



Jenis Artikel: *orginal research*

Desain Algoritma Jaringan Syaraf Tiruan (JST) untuk Identifikasi Covid-19 dan Pneumonia

Tri Mukti Lestari¹, Nurhasanah Nurhasanah¹, Dwiria Wahyuni¹, Hasanuddin Hasanuddin¹, Radhitya Perdhana¹

¹Jurusan Fisika, Fakultas MIPA, Universitas Tanjungpura

Corresponding e-mail: nurhasanah@physics.untan.ac.id

KATA KUNCI: Covid-19, pneumonia, jaringan syaraf tiruan, propagasi balik.

Diserahkan: 4 Desember 2022

Diterima: 15 Desember 2022

Diterbitkan: 31 Januari 2023

Terbitan daring: 31 Januari 2023

ABSTRAK. Kementerian Kesehatan Republik Indonesia menyatakan bahwa penyakit Covid-19 merupakan kasus global yang menyerang lebih dari seratus juta jiwa pada tahun 2021. Sedangkan, pneumonia menjadi penyebab utama kematian pada bayi dan balita di Indonesia. Penyakit yang disebabkan oleh virus ini membutuhkan penanganan yang cepat dan tepat untuk meminimalisir penyebaran. Penyakit ini dapat diidentifikasi dengan analisis gejala, tes PCR, atau pengenalan citra radiologi. Adanya kesamaan gejala serta waktu dan biaya yang tidak sedikit untuk tes PCR, menyebabkan identifikasi penyakit terhambat. Berdasarkan hal tersebut, dibangunlah suatu sistem pendeteksi penyakit Covid-19 dan pneumonia menggunakan Jaringan Syaraf Tiruan (JST) metode propagasi balik. JST yang dibentuk terdiri atas lapisan masukan yang merupakan data numerik fitur citra paru pasien normal, penderita Covid-19 dan pneumonia, lapisan tersembunyi berisi kumpulan proses identifikasi, serta lapisan luaran yang berisi luaran yang diharapkan yaitu jenis paru normal, penderita Covid-19, dan pneumonia. Setelah serangkaian pelatihan untuk sistem mempelajari data, didapatkan hasil yang paling baik yaitu dengan menggunakan kombinasi pembelajaran dengan fungsi aktivasi tanh, *learning rate* 0,001 dan *epoch* 100000 kali. Dari hasil pengujian sistem, didapatkan akurasi dari 66,677% sampai

1. Pendahuluan

COVID-19 dan pneumonia merupakan penyakit yang menyerang organ paru-paru manusia. Covid-19 yang disebabkan oleh virus korona, dan pneumonia yang disebabkan virus influenza mengakibatkan kerusakan pada paru-paru. Kerusakan ini mengakibatkan gangguan pernapasan hingga dapat menyebabkan kematian (Kemenkes, 2021). Penanganan penyakit Covid-19 dan pneumonia dengan langkah yang tepat dapat mengurangi

dampak yang ditimbulkan. Deteksi penyakit ini dapat dilakukan dengan menggabungkan hasil pemeriksaan laboratorium dan hasil pengenalan fitur citra paru pasien. Hasil laboratorium didapatkan melalui tes *Polymerase Chain Reaction* (PCR) untuk mengetahui keberadaan virus di dalam tubuh pasien. Sedangkan, pengenalan fitur citra didapatkan melalui pengolahan data pada citra rontgen paru-paru, untuk mengenali pola kerusakan pada paru-paru yang disebabkan oleh virus (Zu dkk, 2020).

Pengenalan fitur citra penyakit Covid-19 dan pneumonia dapat dilakukan dengan Jaringan Syaraf Tiruan (JST). JST bekerja dengan meniru cara kerja manusia, yaitu mempelajari pola-pola yang sudah ada sebelumnya, mengenali, dan membandingkannya dengan data baru yang belum diketahui. JST mampu memberikan hasil yang sesuai pada data yang belum diketahui (Estomihi, 2019). JST dengan metode propagasi balik (*backpropagation*) mampu menyelesaikan masalah pengenalan dengan pola yang lebih kompleks (Jaya dkk, 2018).

