

**Aplikasi Web Skrining TBC Berbasis Flask: Implementasi Algoritma
Prediksi dan UI Interaktif**

Skripsi

**Sebagai Salah Satu Syarat untuk
memperoleh gelar Sarjana S1 dari
Universitas Lampung**

Oleh

**Andhika Akbar Pratama
NPM: 1915061043
Program Studi Teknik Informatika**



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2026**

ABSTRAK

Aplikasi Web Skrining TBC Berbasis Flask: Implementasi Algoritma Prediksi dan UI Interaktif

Oleh

Andhika Akbar Pratama

Tuberkulosis (TBC) merupakan salah satu penyakit menular yang masih menjadi permasalahan kesehatan masyarakat di Indonesia. Keterbatasan tenaga kesehatan dan luasnya wilayah pelayanan menyebabkan proses skrining TBC belum dapat dilakukan secara optimal. Oleh karena itu, diperlukan sebuah sistem pendukung yang mampu membantu proses skrining awal secara lebih cepat, terstruktur, dan mudah diakses oleh masyarakat serta petugas kesehatan.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem skrining TBC berbasis web yang terintegrasi dengan metode machine learning. Sistem ini dikembangkan menggunakan framework Flask sebagai backend, PostgreSQL sebagai basis data, serta Nginx dan Gunicorn sebagai web server. Model machine learning yang telah dilatih sebelumnya disimpan dalam bentuk file .pkl dan digunakan untuk memprediksi tingkat risiko TBC berdasarkan data gejala yang diinput oleh pengguna.

Sistem yang dibangun terdiri dari dua area akses utama, yaitu area skrining publik yang dapat diakses oleh masyarakat umum untuk melakukan skrining mandiri, serta area SIDAK yang diperuntukkan bagi kader kesehatan untuk melakukan pemantauan, pengelolaan data, dan rekapitulasi hasil skrining.

Hasil pengujian sistem menggunakan metode black-box testing menunjukkan bahwa seluruh fungsi utama sistem berjalan sesuai dengan kebutuhan yang telah ditentukan. Dengan adanya sistem ini, diharapkan proses skrining TBC dapat dilakukan secara lebih efisien dan membantu kader kesehatan dalam melakukan pemantauan serta pengambilan keputusan awal terkait risiko TBC di masyarakat.

Kata kunci: Tuberkulosis, Skrining TBC, Aplikasi Web, Flask, Machine Learning.

ABSTRACT

Flask-Based Web Application for Tuberculosis Screening: Implementation of a Prediction Algorithm and Interactive UI

By

Andhika Akbar Pratama

Tuberculosis (TB) remains one of the major infectious diseases and continues to be a public health problem in Indonesia. The limited number of healthcare workers and the wide coverage area of health services make the TB screening process difficult to carry out optimally. Therefore, a supporting system is needed to assist the initial screening process in a faster, more structured, and more accessible manner for both the public and healthcare workers.

This study aims to design and implement a web-based TB screening system integrated with a machine learning approach. The system is developed using the Flask framework as the backend, PostgreSQL as the database, and Nginx and Gunicorn as the web server. A pre-trained machine learning model is stored in a .pkl file format and is used to predict the level of TB risk based on symptom data input by users.

The developed system consists of two main access areas: a public screening area that can be accessed by the general public for self-screening, and a SIDAK area intended for health cadres to perform monitoring, data management, and recapitulation of screening results.

System testing using the black-box testing method shows that all main system functions operate according to the specified requirements. With the implementation of this system, it is expected that the TB screening process can be carried out more efficiently and can assist health cadres in monitoring and making early decisions regarding TB risk in the community.

Keywords: Tuberculosis, TB Screening, Web Application, Flask, Machine Learning.

Judul Skripsi

APLIKASI WEB SKRINING TBC
BERBASIS FLASK : IMPLEMENTASI
ALGORITMA PREDIKSI DAN UI
INTERAKTIF

Nama Mahasiswa

: **Andhika Akbar Pratama**

Nomor Pokok Mahasiswa

: 1915061043

Program Studi

: S1 Teknik Informatika

Jurusan

: Teknik Elektro

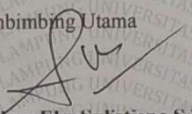
Fakultas

: Teknik

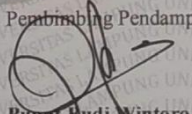
MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

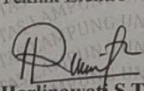

Wahyu Eko Sulistiono, S.T., M.Sc.
NIP. 197412012001121001

Pembimbing Pendamping

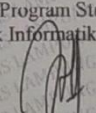

Papat Budi Wintoro, S.Kom., M.T.I.
NIP. 198410312019031004

2. Mengetahui

Ketua Jurusan
Teknik Elektro


Herlinawati, S.T., M.T.
NIP. 197103141999032001

Ketua Program Studi
Teknik Informatika


Yessi Mulyani, S.T., M.T.
NIP. 19731226200012200

MENGESAHKAN

1. Tim Penguji

Ketua : Wahyu Eko Sulistiono, S.T., M.Sc.

Sekretaris : Puput Budi Wintoro, S.Kom., M.T.I.

Penguji : Mona Arif Muda, S.T., M.T.



2. Dekan Fakultas Teknik

Dr. H. Ahmad Herison, S.T., M.T.
NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi : 30 Januari 2026

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan dengan sebenarnya bahwa skripsi saya yang berjudul "Aplikasi Web Skrining TBC Berbasis Flask: Implementasi Algoritma Prediksi dan UI Interaktif" merupakan hasil karya saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 26 Februari 2026
Pembuat Pernyataan



Andhika Akbar Pratama
1915061043

RIWAYAT HIDUP



Penulis dilahirkan pada tanggal 1 September 2000 di Jakarta sebagai anak pertama dari pasangan Ismed Nalfi dan Ermi Yanti. Penulis memulai pendidikan formal di SDN Ciracas 06 Pagi, kemudian melanjutkan pendidikan menengah pertama di SMPN 171 Jakarta Timur.

Pendidikan menengah atas ditempuh di SMKN 2 Jakarta dengan mengambil jurusan Rekayasa Perangkat Lunak.

Setelah menyelesaikan pendidikan menengah, penulis melanjutkan studi pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Lampung. Selama masa perkuliahan, penulis menaruh minat pada bidang pengembangan aplikasi dan sistem informasi. Ketertarikan tersebut kemudian diwujudkan dalam penyusunan skripsi yang berjudul *“Aplikasi Web Skrining TBC Berbasis Flask: Implementasi Algoritma Prediksi dan UI Interaktif”* sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik.

Penulis berharap hasil penelitian dan pengembangan sistem yang dilakukan dalam skripsi ini dapat memberikan manfaat, khususnya dalam mendukung pemanfaatan teknologi informasi di bidang kesehatan.

MOTTO

“Adapt, endure, and overcome.”

“True strength is found through patience and discipline.”

PERSEMBAHAN

Skripsi ini penulis persembahkan kepada kedua orang tua tercinta, yang senantiasa memberikan doa, kasih sayang, dukungan, serta pengorbanan yang tiada henti dalam setiap langkah perjalanan penulis.

Penulis juga mempersembahkan karya ini kepada para dosen pembimbing dan seluruh civitas akademika Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung, atas bimbingan, arahan, dan ilmu yang telah diberikan selama masa perkuliahan hingga terselesaikannya skripsi ini.

Tak lupa, skripsi ini penulis persembahkan kepada teman-teman seperjuangan, baik seangkatan maupun tidak seangkatan, yang telah memberikan semangat, bantuan, serta kebersamaan dalam menjalani proses perkuliahan dan penyusunan skripsi ini.

SANWACANA

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi yang berjudul “Aplikasi Web Skrining TBC Berbasis Flask: Implementasi Algoritma Prediksi dan UI Interaktif” sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

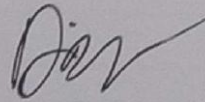
Dalam penyusunan skripsi ini, penulis menyadari bahwa prosesnya tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. H. Ahmad Herison, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Lampung.
2. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
3. Ibu Yessi Mulyani, S.T., M.T., selaku Koordinator Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung.
4. Bapak Wahyu Eko Sulistiono, S.T., M.Sc., selaku Dosen Pembimbing I, yang telah memberikan bimbingan, arahan, serta motivasi selama proses penyusunan skripsi ini.
5. Bapak Puput Budi Wintoro, S.Kom., M.T.I., selaku Dosen Pembimbing II, yang telah memberikan masukan, koreksi, dan saran yang sangat berarti bagi penyempurnaan skripsi ini.
6. Seluruh dosen dan staf di lingkungan Fakultas Teknik Universitas Lampung yang telah memberikan ilmu, bantuan, dan fasilitas selama penulis menempuh pendidikan.

7. Kedua orang tua dan keluarga tercinta, atas doa, dukungan, serta pengorbanan yang tiada henti kepada penulis.
8. Teman-teman seperjuangan, baik seangkatan maupun tidak seangkatan, yang telah memberikan semangat, bantuan, dan kebersamaan selama masa perkuliahan dan penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan dan jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan saran dan kritik yang membangun demi perbaikan di masa yang akan datang. Semoga skripsi ini dapat memberikan manfaat bagi pembaca dan dapat menjadi salah satu kontribusi dalam pengembangan ilmu pengetahuan, khususnya di bidang teknologi informasi dan sistem informasi kesehatan.

Bandar Lampung, 26 Februari 2026
Penulis



Andhika Akbar Pratama

DAFTAR ISI

DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL	xxii
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	5
1.5 Sistematika Penulisan Skripsi	5
II. TINJAUAN PUSTAKA.....	7
2.1 Bootstrap.....	7
2.2 HTML.....	7
2.3 Flask	8
2.4 Python.....	9
2.5 Tuberculosis (TBC).....	10
2.6 PostgreSQL	12
2.7 System Usability Scale (SUS).....	13
2.9 Black Box Testing	13
2.10 Waterfall.....	15
2.11 Unified Modeling Language (UML).....	16
2.12 Entity Relationship Diagram (ERD)	21
2.13 Postman	23
2.14 OWASP ZAP	23

2.15 Xgboost	24
2.16 Penelitian Terkait	25
III. METODOLOGI PENELITIAN	27
3.1 Tempat dan Waktu Penelitian	27
3.1.1 Tempat Penelitian	27
3.2 Alat dan Bahan	28
3.2.1 Alat.....	28
3.2.2 Bahan	29
3.3 Tahapan Penelitian	29
3.3.1 Analisis Kebutuhan.....	30
3.3.2 Merancang Desain Sistem	31
3.3.3 Implementasi.....	32
3.3.4 Pengujian	33
3.3.5 Deployment Sistem dan Evaluasi	34
3.3.6 Pemeliharaan.....	38
IV. HASIL DAN PEMBAHASAN.....	40
4.1 Analisis Kebutuhan	40
4.1.1 Wawancara.....	40
4.2 Perancangan Desain Sistem.....	43
4.2.1 Arsitektur Sistem	43
4.2.3 Use Case Diagram	46
4.2.4 Activity Diagram	49
4.2.4.1 Website Skrining TBC	49
4.2.4.2 SIDAK (Sistem Database Kader).....	57
4.2.5 <i>Wireframe</i>	69
4.2.5.1 Halaman Skrining TBC.....	69

4.2.5.1 Halaman Sistem Database Kader (SIDAK)	74
4.2.6 <i>Class Diagram Sistem</i>	79
4.3 Implementasi Sistem	82
4.3.1 Implementasi Backend	82
4.3.1.1 Konfigurasi Lingkungan Pengembangan	83
4.3.1.2 Manajemen Pengguna	85
4.3.1.3 Autentikasi Dan Login	95
4.3.1.4 Validasi Token dan Ubah Password	99
4.3.1.5 Konfigurasi Dan Inisialisasi SMTP	104
4.3.1.6 Manajemen Profil Pengguna	105
4.3.1.7 Modul Skrining Tuberkulosis	109
4.3.1.8 Manajemen Data Responden	116
4.3.1.9 Logging dan Error Handling	117
4.3.2 Implementasi Frontend	119
4.3.2.1 Konfigurasi Lingkungan <i>Frontend</i>	120
4.3.2.2 Manajemen Antarmuka Pengguna	123
4.3.2.3 Manajemen Autentikasi dan Navigasi	125
4.3.2.4 Antarmuka Manajemen Profil Pengguna	133
4.3.2.5 Antarmuka Skrining TBC	141
4.3.2.6 Dashboard Kader & Statistik	148
4.3.4 Implementasi Keamanan Sistem	159
4.3.4.1 <i>Hashing Password</i>	159
4.3.4.2 Proteksi CSRF (Cross-Site Request Forgery)	160
4.3.4.3 Rate Limiting pada Proses Login	162
4.3.4.4 Penggunaan Hashids pada URL	163
4.3.4.5 Pembatasan Akses Berdasarkan Role (RBAC)	165

