

**SISTEM DETEKSI PENYAKIT TUBERKULOSIS  
BERBASIS CHEST XRAY MENGGUNAKAN  
METODE CONVOLUTIONAL NEURAL NETWORK  
PADA PLATFORM ANDROID**

**LAPORAN TUGAS AKHIR**

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika  
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



**DISUSUN OLEH :**

**ARDIAN ARIF WAHYUDI**

**NIM 32602000069**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG**

**2024**

***FINAL PROJECT***

***TUBERCULOSIS DISEASE DETECTION SYSTEM BASED ON  
CHEST XRAY USING CONVOLUTIONAL NEURAL NETWORK  
METHOD ON ANDROID PLATFORM***

*Proposed to complete the requirement to obtain a bachelor's degree (S1)  
at Informatics Engineering Department of Industrial Technology Faculty  
Sultan Agung Islamic University*



**ARRANGED BY :**

**ARDIAN ARIF WAHYUDI**

**NIM 32602000069**

**MAJORING OF INFORMATICS ENGINEERING  
INDUSTRIAL TECHNOLOGY FACULTY  
SULTAN AGUNG ISLAMIC UNIVERSITY  
SEMARANG**

**2024**

## LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul **“Sistem Deteksi Penyakit Tuberkulosis Berbasis Chest Xray Menggunakan Metode Convolutional Neural Network Pada Platform Android”** ini disusun oleh :

Nama : Ardian Arif Wahyudi

NIM : 32602000069

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :


Hari : Kamis


Tanggal : 29 Februari 2024

Mengesahkan,

Pembimbing I

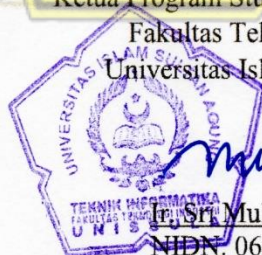
Pembimbing II

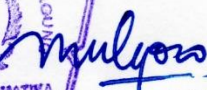
  
Bagus Satrio Waluyo Poetro, S.Kom., M.Cs  
NIDN. 1027118801

  
Ghufron, S.T., M.Kom  
NIDN. 0602079005

Mengetahui,

Ketua Program Studi Teknik Informatika  
Fakultas Teknologi Industri  
Universitas Islam Sultan Agung



  
Ir. Sri Mulyono, M.Eng  
NIDN. 0626066601

## LEMBAR PENGESAHAN PENGUJI


Laporan tugas akhir dengan judul “Sistem Deteksi Penyakit Tuberkulosis Berbasis Chest Xray Menggunakan Metode Convolutional Neural Network Pada Platform Android” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Senin

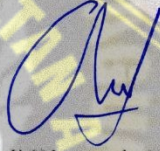
Tanggal : 26 Februari 2024

### TIM PENGUJI

Anggota I

  
Badieah, S.T., M.Kom  
NIDN. 0619018701

Anggota II

  
Andi Riansyah, S.T., M.Kom  
NIDN. 0609108802

## SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Ardian Arif Wahyudi

NIM : 32602000069

Judul Tugas Akhir : Sistem Deteksi Penyakit Tuberkulosis Berbasis Chest Xray Menggunakan Metode Convolutional Neural Network Pada Platform Android

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 28 Februari 2024

Yang Menyatakan,



Ardian Arif Wahyudi

## KATA PENGANTAR

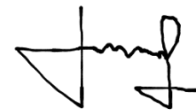
Puji syukur penulis panjatkan kepada Allah SWT, yang telah memberikan rahmat, taufik serta hidayah-Nya, sehingga laporan Tugas Akhir ini dapat terselesaikan.

Tanpa lupa penulis mengucapkan terima kasih kepada :

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto S.H., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Hj. Novi Marlyana, ST., MT., IPU.
3. Dosen pembimbing 1 Bapak Bagus Satrio Waluyo Poetro, S.Kom., M.Cs yang telah membimbing selama tugas akhir berlangsung.
4. Dosen pembimbing 2 Bapak Ghufron, S.T., M.Kom yang telah membimbing selama tugas akhir berlangsung.
5. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini.
6. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan ini masih terdapat banyak kekurangan, untuk itu penulis mengharap kritik dan saran dari pembaca untuk sempurnanya laporan ini. Semoga dengan ditulisnya laporan ini dapat menjadi sumber ilmu bagi setiap pembaca.

Semarang, 12 Januari 2024



Ardian Arif Wahyudi

## DAFTAR ISI

<b>COVER</b> .....	<b>i</b>
<b>HALAMAN PENGESAHAN PEMBIMBING</b> .....	<b>iii</b>
<b>HALAMAN PENGESAHAN PENGUJI</b> .....	<b>iv</b>
<b>SURAT PERNYATAAN KEASLIAN TUGAS AKHIR</b> .....	<b>v</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH</b> .....	<b>vi</b>
<b>KATA PENGANTAR</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>viii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xi</b>
<b>DAFTAR TABEL</b> .....	<b>xiii</b>
<b>ABSTRAK</b> .....	<b>xiv</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah .....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI</b> .....	<b>5</b>
2.1 Tinjauan Pustaka .....	5
2.2 Dasar Teori .....	7
2.2.1 Tuberkulosis .....	7
2.2.2 Chest X-Ray .....	8
2.2.3 Machine Learning .....	9
2.2.4 Deep Learning.....	9
2.2.5 Convolutional Neural Network (CNN).....	11
2.2.6 Tensorflow Lite .....	14
2.2.7 Confusion Matrix .....	15
2.2.8 Flowchart .....	17
<b>BAB III METODE PENELITIAN</b> .....	<b>19</b>

3.1	Metode Penelitian.....	19
3.1.1	Data Collecting.....	19
3.1.2	Data Preprocessing.....	22
3.1.3	Data Cleaning.....	23
3.1.4	Training.....	24
3.1.5	Deployment Model.....	26
3.1.6	Pembuatan Aplikasi Android.....	27
3.2	Analisis Kebutuhan.....	28
3.2.1	Tips Penggunaan Aplikasi.....	28
3.2.2	Unggah Gambar.....	28
3.2.3	Resize Gambar.....	28
3.2.4	Menampilkan Hasil Resize Gambar.....	28
3.2.5	Melakukan Prediksi.....	29
3.2.6	Menampilkan Hasil Prediksi.....	29
3.3	Analisis Sistem.....	29
3.4	Desain Tampilan Antarmuka.....	29
3.4.1	SplashScreen.....	30
3.4.2	Onboarding.....	31
3.4.3	Home.....	32
3.4.4	Camera.....	32
3.4.5	Gallery.....	33
3.4.6	Profile.....	34
<b>BAB IV HASIL DAN ANALISIS PENELITIAN.....</b>		<b>35</b>
4.1	Training Model.....	35
4.2	Cara Kerja Sistem Klasifikasi.....	36
4.3	Implementasi User Interface.....	38
4.3.1	Halaman SplashScreen.....	38
4.3.2	Halaman Onboarding.....	39
4.3.3	Halaman Home.....	40
4.3.4	Halaman Gallery.....	41
4.3.5	Halaman Camera.....	44



4.3.6	Halaman Profile .....	47
4.4	Black Box Testing .....	47
4.5	Hasil dan Analisis .....	49
4.5.1	Training dan Validation .....	49
4.5.2	Testing .....	52
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>55</b>
5.1	Kesimpulan .....	55
5.2	Saran .....	55
<b>DAFTAR PUSTAKA .....</b>		<b>56</b>
<b>LAMPIRAN .....</b>		<b>58</b>



## DAFTAR GAMBAR

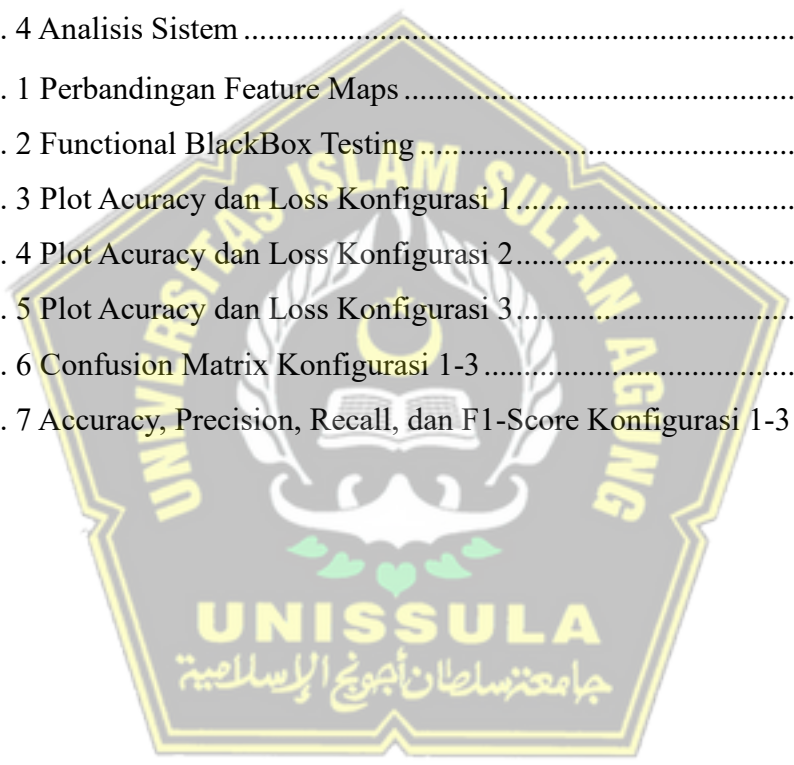
Gambar 2. 1 Tuberkulosis .....	7
Gambar 2. 2 Chest X-Ray .....	8
Gambar 2. 3 Model Lapisan Convolutional Neural Network .....	11
Gambar 2. 4 Convolutional Layer.....	11
Gambar 2. 5 Pooling Layer .....	12
Gambar 2. 6 Fully Connected Layer .....	13
Gambar 3. 1 Flowchart Penelitian.....	19
Gambar 3. 2 Dataset Tuberkulosis .....	19
Gambar 3. 3 Jumlah Data Gambar .....	20
Gambar 3. 4 Dataset TBC NIAID.....	21
Gambar 3. 5 Dataset Mentah Chest X-Ray Tuberkulosis .....	22
Gambar 3. 6 Data Chest X-Ray Ambigu.....	23
Gambar 3. 7 Data Setelah Cleanning .....	24
Gambar 3. 8 Deployment Model TFLite.....	26
Gambar 3. 9 Flowchart Sistem.....	27
Gambar 3. 10 SplashScreen .....	30
Gambar 3. 11 Onboarding.....	31
Gambar 3. 12 Home .....	32
Gambar 3. 13 Camera .....	32
Gambar 3. 14 Gallery.....	33
Gambar 3. 15 Profile .....	34
Gambar 4. 1 Training Model.....	35
Gambar 4. 2 Flowchart Klasifikasi .....	36
Gambar 4. 3 Halaman Splash Screen .....	38
Gambar 4. 4 Halaman Onboarding .....	39
Gambar 4. 5 Halaman Home.....	40
Gambar 4. 6 Halaman Gallery .....	41
Gambar 4. 7 Pilih Gambar Dari Gallery .....	42
Gambar 4. 8 Preview Gambar Halaman Gallery .....	42

Gambar 4. 9 Hasil Prediksi Gallery .....	43
Gambar 4. 10 Halaman Camera .....	44
Gambar 4. 11 Ambil Gambar Dari Kamera .....	45
Gambar 4. 12 Preview Gambar Halaman Camera .....	45
Gambar 4. 13 Hasil Prediksi Camera .....	46
Gambar 4. 14 Halaman Profile .....	47



## DAFTAR TABEL

Tabel 2. 1 Penelitian Deteksi TBC Menggunakan CNN.....	6
Tabel 2. 2 Confusion Matrix .....	15
Tabel 2. 3 Tabel Flowchart.....	17
Tabel 3. 1 Pembagian Dataset .....	24
Tabel 3. 2 Training Layers .....	25
Tabel 3. 3 Konfigurasi Training .....	25
Tabel 3. 4 Analisis Sistem .....	29
Tabel 4. 1 Perbandingan Feature Maps .....	35
Tabel 4. 2 Functional BlackBox Testing .....	48
Tabel 4. 3 Plot Accuracy dan Loss Konfigurasi 1.....	49
Tabel 4. 4 Plot Accuracy dan Loss Konfigurasi 2.....	50
Tabel 4. 5 Plot Accuracy dan Loss Konfigurasi 3.....	51
Tabel 4. 6 Confusion Matrix Konfigurasi 1-3.....	52
Tabel 4. 7 Accuracy, Precision, Recall, dan F1-Score Konfigurasi 1-3.....	53



## ABSTRAK

Tuberkulosis merupakan penyakit yang menginfeksi organ paru bersifat menular yang disebabkan bakteri *Mycobacterium tuberculosis*. Salah satu metode dalam mendiagnosa penyakit ini adalah menggunakan citra rontgen dada. Namun, proses pembacaan hasil rontgen oleh dokter masih dilakukan secara manual, yaitu dengan membaca langsung menggunakan lampu baca rontgen atau x-ray viewer. Akibat hal tersebut, interpretasi hasil rontgen antara dokter berbeda-beda. Oleh karena itu, dibutuhkan suatu sistem yang dapat mendukung keputusan dokter dalam mengidentifikasi hasil rontgen dada. Penelitian ini menggunakan menggunakan algoritma Convolutional Neural Network yang telah teruji mampu mengklasifikasi gambar menggunakan fitur yang dipelajarinya sendiri selama proses konvolusinya. Penelitian ini mengatur jumlah convolution dan pooling layer sebanyak masing-masing 4 lapisan, agar bisa melakukan eksperimen mencari konfigurasi terbaik maka dilakukan perbandingan jumlah kernel/filter dengan jumlah berbeda tiap lapisan dan pada setiap konfigurasinya. Konfigurasi terbaik terdapat pada konfigurasi 1 dengan penggunaan kernel/filter 8-64 dengan hasil evaluasi model 99% pada akurasi, presisi, *recall* dan *f1-score*.

Kata Kunci: Tuberkulosis, Rontgen dada, Convolutional Neural Network, Konfigurasi

## ABSTRACT

*Tuberculosis is an infectious disease of the lungs caused by the bacteria *Mycobacterium tuberculosis*. One of the methods in diagnosing this disease is using chest X-ray images. However, the process of reading x-ray results by doctors is still done manually, namely by reading directly using an x-ray reading lamp or x-ray viewer. As a result of this, the interpretation of x-ray results between doctors is different. Therefore, a system is needed that can support doctors' decisions in identifying chest X-rays. This research uses the Convolutional Neural Network algorithm which has been proven to be able to classify images using features that it learns itself during its convolution process. This research set the number of convolution and pooling layers as many as 4 layers each, in order to experiment to find the best configuration, a comparison of the number of kernels/filters with the number of filters was conducted. comparison of the number of kernels/filters with different numbers of layers and in each configuration. The best configuration is configuration 1 with the use of kernel/filter 8-64 with 99% model evaluation results on accuracy, precision, recall and f1-score.*

*Keywords: Tuberculosis, Chest X-ray, Convolutional Neural Network, Configuration.*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Penyakit infeksius menular bersumber dari bakteri *Mycobacterium tuberculosis* dan menginfeksi organ paru, sering disebut sebagai tuberkulosis paru (TBC) (Fraga dkk., 2021). Penyakit TBC ini penyebarannya melalui udara yaitu udara yang dikeluarkan penderita TBC melalui batuk dan bersin. Gejala utama TBC adalah batuk berdahak dengan durasi kurang lebih dua minggu atau lebih dan serta diikuti dengan gejala lain semisal batuk dengan dahak berdarah, sesak nafas, penurunan berat badan dan nafsu makan, lemas, berkeringat pada waktu malam hari, dan demam selama satu bulan atau lebih (Pertiwi dkk., 2021). Pada tahun 2018, menurut data World Health Organization, sekitar sepuluh juta orang menderita TBC dan 98 ribu di antaranya meninggal dunia (Rochmawanti dkk., 2021). Oleh karena itu, TBC perlu dicegah agar tidak semakin menular penyebarannya di dunia.

