

ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* DALAM KLASIFIKASI *CHEST X-RAYS* PASIEN COVID-19 MENGGUNAKAN FUNGSI AKTIVASI *SIGMOID*

Muhamad Nurhikmat Zain¹, Dwi Agustin Nuriani Sirodj^{2*}

¹Program Studi Statistika, Fakultas Matematika dan Ilmu Pengentahuan
Alam, Universitas Islam Bandung, Indonesia
Email: muhamadzain019@gmail.com

^{2*}Program Studi Statistika, Fakultas Matematika dan Ilmu Pnegerahuan
Alam, Universitas Islam Bandung, Indonesia
Email Korespondensi: dwi.agustinnuriani@unisba.ac.id

ABSTRACT

Convolutional neural network is the development of an artificial neural network. Convolutional neural network can be used to recognize objects from an image. This study will classify chest x-rays of COVID-19 patients. The process of classification using sigmoid activation function for the case of binary classification. Currently, to diagnose COVID-19 using Reverse Transcription Polymerase Chain Reaction (RT-PCR). However, there are several problems with the test mechanism (RT-PCR), has to do it special tools, materials, and it takes quite a long time. Another solution that can help is, use digital images form of chest x-rays to retrieve information and recognize objects automatically using convolutional neural network method. The results of chest x-rays classification using a convolutional neural network method with sigmoid activation function, get an accuracy of 91%, precision of 86%, and recall of 79%. Accuracy of 91% indicates that the model is excellent at classify chest x-rays.

Keywords: *Convolutional neural network, Classification, Chest x-rays, COVID-19.*

ABSTRAK

Convolutional neural network merupakan pengembangan dari artificial neural network. Penelitian ini akan melakukan klasifikasi chest x-rays pasien COVID-19. Proses klasifikasi dilakukan dengan menggunakan bantuan fungsi aktivasi sigmoid untuk kasus klasifikasi biner. Selama ini, untuk mendiagnosa COVID-19 dilakukan dengan tes Reverse Transcription Polymerase Chain Reaction (RT-PCR). Namun, ada beberapa masalah pada mekanisme pengujian tes (RT-PCR) diantaranya perlunya alat dan bahan khusus dan memakan waktu yang cukup lama. Solusi yang lebih cepat dan berbasis data sangat diperlukan dibandingkan melakukan tes (RT-PCR). Salah satu cara yang dapat membantu adalah dengan memanfaatkan citra digital berupa chest x-rays untuk mengambil informasi dan mengenali objek secara otomatis dengan menggunakan metode convolutional neural network. Hasil klasifikasi chest x-rays dengan arsitektur convolutional neural network yang telah dibangun mendapatkan nilai akurasi sebesar 91%, presisi sebesar 86% dan recall sebesar 79%. Akurasi sebesar 91% menandakan model sangat baik dalam melakukan klasifikasi pada chest x-rays.

Kata Kunci: *Convolutional neural network, Klasifikasi, Chest x-rays, COVID-19.*

1. PENDAHULUAN

Machine learning adalah pemrograman komputer untuk mengoptimalkan performa menggunakan data sampel atau berdasarkan pengalaman masa lalu dengan menggunakan teori statistika untuk membangun model (Alpaydın, 2020). *Artificial neural network* merupakan metode *machine learning* yang memiliki tiga hingga empat *layer* (Setiawan, 2020). Tiap *layer* memiliki banyak *neuron*. *Layer-layer* tersebut dinamakan *multilayer perceptron* (MLP) yang berfungsi menghubungkan tiap *neuron*. Namun, *artificial neural network* memiliki keterbatasan pada data yang lebih kompleks. Pengembangan lain dari *artificial neural network* untuk mengatasi masalah tersebut dengan menggunakan *deep learning*. Pada *deep learning* ekstraksi fitur dilakukan secara otomatis sedangkan ekstraksi fitur pada *machine learning* dilakukan secara manual.

