**KLASIFIKASI PENYAKIT TUBERKULOSIS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi

Persyaratan Guna Meraih Gelar Sarjana

Informatika Universitas Muhammadiyah Malang



Husein Abdul Hamid

201810370311202

**Bidang Minat**

Sains Data

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH MALANG**

**2022**

LEMBAR PERSETUJUAN

**KLASIFIKASI PENYAKIT TUBERKULOSIS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**

**TUGAS AKHIR**

**Sebagai Persyaratan Guna Meraih Gelar Sarjana Strata 1**

**Informatika Universitas Muhammadiyah Malang**

Menyetujui,

Malang, 10 Juni 2022

|  |  |
| --- | --- |
| Pembimbing I | Pembimbing II |
|  |  |
| **Yufis Azhar, S.Kom., M.Kom** | **Zamah Sari, S.T., M.T** |
| **NIP. 108.1410.0544** | **NIP. 108.1410.0555** |

# LEMBAR PENGESAHAN

**KLASIFIKASI PENYAKIT TUBERKULOSIS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**

**TUGAS AKHIR**

Sebagai Persyaratan Guna Meraih Gelar Sarjana Strata 1

Informatika Universitas Muhammadiyah Malang

Disusun Oleh:

**Husein Abdul Hamid**

**NIM.** **201810370311202**

Tugas Akhir ini telah diuji dan dinyatakan lulus melalui sidang majelis penguji pada tanggal Tanggal – Bulan - Tahun

Menyetujui,

|  |  |
| --- | --- |
| Penguji I | Penguji II |
|  |  |
| **Nama Dosen** | **Nama Dosen** |
| **NIP.** | **NIP.** |
| Mengetahui, | |
| Ketua Jurusan Program Studi Informatika | |
|  | |
| **Galih Wasis Wicaksono, S.Kom., M.Cs.**  **NIP. 108.1410.0541** | |

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

**NAMA : Husein Abdul Hamid**

**NIM : 201810370311202**

**FAK./JUR. : TEKNIK/INFORMATIKA**

Dengan ini saya menyatakan bahwa Tugas Akhir dengan judul “**KLASIFIKASI PENYAKIT TUBERKULOSIS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**” beserta seluruh isinya adalah karya saya sendiri dan bukan merupakan karya tulis orang lain, baik sebagian maupun seluruhnya, kecuali dalam bentuk kutipan yang telah disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini maka saya siap menanggung segala bentuk resiko/sanksi yang berlaku.

|  |  |
| --- | --- |
| Mengetahui,  Dosen Pembimbing  **Yufis Azhar, S.Kom., M.Kom**. | Malang, Tgl Bulan Tahun  Yang Membuat Pernyataan  Materai 10.000  **Husein Abdul Hamid** |

ABSTRAK

Ringkasan isi laporan tugas akhir dalam bahasa indonesia.

**Kata Kunci:** kata kunci yang ada dalam tugas akhir dalam bahasa indonesia.

ABSTRACT

Abstract (ringkasan isi laporan tugas akhir) in english

**Keywords:** kata kunci in english.

LEMBAR PERSEMBAHAN

Alhamdulillahi rabbil ‘alamin, puji syukur atas kehadiran Allah SWT atas limpahan rahmat serta nikmat-Nya sehingga peneliti dapat menyelesaikan tugas akhir ini. Dalam penyusunan tugas akhir ini tidak luput dari bantuan orang-orang di sekeliling peneliti, oleh karena itu peneliti ingin menyampaikan ucapan terima kasih kepada :

1. Allah SWT yang selalu memberikan kemudahan, kelancaran, dan petunjuk dalam pengerjaan tugas akhir ini.
2. Kedua orang tua saya Bapak Bambang Setiawan dan Ibu Siti Choiriyah yang telah mendidik saya dan memberikan dukungan serta terhadap penelitian ini.
3. Dekan Fakultas Teknik Bapak Prof.Ir. Ilyas Masudin, MLogSCM., Ph.D. dan Keluarga (FT) Serta para Pembantu Dekas Fakultas Teknik dan keluarga besar Universitas Muhammadiyah Malang.
4. Ketua Jurusan Informatika Bapak Ir. Galih Wasis Wicaksono, S.kom. M.Cs. dan Sekretaris Jurusan Informatika Denar Regata Akbi, S.Kom., M.Kom. beserta seluruh jajaran staf.
5. Bapak Yufis Azhar, S.Kom., M.Kom. dan Bapak Zamah Sari, ST., MT. yang tekah meluangkan waktu untuk membimbing penulis dalam menyelesaikan skripsi ini
6. Ibu penguji..
7. Seluruh Civitas Akademika (dosen, asisten, dan karyawan) Universitas Muhammadiyah Malang yang telah membekai ilmu dan membantu penulis selama proses studi.
8. Pasangan, Sahabat dan keluarga saya disini Iky, Yusuf, Fauzan, Krisna, Amirul, Dewa, Anjas, Ridho, Aldi, Rio,
9. …., dst

Malang, tgl bulan tahun

Penulis

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT. Atas limpahan rahmat dan hidayah-NYA sehingga peneliti dapat menyelesaikan tugas akhir yang berjudul: **KLASIFIKASI PENYAKIT TUBERKULOSIS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**”.

Di dalam tulisan ini disajikan pokok-pokok pembahasan yang meliputi desain tampilan aplikasi, kinerja aplikasi, hingga hasil pengujian dari penggunaan aplikasi, yang diharapkan dengan adanya aplikasi ini dapat bermanfaat bagi masyarakat khususnya dalam hal pertanian.

Peneliti menyadari sepenuhnya bahwa dalam penulisan laporan tugas akhir ini masih banyak kekurangan dan keterbatasan. Oleh karena itu peneliti mengharapkan saran yang membangun agar tulisan ini bermanfaat bagi perkembangan ilmu pengetahuan.

Malang, tgl bulan tahun

Penulis

# DAFTAR ISI

[LEMBAR PERSETUJUAN ii](#_Toc105516508)

[LEMBAR PENGESAHAN iii](#_Toc105516509)

[LEMBAR PERNYATAAN iv](#_Toc105516510)

[ABSTRAK v](#_Toc105516511)

[ABSTRACT vi](#_Toc105516512)

[LEMBAR PERSEMBAHAN vii](#_Toc105516513)

[KATA PENGANTAR viii](#_Toc105516514)

[DAFTAR ISI ix](#_Toc105516515)

[DAFTAR BAGAN xi](#_Toc105516516)

[DAFTAR GAMBAR xii](#_Toc105516517)

[DAFTAR TABEL xiii](#_Toc105516518)

[BAB I PENDAHULUAN 1](#_Toc105516519)

[1.1 Latar Belakang 1](#_Toc105516521)

[1.2 Rumusan Masalah 4](#_Toc105516522)

[1.3 Tujuan Penelitian 4](#_Toc105516523)

[1.4 Batasan Masalah 5](#_Toc105516524)

[BAB II TINJAUAN PUSTAKA 6](#_Toc105516525)

[2.1 Studi Literatur 6](#_Toc105516527)

[2.2 Klasifikasi Citra 7](#_Toc105516528)

[2.3 Deep Learning 7](#_Toc105516529)

[2.4 Algoritma CNN 7](#_Toc105516530)

[2.5 Keras 9](#_Toc105516533)

[2.6 TensorFlow 9](#_Toc105516534)

[2.7 Numpy 10](#_Toc105516535)

[2.8 Matplotlib 10](#_Toc105516536)

[2.9 Open CV 10](#_Toc105516537)

[2.10 Pengujian Klasifikasi 10](#_Toc105516538)

[BAB III METODOLOGI PENELITIAN 12](#_Toc105516540)

[3.1 Identifikasi Masalah 13](#_Toc105516543)

[3.2 Pengumpulan DataSet 13](#_Toc105516544)

[3.3 Pembagian Dataset 14](#_Toc105516547)

[3.4 Data Augmentasi 14](#_Toc105516548)

[3.5 Implementation CNN Model 14](#_Toc105516550)

[3.6 Evaluation Result 16](#_Toc105516552)

[BAB IV HASIL DAN PEMBAHASAN 17](#_Toc105516554)

[4.1 Implementasi Skenario Model 1 17](#_Toc105516556)

[4.2 Implementasi Skenario Model 2 20](#_Toc105516564)

[4.3 Implementasi Skenario Model 3 24](#_Toc105516572)

[4.4 Hasil Skenario Model 1 28](#_Toc105516581)

[4.5 Hasil Skenario Model 2 31](#_Toc105516586)

[4.6 Hasil Skenario Model 3 34](#_Toc105516591)

[4.7 Perbandingan Performa Model CNN 37](#_Toc105516596)

[BAB V PENUTUP 40](#_Toc105516599)

[5.1 Kesimpulan 40](#_Toc105516601)

[5.2 Saran 40](#_Toc105516602)

[REFERENSI 41](#_Toc105516603)

# DAFTAR BAGAN

[Bagan 1. Flow Penelitian 12](#_Toc105516542)

[Bagan 2. Data Augmentasi 14](#_Toc105516549)

# DAFTAR GAMBAR

[**Gambar 1.** Convolutional Layer 8](#_Toc105516089)

[**Gambar 2.** Pooling Layer 8](#_Toc105516090)

[**Gambar 3.** Normal. 13](#_Toc105516091)

[**Gambar 4.** Tuberculosis 13](#_Toc105516092)

[**Gambar 5.** Load Data Model 1 17](#_Toc105516093)

[**Gambar 6.** Split Data Model 1 18](#_Toc105516094)

[**Gambar 7.** Augmentasi Data Model 1 18](#_Toc105516095)

[**Gambar 8.** Perancangan Model 1 19](#_Toc105516096)

[**Gambar 9.** Pelatihan Model 1 19](#_Toc105516097)

[**Gambar 10.** Visualisasi dan Loss Model 1 20](#_Toc105516098)

[**Gambar 11.** Pengujian Model 1 20](#_Toc105516099)

[**Gambar 12.** Load Data Model 2 21](#_Toc105516100)

[**Gambar 13.** Split Data Model 2 21](#_Toc105516101)

[**Gambar 14.** Augmentasi Data Model 2 22](#_Toc105516102)

[**Gambar 15.** Perancangan Model 2 22](#_Toc105516103)

[**Gambar 16.** Pelatihan Model 2 23](#_Toc105516104)

[**Gambar 17.** Visualisasi dan Loss Model 23](#_Toc105516105)

[**Gambar 18.** Pengujian Model 2 24](#_Toc105516106)

[**Gambar 19.** Load Data Model 3 24](#_Toc105516107)

[**Gambar 20.** Preprocessing Gathering Data Model 3 25](#_Toc105516108)

[**Gambar 21.** Preprocessing Normalization Model 3 25](#_Toc105516109)

[**Gambar 22.** Preprocessing Labelling Encoder Model 3 26](#_Toc105516110)

[**Gambar 23.** Perancangan Model 3 26](#_Toc105516111)

[**Gambar 24.** Pelatihan Model 3 27](#_Toc105516112)

[**Gambar 25.** Visualisasi dan Loss Model 3 27](#_Toc105516113)

[**Gambar 26.** Pengujian Model 3 28](#_Toc105516114)

[**Gambar 27.** Model 1 Parameter 29](#_Toc105516115)

[**Gambar 28.** Graph Acc Loss Model 1 30](#_Toc105516116)

[**Gambar 29.**Classification Report Model 1 30](#_Toc105516117)

[**Gambar 30.** Confussion Matrix Model 1 31](#_Toc105516118)

[**Gambar 31.** Model 2 Parameter 32](#_Toc105516119)

[**Gambar 32.** Graph Acc Loss Model 2 33](#_Toc105516120)

[**Gambar 33.** Classification Report Model 2 33](#_Toc105516121)

[**Gambar 34.** Confussion Matrix Model 2 34](#_Toc105516122)

[**Gambar 35.** Model 3 Parameter 35](#_Toc105516123)

[**Gambar 36.**Graph Acc Loss Model 3 36](#_Toc105516124)

[**Gambar 37.** Classification Report Model 3 36](#_Toc105516125)

[**Gambar 38.** Confussion Matrix Model 3 37](#_Toc105516126)

# DAFTAR TABEL

[Tabel 1. Pengujian Klasifikasi 11](#_Toc105516539)

[Tabel 2. Modelling 15](#_Toc105516551)

[Tabel 3. Confusion Matrix 16](#_Toc105516553)

[Tabel 4. Performa Model 37](#_Toc105516597)

[**Tabel 5.** Hasil Performa Model 38](#_Toc105516598)

# BAB I

# PENDAHULUAN

1. **Latar Belakang**

Tuberkulosis (TB) adalah penyakit menular yang disebabkan oleh *Mycobacterium Tuberculosis* dimana virus ini menyerang khususnya adalah paru-paru [1]. Penyakit ini menyebabkan kematian dari 1.8 juta orang di seluruh dunia, menurut data dari *World Health Organization* pada tahun 2018 bahwa hampir sekitar 10 juta orang terdampak penyakit TB dan sekitar 98.000 orang di antaranya meninggal dunia [2]. Penyakit ini dapat disembuhkan dan dapat dicegah, tetapi pada komunitas yang miskin sumber dayanya dan terpinggirkan dengan lemahnya infrastruktur kesehatan menyebabkan sulitnya penyakit ini dideteksi sejak dini [3]. Karena sumber daya yang belum memadai untuk mendiagnosa penyakit ini lebih baik dan tindak lanjut pengobatan yang efektif. Untuk mendeteksi penyakit TB sejak dini sangat penting agar penderita dapat segera di obati dan di minimalisir [4].