Salah satu cara dalam mencegah penularan TBC agar tidak semakin meluas adalah melakukan diagnosa pada orang yang memiliki indikasi terjangkit TBC. Rontgen dada (*chest x-ray*) merupakan salah satu metode untuk mendiagona penyakit ini (Fatimah dkk., 2024). Cara kerja rontgen saat digunakan, yaitu sinar-X yang dikeluarkan melalui tabung dan difokuskan pada area bagian tubuh yang akan didiagona (Prayogi dkk., 2021). Namun, proses pembacaan hasil rontgen oleh dokter masih dilakukan secara manual, yaitu dengan membaca langsung menggunakan lampu baca rontgen atau *x-ray viewer*. Akibat hal tersebut, interpretasi hasil rontgen antara dokter berbeda-beda. Oleh karena itu dibutuhkan sebuah sistem untuk mendukung keputusan dokter dalam pembacaan hasil rontgent dengan memanfaatkan teknologi kecerdasan buatan.

Kecerdasan buatan (AI) telah banyak membantu kehidupan manusia terutama dalam pekerjaan di bidang medis. Salah satu dampak positif AI dalam membantu pekerjaan dokter adalah sebagai pendukung keputusan. Oleh sebab itu, akan menambah keyakinan dokter dalam mendiagnosa pasiennya. AI memiliki beberapa cabang ilmu salah satunya *deep learning*. *Deep learning* dapat mendeteksi dan mengklasifikasikan sebuah penyakit berdasarkan citra medis yang telah dilatih sebelumnya. Algoritma *deep learning* yang sering digunakan pada citra medis adalah Convolutional Neural Network (CNN).

CNN sudah terbukti penggunaannya pada penelitian citra gambar medis khususnya pada klasifikasi gambar penyakit karena mampu belajar sendiri mengenali ciri khas setiap penyakit selama proses konvolusi. Berdasarkan permasalahan diatas, penelitian ini bertujuan untuk membantu dokter dalam pendukung pengambilan keputusan pembacaan hasil *chest x-ray*. Oleh sebab itu, penelitian ini dilakukan bertujuan membuat sistem deteksi menggunakan algoritma CNN karena telah terbukti mempunyai akurasi yang tinggi dalam klasifikasi gambar medis dan mengimplemmentasikan pada sistem berupa aplikasi android.

## 1.2 Perumusan Masalah

Beberapa rumusan masalah yang terdapat dalam tugas akhir ini adalah sebagai berikut ini:

1. Apakah CNN mampu mendeteksi tuberkulosis menggunakan basis citra *chest x-ray* pada platform android?
2. Bagaimana performa dari sistem yang dibuat untuk mendeteksi tuberkulosis berbasis citra *chest x-ray* dengan metode *convolutional neural network*?

## 1.3 Pembatasan Masalah

Beberapa batasan masalah yang membatasi penelitian dalam tugas akhir ini adalah sebagai berikut ini:

1. Dataset *chest x-ray* berjumlah 7000 gambar *chest x-ray* dengan rincian 3500 kelas normal berasal dari Kaggle dan 3500 kelas tuberkulosis berasal dari NIAID.

2. Hasil klasifikasi sistem hanya ada 2 kelas yaitu normal atau tuberkulosis serta nilai probabilitasnya setiap kelasnya.
3. File gambar *chest x-ray* yang bisa di-*upload* untuk diklasifikasi di sistem adalah .jpg, .jpeg dan .png.

#### 1.4 Tujuan

Beberapa tujuan yang ingin diwujudkan dalam tugas akhir ini adalah sebagai berikut ini:

1. Mengembangkan sistem untuk mendeteksi tuberkulosis menggunakan citra *chest x-ray* dengan metode CNN.
2. Menghitung kinerja model pada sistem berdasarkan nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

#### 1.5 Manfaat

Beberapa manfaat yang terkandung dalam tugas akhir ini adalah sebagai berikut ini:

1. Membantu meyakinkan dokter dalam pendukung keputusan dalam pembacaan hasil *chest x-ray*.
2. Memberikan inspirasi bagi penelitian tentang CNN pada klasifikasi penyakit menggunakan citra medis untuk selanjutnya.



## 1.6 Sistematika Penulisan

Sistematika penulisan yang akan dipakai penulis dalam tugas akhir ini adalah sebagai berikut:

### **BAB I : PENDAHULUAN**

Memberikan penjelasan tentang alasan untuk memilih judul, perumusan masalah, pembatasan masalah, tujuan penelitian, manfaat dari penelitian, dan sistematika penulisan.

### **BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI**

Memaparkan penelitian 5 tahun terdahulu yang relevan dengan tema penelitian dan dasar teori yang digunakan untuk merancang sistem deteksi.

### **BAB III : METODE PENELITIAN**

Memuat tentang tahapan-tahapan tentang penelitian mulai dari mendapatkan data, mengolah data, melakukan *training* untuk mendapatkan model, mengimplementasikan model ke dalam aplikasi android hingga, hingga pengujian akurasi.

### **BAB IV : HASIL DAN ANALISIS PENELITIAN**

Mendeskripsikan tentang hasil akhir sistem berupa aplikasi android, proses alur klasifikasi, dan uji akurasi sistem

### **BAB V : KESIMPULAN DAN SARAN**

Menjabarkan tentang kesimpulan dan saran dari penulis terhadap penelitian yang telah dilakukan.

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Pada penelitian terdahulu metode *convolutional neural network* telah banyak dipakai untuk penggunaan klasifikasi gambar. Penelitian implementasi *deep learning* menggunakan *convolutional neural network* pada ekspresi manusia yang dilakukan oleh (Nugroho dkk., 2020), menggunakan *dataset train* model yang terdiri dari 3993 gambar mimik marah, 7164 gambar mimik senang, 4938 gambar mimik sedih, serta 3205 gambar mimik kaget. Selain itu, juga ada 4982 gambar mimik netral, 436 gambar mimik jijik, dan 4103 gambar mimik takut. Hasil penelitian tersebut menunjukkan tingkat akurasi pelatihan 90% dan validasi 65% dengan *epoch* 100, *batch size* 128.

Penelitian diagnosa tuberkulosis paru berbasis citra *x-ray* menggunakan CNN pernah dilakukan oleh (Bahri dkk., 2021) dan kawan kawan. Penelitian tersebut memiliki *dataset* 3500 untuk masing-masing kelas yaitu normal dan tuberkulosis. Pengujian data latih melalui pelatihan pada masing masing kelas 2000 citra, ditemukan bahwa penggunaan *Convolutional Neural Network* (CNN) dapat meningkatkan kinerja klasifikasi penderita tuberkulosis. Akurasi mencapai 92,5%, dan pencapaian ini terjadi pada *epoch* kesembilan dari total 30 *epoch* yang dilakukan.

Penelitian lain tentang deteksi tuberkulosis paru berbasis citra *x-ray* menggunakan CNN juga pernah dilakukan oleh (Suharni dkk., 2021) berjudul “Implementasi Model CNN Untuk Klasifikasi Penyakit TBC Berbasis *Desktop*”. Dataset pada penelitian ini berjumlah 240 data masing-masing 120 gambar untuk kelas normal dan tuberkulosis. Pengujian aplikasi yang menggunakan model CNN dengan menerapkan 15 citra sinar-X, terdiri dari 8 citra normal dan 7 citra tuberkulosis, menunjukkan bahwa semua prediksi sesuai dengan ekspektasi. Penelitian ini mencatat kekurangan bahwa jumlah data yang digunakan masih terbatas. Rangkuman hasil penelitian serupa tentang deteksi tuberculosi menggunakan CNN bisa dilihat pada Tabel 2.1 ini.

Tabel 2. 1 Penelitian Deteksi TBC Menggunakan CNN

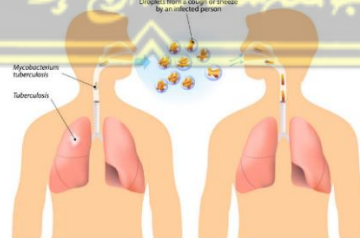
Peneliti (Tahun)	Judul Penelitian	Hasil Penelitian
(Rasyid dkk., 2023)	Klasifikasi Penyakit Tuberculosis (TB) Organ Paru Manusia Berdasarkan Citra Rontgen Thorax Menggunakan Metode <i>Convolutional Neural Network</i>	Penelitian ini dilakukan menggunakan arsitektur <i>MobileNet</i> . <i>MobileNet</i> telah cukup baik untuk mengklasifikasikan penyakit paru menggunakan citra rontgen paru dengan akurasi 98,414%
(Rochmawanti dkk., 2021)	Analisis Peforma <i>Pre-Trained Model Convolutional Neural Network</i> Dalam Mendeteksi Penyakit Tuberculosis	Penelitian ini membandingkan 5 model <i>pre-trained</i> diantaranya <i>MobileNet</i> , <i>DenseNet121</i> , <i>ResNet50</i> , <i>InceptionV3</i> , <i>InceptionResNetV2</i> , dan <i>Xception</i> . Menurut hasil pengujian, model <i>DenseNet121</i> memiliki kemampuan deteksi penyakit TB yang paling akurat sebesar 91,57%.
(Soedradjat dkk., 2022)	Deteksi Gangguan Paru-Paru Berbasis Citra X-Ray Menggunakan <i>Deep Learning</i>	Penelitian ini untuk mendeteksi penyakit pada paru-paru seperti pneumonia, tuberculosis, dan covid-19 dengan metode CNN menggunakan arsitektur <i>ResNet-34</i> . Hasil pengujian pada keempat kelas penyakit menunjukkan 95% akurasi

Peneliti (Tahun)	Judul Penelitian	Hasil Penelitian
		pengujian dan 100% akurasi pelatihan..
(Bahri dkk., 2021)	Diagnosa Tuberculosis Paru Berbasis Citra X-ray Menggunakan <i>Convolutional Neural Network</i>	Pada penelitian ini gambar rontgen paru diubah ke grayscale. Nilai akurasi tertinggi yaitu 92,5% didapatkan pada epoch ke 9 dari 30.

Pada penelitian terdahulu telah menguraikan tentang deteksi tuberkulosis menggunakan CNN yang telah dilakukan dan memiliki akurasi rata rata yang tinggi diatas 90%. Akan tetapi, belum ada aplikasi di platform Android yang dikembangkan untuk tujuan tersebut. Penelitian ini bertujuan untuk memberikan dasar pengetahuan tentang perkembangan penelitian TB dan mengidentifikasi celah penelitian yang dapat diisi melalui pengembangan aplikasi deteksi TB di platform Android

## 2.2 Dasar Teori

### 2.2.1 Tuberkulosis



Gambar 2. 1 Tuberkulosis

Tuberkulosis atau TBC merupakan suatu kondisi tubuh terinfeksi yang diinduksi oleh bakteri *Mycobacterium tuberculosis* (Fraga dkk., 2021). Pada umumnya, penyakit TBC menginfeksi organ paru, namun juga dapat menginfeksi berbagai organ lain, seperti ginjal, tulang, otak, atau kelenjar getah bening. TBC termasuk salah satu penyakit menular yang umum di seluruh dunia dan penularannya bisa melalui air liur penderita saat batuk.

Tuberkulosis seringkali menunjukkan gejala seperti batuk yang menahun, demam, berat badan menurun, dan berkeringat pada waktu malam hari (Dewi, 2020.). Untuk mengobati penyakit TBC, diperlukan penggunaan antibiotik dalam jangka waktu kurang lebih 6-9 bulan dan teratur untuk menghentikan pertumbuhan bakteri penyebabnya. Mendeteksi dan mengatasi TBC dengan cepat sangat penting, karena apabila tidak diobati, penyakit ini bisa berdampak fatal dan menyebar ke organ tubuh lainnya

### 2.2.2 Chest X-Ray



Gambar 2. 2 Chest X-Ray

Pemeriksaan radiografi dada atau rontgen dada yang lebih dikenal sebagai *chest x-ray*, merupakan suatu teknik medis yang menggunakan sinar-X (radiasi elektromagnetik) untuk menghasilkan gambaran visual dari organ dan struktur di dalam rongga dada individu. Teknologi ini memiliki peran yang sangat vital dalam bidang kedokteran untuk membantu dalam proses diagnosis berbagai kondisi dan penyakit yang berhubungan dengan daerah dada, termasuk pneumonia, tuberkulosis, kanker paru-paru, serta cedera pada bagian dada. Salah satu cara untuk mendiagnosis tuberkulosis adalah dengan melakukan pemeriksaan foto rontgen dada.. Pada terduga penderita TBC, hasil rontgen dada (*chest x-ray*) menunjukkan

gambaran bercak putih pada AP (proyeksi depan) dan lateral (proyeksi samping) (Nurmalasari, 2020).

### 2.2.3 Machine Learning

*Machine learning* merupakan komponen dari *artificial intelligence* yang berfokus pada pemahaman pola guna mencari solusi terbaik, memungkinkan komputer untuk secara otomatis menemukan solusi optimal dari data (Siswoyo, 2020). *Machine learning* sudah mengalami perkembangan pesat beberapa tahun terakhir karena kemajuan teknologi.

Tujuan utama dari *machine learning* adalah memungkinkan komputer untuk secara otomatis dan mandiri belajar dari data, dengan tujuan untuk melakukan identifikasi pola dan memahami informasi yang terkandung dalam data tersebut. Dengan pemahaman ini, komputer dapat digunakan untuk mengambil keputusan, membuat prediksi, atau menyelesaikan tugas-tugas kompleks tanpa perlu diprogram secara eksplisit.

Proses pelatihan (*training*) merupakan tahap kunci dalam machine. Saat pelatihan, algoritma *machine learning* diberikan *dataset* yang berisi contoh-contoh data dengan label atau informasi yang ingin diprediksi. Algoritma ini kemudian mencoba untuk menyesuaikan model matematika atau statistiknya sehingga dapat mewakili pola-pola yang ada dalam data tersebut.

### 2.2.4 Deep Learning

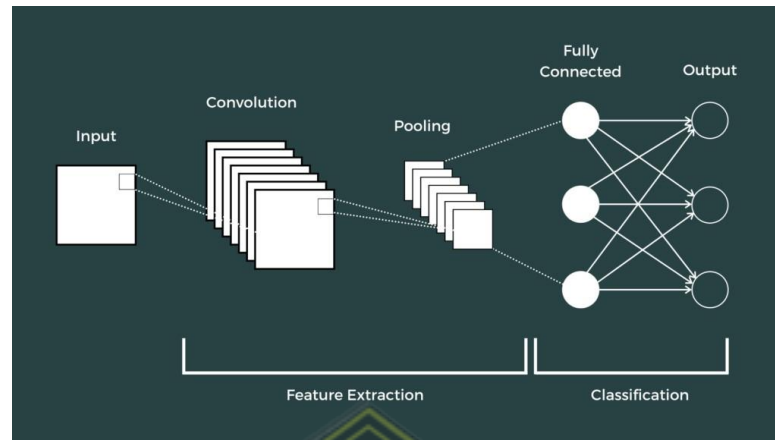
*Deep Learning* adalah komponen dalam bidang kecerdasan buatan dan *machine learning* yang terus berkembang melalui penggunaan jaringan saraf berlapis banyak untuk meningkatkan presisi dalam berbagai tugas seperti identifikasi objek, pemahaman suara, penerjemahan bahasa, dan sejenisnya (Raup, 2022.).

Salah satu aspek yang paling mencolok dalam pemanfaatan *Deep Learning* adalah kemampuannya dalam melakukan klasifikasi gambar. Dalam situasi ini, jaringan saraf mendalam memiliki kapabilitas untuk menganalisis volume besar data gambar dan mengenali objek, pola, atau ciri-ciri khusus dalam gambar tersebut. Proses ini melibatkan proses pembelajaran mesin di mana jaringan saraf "menggali pengetahuan" dari beragam *dataset* pelatihan yang luas, memungkinkan mereka untuk mengidentifikasi pola yang rumit dan memisahkan objek dalam gambar dengan ketepatan prediksi yang tinggi.