JST propagasi balik sudah banyak digunakan pada penelitian sebelumnya. Pada tahun 2019 dilakukan penelitian oleh Estomihi, untuk mengklasifikasikan paru normal, pneumonia bakteri, dan pneumonia virus pada citra *X-ray* dada menggunakan algoritma JST dengan metode propagasi balik (Estomihi, 2019). Adapun pada tahun 2021, JST propagasi balik juga digunakan oleh Yuhandri dan Putra untuk mengidentifikasi Covid-19 dengan menggunakan citra *X-ray* bagian dada (Putra & Yuhandri, 2021). Dari hasil penelitian yang telah dilakukan, didapatkan hasil bahwa JST dengan metode propagasi balik dapat digunakan untuk mengenali suatu penyakit melalui pengenalan pada pola citra *X-ray* pasien.

Penelitian ini menggunakan JST propagasi balik untuk membedakan paru normal, Covid-19, dan pneumonia dengan mempelajari pola citra *X-ray* dada. Penggunaan metode ini diharapkan dapat mempersingkat waktu diagnosis dan memberikan hasil yang akurat dalam upaya pengenalan citra untuk mendeteksi suatu penyakit.

2. Metodologi

2.1 Pengolahan Citra

Pengolahan citra merupakan kumpulan metode untuk memanipulasi atau memproses citra yang dilakukan pada citra digital 2 dimensi (Vincentia dkk, 2019). Proses ini dimulai dengan *grayscale* untuk mengubah citra *Red-Green-Blue* (RGB) menjadi citra *grayscale*. Kemudian, citra *grayscale* dilanjutkan dengan *Contrast Limited Adaptif Histogram Equalization* (CLAHE), untuk memperbaiki persebaran nilai kontras citra (Jaya dkk, 2018). Selanjutnya, citra disegmentasi dengan menggunakan metode *threshold* untuk membedakan antara objek dan latar belakang pada suatu citra. Pada proses ini, akan dihasilkan luaran berupa citra biner dengan dua tingkat keabuan, yaitu hitam dan putih.

2.2 Ekstraksi Citra

Ekstraksi citra menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). Metode ini digunakan untuk menentukan hubungan tekstur antar piksel satu dengan piksel lainnya. Metode GLCM dapat mengetahui nilai kontras, homogenitas, dissimilaritas, energi dan korelasi dari suatu citra dalam 4 arah sudut bertetangga, yaitu 0°, 45°, 90°, dan 135° (Amalia, 2018). Untuk mendapatkan nilai fitur citra, digunakan persamaan (1) sampai (5) (Jaya dkk, 2018).

$$\text{Energi} = \sum_i \sum_j p^2(i, j) \quad (1)$$

$$\text{Kontras} = \sum_i \sum_j (i - j)^2 p(i, j) \quad (2)$$

$$\text{Korelasi} = \frac{1}{\sigma_x \sigma_y} \sum_{i=1} \sum_{j=1} (i - \mu_x)(j - \mu_y) p(i, j) \quad (3)$$

$$\text{Dissimilaritas} = \frac{1}{\sigma_x \sigma_y} \sum_{i=1} \sum_{j=1} (i - j) p(i, j) \quad (4)$$

$$\text{Homogenitas} = \sum_i \sum_j \frac{p(i, j)}{1 + |i - j|} \quad (5)$$

2.3 Deteksi Citra

Pengenalan pola citra dilanjutkan dengan tahapan uji coba pada sampel citra. Metode propagasi balik mempunyai tiga fase yang akan terus berulang sampai kondisi untuk berhenti terpenuhi selama masa pelatihan pengenalan citra. Ketiga fase ini yaitu propagasi maju, propagasi mundur, dan fase perbaikan bobot (Jaya dkk, 2018).

