

MODEL MULTICLASS CLASSIFICATION UNTUK PENYAKIT BERDASARKAN CITRA CHEST X-RAY PARU-PARU DENGAN ENSEMBLE LEARNING

by Lucky Indra Kesuma

Submission date: 03-Aug-2023 02:33PM (UTC+0700)

Submission ID: 2140749514

File name: Laporan_Disertasi_Lucky_Indra_Kesuma.pdf (4.24M)

Word count: 22386

Character count: 143700

DISERTASI

**MODEL *MULTICLASS CLASSIFICATION* UNTUK
PENYAKIT BERDASARKAN CITRA *CHEST X-RAY*
PARU-PARU DENGAN *ENSEMBLE LEARNING***

**Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar
Doktor Dalam Bidang Ilmu Teknik Informatika**



LUCKY INDRA KESUMA

03013682025015

PROGRAM STUDI ILMU TEKNIK PROGRAM DOKTOR

FAKULTAS TEKNIK

UNIVERSITAS SRIWIJAYA

2023

HALAMAN PENGESAHAN

**MODEL *MULTICLASS CLASSIFICATION* UNTUK PENYAKIT
BERDASARKAN CITRA *CHEST X-RAY* PARU-PARU
DENGAN ENSEMBLE LEARNING**

DISERTASI

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Doktor Dalam Bidang Ilmu Teknik Informatika

Oleh:

LUCKY INDRA KESUMA
03013682025015

Promotor,

Indralaya, Juli 2023
KoPromotor,

Dr. Ermatita, M.Kom.
NIP. 196709132006042001

Prof. Dr. Erwin, S.Si., M.Si.
NIP. 197101291994121001

Mengetahui,

Dekan Fakultas Teknik,

Koordinator Program Studi,

Prof. Dr. Ir. H. Joni Arliansyah, M.T.
NIP. 196706151995121002

Prof. Dr. Ir. Nukman, M.T.
NIP. 195903211987031001

HALAMAN PERSETUJUAN

Karya tulis ilmiah berupa Disertasi ini dengan judul “Model *Multiclass Classification* untuk Penyakit Berdasarkan Citra *Chest X-Ray* Paru-Paru dengan *Ensemble Learning*” telah dipertahankan dihadapan Tim Penguji Karya Tulis Ilmiah Program Studi Ilmu Teknik Program Doktor Fakultas Teknik Universitas Sriwijaya pada tanggal Juli 2023.

Palembang, Juli 2023

Tim Penguji Karya Tulis Ilmiah Berupa Disertasi

Ketua:

Prof. Dr. Ir. Nukman, M.T. ()
NIP. 195903211987031001

Anggota:

1. Dr. Yusuf Hartono, M.Sc. ()
NIP. 196411161990031002

2. Dr. Abdiansah, S.Kom., M.Sc. ()
NIP. 198410012009121005

Mengetahui,

Dekan Fakultas Teknik,

Koordinator Program Studi,

Prof. Dr. Ir. H. Joni Arliansyah, M.T.
NIP. 196706151995121002

Prof. Dr. Ir. Nukman, M.T.
NIP. 195903211987031001

RINGKASAN

Model *Multiclass Classification* untuk Penyakit Berdasarkan Citra *Chest X-Ray* Paru-Paru dengan *Ensemble Learning*

Citra *Chest X-ray* (CXR) dapat mendiagnosis penyakit paru-paru. Namun, diagnosis memerlukan waktu dan ketelitian yang tinggi, sehingga diperlukan sistem yang otomatis. Pada prosesnya, citra CXR terlebih dahulu ditingkatkan kualitas citranya menggunakan Morfologi *Opening* dan *Median filter*, dilanjutkan dengan augmentasi data menggunakan rotasi dan *flipping*. Segmentasi citra CXR menggunakan *U-Net Batch Normalization* dilakukan dengan memisahkan objek paru-paru dengan *background*. Hasil proses segmentasi dilakukan menggunakan metode *ensemble learning* terhadap kinerja arsitektur *ResNet*, *EfficientNet*, dan *Inception-v3*. Hasil penelitian menunjukkan keberhasilan dalam meningkatkan kualitas citra CXR menggunakan Morfologi *Opening* dan *Median Filter* dengan rata-rata nilai PSNR sebesar 39,307, MSE sebesar 22,469, dan SSIM sebesar 0,952. Model segmentasi *U-Net Batch Normalization* mencapai akurasi di atas 93% dan nilai *loss* mendekati 1%, menunjukkan kemampuan yang sangat baik dalam mendeteksi paru-paru pada citra CXR. Penerapan *Ensemble Learning* dari *ResNet*, *EfficientNet*, dan *Inception-v3* (ELREI) pada tahap klasifikasi menghasilkan peningkatan yang signifikan dibandingkan dengan metode klasifikasi tunggal. Peningkatan nilai akurasi sebesar 11% (*ResNet*), 3% (*EfficientNet*), dan 1% (*Inception-v3*). Peningkatan nilai presisi adalah 10,5% (*ResNet*), 1% (*EfficientNet*), dan 3% (*Inception-v3*). Sedangkan peningkatan nilai *recall* adalah 10,75% (*ResNet*), 1% (*EfficientNet*), dan 3,25% (*Inception-v3*). Sementara peningkatan nilai *F1-Score* adalah 10,25% (*ResNet*), 3,25% (*EfficientNet*), dan 1% (*Inception-v3*). Rata-rata akurasi, presisi, *recall*, dan *F1-Score* yang dihasilkan oleh metode ELREI adalah 99%, 98,75%, 98,75%, dan 99%. Secara keseluruhan, metode ELREI terbukti kuat dan sangat baik untuk mengklasifikasikan penyakit paru-paru berdasarkan citra CXR, dengan mengelompokkan ke dalam empat kelas: *COVID-19*, normal, *lung opacity*, dan *pneumonia*.

Kata Kunci: Citra *Chest X-ray*, Peningkatan Kualitas Citra, Segmentasi Citra, Klasifikasi Citra, *Ensemble Learning*.

SUMMARY

Multiclass Classification Model for Lung Disease Based on Chest X-Ray Image with Ensemble Learning

Chest X-ray (CXR) images can diagnose lung diseases. However, diagnosis requires time and high accuracy, so an automated system is needed. In the process, the CXR image is first enhanced for image quality using Morphology Opening and Median filter, followed by data augmentation using rotation and flipping. CXR image segmentation using U-Net Batch Normalization is done by separating the lung object from the background. The results of the segmentation process are carried out using the ensemble learning method on the performance of ResNet, EfficientNet, and Inception-v3 architectures. The results successfully improved the quality of CXR images using the morphology opening and Median filter, with an average PSNR value of 39.307, an MSE of 22.469, and an SSIM of 0.952. The U-Net Batch Normalization segmentation model achieved an accuracy of over 93% and a loss value close to 1%, indicating an excellent ability to detect lungs in CXR images. The application of Ensemble Learning from ResNet, EfficientNet, and Inception-v3 (ELREI) in the classification stage resulted in significant improvement compared to the single classification method. The increase in accuracy value is 11% (ResNet), 3% (EfficientNet), and 1% (Inception-v3). The increase in precision value is 10.5% (ResNet), 1% (EfficientNet), and 3% (Inception-v3). The increase in recall value is 10.75% (ResNet), 1% (EfficientNet), and 3.25% (Inception-v3). The increase in F1-Score value is 10.25% (ResNet), 3.25% (EfficientNet), and 1% (Inception-v3). The average accuracy, precision, recall, and F1-Score generated by the ELREI method are 99%, 98.75%, 98.75%, and 99%. Overall, the ELREI method proved to be robust and excellent for classifying lung diseases based on CXR images by categorizing them into four classes: COVID-19, normal, lung opacity, and pneumonia.

Keywords: Chest X-ray Image, Image Quality Enhancement, Image Segmentation, Image Classification, Ensemble Learning.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan kehadiran Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Laporan Disertasi dengan judul “**Model Multiclass Classification untuk Penyakit Berdasarkan Citra Chest X-ray Paru-Paru Dengan Ensemble Learning**”.

Ucapan terima kasih penulis sampaikan pada pihak-pihak yang telah membantu penyusunan laporan ini, terkhusus kepada:

1. Kedua orang tua tercinta.
2. Istri dan anak-anak tersayang.
3. Kakak ayuk tersayang.
4. Prof. Dr. Ir. H. Anis Saggaff, MSCE, IPU, Aseng. Eng. Selaku Rektor Universitas Sriwijaya.
5. Prof. Dr. Ir. H. Joni Arliansyah, M.T. selaku Dekan Fakultas Teknik Universitas Sriwijaya.
6. Prof. Dr. Ir. Nukman, M.T., selaku Koordinator Program Studi Ilmu Teknik Program Doktor, Fakultas Teknik, Universitas Sriwijaya.
7. Dr. Ermatita, M.Kom., selaku Promotor yang telah memberikan bimbingan dan arahan dalam penyelesaian disertasi ini.
8. Prof. Dr. Erwin, S.Si., M.Si., selaku Ko-Promotor yang telah memberikan arahan dalam penyelesaian disertasi ini.
9. Prof. Dr. Ir. Nukman, M.T., Dr. Yusuf Hartono, M.Sc, dan Dr. Abdiansah, S.Kom., M.Sc. selaku Tim Penguji yang telah memberikan saran untuk peningkatan kualitas disertasi ini.
10. Rektor Universitas Sjakhyakirti Palembang, Prof. Dr. Ir. Agoes Thony Ak, M.Si. atas dukungannya.
11. Sahabat dan semua pihak bagian dari *support system* penelitian disertasi ini yang tidak dapat disebutkan satu per satu.

Semoga penelitian disertasi ini menghasilkan ilmu bermanfaat yang diberkahi Allah SWT dan menjadi amal jariah bagi kita semua. Aamiin.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Palembang, Juli 2023

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
RINGKASAN	iv
SUMMARY	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	6
1.3. Tujuan	7
1.4. Manfaat	7
1.5. Batasan Masalah.....	7
1.6. Ruang Lingkup Penelitian.....	7
1.7. Sistematika Penulisan.....	9
BAB II TINJAUAN PUSTAKA.....	10
2.1. <i>Roadmap</i> Penelitian	10
2.2. Penyakit Menular	10
2.3. Citra Digital.....	11
2.4. Citra <i>Chest X-ray</i>	12
2.5. Perbaikan Kualitas Citra	12
2.5.1. Morfologi <i>Opening</i>	13
2.5.2. <i>Median Filter</i>	13
2.5.3. <i>Mean Square Error</i>	14
2.5.4. <i>Peak Signal-to-Noise Ratio</i>	14
2.5.5. <i>Structural Similarity Index Metrics</i>	15
2.6. Augmentasi Data.....	16

2.7. Segmentasi Citra	16
2.8. Klasifikasi Citra	16
2.9. <i>Convolutional Neural Network (CNN)</i>	17
2.9.1. <i>Convolution Layer</i>	17
2.9.2. <i>Batch Normalization</i>	19
2.9.3. Fungsi Aktivasi.....	20
2.9.4. <i>Pooling Layer</i>	21
2.9.5. <i>Transposed Convolution</i>	23
2.9.6. <i>Concatenate Layer</i>	24
2.9.7. <i>Depthwise Separable Convolution</i>	24
2.9.8. <i>Dense Layer</i>	25
2.9.9. <i>Dropout</i>	26
2.9.10. <i>Loss Function</i>	26
2.10. <i>U-Net</i>	27
2.11. <i>Residual Network</i>	28
2.12. <i>EfficientNet</i>	29
2.13. <i>Inception-v3</i>	30
2.14. <i>Ensemble Learning</i>	31
2.15. <i>Confusion Matrix</i>	32
2.16. <i>State of the Art</i>	34
BAB III METODOLOGI PENELITIAN	41
3.1. Tempat Penelitian.....	41
3.2. Waktu Penelitian	41
3.3. Alat.....	41
3.4. Objek Penelitian.....	41
3.5. Pengumpulan Data	41
3.6. Kerangka Kerja Penelitian	42
3.5.1. Peningkatan Kualitas Citra	43
3.5.2. Augmentasi Data	44
3.5.3. Segmentasi Citra.....	45
3.5.4. Klasifikasi Citra.....	47

3.5.4.1. <i>Training</i> Menggunakan Arsitektur <i>ResNet</i>	48
3.5.4.2. <i>Training</i> Menggunakan Arsitektur <i>EfficientNet</i>	50
3.5.4.3. <i>Training</i> Menggunakan Arsitektur <i>Inception-v3</i>	52
3.5.4.4. <i>Training</i> Menggunakan <i>Ensemble Learning</i>	54
3.5.4.5. <i>Testing</i> pada Klasifikasi Citra	55
3.5.4.6. Evaluasi Kinerja Klasifikasi Citra.....	56
BAB IV HASIL DAN PEMBAHASAN	57
4.1. Deskripsi Data	57
4.2. Peningkatan Kualitas Citra CXR	57
4.3. Augmentasi Data pada Citra CXR	61
4.4. Segmentasi Citra CXR	62
4.5. Klasifikasi Citra CXR	64
4.5.1. <i>Training</i> Klasifikasi Citra CXR Menggunakan <i>ResNet</i>	64
4.5.2. <i>Training</i> Klasifikasi Citra CXR Menggunakan <i>EfficientNet</i> ..	66
4.5.3. <i>Training</i> Klasifikasi Citra CXR Menggunakan <i>Inception-v3</i> ..	69
4.5.4. <i>Training</i> Klasifikasi Citra CXR Menggunakan ELREI	71
4.5.5. <i>Testing</i> pada Klasifikasi Citra CXR	73
4.5.6. Evaluasi Kinerja pada Klasifikasi Citra CXR	74
4.6. Analisis dan Interpretasi Hasil	75
BAB V PENUTUP.....	77
5.1. Kesimpulan	77
5.2. Saran.....	78
DAFTAR PUSTAKA	79
LAMPIRAN.....	89

DAFTAR TABEL

Tabel 2.1. <i>Confusion matrix</i> untuk segmentasi paru-paru pada citra CXR	32
Tabel 2.2. <i>Confusion matrix</i> multikelas untuk klasifikasi penyakit <i>COVID-19</i> ...	32
Tabel 2.3. Kategori nilai kinerja arsitektur	34
Tabel 2.4. Beberapa penelitian terdahulu yang telah melakukan peningkatan kualitas citra.....	36
Tabel 2.5. Beberapa penelitian terdahulu yang telah melakukan segmentasi dan klasifikasi citra.....	37
Tabel 4.1. Perbandingan hasil peningkatan citra CXR	58
Tabel 4.2. Perbandingan Hasil Evaluasi Kinerja pada Metode <i>Median Filter</i>	60
Tabel 4.3. Perbandingan Hasil Evaluasi Penelitian ini dengan Penelitian Lain ...	60
Tabel 4.4. Perbandingan Hasil Evaluasi Kinerja Menggunakan Metode ELREI .	74
Tabel 4.5. Perbandingan Hasil Evaluasi Penelitian ini dengan Penelitian Lain ...	75

DAFTAR GAMBAR

Gambar 2.1. <i>Roadmap</i> Penelitian Tahun 2019-2023	10
Gambar 2.2. Proses operasi konvolusi	18
Gambar 2.3. Grafik fungsi aktivasi <i>ReLU</i> dan <i>sigmoid</i>	21
Gambar 2.4. Ilustrasi proses <i>max pooling</i> 2×2	22
Gambar 2.5. Ilustrasi proses <i>average pooling</i> 2×2 dan <i>stride</i> 2	22
Gambar 2.6. Ilustrasi <i>transposed convolution</i> 2×2 dan <i>stride</i> 2	23
Gambar 2.7. Proses <i>depthwise separabel convolution</i> (a) <i>depthwise convolution</i> (b) <i>pointwise convolution</i>	24
Gambar 2.8. Ilustasi arsitektur <i>U-Net</i>	27
Gambar 2.9. Ilustrasi arsitektur <i>ResNet</i>	29
Gambar 2.10. Ilustrasi arsitektur <i>EfficientNet</i>	30
Gambar 2.11. Ilustrasi arsitektur <i>Inception-v3</i>	31
Gambar 3.1. Kerangka kerja penelitian	42
Gambar 3.2. Tahapan Peningkatan Kualitas Citra	43
Gambar 3.3. Tahapan Augmentasi Data	44
Gambar 3.4. Tahapan Segmentasi Citra	45
Gambar 3.5. Tahapan Klasifikasi Citra	48
Gambar 4.1. Contoh Citra CXR Paru-paru (a) <i>COVID-19</i> (b) Normal (c) <i>Lung</i> <i>Opacity</i> (d) <i>Pneumonia</i>	57
Gambar 4.2. Perbandingan Grafik Nilai Evaluasi Peningkatan Kualitas Citra CXR (a) <i>COVID-19</i> (b) Normal (c) <i>Lung Opacity</i> (d) <i>Pneumonia</i>	59
Gambar 4.3. Hasil Augmentasi Data Menggunakan Transformasi Rotasi	61
Gambar 4.4. Arsitektur <i>U-Net</i> pada Citra CXR	62
Gambar 4.5. Grafik <i>Training</i> Segmentasi Citra CXR (a) Akurasi (b) <i>Loss</i>	63
Gambar 4.6. Hasil Segmentasi Citra CXR Paru-paru Menggunakan <i>U-Net Batch</i> <i>Normalization</i>	64
Gambar 4.7. Arsitektur <i>ResNet</i> pada Klasifikasi Citra CXR	65

Gambar 4.8. Grafik <i>Training</i> Klasifikasi Citra CXR Menggunakan Arsitektur <i>ResNet</i> (a) Akurasi (b) <i>Loss</i>	66
Gambar 4.9. Arsitektur <i>EfficientNet</i> pada Klasifikasi Citra CXR	67
Gambar 4.10. Grafik <i>Training</i> Klasifikasi Citra CXR Menggunakan Arsitektur <i>EfficientNet</i> (a) Akurasi (b) <i>Loss</i>	68
Gambar 4.11. Arsitektur <i>Inception-v3</i> pada Klasifikasi Citra CXR	69
Gambar 4.12. Grafik <i>Training</i> Klasifikasi Citra CXR Menggunakan Arsitektur <i>Inception-v3</i> (a) Akurasi (b) <i>Loss</i>	70
Gambar 4.13. Proses ELREI pada Klasifikasi Citra CXR	71
Gambar 4.14. Grafik <i>Training</i> Klasifikasi Citra CXR Menggunakan Metode ELREI (a) Akurasi (b) <i>Loss</i>	72
Gambar 4.15. Perbandingan Evaluasi Kinerja pada Proses Klasifikasi Citra CXR Menggunakan <i>ResNet</i> , <i>EfficientNet</i> , <i>Inception-v3</i> , dan ELREI.....	73

DAFTAR LAMPIRAN

Lampiran 1. Bukti Accepted ICIMCIS 89
Lampiran 2. Bukti Accepted Jurnal IJES..... 90

BAB I

PENDAHULUAN

1.1. Latar Belakang

Penyakit *COVID-19* adalah salah satu penyakit menular pada saluran pernapasan yang disebabkan oleh virus bernama *Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2)*, dimana virus ini pertama kali dilaporkan di kota Wuhan, China pada bulan Desember 2019 [1]. Selama ini, untuk mendiagnosa penyakit *COVID-19* dilakukan dengan menggunakan tes *Reverse Transcription Polymerase Chain Reaction (RT-PCR)* [2]. Namun, tes tersebut memiliki kelemahan yaitu rumah sakit yang memiliki fasilitas RT-PCR terbatas, dan biaya yang dikeluarkan untuk tes tersebut cukup mahal [3]. Hal tersebut membuat para peneliti menemukan alternatif lain yang lebih baru dan mudah untuk mendiagnosa penyakit *COVID-19*, salah satunya yaitu melalui *Chest X-Ray (CXR)*. Menurut Song et al [4] gambar hasil pemeriksaan CXR dinilai cukup efektif dalam mendiagnosa penyakit *COVID-19*, karena melalui hasil pemeriksaan tersebut tenaga medis dapat melihat kelainan yang ada pada paru-paru pasien. Dalam melakukan pembacaan hasil citra CXR, diperlukan tingkat ketelitian yang tinggi, dikarenakan banyak jaringan yang bertumpuk pada citra tersebut [5]. Untuk membantu dokter dan ahli radiologi agar lebih jelas dalam membaca hasil citra CXR, maka diperlukanlah suatu peningkatan kualitas citra agar mendapatkan hasil kualitas citra yang lebih baik lagi dan lebih jelas dibaca sehingga menghasilkan diagnosis yang akurat [5].

Peningkatan kualitas citra merupakan proses mengubah citra asli menjadi citra baru sesuai dengan yang dibutuhkan untuk menghasilkan citra yang lebih jelas dan tajam [6]. Metode peningkatan kualitas citra yang berkembang saat ini ialah perbaikan kontur dan pengurangan *noise*. Salah satu metode yang dapat digunakan untuk perbaikan kontur adalah Morfologi *Opening* [7]. Morfologi *Opening* bekerja dengan menghilangkan objek tipis dan halus agar gambar menjadi lebih halus [8]. Sementara metode yang paling sering digunakan dalam meningkatkan kualitas citra

untuk mengurangi *noise* adalah metode *Filtering*. Metode *Filtering* adalah teknik *filtering* yang digunakan untuk menghilangkan atau mengurangi *noise*. Salah satu metode *filtering* yang sering digunakan saat ini adalah *Median Filter* [9]. *Median Filter* memiliki kemampuan untuk meredam *noise* dengan baik [10]. Selain peningkatan kualitas citra, umumnya citra CXR juga terkendala oleh data yang terbatas. Untuk mengatasi masalah ini, augmentasi data dapat diterapkan untuk meningkatkan jumlah dan keragaman data [11], [12]. Augmentasi data adalah suatu teknik pengolahan data citra yang mengubah citra menjadi data yang baru tanpa menghilangkan ataupun mengubah inti dari citra [13]. Augmentasi akan memperbanyak data dari kelas yang kurang terwakili sehingga membuat dataset menjadi seimbang kelasnya [11]. Augmentasi data yang sering digunakan yaitu rotasi dan *flipping*. Augmentasi rotasi dilakukan dengan memutar citra ke kanan atau ke kiri sesuai dengan derajat yang dipilih, sedangkan *flipping* dilakukan dengan membalikkan citra ke arah vertikal atau horizontal [12].

Penentuan hasil diagnosa melalui CXR membutuhkan waktu yang cukup lama dan ketelitian yang tinggi, sehingga diperlukan suatu sistem otomatis untuk mempermudah tenaga medis dalam mendiagnosa penyakit *COVID-19* secara cepat dan akurat [14].⁶ Segmentasi merupakan salah satu sistem otomatis dalam pengolahan citra yang dilakukan dengan membagi citra menjadi beberapa wilayah atau bagian sesuai dengan karakteristiknya [15].⁴ Proses segmentasi pada citra CXR dilakukan dengan mengambil bagian paru-parunya saja, sedangkan untuk bagian lainnya seperti tulang dada, jantung, dan yang lainnya akan dijadikan sebagai *background* [16].

Saat ini berkembang metode yang handal dalam segmentasi citra yaitu *Convolutional Neural Network* (CNN) [13]. Kemampuan utama CNN terletak pada arsitekturnya, dimana arsitektur yang sering digunakan dalam segmentasi citra adalah *U-Net* [17]. *U-Net* sangat efektif digunakan untuk segmentasi citra berbasis piksel. Kelemahan arsitektur ini selama proses pelatihan sering terjadi perubahan input untuk setiap lapisannya [18]. Input-input setiap lapisan bergantung pada jumlah parameter di lapisan sebelumnya, sehingga membuat terjadinya perubahan distribusi fitur input [19]. Hal ini menyebabkan penurunan kecepatan pelatihan dan

kinerja [18], [19]. Untuk mengatasi masalah perubahan distribusi fitur input tersebut dapat digunakan metode *Batch Normalization* (BN) [19], [20]. BN merupakan metode normalisasi yang banyak digunakan untuk segmentasi biomedis [19]. BN akan menormalkan peta fitur disemua lapisan arsitektur dengan melakukan skala ulang dan menerjemahkan ulang peta fitur yang dinormalisasi[20]. Modifikasi *U-Net* dengan menambahkan lapisan BN telah dilakukan oleh beberapa penelitian. Salah satunya Aboelenen et al [21] mengusulkan arsitektur *U-Net* dengan menambahkan lapisan *Batch Normalization* pada segmentasi tumor otak menghasilkan nilai rata-rata *F-1 Score* 86%. Penelitian lainnya oleh Liu et al [22] mengusulkan arsitektur DCNN *U-Net* dengan penambahan *Batch Normalization* menghasilkan nilai *F-1 Score* 83% dan 84%, Akurasi 92% dan 93%. Dari kedua penelitian diatas terlihat bahwa dengan menggabungkan arsitektur *U-Net* dan BN pada proses segmentasi citra dapat menjadi salah satu alternatif untuk medeteksi kelainan paru-paru pada CXR.

Selain segmentasi, metode CNN paling banyak dimanfaatkan untuk klasifikasi citra. Dalam CNN terdapat banyak arsitektur yang dapat diterapkan untuk pengenalan pola suatu citra. Performa kinerja arsitektur pada CNN dapat ditingkatkan dengan menambahkan lapisan [23]. Namun penambahan lapisan pada arsitektur CNN dapat mengakibatkan kompleksitas lapisan semakin meningkat dan *gradient vanishing* [24]. Untuk mengatasi tersebut dapat digunakan arsitektur *ResNet* [23], [24]. *ResNet* menggunakan *skip connection* di lapisannya untuk menangani masalah *gradient vanishing*. *Skip Connection* dapat mempercepat konvergensi jaringan di *ResNet*. Keunggulan *ResNet* adalah performanya tidak menurun walaupun arsitekturnya semakin dalam. Selain itu, *ResNet* juga memiliki kompleksitas jaringan yang lebih rendah dibandingkan dengan arsitektur CNN lainnya [25]. Beberapa penelitian telah menunjukkan keberhasilannya dalam mengklasifikasikan *COVID-19* menggunakan citra CXR [26], [27]. Penelitian oleh Keles at al. [26] menggunakan *ResNet* untuk mengklasifikasikan *COVID-19*, yang menghasilkan nilai akurasi, *recall*, dan *F1-score* di atas 90%. Sayangnya, penelitian ini tidak menghitung nilai presisinya. Penelitian oleh Karhan and Akal [27] menggunakan *ResNet-50* untuk mengklasifikasikan *COVID-19*. Hasil penelitian

menunjukkan nilai akurasi diatas 90%. Namun penelitian ini tidak menghitung *recall*, presisi, dan *F1-score*. Kelemahan arsitektur *ResNet* adalah sulitnya mendeteksi kesalahan karena lapisan yang dalam [28].

Arsitektur CNN lain yang dapat digunakan untuk klasifikasi citra adalah *EfficientNet*. Arsitektur *EfficientNet* memiliki keunggulan karena mampu menskalakan dimensi seperti kedalaman, lebar, dan resolusi dalam lapisan secara seragam. Penskalaan di *EfficientNet* dilakukan dengan menggunakan kombinasi koefisien yang sederhana dan efektif [29]. Selain itu, *EfficientNet* menghasilkan akurasi klasifikasi yang lebih tinggi daripada arsitektur CNN lainnya [30]. Penelitian oleh Ali et al. [28] menggunakan *EfficientNet-B4* untuk mencapai akurasi dan nilai *F1-Score* di atas 86% untuk klasifikasi citra kanker tetapi tidak mengukur *recall* dan presisi. Penelitian oleh Chen et al. [31] menerapkan *EfficientNet* untuk klasifikasi citra kolposkopi, mencapai skor akurasi lebih dari 90%, tetapi tidak mengukur kinerja lainnya. Kelemahan dari *EfficientNet* adalah memiliki kompleksitas jaringan yang besar, yang dapat menyebabkan hasil pelatihan arsitektur menjadi *overfitting*.

Salah satu arsitektur CNN lainnya yang dapat mengatasi *overfitting* adalah *Inception-v3*. Arsitektur *Inception-v3* merupakan arsitektur yang sering digunakan pada proses klasifikasi citra [32]. Arsitektur *Inception-v3* menggunakan konvolusi kecil yang menumpuk, sehingga jumlah lapisan menjadi lebih kompleks tetapi tanpa menyebabkan masalah *overfitting* [33]. Beberapa penelitian telah berhasil melakukan klasifikasi menggunakan *Inception-v3* [21], [22]. Penelitian oleh Sharma [34] berhasil menerapkan *InceptionNet* pada klasifikasi *COVID-19* dengan akurasi di atas 90% tetapi tidak mengukur performa lainnya. Penelitian oleh Mujahid et al. [35] berhasil menerapkan *InceptionNet* pada klasifikasi *COVID-19* dengan nilai akurasi, presisi, dan *recall* yang masih terbilang rendah. Kelemahan dari *Inception-v3* adalah waktu komputasi yang lama karena menggunakan fitur konvolusi yang lebih mendalam [36].

