



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN SISTEM DETEKSI TUBERKULOSIS
PADA CITRA X-RAY DENGAN METODE *CONVOLUTIONAL
NEURAL NETWORK* (CNN) MENGGUNAKAN *FRAMEWORK
LARAVEL***

TUGAS AKHIR

**ALDI AKBAR ALIMY
0110220262**

**PROGRAM STUDI TEKNIK INFORMATIKA
DEPOK
MEI 2024**



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN SISTEM DETEKSI TUBERKULOSIS
PADA CITRA X-RAY DENGAN METODE *CONVOLUTIONAL
NEURAL NETWORK (CNN)* MENGGUNAKAN *FRAMEWORK
LARAVEL***

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

Aldi Akbar Alimi

0110220262

STT - NF

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Aldi Akbar Alimi
NIM : 0110220262

STT - NF
Depok, 10 Agustus 2024
Tanda Tangan



Aldi Akbar Alimi

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Aldi Akbar Alimi

NIM : 0110220262

Program Studi : Teknik Informatika

Judul Skripsi : **Rancang Bangun Sistem Deteksi Tuberkulosis Pada Citra X-Ray Dengan Metode *Convolutional Neural Network* (CNN) Menggunakan *Framework Laravel***

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing


Ahmad Rio Adriansyah, S.Si. M.Si.

Penguji


Pudy Prima, S.T, M.Kom

Ditetapkan di : Depok.....

Tanggal : 24 Juli 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Kedua orang tua dan semua anggota keluarga yang telah memberikan inspirasi serta dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Tiffany Nabarian, S.Kom, M.T.I. selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Bapak Dr. Lukman Rosyidi, S.T., M.M., M.T. selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Ahmad Rio Adriansyah, S.Si., M.Si. selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
7. Ibu Nurul Janah, S.IIP., M.Hum selaku Dosen Pengampu mata kuliah Tugas Akhir yang telah membantu memberikan pengajaran terkait dengan penulisan ilmiah ini.
8. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
9. Perusahaan mitra program MSIB yang menjadi tempat bagi penulis dalam menuangkan ide dan gagasan yang dimiliki oleh penulis selama program MSIB berlangsung.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 24 Juli 2024

Aldi Akbar Alimi



STT - NF

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Aldi Akbar Alimi

NIM : 0110220262

Program Studi : Teknik Informatika

Jenis karya : Skripsi / Tugas Akhir

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*) atas karya ilmiah saya yang berjudul :

Rancang Bangun Sistem Deteksi Tuberkulosis Pada Citra X-Ray Dengan Metode Convolutional Neural Network (CNN) Menggunakan Framework Laravel

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

STT - NF
Dibuat di : Depok, Jawa Barat...
Pada tanggal : 12 Agustus 2024.

Yang Menyatakan



Aldi Akbar Alimi

ABSTRAK

Nama : Aldi Akbar Alimi
NIM : 0110220262
Program Studi : Teknik Informatika
Judul : Rancang Bangun Sistem Deteksi Tuberkulosis Pada Citra X-Ray Dengan Metode Convolutional Neural Network (CNN) Menggunakan Framework Laravel

Penyakit Tuberkulosis atau TBC merupakan penyakit yang diakibatkan oleh bakteri *Mycobacterium Tuberculosis* yang tingkat penularannya cukup tinggi. Penyakit TBC ini dapat didiagnosis melalui beberapa metode, yaitu menggunakan sampel dahak serta menggunakan scan x-ray. Namun, kedua metode tersebut membutuhkan waktu lama dalam proses pendeteksiannya. Oleh karena itu, dibutuhkan sistem deteksi yang dapat mendeteksi penyakit TBC dengan cepat serta dapat dilakukan oleh siapa saja. Pada penelitian ini, dibuat sistem deteksi yang dapat mendeteksi penyakit TBC melalui gambar x-ray bagian dada. Sistem deteksi yang dibuat merupakan aplikasi berbasis *website* yang dibangun menggunakan *framework Laravel* dan model *machine learning* dengan metode *Convolutional Neural Network* (CNN) untuk analisis gambar x-ray. Penelitian ini akan menerapkan model CNN yang telah dibuat ke dalam aplikasi berbasis *website* melalui API yang dibuat menggunakan *framework FastAPI*. Hasil penelitian pada sistem deteksi yang dibuat menunjukkan bahwa sistem deteksi dapat melakukan pendeteksian pada penyakit TBC. Dibuktikan dengan hasil pengujian yang dilakukan menggunakan metode *blackbox testing* didapatkan bahwa hasil pengujian menunjukkan bahwa tingkat keberhasilan pengujian sebesar 87%. Selain itu, model *machine learning* dengan metode CNN juga dapat memberikan klasifikasi pada gambar x-ray dengan baik dengan akurasi yang didapat sebesar 93% pada data latih serta 85% pada data uji.

Kata kunci : Sistem deteksi, TBC, *Laravel*, CNN, *FastAPI*

ABSTRACT

Name : Aldi Akbar Alimi
NIM : 0110220262
Study Program : Informatics Engineering
Title : Design of Tuberculosis Detection System on X-Ray Image with Convolutional Neural Network (CNN) Method Using Laravel Framework

Tuberculosis or TB is a disease caused by the bacteria *Mycobacterium Tuberculosis*, which has a high level of transmission. TB disease can be diagnosed through several methods, namely using sputum samples and using x-ray scans. However, both methods take a long time to detect. Therefore, a detection system is needed that can detect TB disease quickly and can be done by anyone. In this research, a detection system is created that can detect TB disease through chest x-ray images. The detection system is a web-based application built using the Laravel framework and a machine learning model with the Convolutional Neural Network (CNN) method for x-ray image analysis. This research will apply the CNN model that has been made into a web-based application through an API created using the FastAPI framework. The results of research on the detection system show that the detection system can detect TB disease. Proven by the results of testing conducted using the blackbox testing method, the test results show that the test success rate is 87%. In addition, the machine learning model with the CNN method can also provide classification on x-ray images well where an accuracy of 93% is obtained on training data and 85% on test data.

Key words : detection system, TB, *Laravel*, CNN, *FastAPI*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR.....	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vii
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	vii
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan dan Manfaat Penelitian.....	3
1.4. Batasan Masalah.....	4
1.5. Sistematika Penulisan.....	4
BAB II KAJIAN LITERATUR	6
2.1. Tinjauan Pustaka	6
2.1.1. Sistem Deteksi.....	6
2.1.2. Tuberkulosis.....	7
2.1.3. <i>Convolutional Neural Network</i>	8
2.1.4. API & FastAPI.....	12
2.1.5. <i>Laravel Framework</i>	12
2.1.6. <i>Blackbox Testing</i>	13

2.1.7.	<i>User Acceptance Test (UAT)</i>	14
2.1.8.	<i>Skala Likert</i>	14
2.2.	Penelitian Terkait	15
BAB III METODOLOGI PENELITIAN		17
3.1.	Tahapan Penelitian	17
3.1.1.	Studi Pendahuluan.....	17
3.1.2.	Analisis Sistem.....	18
3.1.3.	Perancangan Sistem	18
3.1.4.	Pengembangan Sistem	19
3.1.5.	Pengujian dan Evaluasi	19
3.2.	Rancangan Penelitian	20
3.2.1.	Jenis Penelitian.....	20
3.2.2.	Metode Analisis Data.....	21
3.2.3.	Metode Pengumpulan Data.....	22
3.2.4.	Metode Pengujian.....	22
3.2.5.	Metode Implementasi dan Evaluasi	25
3.2.6.	Lingkungan Pengembangan.....	25
BAB IV IMPLEMENTASI DAN EVALUASI.....		26
4.1.	Pendefinisian Sistem	26
4.1.1.	Kebutuhan Fungsional	26
4.1.2.	Kebutuhan Non-Fungsional	26
4.1.3.	Spesifikasi Teknis	27
4.2.	Desain Sistem	27
4.2.1.	Desain Arsitektur	27
4.2.2.	Desain Antarmuka.....	28
4.2.3.	Desain Alur Kerja	31

4.2.4. Desain Database	32
4.3. Implementasi Sistem	32
4.3.1. API Sistem Deteksi	33
4.3.2. Frontend dan Backend Sistem Deteksi	35
4.4. Pengujian dan Evaluasi Sistem.....	39
4.4.1. Hasil Pengujian	39
4.4.2. Evaluasi Hasil Pengujian.....	43
BAB V KESIMPULAN DAN SARAN	45
5.1. Kesimpulan.....	45
5.2. Saran	46
DAFTAR PUSTAKA	47



STT - NF

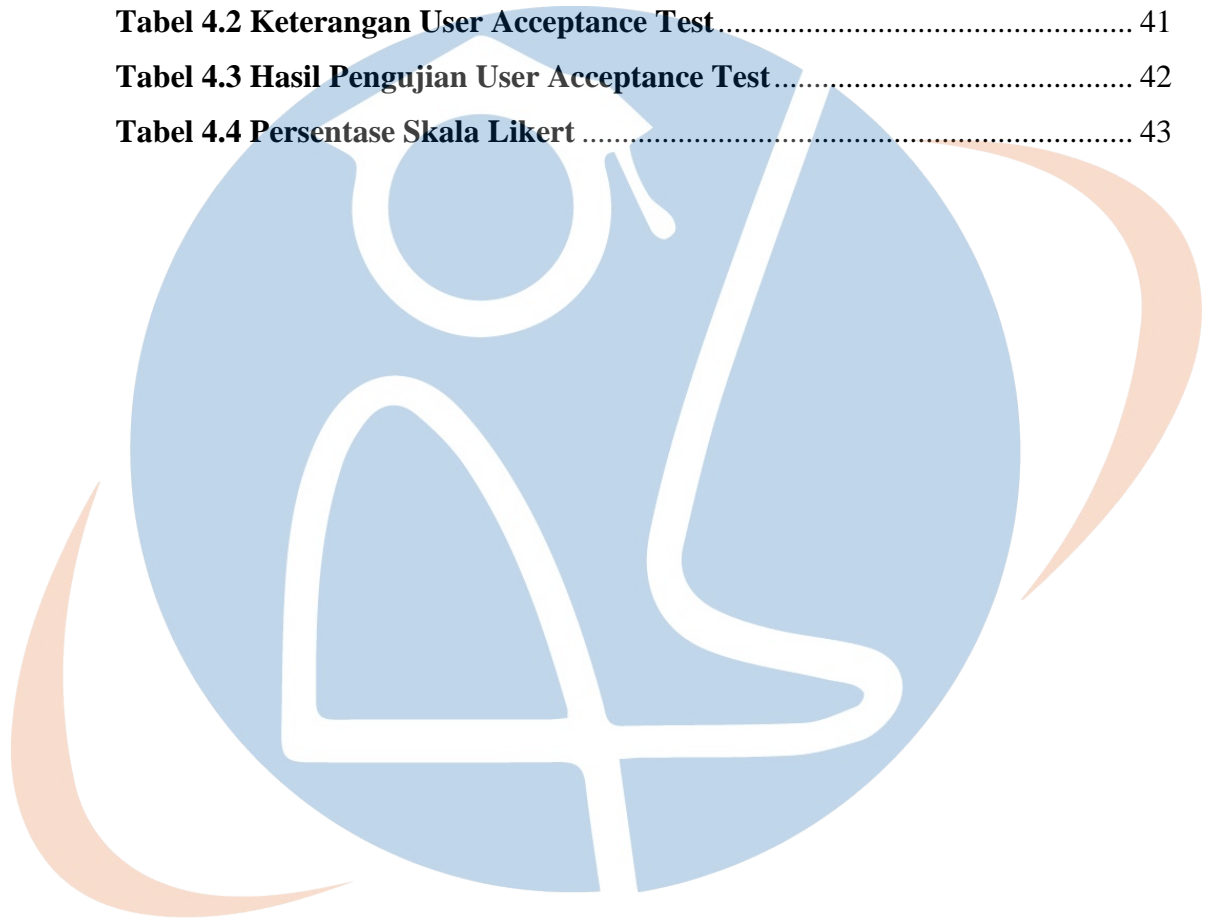
DAFTAR GAMBAR

Gambar 2.1 X-Ray Normal	7
Gambar 2.2 X-Ray TBC	7
Gambar 2.3 Convolutional Neural Network	8
Gambar 2.4 Proses Konvolusi	9
Gambar 2.5 Max Pooling	10
Gambar 2.6 Black Box Testing	13
Gambar 3.1 Tahapan Penelitian	17
Gambar 3.2 Langkah-langkah pengembangan 4D	20
Gambar 4.1 Desain Arsitektur Sistem Deteksi	27
Gambar 4.2 Halaman Utama	28
Gambar 4.3 Halaman Periksa	29
Gambar 4.4 Halaman Hasil Periksa Normal	30
Gambar 4.5 Halaman Hasil Periksa Positif TBC	30
Gambar 4.6 Diagram Alur Kerja Sistem Deteksi	31
Gambar 4.7 Desain Database Sistem Deteksi	32
Gambar 4.8 Pesan grafis berhasil	38
Gambar 4.9 Pesan grafis gagal	37
Gambar 4.10 Hasil Pengujian Kinerja Sistem Deteksi	41

STT - NF

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	15
Tabel 3.1 Skema Pengujian <i>Blackbox Testing</i>	23
Tabel 3.2 Skema Pengujian <i>User Acceptance Test</i>	24
Tabel 4.1 Hasil Pengujian <i>Blackbox Testing</i>	39
Tabel 4.2 Keterangan <i>User Acceptance Test</i>	41
Tabel 4.3 Hasil Pengujian <i>User Acceptance Test</i>	42
Tabel 4.4 Persentase Skala Likert	43



STT - NF

BAB I

PENDAHULUAN

Pada bagian pendahuluan ini, penulis akan menjelaskan mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, serta batasan - batasan masalah yang berkaitan dengan penelitian ini.