Salah satu contohnya, dalam suatu penelitian yang berjudul "Pemanfaatan *Deep Learning* dalam Pengklasifikasian Tanaman Obat Berdasarkan Karakteristik Daun dengan Menggunakan Aplikasi Android" oleh (Alamsyah, 2019) digunakan sepuluh kategori yang berbeda untuk mengidentifikasi tanaman obat, seperti mengkudu, kumis kucing, jambu biji, blimbing wuluh, dan sebagainya. Hasil pengujian pada *train data* menunjukkan akurasi sekitar 75%, sementara pada data uji, tingkat akurasi mencapai sekitar 80%

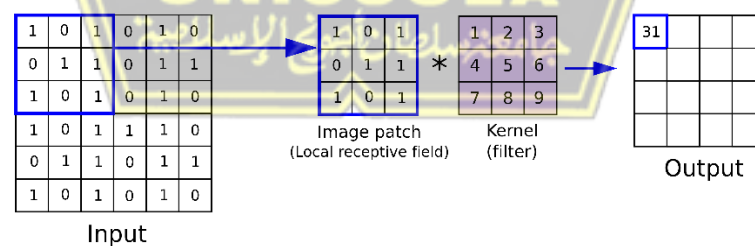
Pemanfaatan *Deep Learning* dalam klasifikasi gambar telah membawa perkembangan yang cukup signifikan dalam berbagai industri, seperti pengenalan wajah untuk keamanan, analisis gambar medis, dan pencarian visual di platform online. Dengan terus berkembangnya teknologi *Deep Learning*, kita dapat mengantisipasi peningkatan lebih lanjut dalam akurasi dan kapabilitas klasifikasi gambar di masa depan, yang akan membantu mengatasi berbagai tantangan kompleks dalam pengolahan gambar dan identifikasi objek.

## 2.2.5 Convolutional Neural Network (CNN)



Gambar 2. 3 Model Lapisan *Convolutional Neural Network*

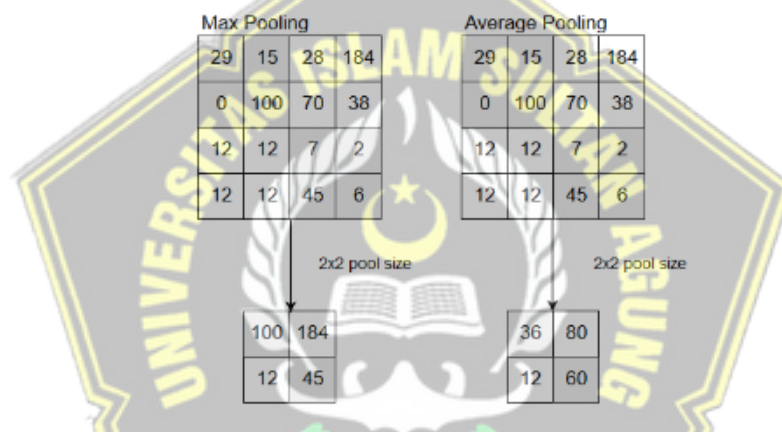
*Convolutional Neural Network* merupakan evolusi dari *Multi Layer Perceptron* yang dirancang khusus untuk mengolah data berdimensi dua. CNN merupakan bagian integral dari *Deep Neural Network* yang sering digunakan dalam analisis gambar dan sering diterapkan dalam penelitian berbasis citra (Alwanda dkk., 2020). Citra merupakan suatu gambaran, kesamaan, atau replikasi dari suatu objek. Arsitektur CNN terdiri dari tiga jenis lapisan, termasuk lapisan input, output, serta beberapa lapisan yang tersembunyi mencakup komponen seperti lapisan konvolusi, lapisan penggabungan, dan lapisan terhubung penuh (Alwanda dkk., 2020).



Gambar 2. 4 *Convolutional Layer*

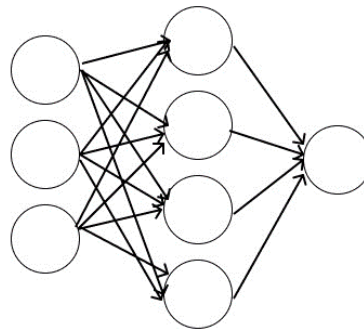


*Convolutional layer* adalah bagian kunci pada CNN berfungsi melakukan operasi konvolusi terhadap keluaran lapisan sebelumnya. Lapisan ini memiliki sejumlah *filter* yang diajarkan secara *random* untuk menjalankan operasi konvolusi, dengan tujuan untuk mengekstraksi fitur yang akan mewakili representasi fitur dari lapisan masukan. Tujuan utama operasi konvolusi terhadap citra adalah mengekstraksi fitur-fitur dari citra masukan. Pada operasi ini, informasi spasial yang terkandung dalam data masukan diubah secara linear oleh transformasi linear. Bobot-bobot dalam lapisan ini menggambarkan kernel konvolusi yang digunakan, dan kernel ini dapat disesuaikan melalui pelatihan menggunakan *input* pada CNN (Alwanda dkk., 2020).



Gambar 2. 5 Pooling Layer

Setelah proses *filtering* selesai, gambar akan masuk ke dalam lapisan *pooling*. Lapisan *pooling* adalah komponen yang bertanggung jawab untuk mengurangi resolusi gambar yang telah melalui tahap pemrosesan sebelumnya. Tujuan dari lapisan *pooling* adalah mengurangi gangguan (*noise*) yang terdapat pada gambar. Terdapat 2 jenis *pooling* diantaranya, yaitu *average pooling* dan *max pooling*. Anda bisa melihat contoh ilustrasinya pada Gambar 5 Average pooling biasanya lebih sering *digunakan* dalam berbagai arsitektur CNN. *Max Pooling* adalah metode *pooling* yang memilih nilai terbesar dari sekelompok nilai dalam suatu wilayah gambar. Sementara itu, *average pooling* mengambil rata-rata nilai dari wilayah gambar yang sama (Hasan Mahmud & Al Faraby, 2019.).



Gambar 2. 6 *Fully Connected Layer*

*Fully connected layer* merupakan elemen dalam jaringan saraf yang bertanggung jawab untuk mengubah dimensi data dengan tujuan mengklasifikasikannya secara linier. Untuk menghasilkan hasil dari lapisan ini, tidak ada kebutuhan untuk melakukan operasi konvolusi; sebaliknya, dilakukan operasi perkalian matriks yang kemudian diikuti dengan penambahan bias. Dalam metode ini, setiap neuron terhubung ke semua aktivasi pada lapisan sebelumnya., itulah sebabnya lapisan ini dikenal sebagai "*fully connected layer*" (Alwanda dkk., 2020).

Fungsi aktivasi yang sering juga disebut sebagai neuron, merupakan fungsi non-linear yang memungkinkan jaringan saraf tiruan menyelesaikan tugas-tugas yang rumit. Setiap fungsi aktivasi menerima input numerik dan melakukan operasi matematika pada input tersebut. Pada arsitektur *convolutional neural network*, fungsi aktivasi ditempatkan pada tahap akhir dari proses konvolusi atau *pooling*, dan berperan dalam menghasilkan peta fitur yang digunakan untuk mengekstraksi pola-pola fitur (Alwanda dkk., 2020).

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} \quad (1)$$

Simbol	Keterangan
$\exp(x_i)$	Fungsi eksponensial standar diterapkan pada setiap elemen dari vektor input. Ini memberikan nilai positif di atas 0, yang akan sangat kecil jika masukkannya negatif, dan sangat besar jika

Simbol	Keterangan
	masukannya besar. Namun, itu masih belum ditetapkan dalam kisaran (0, 1) yang diperlukan dari suatu probabilitas
$\sum_{j=1}^n$	Symbol tersebut adalah istilah normalisasi. Ini memastikan bahwa semua nilai keluaran fungsi akan berjumlah 1 dan masing-masing berada dalam kisaran (0, 1), sehingga merupakan distribusi probabilitas yang valid
K	Jumlah kelas dalam multi-class classifier

*Softmax* adalah salah satu fungsi aktivasi yang penting dalam jaringan syaraf tiruan, terutama dalam konteks klasifikasi. Fungsi ini mengambil sejumlah nilai sebagai input dan menghasilkan probabilitas sebagai output. Dalam arsitektur Convolutional Neural Network (CNN), fungsi *softmax* biasanya digunakan setelah proses perhitungan konvolusi atau pooling pada feature map. Fungsi ini membantu mengubah nilai-nilai yang dihasilkan oleh jaringan menjadi distribusi probabilitas, yang dapat digunakan untuk mengidentifikasi kelas atau kategori yang paling mungkin mewakili *input* tersebut.

### 2.2.6 Tensorflow Lite

TensorFlow Lite adalah *framework* yang membuat perangkat mobile terbatas seperti smartphone untuk menggunakan *machine learning*. TensorFlow Lite dikembangkan oleh Google Brain bersifat *open-source* dan *multiplatform* yaitu mendukung sistem operasi Android dan IOS, Linux, dan mikrokontroler (Christyanto dkk., 2022).

Salah satu fitur utama Tensorflow Lite adalah dukungannya untuk eksekusi model secara *on-device*, yang berarti model *machine learning* dapat dijalankan langsung di perangkat tanpa perlu koneksi internet. Hal ini memungkinkan penggunaan model dengan respons cepat dan menjaga privasi data karena tidak perlu mentransfer data ke *cloud* untuk inferensi.

TFLite mendukung berbagai tipe model *machine learning*, termasuk *neural networks* untuk tugas-tugas seperti klasifikasi gambar, deteksi objek, dan pengenalan suara. Proses konversi model dari TensorFlow ke TensorFlow Lite memungkinkan pengembang untuk mengoptimalkan ukuran model dan meningkatkan kinerja inferensi pada perangkat dengan keterbatasan memori dan daya.

Selain itu, TensorFlow Lite menyediakan API yang mudah digunakan untuk integrasi dengan aplikasi Android dan iOS, mempermudah pengembang dalam mengimplementasikan kecerdasan buatan di aplikasi *mobile*. Dengan demikian, TFLite menjadi alat yang sangat berguna untuk mendukung pengembangan aplikasi *machine learning on-device* dengan fokus pada perangkat terbatas.

### 2.2.7 Confusion Matrix

*Confusion matrix* adalah sebuah tabel yang digunakan dalam evaluasi performa model klasifikasi, terutama dalam *machine learning* yang memperlihatkan jumlah data yang diklasifikasikan dengan benar dan yang salah oleh model. (Hozairi dkk., 2021).

Tabel 2. 2 *Confusion Matrix*

	<i>Actually Positive</i>	<i>Actually Negative</i>
<i>Predicted Positive</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
<i>Predicted Negative</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

*Confusion matrix* memiliki empat nilai utama yang menggambarkan hasil dari proses klasifikasi, yaitu:

1. *True Positive (TP)*: Nilai TP adalah jumlah gambar *chest x-ray* termasuk dalam kelas "tuberkulosis" dan juga diprediksi dengan benar oleh model sebagai "tuberkulosis.". Dalam konteks ini, ini jumlah *chest x-ray* tuberkulosis yang berhasil diidentifikasi dengan benar oleh model
2. *True Negative (TN)*: Nilai TN adalah jumlah gambar *chest x-ray* yang sebenarnya termasuk dalam kelas "normal" dan juga diprediksi dengan benar oleh model sebagai "normal." Dalam konteks ini, ini adalah jumlah *chest x-ray* normal yang berhasil diidentifikasi dengan benar oleh model.

3. *False Positive* (FP): Nilai FP adalah jumlah gambar *chest x-ray* yang sebenarnya termasuk dalam kelas "normal," tetapi model salah memprediksi sebagai "tuberkulosis." Dalam konteks ini, ini adalah jumlah *chest x-ray* normal yang keliru diidentifikasi sebagai *chest x-ray* dengan tuberkulosis oleh model.
4. *False Negative* (FN): Nilai FN adalah jumlah gambar *chest x-ray* yang sebenarnya termasuk dalam kelas "tuberkulosis," tetapi model salah memprediksi sebagai "normal". Dalam konteks ini, ini adalah jumlah *chest x-ray* dengan tuberkulosis yang keliru diidentifikasi sebagai *chest x-ray* normal oleh model.

Dengan nilai-nilai ini, kita dapat melakukan berbagai perhitungan untuk mengevaluasi performa model, seperti:

1. *Accuracy* adalah proporsi dari prediksi yang benar, dihitung dengan rumus sebagai berikut :

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (2)$$

2. *Precision* adalah proporsi dari prediksi positif yang benar. Dalam konteks ini, presisi akan memberi tahu kita seberapa baik model dalam mengidentifikasi *chest x-ray* tuberkulosis. *Precision* dihitung dengan rumus sebagai berikut

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

3. *Recall* adalah proporsi dari data aktual yang benar-benar diidentifikasi sebagai positif oleh model, Dalam konteks ini, *recall* akan memberi tahu kita sejauh mana model mampu mendeteksi *chest x-ray* tuberkulosis. *Recall* dihitung dengan rumus sebagai berikut

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$








4. *F1-Score* adalah metrik gabungan yang memadukan *precision* dan *recall* dalam memberikan perspektif yang lebih luas dan komprehensif tentang performa model. *F1-Score* dihitung dengan rumus sebagai berikut

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

### 2.2.8 Flowchart

Flowchart adalah bagan yang menjelaskan urutan prosedur dan menunjukkan alur kerja yang ada di sistem (Rozaq dkk., 2021). Dengan menggunakan simbol-simbol yang telah ditentukan, flowchart membantu memperjelas langkah-langkah dalam suatu prosedur. Ini memungkinkan pengguna untuk memahami dan menganalisis alur kerja dengan lebih baik. Selain itu, flowchart juga dapat digunakan sebagai panduan untuk mengidentifikasi potensi perbaikan atau penyempurnaan dalam suatu sistem.

Tabel 2. 3 Tabel *Flowchart*

Nama	Simbol	Fungsi
<i>Terminator</i>		Menyatakan awal atau akhir program
<i>Process</i>		Menyatakan suatu proses dalam program
<i>Input/Output</i>		Menyatakan melakukan <i>input</i> atau menghasilkan <i>output</i> pada program
<i>Decision</i>		Menyatakan suatu pilihan/keadaan menghasilkan pilihan “ya” atau “tidak”
<i>Document</i>		Menyatakan <i>input</i> berasal dari sebuah dokumen dalam bentuk fisik dan <i>output</i> perlu dicetak
<i>Predefine Process</i>		Menyatakan suatu bagian atau prosedur
<i>Flow Line</i>		Arah aliran sebuah <i>flowchart</i>

Tabel 2.3 merupakan simbol simbol *flowchart* beserta fungsinya dalam sebuah program. Simbol *flowchart* lebih banyak dari yang ada pada Tabel 2.3, tetapi yang sering dipakai 6 simbol tersebut. Kegunaan simbol tersebut agar pengguna lebih mudah dalam membaca alur sebuah program

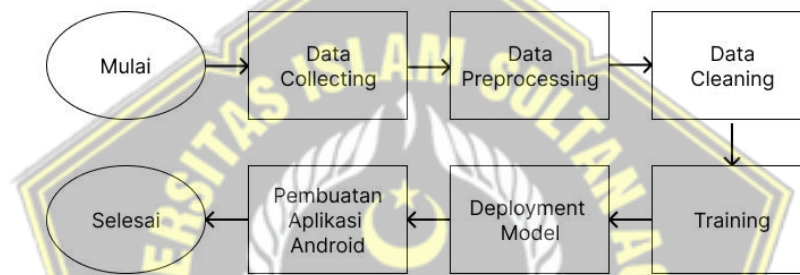


## BAB III

### METODE PENELITIAN

#### 3.1 Metode Penelitian

Pada penelitian pengembangan aplikasi android untuk deteksi tuberkulosis berbasis *chest x-ray*, metode yang digunakan adalah *Convolutional Neural Network*, dimana metode ini nantinya akan melakukan klasifikasi penyakit tuberkulosis berdasarkan hasil rontgen dada dan ada beberapa tahapan yang perlu dilalui dalam penelitian, dapat terlihat pada Gambar 3.1.