Tahap I: Propagasi Alur Maju

1. Semua bobot dan bias diinisialisasi dengan bilangan acak. Bobot dan bias dari lapisan masukan ke lapisan tersembunyi serta bobot dan bias dari lapisan tersembunyi ke lapisan luaran dengan *learning rate sebesar 0,001*. Hasil pada unit tersembunyi (Z_j) ($j=1,2,\dots,p$) dihitung menggunakan persamaan (6). j adalah jumlah unit pada lapisan tersembunyi dan p menunjukkan nomor unit. V_{oj} adalah nilai bias dan x_i adalah unit masukan.

$$Z_j = v_{oj} + \sum_{i=1}^p x_i v_{ij} \quad (6)$$

2. Hasil pada unit luaran y_k ($k=1,2,\dots,m$) dihitung menggunakan persamaan (7):

$$Y_{in\ k} = W_{ok} + \sum_{j=1}^m Z_j W_{jk} \quad (7)$$

$Y_{in\ k}$ adalah hasil dari unit luaran dan Y_{out} adalah fungsi aktivasi. W_{jk} adalah nilai bobot pada lapisan luaran. k adalah jumlah unit pada lapisan luaran, m menunjukkan unit pada lapisan luaran, serta α adalah *learning rate* yang digunakan.

Tahap II: Propagasi Alur Mundur

3. Setiap unit luaran Y_{out} ($k=1,2,\dots,m$) akan menerima hasil yang berkaitan dengan pola masukan pelatihan, kemudian informasi *error*-nya dihitung dengan persamaan (8):

$$\delta_k = (t_k - Y_{in\ k}) \quad (8)$$

δ_k adalah *error* yang akan digunakan pada proses perbaikan bobot lapisan selanjutnya. t_k adalah target yang harus dicapai pada lapisan luaran. Kemudian koreksi bobot W dengan *learning rate* α ($k=1,2,3,\dots,m$; $j=0,1,2,\dots,p$) dihitung menggunakan persamaan (9):

$$\Delta W_{jk} = \alpha \delta_k z_j \quad (9)$$

4. Faktor error δ pada unit tersembunyi dihitung berdasarkan kesalahan pada unit tersembunyi z_j ($j=1,2,\dots,p$) menggunakan persamaan (10):

$$\delta_{in\ j} = \sum_{k=1}^m \delta_k W_{jk} \quad (10)$$

5. Untuk menghitung suku perubahan bobot W_{ji} yang akan digunakan dalam perbaikan V_{ji} ($j=1,2,3,\dots,p$; $i=0,1,2,\dots,n$) digunakan persamaan (11) dan (12):

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (11)$$

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (12)$$

Tahap III: Perubahan bobot

6. Seluruh perubahan bobot yang menuju unit luaran ($k=1,2,3,\dots,m$; $j=0,1,2,\dots,p$) dihitung menggunakan persamaan (13):

$$W(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (13)$$

Perubahan bias yang menuju unit tersembunyi ($j=1,2,3,\dots,p$; $i=0,1,2,\dots,n$) dihitung menggunakan persamaan (14):

$$V(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (14)$$

7. Nilai kesalahan dihitung dengan MSE (*Mean Squared Error*) menggunakan persamaan (15):

$$MSE = \frac{1}{n_{\text{Pola}}} \sum_k^{n_{\text{Pola}}} (t_k - y_{\text{out}})^2 \quad (15)$$

3. Analisis Sistem dan Algoritma

3.1 Analisis Sistem

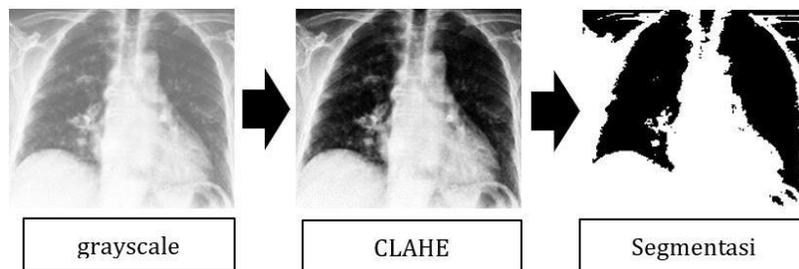
Penelitian ini dilakukan dengan perangkat keras dan perangkat lunak. Perangkat keras yang digunakan yaitu komputer dengan *processor intel® core™ i7-700* dan CPU 3.60 GHz dengan memori (RAM) 8.00 GB. Perangkat lunak yang digunakan yaitu aplikasi *Microsoft Paint* untuk proses *cropping* dan *resizing* citra, aplikasi *ImageJ* versi 1.8.0 untuk proses *grayscale*, CLAHE, dan segmentasi citra, serta *Anaconda3 Jupyter Notebook* versi 3.8 dengan bahasa pemrograman *Python* versi 3.8 yang digunakan untuk proses ekstraksi dan identifikasi citra.