Perkembangan teknologi terkhususnya komputer teknik telah membantu mempercepat diagnosis TB diantara daerah sumber kemiskinan. Salah satunya dengan Teknologi *Computer Vision* dan *Machine Learning* menyebabkan perkembangan dan pemanfaatan di segala aspek termasuk salah satunya berada di bidang kesehatan [2]. Komputer digunakan sebagai salah satu alat untuk memahami informasi serta mengenali penyakit selayaknya seorang dokter yang mampu mendiagnosis seseorang saat terkena penyakit [5]. Salah satu bentuk perkembangannya adalah komputer dapat mengenali penyakit yang terdapat pada citra selayaknya manusia [6]. Salah satu penyakit yang dapat dikenali adalah *Tuberculosis* yang terdapat pada paru-paru manusia

Citra x-ray merupakan hal yang sering dilakukan dalam pemeriksaan radiologi untuk *screening* dan mendiagnosa dari banyaknya penyakit paru-paru [7]. Terdapat banyak teknik yang dapat diperoleh dari Citra x-ray seperti *Magnetic Resonance Imaging* (MRI), *Computed Tomography* (CT), atau *Positron Emission Tomography* (PET) [8]. Teknik tersebut digunakan dengan tujuan melihat serta menggambarkan struktur organ-organ manusia dengan tujuan memudahkan banyak orang dalam mengenali penyakit yang diderita oleh seseorang. Dalam menggunakan citra *x-ray* juga diperlukan analisis agar hasil yang didapatkan dapat diimplementasikan terhadap tindakan selanjutnya seperti pengobatan jika seseorang terindikasi memiliki suatu penyakit pada gambar citra. Tidak semua orang mampu membaca dan melihat hasil dari citra x-ray, disinilah tugas dari seorang radiologi untuk membaca dan melihat dari berbagai sudut pandang apakah pasien tersebut terindikasi atau tidak.

Seiring berkembangnya waktu serta diikuti oleh peningkatan populasi manusia maka perlu adanya langkah yang cepat dan solutif untuk mendeteksi penyakit ini. Sedikitnya ahli yang dapat membaca dan menganalisa dari citra *x-ray* menyebabkan keterlambatan dan kematian pasien [9]. Masalah utamanya diperlukan metode yang cepat, akurat untuk menentukan apakah seseorang ini terindikasi penyakit. Metode yang dapat digunakan untuk melihat segala kemungkinan dan prediksi yang tepat dalam mengklasifikasikan jenis suatu penyakit yang dapat dilihat dari hasil citra x-ray adalah dengan menggunakan *deep learning* [10]*.*

Dalam perkembangan *computer vision* terdapat teknik *deep learning*, teknik ini secara otomatis dapat mendeteksi dan mengklasifikasikan berbagai penyakit dengan akurasi yang lebih baik. Banyak hal yang dapat dilakukan pada teknik ini, dengan pemanfaatan yang benar serta pengolahan data yang baik akan menghasilkan suatu informasi yang bisa diolah dan dikembangkan sehingga bermanfaat oleh banyak orang [11]. Dengan memanfaatkan teknik *Deep Learning* akan meningkatkan hasil dari performa segmentasi paru paru dan klasifikasi berbagai penyakit [12].

Dalam beberapa tahun terakhir, metode *deep learning* menjanjikan untuk kemajuan pembelajaran dan representasi. Tujuannya adalah mempelajari berbagai tingkatan representasi dan abstraksi dari banyaknya data yang dapat dikelola seperti gambar, video, suara dan tulisan[3]. Dalam ilmu Kesehatan terkhususnya gambar analisa kesehatan, beberapa ahli mencoba berbagai macam model untuk dikembangkan seperti pendeteksian tumor, diagnosis kanker, dan pencegahan berbagai penyakit. Penerapan metode *deep learning* saat mengklasifikasikan berbagai penyakit yang terdapat pada paru paru diharapkan menghasilkan ketepatan akurasi serta efisiensi yang lebih baik.

Salah satu metode dari *deep learning* yang mampu menghasilkan akurasi yang tepat serta efisiensi yang lebih baik adalah penggunaan *Convolutional Neural Network* (CNN)[13]. Model ini dapat secara automatis mendeteksi dan mendiagnosa klasifikasi gambar penyakit, mendeteksi suatu objek dengan cepat, dan membuat system tanya jawab visual. Pada penelitian sebelumnya yang akan dijadikan paper referensi utama karya Mustapha Oloko-Oba and Serestina Viriri[13] pada tahun 2020, tujuannya adalah menghasilkan model yang dapat membantu pendeteksian penyakit Tuberkulosis secara akurat. Penelitian ini menggunakan model CNN yang terdiri dari banyak layer seperti  *Convolutional layer, Maxpooling* dan *Full Connected Layer* sebagai layer utamanya tanpa ada data augmentasi. Dengan skema pengujian yang telah dilakukan mendapatkan akurasi sebesar 87%.

Penelitian terkait juga dilakukan oleh Ovy Rochmawanti, Fitri Utaminingrum , Fitra A. Bachtiar pada tahun 2021[2], penelitiaan ini bertujuan menentukan model CNN yang mampu memberikan akurasi tinggi saat mendeteksi Tuberkulosis. Pengujian pada penelitian ini menggunakan lima pre-trained yang disediakan oleh keras diantaranya adalah *ResNet50, DenseNet121, MobileNet, Xception, InceptionV3, InceptionResNetV2*. Tahapan yang dilakukan pada paper ini diantaranya adalah membagi data latih, validasi dan data uji dengan beberapa ukuran inputan serta melakukan proses augmentasi data. Dengan skema pengujian pada penelitian ini menghasilkan satu model dengan akurasi yang sangat baik yaitu *DenseNet121* sebesar 91.57% dalam mendeteksi penyakit TB.

Augementasi data adalah teknik yang umumnya digunakan oleh banyak ahli pemodelan dibidang data untuk memanipulasi suatu gambar data tanpa menghilangkan eksistensi dari gambar asli dari data yang dimiliki[14]. Sehingga pada pengolahan data augementasi ini sangat berpengaruh dalam menghasilkan nilai akurasi yang ingin dicari, misalnya teknik yang sering digunakan adalah memutar gambar dan memperbesar gambar lalu gambar yang telah di modifikasi ini merupakan gambar yang berbeda dari aslinya kemudian komputer akan mencocokkan gambar yang telah di modifikasi dengan gambar asli melalui banyak tahap untuk meningkatkan akurasi dari model yang akan digunakan selanjutnya[15]. Selain meningkatkan akurasi saat menggunakan suatu model, data augmentasi juga dapat melakukan optimalisasi data dengan cara menambahkan variasi data untuk pemrosesan model yang akan digunakan[16].

Berdasarkan paper sebelumnya yang digunakan sebagai rujukan [2], [13], penulis memiliki hipotesa yaitu menggunakan metode yang sama pada paper karya Mustapha Oloko-Oba and Serestina Viriri yaitu penggunaan struktur CNN seperti Convolutional layer, MaxPooling dan Fully connected layer. Metode ini di gunakan pada dataset “*Tuberculosis (TB) Chest X-ray Database*” yang diperoleh dari situs komunitas daring ilmuwan data dan praktisi pembelajaran mesin yaitu Kaggle sebagai data yang akan diolah dan model yang akan dikembangkan pada penelitian kali ini. Oleh karena itu penelitian tugas akhir diputuskan menggunakan metode CNN yang didukung dengan penggunaan augmentasi data serta mencari dan menemukan model yang tepat dalam pemanfaatan citra X-ray sebagai objek penelitian. Sehingga judul yang tepat adalah “KLASIFIKASI PENYAKIT TUBERKULOSIS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK”.

1. **Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat di rumuskan penelitian ini berfokus pada :

1. Bagaimana mengimplementasikan metode CNN untuk mendeteksi penyakit Tuberkulosis pada citra X-Ray?
2. Bagaimana menguji pengaruh data agumentasi terhadap nilai akurasi yang dihasilkan oleh model CNN?
3. Bagaimana melakukan hyperparameter tuning yang tepat untuk memaksimalkan nilai akurasi model CNN?
4. **Tujuan Penelitian**

Penelitian ini bertujuan untuk membuat suatu model yang dapat mendeteksi penyakit tuberkulosisis pada citra X-Ray dengan menggunakan metode Convolutional Neural Network. Untuk dapat menghasilkan akurasi yang terbaik, teknik augmentasi data dan hyperparameter tuning juga akan diterapkan pada model.

1. **Batasan Masalah**

Adapun Batasan masalah pada penelitian kali ini yaitu :

1. Penelitian ini hanya terfokus kepada klasifikasi jenis penyakit paru paru yaitu *Tuberculosis* yang hanya terdiri dari 2 kelas(Normal dan Tuberculosis).
2. Dataset yang digunakan adalah “Tuberculosis (TB) Chest X-ray Database” yang diperoleh dari situs komunitas daring ilmuwan data dan praktisi pembelajaran mesin yaitu Kaggle yang didapatkan dari beberapa peneliti yang saling berkolaborasi. Dengan bantuan doketer Hamad Medical Corporation dan Bangladesh telah membuat database gambar rontgen dada untuk Tuberkulosis (TB).

# BAB II

# TINJAUAN PUSTAKA

1. **Studi Literatur**

Adapun penelitian sebelumnya yang digunakan sebagai rujukan pada penelitian ini sebagai berikut :

Penelitian Mustapha Oloko-Oba and Serestina Viriri dengan judul “*Diagnosing Tuberculosis Using Deep Convolutional Neural Network*” menggunakan sebuah metode *Convolutional Neural Network* (CNN) untuk mengklasifikasikan penyakit tuberkulosis yang terdapat pada paru-paru. Pada penelitian ini berfokus pada 2 kelas yaitu Tuberkulosis dan Normal dengan akurasi yang diperoleh sebesar 87.1%. Dataset yang digunakan berasal dari *The Montgomery County* (MC) CXR dataset dengan jumlah sebesar 138 data berformat png. Pada tahap *preprocessing* menggunakan data augmentasi seperti *horizontal left* dan *right, rando zoom, flip left* dan *right resizing*. Pada CNN struktur menggunakan tahap ekstraksi terdiri dari *convolution layer, batch normalization, relu activation function, dropout,* dan *max pooling* sedangkan tahap klasifikasi berisi *fully connected layer, flatten, density* dan fungsi aktivasi *softmax*.

Penelitian Ovy Rochmawanti , Fitri Utaminingrum , Fitra A. Bachtiar dengan judul “Analisis Performa Pre-Trained Model Convolutional Neural Network Dalam Mendeteksi Penyakit Tuberkulosis”. Pada penelitian ini berfokus pada 2 kelas yaitu Normal dan TB yang didapatkan dari data Shenzhen No.3 People’s Hospital, Guangdong Medical College Shenzhen China dengan jumlah data 662 data berformat png. Pada tahap preprocessing menggunakan teknik data augmentasi.

Augementasi dengan mengubah dan memodifikasi gambar sedemikian rupa seperti membalikkan gambar secara horizontal, merotasi dan melakukan pembesaran (zoom in) gambar secara acak serta menggeser gambar secara *horizontal dan vertical*[2]. Penelitian ini menggunakan model arsitektur CNN yaitu pre-trained model seperti *ResNet50, DenseNet121, MobileNet, Xception, InceptionV3, dan InceptionResNetV2.* Selain pre-trained model, terdapat layer yang ditambahkan seperti Global Average Pooling (GAP) layer, dropout layer, batch normalization dan fully connected layer. Setelah melalui tahap pengujian didapatkan bahwa model DenseNet121 mendapatkan akurasi yang terbaik dengan 91,57%.

