

PROPOSAL TUGAS AKHIR
IMPLEMENTASI DEEP LEARNING UNTUK KLASIFIKASI
CITRA X-RAY PARU-PARU DALAM DETEKSI PNEUMONIA



ABDUR ROCHMAN

3130021021

PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS EKONOMI BISNIS DAN TEKNOLOGI DIGITAL
UNIVERSITAS NAHDLATUL ULAMA SURABAYA

2024

ABSTRAK

Pneumonia adalah kondisi saat bagian paru-paru yaitu alveoli seseorang terisi nanah dan cairan sehingga penderita kesulitan bernafas. Penyakit pneumonia dapat dideteksi menggunakan citra x-ray paru-paru. Citra x-ray paru-paru adalah citra x-ray yang memberikan informasi kondisi paru-paru pasien. Dengan menggunakan citra x-ray ini, Diagnosa penyakit pneumonia dilakukan oleh dokter spesialis paru. Namun jumlah dokter spesialis paru di Indonesia terbatas. Keterbatasan ini berdampak pada lambatnya proses diagnosis dan penanganan pasien. Salah satu upaya untuk mengatasi keterbatasan dalam diagnosa penyakit pneumonia, pada penelitian ini dilakukan klasifikasi citra x-ray paru-paru menggunakan metode deep learning. Deep learning adalah cabang dari machine learning yang berfokus pada pengembangan algoritma berbasis jaringan saraf tiruan yang dirancang untuk belajar dan mengekstraksi pola atau representasi data secara otomatis. Metode deep learning yang efektif dalam pengolahan citra adalah Convolutional Neural Network (CNN). CNN mampu melakukan ekstraksi fitur dari citra secara lebih mendetail, menyimpan dan menjadikan acuan hasil pembelajaran yang telah dilakukan. Dalam penelitian ini digunakan model CNN untuk mengklasifikasikan citra Chest X-Ray paru-paru yaitu DenseNet201 dan Xception. Kedua model ini telah terbukti efektif dalam bidang klasifikasi citra medis karena kemampuannya dalam menangani vanishing gradient dan meningkatkan kemampuan generalisasi. Ada beberapa tahapan yang dilakukan dalam pedekatan pendekatan ini ialah dibagi menjadi 3 tahapan besar yaitu tahap persiapan, implemntasi, dan evaluasi yang mana setiap tahapan besar memiliki sub tahapan diantaranya tahapan persiapan (pengumpulan data, preprocessing, dan augmentasi data), tahapan implementasi (pembuatan model dan evaluasi model), dan terakhir adalah evaluasi (Tabel Perbandingan Klasifikasi Model DenseNet201 dan Xception). Penggunaan kedua model ini diharapkan dapat meningkatkan akurasi dalam mendiagnosis pneumonia.

Kata Kunci – Pneumonia, X-Ray, Deep Learning, DenseNet201, Xception

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pneumonia adalah sebuah kondisi dimana alveoli penderita dipenuhi nanah dan cairan sehingga penderita mengalami kesakitan saat bernafas dan oksigen yang masuk kedalam paru paru kurang (Rindrasari, 2018). Pneumonia termasuk penyakit yang menyebabkan kematian tertinggi didunia khususnya pada anak anak (Elvionita, Sari and Nuryastuti, 2023). Berdasarkan data *World Health Organization* (WHO) pada 2017, terdapat 25.481 kasus kematian anak anak karena infeksi pernafasan akut atau 17 persen dari seluruh kematian anak anak. Indonesia berada diperingkat ke-7 didunia sebagai negara yang memiliki kasus pneumonia tertinggi (Pranita and Sumartiningtyas, 2020). Pada pasien orang dewasa, kasus pneumonia paling banyak terjadi pada pasien dengan rentan umur 56 – 65 tahun. Hal ini terjadi karena pasien usia lanjut mengalami perubahan anatomi fisiologi akibat proses penuaan sehingga terjadi penurunan fungsional paru-paru, kemampuan untuk mengatasi penurunan complaints paru dan peningkatan resistensi saluran napas terhadap infeksi dan penurunan daya tahan tubuh (Farida, Trisna and Nur, 2017).

Peran dokter spesialis paru tentu sangat dibutuhkan. Namun jumlah dokter spesialis paru di Indonesia masih terbatas. Berdasarkan data Perhimpunan Dokter Paru Indonesia (PDPI) pada 2020, tercatat berjumlah 1.206 orang. Jumlah ini belum seimbang dengan jumlah penduduk di Indonesia yang saat ini berjumlah sekitar 250 juta jiwa. Idealnya Indonesia memiliki minimal 2.500 dokter spesialis paru, dengan kata lain saat ini kekurangan 1.294 dokter paru (Manafe, 2020). Keterbatasan ini berdampak pada lambatnya proses diagnosis dan penanganan pasien. Selain itu, diagnosis yang dilakukan oleh dokter spesialis paru sering kali bersifat subjektif. Oleh karena itu dibutuhkanlah sistem yang mampu memberikan

diagnosis cepat, objektif dan akurat dalam meningkatkan kualitas pelayanan kesehatan, khususnya dalam menangani penyakit pneumonia.

Identifikasi pneumonia biasanya melibatkan test klinis, analisis riwayat medis, dan test pencitraan seperti citra *chest x-ray* atau *computed tomography* (CT) (Huy and Lin, 2023). Citra *X-ray* adalah salah satu yang dapat memberikan gambaran kondisi paru-paru, mendeteksi adanya infeksi, peradangan, atau akumulasi cairan yang menjadi ciri khas pneumonia. Hasil dari citra medis *X-ray* akan didiagnosis oleh dokter spesialis untuk dilakukan tindakan teknis selanjutnya. Diagnosis pneumonia yang akurat menjadi faktor krusial untuk memastikan pengobatan yang efektif, sehingga dapat meningkatkan prognosis pasien (Soewu *et al.*, 2022). Dalam konteks penelitian ini, identifikasi pneumonia juga menjadi dasar pengembangan sistem prediksi berbasis deep learning yang diharapkan dapat mempercepat proses diagnosis dan meningkatkan akurasi deteksi penyakit paru-paru secara otomatis. Hal ini menunjukkan potensi besar teknologi kecerdasan buatan dalam mendukung praktik medis modern.

Dengan memanfaatkan kecerdasan buatan, berbagai penelitian telah dilakukan untuk membantu dalam meningkatkan analisis penyakit pneumonia. Salah satu upaya ialah dengan mendeteksi pasien yang terdampak pneumonia dengan cepat dan akurat menggunakan metode *deep learning*. *Deep learning* adalah cabang dari *machine learning* yang berfokus pada pengembangan algoritma berbasis jaringan saraf tiruan yang dirancang untuk belajar dan mengekstraksi pola atau representasi data secara otomatis, mirip dengan cara otak manusia bekerja (Heaton, 2018). Metode *deep learning* diharapkan dapat membantu dokter spesialis paru dan peneliti dalam mendeteksi pasien yang menderita pneumonia menggunakan gambar Citra *Chest X-Ray* yang bahkan cenderung samar dan memiliki kemiripan yang sama antar jenis penyakit pulmonary, seperti pneumonia, tuberkolosis, pneumotrax, pada paru paru pasien (Andreu *et al.*, 2004). Selain meminimalisir waktu dan tenaga medis dalam mendeteksi pasien, penggunaan machine learning dengan teknik deep learning juga mengurangi resiko human error dan mengurangi biaya perawatan.

Salah satu metode yang paling banyak digunakan untuk klasifikasi penyakit adalah *Convolutional Neural Network* (CNN). Penelitian dengan menggunakan CNN antara lain penelitian oleh (Kong and Cheng, 2022) “*Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion*” didapatkan model yang diusulkan mencapai akurasi rata-rata 98.0% untuk klasifikasi biner dan 97.3% untuk klasifikasi tiga kategori dimana menggunakan arsitektur *DenseNet* dan *VGG16*. Kemudian pada penelitian (Upasana, Tewari and Singh, 2022) “*An Attention-based Pneumothorax Classification using Modified Xception Model*” mengenai mendeteksi otomatis citra *X-ray* pneumotoraks, menghasilkan kesimpulan bahwa model *Xception* memiliki nilai akurasi yang baik dengan memiliki akurasi pelatihan 99.18% dan *validation accuracy* 87.53%.

Banyak pendekatan yang telah dikembangkan dan diterapkan untuk mendeteksi pneumonia, namun masih ada ruang untuk peningkatan kerja. Salah satu usulan untuk meningkatkan akurasi dan efisiensi dalam klasifikasi citra *X-Ray* adalah dengan mengimplementasikan metode *DenseNet201* dan *Xception*. Kedua model ini telah terbukti efektif dalam bidang klasifikasi citra medis karena kemampuannya dalam menangani *vanishing gradient* dan meningkatkan kemampuan generalisasi. *DenseNet201* dengan konektivitas yang lebih padat antar layer memungkinkan model untuk memanfaatkan informasi secara lebih efektif, sedangkan *Xception* dengan *depthwise separable* mampu mengurangi jumlah parameter dan operasi yang diperlukan, sehingga membuat model lebih ringan dan cepat. Pengguna kedua model ini diharapkan dapat meningkatkan akurasi dalam mendiagnosis pneumonia.

1.2 Perumusan Masalah

Adapun rumusan masalah yang dapat diangkat dari penelitian ini yaitu: Bagaimana hasil akurasi, presision, sensitivity, dan Fscore dari klasifikasi citra *chest x-ray* paru-paru ?

1.3 Batasan Masalah

Batasan masalah yang digunakan pada penelitian yaitu:

1. Data klasifikasi jenis citra x-ray yang digunakan antara lain : x-ray normal dan x-ray pneumonia
2. Penelitian berfokus pada tingkat akurasi, presision, sensitivity, dan Fscore klasifikasi citra x-ray pneumonia
3. Model dari arsitektur cnn yang digunakan adalah DenseNet201 dan Xception

1.4 Tujuan Penelitian

Dari rumusan masalah yang dilakukan, maka tujuan dari penelitian ini yaitu : Untuk mengetahui hasil akurasi, presision, sensitivity, dan Fscore dari klasifikasi citra x-ray paru-paru

1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu sebagai berikut:

a. Bagi Masyarakat

Hasil dari penelitian tersebut dapat digunakan untuk meningkatkan akurasi dan efisiensi dalam mendeteksi pneumonia melalui analisis citra *chest X-ray*, sehingga dapat mendukung tenaga medis dalam mendiagnosis penyakit lebih cepat dan tepat. Dengan pemanfaatan model dari *deep learning*, sistem dapat membantu mengurangi risiko kesalahan diagnosis serta mempercepat proses pengambilan keputusan. Terutama ditengah keterbatasan jumlah dokter spesialis paru yang sedikit. Hal ini diharapkan dapat berkontribusi pada penanganan yang lebih dini dan efektif, sehingga meningkatkan peluang kesembuhan pasien dan secara keseluruhan mendukung peningkatan kualitas layanan kesehatan Masyarakat.

b. Bagi Pembaca

Hasil dari penelitian ini diharapkan dapat menjadi rujukan atau refrensi bagi mahasiswa Sistem Informasi yang akan melakukan penelitian lanjutan dengan topik *deep learning*.

c. Bagi Penulis

Manfaat yang didapatkan oleh Penulis yakni berupa menambah wawasan pada bidang *deep learning* dan oleh data citra gambar. Selain itu juga pengalaman saat proses pengerjaan tugas akhir dimulai dari proses pengambilan data hingga menulis laporannya.

