

**KLASIFIKASI PENYAKIT *PNEUMONIA*  
MENGGUNAKAN METODE  
*CONVOLUTION NEURAL  
NETWORK (CNN)***

**Oleh**  
**THIA RISKIYANI H. MOSI**  
**T3121008**

**SKRIPSI**

**Untuk memenuhi salah satu syarat ujian  
guna memperoleh gelar Sarjana**



**PROGRAM SARJANA  
FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI TEKNIK INFORMATIKA  
UNIVERSITAS ICHSAN GORONTALO  
GORONTALO  
2025**

## PERSETUJUAN SKRIPSI

### KLASIFIKASI PENYAKIT *PNEUMONIA* MENGGUNAKAN METODE *CONVOLUTION NEURAL NETWORK (CNN)*

Oleh

THIA RISKIYANI H. MOSI

T3121008

#### SKRIPSI

Untuk memenuhi salah satu syarat ujian  
Guna memperoleh gelar Sarjana  
Program Studi Teknik Informatika  
Ini telah disetujui oleh Tim Pembimbing

Gorontalo, Januari 2025

Pembimbing I

Pembimbing II

Yasin Arif Mustafa, M.Kom

NIDN.0926088503

Sudirman Melangi, M.Kom

NIDN.0908017702

PENGESAHAN SKRIPSI

**KLASIFIKASI PENYAKIT PNEUMONIA  
MENGGUNAKAN METODE  
CONVOLUTION NEURAL  
NETWORK (CNN)**

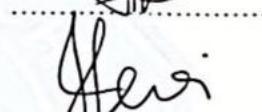
Oleh  
THIARISKIYANI H. MOSI  
T3121008

Diperiksa oleh Panitia ujian Strata Satu (S1)  
Universitas Ichsan Gorontalo

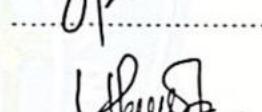
1. Ketua Penguji  
Irma Surya Kumala Idris, M.Kom



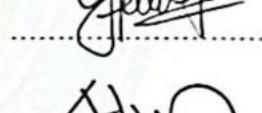
2. Anggota  
Hastuti Dalai, M.Kom



3. Anggota  
Yulianty Lasena, M.Kom



4. Anggota  
Yasin Aril Mustofa, M.Kom



5. Anggota  
Sudirman Melangi, M.Kom



Mengetahui



## PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa :

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan dan penelitian saya sendiri tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah di publikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar, yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 20 Februari 2025

Yang Membuat Pernyataan,



Thia Riskiyani H. Mosi

## ABSTRACT

### **THIA RISKIYANI H. MOSI. T3121008. THE CLASSIFICATION OF PNEUMONIA USING THE CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD**

*Pneumonia is one of the leading causes of death in Indonesia, making early diagnosis crucial for effective treatment. This research aims to develop a pneumonia classification model based on lung X-ray images, utilizing the Convolutional Neural Network (CNN) method. The methodology involves using a dataset of lung X-ray images obtained from Kaggle, which includes two classes: pneumonia and normal. The research process consists of data collection, image pre-processing, CNN model training, and model evaluation using a confusion matrix. The results indicate that the developed CNN model can classify lung X-ray images with an accuracy of 96%. Further evaluation reveals a high level of precision and recall, demonstrating the model's strong capability to accurately identify pneumonia cases. Additionally, the confusion matrix shows that the model performs well in distinguishing between the two classes, with minimal errors. This research contributes to the field of health technology, particularly in the application of CNNs for pneumonia classification. The findings highlight that CNN is an effective method for enhancing the speed and accuracy of pneumonia diagnosis through X-ray image analysis. Future studies with larger and more diverse datasets could further improve the performance of this model.*

*Keywords: Pneumonia, Convolutional Neural Network*



## ABSTRAK

### THIA RISKIYANI H. MOSI, T3121008, KLASIFIKASI PENYAKIT PNEUMONIA MENGGUNAKAN METODE CONVOLUTION NEURAL NETWORK (CNN)

Penyakit *pneumonia* merupakan salah satu penyebab utama kematian di Indonesia dan diagnosis dini sangat penting untuk penanganan yang efektif. Penelitian ini bertujuan untuk mengembangkan model klasifikasi *pneumonia* menggunakan metode Convolution Neural Network (CNN) berbasis citra X-ray paru-paru. Metodologi penelitian ini melibatkan penggunaan dataset citra rontgen paru-paru yang diambil dari Kaggle, yang terdiri dari dua kelas yaitu *pneumonia* dan normal. Tahapan yang dilakukan meliputi pengumpulan data, pre-processing citra, pelatihan model CNN, serta evaluasi model menggunakan confusion matrix. Hasil penelitian menunjukkan bahwa model CNN yang dibangun mampu mengklasifikasikan citra X-ray paru-paru dengan akurasi mencapai 96%. Evaluasi lebih lanjut menunjukkan bahwa model ini memiliki tingkat presisi dan recall yang tinggi, yang menunjukkan kemampuannya dalam mengenali *pneumonia* dengan baik. Selain itu, confusion matrix menunjukkan bahwa model ini memiliki performa yang baik dalam mengidentifikasi kedua kelas dengan minim kesalahan. Penelitian ini memberikan kontribusi dalam bidang teknologi kesehatan, khususnya dalam pemanfaatan CNN untuk klasifikasi penyakit *pneumonia*. Temuan ini menunjukkan bahwa CNN adalah metode yang efektif untuk meningkatkan kecepatan dan akurasi diagnosis *pneumonia* melalui analisis citra X-ray. Ke depannya, penelitian lebih lanjut dengan dataset yang lebih besar dan beragam dapat lebih meningkatkan performa model ini.

Kata kunci : *Pneumonia, Convolution Neural Network*



## KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan penelitian ini dengan judul **“Klasifikasi Penyakit *Pneumonia* Menggunakan Metode *Convolution Neural Network (CNN)*”** untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa skripsi ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo.
2. Ibu Dr. Hj. Juriko Abdussamad, M.Si, selaku Rektor Universitas Ichsan Gorontalo.
3. Bapak Irvan Abraham Salihi, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
4. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo, sekaligus sebagai pembimbing pendamping yang telah membimbing penulis selama penyusunan penelitian ini.
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Pembantu Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
6. Bapak Sudirman S. Panna, M.Kom, selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
7. Bapak Yasin Aril Mustofa, M.Kom, selaku pembimbing utama dalam penelitian ini yang telah membimbing penulis selama penelitian.
8. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis.

9. Cinta pertama dan panutanku, Bapak Haswan R. Mosi dan pintu syurgaku Ibu Hartati G. Antu. Terimakasih atas segala pengorbanan dan tulus kasih yang di berikan. Yang selalu menjadi alasan saya untuk menyelesaikan studi ini, Mereka memang tidak sempat merasakan pendidikan di bangku perkuliahan, namun mereka mampu senantiasa memberikan yang terbaik, tak kenal lelah mendoakan serta memberikan perhatian, motivasi, dorongan dan dukungan kepada penulis. Menjadi suatu kebanggaan memiliki orang tua yang mendukung anaknya untuk mencapai cita-cita. Semoga Bapak dan Ibu sehat, panjang umur dan bahagia selalu.
10. Saudara saya tercinta, Iin Hardiyanti H. Mosi, S.I.K. Terima kasih sudah ikut serta dalam proses penulis menempuh pendidikan selama ini, terima kasih atas semangat, doa dan cinta yang selalu diberikan kepada penulis.
11. Sahabat-sahabat tersayang, tercinta dan terkasih. Sevilaningsih, Yusgiantoro, Fathur Rahman Lasido, dan Aldi Sudibyo Rivai. Terima kasih telah menemani penulis dalam proses penyusunan skripsi ini. Terima kasih karena sudah menjadi pendengar yang baik atas seluruh keluh kesah penulis, selalu memberikan solusi pada setiap masalah yang penulis hadapi, selalu memberikan semangat, dukungan serta masukan kepada penulis. Semoga kita menjadi orang yang sukses dan selalu bersahabat hingga kita tua nanti.
12. Saudara tak sedarah saya selama kuliah, angkatan 2021 Binary, Demisioner HIMTI 05, Pengurus BEM Kabinet Tahun 2024-2025. Bersama kalian penulis mampu merasakan keindahan di dalam perbedaan. Dari kalian pula penulis banyak belajar bahwa keikhlasan, kesabaran dan ketulusan adalah hal terindah dalam hidup.
13. Rekan seperjuangan sekaligus pendengar saya, Moh. Eka Rifqhi Anjasmara dan Moh. Adrian Djailani, yang sudah menjadi pendengar setia atas segala keluh kesah yang penulis alami. Selalu memberikan semangat dan motivasi kepada penulis.
14. Sahabat-sahabatku tercinta, Riski Ngiadi, Thasya Halim, Devi Kadir, dan Dwi Rizky yang selalu membantu, memberikan semangat serta doa terbaiknya.

15. Senior-senior saya yang senantiasa mengarahkan, membantu, memberikan dukungan kepada penulis dalam berbagai kegiatan akademik maupun kemahasiswaan selama proses perkuliahan.
16. Adik-adik tingkat tersayang, Aliya Wahid dan Ariska Tangkudung yang selalu memberikan doa dan dukungan yang tulus kepada penulis selama proses penyusunan skripsi ini.
17. Kepada semua pihak yang ikut membantu dalam penyelesaian skripsi ini yang tak sempat penulis sebutkan satu persatu.
18. *Last but not least*, kepada sang penulis karya tulis ini, diri saya sendiri. Seorang anak bungsu yang beranjak 22 tahun. Sebuah apresiasi terbesar karena sudah bertanggung jawab untuk menyelesaikan apa yang sudah dimulai. Terimakasih telah hadir di dunia dan sudah mampu bertahan sejauh ini. Terimakasih sudah memutuskan untuk tetap berusaha dan tidak menyerah. Terimakasih karena sudah mampu mengendalikan diri dari berbagai tekanan diluar dan tak pernah memutuskan menyerah sesulit apapun rintangan dalam proses penyusunan skripsi ini. Rayakanlah setiap prosesmu. Berbahagialah selalu kapanpun dan dimanapun kamu berada. Sekali lagi, untuk semua rasa takut yang berhasil dikalahkan, untuk semua tantangan yang berhasil dilalui, selamat atas pencapaiannya, Thia Riskiyani H. Mosi.

Semoga Allah SWT melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, 20 Februari 2025

Penulis

## DAFTAR ISI

PERSETUJUAN SKRIPSI .....	ii
PENGESAHAN SKRIPSI .....	iii
PERNYATAAN SKRIPSI .....	iv
ABSTRACT .....	v
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xv
DAFTAR LAMPIRAN .....	xvi
BAB I .....	1
PENDAHULUAN .....	1
1. 1    Latar Belakang .....	1
1. 2    Identifikasi Masalah .....	4
1. 3    Rumusan Masalah .....	4
1. 4    Tujuan Penelitian .....	4
1. 5    Manfaat Penelitian .....	5
BAB II .....	6
LANDASAN TEORI .....	6
2.1    Tinjauan Studi .....	6
2.2    Tinjauan Pustaka .....	7
2. 2. 1    Paru-paru .....	7
2. 2. 2 <i>Pneumonia</i> .....	8

2. 2. 3	Citra .....	9
2. 2. 4	Pengolahan Citra.....	9
2. 2. 5	<i>Computer Vision</i> .....	9
2. 2. 6	Klasifikasi.....	9
2. 2. 7	<i>Convolution Neural Network</i> .....	10
2. 2. 8	Penerapan Metode <i>Convolution Neural Network</i> .....	16
2. 2. 9	Evaluasi .....	19
2.3	Kerangka Pikir .....	21
BAB III .....	22	
METODE PENELITIAN .....	22	
3. 1	Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian .....	22
3. 2	Pengumpulan Data.....	22
3. 3	Pemodelan .....	23
3.3. 1	<i>Pre-Processing</i> .....	23
3.3. 2	Data Training .....	24
3.3. 3	Training CNN .....	24
3.3. 4	Data <i>Testing</i> .....	24
3. 4	Evaluasi Model.....	24
BAB IV .....	25	
HASIL PENELITIAN .....	25	
4. 1	Hasil Pengumpulan Data .....	25
4. 2	Hasil Pemodelan.....	26
4.2.1	Pra Pengolahan Data .....	26
4.2.2	Convolution Neural Network (CNN) .....	27
4. 3	Evaluasi Model.....	34

4.3.1	Confussion Matriks.....	34
BAB V .....		39
PEMBAHASAN.....		39
5.1	Objek Penelitian.....	39
5.2	Pembahasan Model .....	39
5.2.1	Data Citra X-ray Paru-paru.....	39
5.2.2	Ekstraksi Fitur dengan Metode (CNN).....	39
5.2.3	Arsitektur <i>Convolution Neural Network</i> (CNN).....	40
5.2.4	Proses Konvolusi dan Pooling .....	42
5.2.5	Proses Pelatihan Model.....	42
5.2.6	Training dan Validation Loss .....	43
5.2.7	<i>Training</i> dan <i>Validation</i> Akurasi .....	44
5.2.8	<i>Accuracy</i> dan <i>Validation Accuracy</i> .....	46
5.2.9	<i>Training Accuracy Over Epoch</i> .....	47
5.2.10	Evaluasi Model .....	49
5.2.11	Kurva <i>Receiver Operating Characteristic</i> (ROC) .....	51
BAB VI.....		53
PENUTUP .....		53
6.1	Kesimpulan .....	53
6.2	Saran.....	53
DAFTAR PUSTAKA .....		55

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Bagian-bagian Paru-paru.....	8
<b>Gambar 2. 2</b> Citra X-Ray Paru-paru Normal .....	8
<b>Gambar 2. 3</b> Ilustrasi Struktur CNN .....	11
<b>Gambar 2. 4</b> Proses Pooling Layer.....	12
<b>Gambar 2. 5</b> Max Pooling .....	13
<b>Gambar 2. 6</b> Flatten.....	14
<b>Gambar 2. 7</b> Fully Connected Layer .....	15
<b>Gambar 2. 8</b> Dropout.....	15
<b>Gambar 2. 9</b> Model Arsitektur Alexnet.....	17
<b>Gambar 2. 10</b> (a) Grafik Loss (b) Grafik Akurasi .....	18
<b>Gambar 2. 11</b> (a) Klasifikasi Metrik (b) Confusion Matrix .....	18
<b>Gambar 2. 12</b> Confusion Matrix .....	20
<b>Gambar 2. 13</b> Kerangka Pikir.....	21
<b>Gambar 2. 14</b> Pemodelan .....	23
<b>Gambar 4. 1</b> Citra X-Ray Paru Normal.....	25
<b>Gambar 4. 2</b> Citra X-Ray Paru <i>Pneumonia</i> .....	25
<b>Gambar 4. 3</b> Label Encoder .....	26
<b>Gambar 4. 4</b> Arsitektur Convolution Neural Network (CNN).....	27
<b>Gambar 4. 5</b> Operasi Konvolusi.....	27
<b>Gambar 4. 6</b> Sampel Gambar dan Matriks Citra.....	28
<b>Gambar 4. 7</b> Proses Konvolusi dengan Kernel 3 x 3 .....	28
<b>Gambar 4. 8</b> Feature Map.....	30
<b>Gambar 4. 9</b> Max Pooling .....	30
<b>Gambar 4. 10</b> Fully Connected Layer .....	31
<b>Gambar 5. 1</b> Arsitektur Model CNN.....	40
<b>Gambar 5. 2</b> Training Dataset dengan 50 Epoch.....	42
<b>Gambar 5. 3</b> Grafik Training dan Validation Loss .....	43
<b>Gambar 5. 4</b> Grafik Training dan Validasi Akurasi .....	44

<b>Gambar 5. 5</b> Grafik Acuraccy dan Validation Accuracy .....	46
<b>Gambar 5. 6</b> Grafik Training Accuracy Over Epoch.....	47
<b>Gambar 5. 7</b> Hasil Evaluasi Kinerja Model .....	49
<b>Gambar 5. 8</b> Confusion Matriks.....	50
<b>Gambar 5. 9</b> Kurva ROC.....	51

## DAFTAR TABEL

<b>Tabel 2. 1</b> Tinjauan Studi .....	6
<b>Tabel 2. 2</b> Distribusi Data Train, Test dan Valid .....	17
<b>Tabel 2. 3</b> Parameter.....	18
<b>Tabel 4. 1</b> Hasil Data Testing .....	34
<b>Tabel 4. 2</b> Hasil Klasifikasi pada data Testing .....	36
<b>Tabel 4. 3</b> Hasil Confussion Matriks.....	38

## DAFTAR LAMPIRAN

<b>Lampiran 1 : Dataset.....</b>	57
<b>Lampiran 2 : Kode Program .....</b>	59
<b>Lampiran 3 : Riwayat Hidup Peneliti .....</b>	74
<b>Lampiran 4 : Surat Penelitian .....</b>	75
<b>Lampiran 5 : Surat Rekomendasi Penelitian .....</b>	76
<b>Lampiran 6 : Surat Keterangan Bebas Pustaka.....</b>	77
<b>Lampiran 7 : Hasil Turnitin .....</b>	78
<b>Lampiran 8 : Lembar Revisi.....</b>	83

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Penyakit paru-paru menjadi penyebab utama kematian di Indonesia. Paru-paru adalah organ vital dalam sistem pernapasan manusia yang sangat rentan terhadap berbagai penyakit, seperti asma, bronkitis, kanker, tuberculosis, infeksi saluran pernapasan akut (ISPA), dan *pneumonia* [1]. *Pneumonia* merupakan salah satu penyakit yang dianggap serius. Biasanya, penyakit ini lebih rentan menyerang anak-anak di bawah usia 2 tahun serta lansia yang berusia 65 tahun ke atas [2]. *Pneumonia* merupakan penyakit yang disebabkan oleh bakteri, virus, dan jamur. Virus penyebab *pneumonia* adalah *Respiratory Syncial Virus* (RVS). Virus ini kebanyakan menyerang bagian atas saluran pernafasan manusia[3].

*Pneumonia* merupakan penyakit yang dianggap sangat serius. Berdasarkan data surveilans dari Kementerian Kesehatan, jumlah kasus *pneumonia* pada anak-anak lebih tinggi dibandingkan dengan kelompok usia di atas 5 tahun dalam periode 10 tahun terakhir (2011-2021). Selama periode tersebut, tercatat sebanyak 7.475.856 kasus *pneumonia* di seluruh kalangan usia. Kasus *pneumonia* pada anak dilaporkan terjadi di hampir seluruh kabupaten/kota di Indonesia. Pada tahun 2020, dari 34 provinsi yang melaporkan kasus *pneumonia*, terdapat 10 provinsi dengan jumlah kasus tertinggi pada balita, yaitu Jawa Timur, Jawa Barat, Jawa Tengah, DKI Jakarta, Banten, NTB, Lampung, Kalimantan Selatan, Sumatera Selatan, dan Sulawesi Tengah [4].

Diagnosis *pneumonia* umumnya dilakukan melalui pemeriksaan rontgen dada untuk mendeteksi adanya tanda infeksi. Namun analisis gambar rontgen memerlukan proses yang kompleks karena jenis infeksi pada paru-paru sering kali sulit untuk dibedakan dengan kondisi medis lain seperti edema paru atau penyakit pernapasan lainnya. Selain itu, kualitas gambar radiografi yang buruk, seperti resolusi rendah atau gangguan noise, sering menjadi tantangan dalam mengidentifikasi *pneumonia* dengan akurat.

Dengan perkembangan teknologi kecerdasan buatan (AI) terutama dalam *deep learning*, muncul peluang baru untuk mengatasi tantangan tersebut. Teknologi sering digunakan oleh manusia sebagai alat bantu dalam berbagai aktivitas dan pekerjaan sehari-hari. Seiring berjalannya waktu teknologi terus berkembang dan berdampak pada berbagai bidang, salah satunya adalah sektor medis yang semakin mengalami perkembangan digitalisasi [2]. Dengan adanya perkembangan teknologi, klasifikasi menggunakan kecerdasan buatan seperti CNN (*Convolution Neural Network*) menjadi solusi yang cukup baik. CNN adalah metode yang dapat mengidentifikasi pola-pola dalam citra secara otomatis dan lebih cepat dibandingkan manusia.

Berdasarkan penelitian sebelumnya yang dilakukan oleh Fani Nurona Cahya et.al (2021) menghasilkan metode *Convolutional Neural Network* untuk klasifikasi penyakit mata dengan menggunakan dataset sejumlah 610 dataset yang berasal dari *Kaggle*. Dataset tersebut terbagi menjadi 4 kelas yaitu mata normal, *katarak*, *glaucoma*, dan *retina disease*. Penelitian tersebut melalui 3 tahapan proses yaitu *pre-processing*, ekstraksi fitur menggunakan 3 layer (*convolutional layer*, *subsampling layer*, *fully connected layer*). Hasil penelitian tentang klasifikasi penyakit mata menggunakan metode CNN dengan arsitektur model AlexNet, menggunakan 4 kelas dengan melakukan tahapan *pre-processing* mengubah citra menjadi ukuran 224x224 px. Pada tahapan implementasi CNN menggunakan 150 epoch, menghasilkan akurasi sebesar 98,37%[5].