Salah satu metode *deep learning* sebagai pengembangan dari *artificial neural network* yaitu *convolutional neural network* (CNN) yang dapat digunakan untuk mengenali dan mendeteksi objek dari suatu citra. Dibandingkan dengan *artificial neural network*, *convolutional neural network* memiliki kemampuan untuk menghasilkan fitur-fitur yang relevan sesuai input citra yang diberikan (Setiawan, 2020). Secara umum mekanisme *convolutional neural network* harus melewati beberapa tahapan pada tiap-tiap *layer*. *Layer* dari ekstraksi fitur terdiri dari *convolutional layer*, *pooling layer*, dan *Rectified Linier Unit* (ReLU). Sedangkan *layer* klasifikasi terdiri dari *fully connected layer*, *dropout* dan *softmax* untuk klasifikasi *multiclass* (Krizhevsky et al., 2012).

Penelitian mengenai perbandingan metode SVM, ANN dan CNN pada dataset citra jenis daun didapatkan akurasi dari SVM 91%, ANN 94% dan CNN sebesar 99% (Hasan et al., 2019). Kekurangan dari *convolutional neural network* yaitu proses pelatihan model yang cukup lama namun hal tersebut dapat diatasi dengan menggunakan *graphical processing unit* (GPU). Selain itu dengan menggunakan metode *convolutional neural network* dapat mengurangi jumlah parameter yang dihasilkan.

Dunia global termasuk Indonesia sedang menghadapi penyebaran wabah penyakit yang disebabkan oleh virus COVID-19. Selama ini untuk mendiagnosa COVID-19 yang dilakukan oleh dokter dan tenaga kesehatan dilakukan dengan tes *Reverse Transcription Polymerase Chain Reaction* (RT-PCR) Namun, ada beberapa masalah pada mekanisme pengujian tes (RT-PCR) diantaranya perlunya alat dan bahan khusus, memakan waktu yang cukup lama, sensitivitas yang relatif rendah dan membuat pasien tidak nyaman (Hassantabar et al., 2020). Dengan semakin tingginya kasus COVID-19 dan terbatasnya sumber daya manusia diperlukan metode diagnostik yang lain untuk mengatasi masalah tersebut. Solusi yang lebih cepat berbasis data sangat diperlukan dibandingkan melakukan tes (RT-PCR). Salah satu metode yang dapat membantu adalah dengan memanfaatkan citra digital untuk mengambil informasi dan mengenali objek secara otomatis dengan ilmu komputasi. Citra yang dapat digunakan pada penelitian ini adalah citra *chest x-rays*.

Penelitian ini menghasilkan model yang dapat belajar dari citra *chest x-rays* dan dapat mengklasifikasikan citra *chest x-rays* itu sendiri, serta dapat menghasilkan kinerja model yang sangat baik menggunakan *convolutional neural network*. Berdasarkan hal – hal yang sudah dijelaskan sebelumnya, penulis tertarik menggunakan metode *convolutional neural network* untuk mengklasifikasikan citra *chest x-rays* pada pasien COVID-19 menggunakan fungsi aktivasi *sigmoid*.

2. METODOLOGI

Jenis data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari

kaggle www.kaggle.com/ahmadalmahsiri/covid19-radiography-database. Variabel penelitian yang digunakan dalam penelitian ini adalah *chest x-rays* yang telah diberi label yaitu citra *chest x-rays* yang terkena penyakit COVID-19 dan citra *chest x-rays* yang tidak terkena penyakit COVID-19. Citra *chest x-rays* yang terkumpul selama periode Juni 2020 – Desember 2021 ada sebanyak 3.616 citra *chest x-rays* yang terkena penyakit COVID-19 dan 10.192 citra *chest x-rays* normal.

Tahapan analisis data yang dilakukan sebagai berikut:

- a. Input data citra *chest x-rays*, dimisalkan variabel x sebagai citra *x-rays* dan variabel y sebagai label.
- b. Melakukan *preprocessing* pada citra *x-rays*
 - (i) Menyeragamkan *pixels* pada semua citra *chest x-rays* yaitu 64×64 *pixels*.
 - (ii) Memberikan label data menjadi kategori, nilai 0 sebagai label COVID-19 dan nilai 1 sebagai label normal.
 - (iii) Membagi data citra menjadi data *training* dan data *testing* dengan proporsi 80:20. Berdasarkan hasil penelitian oleh (Kabir Anaraki et al., 2018) dalam mengklasifikasikan citra tumor otak menggunakan proporsi data *training* dan data *testing* sebesar 80:20 memberikan hasil akurasi yang paling baik yaitu 94.2% menggunakan metode *convolutional neural network*.
 - (iv) Melakukan *rescale* nilai pada setiap *pixels* dengan melakukan operasi pembagian $1/225$ sehingga mendapatkan nilai diantara 0-1.
- c. Membuat model arsitektur *convolutional neural network* yang terdiri dari *layer feature learning* dan *layer* klasifikasi.

