

**SISTEM PENDETEKSI PNEUMONIA MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR
DENSENET201
LAPORAN TUGAS AKHIR**

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Strata 1 (S1) Pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung



DISUSUN OLEH :

HILDAN MULYO HERMAWAN

32602000030

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2024**

FINAL PROJECT
PNEUMONIA DETECTION SYSTEM USING CONVOLUTIONAL NEURAL
NETWORK METHOD WITH DENSENET201 ARCHITECTURE

Proposed to complete the requirement to obtain a bachelor's degree (SI)
at Informatics Engineering Departement of Industrial Technology Faculty

Sultan Agung Islamic University



Arranged By :

HILDAN MULYO HERMAWAN

32602000030

MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG

2024

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul “SISTEM PENDETEKSI PNEUMONIA MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* DENGAN ARSITEKTUR DENSENET201” ini disusun oleh :

Nama : Hildan Mulyo Hermawan

NIM : 32602000030

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

Hari : Rabu

Tanggal : 28 Februari 2024

Mengesahkan,

Pembimbing I



Muhammad Qomaruddin, S.T.,M.Sc.,Ph.D
NIDN. 0631057101

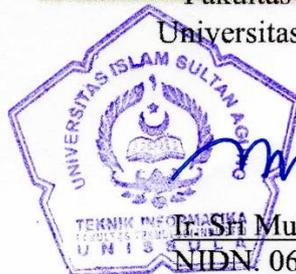
Pembimbing II



Andi Riansyah, S.T.,M.Kom
NIDN. 0609108802

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Sultan Agung




Sri Mulyono, M. Eng
NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul “SISTEM PENDETEKSI PNEUMONIA MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* DENGAN ARSITEKTUR DENSENET201” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : *Rabu*

Tanggal : *28 Februari 2024*

TIM PENGUJI

Anggota I

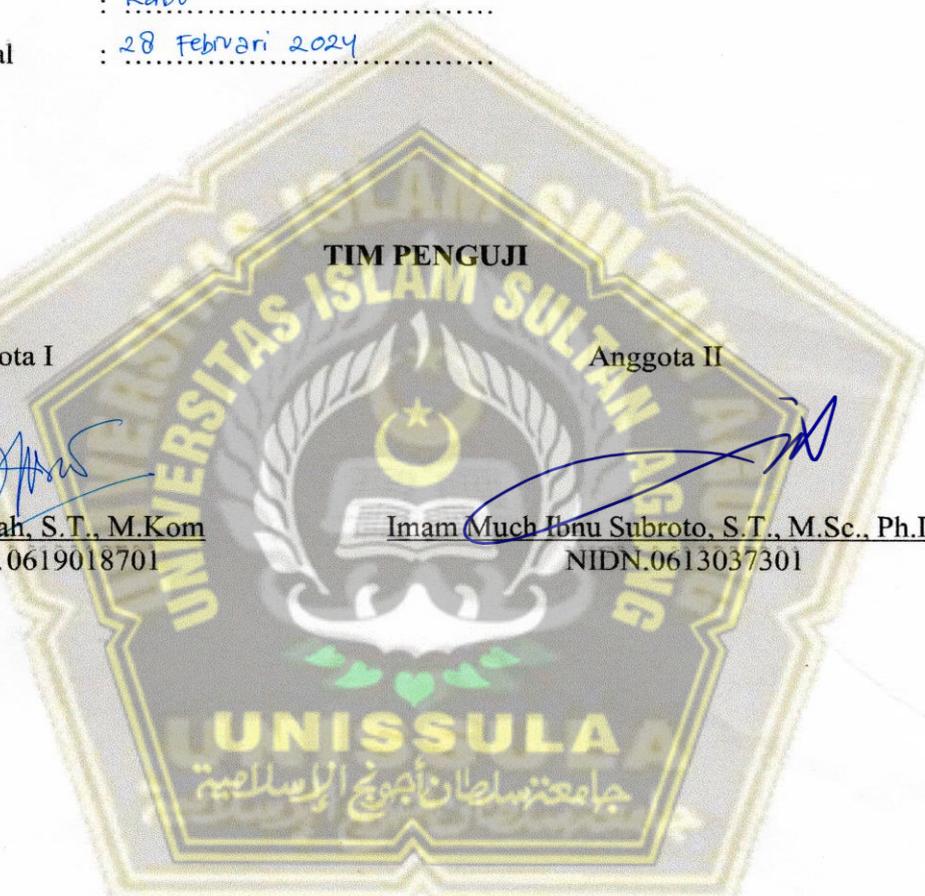


Badie'ah, S.T., M.Kom
NIDN.0619018701

Anggota II



Imam Much Ibnu Subroto, S.T., M.Sc., Ph.D
NIDN.0613037301



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Hildan Mulyo Hermawan

NIM : 32602000030

Judul Tugas Akhir : Sistem Pendeteksi Pneumonia Menggunakan Metode Convolutional Neural Network Dengan Arsitektur Densenet201

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 29 Februari 2024

Yang Menyatakan,



Hildan Mulyo H.

KATA PENGANTAR

Saya bersyukur kepada Allah SWT karena telah memberi saya kesempatan untuk menyelesaikan Tugas Akhir dengan judul "Sistem Pendeteksi Pneumonia Menggunakan Metode Konvolusional Neural Network dengan Arsitektur Densenet201". Tugas Akhir ini merupakan salah satu syarat untuk menyelesaikan studi dan memungkinkan saya untuk memperoleh gelar sarjana (S-1) di Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam.

Sebagai penulis, saya ingin mengucapkan terima kasih kepada berbagai pihak yang telah membantu dalam penyusunan dan pembuatan tugas akhir ini, antara lain:

1. Rektor UNISSULA, Bapak Prof. Dr. H. Gunarto, S.H., M.Hum;
2. Dekan Fakultas Teknologi Industri, Ibu Dr. Hj. Novi Marlyana ST., M.T., IPU, Asean Eng;
3. Dosen Pembimbing I, Bapak Muhammad Qomaruddin, S.T., M.Sc., Ph.D dan Dosen Pembimbing II Bapak Andi Riansyah, S.T., M.Kom, yang telah meluangkan waktu dan memberikan pengetahuan;
4. Dengan rendah hati, penulis menyadari bahwa laporan masih kekurangan kualitas, kuantitas, dan ilmu pengetahuan. Oleh karena itu, penulis mengharapkan kritikan dan saran yang membangun untuk membantu memperbaiki laporan ini.

Semarang, 29 Februari 2024



Hildan Mulyo Hermawan

DAFTAR ISI

COVER	i
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	v
PERNYATAAN PERSUTUJUAN PUBLIKASI ILMIAH	vi
KATA PENGANTAR	vii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiii
ABSTRAK	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Pembatasan Masalah	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	6
2.2.1 Pneumonia	6
2.2.2 <i>Image Augmentation</i>	7
2.2.3 <i>Convolutional Neural Network</i>	8
2.2.4 Sigmoid.....	10
2.2.5 DenseNet.....	11
2.2.6 RESTful API	12
2.2.7 <i>Deep Learning</i>	13

2.2.8	FastAPI.....	14
BAB III	METODE PENELITIAN	15
3.1	Tahapan Penelitian	15
3.2	Studi Literatur.....	15
3.3	Rancangan Sistem	16
3.3.1	Analisis Kebutuhan Sistem	16
3.3.2	<i>Flowchart</i> Sistem.....	16
3.4	Perancangan Model	17
3.4.1	Pengumpulan Data.....	17
3.4.2	Data <i>Preprocessing</i>	18
3.4.3	Data <i>training</i>	19
3.5	Evaluasi Model.....	20
3.6	Pembuatan Sistem	20
3.6.1	<i>Frontend</i>	20
3.6.2	<i>Backend</i>	20
3.7	<i>Deployment</i> Model	21
3.8	<i>Testing</i> Sistem	21
BAB IV	HASIL DAN ANALISIS PENELITIAN	22
4.1	Pengolahan Data.....	22
4.1.1	<i>Preprocessing</i>	22
4.2	Cara Kerja Sistem.....	23
4.2.1	Unggah Gambar	23
4.2.2	Pemrosesan Gambar oleh Model	23
4.2.3	Klasifikasi Model.....	23
4.3	Implementasi <i>Backend</i>	23
4.4	Implementasi <i>Frontend</i>	24
4.5	Pengujian Sistem	26
4.6	Hasil dan Analisis.....	27
4.6.1	Perbandingan <i>Optimizer</i>	27
4.6.2	<i>Testing</i> Setiap <i>Optimizer</i>	31

4.6.3	Perbandingan <i>Batch Size</i>	41
4.6.4	Percobaan tanpa parameter bobot kelas	48
4.6.5	Hasil <i>Testing</i>	50
BAB V Kesimpulan dan Saran		53
5.1	Kesimpulan.....	53
5.2	Saran	53
Daftar Pustaka		54
Lampiran		56



DAFTAR TABEL

Tabel 3. 1 kebutuhan perangkat lunak pada sistem	16
Tabel 3. 2 spesifikasi perangkat keras	16
Tabel 3. 3 pembagian data training dan validation	18
Tabel 4. 1 Pengujian Black Box	27
Tabel 4. 2 Grafik <i>accuracy</i> dan <i>loss</i>	28
Tabel 4. 3 Hasil konfigurasi.....	31
Tabel 4. 4 Skor precision, recall, dan F1-Score Adam	32
Tabel 4. 5 Skor <i>precision</i> , <i>recall</i> , dan <i>F1-Score</i> Adam tiap kelas	32
Tabel 4. 6 <i>Confusion matrix</i> Adam tiap kelas.....	33
Tabel 4. 7 Perhitungan manual <i>precision</i> , <i>recall</i> , dan <i>F1-Score</i> Adam tiap kelas	33
Tabel 4. 8 perhitungan manual <i>precision</i> , <i>recall</i> , dan <i>F1-Score</i> Adam	35
Tabel 4. 9 Skor precision, recall, dan F1-Score Adamax	36
Tabel 4. 10 Skor precision, recall, dan F1-Score Adamax tiap kelas	36
Tabel 4. 11 <i>Confusion matrix</i> Adamax tiap kelas	36
Tabel 4. 12 perhitungan manual <i>precision</i> , <i>recall</i> , dan <i>F1-Score</i> Adamax tiap kelas	37
Tabel 4. 13 perhitungan manual <i>precision</i> , <i>recall</i> , dan <i>F1-Score</i> Adamax.....	38
Tabel 4. 14 skor precision, recall, dan F1-Score Adadelta	39
Tabel 4. 15 skor precision, recall, dan F1-Score Adadelta tiap kelas.....	39
Tabel 4. 16 <i>confusion matrix</i> Adadelta tiap kelas	39
Tabel 4. 17 Perhitungan manual <i>confusion matrix</i> Adadelta tiap kelas.....	40
Tabel 4. 18 Perhitungan Manual <i>Confusion matrix</i> Adadelta.....	41
Tabel 4. 19 Rincian Luaran Terakhir Hasil <i>Train Batch Size 32</i>	42
Tabel 4. 20 Skor <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score Batch Size 32</i>	43
Tabel 4. 21 Rincian Luaran Terakhir Hasil <i>Train Batch Size 128</i>	44
Tabel 4. 22 Skor <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score Batch Size 128</i>	45
Tabel 4. 23 Rincian Luaran Terakhir Hasil <i>Train Batch Size 256</i>	46
Tabel 4. 24 Skor <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score Batch Size 256</i>	47
Tabel 4. 25 Rincian Luaran Terakhir Hasil <i>Train</i> tanpa bobot kelas	48

Tabel 4. 26 Skor *Precision*, *Recall*, dan *F1-Score Batch Size* tanpa bobot kelas . 50

Tabel 4. 27 Hasil *Testing* 51



DAFTAR GAMBAR

Gambar 2. 1 Arsitektur CNN (Phung & Rhee, 2019).....	8
Gambar 2. 2 Grafik Fungsi Sigmoid	10
Gambar 2. 3 Arsitektur DenseNet (Minaee dkk., 2020).....	12
Gambar 3. 1 Alur Penelitian	15
Gambar 3. 2 Diagram Alir Sistem	16
Gambar 3. 3 Gambar pada <i>Dataset</i>	18
Gambar 3. 4 flowchart layer output.....	19
Gambar 3. 5 Rancangan frontend.....	20
Gambar 3. 6 Rancangan <i>backend</i>	20
Gambar 3. 7 <i>flowchart Deployment</i> model.....	21
Gambar 4. 1 Hasil Augmentasi Gambar.....	22
Gambar 4. 2 Dokumentasi API Backend.....	23
Gambar 4. 3 Spesifikasi API Unggah File.....	24
Gambar 4. 4 Halaman Utama Sistem	25
Gambar 4. 5 Tampilan Sistem Saat Prediksi Pneumonia	25
Gambar 4. 6 tampilan sistem saat prediksi normal.....	26
Gambar 4. 7 <i>Confusion matrix</i> adam.....	32
Gambar 4. 8 <i>Confusion matrix</i> Adamax.....	35
Gambar 4. 9 <i>Confusion matrix</i> Adadelata.....	38
Gambar 4. 10 <i>Learning Curve Batch Size 32</i>	42
Gambar 4. 11 <i>Confusion matrix Batch Size 32</i>	43
Gambar 4. 12 <i>Learning Curve Batch Size 128</i>	44
Gambar 4. 13 <i>Confusion matrix Batch Size 128</i>	45
Gambar 4. 14 <i>Learning Curve Batch Size 256</i>	46
Gambar 4. 15 <i>Confusion matrix Batch Size 256</i>	47
Gambar 4. 16 <i>Learning Curve</i> tanpa bobot kelas.....	48
Gambar 4. 17 <i>Confusion matrix</i> tanpa bobot kelas	49

ABSTRAK

Pneumonia adalah penyakit pernapasan umum yang sangat memengaruhi kesehatan fisik dan mental seseorang. Hasil rontgen dada untuk membedakan jenis pneumonia seringkali sulit dipahami, terutama jika gambar tidak jelas atau ada elemen yang mengganggu. Oleh karena itu, diperlukan suatu sistem yang dapat membantu dalam identifikasi rontgen dada. Dalam penelitian ini, sistem berbasis web DenseNet201 digunakan untuk mengklasifikasikan jenis pneumonia pada rontgen dada. Ada banyak konfigurasi eksperimen yang berbeda yang digunakan untuk memaksimalkan kinerja model. Penggunaan optimizer seperti Adam, Adamax, dan Adadelata diuji pada masing-masing konfigurasi selama lima puluh epoch. Selama pelatihan dan validasi, konfigurasi Adamax menunjukkan kinerja terbaik. Keakuratan biner 0,9367, keakuratan validasi biner 0,9420, *validation loss* 0,2603 dan *loss* 0,288. Dengan data yang berbeda dari data pelatihan, model ini mencapai tingkat akurasi 0,96, ketepatan 0,96, *recall* 0,96, dan skor F1 0,96 selama fase pengujian.

Kata Kunci: Pneumonia, Densenet201, sistem, rontgen, klasifikasi

ABSTRACT

Pneumonia is a common respiratory disease that greatly affects a person's physical and mental health. The results of chest x-rays to distinguish the type of pneumonia are often difficult to understand, especially if the picture is unclear or there are disturbing elements. In this study, the web-based DenseNet201 system was used to classify the type of pneumonia in the chest x-rays. There are many different experimental configurations used to maximize the performance of the model. Usage of optimizers like Adam, Adamax, and Adadelata were tested on each configuration for fifty epochs. During training and validation, Adamax configurations showed best performance. Binary accuracy 0.9367, binary validation accuracy 0.9420, validation loss 0.2603 and loss 0.288. With different data from training data, the model achieved a precision level of 0.96, precision of 0,96, recall of 0.96 and F1 score of 0,96 during the test phase.

Keywords: Pneumonia, Densenet201, system, x-ray, classification

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah satu kondisi kesehatan yang sangat berbahaya bagi kehidupan manusia adalah pneumonia. Menurut Suartawan (2019), penyakit ini dapat disebabkan oleh infeksi yang menyerang paru-paru oleh bakteri, virus, jamur, atau parasit. Pneumonia dapat dimanifestasikan dengan batuk, demam, kesulitan bernapas, nyeri dada, dan kelelahan (Chasshidi & Putra, 2021). Dokter biasanya melakukan pemeriksaan fisik, analisis darah, dan pemindaian rontgen dada untuk mendiagnosis pneumonia (Kuswanto dkk., 2022). Namun, menurut Rahman dkk. (2020), interpretasi hasil rontgen dada sering kali sulit, terutama dalam kasus di mana gambar tidak jelas atau ada elemen yang mengganggu. Oleh karena itu, diperlukan alat yang dapat membantu dokter menganalisis hasil rontgen dada dengan cepat dan akurat.