4.3.4.6 Manajemen <i>Session Login</i>	166
4.4 Pengujian	169
4.4.1 Uji Fungsionalitas	170
4.4.1.1 Pengujian Fitur Responden	170
4.4.1.2 Pengujian Fitur Kader	174
4.4.2 Pengujian <i>Endpoint API</i>	184
4.4.2.1 Pengujian Fitur Responden	184
4.4.2.2 Pengujian Fitur Kader	192
4.4.3 Uji Keamanan	203
4.4.3.1 Metode Pengujian (OWASP ZAP)	204
4.4.3.2 Konfigurasi dan Lingkungan Pengujian.....	204
4.4.3.3 Hasil Pengujian	205
4.5 <i>Deployment</i>	211
4.5.1 Persiapan Lingkungan Deployment.....	211
4.5.1.1 Lingkungan Server Production	211
4.5.2 Upload dan Konfigurasi Berkas Sistem.....	212
4.5.3 Konfigurasi Database di Server	212
4.5.4 Konfigurasi SMTP	212
4.6 Evaluasi Sistem Menggunakan Survey Pengguna	213
4.6.1 Penilaian <i>Usability</i> Oleh Responden	214
4.6.2 Penilaian <i>Usability</i> Oleh Kader	215
4.6.3 Evaluasi Tingkat Kepuasan Pengguna.....	216
4.7 Pemeliharaan	217
V. KESIMPULAN DAN SARAN	223
5.1 Kesimpulan.....	223
5.2 Saran	224

Daftar Pustaka.....	225
LAMPIRAN.....	228

DAFTAR GAMBAR

Gambar 2.1 Notasi Use Case Diagram	17
Gambar 2.2 Notasi Activity Diagram.	18
Gambar 2.3 Notasi Sequence Diagram	19
Gambar 2.4 Notasi Class Diagram.....	20
Gambar 2.5 Notasi Chen.	21
Gambar 2.6 Notasi Crow's Foot.	22
Gambar 2.7 Notasi UML.....	22
Gambar 3.1 Tahapan Penelitian Menggunakan Metode Waterfall.....	30
Gambar 4.1 Arsitektur Sistem.....	44
Gambar 4.2 Use Case Diagram Responden.	46
Gambar 4.3 Use Case Diagram Kader.	47
Gambar 4.4 Use Case Diagram Super Kader.....	48
Gambar 4.5 Activity Diagram Registrasi Akun.....	50
Gambar 4.6 Activity Diagram Login.	51
Gambar 4.7 Activity Diagram Fitur Lupa Password.	52
Gambar 4.8 Activity Diagram Melihat Profil.	53
Gambar 4.9 Activity Diagram Mengisi Form Skrining.	54
Gambar 4.10 Activity Diagram Model ML.	55
Gambar 4.11 Activity Diagram Melihat Rekap Skrining Terakhir.....	56
Gambar 4.12 Activity Diagram Login Kader.	57
Gambar 4.13 Activity Diagram Melihat Profil.	58
Gambar 4.14 Activity Diagram Melihat Dashboard Skrining.	59
Gambar 4.15 Activity Diagram Melihat Daftar Responden.	60
Gambar 4.16 Activity Diagram Melihat Grafik Statistik.....	61
Gambar 4.17 Activity Diagram Melihat Daftar Kader.	62
Gambar 4.18 Activity Diagram Menambah Kader.....	63
Gambar 4.19 Activity Diagram Mengedit Data Kader Lain.....	64
Gambar 4.20 Activity Diagram Generate KTA.	65
Gambar 4.21 Activity Diagram Import Data Kader (.csv).....	66
Gambar 4.22 Activity Diagram Kader Export Data Responden.....	67

Gambar 4.23 Kader Export Data Kader.....	68
Gambar 4.24 Wireframe Low Fidelity Halaman Utama Website.....	70
Gambar 4.25 Wireframe Low Fidelity Halaman Login.....	71
Gambar 4.26 Wireframe Low Fidelity Halaman Register.....	72
Gambar 4.27 Wireframe Low Fidelity Halaman Form Skrining.....	73
Gambar 4.28 Wireframe Low Fidelity Halaman Hasil Skrining.....	74
Gambar 4.29 Wireframe Low Fidelity Halaman Login Kader.....	75
Gambar 4.30 Wireframe Low Fidelity Dashboard Kader.....	76
Gambar 4.31 Wireframe Low Fidelity Halaman Daftar Responden.....	77
Gambar 4.32 Wireframe Low Fidelity Halaman Statistik.....	77
Gambar 4.33 Wireframe Low Fidelity Halaman List Kader.....	78
Gambar 4.34 Class Diagram Sistem Web Skrining TBC.....	79
Gambar 4.35 Konfigurasi Database pada .env.....	83
Gambar 4.36 Library Yang Digunakan Untuk Pengembangan.....	84
Gambar 4.37 Struktur Folder Backend.....	85
Gambar 4.38 property role user.....	88
Gambar 4.39 property role kader dan super_kader.....	90
Gambar 4.40 Source Code Decorator Role.....	91
Gambar 4.41 Contoh Penggunaan Otorisasi Role.....	92
Gambar 4.42 Source Code Registrasi Akun Responden.....	93
Gambar 4.43 Source Code Tambah Kader.....	94
Gambar 4.44 Source Code Login Responden.....	96
Gambar 4.45 Source Code Login Kader.....	97
Gambar 4.46 Contoh Potongan Kode Menggunakan hashing.....	98
Gambar 4.47 Source Code Logout.....	99
Gambar 4.48 Source Code Token Aktivasi.....	100
Gambar 4.49 Source code Update Profile Responden.....	101
Gambar 4.50 Source Code Lupa Password dan Token Reset password.....	102
Gambar 4.51 Source Code Ubah Password Kader.....	103
Gambar 4.52 Konfigurasi .env SMTP.....	104
Gambar 4.53 Potongan Kode Inisialisasi SMTP.....	105
Gambar 4.54 Source Code Model RegistrationForm.....	106

Gambar 4.55 Source Code Edit Profil Kader.....	107
Gambar 4.56 Source Code Upload Foto Kader.	108
Gambar 4.57 Source Code model SkriningForm.....	110
Gambar 4.58 Source Code endpoint /skrining/.....	111
Gambar 4.59 Source Code Load Model ML.....	112
Gambar 4.60 Pendefinisian binary_columns.	113
Gambar 4.61 Source Code set_rujukan.....	113
Gambar 4.62 Source Code Hasil Skrining.	114
Gambar 4.63 Source Code Rekap Riwayat Skrining.....	115
Gambar 4.64 Source Code Chart Keterdugaan TBC.	116
Gambar 4.65 Source Code Logging Sistem.....	118
Gambar 4.66 Contoh Error Logging Pada Sistem.	118
Gambar 4.67 Error Handling Pada Halaman Kader.....	119
Gambar 4.68 Struktur Folder Frontend.....	121
Gambar 4.69 Source Code Registrasi Blueprints.....	122
Gambar 4.70 Implementasi fungsi render_template().	123
Gambar 4.71 Pemanfaatan url_for() untuk meload assets.	123
Gambar 4.72 Templating Menggunakan base.html.	124
Gambar 4.73 Komponen opsional pada kader_views.....	125
Gambar 4.74 Halaman Utama Website Skrining TBC.	126
Gambar 4.75 Formulir Pendaftaran Web Skrining TBC.	127
Gambar 4.76 Informasi Aktivasi Akun Responden.	127
Gambar 4.77 Email Aktivasi Akun Web Skrining TBC.....	128
Gambar 4.78 Pop-up Akun Berhasil Diaktivasi.....	128
Gambar 4.79 Formulir Login Responden.	129
Gambar 4.80 Login Berhasil.	129
Gambar 4.81 Login Gagal.....	130
Gambar 4.82 Halaman Utama Setelah Login.	130
Gambar 4.83 Halaman Login Kader.	131
Gambar 4.84 Login Kader Sukses.	131
Gambar 4.85 Halaman Dashboard Kader.	132
Gambar 4.86 Halaman Profil Responden.....	133

Gambar 4.87 Modal Edit Profil Responden.....	134
Gambar 4.88 Pop-up Sukses Mengubah Profil.....	134
Gambar 4.89 Halaman Profil Kader.....	135
Gambar 4.90 Modal Edit Profil Kader.....	136
Gambar 4.91 Pop-up Sukses Edit Profil.....	136
Gambar 4.92 Modal Ubah Password Kader.....	137
Gambar 4.93 pop-up kader berhasil mengubah password.....	137
Gambar 4.94 Pop-up Password Lama Salah.....	138
Gambar 4.95 Modal Mengganti Foto Profil Kader.....	138
Gambar 4.96 Pop-up Berhasil Memperbarui Foto Profil Kader.....	139
Gambar 4.97 Modal Generate KTA Kader.....	139
Gambar 4.98 Kartu Tanda Anggota ILS.....	140
Gambar 4.99 Form skrining TBC Pada Website.....	141
Gambar 4.100 Hasil Skrining Responden Terduga TBC.....	142
Gambar 4.101 Hasil Skrining Responden Tidak Terduga TBC.....	143
Gambar 4.102 Halaman Hasil Skrining Terduga TBC Bersedia Rujuk.....	144
Gambar 4.103 Hasil Skrining Terduga Tetapi Kader Tidak Tersedia.....	145
Gambar 4.104 Hasil Skrining TBC Memilih Tidak Bersedia Rujukan.....	146
Gambar 4.105 Rekap Skrining Jawaban Responden.....	147
Gambar 4.106 Rekomendasi Berdasarkan Domisili Responden.....	148
Gambar 4.107 Grafik Perbandingan Keterdugaan TBC Per Wilayah.....	149
Gambar 4.108 Grafik Perbandingan Rujukan Responden Per Wilayah.....	150
Gambar 4.109 Halaman Daftar Responden.....	150
Gambar 4.110 Hasil Ekspor Data Responden Terfilter.....	151
Gambar 4.111 Halaman Detail Responden.....	152
Gambar 4.112 Ekspor Detail Responden.....	152
Gambar 4.113 Halaman List Kader.....	153
Gambar 4.114 Form Edit Data Kader.....	154
Gambar 4.115 Pop-up Data Kader Berhasil Diperbarui.....	154
Gambar 4.116 Pop-up Peringatan Penghapusan Data Kader.....	155
Gambar 4.117 Pop-up Hapus Data Kader Berhasil.....	155
Gambar 4.118 Form Tambah Kader.....	156

Gambar 4.119 Pop-up Berhasil menambah Kader.....	157
Gambar 4.120 Modal Pilih File Data Kader.	157
Gambar 4.121 Pop-up Sukses Import Data Kader.	158
Gambar 4.122 Password Tersimpan Dalam Database.	159
Gambar 4.123 Pengaplikasian fungsi <code>check_password_hash()</code>	160
Gambar 4.124 Penggunaan <code>form.hidden_tag()</code> Pada Form Skrining.	161
Gambar 4.125 Inisialisasi Form Untuk Digunakan Pada Frontend.	161
Gambar 4.126 Pengaplikasian dekorator <code>@limiter.limit()</code>	162
Gambar 4.127 URL yang sudah dimasking menggunakan Hashids.....	163
Gambar 4.128 Inisialisasi Hashids.....	164
Gambar 4.129 Pengimplementasian Hashids Pada URL Hasil Skrining.....	164
Gambar 4.130 Pengimplementasian Dekorator Role.....	165
Gambar 4.131 Decorator Role <code>@super_kader_required</code>	166
Gambar 4.132 Kick Sesi Kader Yang Sedang Berjalan,.....	168
Gambar 4.133 Pop-up Sesi Berakhir Untuk Kader.....	169
Gambar 4.134 Request endpoint <code>/register</code>	187
Gambar 4.135 Response Endpoint <code>/register</code> Sukses.....	187
Gambar 4.136 Request Endpoint <code>/login</code>	188
Gambar 4.137 Response Endpoint <code>/login</code>	188
Gambar 4.138 Request Endpoint <code>/skrining</code>	189
Gambar 4.139 Response Endpoint <code>/skrining</code>	190
Gambar 4.140 Request Endpoint <code>/set_rujukan</code>	191
Gambar 4.141 Response Endpoint <code>/set_rujukan</code>	191
Gambar 4.142 Endpoint <code>/hasil_skrining/<hashid_history></code>	192
Gambar 4.143 Request Endpoint <code>/kader/login</code>	196
Gambar 4.144 Response Endpoint <code>/kader/login</code>	196
Gambar 4.145 Response Endpoint <code>/kader/upload-profile-photo</code>	197
Gambar 4.146 Request Endpoint <code>/kader/kta/generate_with_photo</code>	197
Gambar 4.147 Response Endpoint <code>/kader/kta/generate_with_photo</code>	198
Gambar 4.148 Request Endpoint <code>/kader/chart-wilayah/data</code>	199
Gambar 4.149 Response Endpoint <code>/kader/chart-wilayah/data</code>	199
Gambar 4.150 Parameter Filter Endpoint <code>/kader/users-list/data</code>	200