Penelitian selanjutnya yang dilakukan oleh Risha Ambar Wati et.al (2020) dengan judul klasifikasi *pneumonia* menggunakan metode *Support Vector Machine* (SVM) melakukan klasifikasi terhadap citra hasil paru-paru yang diperoleh dari website *Kaggle* sejumlah 3.140 citra rontgen paru-paru yang dibagi menjadi dua kelas diantaranya paru-paru normal dan paru-paru *pneumonia*. Metode yang digunakan dalam penelitian tersebut adalah *Support Vector Machine* (SVM) dan *Gray Level Co-Occurrence* (GLCM) untuk metode ekstraksi. Tahapan yang dilakukan sebelum klasifikasi yaitu *cropping*, *resizing*, *contrast stretching*, dan *thresholding*. Selanjutnya dilakukan ekstraksi menggunakan GLCM dan

diklasifikasikan menggunakan SVM. Hasil penelitian menunjukkan bahwa akurasi terbaik yaitu sebesar 62.66%[6].

Berdasarkan penjelasan diatas maka penulis tertarik untuk melakukan penelitian dengan judul : “Klasifikasi Penyakit *Pneumonia* Menggunakan Metode *Convolution Neural Network (CNN)*”. Dalam penelitian klasifikasi citra, khususnya dalam diagnosis penyakit berdasarkan rontgen dada (X-ray), metode *Convolution Neural Network (CNN)* dipilih karena kemampuannya yang sangat baik dalam mengenali pola dan fitur penting dari gambar. CNN (*Convolutional Neural Network*) adalah arsitektur *deep learning* yang dirancang khusus untuk memproses data gambar. Berbeda dengan metode konvensional yang memerlukan ekstraksi fitur manual, CNN secara otomatis dapat mengekstrak fitur kompleks dalam gambar seperti tepi, tekstur, bentuk, dan pola tertentu. Keunggulan utama CNN terletak pada kemampuannya untuk menangani data gambar yang besar dan kompleks dengan efisien. Melalui proses konvolusi dan pooling, CNN dapat mengurangi dimensi gambar tanpa menghilangkan informasi penting, sehingga mempercepat proses penghitungan

Selain itu, CNN memanfaatkan beberapa lapisan tersembunyi untuk melakukan pembelajaran hierarkis, di mana fungsi-fungsi sederhana dipelajari terlebih dahulu di lapisan awal, kemudian fitur yang lebih kompleks diekstraksi di lapisan-lapisan berikutnya. Pendekatan ini memungkinkan CNN untuk menghasilkan representasi fitur yang lebih akurat dan relevan dalam proses klasifikasi. Penggunaan metode *Convolutional Neural Network (CNN)* untuk mengklasifikasikan *pneumonia* sangat tepat, karena metode ini terbukti efektif dalam klasifikasi citra medis, terutama dalam menangani kasus *pneumonia*.

Penelitian ini bertujuan untuk mengukur akurasi metode *Convolution Neural Network (CNN)* dalam mengklasifikasikan *pneumonia* berdasarkan dataset citra x-ray. Penelitian ini akan menggunakan dataset citra rontgen paru-paru yang bersumber dari website *Kaggle* dengan dua kelas yaitu citra rontgen paru-paru normal dan *pneumonia*. CNN menjadi pilihan metode yang tepat untuk penelitian ini karena kemampuannya dalam mengotomatisasi ekstraksi fitur, efisiensi komputasi dan akurasi yang tinggi dalam pengolahan citra medis seperti citra

rontgen dada. Penelitian ini diharapkan dapat memberikan kontribusi penting dalam bidang teknologi kesehatan, khususnya dalam proses diagnosis *pneumonia* menggunakan metode Convolutional Neural Network (CNN) untuk klasifikasi gambar rontgen dada. Dengan diagnosis yang cepat dan akurat, perawatan dapat dimulai lebih awal.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan maka dapat diidentifikasi masalahnya adalah :

1. Tingginya angka kasus *pneumonia* di Indonesia
2. Pengklasifikasian penyakit *pneumonia* menggunakan metode *Convolution Neural Network* (CNN) untuk mengetahui tingkat kecocokan dari metode yang digunakan berdasarkan dataset.

## 1.3 Rumusan Masalah

Berdasarkan latar belakang diatas, rumusan masalah dalam penelitian ini adalah :

1. Bagaimana pengklasifikasian menggunakan model *Convolution Neural Network* (CNN) untuk mengklasifikasikan penyakit *pneumonia* berdasarkan citra X-Ray paru-paru ?
2. Bagaimana hasil akurasi metode *Convolutional Neural Network* (CNN) dalam pengklasifikasian penyakit *pneumonia* dari citra X-ray paru-paru ?

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, tujuan penelitian ini adalah :

1. Untuk menerapkan metode *Convolutional Neural Network* (CNN) dalam mengklasifikasikan penyakit *pneumonia*.
2. Untuk mengetahui hasil akurasi menggunakan metode *Convolutional Neural Network* (CNN) dalam mengklasifikasikan penyakit *pneumonia*.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan mempunyai manfaat, yaitu :

### 1. Pengembangan ilmu

Penelitian ini diharapkan mampu untuk mengembangkan ilmu pengetahuan dibidang teknologi komputer serta mampu menjadi bahan pertimbangan dalam pengambilan keputusan dari hasil akurasi yang akurat dengan menggunakan metode *Convolution Neural Network*.

### 2. Secara Teoritis

Penelitian ini di harapkan dapat mengembangkan ilmu pengetahuan bagi akademis sebagai bahan masukan pemikiran mengenai permasalahan penyakit *pneumonia* dengan efektif dan efisien, dan memberikan sumber informasi bagi mahasiswa apabila melakukan penelitian yang sejenis.

### 3. Secara Praktis

Penelitian ini diharapkan dapat berkontribusi dalam pengembangan pengetahuan bagi perusahaan, memberikan bahan informasi yang diperlukan untuk optimalisasi dan manajemen aplikasi dalam mengklasifikasikan penyakit *pneumonia*.

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Studi

Tinjauan studi sangat berguna bagi penulis dalam memberikan pedoman serta pegangan untuk penulis selanjutnya. Nantinya, dengan adanya penelitian sebelumnya akan mempermudah penelit dalam melakukan penelitian yang sesuai dengan topik yang akan dibahas. Berikut ini terdapat beberapa jurnal yang relevan dengan penelitian ini, yaitu :

**Tabel 2. 1** Tinjauan Studi

NO	PENELITI	JUDUL	HASIL
1.	Risha Ambarwati, Hafiz Irsyad, M. Ezar Al Rivan, 2020[6]	Klasifikasi <i>pneumonia</i> menggunakan metode <i>Support Vector Machine</i>	Hasil pengujian pada penelitian ini dapat disimpulkan bahwa SVM mampu mengklasifikasikan citra paru-paru dengan jumlah dataset testing sebanyak 600 menggunakan fitur GLCM dengan hasil <i>accuracy</i> keseluruhan sebanyak 62,66%. SVM juga mampu mengklasifikasikan citra paru-paru dengan jumlah data <i>testing</i> sebanyak 750 menggunakan fitur GLCM dengan hasil <i>accuracy</i> keseluruhan sebesar 59,2%.
	Fani Nurona Cahya, Nila Hardi, Dwiza Riana, Sri Hadianti, 2021[5]	Klasifikasi penyakit mata pada penelitian ini menggunakan metode CNN dengan model AlexNet, menggunakan 4 kelas, dengan melakukan tahapan <i>pre-processing</i> mengubah ukuran citra menjadi 224x224px. Selanjutnya melakukan	Klasifikasi penyakit mata pada penelitian ini menggunakan metode CNN dengan model AlexNet, menggunakan 4 kelas, dengan melakukan tahapan <i>pre-processing</i> mengubah ukuran citra menjadi 224x224px. Selanjutnya melakukan

			<i>feature extraction</i> dengan 3 lapisan yaitu lapisan <i>convolutional</i> , <i>pooling</i> , dan <i>fully connected</i> . Pada tahapan implementasi CNN menggunakan 150 epoch dan menghasilkan nilai akurasi sebesar 98,37%.
	Sri Dewi, Fanny Ramadhani, Selvia Djasmayena, 2024[7]	Klasifikasi Jenis Jerawat Berdasarkan Gambar Menggunakan Algoritma CNN ( <i>Convolutional Neural Network</i> )	Hasil penelitian ini menunjukkan bahwa model <i>Convolutional Neural Network</i> (CNN) yang digunakan untuk melakukan klasifikasi jenis jerawat pada citra wajah menggunakan 42 data pelatihan dan 8 data pengujian mampu mendapatkan akurasi sebesar 88%, dengan evaluasi model menggunakan <i>confussion matrix</i> dan <i>classification report</i> . Tinggi rendahnya akurasi yang didapatkan tergantung dari kualitas citra yang dimiliki.

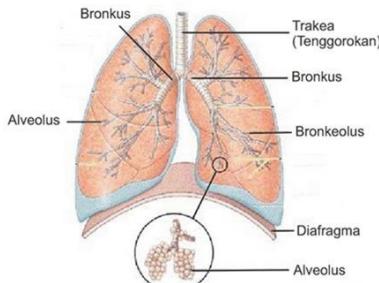
## 2.2 Tinjauan Pustaka

### 2. 2. 1 Paru-paru

Paru-paru adalah organ yang terletak di dalam rongga dada dan memiliki berat antara 300 hingga 400 gram. Organ ini terbagi menjadi dua bagian, yaitu paru-paru kanan dan kiri. Paru-paru kanan terdiri dari tiga lobus yaitu superior, medius, dan inferior, sementara paru-paru kiri hanya memiliki dua lobus yaitu superior dan inferior[8].

Paru-paru berperan penting dalam proses pernapasan yang mempengaruhi aliran oksigen dan karbondioksida dalam tubuh. Fungsi utama paru-paru adalah mentransfer oksigen dari udara ke dalam darah dan mengeluarkan karbondioksida dari darah ke udara. Dalam proses pernapasan, udara masuk melalui mulut atau hidung, melewati *trachea* (tenggorokan), *bronkus*, dan *bronkiolus* hingga mencapai *alveoli*. Di *alveoli*, oksigen diserap dari udara dan disalurkan ke dalam darah untuk

diedarkan ke seluruh tubuh, sementara karbodioksida dikeluarkan dari darah dan bergerak kembali ke *alveoli* untuk dikeluarkan[6]. Paru-paru merupakan organ vital pernafasan manusia yang sangat penting dan rentan terhadap serangan penyakit seperti asma, *bronkitis*, kanker, TBC, ISPA, dan *pneumonia*[1].



**Gambar 2. 1** Bagian-bagian Paru-paru



**Gambar 2. 2** Citra X-Ray Paru-paru Normal

### 2. 2. 2 *Pneumonia*

*Pneumonia* adalah infeksi pada saluran pernapasan yang disebabkan oleh bakteri, virus, atau jamur. Infeksi ini dapat menyerang salah satu atau kedua paru-paru yang mengakibatkan kantung udara di paru-paru kesulitan menyerap oksigen[9]. Gejala yang muncul pada penderita *pneumonia* sangat bervariasi tergantung pada penyebab infeksi, tingkat keparahannya, serta usia dan kondisi kesehatan umum pasien. Gejala ini dapat berkembang dengan cepat atau secara perlahan dalam kurun waktu 24-48 jam.[10].

### 2. 2. 3 Citra

Citra digital terdiri dari sejumlah titik yang disebut piksel (*pixel atau picture element*). Setiap piksel digambarkan sebagai kotak kecil dengan koordinat posisi tertentu. Citra atau gambar dapat dipandang sebagai fungsi dua dimensi  $f(x, y)$ , di mana setiap nilai  $f$  merepresentasikan informasi pada posisi tertentu dalam citra tersebut.[11]. Citra adalah kombinasi dari titik, garis, kotak, dan warna yang menciptakan suatu objek fisik. Citra dapat berupa gambar 2D seperti lukisan, foto, dan video.

### 2. 2. 4 Pengolahan Citra

Pengolahan citra digital (*image processing*) adalah bidang ilmu yang fokus pada teknik-teknik untuk memanipulasi citra. Citra yang dimaksud dapat berupa gambar diam (foto) maupun gambar bergerak, seperti video yang direkam[12].

### 2. 2. 5 Computer Vision

*Computer vision* adalah bidang yang mempelajari cara menganalisis gambar atau video untuk mendapatkan hasil yang serupa dengan kemampuan manusia. Sebagai cabang ilmu komputer, tujuannya adalah memungkinkan komputer untuk melihat objek dengan memproses citra, baik berupa gambar maupun video. Komputer akan menganalisis input citra yang diberikan dan kemudian memberikan informasi berdasarkan analisis tersebut [13].

### 2. 2. 6 Klasifikasi

Klasifikasi merupakan sebuah teknik analisis data yang mengekstrak model untuk mendeskripsikannya ke dalam kelas tertentu[14]. Klasifikasi adalah proses pembelajaran untuk menghasilkan sebuah fungsi yang dapat memetakan setiap himpunan objek ke dalam salah satu label kelas yang telah ditentukan sebelumnya. Klasifikasi termasuk dalam jenis analisis yang digunakan untuk menentukan kelas label dari sejumlah sampel yang akan dikategorikan. Klasifikasi adalah bentuk dasar dari analisis data dan sering dianggap sebagai teknik untuk mengidentifikasi kategori kelas dari data yang ada. Dalam klasifikasi, terdapat berbagai teknik atau algoritma yang digunakan untuk menyelesaikan masalah atau membangun model

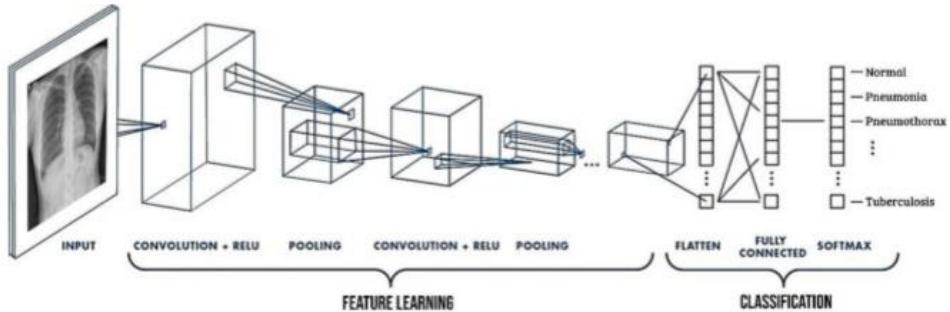
data, seperti algoritma C4.5 (*decision tree*), *naïve bayes*, *neural network*, *k-nearest neighbor*, dan *support vector machine*[15].

### **2. 2. 7 Convolution Neural Network**

*Convolution Neural Network* (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk termasuk dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. CNN pertama kali dikembangkan dengan nama *NeoCognitron* oleh Kunihiko Fukushima, seorang peneliti dari NHK *Broadcasting Science Research Laboratories*, Kinuta, Setagaya, Tokyo, Jepang[5]

CNN (*Convolutional Neural Network*) adalah salah satu arsitektur jaringan saraf yang dirancang khusus untuk klasifikasi citra. CNN dapat dianggap sebagai inovasi dalam pengklasifikasian citra karena merupakan bagian dari jaringan saraf yang terinspirasi oleh cara kerja otak manusia. Salah satu keunggulan CNN adalah kemampuannya untuk mengeliminasi kebutuhan akan ekstraksi fitur pola secara manual yang biasanya memerlukan bantuan pakar. Untuk mencapai hal ini, CNN menggunakan lapisan yang disebut *convolution*, yang berfungsi untuk mengekstrak pola fitur secara otomatis [16]

Dalam penerapannya, metode *Convolutional Neural Network* (CNN) melalui beberapa tahapan, seperti model *training*, model *evaluation*, dan model *prediction*. CNN terbagi menjadi dua bagian utama, yaitu *feature learning* dan *classification*. Proses ekstraksi fitur dalam CNN melibatkan beberapa lapisan tersembunyi, termasuk lapisan konvolusi, fungsi aktivasi, dan pooling layer. *Feature learning* bertugas untuk mengekstrak fitur penting dari input, sementara *classification* berhubungan dengan output setelah proses pembelajaran selesai. Struktur CNN terdiri dari lapisan input, lapisan tersembunyi, dan lapisan output. CNN bekerja secara hierarkis, di mana output dari lapisan konvolusi pertama menjadi input untuk lapisan konvolusi berikutnya [17].



**Gambar 2. 3 Ilustrasi Struktur CNN**

Arsitektur CNN pertama yang dikembangkan adalah LeNet-5, yang dirancang oleh LeCun dan rekan-rekannya untuk mengenali angka tulisan tangan dan berhasil mendeteksi pola citra input tanpa perlu pra-pemrosesan. Namun, karena keterbatasan data pelatihan dan daya komputasi yang rendah, model ini tidak dapat menangani masalah yang lebih kompleks. Pada tahun 2012, Krizhevsky dan timnya memperkenalkan AlexNet, yang meningkatkan kemampuan CNN dengan menambahkan beberapa lapisan tambahan. Perubahan ini secara signifikan meningkatkan akurasi dalam tugas klasifikasi. AlexNet menawarkan beberapa inovasi untuk meningkatkan kinerja CNN, seperti penggunaan fungsi aktivasi ReLU, *dropout*, dan *augmentation*, yang kemudian banyak digunakan. Seiring berjalannya waktu, beberapa arsitektur lain telah dikembangkan dan ditingkatkan lebih lanjut dengan menambahkan lebih banyak lapisan, seperti ResNet, VGGNet, dan GoogLeNet. [18].

### 1. *Convolutional Layer*

*Convolution Layer* adalah lapisan pertama dalam arsitektur CNN yang menerima input citra. Lapisan ini berfungsi untuk mengekstrak fitur dari citra dengan menggunakan filter yang membagi vektor input menjadi blok-blok kecil. Hasil dari operasi konvolusi ini adalah feature maps, yang kemudian digunakan sebagai input untuk lapisan berikutnya dalam jaringan.

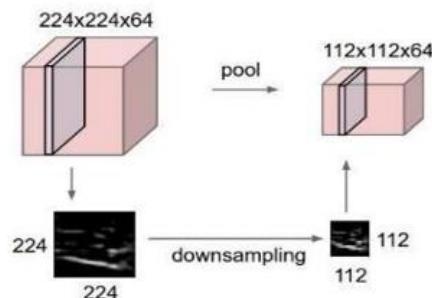
### 2. *Activation Function*

Fungsi aktivasi merupakan salah satu elemen penting dalam arsitektur *deep learning* yang mempengaruhi output, akurasi, dan efisiensi model pelatihan.

Fungsi aktivasi bertugas untuk menentukan apakah neuron akan diaktifkan berdasarkan input tertentu, dengan menghasilkan output yang sesuai. Beberapa contoh fungsi aktivasi antara lain Sigmoid, Tanh, ReLU, Leaky ReLU, Parametrized ReLU, Exponential Linear Unit, Swish, dan SoftMax. Fungsi aktivasi yang paling sering digunakan adalah ReLU. ReLU bekerja dengan cara *thresholding*, di mana nilai piksel yang kurang dari nol pada input citra akan diubah menjadi nol.

### 3. Pooling Layer

*Pooling layer* adalah proses yang digunakan untuk mereduksi input secara spasial atau mengurangi jumlah parameter melalui operasi down sampling. Dimensi dari setiap feature map akan mengecil, namun informasi penting tetap dipertahankan. Fungsi utama dari pooling layer adalah untuk mempercepat proses komputasi dan meningkatkan kinerja secara keseluruhan dari seluruh convolutional layer.

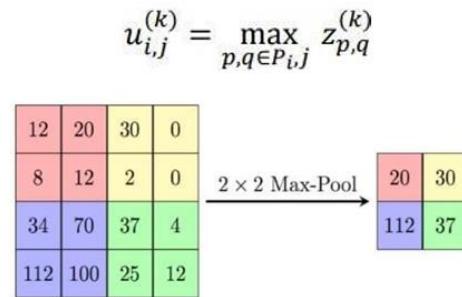


**Gambar 2. 4** Proses Pooling Layer

*Pooling layer* memiliki beberapa macam tipe antara lain sebagai berikut.

#### a. Max Pooling

*Max pooling output* pola fitur (z) didapatkan dengan mencari nilai yang paling besar pada blok spasial pada setiap pergerakan. Metode ini merupakan metode standar yang digunakan dalam penelitian.



**Gambar 2. 5 Max Pooling**

b. *Average Pooling*

*Average Pooling* mendapatkan pola fitur keluaran dengan menghitung nilai rata-rata masukan pola fitur pada blok spasial  $H \times H$  piksel disetiap wilayah pergerakan. *Average pooling* tetap menyimpan informasi spasial pada pola fitur sebelumnya.

$$u_{i,j}^{(k)} = \frac{1}{H^2} \sum_{(p,q) \in P_{i,j}} Z_{p,q}^{(k)}$$

c. *Lp Pooling*

*Lp Pooling* adalah tipe *pooling layer* generalisasi dari *max pooling* dan *average pooling*.

$$u_{i,j}^{(k)} = \left( \frac{1}{H^2} \sum_{(p,q) \in P_{i,j}} (Z_{p,q}^{(k)})^p \right)^{\frac{1}{p}}$$

Ketika nilai  $P=1$ , *pooling layer* bekerja sebagai *average pooling* dan jika  $P=\infty$  *pooling layer* bekerja sebagai *max pooling*. Sebagai contoh, input ke *pooling layer* dengan ukuran  $W_1 \times H_1 \times D_1$ . Untuk mendapatkan ukuran output dari *pooling layer* dapat menggunakan persamaan berikut.