1.1. Latar belakang

Tuberkulosis (TBC) merupakan salah satu penyakit berbahaya akibat bakteri *Mycobacterium Tuberculosis* yang tingkat penularannya cukup tinggi. Penyakit ini menjadi tantangan global dan merupakan salah satu penyakit yang diderita oleh banyak masyarakat di seluruh dunia, tanpa terkecuali di Indonesia. Berdasarkan data yang dikeluarkan oleh *World Health Organization* atau WHO menunjukkan bahwa pada tahun 2018 terdapat sebanyak 10 juta orang mengidap penyakit Tuberkulosis, serta 98 ribu orang di antaranya meninggal dunia. Hal ini menunjukkan bahwa penyakit TBC merupakan salah satu dari sepuluh penyebab kematian terbesar di dunia. Kasus TBC yang telah ditemukan tersebar di berbagai negara, dengan kasus terbanyak berada di wilayah Asia Timur, Asia Tenggara, Afrika hingga ke wilayah Pasifik Barat. Pada tahun 2018 pula, Indonesia menempati peringkat ketiga di dunia setelah China dan India dalam hal jumlah kasus aktif TBC yaitu kurang lebih 842 ribu kasus [1].

Salah satu penyebab banyaknya kasus TBC yang ada di dunia, khususnya di Indonesia yaitu karena kurangnya kesadaran masyarakat mengenai bahayanya penyakit ini. Karena, penyakit ini dapat menyebar dengan cepat ke orang di sekitarnya apabila tidak ditangani dengan cepat. Apabila masyarakat dapat mengenali dan mengetahui bahwa terdapat seseorang yang berpotensi mengidap penyakit ini dengan lebih cepat, maka hal ini akan dapat mengurangi penyebaran sehingga bisa menyelamatkan orang tersebut serta orang di sekitarnya. Semakin cepat proses pendeteksian penyakit TBC, maka semakin cepat pula penderita mendapatkan penanganan dengan baik. Akibatnya, risiko terjadinya komplikasi pada tubuh penderitanya juga semakin berkurang.

Umumnya, penyakit TBC ini dapat didiagnosis melalui beberapa metode, yaitu menggunakan sampel dahak serta menggunakan scan x-ray. Diagnosis menggunakan sampel dahak merupakan cara yang dilakukan dengan melakukan pengujian sampel dahak oleh petugas medis. Namun, diagnosis ini terkadang sulit untuk dilakukan karena tidak semua orang bisa mengeluarkan dahak dengan mudah, yang menyebabkan dahaknya juga tidak bisa diperiksa oleh petugas medis. Selain itu, pemeriksaan menggunakan metode ini juga membutuhkan waktu yang lama, yaitu sekitar 5 - 7 hari. Ditambah pula hasil yang didapatkan terkadang kurang akurat. Diagnosis penyakit TBC juga bisa dilakukan dengan cara melakukan pemeriksaan x-ray pada bagian dada yang dilakukan oleh dokter radiologi di rumah sakit atau fasilitas kesehatan lainnya yang memiliki fasilitas pemeriksaan x-ray. Diagnosis TBC menggunakan x-ray ini dapat memberikan hasil yang lebih baik jika dibandingkan dengan metode sebelumnya. Namun, terdapat juga kelemahan dalam melakukan diagnosis TBC menggunakan metode x-ray ini. Waktu yang dibutuhkan untuk melakukan diagnosis juga cukup lama, sekitar 3 - 5 hari. Meskipun waktu dari hasil deteksi lebih cepat jika dibandingkan dengan deteksi menggunakan dahak, waktu yang dibutuhkan masih cukup lama bagi pasien untuk menerima hasil deteksinya. Hal ini menyebabkan pada periode waktu tersebut, pasien yang belum mendapatkan hasil deteksinya tersebut berpotensi untuk menyebarkan bakteri TBC yang ada di dalam tubuhnya jika hasilnya terbukti positif.

Oleh karena itu, penulis menemukan ide untuk melakukan suatu penelitian, yaitu membuat suatu sistem yang dapat melakukan deteksi penyakit TBC berdasarkan hasil scan x-ray bagian dada. Sistem deteksi akan memanfaatkan metode *Convolutional Neural Network* (CNN) untuk membuat sistem deteksinya, serta disusun menggunakan *framework Laravel* sebagai arsitektur aplikasinya. Harapannya agar proses deteksi penyakit TBC bisa dilakukan dengan lebih cepat sehingga penanganan yang dilakukan juga lebih cepat dan dapat melindungi keluarga di sekitar dari penyakit TBC ini.

1.2. Rumusan Masalah

Masalah yang diangkat dalam penelitian ini, yaitu “Bagaimana membuat sistem deteksi tuberkulosis pada citra X-Ray dengan metode CNN dan framework Laravel”. Berdasarkan identifikasi permasalahan tersebut, berikut merupakan rumusan masalah yang akan dibahas pada penelitian ini, yaitu sebagai berikut:

- Bagaimana proses mengembangkan sistem deteksi penyakit TBC ini?
- Bagaimana cara kerja sistem deteksi yang dikembangkan menggunakan metode CNN di dalam aplikasi sehingga dapat membantu mempercepat proses deteksi penyakit TBC?

1.3. Tujuan dan Manfaat Penelitian

Berdasarkan rumusan masalah yang telah didefinisikan sebelumnya, penulis menentukan tujuan serta manfaat yang ingin dicapai dalam penelitian yang akan dilakukannya.

1.3.1. Tujuan

- Mengetahui proses membangun aplikasi deteksi penyakit TBC menggunakan framework Laravel ini.
- Mengetahui penerapan metode CNN ke dalam aplikasi berbasis *website* yang dapat mempercepat proses deteksi penyakit TBC

1.3.2. Manfaat

a. Bagi Peneliti

- Meningkatkan pengetahuan, serta wawasan mengenai penerapan machine learning untuk mendeteksi penyakit TBC, serta penerapannya dalam aplikasi berbasis web menggunakan *framework Laravel*.
- Meningkatkan kemampuan analisis dalam proses pengembangan aplikasi deteksi penyakit TBC yang dapat digunakan oleh masyarakat

b. Bagi Masyarakat Umum:

- Mendeteksi penyakit TBC secara mandiri menggunakan hasil scan X-Ray sehingga masyarakat dapat mengetahui hasil deteksi dari scan X-Ray nya dengan lebih cepat
- Meningkatkan kesadaran mengenai bahayanya penyakit TBC ini serta cara untuk terhindar dari penyakit ini.

1.4. Batasan Masalah

Batasan masalah pada tugas akhir ini, yaitu:

- Pengembangan aplikasi hanya terbatas pada lingkup aplikasi berbasis web menggunakan Laravel versi 8.0
- Model machine learning yang digunakan untuk sistem deteksi ini telah dibuat dan dilatih sebelumnya
- Sistem deteksi hanya bisa mengenali foto scan x-ray bagian dada
- Sistem deteksi yang digunakan pada penelitian ini bukan merupakan alat medis. Oleh karena itu data dan hasil deteksinya hanya dapat dijadikan sebagai referensi dan tidak boleh digunakan sebagai dasar diagnosis atau pengobatan medis.
- Sistem deteksi yang dibangun hanya berfokus pada pengguna masyarakat umum sehingga belum ada penarikan opini dari tenaga ahli.

1.5. Sistematika Penulisan

Agar penulisan penelitian ini dapat tersusun dengan baik maka akan ditulis dengan sistematika runtut yang terdiri dari enam bab seperti berikut:

BAB I : PENDAHULUAN

Bab ini membahas mengenai latar belakang penulisan, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan yang digunakan dalam menyusun penelitian ini.

BAB II : KAJIAN LITERATUR

Bab ini berisikan berbagai landasan teori mengenai implementasi sistem deteksi penyakit TBC menggunakan CNN serta framework Laravel, disertai dengan beberapa penelitian lainnya yang menunjang penelitian.

BAB III : METODOLOGI PENELITIAN

Bab ini berisikan tahapan penelitian yang akan dilakukan oleh penulis serta metode yang akan digunakan dalam melakukan penelitian ini.

BAB IV : ANALISIS DAN PERANCANGAN

Pada bab ini penulis membahas tentang hasil implementasi rancangan sistem deteksi yang telah dirincikan pada bab sebelumnya, seperti hasil akhir antarmuka sistem, dan hasil pengujian aplikasi pada beberapa responden.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dan saran – saran dari penelitian yang dilakukan oleh penulis agar dapat dilakukan pengembangan lebih lanjut pada penelitian yang akan datang.



STT - NF

BAB II

KAJIAN LITERATUR

Pada bagian kajian literatur ini, penulis akan menjelaskan tentang penggambaran alur penelitian, serta keterkaitan antara penelitian yang penulis buat dengan penelitian - penelitian yang sudah ada sebelumnya.

2.1. Tinjauan Pustaka

Tinjauan pustaka merupakan bagian yang akan menjelaskan teori - teori yang mendukung penelitian yang dilakukan oleh peneliti, yaitu mengenai sistem deteksi penyakit TBC menggunakan CNN serta *framework Laravel*.

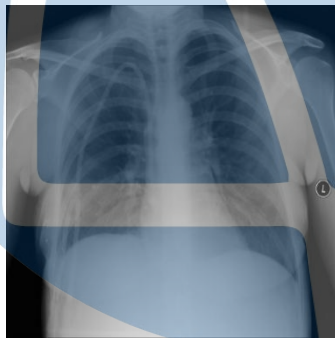
2.1.1. Sistem Deteksi

Sistem deteksi merupakan suatu proses di mana suatu objek diperiksa menggunakan suatu cara atau teknik tertentu yang dikemas dalam suatu sistem tertentu [2]. Tujuan dari sistem deteksi adalah memecahkan suatu masalah dengan berbagai cara tergantung metode yang diterapkan sehingga menghasilkan suatu. Sistem deteksi biasanya dibuat untuk menyelesaikan suatu permasalahan tertentu, seperti mendeteksi kematangan suatu buah, mendeteksi kerusakan dalam suatu barang, ataupun deteksi suatu penyakit. Misalnya dalam sistem pendeteksi suatu penyakit, di mana sistem deteksi akan mengidentifikasi tanda – tanda khusus yang berhubungan dengan penyakit tertentu. Tanda tersebut akan menjadi parameter yang bisa digunakan oleh sistem deteksi untuk menentukan hasil deteksinya.

Cara yang biasa digunakan di sistem deteksi yaitu dengan mengimplementasi suatu algoritma. Algoritma merupakan suatu set aturan yang harus diikuti berdasarkan suatu perhitungan yang dilakukan oleh suatu sistem yang ada pada komputer. Fungsi dari algoritma yaitu untuk mengumpulkan informasi - informasi berdasarkan objek yang didapatnya. Objek yang didapat bisa diambil dari berbagai perangkat seperti smartphone, laptop, drone, dan lain - lain. Informasi - informasi tersebut kemudian akan dibandingkan dengan informasi yang sudah ada pada database menggunakan algoritma yang telah diterapkan pada pendeteksinya. Hasil perbandingan tersebut nantinya akan menjadi penentu dari hasil akhir deteksinya.

2.1.2. Tuberkulosis

Penyakit Tuberkulosis (TBC) merupakan salah satu penyakit yang disebabkan oleh bakteri *Mycobacterium Tuberculosis*. Dalam hal ini, bakteri yang menyebabkan penyakit TBC ini menyerang paru - paru yang bertugas untuk mengelola sistem pernapasan serta sistem kekebalan tubuh. Apabila kondisi imun seseorang sedang menurun, maka resiko untuk tertular TBC akan semakin tinggi [3]. Oleh karena itu, penderita biasanya mengalami berbagai gangguan yang berkaitan dengan sistem pernapasan. Penyakit ini akan menyerang sistem kekebalan tubuh. Jika saat kondisi imun seseorang sedang menurun, maka resiko untuk tertular TBC akan semakin tinggi. Gejala dari penyakit ini yaitu batuk, batuk disertai darah, hingga nyeri pada dada atau pada saat bernafas. Penularan penyakit ini bisa melalui udara, di mana ketika orang yang terkena TBC itu batuk secara langsung di tempat umum tanpa ditutup dengan tangan, maka kuman tersebut akan keluar dan bertebaran melalui udara dan dapat terhirup oleh orang yang ada di sekitarnya yang dapat memungkinkan tertular penyakit ini [4].



Gambar 2.1 X-Ray Normal

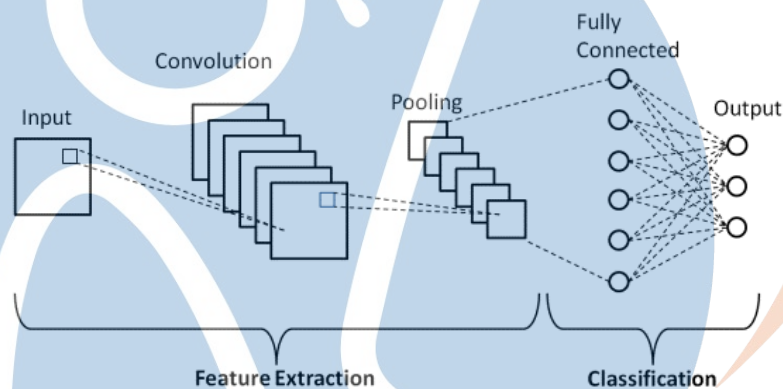


Gambar 2.2 X-Ray TBC

Telah disampaikan pada bagian latar belakang bahwa salah satu cara untuk mendeteksi penyakit TBC menggunakan scan X-Ray. Pada gambar 2.1 merupakan gambar X-Ray dengan kondisi normal sedangkan pada gambar 2.2 merupakan gambar X-Ray dengan kondisi TBC. Perbedaan pada kedua gambar di atas yaitu pada gambar 2.2 terdapat bercak yang ada pada sekitar paru – paru. Penjelasan lebih detail dari gambar x-ray yang terdapat penyakit TBC dapat dikonsultasikan kepada dokter spesialis paru.

2.1.3. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu metode dari algoritma *deep learning* yang mampu melakukan proses pembelajaran mandiri untuk pengenalan objek, ekstraksi objek dan klasifikasi. Algoritma ini terinspirasi dari sel - sel neuron yang ada di dalam otak manusia yang dapat mengenali dan mengingat banyak hal di sekitarnya. Tujuan yang ingin dicapai oleh algoritma ini yaitu untuk membuat suatu jaringan saraf yang dapat mengekstrak fitur penting dari suatu objek yang ingin diidentifikasi. Fitur penting ini akan disimpan dan akan digunakan untuk mengidentifikasi objek serupa dimasa yang akan datang [5].