Gambar 4.151 Response Endpoint /kader/users-list/data.....	201
Gambar 4.152 Parameter Endpoint /kader/user-detail/<hashid_user>/data.	202
Gambar 4.153 Response Endpoint /kader/user-detail/<hashid_user>/data.	203
Gambar 4.154 Konfigurasi Awal Pengujian Menggunakan ZAP.....	205
Gambar 4.155 Akses Situs Tanpa VPN.	206
Gambar 4.156 Akses Situs Menggunakan VPN.	207
Gambar 4.157 Alerts Dari Hasil Scan ZAP.	208
Gambar 4.158 Report Hasil Scanning.....	209
Gambar 4.159 Halaman Connect Domain Hostinger.	213
Gambar 4.160 Pesan Error Ditampilkan dalam Format Default.....	218
Gambar 4.161 Code Sebelum Fix Decorator Role.....	219
Gambar 4.162 Error Connection Refused Port SMTP.....	220
Gambar 4.163 Info Ukuran Penyimpanan Data Server.	221

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	25
Tabel 3.1 Jadwal Penelitian.....	27
Tabel 3.2 Alat Penelitian.....	28
Tabel 3.3 Bahan Penelitian	29
Tabel 3.4 Contoh Pertanyaan SUS.....	35
Tabel 3.5 Contoh Pertanyaan Likert Scale.....	37
Tabel 4.1 Ringkasan Hasil Wawancara	40
Tabel 4.2 Deskripsi Use Case Diagram Responden	46
Tabel 4.3 Deskripsi Use Case Kader	48
Tabel 4.4 Penjelasan Entity Relationship Diagram.....	80
Tabel 4.5 Field Pada Model User.....	86
Tabel 4.6 Field Pada Model DataKader.....	88
Tabel 4.7 Hasil Pengujian Fitur Halaman Website Utama	170
Tabel 4.8 Hasil Pengujian Fitur Sistem Database Kader (SIDAK)	174
Tabel 4.9 Hasil Pengujian Endpoint API Fitur Responden.....	185
Tabel 4.10 Daftar Endpoint API Fitur Kader.....	193
Tabel 4.11 Parameter Database.....	212
Tabel 4.12 Skor SUS Responden	214
Tabel 4.13 Skor SUS Kader	215
Tabel 4.14 Hasil Evaluasi Tingkat Kepuasan Pengguna	217

I. PENDAHULUAN

1.1 Latar Belakang

Tuberkulosis (TBC) merupakan salah satu penyakit infeksi yang masih menjadi tantangan serius dalam dunia kesehatan, baik secara global maupun di Indonesia. Berdasarkan data terbaru dari Organisasi Kesehatan Dunia (WHO), pada tahun 2023 Indonesia menempati peringkat ketiga sebagai negara dengan beban kasus TBC tertinggi di dunia, dengan estimasi lebih dari 820.000 kasus baru setiap tahunnya. Kondisi ini menunjukkan bahwa upaya deteksi dan penanganan dini terhadap TBC menjadi sangat krusial untuk menekan penyebaran penyakit dan meningkatkan angka kesembuhan.

TBC disebabkan oleh bakteri *Mycobacterium tuberculosis*, yang umumnya menyerang paru-paru namun juga dapat menginfeksi organ lain. Salah satu permasalahan utama dalam pengendalian TBC adalah keterlambatan diagnosis yang disebabkan oleh gejala awal yang tidak spesifik, seperti batuk berkepanjangan, demam ringan, kelelahan, dan penurunan berat badan. Gejala-gejala ini sering diabaikan oleh masyarakat, sehingga pasien baru mencari pengobatan setelah infeksi berkembang ke tahap lanjut. Akibatnya, proses pengobatan menjadi lebih kompleks, dan risiko penularan di lingkungan sekitar meningkat signifikan.

Salah satu permasalahan utama dalam pengendalian TBC adalah keterlambatan diagnosis yang disebabkan oleh gejala yang bersifat tidak spesifik pada tahap awal infeksi [1]. Gejala awal yang sering muncul antara lain batuk berkepanjangan, demam ringan, kelelahan, dan penurunan berat badan yang kerap kali diabaikan oleh pasien karena dianggap sebagai kondisi kesehatan ringan. Akibatnya, banyak individu baru mencari pengobatan setelah penyakit mencapai tahap lanjut, sehingga

menyulitkan proses pengobatan dan meningkatkan risiko penyebaran di lingkungan sekitar. Banyak pasien baru menyadari kondisi mereka setelah penyakit berkembang ke tahap lanjut, yang meningkatkan risiko komplikasi serta penyebaran lebih luas di lingkungan sekitarnya. Oleh karena itu, pengembangan sistem skrining yang efektif berbasis gejala klinis dan hasil pemeriksaan menjadi kebutuhan mendesak dalam mendukung upaya deteksi dini.

Inisiatif Lampung Sehat (ILS) merupakan lembaga berbasis masyarakat yang aktif dalam mendukung program pengendalian TBC di 15 kabupaten/kota di Provinsi Lampung. ILS bekerja sama dengan Dinas Kesehatan Provinsi Lampung dalam upaya investigasi kontak erat, edukasi masyarakat, dan pelatihan kader Kesehatan [2]. Selama ini, proses skrining TBC yang dilakukan oleh kader di bawah koordinasi ILS masih bersifat manual, mulai dari pengisian form, verifikasi data, hingga rekapitulasi hasil. Metode ini memiliki sejumlah tantangan, seperti memakan waktu lama, rentan terhadap kesalahan pencatatan, serta menyulitkan analisis capaian per wilayah secara *real-time*.

Selain itu, berdasarkan kebijakan terbaru Kementerian Kesehatan, individu dengan riwayat kontak erat (serumah atau kontak ≥ 8 jam/hari) dengan pasien TBC wajib dirujuk untuk pemeriksaan, terlepas dari gejala yang dirasakan. Proses identifikasi kontak serumah dan gejala penyerta secara manual di lapangan sering kali menimbulkan keterlambatan rujukan dan kesulitan dalam validasi data. Oleh karena itu dibutuhkan sistem skrining yang mampu mengotomatisasi pengumpulan data gejala, melakukan prediksi risiko, dan memberikan rekomendasi tindak lanjut secara cepat dan akurat. Penelitian ini bertujuan untuk mengimplementasikan sistem skrining TBC berbasis web yang berfungsi sebagai alat prediksi awal infeksi. Sistem ini akan dikembangkan menggunakan *framework* Flask pada sisi *backend* dan Bootstrap dengan Jinja2 pada sisi *frontend* untuk antarmuka yang responsif dan *user-friendly*. Pemilihan teknologi ini didasarkan pada fleksibilitas, kemudahan dalam pengembangan, serta dukungan komunitas yang luas dibandingkan dengan alternatif lain seperti Django atau React yang memiliki kompleksitas lebih tinggi.

Sebagai solusi terhadap permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan sistem skrining TBC berbasis web yang dapat digunakan secara mandiri oleh masyarakat maupun kader kesehatan. Sistem ini mengintegrasikan form gejala dengan model prediksi berbasis machine learning, sehingga pengguna dapat langsung memperoleh hasil skrining dalam bentuk probabilitas risiko infeksi. Jika hasil menunjukkan risiko tinggi ($\geq 50\%$) atau terdapat kontak erat, sistem akan memberikan rekomendasi fasilitas kesehatan terdekat untuk pemeriksaan lanjutan.

Sistem ini dikembangkan menggunakan framework Flask untuk *backend*, serta Bootstrap dan Jinja2 untuk *frontend*, guna menghasilkan tampilan yang responsif dan mudah digunakan oleh masyarakat, terutama dewasa dengan tingkat literasi digital bervariasi. Pemilihan teknologi ini didasarkan pada kemudahan pengembangan, fleksibilitas, serta dukungan komunitas yang luas. Sistem ini juga dirancang agar ringan dan dapat diakses melalui perangkat mobile, dengan proses deployment pada platform hosting berbasis Python. Agar hasil dari penelitian sesuai dengan yang diharapkan, maka dilakukan Evaluasi Fungsional dan *System Usability Scaling* (SUS). *System Usability Scale* adalah alat ukur yang digunakan untuk menilai kegunaan (*usability*) dari suatu sistem, seperti perangkat lunak, aplikasi, atau situs web. SUS dikembangkan sebagai metode cepat dan andal untuk mengevaluasi kegunaan sistem berbasis pengalaman pengguna. SUS terdiri dari 10 pernyataan yang dijawab menggunakan skala Likert 5 poin (dari "Sangat Tidak Setuju" hingga "Sangat Setuju"). Skor SUS memberikan gambaran kuantitatif tentang pengalaman pengguna di mana nilai di atas 68 dianggap sebagai rata-rata, dan semakin tinggi skornya semakin baik kegunaannya [3]. Sedangkan Evaluasi Fungsional adalah evaluasi yang dilakukan untuk mengukur fungsi dari aplikasi *website* yang digunakan sudah sesuai dengan yang diharapkan dan tidak ada fungsi yang kurang.

Diharapkan dengan hadirnya sistem ini, kesadaran masyarakat terhadap pentingnya deteksi dini TBC dapat meningkat, serta dapat menjadi alat bantu yang efektif bagi tenaga kesehatan dalam melakukan skrining awal dengan lebih cepat dan akurat.

1.2 Rumusan Masalah

Adapun Rumusan Masalah sebagai berikut.

1. Bagaimana merancang dan mengimplementasikan sistem skrining TBC berbasis web yang dapat digunakan secara mandiri oleh masyarakat untuk mendeteksi gejala awal TBC?
2. Bagaimana merancang dan mengimplementasikan sistem skrining TBC berbasis web menggunakan model *machine learning* untuk meningkatkan efisiensi waktu dalam proses identifikasi gejala dan evaluasi risiko TBC?
3. Seberapa besar pengaruh sistem skrining TBC berbasis web dalam mengefisienkan proses skrining dibandingkan metode manual dari sisi waktu dan potensi penghematan biaya?

1.3 Tujuan Penelitian

Adapun tujuan pada penelitian ini adalah sebagai berikut.

1. Untuk merancang dan mengimplementasikan sistem skrining TBC berbasis web yang sederhana dan mudah digunakan, sehingga dapat membantu proses deteksi dini secara efisien.
2. Mengintegrasikan model machine learning untuk memproses data gejala dan kontak pengguna, serta menghasilkan output berupa probabilitas risiko infeksi TBC dan meningkatkan efisiensi waktu dalam proses skrining TBC.
3. Mengimplementasi model klasifikasi skrining TB untuk prediksi awal penyakit TBC dan untuk mengintegrasikan algoritma skrining TBC ke dalam aplikasi web serta memastikan proses input, pengolahan, dan output data di backend dapat berjalan dengan akurat dan optimal.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Sistem hanya akan menggunakan gejala klinis umum yang sering muncul pada pasien TBC, seperti batuk berkepanjangan, demam, dan keringat malam.
2. Data yang digunakan terbatas pada hasil pemeriksaan dasar seperti riwayat medis pasien, tanpa melibatkan data pencitraan medis seperti rontgen atau CT scan.
3. Aplikasi ini hanya akan diakses melalui antarmuka web dan tidak mencakup pengembangan aplikasi berbasis mobile atau desktop.
4. Sistem ini hanya akan berfungsi sebagai alat bantu untuk skrining awal berdasarkan input gejala pasien dan tidak menggantikan peran dokter dalam diagnosis akhir.
5. Hosting sistem dilakukan menggunakan layanan pihak ketiga berbasis Python dan pengelolaan jangka panjang disesuaikan dengan kebijakan ILS.