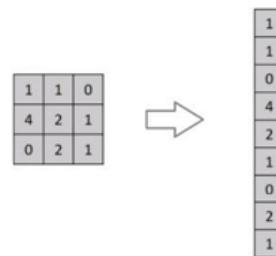
$$W2 = \frac{w1 - F}{S} + 1 = H2$$

$$D2 = D1$$

Dimensi keluaran dari *pooling layer* juga menggunakan rumus yang sama seperti pada *convolution layer*. Tujuan penggunaan *pooling layer* adalah untuk mengurangi ukuran *feature map* (*downsampling*), sehingga mempercepat proses komputasi.

#### 4. *Flatten*

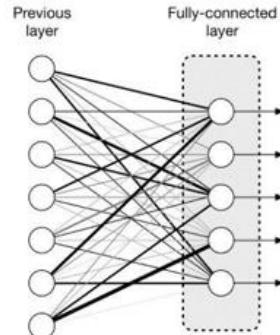
*Flatten* yaitu membentuk ulang fitur menjadi sebuah vector agar bisa digunakan sebagai input dari *fully connected layer*.



**Gambar 2. 6** *Flatten*

#### 5. *Fully Connected Layer*

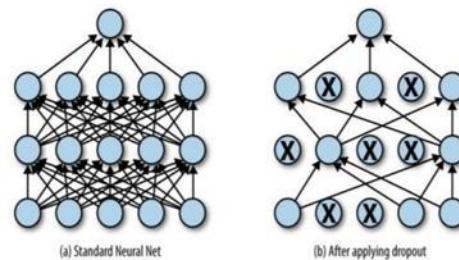
*Fully Connected Layer* adalah struktur dasar dalam *neural network* di mana setiap neuron terhubung dengan semua neuron di lapisan sebelumnya. Lapisan ini biasanya ditempatkan di akhir jaringan. Pada lapisan output, fungsi aktivasi yang sering digunakan adalah softmax, yang bertujuan untuk klasifikasi. Fungsi aktivasi softmax membantu menyelesaikan masalah pembelajaran jaringan saraf multi-kelas dan klasifikasi gambar dengan sekumpulan piksel sebagai input. Output dari softmax akan menghasilkan distribusi probabilitas untuk masing-masing kelas, memungkinkan model untuk menentukan kelas mana yang paling sesuai dengan input yang diberikan[18].



**Gambar 2. 7 Fully Connected Layer**

#### 6. Dropout

Kemungkinan model *fully connected layer* akan menimbulkan *overfitting*. Oleh karena itu, diterapkan *dropout* pada model ini untuk mengurangi hal tersebut.



**Gambar 2. 8 Dropout**

Dropout adalah teknik regularisasi yang secara acak menonaktifkan neuron-neuron tertentu selama proses pelatihan, sehingga tidak semua neuron dan bobot terlibat dalam perhitungan pada setiap iterasi. Dengan cara ini, model dapat menghindari *overfitting* dan menjadi lebih generalis. Probabilitas neuron yang terlibat dalam perhitungan biasanya ditetapkan antara 50% hingga 70%. Neuron yang tidak aktif akan ditandai dengan tanda silang, dan mereka tidak berkontribusi pada perhitungan selama iterasi pelatihan tersebut.

Adapun kelebihan dan kelemahan dari metode *Convolution Neural Network* (CNN) antara lain :

- Kelebihan metode *Convolution Neural Network* (CNN)
  - Kemampuan otomatis dalam ekstraksi fitur

- Memiliki nilai akurasi yang tinggi dalam klasifikasi citra
  - Lebih efektif untuk menangani data dalam jumlah yang besar
  - Selain digunakan dalam bidang medis, CNN mampu diterapkan dalam berbagai bidang
  - CNN memiliki kemampuan untuk mengenali objek meskipun mengalami rotasi atau perubahan yang perspektif.
- b. Kekurangan metode *Convolution Neural Network* (CNN)
- CNN memerlukan dataset yang besar agar dapat berfungsi secara optimal
  - Memiliki arsitektur yang kompleks dengan banyak lapisan sehingga waktu pelatihan dapat menjadi sangat lama terutama pada data set yang berukuran besar
  - Untuk mengolah citra beresolusi tinggi dan model yang dalam, CNN memerlukan konsumsi memori yang sangat besar.

### 2. 2. 8 Penerapan Metode *Convolution Neural Network*

Berikut adalah contoh penerapan metode *Convolution Neural Network* (CNN) yang penulis dapatkan dengan judul kasus klasifikasi penyakit mata menggunakan metode *Convolutional Neural Network* (CNN)[5] Dataset yang digunakan dalam penelitian ini berjumlah 610 dataset yang terbagi menjadi empat kelas. Penelitian ini memaparkan klasifikasi perbedaan mata normal, *katarak*, *glaucoma*, dan *retina disease*. Dimana pengklasifikasian ini akan sangat bermanfaat dan membantu mendeteksi penyakit mata secara tepat dan akurat. Pada tahapan *pre-processing* dilakukan analisis terhadap pra proses data masukan sebelum masuk ke dalam tahap klasifikasi. Adapun tahapan-tahapan *pre-processing* yang dilakukan adalah *grayscale*, *thresholding*, *segmentasi*, dan *resize*.

Arsitektur *Convolutional Neural Network* (CNN) yang digunakan pada penelitian ini yaitu arsitektur Alexnet dengan model yang diterapkan seperti pada gambar berikut.

```

Model: "sequential"
-----  

Layer (type)          Output Shape       Param #
conv2d (Conv2D)      (None, 224, 224, 16)  208
-----  

max_pooling2d (MaxPooling2D) (None, 112, 112, 16)  0
-----  

conv2d_1 (Conv2D)      (None, 112, 112, 32)  2080
-----  

max_pooling2d_1 (MaxPooling2 (None, 56, 56, 32)  0
-----  

conv2d_2 (Conv2D)      (None, 56, 56, 64)  8256
-----  

max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)  0
-----  

dropout (Dropout)      (None, 28, 28, 64)  0
-----  

flatten (Flatten)      (None, 50176)  0
-----  

dense (Dense)          (None, 500)  25088500
-----  

dropout_1 (Dropout)      (None, 500)  0
-----  

dense_1 (Dense)          (None, 4)  2004
-----  

Total params: 25,101,048
Trainable params: 25,101,048
Non-trainable params: 0

```

**Gambar 2. 9** Model Arsitektur Alexnet

Pada tahap *pre-processing* citra di *resize* ukurannya menjadi 224x224px. Dengan ukuran citra yang seragam berguna untuk memudahkan dalam proses komputasi. Selain itu, ukuran citra yang seragam juga dapat memudahkan pada tahap pengenalan. Selanjutnya masuk ke dalam proses distribusi data yang dapat dilihat pada tabel berikut.

**Tabel 2. 2** Distribusi Data Train, Test dan Valid

Dataset	
610	
Train	Test
430	
Train	Valid
439	50
	121

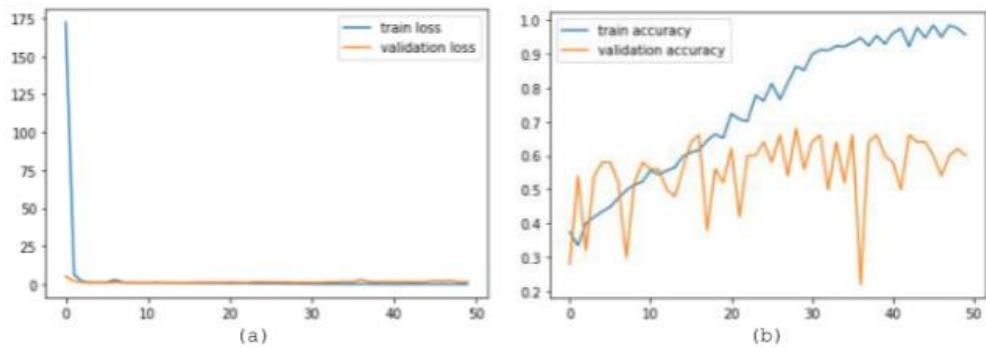
Berdasarkan tabel diatas data *training* dan validasi tersebut digunakan dalam proses *training*, *tuning* dan evaluasi model CNN. Sedangkan data *test* digunakan untuk menguji performa dari model hasil *training* tersebut.

Setelah mendapatkan citra dengan ukuran yang seragam dan distribusi data citra, selanjutnya masuk kedalam tahap pengklasifikasian. Pada tahap ini peneliti menggunakan parameter seperti yang terlihat pada tabel berikut.

**Tabel 2. 3** Parameter

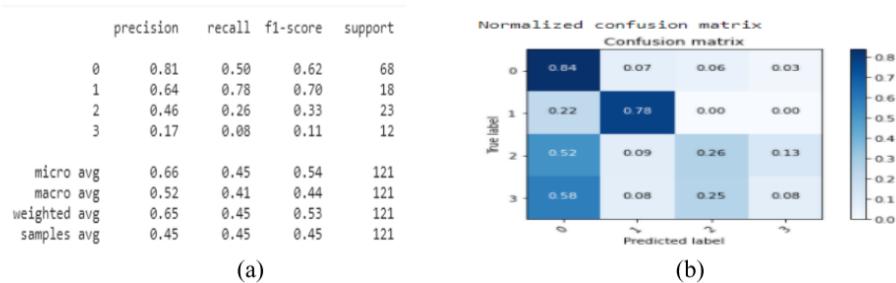
Ukuran Citra	Epoch	batch_size	Optimizer
224x224px	150 epoch	32	Adam

Berdasarkan parameter yang digunakan dengan arsitektur AlexNet dan model yang diterapkan, diperoleh hasil seperti terlihat pada gambar dibawah ini.



**Gambar 2. 10** (a) Grafik Loss (b) Grafik Akurasi

Berdasarkan gambar diatas, dapat dijelaskan bahwa klasifikasi penyakit mata menggunakan *Convolutional Neural Network* (CNN) menghasilkan akurasi sebesar 98.37%. Untuk mengetahui performa klasifikasi, maka perlu dilakukan evaluasi terhadap klasifikasi *metrik* dan *confusion matrix* seperti yang terlihat pada gambar dibawah ini.



**Gambar 2. 11** (a) Klasifikasi Metrik (b) Confusion Matrix

## 2. 2. 9 Evaluasi

Ada banyak cara untuk mengevaluasi model yang sudah dibuat dengan menggunakan ukuran kinerja yang berbeda, seperti *confusion matrix*, *accuracy*, *recall*, *precision*, *sensitivity*, *specificity*, *F1-Score* dan *dice coefficient*[18].

### 1. Akurasi

Akurasi merupakan rasio prediksi benar (positif negative) terhadap keseluruhan data uji. Nilai akurasi dapat diperoleh dengan rumus sebagai berikut[19] :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \%$$

### 2. Presisi

Presisi adalah rasio prediksi positif sejati dibandingkan dengan hasil positif prediksi keseluruhan. Presisi terbaik adalah 1,0 dan yang terburuk adalah 0,0. Nilai presisi dapat diperoleh dengan rumus sebagai berikut [19] :

$$Presisi = \frac{TP}{TP + FP} \times 100 \%$$

### 3. *Recall*

*Recall* adalah menghitung jumlah citra prediktif positif dan positif dibandingkan dengan total citra positif dan data positif. Nilai *recall* terbaik adalah 1,0 sedangkan yang terburuk adalah 0,0. Nilai *recall* dapat diperoleh dengan rumus sebagai berikut[19] :

$$Recall = \frac{TP}{TP + FN} \times 100 \%$$

### 4. *F-Measure*

*F-Measure* (F1-Score) adalah metrik evaluasi yang menggabungkan hasil dari recall dan presisi dalam satu nilai tunggal dan memberikan keseimbangan antara keduanya. Pada situasi tertentu, recall dan presisi mungkin memiliki bobot yang berbeda [19].

$$F - measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} \times 100 \%$$

### 5. Confusion Matrix

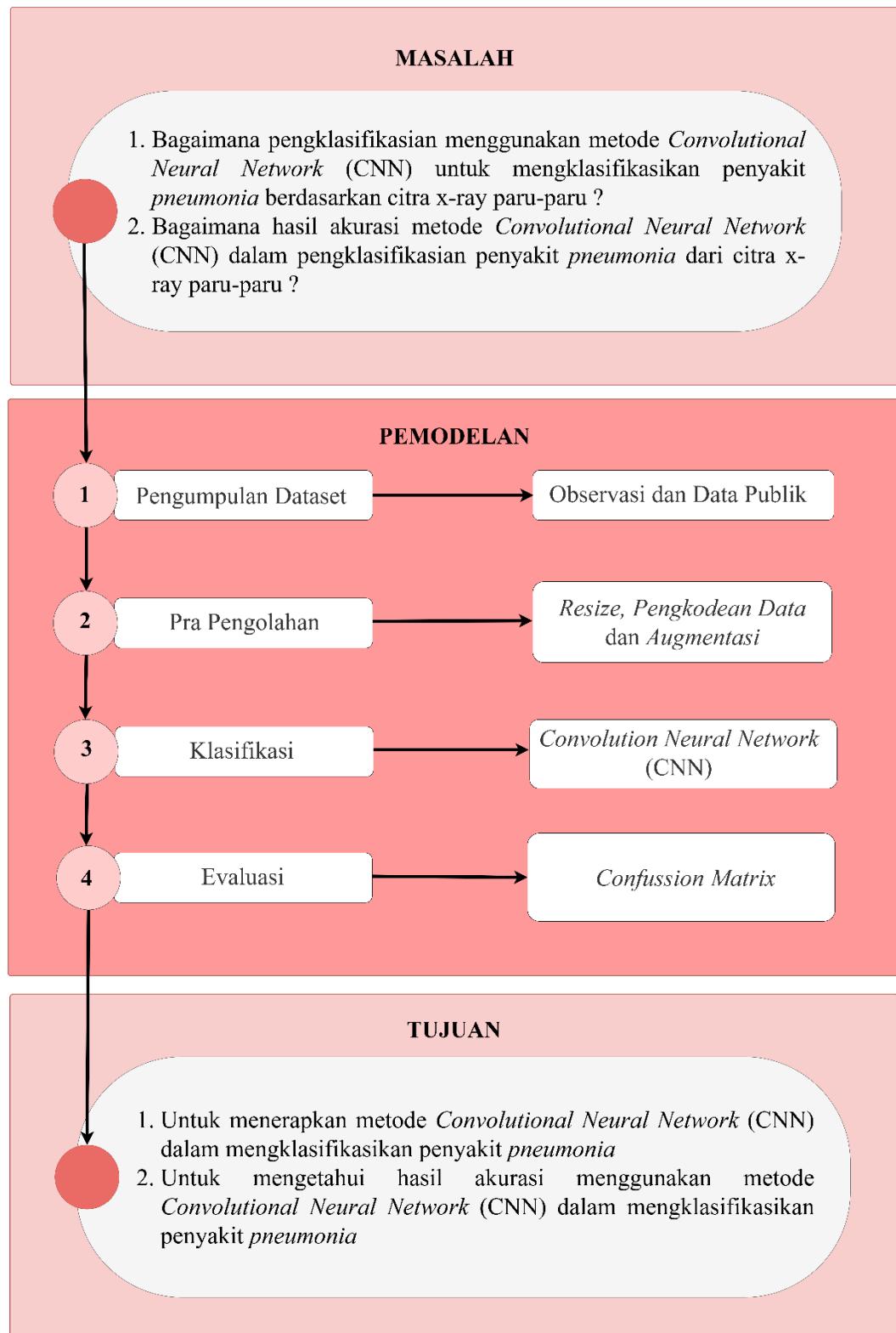
Confusion matriks adalah alat yang digunakan untuk mengevaluasi kinerja suatu classifier dengan menunjukkan bagaimana prediksi model dibandingkan dengan data aktual dalam kategori yang berbeda. Dalam confusion matrix, terdapat empat istilah utama yang merepresentasikan hasil klasifikasi:

- TP ( True Positif) : jumlah data positif aktual yang diprediksi positif (gambar *pneumonia* diidentifikasi sebagai *pneumonia*)
- TN (True Negatif) : jumlah data negatif aktual yang diprediksi negatif (gambar normal diidentifikasi sebagai normal atau sehat)
- FP (False Positif) : jumlah data negative aktual yang diprediksi positif (gambar normal salah diidentifikasi sebagai *pneumonia*)
- FN (False Negatif) : jumlah data positif actual yang diprediksi negative (gambar *pneumonia* salah diidentifikasi sebagai normal atau sehat)[19].

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP (True Positive)	FN (False Positive)
	Negative	FP (False Negative)	TN (True Negative)

**Gambar 2. 12** Confusion Matrix

### 2.3 Kerangka Pikir



**Gambar 2. 13** Kerangka Pikir

## BAB III

### METODE PENELITIAN

#### 3.1 Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian

Di lihat dari tingkat penerapannya, penelitian ini merupakan penelitian terapan. Karena sifat data atau jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif.

Penelitian ini menggunakan metode penelitian *eksperimen*. Dengan demikian, jenis penelitian ini adalah penelitian *eksperimental*.

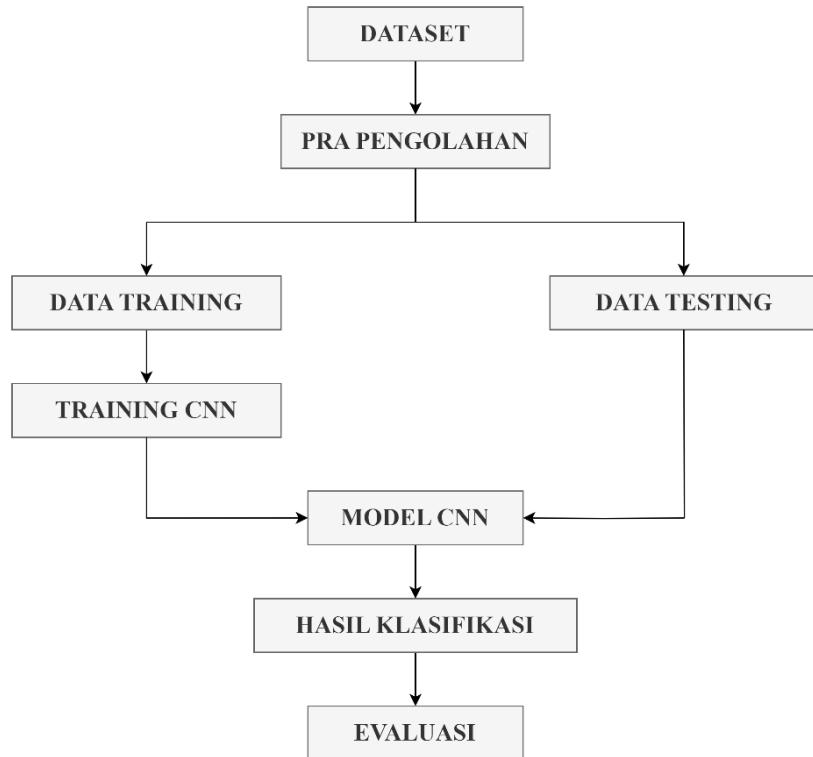
Subjek penelitian ini adalah klasifikasi pada objek gambar citra x-ray paru-paru manusia untuk penyakit *pneumonia* dan *non-pneumonia*. Penelitian ini dimulai dari bulan September 2024.

#### 3.2 Pengumpulan Data

Untuk pengumpulan data digunakan data sekunder. Dataset yang digunakan dalam penelitian ini adalah dataset *public* yang bersumber dari website *Kaggle*. Dataset ini dibagi menjadi dua bagian yaitu data training dan data testing. Berikut penjelasan mengenai data set.