Dalam klasifikasi citra, setiap arsitektur CNN pasti memiliki kelebihan dan kekurangan. Hasil kinerja dari masing-masing metode klasifikasi dapat dimaksimalkan dengan menggunakan metode *ensemble learning*. *Ensemble*

learning adalah metode yang dapat menggabungkan hasil prediksi dari berbagai arsitektur atau metode klasifikasi tunggal menjadi model baru yang lebih akurat [37]. Keuntungan *ensemble learning* adalah dapat mengambil informasi dari beberapa metode klasifikasi dan meminimalkan tingkat kesalahan prediksi klasifikasi tunggal. Beberapa teknik pengambilan keputusan dalam *ensemble learning* sudah tepat dan efisien, seperti *bagging*, *stacking*, *boosting*, *voting*, dan *averaging* [38]. *Voting* adalah teknik pengambilan keputusan yang populer di *Ensemble* [38], [39]. Salah satu teknik *voting* yang sering digunakan adalah *weighted voting*. *Weighted voting* memberikan bobot yang berbeda untuk pengklasifikasi berdasarkan kriteria tertentu dan memilih pengklasifikasi berdasarkan bobot ini [38].

Beberapa penelitian telah menggunakan *ensemble learning* dengan teknik *weighted voting* untuk klasifikasi *COVID-19*. Penelitian oleh Das et al [40] menggunakan teknik *weighted voting* untuk menerapkan *ensemble learning* dari *DenseNet-201*, *ResNet-50*, dan *Inception-v3*, namun akurasi yang diperoleh di bawah 95% dan tidak mengukur kinerja presisi. Penelitian lainnya oleh Siddiquie et al [41] menggunakan teknik *weighted voting* untuk menerapkan *ensemble learning* kelima arsitektur CNN: *Inception*, *ResNet-v2*, *Inception-v3*, *DenseNet121*, dan *Xception*. Sayangnya, *recall* pada penelitian ini masih di bawah 90%. Penelitian oleh Zhuo et al. [42] menerapkan *ensemble learning* dengan *weighted majority vote* dari *AlexNet*, *GoogleNet*, dan *ResNet*. Akurasi yang diperoleh dari penelitian ini sangat baik yaitu 99%, namun tidak mengukur kinerja *recall*. Penelitian oleh Kundu et al. [43] mengimplementasikan *ensemble learning* dari *GoogLeNet*, *ResNet-18*, dan *DenseNet-121* dengan *weighted voting*, tetapi hasil performa untuk akurasi, *recall*, dan *F1-score* masih di bawah 90%. Penelitian-penelitian ini hanya mengumpulkan hasil akhir pada klasifikasi tunggal mana pun untuk digunakan dalam data uji. Metode *ensemble* dari penelitian tersebut tidak berjalan dalam proses *training*, sehingga tidak dapat diketahui bagaimana kinerja metode *ensemble* pada data latih dan data validasi. Metode *ensemble* dari penelitian tersebut tidak menjamin apakah bobot yang diperoleh dari hasil *ensemble* tidak *overfitting*.

Pada penelitian ini akan melakukan peningkatan kualitas citra menggunakan Morfologi *Opening* dan *Median Filter* untuk mendapatkan kualitas citra yang lebih baik. Selanjutnya dilakukan augmentasi data yang sesuai untuk memperbanyak jumlah data citra CXR. Pada tahap segmentasi penelitian ini akan menggabungkan arsitektur *U-Net* dan BN, sehingga dapat menjadi salah satu alternatif metode untuk segmentasi citra medis. Selain itu juga, dalam penelitian ini mengusulkan metode *ensemble learning* baru yang menggabungkan hasil *ResNet*, *EfficientNet*, dan *Inception-v3* yang disebut ELREI (*Ensemble Learning* dari *ResNet*, *EfficientNet*, dan *Inception-v3*) dengan *weighted voting* untuk klasifikasi penyakit paru-paru berdasarkan citra CXR. Berbeda dengan metode *ensemble* pada penelitian lain. Metode ELREI bekerja pada setiap *epoch* pada tahap *training*, bukan pada hasil akhir *training*, sehingga bobot hasil ELREI pada setiap *epoch* pada data *training* dan data validasi dapat dicek pada hasil performa untuk menangani *overfitting*. Untuk mengevaluasi kinerja dari metode yang diusulkan pada tahap segmentasi dan klasifikasi dilakukan dengan menghitung performa kinerja berdasarkan akurasi, presisi, *recall*, dan *F1-Score*.

1.2. Perumusan Masalah

1. Citra CXR pada dataset yang tersedia memiliki kualitas yang buruk karena kontras pada citra masih gelap dan terdapat *noise*, sehingga diperlukan peningkatan kualitas citra dengan menggunakan metode Morfologi *Opening* dan *Median Filter*.
2. Segmentasi menggunakan CNN dengan Arsitektur *U-Net* sudah baik tetapi didalam bidang kedokteran memerlukan tingkat keakuratan yang sangat baik sehingga perlu meningkatkan performa kinerja model dengan memodifikasi arsitektur *U-Net*.
3. Klasifikasi penyakit menggunakan klasifikasi tunggal masih memperoleh kinerja model yang belum akurat sehingga dilakukan penerapan metode *ensemble learning* pada arsitektur *ResNet*, *EfficientNet*, dan *InceptionV3* untuk memperoleh hasil evaluasi yang lebih baik.

1.3. Tujuan

1. Untuk mengetahui perbandingan citra asli dan hasil peningkatan kualitas citra menggunakan metode Morfologi *Opening* dan *Median Filter* pada citra CXR.
2. Untuk mengetahui hasil segmentasi paru-paru dan evaluasi kinerja menggunakan modifikasi arsitektur *U-Net* pada citra CXR.
3. Untuk mengetahui hasil evaluasi kinerja dari penerapan metode *ensemble learning* menggunakan arsitektur *ResNet*, *EfficientNet*, dan *InceptionV3*, pada klasifikasi penyakit *COVID-19*.

1.4. Manfaat

1. Memperoleh hasil segmentasi paru-paru pada citra CXR yang lebih akurat dengan menggunakan arsitektur *U-Net*.
2. Memperoleh model klasifikasi penyakit *COVID-19* yang lebih akurat melalui penerapan *ensemble learning* menggunakan arsitektur *ResNet*, *EfficientNet* dan *Inception-v3*.
3. Membantu tenaga medis untuk mendiagnosis penyakit *COVID-19* melalui penggunaan citra CXR paru-paru yang lebih efisien dan akurat.
4. Dapat digunakan sebagai rujukan bagi pihak yang akan melakukan penelitian mengenai klasifikasi penyakit *COVID-19*.

1.5. Batasan Masalah

Penelitian ini hanya melakukan klasifikasi penyakit menular pada citra CXR kedalam empat label, yaitu *COVID-19*, normal, *lung opacity* dan *pneumonia*. Selain itu, ukuran kinerja yang digunakan pada penelitian ini berdasarkan nilai akurasi, presisi, *recall*, dan *F-1 Score*.

1.6. Ruang Lingkup Penelitian

1. Objek Penelitian

Penelitian ini berfokus pada penggunaan teknik *ensemble learning*, yaitu menggabungkan beberapa model *deep learning* dengan arsitektur yang

berbeda, untuk meningkatkan kinerja pada klasifikasi penyakit paru-paru. Ketiga arsitektur *deep learning* yang digunakan, yaitu *ResNet*, *EfficientNet*, dan *Inception-v3*. Citra CXR digunakan sebagai *input* untuk model *ensemble* ini. Citra-citra ini kemudian diproses oleh masing-masing arsitektur (*ResNet*, *EfficientNet*, dan *Inception-v3*) untuk menghasilkan prediksi klasifikasi terkait penyakit paru-paru yang ada. Selanjutnya, *output* dari ketiga model ini digabungkan melalui teknik *ensemble learning* dengan *weighted voting* untuk menghasilkan prediksi akhir yang lebih akurat.

2. Metode Penelitian

Langkah-langkah yang akan dilakukan pada penelitian ini sebagai berikut:

- Pengumpulan Data

Data ini terdiri dari empat kelas: 3.616 citra *COVID-19*, 10.192 citra normal, 6.012 citra *lung opacity*, dan 1.345 citra *pneumonia*.

- Peningkatan Kualitas Citra

Citra CXR dilakukan perbaikan kontur menggunakan Morfologi *Opening* dan pengurangan *noise* menggunakan *Median Filter*.

- Augmentasi Data

Citra hasil peningkatan kualitas citra selanjutnya dilakukan augmentasi data dengan metode rotasi dan *flipping* untuk meningkatkan jumlah data latih.

- Segmentasi Citra

Tahap berikutnya adalah segmentasi citra, di mana dilakukan segmentasi paru-paru pada citra CXR menggunakan modifikasi arsitektur *U-Net Batch Normalization*.

- Klasifikasi Citra

Setelah itu, dilakukan klasifikasi citra dengan menggunakan tiga metode klasifikasi: *ResNet*, *EfficientNet*, dan *Inception-v3*. Selanjutnya dilakukan klasifikasi menggunakan metode *ensemble learning* dari ketiga model tersebut (ELREI) dengan *weighted voting*. Proses *weighted voting* dilakukan pada setiap *epoch* di tahap *training*, bukan pada hasil akhir setiap arsitektur.

3. Evaluasi Kinerja

Pada tahap evaluasi, dilakukan perbandingan hasil kinerja klasifikasi metode tunggal (*ResNet*, *EfficientNet*, dan *Inception-v3*) dengan metode *ensemble learning* ELREI menggunakan metrik performa model seperti akurasi, presisi, *recall*, dan *F1-Score*.

1.7. Sistematika Penulisan

Untuk memudahkan dalam menyusun disertasi dan memperjelas isi dari setiap bab yang ada pada proposal ini, maka dibuatlah sistematika penulisan yaitu:

BAB 1 Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, ruang lingkup penelitian, dan sistematika penulisan penelitian.

BAB 2 Tinjauan Pustaka

Bab ini menjelaskan tentang landasan-landasan teori yang berkaitan dengan proses pengolahan citra CXR (peningkatan kualitas citra, augmentasi data, segmentasi citra, klasifikasi citra), serta literatur terkait yang relevan dengan topik yang diteliti.

BAB 3 Metodologi Penelitian

Bab ini menjelaskan secara rinci tentang langkah-langkah penyelesaian penelitian yang terdiri dari beberapa tahapan pengolahan citra (peningkatan kualitas citra, augmentasi data, segmentasi citra, klasifikasi citra), termasuk alat dan dataset yang digunakan.

BAB 4 Hasil dan Pembahasan

Bab ini berisi tentang hasil penelitian dan evaluasi performa peningkatan kualitas citra, segmentasi citra, dan klasifikasi citra, termasuk analisis dan interpretasi hasilnya.

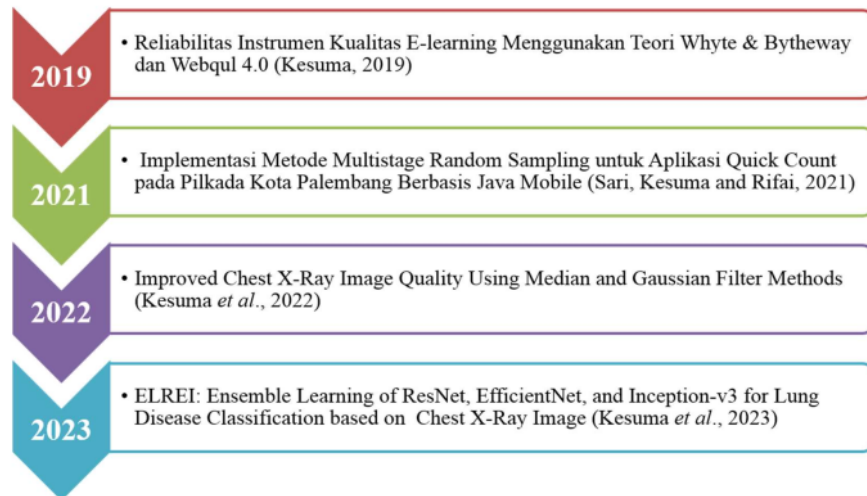
BAB 5 Penutup

Bab ini menyajikan kesimpulan secara singkat dan saran untuk penelitian lanjutan.

BAB II TINJAUAN PUSTAKA

2.1. Roadmap Penelitian

Roadmap penelitian ini disajikan pada Gambar 2.1. Pada tahun 2019, dilakukan pengujian reliabilitas variabel evaluasi menggunakan kerangka kerja *Whyte & Bytheway* dan *WebQual 4.0* untuk mengevaluasi kualitas *e-learning*. Pada tahun 2021, dilakukan pembuatan sistem kontrol melalui teknologi biometrik sidik jari dan pengembangan sistem *quick count* menggunakan Metode *Multistage Random Sampling*. Pada tahun 2022, dilakukan penelitian mengenai peningkatan kualitas Citra *Chest X-ray* dengan menggunakan Metode *Median dan Gaussian Filter*. Sementara untuk tahun 2023, berfokus pada penelitian ELREI: *Ensemble Learning* dari *ResNet*, *EfficientNet*, dan *Inception-v3* untuk klasifikasi penyakit paru-paru berdasarkan citra *Chest X-Ray*.



Gambar 2.1. Roadmap Penelitian Tahun 2019-2023

2.2. Penyakit Menular

Penyakit menular merupakan penyakit yang dapat menular kepada manusia baik secara langsung maupun melalui media yang disebabkan oleh mikroorganisme virus, bakteri, jamur, dan parasit [44]. Saat ini, salah satu penyakit menular yang

menjadi sebuah pandemi di berbagai negara adalah *Coronavirus Disease-2019 (COVID-19)*. *COVID-19* merupakan suatu penyakit menular akut yang menyerang sistem pernapasan secara berkala. Penyakit ini pertama kali dilaporkan di kota Wuhan, Provinsi Hubei, China pada bulan Desember 2019 yang disebabkan oleh virus corona varian baru yaitu *Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2)*, kemudian menyebar secara global dan dinyatakan sebagai pandemi *COVID-19* oleh *World Health Organization (WHO)* [45]. Pada umumnya, gejala yang dirasakan penderita *COVID-19* yaitu demam, sakit tenggorokan, batuk kering, sakit kepala, kelelahan, serta kehilangan indra penciuman dan perasa. Adapun gejala berat yang dirasakan seperti sesak napas, hilangnya kemampuan bergerak dan berbicara, hingga *pneumonia* di paru-paru [46], [47]. Gejala ini akan muncul sekitar 5 hingga 6 hari setelah seseorang terpapar virus *SARS-CoV-2* [48].

Untuk mendeteksi penyakit *COVID-19* dapat dilakukan melalui tes *Reverse Transcription Polymerase Chain Reaction (RT-PCR)* dan tes antibodi [2]. Namun, kedua tes tersebut membutuhkan biaya yang cukup mahal dan tenaga medis yang terbatas [3]. Alternatif lainnya yang lebih baik ditemukan para peneliti yaitu melalui *Chest X-Ray (CXR)* paru-paru. Gambar hasil pemeriksaan *CXR* paru-paru dinilai cukup efektif untuk mendiagnosa penyakit *COVID-19*, karena tenaga medis dapat melihat kelainan pada paru-paru penderita [49].

2.3. Citra Digital

Citra digital merupakan representasi diskrit dari citra yang berisi data dalam bentuk *array* (larik) dengan elemen-elemen yang mewakili intensitas atau tingkat keabuan di setiap koordinat spasial (x, y) dalam citra. Saat sebuah citra direpresentasikan secara digital, intensitas piksel di setiap titik dapat diwakili oleh nilai-nilai bilangan real atau kompleks [50]. Data citra digital dapat diproses dan dianalisis menggunakan komputer dengan metode-metode pengolahan citra digital.

Pengolahan citra digital adalah pemrosesan citra dengan bantuan komputer, dimana *input* dan *output* nya berupa citra atau karakteristik dari suatu gambar yang terkait. Secara umum, pengolahan citra digital terdiri dari beberapa tahapan yaitu peningkatan kualitas citra, segmentasi citra, dan klasifikasi citra [51]. Dalam era

teknologi informasi dan komunikasi saat ini, pengolahan citra digital menjadi bidang penting yang terus berkembang dan memiliki dampak yang besar pada kehidupan saat ini.

Citra yang umum digunakan dalam pengolahan citra digital adalah citra RGB (*Red, Green, Blue*). Citra RGB ini direpresentasikan dalam bentuk matriks dengan elemen-elemennya yaitu x_1 sebagai intensitas warna merah, x_2 sebagai intensitas warna hijau, dan x_3 sebagai intensitas warna biru. Masing-masing dari intensitas warna tersebut akan mewakili 8 bit di setiap pikselnya. Dalam kombinasi ketiganya, citra ini membentuk tampilan warna lengkap yang dapat ditampilkan dan diproses oleh bantuan komputer dalam proses pengolahan citra [52].

2.4. Citra *Chest X-ray*

Citra *Chest X-ray* (CXR) adalah gambar radiografi dari dada seseorang yang diambil dengan menggunakan sinar-X. Prosedur ini merupakan salah satu metode diagnostik medis paling umum untuk melihat struktur dalam dada, seperti jantung, paru-paru, tulang rusuk, dan organ lainnya. Citra CXR dapat membantu dokter radiologi dalam menganalisis dan mendiagnosa berbagai masalah kesehatan, seperti infeksi paru-paru, pneumonia, dan COVID-19 [4].

Citra CXR merupakan alat diagnostik yang cepat dan relatif murah, yang memungkinkan tenaga medis untuk mendapatkan gambaran awal tentang kondisi dada pasien. Namun, interpretasi citra CXR harus dilakukan oleh dokter spesialis atau ahli radiologi yang terlatih dalam membaca dan menganalisis hasil citra tersebut [5]. Namun citra CXR memiliki keterbatasan dalam diagnosis yang lebih mendalam, sehingga diperlukan pemeriksaan lebih lanjut secara otomatis dengan bantuan komputer.

2.5. Perbaikan Kualitas Citra

Perbaikan kualitas citra merupakan tahapan awal yang dilakukan untuk memperoleh kualitas citra yang lebih baik sehingga mempermudah proses pengklasifikasian citra. Beberapa citra medis sering memiliki kontras yang rendah dan *noise* karena berbagai sumber interferensi, seperti proses pencitraan dan

akuisisi data, sehingga citra sulit dianalisis secara visual [53]. Untuk itu, tahapan perbaikan kualitas citra ini perlu dilakukan seperti menghilangkan bagian yang tidak penting atau *noise*, meningkatkan kontras yang gelap, dan mempertahankan semua detail dari citra [54]. Beberapa teknik yang dilakukan pada tahapan ini sebagai berikut:

2.5.1. Morfologi *Opening*

Morfologi *Opening* merupakan teknik peningkatan kualitas citra dalam memperbaiki kontur citra dan menghilangkan objek tipis yang berada di sekitar area citra sehingga citra terlihat lebih jelas [7]. Metode ini bekerja dengan menggunakan operasi erosi yang dikombinasikan dengan operasi dilatasi. Morfologi pembukaan dapat dihitung dengan menggunakan Persamaan (2.1).

$$A \circ B = (A \ominus B) \oplus B \quad (2.1)$$

dengan

$$A \oplus B = \{x | (B)_x \cap A \neq \emptyset\}$$

$$A \ominus B = \{x | (B)_x \cap A^c \neq \emptyset\}$$

dimana, A merupakan citra awal, B merupakan structuring elemen berbentuk matriks operator seperti *line shape*, *disk*, *diamond*, dan lainnya. \oplus merupakan operasi dilasi, \ominus adalah operasi erosi, dan \circ adalah operasi Morfologi *Opening*.

2.5.2. Median Filter

Median Filter adalah fungsi nonlinier yang digunakan untuk pengurangan *noise* dan perataan gambar. *Median Filter* bekerja dengan mengurutkan nilai intensitas dari sekumpulan piksel secara menaik, kemudian mengubah nilai piksel yang diproses ke nilai tertentu [55]. Perhitungan *Median Filter* dapat menggunakan Persamaan (2.2).

$$MF(x, y) = \text{Median}(\text{piksel sekitar}(x, y)) \quad (2.2)$$

dimana, $MF(x, y)$ merupakan nilai piksel hasil proses *Median Filter* pada titik (x, y) dalam citra, sedangkan *Median* merupakan fungsi yang menghitung nilai *Median* dari sekumpulan piksel yang berada di sekitar titik (x, y) .

2.5.3. Mean Square Error

Mean Square Error (MSE) adalah salah satu metode yang umum digunakan untuk mengukur tingkat kesalahan atau perbedaan antara citra asli dengan citra hasil yang telah diproses atau ditingkatkan kualitasnya. Metode ini banyak digunakan dalam bidang pengolahan citra dan evaluasi kualitas gambar. MSE bekerja dengan cara menghitung selisih kuadrat antara setiap piksel pada citra asli (sebelum ditingkatkan) dan citra hasil pemrosesan (setelah ditingkatkan), kemudian mengambil nilai rata-rata dari seluruh selisih kuadrat ini. Semakin kecil nilai MSE, semakin baik kualitas citra hasil pemrosesan, karena artinya selisih piksel antara citra asli dan hasil pemrosesan juga semakin kecil [56]. Perhitungan nilai MSE dapat dilihat pada Persamaan (2.3).

$$MSE = \frac{1}{N} \sum_{i=1}^N (I(i) - K(i))^2 \quad (2.3)$$

dimana, N adalah jumlah piksel dalam citra, $I(i)$ adalah nilai piksel citra asli pada titik ke- i , $K(i)$ adalah nilai piksel hasil peningkatan kualitas citra pada titik ke- i .

2.5.4. Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) adalah metode yang digunakan untuk mengukur kualitas citra dengan membandingkan citra asli dengan citra hasil pemrosesan. PSNR biasanya digunakan untuk mengevaluasi sejauh mana kualitas citra yang dihasilkan mendekati citra asli. PSNR mengukur seberapa besar perbedaan antara citra asli dengan citra hasil pemrosesan dengan menggunakan rasio antara kekuatan sinyal puncak (*peak signal*) dan tingkat *noise* (*noise ratio*). PSNR biasanya diukur dalam satuan desibel (dB). Semakin tinggi nilai PSNR, maka semakin mirip citra hasil pemrosesan dengan citra aslinya sehingga kualitas

citra tersebut dianggap semakin baik. PSNR sekitar 30 dB dianggap sebagai nilai ambang batas yang baik untuk citra digital [10]. Perhitungan nilai PSNR dapat dilihat pada Persamaan (2.4).

$$PSNR = 10 \log_{10} \left(\frac{Max^2}{MSE} \right) \quad (2.4)$$

dimana Max adalah nilai maksimum piksel dalam citra (misalnya, 255 untuk citra 8-bit dengan rentang 0-255), MSE adalah *mean square error*.

2.5.5. Structural Similarity Index Metrics

Structural Similarity Index Metrics (SSIM) adalah metrik yang digunakan dalam evaluasi peningkatan kualitas citra atau perbandingan antara dua citra. SSIM biasanya digunakan untuk mengukur sejauh mana citra yang telah ditingkatkan kualitasnya memiliki kesamaan struktural dengan citra asli. SSIM bekerja dengan membandingkan tiga aspek utama dari dua citra yang ingin dievaluasi yaitu kecerahan (*brightness*), kontras (*contrast*), dan struktur (*structure*). Dengan mempertimbangkan ketiga elemen ini, SSIM dapat memberikan nilai yang mengindikasikan sejauh mana citra hasil pemrosesan mendekati citra asli. SSIM menghasilkan nilai antara -1 dan 1. Nilai 1 berarti citra hasil pemrosesan identik dengan citra asli, sementara nilai mendekati -1 menunjukkan perbedaan yang sangat signifikan antara kedua citra. Semakin tinggi nilai SSIM, semakin mirip citra hasil pemrosesan dengan citra asli, dan ini menunjukkan peningkatan kualitas citra [57], [58]. Perhitungan nilai SSIM dapat dilihat pada Persamaan (2.5).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1) \times (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1) \times (\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.5)$$

dimana μ_x, μ_y adalah rata-rata piksel citra asli dan hasil peningkatan kualitas citra, σ_x, σ_y adalah simpangan baku piksel citra asli dan hasil peningkatan kualitas citra,

adalah σ_{xy} kovariansi antara piksel citra asli dan hasil peningkatan kualitas citra, C_1, C_2 adalah konstanta untuk menghindari pembagian dengan nol.

2.6. Augmentasi Data

Pada proses *training* algoritma berbasis *Deep Learning* (DL) membutuhkan jumlah data yang besar untuk menghindari masalah *overfitting* [59]. Suatu kondisi dimana model terlalu bergantung pada data latih sehingga model tidak dapat melakukan generalisasi pada data baru dan dapat mengurangi kinerja [59], [60]. Augmentasi data merupakan salah satu teknik untuk memperbanyak data latih dengan mentransformasi citra asli pada orientasi yang berbeda, sehingga tercipta keragaman data serta meningkatkan ketahanan model [61]. Transformasi yang paling banyak digunakan dalam augmentasi data yaitu rotasi, refleksi, *flipping*, dan lainnya. Augmentasi rotasi dilakukan dengan memutar citra ke kanan atau ke kiri pada sumbu antara 1° sampai dengan 359° , sedangkan *flipping* merupakan proses augmentasi dengan membalikkan citra ke arah vertikal atau horizontal [12].

2.7. Segmentasi Citra

Segmentasi citra merupakan proses pengambilan objek dari suatu citra dengan membagi piksel citra tersebut ke dalam beberapa bagian berdasarkan tekstur, intensitas, tingkat abu-abu, dan lain-lain [62], [63]. Pada kasus penyakit menular yang menyerang paru-paru, segmentasi paru-paru pada citra CXR dapat menunjukkan kelainan pada paru-paru sehingga mempermudah proses deteksi penyakit, menghemat waktu pengerjaan, serta meningkatkan akurasi [64]. Dalam segmentasi paru-paru, citra CXR akan dibagi menjadi dua bagian yaitu paru-paru sebagai *foreground* dan bagian lainnya selain paru-paru sebagai *background*. Hasil dari segmentasi paru-paru ini dapat dilanjutkan ke tahapan berikutnya yaitu proses klasifikasi penyakit *COVID-19*.

2.8. Klasifikasi Citra

Klasifikasi citra merupakan suatu proses untuk mengkategorikan citra ke dalam salah satu label yang telah ditentukan menggunakan model tertentu [65].

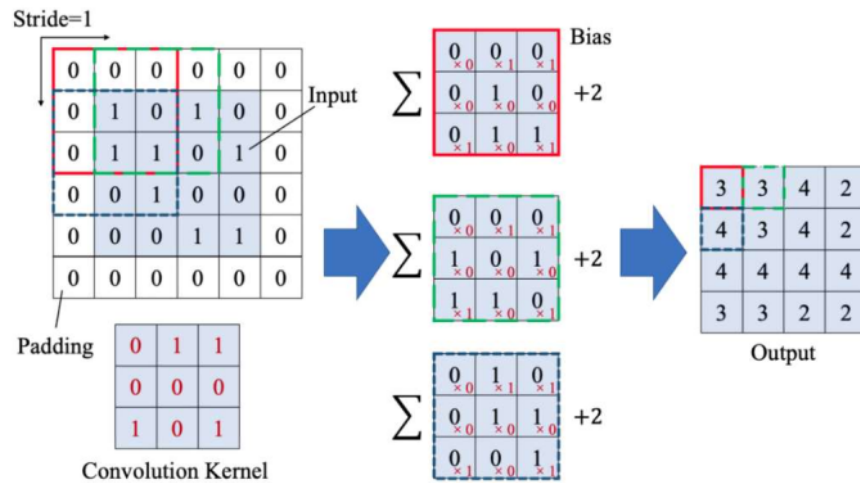
Klasifikasi citra terdiri dari dua proses antara lain tahap *training* dan *testing*. Tahap *training* merupakan suatu proses membangun model klasifikasi dengan melakukan pelatihan model menggunakan data latih yang telah ditentukan, sedangkan tahapan *testing* merupakan suatu proses memberi label pada data citra menggunakan model klasifikasi yang diperoleh dari *training*.

2.9. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu metode *Deep Learning* yang menggabungkan tahapan ekstraksi fitur dan klasifikasi yang telah terbukti keberhasilannya dalam berbagai aplikasi *computer vision* seperti deteksi objek, *natural language processing*, segmentasi, serta klasifikasi citra [66], [67]. Dalam sistem klasifikasi citra, CNN lebih unggul dibandingkan metode konvensional lainnya karena tidak memerlukan tahapan *preprocessing* citra [68]. Komponen CNN terdiri dari tiga lapisan diantaranya *input layer*, *hidden layer*, dan *output layer*. *Hidden layer* pada CNN berfungsi sebagai ekstraksi fitur yang terdiri dari beberapa *convolution layer*, *pooling layer*, dan fungsi aktivasi, sedangkan pada *output layer* dilakukan proses klasifikasi dengan menerapkan *fully connected layer* dengan *output* berupa probabilitas [69].