Gambar 3. 1 *Flowchart* Penelitian

#### 3.1.1 Data Collecting

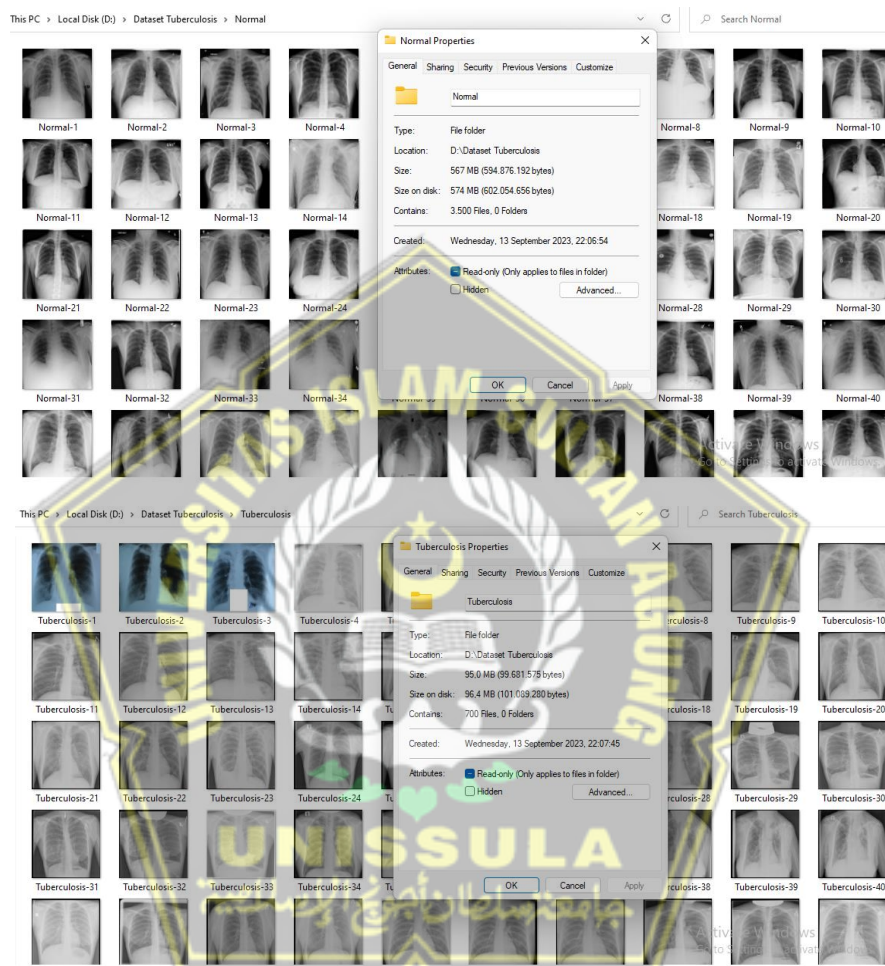
The screenshot shows the Kaggle dataset page for 'Tuberculosis (TB) Chest X-ray Database'. The page header includes the author 'TAWISIFUR RAHMAN AND 2 COLLABORATORS - UPDATED 2 YEARS AGO', a view count of '166', and a 'Download (696 MB)' button. The dataset title is 'Tuberculosis (TB) Chest X-ray Database' with a sub-description 'The largest TB Chest X-ray Database' and a thumbnail image of a chest X-ray. Below the title, there are links for 'Data Card', 'Code (63)', and 'Discussion (3)'. The 'About Dataset' section describes the dataset as being created by researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh, along with collaborators from Malaysia. It mentions that the dataset contains 700 TB images publicly accessible and 2800 TB images available for download. The 'Usability' score is 9.38, and the license is 'Data files © Original Authors'. The 'Expected update frequency' is 'Annually'.

Gambar 3. 2 *Dataset* Tuberkulosis

*Dataset* yang dikumpulkan berupa data berupa citra *chest x-ray* atau rontgen dada yang memiliki 2 kelas yaitu normal dan tuberkulosis. Data tersebut diambil dari kaggle yang bersifat *opensource* dengan *link* sebagai berikut, <https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-x-ray->



dataset. *Dataset* ini telah diuji validasinya oleh dokter medis dari Hamad Medical Corporation. Citra pada *dataset* tersebut telah memiliki resolusi sama semua yaitu 512x512. Pada kelas gambar normal terdapat 3500 gambar, sedangkan pada kelas tuberculosis terdapat 700 gambar.



Gambar 3. 3 Jumlah Data Gambar

Data kelas normal dan tuberculosis memiliki perbedaan jumlah data yang sangat signifikan, dimana pada class normal memiliki 3500 data, sedangkan class tuberculosis memiliki 700 data. Hal ini akan menyebabkan *overfitting*, hasil model yang kurang akurat, dan bias saat melakukan *training*. *Overfitting* merupakan keadaan saat suatu model mempelajari data pelatihan dengan terlalu baik sehingga membuatnya berkinerja buruk pada data baru. *Overfitting* menyebabkan model terlalu cocok dengan data pelatihan dan mengingat noise yang sangat spesifik (Saputra dkk., 2020). Akibatnya, karena pada dasarnya model menghafal contoh

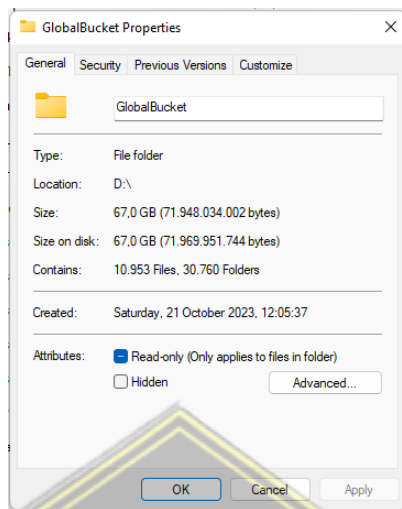
pelatihan daripada mempelajari pola yang lebih luas, model tersebut tidak dapat menggeneralisasi data baru dengan baik.. Oleh karena itu, mencari data gambar *chest x-ray* tambahan ke NIAID. NIAID (The National Institute of Allergy and Infectious Diseases) atau dalam bahasa indonesia Institut Nasional Penyakit Alergi dan Penyakit Menular merupakan badan yang melakukan penelitian dasar dan terapan untuk lebih memahami, mengobati, dan mencegah penyakit menular, imunologi, dan alergi.

Name	Size	Last Modified
<input type="checkbox"/> TB_Portals_Published_CTs_August_2023.zip <input type="checkbox"/> Image Quality.zip <input type="checkbox"/> ReadMe - About the Data.docx <input type="checkbox"/> ReadMe - Downloading the Imaging Data.docx <input type="checkbox"/> ReadMe - Notice of DUA.docx <input type="checkbox"/> TB Portals Data Dictionary.pdf <input type="checkbox"/> TB_Portals_Published_CXRs_August_2023.zip <input type="checkbox"/> TB_Portals_Published_Imaging_Data_August_2023.zip	 3 KB 13.2 KB 17.2 KB 17.7 KB 366 KB 42.2 GB 2.88 MB	 08-10-2023 10:26:26 12-01-2020 09:30:32 11-23-2020 12:49:33 03-17-2021 09:04:08 11-23-2020 12:49:51 08-10-2023 09:02:19 09-11-2023 03:45:23 09-11-2023 10:26:36

Gambar 3. 4 Dataset TBC NIAID

Pada Gambar 3.4 merupakan dataset TBC dari NIAID di *platform collaboration* mereka. Pada *dataset* tersebut terdapat 2 jenis data yaitu CXRs (*Chest X-Ray*) dengan ukuran 42.2 gb dan CTs (*Computed Tomography*) dengan ukuran 134 gb. Pada penelitian ini, menggunakan citra rontgen paru maka memilih datfaset CXRs berukuran 42.2 gb untuk di unduh. Data akhir penelitian ini berjumlah 7000 terdiri dari 3500 kelas normal dan 3500 kelas tuberkulosis.

### 3.1.2 Data Preprocessing



Gambar 3. 5 Dataset Mentah *Chest X-Ray* Tuberkulosis

Pada Gambar 3.5 merupakan folder dataset CXRs setelah diextrak karena sebelumnya berformat zip. Terlihat pada gambar tersebut folder GlobalBucket memiliki 10.953 file berformat .dcm dan 30.760 folder. Format .dcm merupakan format gambar dari mesin *x-ray* yang masih original belum di proses. Pada tahap *preprocessing* kali ini, melakukan beberapa langkah untuk mengolah dataset *chest x-ray* menjadi sama dengan yang diperoleh sebelumnya dari kaggle di antara lain :

1. Mengambil semua file .dcm karena setiap file .dcm terdapat folder di dalam folder sehingga membuat program python untuk mengumpulkan semua file .dcm dalam satu folder
2. Me-*rename* semua file .dcm yang telah berada dalam satu folder dengan program python dengan membuat format nama Tuberculosis-1, Tuberculosis-2 dan seterusnya
3. Mengubah format file .dcm menjadi .png dengan program python untuk menyesuaikan format dataset pada kaggle yang .png
4. Melakukan *resize* dengan program python dikarenakan file.dcm memiliki resolusi tinggi, me-*resize* ke resolusi 512x512 untuk menyamakan pada *dataset* kaggle serta mengurangi ukuran agar *training* lebih cepat
5. Melakukan *copy* file 3500 gambar tuberkulosis untuk mengimbangi kelas normal yang juga memiliki 3500 gambar, dengan menghapus 700 gambar tuberkulosis pada *dataset* kaggle sebelumnya untuk mencegah redudansi data.

*Dataset* ini mengalami proses variasi data pada data pelatihan dengan mengubah, memutar, memanipulasi, atau menerapkan transformasi lainnya. Tujuannya adalah untuk meningkatkan keragaman data pelatihan, sehingga model dapat lebih baik dalam menggeneralisasi pola yang ada dalam data.

Pada dataset citra *chest x-ray* ini terdapat variasi data meliputi:

1. Pemutaran gambar.
2. Pelebaran atau penyempitan gambar.
3. Pemotongan gambar
4. Perubahan kecerahan, kontras, atau saturasi warna.

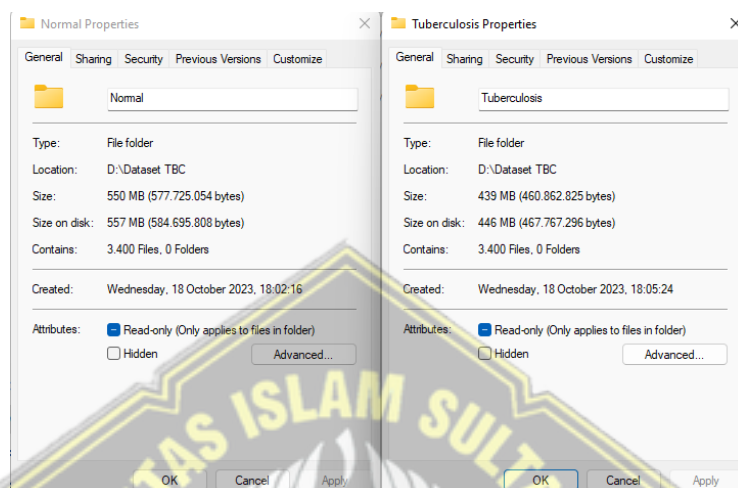
### 3.1.3 Data Cleaning

Menghapus data gambar *chest x-ray* yang ambigu dan tidak jelas, contohnya terlalu dipaksakan memiliki resolusi 512x512 dengan banyak *whitespot* seperti yang terlihat pada Gambar 3.6 agar saat training model yang diharapkan memiliki kemampuan memprediksi yang lebih baik.



Gambar 3. 6 Data *Chest X-Ray* Ambigu

Pada Gambar 3.7 merupakan jumlah data pada kelas data normal dan tuberculosis. Jumlah data pada 2 kelas tersebut sama yaitu 3400 data hal ini agar hasil model yang di-*train* tidak bias pada salah satu kelas dan memiliki akurasi yang tinggi.



Gambar 3. 7 Data Setelah *Cleanning*

### 3.1.4 Training

Sebelum proses training membagi *dataset* yang akan digunakan menjadi beberapa bagian diantaranya yaitu data *training*, *testing*, dan *validation* dengan jumlah sesuai Tabel 3.1 berikut:

Tabel 3. 1 Pembagian *Dataset*

Data	Normal	Tuberculosis
<i>Training</i>	2720 Images	2720 Images
<i>Testing</i>	510 Images	510 Images
<i>Validation</i>	170 Images	170 Images
Jumlah Total	3400 Images	3400 Images

Mengatur *convolution layer* dan *pooling layer* sendiri sesuai dengan kebutuhan. Mengatur *convolution layer* secara manual memungkinkan pengguna untuk menyesuaikan ukuran kernel dan jumlah filter, meningkatkan kontrol terhadap ekstraksi fitur. Sementara itu, pengaturan *pooling layer* secara independen memungkinkan penentuan metode *pooling* dan ukuran *pool* yang optimal,

meningkatkan fleksibilitas dalam mengekstrak informasi penting dari representasi fitur. Semua *layer* yang terdapat di dalam *training* terlihat pada Tabel 3.2.

Tabel 3. 2 *Training Layers*

Layer	Konfigurasi 1	Konfigurasi 2	Konfigurasi 3
Conv2D_1	8 [3x3]	16 [3x3]	32 [3x3]
MaxPool_1	2x2	2x2	2x2
Conv2D_2	16 [3x3]	32 [3x3]	64 [3x3]
MaxPool_2	2x2	2x2	2x2
Conv2D_3	32 [3x3]	64 [3x3]	128 [3x3]
MaxPool_3	2x2	2x2	2x2
Conv2D_4	64 [3x3]	128 [3x3]	256 [3x3]
MaxPool_4	2x2	2x2	2x2
<i>Flatten</i>	-	-	-
<i>Dropout</i>	0.5	0.5	0.5
Dense	128	256	512
Dense	2 - Softmax	2 - Softmax	2 - Softmax

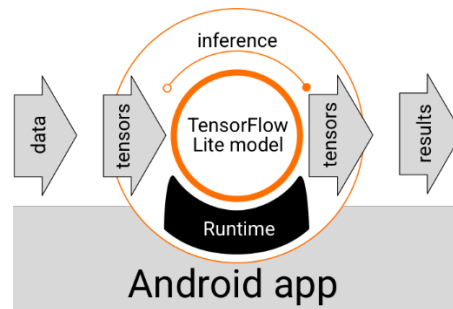
Selain mengatur *convolution layer* dan *pooling layer* sendiri, juga menerapkan konfigurasi tertentu yang sama pada konfigurasi 1-3 saat *training*. Konfigurasi *training* yang peneliti lakukan terdapat pada Tabel 3.3 berikut:

Tabel 3. 3 Konfigurasi *Training*

Nama Konfigurasi	Konfigurasi
batch_size	16
Epoch	30
learning_rate	default(0.001)
Optimizer	RMSprop
Loss	CategoricalCrossentropy

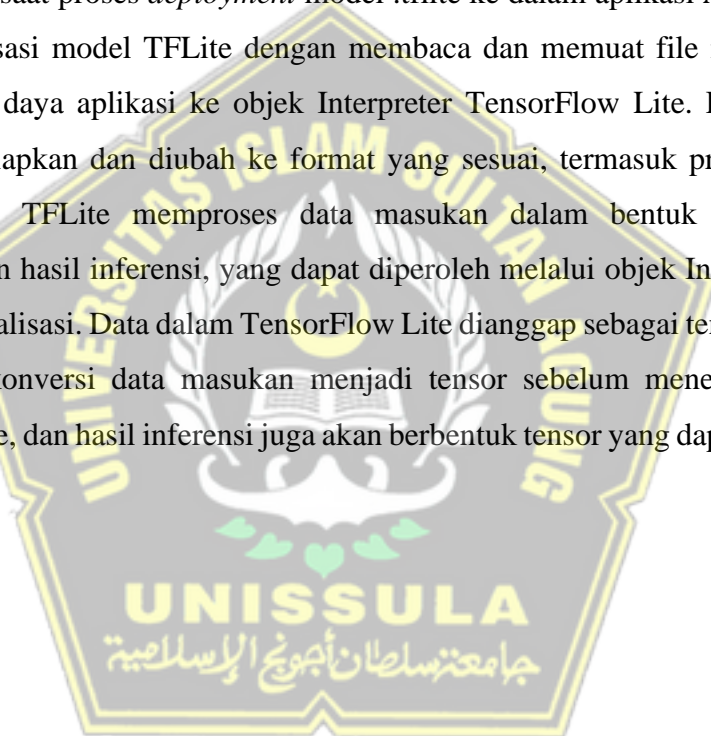
Tabel 3.3 menunjukkan konfigurasi *hyperparameter* yang sama antara konfigurasi 1-3. *Hyperparameter* ditetapkan sama untuk ketiga konfigurasi agar lebih terfokus pada konvolusi layer dengan membandingkan jumlah filter yang berbeda pada setiap lapisannya.