Masukan awal sistem berupa representasi dari kelainan atau kerusakan pada paru-paru akibat terserang virus. Sedangkan, luaran akhir sistem berupa representasi teridentifikasi paru normal, penderita covid-19, dan pneumonia serta hasil akurasi pengujian menggunakan JST propagasi balik.

3.2 Analisis Input

Data citra berjumlah total 120 citra berupa citra *X-ray* bagian dada atau toraks, terdiri atas 40 citra Covid-19 (Cohen dkk, 2020), 40 citra normal, dan 40 citra pneumonia (Wang dkk, 2017), yang berasal dari *international open database collection*. Data dengan format .jpg berukuran 512×512 piksel didapat dari situs publik <https://github.com/muhammedtalo/COVID-19>. Data yang telah terkumpul selanjutnya dikelompokkan berdasarkan jenis penyakit pada citra tersebut. Citra dikelompokkan ke dalam 3 dataset sesuai jenis citra, yaitu citra normal, covid-19, dan pneumonia.

Data yang akan diproses menggunakan JST propagasi balik adalah berupa data numerik. Sehingga, data citra diolah terlebih dahulu untuk mendapatkan hasil berupa nilai fitur citra tersebut. Pengolahan tersebut dimulai dengan memperbaiki citra lalu mengubahnya ke bentuk citra biner, kemudian diekstraksi untuk mendapatkan hasil nilai fitur pada citra. Proses perbaikan citra dilakukan menggunakan aplikasi *imageJ*. Aplikasi ini dapat memproses *grayscale* untuk mengubah citra *Red-green-Blue* (RGB) ke bentuk *greyscale*, hingga *threshold* untuk mengubah citra *greyscale* ke bentuk biner. Gambar 1 menunjukkan proses perubahan bentuk citra setelah melalui proses perbaikan citra.



Gambar 1 citra setelah proses *grayscale*, CLAHE, dan segmentasi.

Tahap ekstraksi fitur dilakukan dengan metode *Grey-Level Co-occurrence Matrix* (GLCM) menggunakan *Jupyter Notebook* dan *Python*. Melalui metode GLCM, akan didapatkan nilai fitur berupa nilai kontras, homogenitas, energi, korelasi, dan dissimilaritas. Ekstraksi dilakukan dengan empat arah sudut bertetangga yaitu 0° , 45° , 90° , dan 135° [Amalia dkk, 2018]. Ekstraksi citra dengan GLCM menggunakan *library greycoprops* dan *greycomatrix*.

Kedua *library* ini dapat mengolah citra untuk mendapatkan nilai setiap fiturnya. Algoritma GLCM disajikan pada Tabel 1.

Tabel 1. Algoritma GLCM

Syntax	Penjabaran
<pre>matrix_cooccurrence = greymatrix(inds, [1], [0, np.pi/4, np.pi/2, 3*np.pi/4], levels=max_value, normed=False, symmetric=False)</pre>	Metode GLCM menggunakan <i>library greymatrix</i> .
<pre>class data: def contrast_feature(matrix_cooccurrence): contrast = greycoprops (matrix_cooccurrence, 'contrast') return "contrast = ", contrast def dissimilarity_feature(matrix_cooccurrence): dissimilarity = greycoprops (matrix_cooccurrence, 'dissimilarity') return "dissimilarity = ", dissimilarity def homogeneity_feature(matrix_cooccurrence): homogeneity = greycoprops (matrix_cooccurrence, 'homogeneity') return "homogeneity = ", homogeneity def energy_feature(matrix_cooccurrence): energy = greycoprops (matrix_cooccurrence, 'energy') return "energy = ", energy def correlation_feature(matrix_cooccurrence): correlation = greycoprops (matrix_cooccurrence, 'correlation') return "correlation = ", correlation</pre>	Ekstraksi nilai fitur GLCM menggunakan <i>library greycoprops</i> .

Ekstraksi citra dengan menggunakan algoritma GLCM akan menghasilkan luaran berupa nilai fitur, salah satunya adalah seperti ditunjukkan pada Tabel 2.