Nama Dataset	:	GAMBAR RONTGEN DADA ( <i>PNEUMONIA</i> )
Jumlah seluruh data	:	5.856
Jumlah data training	:	Normal : 1341 <i>Pneumonia</i> : 3875
Jumlah data testing	:	Normal : 234 <i>Pneumonia</i> : 390
Jumlah data validasi	:	Normal : 8 <i>Pneumonia</i> : 8
Ukuran gambar	:	150 x 150 Pixel
Format	:	JPG

### 3.3 Pemodelan



**Gambar 2. 14 Pemodelan**

#### 3.3. 1 *Pre-Processing*

Pre-processing adalah tahap krusial dalam pengolahan data yang bertujuan untuk mengatasi masalah yang dapat memengaruhi hasil klasifikasi. Pada tahap ini, dilakukan analisis dan persiapan data masukan sebelum diteruskan ke proses klasifikasi. Beberapa tahapan pre-processing yang umum dilakukan meliputi resize, yang mengubah ukuran citra atau data agar sesuai dengan format yang dibutuhkan oleh model. Selain itu, encoding data juga penting untuk mengonversi data kategorikal menjadi format numerik yang bisa dipahami oleh model, seperti dengan menggunakan teknik one-hot encoding. Augmentasi dilakukan untuk memperkaya variasi data, misalnya dengan cara memutar, membalik, atau mengubah kontras citra, yang akan membantu model menjadi lebih tangguh dan meningkatkan akurasinya. Dengan langkah-langkah pre-processing ini, data dipersiapkan dengan lebih baik untuk menghasilkan model yang lebih akurat dan efisien

### **3.3. 2 Data Training**

Data *training* berupa data yang telah diseimbangkan jumlahnya untuk diolah yang nantinya akan dilatih menggunakan algoritma *Convolutional Neural Network* (CNN). Data training

### **3.3. 3 Training CNN**

Training menggunakan metode *Convolutional Neural Network* (CNN) yaitu menjadikan data *training* menjadi data *inputan*. Algoritma *Convolution Neural Network* (CNN) bekerja dengan proses konvolusi dengan menerapkan karnel konvolusi (filter) dengan ukuran tertentu pada gambar, dalam hal ini komputer memperoleh informasi perwakilan baru dengan mengalikan bagian gambar dengan filter yang digunakan. Setelah itu, nilai matriksnya akan dinormalisasikan dengan fungsi aktifasi ReLU, kemudian akan dilakukan proses *pooling layer* untuk mengecilkan ukuran gambar. Ketiga proses diatas sebagai proses fitur ekstraksi dan selanjutnya akan dilakukan proses klasifikasi. Dan dari proses training ini akan menjadi acuan untuk dijadikan pengenalan pada data testing.

### **3.3. 4 Data *Testing***

Data *testing* merupakan data yang telah diseimbangkan jumlahnya yang akan digunakan untuk menguji data *training*. Data testing akan digunakan untuk mengetahui berapa tingkat keberhasilan untuk melakukan klasifikasi penyakit *pneumonia*.

## **3. 4 Evaluasi Model**

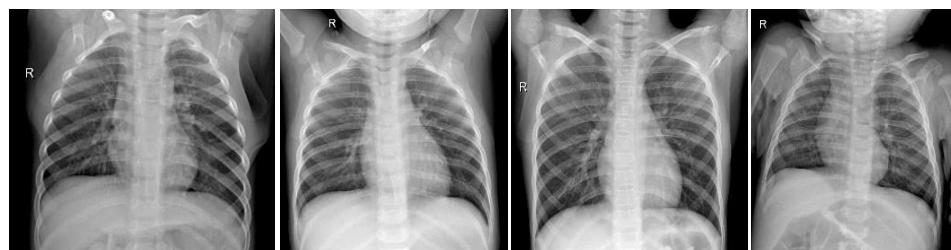
Proses evaluasi bertujuan untuk mengetahui hasil kerja dari metode yang digunakan. Evaluasi dilakukan pada semua data testing kemudian target output yang dihasilkan akan dipetakan ke dalam *Confussion Matrix* untuk dinilai tingkat akurasinya.

## BAB IV

### HASIL PENELITIAN

#### 4.1 Hasil Pengumpulan Data

Data set yang digunakan dalam penelitian ini adalah dataset *public* dengan nama Gambar Rontgen Dada (*pneumonia*) yang berasal dari <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia> ,Dataset ini merupakan dataset yang pada umumnya digunakan dalam eksperimen untuk klasifikasi penyakit *pneumonia*. Ukuran dataset yang digunakan adalah 150 x 150 pixel dengan format gambar jpg. Dengan menggunakan data training 5216 dan data testing 624.



**Gambar 4. 1** Citra X-Ray Paru Normal



**Gambar 4. 2** Citra X-Ray Paru *Pneumonia*

## 4.2 Hasil Pemodelan

### 4.2.1 Pra Pengolahan Data

Pra-pengolahan citra (*image pre-processing*) adalah pemrosesan awal yang dilakukan dalam pengolahan citra sebelum proses utama dilakukan. Pra pengolahan data bertujuan untuk mempermudah proses dalam melakukan klasifikasi citra. Pada tahap ini citra x-ray yang sudah ada dikonversikan agar diperoleh data yang sesuai dengan kebutuhan. Tahap ini berfungsi untuk menormalisasikan data citra x-ray, kemudian dilakukan proses encoding data atau perubahan data label dari kata menjadi numerik. Selanjutnya dilakukan proses augmentasi agar dapat menambah jumlah data menjadi lebih banyak. Pra pengolahan dalam penelitian ini dilakukan dengan beberapa tahap yaitu :

#### 1. *Resize*

*Resize* merupakan tahapan awal dari *pre-processing*. *Resize* bertujuan untuk mengubah ukuran citra asli dari ukuran yang bervariatif menjadi 150 x 150 piksel lebih kecil dari skala citra asli agar dapat mempercepat proses latih.

#### 2. *Encoding Data (Label Encoder)*

Pada proses ini akan dilakukan transformasi label kata menjadi bentuk numerik agar dapat diolah oleh sistem.

Pneumonia	> 0
Normal	> 1

**Gambar 4. 3 Label Encoder**

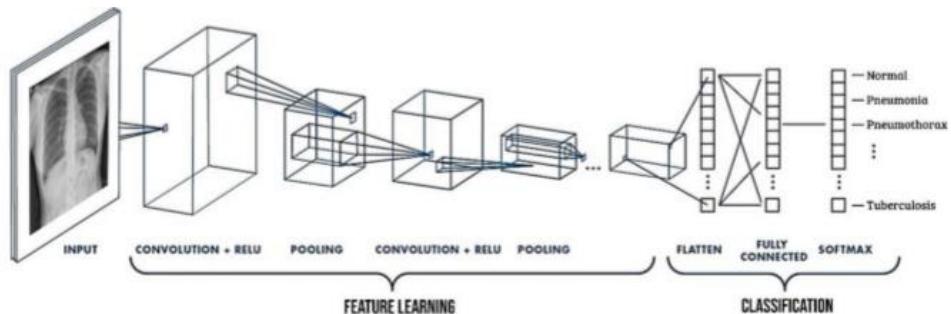
#### 3. Augmantasi Data

Augmantasi adalah teknik manipulasi pada data yang ada tanpa kehilangan inti dari data tersebut. Pada penelitian ini dilakukan teknik augmentasi terhadap data *training* dengan mengatur *rotation range* = 30, *horizontal flip*, *zoom range* = 0,2 , *width* dan *height shift range* = 0,1.

Beberapa cara tersebut membantu model yang akan disusun serta mempermudah dalam melatih model tersebut.

#### 4.2.2 Convolution Neural Network (CNN)

Implementasi metode *Convolution Neural Network* (CNN) terdapat beberapa tahapan mulai dari input data, menentukan pengaturan model yang akan di terapkan sampai pada output dari kelas yang diklasifikasi, serta menentukan nilai epoch untuk melatih data agar menghasilkan hasil yang sesuai. Model arsitektur CNN yang digunakan pada penelitian ini adalah model arsitektur *Sequential*. Berikut ini adalah tahapan implementasi CNN yang peneliti buat :

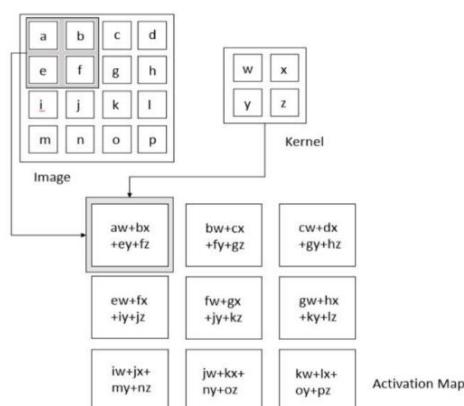


**Gambar 4. 4** Arsitektur *Convolution Neural Network* (CNN)

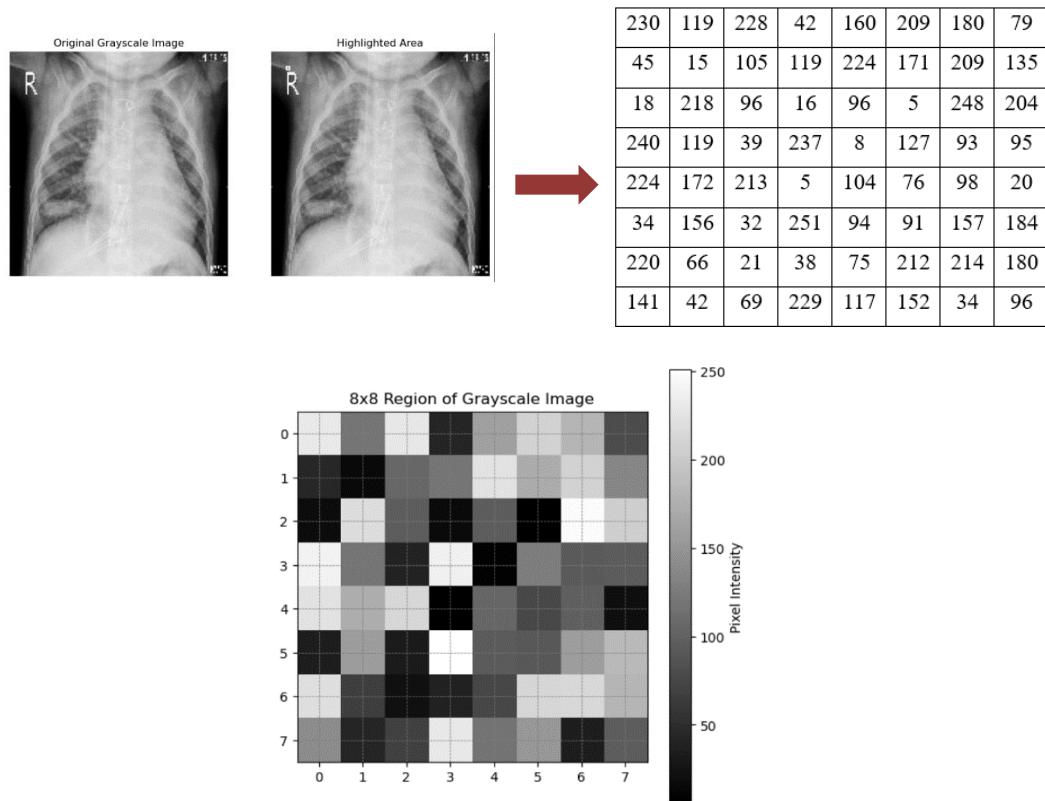
Proses yang pertama dari metode *Convolution Neural Network* (CNN), antara lain:

##### 1. *Convolution Layer*

Pada proses ini akan dilakukan perhitungan antara dua buah matriks, dimana satu matriks adalah parameter yang dipelajari kernel, dan matriks lainnya adalah pixel citra.

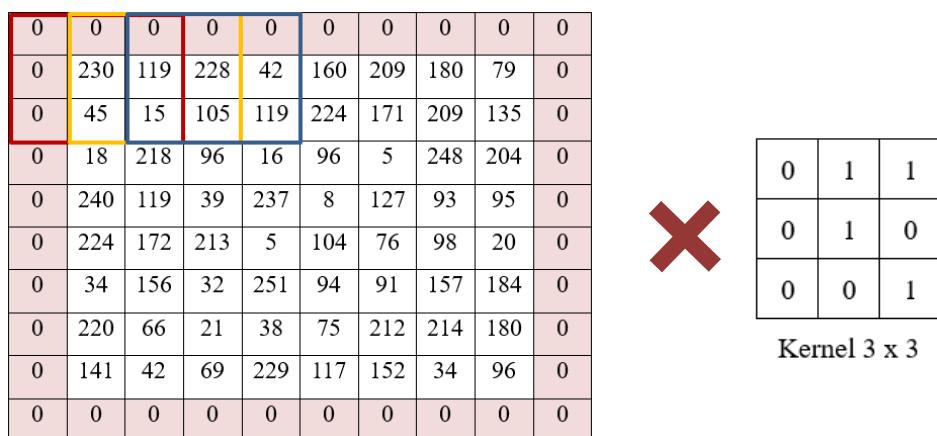


**Gambar 4. 5** Operasi Konvolusi



**Gambar 4.6** Sampel Gambar dan Matriks Citra

Gambar 4.6 adalah gambar input yang menunjukkan dataset yang tersedia. Ukuran pixel dari citra masukan adalah  $224 \times 224$ , sehingga penulis hanya mengambil sebagian dari pixel tersebut yaitu  $8 \times 8$  pixel. Gambar dibawah ini menunjukkan proses konvolusi menggunakan kernel.



**Gambar 4.7** Proses Konvolusi dengan Kernel  $3 \times 3$

Perhitungan manual metode *convolution neural network* pada proses konvolusi dari pixel matriks 150 x 150, diambil contoh dengan ukuran matriks citra = 8 x 8 dengan kernel = 3 x 3, S = 1, P = 1.

Awalnya akan dihitung output volumenya, dengan cara :

$$V = \frac{W - F + 2P}{S} + 1$$

Dimana :

V = Volume

W = Input Size = 8 x 8

F = Kernel Size = 3 x 3

P = Padding = 1

S = Stride = 1

Oleh karena itu, perhitungannya adalah sebagai berikut :

$$Output = \frac{8 - 3 + 2(1)}{1} + 1$$

$$Output = 8.$$

Dari perhitungan diatas, maka diperoleh output dengan ukuran 8 x 8.

Adapun perhitungan yang dilakukan untuk mendapatkan hasil dari Gambar 4.7 adalah sebagai berikut :

- $(0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (230 \times 1) + (119 \times 0) + (0 \times 0) + (45 \times 0) + (15 \times 1) = \mathbf{245}$
- $(0 \times 0) + (0 \times 1) + (0 \times 1) + (230 \times 0) + (119 \times 1) + (228 \times 0) + (45 \times 0) + (15 \times 0) + (105 \times 1) = \mathbf{224}$
- $(0 \times 0) + (0 \times 1) + (0 \times 1) + (119 \times 0) + (228 \times 1) + (42 \times 0) + (15 \times 0) + (105 \times 0) + (119 \times 1) = \mathbf{347}$

Seterusnya cara perhitungan yang dilakukan adalah sama kemudian hasil dari perhitungan diatas diperoleh output konvolusi atau *feature map* dengan ukuran 8 x 8.

245	224	347	266	331	418	315	79
612	458	391	417	598	808	672	214
197	377	557	367	618	478	687	339
648	646	156	453	185	478	565	299
739	362	740	344	330	453	470	115
496	562	288	435	486	479	455	204
452	323	533	500	412	494	651	364
427	129	128	342	404	578	428	276

**Gambar 4. 8 Feature Map**

Setelah mendapatkan output konvolusi atau *feature map*, maka dilakukan *pooling layer* atau mengurangi ukuran dimensi *feature map* sehingga mempercepat komputasi. *Feature map* merupakan pemetaan citra atau intisari citra yang dipetakan dari hasil konvolusi.

## 2. Pooling Layer

Dari hasil proses konvolusi diperoleh *feature map*. Selanjutnya akan dilakukan *pooling layer* menggunakan ukuran filter 2 x 2. Karena menggunakan max pooling maka tiap filter matriks akan diambil nilai paling besar dan akan dimasukkan ke dalam *feature map*. Pada matriks dibawah ini tiap blok warna menandakan tiap pergeseran window dengan matriks 2 x 2. Masing-masing window akan diambil nilai yang paling terbesar, kemudian nilai yang paling besar akan menjadi *feature map* baru.

245	224	347	266	331	418	315	79
612	458	391	417	598	808	672	214
197	377	557	367	618	478	687	339
648	646	156	453	185	478	565	299
739	362	740	344	330	453	470	115
496	562	288	435	486	479	455	204
452	323	533	500	412	494	651	364
427	129	128	342	404	578	428	276

612	417	808	672
648	557	618	687
739	740	486	470
452	533	578	651

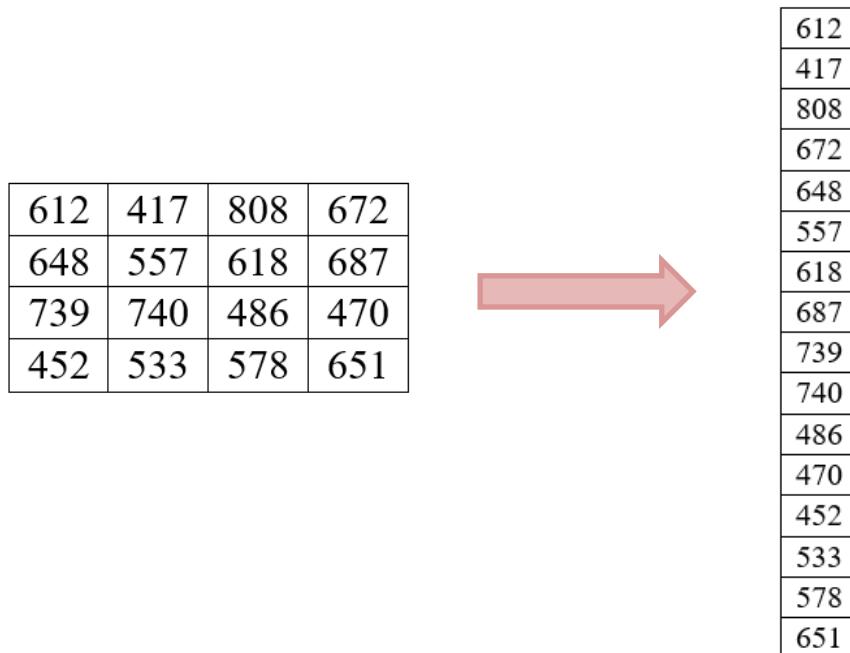
**Gambar 4. 9 Max Pooling**

Dari hasil *pooling layer* didapatkan *feature map* dengan ukuran 4 x 4. Proses konvolusi pada *pooling layer* ini akan terus di ulang sesuai dengan arsitektur model

CNN yang digunakan. Setelah *pooling layer* citra baru akan dilakukan proses *flattening*.

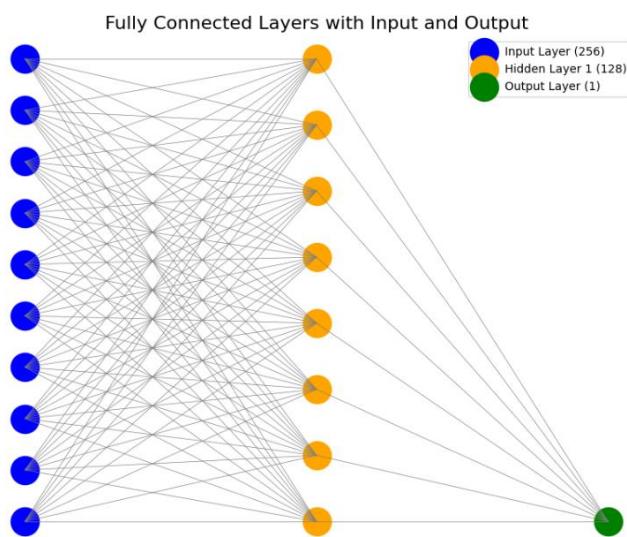
### 3. *Flatten*

Pada tahap ini *feature map* akan dilakukan proses *flattening* atau mengubah matriks menjadi bentuk vector atau satu dimensi.



### 4. *Fully Connected Layer*

*Fully connected layer* berperan untuk mengklasifikasikan data masukkan. Pada proses *training* ini berguna untuk mencari nilai bobot terbaik.



**Gambar 4. 10** *Fully Connected Layer*

1. Inisialisasi semua bobot

Bobot awal atau  $w = 0$ , untuk semua inputan.