Pneumonia adalah penyebab kematian akibat infeksi terbesar pada anak-anak di seluruh dunia. Pada tahun 2019, pneumonia menyebabkan 740.180 kematian pada anak di bawah usia 5 tahun, mencakup 14% dari seluruh kematian anak di bawah usia 5 tahun, tetapi 22% dari seluruh kematian pada anak usia 1 hingga 5 tahun. Pneumonia memengaruhi anak-anak dan keluarga di mana saja, tetapi jumlah kematian paling tinggi terjadi di Asia Selatan dan Afrika sub-Sahara. Anak-anak dapat dilindungi dari pneumonia, dapat dicegah dengan intervensi sederhana, dan dapat diobati dengan obat dan perawatan yang murah dan sederhana. Orang yang berisiko terkena pneumonia juga termasuk orang dewasa berusia di atas 65 tahun dan orang dengan masalah kesehatan predisposisi atau faktor sumber stres yang mempengaruhi individu untuk menghadapi tekanan hidup (WHO, 2022).

Kecerdasan buatan adalah bidang ilmu yang mempelajari bagaimana membuat perangkat lunak atau mesin yang dapat meniru kemampuan kognitif, pembelajaran, dan adaptasi manusia. Dalam bidang kesehatan, kecerdasan buatan telah banyak digunakan untuk diagnosis, pengobatan, pencegahan, dan penelitian. *Watsonx*, *DeepMind Health*, dan *Babylon Health* adalah beberapa contoh penerapan

kecerdasan buatan dalam bidang kesehatan. *Watsonx* adalah sistem kecerdasan buatan yang dikembangkan oleh IBM yang dapat memberikan rekomendasi pengobatan kanker yang sesuai dengan kondisi pasien dengan menganalisis data klinis, jurnal medis, dan studi kasus. *DeepMind Health* adalah contoh penerapan kecerdasan buatan dalam bidang kesehatan. Selain itu, *Watsonx* dapat belajar dari hasil pengobatan sebelumnya dan menyesuaikan rekomendasinya dengan perkembangan terbaru. *Babylon Health* adalah perusahaan kesehatan digital yang menggunakan kecerdasan buatan untuk menyediakan layanan kesehatan yang mudah diakses dan murah, dan *DeepMind Health* mengembangkan *Streams*, sebuah aplikasi yang dapat dengan cepat dan akurat mendeteksi kondisi medis akut seperti sepsis dan gagal ginjal. *Chatbot* dari *Babylon Health* dapat memberikan saran medis, merujuk pasien ke dokter yang tepat, dan menjawab pertanyaan kesehatan umum. Selain itu, fitur *Health Check Babylon Health* memungkinkan Anda menilai tingkat kesehatan dan risiko penyakit seseorang dengan melihat riwayat kesehatan, gaya hidup, dan data pribadi mereka. Robot dapat membantu kesejahteraan masyarakat dan menyelesaikan masalah kesehatan yang rumit. Namun, tantangan dan batasan seperti etika, privasi, keamanan, dan ketergantungan harus diatasi dengan kecerdasan buatan. Akibatnya, pengembangan dan penggunaan AI di bidang kesehatan harus dilakukan dengan hati-hati dan bertanggung jawab.

Metode *Convolutional Neural Network* (CNN) dalam bidang kecerdasan buatan telah menunjukkan kemampuan untuk memahami dan mendiagnosis berbagai penyakit melalui gambar medis, termasuk gambar rontgen pneumonia. Dalam komunitas penelitian ilmu komputer dan kedokteran, metode tersebut telah menjadi sangat penting. Ini karena CNN memiliki kemampuan untuk memahami dan mendiagnosis penyakit dengan lebih baik dengan memanfaatkan pengetahuan yang ada dari model-model yang sudah dilatih pada *dataset* besar. Dalam penelitian yang dilakukan oleh (Irvin dkk, 2019) mereka membuat model *deep learning* yang dapat mengidentifikasi berbagai patologi dalam citra rontgen dada, termasuk pneumonia, dengan menggunakan data besar yang terdiri dari ribuan citra rontgen. Penelitian ini menunjukkan bahwa transfer pembelajaran pada jaringan saraf dapat membantu sistem memahami dan mendiagnosis penyakit dari perspektif medis.

Oleh sebab itu, Penelitian ini bertujuan membuat sistem dalam bidang kesehatan. Densenet201 akan dimanfaatkan untuk membantu dokter atau tenaga medis mendeteksi pneumonia dari gambar rontgen paru-paru. Sistem ini akan menjadi alat yang mudah diakses oleh profesional medis. Mereka dapat dengan mudah memastikan apakah pasien menderita pneumonia dengan mengirimkan hasil citra rontgen pasien ke sistem ini.

1.2 Perumusan Masalah

Bagaimana metode *convolutional neural network* dengan arsitektur densenet201 yang diimplementasikan dalam sistem berbasis website dapat mengklasifikasikan antara paru-paru pneumonia dengan normal dari citra hasil rontgen.

1.3 Pembatasan Masalah

Berikut adalah batasan tugas akhir ini :

1. Sistem hanya dapat mendeteksi pneumonia berdasarkan citra rontgen dada yang diunggah oleh pengguna ke dalam website
2. Sistem tidak dapat mendeteksi penyebab, jenis, atau tingkat keparahan
3. Format ekstensi gambar yang bisa dideteksi oleh sistem adalah .jpeg, .jpg, .png
4. Sistem tidak akan menyimpan inputan gambar hanya memunculkan prediksi saja.

1.4 Tujuan

Tujuan dari penelitian ini, ialah:

1. Menerapkan metode *convolutional neural network* untuk mengklasifikasikan rontgen paru-paru pneumonia dan normal
2. Merancang model *convolutional neural network* dengan arsitektur DenseNet201
3. Membangun sistem yang dapat membedakan paru-paru normal dengan pneumonia

1.5 Manfaat

Sistem yang dibuat dapat membantu pengguna untuk mengklasifikasikan paru-paru pneumonia dengan paru-paru normal.

1.6 Sistematika Penulisan

Sistematika penulisan berikut akan digunakan oleh penulis saat mereka membuat laporan tugas akhir:

BAB I : PENDAHULUAN

Merinci latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan..

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Memaparkan penelitian sebelumnya dengan maksimal tahun terbit dalam 5 tahun terakhir dan teori yang relevan dengan penelitian ini seperti DenseNet 201

BAB III : METODE PENELITIAN

menjelaskan proses penelitian dari pengumpulan data hingga klasifikasi data.

BAB IV : HASIL DAN ANALISIS PENELITIAN

mengungkapkan hasil penelitian, yaitu hasil klasifikasi menggunakan *Convolutional Neural Network* dan pengujian klasifikasi.

BAB V : KESIMPULAN DAN SARAN

Memaparkan kesimpulan serta saran dari penelitian ini

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pneumonia adalah penyakit yang menyerang paru-paru dan dapat disebabkan oleh berbagai mikroba, seperti bakteri, virus, atau jamur. Pneumonia menyebabkan peradangan pada alveoli (kantong udara) paru-paru yang mengganggu pertukaran gas dan menyebabkan gejala sesak napas (Ilmi dkk, 2020). Pneumonia dapat dideteksi dengan menggunakan citra rontgen toraks yang menunjukkan adanya infiltrat atau konsolidasi pada paru-paru. Namun, citra rontgen toraks memiliki tantangan dalam hal kualitas, variasi, dan interpretasi oleh tenaga medis (Hekmatyar dkk, 2022).

Penelitian yang dilakukan oleh (Rahman dkk, 2020) mengenai transfer learning dengan Deep Convolutional Neural Network untuk deteksi Pneumonia menggunakan rontgen dada. Pre-trained Models yang dipakai antara lain Alexnet, ResNet18, DenseNet201, dan SqueezeNet. Lalu optimisasi yang dipakai adalah gradient descent, momentum 0.9, batch 16, serta learning rate 0.0003. Peneliti menggunakan 3 tugas yang salah satunya normal dan pneumonia didapatkan hasil Alexnet dengan akurasi 0.945, ResNet18 dengan akurasi 0.964, DenseNet201 dengan akurasi 0.98, dan SquesNet dengan akurasi 0.961.

Pada peneliatan perbandingan antara DenseNet201 dengan MobileNetV2 yang dilakukan oleh (Dharma Adhinata dkk, 2021) untuk mendeteksi pengguna masker atau tidak. Optimisasi yang digunakan ialah Adam lalu, epoch yang digunakan ada tiga yaitu 10, 20, dan 30. batch size yang dipakai adalah 16. didapatkan hasil pada epoch ke-20 dan ke-30 DenseNet201 mendapatkan hasil lebih baik dari pada MobileNetV2 dengan hasil akuasi 0.995 dan nilai loss 0.00163 . Kemudian peneliti melkukan tes menggunakan 30fps video objek yang dipakai adalah wajah manusia memakai masker dan tidak. hasil tes menunjukkan DenseNet201 dengan *recall* 0.97 sedangkan MobileNetV2 dengan *recall* 0.50. Berdasarkan beberapa penelitian diatas dapat ditarik kesimpulan bahwa penggunaan DenseNet201 Convolutional Neural Network merupakan algoritma

yang tepat dalam penelitian tugas akhir deteksi pneumonia pada citra rontgen paru-paru. DenseNet201 menunjukkan performa bagus dibandingkan MobileNetV2, AlexNet, Resnet18, dan SqueezeNet. DenseNet201 memiliki rata-rata tingkat akurasi diatas 95%. Akurasi dapat terus ditingkatkan dengan mempertimbangkan kualitas citra, pemilihan optimizer, jumlah epoch besar dari learning rate dan masih banyak parameter lainnya.

2.2 Dasar Teori

2.2.1 Pneumonia

Pneumonia adalah suatu kondisi serius yang melibatkan peradangan pada jaringan paru-paru, terutama alveoli (kantong udara paru-paru). Kondisi ini umumnya dipicu oleh infeksi bakteri, virus, atau jamur. Bakteri *Streptococcus pneumoniae* adalah penyebab paling umum dari pneumonia bakterial, sedangkan virus seperti influenza, adenovirus, dan virus pernapasan syncytial (RSV) dapat menyebabkan pneumonia virus. Gejala pneumonia antara lain kesulitan bernapas, demam, batuk berdahak, nyeri dada, dan kelelahan parah. Faktor risikonya antara lain usia tua, sistem kekebalan tubuh yang melemah, merokok, masalah pernapasan kronis, dan kondisi kesehatan lainnya.

Proses diagnosis pneumonia melibatkan evaluasi gejala klinis, pemeriksaan fisik, dan seringkali pemeriksaan radiologi seperti rontgen dada untuk memastikan perubahan pada paru-paru. Pengobatan pneumonia tergantung pada penyebabnya; Infeksi bakteri dapat diobati dengan antibiotik, sedangkan pneumonia virus memerlukan pengobatan simptomatik. Penting untuk mendeteksi dan mengobati pneumonia dengan cepat, karena kondisi ini berpotensi menimbulkan komplikasi serius seperti abses paru, cairan di sekitar paru, atau bahkan syok septik.

Upaya pencegahan pneumonia meliputi vaksinasi, terutama pada kelompok risiko tinggi seperti lansia, anak-anak, dan individu dengan kondisi kesehatan yang melemahkan. Selain itu, menjaga kebersihan tangan dan praktik kebersihan yang baik juga dapat membantu mencegah penyebaran agen penyebab pneumonia. Meskipun pneumonia dapat menyerang siapa saja, kesadaran akan faktor risiko,

deteksi dini, dan pengobatan yang tepat sangat penting untuk mengurangi dampak serius kondisi ini terhadap kesehatan.

2.2.2 Image Augmentation

Image augmentation adalah teknik yang umum digunakan dalam pengolahan citra dan pembelajaran mesin untuk meningkatkan keragaman data pelatihan dengan membuat variasi pada data gambar yang ada. Tujuannya adalah membuat model pembelajaran mesin lebih robust dan mampu mengenali pola atau fitur yang lebih umum.

Berikut adalah beberapa teknik umum dalam image augmentation:

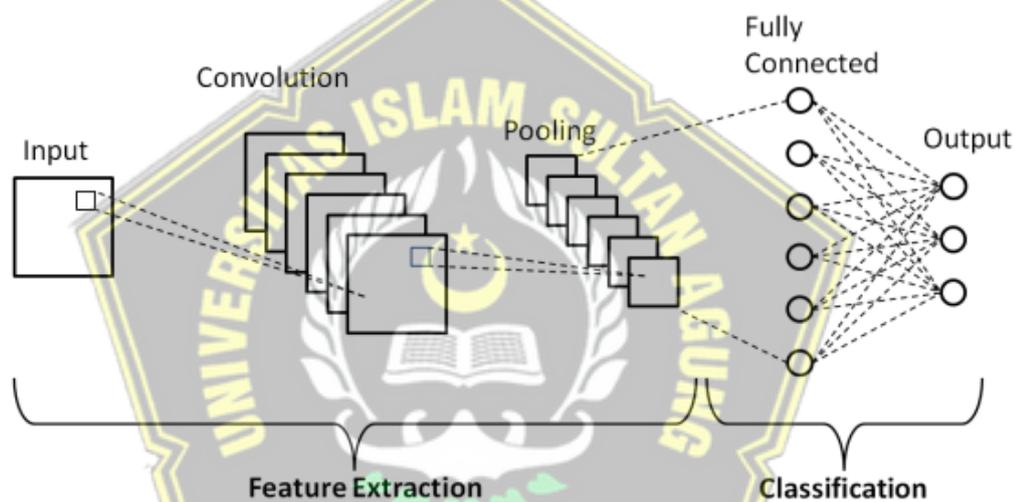
- a. Rotasi: Mengubah sudut rotasi gambar untuk menciptakan variasi.
- b. Flip Horizontal/Vertical: Membalik gambar secara horizontal atau vertikal untuk menambah variasi.
- c. Zoom In/Out: Memperbesar atau memperkecil gambar untuk melatih model terhadap variasi skala objek.
- d. Crop: Memotong bagian acak dari gambar untuk memperkenalkan variasi lokasi objek.
- e. Brightness Adjustment: Menyesuaikan tingkat kecerahan gambar.
- f. Contrast Adjustment: Menyesuaikan tingkat kontras gambar.
- g. Color Saturation Adjustment: Menyesuaikan tingkat saturasi warna gambar.
- h. Noise Injection: Menambahkan noise (misalnya Gaussian noise) ke gambar untuk membuatnya lebih realistis.
- i. Elastic Distortion: Menambahkan distorsi elastis ke gambar untuk memperkenalkan variasi deformasi.
- j. Shear: Memiringkan gambar untuk menciptakan variasi sudut pandang.

Teknik ini efektif untuk mengatasi masalah overfitting, di mana model terlalu fokus pada data pelatihan tertentu dan tidak dapat menggeneralisasi dengan baik pada data baru. Dengan variasi yang diperkenalkan melalui augmentasi, model cenderung lebih baik dalam mengenali objek atau pola pada gambar baru.

2.2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang dibuat untuk memproses gambar dan data lainnya. CNN dapat melakukan banyak hal, seperti pengenalan gambar, deteksi objek, dan segmentasi gambar. Sistem visualisasi hewan dan cara manusia melihat dunia dipengaruhi oleh arsitektur CNN.

CNN pada dasarnya menggunakan prinsip konvolusi dengan menggerakkan *filter* konvolusi tertentu melalui gambar. Komputer dapat memperoleh representasi informasi baru dari hasil *filter* yang diterapkan pada berbagai bagian gambar melalui proses ini (Yumame dkk., 2022).



Gambar 2. 1 Arsitektur CNN (Phung & Rhee, 2019)

Gambar 2.1 di atas menunjukkan gambaran umum arsitektur *Convolutional Neural Network*, yang terdiri dari *input*, *feature extraction*, dan *classification*. Beberapa proses yang terlibat dalam *feature extraction* termasuk:

a. *Convolution Layer*

Convolution Layer bertujuan untuk mengidentifikasi dan mengekstrak fitur-fitur penting dari sebuah gambar dengan menggunakan *filter* atau *kernel*, yang merupakan matriks berukuran kecil yang berisi nilai-nilai tertentu. *Filter* ini akan digeser atau dikonvolusi di setiap bagian gambar, dan menghasilkan sebuah *activation map* atau *feature map*, yang merupakan matriks berisi nilai-nilai hasil operasi *dot product* antara *filter* dan bagian gambar yang bersesuaian. Dengan menggunakan beberapa *filter* yang berbeda, *Convolution Layer* dapat menghasilkan

beberapa *activation map* yang merepresentasikan fitur-fitur yang berbeda dari gambar input. *Convolution Layer* juga biasanya diikuti oleh fungsi aktivasi, seperti ReLU, untuk menambahkan non-linearitas pada *output layer*. Selain itu, *Convolution Layer* juga dapat menggunakan teknik *padding* dan *stride* untuk mengatur ukuran output layer.

b. *Pooling Layer*

Pooling Layer bertujuan untuk melakukan *down sampling* pada peta fitur yang dihasilkan oleh *Convolution Layer*, dengan cara mereduksi ukuran spasial dari peta fitur tanpa menghilangkan informasi penting yang terkandung di dalamnya. *Pooling Layer* juga dapat meningkatkan kecepatan komputasi, mengurangi jumlah parameter, dan mencegah *overfitting*. *Pooling Layer* terdiri dari *filter* atau *kernel*, yang merupakan matriks berukuran kecil yang digeser atau *pooling* di setiap bagian peta fitur. *Filter* ini akan melakukan operasi tertentu pada setiap bagian peta fitur, dan menghasilkan sebuah *output* yang merepresentasikan rangkuman dari bagian tersebut. Operasi *pooling* yang umum digunakan adalah *max pooling* atau mengambil nilai terbesar dari setiap bagian peta fitur, dan *average pooling* atau menghitung nilai rata-rata dari setiap bagian peta fitur. Dengan menggunakan operasi *pooling*, *Pooling Layer* dapat membuat model lebih invarian terhadap pergeseran lokal dari fitur-fitur dalam gambar.

c. *Fully Connected Layer*

Fully Connected Layer terdiri dari neuron-neuron yang masing-masing melakukan transformasi linear pada vektor *input* melalui matriks bobot. Semua neuron pada *layer* sebelumnya, yaitu setelah proses *flattening*, menerima *input* dari *Fully Connected Layer*. Di sini, semua hasil ekstraksi fitur *convolutional neural network* digabungkan. Dalam arsitektur CNN, lapisan penuh terhubung biasanya berada di belakang lapisan *convolutional*, lapisan ReLU, lapisan *pooling*, dan proses *flattening*. *Fully Connected Layer* menghasilkan *output* akhir dari model, seperti kelas atau probabilitas dari gambar *input*. Lapisan ini juga dapat memiliki fungsi aktivasi, seperti softmax atau sigmoid, untuk menambahkan non-linearitas ke *output* lapisan.