1.6 Relevansi

Relevansi pengerjaan penelitian ini berkaitan dengan beberapa mata kuliah seperti:

1. Sistem Cerdas : dalam mata kuliah Sistem Cerdas, konsep dan penerapan algoritma *deep learning* seperti *DenseNet201* dan *Xception* menjadi dasar dalam mengembangkan model untuk mendeteksi pneumonia.
2. Statistika : dalam mata kuliah Statistika berkontribusi pada analisis data, seperti pengolahan statistik deskriptif dan inferensial untuk mengevaluasi performa model yang dikembangkan.
3. Data Science : dalam mata kuliah Data Science menyediakan landasan dalam pengelolaan, pembersihan, dan pemrosesan dataset citra *chest X-ray* termasuk teknik eksplorasi data yang relevan.
4. Visualisasi Informasi : Visualisasi Informasi berperan dalam menyajikan hasil penelitian secara informatif melalui grafik atau diagram yang memudahkan menangkap sebuah informasi oleh pengguna, termasuk tenaga medis.

BAB II

TINJAU PUSTAKA

2.1 Studi Literatur

No	Judul	Pembahasan	Keterkaitan	Perbedaan
1	<i>An Alzheimer's disease classification model using transfer learning Densenet with embedded healthcare decision support system</i> (Saleh et al., 2023).	Menggabungkan beberapa jenis model <i>DenseNet</i> yaitu <i>DenseNet</i> 121,169, dan 201. Menggunakan transfer learning untuk meningkatkan performa klasifikasi pada dataset <i>Magnetic Resonance Imaging</i> (MRI) yang lebih kecil, serta penggunaan augmentasi data untuk meningkatkan performa model.	Menggunakan <i>transfer learning</i> , menggunakan Teknik augmentasi data, dan menggunakan data citra medis.	Data yang digunakan MRI, Penggabungan 3 Jenis <i>DenseNet</i> 121,169, dan 201, dan menggunakan <i>Area Under the Curve</i> untuk evaluasi model.
2	<i>Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion</i> (Kong and Cheng, 2022).	Menambahkan <i>global attention machine block</i> dan <i>category attention block</i> untuk mengekstraksi fitur-fitur dalam. Jaringan residual (<i>ResNet</i>) digunakan untuk segmentasi informasi gambar. Hasil dari pelatihan model menunjukkan bahwa model yang diusulkan	Penelitian sama-sama klasifikasi penyakit citra <i>X-Ray</i> paru-paru. Menggunakan teknologi <i>deep learning</i> yaitu <i>DenseNet</i> sebagai salah satu arsitektur model dalam citra <i>X-Ray</i> . Untuk mengevaluasi	Penelitian COVID-19 mencakup deteksi beberapa penyakit (COVID-19, pneumonia, normal). Penelitian COVID-19 menggabungkan fitur dari dua arsitektur untuk meningkatkan

No	Judul	Pembahasan	Keterkaitan	Perbedaan
		memiliki akurasi yang tinggi.	performa model menggunakan akurasi sebagai indikator utama keberhasilan.	akurasi. Penelitian COVID-19 memiliki pendekatan yang lebih kompleks dengan penambahan mekanisme attention untuk mengoptimalkan performa model. Penelitian
3	<i>FemurTumorNet: Bone tumor classification in the proximal femur using DenseNet model based on radiographs</i> (Pan et al., 2023).	Dengan menggunakan Model <i>DenseNet</i> dapat mengungguli akurasi diagnosis manusia dalam hal sensitivitas, spesitifitas, akurasi dan skor f1. Hasil dari penelitian menunjukkan bahwa model <i>DenseNet</i> memiliki tingkat akurasi yang tinggi.	Objek yang digunakan adalah citra medis, model yang digunakan sama yaitu <i>DenseNet</i> untuk klasifikasi citra medis.	Penyakit yang diklasifikasikan dan bagian tubuh yang menjadi objek penelitian berbeda seperti muskuloskeletal. Penelitian <i>FemurTumorNet</i> hanya fokus pada <i>DenseNet</i> .
4	<i>Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging</i> (Chauhan, Palivela and Tiwari, 2021).	Dengan menggunakan metode transfer learning, untuk meningkatkan performa model dengan cara bobot yang telah data dilatih sebelumnya. Teknik <i>fine-tuning</i> termasuk <i>early-stopping</i> digunakan guna untuk meningkatkan akurasi model. Hasil	penelitian menggunakan citra medis <i>X-Ray</i> dada. Menggunakan <i>DenseNet</i> sebagai model utama dalam <i>deep learning</i> . Memanfaatkan transfer learning	Penelitian <i>DenseNet</i> lebih luas cakupannya seperti (COVID-19, PNEUMONIA dan NORMAL). Hanya fokus pada <i>DenseNet</i> tanpa memperbandingkan. <i>Fine-tuning</i> menjadi bagian

No	Judul	Pembahasan	Keterkaitan	Perbedaan
		menunjukkan bahwa <i>optimizer adam</i> dengan fungsi kerugian <i>Cross Entropy</i> dan <i>scheduler StepLR</i> memberikan hasil akurasi yang tertinggi.	untuk mengatasi keterbatasan dataset medis. Menggunakan evaluasi akurasi model untuk menentukan keberhasilan klasifikasi.	untuk meningkatkan performa model <i>deep learning</i> .
5	<i>AI diagnostics in bone oncology for predicting bone metastasis in lung cancer patients using DenseNet-264 deep learning model and radiomics (Zeng et al., 2024).</i>	Menggunakan Fitur radiomik diekstraksi dari gambar CT dan seleksi fitur dilakukan menggunakan Minimum Redundancy Maximum Relevance (mRMR) dan Least Absolute Shrinkage and Selection Operator (LASSO). Model DenseNet-264 menunjukkan kinerja yang unggul.	Menggunakan model <i>DenseNet</i> . Menggunakan Transfer learning. Evaluasi model pada menggunakan <i>accuracy</i> untuk mengukur performa klasifikasi.	Object yang digunakan kanker metastasis tulang. Penelitian <i>Bone Oncology</i> menggunakan data CT dengan fitur radiomic. Model yang digunakan <i>DenseNet-264</i> . Lebih menonjolkan <i>evaluasi Under the Curve (AUC)</i> untuk mengukur performa.
6	<i>An in-depth analysis of Convolutional Neural Network architectures with transfer learning for skin disease</i>	Menggunakan metode <i>transfer learning</i> untuk menemukan lebih banyak fitur. Mengevaluasi kinerja pendekatan yang diusulkan dengan beberapa arsitektur CNN	Menggunakan data citra medis. Menggunakan <i>transfer learning</i> sebagai metode untuk meningkatkan kinerja model.	Menggunakan citra dermatoskopi. Penelitian kulit mencakup evaluasi lebih banyak arsitektur CNN diantaranya

No	Judul	Pembahasan	Keterkaitan	Perbedaan
	<i>diagnosis</i> (Sadik <i>et al.</i> , 2023).	populer di antara nya, <i>ResNet50</i> , <i>InceptionV3</i> , <i>Inception-ResNet</i> dan <i>DenseNet</i> .	Menggunakan model <i>DenseNet</i> dan <i>Xception</i> . Menggunakan augmentasi data. Mengevaluasi kinerja arsitektur model CNN untuk menentukan model yang paling efektif.	<i>ResNet50</i> , <i>InceptionV3</i> , <i>Inception-ResNet</i> dan <i>DenseNet</i> .
7	<i>An Attention-based Pneumothorax Classification using Modified Xception Model</i> (Upasana, Tewari and Singh, 2022).	Penggunaan radiografi dada digunakan sebagai alat diagnosis yang signifikan untuk mendeteksi gangguan paru-paru. Menggabungkan jaringan <i>Xception</i> dengan attention module untuk meningkatkan akurasi deteksi pneumotoraks pada gambar <i>X-ray</i> dada. Model yang diusulkan memperoleh akurasi yang tinggi dan Area AUC yang sangat baik.	Menggunakan citra <i>X-Ray</i> dada. Menggunakan arsitektur model <i>Xception</i> . Menggunakan <i>transfer learning</i> . Menggunakan metrik evaluasi untuk mengukur tingkat keberhasilan klasifikasi model.	Berfokus pada pneumotoraks. Memodifikasi arsitektur seperti menambahkan <i>attention module</i> . Penelitian Pneumotoraks berfokus pada pengembangan model. Menggunakan metrik AUC sebagai evaluasi kinerja model.
8	<i>Prediction of fetal brain gestational age using multihead attention with</i>	Penggunaan model <i>Xception</i> yang telah dilatih sebelumnya dan mekanisme <i>multihead attention</i> (MHA)	Menggunakan arsitektur model CNN yaitu <i>Xception</i> . Menggunakan	Menggunakan dataset MRI otak janin. Memodifikasi model <i>Xception</i>

No	Judul	Pembahasan	Keterkaitan	Perbedaan
	<i>Xception</i> (Hasan <i>et al.</i> , 2024).	digunakan untuk memprediksi usia gestasi dari gambar MRI otak janin. Penelitian ini menunjukkan bahwa model yang diusulkan dapat membantu klinisi dalam memprediksi usia gestasi dengan akurasi tinggi.	<i>transfer learning</i> . Menggunakan metrik evaluasi untuk mengukur tingkat keberhasilan klasifikasi model.	dengan MHA. Menggunakan <i>Mean Absolute Error</i> (MAE) untuk mengevaluasi hasil regresi.
9	<i>Boosted dipper throated optimization algorithm-based Xception neural network for skin cancer diagnosis: An optimal approach</i> (Tang and Rashid Sheykhahmad, 2024).	Algoritma <i>Boosted Dipper Throated</i> (BDTO) adalah teknik optimasi yang terinspirasi dari perilaku burung <i>Dipper Throated</i> yang dapat menentukan parameter dan bobot optimal untuk arsitektur CNN <i>Xception</i> . Sedangkan dataset <i>International Skin Imaging Collaboration</i> (ISIC) digunakan untuk meningkatkan kualitas dan keragaman gambar, serta menggunakan augmentasi data. Metode ini menunjukkan kinerja yang lebih baik dibandingkan dengan pendekatan kontemporer lainnya.	Menggunakan arsitektur model CNN <i>Xception</i> . Menggunakan augmentasi data.	Menggunakan dataset ISIC yang terdiri dari citra kulit. Menggunakan optimasi model <i>Xception</i> dengan algoritma BDTO.

No	Judul	Pembahasan	Keterkaitan	Perbedaan
10	<i>Concatenated Xception-ResNet50 — A novel hybrid approach for accurate skin cancer prediction</i> (Panthakkan <i>et al.</i> , 2022).	Menggunakan pendekatan hibrida antara dua model <i>Xception</i> dan <i>ResNet50</i> model dilatih dan diuji menggunakan teknik <i>sliding window</i> untuk memastikan kinerja yang konsisten dan menghindari <i>overfitting</i> . Model yang diusulkan mencapai akurasi yang tinggi.	Menggunakan arsitektur model CNN yaitu <i>Xception</i> . Menerapkan penghindaran <i>overfitting</i> dan augmentasi data.	Jenis penyakit yang diklasifikasikan berbeda kanker kulit. Menggunakan pendekatan hibrida antara dua model <i>Xception</i> dan <i>ResNet50</i> .

Tabel 2. 1 Studi Literatur

2.2 Dasar Teori

2.2.1 Pneumonia

Pneumonia adalah salah satu jenis penyakit paru paru yang disebabkan oleh bakteri, virus, jamur atau parasit. Pneumonia memiliki sebuah kondisi dimana alveoli penderita dipenuhi nanah dan cairan sehingga penderita mengalami kesakitan saat bernafas dan oksigen yang masuk kedalam paru paru kurang (Rindrasari, 2018). Pasien yang mengidap pneumonia biasanya mengalami gejala berupa sesak napas, batuk berdahak, demam ataupun menggigil. Pneumonia biasanya ditularkan dengan berbagai cara antara lain melalui batuk dan bersin (Sartiwi *et al.*, n.d).

