

# Deteksi Penyakit Paru Melalui Citra X-Ray Menggunakan *Convolutional Neural Network* *Densenet121* Berbasis *Transfer Learning*

Abdul Malik<sup>1</sup>, Handri Taufik<sup>2</sup>, Richi Setiawan<sup>3</sup>, Tanti Cahya Herdiyani<sup>4</sup>, Toufiq Hidayat<sup>5</sup>

<sup>1,2,3,4</sup>Program Studi Teknik Infomatika S-2, Universitas Pamulang

e-mail: axmalik@yahoo.com<sup>1</sup>, handrith@gmail.com<sup>2</sup>, moijoyo.info@gmail.com<sup>3</sup>,

tanticahyah@gmail.com<sup>4</sup>, tofhid19@gmail.com<sup>5</sup>

**Abstrak**— Penelitian ini bertujuan untuk mengembangkan sistem deteksi otomatis penyakit paru, khususnya tuberkulosis dan beberapa penyakit paru lainnya, melalui citra X-ray menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur DenseNet121 berbasis *transfer learning*. Pendekatan *transfer learning* dipilih untuk memanfaatkan bobot awal yang telah dilatih pada dataset besar *ImageNet*, sehingga proses pelatihan menjadi lebih efisien dan efektif meskipun dengan dataset medis yang relatif terbatas. Dataset citra X-ray yang digunakan terdiri dari empat kelas penyakit, yaitu Tuberkulosis, Viral Pneumonia, Lung Opacity, dan Normal. Dataset tersebut dibagi menjadi data pelatihan, validasi, dan pengujian untuk memastikan evaluasi model yang akurat. Proses pelatihan dilakukan dengan teknik augmentasi data untuk meningkatkan keragaman data dan mencegah overfitting. Hasil eksperimen menunjukkan bahwa model DenseNet121 mampu mencapai akurasi validasi sebesar 94,44%, dengan kemampuan klasifikasi yang baik terhadap keempat kelas penyakit paru. Sistem yang dikembangkan dapat digunakan sebagai alat bantu dalam diagnosis penyakit paru berbasis citra X-ray yang cepat dan akurat, mendukung proses pengambilan keputusan medis.

**Kata kunci:** *Convolutional Neural Network* (CNN), *DenseNet121*, *Transfer learning*.

## I. PENDAHULUAN

Penyakit paru seperti tuberkulosis, *viral pneumonia*, dan *lung opacity* masih menjadi masalah kesehatan global yang serius, terutama di negara berkembang. Tingginya angka sakit dan kematian akibat penyakit ini menunjukkan pentingnya deteksi dan penanganan dini yang akurat. Salah satu metode diagnosis yang paling umum digunakan adalah melalui pencitraan radiologis, khususnya *X-ray* dada, karena prosedurnya yang cepat, murah, dan tersedia secara luas di fasilitas layanan kesehatan.

Namun demikian, interpretasi citra *X-ray* tidak selalu mudah dan sering kali memerlukan tenaga ahli radiologi yang terlatih. Proses diagnosis yang bergantung pada kemampuan subjektif manusia berisiko mengalami kesalahan interpretasi, apalagi dalam kondisi beban kerja yang tinggi atau keterbatasan sumber daya manusia. Oleh karena itu, diperlukan dukungan teknologi yang mampu membantu menganalisis citra medis secara otomatis, cepat, dan akurat.

Dengan kemajuan pesat dalam bidang kecerdasan buatan, khususnya *deep learning* dan *Convolutional Neural Network* (CNN), pendekatan baru dalam klasifikasi citra medis. *Transfer learning* memungkinkan pemanfaatan model pra-latih seperti DenseNet121 yang telah dilatih pada dataset besar seperti *ImageNet* untuk digunakan kembali dalam tugas klasifikasi medis. Pendekatan ini sangat cocok digunakan pada domain seperti citra *X-ray*, di mana ketersediaan data berlabel sering kali terbatas.