Gambar 2.1 Convolutional Neural Network

Terdapat beberapa bagian (dalam CNN hal ini disebut sebagai *layer*) yang digunakan dalam proses identifikasi gambar atau objek yang akan dilakukan identifikasi seperti yang ditunjukkan pada gambar 2.1. Beberapa bagian tersebut antara lain:

- **Input Layer**

Input layer merupakan bagian awal yang merupakan tempat untuk menampung data. Data disini bisa berupa gambar, suara atau objek lainnya yang bisa diidentifikasi oleh algoritma.

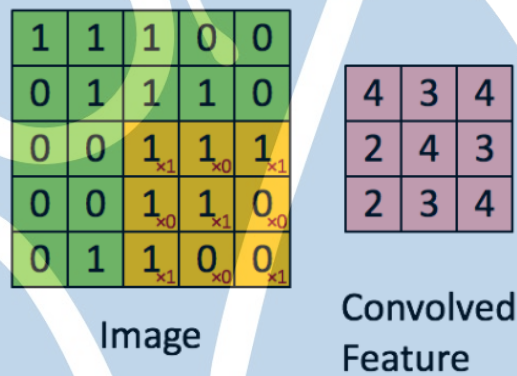
- **Hidden Layer**

Hidden Layer merupakan gabungan dari layer - layer selain *layer input* dan *output* di mana bagian ini digunakan oleh algoritma untuk 'belajar' mengenai objek yang akan diidentifikasi. Belajar disini dalam artian sistem akan melakukan serangkaian proses di mana algoritma dapat mengenali fitur - fitur penting yang ada di suatu objek.

Serangkaian proses tersebut dilakukan oleh *layer - layer* dan juga dua parameter penting yang mempengaruhi hasil prediksi, antara lain sebagai berikut.

a. Convolutional Layer

Convolutional layer merupakan layer pertama pada *hidden layer* yang berfungsi untuk mengekstrak berbagai fitur yang terdapat pada suatu objek. Cara kerja pada layer ini yaitu melakukan perkalian matriks antara area pada gambar dengan filter yang diterapkan. Ilustrasi dari proses konvolusi yaitu sebagai berikut.

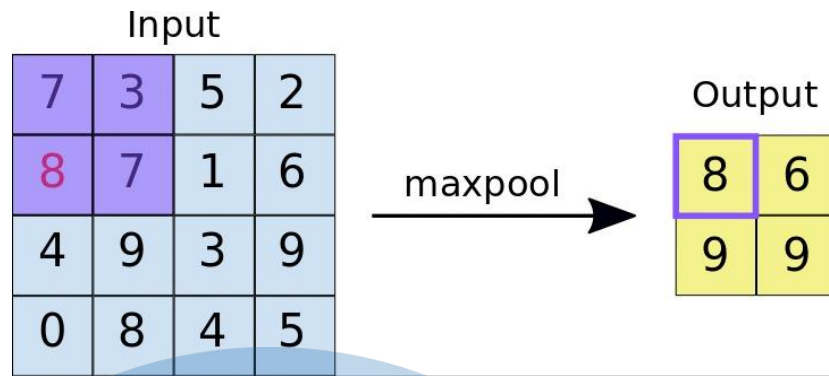


Gambar 2.2 Proses Konvolusi

Pada gambar 2.2 dijelaskan bahwa kotak berwarna hijau merupakan kotak yang merepresentasikan objek atau gambar, sedangkan kotak kuning merepresentasikan filter. Perkalian matriks dilakukan dengan melakukan perkalian antara kotak kuning dengan kotak hijau yang dimulai dari pojok kiri atas, hingga ke bagian kanan bawah. Hasil dari perkalian matriks tersebut direpresentasikan oleh kotak warna merah muda, tersebut yang nantinya akan diproses oleh layer selanjutnya.

b. Max Pooling Layer

Max Pooling layer merupakan suatu layer yang akan melakukan proses pengurangan resolusi pada gambar dengan tetap mempertahankan informasi pada gambar. Max pooling bertujuan untuk mengurangi ukuran setiap gambar pada proses konvolusi. Caranya yaitu seperti pada gambar di bawah.



Gambar 2.3 Max Pooling

Pada gambar 2.3 dijelaskan bahwa input dengan ukuran 4x4 piksel dilakukan proses maxpool yang menghasilkan piksel dengan ukuran 2x2 piksel. Area yang di beri warna berbeda akan dilakukan perhitungan dengan cara mencari angka yang terbesar dari area tersebut. Hal ini merupakan cara kerja dari max pooling itu sendiri, yaitu akan mengambil angka terbesar dari area piksel tertentu, yang angka - angka tersebut digabungkan menjadi output baru dengan dimensi 2x2 piksel. Output dari proses ini akan dibawa ke layer selanjutnya untuk proses berikutnya [6].

c. Fully Connected Layer

Hasil keluaran dari layer sebelumnya masih berupa tipe data array multidimensi yang datanya masih harus dilakukan proses *reshape feature map* menjadi sebuah *vector* agar bisa kita gunakan sebagai input dari *fully connected layer* [7]. *Fully connected layer (FC Layer)* merupakan layer yang memiliki konektivitas penuh dengan *layer-layer* sebelumnya, ataupun setelahnya. *Layer* ini berfungsi untuk membantu melakukan pemetaan dari representasi antara *layer input* dengan *layer output*. Pada *layer* ini, setiap *neuron* yang dihasilkan dari *layer* sebelumnya akan ditransformasi menjadi bentuk *vector*. Caranya yaitu dengan *vector* dikalikan dengan *weight matrix* dari layer. Hasil dari perkalian ini akan berbeda-beda dan akan dioptimasi selagi model melakukan proses *training*.

d. Dropout

Dropout merupakan suatu layer tambahan yang tidak wajib untuk digunakan, namun memiliki peran krusial dalam CNN ini. Layer ini berfungsi untuk mengurangi *overfitting* pada model CNN. *Overfitting* biasa terjadi ketika model CNN memetakan objek dengan sangat baik, bahkan mungkin nyaris sempurna pada data latih, namun tidak dapat melakukan implementasi hal tersebut ke data baru yang dilihat oleh model.

Cara kerja dari *dropout layer* ini yaitu dengan secara acak menghilangkan sekian persen jaringan saraf yang ada pada model CNN yang dibangun, supaya model dapat melakukan identifikasi dengan baik pada data yang sudah dilihat oleh model atau biasa disebut data latih, ataupun data yang baru dilihat oleh model alias data uji.

e. Activation Functions

Activation functions merupakan parameter terpenting dari suatu model CNN. Fungsi ini digunakan untuk mempelajari dan memperkirakan segala jenis hubungan yang berkelanjutan dan kompleks antara variabel-variabel yang ada di jaringan. Ada beberapa fungsi aktivasi yang umum digunakan seperti fungsi **ReLU**, **Softmax**, **tanH** dan **Sigmoid**. Masing-masing fungsi ini memiliki penggunaan yang spesifik. Untuk model CNN klasifikasi biner, fungsi sigmoid dan softmax lebih disukai dan untuk klasifikasi multi-kelas, umumnya digunakan softmax.

- **Output Layer**

Output layer merupakan bagian akhir yang merupakan tempat untuk . mengeluarkan hasil prediksi dari objek yang telah berusaha dikenali. Hasil prediksi disini akan dikeluarkan dalam bentuk angka yang nilainya ditentukan berdasarkan jumlah kelas yang ingin diidentifikasi.

2.1.4. API & FastAPI

Application Programming Interface atau yang biasa disingkat sebagai API yaitu suatu kumpulan kode yang berisi peraturan yang mendefinisikan bagaimana perangkat saling berkomunikasi satu sama lain [8]. Aplikasi atau perangkat yang menggunakan API akan dapat mengakses suatu sumber daya dari perangkat lain. Cara kerja dari API cukup sederhana, perangkat yang ingin mengakses akan mengirimkan *request* kepada API, yang selanjutnya perangkat lain akan menerima request tersebut dan memberikan tanggapan atau biasa disebut sebagai *response* melalui API untuk dikirimkan kembali ke perangkat yang mengirimkan *request*. Saat ini, telah banyak *framework* yang dapat membuat API dengan cepat dan mudah. Salah satu library yang bisa membuat API dengan cepat yaitu *FastAPI*.

FastAPI merupakan salah satu *web framework open-source* yang digunakan untuk membuat API yang dibangun menggunakan bahasa pemrograman *Python* [9]. *FastAPI* dipilih karena bahasa pemrograman yang digunakan sama - sama menggunakan bahasa pemrograman *Python* yang digunakan untuk membuat model *machine learning*. Namun tidak hanya itu, *FastAPI* memiliki kelebihan lain seperti mudah digunakan, lebih ringan, serta mempermudah dalam proses *debugging* pada saat terjadi error. Dokumentasi yang disediakan juga mudah dipahami sehingga semakin mempermudah dalam proses pembuatan API nantinya.

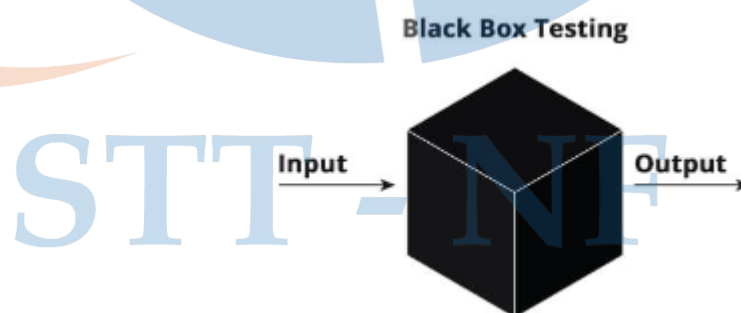
2.1.5. *Laravel Framework*

Laravel merupakan salah satu *framework* yang disusun menggunakan bahasa pemrograman *PHP*. Framework ini bersifat *open-source*, yang artinya kodenya dapat digunakan dan dikembangkan oleh banyak orang secara gratis. *Framework* ini diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola *Model View Controller* (MVC). Struktur pola MVC pada *laravel* sedikit berbeda pada struktur pola MVC pada umumnya. Di *laravel* terdapat routing yang menjembatani antara request dari user dan controller. Jadi, controller tidak langsung menerima request tersebut [10].

Ada berbagai alasan mengapa penulis menggunakan *framework Laravel* dalam penelitian ini. Pertama, *Laravel* merupakan *framework* yang sangat terkenal di kalangan pengembang perangkat lunak yang sudah dilengkapi dengan banyak modul dasar untuk mengoptimalkan kinerja PHP dalam pengembangan aplikasi web. Selain itu, *Laravel* juga memiliki ekosistem yang lengkap didukung oleh *package* dan ekstensi yang kompatibel dengan berbagai tipe *browser*. Penulis bisa saja membangun sistem deteksi pada penelitian ini menggunakan *framework Flask* yang lebih sederhana. Namun, *flask* kurang mendukung untuk *scalability* pada sistem di mana *Laravel* dapat mendukung hal ini.

2.1.6. *Blackbox Testing*

Pengujian pada perangkat lunak merupakan satu rangkaian aktivitas yang direncanakan dan sistematis. Rangkaian aktivitas ini bertujuan untuk menguji atau mengevaluasi dari hasil keluaran yang diinginkan. Selain itu, rangkaian aktivitas ini akan membantu untuk mengetahui kelemahan dari sistem yang diuji. Dengan mengetahui kelemahan dari perangkat lunak yang diuji, maka kesalahan dari perangkat lunak yang dapat terjadi pada saat digunakan oleh pengguna bisa diminimalisir. Pengujian perangkat lunak bisa dilakukan dengan berbagai metode. salah satunya yaitu dengan menggunakan metode *black box testing* [11].



Gambar 2.4 Black Box Testing

Gambar 2.4 menjelaskan mengenai ilustrasi sederhana dari metode pengujian *black box testing*. *Black box Testing* merupakan salah satu metode pengujian yang berfokus pada fungsional dari perangkat lunak. Pengujian perangkat lunak dengan metode *black box testing* ini akan menguji spesifikasi

fungsional dari perangkat lunak tanpa menguji desain dan kode program. Hal ini dimaksudkan untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Selain itu, pengujian ini dilakukan agar data yang dihasilkan sesuai dengan data yang dimasukkan setelah data dieksekusi dan menghindari kekurangan dan kesalahan pada aplikasi sebelum digunakan oleh pengguna [12].

2.1.7. User Acceptance Test (UAT)

User Acceptance Testing (UAT) adalah fase pengujian perangkat lunak yang dilakukan oleh *end user* untuk langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan pengguna. Tujuannya yaitu untuk memastikan sistem memenuhi tidak hanya dari spesifikasi sistemnya, tapi juga kebutuhan sebenarnya dari pengguna, Selama pengujian, perangkat lunak diuji untuk memastikan tugas-tugas apakah sudah sesuai dengan spesifikasinya. Hasil dari pengujian dapat dijadikan bukti bahwa sistem dapat membantu para pengguna mengatasi permasalahan yang dimilikinya [13].

2.1.8. Skala Likert

Skala Likert merupakan suatu skala bipolar yang digunakan untuk mengukur baik tanggapan positif ataupun negatif terhadap suatu pernyataan. Proses penentuan skor pada jawaban responden dilakukan dengan membuat klasifikasi dan kategori yang cocok tergantung kepada tanggapa responden. Terdapat 4 (empat) skala pilihan juga kadang digunakan untuk kuesioner skala likert yang memaksa orang memilih salah satu kutub karena pilihan netral tak tersedia [14].