Pneumonia merupakan masalah kesehatan yang signifikan di Indonesia, berdasarkan data Riskesdas (2013) terjadi peningkatan prevalansi pneumonia pada semua umur, terutama pada balita yang cukup tinggi yaitu 4,5 per 100 balita. Hal ini dari 4,5 per 100 balita menderita penyakit pneumonia. Sementara itu, menurut laporan WHO pada tahun 2017 menunjukkan, 15% kematian anak dibawah umur

5 tahun disebabkan oleh pneumonia. Berdasarkan survey Balitbangkes 2016 jumlah anak yang menderita pneumonia di Indonesia diperkirakan melebihi dari 800.000 anak. Pada pasien dewasa, kasus pneumonia paling banyak terjadi pada rentan umur 56-65 tahun. Kondisi ini dipengaruhi oleh perubahan anatomi dan fisiologi akibat proses penuaan, seperti penurunan elastisitas jaringan paru-paru, melemahnya sistem kekebalan tubuh. Faktor-faktor tersebut dapat menyebabkan penurunan kemampuan paru-paru untuk terkena infeksi paru-paru, sehingga lansia rentan terhadap pneumonia.

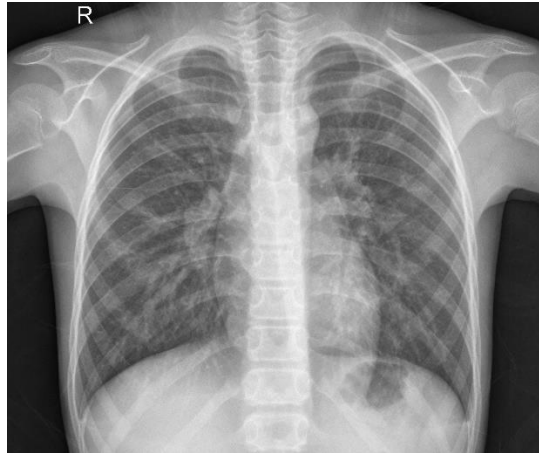
Pneumonia juga dapat dipengaruhi oleh berbagai faktor lain seperti kebiasaan merokok, paparan polusi udara, serta kondisi lingkungan yang tidak higienis (Nurin, 2024). Selain itu seseorang dengan penyakit seperti diabetes, penyakit jantung, atau gangguan sistem imun lebih rentan terjangkit pneumonia (CDC, 2023). Pencegahan penyakit ini dapat dilakukan dengan menerapkan gaya hidup sehat seperti menjaga kebersihan tangan, menghindari kontak dengan orang yang terinfeksi, serta menjalani vaksinasi, seperti vaksin pneumokokus dan influenza . Langkah-langkah pencegahan ini sangat penting untuk mengurangi risiko terkena pneumonia, terutama bagi kelompok rentan seperti anak-anak, lansia, dan individu dengan penyakit kronis (Pittara, 2022).

2.2.2 Chest X-Ray

Chest X-Ray (CXR) adalah suatu proyeksi pencitraan medis yang menggunakan radiasi untuk menghasilkan gambar dari organ atau struktur didalam dada. Termasuk seperti paru-paru, jantung, tulang rusuk dan pembuluh darah. *Chest X-Ray* alat yang sangat umum digunakan untuk mendiagnosis berbagai kondisi seperti infeksi paru-paru, gagal jantung, kanker paru-paru dan masalah yang lainnya (*Chest X-rays – The Heart Clinic*, 2015).

Citra paru-paru dapat dikategorikan normal apabila citra tersebut tidak ada bercak, teksturnya terlihat halus, serta ukuran paru-paru sesuai dengan anatomi

yang normal. Sebaliknya, citra paru-paru dianggap abnormal apabila citra tersebut memiliki bercak, teksturnya tidak terlihat halus, serta perubahan bentuk atau ukuran paru-paru berubah. Perubahan tersebut dapat diketahui dari intensi piksel pada objek paru-paru (Wikanargo and Thenata, 2018).



Gambar 2. 1 Chest X-Ray Paru-Paru

2.2.3 Citra Digital

Citra digital adalah sebuah representasi dari suatu objek yang disimpan dalam suatu bentuk elektronik yang dapat diolah oleh komputer. Citra digital digambarkan sebagai fungsi $f(x,y)$ dengan x dan y merupakan koordinat pada sebuah bidang datar yang mempresentasikan kumpulan pixel dalam dua dimensi (Gonzalez and Woods, 2008). Secara umum citra digital dibagi menjadi tiga, yaitu citra biner, citra *grayscale* dan citra RGB. Citra biner hanya terdiri dari dua nilai, yaitu hitam dan putih, yang mewakili objek dan latar belakang. Citra *grayscale* terdiri dari nilai intensitas abu-abu yang menggambarkan perbedaan kecerahan, sedangkan citra RGB menggabungkan tiga komponen warna dasar, yaitu merah (Red), hijau (Green), dan biru (Blue), untuk membentuk warna yang lebih kompleks (Sulistiyani Ratna, Setyawan Arianto and Komarudin, 2016).

Terdapat banyak metode pengolahan gambar yang dapat digunakan, salah satunya adalah augmentasi gambar. Augmentasi merupakan sebuah metode untuk

memperbanyak data gambar untuk training dengan membuat perubahan untuk memperoleh gambar training yang serupa tetapi berbeda. Memotong gambar dengan cara yang berbeda dapat memberikan posisi gambar yang berbeda, sehingga dapat mengurangi ketergantungan model pada posisi dimana objek muncul. Teknik augmentasi juga dapat memberi penyusaian terhadap kecerahan, warna, serta faktor-faktor lain untuk mengurangi sensitivitas model terhadap warna. Melakukan flip ke kiri dan ke kanan pada gambar umumnya tidak mengubah kategori dari objek. Sehingga, metode ini adalah salah satu metode augmentasi gambar yang sering digunakan (Zhang *et al.*, 2021).

2.2.4 Grayscale

Grayscale adalah format representasi citra yang menggunakan satu chanel untuk setiap piksel dengan nilai kecerahan yang bervariasi dari 0 hingga 255. Nilai ini menggambarkan intensitas Cahaya yaitu, 0 mewakili hitam (tanpa cahaya) dan 255 mewakili putih (intensitas cahaya maksimum) (Gonzalez and Woods, 2002). Dalam aplikasi pemrosesan citra *grayscale* sering digunakan karena informasi warna tidak selalu relevan, terutama pada jenis citra seperti *X-Ray*. Dimana analisis lebih berfokus pada pola intensitas daripada kombinasi warna.

Penggunaan citra *grayscale* memiliki peran penting dalam pengolahan citra medis seperti *X-Ray* paru-paru. Citra *X-Ray* umumnya sudah berupa *grayscale* secara *default*, karena tujuan utama dari citra tersebut adalah untuk menunjukkan detail struktur internal tubuh berdasarkan distribusi intensitas cahaya yang melewati jaringan tubuh. Dengan menggunakan format *grayscale*, analisis fitur seperti pola abnormal, perbedaan densitas jaringan, atau identifikasi anomali menjadi lebih mudah dilakukan (Ramdhan, Bustomi and Faridawati, 2014). Selain itu, penggunaan citra *grayscale* membantu menyederhanakan algoritma pemrosesan, termasuk metode peningkatan kontras seperti *Contrast Limited Adaptive Histogram Equalization* (CLAHE), karena metode ini hanya berfokus pada distribusi intensitas piksel tanpa memperhatikan informasi warna.

2.2.5 Contrast Limited Adaptive Histogram Equalization (CLAHE)

Pre-processing bertujuan untuk meningkatkan kualitas citra, sehingga memudahkan dan mempercepat kinerja sistem dalam mengenali citra *X-Ray* paru-paru. Proses ini sangat penting untuk memastikan bahwa citra yang digunakan oleh model deep learning memiliki kualitas yang optimal. Tanpa *pre-processing* yang tepat sistem dapat kesulitan untuk mengidentifikasi fitur-fitur penting dalam citra. Salah satu metode *pre-processing* citra untuk meningkatkan kontras gambar adalah CLAHE.

CLAHE merupakan sebuah metode dengan memberikan nilai batas pada histogram. Nilai batas ini disebut dengan clip limit yang menyatakan batas maksimum tinggi suatu histogram. Dengan menggunakan CLAHE kontras citra dapat diperbaiki terutama pada area dengan kontras rendah. Metode ini sangat berguna untuk citra medis seperti *X-Ray* yang seringkali memiliki detail yang sulit terlihat karena kontras yang rendah. Metode ini bertugas membagi citra menjadi beberapa bagian kecil dan masing-masing bagian dihitung histogramnya secara terpisah. Proses ini dilakukan untuk menghindari efek noise yang berlebihan di area dengan kontras tinggi.

Proses CLAHE terdiri dari dua tahap, yaitu pada tahap pertama membagi citra kedalam beberapa bagian dengan ukuran yang sama dan merata. Pada tahap kedua menghitung nilai histogram masing-masing bagian sehingga mendapatkan nilai *clip limit* dari citra tersebut. Dengan begitu, area yang memiliki kontras rendah dapat ditingkatkan tanpa merusak detail yang ada pada area dengan kontras tinggi. Cara menghitung clip limit suatu histogram dapat didefinisikan dengan persamaan berikut (Koonsanit *et al.*, 2017).

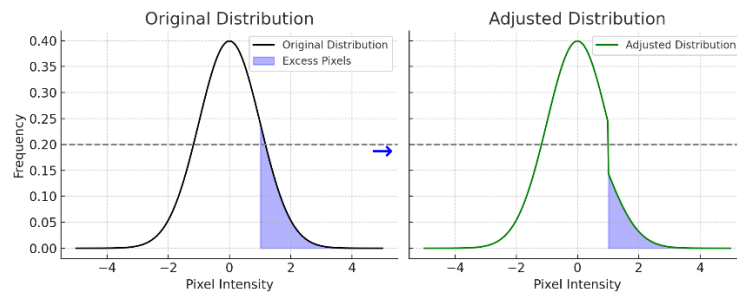
$$\delta = \frac{QR}{C} \left(1 + \frac{a}{100} (gr_{max} - 1) \right) \quad (2.1)$$

Keterangan:

δ = clip limit

$Q \times R$ = luas citra dimensi citra
 C = nilai komponen warna
 a = clip factor (batas limit suatu histogram)
 gr_{max} = nilai gradien maksimum

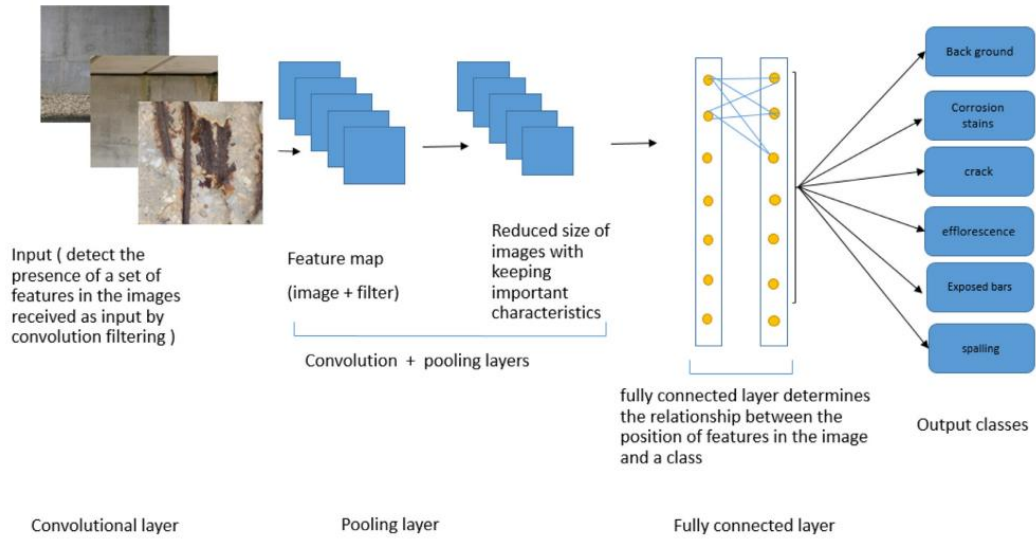
Histogram diatas nilai *clip limit* dianggap kelebihan (*excess*) piksel yang akan didistribusikan kepada area sekitar di bawah *clip limit*, sehingga histogram merata (). Ilustrasi distribusi *excess* piksel dapat dilihat pada gambar 2.2.