Layer feature learning:

- (i) Membuat *convolution layer* dua dimensi dengan input citra 64×64 *pixels*, jumlah kernel 32, ukuran kernel 3×3 , *stride* 1 dan *zero padding*.
- (ii) Melakukan perhitungan pada *convolutional layer* menggunakan persamaan (1) dan ukuran output dan banyaknya parameter yang dihasilkan dapat dihitung menggunakan persamaan (2) dan (3)

$$s(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (1)$$

$$\text{Output} = \frac{W - F + 2P}{S} + 1 \quad (2)$$

$$\text{Banyak Parameter} = (\text{banyak feature map sebelumnya} \times \text{banyak filter} \times \text{ukuran filter}) + \text{bias} \quad (3)$$

Keterangan:

$s(i, j)$ = Fungsi hasil operasi konvolusi (*feature map*) di suatu lokasi

K = Kernel atau filter

I = Input citra

i dan j = Posisi dari kernel atau filter

m dan n = Posisi dari citra

W = Panjang/tinggi input citra

F = Panjang/tinggi filter

P = *padding*

S = *stride*

- (iii) Melakukan aktivasi menggunakan fungsi aktivasi ReLU menggunakan persamaan (4).

$$f(x) = \max(0, x) \quad (4)$$

Keterangan:

$f(x)$ = Fungsi aktivasi ReLU

x = Nilai dari matriks hasil konvolusi

(iv) Melakukan *pooling* menggunakan *max-pooling* dengan *stride* 2x2.

Layer klasifikasi:

(i) Mengubah *layer* dua dimensi menjadi vektor satu dimensi yang akan digunakan sebagai input citra.

(ii) Membuat *fully connected layer* dengan vektor satu dimensi sebagai input citra yang terhubung ke *hidden layer* ke *output layer*. Ukuran output pada *flatten layer* dan banyaknya parameter yang terbentuk pada *fully connected layer* dapat dihitung menggunakan persamaan (5) dan (6).

$$\text{Output} = \text{banyak feature map sebelumnya} \times \text{ukuran output sebelumnya} \quad (5)$$

$$\text{Banyak Parameter} = (\text{banyak input} + 1) \times \text{banyak neuron pada layer} \quad (6)$$

(iii) Melakukan *dropout* pada *fully connected layer* untuk mengeluarkan *neuron* yang tidak akan digunakan pada proses pelatihan.

d. Menetapkan *epoch* sebanyak 40 dan *learning rate* 0.1 untuk melakukan proses pengujian model dengan data *training* dilakukan proses *feedforward* dan *backpropagation* (Srinivasan et al., 2002) menggunakan persamaan (9) s.d. (19) dengan fungsi aktivasi *sigmoid* dari *hidden layer* ke *output layer* menggunakan persamaan (7).

$$y_k = \frac{1}{1 + e^{y_{in_k}}} \quad (7)$$

$$L = -y \log y_k - (1 - y) \log (1 - y_k) \quad (8)$$

Keterangan:

e = Bilangan eksponensial

y_{in_k} = Nilai pada *neuron* output

y = Nilai aktual (0 atau 1)

y_k = Nilai prediksi hasil klasifikasi fungsi *sigmoid*

Proses *Backpropagation*:

(i) Inisialisasi bobot dengan nilai *random* yang cukup kecil. Dalam menentukan nilai acak tersebut pada umumnya dibatasi pada *range* tertentu diantaranya -1 sampai 1.

(ii) Tetapkan *epoch* maksimum dan *learning rate*.

(iii) Ketika kondisi berhenti tidak terpenuhi, lakukan langkah (iv) sampai langkah (xi).

(iv) Setiap unit *input* ($X_i, i = 1, \dots, n$) menerima sinyal *input* x_i .