2. Fungsi ativasi yang digunakan

Aktivasi yang digunakan pada penelitian ini adalah  $\sigma = \text{Sigmoid}$

3. Laju pembelajaran atau *Learning Rate*

Pada penelitian ini ditentukan nilai *learning rate* nya  $\alpha = 0,000001$

4. Menghitung persamaan nilai output :

$$f(x) = \sum_{x=i}^n x_1 w_1 + x_2 w_2 + \cdots + x_n w_n$$

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

$$f(x) = \sum_{x=i}^n x_1 w_{11} + x_2 w_{21} + \cdots + x_n w_n$$

$$f(x) = \sum_{x=i}^{16} 612.0 + 417.0 + 808.0 + 672.0 + 648.0 + 557.0 + 618.0 \\ + 687.0 + 739.0 + 740.0 + 486.0 + 470.0 + 452.0 + 533.0 \\ + 578.0 + 651.0 = 0$$

$$f(x) = \sum_{x=i}^{16} 612.0 + 417.0 + 808.0 + 672.0 + 648.0 + 557.0 + 618.0 \\ + 687.0 + 739.0 + 740.0 + 486.0 + 470.0 + 452.0 + 533.0 \\ + 578.0 + 651.0 = 0$$

$$f(x) = \sum_{x=i}^{16} 612.0 + 417.0 + 808.0 + 672.0 + 648.0 + 557.0 + 618.0 \\ + 687.0 + 739.0 + 740.0 + 486.0 + 470.0 + 452.0 + 533.0 \\ + 578.0 + 651.0 = 0$$

$$f(x) = \sum_{x=i}^n 612.0 + 417.0 + 808.0 + 672.0 + 648.0 + 557.0 + 618.0 \\ + 687.0 + 739.0 + 740.0 + 486.0 + 470.0 + 452.0 + 533.0 \\ + 578.0 + 651.0 = 0$$

*Sigmoid :*

$$\sigma(a) = \frac{1}{1+e^{-a}} = \frac{1}{1+2.71828^{-0}} = 2 =$$

Nilai sama dengan nilai output lainnya dikarenakan menggunakan nilai w = 0

5. Hitung *gradien error* untuk *layer output*

$$e_k(p) = y_{dk}(p) - y_k(p)$$

$$\delta_k(p) = y_{dk}(p) \times [1 - y_k(p)] \times e_k(p)$$

Untuk data pertama, nilai yang diharapkan adalah  $y_{dk} = 1$ , sedangkan keluaran yang didapatkan  $y_{dk} = 2$ , nilai neuron output semuanya sama karena nilai w = 0.

$$e_1(1) = y_d - y_1(1) = 1 - 2 = -1$$

$$\delta_1(1) = y_1(1) \times [1 - y_1(1)] \times e_1(1) = 2 \times [1 - 2] \times -1 = 2$$

6. Menghitung Koreksi Bobot

Untuk  $\Delta w_{11}, \Delta w_{12}, \Delta w_{13}, \Delta w_{14}$  :

$$\Delta w_{11} = a * y_1(1) * \delta_1(1) = 0,000001 * 612 * 2 = 0,001224$$

$$\Delta w_{12} = a * y_1(1) * \delta_1(1) = 0,000001 * 417 * 2 = 0,000834$$

$$\Delta w_{13} = a * y_1(1) * \delta_1(1) = 0,000001 * 808 * 2 = 0,001616$$

$$\Delta w_{14} = a * y_1(1) * \delta_1(1) = 0,000001 * 672 * 2 = 0,001344$$

7. Perbarui Bobot

Dengan di perolehnya nilai pengoreksi bobot, maka proses memperbarui nilai bobot dengan menggunakan persamaan :

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p)$$

$$w_{jk}(p+1) = w_{11}(p) + \Delta w_{11}(p) = 0 + 0,001224 = 0,001224$$

$$w_{jk}(p+1) = w_{12}(p) + \Delta w_{12}(p) = 0 + 0,000834 = 0,000834$$

Setelah itu untuk iterasi selanjutnya, kembali ke langkah 5 dan diulangi sampai mendapatkan hasil yang sesuai. Setelah itu jika didapatkan hasil yang sesuai, lalu diklasifikasikan citra sesuai dengan kelasnya.

### 4.3 Evaluasi Model

#### 4.3.1 Confusion Matriks

Pada penelitian ini menggunakan confusion matriks sebagai metode untuk menghitung akurasi dalam penerapan pengklasifikasian penyakit *pneumonia*. Evaluasi efektivitas kinerja untuk pengklasifikasian penyakit *pneumonia* berdasarkan jumlah subjek benar dan salah yang terdeteksi dapat dilihat pada tabel dibawah ini :

**Tabel 4. 1** Hasil Data Testing

No.	Image	Klasifikasi	Hipotesis	No.	Image	Klasifikasi	Hipotesis
1.		<i>Pneumonia</i>	<i>Pneumonia</i>	16.		Normal	<i>Pneumonia</i>
2.		<i>Pneumonia</i>	<i>Pneumonia</i>	17.		Normal	Normal
3.		<i>Pneumonia</i>	<i>Pneumonia</i>	18.		Normal	<i>Pneumonia</i>
4.		<i>Pneumonia</i>	<i>Pneumonia</i>	19.		Normal	Normal
5.		<i>Pneumonia</i>	<i>Pneumonia</i>	20.		Normal	Normal

No.	Image	Klasifikasi	Hipotesis	No.	Image	Klasifikasi	Hipotesis
6.		Pneumonia	Pneumonia	21.		Normal	Normal
7.		Pneumonia	Pneumonia	22.		Normal	Pneumonia
8.		Pneumonia	Pneumonia	23.		Normal	Normal
9.		Pneumonia	Pneumonia	24.		Normal	Normal
10.		Pneumonia	Pneumonia	25.		Normal	Normal
11.		Pneumonia	Pneumonia	26.		Normal	Normal
12.		Pneumonia	Pneumonia	27.		Normal	Normal

13.		Pneumonia	Pneumonia	28.		Normal	Normal
14.		Pneumonia	Pneumonia	29.		Normal	Normal
15.		Pneumonia	Pneumonia	30.		Normal	Normal

**Tabel 4. 2** Hasil Klasifikasi pada data Testing

	<b>Pneumonia</b>	<b>Normal</b>
<b>Pneumonia</b>	15	3
<b>Normal</b>	0	12

Data yang diklasifikasikan dengan benar = 27

Data yang diklasifikasikan dengan salah = 3

**Menghitung nilai Accuracy :**

$$\text{Accuracy} = \frac{\text{nl benar}}{\text{jumlah data (n)}} \times 100\%$$

$$\text{Accuracy Pneumonia} = \frac{15}{15} \times 100 \% = 100\%$$

$$\text{Accuracy Normal} = \frac{12}{15} \times 100 \% = 80\%$$

$$Rata-rata Accuracy = \frac{15 + 12}{15 + 3 + 0 + 12} \times 100 \% = \frac{27}{30} \times 100 \% = 90 \%$$

Menghitung nilai TP, TN, FP, FN sesuai dengan kelas masing-masing :

- *Pneumonia* : TP = 15      FN = 0      TN = 0      FP = 0
- Normal : TP = 12      FN = 0      TN = 0      FP = 3

Persamaan *precision* dan *recall* :

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

**Menghitung nilai *precision* dan *recall pneumonia* dan normal :**

*Pneumonia* :

$$Precision = \frac{TP}{(TP+FP)} = \frac{15}{(15+0)} = \frac{15}{15} = 1 = 100\%$$

$$Recall = \frac{TP}{(TP+FN)} = \frac{15}{(15+0)} = \frac{15}{15} = 1 = 100\%$$

Normal :

$$Precision = \frac{TP}{(TP+FP)} = \frac{12}{(12+3)} = \frac{12}{15} = 0,8 = 80\%$$

$$Recall = \frac{TP}{(TP + FN)} = \frac{12}{(12 + 0)} = \frac{12}{12} = 1 = 100\%$$

$$Rata-rata Precision = \frac{100 + 80}{2} = \frac{180}{2} = 90\%$$

$$Rata-rata Recall = \frac{100 + 100}{2} = \frac{200}{2} = 10$$

**Tabel 4. 3** Hasil *Confussion Matriks*

Klasifikasi	TP	TN	FP	FN	Hasil %		
					Akurasi	Precision	Recall
Pneumonia	15	0	0	0	100	100	100
Normal	12	0	3	0	80	80	100
Rata-rata					90	90	100

### ***F1-Score***

$$f1-score = 2 \times \frac{(recall \times precision)}{(recall + precision)}$$

$$f1-score = 2 \times \frac{(100 \times 90)}{(100 + 90)} = 2 \times \frac{9000}{190} = 2 \times 47,36 = 94.72 \text{ %}.$$

## BAB V

### PEMBAHASAN

#### 5.1 Objek Penelitian

Data yang digunakan dalam penelitian ini terdiri dari data public dengan nama GAMBAR RONTGEN DADA (*PNEUMONIA*), data yang diambil terdiri atas 2 kategori kelas yaitu normal dan *pneumonia*. Dataset ini diunduh dari website <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia> .

Masing-masing dataset dibagi menjadi 5216 data training dan 624 data testing.

#### 5.2 Pembahasan Model

Berdasarkan hasil yang diperoleh dalam penelitian yang telah dipaparkan pada bab sebelumnya, beberapa hal berikut dapat dijelaskan secara lebih mendetail:

##### 5.2.1 Data Citra X-ray Paru-paru

Penelitian ini berhasil mengumpulkan sebanyak 5.856 gambar citra X-ray paru-paru yang kemudian dibagi menjadi dua kelas, yaitu kelas normal dan *pneumonia*. Berdasarkan visualisasi citra yang dianalisis, kedua kelas tersebut dapat dengan jelas dibedakan. Gambar citra X-ray paru-paru memberikan gambaran yang cukup jelas mengenai kondisi kesehatan paru-paru, yang memungkinkan model untuk mengenali perbedaan mendasar antara gambar paru-paru yang sehat dan yang menunjukkan indikasi *pneumonia*.

##### 5.2.2 Ekstraksi Fitur dengan Metode (CNN)

Pada percobaan yang dilakukan dalam penelitian ini, fitur-fitur citra diekstraksi menggunakan metode CNN (*Convolutional Neural Network*), yang merupakan pendekatan yang sangat efisien karena tidak memerlukan metode ekstraksi fitur tambahan lainnya. CNN secara otomatis melakukan ekstraksi fitur melalui proses konvolusi yang diterapkan pada setiap layer. Proses konvolusi ini diikuti oleh layer pooling, yang bertujuan untuk mengurangi ukuran piksel yang diproses dan memudahkan pengolahan data. Penggunaan layer pooling dalam setiap blok bertujuan untuk mengurangi kompleksitas data, mempercepat waktu komputasi, dan memastikan bahwa algoritma CNN yang digunakan menjadi lebih

efisien. Hal ini juga membuat algoritma lebih optimal untuk diterapkan dalam aplikasi yang akan dikembangkan di masa mendatang, mengingat pengurangan dimensi data yang terjaga tanpa mengorbankan informasi penting yang terkandung dalam citra.

### 5.2.3 Arsitektur *Convolution Neural Network (CNN)*

Arsitektur CNN dalam penelitian ini dibangun dengan enam blok, yang terdiri dari berbagai jenis lapisan yang bekerja secara berurutan untuk mengekstraksi dan memproses fitur citra.

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 150, 150, 32)	320
batch_normalization (BatchNormalization)	(None, 150, 150, 32)	128
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_1 (Conv2D)	(None, 75, 75, 64)	18,496
dropout (Dropout)	(None, 75, 75, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 75, 75, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_2 (Conv2D)	(None, 37, 37, 64)	36,928
batch_normalization_2 (BatchNormalization)	(None, 37, 37, 64)	256
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 64)	0
conv2d_3 (Conv2D)	(None, 18, 18, 128)	73,856
dropout_1 (Dropout)	(None, 18, 18, 128)	0
batch_normalization_3 (BatchNormalization)	(None, 18, 18, 128)	512
max_pooling2d_3 (MaxPooling2D)	(None, 9, 9, 128)	0
conv2d_4 (Conv2D)	(None, 9, 9, 256)	295,168
dropout_2 (Dropout)	(None, 9, 9, 256)	0
batch_normalization_4 (BatchNormalization)	(None, 9, 9, 256)	1,024
max_pooling2d_4 (MaxPooling2D)	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 128)	524,416
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129

Total params: 951,489 (3.63 MB)  
 Trainable params: 950,481 (3.63 MB)  
 Non-trainable params: 1,088 (4.25 KB)

**Gambar 5. 1** Arsitektur Model CNN

Arsitektur jaringan saraf konvolusional (CNN) pada gambar 5.1 mengikuti pendekatan terstruktur untuk tugas klasifikasi gambar, dengan fokus pada diagnosis *pneumonia* menggunakan pembelajaran mendalam. Model dimulai dengan lapisan konvolusional pertama (conv2d) yang menggunakan 32 filter untuk mengekstraksi fitur dari gambar input berukuran 150 x 150, menghasilkan output dengan dimensi yang sama namun dengan 32 saluran. Peta fitur yang dihasilkan kemudian dinormalisasi dengan menggunakan *batch normalization*, yang berfungsi untuk menstabilkan proses pelatihan dengan mengurangi pergeseran kovariat internalutnya, jaringan melanjutkan dengan serangkaian lapisan konvolusional dan lapisan pooling. Setelah blok konvolusional pertama, lapisan *max-pooling* (*max\_pooling2d*) diterapkan untuk mengurangi dimensi spasial dari peta fitur. Proses pooling ini mengurangi kompleksitas komputasi dan mencegah overfitting dengan hanya mempertahankan fitur penting dan mengabaikan informasi yang kurang relevan. Di setiap blok, lapisan *dropout* diterapkan untuk mengatur sebagian unit input menjadi nol secara acak selama pelatihan, bertujuan untuk mengurangi overfitting .

Arsitektur ini juga mengandung beberapa lapisan konvolusional tambahan dengan ukuran filter yang semakin besar, mulai dari 64 hingga 256. Operasi pooling lanjutan dilakukan untuk mengurangi dimensi data lebih lanjut. Lapisan-lapisan konvolusional ini bertujuan untuk menangkap fitur tingkat tinggi dari gambar, seperti tekstur dan tepi, yang penting untuk membedakan gambar paru-paru yang sehat dari yang terinfeksi *pneumonia*. Dalam tahap akhir, peta fitur yang telah diekstraksi diratakan menjadi vektor satu dimensi yang disiapkan untuk diproses oleh lapisan *fully connected* (*dense*). Lapisan terakhir dari model ini, yaitu lapisan *dense*, menghasilkan satu unit yang mewakili probabilitas klasifikasi akhir .

Model ini melibatkan 951.489 parameter, dengan sebagian besar di antaranya merupakan parameter yang dapat dilatih. Lapisan *dropout* yang diterapkan tidak menambah jumlah parameter, tetapi berperan dalam mencegah overfitting dengan melakukan regularisasi selama pelatihan. Desain ini menunjukkan bahwa model cukup kompleks dan mampu mempelajari pola yang rumit dalam data gambar,

sekaligus mengurangi overfitting dengan menerapkan teknik regularisasi seperti dropout dan batch normalization .

#### 5.2.4 Proses Konvolusi dan Pooling

Proses konvolusi dan pooling yang diterapkan pada gambar citra menghasilkan nilai matriks citra yang selanjutnya digunakan sebagai variabel input pada algoritma klasifikasi yang diterapkan dalam penelitian ini. Matriks hasil ekstraksi ini mengandung informasi yang dibutuhkan untuk menentukan kelas dari citra tersebut, apakah normal atau menunjukkan indikasi *pneumonia*.

#### 5.2.5 Proses Pelatihan Model

```

Epoch 1/50
74/74  119s 1s/step - accuracy: 0.8221 - loss: 1.5562 - val_accuracy: 0.7389 - val_loss: 1.9079 - learning_rate: 0.0010
Epoch 2/50
74/74  105s 1s/step - accuracy: 0.9138 - loss: 0.2359 - val_accuracy: 0.7389 - val_loss: 4.6704 - learning_rate: 0.0010
Epoch 3/50
74/74  0s 1s/step - accuracy: 0.9485 - loss: 0.1719
Epoch 3: ReduceLROnPlateau reducing learning rate to 0.0003000000142492354.
74/74  102s 1s/step - accuracy: 0.9485 - loss: 0.1719 - val_accuracy: 0.7389 - val_loss: 4.4035 - learning_rate: 0.0010
Epoch 4/50
74/74  109s 1s/step - accuracy: 0.9478 - loss: 0.1746 - val_accuracy: 0.7389 - val_loss: 6.4345 - learning_rate: 3.0000e-04
Epoch 5/50
74/74  0s 1s/step - accuracy: 0.9754 - loss: 0.0707
Epoch 5: ReduceLROnPlateau reducing learning rate to 0.000000427477062e-05.
74/74  101s 1s/step - accuracy: 0.9754 - loss: 0.0707 - val_accuracy: 0.7389 - val_loss: 4.7632 - learning_rate: 3.0000e-04
Epoch 6/50
74/74  105s 1s/step - accuracy: 0.9816 - loss: 0.0495 - val_accuracy: 0.7389 - val_loss: 4.1971 - learning_rate: 9.0000e-05
Epoch 7/50
74/74  106s 1s/step - accuracy: 0.9871 - loss: 0.0446 - val_accuracy: 0.7406 - val_loss: 2.3349 - learning_rate: 9.0000e-05
Epoch 8/50
74/74  123s 2s/step - accuracy: 0.9883 - loss: 0.0375 - val_accuracy: 0.8413 - val_loss: 0.6887 - learning_rate: 9.0000e-05
Epoch 9/50
74/74  112s 2s/step - accuracy: 0.9901 - loss: 0.0326 - val_accuracy: 0.9334 - val_loss: 0.2789 - learning_rate: 9.0000e-05
Epoch 10/50
74/74  110s 1s/step - accuracy: 0.9943 - loss: 0.0225 - val_accuracy: 0.9539 - val_loss: 0.1202 - learning_rate: 9.0000e-05
Epoch 11/50
74/74  90s 1s/step - accuracy: 0.9937 - loss: 0.0193 - val_accuracy: 0.9522 - val_loss: 0.1751 - learning_rate: 9.0000e-05
Epoch 12/50
74/74  86s 1s/step - accuracy: 0.9943 - loss: 0.0177 - val_accuracy: 0.9573 - val_loss: 0.0886 - learning_rate: 9.0000e-05
Epoch 13/50
74/74  90s 1s/step - accuracy: 0.9954 - loss: 0.0136 - val_accuracy: 0.9556 - val_loss: 0.0846 - learning_rate: 9.0000e-05
Epoch 14/50
74/74  74s 995ms/step - accuracy: 0.9949 - loss: 0.0137 - val_accuracy: 0.9625 - val_loss: 0.0871 - learning_rate: 9.0000e-05
Epoch 35/50
74/74  92s 1s/step - accuracy: 0.9994 - loss: 0.0030 - val_accuracy: 0.9625 - val_loss: 0.1007 - learning_rate: 1.0000e-06
Epoch 36/50
74/74  79s 1s/step - accuracy: 0.9989 - loss: 0.0035 - val_accuracy: 0.9625 - val_loss: 0.1001 - learning_rate: 1.0000e-06
Epoch 37/50
74/74  81s 1s/step - accuracy: 0.9988 - loss: 0.0045 - val_accuracy: 0.9625 - val_loss: 0.1004 - learning_rate: 1.0000e-06
Epoch 38/50
74/74  82s 1s/step - accuracy: 0.9999 - loss: 0.0031 - val_accuracy: 0.9625 - val_loss: 0.0999 - learning_rate: 1.0000e-06
Epoch 39/50
74/74  92s 1s/step - accuracy: 0.9993 - loss: 0.0033 - val_accuracy: 0.9625 - val_loss: 0.1007 - learning_rate: 1.0000e-06
Epoch 40/50
74/74  93s 1s/step - accuracy: 1.0000 - loss: 0.0020 - val_accuracy: 0.9625 - val_loss: 0.1015 - learning_rate: 1.0000e-06
Epoch 41/50
74/74  89s 1s/step - accuracy: 0.9988 - loss: 0.0059 - val_accuracy: 0.9625 - val_loss: 0.1016 - learning_rate: 1.0000e-06
Epoch 42/50
74/74  79s 1s/step - accuracy: 0.9997 - loss: 0.0039 - val_accuracy: 0.9625 - val_loss: 0.1023 - learning_rate: 1.0000e-06
Epoch 43/50
74/74  80s 1s/step - accuracy: 0.9976 - loss: 0.0067 - val_accuracy: 0.9625 - val_loss: 0.1018 - learning_rate: 1.0000e-06
Epoch 44/50
74/74  86s 1s/step - accuracy: 0.9998 - loss: 0.0025 - val_accuracy: 0.9625 - val_loss: 0.1017 - learning_rate: 1.0000e-06
Epoch 45/50
74/74  83s 1s/step - accuracy: 0.9986 - loss: 0.0047 - val_accuracy: 0.9625 - val_loss: 0.1006 - learning_rate: 1.0000e-06
Epoch 46/50
74/74  79s 1s/step - accuracy: 0.9993 - loss: 0.0044 - val_accuracy: 0.9625 - val_loss: 0.1012 - learning_rate: 1.0000e-06
Epoch 47/50
74/74  76s 1s/step - accuracy: 0.9982 - loss: 0.0042 - val_accuracy: 0.9625 - val_loss: 0.1013 - learning_rate: 1.0000e-06
Epoch 48/50
74/74  83s 1s/step - accuracy: 0.9985 - loss: 0.0050 - val_accuracy: 0.9625 - val_loss: 0.1023 - learning_rate: 1.0000e-06
Epoch 49/50
74/74  79s 1s/step - accuracy: 0.9990 - loss: 0.0036 - val_accuracy: 0.9625 - val_loss: 0.1031 - learning_rate: 1.0000e-06
Epoch 50/50
74/74  76s 1s/step - accuracy: 1.0000 - loss: 0.0023 - val_accuracy: 0.9625 - val_loss: 0.1015 - learning_rate: 1.0000e-06

```

**Gambar 5. 2** Training Dataset dengan 50 Epoch

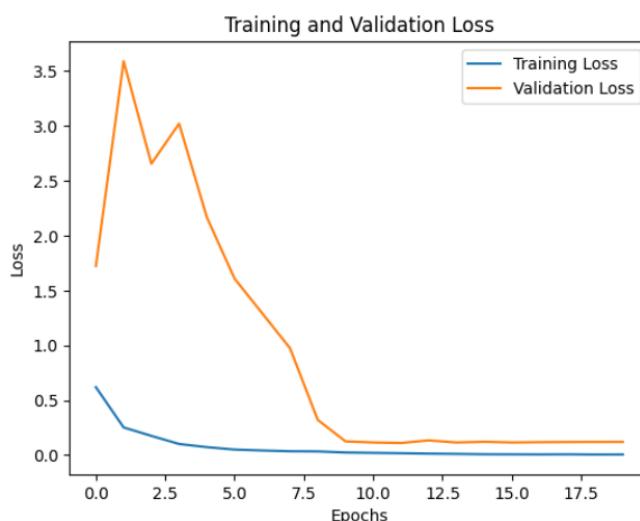
Gambar 5.2 menunjukkan hasil pelatihan model *neural network* yang dilakukan selama 50 epoch. Setiap epoch mencatat informasi terkait akurasi dan

loss baik untuk data pelatihan maupun data validasi. Pada awal pelatihan, akurasi pada data pelatihan mulai dari sekitar 82% dan meningkat secara bertahap hingga mencapai lebih dari 99%, sementara akurasi pada data validasi juga menunjukkan peningkatan yang stabil. Loss pada data pelatihan dan validasi secara konsisten menurun sepanjang pelatihan, menandakan bahwa model semakin baik dalam memprediksi kelas yang benar.