STT - NF

2.2. Penelitian Terkait

Tabel 2.1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	Mawaddah Harahap, Em Manuel Laia, Lilis Suryani Sitanggang, Melda Sinaga, Daniel Franci Sihombing, Amir Mahmud Husein, 2022	Deteksi Penyakit Covid-19 Pada Citra X-Ray Dengan Pendekatan Convolutional Neural Network (CNN)	CNN	Citra X-Ray	Model ML Untuk mendeteksi COVID-19
2	Razief Moch Diar, R. Yunendah Nur Fu'Adah, dan Koredianto Usman, 2022	Klasifikasi Penyakit Paru-Paru Berbasis Pengolahan Citra X Ray Menggunakan Convolutional Neural Network	CNN	Citra X-Ray	Model ML Untuk mendeteksi Penyakit Paru-Paru
3	Brenda Monica Rotty, Angelia Melani Adrian, dan Junaidy Budi Sanger, 2020	Aplikasi Prediksi Penyakit Tuberkulosis Menggunakan Algoritma Convolutional Neural Network (CNN)	Web App	Citra X-Ray	Aplikasi berbasis website untuk deteksi penyakit TBC
4	C. I. Orozco, E. Xamena, C. A. Mart'inez and D. A. Rodriguez, 2021	COVID-XR: A Web Management Platform for Coronavirus Detection on X-Ray Chest Images	Web App	Citra X-Ray	Aplikasi berbasis website untuk deteksi penyakit COVID-19

Berdasarkan tabel 2.1, penulis menemukan beberapa penelitian terdahulu yang dapat menjadi rujukan mengenai topik seperti CNN serta *framework laravel* dengan berbagai macam studi kasus yang berbeda. Pertama, yaitu penelitian yang berjudul 'Deteksi Penyakit Covid-19 Pada Citra X-Ray Dengan Pendekatan *Convolutional Neural Network* (CNN)' [14]. Klasifikasi Penyakit Paru-Paru Berbasis Pengolahan Citra X Ray Menggunakan *Convolutional Neural Network* [15]. Kedua penelitian yang disebutkan sebelumnya berfokus menggunakan

metode CNN yang mana metode tersebut berkaitan dengan penelitian yang sedang penulis buat. Namun, pada kedua penelitian tersebut tidak dibahas mengenai implementasi model yang telah dibuat dalam bentuk aplikasi berbasis website.

Kemudian pada penelitian yang lainnya yaitu penelitian mengenai Aplikasi Prediksi Penyakit Tuberkulosis Menggunakan Algoritma Convolutional Neural Network (CNN) [16] berfokus pada implementasi model prediksi penyakit tuberkulosis, yang implementasinya dalam bentuk website dengan bahasa pemrograman PHP. Namun, pada penelitian ini tidak menggunakan *framework* seperti *laravel* untuk implementasi sistem deteksinya. Lalu, pada acuan penelitian yang terkait yaitu mengenai *COVID-XR: A Web Management Platform for Coronavirus Detection on X-Ray Chest Images* [17]. Penelitian tersebut melakukan pengembangan model yang dapat mendeteksi penyakit covid-19 yang diimplementasikan ke dalam aplikasi berbasis website yang dibangun menggunakan *framework laravel* serta database MySQL.

Penelitian yang dilakukan oleh penulis memang bukan penelitian pertama dan sudah ada penelitian yang membahas mengenai metode ini sebelumnya. Pada penelitian ini pula, penulis memposisikan penelitian ini menyerupai penelitian yang dilakukan oleh saudari Brenda Monica Rotty, yang di mana saudari tersebut melakukan penelitian dengan melakukan implementasi dalam bentuk website dengan bahasa pemrograman PHP. Namun di sini, penulis menambahkan kebaruan yang belum dilakukan di penelitian sebelumnya, yaitu menggabungkan konsep dari metode penelitian yang telah dilakukan menggunakan metode CNN dengan implementasinya menggunakan *framework Laravel* dalam bentuk aplikasi berbasis *website*. Harapannya, kebaruan yang dapat dilakukan pada penelitian ini dapat memberikan manfaat kepada pengguna yang membutuhkan.

BAB III METODOLOGI PENELITIAN

Pada bab ini akan dibahas tahapan serta langkah - langkah dalam penulisan, rancangan, dan juga lingkup penelitian yang dilakukan oleh penulis.

3.1. Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

Berikut adalah penjelasan dari gambar 4 yang merupakan tahapan - tahapan yang dilakukan dalam penelitian ini :

3.1.1. Studi Pendahuluan

Pada tahap awal yaitu studi pendahuluan, penulis melakukan kegiatan observasi terhadap sejumlah permasalahan baik yang dialami oleh penulis ataupun orang - orang di sekitar penulis. Setelah itu, penulis melakukan kajian berdasarkan permasalahan tersebut. Kajian yang dilakukan dengan cara membaca serta mengumpulkan referensi – referensi. Referensi penulis peroleh dari jurnal ilmiah, artikel, serta penelitian yang berkaitan dengan permasalahan yang ingin diteliti. Referensi yang didapat akan dijadikan bahan awal untuk memahami topik yang akan diteliti. Pada tahapan ini juga penulis melakukan pencarian dataset yang akan digunakan untuk sistem deteksi nantinya.

Hasil yang didapat yaitu pemahaman secara umum mengenai topik yang akan diteliti, serta latar belakang yang menjadi alasan penulis ingin melakukan penelitian ini. Lalu, penulis juga berhasil mendapatkan dataset yang berkaitan dengan topik penelitian yang akan digunakan untuk sistem deteksi, Hasil dari tahapan ini akan menjadi bekal utama dalam penulisan dalam penelitian ini.

3.1.2. Analisis Sistem

Pada tahap analisis sistem ini, penulis menguraikan pokok persoalan yang telah didapatkan dari proses sebelumnya. Persoalan tersebut akan dipecah menjadi bagian yang lebih kecil yang sekiranya penting untuk menunjang penelitian ini. Persoalan tersebut nantinya akan dilakukan proses analisis supaya menghasilkan pemahaman lebih lanjut mengenai topik penelitian yang akan diteliti.

Analisis sistem juga merupakan proses yang dilakukan penulis untuk memahami proses pembuatan sistem deteksi yang akan diimplementasikan ke aplikasi berbasis web ini. Dimulai dari proses pengumpulan data, pelabelan data, pemrosesan data, hingga pelatihan serta pengujian model yang telah dibuat. Pemahaman ini diperlukan supaya peneliti bisa memahami cara kerja sistem deteksi yang akan diimplementasikan nantinya. Hasil dari pemahaman ini akan digunakan dalam memahami proses dalam perancangan serta pengembangan sistem nantinya.

3.1.3. Perancangan Sistem

Tahap selanjutnya dari penelitian yang akan dilakukan penulis yaitu melakukan perancangan sistem. Perancangan sistem dilakukan dengan tujuan untuk mempermudah tahap implementasi ke dalam bahasa pemrograman supaya lebih mudah serta lebih terarah. Dengan adanya tahapan perancangan sistem ini, aplikasi yang dibangun akan sesuai dengan kebutuhan pengguna serta dapat berfungsi dengan optimal. Dengan memperhatikan hasil analisis serta pemahaman yang didapat dari tahapan sebelumnya, proses perancangan sistem yang akan dilakukan oleh penulis dapat dilakukan lebih terarah dan sesuai dengan kebutuhan.

Tahap perancangan sistem yang akan dilakukan oleh penulis yaitu melakukan serangkaian proses, seperti merancang desain aplikasi, merancang alur aplikasi, serta merancang arsitektur API yang akan menghubungkan aplikasi dengan sistem deteksi yang telah dibuat. Hasil dari perancangan ini akan dibawa ke tahapan selanjutnya yaitu proses pengembangan sistem.

3.1.4. Pengembangan Sistem

Tahapan selanjutnya sekaligus tahapan utama dari penelitian ini yaitu tahapan pengembangan sistem. Pengembangan sistem merupakan tahapan di mana penulis akan melakukan proses penyusunan sistem yang akan dibuat berdasarkan topik dari penelitian ini. Proses penyusunan sistem ini sangat berkaitan erat dengan proses penulisan kode program menggunakan bahasa pemrograman yang telah ditentukan. Proses pengembangan sistem ini akan dilakukan berdasarkan hasil yang didapatkan dari tahapan perancangan sistem yang telah dilakukan sebelumnya.

Pengembangan sistem yang dilakukan oleh penulis meliputi pembuatan halaman website sistem deteksi menggunakan HTML, CSS serta Javascript. Dilanjutkan dengan membuat logika bahasa pemrograman yang berkaitan dengan sistem deteksi menggunakan *framework Laravel*. Lalu, pembuatan API dengan menggunakan *FastAPI*. Tahapan pengembangan sistem diakhiri dengan menghubungkan seluruhnya menjadi satu kesatuan sistem. Hasil dari tahapan ini yaitu menghasilkan suatu sistem deteksi yang dapat digunakan untuk mengatasi masalah yang telah didefinisikan pada bagian rumusan masalah yang telah dibahas sebelumnya.

3.1.5. Pengujian dan Evaluasi

Tahapan akhir dari penelitian yang dilakukan penulis yaitu tahap pengujian dan evaluasi. Penulis akan menguji serta memberikan evaluasi pada sistem yang telah dibuat sebelumnya. Tujuannya yaitu untuk mengetahui apakah sistem yang telah dibuat telah menjawab permasalahan yang telah didefinisikan di awal penelitian, serta hasilnya bisa digunakan sebagai acuan untuk penelitian di masa yang akan datang.

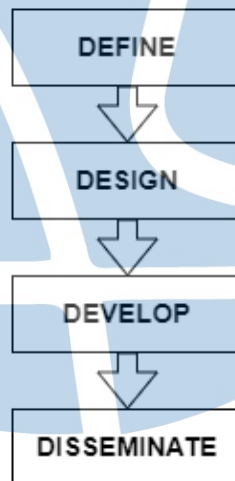
Proses pengujian deteksi yang telah dibangun menggunakan metode pengujian tertentu. Jika pada saat proses pengujian sistem terdapat suatu kendala yang hasilnya tidak sesuai keinginan, maka akan dilakukan proses evaluasi. Evaluasi dilakukan untuk memperbaiki sistem yang telah diuji supaya kendala yang terjadi pada proses pengujian tidak terjadi lagi dimasa yang akan datang. Hasil dari tahapan akhir ini yaitu untuk sistem yang telah dibuat dan dapat digunakan oleh pengguna tanpa ada kendala.

3.2. Rancangan Penelitian

Pada bagian rancangan penelitian ini, penulis memberikan penjelasan lebih dalam mengenai langkah-langkah yang akan dilakukan pada penelitian ini meliputi jenis penelitian, metode analisis data, metode pengumpulan data, metode pengujian, metode implementasi dan pengujian, serta lingkungan pengembangan.

3.2.1. Jenis Penelitian

Jenis penelitian yang akan dilakukan oleh penulis merupakan jenis penelitian dan pengembangan atau yang biasa disebut *Research and Development* (RnD). RnD merupakan jenis penelitian untuk mengembangkan dan menguji produk yang nantinya akan dikembangkan dalam berbagai bidang [18]. Dalam konteks penelitian ini, pengembangan perangkat lunak ini nantinya dapat bermanfaat dalam dunia kesehatan. Model pengembangan yang penulis gunakan untuk penelitian ini yaitu model 4D, yang merepresentasikan *define, design, develop* serta *disseminate*.



Gambar 3.2 Langkah-langkah pengembangan 4D

Adapun rincian dari gambar langkah-langkah yang diilustrasikan pada gambar 3.2 yaitu sebagai berikut:

1. *Define*

Tahapan *define* merupakan tahapan di mana peneliti melakukan pendefinisian mengenai syarat pengembangan. Pada penelitian yang akan dilakukan penulis, proses *define* akan dilakukan disaat proses studi pendahuluan dan analisis sistem. Hasil dari tahapan ini yaitu pemahaman mendalam tentang topik penelitian yang akan dilakukan.

2. *Design*

Tahapan *design* merupakan tahapan di mana peneliti melakukan proses perancangan perangkat lunak. Mulai dari merancang rencana pengembangan, merancang antarmuka aplikasi, merancang arsitektur aplikasi, dan lain – lain. Pada penelitian yang akan dilakukan penulis, proses desain akan dilakukan disaat proses perancangan sistem. Hasil dari tahapan ini yaitu rancangan yang akan diimplementasi pada tahapan selanjutnya.

3. *Develop*

Tahapan *develop* merupakan tahapan di mana peneliti melakukan proses pengembangan perangkat lunak. Pengembangan perangkat lunak biasanya akan dilakukan berdasarkan perencanaan yang telah dibuat pada tahapan sebelumnya. Pada penelitian yang akan dilakukan penulis, proses *develop* akan dilakukan di proses pengembangan sistem. Hasil dari tahapan ini yaitu perangkat lunak yang siap untuk diuji dan dievaluasi.

4. *Disseminate*

Tahapan *disseminate* merupakan tahapan di mana peneliti melakukan proses pengujian dan penyebarluasan dari perangkat lunak. Tahapan Pada penelitian ini, proses *disseminate* akan dilakukan di proses pengujian dan evaluasi sistem. Hasil dari tahapan ini yaitu perangkat lunak yang siap untuk disebarluaskan untuk dapat dimanfaatkan oleh orang yang membutuhkan perangkat lunak ini.

3.2.2. Metode Analisis Data

Metode analisis data yang digunakan pada penelitian ini yaitu metode kuantitatif. Metode kuantitatif merupakan metode menganalisis data yang bentuk datanya berupa angka. Metode kuantitatif dipilih karena pada proses analisis data yang didapat yaitu citra x-ray, data dianalisis dan dibandingkan dengan data yang telah ada pada sistem deteksi menggunakan metode CNN. Setelah itu, metode CNN tersebut mengeluarkan output berupa angka. Angka tersebut yang akan menjadi *threshold* atau Batasan apakah data citra x-ray tersebut terindikasi TBC atau tidak.

3.2.3. Metode Pengumpulan Data

Pada proses pengumpulan data dan informasi, peneliti menggunakan dua metode, yaitu metode observasi serta metode dokumentasi.

1. Metode observasi

Metode observasi yaitu metode pengumpulan data yang datanya diperoleh dari hasil pengamatan tingkah laku, individu atau kelompok yang diteliti secara langsung. Pengamatan yang dilakukan oleh penulis tertuju kepada individu penulis serta keluarga penulis. Kegiatan observasi ini merupakan kegiatan yang sangat penting bagi penulis. Karena, penulis jadi memahami keadaan sesungguhnya yang terjadi yang mengakibatkan dibutuhkan sistem deteksi penyakit TBC seperti yang akan penulis teliti ini.