1. **Klasifikasi Citra**

Merupakan suatu metode yang digunakan untuk mengelompokkan suatu gambar menjadi beberapa kategori dan mengetahui klasifikasi suatu gambar. Klasifikasi gambar merupakan salah satu teknik dari *Machine learning* yang memiliki banyak kegunaan di antaranya mampu mengidentifikasi jenis suatu penyakit berdasarkan gambar yang diperoleh dari citra MRI pasien. Dalam klasifikasi yang terfokus pada *training set* diberi label atau tag. Saat melakukan klasifikasi maka seorang peneliti akan membuat suatu model yang akan dibandingkan dengan label sebenarnya dan akan melihat seberapa akuratnya model yang dibuat dalam mengklasifikasikan suatu objek serta melihat seberapa tinggi kesalahan yang dibuat. Klasifikasi ini memberikan performa yang tinggi dalam mengelompokkan suatu gambar. Salah satu metode yang dapat digunakan dalam jenis klasifikasi terawasi(Classification Supervised) adalah *deep learning.*

1. **Deep Learning**

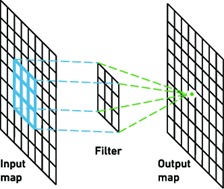
Deep Learning adalah suatu teknik tertentu dengan tujuan teknik ini untuk mempercepat suatu proses pembelajaran dalam *Neural Network* menggunakan beberapa lapisan. Seiring perkembangan zaman teknik ini berkembang menjadi beberapa jenis di antaranya *adalah Deep Auto Encoder, Deep Belief Nets*, *CNN* dan lainnya[17]. *Deep Learning* mampu mencapai hasil yang luar biasa dalam *computer vision.* Deep learning merupakan perkembangan dari teknik *machine learning* yang terinspirasi dari saraf manusia yang saling terhubung dengan menerapkan teknik syaraf buatan[18].

1. **Algoritma CNN**

Convolutional Neural Network merupakan salah satu algoritma yang banyak digunakan untuk membedakan suatu gambar, model ini adalah pengembangan yang sering disebut sebagai *MultiLayer Perceptron*(MLP) untuk mengolah data dua dimensi. Pada kasus citra/gambar, MLP kurang cocok dikarenakan kurang menghasilkan hasil yang baik. CNN pertama kali dikembangkan oleh seorang ilmuwan yang bernama Kunihiko Fukushima dengan nama NeoCognitron[19]. Arsitektur CNN memiliki dua tugas utama yang terdiri dari ekstraksi fitur dan klasifikasi(*Fully Connected Layer*)dan pada setiap layernya memiliki beberapa fungsi aktivasi.

1. **Convolutional Layer**

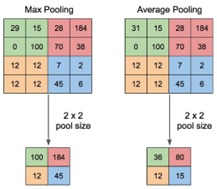
Merupakan proses utama yang terdapat pada arsitektur CNN, proses ini memfilter sebuah gambar dengan ketentuan seperti(tinggi, lebar) tertentu sehingga setiap lapisan akan menjadi gambar yang sudah termodifikasi dan beberapa filter dan menghasilkan activation map atau feature map 2D[19].



**Gambar 1.** Convolutional Layer

1. **Pooling Layer**

Merupakan lapisan yang berfungsi untuk mengurangi ukuran spasial dari fitur sebelumnya yaitu konvolusi, lapisan ini mencegah untuk terjadinya waktu komputasi yang lebih lama dengan cara mengurangi sumber data yang dibutuhkan untuk mempercepat komputasi karena parameter yang diperbaharui semakin sedikit, selain itu proses ini juga membantu agar pelatihan model lebih efektif[20]. Ada 2 jenis *pooling layer* yaitu *maxpooling dan average pooling*.



**Gambar 2.** Pooling Layer

1. **Fully Connected Layer**

Merupakan lapisan yang menghubungkan lapisan-lapisan sebelumnya. Lapisan ini mengolah data menjadi sesuatu untuk diklasifikasikan. Lapisan ini menghubungkan setiap neuron dari layer ke layer lainnya[21]. Setiap aktivitas yang terdapat pada masing-masing layer perlu diubah menjadi satu dimensi terlebih dahulu karena sebelumnya pada tahap *feature extraction* masih berbentuk *multidimensional array*, sehingga diperlukannya *flatten* atau penggabungan agar dapat dihubungkan di semua neuron di lapisan *Fully Connected*.

1. **Batch Normalization**

Merupakan salah satu teknik atau mekanisme yang digunakan untuk menstabilkan atau menormalisasikan distribusi pada lapisan tengah jaringan saraf dalam[22]. Teknik ini mampu mendistribusikan sebuah model untuk dapat mempelajari sebuah model dengan saraf tiruan secara cepat serta lebih stabil walaupun memiliki layer ekstra pada deep neural network.

1. **Keras**

Keras adalah library popular yang sering digunakan pada metode *deep learning* yang merupakan *wrapper* dari *TensorFlow*. Keras memudahkan seseorang dalam mengembangkan metode *deep learning* dikarenakan library ini lebih cepat dan unggul dalam mempelajari sistematika serta memiliki banyak fungsi yang dapat membantu seseorang dalam mempelajari teknik ini dengan lebih baik[23]. Library ini juga dapat digunakan dengan pengujian yang lebih cepat dengan penggunaan CPU dan GPU serta dapat mendukung penggunaan algoritma *Convolutional Neural Network*.

1. **TensorFlow**

Merupakan library yang disediakan oleh google secara gratis untuk pembelajaran mesin. Sistem pembelajaran mesin ini juga beroperasi pada skala besar. Tensorflow menggunakan grafik aliran data untuk mewakili komputasi dan memetakan aliran data di banyak mesin dalam sebuah cluster. Library ini juga berfungsi saat komputasi dan optimalisasi yang berguna dalam machine learning, library ini juga dapat digunakan untuk mengeksekusi kode di CPU atau GPU[24]. Tensorflow juga menawarkan API yang memfasilitasi pengembangan machine learning sehingga pengembang mampu secara konsisten untuk membuat model lebih baik. Dikarenakan framework ini opensource serta dapat dikembangkan membuat peneliti machine learning memiliki keuntungan performa yang lebih cepat dibandingkan framework lainnya.

1. **Numpy**

Numpy adalah library yang disediakan oleh python yang terfokus pada komputasi ilmiah seperti perhitungan *array* dan *object N-dimensional array*. Numpy hampir mirip dengan list python dan memiliki kelebihan di antaranya adalah penggunaan memori yang lebih kecil dan *run time* yang dilakukan relatif cepat. Dengan bantuan library ini banyak orang yang terbantu dalam penggunaan operasi pada vector(1-d array) dan matrix (2-d array).

1. **Matplotlib**

Merupakan library popular dengan pemanfaatan seperti dapat melakukan visualisasi terhadap data lebih menarik dan mudah dipahami. Library ini pada awalnya hanya terfokus pada penampilan grafik saja namun seiring berkembangnya waktu library ini mampu memvisualisasikan data lebih menarik karena data sudah berobject 2D bahkan 3D[25]. Library ini juga mampu memvisualisasikan gambar dengan kualitas yang dapat disimpan sesuai dengan kemauan pengguna dengan menyimpan JPEG dan PNG.

1. **Open CV**

Merupakan library opensource yang dikembangkan untuk mengolah suatu gambar atau video. Library ini juga gratis dan dapat di akses dimanapun dan kapanpun serta dapat diakses siapa saja. Library ini sering digunakan untuk mengolah suatu gambar dengan tujuan salah satunya adalah memperbaiki kualitas gambar dan mengidentifikasi suatu gambar.

1. **Pengujian Klasifikasi**

Pada tahap mengklasifikan suatu model yang telah dibangun dan menemukan model yang cocok, maka diperlukan adanya tahap pengujian untuk melihat sejauh mana performa dari model yang telah dibuat. Dalam penelitian kali ini pengujian menggunakan salah satu konsep yaitu confusion matrix. Berikut adalah tahapan confusion matrix :

**Tabel 1.** Pengujian Klasifikasi

|  |  |  |  |
| --- | --- | --- | --- |
|  | | NILAI SEBENARNYA | |
| TRUE | FALSE |
| NILAI PREDIKSI | TRUE | **TP** | **FP** |
| FALSE | **FN** | **TN** |

Dalam pengujian klasifikasi, terdapat beberapa rumus diantaranya adalah precission, accuracy dan recall.

1. *Precision* : Merupakan perbandingan dari nilai True Positive yang dideteksi sebagai banyaknya data yang diprediksi sebagai positif. Evaluasi tersebut juga melihat ketepatan jumlah prediksi yang berhasil diprediksi dengan baik.

*Precision* =

1. *Accuracy* : Merupakan rasio dari jumlah keseluruhan prediksi benar(positive dan negative) dengan keseluruhan data yang diklasifikasikan. Accuracy akan menentukan kedekatan nilai prediksi dengan nilai yang sebenarnya.

*Accuracy* =

1. *Recall* : Merupakan rasio prediksi benar positif dan dibandingkan dengan keseluruhan data yang benar positif. Recall mampu melihat seberapa benar ketepatan jumlah prediksi yang diperoleh dengan baik dari seluruh data.

*Recall* =

# BAB III

# METODOLOGI PENELITIAN

**Pengumpulan Dataset**

**Pembagian Data Set**

**Data Latih**

**Data Augmentasi**

**Implementation model CNN**

**Data Uji**

**Model**

**Evaluation Result**

**Identifikasi Masalah**

# Bagan 1. Flow Penelitian

1. **Identifikasi Masalah**

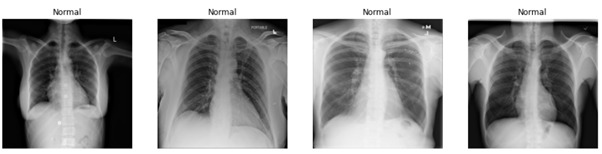
Permasalahan yang akan diselesaikan pada penelitian kali ini adalah membuat arsitektur CNN yang tepat dalam mengklasifikasikan penyakit Tuberkulosis yang terdapat pada paru-paru manusia, terdapat 2 label *Citra-Xray* yaitu Normal dan Tuberkulosis yang akan diuji hasil akurasi dari arsitektur yang tersebut. Untuk mengetahui permasalahan yang terjadi dalam penulisan tugas akhir ini, penulis melakukan studi literatur terhadap penelitian sebelumnya untuk menggali informasi.

1. **Pengumpulan DataSet**

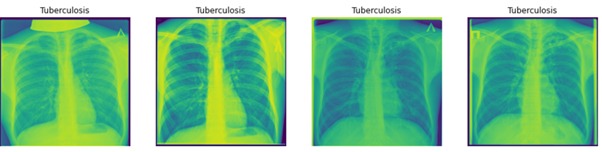
Untuk memenuhi kebutuhan data yang akan digunakan dalam penelitian kali ini maka akan diterapkan proses pengumpulan data sebagai berikut:

1. Studi Pustaka

Dataset yang digunakan dalam penelitian kali ini yaitu data citra Tuberkulosis yang terbentuk dari MRI(Magnetic Resonance Image) yang didalamnya teradapat 2 kelas yaitu Normal dan Tuberkulosis. Penulis mendapatkan data tersebut dari situs komunitas daring ilmuwan data dan praktisi pembelajaran mesin yaitu Kaggle dengan nama “Tuberculosis (TB) Chest X-ray Database”. Dataset ini memiliki jumlah data sebesar 4200 Citra X-ray dengan ukuran gambar seluruhnya yaitu 512 x 5125 x 1 pixel. Dapat dilihat pada gambar 1 adalah 4 gambar Citra-Xray Normal dan pada gambar 2 adalah 4 gambar Citra-Xray Tuberkulosis.



**Gambar 3.** Normal.



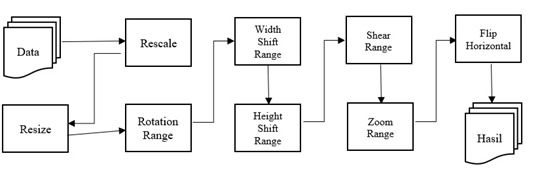
**Gambar 4.** Tuberculosis

1. **Pembagian Dataset**

Dengan jumlah dataset sebanyak 4200, dataset kemudian akan dibagi menjadi 3 bagian yaitu data latih sebesar 80% dengan total gambar latih sebesar 3360, data uji sebesar 10% dengan total gambar uji sebesar 420 dan data testing sebesar 10% dengan total gambar test sebesar 420.

