

Arsitektur *Resnet-152* dengan Perbandingan *Optimizer Adam* dan *RMSProp* untuk Mendeteksi Penyakit Paru – Paru

SITI ASY SYIFA, IRMA AMELIA DEWI

Program Studi Informatika, Institut Teknologi Nasional Bandung
Email: ipaa2240@mhs.itenas.ac.id

Received 2 Agustus 2022 | *Revised* 7 November 2022 | *Accepted* 6 Desember 2022

ABSTRAK

Penyakit paru-paru pada manusia sering terjadi di seluruh dunia umumnya terjadi karena polusi udara dan asap rokok. Untuk mendeteksi penyakit paru-paru manusia ini diperlukan kemampuan secara tepat dengan menggunakan Chest X-Rays (CXR). CXR umumnya sulit dibedakan oleh manusia, maka dari itu pada penelitian ini menerapkan model Deep Learning sebagai sarana untuk mendeteksi penyakit paru-paru manusia melalui citra CXR. Eksperimen dilakukan dengan menggunakan model arsitektur yaitu ResNet-152 serta 2 optimizer yaitu Adam dan RMSProp. Selain itu, pengujian model dilakukan dengan menggunakan accuracy, precision, recall, f1-score, specificity dan grafik Receiver Operating Characteristic (ROC). Pada penelitian ini menunjukkan bahwa model ResNet-152-R10 yang memiliki tingkat accuracy, precision, recall, f1-score dan specificity terbaik yaitu masing-masing 92%, 94%, 92%, 93% dan 96,75%.

Kata kunci: *adam optimizer, rmsprop optimizer, resnet-152, penyakit paru-paru, pneumonia, lung opacity*

ABSTRACT

Lung disease in humans often occurs throughout the world generally occurs due to air pollution and cigarette smoke. To detect human lung disease, it requires the ability to be precise using Chest X-Rays (CXR). CXR is generally difficult to distinguish by humans, therefore this study applies the Deep Learning model as a means of detecting human lung disease through CXR imagery. Experiments were carried out using the architectural model, namely ResNet-152 and 2 optimizers, namely Adam and RMSProp. In addition, model testing is carried out using accuracy, precision, recall, f1-score, specificity and Receiver Operating Characteristic (ROC) graphs. This study shows that the ResNet-152-R10 model has the best levels of accuracy, precision, recall, f1-score and specificity, namely 92%, 94%, 92%, 93% and 96.75% respectively.

Keywords: *adam optimizer, rmsprop optimizer, resnet-152, lung disease, pneumonia, lung opacity*

1. PENDAHULUAN

Penyakit paru-paru pada manusia sering terjadi di seluruh dunia umumnya terjadi karena polusi udara dan asap rokok. Pada tahun 2016, kira-kira terdapat 3,4 juta orang meninggal pada 2016 karena *Chronic Obstructive Pulmonary Disease (COPD)*. Penyakit pada paru-paru yang paling sering diderita oleh manusia yaitu *pneumonia* dan *lung opacity* (**Bharati et al., 2020**). Lalu, pada Desember 2019 terdapat varian baru dari jenis penyakit paru-paru yang banyak menjangkit manusia yaitu *SARS-CoV-2*, atau biasa disebut sebagai *COVID-19* (**Sensusiati et al., 2021**). Dalam diagnosis penyakit paru-paru pada manusia dapat dilakukan menggunakan citra *Chest X-Rays (CXR)* ataupun *Computed Tomography (CT) Scan* (**Chowdhury et al., 2020**). *CT Scan* memiliki sensitivitas yang tinggi untuk mendeteksi penyakit paru-paru, tetapi memiliki kelemahan yaitu tingkat radiasi yang tinggi bagi manusia, terutama untuk wanita hamil dan anak-anak. Selain itu, faktor biaya yang cukup mahal untuk melakukan *CT Scan* menjadikan *CXR* tampaknya menjadi solusi yang tepat dalam membantu tenaga medis untuk mendeteksi penyakit paru-paru (**Chowdhury et al., 2020**).

Penggunaan metode *deep learning* dapat menjadi solusi untuk deteksi penyakit paru-paru menggunakan *CXR* karena mata manusia sangat sulit untuk mengidentifikasi secara visual karena kemiripan yang sangat tinggi (**Jabra et al., 2021**). *Deep learning* merupakan sub bidang dari *artificial intelligence* yang memiliki proses seperti jaringan saraf manusia dan memiliki keunggulan dalam hal akurasi deteksi daripada *machine learning*. *Deep learning* juga membutuhkan banyaknya jumlah *datasets* untuk bekerja lebih baik (**Kumar & L., 2018**). Salah satu arsitektur yang ada pada *deep learning* merupakan *Convolutional Neural Network (CNN)*. *CNN* banyak digunakan saat ini karena dapat otomatis mengekstraksi fitur yang dibutuhkan tanpa penglihatan manusia (**Alzubaidi et al., 2021**). Struktur *CNN* berupa serangkaian lapisan konvolusi, lapisan *pooling* dan lapisan *full-connected* yang terhubung satu sama lain (**Deng & Yu, 2014**). Dengan keunggulan tersebut, *CNN* menjadi pilihan utama para peneliti untuk mendeteksi suatu citra contohnya yaitu deteksi penyakit *COVID-19*. *CNN* memiliki beragam arsitektur yang diantaranya *ResNet*, *DenseNet*, *VGG* dan *Inception* (**Alzubaidi et al., 2021**).