2. Metode Dokumentasi

Metode dokumentasi yaitu metode pengumpulan data dari berbagai sumber tercetak, seperti hasil laporan, buku, berita, hasil penelitian lain, dokumen, laporan pertanggung jawaban. Data ini penulis dapat dari jurnal ilmiah yang telah dilakukan oleh peneliti lain [19]. Data yang didapat dari metode ini berupa citra x-ray bagian dada. Data citra x-ray yang penulis dapatkan ini berjumlah sekitar 4200 data. Data citra x-ray ini sangat berguna untuk proses pelatihan sistem deteksi ini.

3.2.4. Metode Pengujian

Metode pengujian yang penulis gunakan untuk penelitian ini akan dilakukan dengan menggunakan metode pengujian *Blackbox Testing* serta *User Acceptance Test* (UAT). Pengujian ini dilakukan untuk mengetahui apakah sistem deteksi dapat melakukan pendeteksian dengan baik dan dapat memberikan hasil sesuai dengan fungsinya serta memenuhi segala persyaratan yang diharapkan dapat dipenuhi oleh aplikasi.

3.2.4.1. Blackbox Testing

Pengujian *Blackbox Testing* akan dilakukan oleh penulis untuk menguji fungsionalitas dari sistem deteksi.

Tabel 3.1 Skema Pengujian *Blackbox Testing*

Halaman Utama			
No	Skenario Pengujian	Hasil Uji	Keterangan
1.	Menampilkan semua konten di halaman utama	Berhasil / Belum Berhasil	
2.	Menekan tombol 'Periksa Sekarang' untuk beralih ke halaman periksa	Berhasil / Belum Berhasil	
Halaman Periksa			
No	Skenario Pengujian	Hasil Uji	Keterangan
1.	Menampilkan semua konten yang ada di halaman periksa	Berhasil / Belum Berhasil	
2.	Melakukan <i>upload</i> gambar <i>scan</i> x-ray bagian dada pada <i>form upload</i> file	Berhasil / Belum Berhasil	
3.	Melakukan <i>upload</i> gambar selain gambar <i>scan</i> x-ray bagian dada	Berhasil / Belum Berhasil	
4.	Mengisi form laporan kendala dan mengirim laporan tersebut.	Berhasil / Belum Berhasil	
Halaman Hasil Periksa			
No	Skenario Pengujian	Hasil Uji	Keterangan
1.	Mengunjungi halaman hasil periksa setelah berhasil mengunggah gambar x-ray	Berhasil / Belum Berhasil	
2.	Mengunjungi halaman hasil periksa sebelum pengguna berhasil mengunggah gambar x-ray	Berhasil / Belum Berhasil	

Berdasarkan tabel 3.1, penulis menjabarkan poin – poin yang akan diujikan pada sistem deteksi yang dibuat oleh penulis. Pengujian sistem akan dilakukan pada tiga halaman, yaitu halaman utama, periksa, dan hasil periksa. Setiap halaman akan diuji apakah hasil yang diharapkan akan sesuai dengan yang diharapkan atau tidak. Jika pengujian sesuai dengan yang diharapkan, maka hasil dari pengujiannya berhasil. Namun, jika terdapat poin yang tidak berhasil lolos pengujian, maka akan dilakukan proses evaluasi.

3.2.4.2. User Acceptance Test (UAT)

User Acceptance Test akan dilakukan pengujian oleh *end user* untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan yang didefinisikan sebelumnya dan dapat diterima oleh pengguna.

Tabel 3.2 Skema Pengujian *User Acceptance Test*

No.	Poin Pengujian	Hasil	Keterangan
1.	Tata letak layar efisien dan visual yang ditampilkan menarik	Setuju / Tidak Setuju	
2.	Pengguna dapat bernavigasi antar halaman dalam aplikasi dengan mudah	Setuju / Tidak Setuju	
3.	Aplikasi memberikan tujuan yang jelas dan mendukung tujuan dari aplikasi yang dibuat ini	Setuju / Tidak Setuju	
4.	Pengguna dapat dengan mudah menggunakan fitur periksa	Setuju / Tidak Setuju	
5.	Aplikasi dapat memberikan hasil periksa dengan cepat	Setuju / Tidak Setuju	
6.	Aplikasi dapat memberikan hasil periksa sesuai dengan fakta	Setuju / Tidak Setuju	
7.	Aplikasi dapat memberikan informasi yang berguna sesuai dengan hasilnya	Setuju / Tidak Setuju	
8.	Pengguna dapat melaporkan kendala yang dialami ketika menggunakan aplikasi di form keluhan yang disediakan	Setuju / Tidak Setuju	

Berdasarkan tabel 3.2, penulis menjabarkan poin – poin yang akan ditanyakan kepada *end user* mengenai sistem deteksi yang telah dibuat oleh penulis. Pengujian ini akan berfokus kepada pengujian non fungsional sekaligus menguji fitur utama dari sistem deteksi apakah sistem deteksi yang telah dibuat memenuhi kebutuhan dari pengguna atau tidak. Hasil pengujian ini akan dimasukkan ke dalam evaluasi pengujian untuk dilakukan proses analisis apakah terdapat hal yang perlu ditingkatkan pada sistem deteksi ini atau tidak.

3.2.5. Metode Implementasi dan Evaluasi

Metode implementasi yang penulis gunakan dalam penelitian ini yaitu menggunakan *framework laravel*. *Framework laravel* digunakan sebagai salah satu bagian penting dari proses pengembangan sistem deteksi ini. Karena, *laravel* akan digunakan sebagai sarana interaksi utama antara pengguna dengan sistem deteksi ini. Tujuan utamanya yaitu jelas agar pengguna secara umum bisa mendapatkan kemudahan dalam melakukan deteksi penyakit TBC menggunakan perangkat apapun selama terhubung kedalam internet. Sedangkan evaluasi yang akan dilakukan oleh penulis merupakan evaluasi dari hasil pengujian yang telah dilakukan dengan metode yang telah dijabarkan pada sub bab sebelumnya.

3.2.6. Lingkungan Pengembangan

Dalam penelitian dan pengembangan aplikasi ini, diperlukan *software* serta *hardware* untuk pengembangan perangkat lunak berbasis web. *Software* yang digunakan, yaitu *Visual Studio Code*, *Git*, dan *Github*. Selain membutuhkan *software*, penelitian ini juga membutuhkan *hardware*. *Hardware* yang digunakan, yaitu laptop dengan spesifikasi sebagai berikut:

- a. Model : Lenovo Legion 5 15ACH6
- b. Processor : AMD Ryzen 5 5600H
- c. VGA : NVIDIA RTX 3050
- d. RAM : 16 GB
- e. System Type : 64-bit windows operating system

STT - NF

BAB IV

IMPLEMENTASI DAN EVALUASI

Pada bab ini akan membahas implementasi serta hasil pengujian dan evaluasi yang dilakukan oleh penulis dalam penelitian ini.

4.1. Pendefinisian Sistem

Pendefinisian sistem dalam pembuatan sistem deteksi penyakit TBC ini merupakan proses penulis mendefinisikan berbagai hal yang berkaitan dengan sistem deteksi penyakit TBC yang akan dibuat. Pendefinisian sistem ini dilakukan berdasarkan hasil observasi yang telah dilakukan oleh penulis. Observasi yang dilakukan oleh penulis yaitu dengan cara mempelajari pengalaman yang telah penulis alami di masa lalu, serta pengalaman – pengalaman yang diceritakan oleh orang – orang di lingkungan penulis.

Berdasarkan hasil observasi tersebut, penulis mendefinisikan daftar kebutuhan yang harus dipenuhi oleh sistem deteksi yang akan dibuat. Daftar yang akan dibuat yaitu antara lain meliputi kebutuhan fungsional, non-fungsional, serta spesifikasi teknis. Daftar kebutuhan ini akan berguna untuk menentukan apakah sistem deteksi ini akan dapat diterima dengan baik oleh pengguna atau tidak.

4.1.1. Kebutuhan Fungsional

Kebutuhan fungsional ini akan menggambarkan fungsi – fungsi utama yang harus dipenuhi oleh sistem deteksi, yaitu sebagai berikut:

1. **Fitur utama:** Sistem deteksi dapat melakukan pendeteksian pada gambar X-Ray yang diunggah oleh pengguna dengan baik. Lalu,
2. **Fitur pendukung:** Penambahan form untuk melaporkan kendala yang dialami oleh pengguna ketika menggunakan fitur utama dapat meningkatkan pengalaman dari pengguna.

4.1.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dari sistem deteksi penyakit TBC ini akan menjelaskan aspek lain selain fungsi utama yang harus dipenuhi pada sistem deteksi TBC ini, yaitu:

1. **Kinerja:** Sistem deteksi dapat melakukan pendeteksian pada gambar X-Ray yang diunggah oleh pengguna dengan cepat.

2. **Kemudahan:** Sistem deteksi memiliki antarmuka pengguna yang dapat digunakan dengan mudah oleh pengguna.

4.1.3. Spesifikasi Teknis

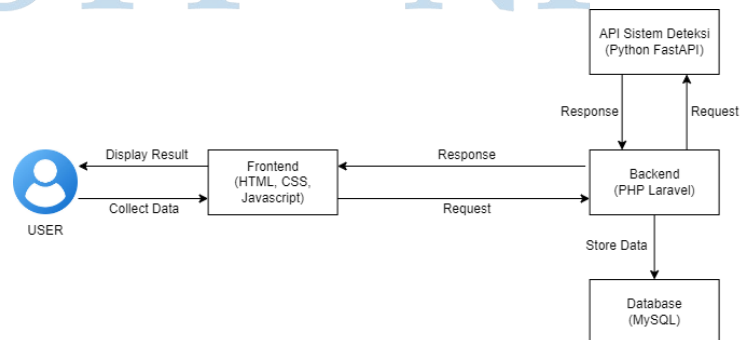
Spesifikasi teknis disini akan menjelaskan teknologi serta infrastruktur yang akan digunakan dalam pembuatan sistem deteksi ini, yaitu sebagai berikut:

1. **Bahasa Pemrograman:** Sistem deteksi akan menggunakan *HTML, CSS* dan *Javascript* sebagai penyusun bagian *frontend* dari sistem deteksi, bahasa pemrograman *PHP* dengan *framework Laravel* sebagai penyusun bagian *backend* dari sistem deteksi, serta bahasa pemrograman *python* dengan *framework FastAPI* yang akan berfungsi sebagai untuk membuat API yang menghubungkan antara aplikasi dengan model sistem deteksinya.
2. **Basis Data:** Sistem deteksi akan menggunakan basis data *MySQL*. *MySQL* ini akan digunakan untuk menyimpan data hasil deteksi serta data keluhan yang dialami oleh pengguna

4.2. Desain Sistem

Desain sistem pada sistem deteksi ini merupakan proses penulis merancang sistem yang akan diimplementasikan pada sistem deteksi ini. Proses perancangan akan meliputi perancangan arsitektur sistem deteksi, perancangan antarmuka sistem deteksi, serta perancangan alur kerja sistem deteksi penyakit TBC ini. Perancangan ini sangat penting untuk dilakukan supaya proses implementasi dalam pembuatan sistem deteksi bisa dilakukan sesuai dengan rancangan yang telah dibuat.

4.2.1. Desain Arsitektur



Gambar 4.1 Desain Arsitektur Sistem Deteksi

Gambar 4.1 menjelaskan mengenai desain arsitektur dari sistem deteksi yang akan dibuat. Desain arsitektur ini tersusun atas empat bagian, yaitu *frontend*, *backend*, *API*, serta *database*. Untuk *frontend* digunakan untuk menampilkan informasi serta sebagai media untuk berinteraksi dengan user. Lalu, *backend* bertanggung jawab untuk menangkap data hasil prediksi serta data keluhan dari pengguna. *API* sistem deteksi digunakan sebagai tempat berkomunikasi antara aplikasi sistem deteksi dengan model *machine learning* yang telah dibuat. Lalu, bagian terakhir yaitu *database* digunakan untuk menyimpan data hasil prediksi serta hasil laporan yang disampaikan oleh pengguna. Keempat bagian tersebut saling berkaitan satu sama lain supaya sistem deteksi dapat berjalan sesuai dengan fungsi utama yang telah didefinisikan sebelumnya.