1. **Data Augmentasi**

Dalam penelitian kali ini dilakukan beberapa teknik augmentasi yang dilakukan pada gambar x-ray. Teknik ini melihat dari arsitektur yang sama digunakan oleh peneliti pada paper [13] seperti melakukan rotasi pada gambar secara acak (*rotation range*), mengubah ukuran gambar (resize), mengubah rentang nilai raster(rescale), mengubah rentang pergeseran lebar (*width shift range*), mengubah rentang pergeseran panjang (*height shift range*), mengubah rentang pergeseran (*shear range*), mengubah rentang perbesaran (*zoom range*) dan flip horizontal. Seperti yang digambarkan pada gambar berikut :



**Bagan 2.** Data Augmentasi

1. **Implementation CNN Model**

Dalam implementasi metode menggunakan arsitektur CNN terdapat beberapa proses yang harus dilalui mulai dari penginputan data, melakukan pengaturan jenis fungsi aktivasi yang akan digunakan, metric loss dan jumlah output dari kelas yang diklasifikasikan serta mengatur jumlah epoch untuk melatih data. Pada tahap pembuatan arsitektur model pada penelitian ini melihat struktur dari paper rujukan karya [13], dan penelitian ini juga mengalami perubahan seperti output size yang berubah menjadi ukuran 100 x 100 serta akan ada beberapa percobaan hyperparameter yang akan digunakan pada arsitektur model kali ini. Berikut adalah tahapan implementasi model yang diusulkan :

**Tabel 2.** Modelling

|  |  |  |
| --- | --- | --- |
| **Layers** | **Output Size** | **Filter** |
| INPUT IMAGE | 100 x 100 x 3 |  |
| CONVO1 | 100 x 100 x 64 | 3 x 3 |
| ACTN | 100 x 100 x 64 |  |
| CONVO2 | 100 x 100 x 64 | 3 x 3 |
| ACTN | 100 x 100 x 64 |  |
| BATCHNORM | 100 x 100 x 64 |  |
| MAXPOOL | 50 x 50 x 64 | 2 x 2 |
| DROPOUT | 50 x 50 x 64 |  |
| CONVO3 | 50 x 50 x 128 | 3 x 3 |
| ACTN | 50 x 50 x 128 |  |
| CONVO4 | 50 x 50 x 128 | 3 x 3 |
| ACTN | 50 x 50 x 128 |  |
| BATCHNORM | 50 x 50 x 128 |  |
| MAXPOOL | 25 x 25 x 128 | 2 x 2 |
| DROPOUT | 25 x 25 x 128 |  |
| FULLY CONN | 512 |  |
| ACTN | 512 |  |
| BATCHNORM | 512 |  |
| DROPOUT | 512 |  |
| SOFTMAX | 2 |  |

Input layer berisi gambar yang telah di ubah sesuai dengan keinginan programmer untuk dapat diolah pada model yang diinginkan . Conv Layer merupakan proses utama dengan parameter yang dibentuk untuk menciptakan berbagai ukuran layer gambar yang akan di proses. Batch Norm digunakan untuk menormalisasikan layer yang akan digunakan untuk memfilter gambar atau daerah yang akan di tandai. Dropout layer digunakan untuk menghilangkan sementara hidden layer untuk mempercepat proses modelling. Fully Connected Layer adalah proses dimana menyatukan seluruh layer yang sudah dibangun sehingga membentuk klasifikasi.

1. **Evaluation Result**

Evaluasi model CNN dalam penelitian ini yaitu melihat skenario pengujian dengan cara membandingkan nilai akurasi dari model yang diusulkan pada penelitian ini dengan metode yang diusulkan oleh peneliti sebelumnya. Evaluasi tersebut juga melihat Plotting Accuracy ,Plotting Loss, Classification Report serta Confusion Matrix Test. Plotting Accuracy dan Plotting Loss akan menggunakan teknik plotting menggunakan library pyplot sehingga akan membentuk graph yang dapat dilihat serta dibandingkan oleh peneliti. Classification Report akan menggunakan teknik dari library sklearn dengan membandingkan model yang diprediksi terhadap kelas Test sehingga outputnya akan melihat prediksi(precision, f1-score, recall, accuracy) terhadap 2 hasil kelas dalam model ini(Tuberkulosis, Normal). Lalu Confusion Matrix menggunakan teknik sklearn matrix sehingga akan membentuk tabel yang diprediksikan sebagai nilai True Positive, False Positive, False Negative, True Negative. Dalam hasil evaluasi ini dapat terlihat sejauh mana model mampu mempelajari citra dengan baik melalui skenario yang telah dibangun. Data test yang digunakan dalam pengujian sebesar 10% dari keseluruhan data yang dimiliki. Berikut tabel *confussion matrix* :

**Tabel 3.** Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Nilai Sebenarnya | |
| TRUE | FALSE |
| NILAI PREDIKSI | TRUE | **TP** | **FP** |
| FALSE | **FN** | **TN** |

Berikut adalah keterangan tabel diatas :

1. TP = Jumlah sampel yang bernilai true positif yang diprediksi sebagai benar.
2. TN = Jumlah sampel yang bernilai false yang salah diprediksi secara benar.
3. FP = Jumlah sampel yang bernilai false yang salah di prediksi sebagai sampel bernilai true.
4. FN = Jumlah sampel yang bernilai true yang salah di prediksi sebagai sampel bernilai true.

# BAB IV

# HASIL DAN PEMBAHASAN

1. **Implementasi Skenario Model 1**

Pada tahap penelitian menggunakan jupyter notebook dan Bahasa pemrograman yang digunakan adalah python. Pada skenario model 1 ini peneliti mencoba membangun aristektur model dengan augmentasi data tanpa melakukan *hyperparameter tunning*. Sebelum masuk kedalam pemodelan, terdapat beberapa proses yang harus dilakukan sesuai dengan ketentuan peneliti.

1. **Load Data Set**

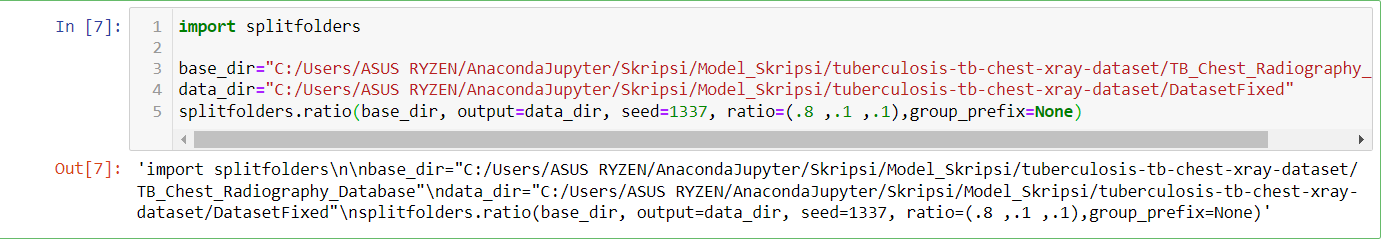
Pada tahap ini peneliti perlu memastikan file yang akan masuk ke dalam proses pemodelan adalah file dalam format png serta peneliti dapat mengetahui jumlah data yang dimiliki.



**Gambar 5.** Load Data Model 1

1. **Splitting Data Set**

Sebelum peneliti masuk kedalam tahap *preprocessing*, maka diperlukan *spiltting dataset*. Langkah ini diperlukan untuk memecah beberapa folder yang terdiri dari data *train*(80%), data *validation*(10%), data *test*(10%). Di masing-masing folder tersebut akan tersedia folder dengan label data Normal dan Tuberculosis.



**Gambar 6.** Split Data Model 1

1. **Preprocessing**

Peneliti mendefinisikan ukuran gambar *height*(100) dan *weight*(100) dengan ukuran *batch\_size*(32) sebelum model diproses. Pada tahap ini juga dilakukan proses *augmentasi* data dengan tujuan agar data yang akan masuk ke dalam proses selanjutnya dapat meningkatkan performa model.



**Gambar 7.** Augmentasi Data Model 1

1. **Perancangan Model CNN**

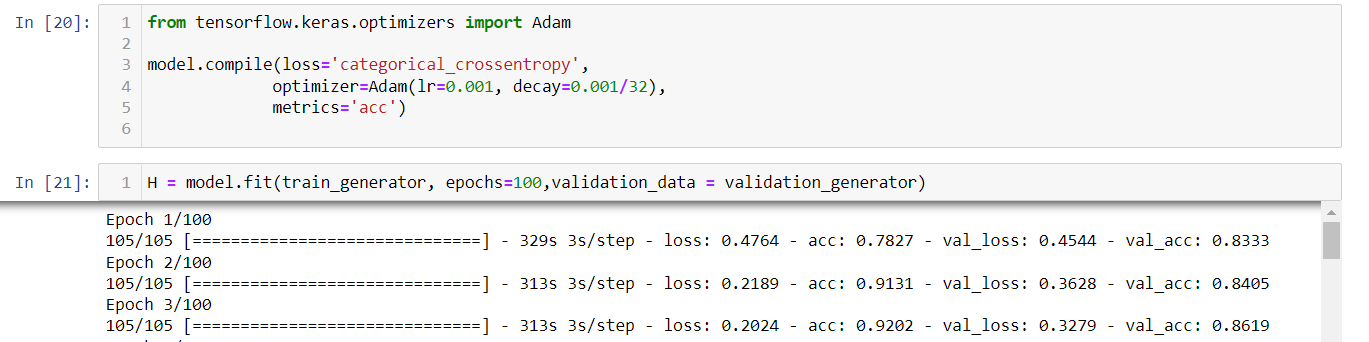
Pada tahap perancangan model menggunakan *input* gambar sebesar (100,100) dengan fungsi aktivasi di tiap layernya adalah ‘Relu’ untuk mendeteksi gambar yang berlabel Normal dan Tuberculosis.



**Gambar 8.** Perancangan Model 1

1. **Pelatihan Model CNN**

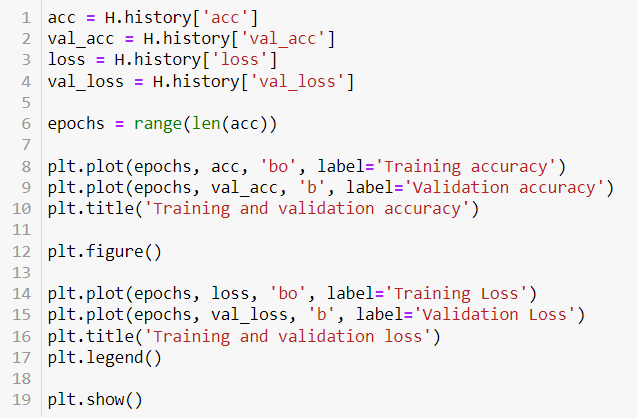
Pada tahap ini peneliti menggunakan fungsi *Optimizers Adam* dengan lr sebesar 0.001 dan decay sebesar 0.001/32. Parameter tersebut digunakan dengan tujuan agar model dapat di pelajari mendetail agar hasil yang diharapkan lebih baik.



**Gambar 9.** Pelatihan Model 1

1. **Visualiasasi Akurasi dan Loss Model CNN**

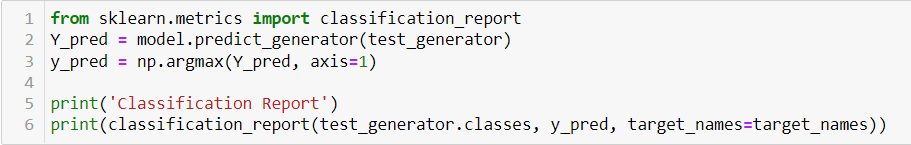
Pada tahap ini adalah hasil untuk melihat sejauh mana model berkembang dan dipelajari melalui grafik yang ada. Performa suatu model juga dapat ditinjau berdasarkan bagaimana model tersebut dapat dilihat, apakah kondisi model yang sudah di pelajari mengalami *overfitting* atau tidak.

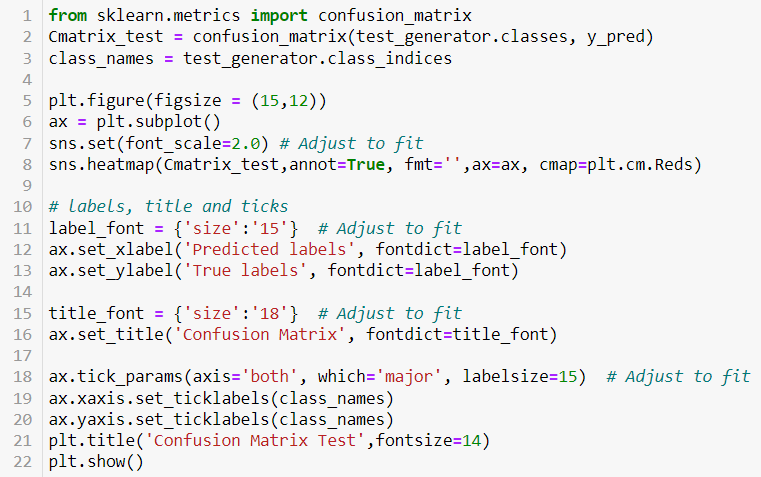


**Gambar 10.** Visualisasi dan Loss Model 1

1. **Pengujian Model CNN**

Pada tahap pengujian, peneliti menggunakan metode *classification report* dan *confussion matrix*. Pengujian model memudahkan seorang peneliti untuk dapat menganalisis model yang sudah dibuat serta dapat melihat sejauh mana performa dari model yang sudah dibangun.