Tabel 2. Luaran GLCM pada Salah Satu Sampel

Fitur	Tes awal
<i>contrast</i>	(([[3.99422853, 5.48097242, 4.18680054, 5.71523164]]))
<i>dissimilarity</i>	(([[0.30214958, 0.38950908, 0.31642536, 0.40429916]]))
<i>homogeneity</i>	(([[0.95943788, 0.957589, 0.95763361, 0.95674694]]))
<i>energy</i>	(([[0.66924478, 0.66960021, 0.66809271, 0.66919581]]))
<i>correlation</i>	(([[0.9663891, 0.9538738, 0.96475727, 0.95190711]]))

Setiap sampel yang diekstraksi mempunyai masing-masing 4 nilai pada setiap fitur. Keempat nilai tersebut menunjukkan nilai pada sudut berturut-turut 0°, 45°, 90°, dan 135°. Setelah seluruh citra diekstraksi, data dari seluruh nilai fitur citra kemudian dirata-ratakan berdasarkan nilai fitur dan sudutnya. Nilai fitur yang dirata-ratakan akan memiliki nilai berupa nilai fitur setiap jenis citra, yaitu nilai kontras, dissimilaritas, homogenitas, energi, dan korelasi pada citra normal, Covid-19, dan pneumonia. Citra normal memiliki nilai homogenitas, energi, dan korelasi yang lebih tinggi dibandingkan citra pneumonia dan Covid-19. Sedangkan, citra Covid-19 memiliki nilai kontras dan dissimilaritas yang lebih tinggi disusul oleh citra pneumonia dan citra normal. Nilai korelasi, energi, serta homogenitas yang tinggi menunjukkan keadaan citra yang homogen. Sedangkan, nilai kontras yang rendah menandakan adanya gangguan pada tekstur citra (Vincentia dkk, 2019).

3.3 Analisis Target Luaran JST

Target luaran akhir pada sistem JST merupakan representasi dari 3 jenis citra paru, yaitu citra normal, Covid-19, dan pneumonia. Sistem akan mengidentifikasi jenis citra berdasarkan data pelatihan yang berjumlah 75% dari total data dengan target luaran seperti ditunjukkan pada Tabel 3.

Tabel 3. Target Luaran JST yang Diharapkan

Jenis Citra	Luaran
Normal	1 1 1
Covid-19	1 0 1
Pneumonia	1 1 0

Hasil luaran berupa angka 0 dan 1. Angka 0 menunjukkan hasil data sistem tidak dapat mengenali citra, sedangkan angka 1 menunjukkan hasil data sistem dapat mengenali citra dengan baik. Setiap angka luaran akan mewakili satu citra hingga total citra yang diuji sebanyak 25% dari total data. Semakin banyak angka 1 yang muncul, akan menunjukkan hasil akurasi sistem JST yang baik.

3.4 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) dengan metode propagasi balik dijalankan menggunakan *Jupyter Notebook* dengan bahasa pemrograman *Python*. Pada JST, terdapat dua proses pengolahan data, yaitu tahap pelatihan dan tahap pengujian. Proses pelatihan menggunakan 20 lapisan masukan, 20 lapisan tersembunyi, dan 3 luaran. Setiap data masukan diberi label berupa nama fitur dan sudutnya, serta jenis citranya sebagai data target selama pelatihan. Algoritma dari JST ditunjukkan pada Tabel 4.

Tabel 4. Algoritma JST

Syntax	Penjabaran
<pre>import pandas as pd import numpy as np my_cols = ["contras0", "contras45", "contras90", "contras135", "dissimilarity0", "dissimilarity45", "dissimilarity90", "dissimilarity135", "homogeneity0", "homogeneity45", "homogeneity90", "homogeneity135", "energy0", "energy45", "energy90", "energy135", "correlation0", "correlation45", "correlation90", "correlation135", "target"]</pre>	Pemberian label data berupa nama fitur dan sudutnya.
<pre>x=my_df.iloc[:,0:20].values y=my_df.iloc[:,21].values</pre>	Pemisahan antara data fitur sebagai data masukan yaitu 'x' dan data target sebagai luaran yaitu 'y'.
<pre>X, y = make_classification(n_samples=120) X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, train_size=0.75)</pre>	Data dibagi ke dalam data latih dan data uji. Data latih menggunakan 75% dari total seluruh data yaitu sebanyak 90 data, dan 25% sebagai data uji yaitu 30 data.
<pre>clf = MLPClassifier(random_state=0, hidden_layer_sizes=(20), activation = 'tanh', solver = 'adam', learning_rate_init=0.001, max_iter=10000).fit(X_train, y_train)</pre>	Selama pelatihan, digunakan fungsi aktivasi tanh, <i>learning rate</i> 0,001 dan iterasi maksimum sebanyak 10.000 kali. Setelah menetapkan parameter, dilakukan pelatihan pada data latih, yaitu 'x_train', dan 'y_train'.