4.2.2. Desain Antarmuka

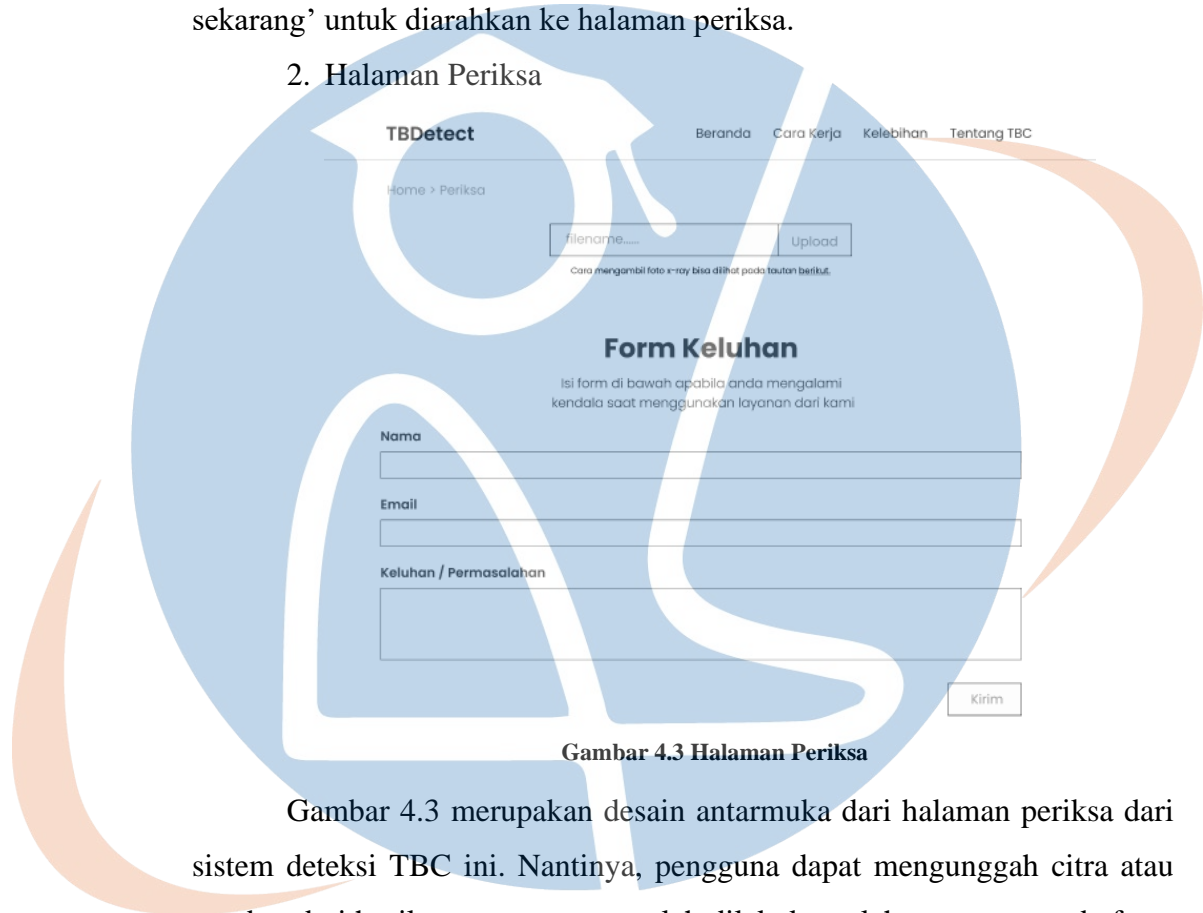
1. Halaman Utama



Gambar 4.2 Halaman Utama

Gambar 4.2 merupakan desain antarmuka dari halaman utama sistem deteksi TBC yang akan dibuat oleh penulis. Halaman ini merupakan halaman yang akan ditampilkan paling awal kepada pengguna ketika mengunjungi *website* sistem deteksi ini. Halaman ini berisi informasi mengenai hal apa yang bisa dilakukan oleh *website* ini kepada pengguna. Jika pengguna ingin menggunakan fitur deteksi TBC, pengguna dapat menekan tombol ‘periksa sekarang’ untuk diarahkan ke halaman periksa.

2. Halaman Periksa

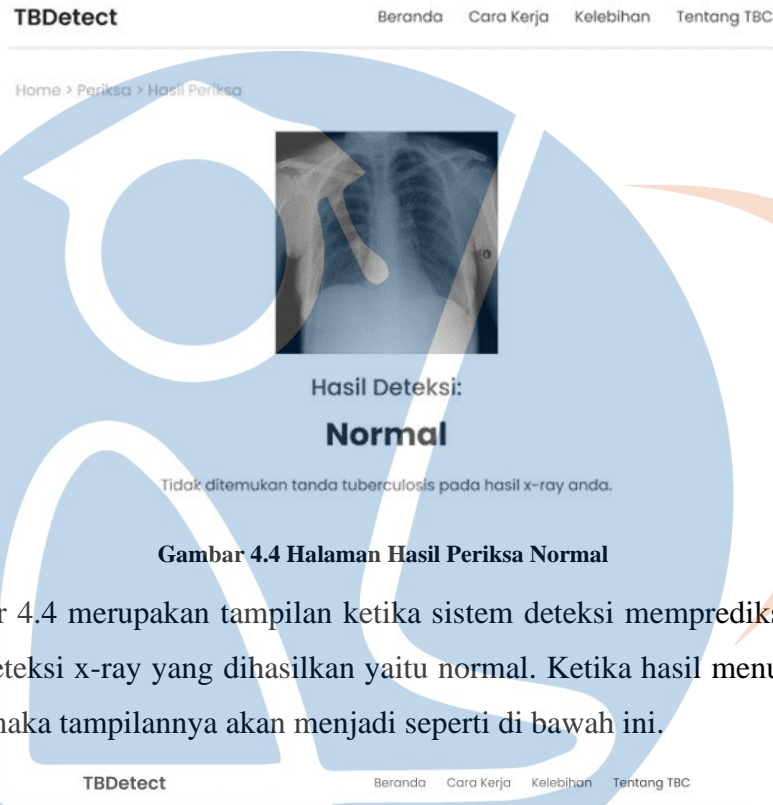


Gambar 4.3 Halaman Periksa

Gambar 4.3 merupakan desain antarmuka dari halaman periksa dari sistem deteksi TBC ini. Nantinya, pengguna dapat mengunggah citra atau gambar dari hasil scan x-ray yang telah dilakukan oleh pengguna pada form yang tersedia pada bagian atas. Setelah pengguna memilih gambar, maka pengguna bisa menekan tombol *upload*. Hasil dari proses deteksinya akan ditampilkan pada halaman lain yang akan dijelaskan setelah ini. Jika pengguna mengalami permasalahan pada saat menggunakan sistem deteksi yang tersedia, pengguna bisa melaporkan permasalahan yang terjadi dengan mengisi form keluhan yang ada di bawah form upload citra x-ray.

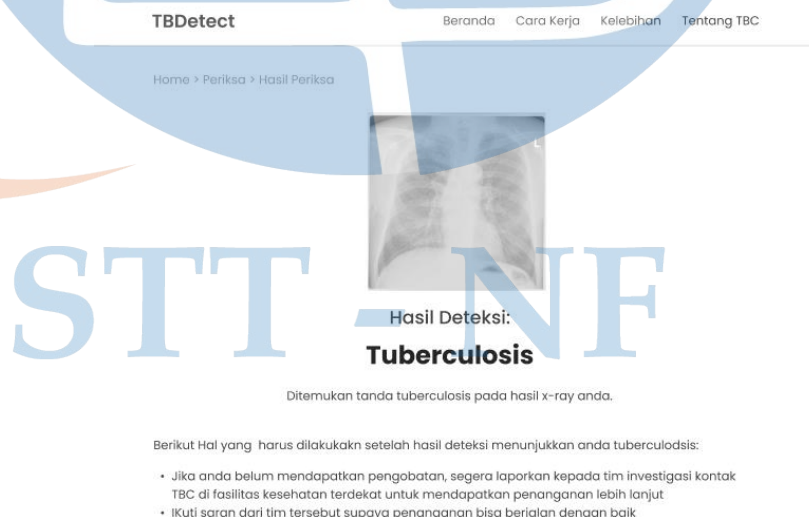
3. Halaman Hasil Periksa

Halaman hasil periksa merupakan halaman yang akan ditampilkan ketika pengguna telah berhasil mengunggah gambar x-ray miliknya dan tersebut telah berhasil diidentifikasi oleh API sistem deteksi. Halaman ini akan memiliki tampilan yang berbeda tergantung dari hasil pemeriksaannya



Gambar 4.4 Halaman Hasil Periksa Normal

Gambar 4.4 merupakan tampilan ketika sistem deteksi memprediksi bahwa hasil deteksi x-ray yang dihasilkan yaitu normal. Ketika hasil menunjukkan TBC, maka tampilannya akan menjadi seperti di bawah ini.

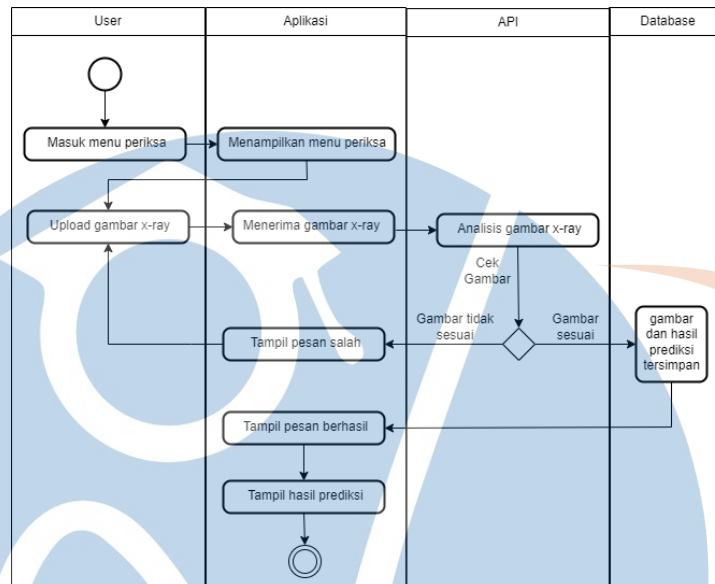


Gambar 4.5 Halaman Hasil Periksa Positif TBC

Gambar 4.5 merupakan gambar desain dari halaman hasil periksa ketika sistem deteksi menunjukkan hasil deteksinya yaitu positif TBC. Ketika hasilnya positif TBC, tampilan akan menunjukkan status hasilnya dengan *font*

berwarna merah dan terdapat informasi mengenai hal dapat dilakukan jika sistem deteksi memberikan hasil deteksinya positif TBC.

4.2.3. Desain Alur Kerja



Gambar 4.6 Diagram Alur Kerja Sistem Deteksi

Gambar 4.6 merupakan desain alur kerja dari fitur utama dari sistem deteksi ini, yaitu fitur deteksi penyakit TBC. Alur kerja akan dimulai dari pengguna akan masuk ke halaman periksa. Setelah aplikasi berhasil menampilkan halaman periksa, user akan mengunggah foto x-ray yang dimiliki di form yang telah disediakan. Setelah pengguna berhasil mengunggah foto x-ray nya, maka sistem akan melakukan pengecekan melalui API sistem deteksi apakah gambar yang diunggah oleh pengguna merupakan gambar x-ray bagian dada atau bukan. Apabila API menerima hasil bahwa gambar yang diunggah bukan gambar x-ray, maka aplikasi akan menampilkan pesan bahwa gambar yang diupload kurang sesuai, lalu meminta pengguna mengunggah kembali gambar x-ray yang lebih sesuai. Namun, apabila gambar yang diupload telah sesuai, aplikasi akan menerima hasil deteksinya yang data hasil deteksinya akan disimpan ke database terlebih dahulu sebelum ditampilkan ke pengguna di halaman hasil periksa. Setelahnya, pengguna akan langsung berpindah otomatis ke halaman hasil periksa dan akan ditampilkan hasil pemeriksaannya serta keterangannya.

4.2.4. Desain Database



tbdetect periksa	
id	int(11)
image	varchar(50)
result	varchar(50)
percentage	float
created_at	timestamp
updated_at	timestamp

tbdetect keluhan	
id	int(11)
nama	varchar(50)
email	varchar(50)
deskripsi	varchar(250)
created_at	timestamp
updated_at	timestamp

Gambar 4.7 Desain Database Sistem Deteksi

Pada gambar 4.7 merupakan desain *database* yang akan digunakan untuk menyimpan data pada sistem deteksi yang akan dibuat. Terdapat dua tabel yang akan digunakan pada sistem deteksi ini, yaitu table periksa dan tabel keluhan. Untuk menyimpan hasil deteksi, sistem akan menyimpan data hasil deteksi tersebut di tabel periksa. Sedangkan untuk data keluhan atau permasalahan yang dialami oleh pengguna akan disimpan di tabel keluhan. Kedua tabel tersebut tidak saling berkaitan satu sama lain sehingga tidak ada relasi diantara kedua tabel tersebut.

4.3. Implementasi Sistem

Implementasi sistem dalam pembuatan sistem deteksi penyakit TBC ini merupakan tahapan yang dilakukan untuk merealisasikan desain atau rancangan sistem yang telah dibuat sebelumnya. Disini, penulis akan melakukan implementasi yang diawali dengan membuat API untuk sistem deteksinya. Lalu, penulis membuat *frontend* dan *backend* dari aplikasi berbasis *website* ini supaya pengguna dapat berinteraksi dengan sistem deteksinya. Setelah dibuat, penulis akan menghubungkan antara API yang telah dibuat dengan *frontend* dan *backend* dari aplikasi yang telah dibuat tadi sehingga pengguna dapat menggunakan sistem deteksi lewat aplikasi berbasis *website* yang telah dibuat.

Penulis telah menyesuaikan seluruh kegiatan pada tahapan implementasi ini dengan persyaratan teknis yang telah didefinisikan sebelumnya. Namun, penulis tidak mencantumkan seluruh kode yang digunakan di sistem deteksi yang dibuat pada tulisan ini. Kode selengkapnya dari sistem deteksi yang telah dibuat bisa dilihat pada alamat *url* berikut ini: <https://github.com/alldinosaur/TBDetect>.