Gambar 2. 2 Distribusi excess pixel pada histogram

2.2.6 Convolutional Neural Network (CNN)

CNN adalah salah satu metode dari deep learning yang digunakan dalam klasifikasi citra gambar. Pada dasarnya, CNN tidak jauh berbeda dengan neural network biasanya yang memiliki bobot, bias dan fungsi aktivasi. CNN mengatur neuron dalam bentuk tiga dimensi yaitu lebar, panjang dan tinggi. Secara umum, arsitektur CNN terdiri dari tiga *layer* utama yaitu *convolutional layer*, *pooling layer* dan *fully connected layer* (Sewak, Karim and Pujari, 2018). Berikut merupakan ilustrasi arsitektur metode CNN yang ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Arsitektur Convolutional Neural Network (CNN)

a. *Convolutional Layer*

Fungsi utama dari *convolution layer* adalah mengekstraksi fitur dari citra input. Proses konvolusi adalah mengalikan sebuah gambar dengan sebuah convolution kernel atau filter yang dinyatakan dalam bentuk matrik dengan ukuran yang biasanya lebih kecil dari ukuran gambar. *Convolutional layer* membutuhkan input yang kemudian mengaplikasikan *convotional kernel* dan memberikan sebuah hasil berupa feature map sebagai output yang diilustrasikan pada Gambar 2.4 (Zhang *et al.*, 2021). Operasi konvolusi apabila input dan kernel berukuran dua dimensi ditulis pada persamaan (2.2).

$$FM_{a,b,l} = bias + \sum_c^C \sum_d^D Z_{c,d,l} X_{a+c-1,b+d-1,l} \quad (2.2)$$

dimana :

$FM_{a,b,l}$ = feature map pada pixel ke-a,b pada citra ke-l

$bias$ = bias pada feature map

$Z_{c,d,l}$ = bobot pada pada convolutional kernel ke-c,d pada citra ke-l

$X = \text{input}$

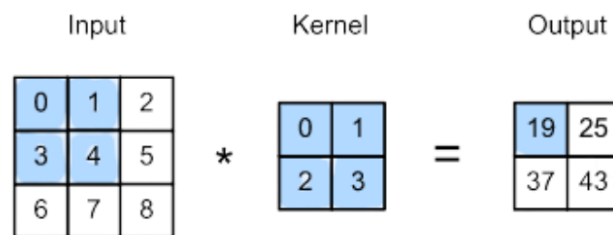
$a = 1, 2, \dots, A$. A merupakan panjang pixel pada *feature map*

$b = 1, 2, \dots, B$. B merupakan lebar pixel pada *feature map*

$c = 1, 2, \dots, C$. C merupakan panjang pixel pada *convolutional kernel*

$d = 1, 2, \dots, D$. D merupakan lebar pixel pada *convolutional kernel*

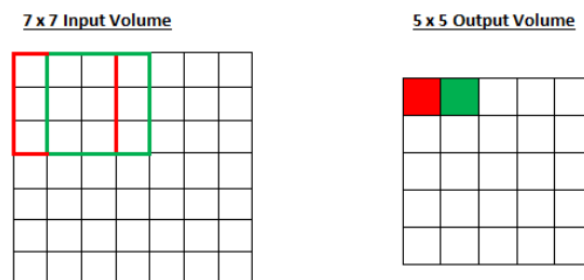
$l = 1, 2, \dots, L$. L merupakan citra yang digunakan



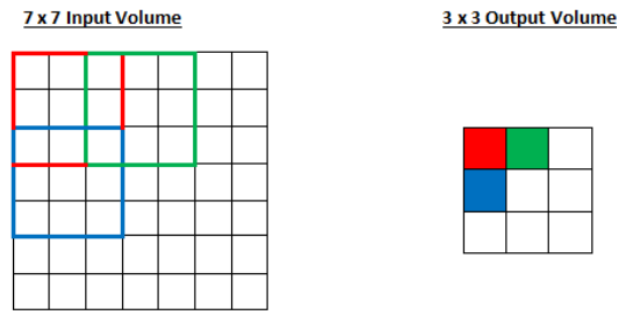
Gambar 2. 4 Ilustrasi Convolutional Layer

Dalam perhitungan output, kernel digeser untuk setiap pixel. Demi meningkatkan efisiensi komputasi, kernel digeser lebih dari satu pixel. Banyaknya pixel yang digeser dalam input disebut stride yang ditampilkan pada Gambar 2.5.

Stride 1

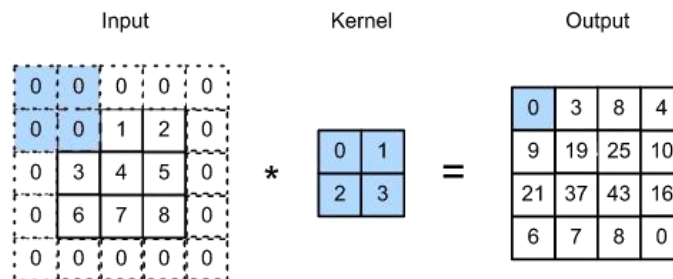


Stride 2



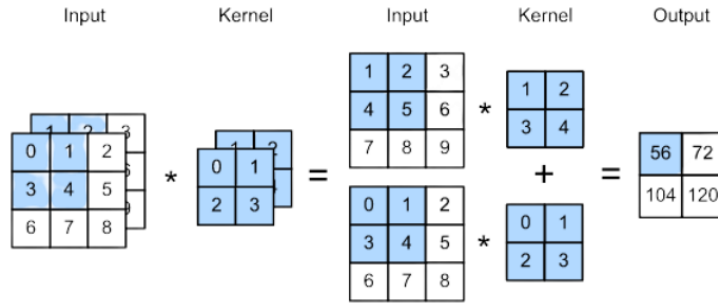
Gambar 2. 5 Perbedaan Antara Stride 1 dan Stride 2

Salah satu masalah yang terjadi akibat pengaplikasian *convutional layer* adalah kehilangan informasi *pixel* yang terletak pada bagian tepi gambar. Solusi yang mudah adalah menambah *pixel* tambahan dipinggiran gambar, sehingga memperbesar dimensi gambar. Biasanya, *pixel* tambahan bernilai 0 yang diilustrasikan pada Gambar 2.6.



Gambar 2. 6 Ilustrasi Padding Berukuran 1x1

Jika input data mengandung banyak *chanel*, maka perlu dibentuk *kernel* konvolusi yang memiliki *chanel* sebanyak *chanel* input sehingga dapat dilakukan korelasi silang. Apabila diasumsikan jika banyaknya *chanel* input adalah c_i , maka kernel konvolusi harus sebanyak c_i juga. Korelasi silang dihitung dengan menjumlahkan operasi konvolusi untuk setiap *chanel*, sehingga didapatkan output berdimensi dua yang diilustrasikan pada gambar 2.7.



Gambar 2. 7 Ilustrasi Padding Berukuran 1x1

Terlepas dari banyaknya input *channel*, sejauh ini menghasilkan output dengan *channel* tunggal. Akan tetapi, arsitektur CNN yang populer pada umumnya memiliki output dengan banyak *channel*. Hal ini dilakukan untuk menyimpan lebih banyak informasi ketika *pooling* dilakukan.

b. Pooling Layer

Pooling layer berada setelah convolution layer dan tidak memiliki parameter. *Pooling layer* bersifat deterministik sehingga fungsi yang bisa digunakan adalah maksimum dan mean untuk mengurangi input data (Zhang *et al.*, 2021). Operasi ini masing-masing dinamakan *max pooling* dan *average pooling*. Fungsi *max pooling* disajikan pada persamaan berikut (Hafemann, Sabourin and Oliveira, 2017).

$$Pool_{a,b,l} = \max_{i,j} (FM_{(a+i),(b+j),l}) \quad (2.3)$$

FM = feature map

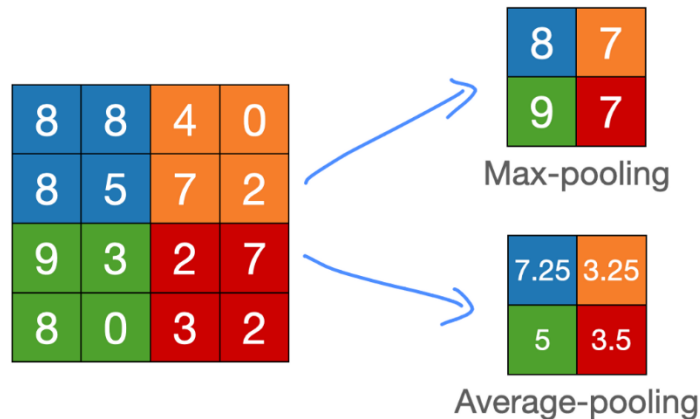
$Pool$ = hasil *pooling layer*

$a = 1, 2, \dots, A$. A merupakan panjang pixel

$b = 1, 2, \dots, B$. B merupakan lebar pixel

$c = 1, 2, \dots, C$. C merupakan citra yang digunakan

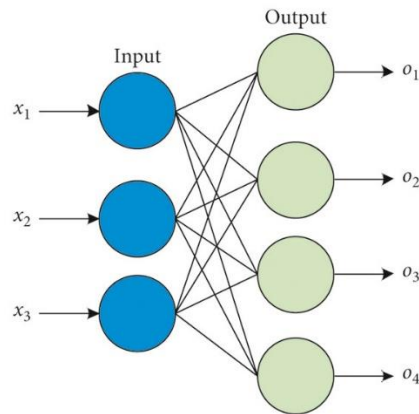
Ilustrasi operasi *max pooling* dan *average pooling* yang dapat dilihat pada gambar. *Pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*.



Gambar 2. 8 Pooling Layer

c. Fully Connected Layer

Fully connected layer merupakan bagian terakhir dari arsitektur CNN yang digunakan untuk melakukan transformasi dimensi data agar dapat diklasifikasikan secara linear. Layer ini mengkoneksasikan semua neuron disatu layer dan kesemua neuron layer lainnya (Zhang *et al.*, 2021). Citra digital yang memiliki dua atau tiga dimensi akan diubah menjadi suatu vector (data berdimensi satu) sebelum masuk ke *fully connected layer* karena input layer ini adalah suatu vektor. Fully connected layer memiliki fungsi aktivasi, *output layer* dan *loss function*. Berikut merupakan ilustrasi *fully connected layer* tanpa menggunakan hidden layer yang akan ditunjukkan pada gambar 2.9.



Gambar 2. 9 Ilustrasi Fully Connected Layer

Metode CNN menggunakan fungsi aktivasi pada *convolutional layer* sebelum *pooling layer* adalah fungsi aktivasi ReLu yang tercantum pada persamaa (2.5) Sedangkan fungsi aktivasi yang digunakan pada layer terakhir adalah fungsi aktivasi sigmoid yang tercantum pada persamaan (2.4), jika kelas klasifikasi yang digunakan adalah biner.

2.2.7 Fungsi Aktivasi Sigmoid

Fungsi aktivasi sigmoid merupakan fungsi aktivasi yang memiliki nilai pada *range* 0 sampai 1 (Zhang *et al.*, 2021). Fungsi aktivasi sigmoid digunakan untuk kelas biner pada klasifikasi yang digunakan pada output layer. Perhitungan fungsi sigmoid ditunjukkan pada persamaan (2.4) sebagai berikut.

