



Aplikasi Medis Pembaca Gambar Rontgen Berbasis AI

Tugas Akhir

**Oleh:
Galang Samudra (4212001020)**

**Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2024**

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : "Sebutkan Judul Tugas Akhir Anda" adalah **hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.** Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 15 Mei 2024



Galang Samudra

NIM: 4212001020


Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr.T)
di
Politeknik Negeri Batam

Oleh:
Galang Samudra (4212001020)

Tanggal Sidang: 8 Mei, 2024


Disetujui oleh:



1. Adlian Jefiza, S.Pd., M.T.
NIK: 119220



1. Dr. Budi Sugandi S.T., M.Eng
NIK: 0021037307



2. Widya Rika Puspita, S.Pd., M.Si.,
Ph. D
NIK: 119211

Aplikasi Medis Pembaca Gambar Rontgen Berbasis AI

Abstrak

Paru-paru merupakan organ pernapasan yang sangat penting bagi tubuh manusia. Apabila tidak diperhatikan dengan baik, maka dapat menyebabkan timbulnya penyakit pada paru-paru. Seperti penyakit pneumonia yang dimana satu anak meninggal akibat pneumonia setiap 39 detik. Indonesia menempati peringkat ketiga setelah India dan Cina dalam kasus TBC paru. Pada pandemi covid-19 terdapat 6.812.127 kasus terkonfirmasi di Indonesia. Kelainan pada paru dapat dilihat melalui X-ray. X-ray merupakan tindakan medis yang menggunakan gelombang radiasi sinar X dengan tipe warna *grayscale*. Ketika dokter spesialis menganalisis gambar x-ray dalam jumlah yang banyak, hal tersebut tentu melelahkan dan membutuhkan fokus yang tinggi. *Convolutional Neural Network* (CNN) saat ini banyak digunakan untuk mengklasifikasi, baik itu gambar ataupun video. Banyak penelitian yang menggunakan metode CNN untuk membaca dan mengidentifikasi penyakit dengan menggunakan gambar rontgen. Serta dengan penggunaan arsitektur turunan CNN yaitu *DenseNet121* yang digunakan untuk melatih kembali model keluaran CNN, yang bertujuan untuk meningkatkan keefektifan model dalam mengklasifikasi gambar rontgen. Dan penelitian ini bertujuan untuk membuat *User Interface* yang menggunakan implementasi dari hasil klasifikasi CNN. Dan dapat membaca gambar x-ray tidak hanya satu tetapi lebih dari dua dalam satu waktu dan proses pengidentifikasian yang cepat dan mudah dalam pengoperasiannya. Dan juga dapat dilihat hasil akurasi pada pembacaan gambar x-ray sebagai hasil akhir dari proses penelitian ini didapatkan hasil akurasi sebesar 92%. Model terintegrasi dengan *Graphical User Interface* (GUI) dapat melakukan klasifikasi terhadap 4 kondisi pada gambar x-ray.

Kata kunci: Paru-paru, Gambar X-ray, *Convolutional Neural Network*, *DenseNet121*.

Medical Application to Read X-ray Images Based on AI

Abstract

The lungs are vital respiratory organs for the human body. If not properly observed, it can lead to disease in the lungs. It's like pneumonia, where one child dies from pneumonia every 39 seconds. Indonesia ranks third after India and China in cases of pulmonary tuberculosis. In the COVID-19 pandemic, there were 6,812,127 confirmed cases in Indonesia. The abnormalities in the lungs can be seen through the X-ray . X-ray s are medical measures that use the waves of x-ray radiation with a grayscale color type. When a doctor analyzes a large number of x-ray s, it's exhausting and requires a high level of focus. Convolutional Neural Network (CNN) is widely used today to classify, be it images or videos. Many researchers have used CNN methods to read and identify diseases using X-ray s, as well as using CNN's derivative architecture, DenseNet121, to re-train the CNN output model, which aims to improve the effectiveness of models in classifying X-ray images. And this research aims to create a Graphic User Interface (GUI) that uses the implementation of the CNN classification results. And can read x-ray images not only one but more than two at a time, and the identification process is fast and easy in its operation. And also can be seen the accuracy results on the reading of x-ray s, as the final result of this research process obtained a result of 92% accuracy. The model integrated with the Graphical User Interfaces (GUI) can perform classification against four conditions on x-ray images.

Keywords: Lungs, X-ray Imaging, Convolutional Neural Network, DenseNet121.

Kata Pengantar

Puji syukur penulis ucapkan atas kehadiran Allah SWT karena atas Rahmat, limpahan dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul “Aplikasi Medis Pembaca Gambar Rontgen Berbasis AI”. Tugas Akhir ini dibuat dan disusun untuk memenuhi persyaratan kelulusan akademis di Jurusan Teknik Elektro, Program Studi D4 Teknik Mekatronika, Politeknik Negeri Batam.

Dalam proses perencanaan, penyusunan, dan menyelesaikan Laporan Tugas Akhir ini, Penulis tidak terlepas dari dukungan, bimbingan dan bantuan serta motivasi dari berbagai pihak yang ada. Oleh karena itu penulis bermaksud mengucapkan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas Rahmat dan Izin-Nya yang telah diberikan kepada penulis.
2. Orang tua dan keluarga yang telah memberikan dukungan moral serta materil yang tak ternilai harganya.
3. Bapak Uuf Brajawidagda, S.T., M.T., Ph. D selaku direktur Politeknik Negeri Batam.
4. Bapak Dr. Budi Sugandi, S.T., M.Eng selaku Kepala Jurusan Teknik Elektro dan selaku pembimbing penulis yang telah membimbing serta mengarahkan penulis proses pengerjaan Tugas Akhir ini dari awal hingga selesai.
5. Bapak Indra Hardian Mulyadi, S.T., M.Eng., Ph.D selaku Kepala Program Studi D-IV Teknik Mekatronika Politeknik Negeri Batam.
6. Bapak Sumantri Kurniawan Risandriya, S.T., M.T. selaku wali dosen penulis
7. Bapak Muhammad Naufal Airlangga Diputra, S.Pd., M.P.H. selaku dosen pengampu mata kuliah tugas akhir
8. Bapak Adlian Jefiza, S.Pd., M.T. dan Ibu Widya Rika Puspita, S.Pd., M.Si. selaku dosen penguji penulis
9. Seluruh Dosen , Staff, dan laboran di Jurusan Teknik Elektro Politeknik Negeri Batam
10. Seluruh rekan-rekan Teknik Mekatronika Angkatan 2020 yang telah membantu dan mendukung penulis
11. Stevany selaku rekan tim proyek MFAI 2023 yang telah banyak membantu dan memberikan masukan dalam proses pembuatan Tugas Akhir ini

Atas bantuan dan kerjasamanya sehingga dapat terselesaikannya buku ini. Penulis menyadari bahwa dalam pembuatan dan penyusunan Tugas Akhir ini masih terdapat banyak kekurangan.Oleh karena itu penulis mengharapkan saran dan kritik yang dapat membangun untuk penulisan karya ilmiah yang akan datang.

Demikian, penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca dan semua pihak yang terkait.

Batam, 1 April 2024

Penulis

Daftar Isi

| | |
|---|-----|
| Pernyataan Keaslian Tugas Akhir | i |
| Lembar Pengesahan | ii |
| Abstrak | iii |
| <i>Abstract</i> | iv |
| Kata Pengantar | v |
| Daftar Isi | vii |
| Daftar Gambar | ix |
| Daftar Tabel | x |
| Bab 1. Pendahuluan | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 3 |
| 1.3. Tujuan | 3 |
| 1.4. Manfaat | 3 |
| 1.5. Batasan | 3 |
| Bab 2. Tinjauan Pustaka | 4 |
| 2.1 Gambar X-ray dan Jenis Kelainan Pada Paru-paru | 4 |
| 2.2 Pengambilan Dataset | 6 |
| 2.3 <i>Deep Learning</i> | 6 |
| 2.3.1 <i>Convolutional Neural Network (CNN)</i> | 6 |
| 2.3.2 <i>DenseNet121</i> | 7 |
| Bab 3. Metodologi Penelitian | 9 |
| 3.1. Tahapan Penelitian | 9 |
| 3.2. Perancangan Perangkat Lunak | 10 |
| 3.2.1 <i>Preprocessing Data</i> | 10 |
| 3.2.2 Training Data | 11 |
| 3.3 Perancangan <i>Graphical User Interface (GUI)</i> | 12 |
| 3.4 Pengujian Sistem | 13 |
| 3.4.1 Confusion Matrix | 13 |
| Bab 4. Hasil dan Pembahasan | 15 |

| | |
|--|----|
| 4.1 Hasil <i>Preprocessing Data</i> | 15 |
| 4.2 <i>Summary</i> Arsitektur | 17 |
| 4.2.1 <i>Summary</i> Arsitektur CNN | 17 |
| 4.2.2 <i>Summary</i> Arsitektur <i>DenseNet121</i> | 18 |
| 4.3 Hasil Pelatihan Model | 19 |
| 4.3.1 Hasil Pelatihan Model CNN | 19 |
| 4.3.2 Hasil Pelatihan Model <i>DenseNet121</i> | 19 |
| 4.4 Hasil <i>Confusion Matrix</i> | 22 |
| 4.4.1 <i>Confusion Matrix</i> model CNN..... | 22 |
| 4.4.2 <i>Confusion Matrix</i> model <i>DenseNet121</i> | 23 |
| 4.5 <i>Graphical User Interface (GUI)</i> | 26 |
| Bab 5. Kesimpulan dan Saran..... | 30 |
| 5.1 Kesimpulan..... | 30 |
| 5.2 Saran | 30 |
| Daftar Pustaka..... | 31 |
| Lampiran | 33 |