Penelitian ini bertujuan untuk membangun model deteksi penyakit paru berbasis CNN menggunakan DenseNet121 yang dilatih ulang melalui metode *transfer learning*. Model dikembangkan untuk mengklasifikasikan citra *X-ray* ke dalam empat kategori yaitu, Tuberkulosis, *Viral Pneumonia*, *Lung Opacity*, dan Normal. Melalui pengujian terhadap dataset yang telah dibagi secara proporsional menjadi data latih, validasi, dan uji, diharapkan sistem ini dapat mendukung proses diagnosis yang lebih akurat dan efisien.

## II. METODE PENELITIAN

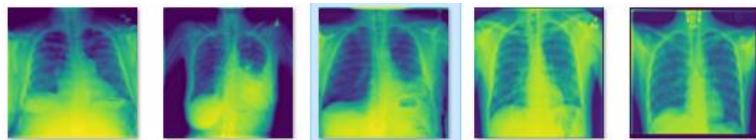
Metode penelitian ini menjelaskan tahapan-tahapan yang dilakukan dalam mengimplementasikan model *Convolutional Neural Network* DenseNet121 dengan menggunakan google collab untuk mendeteksi penyakit paru dari citra *X-Ray*. Tahapan yang dilakukan adalah sebagai berikut :

1. Persiapan Dataset. Pada tahapan ini dilakukan pengumpulan dataset yang didapatkan dari dua sumber data, yaitu dataset gambar paru-paru dengan kelas tuberculosis di dapat dari <https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis>

tb-chest-xray-dataset, dan untuk data paru-paru dengan kelas *Lung\_Opacity*, Normal, serta *Viral\_Pneumonia* didapat dari <https://www.kaggle.com/datasets/fatemehmehrparvar/lung-disease>. Adapun detail dataset ini adalah sebagai berikut:

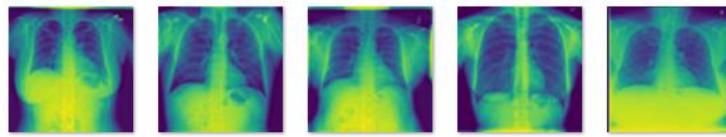
Tuberculosis : 700 image, png  
*Lung\_Opacity* : 1125 image, jpg  
 Normal : 1250 image, jpg  
*Viral\_Pneumonia*: 1100 image, jpg

Image paru paru dengan kondisi *Lung\_Opacity* terlihat pada Gambar 1.



Gambar 1. *Lung\_Opacity*

Image paru paru dengan kondisi Normal tanpa diagnose penyakit apapun terlihat pada Gambar 2.



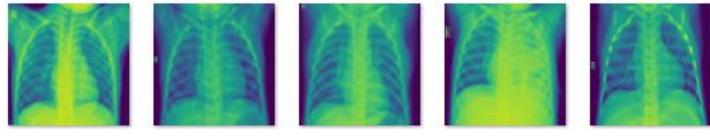
Gambar 2. Normal

Image paru paru dengan kondisi Tuberculosis terlihat pada Gambar 3.



Gambar 3. Tuberculosis

Image paru paru dengan kondisi *Viral\_Pneumonia* terlihat pada Gambar 4.