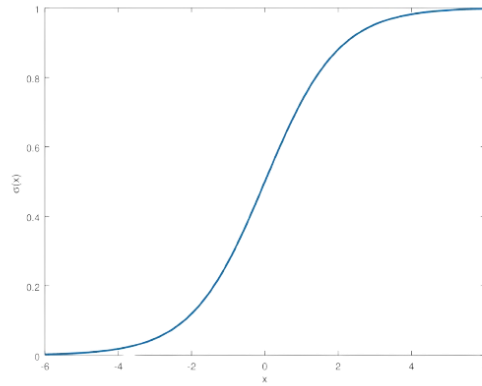
$$p(y_k) = \frac{1}{1 + e^{-r}} \quad (2.4)$$

dimana :

r = output setiap neuron output layer tanpa fungsi aktivasi

$p(y_k)$ = output setiap neuron *output layer* setelah dimasukkan dalam fungsi aktivasi

Berikut merupakan Ilustrasi fungsi sigmoid ditunjukkan pada gambar 2.10.



Gambar 2. 10 Distribusi Fungsi Sigmoid

2.2.8 Fungsi Aktivasi *Rectified Linear Unit* (ReLU)

Fungsi aktivasi ReLu merupakan fungsi aktivasi yang menghilangkan *vanishing gradient* yang cukup populer digunakan (Zhang *et al.*, 2021). Fungsi ReLu digunakan sebagai non-linearitas telah menunjukkan kemungkinan pelatihan CNN tanpa memerlukan unsupervised pre-training. Perhitungan fungsi ReLu ditunjukkan pada persamaan berikut.

$$f(FM_{a,b,l}) = \max(0, FM_{a,b,l}) = \begin{cases} FM_{a,b,l}, & \text{jika } FM_{a,b,l} \geq 0 \\ 0, & \text{jika } FM_{a,b,l} < 0 \end{cases} \quad (2.5)$$

dimana :

FM = *feature map*

$a = 1, 2, \dots, A$. A merupakan panjang *pixel*

$b = 1, 2, \dots, B$. B merupakan lebar *pixel*

$l = 1, 2, \dots, L$. L merupakan citra yang digunakan

2.2.6.3 Loss Function

Loss function adalah nilai error antara nilai aktual dengan nilai yang akan diprediksi. *Loss function* biasanya berupa nilai non-negatif dimana jika bernilai

lebih kecil akan menghasilkan prediksi yang lebih akurat (Zhang *et al.*, 2021). Terdapat banyak metode untuk mengukur nilai error pada *deep learning*. Pada klasifikasi yang memiliki dua kelas perhitungan loss yang digunakan adalah *binary-crossentropy*. Perhitungan *binary-crossentropy* ditunjukkan pada persamaan berikut.

$$Loss = -\frac{1}{N} \sum_{i=1}^N [y_k \times \log(p(y_k)) + (1 - y_k) \times \log(1 - p(y_k))] \quad (2.6)$$

dimana :

N = jumlah data

y_k = nilai target berupa nilai 0 dan 1

$p(y_k)$ = *predicted value* didapatkan dari persamaan (2.4)

2.2.9 Optimasi Parameter *Adaptive Moment Estimation* (ADAM)

Optimasi parameter digunakan untuk meminumkan nilai loss sehingga loss merupakan kunci dalam optimasi parameter bias dan bobot. Optimasi parameter yang digunakan dalam penelitian ini yaitu optimasi parameter ADAM. ADAM menggunakan gradient, lalu estimasi momen pertama dan kedua serta mengoreksi dengan *bias correction*. Penyelesaian gradient pada network ini menggunakan *chain rule* untuk mendapatkan *partial derivative* (Bishop, 2006). Dalam mengoptimasi parameter dengan menggunakan Adam, hal pertama yang harus dilakukan adalah menghitung gradient dari loss function terhadap parameter yaitu bias dan pembobot, sehingga berikut persamaan yang digunakan.

$$g_t = \frac{\partial Loss}{\partial \theta} \quad (2.7)$$

Hal kedua setelah menghitung gradient dari loss function adalah menghitung Momentum term dan RMSProp term, sehingga berikut persamaan yang digunakan.

Momentum term :

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1)g_t \quad (2.8)$$

Dimana :

β_1 = koefisien untuk momentum

RMSProp term :

$$v_t = \beta_1 \times v_{t-1} + (1 - \beta_1)g_t^2 \quad (2.9)$$

Dimana :

β_2 = koefisien untuk estimasi kedua

Hal ketiga setelah menghitung Momentum term dan RMSProp term adalah menghitung Bias correction, sehingga berikut persamaan yang digunakan

Bias correction :

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.10)$$

Langkah terakhir ialah menghitung parameter update, sehingga berikut persamaan yang digunakan.

Dimana :

β_1^t = iterasi saat ini dan koefisien momentum

β_2^t = iterasi saat ini dan koefisien untuk estimasi kedua

Parameter update :

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (2.11)$$

Dimana :

η = Learning rate,

ϵ = Stabilizer untuk menghindari pembagian dari nol.

2.2.10 Transfer Learning

Transfer Learning adalah salah satu metode dari *deep learning* dengan memanfaatkan model yang sudah dilatih terhadap suatu dataset besar untuk menyelesaikan permasalahan lain. Teknik ini digunakan untuk mempercepat proses *training* dan meningkatkan kualitas model. Dengan menggunakan *pre-trained* model dapat menghemat sumber daya komputasi dan waktu yang diperlukan untuk melatih model dari awal (Patterson and Gibson, 2017). Contoh *pre-trained* model adalah *ResNet*, *VGGNet*, *InceptionV3*, *DenseNet*, *EfficientNet* dan *MobilNet*. Model-model ini telah dilatih pada dataset skala besar dan kompleks sehingga memiliki kemampuan yang baik untuk berbagai tugas pengenalan pola dan klasifikasi. Setiap model memiliki arsitektur yang unik dan kelebihan masing-masing yang dapat dipilih berdasarkan kebutuhan spesifik dari permasalahan yang dihadapi.

Salah satu dataset yang sering digunakan untuk melatih *pre-trained* model adalah *ImageNet*. Dataset ini berisi sekitar 1,2 juta gambar dengan 1.000 kelas yang berbeda, mencakup berbagai objek dan scene (Iswari, 2021). Dengan dilatih pada dataset yang besar, *pre-trained* model seperti *DenseNet* dan *Xception* mampu menangkap fitur-fitur yang sangat beragam. Sehingga dapat diterapkan pada berbagai aplikasi dari pengenalan objek hingga segmentasi gambar. Penggunaan *pre-trained* model memungkinkan transfer pengetahuan dari satu domain ke domain lain. Meningkatkan efisiensi dan efektivitas dalam pengembangan model *deep learning*.

Selain efisiensi dan kemampuan generalisasi yang baik, *Transfer Learning* juga memungkinkan pengembangan model dengan dataset yang relatif kecil. Ketika dataset yang tersedia terbatas pelatihan model dari awal bisa menjadi tantangan besar (Hosna *et al.*, 2022). Dalam situasi ini, *pre-trained* model dapat bertindak sebagai dasar yang kuat, di mana model hanya perlu disesuaikan atau di *fine-tune* pada dataset spesifik yang lebih kecil.

a. Ekstraksi Fitur

Ekstraksi fitur adalah proses penting dalam pemrosesan citra yang bertujuan untuk mengambil informasi signifikan atau pola yang merepresentasikan karakteristik utama dari sebuah citra. Proses ini bertujuan untuk mereduksi dimensi data mentah dengan tetap mempertahankan informasi penting yang dapat digunakan untuk analisis atau klasifikasi. Dalam *deep learning* ekstraksi fitur dilakukan secara otomatis oleh lapisan-lapisan dalam model CNN (Huang *et al.*, 2016). Salah satu metode yang umum digunakan dalam *transfer learning* adalah memanfaatkan model *pre-trained*, seperti *DenseNet*, *ResNet*, atau *Xception*, *VGGNet* yang sudah dilatih pada dataset besar seperti *ImageNet*. Model ini berfungsi sebagai base model untuk melakukan ekstraksi fitur dari data baru.

Pada *transfer learning*, lapisan-lapisan dalam model *pre-trained* sering kali dibekukan dengan mengatur properti *False*. Hal ini bertujuan untuk menjaga parameter yang telah dipelajari dari dataset awal sehingga model dapat berfungsi sebagai ekstraktor fitur. Fitur-fitur yang telah diekstraksi kemudian dapat digunakan untuk *custom layers* yang dirancang khusus untuk tugas tertentu, seperti klasifikasi *binary* atau multi-kelas. Ekstraksi fitur otomatis ini memungkinkan, memanfaatkan keunggulan model *pre-trained* tanpa perlu melatih ulang seluruh model, sehingga menghemat waktu dan sumber daya komputasi (He *et al.*, 2015).

b. Task Specifics Layers

Task specific layers adalah lapisan yang ditambahkan ke model *deep learning* yang sudah dilatih sebelumnya untuk menyesuaikan model tersebut dengan tugas atau masalah tertentu (Rahman, Baras and Chellappa, 2025). Dalam konteks *transfer learning* model *pre-trained* seperti *DenseNet*, *VGGNet*, atau *ResNet* digunakan untuk melakukan ekstraksi fitur dari data baru. Fitur-fitur yang telah dipelajari dari dataset besar seperti *ImageNet* digunakan untuk memahami pola-pola dasar dalam data, seperti tepi, tekstur, atau bentuk. Namun, untuk

menyelesaikan tugas tertentu, seperti klasifikasi citra atau deteksi objek, model memerlukan lapisan tambahan yang dirancang khusus untuk tugas tersebut.

Lapisan yang ditambahkan ini sering disebut sebagai *task specific layers*, yang bertujuan untuk mengambil fitur yang telah diekstraksi oleh model *pre-trained* dan memanfaatkan informasi tersebut untuk memecahkan masalah spesifik. Keuntungan utama dari penggunaan *task specific layers* adalah memungkinkan model untuk memanfaatkan data yang telah dipelajari dari data besar tanpa memerlukan pelatihan ulang pada seluruh model, sehingga menghemat waktu dan sumber daya komputasi (Rahman, Baras and Chellappa, 2025).

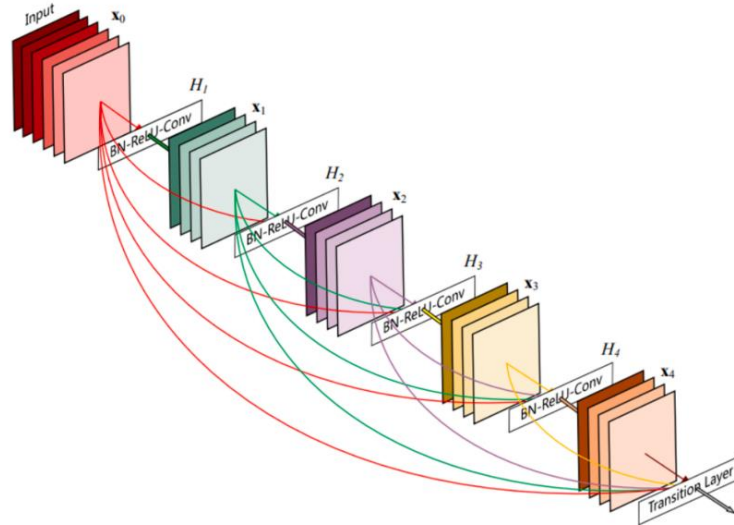
2.2.11 Model *DenseNet201*

DenseNet201 adalah salah satu metode *deep learning* dari CNN yang terdiri dari 201 lapisan, yang diperkenalkan oleh Gao Huang dkk, pada tahun 2017 (Virnodkar *et al.*, 2022). Jaringan ini dikenal karena menggunakan *dense connections* yang terdiri dari *dense blocks* dan *transition layers*. *Dense blocks* yang terdiri dari *Bottleneck Layers*, berfungsi sebagai komponen utama dalam jaringan model. Mode koneksi padat didalam *dense blocks* memungkinkan setiap lapisan terhubung langsung dengan lapisan lainnya, sehingga informasi dapat tersalurkan dengan optimal dan ukuran keluaran tetap konsisten diseluruh lapisan. *DenseNet* mengatur jumlah *channels* melalui *bottleneck layers*, *transition layers* dan *growth layer* (Huang *et al.*, 2016). Pendekatan ini tidak hanya dapat mengurangi jumlah parameter dan mengatasi masalah *overfitting* akan tetapi juga dapat secara signifikan menurunkan beban komputasi. Secara keseluruhan arsitektur *DenseNet201* yang digunakan dapat dilihat pada Gambar 2.11.