Model ini menggunakan teknik *ReduceLROnPlateau*, yang secara otomatis mengurangi *learning rate* ketika model tidak menunjukkan peningkatan signifikan. Hal ini terlihat pada beberapa epoch di mana nilai *learning rate* berkurang, yang menunjukkan bahwa teknik ini diterapkan untuk mengoptimalkan proses pelatihan. Penurunan *learning rate* ini berfungsi untuk meningkatkan stabilitas pelatihan dan membantu model mencapai konvergensi yang lebih baik.

Model ini menunjukkan performa yang sangat baik, dengan akurasi yang meningkat dan loss yang menurun secara signifikan, baik pada data pelatihan maupun data validasi. Tidak ada tanda-tanda overfitting yang terlihat, karena perbedaan antara akurasi data pelatihan dan validasi sangat kecil, menunjukkan bahwa model tidak hanya belajar dengan baik dari data pelatihan tetapi juga dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Dengan demikian, proses pelatihan berjalan efektif dan efisien, menghasilkan model yang dapat melakukan klasifikasi dengan tingkat akurasi yang tinggi.

### 5.2.6 Training and Validation Loss



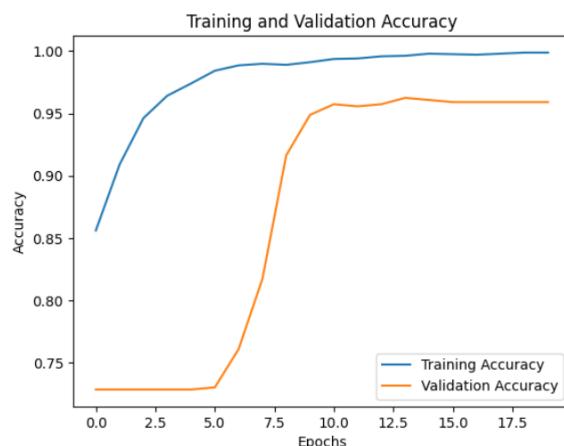
**Gambar 5. 3** Grafik Training dan Validation Loss

Gambar 5.3 menunjukkan grafik yang menggambarkan perubahan training loss dan validation loss selama proses pelatihan model yang diukur sepanjang 20 epoch. Pada sumbu horizontal, terdapat jumlah epoch yang digunakan dalam pelatihan, sementara pada sumbu vertikal, diukur nilai loss yang dihasilkan oleh model baik untuk data pelatihan (training loss) maupun data validasi (validation loss).

Pada awal pelatihan terlihat bahwa training loss turun dengan tajam hampir mencapai nilai yang sangat rendah setelah beberapa epoch pertama, yang menunjukkan bahwa model belajar dengan cepat untuk meminimalkan error pada data pelatihan. Sebaliknya, validation loss mengalami fluktuasi yang lebih besar di awal pelatihan sebelum akhirnya stabil pada nilai yang lebih rendah setelah beberapa epoch. Fluktuasi pada validation loss mungkin menunjukkan adanya variasi dalam performa model saat diterapkan pada data yang tidak terlihat selama pelatihan.

Setelah epoch ke-10, baik training loss maupun validation loss menunjukkan penurunan yang sangat lambat, dengan kedua kurva cenderung mendekati nilai konstan mendekati 0. Hal ini mengindikasikan bahwa model telah mencapai titik di mana ia tidak lagi mengalami peningkatan signifikan dalam hal kemampuannya untuk meminimalkan error, baik pada data pelatihan maupun pada data validasi. Pada tahap ini, pelatihan model mulai mencapai konvergensi, yang dapat menunjukkan bahwa model sudah cukup baik dalam mengklasifikasikan data.

### 5.2.7 *Training dan Validation Akurasi*



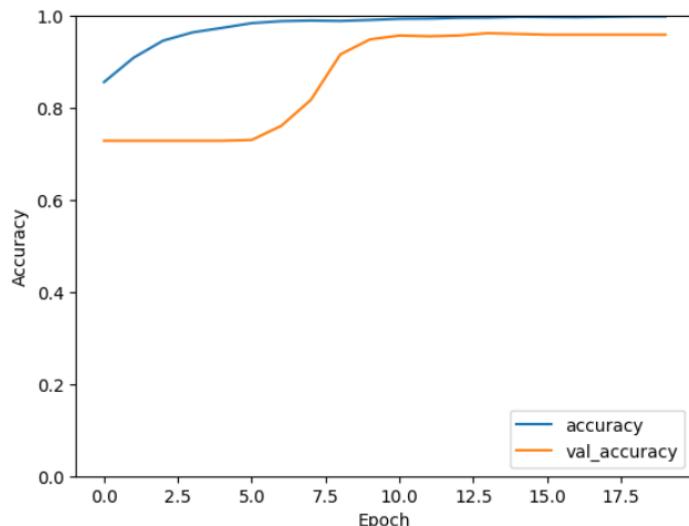
**Gambar 5.4** Grafik Training dan Validasi Akurasi

Gambar 5.4 menunjukkan grafik yang menggambarkan training accuracy dan validation accuracy model sepanjang 20 epoch. Pada awal pelatihan, training accuracy meningkat secara tajam, hampir mencapai 100% dalam beberapa epoch pertama. Peningkatan yang cepat ini menunjukkan bahwa model dapat dengan cepat mempelajari pola-pola dalam data pelatihan. Namun, meskipun training accuracy hampir mencapai 1.0, validation accuracy mengalami peningkatan yang lebih lambat. Hal ini terlihat dari fluktuasi yang lebih besar pada awalnya, yang mengindikasikan bahwa model membutuhkan waktu untuk menggeneralisasi dengan baik pada data yang tidak dilihat sebelumnya.

Setelah beberapa epoch pertama, sekitar epoch ke-10, training accuracy mencapai hampir nilai 1.0. Hal ini menunjukkan bahwa model sangat baik dalam mempelajari data pelatihan. Sementara itu, validation accuracy mengalami peningkatan yang lebih stabil. Meskipun ada fluktuasi minor, nilai akurasi pada data validasi terus meningkat mendekati angka 0.95 setelah beberapa epoch. Perbedaan antara training accuracy dan validation accuracy ini mengindikasikan bahwa meskipun model sangat efisien dalam mempelajari data pelatihan, ia juga mampu mempertahankan kinerjanya dengan baik pada data yang tidak terlihat.

Pada titik sekitar epoch ke-15 hingga ke-50, baik training accuracy maupun validation accuracy cenderung stabil. Training accuracy tetap berada di dekat 1.0, sementara validation accuracy stabil di sekitar 0.95. Hal ini menunjukkan bahwa model telah mencapai konvergensi. Meskipun ada sedikit fluktuasi dalam validation accuracy, perbedaan antara training accuracy dan validation accuracy tetap kecil yang menunjukkan bahwa model tidak mengalami overfitting, karena model mampu menggeneralisasi dengan baik pada data yang tidak dilatih tanpa kehilangan kinerja pada data pelatihan.

### 5.2.8 Accuracy dan Validation Accuracy



**Gambar 5. 5** Grafik Acuraccy dan Validation Accuracy

Gambar 5.5 menunjukkan grafik yang menggambarkan perkembangan accuracy dan validation accuracy sepanjang 20 epoch pelatihan. Pada sumbu horizontal, tercatat jumlah epoch yang digunakan dalam pelatihan, sementara pada sumbu vertikal tercatat nilai akurasi yang dicapai oleh model pada data pelatihan (accuracy) dan data validasi (val\_accuracy).

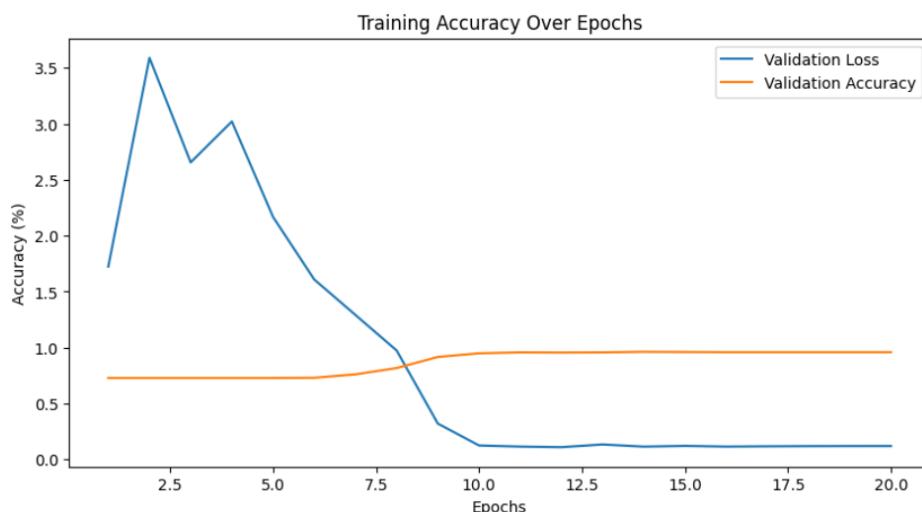
Pada awal pelatihan, grafik menunjukkan peningkatan yang tajam pada *accuracy* yang menunjukkan bahwa model dapat dengan cepat mempelajari pola-pola dalam data pelatihan. *Accuracy* meningkat dengan signifikan dalam beberapa epoch pertama mencapai nilai yang hampir 1.0. Di sisi lain *validation accuracy* menunjukkan pola yang berbeda karena meningkat lebih lambat dan relatif datar pada awal pelatihan namun tetap mengikuti tren yang stabil. Setelah beberapa epoch pertama, *validation accuracy* juga mulai mendekati angka yang cukup tinggi meskipun tidak secepat *accuracy* pada data pelatihan.

Perbedaan yang jelas antara kedua kurva terdapat pada beberapa epoch pertama yang menunjukkan bahwa meskipun model cepat mempelajari data pelatihan, ia memerlukan lebih banyak waktu untuk menggeneralisasi secara efektif pada data yang tidak dilihat sebelumnya. Namun, setelah beberapa epoch *validation accuracy* terus meningkat dan mendekati nilai stabil yang cukup tinggi meskipun

sedikit lebih rendah dibandingkan dengan *accuracy* pada data pelatihan. Pada titik tertentu, kedua kurva menjadi hampir paralel dengan *accuracy* mendekati 1.0 dan *validation accuracy* stabil di sekitar 0.9.

Sedikit perbedaan antara *accuracy* dan *validation accuracy* yang tercatat pada akhir pelatihan menunjukkan bahwa model berhasil meminimalkan kesalahan dalam klasifikasi tanpa menunjukkan tanda-tanda overfitting. Hal ini menunjukkan bahwa meskipun model telah mengoptimalkan akurasinya pada data pelatihan, ia tetap dapat menggeneralisasi dengan baik pada data yang tidak dilihat sebelumnya. Keberhasilan model dalam mencapai akurasi yang tinggi pada kedua set data menunjukkan bahwa model telah dilatih dengan baik dan dapat diandalkan untuk prediksi yang akurat dalam klasifikasi *pneumonia*.

### 5.2.9 Training Accuracy Over Epoch



**Gambar 5. 6** Grafik Training Accuracy Over Epoch

Gambar 5.6 menggambarkan perubahan *training accuracy* dan *validation accuracy* selama 20 epoch pelatihan bersama dengan grafik *validation loss*. Pada sumbu horizontal, tercatat jumlah epoch yang digunakan dalam pelatihan, sementara pada sumbu vertikal tercatat nilai akurasi dan loss yang dicapai oleh model pada data pelatihan dan data validasi.

Pada awal pelatihan, grafik *training accuracy* menunjukkan peningkatan yang sangat tajam mencapai hampir 6% dalam beberapa epoch pertama.

Peningkatan yang tajam ini menunjukkan bahwa model dengan cepat berhasil mempelajari pola dari data pelatihan. Namun pada saat yang sama, *validation accuracy* tetap sangat rendah dan hampir mendekati sekitar angka 0%. Hal ini menunjukkan bahwa meskipun model sangat cepat dalam belajar dari data pelatihan, ia belum dapat menggeneralisasi dengan baik pada data validasi yang tidak dilihat sebelumnya yang menandakan bahwa adanya masalah dalam pengoptimalan atau ketidaksesuaian antara data pelatihan dan data validasi.

Setelah epoch ke-10, *training accuracy* mencapai puncaknya tetapi tetap berada pada tingkat yang relatif rendah. Sementara *validation accuracy* terus menunjukkan angka yang sangat rendah. *Validation loss* tercatat memiliki nilai yang tinggi pada awal pelatihan yang menandakan bahwa model mengalami kesulitan dalam mengurangi kesalahan pada data validasi. Pada epoch-epoch berikutnya, *training accuracy* terus menunjukkan penurunan yang tajam mendekati 1%, sementara *validation accuracy* tetap stagnan. Di sisi lain, *validation loss* menunjukkan penurunan yang sangat lambat yang dapat menunjukkan bahwa meskipun model berhasil meminimalkan kesalahan pada data pelatihan, ia belum berhasil memperbaiki prediksinya pada data yang tidak dilihat sebelumnya.

Ketidakcocokan yang jelas antara *training accuracy* yang meningkat tajam dan *validation accuracy* yang stagnan di awal pelatihan mengindikasikan adanya potensi overfitting pada model. Overfitting terjadi ketika model terlalu fokus pada data pelatihan dan gagal untuk menggeneralisasi dengan baik pada data yang tidak dilihat sebelumnya. Meskipun model mencapai nilai akurasi yang sangat tinggi pada data pelatihan, ketidakmampuan model untuk meningkatkan *validation accuracy* menunjukkan bahwa model tidak dapat menerapkan pembelajaran yang telah diperolehnya pada data validasi.

### 5.2.10 Evaluasi Model

	precision	recall	f1-score	support
PNEUMONIA	0.97	0.97	0.97	2143
NORMAL	0.92	0.92	0.92	785
accuracy			0.96	2928
macro avg	0.95	0.95	0.95	2928
weighted avg	0.96	0.96	0.96	2928

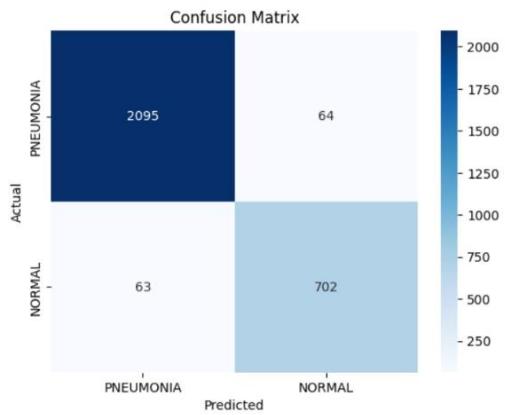
**Gambar 5. 7** Hasil Evaluasi Kinerja Model

Gambar 5.7 menunjukkan hasil evaluasi kinerja model klasifikasi dalam bentuk *classification report* yang mencakup metrik-metrik penting seperti precision, recall, f1-score, dan support untuk masing-masing kelas (*pneumonia* dan *normal*). Pada kelas *pneumonia*, model mencapai precision sebesar 0.97 yang menunjukkan bahwa dari semua prediksi yang diklasifikasikan sebagai *pneumonia*, 97% di antaranya benar-benar merupakan kasus *pneumonia*. Recall pada kelas ini juga mencapai 0.97 yang berarti bahwa dari seluruh kasus *pneumonia* yang sebenarnya, model berhasil mendeteksi 97% di antaranya. F1-score yang dihitung sebagai rata-rata harmonis antara precision dan recall, menghasilkan nilai 0.97 yang menunjukkan bahwa model memiliki keseimbangan yang sangat baik antara keduanya. Angka *support* sebesar 2143 mencerminkan jumlah total kasus *pneumonia* yang ada dalam dataset yang digunakan untuk evaluasi.

Pada kelas *normal*, model menunjukkan precision sebesar 0.92 yang mengindikasikan bahwa 92% dari prediksi yang dikategorikan sebagai *normal* benar-benar merupakan citra *normal*. Recall pada kelas ini adalah 0.92 yang menunjukkan bahwa model berhasil mendeteksi 92% dari semua citra *normal* yang ada. F1-score yang tercatat adalah sebesar 0.92 yang menunjukkan keseimbangan yang baik meskipun sedikit lebih rendah dibandingkan dengan kelas *pneumonia*. Support untuk kelas *normal* adalah 785 yang menunjukkan jumlah total citra *normal* yang ada dalam dataset evaluasi.

Pada bagian *accuracy* yang digunakan untuk mengukur seberapa sering model melakukan prediksi yang benar pada seluruh dataset, tercatat nilai 0.96 yang berarti bahwa model berhasil mengklasifikasikan citra dengan benar 96% dari total

2928 citra. Hal ini merupakan metrik umum yang menunjukkan kinerja keseluruhan model.



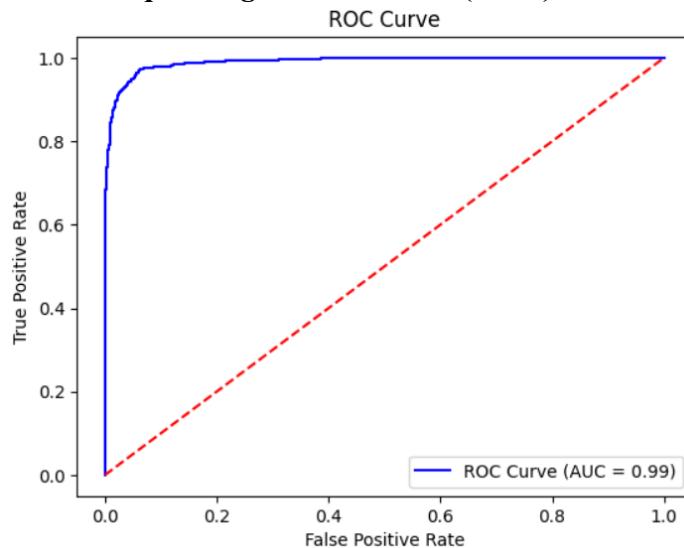
**Gambar 5.8** Confusion Matriks

Gambar 5.8 menunjukkan hasil confusion matrix yang digunakan untuk mengevaluasi kinerja model dalam klasifikasi dua kelas yaitu *pneumonia* dan normal. Matriks ini menunjukkan perbandingan antara kelas aktual dan kelas yang diprediksi oleh model, dengan tujuan untuk menilai seberapa baik model dalam memisahkan kedua kelas tersebut. Matriks ini terdiri dari empat kuadran yang menggambarkan jumlah prediksi yang benar dan salah dari kedua kelas tersebut.

Pada kuadran pertama (*Pneumonia* - *Pneumonia*), angka 2095 menunjukkan jumlah *true positives* (TP) yang mengindikasikan bahwa model berhasil dengan tepat mengklasifikasikan citra sebagai *pneumonia* sesuai dengan kenyataan. Kuadran kedua (Normal - *Pneumonia*) menunjukkan angka 64 yang menggambarkan *false positives* (FP). Hal ini menunjukkan bahwa model mengalami kekeliruan dalam mengklasifikasikan citra yang sebenarnya normal sebagai *pneumonia* yang meningkatkan tingkat kesalahan positif dalam prediksi model. Pada kuadran ketiga (*Pneumonia* - Normal) terdapat angka 63 yang mencerminkan *false negatives* (FN). Hal ini menunjukkan bahwa model gagal dalam mengenali citra yang sebenarnya *pneumonia* sebagai *pneumonia*, yang merupakan kesalahan yang signifikan dalam konteks medis. Di mana kasus *pneumonia* yang tidak terdeteksi dapat berdampak serius pada diagnosis dan pengobatan. Kuadran keempat (Normal - Normal) menunjukkan angka 702 yang

menggambarkan *true negatives* (TN) yaitu jumlah citra normal yang berhasil diklasifikasikan dengan tepat oleh model.