2.2.4 Sigmoid

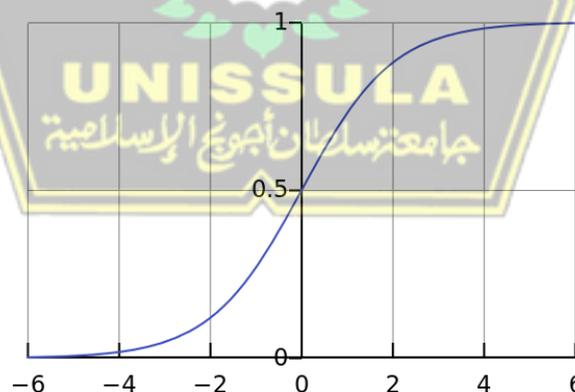
Fungsi sigmoid, juga dikenal sebagai fungsi logistik, adalah fungsi matematika yang sering digunakan dalam pemodelan statistik dan pembelajaran mesin. Fungsi sigmoid memiliki kurva S yang dapat mengubah input kontinu menjadi nilai antara 0 dan 1. Persamaan matematika berikut menjelaskan fungsi ini:

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

keterangan:

- $\sigma(x)$ adalah output dari fungsi sigmoid untuk input x .
- e adalah bilangan Euler, suatu konstanta matematis yang kurang lebih sama dengan 2.71828.

Pembelajaran mesin biasanya menggunakan fungsi sigmoid untuk memetakan nilai input ke dalam rentang $[0, 1]$. Salah satu fungsi utamanya adalah dalam jaringan saraf tiruan, juga dikenal sebagai jaringan saraf tiruan, terutama pada lapisan output, untuk menghasilkan probabilitas atau melakukan fungsi aktivasi di dalam lapisan tersembunyi.



Gambar 2. 2 Grafik Fungsi Sigmoid

Fungsi sigmoid memiliki keuntungan bahwa outputnya berada dalam rentang 0–1, yang dapat dianggap sebagai probabilitas. Namun, sigmoid memiliki beberapa kekurangan, salah satunya adalah masalah yang disebut sebagai "vanishing

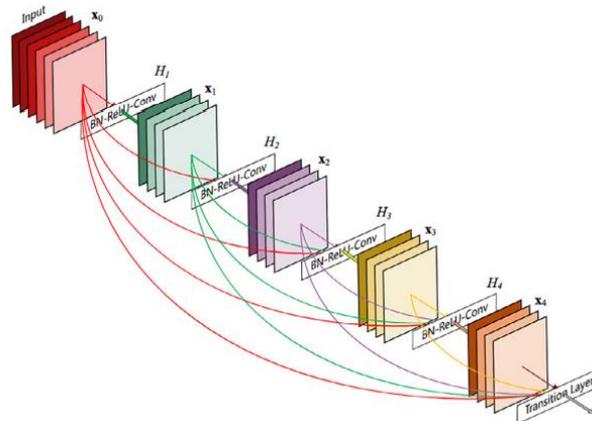
gradient", yang dalam beberapa situasi dapat menghambat pelatihan model. Sebaliknya, fungsi aktivasi tambahan seperti ReLU (*Rectified Linear Unit*) sering digunakan dalam arsitektur jaringan saraf kontemporer.

2.2.5 DenseNet

DenseNet, atau Dense Convolutional Network, adalah arsitektur jaringan saraf konvolusional (CNN) yang dikembangkan untuk meningkatkan efisiensi dan konektivitas antar lapisan jaringan. Dikembangkan oleh Gao Huang, Zhuang Liu, dan Kilian Q. Weinberger, DenseNet memperkenalkan konsep "konektivitas padat" yang membedakannya dari arsitektur CNN lainnya.

Dalam arsitektur DenseNet, setiap lapisan memiliki koneksi langsung dengan semua lapisan di depan dan di belakangnya. Hal ini menciptakan struktur yang padat, atau "padat", di mana informasi dapat mengalir lebih langsung dan efisien melalui jaringan. Dalam jaringan tradisional, seperti ResNet, informasi harus "melompat" atau melewati lapisan yang lebih rendah atau lebih tinggi, namun di DenseNet, setiap lapisan menerima masukan langsung dari semua lapisan di sekitarnya.

Keuntungan utama dari konektivitas padat adalah mengatasi masalah degradasi yang sering terjadi pada jaringan yang lebih dalam. Dengan adanya hubungan langsung antar lapisan, gradien dapat lebih mudah ditransmisikan ke lapisan yang lebih dalam tanpa mengalami degradasi yang berarti. Selain itu, konektivitas yang padat juga membantu mengurangi jumlah parameter yang perlu dipelajari, karena setiap lapisan hanya perlu memahami sisa (perbedaan) keluaran dari lapisan sebelumnya (Farid Naufal & Ferdiana Kusuma, 2021).



Gambar 2.3 Arsitektur DenseNet (Minaee dkk., 2020)

Gambar 2.2 diatas menunjukkan arsitektur DenseNet. arsitektur ini umumnya terdiri dari beberapa blok yang disebut “Dense Blocks”, diikuti dengan lapisan *down sampling* seperti *pooling* atau konvolusi dengan langkah untuk mereduksi dimensi spasial. DenseNet sering digunakan dalam tugas pengenalan gambar dan memberikan kinerja yang baik dengan jumlah parameter yang relatif lebih sedikit dibandingkan dengan beberapa arsitektur serupa lainnya. Dalam beberapa aplikasi, DenseNet juga diperluas ke versi tiga dimensi (3D) untuk menangani data gambar atau video medis.

2.2.6 RESTful API

RESTful API merupakan implementasi praktis dari konsep *Representational State Transfer* (REST) dalam pengembangan perangkat lunak. Hal ini memungkinkan sistem terdistribusi untuk berkomunikasi secara efektif melalui protokol HTTP, menggunakan operasi standar seperti GET, POST, PUT, dan DELETE. Dalam konteks ini, RESTful API mengikuti prinsip dasar REST, termasuk identifikasi sumber daya melalui URI, representasi sumber daya yang dapat dimanipulasi, dan penggunaan metode HTTP standar untuk operasi terbatas pada sumber daya. Penting untuk diperhatikan bahwa URI digunakan sebagai cara untuk mengidentifikasi sumber daya secara unik, sedangkan representasi sumber daya dapat berupa format data seperti JSON atau XML. RESTful API menawarkan keunggulan dalam hal kesederhanaan, skalabilitas, dan keadaan tanpa

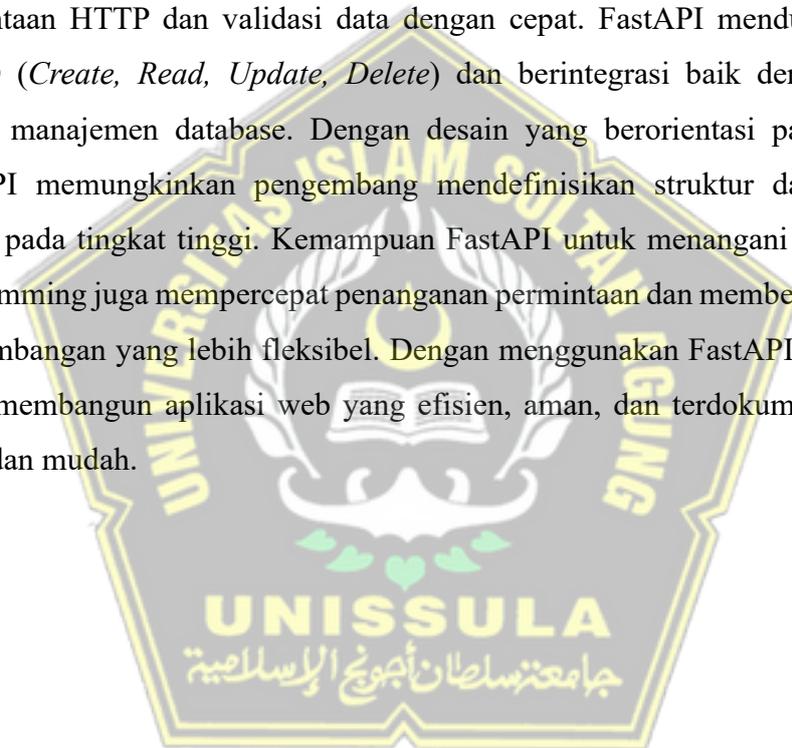
kewarganegaraan. Kesederhanaan ini memungkinkan pengembang untuk memahami dan menggunakan API dengan mudah, sementara skalabilitas mendukung pertumbuhan sistem yang efisien. Sifat stateless memastikan bahwa setiap permintaan dari klien ke server berisi semua informasi yang diperlukan. RESTful API banyak digunakan dalam pengembangan web dan layanan online, memfasilitasi integrasi antara aplikasi dan layanan pihak ketiga. Penggunaan HTTP sebagai protokol komunikasi utama memungkinkan kemudahan akses melalui berbagai platform dan bahasa pemrograman. Dokumen API yang baik juga merupakan komponen penting dalam memanfaatkan RESTful API, memberikan panduan yang jelas kepada pengembang tentang cara menggunakan dan berinteraksi dengan API tersebut. Oleh karena itu, RESTful API menjadi pilihan populer untuk membangun antarmuka yang efisien antara aplikasi dan layanan di lingkungan pengembangan perangkat lunak modern.

2.2.7 Deep Learning

Deep learning adalah salah satu metode dalam kecerdasan buatan yang cara kerjanya mirip seperti otak manusia. Mirip seperti sel-sel pada otak yang saling terhubung satu sama lain membentuk jaringan saraf kompleks. Deep learning menggunakan permodelan abstraksi tingkat tinggi pada data dengan menggunakan fungsi transformasi nonlinear berlapis-lapis. Kompleksitas jaringan saraf deep learning bisa mencapai ratusan lapis. Algoritma ini memiliki kemampuan untuk secara otomatis mengekstraksi fitur dari data. Oleh karena itu, deep learning sangat penting untuk kecerdasan buatan sebab dapat mengurangi beban pemilihan fitur secara eksplisit (Lorentius dkk, 2020).

2.2.8 FastAPI

FastAPI adalah Python yang didesain untuk memberikan performa tinggi, kemudahan penggunaan, dan otomatisasi generasi dokumentasi API. Dikembangkan oleh Sebastián Ramírez, FastAPI memanfaatkan fitur-fitur terbaru dari Python 3.7+ Keunggulan FastAPI terletak pada kecepatan eksekusi dan kemampuannya untuk secara otomatis menghasilkan dokumentasi interaktif dari API yang dibuat (Wijaya dkk, 2022). FastAPI memanfaatkan teknologi seperti Starlette dan Pydantic untuk memastikan performa tinggi dalam menangani permintaan HTTP dan validasi data dengan cepat. FastAPI mendukung operasi CRUD (*Create, Read, Update, Delete*) dan berintegrasi baik dengan berbagai sistem manajemen database. Dengan desain yang berorientasi pada tipe data, FastAPI memungkinkan pengembang mendefinisikan struktur data input dan output pada tingkat tinggi. Kemampuan FastAPI untuk menangani asynchronous programming juga mempercepat penanganan permintaan dan memberikan skenario pengembangan yang lebih fleksibel. Dengan menggunakan FastAPI, pengembang dapat membangun aplikasi web yang efisien, aman, dan terdokumentasi dengan cepat dan mudah.

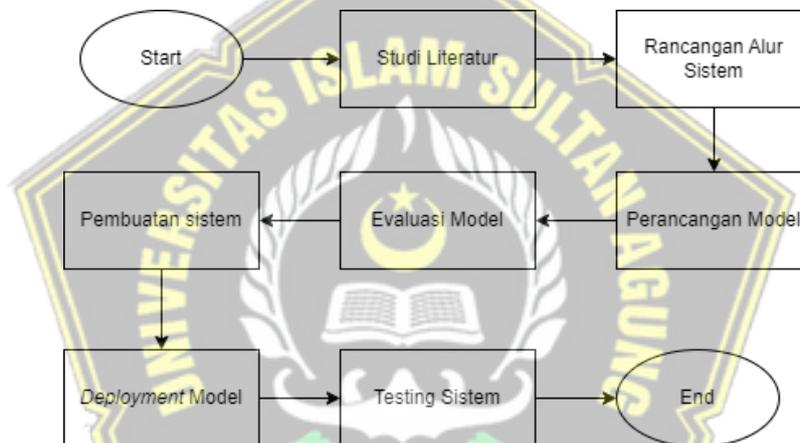


BAB III METODE PENELITIAN

Penelitian ini menguji teori bahwa sistem yang dibangun dapat mendeteksi pneumonia dengan sangat akurat, yang merupakan alasan mengapa penelitian ini dilakukan dengan metode kuantitatif. Hasil rontgen paru-paru dari pasien dengan dan tanpa pneumonia digunakan dalam penelitian ini. Kemudian Densenet201 digunakan untuk menganalisis data.

3.1 Tahapan Penelitian

Tahapan penelitian yang akan dilakukan tertera seperti pada gambar 3.1



Gambar 3. 1 Alur Penelitian

3.2 Studi Literatur

Peneliti akan meninjau beberapa artikel, jurnal, hasil penelitian terdahulu seperti skripsi, serta berbagai situs *Website* untuk bisa mempelajari teori dan berlatih mengenai *deep neural network*, *convolutional neural network*, *transfer learning*, *Website development* dan python.