Setelah diimplementasikan, citra yang diuji akan dikenali oleh sistem JST dan dimasukkan ke dalam salah satu dari kelas Covid-19, pneumonia, atau normal. Setelah menyelesaikan pelatihan, sistem dapat dilanjutkan untuk melakukan pengujian terhadap data uji. Selama pelatihan, dilakukan pengujian akurasi sistem JST terhadap 30 data uji menggunakan variasi *learning rate* 0,05, 0,09, 0,001, 0,005 dan 0,009 serta variasi iterasi maksimum sebanyak 1000, 5000, dan 10000 kali. Hasil akurasi pengujian dengan variasi *learning rate* dan iterasi maksimum terhadap data uji ditunjukkan pada Tabel 5.

Tabel 5. Hasil Akurasi Pengujian dengan Variasi *Learning Rate* dan Iterasi Maksimum terhadap Data Uji

Variasi <i>Learning Rate</i> (α)		Variasi Iterasi Maksimum	
Masukan	Akurasi (%)	Masukan	Akurasi (%)
0,1	<i>error</i>	1000	65
0,05	82,3	5000	74
0,09	88,8	10000	90
0,001	90,3	100000	92
0,005	82,3		
0,009	88,8		

Berdasarkan Tabel 5, akurasi terbaik didapatkan pada variasi *learning rate* 0,001 dan iterasi maksimum 100000 kali. Pemilihan parameter *learning rate* dan iterasi maksimal berpengaruh pada hasil yang kurang optimal. Nilai *learning rate* yang terlalu besar akan mempercepat waktu pembelajaran, akan tetapi akan berpengaruh pada ketepatan identifikasi serta dapat menyebabkan *error* pada sistem JST. Namun, nilai *learning rate* yang terlalu kecil juga dapat mempengaruhi hasil identifikasi. Demikian juga dengan parameter iterasi maksimum yang digunakan, jika terlalu kecil akan menyebabkan akurasi yang kurang optimal. Jumlah iterasi yang terlalu banyak dapat meningkatkan sedikit akurasi, namun akan membutuhkan waktu yang cukup lama (Pangestu dan Widiastuti, 2014).

4. Pengujian

Pengujian sistem dilakukan pada data uji sebanyak 30 data citra. Pada tahap pengujian, sistem melakukan proses identifikasi berdasarkan hasil data pada proses pelatihan sebelumnya. Pengujian pada sistem dilakukan untuk mengetahui apakah sistem JST mampu mengidentifikasi data citra baru melalui data yang telah dipelajari sebelumnya. Pengujian ini dilakukan untuk mengetahui akurasi pengujian dan sebagai pendukung bahwa parameter yang dipilih merupakan kombinasi dari variasi yang sesuai untuk sistem JST untuk mengidentifikasi penyakit Covid-19 dan pneumonia. *Syntax* untuk pengujian JST ditunjukkan pada Tabel 6.

Tabel 6. Algoritma JST

Syntax	Penjabaran
<pre>clf.predict_proba(X_test[:30]) clf.predict(X_test[:3, :]) clf.score(X_test, y_test)</pre>	Pengujian sistem JST terhadap data uji dengan 3 jenis luaran.
<pre>Out[:]: [1 1 0] 93,3333</pre>	Hasil luaran dari proses pengujian. Angka 0 menunjukkan citra tidak teridentifikasi, dan angka 1 menunjukkan citra dapat teridentifikasi. Hasil akhir berupa nilai akurasi pengujian.

Data uji sebanyak 25% dari total data dipilih secara acak dan otomatis oleh sistem. Pemilihan data acak oleh sistem memungkinkan untuk mendapatkan hasil akurasi yang berbeda, sehingga pada penelitian ini dilakukan 10 kali pengujian untuk melihat perbandingan hasil dari pengujian data uji terhadap data target serta akurasi yang dihasilkan. Hasil akurasi dari 10 kali pengulangan pengujian data uji terhadap data target ditunjukkan pada Tabel 7.