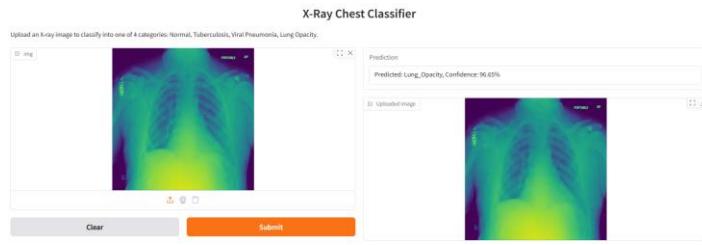
Gambar 4. *Viral\_Pneumonia*

2. Pembagian Dataset. Pada tahapan ini dilakukan pembagian/*split* dataset menjadi latih/*train* (70%), val (15%) dan test (15%) pada masing-masing kelas. Adapun tujuannya adalah dataset latih/*train* untuk melatih model, dataset val untuk mengevaluasi selama proses pelatihan, sedangkan dataset test digunakan untuk ujicoba model, apakah dari gambar test yang kita upload model dapat mengenali kelasnya apa dan nilai confidencenya berapa persen. Berikut detail dataset setelah dilakukan pembagian data :

Tuberculosis : 489 image (*train*), 105 image (*val*), 106 image (*test*)  
*Lung\_Opacity* : 787 image (*train*), 168 image (*val*), 170 image (*test*)  
 Normal : 875 image (*train*), 187 image (*val*), 188 image (*test*)  
*Viral\_Pneumonia*: 770 image (*train*), 165 image (*val*), 165 image (*test*)

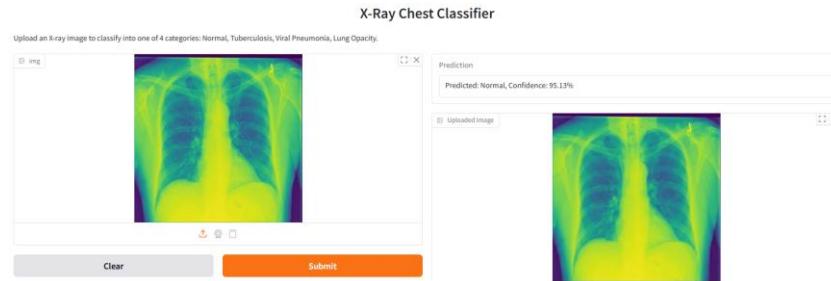
3. Pra-Pemrosesan Data (*Preprocessing*), dilakukan sebelum data diproses oleh model CNN DenseNet121, baik pada dataset *train* maupun pada dataset val. Tujuannya yaitu untuk memastikan bahwa data masuk ke model dalam format, ukuran, dan distribusi nilai yang sesuai. Berikut tahapan *preprocessing* yang dilakukan :
  - a. *Resize* ukuran. DenseNet121 hanya menerima gambar dengan ukuran 224x224 piksel (ukuran ini tidak terlalu besar sehingga cepat diproses GPU dan tidak terlalu kecil sehingga masih mengandung fitur visual penting pada X-Ray paru-paru). Sementara data gambar asli X-Ray itu memiliki ukuran yang besar 1024x1024 piksel bahkan lebih, tergantung dari alat X-Ray yang digunakan.