3.3 Rancangan Sistem

3.3.1 Analisis Kebutuhan Sistem

Perangkat lunak seperti *framework*, *library*, *code editor*, dan bahasa pemrograman yang akan digunakan dalam penelitian ini tercantum pada tabel 3.1

Tabel 3. 1 kebutuhan perangkat lunak pada sistem

Perangkat Lunak	Versi
Python	3.10.11
Tensorflow	2.10.1
Keras	2.10.0
FastApi	0.104.1
Pycharm	2023.2.3
Dataspell	2023.2.2
Jupyter Notebook	1.0.0

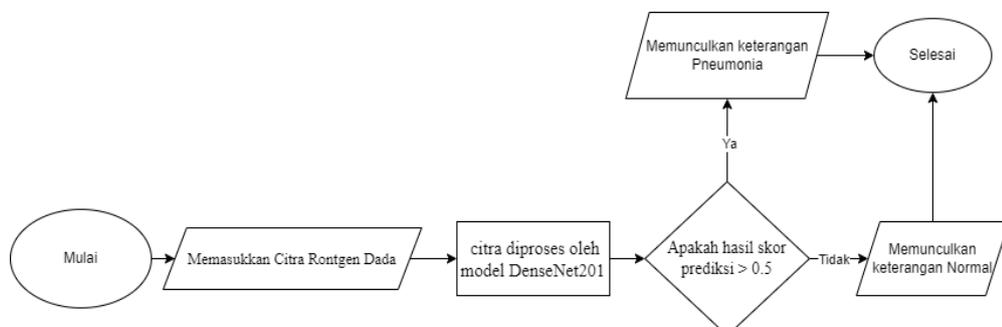
Perangkat keras yang akan digunakan dalam penelitian untuk melatih model adalah laptop HP Omen 15 en0013ax dengan spesifikasi tercantum pada tabel 3.2

Tabel 3. 2 spesifikasi perangkat keras

Nama komponen	spesifikasi
CPU	AMD Ryzen 7 4800H 2.9 Ghz
GPU	Nvidia RTX 2060 6GB
RAM	DDR4 16 GB 3200 Mhz <i>Dual Channel</i>

3.3.2 Flowchart Sistem

Diagram Alir sistem akan seperti pada gambar 7. 2



Gambar 3. 2 Diagram Alir Sistem

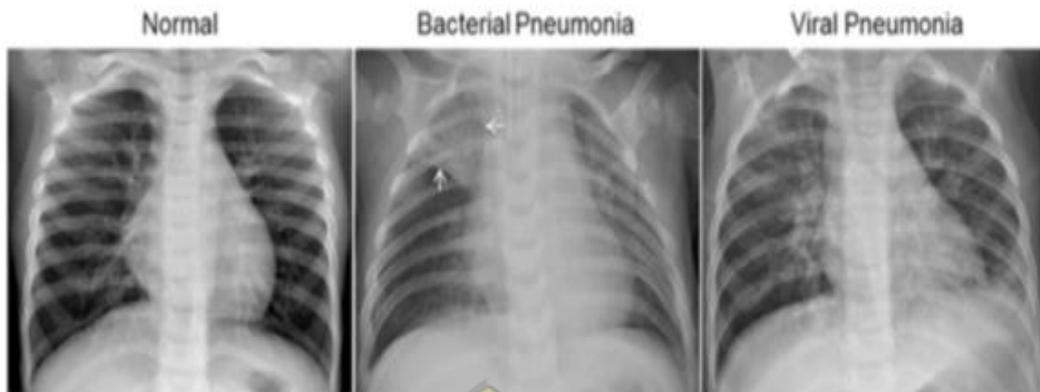
Diagram alir sistem pada Gambar 3. 2 ini terdiri dari beberapa langkah, yaitu:

1. Memasukkan citra rontgen dada. Langkah pertama adalah pengguna memasukkan citra rontgen dada ke dalam sistem. Citra rontgen dada haruslah berformat .jpeg, .png, atau .jpg serta ukuran dari gambar boleh berapa pun.
2. Proses citra oleh model DenseNet201. Citra rontgen dada kemudian diproses oleh model DenseNet20 yang telah dilatih untuk mendeteksi pneumonia pada citra rontgen dada.
3. Prediksi pneumonia. Model DenseNet201 akan memberikan skor dari 0 sampai 1 dan prediksi apakah pasien menderita pneumonia atau tidak.
4. Aktivasi sigmoid pada layer teratas DenseNet201 menghasilkan skor prediksi yang menunjukkan apakah pasien menderita pneumonia atau tidak. Skor yang melebihi 0,5 menunjukkan bahwa pasien menderita pneumonia, sedangkan skor yang kurang atau sama dengan 0,5 menunjukkan bahwa pasien tidak menderita pneumonia. Praktisi kesehatan atau sistem medis yang menggunakan model klasifikasi biner menemukan hasil prediksi lebih mudah dengan fungsi sigmoid dengan batas ambang 0,5.

3.4 Perancangan Model

3.4.1 Pengumpulan Data

Tahap pertama yaitu pengumpulan data berupa citra paru-paru terkena pneumonia dan paru-paru normal yang terdiri dari 1583 Normal dan 4273 Pneumonia. Data tersebut penulis ambil dari jurnal yang dibuat oleh Kermany, dkk (2018). sample data gambar ditunjukkan pada gambar 3.3



Gambar 3. 3 Gambar pada *Dataset*

3.4.2 Data *Preprocessing*

Tabel 3. 3 pembagian data *training* dan *validation*

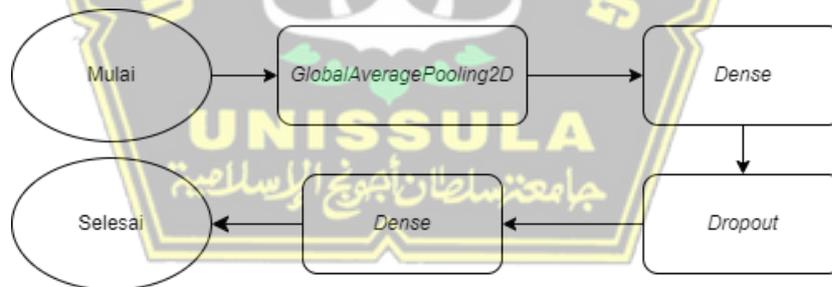
Nama Data	Persentase	Jumlah Data	Pneumonia	Normal
Data <i>Training</i>	80%	4204	1026	3178
Data <i>Validation</i>	20%	1052	257	795

Jumlah data: 5256

Validasi *dataset* menggunakan metode *Holdout Validation* yaitu *dataset* dibagi menjadi dua *training* dan *validation* seperti yang ada pada tabel 3.3. Dengan jumlah data *training* 80% dan *validation* 20%. Citra *dataset* dinormalisasikan menggunakan *preprocess_input* pada parameter *preprocessing_function* di class *ImageDataGenerator* agar gambar yang dimasukkan kedalam model memiliki nilai piksel yang diharapkan oleh model DenseNet. Kemudian citra akan di-*augmentasi* untuk mendapatkan variasi agar model yang dibuat dapat mengetahui gambar secara umum dan menghindari *overfitting*. variasi *augmentasi* yang akan digunakan yaitu *width shift range*, *height shift range*, *shear range*, *rotation range*, dan *zoom range*. Selanjutnya citra akan di-*resize* dengan ukuran 224px * 224px agar semua citra memiliki ukuran yang sama dan ukuran tersebut memberikan keseimbangan yang baik antara kualitas representasi dan sumber daya komputasi yang dibutuhkan.

3.4.3 Data *training*

Data akan dilatih menggunakan pre-trained model DenseNet201. *Batch size* yang akan digunakan adalah 128. Kemudian bobot kelas akan dihitung agar tidak terjadi bias pada pelatihan model yang disebabkan oleh data yang tidak seimbang antara normal dengan pneumonia. Model akan dibuat dengan parameter inputan lebar 224 piksel, tinggi 224 piksel, 3 warna primer. *Fully Connected Layer* bagian atas atau bagian klasifikasi dari model tidak akan disertakan. Bobot yang akan digunakan ialah bobot yang telah dilatih pada *dataset* imagenet. Semua lapisan pada model DenseNet201 akan diatur *untrainable* untuk mempercepat komputasi dan bobot dari model tidak berubah. Selanjutnya pada bagian atas model ditambahkan lapisan GlobalAveragePooling2D, Dense dengan 256 unit neuron dan aktivasi ReLU, Dropout dengan tingkatan 0,4 untuk mengurangi *overfitting*, serta Dense dengan 1 unit neuron dan aktivasi sigmoid. Model kemudian digabungkan antara model dasar densenet201 dengan model yang output yang sudah dibuat. Berikutnya model di-*compile* menggunakan optimizer adam dengan learning rate 0.001, fungsi *loss binary_crossentropy* untuk tugas klasifikasi biner, dan mengukur akurasi biner. gambar 3.4 menggambarkan alur kerja *layer output*.



Gambar 3.4 *flowchart layer output*

Model akan dilatih dengan data latihan dan validasi yang sudah di-*augmentasi* sebelumnya. Jumlah *epochs* yang dipakai adalah 50. parameter *class_weight* akan ditambahkan untuk mengatasi ketidak seimbangan *dataset*.

3.5 Evaluasi Model

Pada tahap ini, *dataset* yang digunakan tidak sama dengan *dataset* yang digunakan untuk pelatihan dan validasi. Tahap ini bertujuan untuk menguji akurasi model yang dibuat. Untuk menghitung akurasi, *confusion matrix* digunakan untuk memberikan informasi tentang ketepatan, akurasi, *recall*, dan skor F1 untuk memastikan model berjalan dengan baik.

3.6 Pembuatan Sistem

3.6.1 Frontend



Gambar 3. 5 Rancangan *frontend*

Pada gambar 3.5 merupakan rancangan tampilan dari frontend sistem. Bahasa yang akan digunakan untuk membuat frontend tersebut ialah html, css, dan javascript.

3.6.2 Backend



Gambar 3. 6 Rancangan *backend*

Gambar 3.6 adalah rancangan *backend* pada sistem akan menggunakan bahasa python dengan *framework* FastApi dengan *endpoint* “/” untuk ke tampilan awal sistem dengan metode GET dan “/uploadfile/” untuk melakukan klasifikasi dengan metode POST.

3.7 Deployment Model



Gambar 3. 7 flowchart *Deployment* model

Gambar 3.7 menjelaskan tentang alur *deployment* model menggunakan FastApi dengan bantuan *library* Tensorflow. model dengan format .h5 akan di-load lalu akan diinputkan image kedalam sistem.

3.8 Testing Sistem

Pada tahap *testing*, sistem akan diuji menggunakan *blackbox* guna menemukan *bug* atau kekurangan pada sistem.

BAB IV HASIL DAN ANALISIS PENELITIAN

4.1 Pengolahan Data

4.1.1 *Preprocessing*

Agar model bisa menghasilkan hasil yang optimal dan data bisa diterima oleh model perlu dilakukan *preprocessing*. penelitian ini menggunakan beberapa tahapan, yaitu:

a. *resize*

Citra diubah menjadi ukuran 224 * 224 piksel menggunakan kelas *ImageDataGenerator* pada pustaka *tensorflow*.

b. *Image Augmentation*

Kumpulan data gambar di-*augmentasi* dengan parameter *shear range*, *zoom range*, *width shift range*, dan *height shift range* nilainya 0.2 serta *rotation range* bernilai 20 menggunakan kelas *ImageDataGenerator*.



Gambar 4. 1 Hasil *Augmentasi* Gambar

Gambar 4.1 merupakan hasil dari *augmentasi* gambar menggunakan kelas *ImageDataGenerator* pada pustaka *tensorflow*. terlihat bahwa setiap gambar memiliki pola variasi yang berbeda. antar satu gambar dengan gambar yang lain divariasikan sehingga menghasilkan luaran yang berbeda-beda satu sama lain.

4.2 Cara Kerja Sistem

4.2.1 Unggah Gambar

Hal yang pertama kali dilakukan pengguna setelah muncul tampilan *website* ialah mengunggah gambar rontgen paru-paru. format gambar yang bisa diterima sistem adalah jpg, png, dan jpeg. sistem hanya bisa mengambil gambar dari penyimpanan lokal.

4.2.2 Pemrosesan Gambar oleh Model

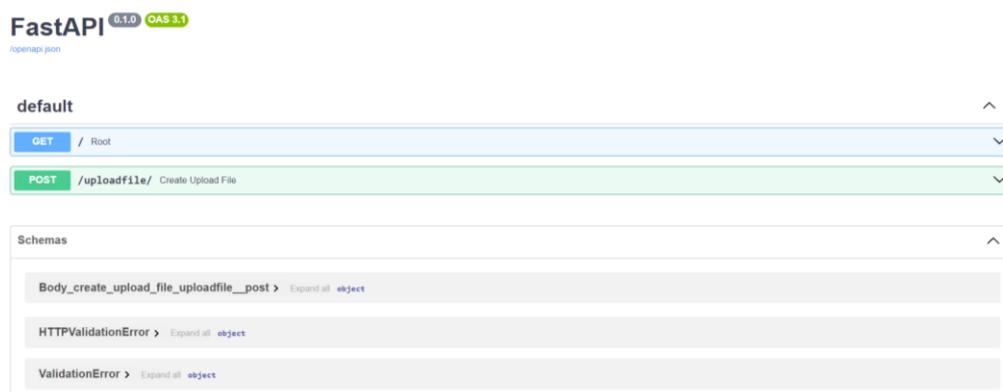
Gambar yang sudah terunggah diubah menjadi ukuran 224*244 piksel. kemudian dikonversi ke bentuk *array*. proses tersebut dilakukan agar gambar sesuai dengan format yang dikehendaki oleh model klasifikasi gambar rontgen paru-paru menggunakan *Convolutional Neural Network* arsitektur DenseNet201.

4.2.3 Klasifikasi Model

Model berformat h5 yang sudah dibuat dimasukkan kedalam kode *backend*. Kemudian setelah Model dimasukkan dan gambar berhasil diubah pada tahap sebelumnya maka gambar tersebut diproses oleh model untuk mendapatkan skor prediksi.

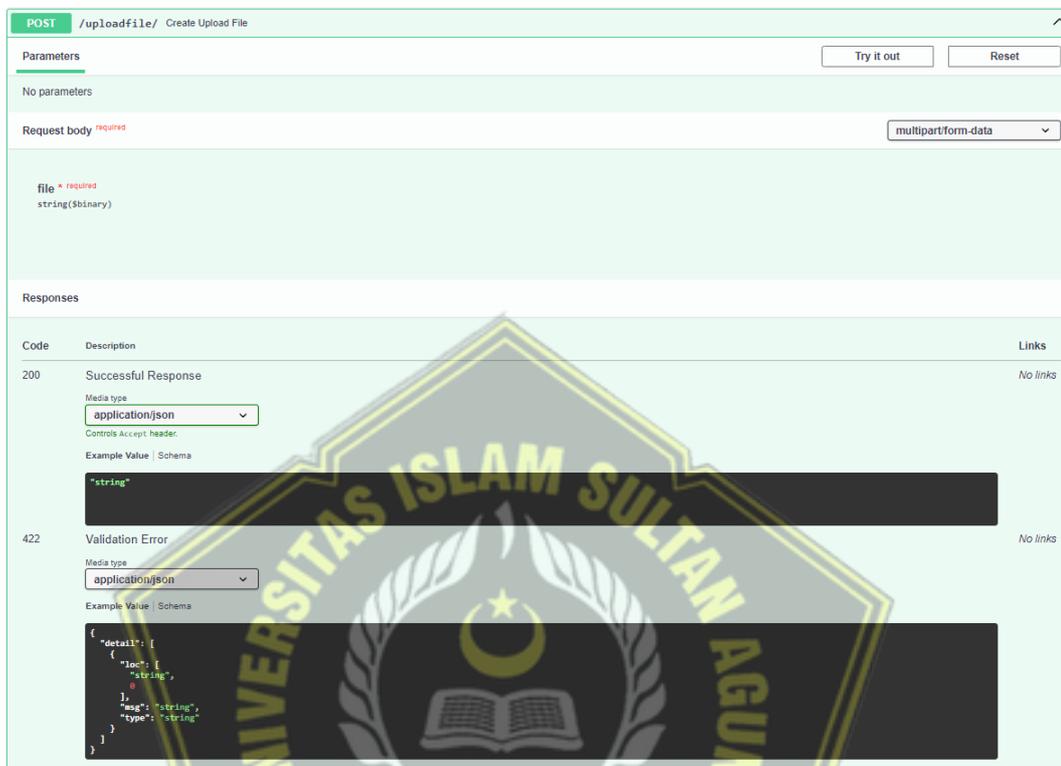
4.3 Implementasi Backend

Tujuan dari pembuatan *Backend* pada sistem ini adalah untuk melakukan *deploy* model yang telah dilatih. *Method* yang dipakai adalah POST dengan *endpoint* “/uploadfile?”. lalu agar dapat mempermudah kerja *frontend* perlu dibuat dokumentasi API salah satunya menggunakan Swagger.



Gambar 4. 2 Dokumentasi API *Backend*

Gambar 4.2 merupakan tampilan dokumentasi API *backend* pada sistem yang terdiri dari dua *endpoint* yaitu “/” dan “/uploadfile”.

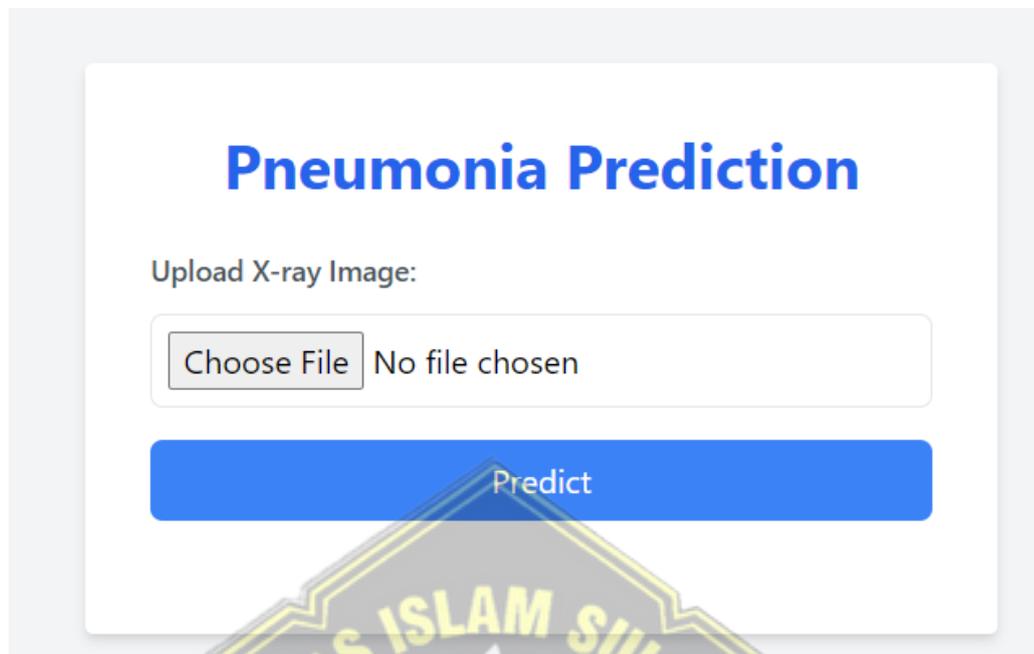


Gambar 4.3 Spesifikasi API Unggah File

Gambar 4.3 merupakan isi *request* dan *response* dari *endpoint* “/uploadfile/” serta respon dari kode status HTTP 200 dan 422.