2.9.1. Convolution Layer

Convolution layer merupakan lapisan utama pada CNN yang melakukan operasi konvolusi untuk mempelajari representasi fitur dari *input* citra [69]. *Convolution layer* menerapkan beberapa kernel untuk menghasilkan sejumlah *feature map* yang mewakili berbagai karakteristik dari *input* citra [70]. Kernel konvolusi (*filter*) merupakan matriks bobot berukuran $n \times n$ dengan entri-entri matriks berupa angka yang dibangkitkan secara acak. Pada proses *convolution layer* terdapat beberapa parameter yang digunakan diantaranya *stride* dan *padding*. *Stride* merupakan jumlah langkah pergeseran kernel pada saat operasi konvolusi dilakukan. *Padding* merupakan penambahan jumlah piksel yang berisi nilai 0 di sekeliling *input* citra untuk menjaga ukuran dimensi citra [69]. Adapun ilustrasi operasi konvolusi dapat dilihat pada Gambar 2.2 [71].



Gambar 2.2. Proses operasi konvolusi

Pada Gambar 2.2 terlihat proses operasi konvolusi antara *input citra* dengan kernel konvolusi. *Input citra* direpresentasikan dalam bentuk matriks berukuran 4×4 serta adanya penambahan *padding same* sehingga matriks *input* berukuran 6×6 . Matriks *input* tersebut dikonvolusi dengan kernel berukuran 3×3 dengan *stride* 1. Operasi konvolusi pertama diawali dengan operasi konvolusi antara area lokal matriks *input* yang ditandai dengan kotak berwarna merah dengan kernel konvolusi. Selanjutnya, kernel konvolusi bergeser satu langkah ke kanan yang ditunjukkan pada kotak berwarna hijau dan dilakukan seperti pada operasi konvolusi pertama. Lakukan proses tersebut hingga operasi konvolusi terakhir sehingga menghasilkan *feature map* berukuran 4×4 . Untuk menghitung hasil dari *convolution layer* dapat menggunakan Persamaan (2.6) sebagai berikut [72]:

$$C_q^p = (A_p * K_q) + b_q \quad (2.6)$$

dengan * merupakan operasi konvolusi dan untuk menghitung entri matriks dari hasil konvolusi dapat menggunakan Persamaan (2.7) sebagai berikut:

$$c_{i,j} = \left(\sum_{u=0}^{n-1} \sum_{v=0}^{n-1} (a_{u+i,v+j} \times k_{u+1,v+1}) \right) + b_q \quad (2.7)$$

dimana, i merupakan baris, j adalah kolom, n adalah ukuran tinggi kernel, A_p merupakan matriks input ke- p , K_q adalah matriks kernel ke- q , b_q adalah bias untuk kernel ke- q , C_q^p merupakan matriks hasil konvolusi (*feature maps*) kernel ke- q pada input ke- p , $a_{i,j}$ adalah entri baris ke- i dan kolom ke- j pada matriks A_p , $k_{i,j}$ adalah entri baris ke- i dan kolom ke- j pada matriks kernel K_q , dan $c_{i,j}$ adalah entri baris ke- i dan kolom ke- j pada matriks C_q^p .

2.9.2. Batch Normalization

Batch normalization merupakan lapisan tambahan yang berfungsi untuk menormalisasikan nilai input pada lapisan berikutnya, mengurangi risiko *overfitting*, dan mempercepat proses *training* [70], [73]. *Batch normalization* mentransformasikan distribusi nilai input menjadi distribusi normal standar dengan rata-rata 0 dan varians 1 [74]. *Batch normalization* menormalisasikan input citra dengan mengurangi setiap entri input citra dengan rata-rata *mini-batch*, kemudian membaginya dengan varians *mini-batch* ditunjukkan pada Persamaan (2.8) berikut ini [75].

$$\hat{c}_{i,j} = \frac{c_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}} \quad (2.8)$$

Rata-rata (μ_j) dan variansi (σ_j^2) untuk masing-masing *mini-batch* dari matriks input $m \times n$ dapat dihitung menggunakan Persamaan (2.9) dan (2.10) sebagai berikut [75]:

$$\mu_j = \frac{1}{m} \sum_{i=1}^m c_{i,j} \quad (2.9)$$

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (c_{i,j} - \mu_j)^2 \quad (2.10)$$

dimana, n adalah jumlah *mini-batch* (kolom), m adalah banyaknya data dalam satu *mini-batch* (baris), c_{ij} merupakan entri matriks *input* C pada baris ke- i dan kolom ke- j , \hat{c}_{ij} merupakan entri matriks *input* C yang telah dinormalisasi pada baris ke- i dan kolom ke- j , dan ε merupakan konstanta positif terkecil.

2.9.3. Fungsi Aktivasi

Dalam CNN, fungsi aktivasi diterapkan pada *output* dalam setiap *convolution layer* dan *fully connected layer* untuk menambahkan faktor nonlinier sehingga mampu mengatasi berbagai masalah yang kompleks seperti meningkatkan kecepatan saat *training* [75]. Fungsi *Rectified Linear Unit (ReLU)*, *sigmoid*, *swish*, dan *softmax* merupakan beberapa contoh fungsi aktivasi yang sering digunakan. *ReLU* merupakan fungsi aktivasi dengan biaya komputasi yang rendah dan konvergensi gradien yang lebih baik [70]. *ReLU* membuat seluruh nilai piksel pada citra yang bernilai negatif menjadi 0, sedangkan akan bernilai sama apabila nilainya positif. Fungsi aktivasi *sigmoid* merupakan fungsi aktivasi dengan *input* berupa bilangan riil dan *output*-nya berada dalam interval (0, 1) [76]. Fungsi aktivasi *swish* merupakan fungsi aktivasi yang lebih efisien untuk memproses data besar dan jaringan kompleks yang lebih dalam. Kemudian, untuk fungsi *softmax* biasanya aktivasi digunakan pada *output layer* untuk menghasilkan *output* dalam distribusi probabilitas sehingga nilai *output* berada dalam interval(0, 1) dan total jumlahnya adalah 1 [76]. Secara matematis, fungsi aktivasi *ReLU*, *sigmoid*, *swish*, dan *softmax* didefinisikan pada Persamaan (2.11), (2.12), (2.13), dan (2.14) [75], [76].

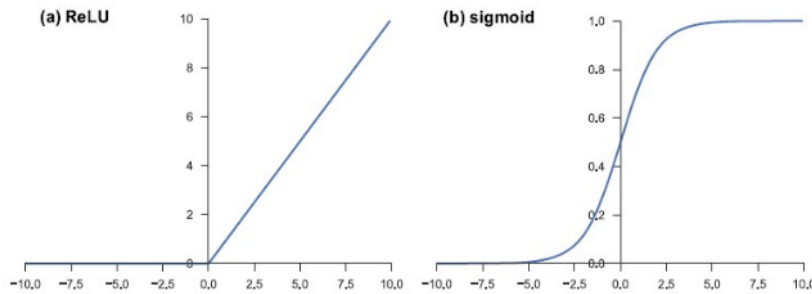
$$r(\hat{c}_{i,j}) = \max(\hat{c}_{i,j}, 0) = \begin{cases} \hat{c}_{i,j} & \text{jika } \hat{c}_{i,j} \geq 0 \\ 0 & \text{jika } \hat{c}_{i,j} < 0 \end{cases} \quad (2.11)$$

$$\sigma(\hat{c}_{i,j}) = \frac{1}{1 + e^{-\hat{c}_{i,j}}} \quad (2.12)$$

$$s(\hat{c}_{i,j}) = \hat{c}_{i,j} \left(\frac{1}{1 + e^{-\beta(\hat{c}_{i,j})}} \right) \quad (2.13)$$

$$p(\hat{c}_{i,j}) = \frac{e^{\hat{c}_{i,j}}}{\sum_{k=1}^n e^{\hat{c}_{i,j}}} \quad ; i = 1, 2, \dots, n \quad (2.14)$$

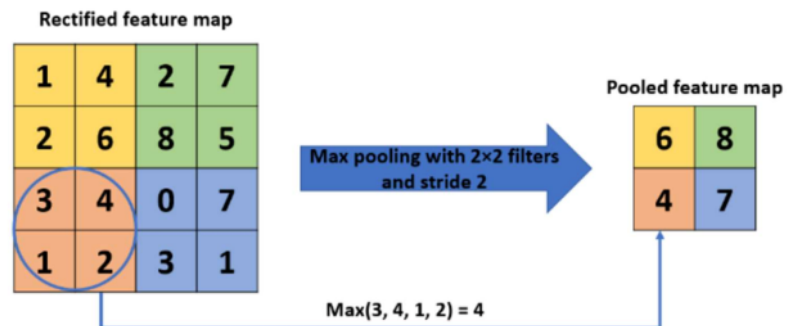
untuk $\hat{c}_{i,j} \in (-\infty, +\infty)$, dimana $r(\hat{c}_{i,j})$ adalah fungsi aktivasi *ReLU*, $\sigma(\hat{c}_{i,j})$ adalah fungsi aktivasi sigmoid dengan $\sigma(\hat{c}_{i,j}) \in (0,1)$, $s(\hat{c}_{i,j})$ adalah fungsi aktivasi *swish* dengan β merupakan parameter yang dipelajari, dan $p(\hat{c}_{i,j})$ merupakan nilai probabilitas *input* ke- i dari fungsi *softmax*. Gambar grafik fungsi *ReLU* dan *sigmoid* dapat dilihat pada Gambar 2.3. sebagai berikut [69]:



Gambar 2.3. Grafik fungsi aktivasi *ReLU* dan *sigmoid*

2.9.4. Pooling Layer

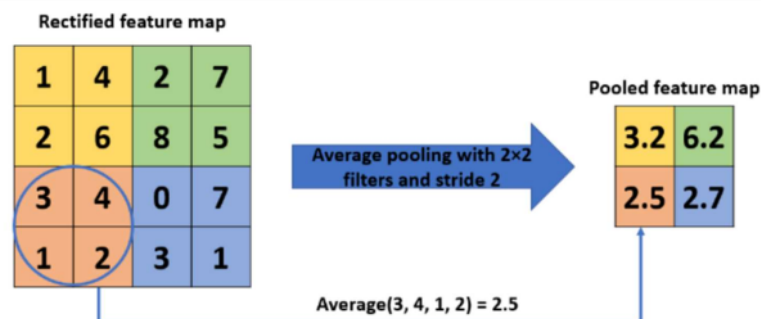
Pooling layer merupakan jenis *hidden layer* yang sangat umum digunakan pada CNN. Lapisan ini membantu memperkecil dimensi *feature map* dengan mempertahankan sebagian besar fitur dominan di setiap langkah tahap *pooling* (Alzubaidi *et al.*, 2021). *Pooling layer* melibatkan dua parameter utama untuk melakukan operasi penyatuan, yaitu ukuran kernel *pooling* dan *stride*. Adapun beberapa jenis *pooling layer* yang digunakan antara lain *max pooling* dan *average pooling*. *Max pooling* bekerja dengan mengambil nilai maksimum pada setiap submatriks yang dihasilkan melalui pergeseran kernel *pooling* dan *stride*. Ilustrasi *max pooling* dapat dilihat pada Gambar 2.4 [77]. Ilustrasi *max pooling* dapat dilihat pada Gambar 2.4. [77].



Gambar 2.4. Ilustrasi proses *max pooling* 2×2

Pada Gambar 2.4. dapat dilihat proses *max pooling* dengan matriks *input* berukuran 4×4 . Matriks *input* akan dipartisi menjadi 4 submatriks karena kernel *pooling* yang digunakan berukuran 2×2 dan stride 2 yang ditunjukkan pada kotak berwarna kuning, hijau, oranye, dan biru. Selanjutnya, rata-rata dari masing-masing submatriks yang telah terbentuk dihitung sehingga menghasilkan *output* berukuran 2×2 .

⁵ *Average pooling* bekerja dengan mengambil nilai rata-rata pada setiap submatriks yang dihasilkan melalui pergeseran kernel *pooling* dan *stride*, sedangkan *global average pooling* dengan mengambil nilai rata-rata dari *input* citra. Ilustrasi *average pooling* dapat dilihat pada Gambar 2.4 [77].

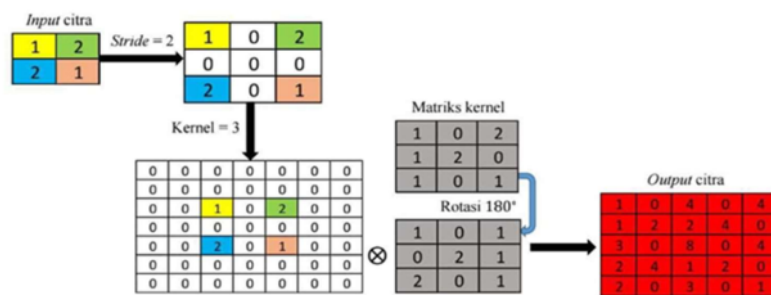


Gambar 2.5. Ilustrasi proses *average pooling* 2×2 dan *stride* 2

Pada Gambar 2.5. dapat dilihat proses *average pooling* dengan matriks *input* berukuran 4×4 menggunakan kernel *pooling* 2×2 dan *stride* 2. Pertama-tama, matriks *input* akan dipartisi menjadi 4 submatriks yang ditunjukkan pada kotak berwarna kuning, hijau, oranye, dan biru. Masing-masing submatriks yang telah terbentuk diambil nilai rata-ratanya sehingga menghasilkan matriks *output* yang berukuran 2×2 .

2.9.5. Transposed Convolution

Transposed Convolution merupakan kebalikan dari *pooling layer* yang mana lapisan ini berfungsi untuk meningkatkan dimensi *feature map* untuk menghasilkan dimensi *feature maps* yang sama dengan *input* citra awal [78]. Ilustrasi proses *transposed convolution* dapat dilihat pada Gambar 2.6.



Gambar 2.6. Ilustrasi *transposed convolution* 2×2 dan *stride* 2

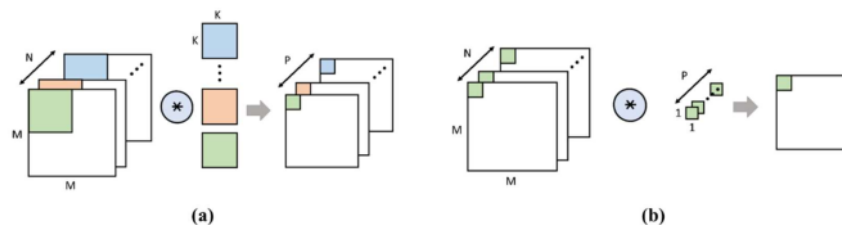
Pada Gambar 2.6 menunjukkan proses *transposed convolution* menggunakan kernel 2×2 dan *stride* 2 terhadap matriks *input* berukuran 2×2 . Proses *transposed convolution* diawali dengan penambahan angka 0 diantara setiap entri dari matriks *input*, dimana jumlahnya yaitu nilai *stride* dikurangi 1. Selanjutnya keseluruhan entri dari matriks *input* ditambahkan dengan 0 mengikuti aturan *padding*. Banyaknya penambahan angka 0 adalah ukuran matriks kernel dikurangi 1. Lalu matriks kernel dirotasi 180° dan dilakukan operasi konvolusi dengan matriks *input* yang berukuran 7×7 . Setelah semua entri di dalam matriks tersebut di konvolusi, maka akan menghasilkan *output* berukuran 6×6 yang ditandai dengan kotak warna merah.

2.9.6. Concatenate Layer

Concatenate layer merupakan lapisan yang berfungsi untuk menggabungkan dua *feature map* yang berdimensi sama [79]. *Concatenate* dapat memekstraksi citra lebih baik dengan menghasilkan fitur yang lebih halus dan memberikan detail yang tajam [80]. Penggunaan *concatenate* juga mampu menangani beberapa detail fitur citra yang hilang sehingga memungkinkan informasi yang relevan dapat digunakan sebagai fitur citra [81].

2.9.7. Depthwise Separable Convolution

Depthwise separable convolution telah terbukti berhasil dalam klasifikasi citra, dimana lapisan ini mampu menghindari ekstraksi fitur yang berlebihan dan mengurangi jumlah parameter serta komputasi yang digunakan dalam operasi konvolusi [82], [83]. Lapisan ini biasanya disebut dengan “*separable convolution*” yang terdiri dari dua lapisan konvolusi yaitu *depthwise convolution* dan *pointwise convolution*. *Depthwise convolution* merupakan konvolusi spasial yang dilakukan secara independen pada setiap *input channel* sehingga setiap *channel*. Lapisan selanjutnya yaitu *pointwise convolution*, proses konvolusi yang menggunakan kernel berukuran 1×1 untuk menggabungkan semua hasil *feature map* dari *depthwise convolution* sehingga menghasilkan *feature map* baru [83], [84]. Ilustrasi proses dalam *depthwise separabel convolution* dapat dilihat pada Gambar 2.7.



Gambar 2.7. Proses *depthwise separabel convolution* (a) *depthwise convolution*
(b) *pointwise convolution*

Pada Gambar 2.7. menunjukkan bagaimana *depthwise convolution* dan *pointwise convolution* bekerja. Sebuah matriks *input* berukuran $N \times N \times L$ dan kernel konvolusi berukuran $N \times N \times L$, dimana L merupakan banyaknya *channel*.

Pada proses *depthwise convolution*, setiap *channel* pada matriks *input* dikonvolusi secara independen dengan masing-masing kernelnya sehingga menghasilkan *feature map* berukuran $N \times N \times L$. Selanjutnya, diproses ke dalam *pointwise convolution*. *Feature map* hasil *depthwise convolution* dikonvolusi dengan kernel berukuran $1 \times 1 \times L$ sehingga menghasilkan *feature map* berukuran $N \times N \times 1$. Adapun untuk menghitung hasil *depthwise convolution* dapat menggunakan Persamaan (2.15) sebagai berikut [83]:

$$D_{l(i,j)} = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} (W_{l(u+i,v+j)} \times K_{l(u+1,v+1)}) \quad ; i, j = i, 2, \dots, n \quad (2.15)$$

dimana D_l merupakan matriks *depthwise convolution channel* ke- l , W_l merupakan matriks *input channel* ke- l , K_l merupakan matriks kernel *channel* ke- l , dan $D_{l(i,j)}$ merupakan entri matriks D_l baris ke- i dan kolom ke- j . Kemudian, dilanjutkan dengan proses *pointwise convolution* menggunakan kernel berukuran 1×1 yang dapat dihitung dengan menerapkan Persamaan (2.16) sebagai berikut [83]:

$$P_{l(i,j)} = \sum_{u=0}^{n-1} (H_{l(i,j,u+1)} \times K_{l(i,j,u+1)}) \quad ; i, j = i, 2, \dots, n \quad (2.16)$$

dimana P_l merupakan matriks *pointwise convolution* ke- l , H_l merupakan matriks *input* ke- l , dan $P_{l(i,j)}$ merupakan entri matriks P_l baris ke- i dan kolom ke- j .

2.9.8. Dense Layer

Dense layer merupakan lapisan yang berfungsi untuk mengklasifikasikan citra yang telah melewati proses ekstraksi fitur ke dalam label yang telah ditentukan. Lapisan ini bekerja dengan mengumpukan seluruh *output* pada lapisan sebelumnya ke semua neuronnya, dimana setiap neuron tersebut akan menghasilkan satu *output* ke lapisan berikutnya dan kemudian diproses ke dalam fungsi aktivasi [85].

2.9.9. Dropout

Pada proses *training*, masalah yang sering terjadi adalah ketidaksesuaian antara model data latih dan jumlah dataset yang tidak seimbang (*overfitting*). Untuk mencegah terjadinya *overfitting* dapat dilakukan dengan melakukan penambahan *dropout*. Cara kerja *dropout* dilakukan dengan menghilangkan neuron secara acak pada setiap lapisan yang ada pada arsitektur sesuai dengan *dropout rate* yang digunakan [86].

2.9.10. Loss Function

Pada *output layer* CNN juga terdapat *loss function* yang digunakan untuk menghitung kesalahan prediksi selama proses *training* [76]. Nilai ini merupakan perbedaan antara nilai aktual dengan hasil prediksi menggunakan metode yang diusulkan. *Loss function* terdiri dari beberapa macam diantaranya *Binary Cross Entropy* (BCE) dan *Categorical Cross Entropy* (CCE). BCE merupakan *loss function* yang digunakan dalam tugas klasifikasi biner, sedangkan CCE digunakan untuk klasifikasi multi-label. *Loss function* BCE dapat dihitung menggunakan Persamaan (2.17) sebagai berikut [87]:

$$L_{BCE} = -\frac{1}{m \times n} \left[\sum_{i=1}^m \sum_{j=1}^n (y_{i,j} \log(p_{ij}) + (1 - y_{ij}) \log(1 - p_{ij})) \right] \quad (2.17)$$

dimana $m \times n$ menunjukkan ukuran matriks hasil prediksi segmentasi, y_{ij} merupakan entri matriks label *ground truth* pada baris ke- i kolom ke- j (0 untuk piksel *background* dan 1 untuk piksel paru-paru), dan p_{ij} merupakan entri matriks hasil prediksi segmentasi pada baris ke- i kolom ke- j , dan L adalah nilai *loss function binary cross entropy*.

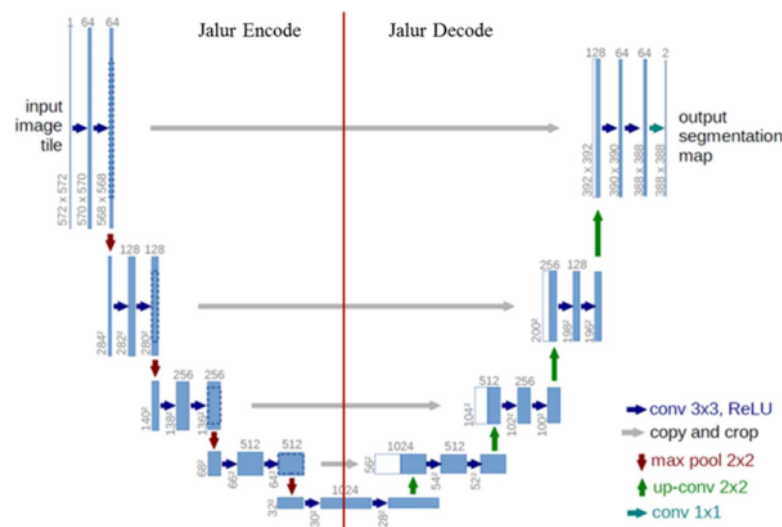
Untuk menghitung hasil *loss function* CCE dapat menggunakan Persamaan (2.18) dengan menerapkan fungsi softmax sebagai berikut [88]:

$$L_{CCE}(p, q) = - \sum_k^K p(x_k) \log q(x_k) \quad ; x_k \in K \quad (2.18)$$

dimana $p(x_k)$ merupakan probabilitas dari label prediksi pada label ke- k , $q(x_k)$ probabilitas dari label aktual pada label ke- k , K adalah himpunan label klasifikasi, dan $L(p, q)$ adalah hasil *loss function categorical cross entropy*.

2.10. ⁶ *U-Net*

U-Net merupakan salah satu arsitektur CNN yang pertama kali diperkenalkan oleh [17] dan telah terbukti keberhasilannya dalam segmentasi citra terutama pada citra medis. Arsitektur ini berbentuk seperti huruf “U” yang terdiri dari ⁴ dua jalur yaitu jalur *encode* dan jalur *decode*. Pada jalur *encode* digunakan untuk mengambil informasi dari *input* citra, sedangkan jalur *decode* untuk menentukan objek yang diinginkan secara efisien [89]. Arsitektur *U-Net* pada citra CXR dapat dilihat pada Gambar 2.8. [17].



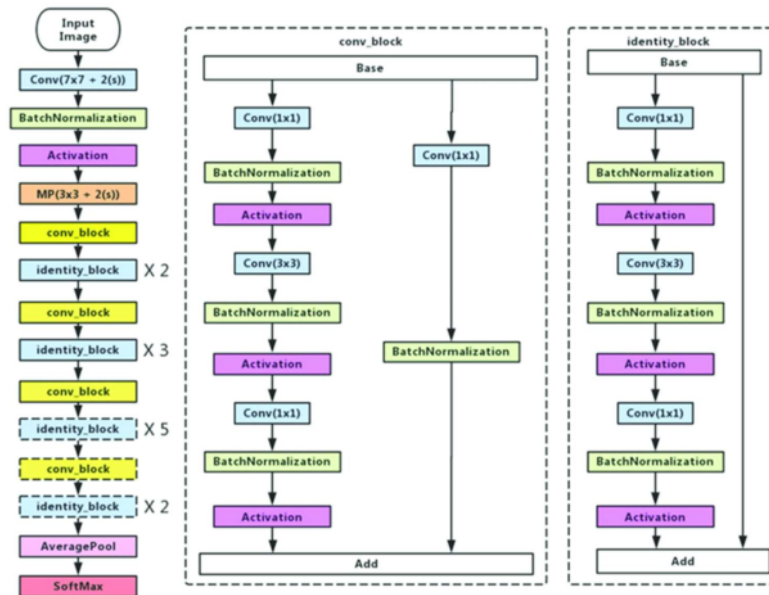
Gambar 2.8. Ilustrasi arsitektur *U-Net*

Pada Gambar 2.8 menunjukkan lapisan-lapisan yang digunakan pada arsitektur *U-Net* dengan *input* citra CXR. ⁵ Pada setiap blok jalur *encode* terdiri dari

dua lapisan konvolusi dengan kernel berukuran 3×3 masing-masing diikuti dengan fungsi aktivasi *ReLU*, kemudian dilakukan *max pooling* 2×2 dengan *stride* 2 untuk pengurangan dimensi citra. Pada setiap langkah *max pooling*, jumlah filter dimulai dari 64, kemudian untuk blok selanjutnya menjadi 128, 256, 512, dan 1024. Pada setiap blok di jalur *decode* terdiri dari *upsampling* 2×2 dan diikuti dengan penggabungan *feature maps* dari jalur *encoder* dan *feature maps* hasil *upsampling* menggunakan *concatenate*. Kemudian dilanjutkan dengan konvolusi sebanyak dua kali menggunakan kernel berukuran 3×3 , dan masing-masing diikuti oleh fungsi aktivasi *ReLU*. Pada lapisan terakhir, digunakan konvolusi 1×1 untuk memetakan setiap nilai piksel *feature maps* ke kelas yang diinginkan. *Output* citra CXR setelah dilakukan konvolusi 1×1 yaitu citras hasil segmentasi dengan paru-paru yang ditandai dengan piksel berwarna putih, dan *background* yang ditandai dengan piksel berwarna hitam.

2.11. Residual Network

Residual Network (ResNet) memanfaatkan “*skip connection*” dalam lapisannya untuk menangani masalah penurunan gradien, sehingga mampu mempercepat konvergensi jaringan dalam [76]. Arsitektur ini memiliki kompleksitas komputasi yang lebih rendah dibandingkan arsitektur lainnya, meskipun kedalamannya diperbesar [25]. Jaringan yang dalam pada arsitektur *ResNet* terbukti lebih baik dalam melakukan tugas klasifikasi karena dapat mengekstraksi fitur lebih representatif [13]. Arsitektur *ResNet* yang paling terkenal adalah *ResNet-50*, yang memiliki 50 lapisan. Namun, ada juga variasi lain seperti *ResNet-18*, *ResNet-101*, dan sebagainya, yang berbeda dalam jumlah lapisan dan kompleksitasnya. Arsitektur *ResNet* telah digunakan dalam banyak tantangan pengenalan citra dan telah menunjukkan performa yang sangat baik dalam banyak tugas, termasuk klasifikasi citra. Adapun ilustrasi arsitektur *ResNet* dapat dilihat pada Gambar 2.9 [90].



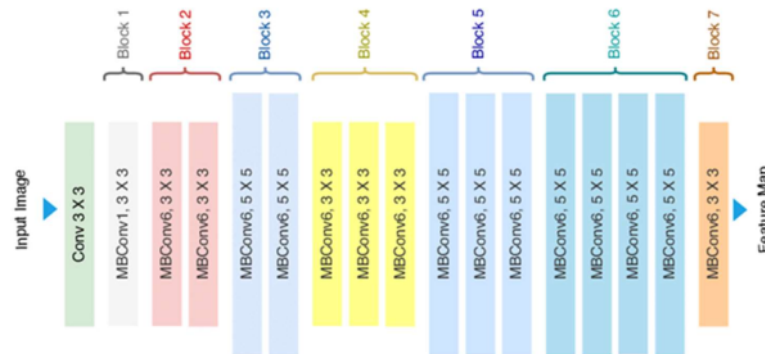
Gambar 2.9. Ilustrasi arsitektur *ResNet*

Pada Gambar 2.9. dapat dilihat bahwa arsitektur *ResNet* terdapat dua jenis *block* yang digunakan yaitu *identity_block* dan *conv_block* yang mana pada *block* tersebut terdapat *skip connection* untuk mengatasi masalah penurunan gradien akibat jaringan yang terlalu dalam. Pada bagian *conv_block*, *input* diproses ke dalam *convolution layer* dan *batch normalization* yang diikuti oleh fungsi aktivasi *ReLU* sebanyak tiga kali. Lalu, *input* juga melewati *residual connection* yang diproses ke dalam *convolution layer* dan *batch normalization*. Kemudian, dilakukan penambahan hasil dari kedua proses tersebut. Pada bagian *identity_block*, *input* diproses ke dalam *convolution layer* dan *batch normalization* yang diikuti oleh fungsi aktivasi *ReLU* sebanyak tiga kali. Kemudian, hasil tersebut ditambahkan dengan *input*.