1.5 Sistematika Penulisan Skripsi

Adapun sistematika Penulisan Skripsi yang terdiri dari 5 (lima) bab yang dijabarkan sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi uraian mengenai latar belakang masalah yang melandasi penelitian, rumusan masalah yang ingin dijawab, serta tujuan dari dilakukannya penelitian. Selain itu, bab ini juga mencakup batasan masalah untuk memperjelas ruang lingkup penelitian agar lebih fokus, serta manfaat dari hasil penelitian baik secara teoritis maupun praktis. Dengan adanya bab pendahuluan, pembaca diharapkan memperoleh gambaran awal mengenai konteks, urgensi, dan arah dari penelitian yang dilakukan.

BAB II TINJAUAN PUSTAKA

Memuat dasar teori mengenai Flask, Python, Bootstrap, HTML, TBC, PostgreSQL, *Wireframe*, UML, SUS, *Waterfall*, ERD, XGBoost, dan Penelitian Terkait.

BAB III METODOLOGI PENELITIAN

Memuat waktu dan tempat penelitian dan tahapan penelitian secara umum identifikasi kebutuhan pengguna, pengembangan scenario, perancangan prototipe, dan melakukan penilaian berdasarkan masukan dari pengguna.

BAB IV PEMBAHASAN

Memuat hasil dari perancangan Aplikasi Web Skrining TBC Berbasis Flask: Implementasi Algoritma Prediksi.

BAB V KESIMPULAN DAN SARAN

Memuat kesimpulan dari kegiatan yang dilakukan pada pengembangan Aplikasi Web Skrining TBC Berbasis Flask: Implementasi Algoritma Prediksi. Saran-saran mengenai hasil evaluasi nilai akurasi untuk melakukam perbaikan dan pengembangan lebih lanjut.

II. TINJAUAN PUSTAKA

2.1 Bootstrap

Bootstrap adalah framework *frontend* yang bersifat *open source* dan dirancang untuk memudahkan pengembangan antarmuka web yang responsif dan *mobile friendly*. Bootstrap pertama kali dikembangkan oleh Mark Otto dan Jacob Thornton di Twitter pada tahun 2011 dan pada Juni 2014 menjadi proyek No.1 di *Github*. *Framework* ini menyediakan berbagai komponen seperti *grid system*, tombol, form, dan komponen lainnya yang memudahkan proses desain antarmuka tanpa perlu menulis kode CSS dari awal [4].

Dengan menggunakan Bootstrap sebagai framework dapat membuat website menjadi responsif. Menurut Ethan Marcotte dalam bukunya *Responsive Web Design*, desain web responsif adalah pendekatan dalam pembangunan website yang memungkinkan tata letak dan elemen-elemen di dalamnya beradaptasi secara dinamis dengan berbagai ukuran layar dan perangkat. Pendekatan ini bertujuan untuk memberikan pengalaman pengguna yang optimal di berbagai resolusi, mulai dari layar ponsel hingga monitor desktop [5].

2.2 HTML

HyperText Markup Language (HTML) adalah bahasa markup standar yang digunakan untuk membuat dan merancang halaman web. HTML berfungsi sebagai fondasi utama dalam pengembangan web, memungkinkan pengembang untuk menyusun dan menyajikan konten dengan berbagai elemen seperti teks, gambar, tautan, dan multimedia. HTML pertama kali dikembangkan oleh Tim Berners-Lee

pada tahun 1991 dan sejak saat itu telah mengalami berbagai pembaruan untuk meningkatkan fungsionalitas dan kompatibilitas dengan teknologi web modern.

Sejak pertama kali diperkenalkan, HTML telah mengalami berbagai evolusi yang signifikan. Versi awal HTML hanya mendukung elemen-elemen dasar untuk menyusun teks dan gambar. Namun, dengan berkembangnya teknologi web, HTML mengalami pembaruan yang lebih kompleks, seperti:

HTML 2.0 (1995): Versi pertama yang didokumentasikan secara luas dan mendukung form serta tabel sederhana.

HTML 3.0 (1998): Dave Ragget memperkenalkan konsep atau rancangan baru mengenai HTML. Versi ini mencakup fitur-fitur yang lebih baik, memberikan kemampuan yang lebih baik dari versi sebelumnya. Akan tetapi, fitur-fitur canggih dari HTML baru menyebabkan browser menjadi lebih lambat dalam menerapkan pengembangan lebih lanjut.

HTML 4.01 (1999): Menyediakan elemen yang lebih kompleks seperti script dan style serta peningkatan pada tabel dan form.

XHTML (2000): Versi yang lebih ketat dengan sintaks XML yang lebih terstruktur.

HTML5 (2014): Versi terbaru yang menambahkan fitur seperti elemen multimedia (audio dan video), elemen semantik, serta API untuk aplikasi web yang lebih interaktif [6].

2.3 Flask

Flask adalah framework web berbasis Python yang ringan dan fleksibel, digunakan untuk membangun aplikasi web dengan cepat dan efisien. Flask dikembangkan oleh Armin Ronacher dan pertama kali dirilis pada tahun 2010. Framework ini mengikuti pola desain mikro, yang berarti hanya menyediakan komponen inti yang diperlukan dan memungkinkan pengembang untuk menambahkan ekstensi sesuai kebutuhan. Flask memiliki 3 dependensi utama yaitu, *routing*, *debugging*, dan *Web*

Server Gateway Interface (WSGI) subsistem berasal dari *Werkzeug*, dukungan template didukung oleh *Jinja2*, dan integrasi *command-line* berasal dari *Click* [7].

Beberapa fitur utama Flask meliputi :

- a. Ringan dan Fleksibel: Flask tidak memiliki dependensi yang berat, memungkinkan pengembang untuk menggunakan hanya komponen yang dibutuhkan.
- b. Routing yang Mudah: Flask menyediakan sistem routing yang sederhana untuk menangani berbagai request.
- c. Template Engine Jinja2: Memungkinkan pengembang untuk membuat tampilan halaman web yang dinamis.
- d. Dukungan untuk RESTful API: Flask mendukung pembuatan API yang mudah diintegrasikan dengan aplikasi lain.
- e. Ekstensi yang Beragam: Tersedia berbagai ekstensi seperti Flask-SQLAlchemy untuk ORM dan Flask-WTF untuk validasi form.

2.4 Python

Python adalah bahasa pemrograman tingkat tinggi yang pertama kali diperkenalkan oleh Guido van Rossum pada tahun 1991. Bahasa ini dirancang dengan tujuan untuk mempermudah proses pengembangan perangkat lunak melalui sintaksis yang bersih dan mudah dibaca. Python mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, prosedural, dan fungsional, sehingga menjadikannya pilihan yang fleksibel dalam berbagai skenario pengembangan perangkat [8]. Python memiliki fitur dynamic typing, yang memungkinkan pengembang untuk mendeklarasikan variabel tanpa menentukan tipe data secara eksplisit, sehingga membuat pengembangan aplikasi menjadi lebih fleksibel dan cepat. Selain itu, fitur garbage collection pada Python secara otomatis mengelola alokasi dan dealokasi memori, yang membantu menghindari kebocoran memori dalam aplikasi. Python juga mendukung integrasi dengan berbagai bahasa pemrograman lain seperti C, C++, dan Java, yang memungkinkan pengembang

untuk memperluas fungsionalitas aplikasi dengan menggunakan pustaka yang tersedia dalam bahasa lain. Dalam pengembangan web, fleksibilitas ini sangat berguna dalam membangun aplikasi yang kompleks dan skalabel, misalnya dengan menggabungkan backend berbasis Python dengan frontend berbasis JavaScript untuk pengalaman pengguna yang lebih interaktif.

Python terus mengalami peningkatan popularitas dan menjadi salah satu bahasa pemrograman yang paling banyak digunakan di dunia. Menurut beberapa survey yang dilakukan oleh StackOverflow dan Github, Python berada di peringkat ke-5 pada StackOverflow dalam jumlah komunitas dan ke-4 dalam bahasa pemrograman paling banyak digunakan pada Github [9]. Python banyak digunakan dalam pengembangan aplikasi web dengan bantuan framework seperti Flask dan Django. Flask cocok untuk proyek yang membutuhkan fleksibilitas dan kesederhanaan, sementara Django menawarkan solusi yang lebih terstruktur dan kaya akan fitur untuk aplikasi berskala besar. Selain itu, Python juga mendukung framework lain seperti FastAPI untuk pengembangan aplikasi berbasis API yang lebih cepat dan asinkron. Python juga digunakan dalam pengembangan web scrapping menggunakan pustaka seperti BeautifulSoup dan Scrapy, serta otomatisasi pengujian aplikasi web dengan Selenium.

2.5 Tuberculosis (TBC)

Tuberkulosis (TBC) adalah penyakit menular yang disebabkan oleh bakteri *Mycobacterium tuberculosis*. Bakteri ini biasanya menyerang paru-paru, tetapi juga dapat menyebar ke organ lain seperti tulang, otak, ginjal, dan sistem limfatik. Penyebaran TBC terjadi melalui udara ketika seseorang yang terinfeksi batuk, bersin, atau bahkan berbicara, melepaskan droplet yang mengandung bakteri ke lingkungan sekitar [10].

Menurut laporan WHO tahun 2024 terdapat peningkatan orang yang terjangkit penyakit TB tapi mulai stabil dikarenakan pandemi COVID-19. Pada tahun 2023, diperkirakan 10.8jt jiwa terkena penyakit TB, sedikit peningkatan dibandingkan

pada tahun 2022 yaitu 10.7jt jiwa, dan lebih tinggi dari tahun 2021 yaitu 10,4jt jiwa. Secara global indikasi TB meningkat sebesar 0.2% dari tahun 2022. Kematian disebabkan oleh TB mengalami penurunan pada tahun 2023 yaitu sebesar 1,25jt kematian, sedangkan pada tahun 2022 sebesar 1,32jt jiwa. Negara yang memiliki kasus terbanyak adalah India (26%), Indonesia (10%), China (6.8%), Philippines (6.8%), dan Pakistan (6.3%) [11].

Gejala umum Tuberkulosis meliputi :

- a. Berat badan turun selama tiga bulan berturut-turut tanpa sebab yang jelas.
- b. Demam meriang lebih dari sebulan.
- c. Sesak napas.
- d. Dada terasa nyeri.
- e. Nafsu makan berkurang.
- f. Mudah lesu.
- g. Berkeringat malam walaupun tanpa aktifitas fisik
- h. Dahak bercampur darah.

Pengobatan tuberkulosis dilakukan dalam dua tahap, yaitu fase awal yang berlangsung selama 2 hingga 3 bulan dan fase lanjutan dengan durasi 4 hingga 7 bulan. Kepatuhan dalam menjalani pengobatan sesuai dengan anjuran dokter merupakan hal yang sangat penting untuk mencegah resistensi bakteri terhadap obat. Kombinasi obat yang digunakan dalam pengobatan ini terdiri dari obat utama serta obat tambahan untuk meningkatkan efektivitas penyembuhan [10]. Obat lini pertama untuk tuberkulosis terdiri dari Isoniazid (INH), Rifampisin, Pirazinamid, Streptomisin, dan Etambutol. Sementara itu, obat tambahan yang dapat digunakan meliputi Kanamisin, Amikasin, dan Kuinolon [12].

Pencegahan atau penanggulangan TB yang dapat dilakukan yaitu :

- a. Pengobatan Pencegahan TBC (TPT – TB Preventive Treatment).
- b. Skrining dan Diagnosis Dini.
- c. Deteksi Infeksi Dini.
- d. Pendekatan Sosial dan Akses Layanan Kesehatan.

- e. Kolaborasi dan Penelitian Inovatif [12].

Pendekatan penanggulangan harus diterapkan secara bersamaan untuk mengurangi beban TBC secara global dan mencapai target eliminasi yang telah ditetapkan oleh WHO.