### 5.2.11 Kurva *Receiver Operating Characteristic* (ROC)



**Gambar 5.9** Kurva ROC

Gambar 5.9 menunjukkan menunjukkan *Receiver Operating Characteristic (ROC) curve* yang digunakan untuk mengevaluasi kinerja model klasifikasi dalam memisahkan dua kelas. Dalam konteks ini, ROC curve menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR), yang keduanya sangat penting dalam menilai kemampuan model untuk melakukan prediksi yang benar dan menghindari kesalahan. Pada sumbu horizontal, terdapat *False Positive Rate* (FPR), yang menggambarkan rasio antara jumlah kasus yang salah diklasifikasikan sebagai positif (*false positives*) terhadap total jumlah kasus negatif. Sementara itu, pada sumbu vertikal terdapat *True Positive Rate* (TPR), yang menunjukkan rasio antara jumlah kasus yang benar diklasifikasikan sebagai positif (*true positives*) terhadap total jumlah kasus yang benar-benar positif.

Kurva yang digambarkan pada gambar 5.9 menunjukkan performa model dalam mengklasifikasikan dua kelas. Garis *dashed red* yang menunjukkan garis acuan ideal menggambarkan performa model yang tidak lebih baik daripada tebakan acak, yaitu situasi di mana model tidak memberikan nilai prediksi yang

berguna, dan TPR serta FPR memiliki hubungan linier 1:1. Garis ini berfungsi sebagai referensi untuk mengevaluasi kinerja model yang sebenarnya.

Kurva *blue line* yang diperoleh dari model menunjukkan performa model yang sangat baik dengan TPR yang tinggi (dekat dengan 1) untuk nilai FPR yang rendah. Ini berarti bahwa model berhasil mendeteksi hampir semua kasus positif tanpa banyak menghasilkan kesalahan positif. Secara khusus, pada awal kurva, terdapat peningkatan tajam pada TPR ketika FPR masih sangat rendah, yang menandakan bahwa model mampu menghasilkan banyak prediksi positif yang benar sambil menjaga jumlah kesalahan yang sangat sedikit.

AUC (*Area Under the Curve*) pada gambar 5.9 adalah 0.99 yang menunjukkan bahwa model memiliki kinerja yang sangat baik. Nilai AUC yang mendekati 1 mengindikasikan bahwa model memiliki kemampuan yang sangat tinggi dalam memisahkan dua kelas dengan sedikit kesalahan dalam membedakan antara kelas positif dan kelas negatif. AUC yang lebih tinggi dari 0.9 menunjukkan bahwa model ini tidak hanya lebih baik daripada tebakan acak, tetapi juga sangat efisien dalam melakukan klasifikasi.

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dan pembahasan yang telah diuraikan sebelumnya, maka dapat disimpulkan :

- 1) Penelitian ini berhasil menerapkan metode *Convolution Neural Network* (CNN) untuk mengklasifikasikan penyakit *pneumonia* berdasarkan citra X-ray paru-paru. Model CNN yang dibangun dapat secara efektif memproses citra dan mengidentifikasi *pneumonia* dengan hasil yang memadai.
- 2) Hasil pengujian model *Convolution Neural Network* (CNN) menunjukkan akurasi yang tinggi dalam mengklasifikasikan citra rontgen paru-paru, dengan tingkat akurasi mencapai 96%. Evaluasi model menggunakan confussion matrix juga menunjukkan performa yang baik, dengan recall dan precision yang tinggi pada klasifikasi *pneumonia* dan normal.

#### 6.2 Saran

Setelah melakukan penelitian ini terdapat beberapa saran yang diharapkan untuk penelitian yang akan datang :

- 1) Penelitian ini menggunakan dataset dengan 5.856 citra X-ray paru-paru. Untuk meningkatkan akurasi model, disarankan untuk menggunakan dataset yang lebih besar dan lebih bervariasi, termasuk gambar dengan kualitas lebih tinggi dan lebih banyak kelas untuk diagnosis penyakit paru-paru lainnya. Penelitian lebih lanjut bisa memanfaatkan dataset dari berbagai sumber untuk meningkatkan generalisasi model.
- 2) Untuk meningkatkan kualitas dan akurasi model, tahapan pre-processing dapat diperluas dengan teknik yang lebih kompleks, seperti *denoising* untuk mengatasi masalah noise pada citra rontgen atau peningkatan kualitas citra melalui algoritma *super-resolution*. Penelitian lebih lanjut dapat mengeksplorasi penggunaan teknik-teknik baru dalam pre-processing citra untuk memaksimalkan hasil pengklasifikasian.

- 3) Meskipun CNN telah menunjukkan hasil yang memuaskan, disarankan untuk mencoba metode lain seperti Transfer Learning menggunakan model pre-trained (misalnya, VGGNet, ResNet, atau Inception) untuk membandingkan performanya dengan model CNN yang dibangun dalam penelitian ini. Metode lain, seperti Support Vector Machine (SVM) atau K-Nearest Neighbor (KNN) juga dapat digunakan untuk perbandingan dalam hal kecepatan dan akurasi.

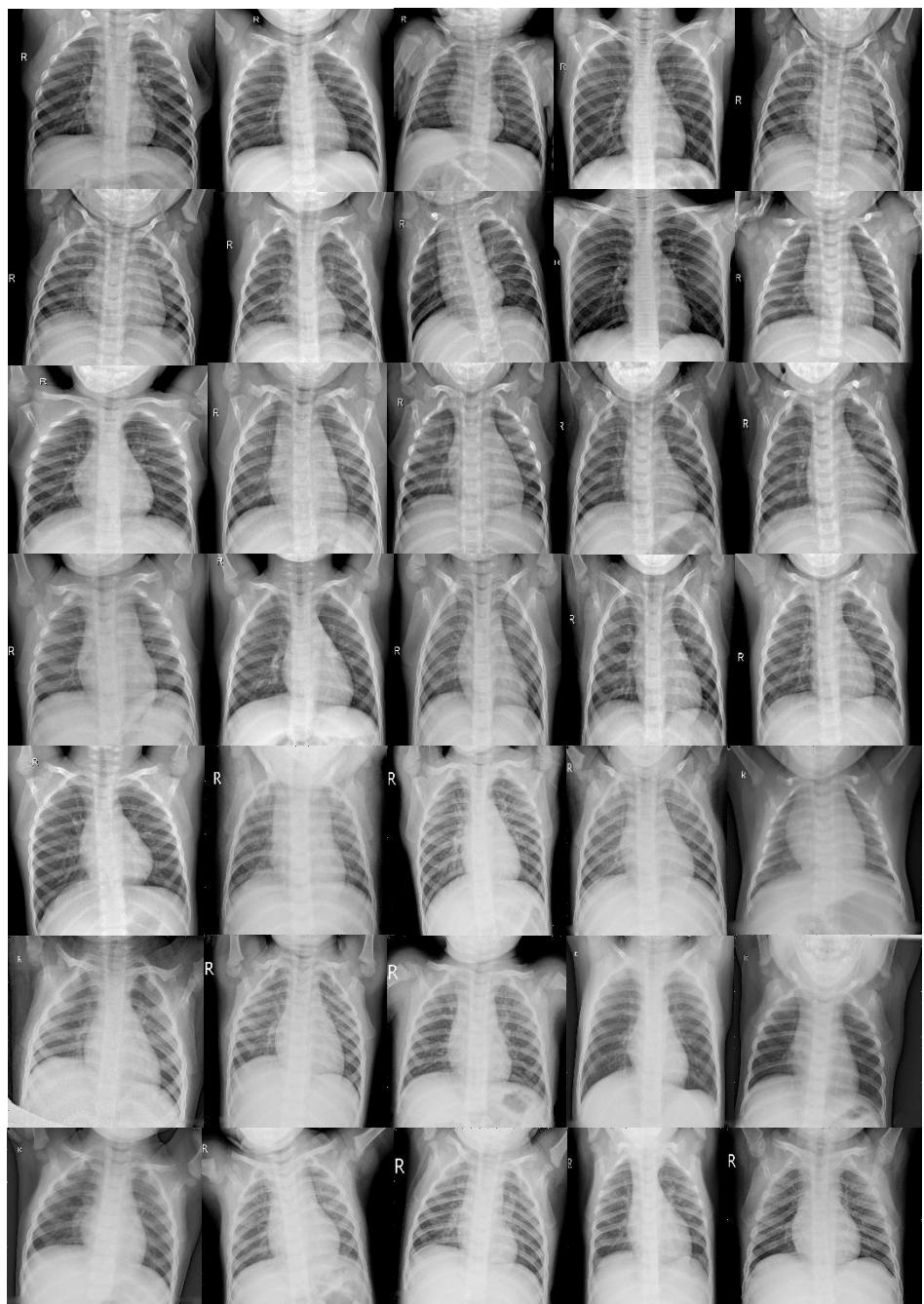
## DAFTAR PUSTAKA

- [1] N. Pratiwi *Et Al.*, “Penerapan Healing Environment Pada Rumah Sakit Khusus Paru-Paru Di Provinsi Gorontalo,” *Jambura Journal Of Architecture*, Vol. 5, No. 2, P. 2023.
- [2] M. A. Alghozali, J. R. Triosaputra, And A. Kayan, “Klasifikasi Penyakit *Pneumonia* Citra Digital X-Ray Menggunakan Metode Convolutional Neural Network Dan Rgb Equalization,” Vol. 3, Pp. 229–236, 2024.
- [3] A. Ainun, D. Halim, And S. Anraeni, “Analisis Klasifikasi Dataset Citra Penyakit *Pneumonia* Menggunakan Metode K-Nearest Neighbor (Knn),” *Indonesian Journal Of Data And Science (Ijodas)*, Vol. 2, No. 1, Pp. 1–12, 2021.
- [4] K. Kesehatan And R. Indonesia, *I Rencana Aksi Nasional Penanggulangan Pneumonia Dan Diare*. 2023.
- [5] F. Nurona Cahya *Et Al.*, “Sistemasi: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network ( Cnn),” Vol. 10, Pp. 618–626, 2021, [Online]. Available: <Http://Sistemasi.Ftik.Unisi.Ac.Id>
- [6] R. A. Wati, H. Irsyad, M. Ezar, And A. Rivan, “Klasifikasi *Pneumonia* Menggunakan Metode Support Vector Machine,” *Jurnal Algoritme*, Vol. 1, No. 1, P. 21, 2020.
- [7] S. Dewi, F. Ramadhani, And S. Djasmayena, “Klasifikasi Jenis Jerawat Berdasarkan Gambar Menggunakan Algoritma Cnn (Convolutional Neural Network),” *Hello World Jurnal Ilmu Komputer*, Vol. 3, No. 2, Pp. 68–73, Jul. 2024, Doi: 10.56211/Helloworld.V3i2.518.
- [8] O. Soedrajat F, R. Magdalena, And K. C. N. Pratiwi, “Deteksi Gangguan Paru-Paru Berbasis Citra X-Ray Menggunakan Deep Learning,” Vol. 8, Pp. 2891–2896, Dec. 2022.
- [9] R. M. Diar, R. Y. N. Fu’adah, And K. Usman, “Klasifikasi Penyakit Paru-Paru Berbasis Pengolahan Citra X Ray Menggunakan Convolutional Neural Network (Classification Of The Lung Diseases Based On X Ray Image Processing Using Convolutional Neural Network),” Vol. 9, Pp. 476–484, Apr. 2022.
- [10] E. Zalukhu, “Analisa Perbandingan Metode Certainly Faktor-Naive Bayes Terhadap Diagnose Penyakit *Pneumonia*,” *Ada Journal Of Information System Research*, Vol. 1, No. 3, Pp. 103–112, Jun. 2024.

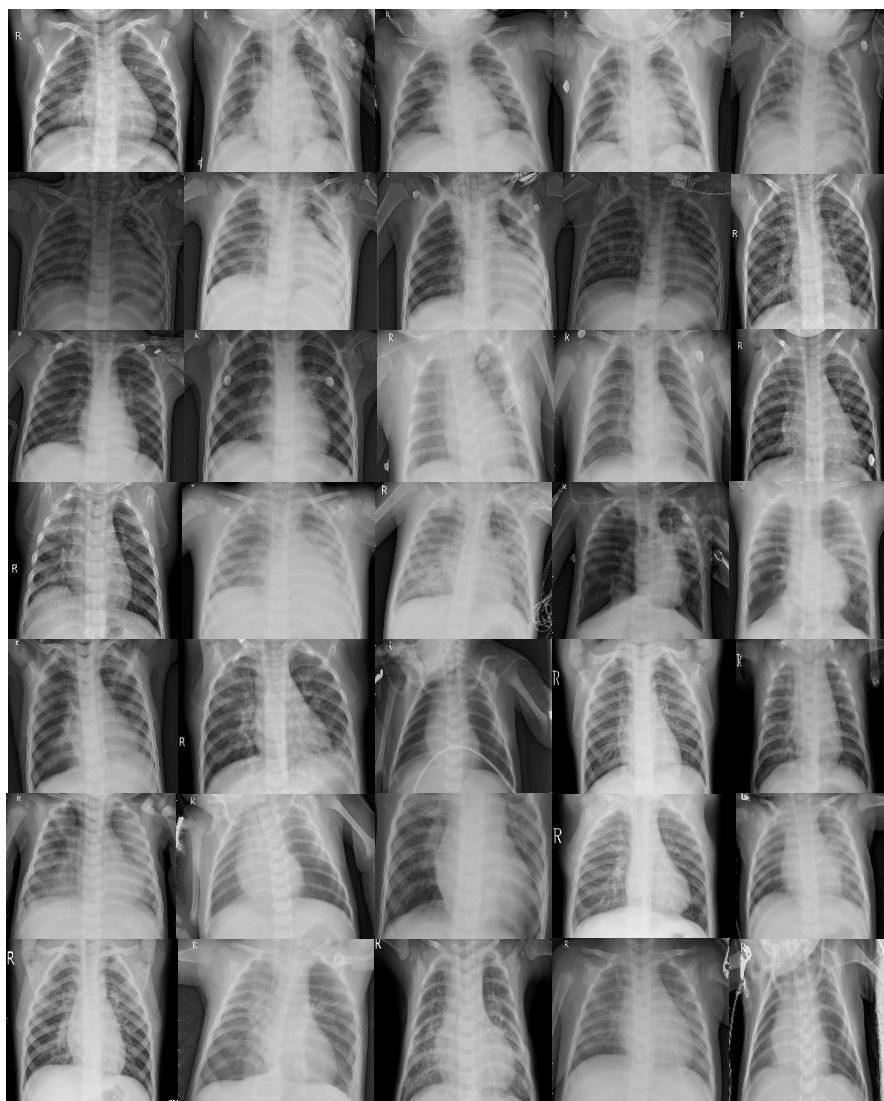
- [11] F. Marpaung, F. Aulia, N. Suryani Skom, And R. Cyra Nabila Skom, *Computer Vision Dan Pengolahan Citra Digital*. 2022. [Online]. Available: [Www.Pustakaaksara.Co.Id](http://Www.Pustakaaksara.Co.Id)
- [12] S. R. Fakultas, T. Informasi, U. Islam, K. Muhammad, And A. Al Banjari, “Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm,” *Technologia*, Vol. 11, No. 3, Pp. 181–186, Jul. 2020.
- [13] A. K. Ramadan And S. B. Laksono, “Rancang Bangun Aplikasi Deteksi Objek Untuk Menghitung Jumlah Pengunjung Restoran Berbasis Computer Vision,” *Jurnal Ikraith-Informatika*, Vol. 7, Pp. 46–57, Mar. 2023, [Online]. Available: <Https://Journals.Upi-Yai.Ac.Id/Index.Php/Ikraith-Informatika/Issue/Archive>
- [14] L. U. Khasanah, Y. N. Nasution, F. Deny, And T. Amijaya, “Klasifikasi Penyakit Diabetes Melitus Menggunakan Algoritma Naïve Bayes Classifier,” Vol. 1, No. 1, Pp. 41–50, 2022, [Online]. Available: <Http://Jurnal.Fmipa.Unmul.Ac.Id/Index.Php/Basis>
- [15] A. Srirahayu And L. Setya Pribadie, “Jurnal Ilmiah Informatika Global Review Paper Data Mining Klasifikasi Data Mining,” *Jurnal Ilmiah Informatika Global*, Vol. 14, Pp. 7–12, Apr. 2023, Doi: 10.36982/Jiig.V13i2.2307.
- [16] I. Md. D. Maysanjaya, “Klasifikasi *Pneumonia* Pada Citra X-Rays Paru-Paru Dengan Convolutional Neural Network,” *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, Vol. 9, No. 2, Pp. 190–195, May 2020, Doi: 10.22146/Jnteti.V9i2.66.
- [17] A. Dian Deva, F. Firdaus, S. Hasyim, B. Yanto, And R. Mai Candra, “Klasifikasi Prediksi Penyakit Paru-Paru Normal Dengan *Pneumonia* Berdasarkan Citra Image X-Ray Dengan Optimasi Adam Convolutional Neural Network (Cnn),” *Riau Journal Of Computer Science*, Vol. 10, No. 2, Pp. 146–155, 2024.
- [18] D. Gunawan And H. Setiawan, “Convolutional Neural Network Dalam Analisis Citra Medis,” Vol. 2, No. 2, 2022.
- [19] I. Kurniawati, R. Akbar, Y. Ardina Kusuma, And I. Fahma Kusumawati, “Identifikasi *Pneumonia* Pada Balita Melalui Citra X-Ray Menggunakan Metode Convolutional Neural Network (Cnn),” *Journal Of Manufacturing In Industrial Engineering & Technology*, Vol. 3, No. 1, Pp. 10–23, Jun. 2024, Doi: 10.30651/Mine-Tech.V3i1.23391.

**Lampiran 1 : Dataset**

Normal :



*Pneumonia :*



## Lampiran 2 : Kode Program

```
# Import library dasar untuk manipulasi data dan visualisasi

import numpy as np # Operasi array/matriks numerik

import pandas as pd # Manipulasi data berbasis tabular

import matplotlib.pyplot as plt # Visualisasi data

import seaborn as sns # Visualisasi statistik tingkat lanjut

# Library untuk pengolahan gambar

import os # Navigasi file/folder

import cv2 # OpenCV untuk pengolahan gambar

# Import library deep learning dari TensorFlow/Keras

from tensorflow.keras.models import Sequential # Arsitektur model berlapis

from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Flatten,
Dropout, BatchNormalization # Layer CNN

from tensorflow.keras.preprocessing.image import ImageDataGenerator # 
Augmentasi dan preprocessing data

from tensorflow.keras.callbacks import ReduceLROnPlateau # Callback
untuk mengatur learning rate

# Library evaluasi model

from sklearn.metrics import classification_report, confusion_matrix # 
Evaluasi performa model

from sklearn.model_selection import train_test_split # Membagi dataset
menjadi training dan testing

# Definisi label kategori
```

```

labels = ['PNEUMONIA', 'NORMAL'] # Kategori target untuk klasifikasi

# Ukuran gambar yang akan digunakan sebagai input ke model CNN

img_size = 150 # Resolusi gambar (150x150 piksel)

```

def get\_training\_data(data\_dir):

"""

Memuat dan memproses data pelatihan dari direktori dataset.

Parameter:

-----

data\_dir : str

Path ke direktori dataset, yang berisi subdirektori untuk setiap kategori label (misalnya, 'PNEUMONIA', 'NORMAL').

Returns:

-----

np.array

Array NumPy dengan format [[gambar yang diproses, label], ...].

- gambar yang diproses : ndarray

Gambar yang telah diubah menjadi grayscale dan diresize ke ukuran (img\_size, img\_size).

- label : int

Label kategori, 0 untuk 'PNEUMONIA' dan 1 untuk 'NORMAL'.