Salah satu keunggulan utama dari *DenseNet201* adalah kemampuannya dalam memitigasi masalah *vanishing gradient* yang sering menjadi tantangan dalam *deep learning*. Dengan menghubungkan setiap lapisan langsung ke setiap lapisan lainnya dalam *dense blocks* *DenseNet* memastikan bahwa gradien dapat mengalir lebih

mudah melalui jaringan selama proses pelatihan. Ini berarti bahwa informasi penting tidak hilang saat melalui banyak lapisan dan pelatihan dapat berlangsung lebih efisien dan efektif (Kong and Cheng, 2022).

DenseNet201 juga memberikan penawaran seperti efisiensi parameter yang tinggi. Karena *dense connections* memungkinkan reuse fitur dari lapisan sebelumnya, sehingga jumlah parameter yang diperlukan jauh lebih sedikit dibandingkan dengan arsitektur jaringan konvolusi tradisional (Devi, Chatrapati and Sandhya, 2024). Ini tidak hanya mengurangi kebutuhan akan sumber daya komputasi yang besar tetapi juga membantu dalam menghindari *overfitting*, terutama ketika bekerja dengan dataset yang lebih kecil.



Gambar 2. 11 Arsitektur DenseNet201

2.2.12 Model Xception

Xception adalah arsitektur model dari CNN *open source* yang diteliti dan dikembangkan oleh Google. Nama *Xception* sendiri merupakan singkatan dari *Extreme version of Inception* yang berarti Versi yang lebih ekstrem dari Inception atau bisa diartikan versi yang lebih intens dari model yang sebelumnya dibuat oleh google. Model ini menggabungkan beberapa inovasi dari Inception dan

mengembangkan pendekatan baru untuk meningkatkan kinerjanya. *Xception* dirancang untuk memberikan fleksibilitas dan performa tinggi dalam berbagai tugas klasifikasi gambar, sehingga yang menjadikannya salah satu model populer dalam bidang klasifikasi gambar (Chollet, 2017).

Xception memiliki lapisan konvolusi yang menggabungkan *pointwise convolution* yang kemudian diikuti oleh *depthwise convolution*. Kombinasi ini memungkinkan *Xception* untuk menghemat jumlah parameter dan meningkatkan efisiensi komputasi sekaligus mempertahankan akurasi. Dari kombinasi tersebut terbukti memberikan kinerja yang lebih baik dalam melakukan klasifikasi gambar pada berbagai dataset besar seperti *ImageNet* (Chollet, 2017).

Arsitektur *Xception* sendiri memiliki lapisan yang terdiri dari 36 lapisan *konvolusi* yang menjadi dasar jaringan ekstraksi fitur. Ke-36 lapisan *konvolusi* tersebut disusun kedalam 14 modul, dimana semuanya memiliki koneksi *residual linear* disekitarnya kecuali untuk modul pertama dan terakhir. Selain itu *Xception* sendiri memiliki arsitektur yang memudahkan dalam memodifikasi pada parameter-parameter yang dimilikinya (Carnegie *et al.*, 2022). Spesifik lengkap tentang jaringan *Xception* dapat dilihat Gambar 2.12.

(FP) atau dikenal sebagai kesalahan tipe I terjadi ketika model salah memprediksi data sebagai kelas positif padahal label sebenarnya adalah negatif. Sedangkan False Negative (FN) atau kesalahan tipe II terjadi ketika model salah memprediksi data sebagai kelas negative, label sebenarnya adalah positif (Choi *et al.*, 2024). Berikut merupakan *confusion matrix* yang memuat keempat nilai tersebut.

Kelas Aktual	Kelas Prediksi	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

Tabel 2. 2 *Confusion Matrix*

Kinerja klasifikasi dapat diukur dengan menggunakan *accuracy*, *precision*, *recall*, dan *Fscore*. *Accuracy* adalah banyak pengamatan yang terklasifikasi secara tepat. *Precision* adalah banyak pengamatan yang tepat terprediksi positif dari keseluruhan dengan hasil prediksi positif. *Recall* adalah banyaknya pengamatan yang tepat diklasifikasikan sesuai kategorinya. *Fscore* didapatkan dari nilai kombinasi antara *precision* dan *recall*. Dalam mengidentifikasi penyakit pada bidang medis diperlukan diagnosis yang tepat sebanyak mungkin sehingga hal yang perlu diperhatikan adalah proporsi penyakit yang teridentifikasi benar pada semua kasus penyakit. Oleh karena itu pada bidang medis sebaiknya menggunakan *sensitivity* (Chen, 2019). Perhitungan *accuracy*, *precision*, *recall* dan *Fscore* dapat dilakukan dengan menggunakan persamaan.

$$accuracy = \frac{TN+TP}{TN+TP+FN+FP} \quad (2.12)$$

$$precision = \frac{TP}{TP+FP} \quad (2.13)$$

$$recall = \frac{TP}{TP+FN} \quad (2.15)$$

$$fscore = \frac{2(precision \times recall)}{precision+recall} \quad (2.16)$$

2.2.14 Python

Python adalah bahasa pemrograman computer serbaguna yang sering digunakan untuk membangun situs website, software atau aplikasi, mengotomasikan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Fleksibilitasnya memungkinkan pengguna untuk menciptakan berbagai jenis program dimulai dari pengolahan data, pengembangan kecerdasan buatan, hingga mengotomasikan dalam bidang industri. Python juga memiliki sintaks yang sederhana dan mudah dipahami sehingga sangat cocok bagi pemula yang baru mulai belajar pemrograman (Sharma *et al.*, 2020).

Popularitas *python* terus meningkat berkat sifatnya yang mudah dipahami dan ekosistem yang luas. Berdasarkan survei pengembangan *Stack Overflow* tahun 2022, python menempati peringkat keempat sebagai bahasa pemrograman terpopuler. Hampir 50% responden mengaku menggunakan *python* untuk menyelesaikan tugas-tugas harian mereka. Dalam bidang pengembangan perangkat lunak *python* banyak digunakan banyak digunakan untuk pengolahan data, pembelajaran mesin dan pengembangan antarmuka *Application Programming Interface* (API). Bahkan python menjadi bahasa pemrograman dasar untuk *framework* seperti *Django* dan *Flask* yang memudahkan pengembangan aplikasi berbasis web (Marchand, 2022).

Nama *python* sendiri memiliki latar belakang yang unik. Ketika Guido van Rossum menciptakan bahasa ini pada akhir 1980-an ia terinspirasi dari acara komedi Inggris *Monty Python's Flying Circus* yang ia tonton saat itu. Menurutnya nama tersebut terdengar singkat, menarik, dan sedikit misterius. Hal ini mencerminkan filosofi *python* sebagai bahasa pemrograman yang sederhana dan mudah dipahami namun kuat (Rossum, 1996).

2.2.15 TensorFlow

TensorFlow adalah sebuah *framework open source* yang dirancang untuk komputasi dan *machine learning*. Dikembangkan oleh tim Google Brain pada tahun 2015. *TensorFlow* menawarkan fleksibilitas tinggi dalam membangun model *machine learning* dan *deep learning*. *Framework* ini mendukung operasi berbagai operasi matematika kompleks yang digunakan dalam pembuatan modeling seperti regresi, klasifikasi, pengenalan pola. *TensorFlow* memiliki arsitektur yang memungkinkan komputasi lintas platform dari perangkat *mobile* hingga *kluster* komputasi dasar menjadikannya pilihan populer dikalangan.

Salah satu fitur utama *TensorFlow* adalah kemampuannya untuk memanfaatkan grafik aliran data. Grafik ini memberitahu perhitungan dalam bentuk *nodes* dan *edges*, dimana *nodes* operasi matematika dan *edges* adalah data *tensor* yang mengalir diantara *nodes*. Pendekatan ini memungkinkan optimalisasi efisien, terutama untuk tugas-tugas yang memerlukan pemrosesan data besar secara paralel. *TensorFlow* mendukung berbagai bahasa pemrograman yang dapat memudahkan integrasi dengan sistem yang sudah ada seperti *Python*, *C++*, dan *javascript* (Abadi *et al.*, 2015).

Framework ini terus berkembang dengan adanya fitur seperti *TensorFlow Lite* untuk pengembangan aplikasi pada perangkat *mobile* seperti *TensorFlow.js* untuk implementasi pada aplikasi berbasis web, *TensorFlow Extended* (TFX) untuk manajemen alur kerja *machine learning* secara end-to-end. Dengan ekosistem yang luas dan komunitatif *TensorFlow* salah satu *framework* yang digunakan dalam pengembangan model kecerdasan buatan. Keunggulannya dalam skalabilitas dan fleksibilitas menjadikannya alat yang sangat efektif untuk menangani permasalahan data yang kompleks (*TensorFlow*, no date).

2.2.16 Google Colab

Google Colab adalah sebuah platform pemrograman berbasis cloud yang dikembangkan oleh Google. Google Colab memberikan layanan menjalankan kode Python langsung di browser tanpa memerlukan pengaturan lingkungan lokal. Dengan integrasi Google Drive, Google Colab mempermudah kolaborasi antar pengguna yang memungkinkan berbagai dan mengedit notebook secara bersamaan. Colab mendukung berbagai aplikasi termasuk data, visualisasi, pengembangan model *deep learning* dan *machine learning* yang menjadikannya populer di kalangan peneliti dan praktisi data (Carneiro *et al.*, 2018).