Pada *deep learning*, penggunaan *optimizer* sangat penting pada tingkat akurasi dari model. *Optimizer* bekerja untuk mengubah *parameter* seperti bobot dan *learning rate* dalam proses pelatihan model. *RMS-prop* merupakan *optimizer* yang bekerja dengan merubah akumulasi *gradient* ke dalam *weighted moving average*. *Adam (Adaptive moment estimation)* merupakan kombinasi dari *RMS-prop* dengan momentum dan *learning rate* adaptif. Cara kerja keduanya hampir sama karena menggunakan *weighted moving average* tetapi *Adam* memiliki momentum (**Bera & Shrivastava, 2020**).

Arsitektur *ResNet* diinisiasi oleh Kaiming He untuk mengatasi masalah degradasi dengan *neural network*, yaitu saat *neural network* semakin dalam, tingkat *error* pada saat *training* sering meningkat. Arsitektur *ResNet* memiliki keunggulan yaitu dapat memecahkan masalah ketika *neural network* yang memiliki lapisan sangat dalam tingkat kesalahan pelatihan menjadi lebih tinggi. Ini dapat terjadi akibat *ResNet* memanfaatkan struktur *residual* yang menjadi *input* dari setiap *layer* (**Liang, 2020**).

Penelitian yang dilakukan oleh (**Hastomo, 2021**) melakukan deteksi penyakit paru-paru dengan 4 kelas *datasets* yaitu 3000 citra *lung opacity*, 1000 citra *viral pneumonia*, 3000 citra *covid* dan 3000 citra normal. Deteksi ini dilakukan menggunakan arsitektur *ResNet-152* dengan 50 *epoch* yang memiliki hasil *testing accuracy* yaitu 98%. Pada penelitian yang dilakukan (**Pranav et al., 2020**) memiliki *datasets* sebanyak 3700 data citra. Dari *datasets* tersebut memiliki 3 kelas yaitu *COVID-19*, *Normal* dan *Viral pneumonia*. Pada penelitian ini

menggunakan 3 arsitektur yang berbeda yaitu *VGGNet19*, *DenseNet121* dan *ResNet50*. Hasil dari penelitian ini menunjukkan bahwa *DenseNet121* dengan nilai 97% memiliki performa yang lebih baik dibandingkan dengan *VGGNet19* dengan nilai 96% dan *ResNet50* dengan nilai 93%. Selain itu, **(Sadre et al., 2021)** melakukan penelitian dengan menyiapkan *datasets* dengan 3 kelas yaitu *normal*, *pneumonia* dan *COVID-19* dengan berbasis citra *x-rays*. Penelitian ini menggunakan beberapa model arsitektur yaitu *ResNet-50*, *AlexNet*, dan *VGG-11*.

Berdasarkan hasil penelitian di atas, pada penelitian ini melakukan eksperimen performa *ResNet-152* untuk deteksi penyakit paru-paru melalui citra *x-rays* dengan menggunakan 2 *optimizer* yang berbeda yaitu *RMSprop* dan *Adam*.

2. METODE PENELITIAN

2.1. Adam Optimizer

Salah satu *optimizer* yang paling banyak diterapkan yakni *Adam Optimizer*, yang identik dengan penggunaannya dalam *deep learning*. Dalam penamaannya sendiri *Adam* berasal dari kata *Adaptive Moment Estimation* yang disingkat menjadi *Adam*. Terlebih karena kemampuannya yang dapat memperbarui bobot dan *learning rate* nya secara otomatis (*adaptive*). Jadi pada saat melakukan pelatihan dengan menggunakan *Adam Optimizer* untuk *learning rate* nya akan berubah-ubah karena kemampuan dari *Adam Optimizer* itu sendiri yang *learning rate* nya dapat bersifat adaptif. Bersamaan dengan itu *Adam* juga menggabungkan kelebihan dari *AdaGrad Optimizer* dan *RMSProp optimizer* yaitu dapat memecahkan masalah *sparse gradients* dan juga bekerja dengan baik dan memanfaatkan rata – rata pada momen pertama dan momen kedua dari gradien untuk mengadaptasi parameter *learning rate*, yang berbeda dengan *RMSProp* yang hanya memanfaatkan momen pertama **(Soydaner, 2020)**.