2.6 PostgreSQL

PostgreSQL adalah manajemen basis data relasional (RDBMS) *open-source* yang telah dikembangkan sejak tahun 1986 di University of California, Berkeley. PostgreSQL memiliki reputasi yang sangat baik dikarenakan memiliki banyak fitur, seperti dukungan terhadap transaksi ACID (*Atomicity, Consistency, Isolation, Durability*), fleksibilitas tinggi dalam pengembangan, serta kepatuhannya terhadap standar SQL yang ketat. PostgreSQL banyak digunakan oleh organisasi dari berbagai skala karena kemampuannya dalam menangani data kompleks serta skalabilitas yang baik. Dengan lisensi *open-source*, PostgreSQL memungkinkan pengguna untuk mengadaptasi, memodifikasi, dan mendistribusikan sistem sesuai dengan kebutuhan spesifik mereka.

PostgreSQL mendukung berbagai bahasa pemrograman dalam pengembangan aplikasi, seperti Python, Java, PHP, C/C++, dan PL/pgSQL. Dukungan terhadap berbagai bahasa pemrograman ini menjadikan PostgreSQL pilihan yang fleksibel dalam pengembangan aplikasi berbasis web dan desktop. PostgreSQL merupakan sistem basis data yang andal, fleksibel, dan dapat diskalakan. Dengan fitur seperti dukungan terhadap SQL, ekstensibilitas tinggi, serta keamanan yang handal, PostgreSQL menjadi pilihan tepat dalam pengelolaan data yang kompleks. Fleksibilitas dan kemampuan integrasi yang luas memungkinkan PostgreSQL untuk digunakan dalam berbagai sistem aplikasi, baik skala kecil maupun besar[13].

2.7 System Usability Scale (SUS)

System Usability Scale (SUS) adalah alat ukur yang digunakan untuk menilai kegunaan (*usability*) dari suatu sistem, seperti perangkat lunak, aplikasi, atau situs web.

SUS dikembangkan oleh John Brooke pada tahun 1986 sebagai metode cepat dan andal untuk mengevaluasi kegunaan sistem berbasis pengalaman pengguna. SUS terdiri dari 10 pernyataan yang dijawab menggunakan skala Likert 5 poin (dari “Sangat Tidak Setuju” hingga “Sangat Setuju”). Skor SUS memberikan nilai kuantitatif tentang pengalaman pengguna, di mana nilai di atas 68 dianggap sebagai rata-rata, dan semakin tinggi skornya, semakin baik kegunaannya [3].

Keunggulan SUS yaitu mudah diterapkan, dapat digunakan pada berbagai jenis sistem, dan dapat dibandingkan dengan sistem lain. Sedangkan untuk kelemahan SUS sistem tidak memberikan analisis yang mendalam, tidak menggali lebih dalam aspek pengguna, dan hasilnya terkadang bervariasi berdasarkan persepsi pengguna. Untuk penggunaan SUS dalam pengukuran kualitas sistem telah digunakan secara luas dalam berbagai bidang contohnya seperti, pengujian aplikasi dan perangkat lunak, sistem informasi dan e-commerce, evaluasi perangkat medis, dan pengujian situs web.

2.9 Black Box Testing

Black Box Testing adalah salah satu metode pengujian perangkat lunak yang fokusnya ada di fungsi sistem dan bukan cara kerja sistem di dalam. Jadi dalam metode ini penguji hanya melihat apa yang masuk (input) dan apa hasil keluaran (output) tanpa melihat bagaimana sistem itu memproses data di belakang layar[14].

Black Box Testing disebut juga sebagai pengujian berbasis fungsional, karena penguji mengacu pada spesifikasi atau kebutuhan sistem untuk memastikan semua fitur berjalan sesuai harapan. Misalnya, saat pengguna isi form gejala TBC, sistem

harus dapat memberikan hasil prediksi risiko dengan akurat. Yang diuji: apakah hasilnya sesuai, bukan bagaimana sistem itu menghitungnya.

Terdapat beberapa Teknik *Black Box Testing* sebagai berikut [15].

1. Equivalence Partitioning

Teknik ini membagi data input ke dalam beberapa kelompok (partisi), di mana setiap partisi dianggap memberikan hasil yang serupa. Tujuannya adalah untuk mengurangi jumlah test case, karena cukup diambil satu contoh dari tiap partisi untuk diuji.

2. Boundary Value Analysis (BVA)

Teknik ini fokus menguji nilai batas input, seperti nilai minimum, maksimum, tepat di dalam batas, dan tepat di luar batas. Ini karena error sering muncul di area batas, sehingga penting dilakukan pengujian di titik-titik ekstrem tersebut.

3. Fuzzing

Fuzz testing dilakukan dengan cara memberikan input data yang tidak valid, acak, atau setengah valid ke sistem, untuk melihat apakah sistem bisa menangani data tersebut tanpa crash atau error. Biasanya dilakukan secara otomatis atau semi-otomatis.

4. Cause-Effect Graphing

Teknik ini membuat grafik hubungan antara sebab (input/kondisi) dan akibat (output/hasil). Grafik ini digunakan untuk merancang test case berdasarkan logika hubungan antar variabel, seperti menggunakan simbol AND, OR, dan NOT.

5. Orthogonal Array Testing (OAT)

Digunakan ketika domain input cukup besar tapi tidak mungkin diuji semuanya. OAT memungkinkan pengujian secara efisien dengan kombinasi input tertentu yang diatur dalam bentuk array.

6. All Pair Testing

Teknik ini menyusun test case agar semua kemungkinan pasangan kombinasi input dapat diuji. Cocok untuk pengujian dengan banyak parameter input, namun ingin meminimalkan jumlah test case.

7. State Transition Testing

Cocok untuk sistem yang memiliki berbagai status (state), seperti halaman login, halaman utama, form input, dan hasil. Pengujian dilakukan untuk mengecek apakah perpindahan antar status berjalan sesuai alur.

2.10 Waterfall

Metode Waterfall merupakan salah satu pendekatan klasik dalam pengembangan perangkat lunak yang memiliki alur kerja berurutan dan sistematis. Setiap tahap dalam metode ini harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Model ini dianggap cocok untuk pengembangan sistem dengan kebutuhan yang sudah terdefinisi dengan jelas sejak awal, sehingga dapat meminimalkan perubahan selama proses implementasi.

Metode Waterfall terdiri dari lima tahapan utama, yaitu:

1. Analisis Kebutuhan, yaitu proses pengumpulan dan identifikasi kebutuhan sistem secara menyeluruh untuk memastikan semua fungsionalitas sistem sesuai dengan kebutuhan pengguna.
2. Perancangan Sistem, meliputi perancangan struktur sistem secara teknis, termasuk diagram UML, ERD (Entity Relationship Diagram), dan rancangan antarmuka.
3. Implementasi (Coding), yaitu tahapan penerjemahan desain sistem ke dalam bentuk kode program.
4. Pengujian (Testing), dilakukan untuk memastikan sistem berjalan sesuai dengan kebutuhan dan bebas dari kesalahan atau bug.
5. Deployment, dilakukan setelah pengujian berhasil dilakukan dengan kondisi sistem sesuai dengan kebutuhan.
6. Pemeliharaan (Maintenance), dilakukan setelah sistem digunakan untuk memperbaiki kesalahan atau menambahkan fitur baru sesuai kebutuhan pengguna.

Penerapan metode ini dalam penelitian mereka digunakan untuk mengembangkan sistem informasi penggajian berbasis web, dan dinilai mampu menghasilkan sistem yang terstruktur, efisien, serta mudah diimplementasikan [16], [17].

2.11 Unified Modeling Language (UML)

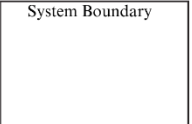
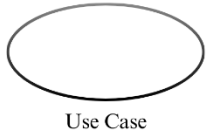
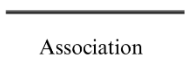
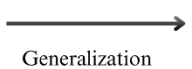
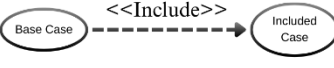
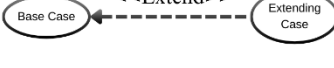
Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan secara luas dalam pengembangan perangkat lunak untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan komponen-komponen dari suatu sistem. UML menyediakan berbagai jenis diagram yang digunakan untuk merepresentasikan aspek struktural maupun perilaku dari sistem perangkat lunak.

UML bertujuan untuk menyatukan berbagai metode pemodelan perangkat lunak ke dalam satu bahasa standar yang dapat digunakan dalam analisis dan perancangan sistem berbasis objek [18].

Terdapat 4 jenis diagram, yaitu sebagai berikut.

a. Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan fungsionalitas sistem dari sudut pandang pengguna (aktor). Diagram ini menjelaskan interaksi antara pengguna dengan sistem melalui serangkaian use case, dan sangat berguna dalam tahap awal perancangan sistem untuk mendefinisikan ruang lingkup fungsionalitas.

SIMBOL	KETERANGAN
 Actor	Merepresentasikan pengguna atau sistem eksternal yang berinteraksi dengan sistem. Bisa manusia, aplikasi lain, atau perangkat.
 System Boundary	Menunjukkan batas sistem, mana yang termasuk dalam sistem dan mana yang eksternal.
 Use Case	Menunjukkan fungsi atau layanan yang disediakan sistem untuk aktor. Misalnya "Login", "Melakukan Skrining".
 Association	Menggambarkan interaksi atau komunikasi antara aktor dengan fungsi sistem.
 Generalization	Menunjukkan aktor atau use case yang lebih spesifik merupakan turunan dari yang lebih umum. Contoh: aktor "Admin" dan "User" bisa merupakan generalisasi dari "Pengguna".
	Menggambarkan suatu use case selalu menyertakan use case lain (wajib).
	Menunjukkan suatu use case dapat menambahkan perilaku tambahan yang memiliki sifat opsional.

Gambar 2.1 Notasi Use Case Diagram

b. Activity Diagram

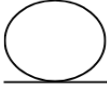
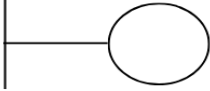
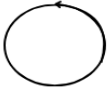
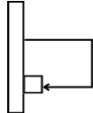
Activity Diagram merepresentasikan alur proses atau aktivitas dalam sistem, baik yang dilakukan oleh pengguna maupun sistem itu sendiri. Diagram ini sering digunakan untuk menggambarkan proses bisnis, logika alur kerja, atau prosedur eksekusi dalam aplikasi.

SIMBOL	KETERANGAN
 Initial State	Menandakan awal proses atau aktivitas dimulai. Activity / Action
 Final State	Menandakan akhir dari proses/aktivitas.
 Activity/Action	Merepresentasikan suatu aktivitas, aksi, atau langkah yang dilakukan dalam proses.
 Decision	Digunakan untuk pengambilan keputusan dengan percabangan (ya/tidak, benar/salah).
 Merge	Digunakan untuk menggabungkan kembali alur dari beberapa cabang keputusan menjadi satu alur.
 Fork Node	Memecah satu alur menjadi beberapa aktivitas paralel.
 Join Node	Menggabungkan beberapa alur paralel menjadi satu alur.
 Transition	Menunjukkan urutan aliran proses dari satu aktivitas ke aktivitas lain.
 Swimlane	Digunakan untuk memisahkan aktivitas berdasarkan pelaku (aktor atau bagian sistem).

Gambar 2.2 Notasi Activity Diagram.

c. Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek dalam sistem dalam urutan waktu tertentu. Diagram ini menekankan pengiriman pesan antar entitas (biasanya objek atau modul) dan sangat berguna dalam menggambarkan implementasi logika sistem secara detail.

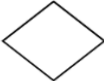
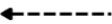
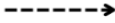
Simbol	Keterangan
 Entity Class	Entity Class : Gambaran sistem sebagai dasar dalam menyusun basis data
 Boundary Class	Boundary Class : Menangani komunikasi antar lingkungan sistem
 Control Class	Control Class : Bertanggung jawab terhadap kelas-kelas objek yang berisik logika
 Recursive	Recursive : Menggambarkan pesan ke dirinya sendiri
 Activation	Activation : Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
 Life Line	Life line : Komponen yang digambarkan dengan garis putus sepanjang <i>Life Line</i> terdapat <i>activation</i> .