Daftar Gambar

| | |
|--|----|
| Gambar 1. Kondisi Paru-Paru Normal | 4 |
| Gambar 2. Kondisi Paru-Paru Pneumonia..... | 4 |
| Gambar 3 Kondisi Paru-Paru Penderita Tuberculosis | 5 |
| Gambar 4 Kondisi Paru-Paru Penderita Covid-19 | 5 |
| Gambar 5 Convolutional Neural Network (CNN) | 7 |
| Gambar 6 Densely Connected Convolutional Network..... | 8 |
| Gambar 7. Tahapan Penelitian..... | 9 |
| Gambar 8 Alur <i>Preprocessing Data</i> | 10 |
| Gambar 9 Proses pelatihan model..... | 11 |
| Gambar 10. Sistem Perancangan Aplikasi..... | 12 |
| Gambar 11 Gambar sampel | 15 |
| Gambar 12 Gambar sampel setelah di- <i>preprocessing</i> | 16 |
| Gambar 13 Arsitektur CNN | 17 |
| Gambar 14 Detail Arsitektur DenseNet121 | 18 |
| Gambar 15 Hasil Pelatihan Model CNN | 19 |
| Gambar 16 Hasil Pelatihan Model <i>DenseNet121</i> | 19 |
| Gambar 17 Train & Validation Loss, Train & Validation Accuracy dari Model CNN | 20 |
| Gambar 18 Train & Validation Loss, Train & Validation Accuracy dari Model <i>DenseNet121</i> | 21 |
| Gambar 19 <i>Confusion Matrix</i> CNN | 22 |
| Gambar 20 <i>Confusion Matrix DenseNet121</i> | 23 |
| Gambar 21 Tampilan GUI..... | 26 |
| Gambar 22 Proses Klasifikasi Gambar..... | 27 |
| Gambar 23 Gambar Tersimpan dalam format PDF..... | 28 |

Daftar Tabel

Tabel 1 Detail dataset 6
Tabel 2 *Confusion Matrix* 13
Tabel 3 *Classification Report* Model CNN 24
Tabel 4 *Classification Report* Model *DenseNet121* 24

Bab 1. Pendahuluan

1.1. Latar Belakang

Paru-paru merupakan organ pernapasan yang sangat penting bagi tubuh manusia. Paru-paru memiliki peran sebagai organ yang bekerja dengan melakukan pertukaran karbon dioksida dari dalam tubuh dengan oksigen dari lingkungan luar. Walaupun paru-paru sangat vital bagi tubuh, namun faktanya masih banyak masyarakat yang tidak memperhatikan kesehatan paru-parunya yang dapat membuat organ ini menjadi terganggu. Apabila fungsi pernapasan pada manusia terganggu, maka dapat dipastikan akan memengaruhi manusia tersebut secara langsung. Diantara gangguan pernapasan yang terjadi, apabila tidak ditangani secara langsung dapat mengakibatkan timbulnya penyakit pada paru-paru.

Penyakit yang terjadi pada paru-paru umumnya disebabkan oleh berbagai mikroorganisme seperti bakteri, virus, jamur. Selain dari mikroorganisme, penyakit pada paru-paru juga dapat disebabkan dari kebiasaan merokok, terpapar polusi udara atau asap kendaraan berat. Yang dimana hal ini dapat menyebabkan berbagai kelainan kondisi pada paru-paru yang jika dibiarkan akan berakibat munculnya penyakit seperti Pneumonia, Tuberculosis (TBC), Bronkitis, kanker paru-paru.

Lembaga Kesehatan dan Anak memperingatkan satu anak meninggal akibat pneumonia setiap 39 detik. Sebagian besar kematian terjadi pada anak dibawah usia 2 tahun. Pneumonia menjadi salah satu penyebab kematian Balita terbesar di Indonesia. Pada tahun 2018, diperkirakan sekitar 19.000 anak meninggal dunia akibat pneumonia. Risiko terjangkit pneumonia jauh lebih besar dialami anak-anak yang sistem imun tubuhnya lemah akibat penyakit lain seperti HIV atau malnutrisi [1].

Tuberculosis (TBC) adalah penyakit yang disebabkan oleh patogen bakteri *Mycobacterium tuberculosis* (*M. tuberculosis*) dan termasuk penyakit menular. TBC pada paru dapat menyebabkan kematian apabila tidak mengonsumsi obat dengan teratur pada rentang 6 bulan. TBC masih menjadi masalah kesehatan global dan WHO pada tahun 2017 melaporkan terdapat 1,3 juta kematian akibat TBC paru. Indonesia merupakan peringkat ketiga setelah India dan Cina dalam kasus TBC paru, yakni dengan jumlah 824 ribu kasus dan 93 ribu kematian pertahun [2]. Pada tahun 2022 Kementerian Kesehatan bersama seluruh tenaga kesehatan berhasil mendeteksi TBC sebanyak lebih dari 700 ribu kasus. Angka tersebut merupakan angka tertinggi sejak TBC menjadi program prioritas nasional [3].

Penyakit gangguan pernapasan akut atau lebih dikenal dengan nama *corona virus disease 2019* (COVID-19) pertama kali dilaporkan muncul di kota Wuhan, ibukota provinsi Hubei, Tiongkok pada akhir tahun 2019. Penyakit ini menjadi perhatian dunia karena tingkat penularannya yang sangat tinggi dan

menyebabkan kasus kematian pada beberapa penderita di kota Wuhan sehingga pemerintah setempat memberlakukan karantina wilayah kota Wuhan secara penuh (*lockdown*) [4]. Dan informasi terbaru dari situs Kementerian Kesehatan Republik Indonesia (INFEKSIEMERGING) mengenai perkembangan kasus masyarakat yang terinfeksi virus covid-19 pertanggal 05 Juli 2023 adalah 6.812.127 kasus terkonfirmasi, 6.642.003 kasus sembuh (97,5%), 161.879 kasus meninggal (2,4%), dan 8.245 kasus aktif (0,1%) [5].

Rontgen merupakan tindakan medis yang menggunakan radiasi gelombang elektromagnetik untuk mengambil gambar bagian atau organ dalam dari tubuh seseorang. Hasil dari foto rontgen ini sendiri nantinya akan digunakan oleh dokter spesialis radiologi dalam rangka mendeteksi, menentukan diagnosis, dan mengobati suatu penyakit menggunakan prosedur pencitraan. Seperti patah tulang, tumor, masalah jantung, dan penyakit pada paru-paru [6].

Dalam melakukan *Rontgen*, menghasilkan gambar dari proses penyinaran sinar *X-ray* ke anggota tubuh yang akan diperiksa. Dalam pembacaan gambar *X-ray*, menghasilkan keluaran berupa gambar *X-ray* dengan jenis citra *grayscale*.

Salah satu metode untuk mengenali dan mendeteksi objek pada sebuah gambar yaitu dengan menggunakan algoritma CNN (*Convolutional Neural Networks*) [7]. CNN digunakan untuk mengenali benda, melakukan deteksi serta segmentasi pada objek. CNN bekerja dengan melakukan klasifikasi yang digunakan untuk pengolahan data gambar. Algoritma CNN ini dapat digunakan pada proses klasifikasi data yang sudah terlabel dengan menggunakan metode *supervised learning*. Cara kerja dari *supervised learning* yaitu jika sudah terdapat *dataset* yang akan dilatih dan sudah terdapat variabel yang akan ditargetkan sebagai keluaran, sehingga tujuan dari metode ini adalah dengan mengelompokkan data ke data yang telah ada [8].

Pada penelitian Novanto Yudistira dkk (2020) [9], membuat sistem pendeteksian COVID-19 menggunakan metode *Convolutional Neural Network*. Yang menggunakan 1125 gambar x-ray dan menghasilkan akurasi sebesar 86.93%

Berdasarkan penelitian sebelumnya menggunakan metode *command type* ketika akan melakukan pembacaan gambar x-ray. Dan harapan untuk penelitian ini adalah, dengan adanya pembuatan *User Interface* (UI) yang nantinya akan diimplementasikan dalam bentuk aplikasi yang dapat digunakan. Sehingga dapat lebih efisien dalam membantu pengoperasian untuk melakukan pembacaan banyak gambar x-ray dalam satu waktu.

1.2. Rumusan Masalah

Dari latar belakang masalah tersebut, adapun rumusan masalah yang akan dibahas pada penelitian tugas akhir ini adalah:

1. Bagaimana cara mendeteksi kelainan pada paru-paru pada gambar X-ray ?
2. Bagaimana cara untuk mengklasifikasi jenis kelainan pada paru-paru gambar X-ray ?.

1.3. Tujuan

Adapun tujuan dari penulisan proyek akhir ini yaitu:

1. Melatih model klasifikasi CNN menggunakan arsitektur *DenseNet121* pada gambar x-ray .
2. Model dapat mendeteksi dan memprediksi penyakit pada paru-paru yang ada pada gambar x-ray
3. Merancang antarmuka untuk memudahkan pengguna dalam melakukan klasifikasi gambar x-ray .

1.4. Manfaat

1. Manfaat dari perancangan sistem deteksi kelainan paru-paru gambar x-ray menggunakan kecerdasan buatan ini adalah untuk membantu dokter spesialis radiologi dalam mendiagnosis penyakit yang diidap pasien.
2. Meningkatkan efisiensi waktu dalam menentukan diagnosis penyakit pneumonia.

1.5. Batasan

Adapun batasan masalah dari pembuatan proyek akhir ini diantaranya:

1. Pembacaan pada gambar x-ray hanya mendeteksi kelainan di paru-paru
2. Kondisi yang akan dideteksi oleh model yaitu Covid-19, Normal, *pneumonia*, *tuberculosis* (TBC) pada paru-paru
3. Data gambar yang digunakan terbatas dan bukan format gambar x-ray aslinya (Dicom).