### 3.1.5 Deployment Model



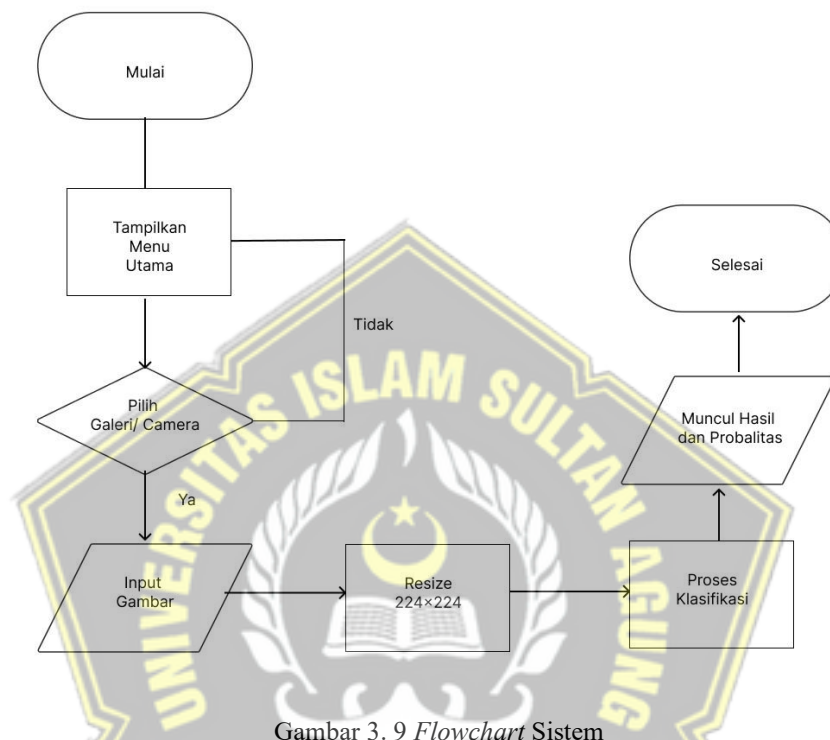
Gambar 3. 8 *Deployment Model TFLite*

Pada saat proses *deployment* model .tflite ke dalam aplikasi Android, perlu menginisialisasi model TFLite dengan membaca dan memuat file model TFLite dari sumber daya aplikasi ke objek Interpreter TensorFlow Lite. Data inferensi harus dipersiapkan dan diubah ke format yang sesuai, termasuk pra-pemrosesan data. Model TFLite memproses data masukan dalam bentuk tensor untuk menghasilkan hasil inferensi, yang dapat diperoleh melalui objek Interpreter yang sudah diinisialisasi. Data dalam TensorFlow Lite dianggap sebagai tensor, sehingga diperlukan konversi data masukan menjadi tensor sebelum meneruskannya ke model TFLite, dan hasil inferensi juga akan berbentuk tensor yang dapat diolah atau ditampilkan.



### 3.1.6 Pembuatan Aplikasi Android

Sistem deteksi tuberkulosis berbasis *chest x-ray* dengan metode CNN yang dibuat merupakan sebuah aplikasi android. Alur sistem aplikasi android terdapat pada Gambar 3.9.



Gambar 3. 9 Flowchart Sistem

Pada Gambar 3.9 menunjukkan alur sistem dari aplikasinya, dimulai dari halaman *home* akan ada 2 pilihan *button* yaitu *Camera* atau *Gallery*. Apabila pengguna memilih *Camera* atau *Gallery* pengguna tinggal menginput gambar atau foto apabila tidak maka kembali ke menu utama. Hasil gambar yang sudah ada baik dari camera maupun gallery akan mengalami *image preprocessing* di *resize* ke 224x224 secara RGB karena model hanya bisa inputan tersebut. Apabila gambar tidak melalui *preprocessing* sebagaimana yang dijelaskan sebelumnya maka aplikasi akan *force close*. Jika *input* sudah sesuai dengan yang diminta model .tflite selanjutnya yaitu menuju dalam proses pengklasifikasian oleh model TFLite, setelah itu aplikasi akan menunjukkan hasil klasifikasi citra *chest x-ray* yaitu normal atau tuberkulosis



## 3.2 Analisis Kebutuhan

Pada fase analisis kebutuhan ini, dilakukan evaluasi terhadap kemampuan sistem, termasuk dari tahap penerimaan input hingga penghasilan hasil klasifikasi oleh sistem. Terdapat beberapa proses atau fungsi yang perlu disertakan dalam sistem ini diantaranya sebagai berikut:

### 3.2.1 Tips Penggunaan Aplikasi

Tips penggunaan aplikasi adalah hal yang penting agar pengguna tidak salah dalam memasukkan citra. Citra *chest x-ray* atau rontgen dada yang dimasukkan harus di-*resize* resolusinya menjadi 224x224 dengan 3 *channel* warna atau RGB agar sesuai dengan input yang diminta model. Terdapat beberapa aturan untuk memaksimalkan hasil prediksi, yaitu sebagai berikut:

1. Memiliki gambar rontgen dada dengan ratio 1:1 karena akan di-*resize* menjadi 224x224
2. Apabila gambar ratio tidak mendekati 1:1 dikhawatirkan saat *resize* gambar menjadi gepeng sehingga akurasi berkurang

### 3.2.2 Unggah Gambar

Fitur unggah gambar merupakan komponen utama dalam sistem ini, di mana pengguna diberi pilihan yaitu apakah pengguna dapat memilih apakah ingin mengambil gambar langsung dari kamera atau memilih gambar yang telah tersimpan dalam galeri.

### 3.2.3 Resize Gambar

Fitur *resize* gambar adalah fitur yang fundamental untuk menghasilkan citra yang siap untuk diprediksi. Apabila gambar tidak diresize ke ukuran 224x224 ketika diprediksi oleh model maka aplikasi akan *force close*. Hal tersebut karena *ByteBuffer* yang dihasilkan gambar yang tidak diresize melebihi *ByteBuffer* yang bisa ditampung oleh model .tflite.

### 3.2.4 Menampilkan Hasil Resize Gambar

Apabila gambar telah di-*resize* ke 224x224 proses selanjutnya adalah menampilkan gambar setelah di *resize* untuk memastikan hasil gambar sudah tepat dan sesuai dengan tips penggunaan aplikasi.

### 3.2.5 Melakukan Prediksi

Proses ini adalah melakukan prediksi pada citra yang telah di-*resize* sebelumnya sesuai dengan panduan penggunaan aplikasi. Proses prediksi dilakukan menggunakan model TensorFlow Lite yang telah diubah dari model CNN yang telah dilatih sebelumnya.

### 3.2.6 Menampilkan Hasil Prediksi

Hasil akhir dari sistem ini adalah untuk menampilkan kepada pengguna hasil prediksi citra *chest x-ray* atau rontgen dada, memberikan informasi apakah itu normal atau tuberkulosis, dan juga menampilkan tingkat probabilitasnya

## 3.3 Analisis Sistem

Pada tahap analisis sistem, kumpulan dari tools yang digunakan dalam mengembangkan sistem berupa aplikasi android untuk deteksi tuberkulosis dapat dilihat pada Tabel 3.4:

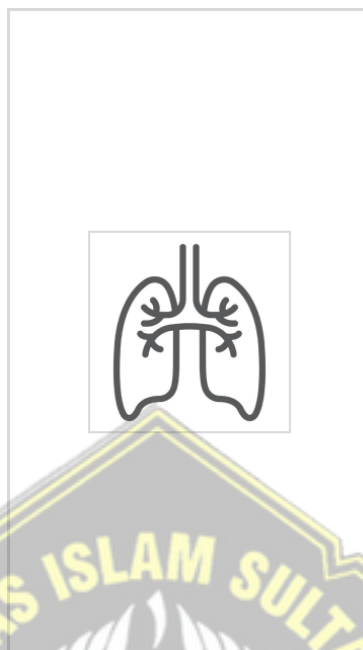
Tabel 3. 4 Analisis Sistem

Tools	Version
Lenovo Ideapad 320-14-AST	Windows 11
Samsung Galaxy A24	Android 13
Google Colaboratory	T4 GPU
Android Studio	Flamingo
Gradle	8.0
Kotlin	1.8

## 3.4 Desain Tampilan Antarmuka

Desain tampilan antarmuka ini merupakan desain *mockup* yang akan diaplikasikan ke dalam sistem yang dibuat dengan figma. Berikut adalah UI design dari Sistem Deteksi Untuk Penyakit Tuberkulosis Berbasis *Chest X-Ray* Dengan Metode *Convolutional Neural Network* Pada Platform Android

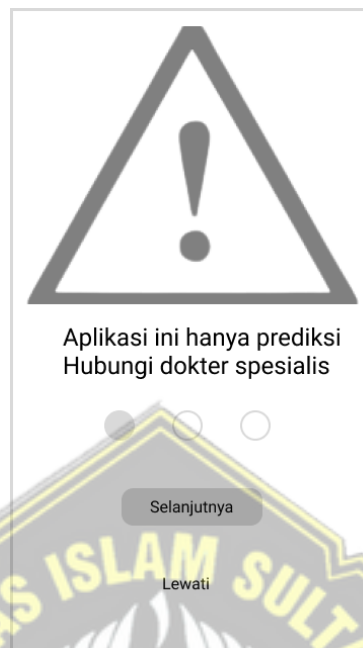
### 3.4.1 SplashScreen



Gambar 3. 10 *SplashScreen*

Gambar 3.10 adalah *mockup* dari *user interface* pada bagian *SplashScreen*. *Splashscreen* adalah layar/tampilan pertama kali saat pengguna membuka sebuah aplikasi. Pada rancangan *splashscreen* ini memuat gambar atau *icon* dari penyakit tuberkulosis.

### 3.4.2 Onboarding

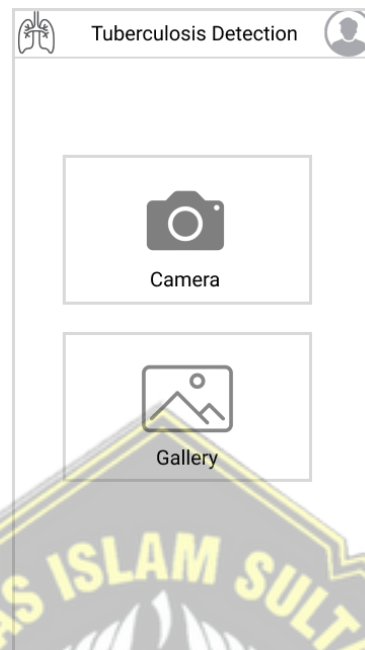


Gambar 3.11 *Onboarding*

Gambar 3.11 adalah *mockup* dari *user interface* pada bagian *Onboarding*. *Onboarding* adalah layar/tampilan panduan tata cara dan fitur fitur yang ada pada sebuah aplikasi. Pada rancangan *onboarding* ini memuat 3 halaman diantara lain:

1. Peringatan: Aplikasi ini hanya prediksi, hubungi dokter spesialis lebih lanjut
2. Citra : Citra yang digunakan harus berupa *chest x-ray* atau rontgen dada
3. Fitur : Fitur citra bisa diambil dari galeri atau kamera langsung

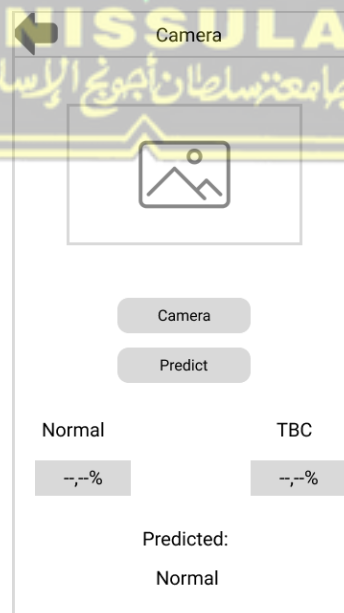
### 3.4.3 Home



Gambar 3. 12 Home

Gambar 3.12 adalah *mockup* dari *user interface* pada bagian *Home*. *Home* adalah layar/tampilan utama yang biasanya menjadi penghubung dengan menu lain pada sebuah aplikasi. Pada rancangan *home* ini terdapat 2 *button* yaitu *camera* atau *gallery* dan 1 *menu* yaitu *menu profile*.

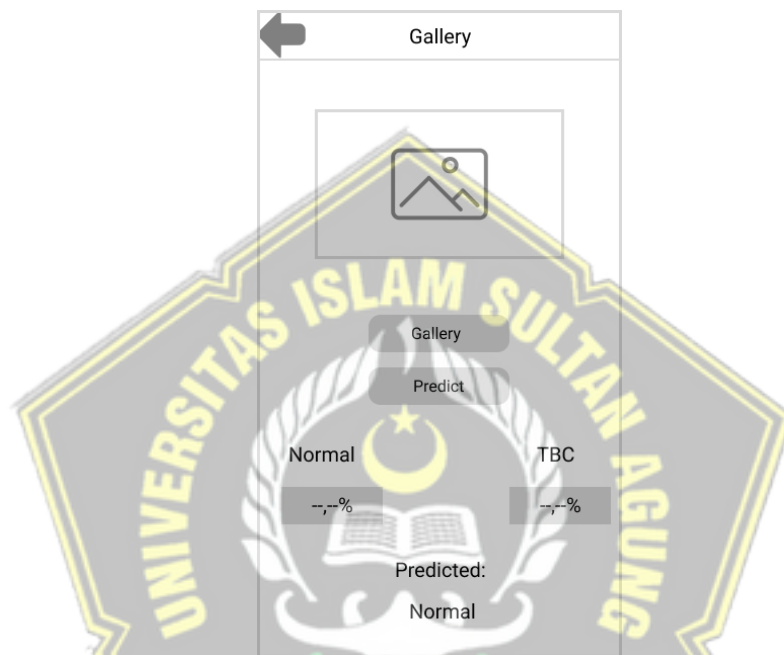
### 3.4.4 Camera



Gambar 3. 13 Camera

Gambar 3.13 adalah *mockup* dari *user interface* pada bagian *Camera*. *Camera* adalah layar/tampilan mengambil gambar, menampilkan gambar, dan prediksi yang berhubungan dengan kamera. Pada rancangan *camera* ini juga akan ditampilkan presentasi bobot tiap kelas dari gambar *chest x-ray* yang diambil langsung dari kamera

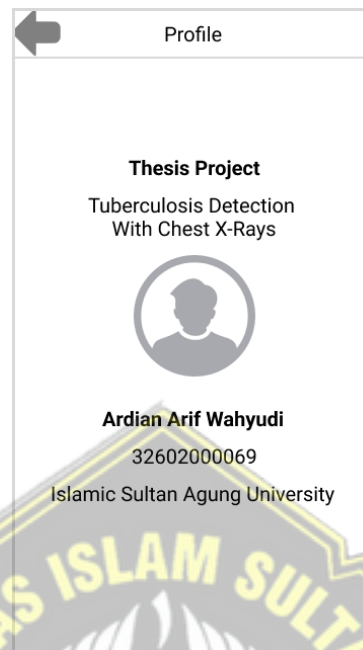
### 3.4.5 Gallery



Gambar 3. 14 *Gallery*

Gambar 3.14 adalah *mockup* dari *user interface* pada bagian *Gallery*. *Gallery* adalah layar/tampilan mengambil gambar, menampilkan gambar, dan prediksi yang berhubungan dengan galeri. Pada rancangan *camera* ini juga akan ditampilkan presentasi bobot tiap kelas dari gambar *chest x-ray* yang diambil dari galeri atau album.