2.12. *EfficientNet*

EfficientNet merupakan salah satu arsitektur CNN yang memiliki parameter yang sedikit dan menghasilkan akurasi klasifikasi yang lebih baik [91]. Arsitektur ini diperkenalkan oleh Tan and Le [29], dalam penelitiannya arsitektur *EfficientNet*

mengungguli jumlah parameter dan akurasi dari semua arsitektur sebelumnya pada dataset *ImageNet*. Keunggulan arsitektur *EfficientNet* dibandingkan arsitektur lainnya yaitu dapat menskalakan lebar lapisan, resolusi citra, kedalaman lapisan, dan lainnya secara efisien, serta mengurangi set parameter sehingga proses *training* lebih efisien [30]. Blok utama yang menyusun arsitektur ini adalah *Mobile inverted bottleneck convolution (MBCConv)* yang dapat dilihat pada Gambar 2.10. [92].

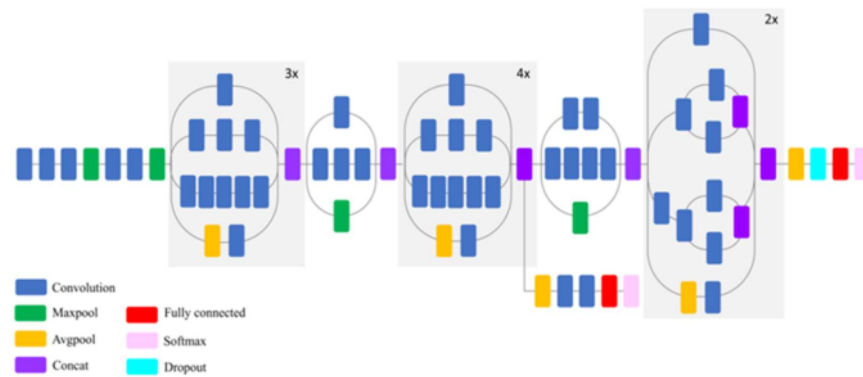


Gambar 2.10. Ilustrasi arsitektur *EfficientNet*

Pada Gambar 2.10. terlihat bahwa arsitektur *EfficientNet* terdiri dari beberapa lapisan diantaranya *convolution layer* dengan kernel berukuran 3×3 dan *MBCConv*. Terdapat dua tipe *MBCConv* yang digunakan yaitu *MBCConv1* dan *MBCConv6*. *MBCConv1* hanya digunakan satu kali dengan kernel berukuran 3×3 , sedangkan *MBCConv6* digunakan setelahnya dengan ukuran kernel yang berbeda (3×3 dan 5×5). *MBCConv* terdiri dari *Depthwise Convolution*, *Batch Normalization*, fungsi aktivasi *Swish*, *Squeeze and Excitation (SE)*, *Convolution*, dan *Add*.

2.13. Inception-v3

Arsitektur *Inception-v3* diperkenalkan pertama kali oleh Szegedy et al [93] yang menggunakan blok dasar yaitu *inception block*. Blok ini berfungsi untuk meminimalkan biaya komputasi tanpa mempengaruhi jaringan yang lebih dalam dan *overfitting* melalui pengurangan dimensi dengan konvolusi 1×1 yang bertumpuk [94]. Arsitektur ini memiliki 48 lapisan yang terdiri dari 11 *inception block* yang dapat dilihat pada Gambar 2.11 [95].



Gambar 2.11. Ilustrasi arsitektur *Inception-v3*

Pada Gambar 2.10 menunjukkan bahwa lapisan-lapisan yang menyusun arsitektur *Inception-v3* antara lain *convolution layer*, *max pooling*, *inception block*, *global average pooling*, *concatenate*, *dense layer*, dan *softmax*. Arsitektur ini terdiri dari 11 blok awal (3 blok A, 1 blok B, 4 blok C, 1 blok D, dan 2 blok E), masing-masing dengan jumlah lapisan konvolusi dan ukuran kernel yang bervariasi. *Output* dari setiap jalur di *inception block* digabungkan menggunakan *concatenate*.

2.14. Ensemble Learning

Pendekatan *ensemble learning* dilakukan dengan melatih beberapa arsitektur CNN dan menggabungkan hasil prediksinya untuk menghasilkan model baru yang lebih akurat daripada hasil klasifikasi tunggal [96]. Keuntungan yang dimiliki metode ini yaitu dapat memanfaatkan informasi dari beberapa pengklasifikasian, meminimalkan kesalahan prediksi, serta kesalahan pembelajaran fitur masih dapat diklasifikasikan dengan benar menggunakan pola yang dipelajari oleh pengklasifikasian lainnya [97]. Teknik yang sering digunakan untuk menggabungkan hasil klasifikasi tunggal dari masing-masing model pada metode *ensemble learning* adalah *weighted voting*. Teknik ini memanfaatkan bobot dari masing-masing arsitektur yang digunakan.

Untuk menentukan hasil prediksi menggunakan teknik *weighted voting* dapat menerapkan Persamaan (2.19) sebagai berikut:

$$f_i = \sum_{i=1}^n (w_i g_i) \quad (2.19)$$

dimana f_i merupakan hasil prediksi label untuk $f_i \in [0, 1]$, w_i merupakan bobot hasil *training* untuk arsitektur ke- i , dan g_i merupakan hasil fungsi aktivasi *softmax* pada arsitektur ke- i .

2.15. Confusion Matrix

Confusion matrix merupakan konsep dari *deep learning* yang digunakan untuk mengukur kinerja model, yang berisi informasi tentang label aktual dengan hasil prediksi [65], [94]. Secara umum, *confusion matrix* direpresentasikan sebagai matriks bujur sangkar ($n \times n$) dimana baris menyatakan label aktual dan kolom menyatakan hasil prediksi, dimana entri-entri-nya terdiri dari *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Adapun *confusion matrix* untuk segmentasi biner dan klasifikasi multikelas pada citra CXR dapat dilihat pada Tabel 2.1. dan 2.2 [62], [98].

Tabel 2.1. *Confusion matrix* untuk segmentasi paru-paru pada citra CXR

<i>Ground Truth</i>	Hasil Segmentasi	
	Paru-paru	<i>Background</i>
Paru-paru	TP	FN
<i>Background</i>	FP	TN

Tabel 2.2. *Confusion matrix* multikelas untuk klasifikasi penyakit *COVID-19*

Aktual	Hasil Klasifikasi			
	Normal	<i>COVID-19</i>	<i>Pneumonia</i>	<i>Lung Opacity</i>
Normal	TN	FP	TN	TN
<i>COVID-19</i>	FN	TP	FN	FN
<i>Pneumonia</i>	TN	FP	TN	TN
<i>Lung Opacity</i>	TN	FP	TN	TN

Berdasarkan Tabel 2.1 dan 2.2 nilai TP menunjukkan jumlah citra/pikselyang termasuk dalam kelas positif yang diprediksi dengan benar, nilai TN menunjukkan

jumlah citra/piksel yang termasuk kelas negatif yang diprediksi dengan benar, nilai FN menunjukkan jumlah citra/piksel yang termasuk kelas positif namun diprediksi sebagai kelas negatif, dan nilai FP menunjukkan jumlah citra/piksel yang termasuk dalam kelas negatif namun diprediksi sebagai kelas positif [94], [99].

Nilai TP, TN, FP, dan FN pada *confusion matrix* dapat digunakan untuk menghitung kinerja metode segmentasi dan klasifikasi seperti akurasi, presisi, *recall*, dan *F1-Score*. Akurasi merupakan ukuran untuk melihat keakuratan arsitektur berdasarkan rasio jumlah prediksi yang benar terhadap jumlah total prediksi. Presisi merupakan ukuran untuk melihat kemampuan arsitektur dalam memprediksi kelas negatif dengan benar (TN). *Recall* merupakan ukuran untuk melihat kemampuan arsitektur dalam memprediksi kelas positif dengan benar (TP). *F1-Score* merupakan ukuran untuk melihat kecocokan antara label aktual dengan hasil prediksi dengan menghitung rata-rata harmonik antara presisi dan *recall* [62], [76], [94]. Secara matematis, untuk menghitung nilai akurasi, presisi, *recall* dan *F1 Score* dapat menggunakan Persamaan (2.20), (2.21), (2.22), dan (2.23) [62], [100].

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.20)$$

$$Presisi = \frac{TP}{TP + FP} \times 100\% \quad (2.21)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.22)$$

$$F1-Score = \frac{2TP}{2TP + FP + FN} \times 100\% \quad (2.23)$$

Ukuran kinerja suatu algoritma dapat dikategorikan ke dalam 5 kriteria yang dapat dilihat pada Tabel 2.3. sebagai acuan kesuksesan algoritma dalam melakukan segmentasi dan klasifikasi [101].

Tabel 2.3. Kategori nilai kinerja arsitektur

Nilai Kinerja (NK) (%)	Kategori
$NK > 90$	Sangat Baik
$80 < NK \leq 90$	Baik
$70 < NK \leq 80$	Cukup
$60 < NK \leq 70$	Kurang baik
$NK \leq 60$	Gagal

Pada Tabel 2.3. Nilai Kinerja (NK) dapat digunakan untuk nilai akurasi, presisi, *recall* dan *F1-score*. Jika nilai kinerja yang diperoleh lebih dari 90% maka nilai tersebut dikategorikan sangat baik. Jika nilai kinerja lebih dari 80% dan kurang dari sama dengan 90% maka nilai tersebut dikategorikan baik. Jika nilai kinerja lebih dari 70% dan kurang dari sama dengan 80% maka nilai tersebut dikategorikan cukup baik. Jika nilai kinerja lebih dari 60% dan kurang dari sama dengan 70% maka nilai tersebut dikategorikan kurang baik. Jika nilai kinerja kurang dari sama dengan 60% maka nilai tersebut dikategorikan gagal.

2.16. State of the Art

Penyakit *COVID-19* adalah penyakit menular pada saluran pernapasan yang disebabkan oleh virus SARS-CoV-2. Citra *Chest X-ray* (CXR) dapat digunakan sebagai alternatif untuk mendiagnosa penyakit ini, namun memerlukan tingkat ketelitian yang tinggi. Peningkatan kualitas citra CXR telah banyak dilakukan oleh beberapa penelitian terdahulu yang dapat dilihat pada Tabel 2.4.

Berdasarkan Tabel 2.4, metode peningkatan kualitas citra melalui Morfologi *Opening*, *Median Filter*, dan *Gaussian Filter* bertujuan untuk menghasilkan citra CXR yang lebih jelas dan tajam sehingga memudahkan tenaga medis dalam melihat kelainan pada paru-paru pasien. Selanjutnya dilakukan segmentasi citra CXR untuk memfokuskan pada bagian paru-parunya saja. Metode yang berkembang saat ini untuk segmentasi citra adalah *Convolutional Neural Network* (CNN), khususnya arsitektur *U-Net*. Arsitektur *U-Net* efektif digunakan untuk segmentasi citra berbasis piksel. Namun, selama proses *training*, terkadang terjadi perubahan

distribusi fitur input yang dapat menghambat kecepatan pelatihan dan kinerja. Untuk mengatasi masalah ini, diperlukan metode *Batch Normalization* (BN) yang akan menormalkan peta fitur disemua lapisan arsitektur.

Selain segmentasi citra, metode CNN juga banyak digunakan untuk klasifikasi citra. Beberapa arsitektur CNN yang populer adalah *ResNet*, *EfficientNet*, dan *Inception-v3*. *ResNet* menggunakan *skip connection* untuk menangani masalah *vanishing gradient*, sedangkan *EfficientNet* memiliki keunggulan dalam penskalaan dimensi secara seragam. *Inception-v3* menggunakan konvolusi kecil yang menumpuk untuk menghindari *overfitting*. Setiap arsitektur CNN memiliki kelebihan dan kekurangan dalam klasifikasi citra.

Untuk meningkatkan kinerja klasifikasi, metode *ensemble learning* dapat digunakan. *Ensemble learning* menggabungkan hasil prediksi dari berbagai arsitektur atau metode klasifikasi tunggal menjadi model baru yang lebih akurat. Metode *weighted voting* pada *ensemble learning* memberikan bobot yang berbeda untuk pengklasifikasi berdasarkan kriteria tertentu, sehingga meminimalkan tingkat kesalahan prediksi klasifikasi tunggal. Metode *ensemble learning* baru yang diusulkan adalah ELREI (*Ensemble Learning* dari *ResNet*, *EfficientNet*, dan *Inception-v3*) dengan *weighted voting* untuk klasifikasi penyakit paru-paru berdasarkan citra CXR. Metode ELREI berjalan pada setiap *epoch* pada tahap *training*, sehingga bobot hasil ELREI pada setiap *epoch* pada data *training* dan data validasi dapat dicek pada hasil performa untuk menangani *overfitting*.⁴ Beberapa penelitian terdahulu telah melakukan segmentasi dan klasifikasi citra yang dapat dilihat seperti pada Tabel 2.5.

Tabel 2.4. Beberapa penelitian terdahulu yang telah melakukan peningkatan kualitas citra

No	Tahun	Penulis	Metode	Hasil dan Kelemahan
1	2018	Susanto [102]	Menggunakan HE, CLAHE, <i>Contrast Stretching</i> (CS), dan <i>Sharpened Image</i> (SI) dikombinasikan dengan <i>Median Filter</i>	Pengabungan metode <i>Median Filter</i> menghasilkan nilai rata-rata MSE sebesar 81,78 serta nilai rata-rata PSNR sebesar 29,04.
2	2020	Umamaheswari & Geetha [103]	Menggunakan metode <i>Gamma Corrected Gaussian Filtering</i> (GCGF)	Menghasilkan nilai rata-rata PSNR sebesar 24,61 dan nilai MSE rata-rata sebesar 229,92. Hasil penelitian ini tergolong rendah karena nilai PSNR masih dibawah 30 dan nilai MSE yang terlalu tinggi serta tidak menghitung nilai SSIM.
3	2020	Roy & Maity [104]	Menggunakan kombinasi metode <i>Wiener Filter</i> dan <i>Poisson Noise</i>	Menghasilkan nilai rata-rata PSNR sebesar 33, nilai rata-rata MSE sebesar 29, dan nilai rata-rata SSIM sebesar 0,7876. Hasil penelitian ini tergolong cukup rendah karena nilai rata-rata PSNR masih dibawah 35 dan nilai SSIM masih dibawah 80.
4	2022	Basset et al [105]	Menggunakan metode <i>Type-2 Neurosophic Set</i>	Menghasilkan nilai rata-rata PSNR sebesar 28,58, nilai rata-rata MSE sebesar 23,6, dan nilai rata-rata SSIM sebesar 0,9. Nilai rata-rata PSNR pada penelitian ini cukup rendah karena dibawah 30.
5	2022	Kesuma et al [106] ¹	Membandingkan antara Metode <i>Morfologi Opening dan Median Filter</i> serta <i>Morfologi Opening dan Gaussian Filter</i>	Hasil penelitian pada <i>Morfologi Opening dan Median Filter</i> yaitu PSNR sebesar 39,187, MSE 22,252, dan SSIM 0,952. Sedangkan pada <i>Morfologi Opening dan Gaussian Filter</i> yaitu PSNR sebesar 38,717, MSE 23,917, dan SSIM 0,956.

2

¹ Kesuma, L. I., Ermatita, Erwin, Sari, P., & Purabaya, R. H. (2022). Improved Chest X - Ray Image Quality Using Median and Gaussian Filter Methods. 2022 *International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. (Lampiran 1)

Tabel 2.5. Beberapa penelitian terdahulu yang telah melakukan segmentasi dan klasifikasi citra

No	Tahun	Penulis	Metode	Hasil dan Kelemahan
Segmentasi menggunakan Metode U-Net Batch Normalization				
1	2018	Tong et al. [107]	Menerapkan modifikasi arsitektur U-Net dengan <i>batch normalization</i> dan <i>residual connection</i> untuk segmentasi modul paru-paru pada citra CT Scan paru-paru	Menghasilkan nilai <i>IoU</i> tertinggi dibandingkan dengan arsitektur U-Net dasar yaitu sebesar 73,6%. Tetapi tidak menjelaskan hasil akurasi, sensitivitas, spesifisitas, dan <i>F1-Score</i> sebagai evaluasi kinerja segmentasi citra lainnya
2	2020	Smith et al. [108]	Menggunakan modifikasi arsitektur U-Net menggunakan <i>Group normalization</i> untuk segmentasi akar tanaman pada citra tanah	Hasil akurasi, sensitivitas, dan <i>F1-Score</i> masing-masing sebesar 99,7%; 74,8%; dan 70,1%. Namun, tidak menjelaskan hasil spesifisitas, dan <i>IoU</i> sebagai evaluasi kinerja segmentasi citra lainnya
Klasifikasi menggunakan Metode Konvensional				
3	2021	Mohammed et al. [67]	Menggunakan <i>machine learning (Random Forest dan CN2 Rule Inducer)</i>	<ul style="list-style-type: none"> ➢ Rata-rata nilai ukuran kinerja pada keempat metode yang diusulkan masih dibawah 90% kecuali hasil sensitivitas pada metode Xception. ➢ Tidak menjelaskan hasil <i>F1-Score</i> sebagai evaluasi kinerja klasifikasi citra lainnya.
4	2021	Akinuwesi et al. [109]	Menggunakan metode <i>Logistic Regression (LR), Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), Multilayer Perceptron (MLP), Fuzzy Cognitive Map (FCM) dan Deep Neural Network (DNN)</i>	<ul style="list-style-type: none"> ➢ Rata-rata nilai ukuran kinerja pada ketujuh metode yang diusulkan masih dibawah 85%.
5	2019	Kadry et al. [110]	Menggunakan metode <i>Naive Bayes (NB), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF) and Support Vector Machine with linear kernel (SVM)</i>	<ul style="list-style-type: none"> ➢ Rata-rata nilai kinerja evaluasi yang diperoleh masih dibawah 84%, selain itu nilai performa kinerja metode lainnya seperti <i>F1-score</i> tidak dihitung.
6	2020	Vinod & Prabaharan [111]	Menggunakan metode <i>Decision Tree</i>	<ul style="list-style-type: none"> ➢ Nilai kinerja evaluasi yang dihasilkan pada dataset <i>CT-Scan</i> untuk Presisi dan <i>F-1 Score</i> masih cukup rendah yaitu masih dibawah 85%, selain itu nilai kinerja lain seperti Akurasi tidak dihitung.

7	2021	Sharma et al. [112]	Menggunakan metode <i>Improved Random Forest (IRF)</i> , <i>Support Vector Machine (SVM)</i> , dan <i>Random Forest (RF)</i>	<ul style="list-style-type: none"> ➢ Nilai evaluasi kinerja masih kurang baik dalam mengklasifikasikan penyakit <i>COVID-19</i> karena nilai yang diperoleh rata-rata masih dibawah 77%. ➢ Tidak menghitung nilai evaluasi kinerja lainnya seperti akurasi.
Klasifikasi menggunakan Metode ResNet				
8	2021	Bharati et al. [70]	Mengusulkan algoritma baru yaitu <i>Optimized Residual Network for COVID-19 (CO-ResNet)</i>	<ul style="list-style-type: none"> ➢ Jumlah data citra untuk setiap kelas tidak seimbang ➢ Tidak menjelaskan hasil sensitivitas, spesifisitas, dan <i>F1-Score</i> sebagai evaluasi kinerja klasifikasi citra.
9	2020	Wu et al. [113]	Menggunakan metode <i>ResNet</i> berbasis <i>multi-view fusion</i>	<ul style="list-style-type: none"> ➢ Nilai evaluasi kinerja masih kurang baik dalam mengklasifikasikan penyakit <i>COVID-19</i> karena nilai yang diperoleh rata-rata masih dibawah 82%. ➢ Tidak menghitung nilai evaluasi kinerja lainnya seperti <i>F1-Score</i>
10	2021	Shah et al. [114]	Menggunakan metode <i>Cnet-10</i> , <i>Dense-169</i> , <i>VGG-16</i> , <i>ResNet-50</i> , <i>Inception-v3</i> , dan <i>VGG-19</i>	<p>Nilai akurasi untuk metode <i>Inception-v3</i> dan <i>ResNet</i> masih kurang baik mengklasifikasikan penyakit <i>COVID-19</i> karena nilai akurasinya dibawah 60%. Selain itu nilai evaluasi kinerja lainnya seperti sensitivitas, spesifisitas, <i>F1-Score</i> tidak dihitung.</p>
Klasifikasi menggunakan Metode Inception-v3				
11	2021	Albahli et al. [115]	Menggunakan metode <i>Densenet</i> , <i>Inception-v3</i> , dan <i>Inception-ResNetv4</i>	<p>Hanya menghitung nilai akurasi dengan menghasilkan nilai masih dibawah 90%. Selain itu tidak menghitung nilai evaluasi kinerja lainnya seperti sensitivitas, spesifisitas, dan <i>F1-Score</i>.</p>
12	2021	Dimas et al. [116]	Membandingkan hasil evaluasi kinerja antara metode CNN dengan <i>Inception-v3</i> dalam mengklasifikasikan citra CXR ke dalam dua kelas yaitu positif <i>COVID-19</i> dan normal	<p>Hasil penelitian menunjukkan bahwa model <i>Inception-v3</i> mencapai nilai akurasi, sensitivitas lebih baik dari pada model CNN masing-masing sebesar 82,41%; 76,24%, sedangkan pada model CNN masing-masing sebesar 80,90%; 72,28%. Selain itu tidak menjelaskan hasil spesifisitas, dan <i>F1-Score</i> sebagai evaluasi.</p>

Klasifikasi menggunakan Metode <i>EfficientNet</i>				
13	2021	Monshi et al. [60]	Menerapkan model <i>CovidXrayNet</i> yang didasarkan dari arsitektur <i>EfficientNet-B0</i> dalam mengklasifikasikan citra CXR	Hasil kinerja metode yang diusulkan untuk dataset <i>COVIDcxr</i> nilai akurasi dan <i>F1-Score</i> masing-masing sebesar 88,54% dan 88,062%. Selain itu tidak menjelaskan hasil spesifisitas dalam nilai evaluasi.
14	2020	Anwar & Zakir [117]	Menggunakan metode <i>Efficient-Net</i> dengan 3 <i>learning rate</i> yaitu <i>Plateau</i> , <i>Cyclic</i> , dan <i>Constant</i>	Hasil kinerja evaluasi yang diperoleh ada 4 nilai akurasi, <i>F-1 Score</i> masih dibawah 90%. Selain itu tidak menghitung nilai evaluasi kinerja lainnya seperti sensitivitas, dan spesifisitas.
Klasifikasi menggunakan Metode <i>Ensemble Learning</i>				
15	2021	Paladini et al. [25]	Membandingkan dua pendekatan metode <i>ensemble</i> yaitu <i>Mean-Ensemble-CNNs</i> dan <i>NN-Ensemble-CNN</i> untuk mengklasifikasikan tipe jaringan <i>Colorectal Cancer</i> (CRC). Model CNN yang digunakan yaitu <i>ResNet101</i> , <i>ResNeXt50</i> , <i>Inception-v3</i> , dan <i>DenseNet161</i>	Hasil nilai akurasi dan <i>F1-Score</i> menggunakan <i>Mean-Ensemble-CNNs</i> yaitu 86,97% dan 86,99%; sedangkan menggunakan <i>NN-Ensemble-CNN</i> sebesar 87,26% dan 87,27%. Tidak menjelaskan hasil sensitivitas, dan spesifisitas sebagai evaluasi kinerja. Selain itu, <i>ensemble</i> yang dilakukan hanya menggabungkan hasil dari klasifikasi tunggal tanpa dilakukan <i>training</i> .
16	2021	Pratiwi et al. [118]	Menerapkan metode <i>ensemble</i> menggunakan tiga arsitektur <i>DCNNs</i> diantaranya <i>Inception-v3</i> , <i>Inception ResNetV2</i> , dan <i>DenseNet201</i> untuk klasifikasi lesi kulit	Hasil nilai akurasi, sensitivitas, spesifisitas, dan <i>F1-Score</i> berturut-turut sebesar 97,23%; 90,12%; 97,73%; dan 85,01. <i>Ensemble</i> yang dilakukan hanya menggabungkan hasil dari klasifikasi tunggal tanpa dilakukan <i>training</i> .
17	2021	Tang et al. [119]	Menggunakan metode <i>Ensemble Deep Learning</i>	Menghasilkan nilai akurasi sebesar 97,8%, presisi sebesar 97,83%, sensitivitas sebesar 97,83, dan <i>F1-score</i> sebesar 97,77%. <i>Ensemble</i> yang dilakukan hanya menggabungkan hasil dari klasifikasi tunggal tanpa dilakukan <i>training</i> .
18	2021	Shrivastava et al. [120]	Menggunakan metode kombinasi <i>DCNN Ensemble</i>	Menghasilkan nilai akurasi sebesar 97,47% dan sensitivitas sebesar 98,18%. Tidak menghitung nilai spesifisitas dan <i>F1-Score</i> . Selain itu, <i>ensemble</i> yang dilakukan hanya menggabungkan hasil dari klasifikasi tunggal tanpa dilakukan <i>training</i> .

19	2022	Ho & Gwak [121]	Menggunakan metode kombinasi antara <i>Ensemble</i> , <i>ResNet18</i> dan <i>DenseNet121</i>	Menghasilkan nilai akurasi sebesar 94,1%, presisi sebesar 94,5%, sensitivitas sebesar 94,1%, dan <i>F1-score</i> sebesar 94%. <i>Ensemble</i> yang dilakukan hanya menggabungkan hasil dari klasifikasi tunggal tanpa dilakukan <i>training</i> .
20	2022	Visuna et al. [122]	Menggunakan kombinasi metode CNN dan <i>ensemble learning</i>	Menghasilkan nilai akurasi sebesar 98%, presisi sebesar 98%, sensitivitas sebesar 98,25%, dan <i>F1-score</i> sebesar 97,75%. <i>Ensemble</i> yang dilakukan hanya menggabungkan hasil dari klasifikasi tunggal tanpa dilakukan <i>training</i> .
21	2023	Kesuma et al. [123] ²	Menggunakan metode <i>Ensemble Learning</i> dari <i>ResNet</i> , <i>EfficientNet</i> , dan <i>Inception-v3</i> (ELREI)	Menghasilkan nilai akurasi 99%, presisi 98,75%, <i>recall</i> 98,75%, dan <i>F1-Score</i> 99%. Kelebihan dari penelitian ini yaitu Metode ELREI bekerja pada setiap <i>epoch</i> pada tahap <i>training</i> , bukan pada hasil akhir <i>training</i> .

² Kesuma, L. I., Ermatita, & Erwin. (2023). ELREI : Ensemble Learning of ResNet , EfficientNet , and Inception-v3 for Lung Disease Classification based on Chest X-Ray Image. *International Journal of Intelligent Engineering and Systems*, 16(5), 149–161. (Lampiran 2)

BAB III

METODOLOGI PENELITIAN

3.1. Tempat Penelitian

Penelitian ini dilakukan di Program Studi Ilmu Teknik Fakultas Teknik Universitas Sriwijaya. Sumber referensi yang digunakan diperoleh dari Perpustakaan Universitas Sriwijaya dan penelusuran internet.

3.2. Waktu Penelitian

Penelitian ini dilaksanakan selama 3 (tiga) tahun, yaitu dari Tahun 2020 hingga Tahun 2023.

3.3. Alat

Penelitian ini dilakukan menggunakan *software Jupyter Notebook* pada aplikasi *Anaconda* yang berbasis bahasa pemrograman *Python*.

3.4. Objek Penelitian

Pada penelitian ini data citra yang digunakan yaitu Citra *Chest X-ray (CXR)* yang merupakan citra yang dihasilkan melalui proses teknik *rontgen X-ray* pada bagian dada. Penelitian ini berfokus pada penggunaan teknik *ensemble learning* untuk meningkatkan kinerja klasifikasi penyakit paru-paru. Teknik ini menggabungkan tiga model *deep learning* dengan arsitektur yang berbeda, yaitu *ResNet*, *EfficientNet*, dan *Inception-v3*. Setiap arsitektur akan memproses citra CXR dan menghasilkan prediksi klasifikasi terkait penyakit paru-paru. Selanjutnya, *output* dari ketiga model ini digabungkan menggunakan *weighted voting* melalui teknik *ensemble learning* untuk menghasilkan prediksi akhir yang lebih akurat.