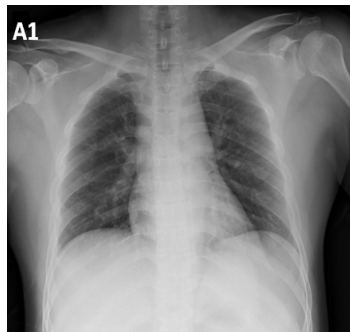
Bab 2. Tinjauan Pustaka

2.1 Gambar X-ray dan Jenis Kelainan Pada Paru-paru

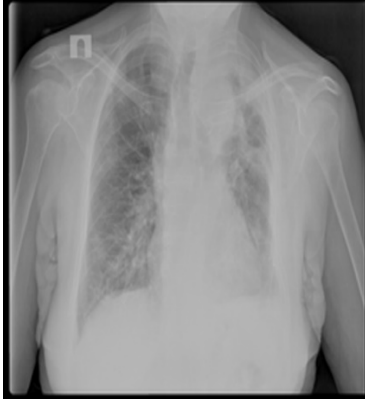
Gambar *rontgen* atau *x-ray* merupakan hasil dari proses mengambil foto organ dalam tubuh manusia dengan menggunakan radiasi dari sinar X. Dan hasil dari proses rontgen akan dicetak dalam bentuk lembaran fail gambar dengan tipe citra *grayscale*. Pada bagian dada (*thorax*) di gambar *x-ray*, dokter spesialis melihat dan menganalisis organ-organ yang tampak pada gambar [10]. Kondisi paru-paru normal dan yang mengalami kelainan dapat langsung terlihat. Seperti lebih banyak bercak warna putih yang ada pada paru-paru bagian bawah. Hal tersebut dapat mengindikasikan bahwa pasien mengidap penyakit pada paru-parunya yang dapat dilihat pada Gambar 2.



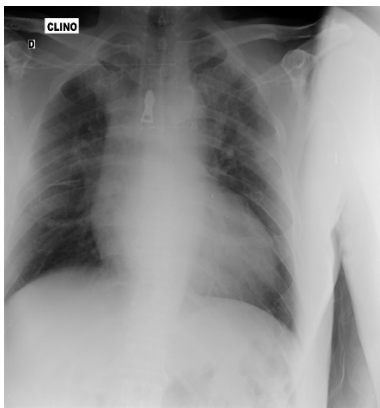
Gambar 1. Kondisi Paru-Paru Normal



Gambar 2. Kondisi Paru-Paru Pneumonia



Gambar 3 Kondisi Paru-Paru Penderita Tuberculosis



Gambar 4 Kondisi Paru-Paru Penderita Covid-19

2.2 Pengambilan Dataset

Pada penelitian ini, dataset dari gambar x-ray rontgen didapat dari beberapa sumber yang berbeda. Seperti pada situs *Kaggle* yang merupakan situs untuk berbagi ide, dan juga merupakan platform *data sciene* [11]. Kemudian *dataset* yang telah diperoleh ini adalah kumpulan dari gambar x-ray dengan kondisi normal dan kondisi mengidap Covid-19, *pneumonia*, *tuberculosis*, yang akan menjadi bahan untuk pelatihan bagi model kecerdasan buatan. Jumlah sampel gambar x-ray dengan diagnosis penyakit yang akan digunakan untuk menjadi bahan latih kecerdasan buatan akan bergantung dari ketersediaan database yang ada di internet. Detail data gambar x-ray yang terkumpul pada penelitian ini dapat dilihat pada tabel 1.

Tabel 1 Detail dataset

| Kelas kondisi | Jumlah data | Data Latih (90%) | Data Test (10%) |
|---------------|---------------|------------------|-----------------|
| Normal | 11.995 | 10.795 | 1.200 |
| Covid-19 | 5.243 | 4.718 | 525 |
| Pneumonia | 3.146 | 2.831 | 315 |
| Tuberculosis | 1.221 | 1.099 | 122 |
| Total | 21.606 | 19.444 | 2.162 |

Pada tabel data di atas *dataset* dibagi menjadi 2 kelompok, yaitu data latih sebagai data yang digunakan untuk proses pelatihan yang diambil sebanyak 90% dari jumlah total *dataset* dan data test sebagai data untuk memvalidasi hasil prediksi model yang diambil sebanyak 10% dari total *dataset* yang ada.

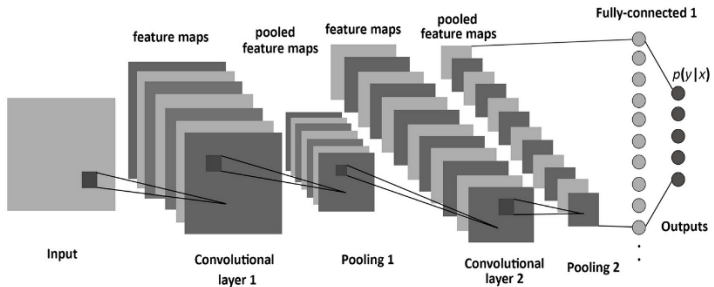
2.3 Deep Learning

Deep learning adalah bagian dari kecerdasan buatan (*artificial intelligence*) dan pembelajaran mesin (*machine learning*) yang menggunakan lapisan saraf (*neuron layer*) dari jaringan saraf tiruan melakukan tugasnya. *Deep learning* terdiri dari seperangkat algoritma *neural network* dan *machine learning* yang berusaha belajar dalam berbagai tingkatan berdasarkan input informasi dari setiap neuron yang ada pada algoritma dan *machine learning*. Yang dimana hasil pembelajaran tersebut akan selalu dilakukan selama neuron mendapatkan input baru dari lapisan neuron sebelumnya hingga mencapai lapisan output [12].

2.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan hasil pengembangan dari jaringan saraf tiruan (*neural network*) *Multilayer Perceptron* (MLP) yang digunakan untuk mengolah data dua dimensi. CNN banyak diaplikasikan pada data citra karena kemampuan jaringan yang tinggi. Algoritma CNN ini termasuk kedalam

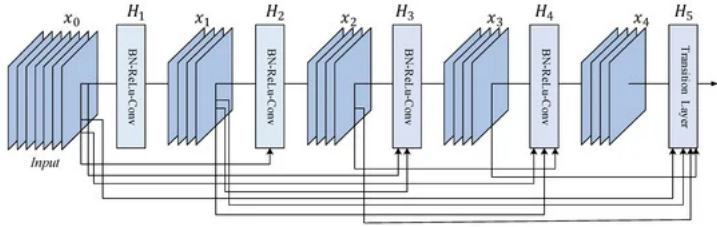
jenis *Deep Neural Network* yang konvensionalnya digunakan pada proses perancangan *deep learning*. Dan karena kemampuannya, CNN banyak digunakan dalam mengolah data citra yang kemudian salah satu penggunaannya adalah untuk melakukan klasifikasi gambar (*image classification*), pengenalan objek (*object detection*) [13]. CNN menggunakan beberapa lapisan (*layer*) yaitu *convolutional layer*, *pooling layer*, dan *fully connected layer*. *Convolutional layer* berfungsi untuk mengekstraksi fitur dari input dengan filter yang dasarnya adalah bentuk estimasi. Hasil ekstraksi fitur itu akan menjadi pola gambar yang akan diidentifikasi dan menghasilkan *weights* atau bobot untuk menggambarkan pola gambar yang telah teridentifikasi. Kemudian diikuti dengan *pooling layer* yang akan mengurangi ukuran spasial dan dapat mengurangi sumber daya komputasi yang dibutuhkan untuk memproses data melalui pengurangan dimensi dari *feature map* (*down sampling*) yang dapat mempercepat komputasi karena parameter yang diperbarui semakin sedikit [14]. *Pooling layer* dilakukan dengan *max-pooling*, yaitu dengan mengambil nilai maksimal dari *region* sebelumnya seperti Gambar berikut.



Gambar 5 Convolutional Neural Network (CNN)

2.3.2 DenseNet121

DenseNet (*Densely Connected Convolutional Network*) merupakan model arsitektur turunan dari CNN yang melatih kembali hasil keluaran dari model CNN. *DenseNet121* menggunakan keluaran dari sebuah lapisan model untuk digunakan sebagai masukan pada lapisan neuron selanjutnya [15]. *DenseNet121* yang menghubungkan setiap lapisan neuron, seperti pada lapisan pertama dihubungkan ke lapisan kedua, ketiga dan seterusnya. Yang dimana hal ini bertujuan untuk memberdayakan pergerakan data terbesar antar lapisan neuron [16].

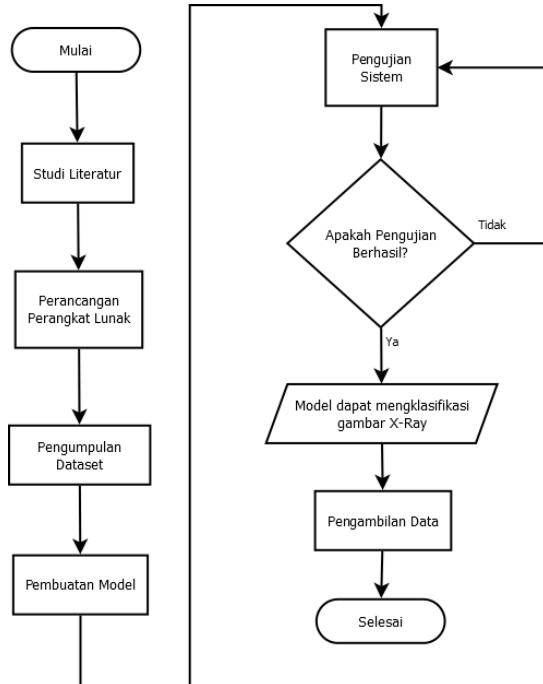


Gambar 6 Densely Connected Convolutional Network

Bab 3. Metodologi Penelitian

3.1. Tahapan Penelitian

Dalam merancang sistem aplikasi, dibutuhkan perancangan dalam proses pembuatannya. Proses perancangan aplikasi ini meliputi tahapan penelitian, perancangan perangkat lunak, pengujian sistem, dan analisis hasil pengujian. Rancang diagram alir tahapan penelitian dapat dilihat pada Gambar 6.



Gambar 7. Tahapan Penelitian

Perancangan penelitian dilakukan dengan melakukan studi literatur yang dilakukan dengan mengumpulkan berbagai macam informasi dan keilmuan mengenai penerapan *artificial intelligent* dalam bidang klasifikasi gambar medis. Dilanjutkan dengan mengumpulkan dataset yang akan diolah menjadi model (hasil dari proses pelatihan) yang digunakan sebagai *comparison* pembacaan, dilanjutkan dengan melakukan pengujian pada sistem yang telah dibuat dengan

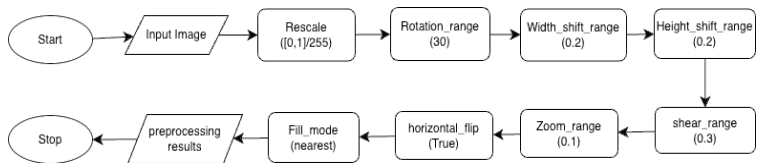
memasukkan model sebagai data pembacaan untuk perbandingan. Dilakukan verifikasi dan validasi pada model dikarenakan jika terdapat hasil pembacaan yang kurang akurat jika sudah akurat maka dilanjutkan dengan pengambilan data.