(v) Setiap unit *hidden* ($Z_j, j = 1, \dots, p$) menghitung nilai z_j .

$$z_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (9)$$

(vi) Setiap unit *output* ($Y_k = k = 1, \dots, m$) menghitung nilai y_{in_k} dan y_k .

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (10)$$

$$y_k = f(y_{in_k}) \quad (11)$$

(vii) Setiap unit *output* ($Y_k = k = 1, \dots, m$) menghitung nilai *loss* pada *output layer* menggunakan persamaan (8).

(viii) Menghitung nilai *gradient loss* terhadap bobot dan *bias* pada *output layer* ke *hidden layer*

$$\frac{\partial L}{\partial w_{jk}} = \frac{\partial L}{\partial y_k} \times \frac{\partial y_k}{\partial y_{in_k}} \times \frac{\partial y_{in_k}}{\partial w_{jk}} \quad (12)$$

$$\frac{\partial L}{\partial w_{0k}} = \frac{\partial L}{\partial y_k} \times \frac{\partial y_k}{\partial y_{in_k}} \times \frac{\partial y_{in_k}}{\partial w_{0k}} \quad (13)$$

(ix) *Update* nilai bobot dan *bias* pada *output layer* ke *hidden layer*.

$$w_{jk}^{(l+1)} = w_{jk}^{(l)} - \alpha \frac{\partial L}{\partial w_{jk}} \quad (14)$$

$$w_{0k}^{(l+1)} = w_{0k}^{(l)} - \alpha \frac{\partial L}{\partial w_{0k}} \quad (15)$$

(x) Menghitung nilai *gradient loss* terhadap bobot dan *bias* pada *hidden layer* ke *input layer*.

$$\frac{\partial L}{\partial v_{ij}} = \frac{\partial L}{\partial y_k} \times \frac{\partial y_k}{\partial y_{in_k}} \times \frac{\partial y_{in_k}}{\partial z_j} \times \frac{\partial z_j}{\partial v_{ij}} \quad (16)$$

$$\frac{\partial L}{\partial v_{0j}} = \frac{\partial L}{\partial y_k} \times \frac{\partial y_k}{\partial y_{in_k}} \times \frac{\partial y_{in_k}}{\partial z_j} \times \frac{\partial z_j}{\partial v_{0j}} \quad (17)$$

(xi) *Update* nilai bobot dan *bias* pada *hidden layer* ke *input layer*.

$$v_{ij}^{(l+1)} = v_{ij}^{(l)} - \alpha \frac{\partial L}{\partial v_{ij}} \quad (18)$$

$$v_{0j}^{(l+1)} = v_{0j}^{(l)} - \alpha \frac{\partial L}{\partial v_{0j}} \quad (19)$$

(xii) Uji kondisi berhenti. Kondisi berhenti ini terpenuhi jika nilai *loss* yang dihasilkan lebih kecil dari nilai target *loss*. Sebagai inisialisasi awal ditetapkan target *loss* 0.1.

Keterangan:

- n = Banyaknya *neuron* pada *input layer*
- p = Banyaknya *neuron* pada *hidden layer*
- m = Banyaknya *neuron* pada *output layer*
- L = Nilai *loss* pada *output layer*
- x_i = Unit *input* ke- i
- z_j = Unit *hidden* ke- j
- y_k = Unit *output* ke- k
- α = *Learning rate*
- v_{0j} = *Bias* untuk unit *hidden* ke- j
- v_{ij} = Bobot dari unit *input* ke- i ke unit *hidden* ke- j
- w_{0k} = *Bias* untuk unit *output* ke- k
- w_{jk} = Bobot dari unit *hidden* ke- j ke unit *output* ke- k

e. Melakukan pengujian model menggunakan data *testing*. Proses pengujian model menggunakan data *testing* sama seperti point “d” hanya saja pengujiannya hanya melakukan proses *feed forward*.

- f. Membuat *confusion matrix* untuk mengevaluasi model dengan melihat tingkat akurasi, presisi dan *recall* menggunakan persamaan (20), (21), dan (22).