4.4 Implementasi *Frontend*

Setelah membuat *backend* langkah selanjutnya adalah membuat *frontend*. pada sistem ini *frontend* dipanggil pada *endpoint* “/”. Hal itu dapat terjadi karena menggunakan *engine* jinja2 untuk bisa menampilkan *static web* pada FastApi.



Pneumonia Prediction

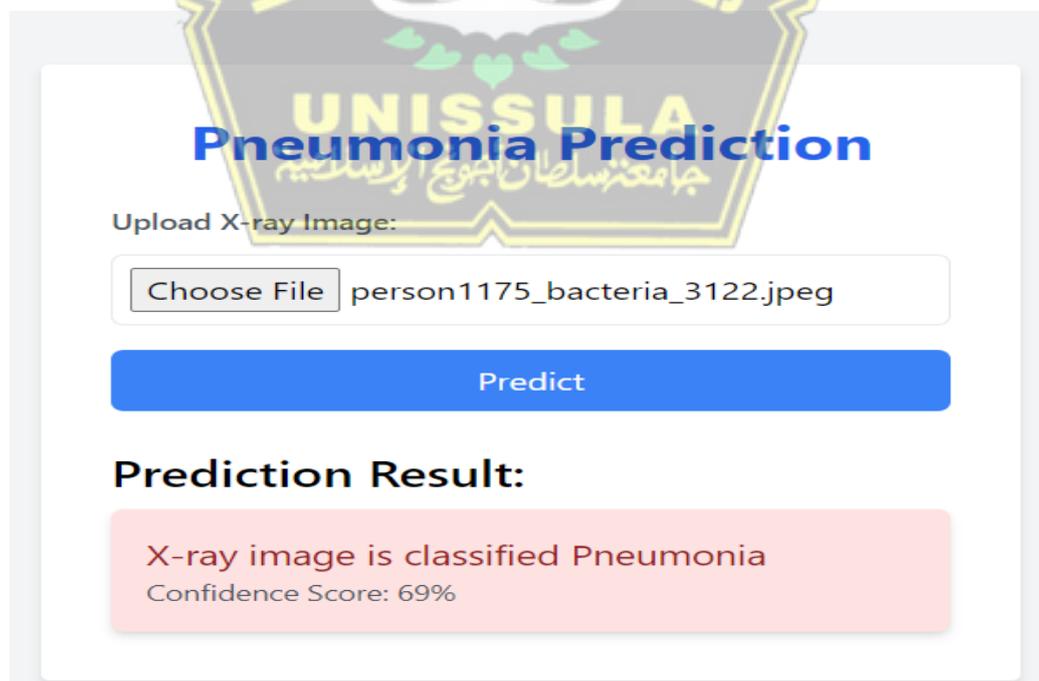
Upload X-ray Image:

Choose File No file chosen

Predict

Gambar 4. 4 Halaman Utama Sistem

Gambar 4. 3 adalah tampilan awal dari sistem. pengguna harus menekan tombol “Choose File” setelah itu memilih gambar yang dikehendaki. kemudian menekan tombol “Predict” untuk mendapatkan skor prediksi.



Pneumonia Prediction

Upload X-ray Image:

Choose File person1175_bacteria_3122.jpeg

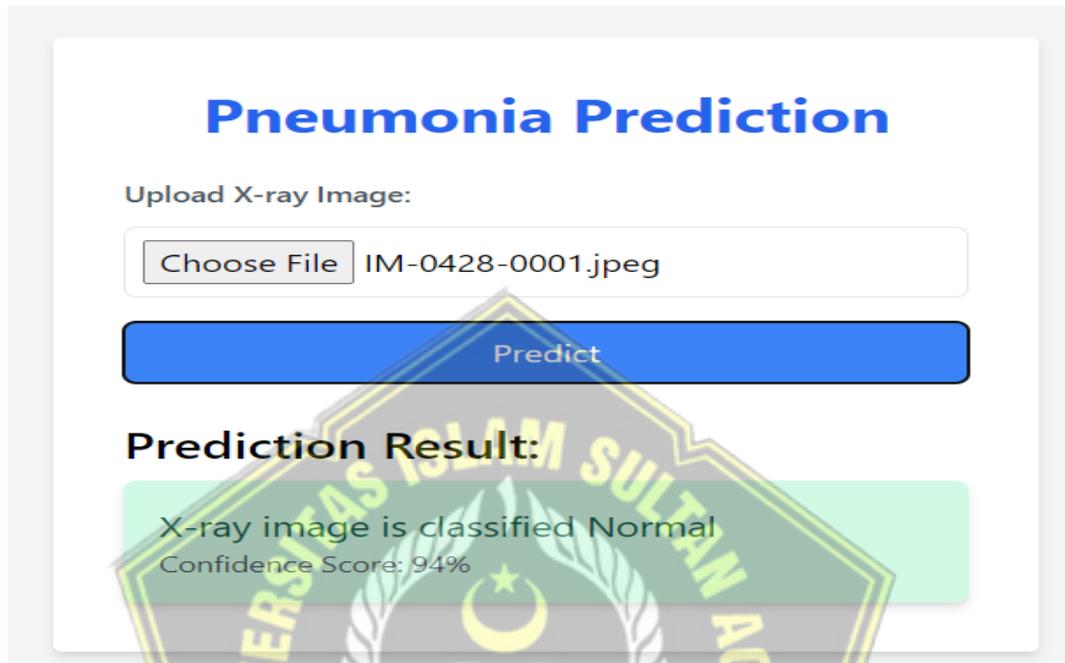
Predict

Prediction Result:

X-ray image is classified Pneumonia
Confidence Score: 69%

Gambar 4. 5 Tampilan Sistem Saat Prediksi Pneumonia

Gambar 4. 4 menunjukkan hasil gambar rontgen paru-paru terindikasi pneumonia. dengan tampilan berwarna merah dan ada hasil skornya.



Gambar 4. 6 tampilan sistem saat prediksi normal

Gambar 4. 5 menunjukkan hasil gambar rontgen paru-paru terindikasi normal. dengan tampilan berwarna hijau dan ada hasil skornya.

4.5 Pengujian Sistem

Pada tahap ini, sistem diuji menggunakan metode *black box testing*. Metode ini adalah teknik pengujian perangkat lunak yang bertujuan untuk menguji fungsionalitas sistem tanpa mengetahui rincian struktur atau cara kerja internalnya. Function testing, non-function testing, dan regression testing adalah beberapa jenis black box testing yang umum digunakan.

Penelitian ini menguji semua fungsi sistem. Tabel 4.1 menunjukkan hasil pengujian berbagai fungsi pada sistem ini, yang bertujuan untuk memastikan bahwa semua fungsi berjalan dengan benar dan sesuai dengan spesifikasi yang telah ditetapkan. Tabel ini juga menunjukkan apakah fungsi-fungsi tersebut berhasil berjalan dengan benar atau apakah ada kesalahan yang perlu diperbaiki.

Tabel 4. 1 Pengujian *Black Box*

Skenario Pengujian	Kasus Pengujian	Hasil Pengujian	Kesimpulan
Membuka <i>website</i>	Menampilkan halaman utama	Sesuai	Valid
unggah gambar dari penyimpanan internal	Melakukan <i>upload</i> gambar rontgen paru-paru dari <i>storage</i>	Sesuai	Valid
Menekan <i>button</i> “Klasifikasi”	Menekan <i>button</i> untuk menampilkan hasil klasifikasi	Sesuai	Valid
<i>Testing</i> gambar rontgen paru-paru pneumonia	Melakukan <i>testing</i> pada gambar paru-paru yang menunjukkan pneumonia	Sesuai	Valid
<i>Testing</i> gambar rontgen paru-paru normal	Melakukan <i>testing</i> pada gambar paru-paru yang menunjukkan normal	Sesuai	Valid
<i>Testing</i> <i>result</i> klasifikasi	Mengecek hasil klasifikasi gambar rontgen	Sesuai	Valid

4.6 Hasil dan Analisis

4.6.1 Perbandingan *Optimizer*

Dataset yang telah divariasikan digunakan untuk pelatihan dan validasi. Data pelatihan terdiri dari 4204 sampel, dengan 1026 sampel dalam kelas normal dan 3178 sampel dalam kelas pneumonia; validasi terdiri dari 1052 sampel yang ditambahkan, dengan 257 sampel dalam masing-masing kelas.

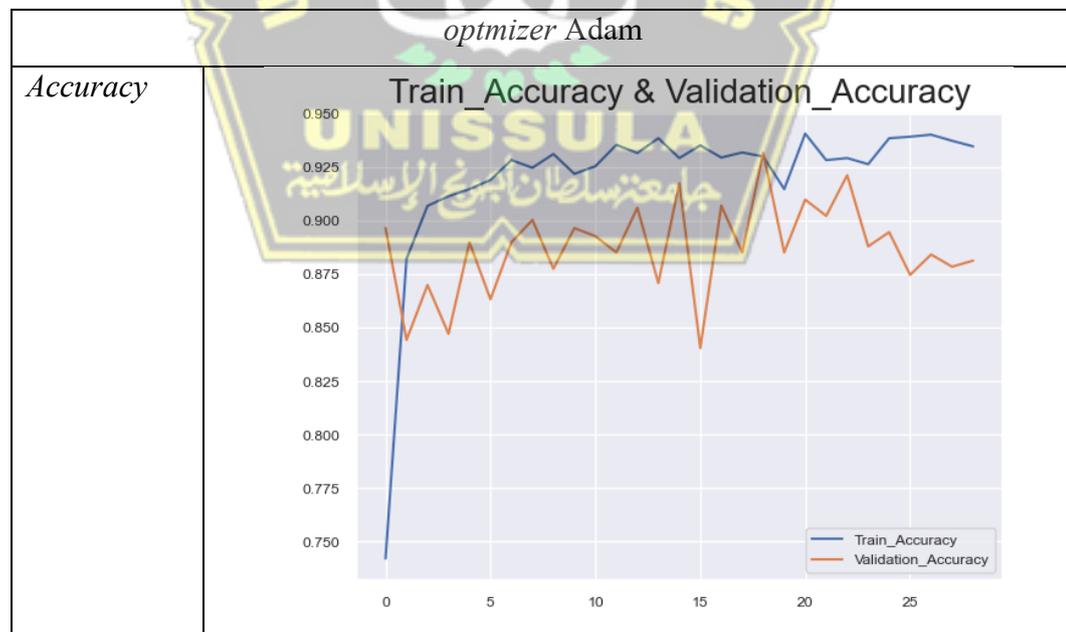
Callback early stopping, *reduce lr*, dan *model checkpoint* digunakan pada penelitian ini dengan parameter monitornya adalah *validation loss*. *early stopping* berfungsi untuk menghentikan proses pelatihan apabila tidak terjadi peningkatan

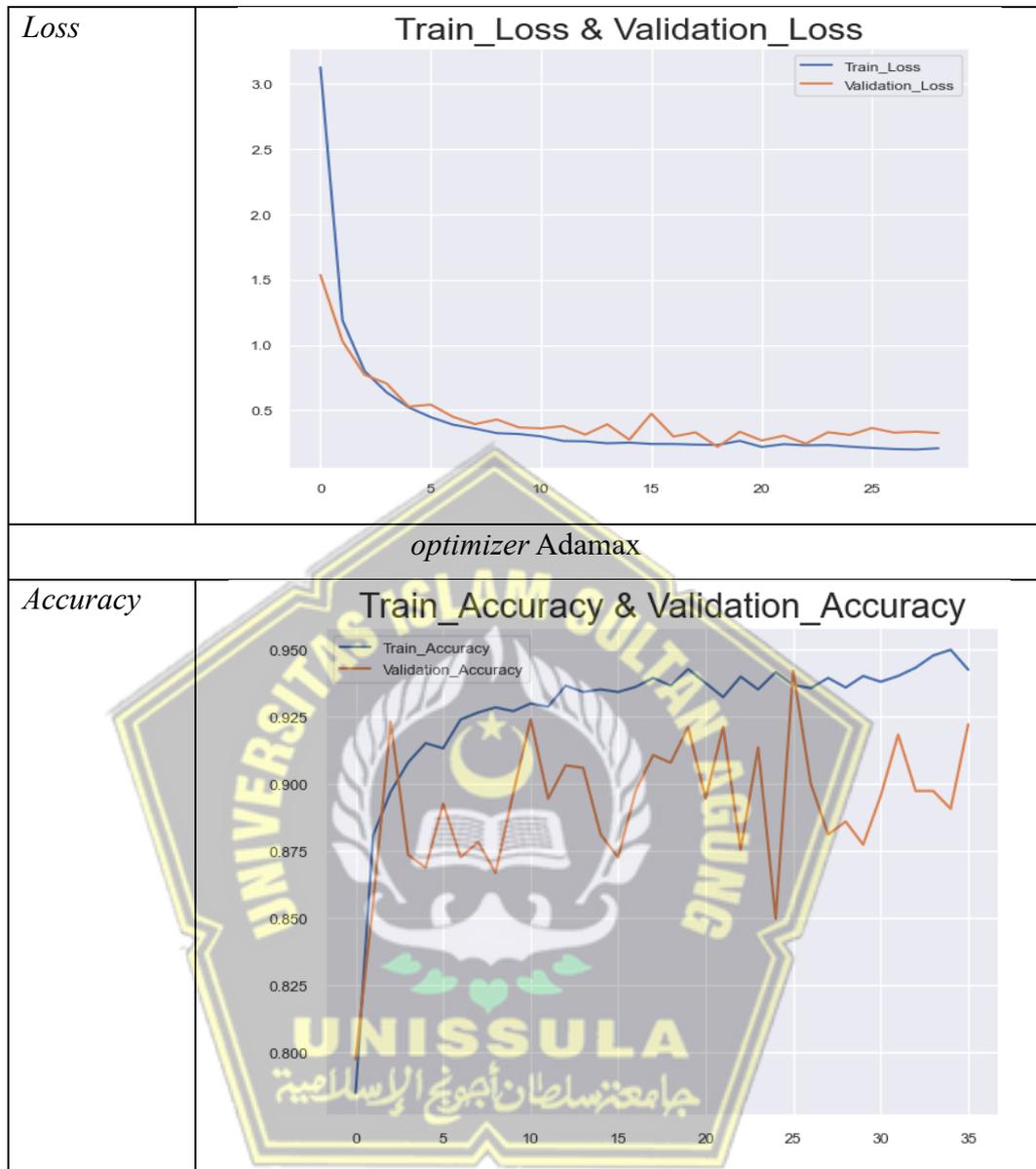
akurasi. *reduce lr* berguna untuk mengurangi *learning rate* pada saat proses pelatihan jika tidak terjadi peningkatan akurasi. *model checkpoint* berperan untuk menyimpan model terbaik berdasarkan akurasi tertinggi. *patience* atau waktu tunggu jika tidak terjadi peningkatan akurasi dari *early stopping* 10 kali sedangkan *reduce lr* 5 kali.