3.2. Perancangan Perangkat Lunak

Pada perancangan perangkat lunak terdiri dari tiga proses utama, yaitu *Preprocessing*, *training* dan *data testing* untuk mengetahui kelainan paru-paru pada gambar x-ray . Pada prosesnya, dataset yang berisi data latih berupa gambar x-ray paru-paru normal dan gambar x-ray paru-paru yang mengalami kelainan. Sebelum memasuki proses *training* dan *testing*, dataset akan di-*preprocessing* untuk mengubah dan menghilangkan data yang tidak diperlukan dan agar sesuai dengan format yang diperlukan pada proses *training* nantinya.

3.2.1 Preprocessing Data

Preprocessing Data merupakan proses untuk mengubah data mentah ke dalam bentuk yang lebih mudah dipahami. Yang bertujuan untuk memperbaiki kesalahan pada data mentah yang sering tidak lengkap serta format yang tidak teratur. Proses dari perancangan kecerdasan buatan yaitu melalui proses *preprocessing data* untuk mengubah dan memperbaiki apabila ada data yang tidak sesuai. Dan selanjutnya pada proses pelatihan dan validasi data. Akan dibagi menjadi 4 kelas sesuai dengan klasifikasi penyakit yang telah ditentukan.



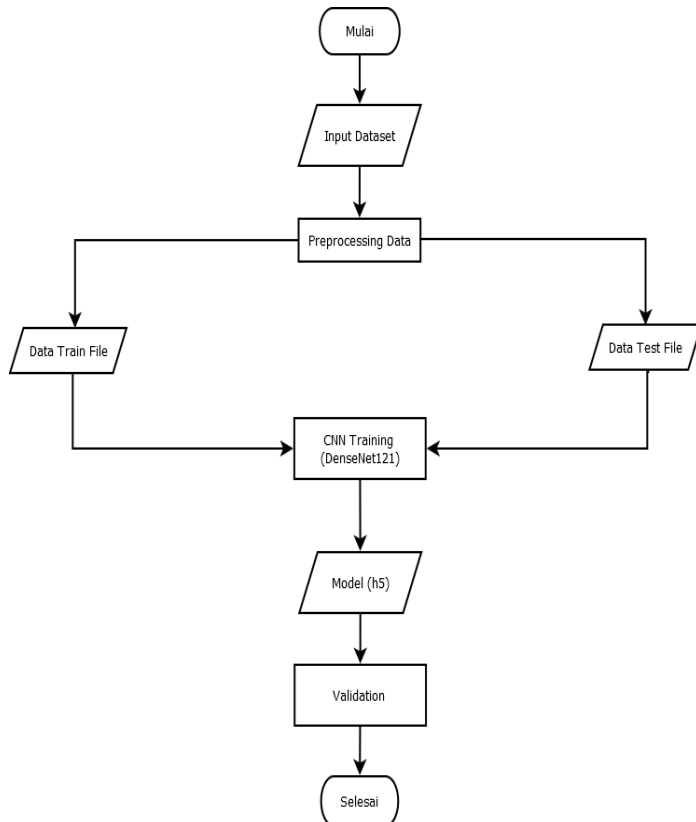
Gambar 8 Alur *Preprocessing Data*

Pada *processing data*, dataset gambar x-ray dimasukkan. Tahap pertama yang dilakukan adalah menormalisasi ukuran piksel gambar menjadi 0 dan 1 dengan membagi setiap nilai piksel dengan 255 yang bertujuan data agar lebih mudah dikelola oleh model. Selanjutnya gambar dirotasi secara acak yang telah ditentukan sebesar 30 derajat, yang berarti gambar dapat diputar sebanyak maksimal 30 derajat searah atau berlawanan jarum jam dan menggeser 0.2 atau 20% dari lebar dan tinggi gambar. Pada *shear_range* akan mentransformasi geser yang membuat bentuk gambar menjadi miring, hal ini menciptakan regangan oada gambar dengan sudut kemiringan sebesar 30 derajat. Kemudian pada *zoom_range* akan memungkinkan gambar diperbesar atau diperkecil sebesar 10% dan diputar secara mendatar (*horizontal*). Dan *fill_mode (nearest)* digunakan untuk mengisi piksel kosong terdekat yang muncul pada gambar setelah proses rotasi atau

transformasi. Hasil dari proses ini meningkatkan keberagaman data yang digunakan untuk pelatihan model, sehingga model dapat menggeneralisasi data yang belum pernah dilihat sebelumnya.

3.2.2 Training Data

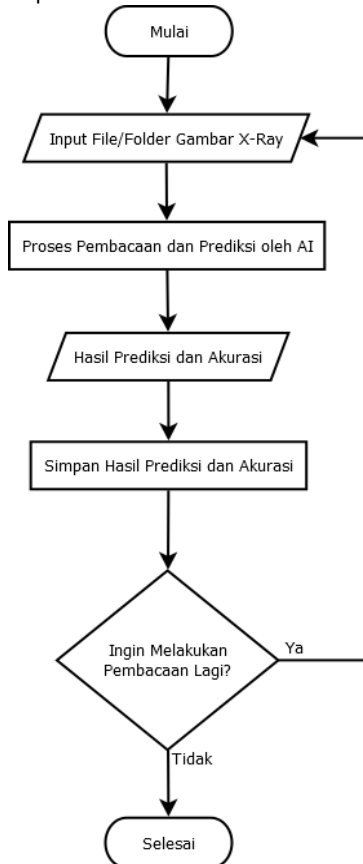
Setelah persiapan *dataset* dan *preprocessing*, proses pelatihan model dimulai. Pada tahap ini, arsitektur jaringan syaraf tiruan membuat dan melatih model kecerdasan buatan dengan menggunakan gambar dari dataset yang telah melalui tahap *preprocessing*, yang memungkinkan mereka untuk mengklasifikasi dan membuat prediksi pada gambar x-ray .



Gambar 9 Proses pelatihan model

3.3 Perancangan *Graphical User Interface* (GUI)

Graphical User Interface merupakan operasi sistem antar muka berbasis grafis yang menggunakan ikon, daftar menu, dan tetikus (mouse) untuk mengklik ikon atau menarik ke atas dan bawah daftar menu agar dapat berinteraksi dengan sistem. Dengan adanya GUI ini yang dirancang menjadi satu menjadi aplikasi padu, dapat mempermudah dalam mengoperasikan sistem untuk melakukan pembacaan gambar dan menganalisis hasilnya. Tidak perlu melalui perintah/kode program ketika masih berbentuk kumpulan kode perintah. Rancangan GUI pada penelitian ini dapat dilihat pada Gambar 6.



Gambar 10. Sistem Perancangan Aplikasi

3.4 Pengujian Sistem

3.4.1 Confusion Matrix

Confusion matrix digunakan untuk mengukur kinerja dari hasil model klasifikasi CNN dan menggambarkan jumlah yang diklasifikasikan dengan benar atau tidak. Terdapat empat nilai yang dihasilkan, yaitu:

- TP (*True Positive*) adalah jumlah data yang bernilai positif dan diprediksi benar sebagai Positif.
- FP (*False Positive*) adalah jumlah data yang bernilai Negatif, tetapi diprediksi Positif.
- FN (*False Negative*) adalah jumlah data yang Positif, tetapi diprediksi sebagai Negatif
- TN (*True Negative*) adalah jumlah data yang Negatif dan diprediksi sebagai Negatif

Tabel 2 *Confusion Matrix*

| | | Predicted Label | |
|------------|--------------|-------------------------------|-------------------------------|
| | | Positive (1) | Negative (0) |
| True Label | Positive (1) | TP (True Positive) | FN (False Negative) |
| | Negative (0) | FP (False Positive) | TN (True Negative) |

Setelah hasil dari *confusion matrix* didapatkan. Maka dilakukan evaluasi terhadap hasilnya. Dengan melakukan *classification report* yang menggunakan parameter sebagai berikut:

- *Accuracy* menghitung proporsi prediksi benar yang dihasilkan dari model klasifikasi dari total jumlah prediksi.

$$\frac{(TP + TN)}{TP + TN + FP + FN} \tag{1}$$

- *Precision* menghitung jumlah proporsi dari *True Positive* dari semua jumlah prediksi benar yang dihasilkan model klasifikasi

$$\frac{TP}{(TP + FP)} \quad (2)$$

- *Recall* menghitung jumlah proporsi *True Positive* dari total jumlah aktual positif pada dataset.