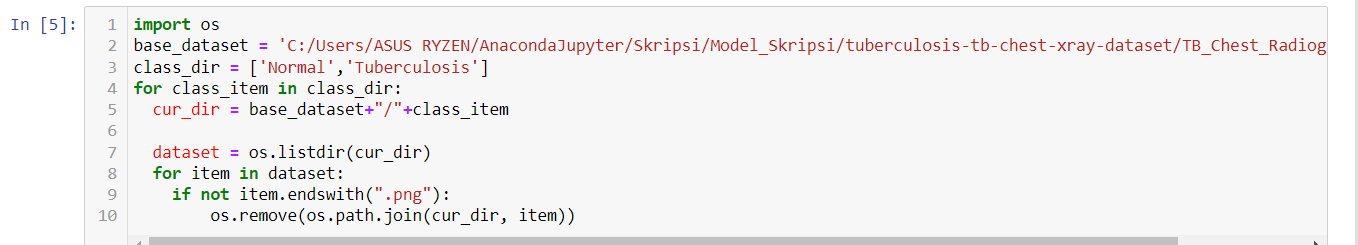
**Gambar 11.** Pengujian Model 1

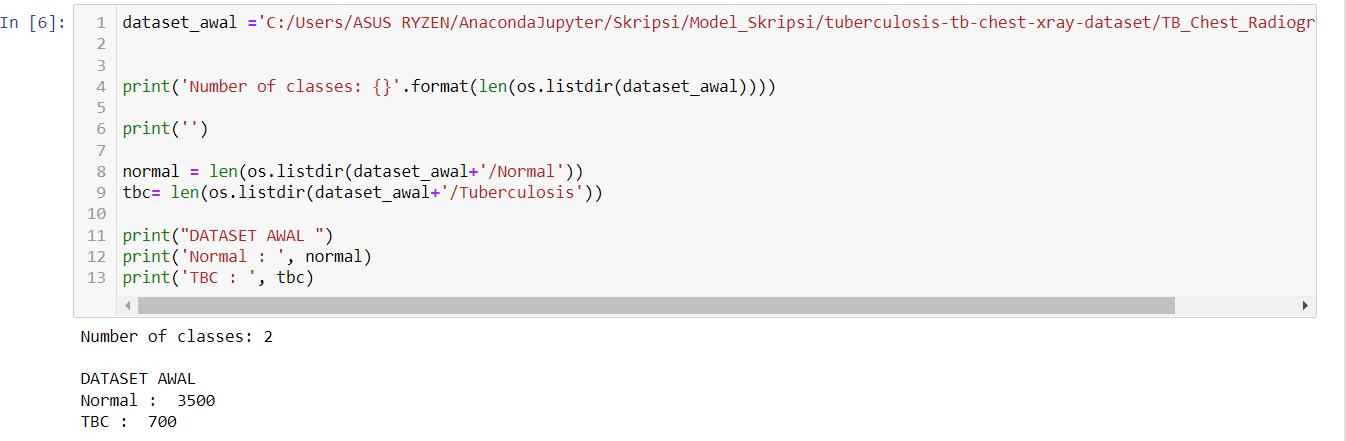
1. **Implementasi Skenario Model 2**

Pada tahap penelitian ini menggunakan jupyter notebook dengan menggunakan Bahasa python. Tahap penelitian skenario model 2 dimulai dari pengumpulan data hingga mengimplementasikan algoritma CNN. Data tersebut juga akan melewati proses yaitu split data sesuai dengan ketentuan peneliti.

1. **Load Data Set**

Pada tahap ini peneliti ingin memastikan file atau dataset yang tersimpan memiliki format yang sama yakni png dan peneliti ingin memastikan jumlah data set yang dimiliki.

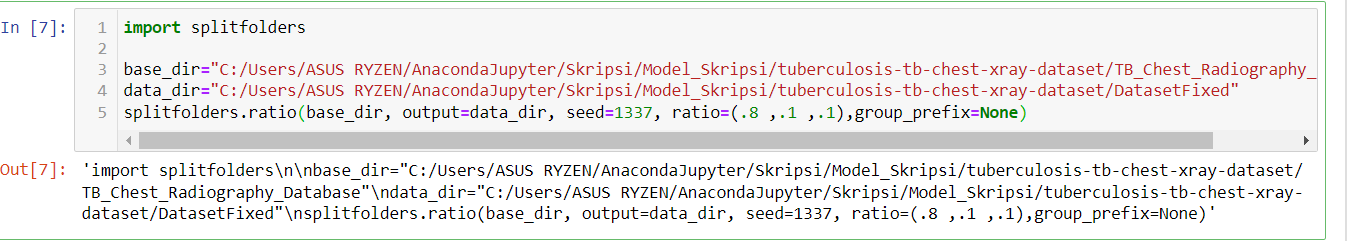




**Gambar 12.** Load Data Model 2

1. **Splitting Dataset**

Peneliti mendefinisikan untuk masing-masing folder seperti data *train* (80%), data *validation* (10%), data *testing* (10%). Data yang telah dipecah tersebut memiliki sub folder khusus yaitu data Normal dan Tuberkulosis. *Splitting dataset* memudahkan peneliti agar dalam penelitian dapat memaksimalkan performa model.



**Gambar 13.** Split Data Model 2

1. **Preprocessing**

Peneliti mendefinisikan gambar dengan ukuran *height*(100), *weight*(100), *batch\_size*(32) sebelum masuk ketahap pelatihan model. Peneliti memanggil path yang sudah di pisahkan dan akan masuk kedalam tahap augmentasi, Tahap augmentasi berguna agar mesin dapat mengenali suatu gambar lebih baik lagi, augmentasi data juga dapat meningkatkan performa dari model.



**Gambar 14.** Augmentasi Data Model 2

1. **Perancangan Model CNN**

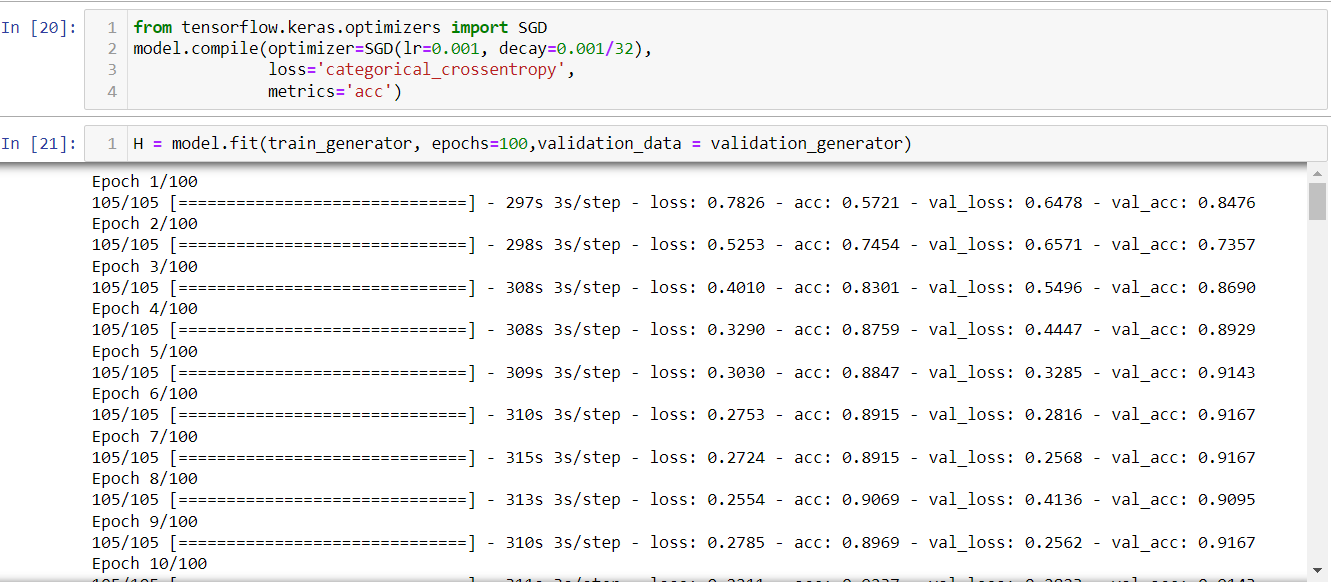
Pada tahap ini peneliti merancang model dengan skema model CNN untuk mendeteksi gambar dengan label Normal dan Tuberculosis.   
Saat mengimplementasikannya, peneliti membuat skenario model kedua dengan penggunaan hyperparameter tuning yaitu fungsi aktivasi yang digunakan setiap layernya adalah *relu.*



**Gambar 15.** Perancangan Model 2

1. **Pelatihan Model CNN**

Pada tahap pelatihan model, peneliti menggunakan salah satu parameter *hyperparameter tuning* untuk dapat meningkatkan hasil akurasi yang lebih baik serta dapat menekan kondisi overfitting. Parameter yang digunakan adalah optimizers SGD yang disediakan oleh keras. Tahap penelitian memakai *lr* 0.001, decay 0.001/32, epoch 100.



**Gambar 16.** Pelatihan Model 2

1. **Visualisasi Akurasi & Loss Model CNN**

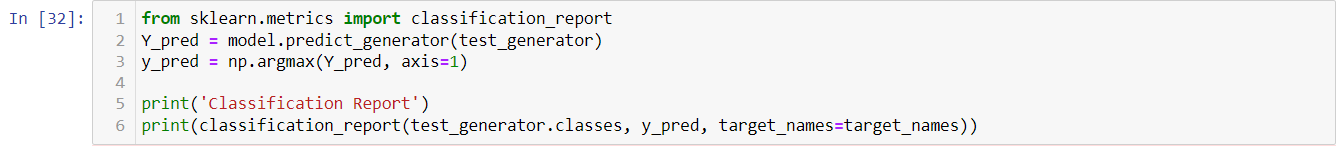
Pada tahap penelitian ini, model yang telah dipelajari akan dilihat sejauh mana model dapat menghasilkan nilai akurasi lebih baik. Performa model sangat bermanfaat semisal dapat melihat apakah kondisi model yang telah dipelajari mengalami *overfitting/underfitting* terhadap data train dan validasi.



**Gambar 17.** Visualisasi dan Loss Model

1. **Pengujian Model CNN**

Pada tahap pengujian, peneliti menggunakan metode *classification report* dan *confussion matrix*. Hal ini dapat memudahkan peneliti dan pembaca dalam melihat performa yang dihasilkan model. Pengujian ini menggunakan data test sebanyak 420 data.



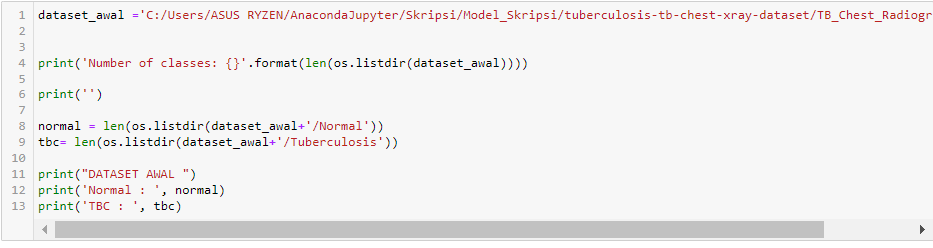
**Gambar 18.** Pengujian Model 2

1. **Implementasi Skenario Model 3**

Pada tahap penelitian ke tiga kali peneliti menggunakan pemodelan CNN dasar dengan ketentuan *penggunaan Gathering Data, Normalization, Encoder Labelling* dan masuk ke dalam tahap pemodelan. Terdapat beberapa proses yang harus dilalui agar model dapat menghasilkan akurasi yang lebih baik.

1. **Load Data Set**

Pada tahap ini peneliti memastikan jumlah data yang masuk sehingga dapat ditentukan proses selanjutnya.