- b. *Augmentation*, dilakukan menggunakan *RandomHorizontalFlip* (membalikkan gambar secara horizontal), teknik ini dipilih karena aman untuk gambar *X-Ray* (tidak mengubah makna dan arti medis), efektif menambah variasi data, dan tidak mengubah struktur anatomii penting.
  - c. Standarisasi *channel* input dengan mengubah gambar ke format RGB. Hal ini perlu dilakukan karena DenseNet121 dilatih dengan gambar RGB 3 *channel* dari ImageNet, sementara pada dataset ada gambar yang pakai hitam putih/*grayscale* (1 *channel*) dan ada yang pakai RGB (3 *channel*). Oleh karena itu standarisasi *channel* input ini perlu dilakukan agar konsisten, tidak *error* dan sesuai dengan *pretrained* model yang digunakan yaitu RGB (3 *channel*).
  - d. Konversi gambar menjadi tensor PyTorch, Mengubah gambar dari format Python *Imaging Library* (PIL) menjadi tensor agar bisa dikenali dan diproses oleh model.
4. *Dataloader*. Pada tahapan ini, dataset yang sudah dilakukan *preprocessing* kemudian dimasukkan ke *dataloader* sebelum masuk sebagai inputan model CNN DenseNet121, tujuannya adalah untuk pengaturan distribusi data ke model dapat dilakukan menjadi *batch* otomatis, *shuffle*, *multithreading*, sehingga inputan ke model bisa efisien dan optimal.
5. *Load* Model CNN DenseNet121. Pada tahapan ini tidak membuat CNN dari awal, melainkan menggunakan *transfer learning* dari model DenseNet121 yang mengklasifikasikan dataset ImageNet sebanyak 1000 kelas. Namun, yang akan digunakan pada penelitian ini hanya sebanyak 4 kelas, sesuai dengan klasifikasi pada dataset yang ada dan telah disesuaikan dengan tujuan penelitian.
6. Pelatihan model, berikut tahapan:
- a. *Loss Function* dan *Optimizer*, menggunakan *CrossEntropyLoss* untuk klasifikasi dengan membandingkan *output* dengan label aslinya dan *optimizer* Adam untuk update bobotnya agar hasil prediksi yang didapatkan semakin baik.
  - b. Ulang/*Loop* proses pelatihan untuk dataset *training*. Proses *training* akan diulang sebanyak *num\_epoch* yaitu 5 kali dan dalam 1 tahapan *epoch* akan melihat seluruh dataset *training* satu kali penuh. Proses yang dilakukan adalah :
    - a. Dataset diproses sebanyak 8 gambar sekaligus per *batch*, cukup kecil mengingat keterbatasan *memory* RAM yang disediakan oleh Colab (versi gratis hanya sebesar 12 GB).
    - b. Membaca gambar dari *train\_loader* (*dataloader*), data langsung dikirim ke GPU jika tersedia
    - c. Prediksi oleh model
    - d. Hitung *loss*, membandingkan prediksi dengan label aslinya
    - e. *Backpropagation*, model belajar dari kesalahan prediksi dengan menghitung gradien (turunan) dari fungsi *loss*
    - f. *Optimizer*, memperbarui bobot model agar pada *batch* berikutnya prediksi bisa lebih akurat
    - g. Hitung metrix evaluasi selama 1 *epoch*
    - h. Mengukur kinerja model setelah semua *batch* selesai diproses
  - c. *Loop validation* (val), dimana prosesnya dipisah dari *loop* pelatihan (train) tujuannya adalah untuk mengukur generalisasi model ke data yang belum pernah dilihat sebelumnya (tanpa mempengaruhi bobot model). Tahapannya sama dengan *loop* pelatihan namun perbedaannya pada tidak ada update bobot, tidak ada *backward pass*.
7. Simpan model terbaik berdasarkan val yang memiliki akurasi paling tinggi.
8. *Load* model terbaik yang berhasil disimpan pada saat proses *valuation*.
9. Untuk *testing* kita gunakan dataset *test* yang belum pernah digunakan sama sekali oleh model, baik dalam pelatihan model (*train*), maupun *validation* (val). Hal ini dilakukan untuk menguji apakah hasil dari model sesuai label, dan persentase tingkat akurasinya.



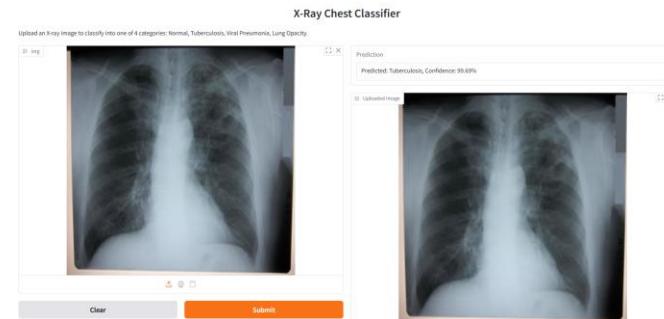
Gambar 5. Test Lung\_Opacity

Gambar 5. di atas, menunjukkan proses testing sebuah image dan dieksekusi hasilnya. Hasil testing menunjukkan bahwa image yang dimasukan terdeteksi sebagai image dengan kondisi *Lung Opacity*. Sedangkan pada Gambar 6, hasil test image menunjukkan bahwa kondisi paru-paru dalam keadaan Normal.