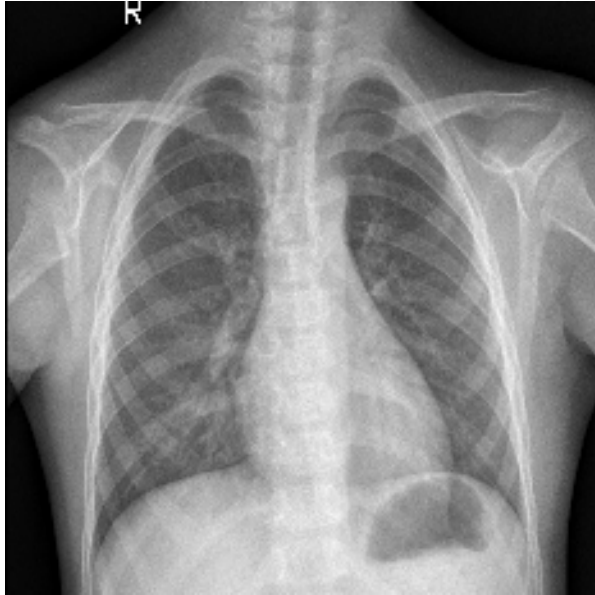
$$\frac{TP}{(TP + FN)} \quad (3)$$

- *F1-Score* adalah nilai rata-rata *precision* dan *recall*

$$F1 = 2 \cdot \frac{(precision \cdot recall)}{(precision + recall)} \quad (4)$$

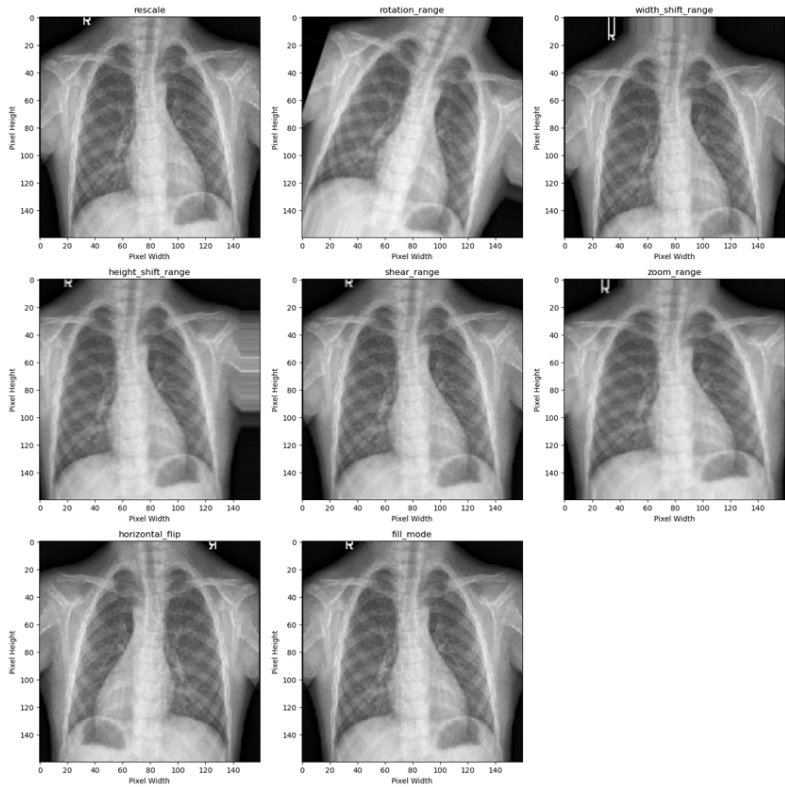
Bab 4. Hasil dan Pembahasan

4.1 Hasil *Preprocessing Data*



Gambar 11 Gambar sampel

Pada Gambar 11 merupakan salah satu sampel gambar x-ray dari *dataset* yang digunakan untuk pelatihan model. Merupakan gambar dari kelas NORMAL yang memiliki ukuran 232x232 piksel dan saluran warna RGB.



Gambar 12 Gambar sampel setelah di-*preprocessing*

Gambar 12 merupakan sampel gambar x-ray sebelumnya yang telah dilakukan proses *preprocessing*. Pada *rescale* ukuran gambar diubah dari yang awalnya berukuran 232x232 piksel menjadi 160x160 piksel. *Rotation_range* memutar gambar sebesar 30 derajat. *Width_shift_range* dan *height_shift_range* menggeser lebar dan tinggi gambar sebesar 20% dari ukuran asli. *Shear_rang* meregangkan gambar. *Zoom_range* memperbesar gambar sebesar 10% dari gambar asli. *Horizontal_flip* memutar gambar secara mendatar (horizontal). Dan *fill_mode* mengisi piksel yang kosong pada gambar setelah proses preprocessing data.

4.2 Summary Arsitektur

4.2.1 Summary Arsitektur CNN

```
Model: "sequential_2"
```

| Layer (type) | Output Shape | Param # |
|--------------------------------------|----------------------|---------|
| conv2d_6 (Conv2D) | (None, 158, 158, 16) | 448 |
| max_pooling2d_6 (MaxPooling2D) | (None, 79, 79, 16) | 0 |
| conv2d_7 (Conv2D) | (None, 77, 77, 32) | 4640 |
| max_pooling2d_7 (MaxPooling2D) | (None, 38, 38, 32) | 0 |
| conv2d_8 (Conv2D) | (None, 36, 36, 64) | 18496 |
| max_pooling2d_8 (MaxPooling2D) | (None, 18, 18, 64) | 0 |
| flatten_2 (Flatten) | (None, 20736) | 0 |
| dense_9 (Dense) | (None, 200) | 4147400 |
| dropout_4 (Dropout) | (None, 200) | 0 |
| ... | | |
| Total params: 4273488 (16.30 MB) | | |
| Trainable params: 4273488 (16.30 MB) | | |
| Non-trainable params: 0 (0.00 Byte) | | |

Gambar 13 Arsitektur CNN

Pada *model summary* di atas terdapat lapisan konvolusi dengan 16 filter, dan menggunakan fungsi aktivasi *Rectified Linear Units* (ReLU). Memiliki input gambar dengan ukuran 160x160 piksel dan 3 saluran warna RGB. Dan pada lapisan *max pooling* mengurangi ukuran keluaran dari 158x158 menjadi 79x79 dan lapisan seterusnya. Lapisan *dense* merupakan lapisan *fully connected* dengan 200 unit neuron.

4.2.2 Summary Arsitektur *DenseNet121*

Model: "densenet121"

| Layer (type) | Output Shape | Param # | Connected to |
|---|---------------------|---------|-----------------------------|
| input_7 (InputLayer) | (None, 160, 160, 3) | 0 | [] |
| zero_padding2d_12 (ZeroPadding2D) | (None, 166, 166, 3) | 0 | ['input_7[0][0]'] |
| conv1/conv (Conv2D) | (None, 80, 80, 64) | 9408 | ['zero_padding2d_12[0][0]'] |
| conv1/bn (BatchNormalization) | (None, 80, 80, 64) | 256 | ['conv1/conv[0][0]'] |
| conv1/relu (Activation) | (None, 80, 80, 64) | 0 | ['conv1/bn[0][0]'] |
| zero_padding2d_13 (ZeroPadding2D) | (None, 82, 82, 64) | 0 | ['conv1/relu[0][0]'] |
| pool1 (MaxPooling2D) | (None, 40, 40, 64) | 0 | ['zero_padding2d_13[0][0]'] |
| conv2_block1_0_bn (BatchNormalization) | (None, 40, 40, 64) | 256 | ['pool1[0][0]'] |
| conv2_block1_0_relu (Activation) | (None, 40, 40, 64) | 0 | ['conv2_block1_0_bn[0][0]'] |
| ... | | | |
| Total params: 7037504 (26.85 MB) | | | |
| Trainable params: 6953856 (26.53 MB) | | | |
| Non-trainable params: 83648 (326.75 KB) | | | |

Gambar 14 Detail Arsitektur DenseNet121

Pada data ringkasan dari model arsitektur *DenseNet121* terdiri dari *layer* (lapisan) yang terdiri dari *input layer*, *convolutional layer*, *batch normalization layer*, *activation layer*, dan lainnya. Dan pada model ini menggunakan *activation function* ReLU (Rectified Linear Units). Pada *Output Shape* merupakan keluaran yang dihasilkan oleh tiap *layer*, seperti pada (None, 160, 160, 3) menunjukkan bahwa *input* yang masuk ke model merupakan gambar yang memiliki ukuran 160x160 piksel dengan 3 *channel* warna (RGB). *Param #* merupakan jumlah parameter yang dapat diubah dalam *layer* tersebut. *Total params* adalah total jumlah parameter yang dapat dipelajari oleh model, *Trainable params* merupakan jumlah parameter yang dapat diubah nilainya selama proses pelatihan model, dan *Non-trainable params* adalah jumlah parameter yang nilainya tidak diperbarui selama proses pelatihan (tetap).

4.3 Hasil Pelatihan Model

4.3.1 Hasil Pelatihan Model CNN

Pada bagian ini merupakan hasil pelatihan model CNN yang dilakukan dengan 25 siklus pelatihan (epochs).

```
Epoch 1/25 [=====] - 82s 43ms/step - loss: 0.8271 - accuracy: 0.6586 - precision_3: 0.6933 - recall_3: 0.5563 - auc_3: 0.6799 - categorical_crossentropy: 0.8271 - val_loss: 0.6394 - val_accuracy: 0.8000
Epoch 2/25 [=====] - 78s 48ms/step - loss: 0.6395 - accuracy: 0.7255 - precision_3: 0.7573 - recall_3: 0.6751 - auc_3: 0.9267 - categorical_crossentropy: 0.6395 - val_loss: 0.5287 - val_accuracy: 0.8000
Epoch 3/25 [=====] - 77s 48ms/step - loss: 0.5929 - accuracy: 0.7553 - precision_3: 0.7843 - recall_3: 0.7114 - auc_3: 0.9378 - categorical_crossentropy: 0.5929 - val_loss: 0.6526 - val_accuracy: 0.7000
Epoch 4/25 [=====] - 77s 48ms/step - loss: 0.5458 - accuracy: 0.7789 - precision_3: 0.8033 - recall_3: 0.7449 - auc_3: 0.9475 - categorical_crossentropy: 0.5458 - val_loss: 0.6434 - val_accuracy: 0.6000
Epoch 5/25 [=====] - 77s 48ms/step - loss: 0.5859 - accuracy: 0.7997 - precision_3: 0.8204 - recall_3: 0.7726 - auc_3: 0.9547 - categorical_crossentropy: 0.5859 - val_loss: 0.5683 - val_accuracy: 0.7000
Epoch 6/25 [=====] - 77s 48ms/step - loss: 0.4782 - accuracy: 0.8142 - precision_3: 0.8338 - recall_3: 0.7899 - auc_3: 0.9591 - categorical_crossentropy: 0.4782 - val_loss: 0.1976 - val_accuracy: 1.0000
Epoch 7/25 [=====] - 77s 48ms/step - loss: 0.4492 - accuracy: 0.8246 - precision_3: 0.8487 - recall_3: 0.8025 - auc_3: 0.9638 - categorical_crossentropy: 0.4492 - val_loss: 0.5762 - val_accuracy: 0.7000
Epoch 8/25 [=====] - 77s 48ms/step - loss: 0.4459 - accuracy: 0.8279 - precision_3: 0.8443 - recall_3: 0.8052 - auc_3: 0.9642 - categorical_crossentropy: 0.4459 - val_loss: 0.1216 - val_accuracy: 1.0000
Epoch 9/25 [=====] - 77s 48ms/step - loss: 0.4249 - accuracy: 0.8387 - precision_3: 0.8532 - recall_3: 0.8193 - auc_3: 0.9675 - categorical_crossentropy: 0.4249 - val_loss: 0.2359 - val_accuracy: 0.8000
Epoch 10/25 [=====] - 77s 48ms/step - loss: 0.4062 - accuracy: 0.8468 - precision_3: 0.8613 - recall_3: 0.8318 - auc_3: 0.9699 - categorical_crossentropy: 0.4062 - val_loss: 0.1127 - val_accuracy: 1.0000
Epoch 11/25 [=====] - 77s 48ms/step - loss: 0.4043 - accuracy: 0.8471 - precision_3: 0.8631 - recall_3: 0.8318 - auc_3: 0.9701 - categorical_crossentropy: 0.4043 - val_loss: 0.3334 - val_accuracy: 0.8000
Epoch 12/25 [=====] - 77s 48ms/step - loss: 0.3988 - accuracy: 0.8538 - precision_3: 0.8666 - recall_3: 0.8381 - auc_3: 0.9709 - categorical_crossentropy: 0.3988 - val_loss: 0.3144 - val_accuracy: 0.8000
Epoch 13/25 [=====]
...
Epoch 24/25 [=====]
Epoch 25/25 [=====] - 78s 48ms/step - loss: 0.3411 - accuracy: 0.8772 - precision_3: 0.8872 - recall_3: 0.8667 - auc_3: 0.9782 - categorical_crossentropy: 0.3411 - val_loss: 0.1239 - val_accuracy: 1.0000
Epoch 19/25 [=====] - 77s 48ms/step - loss: 0.3418 - accuracy: 0.8886 - precision_3: 0.8912 - recall_3: 0.8782 - auc_3: 0.9785 - categorical_crossentropy: 0.3418 - val_loss: 0.2111 - val_accuracy: 0.8000
```