Adam Optimizer memerlukan momen pertama dan momen kedua dengan variabel m dan u , yang kemudian setelah menghitung gradien estimasi momen pertama dan kedua yang mengalami bias akan diperbarui pada setiap waktu langkah (t) masing – masing seperti yang ditunjukkan pada Persamaan (1) dan Persamaan (2) **(Soydaner, 2020)**.

$$m_t \leftarrow \rho_1 m_{t-1} + (1 - \rho_1) g_t \quad (1)$$

$$u_t \leftarrow \rho_2 u_{t-1} + (1 - \rho_2) g \odot g \quad (2)$$

Setelah bias diperbaiki pada momen pertama dan kedua, maka momen yang sudah diperbaiki digunakan untuk mengestimasi pembaruan parameter yang akan dihitung dan diterapkan, seperti yang tercantum pada Persamaan (3) hingga Persamaan (6) **(Soydaner, 2020)**.

$$\hat{m}_t \leftarrow \frac{m_t}{1 - \rho_1^t} \quad (3)$$

$$\hat{u}_t \leftarrow \frac{u_t}{1 - \rho_2^t} \quad (4)$$

$$\Delta\theta = -\epsilon \frac{\hat{m}_t}{\sqrt{\hat{u}_t + \delta}} \quad (5)$$

$$\theta_t \leftarrow \theta_{t-1} + \Delta\theta \quad (6)$$

2.2. RMSProp Optimizer

RMSProp adalah algoritma tambahan yang dimodifikasi dari AdaGrad. RMSProp diusulkan untuk lebih baik pada pengaturan *nonconvex* dengan mengubah akumulasi dari gradien ke dalam *moving average* eksponensial (**Soydner, 2020**). RMSProp merupakan singkatan dari *Root Mean Square* karena membagi gradien dengan akar kuadrat rata-rata. RMSProp dikembangkan pertama kali oleh Geoffrey Hinton sebagai teknik stokastik untuk *mini-batch*. Sama seperti Adam, RMSProp menggunakan *learning rate* yang adaptif pada saat proses *training* (**Wibowo et al., 2019**). Langkah untuk melakukan RMSProp dapat dengan melakukan gradien kuadrat diakumulasikan setelah menghitung gradien seperti Persamaan (7)

$$r \leftarrow \rho r + (1 - \rho)g \odot g \quad (7)$$

Kemudian, pembaruan dari parameter dihitung seperti Persamaan (8) dan (9)

$$\Delta\theta = -\epsilon \frac{e}{\sqrt{\delta+r}} \odot g \quad (8)$$

$$\theta \leftarrow \theta + \Delta\theta \quad (9)$$

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan jenis Deep Learning karena kedalaman jaringan yang tinggi dan kompleks. Selain itu, *CNN* juga pengembangan dari *Multilayer Perceptron (MLP)* yang dapat mengolah data dua dimensi (**E. P et al., 2016**). Sesuai dengan namanya, *CNN* menggunakan operasi matematika yang disebut konvolusi. Konvolusi membutuhkan kernel yang bergerak dari kiri atas hingga ke kanan bawah untuk mengekstraksi fitur dari citra (**Muniasamy & Alasiry, 2020**). *CNN* sangat populer dan banyak digunakan pada penelitian untuk pengenalan, identifikasi dan klasifikasi dari memanfaatkan teknik pemrosesan gambar dan pengenalan pola (**Khaliluzzaman et al., 2021**). Pada proses konvolusi terdapat *stride* untuk mengurangi parameter lebih banyak pada *CNN*. *Stride* merupakan langkah dari kernel untuk melakukan konvolusi. Selain itu, penggunaan *zero padding* dapat mengatasi masalah untuk kehilangan informasi dari *border* pada citra. *Zero padding* menambah piksel 0 pada citra (**Albawi et al., 2017**).

Arsitektur *CNN* yang dipilih pada penelitian ini yaitu *ResNet (Residual Network)*. Konsep utama *ResNet* merupakan *identity shortcut link*. Ketika jaringan menjadi lebih dalam, kinerjanya semakin menurun dengan cepat. Masalah tersebut dapat diatasi menggunakan *identity shortcut link*.

ResNet bekerja dengan melewati 2 hingga 3 *layer* pada arsitektur sehingga *residual network* tidak memiliki parameter tambahan seperti *neural network* biasanya. Arsitektur *ResNet-152* memiliki *top-1 error* dan *top-5 error* yang sangat rendah masing-masing sekitar 21,43 % dan 5,71 % (**He et al., 2016**). Pada Gambar 1 terdapat arsitektur dari semua versi *ResNet* yaitu *ResNet-18*, *ResNet-34*, *ResNet-50*, *ResNet-101*, dan *ResNet-152*. Pada penelitian ini menggunakan *ResNet-152* yang memiliki komposisi lapisan yang hampir sama dengan versi *ResNet* lainnya. *ResNet-152* diawali dengan *convolution 7x7* dengan *stride 2*. Kemudian melakukan *max pooling 3x3* dengan *stride 2*. Kemudian proses *batch normalization 1x1*, *ReLU 1x1* dan *convolution 3x3* pada setiap *conv2_x*, *conv3_x*, *conv4_x* dan *conv5_x*. Kemudian dilakukan *average pooling* dan *softmax* untuk hasil klasifikasi.

