "WHO Nyatakan Wabah COVID-19 jadi Pandemi, Apa Maksudnya?," 2020. <https://www.cnbcindonesia.com/news/20200312075307-4-144247/who-nyatakan-wabah-covid-19-jadi-pandemi-apa-maksudnya> (diakses Mar 25, 2020).
- [2] W. H. Organization, "T3: Test. Treat. Track. Scaling up diagnostic testing, treatment and surveillance for malaria," 2012. [https://www.who.int/malaria/publications/atoz/t3\\_brochure/en/](https://www.who.int/malaria/publications/atoz/t3_brochure/en/) (diakses Mar 22, 2020).
- [3] R. H. Permana, "Strategi Korea Selatan Pukul Mundur Corona: Lacak, Uji, Obati!," 2020. <https://news.detik.com/berita/d-4946883/strategi-korea-selatan-pukul-mundur-corona-lacak-uji-obati> (diakses Mar 25, 2020).
- [4] T. administrator situs KawalCOVID19.id, "Rapid test atau swab test: Apa bedanya? Mana yang lebih baik?," 2020. <https://kawalcovid19.id/content/1183/rapid-test-atau-swab-test-apa-bedanya-mana-yang-lebih-baik> (diakses Apr 15, 2020).
- [5] A. Herráez, "Rapid tests for coronavirus: how do they work?," 2020. <https://network.febs.org/posts/64073-rapid-tests-for-coronavirus-how-do-they-work> (diakses Mar 27, 2020).
- [6] L. Ying *et al.*, "Diagnostic Indexes of a Rapid IgG/IgM Combined Antibody Test for SARS-CoV-2," *ivmedRx*, 2020, doi: <https://doi.org/10.1101/2020.03.26.20044883>.
- [7] W. J. Palmer, "ACR Releases CT and Chest X-ray Guidance Amid COVID-19 Pandemic," 2020. <https://www.diagnosticsimaging.com/view/acr-releases-ct-and-chest-x-ray-guidance-amid-covid-19-pandemic> (diakses Mar 22, 2020).
- [8] W. Rawat dan Z. Wang, "Deep Convolutional Neural Networks for Image Classification : A Comprehensive Review," *Neural Comput.* 29, hal. 2352–2449, 2017, doi: 10.1162/NECO\_a\_00990.
- [9] C. Liu *et al.*, "TX-CNN: Detecting Tuberculosis in Chest X-ray Images using Convolutional Neural Network," in *Proceedings - International Conference on Image Processing, ICIP*, 2017, hal. 2314–2318. doi: 10.1109/ICIP.2017.8296695.
- [10] S. U. K. Bukhari, S. S. K. Bukhari, A. Syed, dan S. S. H. Shah, "The Diagnostic Evaluation of Convolutional Neural Network (CNN) for The Assessment of Chest X-ray of Patients Infected with COVID-19," *MedRxiv*, 2020, doi: <https://doi.org/10.1101/2020.03.26.20044610>.
- [11] I. Gogul dan V. S. Kumar, "Flower Species Recognition System Using Convolution Neural Networks and Transfer Learning," 2017. doi: 10.1109/ICSCN.2017.8085675.
- [12] B. P. Hartato, T. Astuti, I. Pratika, R. Wahyudi, I. Santiko, dan A. D. Riyanto, "Artificial Neural Network Utilization for Analyzing Sentiment Polarity in Electronics Product Reviews," in *3rd International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2018, hal. 209–214. doi: 10.1109/ICITISEE.2018.8720987.
- [13] M. Arfan, Ahmad Nurjalal, Maman Somantri, dan Sudjadi, "Pengenalan Aktivitas Manusia pada Area Tambak Udang dengan Convolutional Neural Network," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, hal. 174–179, 2021, doi: 10.29207/resti.v5i1.2888.
- [14] N. Aloysius dan G. M., "A Review on Deep Convolutional Neural Networks," in *International Conference on Communication and Signal Processing (ICCSP)*, 2017, hal. 588–592. doi: 10.1109/ICCSP.2017.8286426.

- [15] T. Rahman, M. Chowdhury, dan A. Khandakar, "COVID-19 Radiography Database." 2020. [Daring]. Tersedia pada: <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>
- [16] M. E. H. Chowdhury *et al.*, "Can AI Help in Screening Viral and COVID-19 Pneumonia?," *IEEE Access*, vol. 8, hal. 132665–132676, 2020, doi: 10.1109/ACCESS.2020.3010287.
- [17] S. Sharma, "Activation Functions in Neural Networks," 2017. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> (diakses Nov 20, 2020).
- [18] R. R. Rao dan K. Makkithaya, "Learning from a Class Imbalanced Public Health Dataset : a Cost-based Comparison of Classifier Performance," *Int. J. Electr. Eng.*, vol. 7, no. 4, hal. 2215–2222, 2017, doi: 10.11591/ijece.v7i4.pp2215-2222.