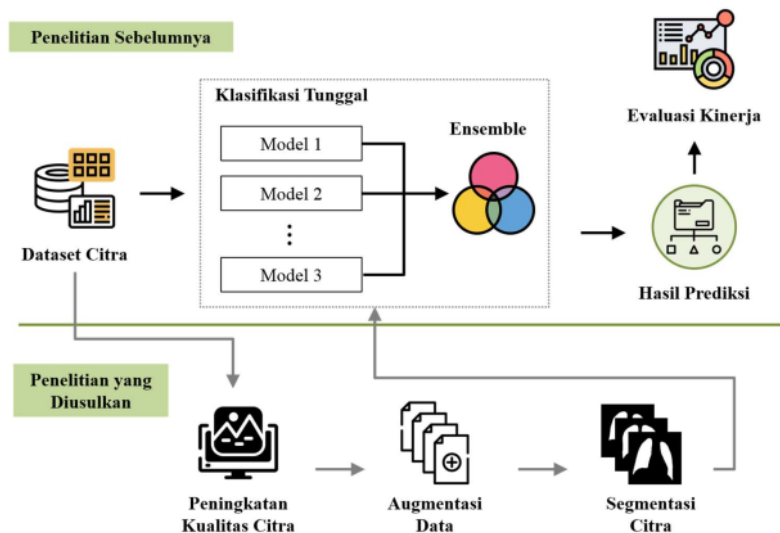
3.5. Pengumpulan Data

Data pada penelitian ini berasal dari sumber data sekunder, yaitu dataset *COVID-19 Radiography Database* yang diperoleh melalui laman <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>.

Dataset ini merupakan kumpulan citra *Chest X-Ray* (CXR) dari proyek penelitian Qatar University, Doha, Qatar, dan University of Dhaka, Bangladesh yang berkolaborasi bersama Pakistan dan Malaysia. Terdapat 4 kelas yang digunakan pada penelitian ini yaitu 3.616 citra *COVID-19*, 10.192 citra normal, 6.012 citra *lung opacity (non-COVID)*, dan 1.345 citra *pneumonia*.

3.6. Kerangka Kerja Penelitian

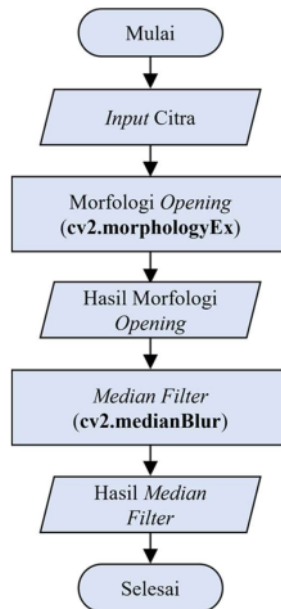
Kerangka kerja pada penelitian *multiclass classification* untuk penyakit berdasarkan citra CXR paru-paru dengan menerapkan *ensemble learning* dapat dilihat pada Gambar 3.1. Pada bagian atas menunjukkan penerapan *ensemble learning* yang umum digunakan berdasarkan literatur dalam beberapa tahun terakhir, sedangkan pada bagian bawah menunjukkan penerapan *ensemble learning* yang diusulkan dengan mengimplementasikan beberapa tahap sebelum memasuki proses klasifikasi citra. Tahap-tahap tersebut yaitu perbaikan kualitas citra untuk memperoleh kualitas citra yang lebih baik sehingga mempermudah proses pengklasifikasian citra, augmentasi citra untuk memperbanyak data latih dengan mentransformasi citra asli pada orientasi yang berbeda, dan segmentasi citra CXR untuk memisahkan objek paru-paru dari objek lainnya.



Gambar 3.1. Kerangka kerja penelitian

3.5.1. Peningkatan Kualitas Citra

Proses peningkatan kualitas citra merupakan tahapan awal dalam proses pengolahan data setelah pengumpulan data yang bertujuan untuk meningkatkan kualitas citra. Tahapan peningkatan kualitas citra pada penelitian ini disajikan dalam bentuk *flowchart* yang dapat dilihat pada Gambar 3.2.



Gambar 3.2. Tahapan Peningkatan Kualitas Citra

Proses peningkatan kualitas citra yang terlihat pada Gambar 3.2 dilakukan dengan langkah-langkah berikut:

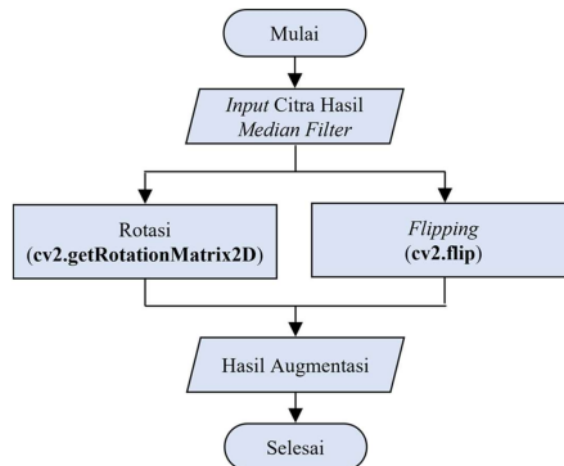
1. Citra CXR yang diperoleh dari dataset *COVID-19 Radiography* dijadikan sebagai *input* citra.
2. Citra yang telah di *input* kemudian dilakukan operasi Morfologi *Opening* dengan menggunakan fungsi `cv2.morphologyEx`. Operasi ini bertujuan untuk memperbaiki kontur dan menghilangkan objek-objek tipis yang ada pada citra, sehingga menghasilkan citra yang lebih halus dan bersih dengan tepi yang lebih tajam.
3. Citra hasil dari proses Morfologi *Opening* selanjutnya dilakukan proses *Median Filter* menggunakan fungsi `cv2.medianBlur`. *Median Filter* bertujuan untuk

mengurangi *noise* pada citra dengan memperhatikan distribusi intensitas piksel dalam citra, khususnya piksel-piksel yang memiliki nilai ekstrem.

4. Citra hasil *Median Filter* dijadikan sebagai *Output* sekaligus dapat digunakan untuk proses selanjutnya.

3.5.2. Augmentasi Data

Tahapan augmentasi data bertujuan untuk memperbanyak data latih sehingga setiap kelas akan memiliki jumlah data yang seimbang. Data yang telah dilakukan perbaikan kualitas citra diperbanyak menggunakan teknik transformasi rotasi dan *flipping* yang disajikan dalam bentuk *flowchart* yang dapat dilihat pada Gambar 3.3.



Gambar 3.3. Tahapan Augmentasi Data

Tahapan augmentasi data yang terlihat pada Gambar 3.3 dilakukan dengan langkah-langkah berikut:

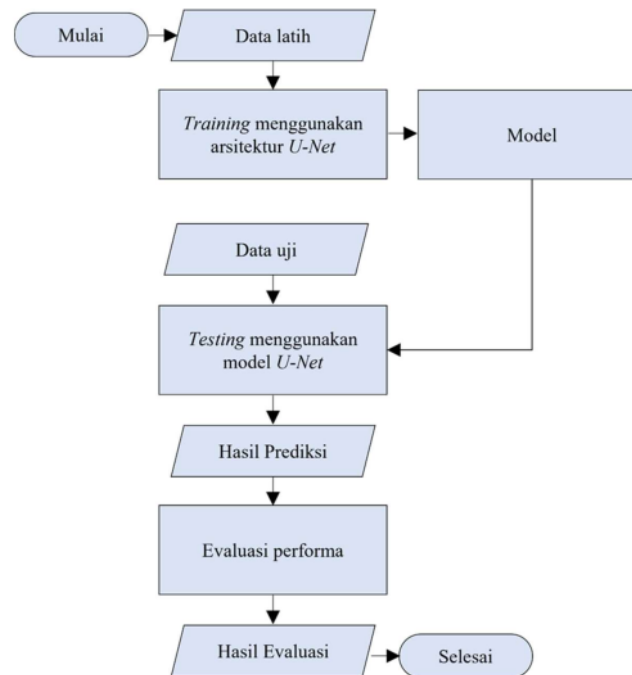
1. Citra hasil *Median Filter* dijadikan sebagai *input* pada proses augmentasi data.
2. Citra yang telah di *input* kemudian dilakukan proses augmentasi dengan menggunakan dua teknik transformasi yaitu rotasi dan *flipping*. Teknik transformasi rotasi menggunakan fungsi `cv2.getRotationMatrix2D` dengan mengubah sudut antara 1° hingga 359° pada citra. Sedangkan teknik transformasi *flipping* dilakukan dengan memutar gambar secara horizontal dan

vertikal menggunakan fungsi `cv2.flip`. Kedua teknik ini memberikan variasi baru pada dataset CXR dengan mengubah arah dan sudut orientasi citra sehingga menciptakan variasi yang lebih beragam.

3. Citra hasil augmentasi data dijadikan sebagai *Output* sekaligus dapat digunakan untuk proses selanjutnya.

3.5.3. Segmentasi Citra

Pada tahapan ini, citra hasil augmentasi dilakukan proses segmentasi yang bertujuan untuk membagi citra menjadi dua bagian, yaitu paru-paru (*foreground*) sebagai piksel berwarna putih dan bagian bukan paru-paru (*background*) sebagai piksel berwarna hitam. Segmentasi citra paru-paru dapat membantu mendeteksi penyakit atau kelainan pada paru-paru. Proses segmentasi citra paru-paru terdiri dari dua tahap utama, yaitu *training* data dan *testing* data dengan menggunakan arsitektur *U-Net*. Tahapan segmentasi citra CXR pada penelitian ini disajikan dalam bentuk *flowchart* seperti pada Gambar 3.4.



Gambar 3.4. Tahapan Segmentasi Citra

Pada Gambar 3.4 terlihat bahwa data yang dihasilkan dari proses augmentasi dibagi menjadi dua proses yaitu data latih dan data uji. Data latih digunakan untuk proses *training*, sedangkan data uji digunakan untuk proses *testing*.

- *Training* data

Training data dilakukan untuk membangun model dengan melatih arsitektur terhadap data latih agar dapat mengenali pola dan fitur-fitur yang tersedia pada data. Proses *training* data menggunakan arsitektur *U-Net* dilakukan dengan langkah-langkah sebagai berikut:

1. Inisialisasi parameter: jumlah *epoch* dan ukuran *batch size* untuk satu *epoch*.
2. Lakukan inisialisasi nilai bobot pada *epoch* pertama.
3. *Split* data: Bagi data hasil augmentasi menjadi dua bagian yaitu data latih dan data validasi. Selanjutnya data tersebut diproses ke dalam arsitektur *U-Net*.
4. Jalur *encoder*:
 - a. *Input* citra dilakukan operasi *convolutional layer* dengan kernel berukuran 3×3 yang diikuti dengan fungsi aktivasi ReLU.
 - b. Hasil dari operasi konvolusi (*feature maps*) selanjutnya dilakukan normalisasi menggunakan *batch normalization*.
 - c. Kemudian lakukan operasi *max pooling* dengan ukuran 2×2 untuk mengurangi dimensi *feature maps*.
 - d. Ulangi langkah (a) sampai (c) sebanyak empat kali.
5. Jalur *decoder*:
 - a. Lakukan peningkatan ukuran *feature maps* menggunakan *transposed convolution* berukuran 2×2 .
 - b. Gabungkan ⁴*feature maps* hasil proses konvolusi pada jalur *encoder* dengan hasil *transposed convolution* pada jalur *decode* menggunakan *concatenate*.
 - c. Ulangi langkah (a) dan (b) sebanyak tiga kali.
6. Lakukan kembali proses *convolution layer* dengan kernel berukuran ⁴ 1×1 dan dilanjutkan dengan fungsi aktivasi *sigmoid*.
7. Hitung nilai ⁴*loss function binary cross entropy*. Jika nilai *loss function* untuk data validasi kurang dari nilai validasi *loss* pada *epoch* sebelumnya maka

bobot akan disimpan dan digunakan untuk *epoch* selanjutnya, jika tidak maka bobot akan diperbarui untuk *epoch* selanjutnya.

8. Ulangi langkah (2) sampai (7) hingga mencapai *epoch* terakhir.
9. Simpan bobot terakhir ke dalam model *U-Net* untuk digunakan pada saat *testing* data.

- *Testing* data

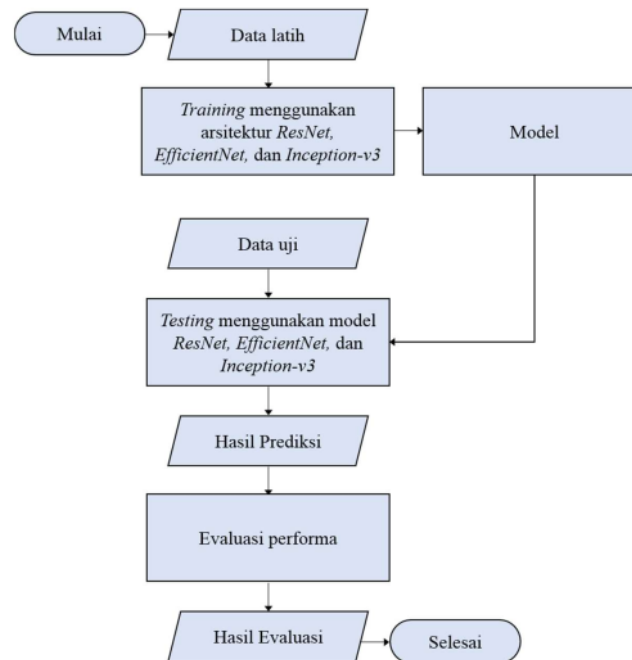
Testing data dilakukan untuk menguji model *U-Net* yang didapatkan dari proses *training* dengan menggunakan data *testing*. Hasil prediksi segmentasi menggunakan model *U-Net* menunjukkan bahwa piksel dengan nilai 1 (berwarna putih) merepresentasikan paru-paru, sementara piksel dengan nilai 0 (berwarna hitam) merepresentasikan *background*. Selanjutnya, nilai-nilai ini akan termuat dalam *confusion matrix* yang dapat digunakan untuk menghitung kinerja model *U-Net*.

- Evaluasi kinerja segmentasi

Evaluasi kinerja ini bertujuan untuk mengevaluasi sejauh mana model segmentasi yang diusulkan mampu melakukan segmentasi dengan baik. Pada penelitian ini, evaluasi kinerja yang dihitung yaitu nilai akurasi, presisi, *recall*, dan *F1-Score*.

3.5.4. Klasifikasi Citra

Pada tahapan ini dilakukan klasifikasi penyakit paru-paru menggunakan citra CXR yang telah disegmentasi. Penerapan metode *ensemble learning* dengan teknik *weighted voting* menggunakan arsitektur *ResNet*, *EfficientNet*, dan *Inception-v3* diterapkan untuk mengklasifikasikan citra ke dalam 4 kelas yaitu *COVID-19*, normal, *lung opacity (non-COVID)*, dan *pneumonia*. Tahapan klasifikasi citra pada penelitian ini disajikan dalam bentuk *flowchart* yang dapat dilihat pada Gambar 3.5.



Gambar 3.5. Tahapan Klasifikasi Citra

Pada Gambar 3.5 terlihat bahwa data yang dihasilkan dari proses segmentasi dibagi menjadi dua yaitu data latih dan data uji. Data latih digunakan untuk proses *training*, sedangkan data uji digunakan untuk proses *testing*. Selanjutnya dilakukan proses *training* menggunakan arsitektur *ResNet*, *EfficientNet*, dan *Inception-v3*. Hasil *training* berupa model yang akan diuji pada tahap *testing* menggunakan data uji. Proses *testing* akan menghasilkan nilai evaluasi dan prediksi kedalam 4 kelas yaitu *COVID-19*, *normal*, *lung opacity*, dan *pneumonia*.

3.5.4.1. *Training Menggunakan Arsitektur ResNet*

Proses *training* data menggunakan arsitektur *ResNet* dilakukan dengan langkah-langkah sebagai berikut:

1. Inisialisasi parameter: jumlah *epoch* dan ukuran *batch size* untuk satu *epoch*.
2. Inisialisasi nilai bobot pada *epoch* pertama.

3. *Split* data: Bagi data hasil segmentasi menjadi dua bagian yaitu data latih dan data validasi. Selanjutnya data tersebut diproses ke dalam arsitektur *ResNet*.
4. *Input* citra dilakukan operasi *convolutional layer* dengan kernel berukuran 7×7 yang diikuti dengan fungsi aktivasi ReLU.
5. Hasil dari operasi konvolusi (*feature maps*) selanjutnya dilakukan normalisasi menggunakan *batch normalization*.
6. Kemudian lakukan operasi *max pooling* dengan ukuran 3×3 untuk mengurangi dimensi *feature maps*.
7. Selanjutnya, hasil operasi *max pooling* diproses ke dalam *conv_block*. Lakukan seperti langkah (4) sampai (6) sebanyak tiga kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 , 3×3 , dan 1×1 . Namun, pada saat proses konvolusi ketiga tidak disertai dengan fungsi aktivasi *ReLU*.
8. Hasil operasi *max pooling* juga diproses ke dalam jalur *residual connection*, dimana dilakukan proses *convolutional layer* dengan kernel berukuran 1×1 yang diikuti dengan operasi *batch normalization*.
9. Lakukan penambahan hasil *feature maps* dari langkah (7) dan (8) menggunakan operasi *add*. Selanjutnya hasil tersebut dilakukan operasi fungsi aktivasi ReLU.
10. Hasil dari fungsi aktivasi ReLU diproses ke dalam *identity_block*, dimana proses ini dilakukan seperti pada langkah (7).
11. Lakukan penambahan hasil *feature maps* dari langkah (9) dan (10) menggunakan operasi *add*.
12. Ulangi langkah (10) dan (11) dengan menggunakan hasil *feature maps* dari langkah (11).
13. Ulangi langkah (6) sampai (11) sebanyak 3 kali, namun pada *identity_blok* diulang secara berturut-turut sebanyak tiga kali, lima kali, dan dua kali.
14. Lakukan pengurangan ukuran *feature maps* menggunakan *global average pooling*. Selanjutnya hasil tersebut diproses ke dalam *dense layer*, kemudian disubstitusikan ke dalam fungsi aktivasi *sigmoid*.

15. Hitung *loss function categorical cross entropy*. Jika nilai *loss function* untuk data validasi kurang dari nilai validasi *loss* pada *epoch* sebelumnya maka bobot akan disimpan dan digunakan untuk *epoch* selanjutnya, jika tidak maka bobot akan diperbarui untuk *epoch* selanjutnya.
16. Ulangi langkah (2) sampai (15) hingga mencapai *epoch* terakhir.
17. Simpan bobot terakhir ke dalam model *ResNet* untuk digunakan pada saat *testing* data.

3.5.4.2. Training Menggunakan Arsitektur *EfficientNet*

Proses *training* data menggunakan arsitektur *EfficientNet* dilakukan dengan langkah-langkah sebagai berikut:

1. Inisialisasi parameter: jumlah *epoch* dan ukuran *batch size* untuk satu *epoch*.
2. Inisialisasi nilai bobot pada *epoch* pertama.
3. *Split* data: Bagi data hasil segmentasi menjadi dua bagian yaitu data latih dan data validasi. Selanjutnya data tersebut diproses ke dalam arsitektur *EfficientNet*.
4. *Input* citra dilakukan operasi *convolutional layer* dengan kernel berukuran 3×3 yang diikuti dengan *batch normalization* dan fungsi aktivasi ReLU.
5. Selanjutnya hasil dari langkah (4) diproses ke dalam blok *MBConv1* sebanyak satu kali. Pada blok *MBConv1* ini, *feature maps* diproses ke dalam *depthwise convolution* dengan kernel 3×3 , kemudian dilakukan normalisasi menggunakan *batch normalization* yang diikuti dengan fungsi aktivasi ReLU.
6. Selanjutnya, lakukan pengurangan ukuran *feature maps* menggunakan *global average pooling*. Kemudian, dilanjutkan kembali proses *convolution layer* dengan kernel 1×1 sebanyak dua kali.
7. Lakukan proses perkalian antara hasil dari langkah (5) dan (6). Kemudian, dilanjutkan ke dalam proses *convolution layer* dengan kernel 1×1 yang diikuti proses *batch normalization*.

8. Selanjutnya hasil dari langkah (7) diproses ke dalam blok *MBCConv6* sebanyak dua kali. Pada blok *MBCConv6* ini, dilakukan seperti pada langkah (5) sampai (7) dimana pada proses *depthwise convolution* digunakan kernel berukuran 3×3 .
9. Hasil pada langkah (8) diproses ke dalam lapisan *dropout* dan dilanjutkan penambahan *feature maps* hasil pada langkah (7) dan (8).
10. Lakukan kembali proses blok *MBCConv6* seperti langkah (8) dan (9) dengan kernel *depthwise convolution* berukuran 5×5 sebanyak dua kali.
11. Lakukan kembali proses blok *MBCConv6* seperti langkah (8) dan (9) dengan kernel *depthwise convolution* berukuran 3×3 sebanyak tiga kali.
12. Lakukan kembali proses blok *MBCConv6* seperti langkah (8) dan (9) dengan kernel *depthwise convolution* berukuran 5×5 sebanyak tujuh kali.
13. Lakukan kembali proses blok *MBCConv6* seperti langkah (8) dan (9) dengan kernel *depthwise convolution* berukuran 3×3 sebanyak satu kali.
14. Lakukan kembali proses *convolutional layer* dengan kernel berukuran 1×1 , kemudian dilanjutkan dengan operasi *batch normalization* yang diikuti fungsi aktivasi ReLU.
15. Lakukan pengurangan ukuran *feature maps* menggunakan *global average pooling*.
16. Hasil dari langkah (14) diproses ke dalam *dense layer*, kemudian disubstitusikan ke dalam fungsi aktivasi *softmax*.
17. Hitung *loss function categorical cross entropy*. Jika nilai *loss function* untuk data validasi kurang dari nilai validasi *loss* pada *epoch* sebelumnya maka bobot akan disimpan dan digunakan untuk *epoch* selanjutnya, jika tidak maka bobot akan diperbarui untuk *epoch* selanjutnya.
18. Ulangi langkah (2) sampai (17) hingga mencapai *epoch* terakhir.
19. Simpan bobot terakhir ke dalam model *EfficientNet* untuk digunakan pada saat *testing data*.

3.5.4.3. *Training Menggunakan Arsitektur Inception-v3*

Proses *training* data menggunakan arsitektur *Inception-v3* dilakukan dengan langkah-langkah sebagai berikut:

1. Inisialisasi parameter: jumlah *epoch* dan ukuran *batch size* untuk satu *epoch*.
2. Inisialisasi nilai bobot pada *epoch* pertama.
3. *Split* data: Bagi data hasil segmentasi menjadi dua bagian yaitu data latih dan data validasi. Selanjutnya data tersebut diproses ke dalam arsitektur *Inception-v3*.
4. *Input* citra dilakukan operasi *convolutional layer* dengan kernel berukuran 3×3 yang diikuti dengan operasi *batch normalization* dan fungsi aktivasi ReLU.
5. Lakukan langkah (4) sebanyak dua kali.
6. Lakukan pengurangan ukuran *feature maps* menggunakan *max pooling*.
7. Ulangi langkah (5) dan (6) dengan kernel konvolusi berukuran 1×1 .
8. Selanjutnya hasil *max pooling* dari langkah (7) diproses ke dalam blok *Inception A* sebanyak tiga kali. Pada blok *Inception A* terdapat empat jalur, dimana pada jalur pertama dilakukan seperti langkah (iv) sebanyak tiga kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 , 3×3 , dan 3×3 . Pada jalur kedua dilakukan seperti langkah (iv) sebanyak dua kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 dan 5×5 . Pada jalur ketiga dilakukan proses pengurangan dimensi *feature maps* menggunakan *average pooling*, kemudian dilanjutkan seperti langkah (4) dengan kernel konvolusi berukuran 1×1 . Pada jalur keempat dilakukan seperti langkah (4) dengan kernel konvolusi berukuran 1×1 . Selanjutnya, hasil pada setiap jalur digabungkan menggunakan *concatenate*.
9. Hasil dari langkah (8) diproses ke dalam blok *Inception B*. Pada blok *Inception B* terdapat tiga jalur, dimana pada jalur pertama dilakukan seperti langkah (4) sebanyak tiga kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 , 3×3 , dan 3×3 . Pada jalur kedua

dilakukan proses pengurangan dimensi *feature maps* menggunakan *max pooling*. Pada jalur ketiga, dilakukan seperti langkah (4) sebanyak satu kali. Selanjutnya, hasil pada setiap jalur digabungkan menggunakan *concatenate*.

10. Hasil dari langkah (9) diproses ke dalam blok *Inception C* sebanyak empat kali. Pada blok *Inception C* terdapat empat jalur, dimana pada jalur pertama dilakukan seperti langkah (4) sebanyak lima kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 , 7×1 , 1×7 , 7×1 , dan 1×7 . Pada jalur kedua dilakukan seperti langkah (4) sebanyak tiga kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 , 1×7 , dan 7×1 . Pada jalur ketiga dilakukan proses pengurangan dimensi *feature maps* menggunakan *average pooling*, kemudian dilanjutkan seperti langkah (4) dengan kernel konvolusi berukuran 1×1 . Pada jalur keempat dilakukan seperti langkah (4) dengan kernel konvolusi berukuran 1×1 . Selanjutnya, hasil pada setiap jalur digabungkan menggunakan *concatenate*.
11. Hasil dari langkah (10) diproses ke dalam blok *Inception D*. Pada blok *Inception D* terdapat tiga jalur, dimana pada jalur pertama dilakukan seperti langkah (4) sebanyak empat kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 , 1×7 , 7×1 , dan 3×3 . Pada jalur kedua dilakukan seperti langkah (4) sebanyak dua kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 , dan 3×3 . Pada jalur ketiga dilakukan proses pengurangan dimensi *feature maps* menggunakan *max pooling*, Selanjutnya, hasil pada setiap jalur digabungkan menggunakan *concatenate*.
12. Hasil dari langkah (11) diproses ke dalam blok *Inception E* sebanyak dua kali. Pada blok *Inception E* terdapat empat jalur, dimana pada jalur pertama dilakukan seperti langkah (4) sebanyak dua kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×1 , dan 3×3 . Selanjutnya, dipecah menjadi dua jalur dimana masing-masing jalur dilakukan seperti langkah (4) sebanyak satu kali dengan ukuran kernel

konvolusi secara berurutan berukuran 1×3 , dan 3×1 dan di gabungkan menggunakan *concatenate*. Pada jalur kedua dilakukan seperti langkah (4) satu kali dengan kernel berukuran 1×1 . Selanjutnya, dipecah menjadi dua jalur dimana masing-masing jalur dilakukan seperti langkah (4) sebanyak satu kali dengan ukuran kernel konvolusi secara berurutan berukuran 1×3 , dan 3×1 dan digabungkan menggunakan *concatenate*. Pada jalur ketiga dilakukan proses pengurangan dimensi *feature maps* menggunakan *average pooling*, kemudian dilanjutkan seperti langkah (4) dengan kernel berukuran 1×1 . Pada jalur keempat dilakukan seperti langkah (4) dengan kernel berukuran 1×1 . Selanjutnya, hasil pada setiap jalur digabungkan menggunakan *concatenate*.

13. Lakukan pengurangan ukuran *feature maps* menggunakan *global average pooling*.
14. Hasil dari langkah (13) diproses ke dalam *dense layer*, kemudian disubstitusikan ke dalam fungsi aktivasi *softmax*.
15. Hitung nilai *loss function categorical cross entropy*. Jika nilai *loss function* untuk data validasi kurang dari nilai validasi *loss* pada *epoch* sebelumnya maka bobot akan disimpan dan digunakan untuk *epoch* selanjutnya, jika tidak maka bobot akan diperbarui untuk *epoch selanjutnya*.
16. Ulangi langkah (2) sampai (15) hingga mencapai *epoch* terakhir.
17. Simpan bobot terakhir ke dalam model *Inception-v3* untuk digunakan pada saat *testing* data.

3.5.4.4. Training Menggunakan Ensemble Learning

Proses *training* data menggunakan *ensemble learning* dilakukan dengan langkah-langkah sebagai berikut:

1. *Input* citra akan diproses kedalam tiga arsitektur, yaitu *ResNet*, *EfficientNet*, dan *Inception-V3*. Pada proses *training* ini, ketiga arsitektur tidak akan melakukan perubahan nilai bobot, tetapi ketiga arsitektur ini akan mengeluarkan nilai probabilitas.