4.3.1. API Sistem Deteksi

Implementasi *API* pada sistem deteksi penyakit TBC ini bertujuan sebagai media komunikasi antara model *machine learning* yang dapat memprediksi penyakit TBC pada gambar x-ray dengan aplikasi berbasis *website* yang digunakan untuk berinteraksi dengan pengguna. Implementasi ini menggunakan bahasa pemrograman *python* dengan *framework FastAPI*.

```
app = FastAPI()

app.add_middleware(
    CORSMiddleware,
    allow_origins = origins,
    allow_credentials = True,
    allow_methods = methods,
    allow_headers = headers
)

model_dir = "models/TBDetect.h5"
model = load_model(model_dir)
class_names = ['NORMAL', 'TUBERCULOSIS']
```

Kode sebelumnya merupakan kode dari implementasi *API* untuk menginisiasi model *machine learning* yang akan digunakan. Pertama, penulis menginisiasi kelas yang berisi *library FastAPI* didalamnya. Lalu, penulis menginisiasi model yang telah dibuat serta kelas yang akan dijadikan sebagai nilai dari hasil deteksi. Selanjutnya, model akan melakukan prediksi berdasarkan gambar yang dimasukkan oleh pengguna.

```
loaded_image =
load_image_into_numpy_array(uploaded_file.file.read()
)

img_array = expand_dims(loaded_image, 0)

image = np.vstack([img_array])

pred = model.predict(image)
```

Gambar yang diunggah oleh pengguna akan masuk ke dalam variabel dan disesuaikan terlebih dahulu sebelum dilakukan prediksi oleh model. Seperti yang ditunjukkan pada kode yang ditulis di bagian awal halaman ini, gambar akan diubah menjadi bentuk yang dapat dikenali oleh model terlebih dahulu, sebelum akhirnya gambar dimasukkan ke dalam model yang akan memprediksi gambar tersebut. Setelah diprediksi, model akan memberikan keluaran berupa angka yang nilainya diantara 0 – 1. Lalu, Nilai keluaran tersebut akan diklasifikasikan sesuai dengan kelas yang telah diinisiasikan sebelumnya seperti kode di bawah ini.

```
if pred[0] < 0.3:
    status = "berhasil"
    class_pred = class_names[0]
    model_score = 100 - round(max(pred[0]) * 100, 2)
elif pred[0] > 0.7:
    status = "berhasil"
    class_pred = class_names[1]
    model_score = round(max(pred[0]) * 100, 2)
else:
    status = "gagal"
    class_pred = "Gambar tidak sesuai atau kurang
    jelas, silahkan coba lagi."

json = {
    "status": status,
    "model_prediction": class_pred,
    "model_prediction_confidence_score": model_score
}

return json
```

Nilai keluaran dari variabel prediksi tersebut diklasifikasikan berdasarkan besar nilainya. Apabila nilai keluarannya kurang dari 0.3 maka kelas yang menjadi keluarannya yaitu **NORMAL**. Jika nilai keluarannya lebih dari 0.7 maka hasil keluarannya yaitu **TUBERCULOSIS**. Namun, jika nilai nya tidak diantara kedua nilai diatas, maka kemungkinan besar pengguna

mengunggah gambar yang kurang sesuai atau kurang jelas. Akibatnya, sistem akan diminta untuk mengunggah ulang kembali gambar x-ray nya. Setelah berhasil diklasifikasi, maka hasil klasifikasi akan dijadikan sebagai nilai keluaran yang akan digunakan oleh aplikasi dalam bentuk *json*.

Sebagai informasi tambahan, untuk model sistem deteksi yang digunakan pada *API* sistem deteksi ini merupakan model *machine learning* yang telah dibuat menggunakan *framework Tensorflow* dengan menggunakan arsitektur CNN. Model ini dilatih dengan data gambar x-ray bagian dada dengan total data kurang lebih sebesar 4200 data. Setelah model dilakukan proses *training*, model ini memiliki akurasi sekitar 93% pada data latih dan sekitar 85% pada data uji. Untuk proses pembuatan model *machine learning* ini tidak dibahas pada penelitian kali ini karena sudah termasuk pada sub bab batasan masalah yang telah dibahas pada awal penulisan penelitian ini.

4.3.2. *Frontend dan Backend Sistem Deteksi*

```
<form name="periksa" class="periksa" method="post"
enctype="multipart/form-data">

    @csrf

    <input type="file" class="form-control scan"
name="scan" id="inputGroupFile04" aria-
describedby="inputGroupFileAddon04" aria-
label="Upload" accept="image/png, image/jpeg">

</form>

<button class="btn btn-outline-secondary"
type="button" id="inputGroupFileAddon04"
onClick="tambah()">Upload</button>
```

Kode tersebut merupakan *form HTML* yang dapat digunakan untuk memasukkan gambar x-ray untuk diperiksa oleh sistem deteksi. Setelah pengguna berhasil memasukkan gambar x-ray nya, form tersebut akan menjalankan fungsi **tambah** yang akan memproses gambar tersebut.

```

async function tambah(){

    let form = $('#periksa')[0];

    let formData = new FormData(form);

    let image = $('#scan')[0].files[0];

    if (!image) {

        return alert('Please select a file to upload.');
```

Bagian awal dari fungsi **tambah** seperti yang tertera pada kode berguna untuk menyimpan gambar yang diunggah pada form yang telah dibuat. Lalu, gambar tersebut akan dicek atau dideteksi menggunakan fungsi *check_image*. Setelah itu, fungsi **tambah** akan menunggu fungsi *check_image* selesai dijalankan, baru akan melanjutkan ke kode selanjutnya.

```

async function check_image(image) {

    var form = new FormData();

    form.append("uploaded_file", image);

    return $.ajax({

        url: "http://127.0.0.1:5000/predict",

        type: "POST",

        processData: false,

        mimeType: "multipart/form-data",

        contentType: false,

        data: form,

    });

}
```

Fungsi *check_image* akan mengirimkan gambar yang telah disimpan tadi ke dalam API yang telah dibuat pada sub bab sebelumnya. Setelah gambar dikirimkan, fungsi ini akan menerima hasil deteksi atau nilai yang dikirimkan oleh API dan akan ditampilkan ke pengguna. Hasil dari fungsi

check_image akan dilanjutkan kembali di fungsi **tambah** yang sebelumnya masih menunggu fungsi *check_image* untuk menyelesaikan tugasnya.

```
let result = JSON.parse(image_result);
if (result.status == "gagal") {
    Swal.fire({
        title: "Gagal",
        text: result.model_prediction,
        icon: "error",
        showConfirmButton: true,
    });
} else {
    formData.append("result", result.model_prediction);
    formData.append("percentage",
    result.model_prediction_confidence_score);
    $.ajax({
        type: "POST",
        data: formData,
        processData: false,
        contentType: false,
        url: "{{ route('app.simpan-periksa') }}",
        success: function(response) {
            if (response.status == "berhasil") {
                Swal.fire({
                    title: "Berhasil",
                    text: "Data berhasil disimpan",
                    icon: "success",
                    showConfirmButton: true,
                });
                window.location.href = "{{
route('app.hasil-periksa') }}";
            } else if (response.status == "gagal") {
                alert('gagal');
            }
        }
    })
}
```

Kode lanjutan dari fungsi **tambah** akan menampilkan hasil prediksi yang dikeluarkan oleh *API*. Apabila status dari prediksinya gagal, maka akan keluar pesan grafis gagal yang menandakan pengguna harus mengunggah ulang gambar x-ray yang dimiliki. Namun, apabila status dari hasil

prediksinya berhasil, maka akan keluar infografis bahwa gambar berhasil diprediksi.



Berhasil

Data berhasil disimpan

OK

Gambar 4.8 Pesan grafis berhasil



Gagal

Gambar tidak sesuai atau kurang jelas, silahkan coba lagi

OK

Gambar 4.9 Pesan grafis gagal

Gambar 4.8 menandakan bahwa gambar berhasil diprediksi. Lalu, data hasil prediksinya akan dimasukkan ke *database* dan pengguna akan diarahkan ke halaman hasil periksa untuk melihat hasil prediksinya.

```
public function simpanPeriksa(Request $request){
    $prediksi = new AppModel();
    $imageName = time().'.'.$request->scan->extension();
    $request->scan->move(public_path('images'), $imageName);

    $prediksi->image = $imageName;
    $prediksi->result = $request->result;
    $prediksi->percentage = $request->percentage;
    $prediksi->save();

    return response()->json([
        'status' => 'berhasil',
        'message' => 'data berhasil disimpan'
    ]);
}
```

Kode sebelumnya merupakan sebuah fungsi yang akan menyimpan data hasil prediksi ke dalam *database*. Setelah data disimpan, data tersebut akan ditampilkan di halaman hasil periksa sebagai *output* dari hasil deteksinya. Untuk melihat kode dari halaman hasil periksa serta kode lengkap

dari tahapan implementasi ini dapat dilihat pada alamat *url* berikut ini:
<https://github.com/alldinosaur/TBDDetect>.

4.4. Pengujian dan Evaluasi Sistem

Pengujian dan evaluasi ini merupakan kegiatan pengujian sistem pada sistem deteksi penyakit TBC. Kegiatan ini mulai dilakukan setelah proses implementasi sistem selesai dilakukan. Pengujian dilakukan dengan menggunakan metode *black-box testing* yang memeriksa *input* serta *output* dari sistem yang diujikan. Pengujian akan dilakukan ke semua halaman yang ada di sistem deteksi ini. Setelah dilakukan pengujian, akan dilakukan proses evaluasi kepada sistem yang telah diujikan.

4.4.1. Hasil Pengujian

Berikut hasil pengujian dari sistem deteksi yang telah dilakukan menggunakan metode blackbox testing dan user acceptance test.

4.4.1.1. Hasil Pengujian Blackbox Testing

Tabel 4.1 Hasil Pengujian *Blackbox Testing*

Halaman Utama				
No	Skenario Pengujian	Harapan	Hasil Uji	Keterangan
1.	Menampilkan semua konten di halaman utama	Semua konten berhasil ditampilkan	Berhasil	-
2.	Menekan tombol 'Periksa Sekarang' untuk beralih ke halaman periksa	Tombol bekerja dan berhasil beralih ke halaman periksa	Berhasil	-
Halaman Periksa				
No	Skenario Pengujian	Harapan	Hasil Uji	Keterangan
1.	Menampilkan semua konten yang ada di halaman periksa	Semua konten berhasil ditampilkan	Berhasil	-
2.	Melakukan <i>upload</i> gambar <i>scan</i> x-ray bagian dada pada <i>form upload</i> file	Gambar scan x-ray terkirim ke API dan berhasil di validasi oleh sistem deteksi dengan baik	Berhasil	Terdapat <i>alert</i> dengan keterangan berhasil ketika form berhasil diisi
3.	Melakukan <i>upload</i> gambar selain gambar	Gambar terkirim ke API,	Berhasil	Terdapat <i>alert</i> dengan

	scan x-ray bagian dada	namun gagal di validasi oleh sistem deteksi		keterangan gagal
4.	Mengisi form laporan kendala dan mengirim laporan tersebut.	<i>Form</i> berfungsi dengan baik dan data sukses terkirim ke API	Berhasil	Terdapat <i>alert</i> dengan keterangan berhasil
Halaman Hasil Periksa				
No	Skenario Pengujian	Harapan	Hasil Uji	Keterangan
1.	Mengunjungi halaman hasil periksa setelah berhasil mengunggah gambar x-ray	Halaman dapat menampilkan hasil deteksi berdasarkan response API yang telah diberikan dengan baik	Berhasil	-
2.	Mengunjungi halaman hasil periksa sebelum pengguna berhasil mengunggah gambar x-ray	Halaman hasil deteksi tidak dapat ditampilkan	Tidak berhasil	Halaman hasil deteksi masih dapat ditampilkan

Tabel 4.1 merupakan tabel hasil pengujian fungsionalitas yang telah dilakukan kepada dari sistem deteksi penyakit TBC. Dari delapan skenario pengujian yang telah dilakukan, tujuh pengujian berhasil dilakukan dengan persentase keberhasilan yaitu 87%. Terdapat satu pengujian yang tidak berhasil diujikan, yaitu halaman hasil periksa masih dapat dikunjungi atau dilihat meskipun belum mengirimkan gambar x-ray pada form yang ada di halaman periksa. Faktor yang menyebabkan satu pengujian tidak berhasil dilakukan karena belum ada pembatasan yang membatasi akses yang seharusnya diterapkan pada halaman hasil periksa. Akibatnya, meskipun pengguna belum mengirimkan gambar x-ray pada form yang ada di halaman periksa, pengguna masih bisa dapat melihat hasil pemeriksaan yang mungkin saja telah dilakukan oleh orang lain.

Pengujian juga penulis lakukan untuk memenuhi kebutuhan non-fungsional dari sistem deteksi, yaitu dari sisi kinerja. Sisi kinerja ini dinilai dari seberapa cepat sistem deteksi dapat menampilkan hasil deteksinya kepada pengguna.

```

1/1 0s 178ms/step
INFO: 127.0.0.1:59912 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 23ms/step
INFO: 127.0.0.1:59922 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 15ms/step
INFO: 127.0.0.1:59926 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 18ms/step
INFO: 127.0.0.1:59963 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 16ms/step
INFO: 127.0.0.1:59970 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 16ms/step
INFO: 127.0.0.1:59989 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 18ms/step
INFO: 127.0.0.1:60015 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 17ms/step
INFO: 127.0.0.1:60041 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 17ms/step
INFO: 127.0.0.1:60044 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 16ms/step
INFO: 127.0.0.1:60094 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 19ms/step
INFO: 127.0.0.1:60099 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 16ms/step
INFO: 127.0.0.1:60167 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 19ms/step
INFO: 127.0.0.1:60167 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 16ms/step
INFO: 127.0.0.1:60173 - "POST /predict HTTP/1.1" 200 OK
1/1 0s 18ms/step
INFO: 127.0.0.1:60180 - "POST /predict HTTP/1.1" 200 OK

```

Gambar 4.10 Hasil Pengujian Kinerja Sistem Deteksi

Gambar 4.10 menunjukkan bahwa pengujian yang dilakukan oleh penulis dengan melakukan pemeriksaan pada 15 gambar x-ray bagian dada pada sistem deteksi memberikan hasil yang sangat baik. Waktu tercepatnya yaitu 15ms dan waktu terlama yaitu 178ms. Hasil ini menunjukkan bahwa sistem deteksi dapat memberikan hasil prediksi dengan sangat cepat, hingga di bawah dari 1 detik yaitu sekitarr 0.1 detik hingga 0.2 detik. Ini membuktikan bahwa kinerja dari sistem deteksi ini cukup sesuai dengan kebutuhan non-fungsional yang diinginkan oleh pengguna.

4.4.1.2. Hasil Pengujian User Acceptance Test

Tabel 4.2 Keterangan User Acceptance Test

Keterangan	Deskripsi	Nilai
SS	Sangat Setuju	4
S	Setuju	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

Tabel 4.2 merupakan tabel keterangan dari setiap pertanyaan yang ditanyakan kepada user. Terdapat empat keterangan yang dapat dipilih oleh user yang setiap keterangannya akan diubah dalam bentuk angka. Angka tersebut akan digabungkan dan digunakan dalam perhitungan menggunakan *skala likert* untuk menghasilkan persentase kelayakan dari sistem deteksi.