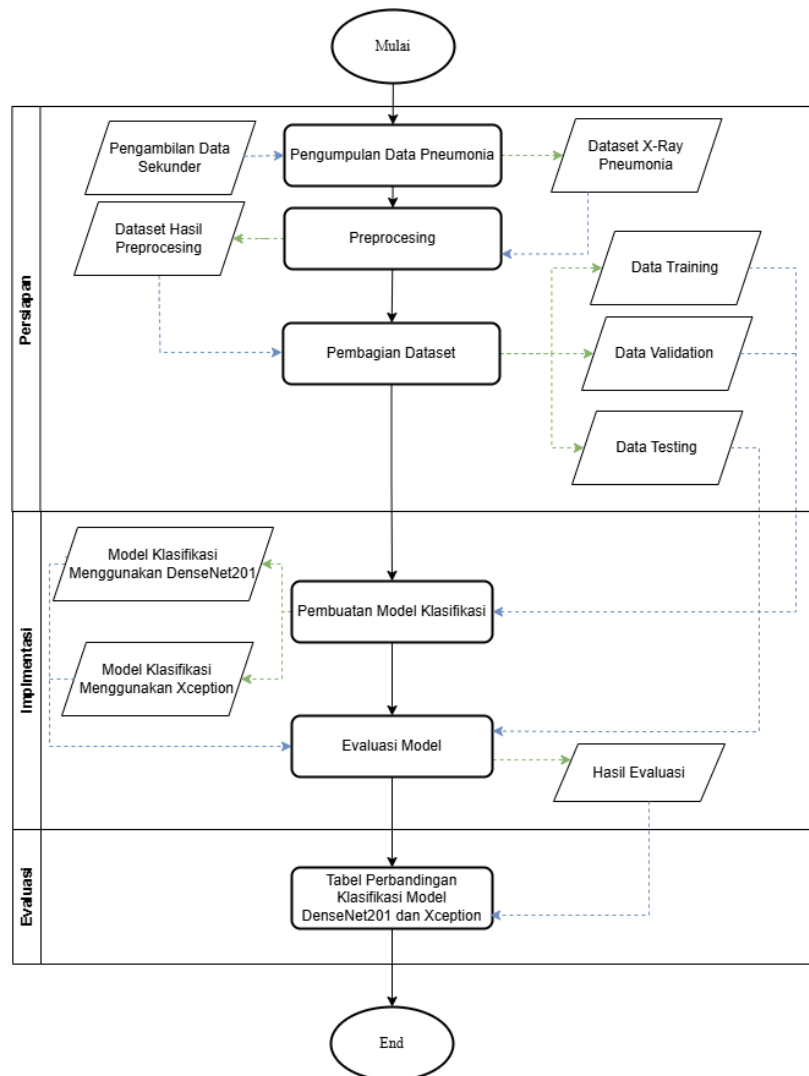
Colab menawarkan akses gratis ke sumber daya komputasi yang kuat seperti GPU dan TPU, yang dapat digunakan untuk mempercepat pelatihan model dalam *deep learning* dan *machine learning*. Google Colab juga menyediakan akses mudah ke berbagai library Python populer seperti *TensorFlow*, *NumPy*, dan *Matplotlib* tanpa memerlukan instalasi manual. Google Colab juga dapat mengintegrasikan dengan layanan cloud lain seperti *Google Cloud Storage* sehingga memperluas kemampuannya untuk mengelola data besar (Scholar and Bagane, 2024).

BAB 3

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Pada tahapan penelitian ini, akan dijelaskan mengenai alur penelitian yang digunakan dalam pengerjaan tugas akhir. Tahapan penelitian ini memberi pedoman berupa alur penelitian yang dilakukan selama penelitian berlangsung.



Gambar 3. 1 Flowchart Tahapan Penelitian

3.1.1 Persiapan

a. Pengumpulan Data Pneumonia

Pada proses ini merupakan tahapan yang digunakan untuk pengambilan data yang dibutuhkan dalam pengembangan model klasifikasi gambar. Dataset yang digunakan mencakup data citra *X-Ray* pneumonia dan *X-Ray* normal. Populasi penelitian ini berasal dari data sekunder yang diperoleh pada platform kaggle. Sampel penelitian ini berjumlah 5.863 gambar *X-Ray* berformat JPEG.

b. *Preprocessing*

Tahap *preprocessing* data terhadap data citra merupakan tahap awal yang penting untuk dilakukan sebelum melakukan proses pengenalan citra. Tahap ini bertujuan untuk mempersiapkan data citra agar dapat diolah dengan baik oleh model pengenalan citra.

1. Pengurangan data citra

Pengurangan data citra merupakan suatu proses dari *preprocessing* data yang bertujuan untuk menyederhanakan dataset tanpa mengurangi representasi atau kualitas informasi yang relevan. Proses ini dilakukan untuk memastikan dalam pengembangan model lebih terfokus dan efisien, baik dari segi waktu pemrosesan maupun kebutuhan sumber daya komputasi. Pengurangan data citra dilakukan secara manual dengan mengurangi jumlah data menjadi 1000 data untuk setiap label.

2. Menerapkan CLAHE pada data citra

CLAHE merupakan langkah yang digunakan meningkatkan kontras gambar sehingga fitur penting dalam citra menjadi lebih terlihat dan dapat diolah dengan baik oleh model. CLAHE bekerja dengan menyesuaikan histogram lokal pada setiap gambar, menghindari peningkatan kontras yang secara berlebihan pada data (Koonsanit *et al.*, 2017).

3. Augmentasi pada citra

Augmentasi data pada citra dilakukan untuk meningkatkan variasi dataset tanpa menambah jumlah gambar asli secara langsung. Augmentasi citra dilakukan menggunakan berbagai metode. Augmentasi citra yang tepat akan meningkatkan performa model pengenalan citra.

c. Pembagian Dataset

Dalam penelitian ini, terdapat 2000 data citra *X-Ray* yang dibagi menjadi tiga kelompok yaitu, data latih, data uji, dan data validasi. Data tersebut diambil secara acak dari keseluruhan dataset citra *X-Ray*. Sebanyak 1400 data *X-Ray* digunakan sebagai data latih untuk membangun sistem. Sebanyak 300 data validasi digunakan selama pelatihan untuk memonitor kinerja model. Sebanyak 300 data digunakan untuk data uji untuk mengevaluasi kinerja akhir model setelah pelatihan selesai.

3.1.2 Implementasi

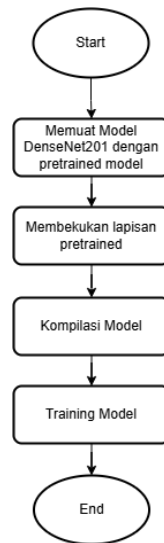
a. Pembuatan Model

Dalam tahap pembuatan model ini, model yang digunakan adalah *DenseNet201* dan *Xception*. Berikut adalah tahapan pembuatan model *DenseNet201* dan *Xception*.

1. Model *DenseNet201*

Model *DenseNet201* adalah yang terdiri dari 201 lapisan, yang mempunyai *dense connections* yang terdiri dari *dense blocks* dan *transition layers*. Setiap lapisan terhubung langsung dengan lapisan lainnya, sehingga informasi dapat tersalurkan dengan optimal dan ukuran keluaran tetap konsisten diseluruh lapisan (Huang *et al.*, 2016). Terdapat beberapa tahap untuk pembuatan model *DenseNet201* antara lain, model menggunakan *ImageNet* dengan memanfaatkan pretrained untuk membantu model dalam mengklasifikasikan, karena model tidak perlu lagi untuk mempelajari fitur-

fitur dasar dari awal yang memerlukan dataset banyak. Menambahkan *freeze* model yang memiliki fungsi agar lapisan Model yang sudah dilatih sebelumnya tidak akan diperbarui selama pelatihan, hal ini dilakukan untuk mencegah overfitting. Menambahkan kompilasi model yang digunakan untuk optimalisasi model seperti fungsi *loss*, kompilasi, pelatihan model, dan metrik evaluasi (Cobilla *et al.*, 2023).

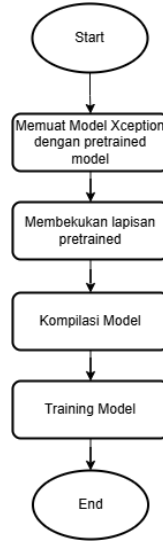


Gambar 3. 2 Alur Pembuatan Model DenseNet201

2. Model *Xception*

Model *Xception* adalah model memiliki lapisan konvolusi yang menggabungkan *pointwise convolution* yang kemudian diikuti oleh *depthwise convolution*, yang memiliki kelebihan menghemat jumlah parameter dan meningkatkan efisiensi komputasi sekaligus mempertahankan akurasi (Chollet, 2017). Terdapat beberapa tahap untuk pembuatan model *Xception* antara lain, model menggunakan *ImageNet* dengan memanfaatkan pretrained untuk membantu model dalam mengklasifikasikan, karena model tidak perlu lagi untuk mempelajari fitur-fitur dasar dari awal yang memerlukan dataset banyak. Menambahkan *freeze* model yang memiliki fungsi agar lapisan Model yang sudah dilatih sebelumnya tidak akan diperbarui selama pelatihan, hal ini dilakukan untuk

mencegah *overfitting*. Menambahkan kompilasi model yang digunakan untuk optimalisasi model seperti fungsi *loss*, kompilasi, pelatihan model (Cobilla *et al.*, 2023).



Gambar 3. 3 Alur Pembuatan Model Xception

b. Evaluasi Model

Kinerja klasifikasi Model DenseNet201 dan Model Xception diukur dengan menggunakan *accuracy*, *precision*, *sensitivity*, dan *Fscore* yang diambil dari dataset testing. Untuk mengetahui nilai dari *accuracy*, *precision*, *sensitivity*, dan *Fscore* dapat diketahui dari nilai TN, TP, FN, dan FP yang diperoleh dari *confusion matrix* (hart *et al.*, 2024). Perhitungan *accuracy*, *precision*, *recall* dan *Fscore* dapat dilakukan dengan menggunakan persamaan berikut.

$$akurasi = \frac{TN+TP}{TN+TP+FN+FP} \quad (3.1)$$

$$presisi = \frac{TP}{TP+FP} \quad (3.2)$$

$$recall = \frac{TP}{TP+TN} \quad (3.3)$$

$$f_{s1}core = \frac{2(precision \times recall)}{precision+recall} \quad (3.3)$$

Akurasi ini memiliki tingkat nilai diagnosa yaitu :

- Akurasi antara 0.90 dan 1.00 diklasifikasikan sebagai berikut “excellent classification” (klasifikasi sangat baik).
- Akurasi antara 0.80 dan 0.90 diklasifikasikan sebagai berikut “good classification” (klasifikasi baik).
- Akurasi antara 0.70 dan 0.80 diklasifikasikan sebagai berikut “fair classification” (klasifikasi cukup).
- Akurasi antara 0.60 dan 0.70 diklasifikasikan sebagai berikut “poor classification” (klasifikasi kurang baik).
- Akurasi antara 0.50 dan 0.60 diklasifikasikan sebagai berikut “failure” (gagal).

3.1.3 Evaluasi

a. Tabel Perbandingan Klasifikasi Model *DenseNet201* dan *Xception*

Perbandingan kinerja model CNN dengan arsitektur *DenseNet201* dan *Xception* dapat dilihat dari hasil kinerja klasifikasi yang terbaik, dengan membandingkan nilai akurasi, presision, recall, dan fscore pada tabel *confusion matrix* yang diperoleh dari hasil klasifikasi terbaik.

3.2 Jadwal Kegiatan

No	KEGIATAN	Bulan I				Bulan II				Bulan III				Bulan IV			
		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
1	Pengumpulan Data Pneumonia																
2	Preprocessing Data																
3	Pembagian dataset																
4	Permodelan Model Klasifikasi																
5	Evaluasi Model																
6	Perbandingan Model																

Tabel 3. 1 Tabel Kegiatan

DAFTAR PUSTAKA

Abadi, M. *et al.* (2015) *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. Available at: www.tensorflow.org.

Andreu, J. *et al.* (2004) ‘Radiological manifestations of pulmonary tuberculosis’, *European Journal of Radiology*, 51(2), pp. 139–149. Available at: <https://doi.org/10.1016/j.ejrad.2004.03.009>.

Bishop, C.M.. (2006) *Pattern recognition and machine learning*. Springer Science + Business Media.

Carnegie, J.O. *et al.* (2022) ‘Essential Oil Plants Image Classification Using Xception Model’, in *Procedia Computer Science*. Elsevier B.V., pp. 395–402. Available at: <https://doi.org/10.1016/j.procs.2022.08.048>.

Carneiro, T. *et al.* (2018) ‘Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications’, *IEEE Access*, 6, pp. 61677–61685. Available at: <https://doi.org/10.1109/ACCESS.2018.2874767>.

CDC (2023) *Risk Factors for Pneumonia | Pneumonia | CDC*. Available at: <https://www.cdc.gov/pneumonia/risk-factors/index.html?form=MG0AV3> (Accessed: 21 December 2024).