### 3.4.6 Profile



Gambar 3. 15 *Profile*

Gambar 3.15 adalah *mockup* dari *user interface* pada bagian *Profile*. *Profile* adalah layar/tampilan yang menampilkan informasi tentang profil baik tentang aplikasi itu sendiri maupun pembuat aplikasi. Pada rancangan *profile* ini menampilkan tentang deskripsi aplikasi dan asal usul pembuat aplikasi.

## BAB IV

### HASIL DAN ANALISIS PENELITIAN

#### 4.1 Training Model

```

[1] import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, AveragePooling2D, Flatten, Dense, Dropout
from keras.callbacks import EarlyStopping

[2] # Mengatur folder path dari Google Drive
from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

[3] # Path ke dataset di Google Drive
base_path = "/content/gdrive/MyDrive/Dataset Tuberkulosis"
train_path = os.path.join(base_path, "Train")
test_path = os.path.join(base_path, "Test")
val_path = os.path.join(base_path, "Val")

[4] # Inisialisasi generator untuk training data
train_datagen = ImageDataGenerator(
    rescale=1./255, # Normalisasi nilai pixel menjadi 0-1
    shear_range=0.2, # Pergeseran sudut gambar

```

Gambar 4. 1 *Training Model*

Gambar 4.1 adalah *source code* dari pelatihan model menggunakan Google Colab. Pada gambar tersebut, kita dapat melihat antarmuka Google Colab yang menunjukkan langkah-langkah pelatihan model, termasuk kode program, output, dan grafik performa. Proses tersebut mencerminkan bagaimana Colab menyediakan lingkungan cloud computing untuk melatih model machine learning tanpa memerlukan pengaturan keras atau perangkat keras lokal. Pelatihan tersebut menghasilkan 3 model yang berbeda menggunakan konfigurasi 1, konfigurasi 2 dan konfigurasi 3. Perbedaan konfigurasi tersebut menghasilkan sebuah future maps yang berbeda juga, perbandingan feature maps antar konfigurasi terdapat pada Tabel 4.1 berikut.

Tabel 4. 1 Perbandingan Feature Maps

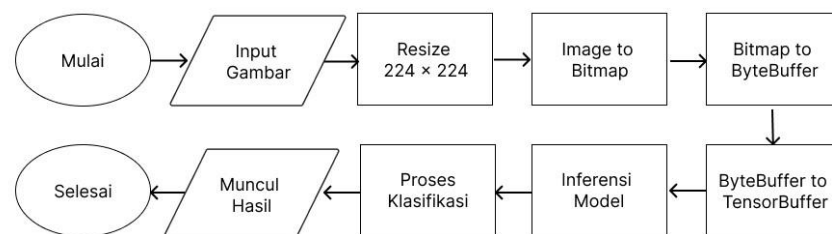
Layer	Konfigurasi 1	Konfigurasi 2	Konfigurasi 3
Input Layer	224x224x3	224x224x3	224x224x3
Conv2D_1	222x222x8	222x222x16	222x222x32



Layer	Konfigurasi 1	Konfigurasi 2	Konfigurasi 3
MaxPool_1	111x111x8	111x111x16	111x111x32
Conv2D_2	109x109x16	109x109x32	109x109x64
MaxPool_2	54x54x16	54x54x32	54x54x64
Conv2D_3	52x52x32	52x52x64	52x52x128
MaxPool_3	26x26x32	26x26x64	26x26x128
Conv2D_4	24x24x64	24x24x128	24x24x256
MaxPool_4	12x12x64	12x12x128	12x12x256
Flatten	1x9216	1x18432	1x36864
Dropout	-	-	-
Dense	1x128	1x256	1x512
Dense	1x2	1x2	1x2

Tabel 4.1 menunjukan feature maps model dari ketiga konfigurasi yang telah di training. Pada setiap operasi konvolusi dengan filter/kernel berukuran 3x3 akan mengurangi resolusi 2 piksel pada setiap panjang dan lebarnya karena setiap bagi tepi tidak ikut terkena operasi konvolusi. Pada operasi pooling layer menggunakan max pooling dengan ukuran 2x2 yang akan mengurangi ukuran menjadi setengahnya. Jumlah neuron saat di flatten tergantung seberapa banyak jumlah filter/kernel, semakin banyak jumlah filter/kernel maka neuron akan semakin banyak.

#### 4.2 Cara Kerja Sistem Klasifikasi



Gambar 4. 2 *Flowchart* Klasifikasi

Pada gambar 4.2 merupakan *flowchart* dari klasifikasi yang dilakukan oleh sistem. Berikut ini penjelasan lebih lengkap tiap bagian dari *flowchart* klasifikasi sebagai berikut:

1. Input Gambar : *User* akan melakukan *input* gambar yang dipilih oleh kamera maupun galeri. Format gambar umum biasanya diantaranya .jpg, .png, .jpeg dan sebagainya.
2. Resize 224x224 : Gambar yang telah diinputkan akan diresize ke 224x224 agar ke depan *ByteBuffer* yang ada gambar sesuai *ByteBuffer* yang diminta oleh model. Apabila *ByteBuffer* gambar tidak cocok oleh *ByteBuffer* yang diminta model baik kelebihan maupun kekurangan maka akan *force close*.
3. Image to Bitmap : Gambar yang sudah beresolusi 224x224 akan dikonversi menjadi *Bitmap*. *Image* merupakan konsep umum untuk segala jenis representasi visual, sedangkan *Bitmap* merujuk pada representasi khusus dari gambar di mana setiap elemen atau piksel pada gambar direpresentasikan secara langsung oleh data yang menggambarkan warna atau kecerahan pada posisi tersebut. *Bitmap* diperlukan karena model memerlukan input dalam bentuk matriks piksel, yang sesuai dengan representasi *Bitmap* di tingkat android.
4. Bitmap to ByteBuffer : *Bitmap* yang sudah ada kemudian dikonversi menjadi *ByteBuffer*. Hal itu disebabkan karena model memerlukan data input dalam bentuk array byte atau struktur data tertentu yang tidak langsung diperoleh dari objek *Bitmap*. Oleh karena itu, perlu mengonversi *Bitmap* ke *ByteBuffer* yang dikenal juga lebih hemat memori sehingga membantu membantu mengoptimalkan penggunaan memori karena kita dapat mengontrol cara data gambar direpresentasikan dan memastikan hanya informasi yang diperlukan oleh model yang disertakan.
5. *ByteBuffer* to *TensorBuffer* : Setelah menjadi *ByteBuffer* maka perlu mensalin *ByteBuffer* ke *TensorBuffer* agar sesuai dengan format yang dapat dipahami oleh model, sehingga *TensorBuffer* dapat digunakan sebagai input untuk model TensorFlow Lite.
6. Inferensi Model : Inferensi adalah proses penggunaan model yang sudah diinisialisasi untuk membuat prediksi atau menghasilkan output berdasarkan data *input*. Mendapatkan hasil atau prediksi dari model berdasarkan data *input* tertentu

7. Proses Klasifikasi : Setelah proses inferensi, model tflite akan bekerja melakukan klasifikasi. Hasil klasifikasi oleh model merupakan TensorBuffer berformat float.
8. Muncul Hasil : Hasil float `outputFeature0AsTensorBuffer[0]` merupakan hasil probabilitas kelas “Normal” sedangkan `outputFeature0AsTensorBuffer[1]` merupakan hasil probabilitas kelas “Tuberculosis”. Cara mendapatkan kelas terprediksi adalah membuat fungsi membandingkan jumlah `outputFeature0AsTensorBuffer[0]` dengan `outputFeature0AsTensorBuffer[1]` yang memiliki nilai lebih tinggi termasuk kelas yang terprediksi.

### 4.3 Implementasi User Interface

#### 4.3.1 Halaman SplashScreen

Pada halaman *splash screen* ini, merupakan halaman pertama yang akan dilihat oleh pengguna saat pertama membuka aplikasi, dimana terdapat logo atau *icon* paru paru dengan terdapat text tuberculosis dibawahnya.



Gambar 4. 3 Halaman Splash Screen

Pada Gambar 4.3 adalah tampilan halaman *splash screen* dari aplikasi yang dibuat. Waktu menunggu yang diperlukan adalah 4 detik agar aplikasi bisa melanjutkan ke halaman *onboarding*.

### 4.3.2 Halaman Onboarding

Pada halaman *onboarding* ini, merupakan halaman yang akan muncul setelah splash screen, dimana ada 3 slide halaman yang akan petunjuk maupun fitur yang tersedia pada aplikasi.



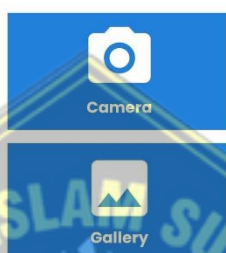
Gambar 4. 4 Halaman *Onboarding*

Pada Gambar 4.4 adalah tampilan halaman *onboarding* dari aplikasi yang buat. Pada halaman *onboarding* ini kita bisa menggeser setiap halaman untuk membacanya atau bisa melewatinya langsung. Pada halaman *onboarding* ini memuat 3 halaman diantara lain:

1. Peringatan: Aplikasi ini hanya prediksi, hubungi dokter spesialis lebih lanjut. Hal tersebut menegaskan bahwa tidak mempercayai 100% aplikasi deteksi ini, lebih baik konsultasikan hasil dengan dokter tentang hasil x-ray kamu.
2. Citra : Citra yang digunakan harus berupa *chest x-ray* atau rontgen dada. Hal tersebut merupakan petunjuk citra yang harus dimasukkan ke sistem harus berupa *x-ray* atau rontgen lebih tepatnya bagian dada.
3. Fitur : Fitur citra bisa diambil dari galeri atau kamera langsung Hal tersebut merupakan penjelasan fitur untuk meng-*upload* gambar ke sistem untuk di deteksi bisa melalui galeri dari foto chest x-ray yang telah tersimpan atau memotret gambar langsung dari kamera.

### 4.3.3 Halaman Home

Pada halaman *home* ini, merupakan halaman yang akan muncul setelah *onboarding*, dimana halaman home ini merupakan halaman utama dan juga pusat navigasi ke halaman halaman lainnya.



Gambar 4. 5 Halaman *Home*

Pada Gambar 4.5 adalah tampilan halaman *home* dari aplikasi yang buat. Pada halaman *home* ini kita dapat terhubung ke halaman *camera*, *gallery*, dan *realtime detection*. Pada bagian *toolbar* atas *icon profile* yang apabila kita klik akan terhubung ke menu *profile* untuk melihat *profile* pembuat aplikasi.

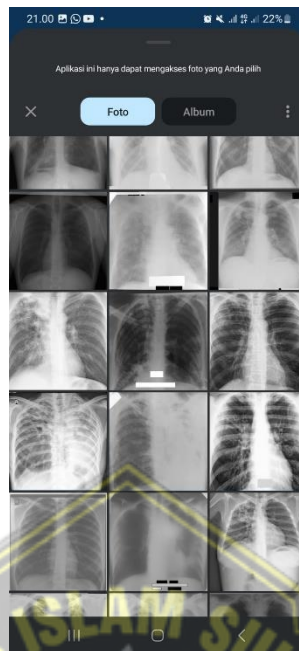
#### 4.3.4 Halaman Gallery

Pada halaman *gallery* ini, merupakan halaman yang akan muncul setelah *button gallery* di halaman *home* di klik, dimana halaman *gallery* inilah merupakan proses klasifikasi gambar *chest x-ray* dilakukan.



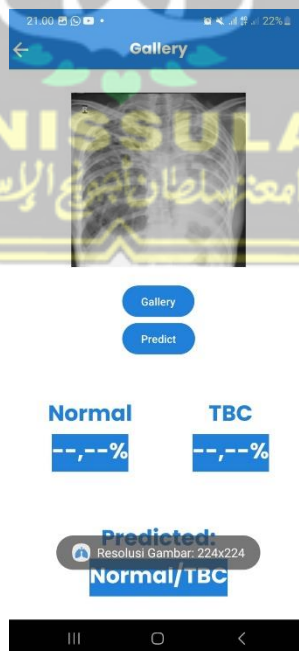
Gambar 4. 6 Halaman *Gallery*

Pada Gambar 4.6 adalah tampilan awal halaman *gallery* dari aplikasi yang buat. Pada tampilan tersebut merupakan tampilan saat belum diinputkan gambar maupun diklik prediksi sehingga kelas yang terprediksi dan probabilitas tiap kelasnya masih kosong.



Gambar 4. 7 Pilih Gambar Dari Gallery

Pada Gambar 4.7 adalah tampilan ketika *button gallery* di klik pada halaman *gallery*. Dalam tahap ini maka *user* akan diminta memilih foto *chest x-ray* dari *gallery*. *User* juga bisa memilih foto dari album yang telah dikategorikan sebelumnya untuk mempermudah pencarian.



Gambar 4. 8 Preview Gambar Halaman Gallery

Pada Gambar 4.8 adalah tampilan ketika gambar dari galeri berhasil dipilih dan ditampilkan pada *preview*. Terdapat juga *toast* tentang “Resolusi Gambar: 224x224” yang menunjukkan bahwa *function resize* yang ada di aplikasi berhasil dijalankan, jadi ukuran gambar berapapun yang di-*input* akan di-*resize* ke 224x224.



Gambar 4. 9 Hasil Prediksi *Gallery*

Pada Gambar 4.9 adalah tampilan ketika *button predict* ditekan Model akan melakukan klasifikasi gambar yang telah di-*resize* sebelumnya. Hasil dari prediksi tersebut adalah kelas yang terprediksi yaitu kelas “Normal” atau “Tuberculosis”. Pada hasil juga ditampilkan probabilitas tiap kelasnya, yaitu 24,70% kelas normal dan 75,30% kelas tuberkulosis. Tuberkulosis akan tampil sebagai kelas yang terprediksi karena memiliki probabilitas lebih besar.



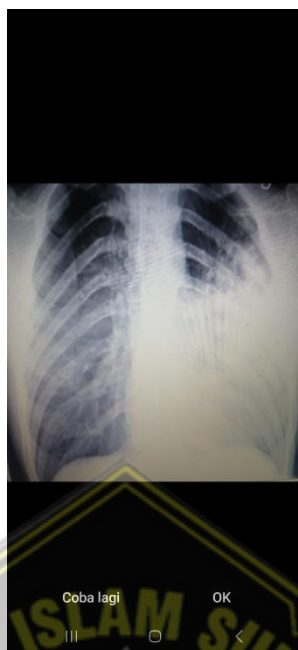
### 4.3.5 Halaman Camera

Pada halaman *camera* ini, merupakan halaman yang akan muncul setelah *button camera* di halaman *camera* di klik, dimana halaman *camera* inilah merupakan proses klasifikasi gambar *chest x-ray* dilakukan.