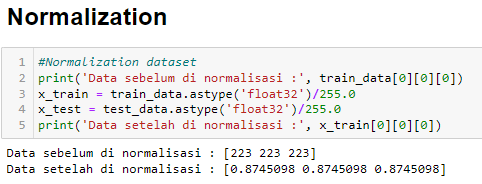
**Gambar 19.** Load Data Model 3

1. **Preprocessing**

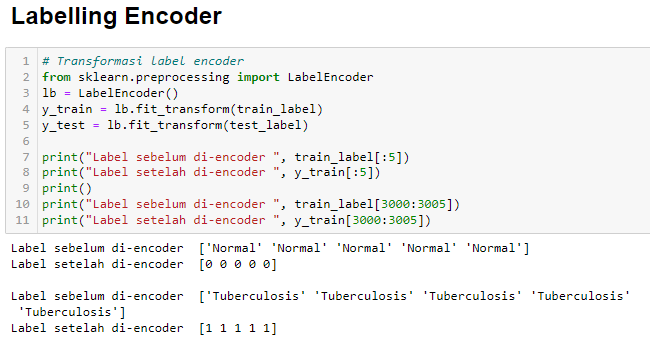
Pada tahapan ini peneliti tidak menggunakan Image Data Generator dalam proses *preprocessing*, melainkan menggunakan beberapa tahap yaitu *gathering data* terhadap data *train* dan *test* lalu masuk ke tahap *normalization* dan diakhiri dengan *labelling encoder*.



**Gambar 20.** Preprocessing Gathering Data Model 3



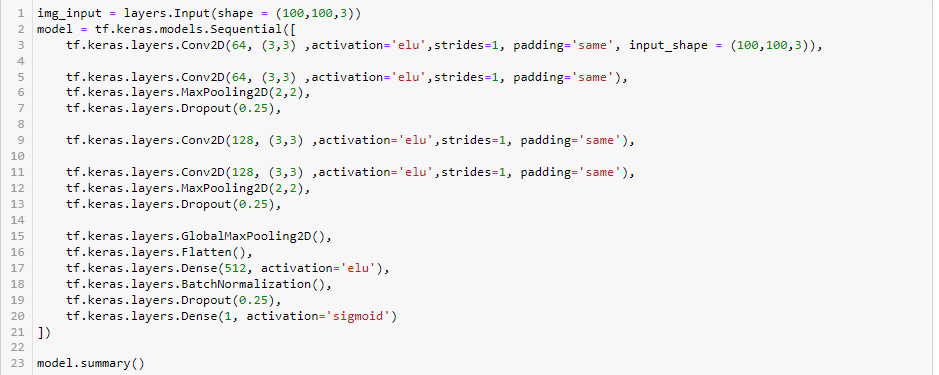
**Gambar 21.** Preprocessing Normalization Model 3



**Gambar 22.** Preprocessing Labelling Encoder Model 3

1. **Perancangan Model CNN**

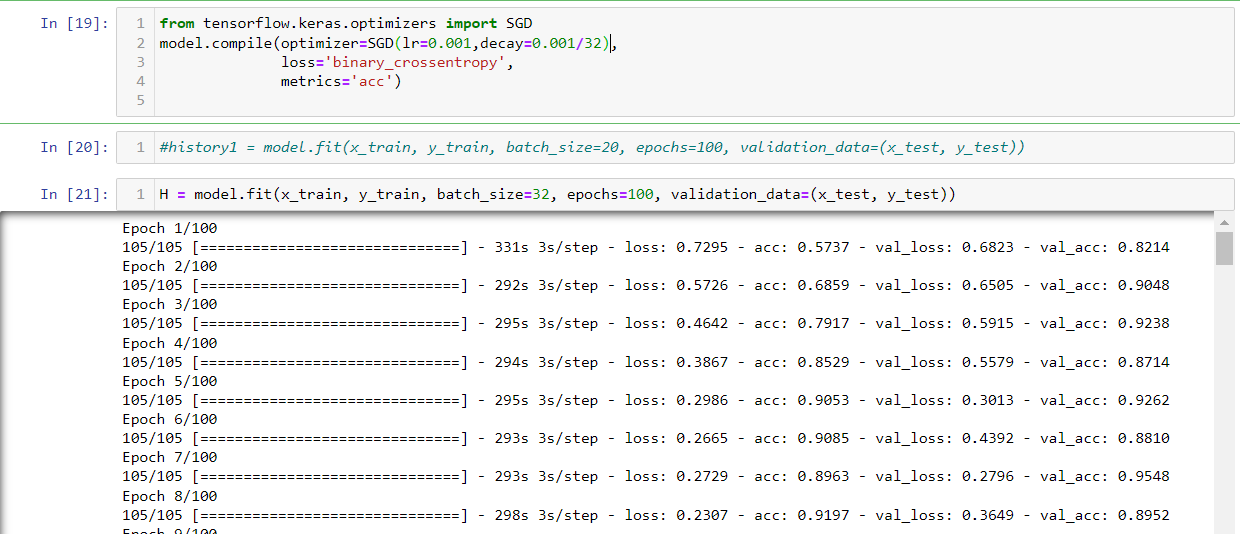
Pada tahap perancangan model yang akan dibangun kali ini peneliti mendefisinikan preprocessing sesuai ketentuan sebelum masuk ke tahap perancangan. Dalam tahap perancangan model menggunakan layer model yang sama dengan model sebelumnya dan tentunya juga mendefinisikan fungsi aktivasi di tiap layernya dengan “relu”.



**Gambar 23.** Perancangan Model 3

1. **Pelatihan Model CNN**

Pada tahap pelatihan model, peneliti menggunakan salah satu parameter *hyperparameter tuning* seperti model sebelumnya. Parameter yang digunakan adalah *optimizers SGD* yang disediakan oleh keras. Tahap penelitian memakai lr 0.001, epoch 100, karena data yang sudah diolah menjadi data *binary* maka pada fungsi *loss* menggunakan ‘*binary\_crossentropy’*.



**Gambar 24.** Pelatihan Model 3

1. **Visualisasi Akurasi & Loss Model CNN**

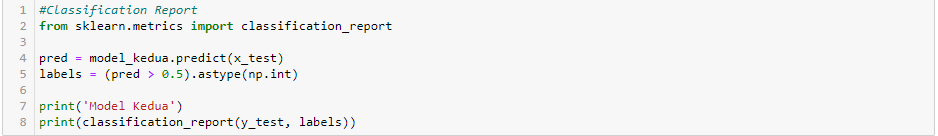
Pada tahap visualisasi ini , peneliti melihat visualisasi sejauh mana model ini dapat berjalan dan apakah performa model yang telah dipelajari mengalami *overfitting/underfitting* terhadap data *train* dan *test* dalam pemodelan.

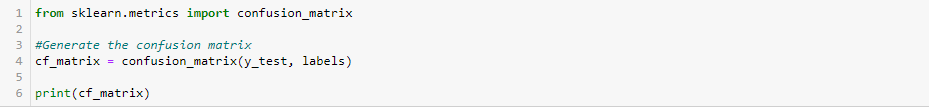


**Gambar 25.** Visualisasi dan Loss Model 3

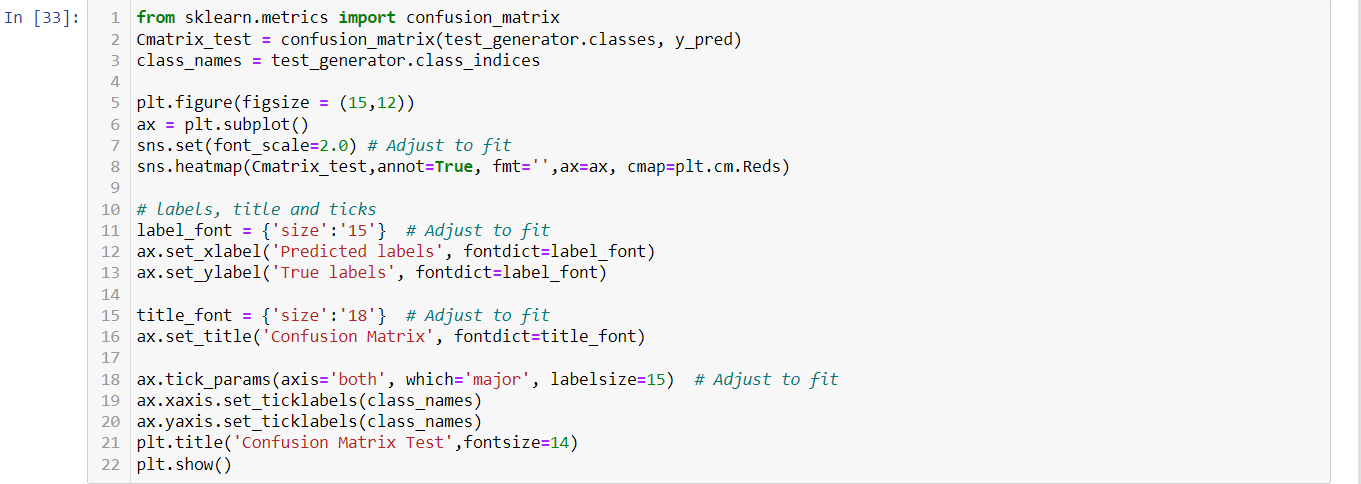
1. **Pengujian Model CNN**

Tahap pengujian menggunakan metode *classification report* dengan tujuan melihat sejauh mana performa model yang di peroleh berdasarkan data x\_test dan y\_test serta menggunakan *confusion\_matrix* sebagai pengujian data yang diprediksi benar.





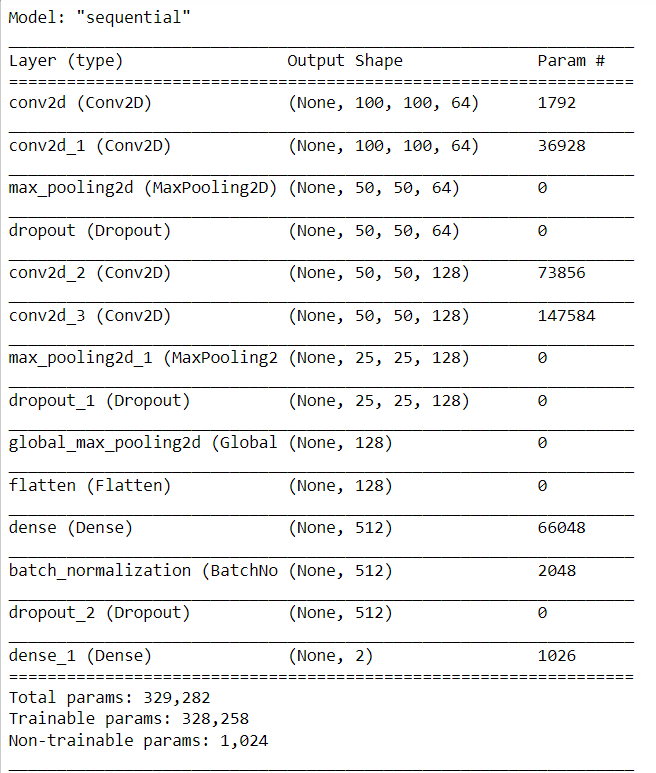




**Gambar 26.** Pengujian Model 3

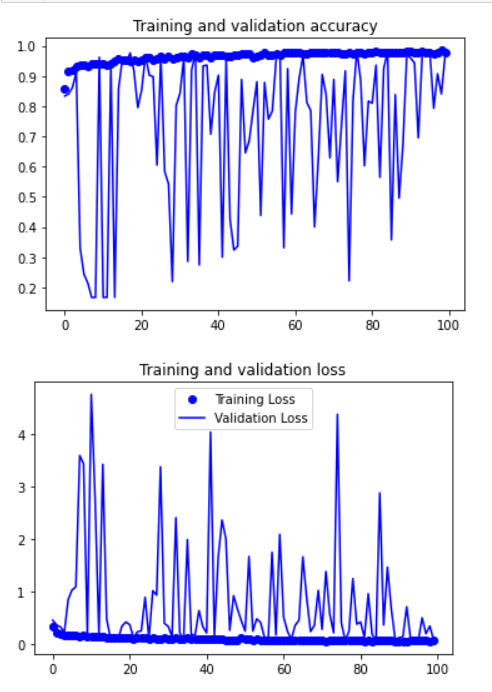
1. **Hasil Skenario Model 1**

Pada tahap skenario model pengujian pertama menggunakan proses augmentasi dan tidak menggunakan hyper parameter tunning. Saat model dapat dipelajari selanjutnya peneliti melakukan Analisa terhadap model seperti akurasi, presisi, recall confusion matrix. Pengujian model pertama memiliki jumlah parameter 329,282 yang terdiri dari 4 convolutional layer. Pada layer pertama menggunakan kernel 64 dengan fungsi aktivasi Relu. Layer kedua menggunakan kernel 64 dengan fungsi aktivasi Relu serta Maxpooling2D dan dropout 25%. Layer ketiga menggunakan kernel berjumlah 128 dengan aktivasi Relu. Layer ke empat menggunakan kernel berjumlah 128 dengan fungsi aktivasi Relu serta Maxpooling2D dan dropout 25%. Memiliki 1 Fully Connected Layer sebesar 512 dengan fungsi aktivasi Relu, BatchNormalization, dropout sebesar 25%, output layer mengggunakan fungsi aktivasi softmax.