2. Nilai probabilitas yang dihasilkan oleh masing-masing arsitektur selanjutnya dihitung nilai *softmax*, dan selanjutnya digunakan teknik *weighted voting*.
3. Setelah itu, *Fully Connected Layer* digunakan pada tahap berikutnya untuk penentuan bobot akhir dan menangani *overfitting* selama *training*.
4. Model *ensemble* akan menghasilkan probabilitas baru, dimana *output* tersebut akan dihitung nilai *loss function categorical cross entropy* dan nilai akurasi nya untuk setiap *epoch*.
5. Lakukan penyimpanan bobot pada *layer weighted voting*. Bobot akan disimpan jika nilai *loss* pada data validasi lebih kecil dari *epoch* sebelumnya, jika tidak maka bobot akan diperbarui untuk *epoch* selanjutnya.
6. Ulangi langkah (2) sampai (5) hingga mencapai *epoch* terakhir.
7. Simpan bobot terakhir ke dalam model *ensemble learning* untuk digunakan pada saat *testing* data.

3.5.4.5. Testing pada Klasifikasi Citra

Proses *testing* dilakukan untuk mengevaluasi kinerja model dari masing-masing arsitektur yang telah di *training* dan membandingkannya dengan hasil dari metode *ensemble learning*. Proses ini dilakukan dengan menggunakan data uji yang telah dibagi sebelumnya. Adapun langkah-langkah pada tahapan ini sebagai berikut:

1. Implementasikan bobot yang dihasilkan selama proses *training* pada masing-masing arsitektur.
2. Simpan hasil dari proses *testing* pada setiap arsitektur agar dapat dibandingkan dengan metode *ensemble learning*.
3. Kemudian lakukan prediksi menggunakan model ketiga arsitektur dan model *ensemble learning*. Hasil prediksi tersebut berupa nilai bobot.
4. Selanjutnya, nilai-nilai ini akan termuat dalam *confusion matrix* yang dapat digunakan untuk menghitung evaluasi kinerja model.

3.5.4.6. Evaluasi Kinerja Klasifikasi Citra

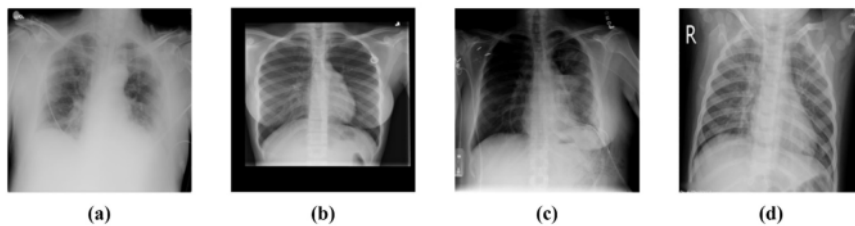
Pada tahapan ini akan membandingkan hasil kinerja klasifikasi tunggal (*ResNet*, *EfficientNet*, dan *Inception-v3*) dengan hasil kinerja metode *ensemble learning* menggunakan ukuran kinerja model berupa nilai akurasi, presisi, *recall*, dan *F1-Score*.

3 BAB IV

HASIL DAN PEMBAHASAN

4.1. Deskripsi Data

Dataset yang digunakan pada penelitian ini yaitu *COVID-19 Radiography Database*. Dataset ini merupakan kumpulan citra CXR dari proyek penelitian *Qatar University* dan *University of Dhaka* yang berkolaborasi bersama Pakistan dan Malaysia. Dataset ini digunakan untuk analisis dan deteksi kasus *COVID-19*, serta kondisi kesehatan paru-paru lainnya. Terdapat 4 kelas yang digunakan pada penelitian ini yaitu 3.616 citra *COVID-19*, 10.192 citra normal, 6.012 citra *lung opacity*, dan 1.345 citra *pneumonia*. Adapun contoh citra CXR masing-masing kelas dapat dilihat pada Gambar 4.1.















Gambar 4.1. Contoh Citra CXR Paru-paru (a) *COVID-19* (b) Normal (c) *Lung Opacity* (d) *Pneumonia*

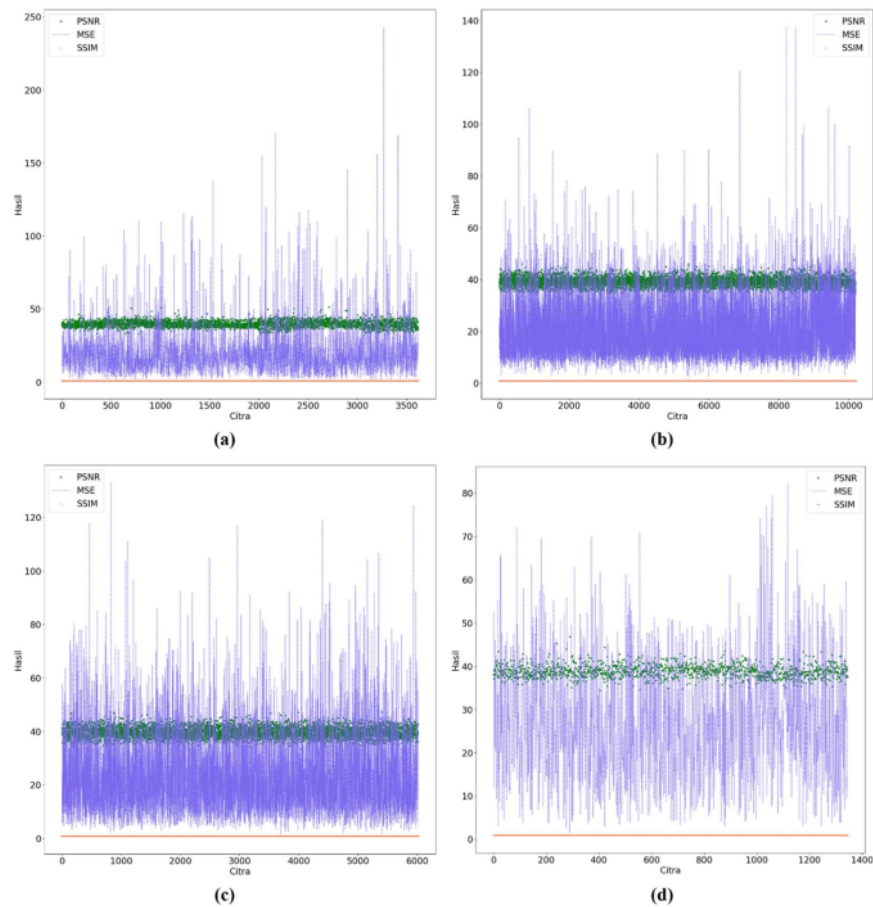
4.2. Peningkatan Kualitas Citra CXR

Citra CXR pada penelitian ini memiliki intensitas citra yang tidak merata pada skala keabuan sehingga sulit untuk mendeteksi *detail* struktur paru-paru. Untuk itu diperlukan teknik yang dapat meningkatkan kualitas citra CXR sehingga dapat memudahkan dokter dan ahli radiologi dalam mendeteksi penyakit paru-paru. Penelitian ini menggunakan Morfologi *Opening* untuk memperbaiki kontur dan menghilangkan objek-objek tipis yang ada pada citra. Selanjutnya dilakukan pengurangan *noise* menggunakan *Median Filter*. Hasil perbandingan peningkatan citra dengan menggunakan metode yang digunakan pada masing-masing kelas citra dapat dilihat pada Tabel 4.1.

Tabel 4.1. Perbandingan hasil peningkatan citra CXR

Nama file	Original	Morfologi Opening	Median Filter
COVID-1.png			
Normal-1.png			
Lung Opacity-1.png			
Pneumonia-1.png			

Terlihat pada Tabel 4.1 menampilkan perbandingan visual dari hasil peningkatan citra untuk setiap kelas citra. Meskipun hasil visual menunjukkan perbaikan, namun sulit untuk secara subjektif menentukan kualitas citra yang terbaik. Untuk mempermudah penentuan kualitas citra yang terbaik, penelitian ini menggunakan pengukuran kuantitatif dengan menghitung nilai PSNR, MSE, dan SSIM pada citra asli dan citra hasil peningkatan kualitas. Nilai PSNR mengukur seberapa baik citra hasil peningkatan dibandingkan citra asli, semakin tinggi nilai PSNR, semakin baik citra hasil peningkatan. MSE mengukur rata-rata kesalahan per piksel antara citra asli dan hasil peningkatan, semakin rendah nilai MSE, semakin baik kualitas citra hasil peningkatan. Sedangkan SSIM mengukur tingkat kesamaan struktural antara citra asli dan hasil peningkatan, nilai SSIM mendekati 1 menunjukkan kualitas citra yang lebih baik. Grafik perbandingan nilai PSNR, MSE dan SSIM dapat dilihat seperti pada Gambar 4.2.



Gambar 4.2. Perbandingan Grafik Nilai Evaluasi Peningkatan Kualitas Citra CXR
(a) *COVID-19* (b) *Normal* (c) *Lung Opacity* (d) *Pneumonia*

Terlihat bahwa pada Gambar 4.2 grafik perbandingan nilai PSNR, MSE, dan SSIM antara citra asli dan hasil peningkatan kualitas citra menunjukkan perbedaan pada keempat kelas citra, yaitu citra *COVID-19*, *Normal*, *Lung Opacity*, dan *Pneumonia*. Kelas citra *Pneumonia* menunjukkan nilai evaluasi yang lebih bervariasi dibandingkan dengan kelas lainnya. Hasil evaluasi kinerja pada metode Morfologi *Opening* dan *Median Filter* menghasilkan parameter kualitas citra pada setiap kelas dengan rata-rata nilai PSNR, MSE, dan SSIM ditunjukkan seperti pada Tabel 4.2.

Tabel 4.2. Perbandingan Hasil Evaluasi Kinerja pada Metode *Median Filter*

	PSNR	MSE	SSIM
Citra <i>COVID-19</i>	39,714	19,891	0,952
Citra Normal	39,022	19,965	0,954
Citra <i>Lung Opacity</i>	39,669	23,121	0,951
Citra <i>Pneumonia</i>	38,825	26,901	0,951

Terlihat bahwa pada Tabel 4.2 hasil evaluasi kinerja menggunakan metode Morfologi *Opening* dan *Median Filter* menunjukkan nilai rata-rata PSNR, MSE, dan SSIM yang telah mendekati atau mencapai angka yang dianggap baik. Nilai rata-rata PSNR berada pada rentang 38-39, nilai MSE berada pada rentang 19-27, dan SSIM berada pada rentang 0,95-0,96. Hal ini menunjukkan bahwa kualitas citra setelah dilakukan peningkatan sudah baik. Untuk lebih memvalidasi nilai evaluasi kinerja yang diusulkan, hasil peningkatan kualitas citra akan dibandingkan dengan hasil dari metode penelitian lainnya yang dapat dilihat seperti pada Tabel 4.3.

Tabel 4.3. Perbandingan Hasil Evaluasi Penelitian ini dengan Penelitian Lain

Metode	PSNR	MSE	SSIM
<i>Region Growing</i> dan <i>Wavelet Transform</i> [124]	37.122	25.5911	-
<i>Gamma Correction, Gaussian Filter</i> [103]	24.61	229.92	-
<i>Wiener Filter</i> dan <i>Poisson Noise</i> [104]	33	29	0.7876
<i>Type-2 Neutrosophic Set</i> [105]	28.58	23.6	0.9
Morfologi <i>Opening</i> dan <i>Median Filter</i> [106]³	39.307	22.469	0.952

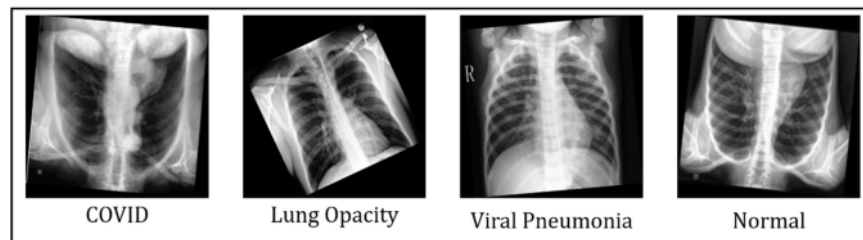
Pada Tabel 4.3 penelitian yang dilakukan oleh Allam Zanaty & Mostafa Ibrahim [124] dan Umamaheswari & Geetha [103] hanya menghitung nilai PSNR dan MSE, sedangkan SSIM tidak dihitung. Penelitian oleh Umamaheswari & Geetha [103] juga didapatkan nilai PSNR terendah dan MSE tertinggi dibandingkan

³ Kesuma, L. I., Ermatita, Erwin, Sari, P., & Purabaya, R. H. (2022). Improved Chest X - Ray Image Quality Using Median and Gaussian Filter Methods. *2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. (Lampiran 1)

penelitian lainnya. Sementara itu, penelitian lain oleh Roy & Maity [104] dan Abdel-Basset et al [105] menghitung semua nilai yang digunakan. Penelitian oleh Roy & Maity [104] memperoleh skor SSIM terendah dibandingkan penelitian lainnya. Pada metode yang diusulkan menggunakan Morfologi *Opening* dan *Median Filter* memperoleh hasil PSNR dan SSIM tertinggi serta MSE terendah dibandingkan penelitian lainnya. Secara keseluruhan, penelitian ini berhasil mengatasi masalah intensitas citra yang tidak merata pada skala keabuan pada citra CXR dan meningkatkan kualitas citra secara signifikan menggunakan teknik Morfologi *Opening* dan *Median Filter*.

4.3. Augmentasi Data pada Citra CXR

Pada dataset Citra CXR yang digunakan dalam penelitian ini, terdapat ketidakseimbangan jumlah data pada setiap kelas. Ketidakseimbangan ini dapat menyebabkan model cenderung cenderung memprediksi kelas mayoritas dan mengabaikan kelas minoritas, sehingga performa model menjadi tidak optimal. Untuk mengatasi masalah ini, digunakan teknik augmentasi data pada citra CXR. Augmentasi data digunakan untuk memperbanyak data dengan cara mengubah citra asli menjadi variasi baru, namun tetap mempertahankan informasi yang sama. Pada penelitian ini, augmentasi data yang digunakan yaitu rotasi dan *flipping*. Teknik rotasi dilakukan dengan memutar citra CXR sebesar sudut tertentu. Sementara teknik *flipping* dilakukan dengan membalik citra secara horizontal dan vertikal. Ilustrasi beberapa contoh gambar CXR yang diperoleh dari augmentasi data dapat dilihat seperti pada Gambar 4.3 berikut.

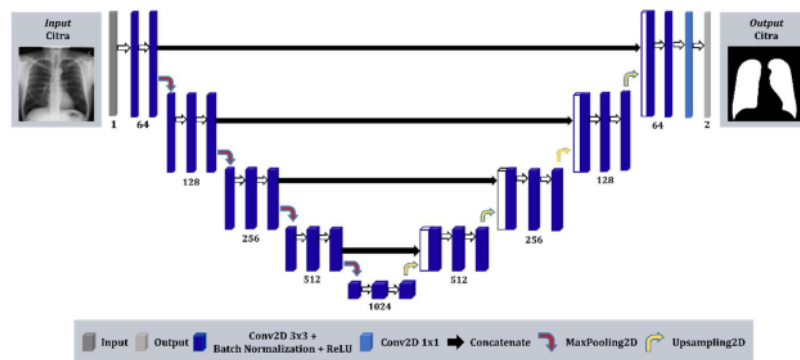


Gambar 4.3. Hasil Augmentasi Data Menggunakan Transformasi Rotasi

Terlihat bahwa pada Gambar 4.3 beberapa contoh citra CXR yang telah mengalami proses augmentasi data ditampilkan. Terlihat bahwa transformasi rotasi dengan sudut 1° hingga 359° diterapkan pada citra-citra yang termasuk dalam kelas minoritas. Sebagai contoh, pada kelas pneumonia yang semula hanya berisi 1.345 citra, melalui metode augmentasi data, jumlahnya berhasil diperbanyak menjadi 10192 citra. Total keseluruhan citra yang dihasilkan dari proses augmentasi data adalah 40.768 citra dengan setiap kelas pada dataset memiliki jumlah citra yang seimbang yaitu 10.192 citra. Dengan meningkatnya jumlah data pada setiap kelas, diharapkan bahwa model yang dihasilkan dapat memiliki performa yang lebih baik dalam mengenali citra CXR dari berbagai kelas, termasuk kelas minoritas yang sebelumnya mungkin kurang terwakili dalam data latih.

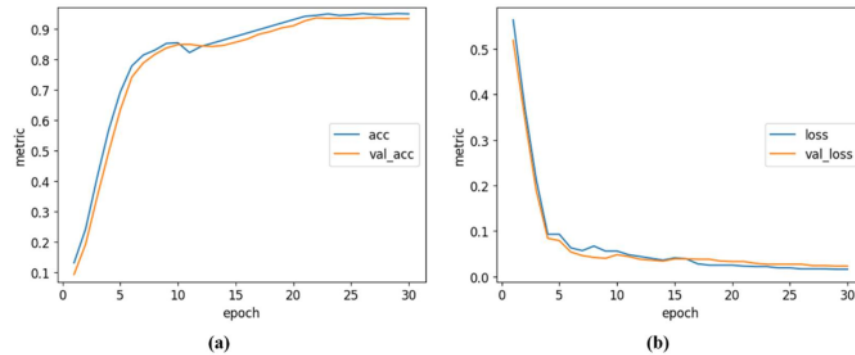
4.4. Segmentasi Citra CXR

Segmentasi citra paru-paru dapat membantu dalam mendeteksi kelainan pada paru-paru. Citra CXR akan dipisahkan menjadi dua bagian selama segmentasi citra yaitu paru-paru sebagai *foreground* dan daerah selain paru-paru sebagai *background*. Data yang dihasilkan dari proses augmentasi dipisahkan menjadi dua jenis: data latih dan data uji. Data latih dipisahkan lagi selama proses *training* menjadi data latih dan data validasi. Proses *training* segmentasi adalah untuk membangun model dari arsitektur yang diusulkan. Hasil dari proses *training* berupa bobot. Metode yang digunakan dalam penelitian ini yaitu modifikasi arsitektur *U-Net Batch Normalization* yang dapat dilihat seperti pada Gambar 4.4.



Gambar 4.4. Arsitektur *U-Net* pada Citra CXR

Pada Gambar 4.4 terlihat bahwa Arsitektur U-Net terdiri dari jalur encoder (sisi kiri) dan jalur decoder (sisi kanan). Jalur encoder memiliki empat blok konvolusi dengan fungsi aktivasi *ReLU*, dilanjutkan dengan *batch normalization* dan *max pooling* untuk mengurangi dimensi. Jumlah *feature maps* di jalur *encoder* akan berlipat ganda. Jalur *decoder* menggunakan *transposed convolution* untuk mengembalikan dimensi *feature maps* dan dilakukan *concatenate* dengan *feature maps* dari jalur *encoder*. Kemudian, dilakukan empat blok konvolusi dengan jumlah *feature maps* berkurang dua kali lipat hingga mencapai jumlah awal. Tahap terakhir adalah konvolusi 1×1 dengan fungsi aktivasi *softmax* untuk menghasilkan gambar hasil segmentasi dengan probabilitas untuk setiap kelas objek. Grafik akurasi dan *loss* tahap segmentasi selama proses *training* dapat dilihat seperti pada Gambar 4.5.

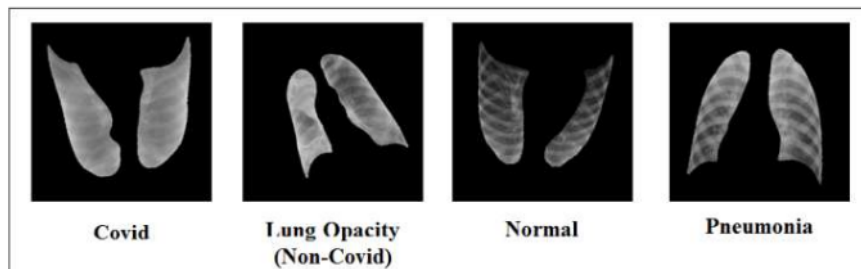


Gambar 4.5. Grafik *Training* Segmentasi Citra CXR (a) Akurasi (b) *Loss*

Pada Gambar 4.5 terlihat bahwa grafik akurasi data latih pada segmentasi citra mengalami peningkatan pada setiap *epoch*. Untuk *epoch* pertama nilai akurasinya 13%, kemudian meningkat terus menerus hingga 95%. Grafik akurasi untuk data validasi pada segmentasi citra juga meningkat pada setiap *epoch*. Untuk *epoch* pertama nilai akurasinya adalah 9%, kemudian meningkat terus menerus menjadi 93%. Grafik *loss* untuk data latih mengalami penurunan pada setiap *epoch*. Untuk *epoch* pertama, nilai *loss* 57%, kemudian terus menurun hingga mencapai 1%. Grafik *loss* untuk data validasi pada segmentasi citra juga mengalami penurunan pada setiap *epoch*. Untuk *epoch* pertama nilai *loss* adalah 52%, kemudian terus menurun hingga 2%. Berdasarkan Gambar 4.5, kinerja model pada

metode *U-Net Batch Normalization* sangat baik dalam melakukan segmentasi paru-paru pada citra CXR karena nilai akurasi di atas 93% dan nilai *loss* mendekati 1%.

Tahap *testing* dilakukan untuk menguji bobot yang dihasilkan selama proses *training* yang digunakan pada data uji. Bobot terbaik hasil proses *training* menggunakan *U-Net* digunakan untuk menguji segmentasi citra CXR paru-paru dengan data yang baru. Hasil segmentasi CXR paru-paru dipisahkan menjadi dua bagian yaitu paru-paru sebagai *foreground* dan daerah selain paru-paru sebagai *background*. Ilustrasi beberapa contoh citra CXR yang diperoleh dari proses segmentasi dapat dilihat pada Gambar 4.6.



Gambar 4.6. Hasil Segmentasi Citra CXR Paru-paru Menggunakan *U-Net Batch Normalization*

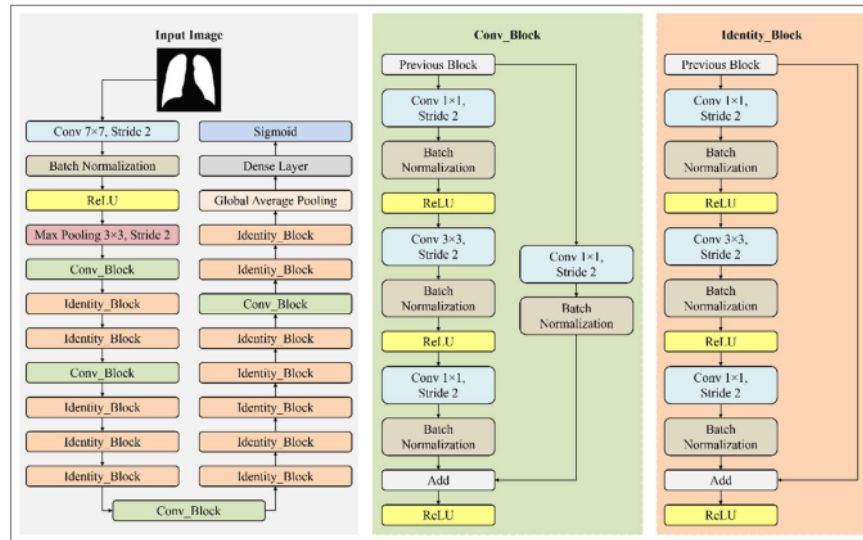
4.5. Klasifikasi Citra CXR

Pada tahap klasifikasi citra, data yang digunakan adalah data hasil segmentasi citra CXR. Data ini dibagi menjadi data latih dan data uji. Proses *training* untuk data latih selanjutnya dibagi lagi menjadi data latih dan data validasi. Pada tahap ini digunakan 40.768 data latih, dengan 90% (36.691 data latih) dan 10% (4.077 data validasi).

4.5.1. *Training* Klasifikasi Citra CXR Menggunakan *ResNet*

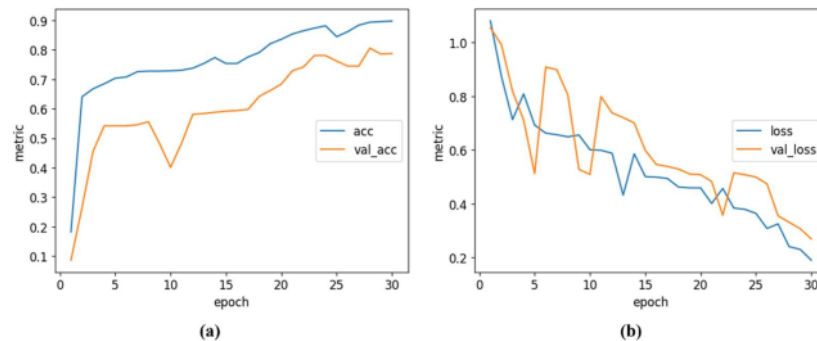
ResNet merupakan arsitektur CNN yang dikenal dalam mengatasi masalah *vanishing gradient*. Dengan menggunakan *residual block*, *ResNet* dapat menyederhanakan proses pelatihan jaringan yang dalam dan memungkinkan untuk melatih model yang lebih dalam. Pada klasifikasi citra CXR, *ResNet* dapat

mempelajari fitur-fitur kompleks yang ada dalam citra dan menghasilkan representasi fitur yang akurat. Arsitektur *ResNet* pada klasifikasi citra CXR dapat dilihat seperti pada Gambar 4.7.



Gambar 4.7. Arsitektur *ResNet* pada Klasifikasi Citra CXR

Pada Gambar 4.7 terlihat bahwa pada arsitektur *ResNet*, citra CXR diproses melalui *convolutional layer* dengan fungsi aktivasi *ReLU* dan *batch normalization*. Selanjutnya, dilakukan operasi *max pooling* untuk mengurangi dimensi *feature maps*. *Feature maps* tersebut kemudian diproses melalui *conv_block* dan jalur *residual connection*. Hasil yang diperoleh selanjutnya dijumlahkan dan dilakukan fungsi aktivasi menggunakan *ReLU*. Langkah ini diulangi beberapa kali, termasuk pengulangan pada *identity_block*. Setelah beberapa kali pengulangan, ukuran *feature maps* dikurangi menggunakan *global average pooling*. Selanjutnya, hasilnya diproses melalui *dense layer* dan fungsi aktivasi *sigmoid* untuk menghasilkan probabilitas kelas. Grafik akurasi dan *loss* tahap klasifikasi citra menggunakan arsitektur *ResNet* ³ selama proses *training* dapat dilihat seperti pada Gambar 4.8.



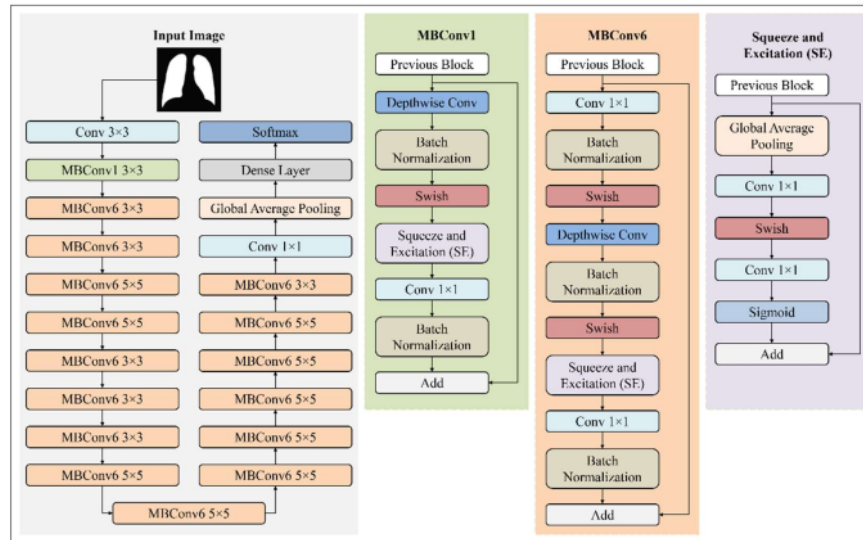
Gambar 4.8. Grafik Training Klasifikasi Citra CXR Menggunakan Arsitektur ResNet (a) Akurasi (b) Loss

Pada Gambar 4.8 terlihat bahwa grafik akurasi untuk data latih dalam arsitektur ResNet meningkat di setiap epoch. Untuk epoch pertama, nilai akurasi pada data latih adalah 18%, kemudian meningkat menjadi 89%. Selain itu, grafik akurasi untuk data validasi pada arsitektur ResNet mengalami peningkatan dan penurunan. Untuk epoch pertama sebesar 8% dan meningkat menjadi 55% di epoch kedelapan. Pada epoch kesepuluh, nilainya menurun menjadi 40% dan kemudian meningkat menjadi 78% pada epoch berikutnya. Sementara untuk grafik loss pada data latih mengalami penurunan pada setiap epoch. Untuk epoch pertama, nilai loss lebih dari 100%, kemudian terus menurun hingga mencapai 20%. Sedangkan untuk grafik loss pada data validasi mengalami penurunan dan peningkatan pada setiap epoch. Untuk epoch pertama nilai loss lebih dari 100%, kemudian pada epoch kelima menurun menuju 55%. Pada epoch kesebelas, nilainya kembali meningkat ke 79% dan kemudian menurun menuju 23% pada epoch berikutnya. Terlihat bahwa grafik yang diperoleh arsitektur ResNet masih mengalami *overfitting* karena model hanya mampu mengenali pola pada data latih.