Chauhan, T., Palivela, H. and Tiwari, S. (2021) ‘Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging’, *International Journal of Information Management Data Insights*, 1(2). Available at: <https://doi.org/10.1016/j.jjime.2021.100020>.

Chen, X. (2019) *Image enhancement effect on the performance of convolutional neural networks*. Available at: www.bth.se.

Chest X-rays – The Heart Clinic (2015). Available at: <https://heartclinics.org/terminology/chest-x-rays/> (Accessed: 8 December 2024).

Choi, Y. *et al.* (2024) *TP / FP / FN / TN: Learn about true positives, false positives, false negatives, and true negatives to evaluate ML model performance · Testing with Kolena*. Available at: <https://docs.kolena.com/metrics/tp-fp-fn-tn/?form=MG0AV3> (Accessed: 23 December 2024).

Chollet, F. (2017) ‘Xception: Deep learning with depthwise separable convolutions’, in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Institute of Electrical and Electronics Engineers Inc., pp. 1800–1807. Available at: <https://doi.org/10.1109/CVPR.2017.195>.

Cobilla, R. *et al.* (2023) ‘Classification of the Type of Brain Tumor in MRI Using Xception Model’, in *2023 International Conference on Electronics, Information, and*

Communication, ICEIC 2023. Institute of Electrical and Electronics Engineers Inc. Available at: <https://doi.org/10.1109/ICEIC57457.2023.10049979>.

Devi, B.S., Chatrapati, K.S. and Sandhya, N. (2024) ‘Enhanced Sugarcane Disease Detection Using DenseNet201 and DenseNet264 with Transfer Learning and Fine-tuning’, *Frontiers in Health Informatics*, 13(3). Available at: www.healthinformaticsjournal.com.

Elvionita, C., Sari, I.P. and Nuryastuti, T. (2023) ‘Evaluation the Rationality of Clinical Outcomes of Antibiotic Use and Patterns of Bacterial Resistance to Antibiotics in Children with Pneumonia’, *Majalah Farmaseutik*, 19(1). Available at: <https://doi.org/10.22146/farmaseutik.v19i1.76103>.

Farida, Y., Trisna, A. and Nur, D. (2017) *Study of Antibiotic Use on Pneumonia Patient in Surakarta Referral Hospital Studi Penggunaan Antibiotik Pada Pasien Pneumonia di Rumah Sakit Rujukan Daerah Surakarta*, *Journal of Pharmaceutical Science and Clinical Research*.

Gonzalez, R.C.. and Woods, R.E.. (2002) *Digital image processing*. Prentice Hall.

Gonzalez, R.C.. and Woods, R.E.. (2008) *Digital image processing*. Prentice Hall.

Hafemann, L.G., Sabourin, R. and Oliveira, L.S. (2017) ‘Learning features for offline handwritten signature verification using deep convolutional neural networks’, *Pattern Recognition*, 70, pp. 163–176. Available at: <https://doi.org/https://doi.org/10.1016/j.patcog.2017.05.012>.

hart, gordon *et al.* (2024) *Confusion Matrix: How to create and interpret confusion matrices to evaluate ML model performance · Testing with Kolena*. Available at: <https://docs.kolena.com/metrics/confusion-matrix/> (Accessed: 5 January 2025).

Hasan, M.A. *et al.* (2024) ‘Prediction of fetal brain gestational age using multihead attention with Xception’, *Computers in Biology and Medicine*, 182. Available at: <https://doi.org/10.1016/j.compbio.2024.109155>.

He, K. *et al.* (2015) ‘Deep Residual Learning for Image Recognition’. Available at: <http://arxiv.org/abs/1512.03385>.

Heaton, J. (2018) ‘Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning’, *Genetic Programming and Evolvable Machines*, 19(1–2), pp. 305–307. Available at: <https://doi.org/10.1007/s10710-017-9314-z>.

Hosna, A. *et al.* (2022) ‘Transfer learning: a friendly introduction’, *Journal of Big Data*, 9(1). Available at: <https://doi.org/10.1186/s40537-022-00652-w>.

Huang, G. *et al.* (2016) ‘Densely Connected Convolutional Networks’. Available at: <http://arxiv.org/abs/1608.06993>.

Huy, V.T.Q. and Lin, C.-M. (2023) ‘An Improved Densenet Deep Neural Network Model for Tuberculosis Detection Using Chest X-Ray Images’, *IEEE Access*, 11, pp. 42839–42849. Available at: <https://doi.org/10.1109/ACCESS.2023.3270774>.

Iswari, R.A. (2021) ‘KLASIFIKASI PNEUMONIA PADA GAMBAR X-RAY PARU-PARU MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK’.

Kong, L. and Cheng, J. (2022) ‘Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion’, *Biomedical Signal Processing and Control*, 77. Available at: <https://doi.org/10.1016/j.bspc.2022.103772>.

Koonsanit, K. *et al.* (2017) ‘Image enhancement on digital x-ray images using N-CLAHE’, in *2017 10th Biomedical Engineering International Conference (BMEiCON)*, pp. 1–4. Available at: <https://doi.org/10.1109/BMEiCON.2017.8229130>.

Manafe, D. (2020) *Covid-19 Masih Panjang, Indonesia Kekurangan 1.294 Dokter Paru*. Available at: <https://www.beritasatu.com/news/674573/covid19-masih-panjang-indonesia-kekurangan-1294-dokter-paru> (Accessed: 1 January 2025).

Marchand, W.R. (2022) *The Rising Popularity of Python* - <https://pythoncircle.com>. Available at: <https://pythoncircle.com/post/763/the-rising-popularity-of-python/?form=MG0AV3> (Accessed: 25 December 2024).

Nurin, F. (2024) *9 Faktor Risiko Pneumonia yang Perlu Anda Waspadai*. Available at: <https://hellosehat.com/pernapasan/pneumonia/faktor-risiko-pneumonia/?form=MG0AV3> (Accessed: 21 December 2024).

Pan, C. *et al.* (2023) ‘FemurTumorNet: Bone tumor classification in the proximal femur using DenseNet model based on radiographs’, *Journal of Bone Oncology*, 42. Available at: <https://doi.org/10.1016/j.jbo.2023.100504>.

Panthakkan, A. *et al.* (2022) ‘Concatenated Xception-ResNet50 — A novel hybrid approach for accurate skin cancer prediction’, *Computers in Biology and Medicine*, 150. Available at: <https://doi.org/10.1016/j.combiomed.2022.106170>.

Patterson, J. and Gibson, A. (2017) *Deep Learning: A Practitioner’s Approach*. O’Reilly Media. Available at: <https://books.google.co.id/books?id=qrcuDwAAQBAJ>.

Pittara (2022) *Pencegahan Pneumonia - Alodokter*. Available at: <https://www.alodokter.com/pneumonia/pencegahan?form=MG0AV3> (Accessed: 21 December 2024).

Pranita, E. and Sumartiningtyas, H.K.N. (2020) *Indonesia Peringkat Ketujuh Kematian Balita akibat Pneumonia di Dunia, Apa Sebabnya? Halaman all - Kompas.com*. Available at: <https://www.kompas.com/sains/read/2020/11/09/173300723/indonesia->

peringkat-ketujuh-kematian-balita-akibat-pneumonia-di-dunia-apa?page=all
(Accessed: 1 January 2025).

Rahman, T., Baras, A.S. and Chellappa, R. (2025) 'Evaluation of a Task-Specific Self-Supervised Learning Framework in Digital Pathology Relative to Transfer Learning Approaches and Existing Foundation Models', *Modern Pathology*, 38(1). Available at: <https://doi.org/10.1016/j.modpat.2024.100636>.

Ramadhan, A., Bustomi, M.A. and Faridawati (2014) 'KLASIFIKASI CITRA RONTGEN PARU PARU DENGAN EKSTRAKSI FITUR HISTOGRAM DAN METODE NAIVE BAYES'.

Rindrasari, R. (2018) *KLASIFIKASI KELAS RISIKO PASIEN PNEUMONIA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE-GENETIC ALGORITHM (SVM-GA) HYBRID*.

Rossum, G. van (1996) *Foreword for 'Programming Python' (1st ed.)* | *Python.org*. Available at: <https://www.python.org/doc/essays/foreword/> (Accessed: 25 December 2024).

Sadik, R. *et al.* (2023) 'An in-depth analysis of Convolutional Neural Network architectures with transfer learning for skin disease diagnosis', *Healthcare Analytics*, 3. Available at: <https://doi.org/10.1016/j.health.2023.100143>.

Saleh, A.W. *et al.* (2023) 'An Alzheimer's disease classification model using transfer learning Densenet with embedded healthcare decision support system', *Decision Analytics Journal*, 9. Available at: <https://doi.org/10.1016/j.dajour.2023.100348>.

Sartiwi, W. *et al.* (no date) 'Jurnal Abdimas Saintika LATIHAN BATUK EFEKTIF PADA PASIEN PNEUMONIA DI RSUD SAWAHLUNTO'. Available at: <https://jurnal.syedzasaintika.ac.id>.

Scholar, M.T. and Bagane, M. (2024) 'Google Colab: The Free Cloud Platform Powering Machine Learning', 12, p. 1.

Sewak, M., Karim, Md.R. and Pujari, P. (2018) *Practical Convolutional Neural Networks: Implement advanced deep learning models using Python*. Packt Publishing.

Sharma, A. *et al.* (2020) 'Python: The Programming Language of Future'.

Soewu, T. *et al.* (2022) 'Lung Cancer Detection using Image Processing', in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 1206–1211. Available at: <https://doi.org/10.1109/IC3I56241.2022.10072589>.

Sulistiyani Ratna, S., Setyawan Arianto, F. and Komarudin, M. (2016) *PENGOLAHAN CITRA; DASAR DAN CONTOH PENERAPANNYA*.

Tang, X. and Rashid Sheykhahmad, F. (2024) ‘Boosted dipper throated optimization algorithm-based Xception neural network for skin cancer diagnosis: An optimal approach’, *Heliyon*, 10(5). Available at: <https://doi.org/10.1016/j.heliyon.2024.e26415>.

TensorFlow (no date). Available at: <https://www.tensorflow.org/> (Accessed: 25 December 2024).

Tharwat, A. (2018) ‘Classification assessment methods’, *Applied Computing and Informatics*, 17(1), pp. 168–192. Available at: <https://doi.org/10.1016/j.aci.2018.08.003>.

Upasana, C., Tewari, A.S. and Singh, J.P. (2022) ‘An Attention-based Pneumothorax Classification using Modified Xception Model’, in *Procedia Computer Science*. Elsevier B.V., pp. 74–82. Available at: <https://doi.org/10.1016/j.procs.2022.12.403>.

Virnodkar, S.S. *et al.* (2022) ‘CaneSat dataset to leverage convolutional neural networks for sugarcane classification from Sentinel-2’, *Journal of King Saud University - Computer and Information Sciences*, 34(6), pp. 3343–3355. Available at: <https://doi.org/10.1016/j.jksuci.2020.09.005>.

Wikanargo, M.A. and Thenata, A.P. (2018) ‘IMAGE SEGMENTATION OF CHEST X-RAYS FOR ABNORMALITY PATTERN RECOGNATION IN LUNGS USING FUZZY C-MEANS METHOD’, *Jurnal Terapan Teknologi Informasi*, 2(2), pp. 101–111. Available at: <https://doi.org/10.21460/jutei.2018.22.98>.

Zeng, T. *et al.* (2024) ‘AI diagnostics in bone oncology for predicting bone metastasis in lung cancer patients using DenseNet-264 deep learning model and radiomics’, *Journal of Bone Oncology*, 48. Available at: <https://doi.org/10.1016/j.jbo.2024.100640>.

Zhang, A. *et al.* (2021) *Dive into Deep Learning — Dive into Deep Learning 1.0.3 documentation*. Available at: <https://d2l.ai/> (Accessed: 12 December 2024).