Arsitektur Resnet-152 dengan Perbandingan Optimizer Adam dan RMSProp untuk Mendeteksi Penyakit Paru - Paru

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

Gambar 1. Arsitektur ResNet

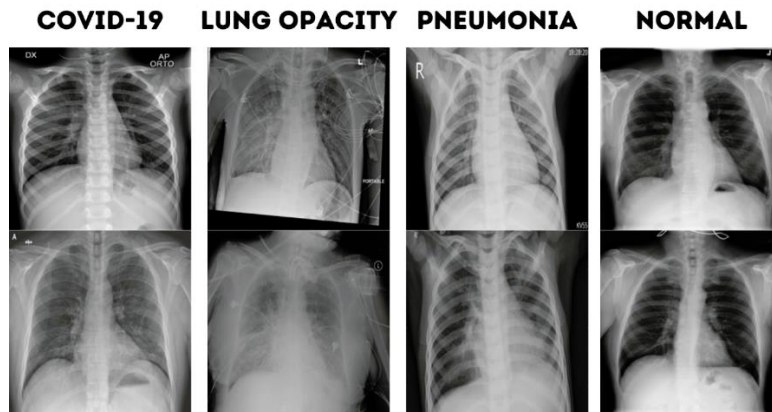
Dari 5 versi arsitektur pada *ResNet*, arsitektur *ResNet-152* memiliki *top-1* dan *top-5* error yang paling rendah. Ini membuktikan bahwa *ResNet-152* merupakan versi terbaik dari semua model *ResNet* seperti terlihat pada Gambar 2.

model	top-1 err.	top-5 err.
VGG-16 [40]	28.07	9.33
GoogLeNet [43]	-	9.15
PReLU-net [12]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	21.43	5.71

Gambar 2. Top-1 dan Top-5 Error ResNet

2.4. Deskripsi Datasets

Sumber data yang digunakan pada penelitian ini berupa citra *x-rays* paru – paru yang memiliki format citra *png* yang berjumlah 21165 buah citra dan terdiri dari 4 kelompok yaitu normal, *covid-19*, *lung opacity* dan *viral pneumonia*. Data ini diperoleh melalui *Kaggle* dalam judul *Covid-19 Radiography Database*, data tersebut dibuat oleh (**Chowdhury et al., 2020**) dan (**Rahman et al., 2021**).