4.5.2. Training Klasifikasi Citra CXR Menggunakan EfficientNet

EfficientNet adalah arsitektur CNN yang dikenal karena berhasil mengatasi masalah penskalaan dimensi dengan seragam dan seimbang pada kedalaman (*depth*), lebar (*width*), serta resolusi di setiap lapisannya. Dengan melakukan penyesuaian dimensi secara seragam, arsitektur ini mampu meningkatkan

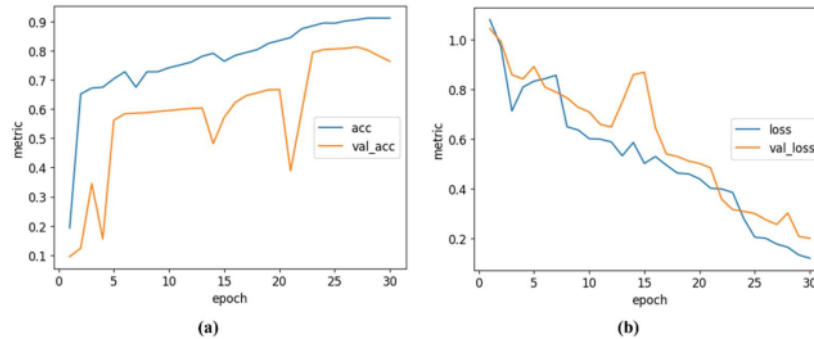
kemampuannya dalam menangkap fitur-fitur yang kompleks dan abstrak. Arsitektur *EfficientNet* pada klasifikasi citra CXR dapat dilihat seperti pada Gambar 4.9.



Gambar 4.9. Arsitektur *EfficientNet* pada Klasifikasi Citra CXR

Pada Gambar 4.9 terlihat bahwa pada arsitektur *EfficientNet*, citra CXR diproses melalui *convolutional layer* yang diikuti dengan operasi *batch normalization* dan fungsi aktivasi *ReLU*. Selanjutnya hasil tersebut diproses ke dalam blok *MBConv*. Terdapat dua jenis blok *MBConv* yang digunakan pada penelitian ini yaitu *MBConv1* dan *MBConv6*. *MBConv1* digunakan satu kali setelah proses konvolusi dengan kernel berukuran 3×3 . Sedangkan *MBConv6* digunakan setelahnya dengan ukuran kernel yang berbeda yaitu (3×3 dan 5×5). *MBConv* terdiri dari operasi *depthwise convolution*, *batch normalization*, fungsi aktivasi *swish*, *squeeze and excitation*, *convolution layer*, dan *add*. Selanjutnya hasil yang diperoleh dari proses *MBConv6* dilakukan operasi *convolutional layer* kembali dengan kernel 1×1 yang diikuti operasi *batch normalization* dan fungsi aktivasi *ReLU*. Setelah itu dilakukan pengurangan ukuran *feature maps* menggunakan operasi *global average pooling*. *Feature maps* tersebut kemudian diproses melalui *dense layer* dengan aktivasi *softmax* untuk menghasilkan probabilitas kelas. Grafik

akurasi dan *loss* tahap klasifikasi citra menggunakan arsitektur *EfficientNet* selama proses *training* dapat dilihat seperti pada Gambar 4.10.

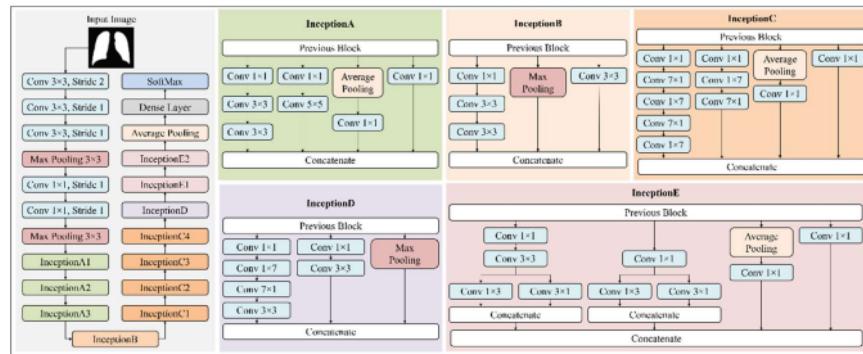


Gambar 4.10. Grafik *Training* Klasifikasi Citra CXR Menggunakan Arsitektur *EfficientNet* (a) Akurasi (b) *Loss*

Pada Gambar 4.10 terlihat bahwa grafik akurasi untuk data latih pada arsitektur *EfficientNet* mengalami peningkatan pada setiap *epoch* namun mengalami penurunan pada beberapa *epoch*. Pada *epoch* pertama nilai akurasi pada data *training* adalah 19% yang kemudian meningkat menjadi 91%. Sementara itu, grafik akurasi untuk validasi data pada arsitektur *EfficientNet* mengalami peningkatan dan penurunan di setiap *epoch*. Pada *epoch* pertama sebesar 9%, kemudian meningkat menjadi 34%, dan menurun lagi pada *epoch* ke-4, 14, dan 20. Selanjutnya, akurasi meningkat dan menurun lagi, mencapai 76%. Sementara untuk grafik *loss* pada data latih mengalami penurunan pada setiap *epoch*. Untuk *epoch* pertama, nilai *loss* lebih dari 100%, kemudian terus menurun hingga mencapai 12%. Sedangkan untuk grafik *loss* pada data validasi mengalami penurunan dan peningkatan pada setiap *epoch*. Untuk *epoch* pertama nilai *loss* lebih dari 100%, kemudian pada *epoch* ke-12 menurun menuju 64%. Pada *epoch* ke-15, nilainya kembali meningkat ke 87% dan kemudian menurun menuju 20% pada *epoch* berikutnya. Terlihat bahwa grafik yang diperoleh arsitektur *EfficientNet* juga masih mengalami *overfitting* karena model hanya mampu mengenali pola pada data latih dan tidak dapat mengenali data yang baru.

4.5.3. Training Klasifikasi Citra CXR Menggunakan *Inception-v3*

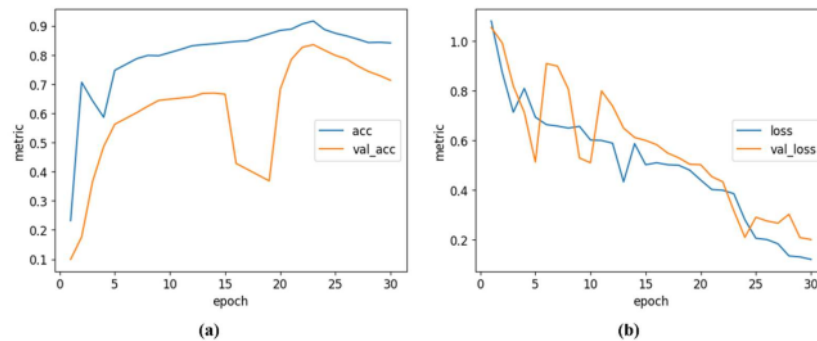
Inception-v3 merupakan arsitektur CNN yang dapat mengatasi masalah kompleksitas jaringan. Dengan menggunakan *inception block*, arsitektur *Inception-v3* mempelajari representasi jaringan yang lebih baik dari data *input*. Arsitektur *Inception-v3* pada klasifikasi citra CXR dapat dilihat seperti pada Gambar 4.11.



Gambar 4.11. Arsitektur *Inception-v3* pada Klasifikasi Citra CXR

Pada Gambar 4.11 terlihat bahwa arsitektur *Inception-v3* terdiri dari beberapa blok *Inception*, yakni A, B, C, D, dan E, setiap blok *Inception* memiliki jalur-jalur yang berbeda untuk melakukan operasi konvolusi dengan ukuran kernel yang berbeda. Proses dimulai dengan operasi konvolusi 3×3 , diikuti oleh *batch normalization* dan fungsi aktivasi *ReLU*. Kemudian, dilakukan operasi *max pooling* untuk mengurangi dimensi *feature maps*. Setelah itu, dilakukan kembali operasi konvolusi dengan kernel 1×1 dan kemudian *max pooling* lagi. Hasil dari *max pooling* diproses ke dalam blok *Inception A* sebanyak tiga kali. Pada setiap blok tersebut, terdapat empat jalur yang masing-masing melakukan operasi konvolusi dengan ukuran kernel yang berbeda dan pengurangan dimensi menggunakan *average pooling*. Hasil dari setiap jalur digabungkan menggunakan *concatenate*. Hasil dari blok *Inception A* kemudian diproses ke dalam blok *Inception B*. Pada blok ini, terdapat tiga jalur yang melibatkan operasi konvolusi dan penggunaan *max pooling* untuk pengurangan dimensi *feature maps*. Setelah itu, hasil dari setiap jalur pada *Inception B* digabungkan menggunakan *concatenate* dan diproses ke dalam blok *Inception C* sebanyak empat kali. Blok ini memiliki empat jalur dengan

kombinasi operasi konvolusi dan pengurangan dimensi menggunakan *average pooling*. Hasil dari setiap jalur pada *inception C* juga digabungkan menggunakan *concatenate* dan diproses ke dalam blok *Inception D* serta *E* secara berulang sesuai jumlah blok masing-masingnya. Hasil pada blok *Inception E* dilakukan pengurangan ukuran *feature maps* menggunakan *global average pooling*. Kemudian, hasil tersebut diproses ke dalam *dense layer* dan diaktivasi dengan *softmax* untuk mendapatkan output klasifikasi. Grafik akurasi dan *loss* tahap klasifikasi citra menggunakan arsitektur *Inception-v3* selama proses *training* dapat dilihat seperti pada Gambar 4.12.



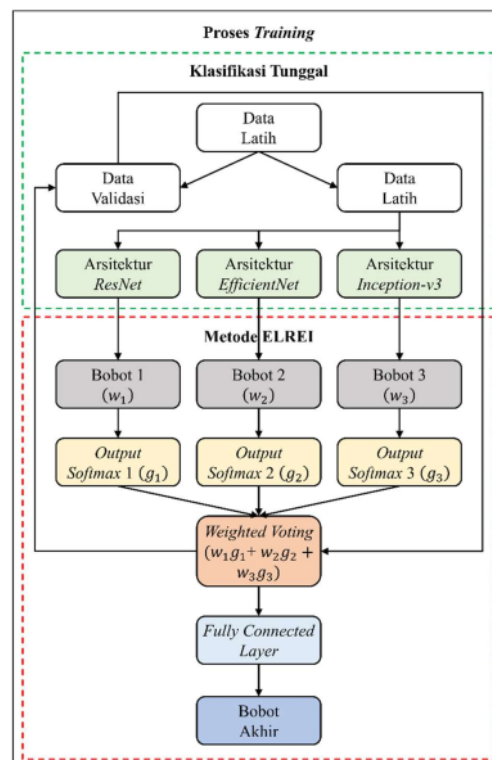
Gambar 4.12. Grafik *Training* Klasifikasi Citra CXR Menggunakan Arsitektur *Inception-v3* (a) Akurasi (b) *Loss*

Pada Gambar 4.12 terlihat bahwa grafik akurasi untuk data pelatihan dan validasi menggunakan arsitektur *Inception-v3*. Sama halnya dengan arsitektur *EfficientNet*, nilai akurasi mengalami kenaikan dan penurunan pada setiap *epoch*. Pada *epoch* pertama nilai akurasi pada data latih sebesar 23% dan terus meningkat menjadi 91% pada *epoch* ke-23. Selain itu, nilai akurasi data validasi meningkat dari 10% menjadi 83%. Namun, untuk *epoch* selanjutnya menurun menjadi 71%. Sementara untuk grafik *loss* pada data latih mengalami penurunan pada setiap *epoch*. Untuk *epoch* pertama, nilai *loss* lebih dari 100%, kemudian terus menurun hingga mencapai 11%. Sedangkan untuk grafik *loss* pada data validasi mengalami penurunan dan peningkatan pada setiap *epoch*. Untuk *epoch* pertama nilai *loss* lebih dari 100%, kemudian pada *epoch* kelima menurun menuju 51%. Pada *epoch* keenam, nilainya kembali meningkat ke 91% dan kemudian menurun menuju 50%

pada *epoch* ke-10. Selanjutnya pada *epoch* ke-11 kembali meingkat ke 80% dan menurun menuju 19% pada *epoch* berikutnya. Terlihat bahwa grafik yang diperoleh arsitektur *Inception-v3* sama seperti arsitektur *ResNet* dan *EfficientNet* yaitu masih mengalami *overfitting* karena model hanya mampu mengenali pola pada data latih.

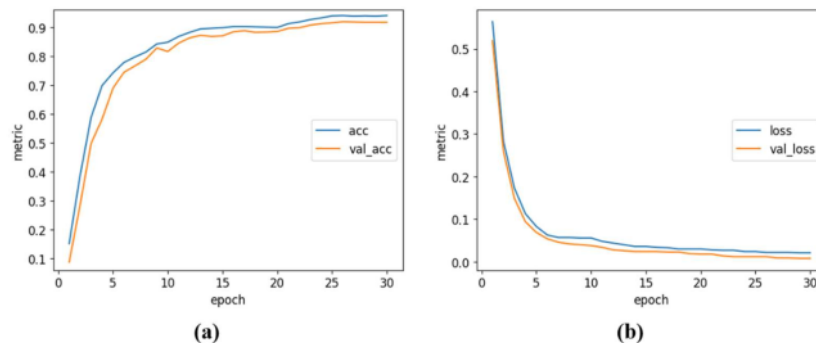
4.5.4. Training Klasifikasi Citra CXR Menggunakan ELREI

ELREI (*Ensemble Learning* dari *ResNet*, *EfficientNet*, dan *Inception-v3*) mengatasi keterbatasan dan memanfaatkan keunggulan arsitektur masing-masing arsitektur CNN (*ResNet*, *EfficientNet*, dan *Inception-v3*) dengan menggabungkan hasil klasifikasinya. ELREI bekerja selama tahap *training* di setiap *epoch*. Tahapan klasifikasi citra pada proses *training* data menggunakan ELREI dapat dilihat pada Gambar 4.13.



Gambar 4.13. Proses ELREI pada Klasifikasi Citra CXR

Pada Gambar 4.13 terlihat bahwa proses awal *training* data dilakukan klasifikasi tunggal dengan menggunakan arsitektur *ResNet*, *EfficientNet*, dan *Inception-v3*. Hasil dari proses *training* ini adalah bobot terbaik setiap *epoch*-nya dari masing-masing arsitektur. Selanjutnya, dilakukan klasifikasi dengan menggunakan metode *Ensemble Learning* dari *ResNet*, *EfficientNet*, dan *Inception-v* (ELREI). Metode ELREI bekerja pada setiap *epoch* dalam proses *training*. Bobot yang dihasilkan dari setiap arsitektur kemudian dihitung nilai *softmax* nya, dan kemudian dilakukan teknik *weighted voting* untuk menggabungkan hasil dari ketiga arsitektur tersebut. Setelah itu, *Fully Connected Layer* (FCN) digunakan pada tahap berikutnya untuk penentuan bobot akhir dan menangani masalah *overfitting* selama proses *training*. FCN mempelajari pola-pola dari data secara menyeluruh tanpa kehilangan informasi spasialnya, sehingga dapat digunakan untuk melakukan klasifikasi penyakit paru-paru berdasarkan citra CXR. Grafik akurasi dan *loss* tahap klasifikasi citra menggunakan metode ELREI selama proses *training* dapat dilihat seperti pada Gambar 4.14.



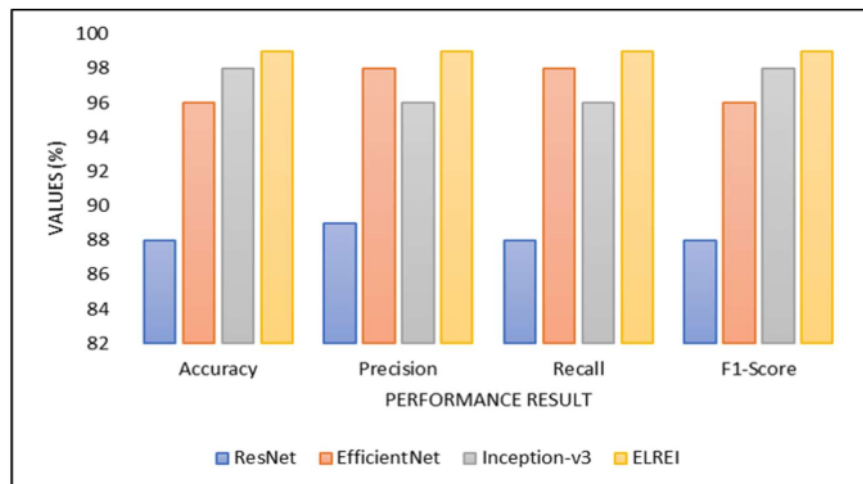
Gambar 4.14. Grafik *Training* Klasifikasi Citra CXR Menggunakan Metode ELREI (a) Akurasi (b) *Loss*

Pada Gambar 4.14 terlihat bahwa grafik grafik akurasi data latih pada metode ELREI mengalami peningkatan pada setiap *epoch*. Untuk epoch pertama nilai akurasinya 15%, kemudian meningkat terus menerus hingga 94%. Grafik akurasi untuk data validasi pada metode ELREI juga meningkat pada setiap *epoch*. Untuk *epoch* pertama nilai akurasinya adalah 9%, kemudian meningkat terus

menerus menjadi 92%. Grafik *loss* untuk data latih mengalami penurunan pada setiap *epoch*. Untuk *epoch* pertama, nilai *loss* 56%, kemudian terus menurun hingga mencapai 1%. Grafik *loss* untuk data validasi pada metode ELREI juga mengalami penurunan pada setiap *epoch*. Untuk *epoch* pertama nilai *loss* adalah 51%, kemudian terus menurun hingga 0%. Grafik yang diperoleh menunjukkan bahwa metode ELREI tidak mengalami *overfitting*. ELREI dapat mengenali dan mempelajari pola dalam data yang telah terlatih maupun data yang tidak terlatih. Berdasarkan Gambar 4.14, kinerja model pada metode ELREI sangat baik dalam mengkategorikan citra CXR paru-paru karena nilai akurasinya diatas 92% dan nilai *loss* mendekati 0%.

4.5.5. Testing pada Klasifikasi Citra CXR

Pada tahap *testing*, data uji digunakan untuk melihat sejauh mana keberhasilan metode klasifikasi dalam memprediksi citra CXR paru-paru. Ukuran kinerja evaluasi yang digunakan pada klasifikasi citra CXR adalah akurasi, presisi, *recall*, dan *F1-Score*. Perbandingan nilai evaluasi kinerja pada metode klasifikasi tunggal dan metode ELREI dapat dilihat seperti pada Gambar 4.15.



Gambar 4.15. Perbandingan Evaluasi Kinerja pada Proses Klasifikasi Citra CXR Menggunakan *ResNet*, *EfficientNet*, *Inception-v3*, dan ELREI

Pada Gambar 4.15 terlihat bahwa hasil kinerja dengan menggunakan metode ELREI pada klasifikasi citra CXR mengalami peningkatan kinerja dibandingkan dengan ResNet, EfficientNet, dan Inception-v3. Peningkatan nilai akurasi sebesar 11% (*ResNet*), 3% (*EfficientNet*), dan 1% (*Inception-v3*). Peningkatan nilai presisi adalah 10,5% (*ResNet*), 1% (*EfficientNet*), dan 3% (*Inception-v3*). Sedangkan peningkatan nilai *recall* adalah 10,75% (*ResNet*), 1% (*EfficientNet*), dan 3,25% (*Inception-v3*). Sementara peningkatan nilai *F1-Score* adalah 10,25% (*ResNet*), 3,25% (*EfficientNet*), dan 1% (*Inception-v3*).

4.5.6. Evaluasi Kinerja pada Klasifikasi Citra CXR

Hasil evaluasi kinerja klasifikasi citra menggunakan metode ELREI pada citra CXR telah dilakukan pada penelitian ini. Ukuran kinerja evaluasi yang digunakan yaitu nilai akurasi, presisi, *recall* dan *F1-Score* pada setiap kelas yang ditunjukkan seperti pada Tabel 4.4.

Tabel 4.4. Perbandingan Hasil Evaluasi Kinerja Menggunakan Metode ELREI

Kelas	Akurasi (%)	Presisi (%)	<i>Recall</i> (%)	<i>F1-Score</i> (%)
<i>COVID-19</i>	99	99	99	99
Normal		96	99	98
<i>Lung Opacity</i>		100	97	99
<i>Pneumonia</i>		100	100	100

Terlihat bahwa pada Tabel 4.4, metode ELREI memperoleh nilai presisi, *recall*, dan *F1-Score* pada setiap kelas dengan nilai yang berbeda-beda. Sementara nilai akurasi dihitung keseluruhan bukan pada setiap kelasnya. Untuk nilai presisi terendah diperoleh kelas normal, sedangkan untuk nilai presisi tertinggi diperoleh kelas *lung opacity* dan *pneumonia*. Untuk nilai *recall* terendah diperoleh kelas *lung opacity*, sedangkan untuk nilai *recall* tertinggi diperoleh kelas *pneumonia*. Untuk nilai *F1-Score* terendah diperoleh kelas normal, sedangkan untuk nilai *F1-Score* tertinggi diperoleh kelas *pneumonia*. Nilai rata-rata akurasi yang dihasilkan sebesar 99%, rata-rata presisi 98,75%, rata-rata *recall* sebesar 98,75%, dan rata-rata *F1-Score* 99%.

4.6. ³ Analisis dan Interpretasi Hasil

Pada penelitian ini telah dilakukan klasifikasi citra CXR paru-paru menggunakan metode ELREI. Penelitian mengevaluasi kinerja metode ini dan membandingkannya dengan penelitian sebelumnya. Perbandingan hasil klasifikasi citra CXR paru-paru pada penelitian ini dengan penelitian lain dapat dilihat pada Tabel 4.5.

Tabel 4.5. Perbandingan Hasil Evaluasi Penelitian ini dengan Penelitian Lain

Metode	Tahun	Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)
EDL-COVID[119]	2021	97.8	97.83	97.77	97.77
Ensemble DCNN[120]	2021	97.47	-	98.18	-
Ensemble ResNet & DenseNet [121]	2022	94.1	94.5	94.1	94
CNN-Ensemble[122]	2022	98	98	98.25	97,75
Metode yang Diusulkan (ELREI) [123]⁴	2023	99	98,75	98,75	99

Pada Tabel 4.5 merupakan perbandingan hasil penelitian dengan menggunakan dataset yang sama pada citra CXR untuk klasifikasi penyakit paru-paru. Penelitian oleh [119] dan [120] memiliki nilai akurasi dan recall sama dengan atau lebih dari 95% tetapi tidak mengukur nilai presisi dan *F1-score*. Penelitian oleh [121] mencapai akurasi, presisi, *recall*, dan *F1-score* terendah dengan nilai di bawah 95%. Metode yang diusulkan (ELREI) mencapai akurasi, presisi, *recall*, dan *F1-score* tertinggi dengan nilai di atas 98%. Nilai akurasi dari metode yang diusulkan menunjukkan bahwa secara akurat mengklasifikasikan penyakit paru-paru pada gambar CXR. Nilai presisi dan *recall* menunjukkan bahwa model bekerja dengan baik dalam mengidentifikasi kasus positif dengan meminimalkan *false positive* dan *false negatives*. Nilai *F1-Score* menunjukkan performa yang seimbang

⁴ Kesuma, L. I., Ermatita, & Erwin. (2023). ELREI : Ensemble Learning of ResNet , EfficientNet , and Inception-v3 for Lung Disease Classification based on Chest X-Ray Image. *International Journal of Intelligent Engineering and Systems*, 16(5), 149–161. (Lampiran 2)

antara presisi dan *recall*. Berdasarkan hasil perbandingan pada Tabel 4.3, metode ELREI untuk klasifikasi citra CXR paru-paru memiliki kinerja yang sangat baik. Dilihat dari nilai performansi metode ELREI, metode ini dapat mengklasifikasikan citra CXR paru-paru dengan sangat baik.

BAB V PENUTUP

5.1. Kesimpulan

1. Hasil peningkatan kualitas citra menggunakan Morfologi *Opening* dan *Median Filter* pada citra CXR memberikan hasil yang sangat baik dalam mengatasi kontras citra yang gelap dan *noise* yang berlebihan. Hal ini terlihat dari rata-rata nilai PSNR, MSE, dan SSIM pada metode Morfologi *Opening* dan *Median Filter* menghasilkan rata-rata nilai PSNR telah mendekati nilai 40, SSIM mendekati 1, dan MSE sebesar 20. Hasil ini menunjukkan bahwa penggunaan metode Morfologi *Opening* dan *Median Filter* efektif meningkatkan kualitas citra CXR secara signifikan.
2. Kinerja model arsitektur *U-Net Batch Normalization* dalam segmentasi paru-paru pada citra CXR yang dihasilkan sudah sangat baik dengan nilai akurasi diatas 93% dan nilai *loss* mendekati 1%. Hasil ini menunjukkan bahwa model berhasil dalam mendeteksi objek paru-paru dan background dengan sangat baik pada citra CXR.
3. Penerapan metode klasifikasi penyakit menggunakan klasifikasi tunggal pada arsitektur *ResNet*, *EfficientNet*, dan *Inception-v3* menghasilkan kinerja model yang belum akurat. Hasil kinerja dari masing-masing arsitektur rata-rata masih dibawah 90% dan mengalami *overfitting*. Selanjutnya diterapkan metode *Ensemble Learning* dari *ResNet*, *EfficientNet*, dan *Inception-v3* (ELREI) dengan *weighted voting*. Proses *weighted voting* dilakukan pada setiap *epoch* di tahap *training* menghasilkan peningkatan kinerja yang signifikan dibandingkan dengan metode klasifikasi tunggal. Metode ELREI mencapai hasil yang sangat baik dalam hal akurasi, presisi, *recall*, dan *F1-score*, semuanya di atas 98%. Grafik *training* ELREI juga menunjukkan kemampuannya untuk mengatasi *overfitting*. Secara keseluruhan, metode ELREI terbukti kuat dan sangat baik untuk mengklasifikasikan penyakit paru-paru berdasarkan

citra CXR, dengan mengelompokkan menjadi empat kelas: normal, *COVID-19*, *lung opacity*, dan *pneumonia*.

5.2. Saran

1. Mengkombinasikan lebih banyak metode lainnya dalam meningkatkan kualitas citra secara lebih lanjut, sehingga dapat mengatasi permasalahan *noise* dan kontras pada citra yang lebih kompleks..
2. Penelitian ini terbatas pada klasifikasi citra CXR untuk penyakit paru-paru dengan hanya mempertimbangkan 4 kelas saja, sehingga diharapkan dapat memperluas cakupan klasifikasi untuk penyakit paru-paru lainnya atau penyakit lain yang dapat didiagnosis melalui citra CXR..

DAFTAR PUSTAKA

- [1] N. Zhu *et al.*, “A Novel Coronavirus from Patients with Pneumonia in China, 2019,” *N. Engl. J. Med.*, vol. 382, no. 8, pp. 727–733, 2020.
- [2] X. Xu *et al.*, “A Deep Learning System to Screen Novel Coronavirus Disease 2019 Pneumonia,” *Engineering*, vol. 6, no. 10, pp. 1122–1129, 2020.
- [1] H. Munusamy, J. M. Karthikeyan, G. Shriram, S. Thanga Revathi, and S. Aravindkumar, “FractalCovNet architecture for COVID-19 Chest X-ray image Classification and CT-scan image Segmentation,” *Biocybern. Biomed. Eng.*, vol. 41, no. 3, pp. 1025–1038, 2021.
- [1] F. Song *et al.*, “Emerging Coronavirus 2019-nCoV Pneumonia,” *Radiology*, 2019.
- [2] I. Tinaliah and T. Elizabeth, “Peningkatan Kualitas Citra X-Ray Paru-Paru Pasien Covid-19 Menggunakan Metode Contrast Limited Adaptive Histogram Equalization,” *J. Teknol. Inf.*, vol. 4, no. 2, pp. 345–349, 2020.
- [2] S. S. Sumijan, A. W. Purnama, and S. Arlis, “Peningkatan Kualitas Citra CT-Scan dengan Penggabungan Metode Filter Gaussian dan Filter Median,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 6, p. 591, 2019.
- [2] A. Desiani, D. A. Zayanti, R. Primartha, F. Efriliyanti, and N. A. C. Andriani, “Variasi Thresholding untuk Segmentasi Pembuluh Darah Citra Retina,” *J. Edukasi dan Penelit. Inform. JEPIN*, vol. 7, no. 2, pp. 255–262, 2021.
- [8] B. Nugroho, W. S. J. Saputra, and E. Y. Puspaningrum, “Pengenalan Multi-Wajah Menggunakan Metode Ekstraksi Fitur Viola Jones Untuk Presensi Perkuliahan,” in *Seminar Nasional informatika (SANTIKA)*, 2019, pp. 6–11.
- [9] M. R. Khilmawan and A. A. Riadi, “Implementasi Pengurangan Noise pada Citra Tulang Menggunakan Metode Median Filter dan Gaussian Filter,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 3, no. 2, pp. 116–121, 2018.
- [2] M. M. Sebatubun, “Peningkatan Kualitas Citra X-Ray Paru-Paru Menggunakan Contrast Limited Adaptive Histogram Equalization dan Gaussian Filter,” in *Seminar Riset Teknologi Informasi (SRITI)*, 2016, pp. 241–247.
- [11] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, “A review of medical image data augmentation techniques for deep learning applications,” *J. Med. Imaging Radiat. Oncol.*, vol. 65, no. 5, pp. 545–563, 2021.
- [12] C. Shorten and T. M. Khoshgoftaar, “A Survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 60, 2019.
- [1] T. A. Soomro *et al.*, “Deep Learning Models for Retinal Blood Vessels Segmentation: A Review,” *IEEE Access*, vol. 7, pp. 71696–71717, 2019.