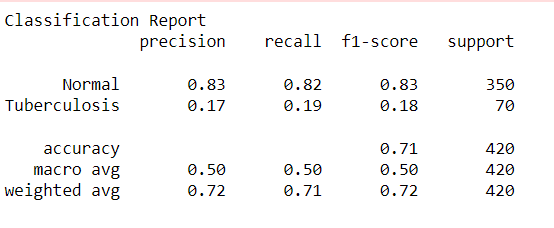
**Gambar 27.** Model 1 Parameter

Grafik akurasi dan loss pada model skenario 1 dapat dilihat pada gambar dibawah ini. Akurasi train dan validation sejak pertama kali di latih hingga epoch 100 mengalami kondisi overfitting.



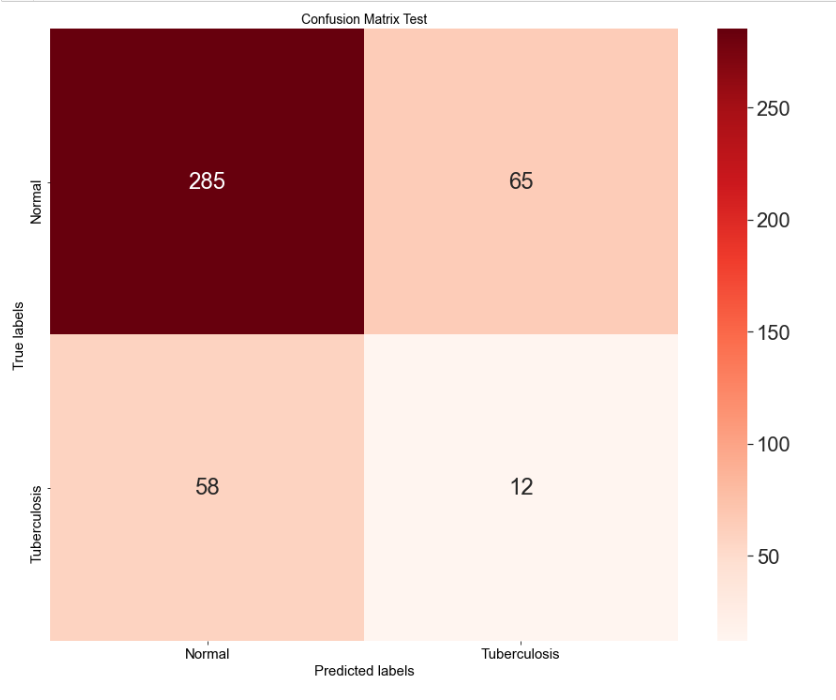
**Gambar 28.** Graph Acc Loss Model 1

Pada gambar selanjutnya adalah hasil pengujian classification report pada skenario model 1, Classification report berguna agar peneliti mampu melihat sejauh mana model berjalan dan standarisasi berdasarkan precision recall, f1-score dalam pendeteksian label Normal dan Tuberculosis.



**Gambar 29.**Classification Report Model 1

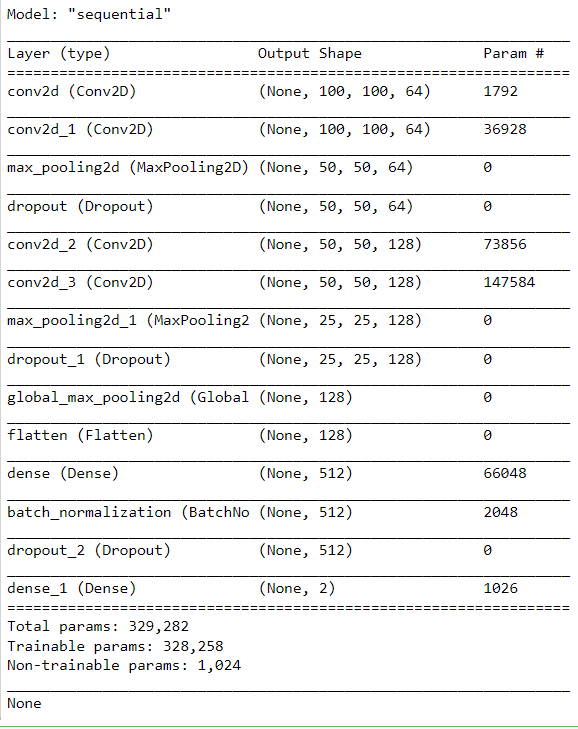
Pada gambar dibawah adalah hasil pengujian pada skenario model 1 dengan jumlah data test sebesar 420 dan disajikan dalam bentuk confussion matrix.



**Gambar 30.** Confussion Matrix Model 1

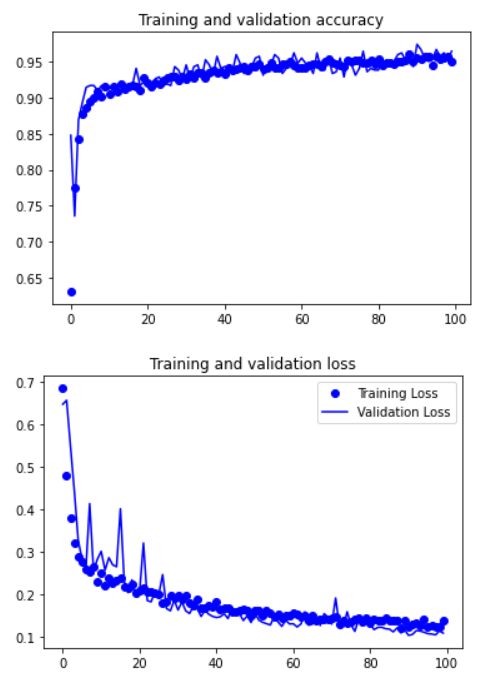
1. **Hasil Skenario Model 2**

Pada tahap ini adalah skenario pengujian dengan model CNN yang menggunakan proses augmentasi serta menggunakan hyper parameter tunning. Dalam skenario model 2 ini melakukan Analisa terhadap model seperti akurasi, presisi, recall serta confussion matrix. Pada skenario model pertama mempunyai 329,282 parameter yang terdiri dari 4 convolutional layers dengan jumlah kernel 64(Conv-1 dan Conv-2), 128(Conv-3 dan Conv-4) dan ukuran kernel yang digunakan adalah 3 x 3, 2 max-pooling layer dengan ukuran 2 x 2 kemudian diikuti 1 fully connected layers yaitu FC-1 sebesar 512, pada output layer menggunakan fungsi aktivasi softmax. Fungsi aktivasi (Elu) sebagai hyper parameter tunning digunakan di semua hidden layer kecuali output layer. Dropout layer ditambahkan pada layer 2 dan 4 setelah max-pooling layers sebesar 25% dan fully connected layers untuk dropout yang digunakan sebesar 25%.



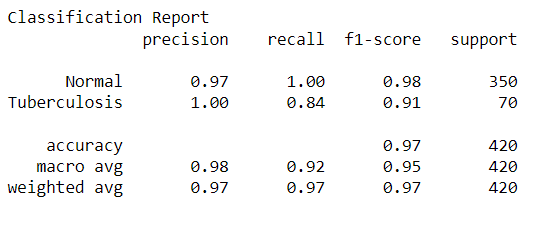
**Gambar 31.** Model 2 Parameter

Grafik akurasi dan loss pada model skenario 2 dapat dilihat pada gambar dibawah ini. Akurasi train dan validation meningkat pada tahap 20 epoch pertama. Pada grafik hasil akurasi tidak terjadi overfitting maupun underfitting.



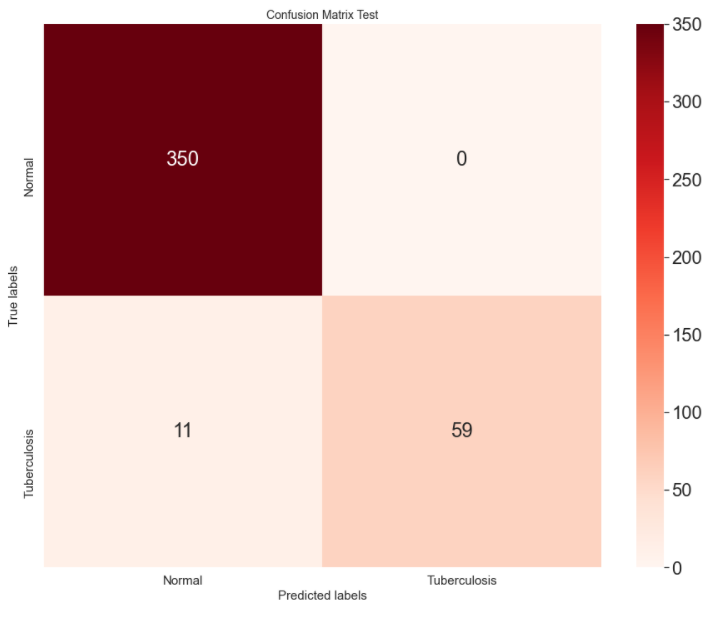
**Gambar 32.** Graph Acc Loss Model 2

Pada gambar selanjutnya adalah hasil pengujian classification report pada skenario model 2, hal ini diperlukan untuk melihat standarisasi sejauh mana model ini dilihat berdasarkan precission, recall, f1-score dalam pendeteksian label Normal dan Tuberculosis.



**Gambar 33.** Classification Report Model 2

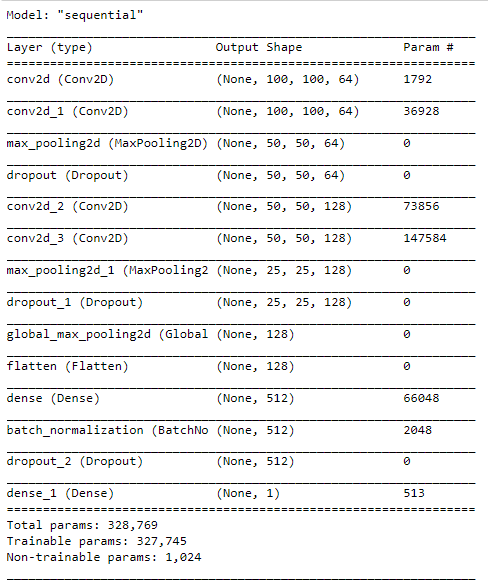
Pada gambar dibawah adalah hasil pengujian pada skenario model 2 dengan jumlah data test sebesar 420 dan disajikan dalam bentuk confussion matrix.



**Gambar 34.** Confussion Matrix Model 2

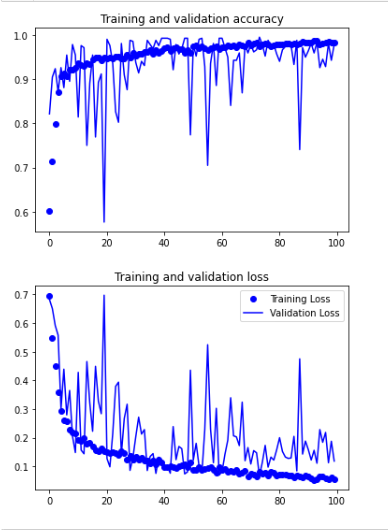
1. **Hasil Skenario Model 3**

Pada tahap skenario pengujian model ke 3 tidak menggunakan proses augmentasi namun menggunakan metode penggunaan data *binary processing* terdiri dari gathering data dengan ketentuan data train dan data test mengalami *resize*(100,100) dan data train menjadi ukuran (3360, 100,100,3) , Train label (3360), Test data(420,100,100,3), Test label(420) data. Selanjutnya masuk ke tahap Normalization dan Labelling Encoder. Pada skenario model ini tetap menggunakan metode *Hyperparamter Tunning* dengan ketentuan yang sama yakni penggunaan fungsi aktivasi ‘elu’ dengan total parameter 328,769 terdiri dari 4 layer yang sama dengan skenario model ke 2 dengan diakhiri oleh 1 Fully Connected Layer dengan fungsi aktivasi output layernya adalah ‘Sigmoid’.



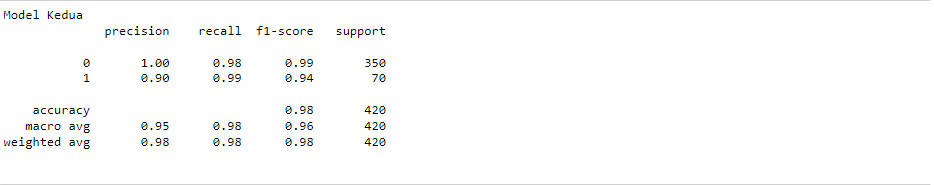
**Gambar 35.** Model 3 Parameter

Grafik akurasi pada skenario pemodelan 3 dapat dilihat seperti gambar dibawah ini. Selama proses berjalan *Training Accuracy* serta *Validation Accuracy* epoch 0 hingga 100 mengalami kenaikan. Namun saat model berjalan data *Training* serta data *Validation* tidak dapat berjalan dengan baik sehingga model yang sedang berjalan tidak stabil atau yang sering disebut sebagai *data bouncing*. Grafik ini juga terlihat pada data Training dan Validation loss juga menunjukkan bahwa grafik dari pemodelan yang telah dijalankan mengalami kondisi tidak stabil atau overfitting.