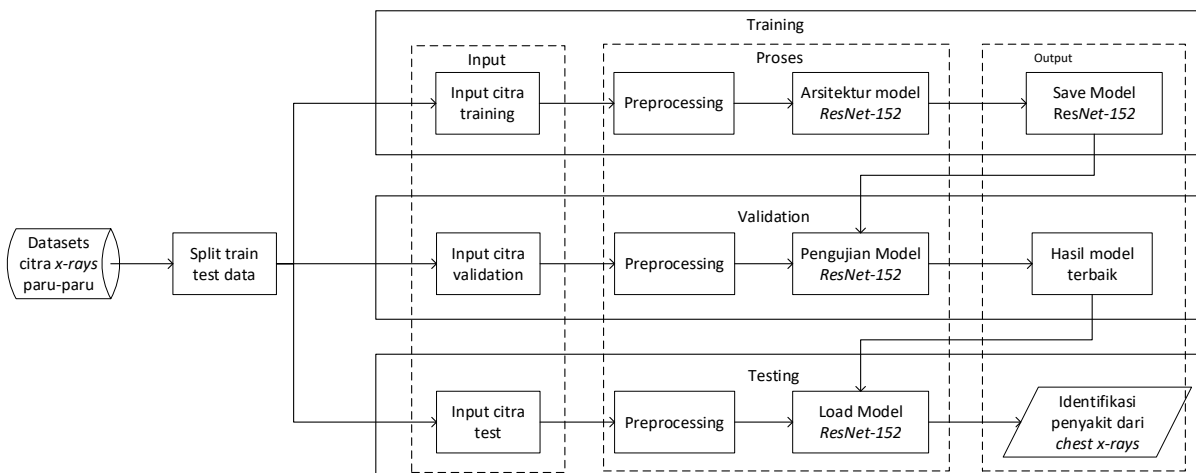


Gambar 3. Contoh Dataset

2.6 Perancangan Umum

2.6.1 Blok Diagram

Keseluruhan *datasets* yang akan digunakan pada penelitian akan dibagi menjadi 3 proses yaitu *training*, *validation* dan *testing*, untuk kejelasan proses nya dapat dilihat pada Gambar 4. Dimulai dengan memasukkan *datasets* citra yang telah dibagi untuk setiap proses nya lalu masuk ke dalam proses *preprocessing* yang dimana citra akan di olah dengan salah satu proses *preprocessing* yaitu menggunakan proses *resize* atau pengubahan ukuran citra dari yang awalnya berukuran 299 x 299 akan di *resize* menjadi 224 x 224 untuk keperluan masukkan pada tahap selanjutnya. Setelah citra selesai dilakukan *preprocessing* maka selanjutnya akan masuk ke tahapan pelatihan model yaitu model *ResNet-152* untuk proses *training*, yang dimana setelah itu akan menghasilkan hasil pelatihan model. Kemudian model dari hasil *training* akan dievaluasi di tahap *validation* dengan menggunakan *datasets validation* sebanyak 10%. Selanjutnya ketika model dari hasil *training* sudah siap maka akan disimpan dan akan dipanggil serta digunakan kembali pada saat proses *testing*.



Gambar 4. Blok Diagram Pembuatan Model

3. HASIL DAN PEMBAHASAN

3.1 Penerapan *Datasets*

Data tersebut dilakukan proses pemecahan untuk kemudian dibagi – bagi menjadi proses *training*, *testing* dan *validation* dengan masing – masing jumlah pemecahan sebanyak 70%, 20% dan 10%, mengenai rincian pemecahan tersebut tercantum dalam Tabel 1.

Tabel 1. Pembagian *Datasets*

Data	<i>Training (70%)</i>	<i>Testing (20%)</i>	Validation (10%)	Jumlah
Normal	7.134	2.038	1.019	10.192
Covid-19	2.531	723	362	3.616
Lung Opacity	4.208	1.203	601	6.012
Viral Pneumonia	942	269	135	1.345
Total	14.815	4.233	2.117	21.165

Seperti yang tercantum pada Tabel 1 untuk tahapan *training* terdiri dari 14.815 buah citra, tahapan *testing* terdiri dari 4.233 buah citra dan untuk tahapan *validation* terdiri dari 2.117 buah citra. Seluruh jumlah citra untuk setiap tahapan tersebut merupakan gabungan dari setiap kelas atau kelompok pada *datasets* yang akan diteliti.

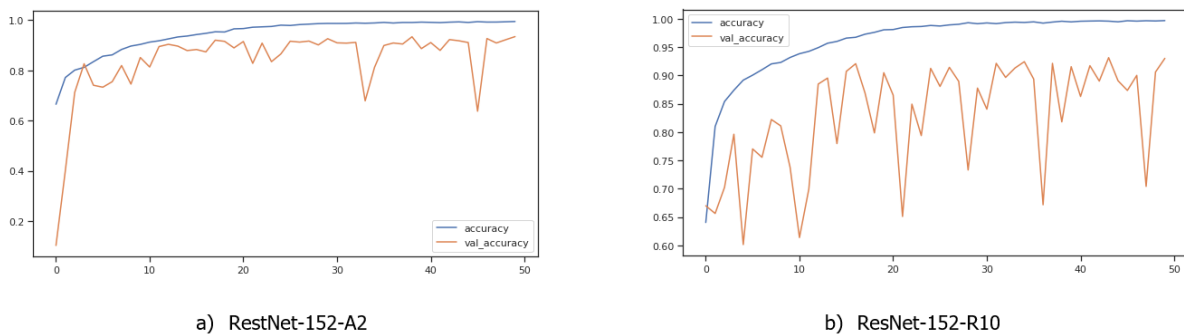
3.2 Pelatihan Model (*Training Model*)

Pada tahap pelatihan model ini dilakukan dengan cara membuat model *ResNet-152* yang kemudian dengan mengatur *hyperparameter* yang akan digunakan seperti yang tercantum dalam Tabel 2.

Tabel 2. *Hyperparameter* Setiap Model

Model	<i>Hyperparameter</i>
<i>ResNet-152-R10</i>	<i>RMSProp Optimizer</i>
	<i>Learning Rate 0,0001</i>
	<i>Batch Size 8</i>
<i>ResNet-152-A2</i>	<i>Adam Optimizer</i>
	<i>Learning rate 0,001</i>
	<i>Batch Size 8</i>