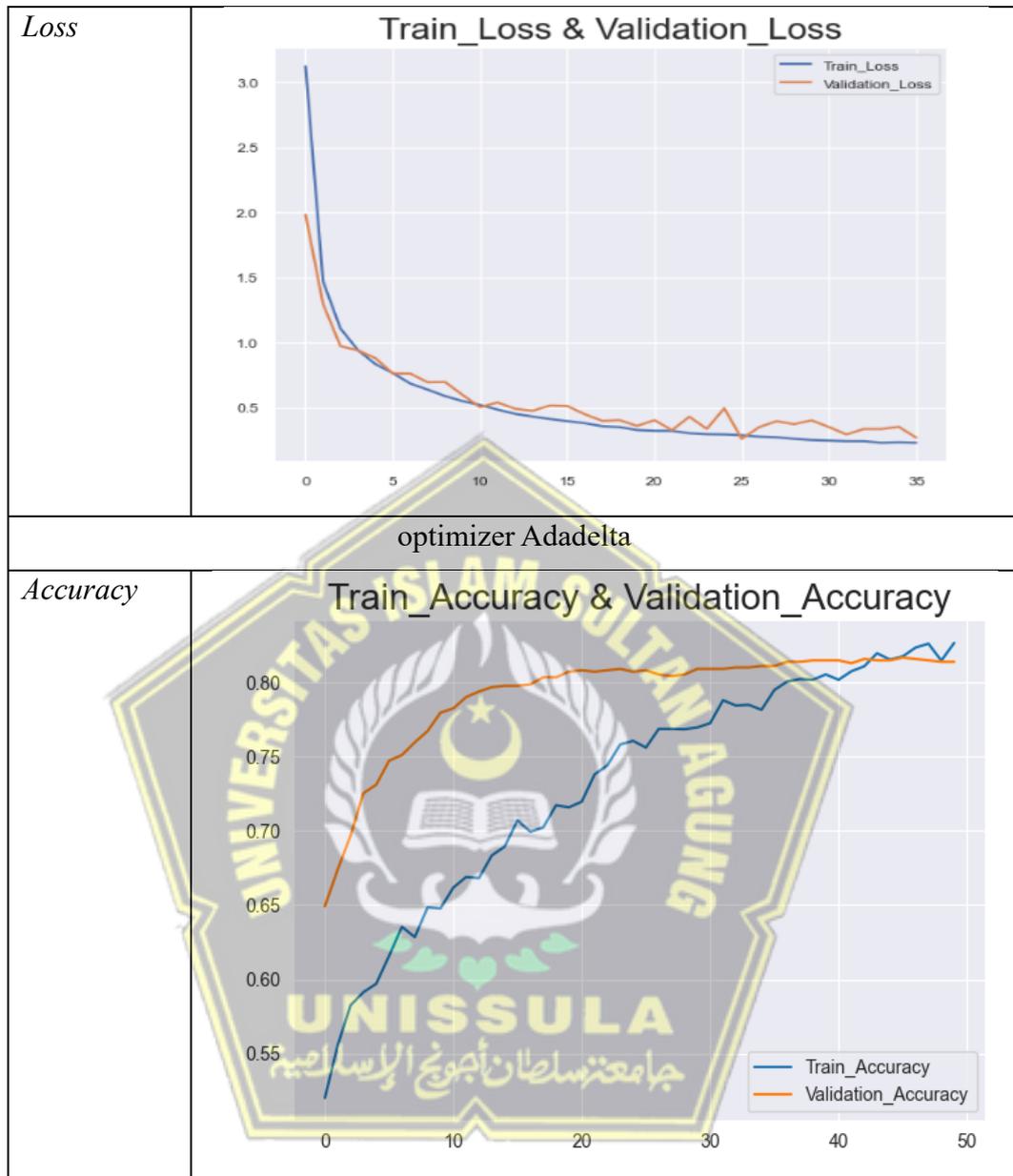
Hasil pelatihan dan validasi ditampilkan dalam grafik yang menunjukkan *accuracy* dan *loss*. Grafik juga menunjukkan hasil konfigurasi yang telah digunakan. Untuk mendapatkan pemahaman visual tentang bagaimana model berfungsi selama proses pelatihan dan validasi, grafik ketepatan dan kehilangan nilai dapat digunakan. Sebaliknya, konfigurasi yang digunakan dalam penelitian ini mengacu pada *optimizer* yang digunakan.

Grafik yang menunjukkan ketepatan dan kehilangan untuk berbagai konfigurasi dapat ditemukan di Tabel 4.2. Sementara grafik pelatihan dan validasi semakin dekat, grafik ketepatan semakin meningkat dan grafik kehilangan semakin menurun. Ini menunjukkan peningkatan kinerja model tersebut.

Tabel 4. 2 Grafik *accuracy* dan *loss*









Namun, model terbaik untuk setiap konfigurasi pada tahap pelatihan disajikan dalam tabel 4.3. Karena memiliki akurasi binar 0,9367, kehilangan 0,2881, akurasi binar validasi 0,9420, dan kehilangan validasi 0,2603, Adamax adalah konfigurasi terbaik dari ketiga konfigurasi tersebut. Sementara grafik ketepatan menunjukkan penurunan, kecenderungan menunjukkan peningkatan. Selain itu, grafik pelatihan dan validasi pada konfigurasi Adamax akan terlihat saling berdekatan, menunjukkan bahwa model tersebut bekerja dengan baik dan stabil dalam melakukan tugasnya.

Tabel 4. 3 Hasil konfigurasi

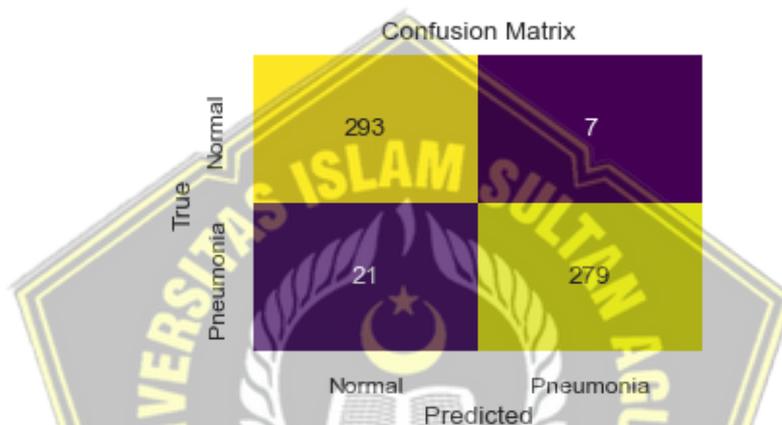
	<i>Epoch</i>	<i>loss</i>	<i>binary accuracy</i>	<i>val-loss</i>	<i>val-binary accuracy</i>	<i>learning rate</i>
Adam	19/50	0.2358	0.9298	0.2207	0.9316	0.001
Adamax	26/50	0.2881	0.9367	0.2603	0.9420	0.001
Adadelta	50/50	4.6946	0.8264	4.7140	0.8137	0.001

4.6.2 Testing Setiap Optimizer

Proses pengujian dilakukan untuk mengevaluasi seberapa akurat sistem dalam mengklasifikasikan data pengujian, yang terdiri dari dua kelas dan total 600 gambar. Untuk mengukur tingkat akurasi, *confusion matrix* digunakan untuk

menghitung nilai akurasi, *precision*, *recall*, dan *F1-Score*. Proses perhitungan ini dilakukan terhadap ketiga konfigurasi yang berbeda dalam arsitektur DenseNet201. Tujuan dari langkah ini adalah untuk menentukan konfigurasi mana yang akan melakukan klasifikasi rontgen paru-paru dengan kinerja terbaik berdasarkan nilai akurasi, *precision*, *recall*, dan *F1-Score* dari masing-masing konfigurasi. Proses perhitungan yang dilakukan terhadap ketiga konfigurasi yang berbeda dijelaskan di sini:

a. Adam



Gambar 4. 7 Confusion matrix adam

Tabel 4.4 dan 4.5 menunjukkan hasil dari *precision*, *accuracy*, *F1-Score*, *recall* dari optimisasi Adam, yang dapat digunakan untuk menghitung nilai matriks berdasarkan mereka.

Tabel 4. 4 Skor *precision*, *recall*, dan *F1-Score* Adam

	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>optimizer</i> Adam	0.95	0.95	0.95	0.95

Tabel 4.5 menampilkan hasil untuk masing-masing kelas yang diuji dengan optimisasi Adam, termasuk ketepatan, *recall*, skor f1, dan jumlah data.

Tabel 4. 5 Skor *precision*, *recall*, dan *F1-Score* Adam tiap kelas

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.93	0.98	0.95
Pneumonia	0.98	0.93	0.95

Tabel 4.6 dan 4.8 menunjukkan hasil *precision*, *accuracy*, *recall*, dan *F1-Score* untuk masing-masing kelas. Nilai *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN) ditunjukkan untuk masing-masing kelas jika dihitung secara manual, tetapi hasilnya tidak jauh berbeda dan hampir sama dengan perhitungan otomatis.

Tabel 4. 6 *Confusion matrix* Adam tiap kelas

Kelas	TP	FP	FN	TN
Normal	293	21	7	279
Pneumonia	279	7	21	293

Tabel 4.7 berikut menampilkan perhitungan *precision*, *recall*, dan *F1-Score* untuk masing-masing kelas berdasarkan nilai TP, FP, dan juga FN.

Tabel 4. 7 Perhitungan manual *precision*, *recall*, dan *F1-Score* Adam tiap kelas

Kelas	Perhitungan	Hasil
	$Precision = \frac{TP}{TP+FP}$	
P_{Normal}	$\frac{TP_{Normal}}{TP_{Normal} + FP_{Normal}} = \frac{293}{293 + 21} = \frac{293}{314}$	0.93
$P_{Pneumonia}$	$\frac{TP_{Pneumonia}}{TP_{Pneumonia} + FP_{Pneumonia}} = \frac{279}{279 + 7} = \frac{279}{284}$	0.98

$Recall = \frac{TP}{TP + FN}$		
R_{Normal}	$\frac{TP_{Normal}}{TP_{Normal} + FN_{Normal}} = \frac{293}{293 + 7} = \frac{293}{300}$	0.976
$R_{Pneumonia}$	$\frac{TP_{Pneumonia}}{TP_{Pneumonia} + FN_{Pneumonia}} = \frac{279}{279 + 21}$ $= \frac{279}{300}$	0.93
$F1 - Score = \frac{2 * precision * recall}{precision + recall}$		
$F1_{Normal}$	$\frac{2 * (P_{Normal} * R_{Normal})}{(P_{Normal} + R_{Normal})}$ $= \frac{2 * (0.93 * 0.976)}{0.93 + 0.976} = \frac{1.81}{1.90}$	0.95
$F1_{Pneumonia}$	$\frac{2 * (P_{Pneumonia} * R_{Pneumonia})}{(P_{Pneumonia} + R_{Pneumonia})}$ $= \frac{2 * (0.98 * 0.93)}{0.98 + 0.93} = \frac{1.82}{1.91}$	0.95

Tabel 4.8 berikut menunjukkan perhitungan *precision*, *accuracy*, *recall*, dan *F1-Score* konfigurasi Adam yang dilakukan secara manual. Perhitungan ini didasarkan pada nilai TP, jumlah data, jumlah kelas, jumlah *precision*, jumlah *recall*, dan jumlah skor *F1-Score* masing-masing kelas.

Tabel 4. 8 perhitungan manual *precision*, *recall*, dan *F1-Score* Adam

	Perhitungan	Hasil
<i>Accuracy</i>	$\frac{TP_{Normal} + TP_{Pneumonia}}{Jumlah\ Data} = \frac{293 + 279}{600}$ $= \frac{572}{600}$	0.95
<i>Precision</i>	$\frac{P_{Normal} + P_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.93 + 0.98}{2} = \frac{1.91}{2}$	0.95
<i>Recall</i>	$\frac{R_{Normal} + R_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.93 + 0.976}{2}$ $= \frac{1.9}{2}$	0.95
<i>F1-Score</i>	$\frac{F1_{Normal} + F1_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.95 + 0.95}{2}$ $= \frac{1.9}{2}$	0.95

b. Adamax

Confusion Matrix

True	Normal	293	7
	Pneumonia	17	283
		Normal	Pneumonia
		Predicted	

Gambar 4. 8 *Confusion matrix* Adamax

Tabel 4.9 dan 4.10 menunjukkan hasil untuk *precision*, *accuracy*, *recall*, dan *F1-Score* dari Adamax, yang dapat digunakan untuk menghitung nilai matriks pada gambar 4.7.

Pada tabel 4.9 menampilkan hasil *accuracy*, *precision*, *recall* dan *F1-Score* dari *optimizer* Adamax.

Tabel 4. 9 Skor *precision*, *recall*, dan *F1-Score* Adamax

	<i>accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>optimizer</i> Adamax	0.96	0.96	0.96	0.96

Pada tabel 4.10 ini menampilkan hasil *precision*, *recall*, *F1-Score*, dan jumlah data pada masing-masing kelas yang dilakukan *testing* menggunakan Adamax.

Tabel 4. 10 Skor *precision*, *recall*, dan *F1-Score* Adamax tiap kelas

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.95	0.98	0.96
Pneumonia	0.98	0.94	0.96

Tabel 4.11 dan 4.13 menunjukkan hasil *precision*, *accuracy*, *recall*, dan *F1-Score* untuk masing-masing kelas. Nilai *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN) ditunjukkan untuk masing-masing kelas. Namun, jika dihitung secara manual, hasilnya tidak jauh berbeda dan hampir sama dengan perhitungan otomatis.

Tabel 4. 11 *Confusion matrix* Adamax tiap kelas

Kelas	TP	FP	FN	TN
Normal	293	17	7	283
Pneumonia	283	7	17	293

Tabel 4.12 berikut menunjukkan perhitungan *precision*, *recall*, dan *F1-Score* untuk masing-masing kelas berdasarkan nilai TP, FP, dan FN.

Tabel 4. 12 perhitungan manual *precision*, *recall*, dan *F1-Score* Adamax tiap kelas

Kelas	Perhitungan	Hasil
$Precision = \frac{TP}{TP+FP}$		
P_{Normal}	$\frac{TP_{Normal}}{TP_{Normal} + FP_{Normal}} = \frac{293}{293 + 17} = \frac{293}{310}$	0.945
$P_{Pneumonia}$	$\frac{TP_{Pneumonia}}{TP_{Pneumonia} + FP_{Pneumonia}} = \frac{283}{283 + 7}$ $= \frac{283}{290}$	0.975
$Recall = \frac{TP}{TP + FN}$		
R_{Normal}	$\frac{TP_{Normal}}{TP_{Normal} + FN_{Normal}} = \frac{293}{293 + 7} = \frac{293}{300}$	0.976
$R_{Pneumonia}$	$\frac{TP_{Pneumonia}}{TP_{Pneumonia} + FN_{Pneumonia}} = \frac{283}{283 + 17}$ $= \frac{283}{300}$	0.943
$F1 - Score = \frac{2 * precision * recall}{precision + recall}$		
$F1_{Normal}$	$\frac{2 * (P_{Normal} * R_{Normal})}{(P_{Normal} + R_{Normal})}$ $= \frac{2 * (0.945 * 0.976)}{0.945 + 0.976}$ $= \frac{1.84}{1.92}$	0.958
$F1_{Pneumonia}$	$\frac{2 * (P_{Pneumonia} * R_{Pneumonia})}{(P_{Pneumonia} + R_{Pneumonia})}$ $= \frac{2 * (0.975 * 0.943)}{0.975 + 0.943} = \frac{1.83}{1.91}$	0.958

Pada tabel 4.13 berikut ini menampilkan perhitungan *accuracy*, *precision*, *recall* dan *F1-Score* Adamax yang dilakukan secara manual berdasarkan nilai TP, jumlah data, jumlah kelas, jumlah *precision* setiap kelas, jumlah *recall* setiap kelas dan jumlah *F1-Score* setiap kelas.

Tabel 4. 13 perhitungan manual *precision*, *recall*, dan *F1-Score* Adamax

	Perhitungan	Hasil
<i>Accuracy</i>	$\frac{TP_{Normal} + TP_{Pneumonia}}{Jumlah\ Data} = \frac{293 + 283}{600}$ $= \frac{576}{600}$	0.96
<i>Precision</i>	$\frac{P_{Normal} + P_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.945 + 0.975}{2}$ $= \frac{1.92}{2}$	0.96
<i>Recall</i>	$\frac{R_{Normal} + R_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.976 + 0.943}{2}$ $= \frac{1.919}{2}$	0.959
<i>F1-Score</i>	$\frac{F1_{Normal} + F1_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.958 + 0.958}{2}$ $= \frac{1.916}{2}$	0.958

c. Adadelta

Confusion Matrix

True	Normal	298	2
	Pneumonia	78	222
		Normal	Pneumonia
		Predicted	

Gambar 4. 9 *Confusion matrix* Adadelta

Tabel 4.14 dan 4.15 menunjukkan hasil yang dapat digunakan untuk menghitung akurasi, ketepatan, *recall*, dan skor f1 berdasarkan nilai matriks tersebut.

Tabel 4. 14 skor *precision*, *recall*, dan *F1-Score* Adadelta

	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>optimizer</i> Adadelta	0.87	0.89	0.87	0.86

Tabel 4.15 ini menunjukkan hasil untuk masing-masing kelas yang dilakukan tes dengan konfigurasi Adadelta, termasuk *accuracy*, *recall*, *F1-Score*, dan jumlah data.

Tabel 4. 15 skor *precision*, *recall*, dan *F1-Score* Adadelta tiap kelas

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.79	0.99	0.88
Pneumonia	0.99	0.74	0.85

Tabel 4.16–4.18 menunjukkan hasil *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN) untuk masing-masing kelas. Namun, jika dihitung secara manual untuk *precision*, *accuracy*, *recall*, dan *F1-Score* berdasarkan *confusion matrix* pada gambar 4.8, hasilnya juga tidak jauh berbeda dan hampir sama dengan perhitungan otomatis.

Tabel 4. 16 *confusion matrix* Adadelta tiap kelas

Kelas	TP	FP	FN	TN
Normal	298	78	2	222
Pneumonia	222	2	78	298

Tabel 4.17 berikut menunjukkan perhitungan *precision*, *recall*, dan *F1-Score* untuk masing-masing kelas berdasarkan nilai TP, FP, dan FN.