Tabel 7. Hasil Akurasi Pengujian Data Uji terhadap Data Target

Pengujian ke-	Covid-19	Pneumonia	Normal	Akurasi (%)
1	1	1	0	83,34
2	1	1	0	93,34
3	0	1	1	90
4	1	1	1	96,67
5	0	1	1	83,34
6	0	1	1	90
7	0	1	0	66,67
8	1	0	0	66,67
9	1	1	0	83,34
10	0	1	1	90

Berdasarkan pengujian yang telah dilakukan, tingkat akurasi yang didapat dalam upaya pengelompokan citra normal, Covid-19, dan pneumonia dengan menggunakan JST propagasi balik pada penelitian ini berada dalam rentang nilai data pengujian dengan akurasi 66,67% sampai 96,67%. Hal ini menunjukkan bahwa pemanfaatan JST dengan propagasi balik dapat diterapkan untuk mengenali citra toraks pasien normal, Covid-19, dan pneumonia dengan akurat.

5. Kesimpulan

Jaringan syaraf tiruan dengan metode propagasi balik berhasil digunakan dalam pengenalan citra toraks pasien normal, Covid-19, dan pneumonia. Proses pelatihan dilakukan dengan *learning rate* 0,001 dan iterasi maksimal 10000 kali yang memberikan akurasi optimal. Dari data pengujian pada penelitian ini didapatkan nilai rentang akurasi sebesar 66,67% sampai dengan 96,67%.

Keterlibatan Penulis

TML melakukan analisis data dan menulis manuskrip awal. N memberi gagasan pokok penelitian dan merancang alur penelitian, DW merancang alur penelitian dan menulis manuskrip revisi. H dan RP melakukan revisi terhadap analisis data.

Daftar Pustaka

- Amalia, I. 2018. Ekstraksi Fitur Citra Songket Berdasarkan Tekstur Menggunakan Metode *Gray Level Co-occurrence Matrix (GLCM)*. *Jurnal Infomedia*, Vol. 3(2), 64–68.
- Estomih, J. 2019. Klasifikasi Penyakit Pneumonia Dari Citra X-Ray Menggunakan Backpropagation. 1–74.
- Jaya, H., Sabran, D., Djawad, Y. A., Ilham, A., Ahmar, A. 2018. Kecerdasan Buatan. *Journal of Chemical Information and Modeling*, Vol. 53(9).
- Kemenkes, & Gernas. 2021. Situasi Terkini Perkembangan Novel Coronavirus (COVID-19).
- Ozturk, T., Talo, M., Yildirim, E.A., Baloglu, U.B., Yildirim, O., & Acharya, U.R. 2020. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in Biology and Medicine*. Vol. 121
- Cohen, J.P., Morrison, P., & Dao. 2020, COVID-19 Image Data Collection, arXiv preprint arXiv:2003.11597
- Pangestu, D. P., & Widiastuti, N. I. 2014. Pembangunan Sistem Pendeteksi Penyalahgunaan Narkoba Menggunakan Algoritma Jaringan Syaraf Tiruan Metode Backpropagation. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, Vol. 3(1), 27-34.
- Putra, H. R. W. & Yuhandri, Y. 2021. Identifikasi Penderita COVID-19 Berdasarkan Chest X-Ray Menggunakan Algoritma Jaringan Syaraf Tiruan Backpropagation. *Jurnal Sistem Informasi dan Teknologi*, Vol. 3, 197-202.
- Vincentia, V., Nurhasanah, & Sanubary, I. 2019. Deteksi Awal Retinopati Hipertensi Menggunakan Jaringan Syaraf Tiruan pada Citra Fundus Mata. *Jurnal Fisika*, Vol. 9(1), 9–20.

- Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R.M., 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016), pp. 770-778
- Zu, Z. Y., Jiang, M. D., Xu, P. P., Chen, W., Ni, Q. Q., Lu, G. M., Zhang, L. J. 2020. *Coronavirus Disease 2019 (COVID-19): A Perspective From China*. *radiology.rsna.org*. 296:E15-E25.