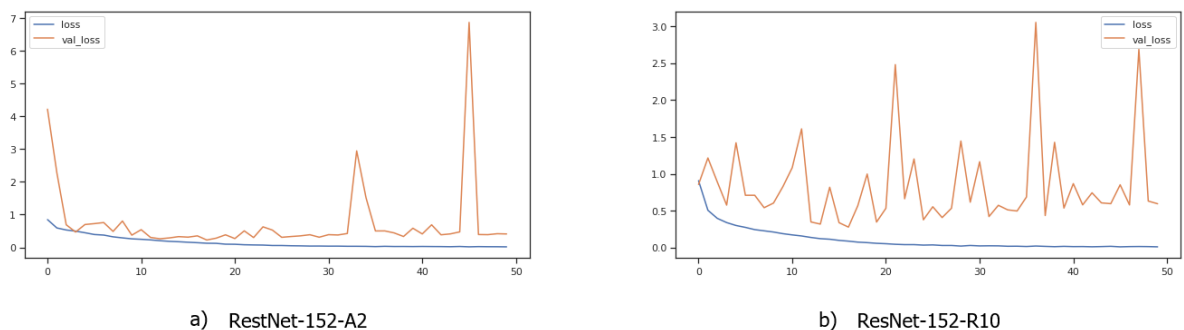
Percobaan pada tahap pelatihan ini menggunakan beberapa *hyperparameter* yang sudah tercantum pada Tabel 2, lalu kemudian masing – masing model juga menggunakan *epoch* yang sama untuk setiap proses *training*-nya yaitu sebesar 50 *epoch*. Kedua model tersebut akan dilakukan *training* dengan melihat dari hasil *accuracy* dan *validation accuracy*. *Accuracy* berasal dari data *training* 70%, sedangkan untuk *validation* berasal dari data *validation* 10% dari total *datasets*. *Validation* sangat berpengaruh apakah model bekerja dengan baik atau tidak. Idealnya, model yang baik pada saat *training* memiliki *validation* yang mendekati hasil dari *accuracy* karena model melakukan prediksi dari data yang belum dilihat pada saat *training*. Dari hasil percobaan tersebut didapatkan hasil *accuracy*, dan *validation accuracy* seperti yang tercantum pada Gambar 5.



Gambar 5. Grafik Accuracy Training

Pada Gambar 5 menampilkan perubahan – perubahan *accuracy* dan *validation accuracy* dimulai dari *epoch* pertama hingga *epoch* terakhir. Dapat dilihat pada grafik bahwa *validation accuracy* pada *ResNet-152-A2* cenderung lebih stabil atau mendekati nilai *accuracy* pada setiap *epoch*-nya. Tetapi jika dilihat dari nilai akurasi akhir-nya pun tidak berbeda jauh yaitu untuk model *ResNet-152-A2* nilai akurasi sekitar 99,46% dan validasi sekitar 93,45%, model *ResNet-152-R10* nilai akurasi sekitar 99,70% dan validasi sekitar 93,03%.

Selain grafik proses *training accuracy* juga terdapat grafik proses *training loss* pada saat pelatihan model. *Loss* dapat diartikan sebagai kesalahan atau *error* yang dilakukan oleh model ketika *training*. Nilai *loss* yang baik yaitu mendekati 0. Pada Gambar 6 memperlihatkan grafik *training loss* pada model.



Gambar 6. Grafik Loss Training

Pada Gambar 6 terdapat grafik *loss* dan *validation loss* dari keempat model. Dari grafik tersebut terlihat untuk garis *validation loss* model *ResNet-152-R10* sangat tidak stabil dalam proses pelatihan model. Model *ResNet-152-A2* mendapatkan nilai *loss* 0,016 dan *validation loss* 0,410. Sedangkan model *Resnet-152-R10* mendapatkan nilai *loss* 0,010 dan *validation loss* 0,594. Dari nilai *loss* dan *validation loss* terlihat bahwa Model *Resnet-152-R10* merupakan model yang mempunyai nilai *loss* terbaik dan Model *ResNet-152-A2* mempunyai nilai *validation loss* terbaik.

3.3 Pengujian Model (*Testing Model*)

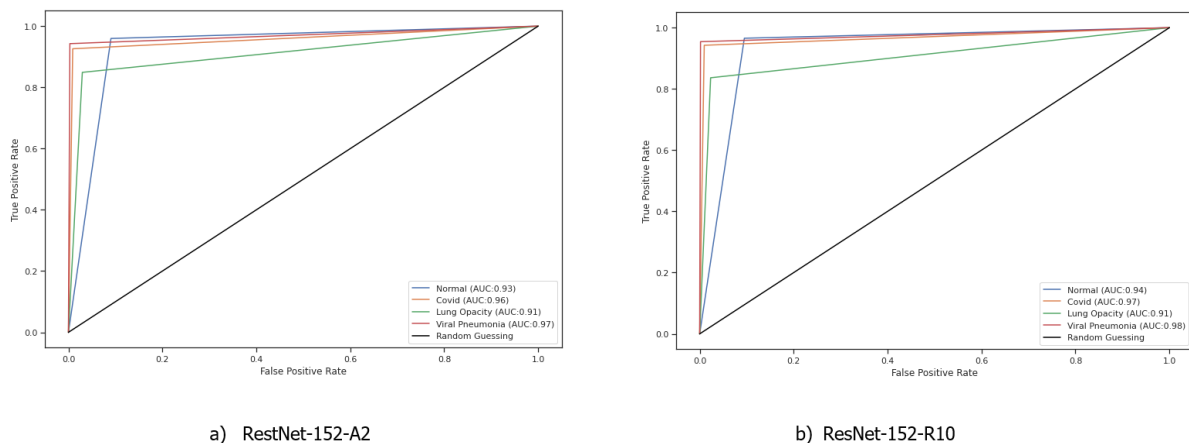
Setelah tahapan *training* selesai selanjutnya masuk ke tahapan *testing* atau pengujian model dengan menggunakan *datasets* sejumlah 20% dari total keseluruhan *datasets*. Pada Tabel 1 terlihat *datasets* yang ada pada setiap kelas berbeda-beda jumlahnya sehingga untuk melakukan pengujian model tidak bisa menggunakan metrik *accuracy* saja, melainkan dibutuhkan metrik lain seperti *precision* ataupun *recall* yang cocok dengan kasus ini. Maka dari