Gambar 15 Hasil Pelatihan Model CNN

Pada Gambar 15 dapat dilihat proses pelatihan model CNN. Kinerja model meningkat yang mengindikasikan bahwa model terus belajar dan memperbaiki performanya hingga *validation accuracy* mencapai 99% dan 25 *epochs*, proses pelatihan berhenti.

4.3.2 Hasil Pelatihan Model DenseNet121

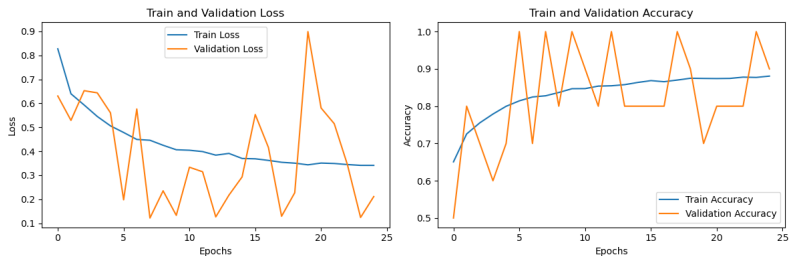
Pada bagian ini merupakan hasil pelatihan model *DenseNet121* yang dilakukan dengan 15 siklus pelatihan (epochs).

```
Epoch 1/15 [=====] - 860s 438ms/step - loss: 0.5210 - accuracy: 0.8016 - val_loss: 0.9812 - val_accuracy: 0.6000
Epoch 2/15 [=====] - 857s 441ms/step - loss: 0.3638 - accuracy: 0.8648 - val_loss: 4.0277 - val_accuracy: 0.3000
Epoch 3/15 [=====] - 1113s 573ms/step - loss: 0.3108 - accuracy: 0.8874 - val_loss: 0.1662 - val_accuracy: 0.9000
Epoch 4/15 [=====] - 1230s 633ms/step - loss: 0.2696 - accuracy: 0.9043 - val_loss: 0.4704 - val_accuracy: 0.8000
Epoch 5/15 [=====] - 873s 449ms/step - loss: 0.2248 - accuracy: 0.9176 - val_loss: 2.4637 - val_accuracy: 0.2000
Epoch 6/15 [=====] - 3066s 2s/step - loss: 0.2097 - accuracy: 0.9297 - val_loss: 0.8425 - val_accuracy: 1.0000
Epoch 7/15 [=====] - 3743s 2s/step - loss: 0.1934 - accuracy: 0.9308 - val_loss: 0.6132 - val_accuracy: 0.7000
Epoch 8/15 [=====] - 4017s 2s/step - loss: 0.1781 - accuracy: 0.9380 - val_loss: 0.8044 - val_accuracy: 1.0000
Epoch 9/15 [=====] - 1537s 791ms/step - loss: 0.1687 - accuracy: 0.9411 - val_loss: 0.3135 - val_accuracy: 0.8000
Epoch 10/15 [=====] - 2863s 1s/step - loss: 0.1476 - accuracy: 0.9488 - val_loss: 1.0273 - val_accuracy: 0.7000
Epoch 11/15 [=====] - 2046s 1s/step - loss: 0.1415 - accuracy: 0.9517 - val_loss: 0.8638 - val_accuracy: 1.0000
```

Gambar 16 Hasil Pelatihan Model DenseNet121

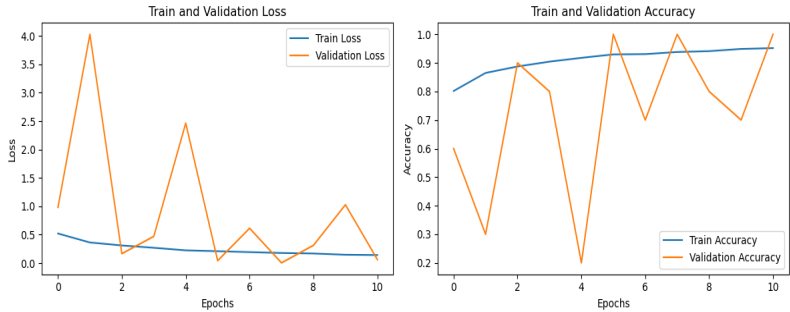
Pada Gambar 16 dapat dilihat proses pelatihan model *DenseNet121*. Kinerja model meningkat secara signifikan yang mengindikasikan bahwa model terus belajar dan memperbaiki performanya dari model CNN konvensional hingga *validation accuracy* mencapai 99% dan 11 *epochs*, proses pelatihan berhenti.

Train Loss mengukur nilai yang hilang ketika proses pelatihan berlangsung dan mengukur seberapa baik model melakukan prediksi yang benar, sehingga proses latihan ini ada untuk mengurangi *loss* agar model dapat memprediksi dengan baik. *Validation Loss* merupakan ukuran kesalahan rata-rata dari model saat diuji dengan data validasi dan memberikan indikasi seberapa baik model dapat melakukan prediksi dari data pelatihan ke data baru yang belum pernah dilihat sebelumnya. *Train Accuracy* merupakan tingkat prediksi benar yang dibuat oleh model pada data latih dan memberikan indikasi seberapa baik model dapat memprediksi pola-pola yang ada dalam data latih. *Validation Accuracy* merupakan tingkat prediksi benar yang dibuat oleh model terhadap data validasi.



Gambar 17 Train & Validation Loss, Train & Validation Accuracy dari Model CNN

Pada Gambar 17 menunjukkan perubahan nilai los pada grafik *Train*, dan *Validation* dari hasil proses pelatihan model menggunakan arsitektur CNN. Dapat dilihat pada *Train Loss* menurun secara signifikan seiring bertambahnya siklus pengulangan (*epochs*) yang menunjukkan model sedang belajar dan memperbaiki performanya pada data latih. Pada grafik *Validation Loss* terlihat sangat fluktuatif, yang dimana hal ini mengindikasikan terjadinya *overfitting* (model mempelajari data dengan sangat detail sehingga tidak dapat menggeneralisi dengan baik terhadap data yang baru). Pada grafik *Train Accuracy* terlihat meningkat secara signifikan dan stabil seiring bertambahnya *epochs*. Yang menunjukkan bahwa model memiliki performa yang baik pada proses pelatihan. *Validation Accuracy* terlihat bergerak secara fluktuatif dan tidak menunjukkan peningkatan yang konsisten. Hal ini juga mengindikasikan kemungkinan terjadinya *overfitting*.



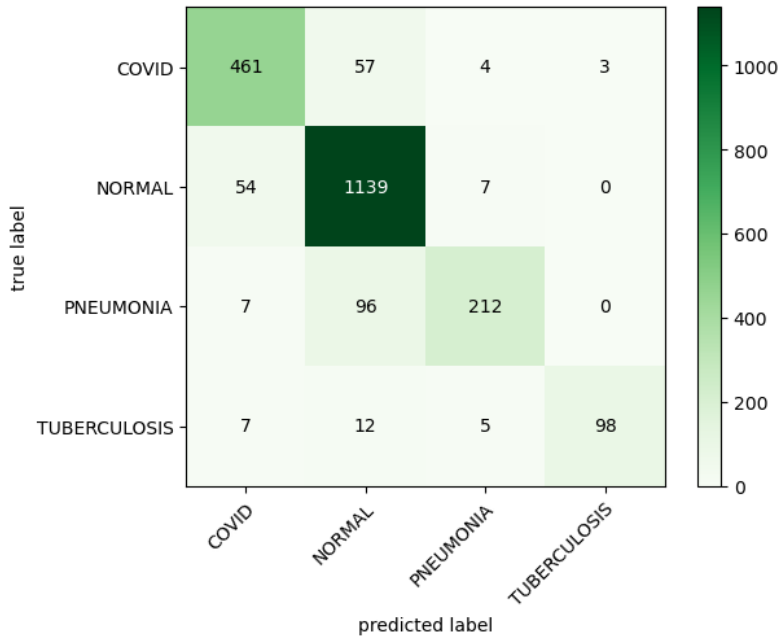
Gambar 18 Train & Validation Loss, Train & Validation Accuracy dari Model *DenseNet121*

Pada Gambar 18 merupakan grafik *Loss dan Accuracy* dari *Train & Validation* untuk model *DenseNet121*. Pada *Train Loss* dapat dilihat menurun secara bertahap dan konsisten dibanding arsitektur CNN konvensional setiap penambahan *epochs*. Pada *Validation Loss* grafik bergerak sangat fluktuatif tapi masih menunjukkan penurunan secara keseluruhan, yang mengindikasikan bahwa model dapat melakukan generalisasi data yang lebih baik dari arsitektur CNN. Kemudian untuk grafik *Train Accuracy* meningkat dan stabil seperti pada model CNN konvensional. Pada *Validation Accuracy* menunjukkan pergerakan yang fluktuatif, namun tetap terjadi peningkatan.