Tabel 1 Pedoman parameter hasil klasifikasi

Rentang	Hasil Klasifikasi
90 – 100%	<i>Excellent classification</i>
80-90%	<i>Good classification</i>
70-80%	<i>Fair classification</i>
60-70%	<i>Poor classification</i>
50-60%	<i>Failure</i>

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\% \quad (20)$$

$$\text{Presisi} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (21)$$

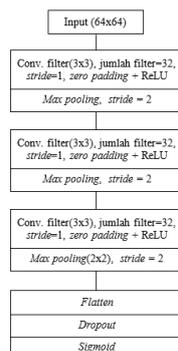
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (22)$$

- g. Melakukan interpretasi dari hasil analisis yang telah dilakukan.
 h. Menarik kesimpulan dan saran.

3. HASIL DAN PEMBAHASAN

3.1. Arsitektur *Convolutional Neural Network*

Arsitektur *convolutional neural network* yang telah dibangun pada penelitian ini adalah sebagai berikut:



Gambar 1 Arsitektur *Convolutional Neural Network*.

Gambar 1 merupakan arsitektur *convolutional neural network* yang dibangun untuk proses pelatihan data. Input data citra yang digunakan pada penelitian ini berukuran 64x64 *pixels*. Proses *feature learning* dilakukan sebanyak tiga kali konvolusi. Konvolusi pertama dilakukan dengan filter 3x3 dengan jumlah filter sebanyak 32 filter, *stride* 1 dan *max-pooling* dengan *stride* 2 sehingga ukuran matriks mejadi 32x32 *pixels*. Begitupun dengan konvolusi kedua dan ketiga menggunakan inisiasi yang sama sehingga ukuran matriks terkahir menjadi 8x8 *pixels*. Selanjutnya melakukan proses klasifikasi pada *fully connected layer* dengan satu *hidden layer* menggunakan fungsi aktivasi *sigmoid*. *Sigmoid* akan menerima suatu nilai bilangan real dan mengubah nilai tersebut yang memiliki *range* berkisar 0-1. Sehingga

dengan ditetapkannya $threshold = 0.5$, maka jika nilai hasil klasifikasi dari fungsi $sigmoid < 0.5$ termasuk kedalam kategori COVID-19 dan jika nilai hasil klasifikasi dari fungsi $sigmoid > 0.5$ termasuk kategori normal. Berdasarkan penjelasan arsitektur *convolutional neural network* diatas, didapat model yang terbentuk sebagai berikut:

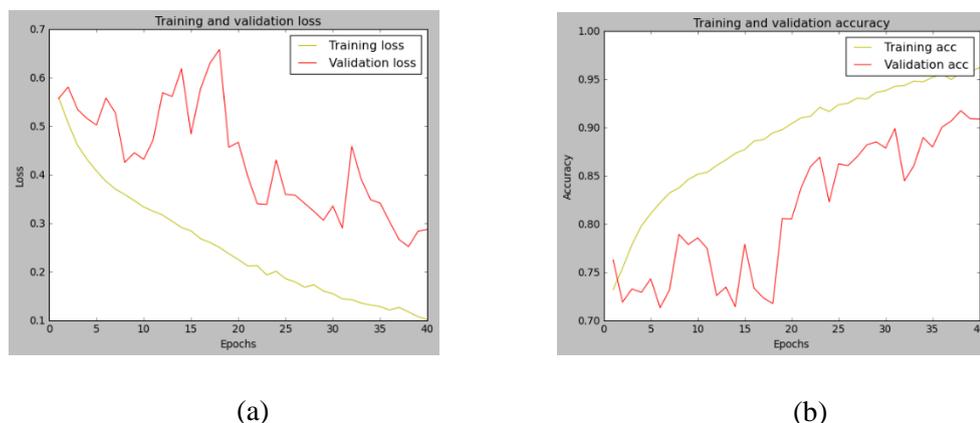
Layer (type)	Output Shape	Param #
conv2d_21 (Conv2D)	(None, 64, 64, 32)	896
activation_33 (Activation)	(None, 64, 64, 32)	0
max_pooling2d_21 (MaxPooling)	(None, 32, 32, 32)	0
conv2d_22 (Conv2D)	(None, 32, 32, 32)	9248
activation_34 (Activation)	(None, 32, 32, 32)	0
max_pooling2d_22 (MaxPooling)	(None, 16, 16, 32)	0
conv2d_23 (Conv2D)	(None, 16, 16, 32)	9248
activation_35 (Activation)	(None, 16, 16, 32)	0
max_pooling2d_23 (MaxPooling)	(None, 8, 8, 32)	0
flatten_7 (Flatten)	(None, 2048)	0
dense_14 (Dense)	(None, 64)	131136
dropout_7 (Dropout)	(None, 64)	0
dense_15 (Dense)	(None, 1)	65
activation_36 (Activation)	(None, 1)	0
Total params: 150,593		
Trainable params: 150,593		
Non-trainable params: 0		