Tabel 4.3 Hasil Pengujian User Acceptance Test

No.	Poin Pengujian	STS	TS	S	SS
1.	Tata letak layar efisien dan visual yang ditampilkan menarik	0	1	6	3
2.	Pengguna dapat bernavigasi antar halaman dalam aplikasi dengan mudah	0	0	7	3
3.	Aplikasi memberikan tujuan yang jelas dan mendukung tujuan dari aplikasi yang dibuat ini	0	3	5	2
4.	Pengguna dapat dengan mudah menggunakan fitur periksa	0	2	7	1
5.	Aplikasi dapat memberikan hasil periksa dengan cepat	0	0	2	8
6.	Aplikasi dapat memberikan hasil periksa sesuai dengan fakta	0	1	8	1
7.	Aplikasi dapat memberikan informasi yang berguna sesuai dengan hasilnya	0	3	4	3
8.	Pengguna dapat melaporkan kendala yang dialami ketika menggunakan sistem deteksi di form keluhan	0	2	5	3

Rumus total skor :

Total skor = skor SS + skor S + skor TS + skor STS

Total skor = (23 x 4) + (44 x 3) + (12 x 2) + 0 x 1)

Total skor = 92 + 132 + 24 + 0 = 248

Rumus nilai maksimum :

Maksimum = Jumlah responden x nilai tertinggi likert x total pertanyaan

Maksimum = 10 x 4 x 8 = 320

Rumus persentase pengujian:

$$\text{Persentase} = \frac{\text{Total skor}}{\text{Maksimum}} \times 100\%$$

$$\text{Persentase} = \frac{248}{320} \times 100\% = 77,5\%$$

Rumus interval likert :

$$\text{interval} = \frac{100}{\text{Total skor likert}} \times 100\%$$

$$\text{interval} = \frac{100}{4} \times 100\% = 25\%$$

Tabel 4.4 Persentase Skala Likert

No	Persentase	Interpretasi
1	0% - 24%	Sangat Tidak Layak
2	25% - 49%	Tidak Layak
3	50% - 74%	Cukup Layak
4	75% - 100%	Layak

Tabel 4.3 merupakan hasil pengujian yang telah dilakukan kepada *end user* mengenai fungsionalitas dan non fungsionalitas dari sistem deteksi penyakit TBC. Dari delapan pertanyaan yang ditanyakan kepada sepuluh orang *end user*, didapat bahwa hasil pengujian menggunakan metode UAT ini menghasilkan *output* yang ‘Layak’ berdasarkan tabel persentase *skala likert* yang ada pada tabel 4.4 dengan persentase sebesar 77,5%.

4.4.2. Evaluasi Hasil Pengujian

Berdasarkan hasil dari dua pengujian yang telah dilakukan sebelumnya, sistem deteksi yang telah dibuat telah cukup untuk menjawab persyaratan kebutuhan yang harus dipenuhi oleh sistem deteksi penyakit TBC ini. Untuk aspek fungsional, sistem deteksi telah memenuhi fungsi utama dan fungsi pendukung yang dipersyaratkan oleh sistem deteksi. Sementara untuk aspek non-fungsional, sistem deteksi mampu memberikan hasil deteksi dengan cepat, bahkan jauh lebih cepat dari hasil deteksi yang dilakukan secara konvensional. Lalu, sistem deteksi juga berhasil mempermudah pengguna dalam melakukan deteksi penyakit TBC. Pengguna hanya perlu

mengunggah gambar x-ray milik pengguna dan pengguna sudah dapat mengetahui hasil deteksinya. Ditambah dengan navigasi dari aplikasi sistem deteksi yang mudah dipahami sehingga makin mempermudah pengguna untuk menggunakan sistem deteksi penyakit TBC ini.

Namun, terdapat satu pengujian pada pengujian *blackbox testing* yang belum berhasil pada sistem deteksi penyakit TBC ini. Oleh sebab itu, penulis melakukan evaluasi terhadap sistem deteksi yang telah dibuat. Evaluasi yang dilakukan oleh penulis yaitu dengan memberikan pembatasan akses kepada halaman hasil periksa. Pembatasan ini dilakukan dengan tujuan agar halaman hasil periksa hanya bisa diakses jika pengguna telah berhasil mengunggah gambar x-ray miliknya, serta lebih meningkatkan keamanan aplikasi berbasis *website* dari sistem deteksi penyakit TBC ini. Cara yang dilakukan penulis yaitu dengan memberikan akses untuk membuka halaman hasil periksa hanya jika pengguna menekan tombol **OK** yang terdapat pada *alert* yang muncul sesaat setelah gambar x-ray berhasil diunggah. Dengan ini, pengguna hanya akan dapat masuk ke halaman hasil deteksi apabila telah berhasil mengunggah gambar x-ray nya dan menekan tombol **OK**.

Selanjutnya pada pengujian UAT kepada *end user*, evaluasi yang dapat dilakukan oleh penulis untuk poin yaitu 'sistem dapat memberikan informasi yang lebih memberikan manfaat kepada pengguna'. Sistem juga seharusnya dapat memberikan informasi mengenai langkah – langkah preventif untuk mencegah penularan yang mungkin dapat terjadi kepada orang – orang disekitar pengguna. Namun, untuk evaluasi yang ada pada poin pertama yaitu sistem deteksi pada suatu waktu masih belum memberikan hasil deteksi sesuai dengan fakta atau realitanya. Disini, penulis hanya dapat memberikan evaluasi sedikit untuk poin tersebut karena poin tersebut berhubungan dengan model *deep learning* yang dibuat menggunakan metode CNN yang masuk ke dalam batasan masalah dari penelitian ini. Penulis juga hanya dapat memberikan masukan untuk dapat mengembangkan lagi model *deep learning* yang sudah ada sehingga dapat meminimalisir kesalahan dalam memberikan hasil deteksinya.

BAB V

KESIMPULAN DAN SARAN

Pada bab ini sebagai penutup akan dijelaskan kesimpulan dan saran atas hasil yang telah dilakukan pada tahap sebelumnya yaitu implementasi dan perancangan. Kesimpulan adalah jawaban dari hasil rumusan masalah yang dijelaskan pada awal bagian tulisan ini. Sedangkan, saran adalah masukan yang diberikan untuk tahap pengembangan selanjutnya untuk sistem deteksi ini.

5.1. Kesimpulan

Berdasarkan dari hasil pengujian dan evaluasi yang telah dilakukan sebelumnya, tujuan penelitian dapat tercapai serta dapat menjawab rumusan masalah yang telah didefinisikan pada bagian awal penelitian. Berikut ini dapat diperoleh kesimpulan bahwa:

- Proses pengembangan sistem deteksi penyakit TBC dilakukan dengan beberapa tahapan mulai dari pendefinisian sistem, pembuatan desain, implementasi, pengujian serta evaluasi dari hasil pengujian.
- Sistem deteksi yang telah dibuat telah memenuhi seluruh persyaratan, mulai dari persyaratan fungsional, non-fungsional, hingga persyaratan teknis berhasil dipenuhi oleh sistem deteksi. Hal ini dibuktikan dari hasil pengujian yang dilakukan pada sistem deteksi penyakit TBC menggunakan metode *blackbox testing* serta *User Acceptance Test*.
 - Pengujian menggunakan *blackbox testing* menunjukkan tingkat keberhasilan pengujian sebesar 87.5%. Hal ini menunjukkan bahwa sistem deteksi mampu dalam melakukan prediksi penyakit TBC melalui gambar x-ray yang diunggah oleh pengguna. Namun dengan catatan, hasil deteksi yang dikeluarkan dari sistem deteksi ini hanya dapat dijadikan sebagai referensi saja dan tidak boleh digunakan sebagai dasar diagnosis atau pengobatan medis.
 - Pengujian menggunakan UAT kepada *end user* berjalan dengan cukup baik dengan tingkat keberhasilan sebesar 75%. Hasil pengujian ini memberikan informasi bahwa sistem deteksi yang telah dibuat oleh penulis dapat diterima dengan baik dan

memenuhi persyaratan kebutuhan yang diinginkan oleh pengguna.

- Cara kerja dari sistem deteksi penyakit TBC ini yaitu dimulai dengan pengguna mengunggah gambar x-ray bagian dada yang ingin diperiksa. Setelah berhasil diunggah, sistem akan mengirimkan gambar x-ray tersebut ke API untuk diklasifikasi oleh model *deep learning* yang telah dikembangkan dengan metode CNN. Setelah berhasil diklasifikasi, API akan mengembalikan hasilnya ke sistem untuk disimpan ke *database* dan ditampilkan ke halaman hasil periksa.

5.2. Saran

Saran merupakan masukan untuk pengembangan selanjutnya agar sistem yang telah dibuat dapat berkembang lebih baik lagi. Berikut beberapa saran yang diberikan agar sistem yang telah diteliti ini dapat berkembang lebih baik lagi:

- Sistem perlu menambahkan fitur - fitur baru yang dapat bermanfaat bagi pengguna. Fitur seperti layanan konsultasi terkait penyakit TBC akan membantu pengguna dalam membuat keputusan yang lebih baik.
- Sistem perlu memberikan contoh kepada *user* bagaimana gambar x-ray yang baik dan yang kurang baik sehingga pengguna dapat lebih mudah dalam mengambil gambar x-ray yang dimilikinya.
- Sistem perlu memberikan lebih banyak visualisasi mengenai hasil deteksi yang dikeluarkan oleh sistem deteksi sehingga informasi yang didapat dari hasil deteksi menjadi lebih banyak lagi.
- Sistem dapat memberikan informasi – informasi tambahan mengenai penyakit TBC seperti cara penyebarannya, pengobatannya, dan hal lain yang berkaitan dengan penyakit TBC.

DAFTAR PUSTAKA

- [1] O. Rochmawanti, F. Utamingrum, and F. A. Bachtiar, “ANALISIS PERFORMA PRE-TRAINED MODEL CONVOLUTIONAL NEURAL NETWORK DALAM MENDETEKSI PENYAKIT TUBERKULOSIS”, doi: 10.25126/jtiik.202184441.
- [2] D. S. Wahyuni, Aswasulasikin, and A. Aziz, “DETEKSI DINI SISWA BERKEBUTUHAN KHUSUS DI SEKOLAH DASAR NEGERI 1 KESIK,” 2023.
- [3] A. A. Agyeman and R. Ofori-Asenso, “Tuberculosis—an overview,” *J Public Health Emerg*, vol. 1, pp. 7–7, Jan. 2017, doi: 10.21037/jphe.2016.12.08.
- [4] TBC Indonesia, “Apakah kalian tahu apa itu TBC ?” Accessed: Feb. 15, 2024. [Online]. Available: <https://tbindonesia.or.id/apakah-kalian-tahu-apa-itu-tbc/>
- [5] N. P. Ekananda and D. Riminarsih, “IDENTIFIKASI PENYAKIT PNEUMONIA BERDASARKAN CITRA CHEST X-RAY MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK,” *Jurnal Ilmiah Informatika Komputer*, vol. 27, no. 1, pp. 79–94, 2022, doi: 10.35760/ik.2022.v27i1.6487.
- [6] Dicoding, “Convolutional Neural Network.” Accessed: Feb. 14, 2024. [Online]. Available: <https://www.dicoding.com/academies/184/tutorials/8512>
- [7] Diego Unzueta, “Fully Connected Layer vs. Convolutional Layer: Explained.”
- [8] IBM, “What is a REST API?” Accessed: Mar. 30, 2024. [Online]. Available: <https://www.ibm.com/topics/rest-apis#:~:text=the%20next%20step-,What%20is%20a%20REST%20API%3F,representational%20state%20transfer%20architectural%20style.>
- [9] FastAPI, “FastAPI Documentation.” Accessed: Jan. 30, 2024. [Online]. Available: <https://fastapi.tiangolo.com/>
- [10] D. Purnama Sari, R. Wijanarko, and J. X. Menoreh Tengah, “Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang),” vol. 2, no. 1, pp. 32–36, 2019.
- [11] T. Hidayat, H. D. Putri, and A. Uk, “Pengujian Portal Mahasiswa pada Sistem Informasi Akademik (SINA) menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis CORE View metadata,

citation and similar papers at core,” *JUTIS*, vol. 7, no. 1, 2019, [Online].

Available: www.ccsenet.org/cis

- [12] N. Made, D. Febriyanti, A. A. Kompiang, O. Sudana, and N. Piarsa, “Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen,” 2021.
- [13] I. Wahyudi and F. Alameka, “ANALISIS BLACKBOX TESTING DAN USER ACCEPTANCE TESTING TERHADAP SISTEM INFORMASI SOLUSIMEDSOSKU,” *Jurnal Teknosains Kodepena /*, vol. 04, pp. 1–9, 2023.
- [14] B. W. Widagdo, M. Handayani, and D. A. Suharto, “DAMPAK PANDEMI COVID-19 TERHADAP PERILAKU PESERTA DIDIK PADA PROSES PEMBELAJARAN DARING MENGGUNAKAN METODE PENGUKURAN SKALA LIKERT (Studi Kasus di Kabupaten Tangerang Selatan).”
- [15] M. Harahap, Em Manuel Laia, Lilis Suryani Sitanggung, Melda Sinaga, Daniel Franci Sihombing, and Amir Mahmud Husein, “Deteksi Penyakit Covid-19 Pada Citra X-Ray Dengan Pendekatan Convolutional Neural Network (CNN),” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, pp. 70–77, Feb. 2022, doi: 10.29207/resti.v6i1.3373.
- [16] “Klasifikasi Penyakit Paru-Paru Berbasis Pengolahan Citra X Ray Menggunakan Convolutional Neural Network (Classification Of The Lung Diseases Based Ona X Ray Image Processing Using Convolutional Neural Network).”
- [17] Brenda Monica Rotty, Angelia Melani Adrian, and Junaidy Budi Sanger, “Aplikasi Prediksi Penyakit Tuberkulosis Menggunakan Algoritma Convolutional Neural Network (CNN),” *UNIVERSITAS KATOLIK DE LA SALLE.*, 2020.
- [18] I. Latin and A. Transactions, “COVID-XR: A Web Management Platform for Coronavirus Detection on X-Ray Chest Images,” 2021. [Online]. Available: <http://di.unsa.edu.ar>
- [19] A. Maydiantoro, “Model-Model Penelitian dan Pengembangan,” 2022.
- [20] T. Rahman *et al.*, “Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization,” *IEEE Access*, vol. 8, pp. 191586–191601, 2020, doi: 10.1109/ACCESS.2020.3031384.