Tabel 4. 17 Perhitungan manual *confusion matrix* Adadelta tiap kelas

Kelas	Perhitungan	Hasil
$Precision = \frac{TP}{TP+FP}$		
P_{Normal}	$\frac{TP_{Normal}}{TP_{Normal} + FP_{Normal}} = \frac{298}{298 + 78} = \frac{298}{376}$	0.792
$P_{Pneumonia}$	$\frac{TP_{Pneumonia}}{TP_{Pneumonia} + FP_{Pneumonia}} = \frac{222}{222 + 2}$ $= \frac{222}{224}$	0.991
$Recall = \frac{TP}{TP + FN}$		
R_{Normal}	$\frac{TP_{Normal}}{TP_{Normal} + FN_{Normal}} = \frac{298}{298 + 2} = \frac{298}{300}$	0.993
$R_{Pneumonia}$	$\frac{TP_{Pneumonia}}{TP_{Pneumonia} + FN_{Pneumonia}} = \frac{222}{222 + 78}$ $= \frac{222}{300}$	0.74
$F1 - Score = \frac{2 * precision * recall}{precision + recall}$		
$F1_{Normal}$	$\frac{2 * (P_{Normal} * R_{Normal})}{(P_{Normal} + R_{Normal})}$ $= \frac{2 * (0.792 * 0.993)}{0.792 + 0.993}$ $= \frac{1.57}{1.78}$	0.88
$F1_{Pneumonia}$	$\frac{2 * (P_{Pneumonia} * R_{Pneumonia})}{(P_{Pneumonia} + R_{Pneumonia})}$ $= \frac{2 * (0.991 * 0.74)}{0.991 + 0.74} = \frac{1.46}{1.73}$	0.84

Tabel 4.18 berikut menunjukkan perhitungan *precision*, *accuracy*, *recall*, dan *F1-Score* Adadelta yang dilakukan secara manual. Perhitungan ini didasarkan pada nilai TP, jumlah data, jumlah kelas, jumlah *precision*, jumlah *recall*, dan jumlah *F1-Score* untuk masing-masing kelas.

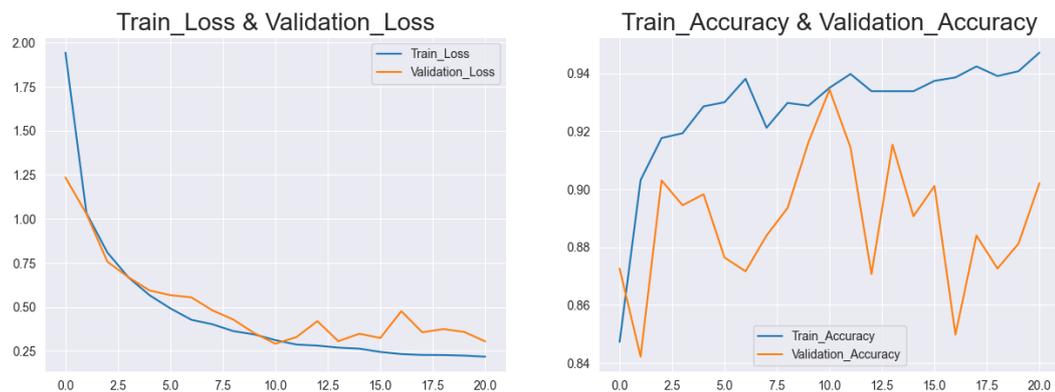
Tabel 4. 18 Perhitungan Manual *Confusion matrix* Adadelta

	Perhitungan	Hasil
<i>Accuracy</i>	$\frac{TP_{Normal} + TP_{Pneumonia}}{Jumlah\ Data} = \frac{298 + 222}{600}$ $= \frac{520}{600}$	0.866
<i>Precision</i>	$\frac{P_{Normal} + P_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.792 + 0.993}{2}$ $= \frac{1.78}{2}$	0.892
<i>Recall</i>	$\frac{R_{Normal} + R_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.993 + 0.74}{2}$ $= \frac{1.733}{2}$	0.866
<i>F1-Score</i>	$\frac{F1_{Normal} + F1_{Pneumonia}}{Jumlah\ Kelas} = \frac{0.88 + 0.84}{2}$ $= \frac{1.72}{2}$	0.86

Hasil percobaan menunjukkan bahwa *optimizer* Adamax memiliki nilai tinggi. mencapai *accuracy* 0.96, *precision* 0.96, *recall* 0.96, dan *F1-Score* 0.96.

4.6.3 Perbandingan *Batch Size*

Setelah membandingkan *optimizer* penelitian ini selanjutnya membandingkan pengaruh *batch size* terhadap akurasi dan *overfitting*. konfigurasi yang digunakan sama seperti pada sub bab 4.6.1 dengan *optimizer* yang dipakai adalah Adamax sebab memiliki hasil akurasi yang bagus dan tidak terjadi *overfitting* dibandingkan Adam dan Adadelta.



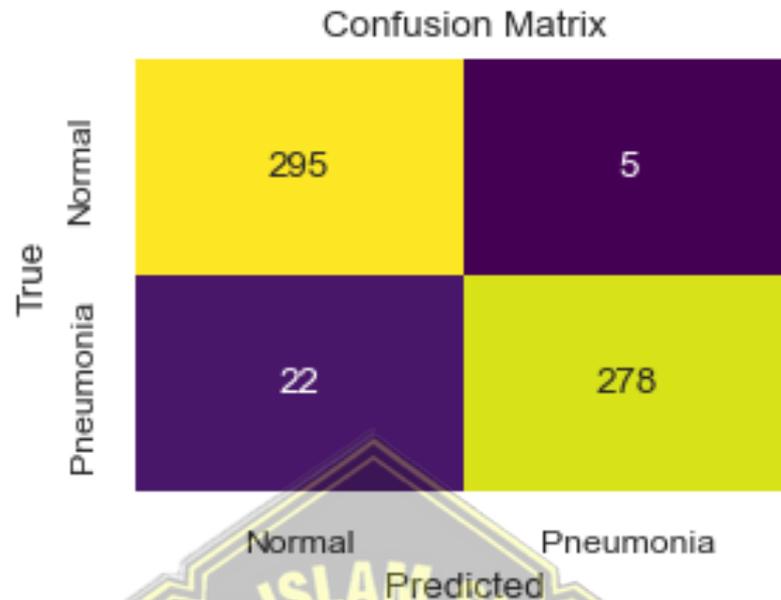
Gambar 4. 10 Learning Curve Batch Size 32

Gambar 4.10 merupakan *learning curve* dari hasil pelatihan model dengan *batch size* 32. terlihat curva *train* memiliki gap yang lebar. rincian dari hasil akhir pelatihan terdapat pada tabel 4.19

Tabel 4. 19 Rincian Luaran Terakhir Hasil *Train Batch Size* 32

<i>loss</i>	<i>binary accuracy</i>	<i>validation loss</i>	<i>validation binary accuracy</i>	<i>learning rate</i>
0.218218	0.947193	0.304652	0.902091	0.00025

Tabel 4.19 menunjukkan metrik kinerja model pada suatu waktu. Ini termasuk kehilangan, keakuratan biner, keakuratan validasi biner, dan tingkat pembelajaran. Untuk data pelatihan, model menunjukkan kehilangan sebesar 0,218218 dan akurasi biner sebesar 94,72%; namun, ada perbedaan antara kehilangan dan akurasi sebesar 0,045 dan 0,86. Setelah validasi, model menunjukkan kehilangan sebesar 0,304652 dan akurasi biner sebesar 90,21%, yang menunjukkan potensi generalisasi. Pada saat ini, tingkat pembelajaran adalah 0.00025. Hasil ini memberikan gambaran tentang kinerja model dan menunjukkan bahwa ada kemungkinan parameter dapat diubah untuk meningkatkan kinerja.



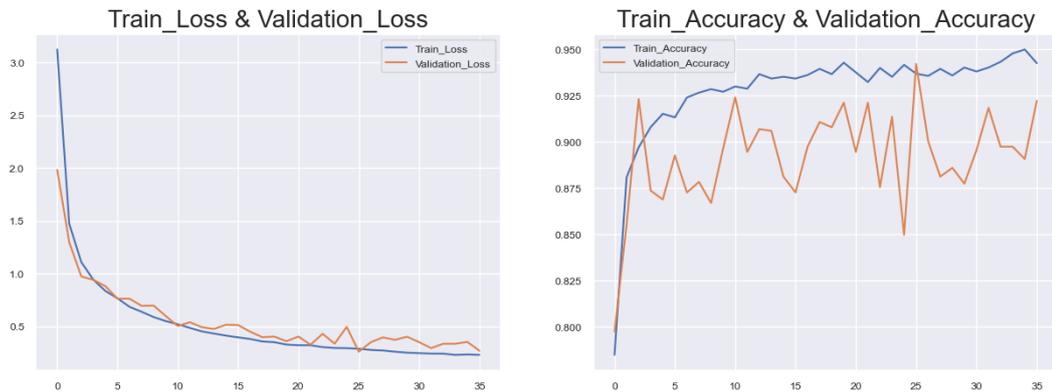
Gambar 4. 11 *Confusion matrix Batch Size 32*

Data *confusion matrix* dengan detail ditunjukkan pada Gambar 4.11. Data ini mencakup 295 pasien yang diprediksi dengan benar sebagai normal, 5 pasien yang diprediksi sebagai pneumonia yang salah, 22 pasien yang diprediksi sebagai normal yang salah, dan 278 pasien pneumonia yang diprediksi dengan benar.

Tabel 4. 20 Skor *Precision*, *Recall*, dan *F1-Score Batch Size 32*

<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy</i>
0.96	0.95	0.95	0.95

Tabel 4.20 evaluasi kinerja model klasifikasi tersebut menunjukkan bahwa, dari berbagai sudut pandang, kinerja model tersebut cukup baik. Dengan presisi sebesar 96%, sebagian besar prediksi positif yang dibuat oleh model benar, dan *recall* sebesar 95% menunjukkan kemampuan model untuk menemukan sebagian besar kasus positif yang sebenarnya. *F1-Score*, yang berfungsi sebagai harmonisasi antara presisi dan *recall*, memiliki nilai 0,95, yang menunjukkan keseimbangan yang baik antara ketepatan dan kepekaan model. Model ini secara keseluruhan memberikan prediksi yang benar untuk 95 persen dari semua kasus..



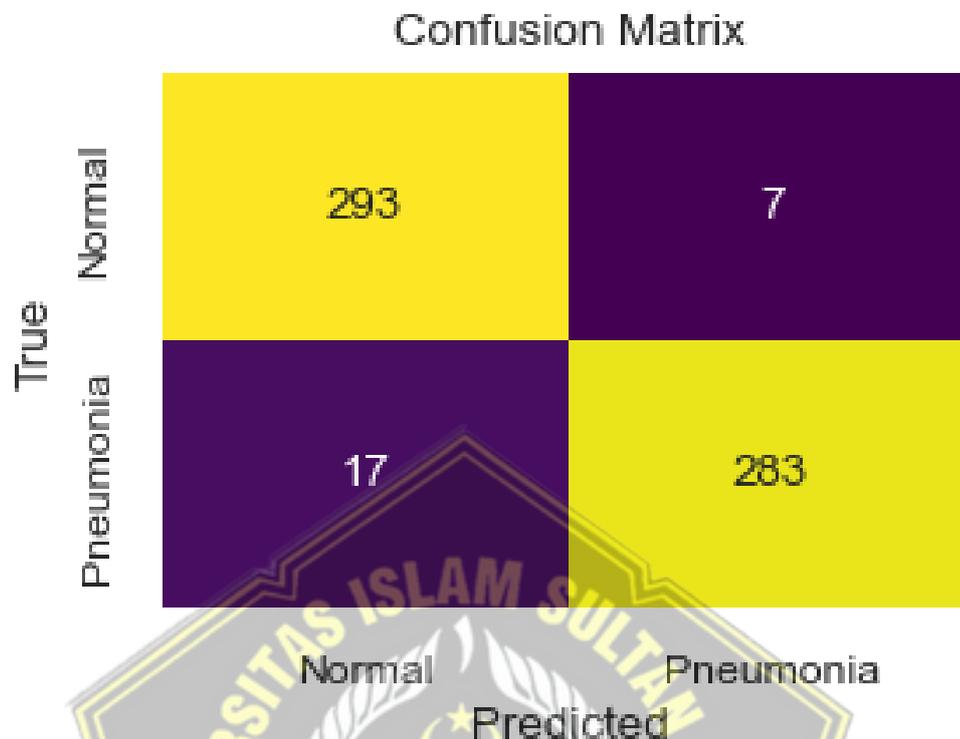
Gambar 4. 12 *Learning Curve Batch Size 128*

Gambar 4.12 adalah *learning curve* dari konfigurasi *batch size* 128. Menunjukkan hasil lebih baik dari hasil pada gambar 4.10. hasil terakhir dari pelatihan model konfigurasi *batch size* 128 terdapat pada tabel 4.21.

Tabel 4. 21 Rincian Luaran Terakhir Hasil *Train Batch Size 128*

<i>loss</i>	<i>binary accuracy</i>	<i>validation loss</i>	<i>validation binary accuracy</i>	<i>learning rate</i>
0.230625	0.942436	0.269844	0.922053	0.00025

Tabel 4.21 memiliki hasil yang lebih bagus dibanding Tabel 4.19. hal tersebut ditunjukkan pada gap antara *loss* dengan *validation loss* terlihat bahwa lebih kecil konfigurasi *batch size* 128 dibanding 32.



Gambar 4. 13 *Confusion matrix Batch Size 128*

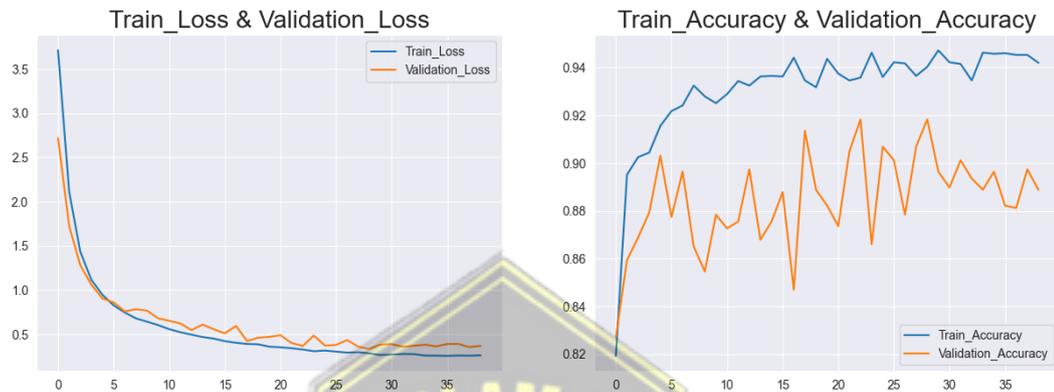
Gambar 4.13 menunjukkan data *confusion matrix* dengan detail 293 *True Positive*, 17 *False Positive*, 7 *False Negatif*, dan 283 *True Negatif*.

Tabel 4. 22 *Skor Precision, Recall, dan F1-Score Batch Size 128*

<i>accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
0.96	0.96	0.96	0.96

Tabel evaluasi kinerja model klasifikasi terbaru menunjukkan peningkatan signifikan dengan semua metrik evaluasi (*akurasi*, *presisi*, *recall*, dan *F1-Score*) mencapai nilai 96. Peningkatan ini menunjukkan kemampuan model yang lebih baik dalam mengklasifikasikan *instance* positif dan negatif dibandingkan dengan tabel sebelumnya yang memiliki nilai 95 untuk semua metrik. Meskipun perbedaannya kecil, konsistensi tinggi dalam peningkatan nilai tersebut mencerminkan performa model yang lebih handal dan akurat. Seiring nilai yang sangat dekat antara *presisi*, *recall*, dan *F1-Score*, model ini menunjukkan

keseimbangan yang baik antara ketepatan dan kepekaan dalam pengklasifikasian. Evaluasi lebih lanjut tetap disarankan, dengan mempertimbangkan konteks aplikasi dan karakteristik spesifik dari masalah klasifikasi yang dihadapi.