- [14] S. Ahuja, B. K. Panigrahi, N. Dey, V. Rajinikanth, and T. K. Gandhi, "Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices," *Appl. Intell.*, vol. 51, no. 1, pp. 571–585, 2021.
- [15] V. Kumar, T. Lal, P. Dhuliya, and D. Pant, "A study and comparison of different image segmentation algorithms," *Proc. - 2016 Int. Conf. Adv. Comput. Commun. Autom. (Fall), ICACCA 2016*, 2016.
- [16] M. Kiran, I. Ahmed, N. Khan, and A. G. Reddy, "Chest X-ray segmentation using Sauvola thresholding and Gaussian derivatives responses," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 10, pp. 4179–4195, 2019.
- [17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Springer Int. Publ.*, 2015.
- [18] S.-H. Wang *et al.*, "Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling," *Front. Neurosci.*, vol. 12, 2018.
- [19] X. Y. Zhou and G. Z. Yang, "Normalization in Training U-Net for 2-D Biomedical Semantic Segmentation," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1792–1799, 2019.
- [20] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proc. 32nd Int. Conf. Mach. Learn. Lille, Fr. 2015*, vol. 37, pp. 1–9, 2015.
- [21] N. M. Aboelenein, P. Songhao, A. Koubaa, A. Noor, and A. Afifi, "HTTU-Net: Hybrid Two Track U-Net for Automatic Brain Tumor Segmentation," *IEEE Access*, vol. 8, pp. 101406–101415, 2020.
- [22] L. Liu, L. Mou, X. X. Zhu, and M. Mandal, "Skin Lesion Segmentation Based on Improved U-net," *2019 IEEE Can. Conf. Electr. Comput. Eng. CCECE 2019*, pp. 1–4, 2019.
- [23] A. L. Aswathy, A. H. S., and V. C. Vinod, "COVID-19 diagnosis and severity detection from CT-images using transfer learning and back propagation neural network," *J. Infect. Public Health*, vol. 14, no. 10, pp. 1435–1445, 2021.
- [24] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer," *Procedia Comput. Sci.*, vol. 179, no. 2019, pp. 423–431, 2021.
- [25] E. Paladini, E. Vantaggiato, F. Bougourzi, C. Distanto, A. Hadid, and A. Taleb-Ahmed, "Two Ensemble-CNN Approaches for Colorectal Cancer Tissue Type Classification," *J. Imaging*, vol. 7, no. 3, p. 51, 2021.
- [26] A. Keles, M. B. Keles, and A. Keles, "COV19-CNNNet and COV19-ResNet: Diagnostic Inference Engines for Early Detection of COVID-19," *Cognit. Comput.*, no. 0123456789, 2021.

- [27] Z. Karhan and F. Akal, "Covid-19 Classification Using Deep Learning in Chest X-Ray Images," *TIPTEKNO 2020 - Tip Teknol. Kongresi - 2020 Med. Technol. Congr. TIPTEKNO 2020*, pp. 2021–2024, 2020.
- [28] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, "Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer," *Neurosci. Informatics*, vol. 2, no. 4, p. 100034, 2022.
- [29] M. Tan and Q. V Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 6105–6114.
- [30] S. Chaudhary, V. Jakhetiya, B. N. Subudhi, U. Baid, and S. C. Guntuku, "Detecting Covid-19 and Community Acquired Pneumonia Using Chest CT Scan Images with Deep Learning," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8583–8587.
- [31] X. Chen *et al.*, "Application of EfficientNet-B0 and GRU-based deep learning on classifying the colposcopy diagnosis of precancerous cervical lesions," *Cancer Med.*, no. July 2022, pp. 1–10, 2023.
- [32] C. Wang *et al.*, "Pulmonary image classification based on inception-v3 transfer learning model," *IEEE Access*, vol. 7, pp. 146533–146541, 2019.
- [33] W. Li *et al.*, "Classification of High-Spatial-Resolution Remote Sensing Scenes Method Using Transfer Learning and Deep Convolutional Neural Network," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, no. X, pp. 1986–1995, 2020.
- [34] S. Sharma, "COVID-19 prediction from chest X-ray images using deep convolutional neural network," *Artif. Intell. Mach. Learn. EDGE Comput.*, no. March 2020, pp. 315–324, 2022.
- [35] M. Mujahid, F. Rustam, R. Álvarez, J. Luis Vidal Mazón, I. de la T. Díez, and I. Ashraf, "Pneumonia Classification from X-ray Images with Inception-V3 and Convolutional Neural Network," *Diagnostics*, vol. 12, no. 5, pp. 1–16, 2022.
- [36] J. Cao, M. Yan, Y. Jia, X. Tian, and Z. Zhang, "Application of a modified Inception-v3 model in the dynasty-based classification of ancient murals," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 1, 2021.
- [37] J. Gao, C. Leung, and C. Miao, "Diabetic Retinopathy Classification Using an Efficient Convolutional Neural Network," *Proc. - 2019 IEEE Int. Conf. Agents, ICA 2019*, pp. 80–85, 2019.
- [38] V. C. Osamor and A. F. Okezie, "Enhancing the weighted voting ensemble algorithm for tuberculosis predictive diagnosis," *Sci. Rep.*, vol. 11, no. 1, pp. 1–12, 2021.
- [39] A. Desiani *et al.*, "Majority Voting as Ensemble Classifier for Cervical Cancer Classification," *Sci. Technol. Indones.*, vol. 8, no. 1, pp. 84–92, 2023.

- [40] A. Das, "Adaptive UNet-based Lung Segmentation and Ensemble Learning with CNN-based Deep Features for Automated COVID-19 Diagnosis," *Multimed. Tools Appl.*, vol. 81, no. 4, pp. 5407–5441, 2022.
- [41] S. A. Siddiqui, N. Fatima, and A. Ahmad, "Chest X-ray and CT Scan Classification using Ensemble Learning through Transfer Learning," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 22, no. 6, pp. 1–9, 2022.
- [42] T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, and Y. Dong, "The Ensemble Deep Learning Model for Novel COVID-19 on CT Images," *Appl. Soft Comput.*, vol. 98, p. 106885, 2021.
- [43] R. Kundu, R. Das, Z. W. Geem, G. T. Han, and R. Sarkar, "Pneumonia Detection in Chest X-Ray Images using an Ensemble of Deep Learning Models," *PLoS One*, vol. 16, no. 9 September, 2021.
- [44] K. Khaksari *et al.*, "Review of the efficacy of infrared thermography for screening infectious diseases with applications to COVID-19," *J. Med. Imaging*, vol. 8, no. S1, pp. 1–15, 2021.
- [45] K. Yuki, M. Fujiogi, and S. Koutsogiannaki, "COVID-19 pathophysiology: A review," *Clin. Immunol.*, vol. 215, no. April, 2020.
- [46] S. Sehrawat and B. T. Rouse, "COVID-19: disease, or no disease? - that is the question. It's the dose stupid!," *Microbes Infect.*, vol. 23, no. 1, p. 104779, 2021.
- [47] M. Rahimzadeh, A. Attar, and S. M. Sakhaei, "A fully automated deep learning-based network for detecting COVID-19 from a new and large lung CT scan dataset," *Biomed. Signal Process. Control*, vol. 68, no. December 2020, p. 102588, 2021.
- [48] Y. Tang, J. E. Schmitz, D. H. Persing, and C. W. Stratton, "Laboratory Diagnosis of COVID-19: Current Issues and Challenges," *J. Clin. Microbiol.*, vol. 58, no. 6, pp. 1–9, 2020.
- [49] F. Song *et al.*, "Emerging 2019 Novel Coronavirus (2019-NCoV) Pneumonia," *Radiology*, vol. 295, no. 1, pp. 210–217, 2020.
- [50] N. Z. Munantri, H. Sofyan, and M. Y. Florestiyanto, "Aplikasi Pengolahan Citra Digital Untuk Identifikasi Umur Pohon," *Telematika*, vol. 16, no. 2, p. 97, 2020.
- [51] I. R. S. Valente, P. C. Cortez, E. C. Neto, J. M. Soares, V. H. C. de Albuquerque, and J. M. R. S. Tavares, "Automatic 3D pulmonary nodule detection in CT images: A survey," *Comput. Methods Programs Biomed.*, vol. 124, pp. 91–107, 2016.
- [52] A. Pumama, "Aplikasi matriks dalam pengolahan gambar," *Makal. IF2123 Aljabar*, no. May, p. 18650054, 2016.
- [53] A. Tahir *et al.*, "Coronavirus : Comparing COVID-19 , SARS, and MERS in The Eyes of AI," *arXiv*, pp. 1–35, 2021.

- [54] M. A. Kamel, M. Abdelshafy, M. AbdulRazek, O. Abouelkhir, A. Fawzy, and A. T. Sahlol, "Efficient Classification Approach Based on COVID-19 CT Images Analysis with Deep Features," in *International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021.
- [55] Nurhidayah, B. Abdul Samad, and B. Abdullah, "Perbandingan Metode Contrast Enhancement pada Citra CT-Scan Kanker Paru-paru," *Gravitasi*, vol. 19, no. 2, pp. 24–28, 2020.
- [56] M. Wulandari, "Index Quality Assesment Citra Terinterpolasi (SSIM dan FSIM)," *J. Terap. Teknol. Inf.*, vol. 1, no. 1, pp. 11–20, 2017.
- [57] A. Ramadhan, F. Mahmood, and A. Elci, "Image Denoising by Median Filter in Wavelet Domain," *Int. J. Multimed. Its Appl.*, vol. 9, no. 1, pp. 31–40, 2017.
- [58] U. Sara, M. Akter, and M. S. Uddin, "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study," *J. Comput. Commun.*, vol. 07, no. 03, pp. 8–18, 2019.
- [59] M. Elgendi *et al.*, "The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19 : A Geometric Transformation Perspective," *Front. Med.*, vol. 8, no. March, pp. 1–12, 2021.
- [60] M. M. A. Monshi, J. Poon, V. Chung, and F. M. Monshi, "CovidXrayNet: Optimizing Data Augmentation and CNN Hyperparameters for Improved COVID-19 Detection from CXR," *Comput. Biol. Med.*, vol. 133, no. March, p. 104375, 2021.
- [61] D. Keidar, D. Yaron, E. Goldstein, Y. Shachar, A. Blass, and L. Charbinsky, "COVID-19 Classification of X-Ray Images Using Deep Neural Networks," *Eur. Radiol.*, vol. 31, pp. 9654–9663, 2021.
- [62] I. R. I. Haque and J. Neubert, "Deep learning approaches to biomedical image segmentation," *Informatics Med. Unlocked*, p. 100297, 2020.
- [63] A. Desiani, B. Suprihatin, S. Yahdin, A. I. Putri, and F. R. Husein, "Bi - path Architecture of CNN Segmentation and Classification Method for Cervical Cancer Disorders Based on Pap - smear Images," *IAENG Int. J. Comput. Sci.*, vol. 48, no. 3, 2021.
- [64] Y. Jalali, M. Fateh, M. Rezvani, V. Abolghasemi, and M. H. Anisi, "ResBCDU-Net: A Deep Learning Framework for Lung CT Image Segmentation," *Sensors*, vol. 21, no. 1, pp. 1–24, 2021.
- [65] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An Improved Method to Construct basic Probability Assignment based on The Confusion Matrix for Classification Problem," *Inf. Sci. (Ny)*, vol. 340–341, pp. 250–261, 2016.
- [66] J. Rasheed, A. A. Hameed, C. Djeddi, A. Jamil, and F. Al-Turjman, "A Machine Learning-based Framework for Diagnosis of COVID-19 from Chest X-ray Images," *Interdiscip. Sci. Comput. Life Sci.*, vol. 13, no. 1, pp. 103–117, 2021.

- [67] M. A. Mohammed *et al.*, “A Comprehensive Investigation of Machine Learning Feature Extraction and Classification Methods for Automated Diagnosis of COVID-19 based on X-ray Images,” *Comput. Mater. Contin.*, vol. 66, no. 3, pp. 3289–3310, 2021.
- [68] G. Marques, D. Agarwal, and I. de la Torre Díez, “Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network,” *Appl. Soft Comput. J.*, vol. 96, p. 106691, 2020.
- [69] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition,” *Insights Imaging*, vol. 9, no. 4, pp. 21–30, 2018.
- [70] S. Bharati, P. Podder, M. R. H. Mondal, and V. B. S. Prasath, “CO-ResNet: Optimized ResNet model for COVID-19 diagnosis from X-ray images,” *Int. J. Hybrid Intell. Syst.*, vol. 17, no. 1–2, pp. 71–85, 2021.
- [71] J. Shi *et al.*, “Improvement of Damage Segmentation Based on Pixel-Level Data Balance using VGG-UNet,” *Appl. Sci.*, vol. 11, no. 2, pp. 1–17, 2021.
- [72] W. Chen, B. Yang, J. Li, and J. Wang, “An Approach to Detecting Diabetic Retinopathy based on Integrated Shallow Convolutional Neural Networks,” *IEEE Access*, vol. 8, pp. 178552–178562, 2020.
- [73] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, France: JMLR: W&CP, 2015, pp. 448–456.
- [74] S. Wang, K. Muhammad, J. Hong, A. Kumar, S. Y. Zhang, and S. Wang, “Alcoholism Identification via Convolutional Neural Network based on Parametric ReLU, Dropout, and Batch Normalization,” *Neural Comput. Appl.*, vol. 32, pp. 665–680, 2018.
- [75] Y. Tian, “Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm,” *IEEE Access*, vol. 8, pp. 125731–125744, 2020.
- [76] L. Alzubaidi *et al.*, “Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions,” *J. Big Data*, vol. 8, no. 53, 2021.
- [77] H. Gholamalinezhad and H. Khosravi, “Pooling Methods in Deep Neural Networks, a Review,” *arXiv*, 2020.
- [78] D. Im, D. Han, S. Choi, S. Kang, and H. J. Yoo, “DT-CNN: an energy-efficient dilated and transposed convolutional neural network processor for region of interest based image segmentation,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 67, no. 10, pp. 3471–3483, 2020.
- [79] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Shi, “An Embedding Cost Learning Framework Using,” *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 839–851, 2019.

- [80] T. A. Soomro, O. Hellwich, A. J. Afifi, M. Paul, J. Gao, and L. Zheng, "Strided U-Net Model: Retinal Vessels Segmentation using Dice Loss," *2018 Int. Conf. Digit. Image Comput. Tech. Appl. DICTA 2018*, pp. 1–8, 2018.
- [81] Q. Li *et al.*, "Tumor Segmentation in Contrast-Enhanced Magnetic Resonance Imaging for Nasopharyngeal Carcinoma: Deep Learning with Convolutional Neural Network," *Biomed Res. Int.*, vol. 2018, 2018.
- [82] G. Brain, A. N. Gomez, and G. Brain, "Depthwise Separable Convolutions for Neural Machine Translation," in *International Conference on Learning Representations (ICLR) 2018*, 2018.
- [83] R. Stolkin, "Dense Connection and Depthwise Separable Convolution based CNN for Polarimetric SAR Image Classification," *Knowledge-Based Syst.*, vol. 194, 2020.
- [84] Z. Younas and Z. Niu, "CNN with Depthwise Separable Convolutions and Combined Kernels for Rating Prediction," *Expert Syst. Appl.*, vol. 170, p. 114528, 2021.
- [85] D. R. Nayak, N. Padhy, P. K. Mallick, M. Zymbler, and S. Kumar, "Brain Tumor Classification Using Dense Efficient-Net," *Axioms*, vol. 11, no. 1, 2022.
- [86] H. Salehinejad and S. Valaee, "ISING-DROPOUT : A REGULARIZATION METHOD FOR TRAINING AND COMPRESSION OF DEEP NEURAL NETWORKS," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3602–3606.
- [87] S. Lian, L. Li, G. Lian, X. Xiao, Z. Luo, and S. Li, "A Global and Local Enhanced Residual U-Net for Accurate Retinal Vessel Segmentation," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 14, no. 8, pp. 1–10, 2019.
- [88] H. Alaiz-moreton, J. Aveleira-mata, J. Ondicol-garcia, A. L. Muñoz-castañeda, and C. Benavides, "Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol," *Hindawi*, vol. 2019, 2019.
- [89] H. Shaziya, K. Shyamala, and R. Zaheer, "Automatic Lung Segmentation on Thoracic CT Scans using U-Net Convolutional Network," in *2018 International Conference on Communication and Signal Processing (ICCSP)*, IEEE, 2018, pp. 643–647.
- [90] Q. Ji, J. Huang, W. He, and Y. Sun, "Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images," *Algorithms*, vol. 12, no. 3, pp. 1–12, 2019.
- [91] H. Alhichri and A. S. Alswayed, "Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model With Attention," *IEEE Access*, vol. 9, pp. 14078–14094, 2021.
- [92] T. Ahmed, "Classification and Understanding of Cloud Structures via Satellite Images with EfficientUNet," *SN Comput. Sci.*, vol. 3, no. 99, 2022.

- [93] C. Szegedy *et al.*, "Going Deeper with Convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [94] A. Gupta, Anjum, S. Gupta, and R. Katarya, "InstaCovNet-19: A Deep Learning Classification Model for The Detection of COVID-19 Patients Using Chest X-ray," *Appl. Soft Comput.*, vol. 99, no. xxxx, p. 106859, 2021.
- [95] C. Land, C. Mapping, U. Multispectral, and R. S. Imagery, "Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery," *Remote Sens.*, vol. 10, no. 7, 2018.
- [96] F. Ahmad, A. Farooq, and M. U. Ghani, "Deep Ensemble Model for Classification of Novel Coronavirus in Chest X-Ray Images," *Comput. Intell. Neurosci.*, vol. 2021, 2021.
- [97] J. F. Hern, I. Researcher, J. Chardin, and S. Borja, "An Ensemble Approach for Multi-Stage Transfer Learning Models for COVID-19 Detection from Chest CT Scans," *Intell. Med.*, vol. 5, no. 2020, 2021.
- [98] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies*, vol. 9, no. 4, p. 81, 2021.
- [99] A. Desiani, R. Primartha, M. Arhami, and O. Orsalan, "Naive Bayes Classifier for Infant Weight Prediction of Hypertension Mother," *J. Phys. Conf. Ser.*, vol. 1282, no. 1, 2019.
- [100] D. Chicco, M. J. Warrens, and G. Jurman, "The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment," *IEEE Access*, vol. 9, pp. 78368–78381, 2021.
- [101] F. Li and H. He, "Assessing the Accuracy of Diagnostic Tests," *Shanghai Arch. Psychiatry*, vol. 30, no. 3, pp. 207–212, 2018.
- [102] A. R. Susanto, "Analisis Perbandingan Metode Perbaikan Kontras Citra pada Citra Medis Xray," *J. Inform.*, vol. 12, no. 1, pp. 1–9, 2018.
- [103] D. Umamaheswari and S. Geetha, "Visual Quality and Illumination Enhancement Using Gamma Corrected Gaussian Filtering Framework for Covid-19 Images," *ICTACT J. Image Video Process.*, vol. 11, no. 1, pp. 2268–2274, 2020.
- [104] A. Roy and P. Maity, "A Comparative Analysis of Various Filters to Denoise Medical X-Ray Images," in *2020 4th International Conference on Electronics, Materials Engineering and Nano-Technology, IEMENTech 2020*, 2020, pp. 20–24.
- [105] M. Abdel-Basset, N. N. Mostafa, K. M. Sallam, I. Elgendi, and K. Munasinghe, "Enhanced COVID-19 X-ray Image Preprocessing Schema Using Type-2 Neutrosophic Set," *Appl. Soft Comput.*, vol. 123, p. 108948,

2022.

- [106] L. I. Kesuma, Ermatita, Erwin, P. Sari, and R. H. Purabaya, "Improved Chest X-Ray Image Quality Using Median and Gaussian Filter Methods," in *2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, Jakarta: IEEE, 2022.
- [107] G. Tong, Y. Li, H. Chen, Q. Zhang, and H. Jiang, "Improved U-NET Network for Pulmonary Nodules Segmentation," *Optik (Stuttg.)*, vol. 174, no. August, pp. 460–469, 2018.
- [108] A. G. Smith, J. Petersen, R. Selvan, and C. R. Rasmussen, "Segmentation of Roots in Soil with U-Net," *Plant Methods*, vol. 16, no. 1, pp. 1–15, 2020.
- [109] B. A. Akinnuwesi *et al.*, "Application of intelligence-based computational techniques for classification and early differential diagnosis of COVID-19 disease," *Data Sci. Manag.*, vol. 4, no. December, pp. 10–18, 2021.
- [110] S. Kadry, V. Rajinikanth, S. Rho, N. Sri, and M. Raja, "Development of a Machine-Learning System to Classify Scan Images into Normal / COVID-19 Class," vol. 2, no. December, 2019.
- [111] D. N. Vinod and S. R. S. Prabaharan, "Data science and the role of Artificial Intelligence in achieving the fast diagnosis of Covid-19," vol. 140, 2020.
- [112] S. K. Sharma, U. K. Lilhore, S. Simaiya, and N. K. Trivedi, "An improved random forest algorithm for predicting the COVID-19 pandemic patient health," *Ann. Rom. Soc. Cell Biol.*, vol. 25, no. 1, pp. 67–75, 2021.
- [113] X. Wu *et al.*, "Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: A multicentre study," *Eur. J. Radiol.*, vol. 128, no. March, pp. 1–9, 2020.
- [114] V. Shah, R. Keniya, A. Shridharani, M. Punjabi, J. Shah, and N. Mehendale, "Diagnosis of COVID-19 using CT scan images and deep learning techniques," *Emerg. Radiol.*, vol. 28, no. 3, pp. 497–505, 2021.
- [115] S. Albahli, N. Ayub, and M. Shiraz, "Coronavirus disease (COVID-19) detection using X-ray images and enhanced DenseNet," *Appl. Soft Comput.*, vol. 110, p. 107645, 2021.
- [116] S. R. D. Dimas, B. S. Negara, S. Sanjaya, and E. Satria, "COVID-19 Classification for Chest X-Ray Images using Deep Learning and Resnet-101," in *2021 International Congress of Advanced Technology and Engineering, ICOTEN 2021*, IEEE, 2021, pp. 20–23.
- [117] T. Anwar and S. Zakir, "Deep learning based diagnosis of COVID-19 using chest CT-scan images," *Proc. - 2020 23rd IEEE Int. Multi-Topic Conf. INMIC 2020*, pp. 6–10, 2020.
- [118] R. A. Pratiwi, S. Nurmaini, D. P. Rini, M. N. Rachmatullah, and A. Darmawahyuni, "Deep Ensemble Learning for Skin Lesions Classification with Convolutional Neural Network," *IAES Int. J. Artif. Intell.*, vol. 10, no.

- 3, pp. 563–570, 2021.
- [119] S. Tang *et al.*, “EDL-COVID: Ensemble Deep Learning for COVID-19 Case Detection from Chest X-Ray Images,” *IEEE Trans. Ind. Informatics*, vol. 17, no. 9, pp. 6539–6549, 2021.
- [120] P. Shrivastava, A. Singh, S. Agarwal, H. Tekchandani, and S. Verma, “Covid Detection in CT and X-Ray Images using Ensemble Learning,” in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, 2021, pp. 1085–1090.
- [121] T. K. K. Ho and J. Gwak, “Feature-Level Ensemble Approach for COVID-19 Detection using Chest X-ray Images,” *PLoS One*, vol. 17, no. 7 July, pp. 1–19, 2022.
- [122] L. Visuna, D. Yang, J. Garcia-Blas, and J. Carretero, “Computer-Aided Diagnostic for Classifying Chest X-ray Images using Deep Ensemble Learning,” *BMC Med. Imaging*, vol. 22, no. 1, pp. 1–16, 2022.
- [123] L. I. Kesuma, Ermatita, and Erwin, “ELREI : Ensemble Learning of ResNet , EfficientNet , and Inception-v3 for Lung Disease Classification based on Chest X-Ray Image,” *Int. J. Intell. Eng. Syst.*, vol. 16, no. 5, pp. 149–161, 2023.
- [124] ² E. Allam Zanaty and S. Mostafa Ibrahim, “Medical Image Compression Based on Combining Region Growing and Wavelet Transform,” *Int. J. Med. Imaging*, vol. 7, no. 3, pp. 57–65, 2019.

LAMPIRAN

Lampiran 1. Bukti Accepted ICIMCIS



4th International Conference Informatics, Multimedia, Cyber and Information System

AI-Powered Solutions in transformed Business and Society into Super Smart Society 5.0

No : 103/UN61/FIK/2022 Jakarta, November 8th, 2022
Subject : Letter of Acceptance as Submission
at ICIMCIS 2022

Lucky Kesuma
Universitas Sjakhyakirti Palembang, Indonesia

Dear author,

Congratulation, your final manuscript submission #1570852024 with the title **(Improved Chest X-Ray Image Quality Using Median and Gaussian Filter Methods)** has been ACCEPTED as part of the technical program for the 2022 International Conference on Informatics, Multimedia, Cyber and Information Systems (ICIMCIS). Hopefully, this paper will be presented (oral or online) at the conference as our technical program reports. The conference organizers and our attendees expect each and every accepted paper to be presented which contributes to the success of the event.

Your final submission must adhere to the IEEE PDF compliance which is checked through <https://2022.icimcis.org/final-camera-ready-guidelines/>. To confirm attendance at the conference, please complete registration through the link below:

<https://bit.ly/ICIMCIS2022-Registration-Form>

Please be noticed, only registered and presented papers will proceed to IEEE explore. Upon receipt of your registration, we appreciate taking this opportunity to thank you for your participation in the 4th ICIMCIS 2022. Please do not hesitate to contact us for further information or visit our link below :

<https://2022.icimcis.org/>

We look forward to meeting you at the 4th ICIMCIS on November 16th - 17th, 2022 at **Virtual Conference, in Horison Grand Serpong, Indonesia**. For more info please join this WhatsApp Group:

<https://chat.whatsapp.com/EiZ3X7dRHFcBbbYgNGufNE>



Lampiran 2. Bukti Accepted Jurnal IJIES

Intelligent Networks and Systems Society

Acceptance Letter

International Journal of Intelligent Engineering and Systems (IJIES)

June 27, 2023

Dear Erwin,

Manuscript Title: ELREI: Ensemble Learning of ResNet, EfficientNet, and Inception-v3 for Lung Disease Classification based on Chest X-Ray Image

Author(s): Lucky Indra Kesuma, Ermatita, Erwin

Thank you for submitting your paper to the International Journal of Intelligent Engineering and Systems (IJIES). Based on double blind review process, we are pleased to inform you that our Review Committee has accepted your paper.

The paper will be included in the IJIES, which will be published with ISSN (ISSN: 2185-3118) in online on the website (<http://www.inass.org/publications.html>). We are looking forward to your further contribution to our journal.

Kind regards

Prof. Dr. Kei EGUCHI
Editor-in-Chief, International Journal of Intelligent Engineering and Systems

Department of Information Electronics
Fukuoka Institute of Technology

A handwritten signature in black ink that reads "Kei Eguchi".

E-mail: ijies@inass.org

MODEL MULTICLASS CLASSIFICATION UNTUK PENYAKIT BERDASARKAN CITRA CHEST X-RAY PARU-PARU DENGAN ENSEMBLE LEARNING

ORIGINALITY REPORT

9%

SIMILARITY INDEX

8%

INTERNET SOURCES

4%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1	comengapp.unsri.ac.id Internet Source	3%
2	Lucky Indra Kesuma, Ermatita Ermatita, Erwin Erwin, Purwita Sari, Rudhy Ho Purabaya. "Improved Chest X-Ray Image Quality Using Median and Gaussian Filter Methods", 2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), 2022 Publication	2%
3	repository.unsri.ac.id Internet Source	1%
4	publikasi.dinus.ac.id Internet Source	1%
5	Submitted to Sriwijaya University Student Paper	1%
6	ojs.uma.ac.id Internet Source	1%

Exclude quotes Off

Exclude matches < 1%

Exclude bibliography Off