Gambar 2 Model arsitektur *convolutional neural network*.

Gambar 2 merupakan model yang telah dibangun untuk melakukan proses *training*. Dengan menggunakan arsitektur yang sudah dibangun berdasarkan Gambar 1, menghasilkan jumlah banyaknya parameter yang terbentuk sebanyak 150.593 parameter. Parameter tersebut yang akan digunakan untuk melakukan proses *training*.

3.2. Proses Training

Proses *training* terdiri dari tahapan *feedforward* dan *backpropagation* untuk mengupdate bobot dan bias dengan ditetapkan 40 *epoch* dan nilai *learning rate* 0.1. Berikut hasil grafik dari proses *training*:



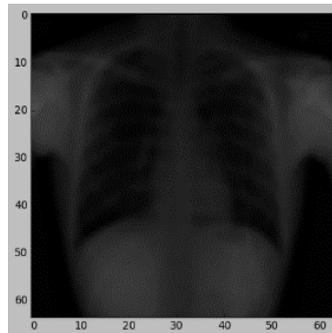
Gambar 3 Grafik dari proses *training* dengan (a) nilai *loss* dari data *training* dan *testing* dan (b) nilai akurasi dari data *training* dan *testing*.

Gambar 3 bagian (a) menunjukkan nilai *loss* dari data *training* dan *testing* dan bagian

(b) menunjukkan nilai akurasi dari data *training* dan *testing* terlihat bahwa semakin banyak jumlah *epoch* semakin banyak bobot dan bias yang berubah sehingga semakin banyak jumlah waktu yang diperlukan untuk melakukan proses *training*, selain itu terlihat bahwa nilai *loss* pada *epoch* ke-40 telah mencapai 0.1.

3.3. Proses Testing

Proses *testing* dilakukan dengan tahapan *feedforward* dengan menggunakan bobot dan bias dari hasil proses *training*. Pada proses *testing* menghasilkan hasil klasifikasi yang disajikan dengan *confusion matrix*. Output hasil klasifikasi berkisar diantara *range* 0-1 dengan nilai mendekati 0 termasuk kedalam kategori COVID-19 dan nilai mendekati 1 termasuk kedalam kategori normal. Sehingga ditetapkan *threshold* sebesar 0.5, jika nilai hasil klasifikasi dari fungsi *sigmoid* $<$ *threshold* termasuk kedalam kategori COVID-19 dan jika nilai hasil klasifikasi dari fungsi *sigmoid* $>$ *threshold* termasuk kategori normal. Sebagai contoh akan dilakukan proses *testing* menggunakan satu data *testing*:



Gambar 4 Data testing 64x64 pixels.

Berdasarkan Gambar 4, setelah dilakukan proses *testing* dengan tahapan *feedforward* diperoleh nilai prediksi sebesar 0.98 sedangkan nilai aktual nya adalah 1 atau normal. Dengan nilai hasil klasifikasi $>$ 0.5 maka data citra tersebut termasuk kedalam kategori normal dan termasuk sebagai representasi dari *true negative* (TN). Proses *testing* tersebut dilanjutkan sampai data *testing* terakhir dan disimpan kedalam *confusion matrix*. Berikut hasil *confusion matrix* menggunakan data *testing*:

Tabel 2 Confusion matrix

		Nilai Aktual	
		COVID-19	Normal
Nilai Prediksi	COVID-19	571	96
	Normal	156	1939

Tabel 2 menunjukkan hasil dari proses *testing* berupa *confusion matrix*. Prediksi terhadap kategori COVID-19 yang diklasifikasikan dengan benar sebagai COVID-19 ada sebanyak 571. Prediksi terhadap kategori normal yang diklasifikasikan dengan benar sebagai normal ada sebanyak 1939. Prediksi terhadap kategori normal yang diklasifikasikan sebagai COVID-19 ada sebanyak 96. Dan prediksi terhadap kategori COVID-19 yang

diklasifikasikan sebagai normal ada sebanyak 156. Dari hasil *confusion matrix* dapat dihitung nilai akurasi, *presisi* dan *recall*. Berikut hasil akurasi, *presisi* dan *recall*:

$$\text{Akurasi} = \frac{571 + 1939}{571 + 1939 + 96 + 156} \times 100\% = 91\%$$

$$\text{Presisi} = \frac{571}{571 + 96} \times 100\% = 86\%$$

$$\text{Recall} = \frac{571}{571 + 156} \times 100\% = 79\%$$

Berdasarkan perhitungan diatas dengan arsitektur *convolutional neural network* yang telah dibangun diperoleh nilai akurasi dari data *testing* sebesar 91%, *presisi* 86% dan *recall* 79%. Nilai akurasi sebesar 91% menandakan model sangat baik dalam melakukan klasifikasi (*excellent classification*). Nilai *presisi* sebesar 86% menandakan persentase pasien yang benar menderita COVID-19 dari keseluruhan pasien yang diprediksi menderita COVID-19. Nilai *recall* sebesar 79% menandakan persentase pasien yang diprediksi COVID-19 dibandingkan keseluruhan pasien yang sebenarnya COVID-19.

4. SIMPULAN

Berdasarkan hasil analisis pada arsitektur *convolutional neural network* yang telah dibangun dengan input citra 64x64 *pixels*, jumlah *epoch* sebanyak 40, nilai *loss* 0.1 dan nilai *learning rate* 0.1, model dapat memprediksi 571 citra *x-rays* yang termasuk kategori COVID-19 dari 727 citra *x-rays* COVID-19 dan 1.939 citra *x-rays* yang termasuk kategori normal dari 2.035 citra *x-rays* normal. Diperoleh nilai akurasi sebesar 91%, *presisi* sebesar 86% dan *recall* sebesar 79%. Nilai akurasi sebesar 91% menunjukkan bahwa dengan menggunakan algoritma *convolutional neural network* dalam klasifikasi biner pasien COVID-19 menggunakan fungsi aktivasi *sigmoid* sudah sangat baik dalam melakukan klasifikasi. Dari hasil penelitian ini diharapkan dapat membantu tenaga kesehatan dalam mendiagnosis pasien yang memiliki gejala COVID-19.

5. DAFTAR PUSTAKA

- Alpayđın, E. (2020). *Introduction to Machine Learning* (4th ed.). MIT Press.
- Hasan, M., Ullah, S., Khan, M. J., & Khurshid, K. (2019). Comparative Analysis of SVM, ANN and CNN for Classifying Vegetation Species using Hyperspectral Thermal Infrared Data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(2/W13), 1861–1868. <https://doi.org/10.5194/isprs-archives-XLII-2-W13-1861-2019>
- Hasantabar, S., Ahmadi, M., & Sharifi, A. (2020). Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network approaches. *Chaos, Solitons and Fractals*, 140, 110170. <https://doi.org/10.1016/j.chaos.2020.110170>
- Kabir Anaraki, A., Ayati, M., & Kazemi, F. (2018). Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. *Biocybernetics and Biomedical Engineering*, 39(1), 63–74. <https://doi.org/10.1016/j.bbe.2018.10.004>

- Krizhevsky, A., Sutskeve, I., & Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. 60, 84–90. <https://doi.org/10.1145/3383972.3383975>
- Setiawan, W. (2020). *Deep Learning Menggunakan Convolutional Neural Network*. MNC.
- Srinivasan, N., Ravichandran, V., Chan, K. L., Vidhya, J. R., Ramakrishnan, S., & Krishnan, S. M. (2002). Exponentiated backpropagation algorithm for multilayer feedforward neural networks. *ICONIP 2002 - Proceedings of the 9th International Conference on Neural Information Processing: Computational Intelligence for the E-Age, I*(December), 327–331. <https://doi.org/10.1109/ICONIP.2002.1202187>