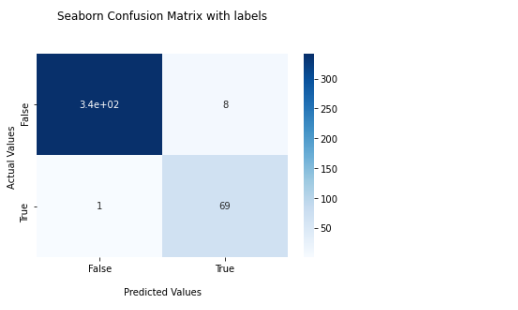
**Gambar 36.**Graph Acc Loss Model 3

Pada tahap selanjutnya peneliti melihat hasil pengujian classification report pada skenario model 2. Tujuannya adalah melihat sejauh mana model berkembang dan standarisasi yang dibutuhkan seperti melihat hasil akurasi, precission, recall, f-score pada data yang berlabel Normal dan Tuberculosis. Hasil classification report dapat dilihat pada gambar dibawah, terdapat label binary 0 ‘Normal’ dan label binary 1 ‘Tuberculosis’. Data tersebut menunjukkan akurasi pada model skenario 3 mendapatkan hasil 98%.



**Gambar 37.** Classification Report Model 3

Pada gambar dibawah adalah hasil pengujian pada skenario model 3 dengan jumlah data test sebesar 420 dan disajikan dalam bentuk confussion matrix.



**Gambar 38.** Confussion Matrix Model 3

1. **Perbandingan Performa Model CNN**

Peneliti meneliti 3 model yang dibangun dengan arsitektur yang berbeda, peniliti menginginkan model yang dibangun dapat menghasilkan akurasi yang lebih baik serta penggunaan data yang efisien dalam mendeteksi penyakit Tuberkulosis. Dari ketiga model yang sudah diteliti dapat dilihat bahwa model ke 2 adalah model terbaik karena hyperparameter tunning serta penggunaan data augmentasi dapat menekan jumlah overfitting.

**Tabel 4.** Performa Model

|  |  |
| --- | --- |
| ***Factor(s)*** | ***Values*** |
| Jumlah Convolutional + elu layers | 1,2,3,4 |
| Kernel Convolutional layer | 64, 128 |
| Jumlah dropout layers | 2,4 |
| Epoch | 100 |
| Pooling layers | Max Pooling |
| Batch\_size | 32 |
| Dropout rate | 0.25 |
| Optimizers | SGD |

Pada tabel diatas merupakan model yang digunakan pada jurnal sebelumnya yang menjadi rujukan utama pemodelan dan peneliti juga menambah beberapa arsitektur seperti penggantian fungsi aktivasi pemodelan yang dibangun. Seperti penggantian fungsi ‘elu’ di tiap layernya(Conv1-4) dengan ukuran kernel\_size di tiap layernya(3,3), filters 64 untuk (Conv 1-2), filters 128 untuk (Conv 3-4), penghapusan batchnorm di tiap layernya dan diganti batchnorm pada Fully Connected Layer, Maxpooling pada layer(Conv2 dan Conv4). Teknik dropout terdapat pada hidden layer(Conv2 dan Conv4) serta teknik dropout pada Fully Connected layer sebesar 25%.

Pada tabel dibawah ini adalah perbandingan performa model 1 hingga 3 menggunakan metode confussion matrix yang sudah didefinisikan di tiap modelnya dan tentunya sudah dihitung secara sistematis terhadap model yang telah dibangun.

**Tabel 5.** Hasil Performa Model

|  |  |  |  |
| --- | --- | --- | --- |
| **Performa (%)** | **Model Klasifikasi** | | |
| **1** | **2** | **3** |
| Accurcy | 69% | 97% | 97% |
| Precission | 80% | 100% | 97% |
| Recall | 95% | 96% | 99% |
| F1-Score | 86% | 97% | 97% |

Dari tabel diatas secara perhitungan menggunakan accuracy () , *precission* () , *recall* () , *f1-score* () dapat disimpulkan bahwa perancangan model ke 2 dan ke 3 adalah model yang terbaik. Model ke 2 menggunakan *hyperparameter tunning* dan *image data generator* sesuai dengan keinginan peneliti dan dari hasil *plotting* grafik menunjukkan mampu menekan *overfitting*. Sedangkan pada model ke 3 tidak menggunakan *image data generator* dan hanya menggunakan *hyperparameter tunning* yang sama dengan model sebelumnya menghasilkan performa akurasi yang juga hampir mendekati model ke 2 namun hanya saja saat model sedang berjalan terjadi overfitting antara data *train* dan data *test* sehingga walaupun akurasi yang ditunjukkan mampu menyamai model ke 2 namun tidak dapat dikatakan sebagai pemodelan yang sempurna.

# BAB V

# PENUTUP

1. **Kesimpulan**

Berdasarkan hasil pengujian, metode yang diusulkan oleh peneliti pada tahap penelitian ini lebih baik dibandingkan penelitian sebelumnya dikarenakan arsitektur yang di rancang sudah dalam tahap pemrosesan sedemikian rupa dan tentunya dengan penggunaan *image data generator* atau *data augmentasi* serta penggunaan *hyperparameter tunning* mampu menghasilkan performa model yang lebih baik. Dalam penyununan arsitekturpun peneliti juga membandingkan model yang dibangun dengan ketentuan peneliti dengan model yang menggunakan arsitektur pemodelan tanpa image data generator serta hyperparameter tunning.

1. **Saran**

Penulis berharap ada perbaikan pada penelitian ini dan pada penelitian selanjutnya melakukan pengembangan menggunakan variasi layer, *augmentasi data* yang berbeda, serta mencoba perkembangan model menggunakan data set yang lebih banyak jumlahnya daripada *dataset* yang digunakan kali ini.

# REFERENSI

[1] W. Supartini and H. Hindarto, “Sistem Pakar Berbasis Web Dengan Metode Forward Chaining Dalam Mendiagnosa Dini Penyakit Tuberkulosis Di Jawa Timur,” *Kinetik*, vol. 1, no. 3, p. 147, 2016, doi: 10.22219/kinetik.v1i3.123.

[2] O. Rochmawanti, F. Utaminingrum, and F. A. Bachtiar, “Analisis Performa Pre-Trained Model Convolutional Neural Network dalam Mendeteksi Penyakit Tuberkulosis,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 4, p. 805, 2021, doi: 10.25126/jtiik.2021844441.

[3] C. Liu *et al.*, “TX-CNN: Detecting tuberculosis in chest X-ray images using convolutional neural network,” *Proc. - Int. Conf. Image Process. ICIP*, vol. 2017-Septe, pp. 2314–2318, 2018, doi: 10.1109/ICIP.2017.8296695.

[4] N. W. Septarini, “Metode Pengendalian Penyakit Menular,” *Fak. Kedokteran, Univ. Udayana*, pp. 121–125, 2017.

[5] B. P. Putra, Y. Yunus, and Sumijan, “Sistem Pakar dalam Mendiagnosis Penyakit Mata dengan Menggunakan Metode Forward Chaining,” *J. Inf. dan Teknol.*, vol. 3, pp. 128–133, 2021, doi: 10.37034/jidt.v3i3.122.

[6] A. Ahyuna and K. Aryasa, “Sistem Pakar Diagnosa Dan Tatalaksana Penyakit Demam Berdarah Dengue Menggunakan Jaringan Saraf Tiruan Backpropagation,” *E-JURNAL JUSITI J. Sist. Inf. dan Teknol. Inf.*, vol. 6, no. 1, pp. 1–11, 2017, [Online]. Available: http://ejurnal.dipanegara.ac.id/index.php/jusiti/article/view/40.

[7] R. Rahmadewi and R. Kurnia, “Klasifikasi Penyakit Paru Berdasarkan Citra Rontgen dengan Metoda Segmentasi Sobel,” *J. Nas. Tek. Elektro*, vol. 5, no. 1, p. 7, 2016, doi: 10.25077/jnte.v5n1.174.2016.

[8] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “ChestX-ray: Hospital-Scale Chest X-ray Database and Benchmarks on Weakly Supervised Classification and Localization of Common Thorax Diseases,” *Adv. Comput. Vis. Pattern Recognit.*, pp. 369–392, 2019, doi: 10.1007/978-3-030-13969-8\_18.

[9] D. Litchfield, L. J. Ball, T. Donovan, D. J. Manning, and T. Crawford, “Viewing another person’s eye movements improves identification of pulmonary nodules in chest x-ray inspection,” *J. Exp. Psychol. Appl.*, vol. 16, no. 3, pp. 251–262, 2010, doi: 10.1037/a0020082.

[10] N. Suhermi, S. Suhartono, I. M. G. M. Dana, and D. D. Prastyo, “Pemilihan Arsitektur Terbaik pada Model Deep Learning Melalui Pendekatan Desain Eksperimen untuk Peramalan Deret Waktu Nonlinier,” *Stat. J. Theor. Stat. Its Appl.*, vol. 18, no. 2, pp. 153–159, 2019, doi: 10.29313/jstat.v18i2.4545.

[11] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach, “Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, 2019, doi: 10.1038/s41598-019-42294-8.

[12] H. Liu, L. Wang, Y. Nan, F. Jin, Q. Wang, and J. Pu, “SDFN: Segmentation-based deep fusion network for thoracic disease classification in chest X-ray images,” *Comput. Med. Imaging Graph.*, vol. 75, pp. 66–73, 2019, doi: 10.1016/j.compmedimag.2019.05.005.

[13] G. Kim and C. Lee, *Convolutional Neural Network 를 이용한 한국어 영화평 감성 분석 Using Convolutional Neural Network*, vol. 2644. Springer International Publishing, 2016.

[14] D. A. V. Dyk and X. L. Meng, “The art of data augmentation,” *J. Comput. Graph. Stat.*, vol. 10, no. 1, pp. 1–50, 2001, doi: 10.1198/10618600152418584.

[15] J. Sanjaya and M. Ayub, “Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup,” *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

[16] B. Liu, X. Wang, R. Kwitt, and N. Vasconcelos, “Feature Space Transfer for Data Augmentation Mandar Dixit Microsoft,” *Cvpr*, pp. 9090–9098, 2018, [Online]. Available: http://openaccess.thecvf.com/content\_cvpr\_2018/papers/Liu\_Feature\_Space\_Transfer\_CVPR\_2018\_paper.pdf.

[17] A. Ahmad Hania, “Mengenal Artificial Intelligence, Machine Learning, & Deep Learning,” *J. Teknol. Indones.*, no. June, 2017, [Online]. Available: https://amt-it.com/mengenal-perbedaan-artificial-inteligence-machine-learning-deep-learning/.

[18] A. Santoso and G. Ariyanto, “Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah,” *Emit. J. Tek. Elektro*, vol. 18, no. 01, pp. 15–21, 2018, doi: 10.23917/emitor.v18i01.6235.

[19] W. S. Eka Putra, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.

[20] M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, “Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle,” *J. Algoritm.*, vol. 1, no. 1, pp. 45–56, 2020, doi: 10.35957/algoritme.v1i1.434.

[21] H. Abhirawa, M. S. Jondri, and A. Arifianto, “Pengenalan Wajah Menggunakan Convolutional Neural Network Face Recognition Using Convolutional Neural Network,” *e-Proceeding Eng.*, vol. 4, no. 3, pp. 4907–4916, 2017.

[22] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?,” *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, no. NeurIPS, pp. 2483–2493, 2018.

[23] N. K. Manaswi, “Deep Learning with Applications Using Python,” *Deep Learn. with Appl. Using Python*, pp. 31–43, 2018, doi: 10.1007/978-1-4842-3516-4.

[24] F. Ertam, “Data classification with deep learning using tensorflow,” *2nd Int. Conf. Comput. Sci. Eng. UBMK 2017*, pp. 755–758, 2017, doi: 10.1109/UBMK.2017.8093521.

[25] K. Thirunavukkarasu, A. S. Singh, P. Rai, and S. Gupta, “Classification of IRIS dataset using classification based KNN Algorithm in supervised learning,” *2018 4th Int. Conf. Comput. Commun. Autom. ICCCA 2018*, pp. 1–4, 2018, doi: 10.1109/CCAA.2018.8777643.