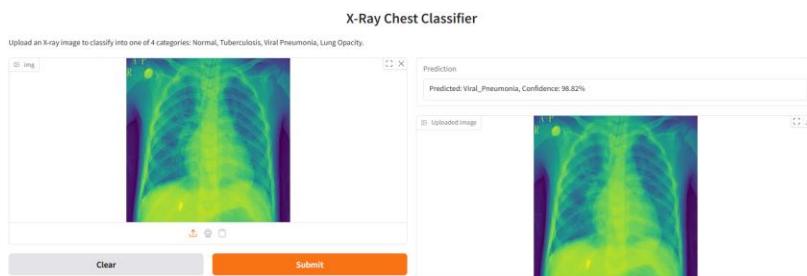


Gambar 6. Test Normal

Masih dengan pola testing yang sama, Gambar 7. di bawah, menunjukkan proses testing yang menunjukkan hasil bahwa image yang dimasukan terdeteksi sebagai image dengan kondisi paru-paru terindikasi *Tuberculosis*. Sedangkan pada Gambar 8, hasil test image menunjukkan bahwa kondisi paru-paru dalam mengalami *Viral Pneumonia*.



Gambar 7. Test Tuberculosis



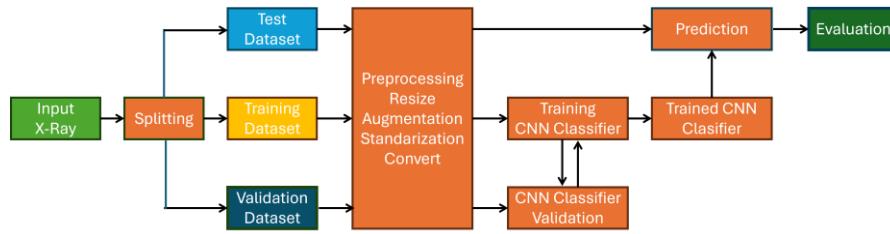
Gambar 8. Test viral pneumonia

Dari Gambar X-Ray yang di inputkan, terlihat bahwa model mampu mengenali masing-masing kelas dengan baik sekali, dan memperoleh nilai *confidence* diatas 90%.

### III. METODE PENELITIAN

#### a. Architectur Workflow

Pada Gambar 9 di bawah ini menunjukkan peta dari keseluruhan metode penelitian yang telah disampaikan sebelumnya, mulai dari persiapan data, proses belajar, hingga pengujian akhir untuk mengevaluasi model yang telah dibuat.



Gambar 9. Architectur workflow

Penjelasan Alur Kerja (Workflow) yang di representasikan pada Gambar 9 adalah :

1. Input X-Ray (Data Mentah), merupakan titik awal dimana ada sekumpulan besar gambar rontgen yang masih mentah dan belum mengalami proses apapun.
2. *Splitting* (Pembagian Data), proses ini membagi kumpulan data mentah menjadi tiga bagian terpisah, yaitu : *Training, Validation, & Test Dataset*.
3. *Preprocessing* (Persiapan Data), merupakan proses untuk menggunakan "kacamata standar" pada setiap gambar sebelum ditunjukkan ke model AI.
  - *Resize*, untuk menyamakan ukuran semua gambar.
  - *Augmentation*, untuk menambahkan data latih dari satu sumber gambar (misalnya, membalik gambar).
  - *Standardization*, untuk menyamakan tingkat kecerahan atau kontras.
  - *Convert*, untuk mengubah gambar menjadi format Tensor yang dimengerti oleh AI.
4. *Training & Validation Loop*. Adalah jantung dari proses belajar.
5. *Trained CNN Classifier*, merupakan hasil akhir dari proses belajar, yaitu "otak" model kita dalam kondisi terpintar dan sudah siap bekerja.
6. *Prediction* (Proses Prediksi), merupakan proses menguji model yang sudah kita latih menggunakan data baru. Di tahap ini proses ujian sesungguhnya berlangsung.
7. *Evaluation* (Evaluasi/Penilaian Akhir), merupakan hasil interpretasi nilai dari model. Pada tahap ini akan dihitung nilai akurasinya.