itu, di tahapan pengujian model ini menghasilkan beberapa bentuk evaluasi model seperti *accuracy*, *precision*, *recall*, *F1-score*, *specificity* serta grafik *roc*. Seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Hasil Pengujian

Model	Accuracy	Precision	Recall/ Sensitivity	F1-Score	Specificity
<i>ResNet-152-A2</i>	92,00%	93,00%	92,00%	93,00%	96,70%
<i>ResNet-152-R10</i>	92,00%	94,00%	92,00%	93,00%	96,75%

Di dalam Tabel 3 dapat dilihat bahwa model terbaik diantara 2 model yaitu *ResNet-152-R10* dengan nilai *precision* dan *specificity* yang paling tinggi walaupun tidak berbeda jauh nilainya. Selain itu juga dapat dilihat dari grafik *roc* yang tercantum pada Gambar 7.



Gambar 7. Grafik ROC

Pada grafik ROC ini akan menampilkan TPR (*True Positive Rate*) dan FPR (*False Positive Rate*) yang dimana jika setiap garis dari setiap *class* mendekati nilai 1 pada sumbu TPR dan menjauhi nilai 1 pada sumbu FPR maka model tersebut dapat dikatakan sebagai model yang baik. Selain itu bisa juga dilihat dari *ROC AUC Score* yang tercantum dalam Tabel 4.

Tabel 4. Nilai ROC dan AUC

Model	ROC AUC Score
<i>ResNet-152-A2</i>	0,943
<i>ResNet-152-R10</i>	0,946

Dapat dilihat bahwa pada Tabel 4 model *ResNet-152-R10* memiliki nilai *ROC AUC Score* yang lebih tinggi dibanding model yang lainnya, sehingga semakin membuktikan bahwa model *ResNet-152-R10* adalah model yang cukup baik.

Pada penelitian ini meneliti mengenai perbandingan performa model ResNet seperti yang dilakukan pada penelitian (**Pranav et al., 2020**) dan (**Sadre et al., 2021**) tetapi dengan menggunakan versi yang terbaru yaitu *ResNet-152* dan menggunakan 2 optimizer yang

berbeda yaitu *Adam* dan *RMSProp*. Selain itu, dilakukan uji *hyperparameter* lain dengan mengatur *learning rate* dan *batch size* pada kedua model arsitektur.

4. KESIMPULAN

Penelitian yang telah dilakukan adalah dengan membandingkan arsitektur *ResNet-152* beserta dengan 2 *optimizer* yaitu *Adam* dan *RMSProp* untuk deteksi penyakit paru-paru pada manusia. Hasil dari penelitian mendapatkan 2 model berbeda dengan menunjukkan nilai *accuracy*, *precision*, *recall*, *f1-score* dan *specificity* di atas 92%. Model *ResNet-152-R10* merupakan model terbaik dengan memiliki nilai *accuracy* sebesar 92,00%, *precision* sebesar 94,00%, *recall* sebesar 92,00%, *f1-score* sebesar 93,00% dan *specificity* sebesar 96,75%. Selain itu, nilai *AUC* pada model *ResNet-152-R10* memiliki nilai 0,946 yaitu tertinggi dari model lain. Model tersebut mempunyai *hyperparameter* yaitu menggunakan *RMSProp optimizer*, *learning rate* 0.001 dan *batch size* 8. Pada penelitian yang telah dilakukan, model *ResNet* yang memiliki *RMSProp optimizer* memiliki performa yang lebih baik karena memiliki *learning rate* yang adaptif dan cocok digunakan untuk *datasets mini-batch*. Pada penelitian ini menggunakan *batch size* yaitu 8 yang termasuk *datasets mini-batch*.