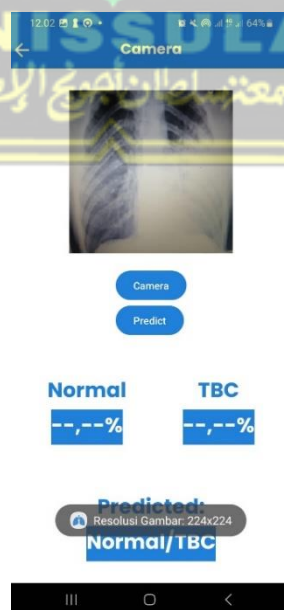
Gambar 4. 10 Halaman *Camera*

Pada Gambar 4.10 adalah tampilan awal halaman *camera* dari aplikasi yang buat. Pada tampilan tersebut merupakan tampilan saat belum diinputkan gambar maupun diklik dan probabilitas tiap kelasnya masih kosong.



Gambar 4. 11 Ambil Gambar Dari Kamera

Pada Gambar 4.11 adalah tampilan ketika *button camera* di klik pada halaman *gallery*. Dalam tahap ini aplikasi akan beralih ke kamera dan *user* akan diminta mengambil foto *chest x-ray*. Pada pengaturan foto di kamera perhatikan tips penggunaan aplikasi agar foto beratio 1:1 agar menghindari foto gepeng saat di-*resize*. Gambar yang menjadi gepeng akan memengaruhi hasil prediksi yang dilakukan oleh model



Gambar 4. 12 Preview Gambar Halaman Camera

Pada Gambar 4.12 adalah tampilan ketika gambar dari kamera berhasil dipilih dan ditampilkan pada *preview*. Terdapat juga *toast* tentang “Resolusi Gambar: 224x224” yang menunjukkan bahwa *function resize* yang ada di aplikasi berhasil dijalankan, jadi ukuran gambar berapapun yang di-*input* akan di-*resize* ke 224x224.



Gambar 4. 13 Hasil Prediksi *Camera*

Pada Gambar 4.13 adalah tampilan ketika *button predict* ditekan Model akan melakukan klasifikasi gambar yang telah di-*resize* sebelumnya. Hasil dari prediksi tersebut adalah kelas yang terprediksi yaitu kelas “Normal” atau “Tuberculosis”. Pada hasil juga ditampilkan probabilitas tiap kelasnya, yaitu 0,39% kelas normal dan 99,61% kelas tuberkulosis. Tuberkulosis akan tampil sebagai kelas yang terprediksi karena memiliki probabilitas lebih besar.

### 4.3.6 Halaman Profile

Pada halaman *profile* ini, merupakan halaman yang akan muncul setelah *icon profile* di halaman *home* dibagian *toolbar* di klik, dimana halaman ini berisi tentang project aplikasi dan pembuatnya.



Gambar 4. 14 Halaman *Profile*

Pada Gambar 4.14 adalah tampilan halaman profile dari aplikasi yang telah dibuat. Tampilan profile menunjukkan deskripsi project yaitu sebuah “Thesis Project” yang merupakan sebuah project skripsi. Judul skripsi tersebut adalah “Tuberculosis Detection with Chest X-Rays” dan pembuat aplikasi yaitu “Ardian Arif Wahyudi” yang merupakan mahasiswa dari “Universitas Islam Sultan Agung” yang memiliki nomor induk mahasiswa “32602000069”

## 4.4 Black Box Testing

Pada tahap pengujian aplikasi ini dilakukan menggunakan metode Black Box Testing. Pengujian *Black Box*, juga dikenal sebagai pengujian perilaku, dilakukan setelah pembuatan perangkat lunak selesai untuk memastikan bahwa itu berfungsi dengan baik. Pengujian ini dilakukan tanpa memahami struktur kode perangkat lunak. Jenis pengujian black box testing yang dipilih ialah functional testing atau berdasarkan fungsinya yang dibisa dilihat pada Tabel 4.2 berikut ini:

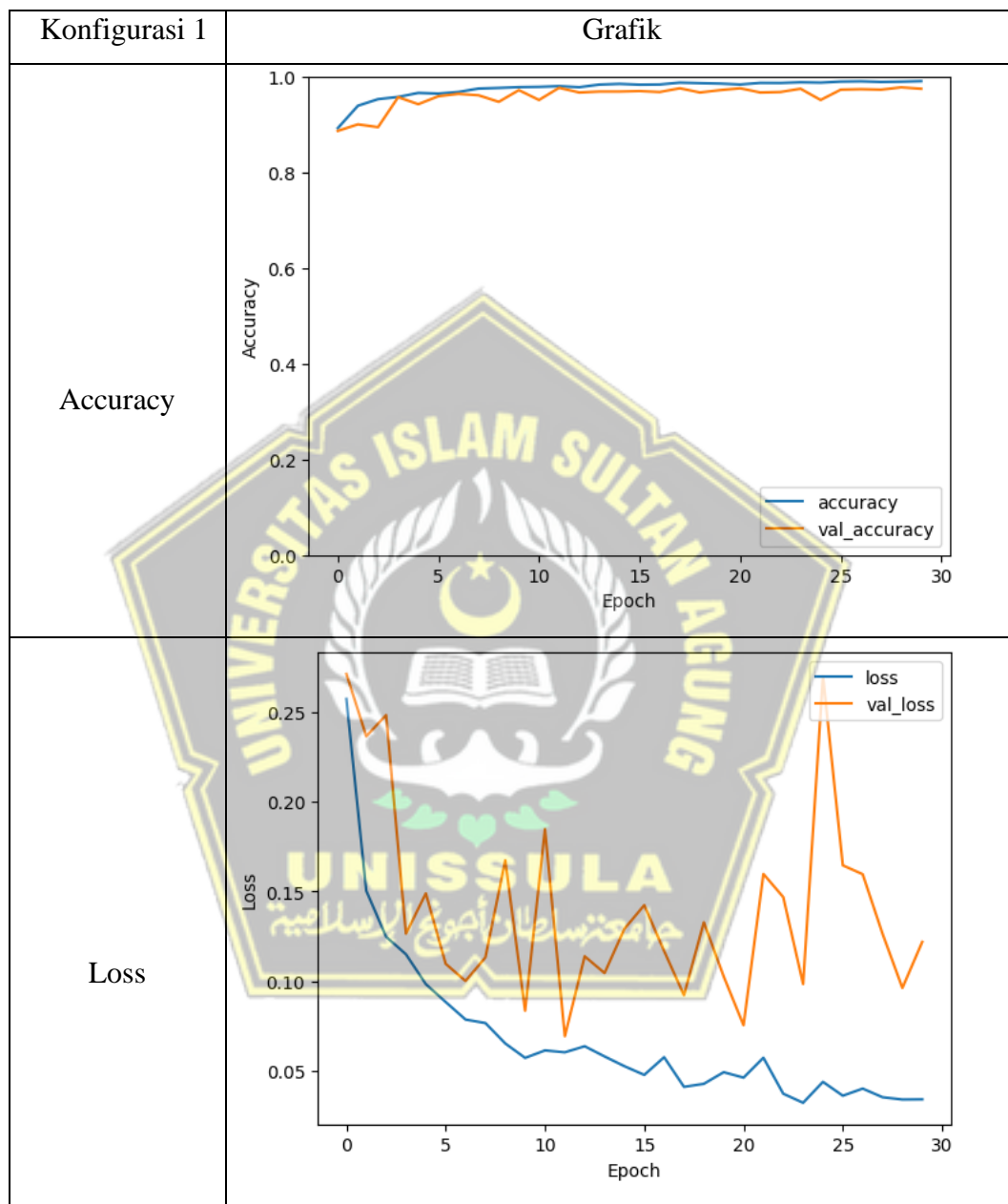
Tabel 4. 2 *Functional BlackBox Testing*

No	Skenario Pengujian	Ekspetasi Hasil	Hasil Pengujian	Kesimpulan
1	Membuka Semua Halaman Aplikasi	Semua Halaman Bisa Diklik	Semua Halaman Bisa Diklik	Terpenuhi
2	Mengambil Foto Dari Galeri atau Kamera	Foto Berhasil Diinput	Foto Berhasil Diinput	Terpenuhi
3	Menampilkan Preview Foto Dari Galeri atau Kamera	Foto Berhasil Ditampilkan	Foto Berhasil Ditampilkan	Terpenuhi
4	Resize Gambar ke 224x224	Toast Menampilkan Resolusi 224x224	Toast Menampilkan Resolusi 224x224	Terpenuhi
5	Melakukan Prediksi Saat Ada Gambar	Muncul Hasil dan Probabilitas	Muncul Hasil dan Probabilitas	Terpenuhi
6	Melakukan Prediksi Saat Belum Ada Gambar	Akan Muncul Peringatan Harus Memasukan Gambar	Akan Muncul Peringatan Harus Memasukan Gambar	Terpenuhi
7	Melakukan Prediksi Data Validasi Normal	Terprediksi Normal	Terprediksi Normal	Terpenuhi
8	Melakukan Prediksi Data Validasi Tuberkulosis	Terprediksi Tuberkulosis	Terprediksi Normal	Terpenuhi

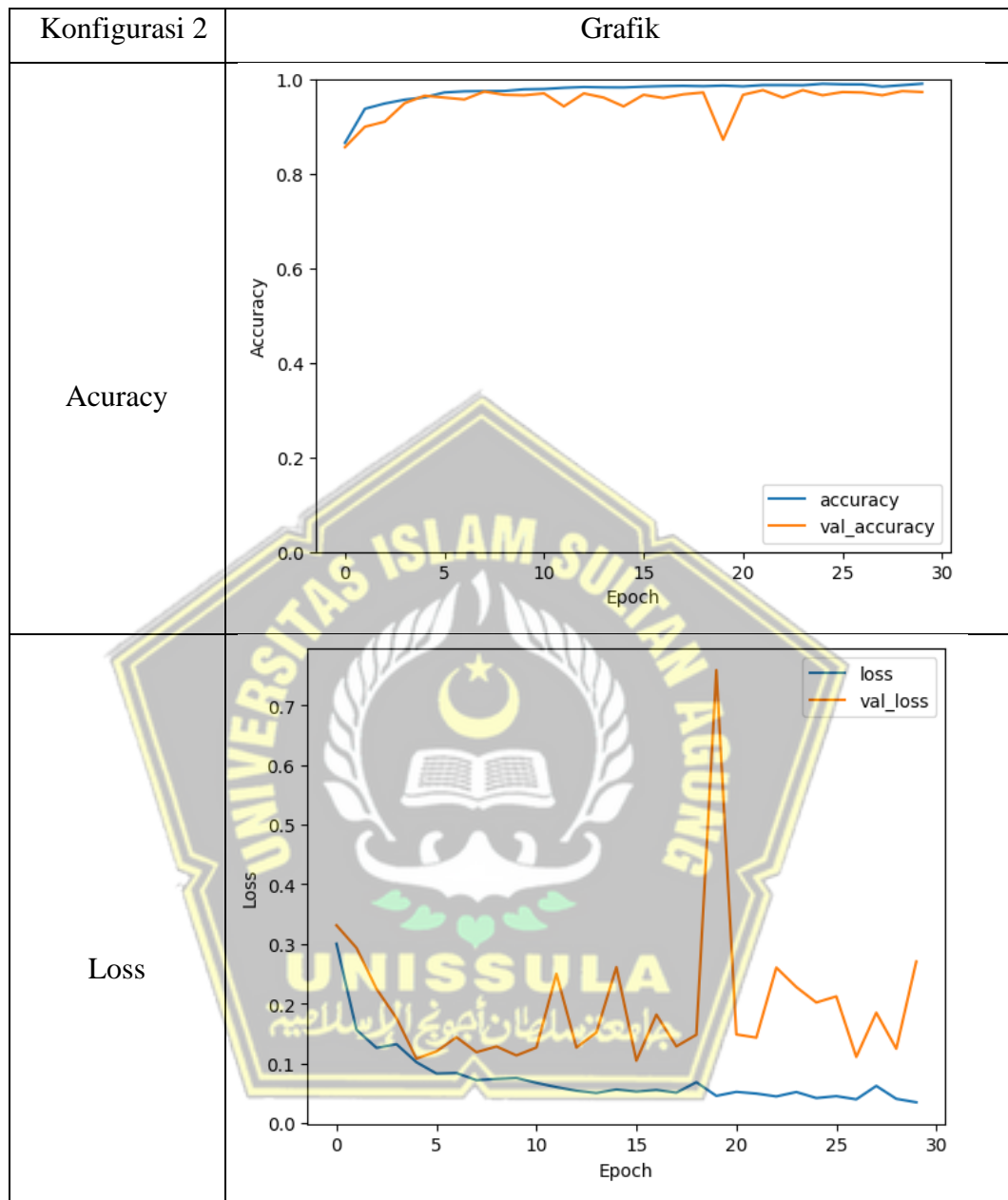
## 4.5 Hasil dan Analisis

### 4.5.1 Training dan Validation

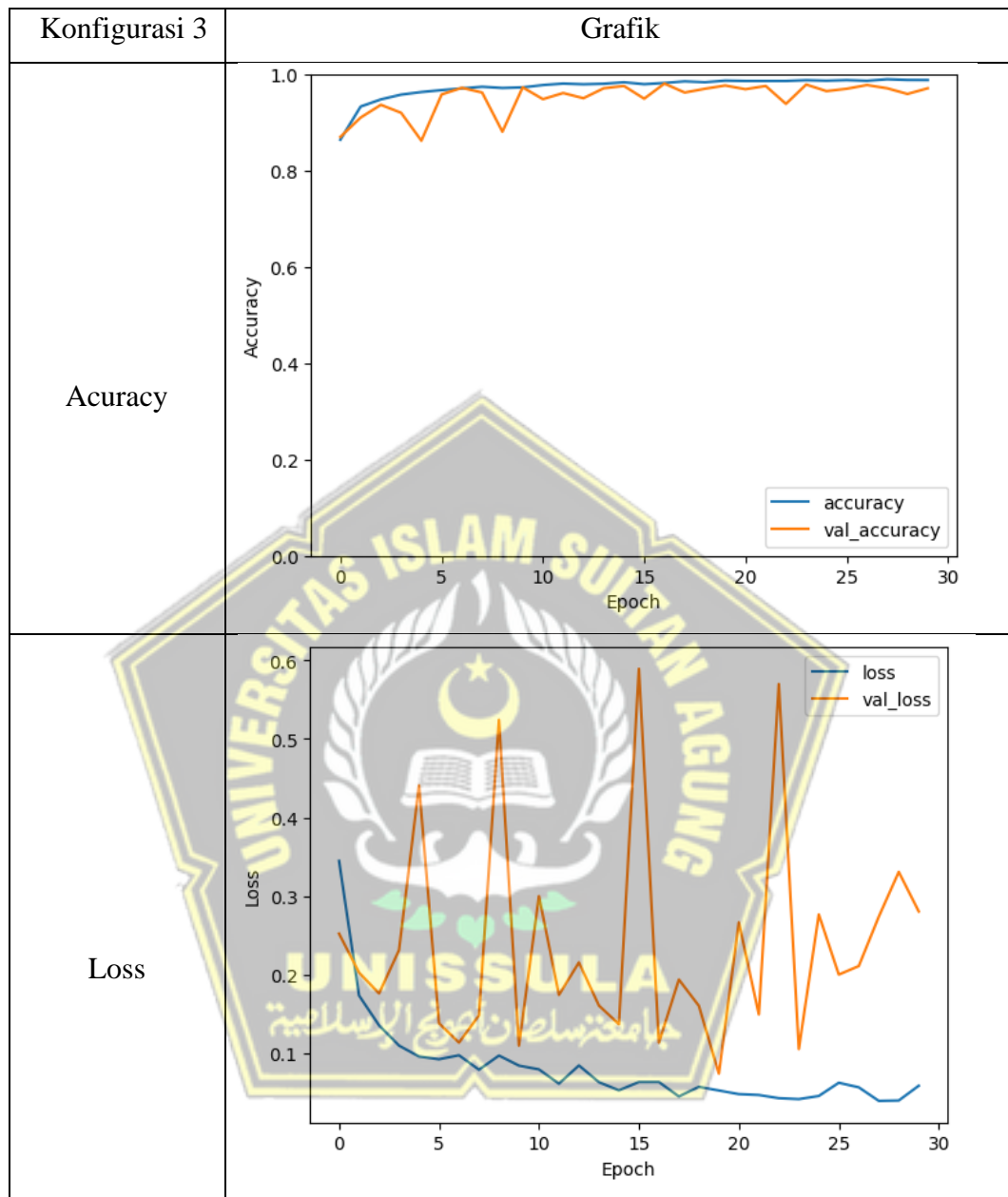
Tabel 4. 3 Plot *Accuracy* dan *Loss* Konfigurasi 1



Tabel 4.3 menunjukkan plot *accuracy* dan *loss* pada konfigurasi 1. Pada plot *accuracy* menunjukkan tren selalu naik pada *train accuracy* sedangkan *val accuracy* juga menunjukkan tren naik walau terkadang pada beberapa epoch mengalami *drop*. Pada plot *loss* menunjukkan tren selalu turun *train loss* sedangkan *val loss* juga menunjukkan tren turun walau terkadang pada beberapa epoch mengalami kenaikan.