#### b. Formula Matematika

- Operasi Konvolusi, merupakan operasi dasar dalam CNN yang digunakan untuk mengekstraksi fitur spasial dari gambar input. Operasinya dirumuskan sebagai :

$$y(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(i + m, j + n) \cdot k(m, n) \dots \dots \dots (1)$$

Keterangan :

$x(i + m, j + n)$ : Input Citra

$k(m, n)$  : Kernel Filter

$y(i, j)$  : Output fitur map

- Fungsi Loss-Cross Entropy, digunakan untuk mengukur perbedaan antara distribusi prediksi dan label actual :

$$L = - \sum_{i=1}^K y_i \log(\hat{y}_i) \dots \dots \dots (2)$$

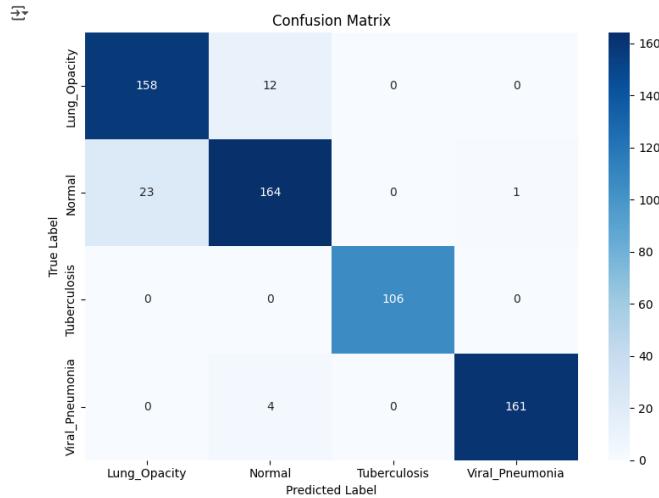
Keterangan :

$y_i$  : label asli dalam bentuk *one hot*

$\hat{y}_i$  : Probabilitas prediksi dari model

#### IV. HASIL PENELITIAN

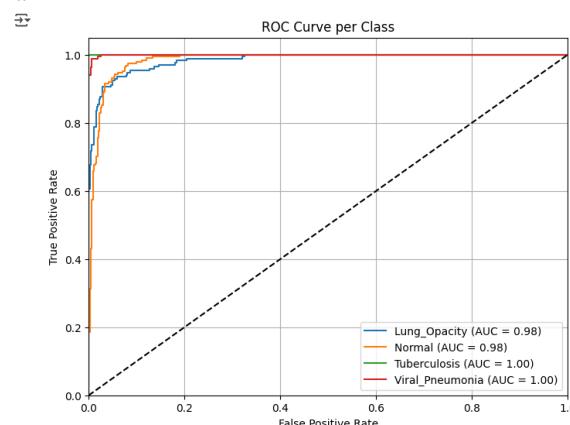
1. Uji Akurasi, menggunakan dataset test yang belum digunakan baik pada proses pelatihan maupun validasi, dengan hasil 93,48% (cukup baik), artinya dari 629 gambar (Tuberculosis, *Lung\_Opacity*, Normal, *Viral\_Pneumonia*) yang terdeteksi sesuai kelasnya yaitu sebanyak 588 gambar.
2. Evaluasi dengan Confusion Matrik terlihat sebagaimana Gambar 10. di bawah ini :