Gambar 2.3 Notasi Sequence Diagram

d. Class Diagram

Class Diagram merupakan salah satu jenis diagram struktural dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan struktur statis dari sistem perangkat lunak. Diagram ini merepresentasikan kelas-kelas yang ada di

dalam sistem, termasuk atribut, metode, serta hubungan antar kelas seperti asosiasi, pewarisan (inheritance), agregasi, dan komposisi.

Simbol	Keterangan
 Generalization	Generalization : Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan strukdata dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
 N-ary Association	N-ary Association : Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
 Class	Class : Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
 Collaboration	Collaboration : Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
 Realization	Realization : Operasi yang benar-benar dilakukan oleh suatu objek.
 Dependency	Dependency : Hubungan yang dimana perubahan terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
 Association	Association : Garis yang menghubungkan antara objek satu dengan objek lainnya.

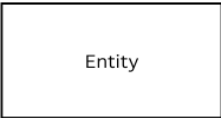
Gambar 2.4 Notasi Class Diagram

2.12 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah representasi grafis dari struktur konseptual basis data yang digunakan untuk memodelkan hubungan antar data dalam suatu sistem [19]. ERD menjadi alat penting dalam tahap perancangan basis data karena mampu menggambarkan bagaimana entitas saling berinteraksi melalui atribut dan relasi. ERD adalah model yang dirancang untuk memvisualisasikan struktur logis dari suatu sistem penyimpanan data, yang terdiri dari tiga komponen utama: entitas, atribut, dan relasi antar entitas[20].

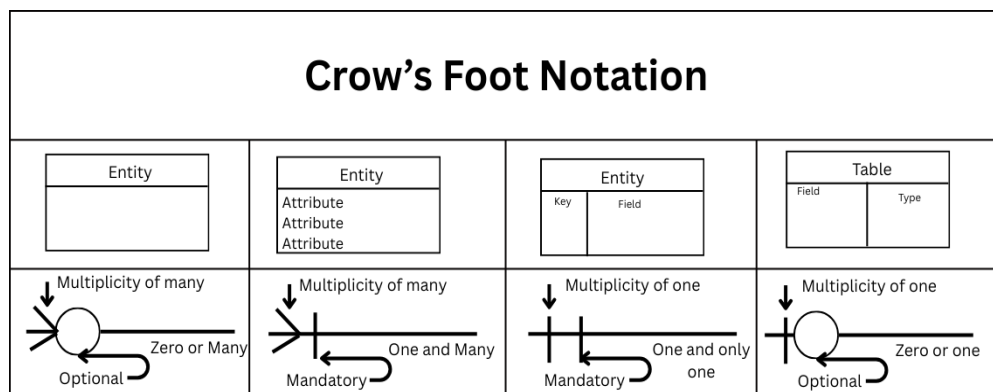
Terdapat beberapa notasi ERD yang umum digunakan:

1. Notasi Chen: Menggunakan simbol grafis seperti kotak (entitas), oval (atribut), dan diamond (relasi), cocok untuk pemodelan detail tetapi cenderung kompleks

Chen ERD Notation		
		
		 Mandatory Relationship
		 Optional Relationship
		 Partial Participation  Total Participation

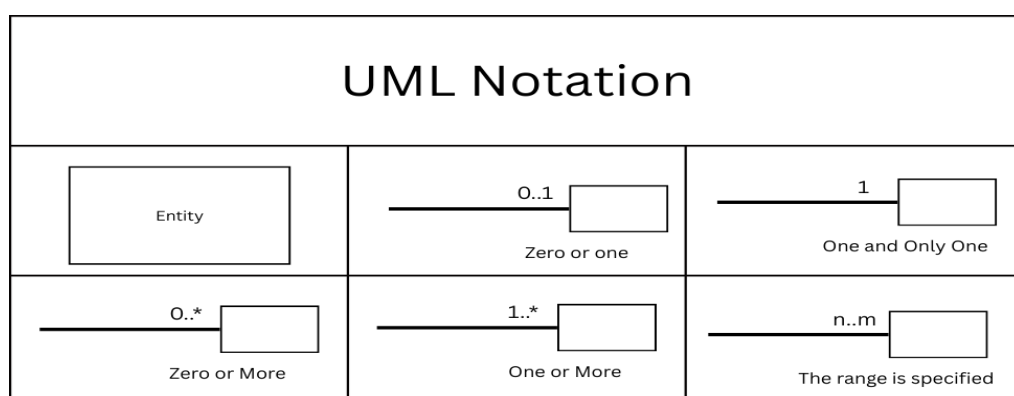
Gambar 2.5 Notasi Chen.

2. Crow's Foot: Fokus pada kardinalitas relasi (contoh: one-to-many) dengan simbol intuitif, sering dipakai dalam tools desain database seperti MySQL Workbench.



Gambar 2.6 Notasi Crow's Foot.

3. UML Class Diagram: Memadukan konsep ERD dengan pemodelan berorientasi objek, ideal untuk integrasi dengan pengembangan perangkat lunak.



Gambar 2.7 Notasi UML.

Pemilihan notasi bergantung pada kompleksitas sistem dan kebutuhan pemangku kepentingan. Notasi Crow's Foot lebih populer untuk proyek sederhana, sementara UML digunakan untuk sistem berbasis objek atau hierarki[19].

2.13 Postman

Postman adalah platform yang digunakan untuk pengujian, pengembangan, dan dokumentasi *Application Programming Interface* (API). Postman memungkinkan pengguna untuk mengirim *request* ke server dan melihat *response* secara langsung seperti kode status, header, dan hasil dari *response* tanpa perlu menuliskan perintah secara manual melalui *command line*[21].

Postman digunakan pada penelitian ini sebagai alat bantu utama untuk pengujian API karena mudah digunakan, lingkungan terstruktur, dan mendukung proses verifikasi fungsionalitas tiap *endpoint*.

2.14 OWASP ZAP

Zed Attack Proxy (ZAP) adalah salah satu alat *penetration testing* dan *vulnerability scanning* yang dikembangkan oleh *Open Web Security Project* (OWASP). Zap digunakan untuk mendeteksi berbagai kerentanan pada aplikasi web, terutama pada tahap pengembangan ataupun pengujian keamanan [22].

ZAP bekerja dengan cara menganalisis lalu lintas HTTP/HTTPS antara klien dan server, kemudian melakukan serangkaian pengujian otomatis ataupun manual untuk menemukan potensi celah keamanan. Fitur utama ZAP yaitu *spidering*, *active scanning*, *passive scanning*, *fuzzer*, dan *monitoring* dan analisis permintaan secara realtime.

Dalam penelitian ini, ZAP digunakan untuk pendeteksian kerentanan aplikasi web secara otomatis. Pengujian dilakukan setelah *dideploy* ke lingkungan produksi agar evaluasi keamanan mencerminkan kondisi nyata ketika aplikasi sudah aktif digunakan oleh pengguna.

2.15 Xgboost

XGBoost (Extreme Gradient Boosting) adalah salah satu algoritma *machine learning* berbasis pohon keputusan dirancang untuk memberikan kinerja tinggi dan efisiensi dalam pemrosesan data berskala besar. Algoritma ini dikembangkan dengan berbagai optimisasi, termasuk penggunaan algoritma pencarian *split* yang efisien, pengelolaan data *sparsity*, serta sistem pembelajaran paralel dan terdistribusi. Keunggulan utama XGBoost terletak pada kemampuannya untuk menangani dataset besar dengan cepat serta menghasilkan performa prediksi yang kompetitif dalam berbagai kompetisi data science seperti *Kaggle*.

XGBoost menggunakan pendekatan gradient tree boosting yang diperkuat dengan teknik regularisasi untuk mencegah overfitting. Selain itu, algoritma ini juga mendukung pembelajaran menggunakan data sparse (bernilai nol atau kosong) secara langsung melalui mekanisme *sparsity-aware*. Dalam pengolahan data berskala besar, XGBoost mengimplementasikan sistem out-of-core learning, cache-aware computation, dan pemrosesan berbasis blok untuk meningkatkan efisiensi penggunaan memori dan disk.

Keunggulan ini menjadikan XGBoost sebagai algoritma andalan dalam berbagai tugas seperti klasifikasi, prediksi risiko, dan pemeringkatan. Implementasinya juga mendukung berbagai bahasa pemrograman seperti Python, R, dan Scala, serta bisa dijalankan secara terdistribusi menggunakan Spark dan Hadoop [23].

2.16 Penelitian Terkait

Berikut adalah penelitian-penelitian terkait dalam bentuk jurnal yang berkaitan dengan tuberkulosis dan skrining.

Tabel 2.1 Penelitian Terkait

No	Judul	Penulis	Tahun	Hasil
1	Early Detection of Tuberculosis Application (E-TIBI): A New Paradigm to Detect New Case of Tuberculosis	(Triyono et.al 2023) [24]	2023	Pengguna aplikasi E-TIBI mencapai 4.658 pengguna. Dari hasil yang didapatkan terdapat 20.8% pengguna diduga menderita TBC. Kelompok berisiko tinggi yang teridentifikasi dalam skrining meliputi perokok (16.60%) dan orang dengan HIV (20.60%).
2	Implementasi Aplikasi Deteksi Dini Suspek Tuberkulosis Berbasis mHealth di Antara Kontak Serumah: Tinjauan Sistematis	(Pramono et.al 2023) [25]	2023	Penelitian menggunakan model PRISMA sebagai metode pencarian artikel di database seperti <i>Google Scholar</i> , <i>ScienceDirect</i> , <i>EBSCO</i> , dan <i>ProQuest</i> . Aplikasi mHealth terbukti efektif dalam mendeteksi dini kasus TBC di Masyarakat yang berisiko.
3	Pengembangan Aplikasi SIKRIBO (Sistem Skrining Tuberkulosis)	(Rahayu et.al 2022) [26]	2022	Hasil uji coba menunjukkan bahwa aplikasi SKIRIBO berfungsi dengan baik dan dapat digunakan untuk membantu mendeteksi TBC dini.

Tabel 2.1 (lanjutan)

No	Judul	Penulis	Tahun	Hasil
4	Capacity Building Skrining TBC Anak di Kabupaten Jember	(Damayanti et.al 2023) [27]	2024	Penelitian ini bertujuan meningkatkan kapasitas tenaga kesehatan dan kader dalam pelaksanaan skrining TBC anak secara lebih aktif. Peserta dibekali pemahaman dasar tentang TBC, penularannya, dan pentingnya deteksi dini, serta dilatih menggunakan alat skrining berbasis Google Form yang dikembangkan Universitas Airlangga agar data dapat dikumpulkan dan dianalisis secara real-time.
5	Tuberculosis Research Trends in Indonesian Health Scientific Journals: A Content Analysis	(Ratnasari and Handayani 2023) [28]	2023	Jumlah penelitian tentang TBC di jurnal SINTA-1 mengalami perkembangan yang fluktuatif dalam 10 tahun terakhir, dengan puncaknya terjadi pada 2020 setelah sempat menurun pada 2019. Topik penelitian perlu lebih bervariasi, terutama dalam aspek skrining dini, edukasi pencegahan, serta inovasi teknologi dalam deteksi dan pengobatan TBC.
6	Integrated E-Healthcare Management System Using Machine Learning and Flask	(Kumar S, Ojha J, Tripathi M, Garg K 2023) [29]	2023	Hasil pengujian menunjukkan bahwa sistem tersebut mampu meningkatkan efisiensi layanan kesehatan dengan akurasi model prediksi mencapai 95%. Pengguna dapat mengakses layanan kesehatan secara cepat tanpa harus hadir langsung ke Fasilitas Kesehatan.

III. METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Waktu penelitian Agustus 2025 - November 2025 yang telah dilakukan di Jakarta Timur dan Bandarlampung. Tahapan penelitian secara umum identifikasi kebutuhan, desain sistem, implementasi, pengujian, *deployment*, evaluasi sistem oleh pengguna, dan pemeliharaan.