Tabel 4. 4 Plot *Accuracy* dan *Loss* Konfigurasi 2

Tabel 4.4 menunjukkan plot *accuracy* dan *loss* pada konfigurasi 2. Pada plot *accuracy* menunjukkan tren selalu naik pada *train accuracy* sedangkan *val accuracy* juga menunjukkan tren naik walau tetapi memiliki drop yang lebih banyak daripada konfigurasi 1. Pada plot *loss* menunjukkan tren selalu turun pada *train loss* sedangkan *val loss* menunjukkan naik turun tetapi memiliki kenaikan *loss* yang lebih banyak daripada konfigurasi 1.

Tabel 4. 5 Plot *Acuracy* dan *Loss* Konfigurasi 3

Tabel 4.5 menunjukkan plot *accuracy* dan *loss* pada konfigurasi 3. Pada plot *accuracy* menunjukkan tren selalu naik pada *train accuracy* sedangkan *val accuracy* juga menunjukkan tren naik walau tetapi memiliki drop yang lebih banyak daripada konfigurasi 2. Pada plot *loss* menunjukkan tren turun pada *train loss* sedangkan *val loss* menunjukkan tren naik turun tetapi memiliki kenaikan *loss* yang lebih banyak daripada konfigurasi 2.



## 4.5.2 Testing

Tabel 4. 6 *Confusion Matrix* Konfigurasi 1-3

Konfigurasi	TP	TN	FP	FN	Bukti Pengujian
Konfigurasi 1	169	169	1	1	⇒ Confusion Matrix: [[169 1] [ 1 169]]
Konfigurasi 2	168	165	2	5	⇒ Confusion Matrix: [[168 2] [ 5 165]]
Konfigurasi 3	167	165	3	5	⇒ Confusion Matrix: [[167 3] [ 5 165]]

Tabel 4.6 merupakan tabel hasil *confusion matrix* dari setiap pengujian konfigurasi 1-3. Nilai *true* berarti prediksi benar sedangkan *false* salah memprediksi. Nilai *positive* merupakan nilai dari kelas tuberkulosis, sedangkan nilai *negative* nilai dari kelas normal. Hasil dari TP, TN, FP, dan FN akan dipakai untuk menentukan nilai akurasi, presisi, *recall* dan *f-1 score*.

### 1. Evaluasi Model Konfigurasi 1

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} = \frac{169+169}{169+1+1+169} = \frac{338}{340} = 0,994$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{169}{169+1} = \frac{169}{170} = 0,994$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{169}{169+1} = \frac{169}{170} = 0,994$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * 0,99 * 0,99}{0,99 + 0,99} = \frac{1,9602}{1,98} = 0,99$$

### 2. Evaluasi Model Konfigurasi 2

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} = \frac{168+165}{169+1+1+169} = \frac{333}{340} = 0,979$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{168}{168+2} = \frac{168}{170} = 0,988$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{168}{168+5} = \frac{168}{173} = 0,971$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * 0,99 * 0,97}{0,99 + 0,97} = \frac{1,9206}{1,96} = 0,979$$

## 3. Evaluasi Model Konfigurasi 3

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} = \frac{167+165}{169+1+1+169} = \frac{332}{340} = 0,976$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{167}{167+3} = \frac{167}{170} = 0,982$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{167}{167+5} = \frac{167}{172} = 0,97$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * 0,98 * 0,97}{0,98 + 0,97} = \frac{1,9012}{1,95} = 0,9749$$

Tabel 4. 7 Accuracy, Precision, Recall, dan F1-Score Konfigurasi 1-3

Konfigurasi	Accuracy, Precision, Recall, dan F1-Score																														
Konfigurasi 1	<p>Classification Report:</p> <table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Normal</td> <td>0.99</td> <td>0.99</td> <td>0.99</td> <td>170</td> </tr> <tr> <td>Tuberkulosis</td> <td>0.99</td> <td>0.99</td> <td>0.99</td> <td>170</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.99</td> <td>340</td> </tr> <tr> <td>macro avg</td> <td>0.99</td> <td>0.99</td> <td>0.99</td> <td>340</td> </tr> <tr> <td>weighted avg</td> <td>0.99</td> <td>0.99</td> <td>0.99</td> <td>340</td> </tr> </tbody> </table> <p>Accuracy: 0.99</p>		precision	recall	f1-score	support	Normal	0.99	0.99	0.99	170	Tuberkulosis	0.99	0.99	0.99	170	accuracy			0.99	340	macro avg	0.99	0.99	0.99	340	weighted avg	0.99	0.99	0.99	340
	precision	recall	f1-score	support																											
Normal	0.99	0.99	0.99	170																											
Tuberkulosis	0.99	0.99	0.99	170																											
accuracy			0.99	340																											
macro avg	0.99	0.99	0.99	340																											
weighted avg	0.99	0.99	0.99	340																											
Konfigurasi 2	<p>Classification Report:</p> <table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Normal</td> <td>0.97</td> <td>0.99</td> <td>0.98</td> <td>170</td> </tr> <tr> <td>Tuberkulosis</td> <td>0.99</td> <td>0.97</td> <td>0.98</td> <td>170</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.98</td> <td>340</td> </tr> <tr> <td>macro avg</td> <td>0.98</td> <td>0.98</td> <td>0.98</td> <td>340</td> </tr> <tr> <td>weighted avg</td> <td>0.98</td> <td>0.98</td> <td>0.98</td> <td>340</td> </tr> </tbody> </table> <p>Accuracy: 0.98</p>		precision	recall	f1-score	support	Normal	0.97	0.99	0.98	170	Tuberkulosis	0.99	0.97	0.98	170	accuracy			0.98	340	macro avg	0.98	0.98	0.98	340	weighted avg	0.98	0.98	0.98	340
	precision	recall	f1-score	support																											
Normal	0.97	0.99	0.98	170																											
Tuberkulosis	0.99	0.97	0.98	170																											
accuracy			0.98	340																											
macro avg	0.98	0.98	0.98	340																											
weighted avg	0.98	0.98	0.98	340																											
Konfigurasi 3	<p>Classification Report:</p> <table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Normal</td> <td>0.97</td> <td>0.98</td> <td>0.98</td> <td>170</td> </tr> <tr> <td>Tuberkulosis</td> <td>0.98</td> <td>0.97</td> <td>0.98</td> <td>170</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.98</td> <td>340</td> </tr> <tr> <td>macro avg</td> <td>0.98</td> <td>0.98</td> <td>0.98</td> <td>340</td> </tr> <tr> <td>weighted avg</td> <td>0.98</td> <td>0.98</td> <td>0.98</td> <td>340</td> </tr> </tbody> </table> <p>Accuracy: 0.98</p>		precision	recall	f1-score	support	Normal	0.97	0.98	0.98	170	Tuberkulosis	0.98	0.97	0.98	170	accuracy			0.98	340	macro avg	0.98	0.98	0.98	340	weighted avg	0.98	0.98	0.98	340
	precision	recall	f1-score	support																											
Normal	0.97	0.98	0.98	170																											
Tuberkulosis	0.98	0.97	0.98	170																											
accuracy			0.98	340																											
macro avg	0.98	0.98	0.98	340																											
weighted avg	0.98	0.98	0.98	340																											

Tabel 4.7 merupakan tabel hasil nilai *accuracy*, *precision*, *recall*, dan *f1-score* yang dihitung saat training berdasarkan nilai TP, TN, FP, dan FN pada confusion matrix sebelumnya dari setiap pengujian konfigurasi 1-3. Akurasi perhitungan manual memiliki persamaan dengan hasil pengujian dengan sistem.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Bedasarkan hasil penelitian menunjukkan bahwa sistem deteksi tuberkulosis berbasis citra *chest x-ray* dapat bekerja dengan baik pada platform android. Berikut adalah beberapa kesimpulan yang bisa diambil dari penelitian ini.

1. CNN mampu deteksi tuberkulosis pada citra *chest x-ray* dengan hasil yang cukup baik pada platform android karena mampu belajar mengekstrasi fitur dengan *dataset* yang diberikan saat *training* pada proses konvolusi.
2. Pada tahap *training*, konfigurasi terbaik terdapat pada konfigurasi 1, yang memiliki plot *accuracy* yang terus naik dan *loss* yang terus turun walau serta memiliki drop yang lebih sedikit
3. Pada tahap *testing*, konfigurasi terbaik terdapat pada konfigurasi 1 mendapat nilai 0.99 atau 99% pada akurasi, presisi, *recall* dan *f1-score*.
4. Jumlah kernel pada operasi kovolusi yang lebih banyak tidak menjamin bahwa akurasi yang dihasilkan lebih tinggi

#### 5.2 Saran

Bedasarkan penelitian yang telah dilakukan menunjukkan performa yang cukup baik. Akan tetapi, untuk penelitian selanjutnya peneliti menyarankan:

1. Menggunakan arsitektur *convolutional neural network* yang telah teruji semisal ResNet50, VGG19, Densenet, AlexNet dan sebagainya
2. Mengembangkan pada platform selain android yaitu pada IOS agar bisa berjalanan pada *multiplatform*.

## DAFTAR PUSTAKA

- Alamsyah. (2019). Implementasi Deep Learning Untuk Klasifikasi Tanaman Toga Berdasarkan Ciri Daun Berbasis Android. *Ubiquitous: Computers and Its Applications Journal*, 2(2), 113–122.
- Alwanda dkk. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. In *Jurnal Algoritme* (Vol. 1, Issue 1).
- Bahri dkk. (2021). IJCIT (Indonesian Journal on Computer and Information Technology) Diagnosa Tuberculosis Paru Berbasis Citra X-ray Menggunakan Convolutional Neural Network. In *IJCIT (Indonesian Journal on Computer and Information Technology)* (Vol. 6, Issue 2).
- Christyanto dkk. (2022). *Pengembangan Aplikasi Android Presensi Kehadiran Realtime menggunakan Pengenalan Wajah dengan Model Facenet* (Vol. 6, Issue 10). <http://j-ptiik.ub.ac.id>
- Dewi dkk. (2020). *Gambaran Karakteristik Pasien Tuberculosis di Poliklinik Paru RSUP Sanglah Denpasar*. <https://doi.org/10.24843.MU.2020.V9.i6.P02>
- Fatimah dkk. (2024). *Upaya Peningkatan Pengetahuan Remaja Mengenai Peran Rontgen Toraks Pada Penegakkan Diagnosis Tuberkulosis Paru di SMK Kader Bangsa Palembang*. <https://doi.org/10.37081/adam.v3i1.1727>
- Fraga dkk. (2021). Evaluasi Penggunaan Obat Anti Tuberkulosis Pasien Baru Tuberkulosis Paru di Puskesmas Oebobo Kupang. *Jurnal Farmagazine*, 8(1), 17. <https://doi.org/10.47653/farm.v8i1.530>
- Hozairi dkk. (2021). Implementasi Orange Data Mining Untuk Klasifikasi Kelulusan Mahasiswa Dengan Model K-Nearest Neighbor, Decision Tree Serta Naive Bayes. In *Jurnal Ilmiah NERO* (Vol. 6, Issue 2).
- Mahmud & Faraby. (2019). *Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network*.
- Nugroho dkk. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network ( CNN ) Pada Ekspresi Manusia. *JURNAL ALGOR*, 2(1). <https://jurnal.buddhidharma.ac.id/index.php/algor/index>

- Nurmalasari. (2020). Pemeriksaan Radiografi Thorax dengan Kasus Tuberkulosis Paru. In *KOCENIN Serial Konferensi* (Issue 1).
- Pertiwi dkk. (2021). *Pengaruh Pengawas Minum Obat Terhadap Keberhasilan Pengobatan Pasien Tuberkulosis Paru: A Systematic Review*. 2(4).
- Prayogi dkk. (2021). *Implementasi Reduksi Noise Pada Citra Rontgen Menggunakan Algoritma Arithmetic Mean Filter*. 3(3).
- Rasyid dkk. (2023). Klasifikasi Penyakit Tuberculosis (TB) Organ Paru Manusia Berdasarkan Citra Rontgen Thorax Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Manajemen Informasi Kesehatan Indonesia (JMiki)*, 11(1). <https://doi.org/10.33560/jmiki.v11i1.484>
- Raup, A., Ridwan, W., Khoeriyah, Y., Yuliati Zaqiah, Q., & Islam Negeri Sunan Gunung Djati Bandung, U. (2022). *Deep Learning dan Penerapannya dalam Pembelajaran*. <http://jiip.stkipyapisdompnu.ac.id>
- Rochmawanti dkk. (2021). *Analisis Performa Pre-Trained Model Convolutional Neural Network Dalam Mendeteksi Penyakit Tuberculosis*. <https://doi.org/10.25126/jtiik.202184441>
- Rozaq dkk. (2021). *Rancangan Database Sistem Informasi Program Studi D3 Komunikasi Penerbangan di Politeknik Penerbangan Surabaya*.
- Saputra dkk. (2020). *Identifikasi Mutu Biji Kopi Arabika Berdasarkan Cacat Dengan Teknik Convolutional Neural Network*.
- Siswoyo, B. (2020). MultiClass Decision Forest Machine Learning Artificial Intelligence. In *Journal of Applied Informatics and Computing (JAIC)* (Vol. 4, Issue 1). <http://jurnal.polibatam.ac.id/index.php/JAIC>
- Soedradjat dkk. (2022). *Deteksi Gangguan Paru-Paru Berbasis Citra X-Ray Menggunakan Deep Learning*.
- Suharni dkk. (2021). *Implementasi Model Convolutional Neural Network (CNN) Untuk Klasifikasi Penyakit TBC Berbasis Desktop* (Vol. 16).

## LAMPIRAN

### SISTEM DETEKSI PENYAKIT TUBERKULOSIS BERBASIS CHEST XRAY MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK PADA PLATFORM ANDROID

ORIGINALITY REPORT			
<b>9%</b>	<b>9%</b>	<b>3%</b>	<b>2%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
<b>1</b>	<b>dspace.uii.ac.id</b> Internet Source		<b>1%</b>
<b>2</b>	<b>docplayer.info</b> Internet Source		<b>1%</b>
<b>3</b>	<b>repository.itk.ac.id</b> Internet Source		<b>&lt;1%</b>
<b>4</b>	<b>Submitted to Sultan Agung Islamic University</b> Student Paper		<b>&lt;1%</b>
<b>5</b>	<b>Submitted to STIKOM Surabaya</b> Student Paper		<b>&lt;1%</b>
<b>6</b>	<b>home.simula.no</b> Internet Source		<b>&lt;1%</b>
<b>7</b>	<b>repository.ar-raniry.ac.id</b> Internet Source		<b>&lt;1%</b>
<b>8</b>	<b>id.123dok.com</b> Internet Source		<b>&lt;1%</b>