Gambar 10. *Confusion Matrix*

Penjelasan dari evaluasi *Confusion Matrix* pada Gambar 10. di atas adalah :

- a. Prediksi *Lung\_Opacity*. Terdeteksi Actual sebagai *Lung\_Opacity* sebanyak 158 (True Positif), sebagai Normal sebanyak 23 (salah prediksi), sebagai Tuberculosis dan *Viral\_Pneumonia* tidak ada (0).
  - b. Prediksi Normal. Terdeteksi Actual sebagai Normal sebanyak 164 (True Positif), sebagai *Lung\_Opacity* sebanyak 12 (salah prediksi), sebagai *Viral\_Pneumonia* 4 (salah prediksi), sebagai Tuberculosis tidak ada (0).
  - c. Prediksi Tuberculosis. Terdeteksi Actual sebagai Tuberculosis sebanyak 106 (True Positif), sebagai *Viral\_Pneumonia*, Normal dan *Lung\_Opacity* tidak ada (0) artinya model tidak pernah salah memprediksi gambar X-Ray Tuberculosis.
  - d. Prediksi *Viral\_Pneumonia*. Terdeteksi actual sebagai *Viral\_Pneumonia* sebanyak 161 (True Positif), sebagai normal sebanyak 1, sebagai Tuberculosis dan *Lung\_Opacity* tidak ada (0).
3. Gambar 11 di bawah ini memperlihatkan hasil evaluasi menggunakan ROC (*Receiver Operating Characteristic*) Curve dan nilai AUC (*Area Under Curve*), memiliki nilai yang mendekati 1, artinya model memiliki kemampuan diskriminasi sangat tinggi, bahkan untuk kelas Tuberculosis dan *Viral\_pneumonia* model memberi skor 1 (sempurna) hal ini karena AUC menggunakan skor kepercayaan bukan prediksi akhir, sedangkan confusion matrix sangat sensitive meski kesalahan 1 tetap dicatat sebagai kesalahan



Gambar 11. *ROC Curve*

## V. KESIMPULAN

Berdasarkan hasil penelitian ini, implementasi arsitektur *Convolutional Neural Network* (CNN) menggunakan DenseNet121 terbukti mampu melakukan klasifikasi citra X-ray dada ke dalam empat kategori, yaitu Normal, *Lung Opacity*, *Tuberculosis*, dan *Viral Pneumonia*, dengan tingkat akurasi yang sangat tinggi, yakni hampir mencapai 94%. Proses pelatihan model dilakukan dengan membagi dataset menjadi data *training*, validasi, dan pengujian, serta dilakukan *preprocessing* yang sesuai agar gambar dapat diolah secara optimal oleh model. Evaluasi performa model melalui akurasi dan *confusion matrix* berada pada kisaran nilai yang sangat baik untuk semua kelas. Selain itu, analisis lebih lanjut menggunakan *Receiver Operating Characteristic (ROC) Curve* memperlihatkan *Area Under Curve* (AUC) mendekati 1 di setiap kelas, yang menunjukkan bahwa model memiliki kemampuan klasifikasi yang sangat akurat. Hasil tersebut menunjukkan bahwa transfer learning menggunakan DenseNet121, meskipun awalnya dilatih dengan dataset ImageNet, tetap efektif untuk diterapkan dalam domain medis seperti deteksi penyakit paru-paru. Dengan demikian, penelitian ini membuktikan bahwa pendekatan CNN berbasis DenseNet121 dapat dijadikan solusi yang andal sebagai alat bantu diagnosis radiologis secara otomatis untuk mendeteksi penyakit paru berbasis citra *X-ray*.