Tabel 3.1 Jadwal Penelitian

No	Kegiatan	2025						2026
		Jul	Agu	Sep	Okt	Nov	Des	Jan
1	Studi Literatur							
2	Analisis Kebutuhan							
3	Merancang Desain Sistem							
4	Implementasi							
5	Pengujian							
6	Deployment dan Evaluasi							
7	Pemeliharaan							

3.2 Alat dan Bahan

3.2.1 Alat

Tabel 3.2 Alat Penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1	Komputer	Intel Core i5 9400F, 16GB RAM, Windows 10	Perangkat yang digunakan untuk pengembangan website.
2	Visual Studio Code	Versi 1.106.3	Sebagai kode editor yang digunakan pada saat pengembangan.
3	Flask	Versi 3.1.0	Sebagai <i>framework</i> utama yang menangani <i>backend</i> .
4	Jinja2	Versi 3.1.6	<i>Template engine</i> dari Flask untuk membuat halaman HTML dinamis dari sisi server.
5	Github Desktop	Versi 3.5.4	Sebagai <i>version control</i> dan dokumentasi kode.
6	PostgreSQL	Versi 10.0.9	<i>Database</i> yang digunakan untuk menyimpan data sistem.
7	pgAdmin 4	Versi 9.6	<i>Tool</i> yang digunakan untuk pengelolaan database PostgreSQL di lokal.
8	Postman	Versi 11.73.5	Sebagai alat pengujian API.
9	ZAP	Versi 2.16.1	Sebagai alat pengujian keamanan website.

Tabel 3.2 (Lanjutan)

No	Nama Alat	Spesifikasi	Deskripsi
10	Server	VPS	<i>Virtual Private Server</i> yang digunakan untuk penyediaan <i>hosting</i> website skrining TBC.
11	SMTP <i>Email Service</i>	SMTP Port:465, SSL <i>enabled</i> , SMTP Host: smtp.hostinger.com	Digunakan untuk pengiriman <i>email</i> aktivasi akun dan <i>reset password</i> .
12	Domain	.org	Domain yang digunakan untuk mengakses website secara <i>online</i> setelah dideploy.

3.2.2 Bahan

Tabel 3.3 Bahan Penelitian

No	Nama Bahan	Spesifikasi	Deskripsi
1	Kuesioner Evaluasi	SUS dan Likert	Digunakan untuk evaluasi <i>usability</i> dan kepuasan pengguna.
2	Model <i>Machine Learning</i>	XGBoost	Digunakan untuk proses prediksi hasil skrining TBC.

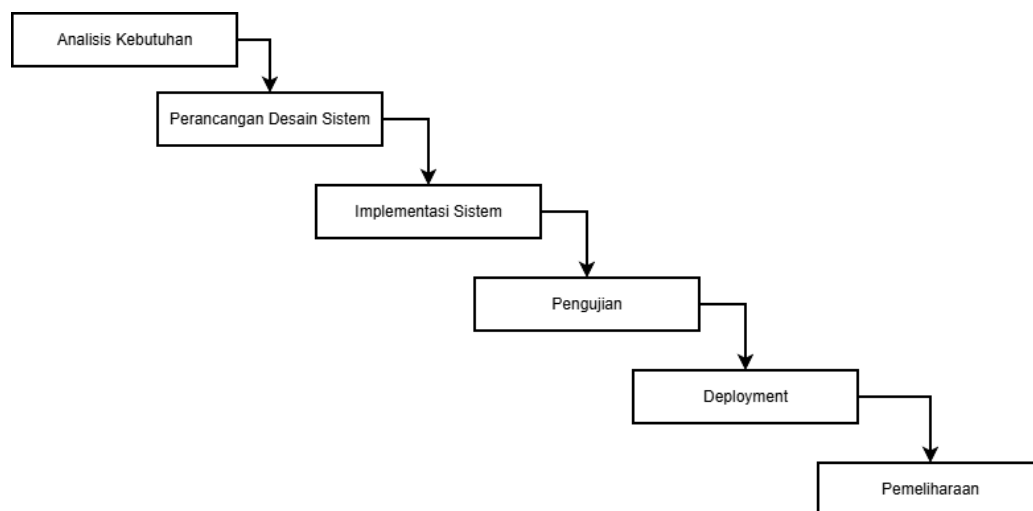
3.3 Tahapan Penelitian

Metodologi pengembangan yang digunakan adalah metodologi *Waterfall*. Model ini dipilih karena menawarkan pendekatan terstruktur dan sistematis di mana tiap tahapan dilakukan secara berurutan mulai dari analisis hingga pemeliharaan sistem.

Model *Waterfall* efektif diterapkan ketika ruang lingkup proyek bersifat stabil dan minim perubahan karena proses dokumentasi dan perencanaan dilakukan secara menyeluruh di awal pengembangan [30].

Berbeda dengan metode *Agile* yang mengedepankan pengembangan secara iterative dan adaptif terhadap perubahan. *Waterfall* lebih sesuai untuk pengembangan sistem skrining TBC ini dikarenakan fitur-fitur utama seperti form input gejala dan prediksi hasil skrining telah ditentukan di awal sebelum pengembangan dimulai. *Agile* memang lebih fleksibel dalam menanggapi perubahan [31], akan tetapi dalam proyek ini struktur tetap dan dokumentasi lengkap lebih diutamakan.

Berikut ini adalah tahap penelitian secara umum yang akan dilakukan pada penelitian ini.



Gambar 3.1 Tahapan Penelitian Menggunakan Metode Waterfall.

3.3.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan sebagai tahap awal dalam proses pengembangan sistem untuk memahami secara menyeluruh permasalahan yang ada, kebutuhan pengguna, serta ruang lingkup sistem yang akan dibangun. Proses ini bertujuan agar sistem yang dikembangkan benar-benar dapat menjawab kebutuhan lapangan dan

selaras dengan tujuan penelitian, yaitu menghasilkan aplikasi web skrining Tuberkulosis (TBC) berbasis *Machine Learning* dengan algoritma XGBoost.

Tahap analisis dilakukan melalui dua pendekatan utama, yaitu studi literatur dan pengumpulan data lapangan.

- a. Studi literatur dilakukan dengan menelaah berbagai sumber tertulis seperti buku, jurnal ilmiah, laporan penelitian, serta artikel daring yang relevan dengan tema skrining TBC, pengembangan sistem berbasis web, dan penerapan algoritma Machine Learning untuk klasifikasi penyakit. Tujuan dari studi literatur ini adalah untuk memperoleh dasar teoritis yang kuat serta memahami penelitian-penelitian terdahulu yang dapat dijadikan acuan dalam perancangan sistem.
- b. Analisis kebutuhan juga diperkuat melalui wawancara langsung dengan narasumber dari organisasi Inisiatif Lampung Sehat (ILS), yang memiliki pengalaman langsung dalam pelaksanaan program penanggulangan TBC di lapangan. Hasil wawancara ini memberikan pemahaman kontekstual mengenai kendala, kebutuhan fitur, serta ekspektasi pengguna terhadap sistem skrining yang akan dikembangkan.

Melalui hasil analisis kebutuhan tersebut, diperoleh informasi mengenai fitur-fitur utama yang harus dimiliki sistem, alur kerja yang diharapkan pengguna, serta persyaratan non-fungsional seperti keamanan, kemudahan penggunaan, dan skalabilitas. Tahapan ini menjadi dasar bagi proses desain dan implementasi yang dilakukan pada tahap berikutnya.

3.3.2 Merancang Desain Sistem

Perancangan desain sistem merupakan tahap lanjutan setelah analisis kebutuhan selesai dilakukan. Tujuan dari tahap ini adalah untuk menerjemahkan hasil analisis kebutuhan ke dalam bentuk rancangan sistem yang lebih konkret, baik dalam

bentuk diagram, struktur basis data, maupun arsitektur sistem yang akan digunakan sebagai acuan pada tahap implementasi.

Desain sistem dalam penelitian ini menggunakan pendekatan perancangan terstruktur yang meliputi beberapa komponen utama, yaitu:

1. Perancangan Arsitektur Sistem menggambarkan hubungan antara komponen utama sistem seperti pengguna, antarmuka web (*frontend*), server aplikasi (*backend* Flask), model *Machine Learning* XGBoost, dan basis data PostgreSQL.
2. Perancangan Basis Data dilakukan untuk menentukan entitas, atribut, serta relasi antar tabel yang digunakan dalam penyimpanan data pengguna, hasil skrining TBC, riwayat hasil skrining tbc, dan data kader.
3. Perancangan Antarmuka (UI/UX) terfokus pada kemudahan penggunaan bagi masyarakat umum dengan tampilan yang sederhana, informatif, dan mudah diakses melalui berbagai perangkat.
4. Perancangan Logika Sistem Skrining (*backend*) perancangan alur proses prediksi berdasarkan input gejala pengguna, data pengguna, hasil rekomendasi, penyimpanan riwayat skrining pengguna, dan *dashboard* kader (statistik, export data, import data).

Perancangan ini disusun berdasarkan prinsip *modularity* agar sistem mudah dikembangkan, serta mempertimbangkan aspek *security* dan *usability* sesuai hasil wawancara dengan pihak ILS. Hasil dari tahap perancangan ini kemudian digunakan sebagai dasar dalam proses implementasi sistem pada bab berikutnya.

3.3.3 Implementasi

Implementasi sistem dilakukan menggunakan bahasa pemrograman Python karena memiliki sintaks yang sederhana, mendukung berbagai pustaka machine learning, dan kompatibel untuk pengembangan aplikasi web. *Framework* yang digunakan

adalah Flask, yang merupakan *microframework* berbasis Python dengan arsitektur ringan namun fleksibel, sehingga sangat sesuai untuk membangun sistem berbasis web yang modular dan mudah dikembangkan.

Pada sisi tampilan (*frontend*), digunakan Bootstrap sebagai kerangka kerja CSS yang mempermudah perancangan antarmuka responsif dan adaptif terhadap berbagai ukuran layar perangkat. Kombinasi Flask dan Bootstrap memungkinkan integrasi antara *backend* dan *user interface* berjalan secara dinamis melalui *template engine* Jinja2, yang digunakan untuk menampilkan data dari server ke halaman web secara *real-time*.

Untuk penyimpanan data, sistem menggunakan PostgreSQL sebagai basis data utama. PostgreSQL dipilih karena memiliki performa yang tinggi, mendukung transaksi kompleks, serta menjamin konsistensi data. Selain itu, PostgreSQL juga mendukung tipe data terstruktur dan relasional yang sesuai untuk penyimpanan data hasil skrining, data pengguna, dan hasil prediksi dari model *machine learning*.

Dengan kombinasi teknologi tersebut, sistem skrining TBC ini diharapkan dapat berjalan secara optimal baik dari sisi performa maupun keamanan, serta memudahkan proses pengembangan dan pemeliharaan di tahap berikutnya.

3.3.4 Pengujian

Pengujian akan dilakukan dengan metode *Black Box Testing* yang berfokus pada fungsional sistem tanpa melihat logika internal atau kode program. Metode ini dipilih karena cocok untuk pengujian sistem berbasis web, khususnya dalam proses validasi *form* skrining, proses pengolahan prediksi, tampilan hasil prediksi, dsb. Tujuan dari pengujian ini adalah memastikan fitur pada sistem berjalan sesuai fungsinya, mengetahui apakah sistem mampu menghasilkan output yang benar berdasarkan input pengguna, dan mengidentifikasi adanya *bug* atau kesalahan sistem dalam pengolahan input.

3.3.5 Deployment Sistem dan Evaluasi

Tahap deployment sistem merupakan proses penerapan aplikasi ke lingkungan produksi agar dapat diakses dan digunakan oleh pengguna sesungguhnya. Pada penelitian ini, deployment dilakukan menggunakan layanan *Virtual Private Server* (VPS) yang mendukung bahasa pemrograman Python. Pemilihan server mempertimbangkan efisiensi biaya, kemudahan konfigurasi, serta kompatibilitas dengan framework Flask dan basis data PostgreSQL yang digunakan dalam pengembangan sistem.

Proses deployment meliputi beberapa tahapan, yaitu pengaturan server environment, pengunggahan kode program dan model machine learning yang telah dilatih, konfigurasi koneksi basis data, serta pengujian fungsi untuk memastikan sistem berjalan stabil di lingkungan daring. Selain itu, dilakukan pula pengujian akses dari berbagai perangkat dan browser untuk memastikan tampilan antarmuka tetap responsif dan dapat digunakan secara optimal oleh masyarakat maupun pihak Inisiatif Lampung Sehat (ILS).