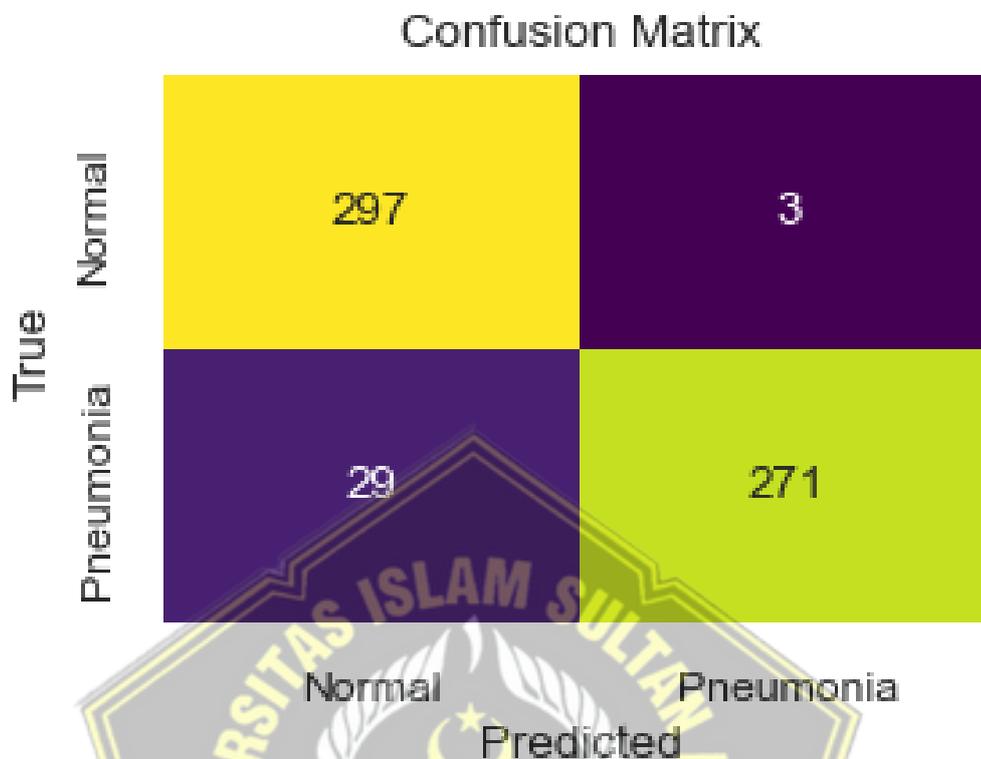
Gambar 4. 14 *Learning Curve Batch Size 256*

Gambar 4.14 menunjukkan grafik *learning curve batch size 256*. Luaran dari *learning curve* gambar 4.14 lebih buruk dari gambar 4.12 sebab garis grafiknya melebar dan gapnya lebih besar.

Tabel 4. 23 Rincian Luaran Terakhir Hasil *Train Batch Size 256*

<i>loss</i>	<i>binary accuracy</i>	<i>validation loss</i>	<i>validation binary accuracy</i>	<i>learning rate</i>
0.262084	0.941960	0.369037	0.888783	0.000125

Tabel 4.23 memiliki hasil tidak lebih baik dari Tabel 4.21 sebab gap *loss* lebih lebar. lebarnya gap tersebut mengindikasikan bahwa overfitting konfigurasi *batch size 256* lebih overfit dibanding konfigurasi *batch size 128*



Gambar 4. 15 *Confusion matrix Batch Size 256*

Gambar 4.13 menunjukkan data *confusion matrix* dengan detail 297 *True Positive*, 29 *False Positive*, 3 *False Negatif*, dan 271 *True Negatif*.

Tabel 4. 24 *Skor Precision, Recall, dan F1-Score Batch Size 256*

<i>accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
0.95	0.95	0.95	0.95

Tabel 4.24 menunjukkan nilai yang stabil dan tinggi untuk akurasi, presisi, *recall*, dan *F1-Score*, dengan semua metrik memiliki nilai 0.95. Meskipun tidak ada peningkatan numerik dari tabel sebelumnya, model ini menunjukkan konsistensi tinggi dalam kemampuan klasifikasinya. Sebagai perbandingan, tabel sebelumnya menunjukkan peningkatan kecil namun signifikan, mencapai nilai 0.96 untuk semua metrik. Dalam kedua kasus, model menunjukkan kemampuan yang baik

dalam mengklasifikasikan instance positif dan negatif dengan keseimbangan yang baik antara ketepatan dan kepekaan.

4.6.4 Percobaan tanpa parameter bobot kelas



Gambar 4. 16 *Learning Curve* tanpa bobot kelas

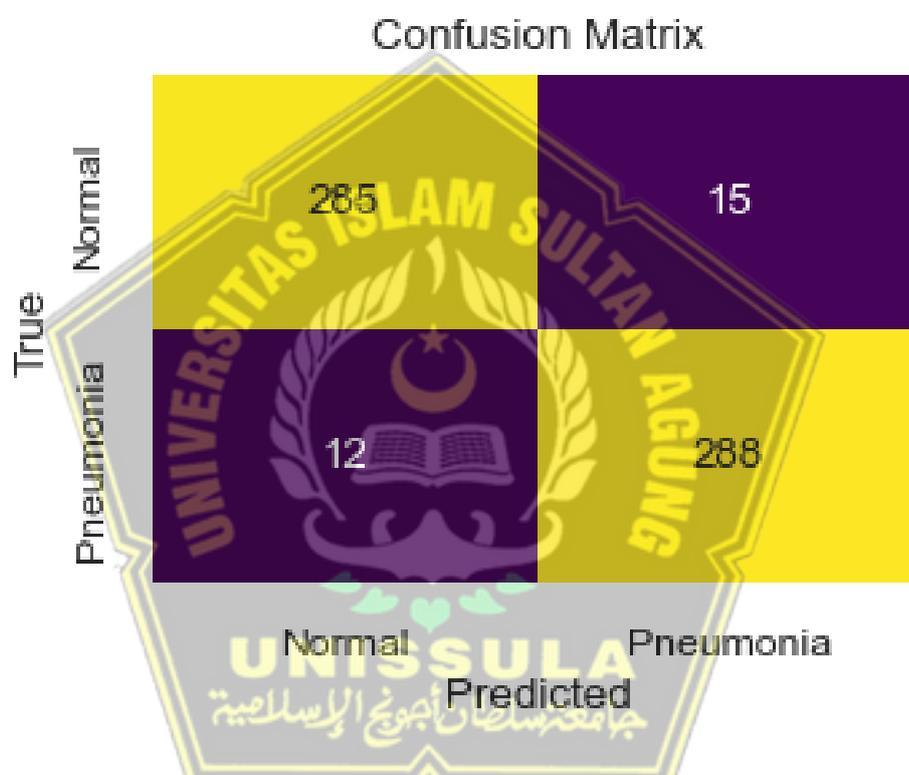
Learning Curve Gambar 4.16 menunjukkan hasil yang kurang memuaskan karena terjadi pelebaran garis pada grafik curva. hal ini lebih buruk ketimbang grafik pelatihan model dengan memperhatikan bobot kelas

Tabel 4. 25 Rincian Luaran Terakhir Hasil *Train* tanpa bobot kelas

<i>loss</i>	<i>binary accuracy</i>	<i>validation loss</i>	<i>validation binary accuracy</i>	<i>learning rate</i>
0.125078	0.955519	0.140963	0.943916	0.00050

Tabel evaluasi kinerja model menampilkan beberapa metrik yang umum digunakan dalam pelatihan dan validasi model *neural network*. Nilai *loss* pada 0.125078 mengukur seberapa baik model dapat mengurangi kesalahan selama proses pelatihan, dan semakin rendah nilainya, semakin baik model berkinerja. *Binary accuracy*, dengan nilai 0.955519, mencerminkan akurasi model dalam mengklasifikasikan *instance biner* pada data pelatihan. *Validation loss* dan *validation binary accuracy*, dengan nilai masing-masing 0.140963 dan 0.943916, memberikan gambaran performa model pada data validasi yang tidak digunakan selama pelatihan. *Validation loss* yang rendah dan *validation binary accuracy* yang tinggi menunjukkan kemampuan model untuk melakukan generalisasi pada data yang belum pernah dilihat sebelumnya. *Learning rate* sebesar 0.00050

mengindikasikan seberapa besar langkah yang diambil oleh *optimizer* untuk menyesuaikan bobot model selama pelatihan. Keseluruhan, nilai-nilai ini menunjukkan model yang memiliki performa yang baik dan mampu melakukan generalisasi pada data validasi dengan baik. Evaluasi lebih lanjut, termasuk analisis terhadap kurva pembelajaran dan validasi, tetap penting untuk memahami secara menyeluruh kinerja model dan menentukan apakah *hyperparameter*, seperti *learning rate*, sudah dioptimalkan dengan baik.



Gambar 4. 17 *Confusion matrix* tanpa bobot kelas

Gambar 4.17 menampilkan *confusion matrix* yang merinci hasil klasifikasi model. Dalam matrix tersebut, terdapat 285 *True Positive*, 12 *False Positive*, 15 *False Negative*, dan 288 *True Negative*. Data ini memungkinkan perhitungan metrik evaluasi seperti presisi, *recall*, dan akurasi. Presisi diukur dengan rasio *True Positive* terhadap total prediksi positif, *recall* diukur dengan rasio *True Positive* terhadap total instance positif sebenarnya, dan akurasi diukur dengan rasio total prediksi yang benar terhadap total instance. *Confusion matrix* juga memberikan

informasi untuk menghitung *F1-Score*, yaitu nilai gabungan antara presisi dan *recall*.

Tabel 4. 26 Skor *Precision*, *Recall*, dan *F1-Score Batch Size* tanpa bobot kelas

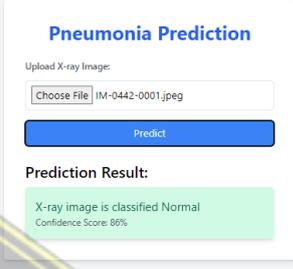
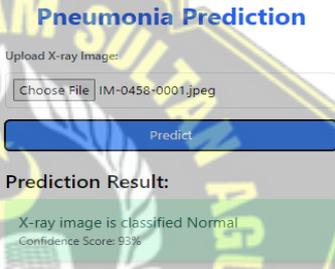
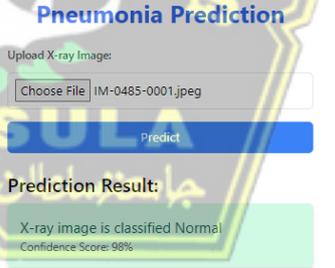
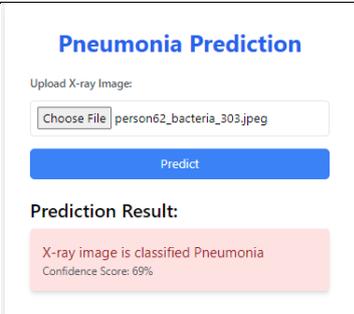
<i>accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
0.95	0.95	0.95	0.95

Tabel 4.26 evaluasi model menunjukkan nilai stabil dan tinggi untuk akurasi (0.95), presisi (0.95), *recall* (0.95), dan *F1-Score* (0.95). Hasil ini menunjukkan bahwa model memiliki performa baik dalam mengklasifikasikan instance dengan seimbang antara akurasi, presisi, *recall*, dan *F1-Score*. Evaluasi lebih lanjut diperlukan untuk pemahaman kontekstual yang lebih mendalam terkait kinerja model.

4.6.5 Hasil *Testing*

Hasil percobaan menggunakan model yang dipilih akan ditunjukkan di sini. Percobaan ini menggunakan konfigurasi Adamax karena proses pelatihan konfigurasi ini menunjukkan *binary accuracy* 0,9367, *loss* 0,2881, *validation binary accuracy* 0,9420, dan *validation loss* 0,2603. Selain itu, pemeriksaan dengan data uji menunjukkan *accuracy* 0.96, *precision* 0.96, *recall* 0.96, dan *F1-Score* 0.96. Tabel 4.19 menunjukkan hasil tes dengan model konfigurasi Adamax.

Tabel 4. 27 Hasil *Testing*

Gambar	Hasil Prediksi	Keterangan
Normal		
		Sesuai
		Sesuai
		Sesuai
Pneumonia		
		Sesuai

	<p>Pneumonia Prediction</p> <p>Upload X-ray Image:</p> <p>Choose File person65_virus_123.jpeg</p> <p>Predict</p> <p>Prediction Result:</p> <p>X-ray image is classified Pneumonia Confidence Score: 99%</p>	Sesuai
	<p>Pneumonia Prediction</p> <p>Upload X-ray Image:</p> <p>Choose File person66_bacteria_326.jpeg</p> <p>Predict</p> <p>Prediction Result:</p> <p>X-ray image is classified Pneumonia Confidence Score: 98%</p>	Sesuai



BAB V

Kesimpulan dan Saran

5.1 Kesimpulan

Hasil penelitian menunjukkan bahwa arsitektur DenseNet201 menunjukkan kinerja yang baik untuk mengklasifikasikan pneumonia pada rontgen paru-paru. Oleh karena itu, hasil penelitian ini adalah sebagai berikut:

1. Konfigurasi 2 yang menggunakan optimizer Adamax adalah yang terbaik, dengan *binary accuracy* 0,9367, *loss* 0,2881, *validation binary accuracy* 0,9420, dan *validation loss* 0,2603.
2. Model arsitektur DenseNet201 bekerja dengan baik dalam mengklasifikasikan pneumonia pada rontgen paru-paru. Hasil pengujian menggunakan data tes menunjukkan *accuracy* 0,96, *precision* 0,96, dan *F1-Score* 0,96.

5.2 Saran

Penelitian ini menunjukkan kinerja yang baik. Namun, peneliti menyarankan untuk mengembangkan sistem ini agar dapat bekerja dengan *platform mobile* seperti Android atau iOS.

Daftar Pustaka

- Chasshidi, T. A., & Putra, M. R. (2021). Sistem Pakar untuk Mendiagnosa Penyakit Pneumonia Menggunakan Metode Certainty Factor dan Fuzzy Logic Tsukamoto Berbasis WEB. *Jurnal KomtekInfo*, 8(2).
<https://doi.org/10.35134/komtekinfo.v8i2.106>
- Dharma Adhinata, F., Putra Rakhmadani, D., Wibowo, M., & Jayadi, A. (2021). *A Deep Learning Using DenseNet201* (Vol. 9, Issue 1).
- Farid Naufal, M., & Ferdiana Kusuma, S. (2021). *PENDETEKSI CITRA MASKER WAJAH MENGGUNAKAN CNN DAN TRANSFER LEARNING*. 8(6), 1293–1300.
<https://doi.org/10.25126/jtiik.202185201>
- Hekmatyar, H. D., Saputra, W. A., & Ramdani, C. (2022). Klasifikasi Pneumonia Dengan Deep Learning Faster Region Convolutional Neural Network Arsitektur VGG16 dan ResNet50. *InComTech : Jurnal Telekomunikasi Dan Komputer*, 12(3).
<https://doi.org/10.22441/incomtech.v12i3.15112>
- Ilmi, T., Yulia, R., & Herawati, F. (2020). *EVALUATION OF ANTIBIOTIC USE ON PNEUMONIA PATIENTS IN TULUNGAGUNG REGIONAL HOSPITAL*. 1(2).
- Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Illcus, S., Chute, C., Marklund, H., Haghighi, B., Ball, R., Shpanskaya, K., Seekins, J., Mong, D. A., Halabi, S. S., Sandberg, J. K., Jones, R., Larson, D. B., Langlotz, C. P., Patel, B. N., Lungren, M. P., & Ng, A. Y. (2019). *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison*. <http://arxiv.org/abs/1901.07031>.
- Kerem, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., ... Zhang, K. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5), 1122-1131.e9. <https://doi.org/10.1016/j.cell.2018.02.010>.
- Kuswanto, J., Dapiokta, J., Ki Ratu Penghulu No, J., Sari Baturaja, K., & Selatan, S. (n.d.). *Penerapan Metode Forward Chaining Untuk Diagnosa Penyakit Pneumonia*. 15(1), 2022.

- Lorentius, C. A., Adipranata, R., & Tjondrowiguno, A. (n.d.). *Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network*.
- Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., & Jamalipour Soufi, G. (2020). Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Medical Image Analysis*, 65. <https://doi.org/10.1016/j.media.2020.101794>
- Phung, V. H., & Rhee, E. J. (2019). A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. *Applied Sciences (Switzerland)*, 9(21). <https://doi.org/10.3390/app9214500>
- Rahman, T., Chowdhury, M. E. H., Khandakar, A., Islam, K. R., Islam, K. F., Mahbub, Z. B., Kadir, M. A., & Kashem, S. (2020). Transfer learning with deep Convolutional Neural Network (CNN) for pneumonia detection using chest X-ray. *Applied Sciences (Switzerland)*, 10(9). <https://doi.org/10.3390/app10093233>
- Suartawan, I. P. (2019). BRONKOPNEUMONIA PADA ANAK USIA 20 BULAN. *JURNAL KEDOKTERAN*, 5(1). <https://doi.org/10.36679/kedokteran.v5i1.177>
- Wijaya, A. C., Gede, I., Wibawa, A., Dewa, D. I., Bayu, M., & Darmawan, A. (n.d.). *PENGEMBANGAN RESTFUL API UNTUK MODEL MACHINE LEARNING INDOOR-OUTDOOR DALAM APLIKASI PEMINJAMAN RUANGAN*.
- World Health Organization. (2022, 11 November). Pneumonia in children. Diakses pada 20 September 2023. <https://www.who.int/news-room/fact-sheets/detail/pneumonia>.
- Yumame, R. O., Pranoto, Y. A., & Dedy Irawan, J. (2022). RANCANG BANGUN PENDETEKSI MASKER DAN ALAT PEMBERSIH TANGAN OTOMATIS BERBASIS ARDUINO MENGGUNAKAN METODE CNN (CONVOLUTION NEURAL NETWORK) DI GEREJA GKI JAYAPURA PAPUA. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 6, Issue 2).

Lampiran