4.4 Hasil *Confusion Matrix*

4.4.1 *Confusion Matrix* model CNN

Pada hasil *confusion matrix* menunjukkan tabel hasil dari pelatihan model menggunakan arsitektur CNN. Dan dapat dilihat tabel matriks yang berisi jumlah data yang telah terklasifikasi dengan benar dan yang salah terklasifikasi

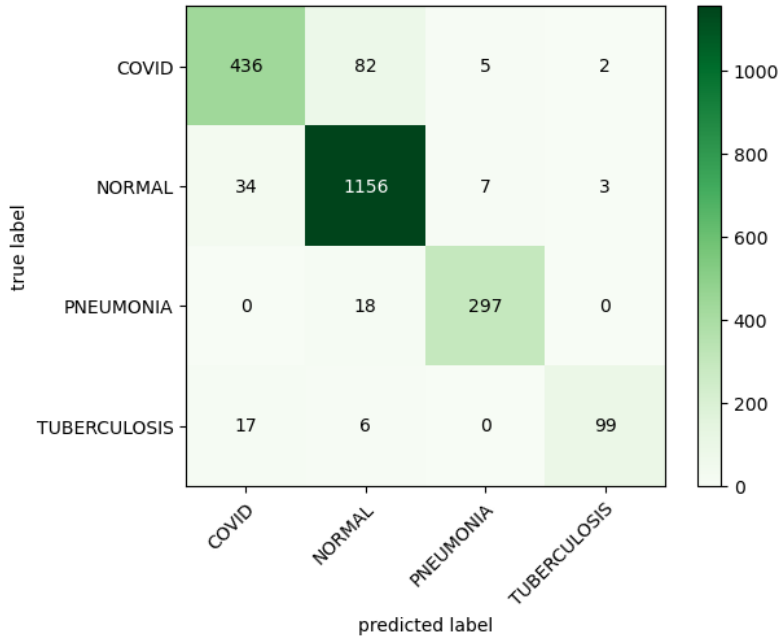


Gambar 19 *Confusion Matrix* CNN

Confusion matrix dari model CNN pada Gambar 17 dapat dilihat dari diagonal utama bahwa model memiliki kinerja yang baik dalam mengklasifikasi gambar dengan kelas NORMAL, PNEUMONIA, dan TUBERCULOSIS, karena jumlah *True Positive* (TP) untuk kelas-kelas tersebut relative tinggi. Dan dari 2162 data validasi, terdapat 252 prediksi data yang salah oleh model.

4.4.2 Confusion Matrix model DenseNet121

Pada hasil *confusion matrix* menunjukkan tabel hasil dari pelatihan model menggunakan arsitektur *DenseNet121*. Dan dapat dilihat tabel matriks yang berisi jumlah data yang telah terklasifikasi dengan benar dan yang salah terklasifikasi



Gambar 20 Confusion Matrix DenseNet121

Dan pada *confusion matrix* model DenseNet121 pada Gambar 18 dapat dilihat dari diagonal utama bahwa model memiliki kinerja yang baik lebih baik. Yang ditandai dengan meningkatnya jumlah prediksi kelas NORMAL dan PNEUMONIA, dan pada kelas TUBERCULOSIS tidak mengalami perubahan yang signifikan. Namun pada kelas COVID terdapat penurunan jumlah prediksi, karena model kesulitan dalam mengidentifikasi gambar dengan kelas COVID, karena jumlah *False Negative* (FN) untuk kelas ini cukup tinggi. Dari 2162 data validasi, terdapat 174 prediksi yang salah oleh model. Hal ini menunjukkan bahwa terdapat peningkatan hasil prediksi oleh model *DenseNet121* walau masih terdapat beberapa kesalahan prediksi pada tiap kelas yang ada oleh model.

Untuk menganalisis lebih lanjut hasil dari kedua model yaitu dengan melakukan evaluasi *classification report*, yang dapat dilihat pada Tabel 3 dan 4.

Tabel 3 Classification Report Model CNN

| | Accuracy | Precision | Recall | F1-score | support |
|--------------|----------|-----------|--------|----------|---------|
| COVID | 0.88 | 0.87 | 0.88 | 0.87 | 525 |
| NORMAL | 0.95 | 0.87 | 0.95 | 0.91 | 1200 |
| PNEUMONIA | 0.67 | 0.93 | 0.67 | 0.78 | 315 |
| TUBERCULOSIS | 0.80 | 0.97 | 0.80 | 0.88 | 122 |
| accuracy | 0.88 | | | | 2162 |
| Macro avg | | 0.91 | 0.83 | 0.86 | 2162 |
| Weighted avg | | 0.89 | 0.88 | 0.88 | 2162 |

Pada tabel 3 merupakan *classification report* dari model CNN konvensional. Dapat dilihat akurasi tiap kelas yang ada yaitu COVID 88%, NORMAL 95%, PNEUMONIA 67%, dan TUBERCULOSIS 80%. Untuk kelas PNEUMONIA memiliki akurasi yang tidak terlalu baik, sehingga didapat akurasi total sebesar 88%.

Tabel 4 Classification Report Model DenseNet121

| | Accuracy | Precision | Recall | F1-score | support |
|--------------|----------|-----------|--------|----------|---------|
| COVID | 0.85 | 0.94 | 0.85 | 0.89 | 525 |
| NORMAL | 0.97 | 0.93 | 0.97 | 0.96 | 1200 |
| PNEUMONIA | 0.97 | 0.95 | 0.97 | 0.96 | 315 |
| TUBERCULOSIS | 0.89 | 0.96 | 0.89 | 0.92 | 122 |
| accuracy | 0.92 | | | | 2162 |
| Macro avg | | 0.94 | 0.92 | 0.93 | 2162 |
| Weighted avg | | 0.94 | 0.94 | 0.93 | 2162 |

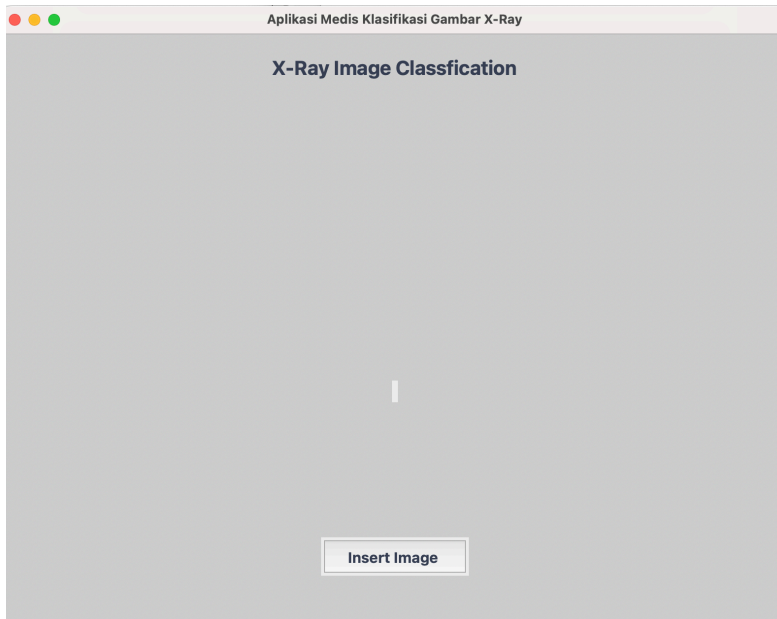
Pada tabel 4 merupakan *classification report* dari model *DenseNet121*. Dapat dilihat akurasi tiap kelas yang ada yaitu COVID 85%, NORMAL 97%, PNEUMONIA 97%, dan TUBERCULOSIS 89%. Sehingga didapat akurasi total sebesar 92%. Terdapat peningkatan yang signifikan terhadap akurasi model setelah dilakukan pelatihan menggunakan arsitektur *DenseNet121*.

Pada tabel hasil *classification report* dari model CNN dan *DenseNet121* di atas, didapatkan nilai *accuracy*, *precision*, *recall*, dan *f1-score*. Dapat dilihat tingkat akurasi pada model CNN yaitu sebesar 88%, rata-rata presisi 91%, *recall* 83%, dan *f1-score* 86%. Kemudian pada hasil model *DenseNet121* didapat tingkat akurasi yaitu sebesar 92%, rata-rata presisi 93%, *recall* 89%, dan *f1-score* 91%. Terdapat peningkatan hasil *classification report* dari pelatihan model menggunakan arsitektur *DenseNet121*. Yang dimana nilai *precision* menunjukkan bahwa model tidak membuat banyak kesalahan dalam melakukan prediksi terhadap kelas tertentu sebagai 'positif' dan nilai *recall* menunjukkan bahwa model memiliki

kemampuan untuk mengingat (*recall*) sebagian besar data yang relevan, sehingga data yang tidak terdeteksi dapat diminimalkan. Dan nilai *f1-score* yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara *precision* dan *recall*. Dari rata-rata yang ada, dapat dilihat bahwa akurasi (*accuracy*) dari model CNN sebelumnya meningkat dari 86% menjadi 92%, yang mengindikasikan model dari *DenseNet121* memiliki kinerja yang baik secara keseluruhan dalam mengklasifikasi gambar X-ray sesuai dengan kelas yang ada setelah dihubungkan dengan model CNN.