"""

```

data = [] # List untuk menyimpan data gambar dan label

for label in labels:

    path = os.path.join(data_dir, label) # Path ke folder kategori

    class_num = labels.index(label) # Index label sebagai kelas

    for img in os.listdir(path): # Iterasi melalui semua file gambar di folder

        try:

            # Baca gambar dalam mode grayscale

            img_arr = cv2.imread(os.path.join(path, img),
cv2.IMREAD_GRAYSCALE)

            if img_arr is not None: # Pastikan gambar berhasil dibaca

                # Resize gambar ke ukuran target

                resized_arr = cv2.resize(img_arr, (img_size, img_size))

                data.append([resized_arr, class_num]) # Tambahkan gambar
dan label ke dataset

            except Exception as e:

                print(f"Error loading image {img}: {e}")

        return np.array(data, dtype=object) # Kembalikan array NumPy

# Path ke dataset untuk pelatihan, pengujian, dan validasi

train_path = 'chest_xray-pneumonia/chest_xray/train' # Path ke folder data
pelatihan

```

```
test_path = 'chest_xray-pneumonia/chest_xray/test'    # Path ke folder data pengujian

val_path = 'chest_xray-pneumonia/chest_xray/val'      # Path ke folder data validasi

# Load data

train_data = get_training_data(train_path)

test_data = get_training_data(test_path)

val_data = get_training_data(val_path)

def count_images(data_dir):

    for label in labels:

        path = os.path.join(data_dir, label)

        num_images = len(os.listdir(path))

        print(f'{label}: {num_images} images in {data_dir}')

    count_images(train_path)

    count_images(test_path)

    count_images(val_path)

    print(f'Train data size: {len(train_data)}')

    print(f'Test data size: {len(test_data)}')

    print(f'Validation data size: {len(val_data)}')

# Menampilkan beberapa gambar sampel

for i in range(5):

    plt.imshow(train_data[i][0], cmap='gray')
```

```
plt.title(f"Label: {train_data[i][1]}")  
plt.show()  
  
# Combine all data for splitting  
  
all_data = np.concatenate([train_data, test_data, val_data])  
  
np.random.shuffle(all_data) # Shuffle for randomness  
  
# Split data for input and output (50% each)  
  
total_data = len(all_data)  
  
train_limit = total_data // 2  
  
input_data = all_data[:train_limit]  
  
output_data = all_data[train_limit:]  
  
  
# Prepare input and output data  
  
x_input, y_input = [], []  
  
x_output, y_output = [], []  
  
for feature, label in input_data:  
  
    x_input.append(feature)  
  
    y_input.append(label)  
  
for feature, label in output_data:  
  
    x_output.append(feature)  
  
    y_output.append(label)  
  
# Normalize the data
```

```
x_input = np.array(x_input) / 255

x_output = np.array(x_output) / 255

# Reshape data for deep learning input

x_input = x_input.reshape(-1, img_size, img_size, 1)

y_input = np.array(y_input)

x_output = x_output.reshape(-1, img_size, img_size, 1)

y_output = np.array(y_output)

# Data augmentation to prevent overfitting

datagen = ImageDataGenerator(

    rotation_range=30,

    zoom_range=0.2,

    width_shift_range=0.1,

    height_shift_range=0.1,

    horizontal_flip=True

)

datagen.fit(x_input)

# Visualisasi hasil augmentasi

augmented_images = []

augmented_labels = []

# Ambil satu batch data augmentasi
```

```
for x_batch, y_batch in datagen.flow(x_input, y_input, batch_size=16):

    augmented_images = x_batch # Simpan gambar yang diaugmentasi

    augmented_labels = y_batch # Simpan label yang diaugmentasi

    break # Ambil satu batch saja

# Tampilkan beberapa gambar hasil augmentasi

plt.figure(figsize=(12, 12))

for i in range(9): # Tampilkan 9 gambar

    plt.subplot(3, 3, i + 1)

    plt.imshow(augmented_images[i].reshape(img_size, img_size), cmap='gray')

    plt.title(f"Label: {augmented_labels[i]}")

    plt.axis('off')

plt.tight_layout()

plt.show()

# Build the CNN model

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(img_size,
    img_size, 1)),
    BatchNormalization(),
    MaxPool2D((2, 2)),
```

```
Conv2D(64, (3, 3), activation='relu', padding='same'),  
Dropout(0.1),  
BatchNormalization(),  
MaxPool2D((2, 2)),  
Conv2D(64, (3, 3), activation='relu', padding='same'),  
BatchNormalization(),  
MaxPool2D((2, 2)),  
Conv2D(128, (3, 3), activation='relu', padding='same'),  
Dropout(0.2),  
BatchNormalization(),  
MaxPool2D((2, 2)),  
Conv2D(256, (3, 3), activation='relu', padding='same'),  
Dropout(0.2),  
BatchNormalization(),  
MaxPool2D((2, 2)),  
Flatten(),  
Dense(128, activation='relu'),  
Dropout(0.2),  
Dense(1, activation='sigmoid')  
])
```

```
model.compile(optimizer='rmsprop', loss='binary_crossentropy',  
metrics=['accuracy'])  
  
model.summary()  
  
# Re-import required libraries due to state reset  
  
import matplotlib.pyplot as plt  
  
import numpy as np  
  
# Jumlah neuron di setiap lapisan Fully Connected  
  
input_neurons = 256 # Jumlah fitur input setelah Flatten (perkiraan berdasarkan  
model sebelumnya)  
  
hidden_layers = [128] # Jumlah neuron di Dense layers  
  
output_neurons = 1 # Jumlah neuron output  
  
  
# Fungsi untuk menggambar Fully Connected Layer  
  
def plot_fully_connected(input_neurons, hidden_layers, output_neurons):  
  
    fig, ax = plt.subplots(figsize=(10, 8))  
  
    ax.axis("off")  
  
    ax.set_title("Fully Connected Layers with Input and Output", fontsize=16)  
  
    # Koordinat untuk setiap lapisan  
  
    layer_x = [0] + list(range(1, len(hidden_layers) + 1)) + [len(hidden_layers) + 1]
```

```

# Plot neuron input

input_y = np.linspace(-1, 1, 10) # Representasikan hanya 10 neuron untuk
visualisasi

ax.scatter([layer_x[0]] * len(input_y), input_y, s=600, color="blue",
label="Input Layer (256)")

# Plot hidden layers

prev_y = input_y

for i, (neurons, x) in enumerate(zip(hidden_layers, layer_x[1:-1])):

    layer_y = np.linspace(-1, 1, neurons // 16) # Representasikan sebagian neuron
untuk visualisasi

    ax.scatter([x] * len(layer_y), layer_y, s=600, color="orange", label=f"Hidden
Layer {i + 1} ({neurons})")

    # Koneksi dari lapisan sebelumnya ke lapisan ini

    for y1 in prev_y:

        for y2 in layer_y:

            ax.plot([layer_x[i], x], [y1, y2], color="gray", lw=0.5)

    prev_y = layer_y

# Plot neuron output

output_y = np.linspace(-1, 1, output_neurons)

ax.scatter([layer_x[-1]] * len(output_y), output_y, s=600, color="green",
label="Output Layer (1)")

# Koneksi dari hidden layer terakhir ke output

for y1 in prev_y:

```

```

for y2 in output_y:
    ax.plot([layer_x[-2], layer_x[-1]], [y1, y2], color="gray", lw=0.5)

# Tampilkan diagram
ax.legend(loc="upper right", fontsize=10)
plt.show()

# Panggil fungsi untuk menggambar
plot_fully_connected(input_neurons, hidden_layers, output_neurons)

# Learning rate reduction
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy',
patience=2, verbose=1, factor=0.3, min_lr=0.000001)

# Train the model
history = model.fit(
    x_input, y_input,
    epochs=50,
    validation_split=0.2, # 20% of input data used for validation
    batch_size=32,
    callbacks=[learning_rate_reduction]
)

# Evaluate the model on output data
evaluation = model.evaluate(x_output, y_output)
print("Loss on output data: ", evaluation[0])

```

```
print("Accuracy on output data: ", evaluation[1] * 100, "%")

# Jumlah epoch

num_epochs = 50

# Menyimpan loss dan akurasi dari training dan validation

train_losses = history.history['loss']      # Loss pada data pelatihan

val_losses = history.history['val_loss']    # Loss pada data validasi

train_accuracies = history.history['accuracy'] # Akurasi pada data pelatihan

val_accuracies = history.history['val_accuracy'] # Akurasi pada data validasi

plt.plot(range(num_epochs), train_losses, label='Training Loss')

plt.plot(range(num_epochs), val_losses, label='Validation Loss')

plt.title('Training and Validation Loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()

plt.plot(range(num_epochs), train_accuracies, label='Training Accuracy')

plt.plot(range(num_epochs), val_accuracies, label='Validation Accuracy')

plt.title('Training and Validation Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()
```

```
plt.show()

# Plot akurasi pelatihan dan validasi

plt.plot(history.history['accuracy'], label='accuracy') # Akurasi pada data pelatihan

plt.plot(history.history['val_accuracy'], label='val_accuracy') # Akurasi pada data validasi

# Label untuk sumbu dan grafik

plt.xlabel('Epoch') # Label untuk sumbu X

plt.ylabel('Accuracy') # Label untuk sumbu Y

plt.ylim([0, 1]) # Rentang akurasi antara 0 dan 1

plt.legend(loc='lower right') # Menampilkan legenda di pojok kanan bawah

plt.show() # Menampilkan grafik


# Plot kurva akurasi dan loss

plt.figure(figsize=(10, 5)) # Atur ukuran plot

plt.plot(range(1, num_epochs + 1), val_losses, label='Validation Loss')

plt.plot(range(1, num_epochs + 1), val_accuracies, label='Validation Accuracy')

plt.title('Training Accuracy Over Epochs') # Judul plot

plt.xlabel('Epochs') # Label sumbu X

plt.ylabel('Accuracy (%)') # Label sumbu Y

plt.legend() # Tambahkan legenda

plt.show() # Tampilkan plot
```

```
# Membuat prediksi

predictions = (model.predict(x_output) > 0.5).astype("int32") # Prediksi label biner
# (0 atau 1)

# Menghasilkan laporan klasifikasi

print(classification_report(y_output, predictions, target_names=labels)) # labels
# adalah ['PNEUMONIA', 'NORMAL']

# Confusion Matrix

cm = confusion_matrix(y_output, predictions)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels,
# yticklabels=labels)

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

from sklearn.metrics import roc_curve, auc

y_probs = model.predict(x_output).ravel() # Probabilitas prediksi

fpr, tpr, _ = roc_curve(y_output, y_probs)

roc_auc = auc(fpr, tpr)
```

```
plt.figure()

plt.plot(fpr, tpr, color='blue', label=f"ROC Curve (AUC = {roc_auc:.2f})")

plt.plot([0, 1], [0, 1], color='red', linestyle='--') # Garis diagonal sebagai baseline

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve')

plt.legend()

plt.show()
```

**Lampiran 3 : Riwayat Hidup Peneliti****RIWAYAT HIDUP**

**Nama** : Thia Riskiyani H. Mosi  
**Nim** : T3121008  
**Tempat Tanggal Lahir** : Boliyohuto, 08 Juni 2003  
**Email** : thiariskiyani31@gmail.com

**Riwayat Pendidikan :**

1. Tahun 2015, menyelesaikan Pendidikan di Sekolah Dasar Negeri 07 Boliyohuto, Kecamatan Boliyohuto, Kabupaten Gorontalo, Provinsi Gorontalo
2. Tahun 2018, menyelesaikan Pendidikan di Sekolah Menengah Pertama 01 Boliyohuto, Kecamatan Boliyohuto, Kabupaten Gorontalo Provinsi Gorontalo
3. Tahun 2021, menyelesaikan Pendidikan di Sekolah Menengah Atas 01 Boliyohuto, Kecamatan Boliyohuto, Kabupaten Gorontalo Provinsi Gorontalo.
4. Tahun 2021, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo

## Lampiran 4 : Surat Penelitian



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
UNIVERSITAS ICHSAN GORONTALO  
LEMBAGA PENELITIAN**

Jl. Achmad Nadjamuddin No.17, Kampus Unisan Gorontalo Lt.1 Kota Gorontalo 96128  
Website: lembagapenelitian.unisan.ac.id, Email: lembagapenelitian@unisan.ac.id

---

Nomor : 48/PIP/B.04/LP-UIG/2024  
 Lampiran : -  
 Hal : Permohonan Izin Penelitian (Pengambilan Data)

Kepada Yth.,  
 Dekan Fakultas Ilmu Komputer UNISAN Gorontalo  
 di -  
 Tempat

Yang bertandatangan di bawah ini:

Nama : Dr. Rahmisyari, ST., SE., MM  
 NIDN : 0929117202  
 Pangkat Akademik : Lektor Kepala  
 Jabatan : Ketua Lembaga Penelitian Universitas Ichsan Gorontalo

Meminta kesediaannya untuk memberikan izin pengambilan data dalam rangka penyusunan **Proposal/Skripsi**, kepada:

Nama : Thia Riskiyani H. Mosi  
 NIM : T3121008  
 Fakultas : Ilmu Komputer  
 Program Studi : Teknik Informatika  
 Judul Penelitian : KLASIFIKASI PENYAKIT PARU-PARU (PNEUMONIA)  
 MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Lokasi Penelitian : FAKULTAS ILMU KOMPUTER UNISAN GORONTALO  
 Demikian surat ini saya sampaikan, atas bantuan dan kerjasamanya diucapkan banyak terima kasih.

Dikeluarkan di Gorontalo  
 Tanggal, 22/10/2024



Dr. Rahmisyari, ST., SE., MM  
 NIDN: 0929117202

## Lampiran 5 : Surat Rekomendasi Penelitian



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
UNIVERSITAS IHSAN GORONTALO**

**FAKULTAS ILMU KOMPUTER  
SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001  
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo**

### **SURAT KETERANGAN PENELITIAN**

Nomor : 028 /FIKOM-UIG/SKP/I/2025

Yang bertanda tangan dibawah ini :

N a m a : Irvan Abraham Salihi, M. Kom  
Jabatan : Dekan Fakultas Ilmu Komputer

Dengan ini Menerangkan bahwa :

N a m a Mahasiswa : Thia Riskiyani H. Mosi  
N I M : T3121008  
Program Studi : Teknik Informatika

Bawa yang bersangkutan benar-benar telah melakukan penelitian tentang **“Klasifikasi Penyakit Pneumonia Menggunakan Metode Convolutional Neural Network (CNN) ”** Guna untuk meyelesaikan Studi pada Program Studi Teknik Informatika Fakultas Ilmu Komputer, dan bersangkutan telah menyelesaikan penelitian tersebut pada **TGL 30 Januari 2025** sesuai dengan waktu yang telah ditentukan.

Demikian Surat Keterangan ini dibuat dan digunakan untuk seperlunya.



**Lampiran 6 : Surat Keterangan Bebas Pustak**

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
UNIVERSITAS ICHSAN GORONTALO  
FAKULTAS ILMU KOMPUTER  
UPT. PERPUSTAKAAN FAKULTAS  
SK. MENDIKNAS RI NO. 84/D/0/2001

Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

**SURAT KETERANGAN BEBAS PUSTAKA**

No : 001/Perpustakaan-Fikom/II/2025

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota : Thia Riskiyani H. Mosi  
No. Induk : T3121008  
No. Anggota : M20251

Terhitung mulai hari, tanggal : Sabtu, 01 Februari 2025, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

**Gorontalo, 01 Februari 2025**

**Mengetahui,  
Kepala Perpustakaan**



**Apriyanto Alhamad, M.Kom**  
NIDN : 0924048601

## Lampiran 7 : Bebas Plagiasi & Hasil Turnitin



**KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI**  
**UNIVERSITAS IHSAN GORONTALO**  
**FAKULTAS ILMU KOMPUTER**  
**SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001**  
**Jl. Achmad Najamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo**

**SURAT REKOMENDASI BEBAS PLAGIASI**  
**No. 031/FIKOM-UIG/R/II/2025**

Yang bertanda tangan di bawah ini :

Nama : Irvan Abraham Salihi, M.Kom  
 NIDN : 0928028101  
 Jabatan : Dekan Fakultas Ilmu Komputer

Dengan ini menerangkan bahwa :

Nama Mahasiswa : Thia Riskiyani H.Mosi  
 NIM : T3121008  
 Program Studi : Teknik Informatika (S1)  
 Fakultas : Fakultas Ilmu Komputer  
 Judul Skripsi : Klasifikasi Penyakit Pneumonia Menggunakan Metode Convolution Neural Network [CNN]

Sesuai hasil pengecekan tingkat kemiripan skripsi melalui aplikasi **Turnitin** untuk judul skripsi di atas diperoleh hasil *Similarity* sebesar **12%**, berdasarkan Peraturan Rektor No. 32 Tahun 2019 tentang Pendekripsi Plagiat pada Setiap Karya Ilmiah di Lingkungan Universitas Ihsan Gorontalo dan persyaratan pemberian surat rekommendasi verifikasi calon wisudawan dari LLDIKTI Wil. XVI, bahwa batas kemiripan skripsi maksimal 30%, untuk itu skripsi tersebut di atas dinyatakan **BEBAS PLAGIASI** dan layak untuk diujangkan.

Demikian surat rekommendasi ini dibuat untuk digunakan sebagaimana mestinya.

Mengetahui  
 Dekan  
  
Irvan Abraham Salihi, M.Kom  
 NIDN. 0928028101

Gorontalo, 08 Februari 2025  
 Tim Verifikasi,

  
Zulfrianto V. Lamasigi, M.Kom  
 NIDN. 0914089101

Terlampir :  
 Hasil Pengecekan Turnitin



## 12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

### Filtered from the Report

- ▶ Bibliography

---

### Top Sources

- |    |                                  |
|----|----------------------------------|
| 0% | Internet sources                 |
| 9% | Publications                     |
| 8% | Submitted works (Student Papers) |
- 

### Integrity Flags

#### 0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.





## Top Sources

- 0% Internet sources
- 9% Publications
- 8% Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
	Universitas Mercu Buana	1%
2	Student papers	
	LL Dikti IX Turnitin Consortium	<1%
3	Student papers	
	Universitas Sam Ratulangi	<1%
4	Student papers	
	Sriwijaya University	<1%
5	Student papers	
	Universitas Dian Nuswantoro	<1%
6	Publication	
	Andi Ainun Dzariah Halim, Siska Anraeni. "Analisis Klasifikasi Dataset Citra Peny..."	<1%
7	Publication	
	Shaheen Fatima. "Image Retrieval Framework for Anomalies Detection in X-Ray R..."	<1%
8	Publication	
	Dede Husen. "Klasifikasi Citra MRI Tumor Otak Menggunakan Metode Convolutio..."	<1%
9	Publication	
	Andrian Herbert Parsaoran Sitohang, Teguh Iman Hermanto, Candra Dewi Lestari...	<1%
10	Publication	
	Clairine Aurellia Sanjaya, Minto Waluyo. "ANALISIS PERBANDINGAN METODE TRA..."	<1%
11	Publication	
	Jopa Yopento, Ernawati Ernawati, Funny Farady Coastera. "Identifikasi Pneumoni..."	<1%



12	Publication	Novia Pramesti Aprilia, Theresia Herlina Rochadiani. "Image Captioning untuk Ga..."	<1%
13	Publication	Ananda Rizki Dani, Irma Handayani. "Classification of Yogyakarta Batik Motifs Usi..."	<1%
14	Publication	Dandi Dandi, Niniek Pratiwi, Kalih Trumansyahjaya. "PENERAPAN HEALING ENVIR..."	<1%
15	Student papers	Universitas Brawijaya	<1%
16	Student papers	Universitas Pamulang	<1%
17	Publication	Alfito Herdiansyah, Arief Hermawan, Sutarmen. "Implementasi Algoritma Conv..."	<1%
18	Publication	Ivo Colanus Rally Drajana, Novriyanti Polimengo, Annahli Riadi. "SPK Penilaian Kin..."	<1%
19	Student papers	Universitas Negeri Jakarta	<1%
20	Publication	Zahra Purwanti, Sugiyono. "Pemodelan Text Mining untuk Analisis Sentimen Ter..."	<1%
21	Publication	Desya Risty Putri, Eva Yulia Puspaningrum, Hendra Maulana. "KLASIFIKASI SENT..."	<1%
22	Publication	Dede Husen. "EVALUASI TEKNIK AUGMENTASI DATA UNTUK KLASIFIKASI TUMOR ..."	<1%
23	Publication	Dhini Septhya, Rahmaddeni, Susanti, Agustin. "Penerapan Algoritma Convolutio..."	<1%
24	Publication	Dyan Prawita Sari, Zuhri Halim, Irlon Irlon, Bayu Waseso, Saromah Saromah. "Im..."	<1%
25	Student papers	IAIN Bengkulu	<1%

26	Publication	Ika Amelia, Sugiyono, Francis Matheos Sarimole, Tundo. "Analisis Sentimen Tan...<1%
27	Publication	Aang Alim Murtopo, Maulana Aditya, Pingky Septiana Ananda, Gunawan Gunaw...<1%
28	Student papers	Universitas Jenderal Soedirman <1%
29	Student papers	Universiti Malaysia Sabah <1%
30	Publication	Didi Kurniawan, Dhani Ariatmanto. "IDENTIFIKASI VARIETAS BIBIT DURIAN MEN...<1%
31	Student papers	President University <1%
32	Publication	SATRIA SATRIA, Sumijan, Billy Hendrik. "Implementasi Convolutional Neural Net...<1%
33	Student papers	Universitas Diponegoro <1%
34	Publication	Ilham Jawaz, Reni Rahmadewi. "Sistem Deteksi Pneumonia Paru-Paru dengan Pe...<1%
35	Student papers	Universitas Muria Kudus <1%
36	Student papers	Universitas Muslim Indonesia <1%

## Lampiran 8 : Lembar Revisi



**KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI**  
**UNIVERSITAS ICHSAN GORONTALO**  
**FAKULTAS ILMU KOMPUTER**  
**SK MENDIKNAS NOMOR 84/D/O/2001**  
**JL. Achmad Nadjamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo**

**Berita Acara Perbaikan/Revisi Ujian Skripsi**

Pada hari ini, Kamis 20 Februari 2025, Pukul 13.00-14.30 Wita. Telah dilaksanakan Ujian Skripsi mahasiswa Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Nama	:	Thia Riskiyani H. Mosi
Nim	:	T3121008
Pembimbing I	:	Yasin Aril Mustofa, M.Kom
Pembimbing II	:	Sudirman Melangi, M.Kom
Judul Skripsi	:	Klasifikasi Penyakit Paru-Paru (Pneumonia) Menggunakan Metode Convolution Neural Network (CNN)

Oleh Komite Seminar sebagai berikut :

No	Komite Seminar	Status	Tanda Tangan
1	Irma Surya Kumala Idris, M.Kom	Ketua	
2	Hastuti Dalai, M.Kom	Anggota	
3	Yulianty Lasena, M.Kom	Anggota	
4	Yasin Aril Mustofa, M.Kom	Anggota	
5	Sudirman Melangi, M.Kom	Anggota	