DAFTAR RUJUKAN

- Albawi, S., Mohammed, T. A. M., & Alzawi, S. (2017). Layers of a Convolutional Neural Network. *Ieee*, 16.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In *Journal of Big Data* (Vol. 8, Issue 1). Springer International Publishing. <https://doi.org/10.1186/s40537-021-00444-8>
- Bera, S., & Shrivastava, V. K. (2020). Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. *International Journal of Remote Sensing*, 41(7), 2664–2683. <https://doi.org/10.1080/01431161.2019.1694725>
- Bharati, S., Podder, P., & Mondal, M. R. H. (2020). Hybrid deep learning for detecting lung diseases from X-ray images. *Informatics in Medicine Unlocked*, 20, 100391. <https://doi.org/10.1016/j.imu.2020.100391>
- Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. Bin, Islam, K. R., Khan, M. S., Iqbal, A., Emadi, N. Al, Reaz, M. B. I., & Islam, M. T. (2020). Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access*, 8, 132665–132676. <https://doi.org/10.1109/ACCESS.2020.3010287>
- Chowdhury, N. K., Kabir, M. A., Rahman, M. M., & Rezoana, N. (2020). *ECOVNet: An Ensemble of Deep Convolutional Neural Networks Based on EfficientNet to Detect COVID-19 From Chest X-rays*. <https://doi.org/10.7717/peerj-cs.551>
- Deng, L., & Yu, D. (2014). *REAL-TIME COLOR IMAGE CLASSIFICATION BASED ON DEEP LEARNING NETWORK*. 54(5), 1–134. <https://doi.org/10.1227/01.NEU.0000255452.20602.C9>

- E. P, I. W. S., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101. *Jurnal Teknik ITS*, 5(1), 76. <http://repository.its.ac.id/48842/>
- Hastomo, W. (2021). Diagnosa COVID-19 Chest X-Ray Dengan Convolution Neural Network Arsitektur Resnet-152. *KERNEL: Jurnal Riset Inovasi Bidang Informatika Dan Pendidikan Informatika*, 2(1), 26–33. <https://doi.org/10.31284/j.kernel.2021.v2i1.1884>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Jabra, M. Ben, Koubaa, A., Benjdira, B., Ammar, A., & Hamam, H. (2021). Covid-19 diagnosis in chest x-rays using deep learning and majority voting. *Applied Sciences (Switzerland)*, 11(6). <https://doi.org/10.3390/app11062884>
- Khaliluzzaman, M., Md. Abu Bakar Siddiq Sayem, & Lutful KaderMisbah. (2021). HActivityNet: A Deep Convolutional Neural Network for Human Activity Recognition. *EMITTER International Journal of Engineering Technology*, 9(2), 357–376. <https://doi.org/10.24003/emitter.v9i2.642>
- Kumar, V., & L., M. (2018). Deep Learning as a Frontier of Machine Learning: A Review. *International Journal of Computer Applications*, 182(1), 22–30. <https://doi.org/10.5120/ijca2018917433>
- Liang, J. (2020). Image classification based on RESNET. *Journal of Physics: Conference Series*, 1634(1). <https://doi.org/10.1088/1742-6596/1634/1/012110>
- Muniasamy, A., & Alasiry, A. (2020). Deep learning: The impact on future eLearning. *International Journal of Emerging Technologies in Learning*, 15(1), 188–199. <https://doi.org/10.3991/IJET.V15I01.11435>
- Pranav, J. V., Anand, R., Shanthi, T., Manju, K., Veni, S., & Nagarjun, S. (2020). Detection and identification of COVID -19 based on chest medical image by using convolutional neural networks. *International Journal of Intelligent Networks*, 1(November), 112–118. <https://doi.org/10.1016/j.ijin.2020.12.002>
- Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Abul Kashem, S. Bin, Islam, M. T., Al Maadeed, S., Zughaier, S. M., Khan, M. S., & Chowdhury, M. E. H. (2021). Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in Biology and Medicine*, 132(November 2020), 104319. <https://doi.org/10.1016/j.compbiomed.2021.104319>
- Sadre, R., Sundaram, B., Majumdar, S., & Ushizima, D. (2021). Validating deep learning inference during chest X-ray classification for COVID-19 screening. *Scientific Reports*, 11(1), 1–10. <https://doi.org/10.1038/s41598-021-95561-y>
- Sensusiati, A. D., Pramulen, A. S., Rumala, D. J., & ... (2021). A New Approach to Detect COVID-19 in X-Ray Images of Indonesians. *Journal of Hunan ...*, 48(6). <http://www.jonuns.com/index.php/journal/article/view/595>
- Soydaner, D. (2020). A Comparison of Optimization Algorithms for Deep Learning.

International Journal of Pattern Recognition and Artificial Intelligence, 34(13), 1–26.
<https://doi.org/10.1142/S0218001420520138>

Wibowo, A., Wiryawan, P. W., & Nuqoyati, N. I. (2019). Optimization of neural network for cancer microRNA biomarkers classification. *Journal of Physics: Conference Series*, 1217(1). <https://doi.org/10.1088/1742-6596/1217/1/012124>