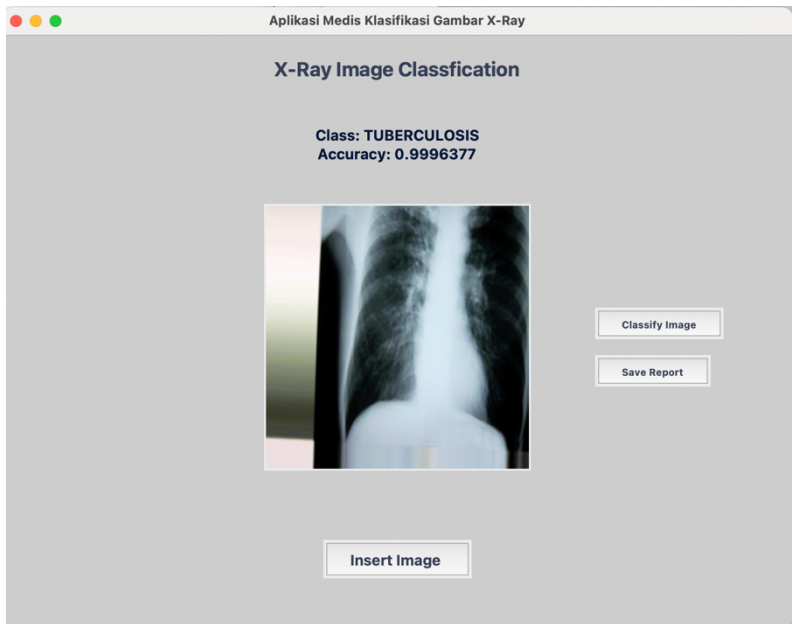
4.5 Graphical User Interface (GUI)

Setelah model selesai dilatih dan dievaluasi, model belum sepenuhnya dapat digunakan untuk pembacaan berulang karena hanya dengan program tunggal untuk melakukan pembacaan akan memakan banyak waktu dan tidak akan efektif untuk digunakan. Sehingga GUI diperlukan untuk melakukan klasifikasi gambar karena mempermudah interaksi antar pengguna dan model kecerdasan buatan. Karena dibekali dengan tombol dan alat yang akan melakukan klasifikasi secara langsung.



Gambar 21 Tampilan GUI

Setelah aplikasi terbuka, tombol *Insert Image* akan membuka *filedialog* untuk memasukkan gambar yang ingin diklasifikasi.



Gambar 22 Proses Klasifikasi Gambar

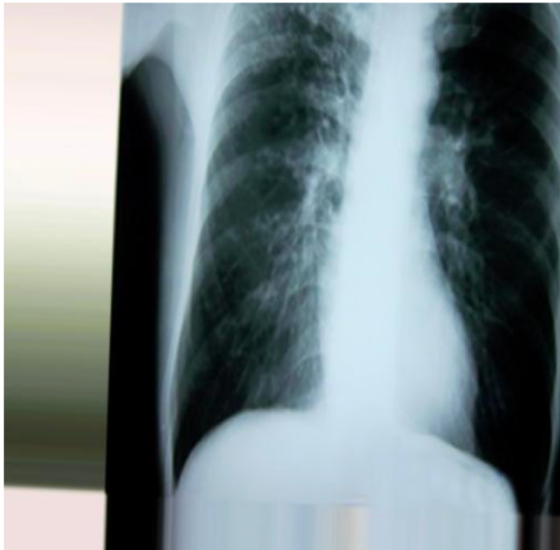
Apabila gambar telah masuk pada tampilan, maka tombol klasifikasi dan simpan akan muncul. Tombol *Classify Image* akan mengaktifkan model untuk melakukan klasifikasi pada gambar. Hasil prediksi dan tingkat akurasi akan keluar di atas gambar.

Classification Report

Class: TUBERCULOSIS

Accuracy: 0.9996377

File Path: /Users/teknikelektro/Documents/WOI/Validasi/TUBERCULOSIS/test_0_2723.jpeg



Gambar 23 Gambar Tersimpan dalam format PDF

Pada Gambar 23 merupakan contoh dari gambar hasil pembacaan yang telah tersimpan dalam bentuk format PDF beserta dengan informasi yang ada di dalamnya.

Bab 5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil penelitian Tugas Akhir ini, penulis dapat menyimpulkan bahwa penggunaan metode arsitektur *DenseNet121* yang dihubungkan dengan model dari arsitektur CNN dalam melakukan klasifikasi pada gambar x-ray memiliki performa kinerja yang lebih baik. Hal ini ditandai dengan tingkat akurasi yang awalnya sebesar 88% meningkat menjadi 92%. Dan dari hasil klasifikasi gambar rontgen terhadap 4 kelas kondisi pada gambar x-ray yang terintegrasi dengan GUI ini praktis digunakan dan efisien. Yang diharapkan dapat membantu serta mempersingkat waktu dalam proses diagnosis terhadap paru-paru pasien.

5.2 Saran

Dalam pengerjaan Tugas Akhir ini masih terdapat kesalahan dan kekurangan. Adapun saran dari penulis, yaitu:

1. Penambahan jumlah data yang digunakan
Penggunaan kecerdasan buatan untuk klasifikasi gambar memerlukan banyak data gambar untuk meningkatkan hasil akurasi dari prediksi kecerdasan buatan. Oleh karena itu, penulis memberikan saran untuk mencari dataset dari sumber yang terpercaya dengan jumlah yang memadai serta dengan format yang sesuai agar lebih banyak keragaman data yang dimiliki.
2. Penambahan label dan anotasi
Penulis menyarankan untuk menambahkan label dan anotasi pada pelatihan model terhadap gambar x-ray, dengan harapan dokter spesialis atau ahli radiologi dapat fokus menganalisis bagian atau daerah pada paru yang mengalami kelainan dari kondisi normal yang seharusnya. diharapkan adanya kerjasama dan sinergi dengan ahli radiologi atau dokter spesialis paru-paru agar dapat mengembangkan aplikasi klasifikasi gambar x-ray yang lebih baik.

Daftar Pustaka

- [1] Unicef, "Unicef.org," 12 11 2019. [Online]. Available: <https://www.unicef.org/indonesia/id/siaran-pers/lembaga-kesehatan-dan-anak-memeringatkan-satu-anak-meninggal-akibat-pneumonia-setiap-39>. [Accessed 25 07 2023].
- [2] T. D. Kristini and R. Hamidah, "Potensi Penularan Tuberculosis Paru pada Anggota Keluarga Penderita," *JURNAL KESEHATAN MASYARAKAT INDONESIA*, vol. 15, no. 1, pp. 24-28, 2020.
- [3] K. K. R. Indonesia, "kemenkes.go.id," 31 3 2023. [Online]. Available: <https://www.kemkes.go.id/article/view/23033100001/deteksi-tbc-capai-rekor-tertinggi-di-tahun-2022.html>. [Accessed 25 07 2023].
- [4] J. Prayitno, R. Admirasari, J. P. Susanto and R. Nugroho, "TINJAUAN TEKNOLOGI INAKTIVASI VIRUS UNTUK PENANGGULANGAN PANDEMI COVID-19," *JURNAL BIOTEKNOLOGI & BIOSAINS INDONESIA*, vol. 8, no. 1, pp. 137-154, 2021.
- [5] K. K. R. Indonesia, "INFEKSIEMERGING," 05 07 2023. [Online]. Available: <https://infeksiemerging.kemkes.go.id/dashboard/covid-19>. [Accessed 27 07 2023].
- [6] d. V. V. Handayani, "halodoc.com," 09 12 2022. [Online]. Available: <https://www.halodoc.com/artikel/ketahui-tugas-tugas-dari-spesialis-radiologi>. [Accessed 24 March 2023].
- [7] M. Saiful, LM.Samsu and Fathurrahman, "Sistem Deteksi Infeksi COVID-19 Pada Hasil X-ray Rontgen menggunakan Algoritma Convolutional Neural Network (CNN)," *Infotek : Jurnal Informatika dan Teknologi*, vol. 4, no. 2, pp. 217-227, 2021.
- [8] N. H. Harani, C. Prianto and M. Hasanah, "Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python," *Jurnal Teknik Informatika*, vol. 11, no. 3, pp. 47-53, 2019.
- [9] N. Yudistira, A. W. Widodo and B. Rahayudi, "DETEKSI COVID-19 PADA CITRA SINAR-X DADA MENGGUNAKAN DEEP LEARNING YANG EFISIEN," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 7, no. 6, pp. 1289-1296, 2020.
- [10] F. A. Natasya, "TATALAKSANA PNEUMONIA," *Jurnal Medika Utama*, vol. 3, no. 2, pp. 2392-2399, 2022.

- [11] W. H. KHOONG, "Kaggle.com," 12 2020. [Online]. Available: <https://www.kaggle.com/datasets/khoongweihao/covid19-xray-dataset-train-test-sets>. [Accessed 24 3 2023].
- [12] A. Raup, W. Ridwan, Y. Khoeriyah, Supiana and Q. Y. Zaqiah, "Deep Learning dan Penerapannya dalam Pembelajaran," *JlIP (Jurnal Ilmiah Ilmu Pendidikan)*, vol. 5, no. 9, pp. 3258-3267, 2022.
- [13] I. W. S. E. P, A. Y. Wijaya and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," *JURNAL TEKNIK ITS*, vol. 5, no. 1, pp. A65-A69, 2016.
- [14] B. P. Hartato, "Penerapan Convolutional Neural Network pada Citra Rontgen Paru-Paru untuk Deteksi SARS-CoV-2," *Jural Resti*, vol. 5, no. 4, pp. 747-759, 2021.
- [15] R. R. I. A. F. Wahyu Wijaya Kusuma, "ANALISIS PERBANDINGAN MODEL CNN VGG16 DAN DENSENET121 MENGGUNAKAN KERANGKA KERJA TENSORFLOW UNTUK DETEKSI JENIS HEWAN," *Jurnal Teknik Komputer*, vol. 1, no. 4, pp. 141-147, 2023.
- [16] D. A. L. P. Jasman Pardede, "Implementasi DenseNet Untuk Mengidentifikasi Jenis Kanker Kulit Melanoma," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 3, pp. 425-433, 2020.

Lampiran