## VI. DAFTAR PUSTAKA

- [1] Rajvardhan Nalawade et al. (2025). *Alzheimer's disease detection from brain MRI: Comparative Study of ResNet50, VGG19, DenseNet121 Multibranch CNN Model*, Procedia Computer Science.
- [2] Priyanka R et all (2025). *PediaPulmoDX: Harnessing cutting edge preprocessing and explainable AI for pediatric chest X-ray Classification with DenseNet121*, Result in Engineering 104320.
- [3] Xiaurui Zhang et all (2021). *Vehicle Re-Identification Model Based on Optimized DenseNet121 with Joint Loss*, Computers, Materials, and Continua Vol 67, Issue 3 pages 3933-3948
- [4] Jacinta Potsangbam et all (2024). *Classification of Breast Cancer Histopathological Image Using Transfer Learning with DenseNet121*, Procedia Computer Science 235, page 1990-1997
- [5] G Krishna Lava Kumar et all (2024). *An efficient diagnosis of heart disease using optimized cross-layer DenseNet121 pyramid mutual attention network*, Computer and Electrical Engineering Vol 119, Part B,
- [6] Ogundokun et all (2023). *MobileNet-SVM Lightweight Deep Transfer Learning Model to Diagnose BCH Scans for IoMT-Based Imaging Sensor*”, Sensor 23 (2), 656
- [7] Morovati et all (2023). *Reduced Deep Convolutional Activation Features (R-DeCAF) in Histopathology Images to Improve the Classification Performance for Breast Cancer Diagnosis*, arXivpreprint arXiv:2301.01931.
- [8] U. Chutia, A.S. Tewari, J.P. Singh, V.K. Raj (2024), *Classification of lung diseases using an attention-based modified DenseNet model*, J. Imag. Inf. Med. 37 (4) 1625–1641
- [9] K.V. Priya, J.D. Peter (2021). *A federated approach for detecting the chest diseases using DenseNet for multi-label classification*, Complex Intell. Syst. 8 (4) 3121–3129.
- [10] N. Hasan, Y. Bao, A. Shawon, Y. Huang (2021). *DenseNet convolutional neural networks application for predicting COVID-19 using CT image*, SN Comput. Sci. 2 (5).
- [11] M. Kholiavchenko, et al (2020). *Contour-aware multi-label chest X-ray organ segmentation*, Int. J. Comput. Assisted Radiol. Surg. 15 (3) 425–436
- [12] M.W. Kusk, S. Lysdahlgaard (2022). *The effect of Gaussian noise on pneumonia detection on chest radiographs, using convolutional neural networks*, Radiography 29 (1) 38–43
- [13] A. Gielczyk, A. Marciniak, M. Tarczewska, Z. Lutowski (2022), *Pre-processing methods in chest X-ray image classification*, PLoS ONE 17 (4) e0265949,
- [14] C. Ortiz-Toro, A. García-Pedrero, M. Lillo-Saavedra, C. Gonzalo-Martín (2022) *Automatic detection of pneumonia in chest X-ray images using textural features*, Comput. Biol. Med. 145 105466
- [15] I.F. Jassam, S.M. Elkaffas, A.A. El-Zoghabi (2021), *Chest X-ray pneumonia detection by Dense-Net*, <https://doi.org/10.1109/iccta54562.2021.9916637>
- [16] N. Ullah, et al., (2024). *ChestCovidNet: an effective DL-based approach for COVID-19, lung opacity, and pneumonia detection using chest radiographs images*, Biochem. Cell Biol. <https://doi.org/10.1139/bcb-2023-0265>.
- [17] V. Parthasarathy, S. Saravanan. (2024) *Computer aided diagnosis using Harris Hawks optimizer with deep learning for pneumonia detection on chest X-ray images*, Int. J. Inf. Technol. 16 (3) 1677–1683, <https://doi.org/10.1007/s41870-023-01700-1>.
- [18] Swathi S Kundaram and Ketki C Pathak.(2021) *Deep learning-based Alzheimer disease detection*. In: Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems: MCCS 2019. Springer. 2021, pp. 587–597