Setelah sistem berhasil dijalankan di lingkungan server, dilakukan proses evaluasi untuk menilai kinerja dan kelayakan sistem dari sisi pengguna. Evaluasi ini mencakup dua pendekatan, yaitu usability testing menggunakan metode System Usability Scale (SUS) dan penilaian kelayakan menggunakan skala Likert. Kedua metode ini dipilih karena mampu memberikan gambaran kuantitatif mengenai tingkat kemudahan penggunaan (usability), tampilan antarmuka, kecepatan, serta kepuasan pengguna secara umum terhadap sistem.

Tujuan utama dari tahap ini adalah untuk memastikan bahwa seluruh fungsi sistem berjalan dengan baik pada lingkungan produksi, sistem dapat diakses secara mudah oleh pengguna, serta memperoleh umpan balik yang valid mengenai kualitas dan kelayakan aplikasi dari sudut pandang pengguna akhir.

Berikut adalah perkiraan pertanyaan yang akan diajukan pada saat pengambilan nilai SUS dari sistem yang diperkenalkan kepada responden. Terdapat 10 pertanyaan mengenai rancangan yang telah dibuat.

Tabel 3.4 Contoh Pertanyaan SUS

No.	Pertanyaan	1	2	3	4	5
1	Saya rasa saya akan menggunakan sistem ini dengan sering					
2	Sistem terlalu rumit untuk dipahami					
3	Sistem sangat mudah untuk dimengerti					
4	Saya rasa saya membutuhkan teknisi ahli untuk memahami sistem ini					
5	Fungsi yang terdapat pada sistem terhubung dengan baik					
6	Saya rasa banyak fungsi yang tidak diperlukan dalam sistem ini					
7	Saya rasa banyak orang yang akan dengan mudah memahami sistem ini					
8	Saya rasa tampilan sistem ini terlalu rumit untuk dipahami					
9	Saya merasa sangat percaya diri untuk menggunakan sistem ini					
10	Saya butuh mempelajari banyak hal untuk menggunakan sistem ini					

Setelah dilakukan pengisian kuesioner dinilai dari sangat tidak setuju hingga sangat setuju (nilai 1-5) maka data yang terkumpul akan dihitung dengan menggunakan rumus *System Usability Scale* (SUS).

Berikut adalah Rumus yang digunakan untuk menghitung skor SUS [32].

Setelah pengumpulan jawaban dari responden maka dilakukan penghitungan berdasarkan pertanyaan nomor ganjil dan genap dengan cara penghitungan skor yang berbeda.

Skor Nomor Genap (2, 4, 6, 8, dan 10) (2.1)

Skor Nomor Genap = 5 – Skor (2.2)

Skor Nomor Ganjil (1, 3, 5, 7, dan 9) (2.3)

$$\text{Skor Nomor Ganjil} = \text{Skor} - 1 \quad (2.4)$$

Kemudian dijumlahkan semua hasil untuk mendapatkan total skornya.

$$\text{Total Skor} = \sum (\text{Skor Genap} + \text{Skor Ganjil}) \quad (2.5)$$

Untuk mendapatkan hasil akhir dari SUS.

$$\text{Skor SUS} = \left(\sum (\text{Skor Genap} + \text{Skor Ganjil}) \right) * 2.5 \quad (2.6)$$

Kemudian digunakan juga Likert Scale yang akan menghitung hal spesifik mengenai seperti kepuasan dan kemudahan penggunaan. Berikut adalah contoh kuesioner yang akan diberikan ke pada pengguna.

Tabel 3.5 Contoh Pertanyaan Likert Scale

No	Pertanyaan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Antarmuka website mudah dipahami					
2	Saya merasa nyaman menggunakan website ini.					
3	Fitur yang tersedia di website mudah diakses.					
4	Website mampu menampilkan informasi yang relevan dengan fungsinya.					
5	Hasil skrining oleh website akurat dan sesuai harapan saya.					
6	Website merespon dengan cepat.					
7	Saya merasa puas dengan kinerja keseluruhan website.					
8	Website memenuhi kebutuhan saya dalam tahap awal skrining TBC.					
9	Saya akan merekomendasikan website ini kepada orang lain.					

Selanjutnya, proses penghitungan hasil evaluasi dilakukan untuk memperoleh nilai rata-rata dari setiap pernyataan. Rumus ini digunakan untuk mengubah data kualitatif berupa tanggapan responden menjadi data kuantitatif yang dapat dianalisis lebih lanjut. Adapun rumus yang digunakan dalam perhitungan disajikan sebagai berikut.

Sangat Tidak setuju = 1

Tidak Setuju = 2

Netral = 3

Setuju = 4

Sangat Setuju = 5

Untuk menghitung tiap skor yaitu mengkalikan berapa banyak respon untuk tiap point dengan nilai respon kemudian dijumlahkan hasilnya.

Menghitung skor rata rata :

$$Average\ Score = \frac{Total\ Score}{Total\ Responses} \quad (2.8)$$

Untuk menilai Kesimpulan dari hasil perhitungan dapat diartikan sebagai berikut :

Skor mendekati 5 menandakan kepuasan yang tinggi atau kesetujuan yang tinggi.

Skot mendekati 1 menandakan kepuasan yang rendah atau ketidaksetujuan yang tinggi.

3.3.6 Pemeliharaan

Pada tahap pemeliharaan akan dilakukan perbaikan apabila terdapat *bug/error* setelah *deployment* sistem berjalan serta optimasi dan pembaruan model/data atau fitur jika diperlukan. Hal ini ditujukan untuk menjaga kualitas Web Skrining TBC dan kenyamanan pengguna dalam menggunakan layanan tersebut. Pemantauan sistem akan dilakukan secara berkala untuk mengevaluasi perfoma sistem dan respon pengguna.

Terdapat 4 kategori utama dalam pemeliharaan [17], yaitu :

1. Corrective Maintenance
Perbaikan atas kesalahan atau bug yang muncul setelah sistem digunakan oleh pengguna.
2. Adaptive Maintenance
Penyesuaian sistem terhadap perubahan lingkungan operasional seperti pembaruan *browser*, versi python, atau platform hosting.
3. Perfective Maintenance
Penyempurnaan atau penambahan fitur sesuai kebutuhan pengguna atau instansi tanpa mengubah struktur utama.
4. Preventive Maintenance
Aktivitas preventif untuk mencegah munculnya masalah kedepannya seperti refactoring kode, dokumentasi ulang, atau audit performa agar sistem tetap stabil dan hemat biaya.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan proses penelitian, analisis, perancangan, implementasi, serta pengujian yang telah dilakukan, dapat disimpulkan bahwa:

1. Sistem Skrining TBC berbasis web berhasil dikembangkan menggunakan *framework Flask* dengan dimulai dari analisis kebutuhan, perancangan desain sistem, implementasi sistem, pengujian, deployment, hingga dilakukannya pemeliharaan pada lingkungan production. Sistem mampu melakukan proses skrining awal TBC dengan model *Machine Learning* yang telah diintegrasikan dengan baik menggunakan pendekatan *lazy loading* sehingga dapat mencegah beban berlebih pada server.
2. Integrasi model Machine Learning untuk skrining gejala TBC telah berjalan efektif dan stabil pada lingkungan shared hosting melalui penerapan mekanisme *lazy loading*. Hasil implementasi menunjukkan bahwa model dapat dipanggil dan digunakan secara konsisten tanpa membebani server, sehingga mendukung proses skrining awal di lapangan dengan baik.
3. Hasil pengujian fungsional melalui metode *Black Box Testing* dan Postman menunjukkan seluruh fitur berjalan sesuai kebutuhan dengan tingkat keberhasilan 100% tanpa *error* yang menghambat proses Website Utama Skrining TBC dan Sistem Database Kader (SIDAK). Pengujian keamanan menggunakan OWASP ZAP juga dilakukan; pemindaian otomatis terhenti sekitar 16% karena pemblokiran IP oleh mekanisme proteksi server, namun meskipun tidak lengkap, ZAP tetap mengidentifikasi beberapa temuan dengan tingkat risiko rendah hingga menengah yang menjadi bahan perhatian untuk pengembangan selanjutnya. Hal ini membuktikan bahwa website sudah siap 100% untuk digunakan secara umum.

4. Evaluasi sistem melalui kuesioner SUS menunjukkan bahwa SIDAK dan website Skrining TBC mudah digunakan serta mendukung alur kerja kader, dengan skor 80,64%. Fitur inti seperti manajemen data kader dan statistik responden dapat dioperasikan dengan lancar. SUS responden yang fokus pada penggunaan website untuk skrining mandiri, memperoleh skor 82,83%, menandakan sistem cukup intuitif. Hasil Skala Likert dari kedua kelompok pengguna menunjukkan tingkat kepuasan sebesar 95,7%, sehingga sistem tidak hanya memenuhi kebutuhan fungsional, tetapi juga diterima dengan baik di lapangan.

5. Secara keseluruhan sistem Skrining TBC yang dikembangkan berhasil memenuhi tujuan penelitian. Sistem mampu menjalankan fitur fungsional dengan baik dan menyediakan mekanisme skrining awal gejala TBC yang dapat digunakan oleh kader dan responden. Integrasi model *Machine Learning* dengan proses skrining, pengelolaan data responden, dan antarmuka yang mudah dipahami menjadikan sistem ini berpotensi mendukung proses identifikasi awal pasien terduga TBC, walaupun implementasi di lapangan ke depannya masih perlu pengamatan lebih lanjut.

5.2 Saran

1. Menambahkan fitur notifikasi kader dari hasil skrining responden yang berada di domisili yang sama dengan kader.
2. Meningkatkan spesifikasi server atau mempertimbangkan penggunaan server dengan kapasitas lebih tinggi untuk mendukung beban proses model Machine Learning, notifikasi real-time, serta akses simultan oleh pengguna di lapangan.
3. Menambahkan mekanisme keamanan yang lebih komprehensif seperti audit keamanan berkala, dan enkripsi data sensitif.

DAFTAR PUSTAKA

- [1] S. D. Pralambang and S. Setiawan, "Faktor Risiko Kejadian Tuberkulosis di Indonesia," *Jurnal Biostatistik, Kependudukan, dan Informatika Kesehatan*, vol. 2, no. 1, Nov. 2021.
- [2] "Inisiatif Lampung Sehat (ILS)." Accessed: Jul. 29, 2025. [Online]. Available: <https://lampungsehat.org>
- [3] J. R. Lewis, "The System Usability Scale: Past, Present, and Future," *Int. J. Hum. Comput. Interact.*, vol. 34, no. 7, pp. 577–590, 2018.
- [4] C. Alves, *Bootstrap Reference Guide: Web Development with Bootstrap*, 2nd ed. Independently Published, 2021.
- [5] E. Marcotte, *Responsive Web Design*, 2nd ed. NY, USA: A Book Apart, 2014.
- [6] R. Tabarés, "HTML5 and the evolution of HTML; tracing the origins of digital platforms," *Technol. Soc.*, vol. 65, p. 101529, 2021, Accessed: Feb. 19, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0160791X2100004X>
- [7] M. Grinberg, *Flask Web Development*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
- [8] C. Severance, "Guido van Rossum: The Early Years of Python," *Computer (Long Beach. Calif.)*, vol. 48, no. 2, pp. 7–9, 2015.
- [9] K. R. Srinath, "Python-The Fastest Growing Programming Language," *International Research Journal of Engineering and Technology*, 2017, Accessed: Feb. 19, 2026. [Online]. Available: www.irjet.net
- [10] K. Mar'iyah and Z. Zulkarnain, "Patofisiologi penyakit infeksi tuberkulosis," in *Prosiding Seminar Nasional Biologi*, Gowa, Nov. 2021, pp. 88–92. Accessed: Feb. 19, 2026. [Online]. Available: <https://journal.uin-alauddin.ac.id/index.php/psb/article/view/23169>
- [11] World Health Organization, "Global Tuberculosis Report 2024," 2024. [Online]. Available: <https://iris.who.int/bitstream/handle/10665/379339/9789240101531-eng.pdf?sequence=1>
- [12] World Health Organization, "Global Tuberculosis Report 2024," Geneva, Switzerland, 2024. Accessed: Feb. 19, 2026. [Online]. Available: <https://iris.who.int/server/api/core/bitstreams/7292c91e-ffb0-4cef-ac39-0200f06961ea/content>

- [13] M. Stonebraker, L. Rowe, and M. Hirohama, "The Implementation Of Postgres," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 2, pp. 125–142, Mar. 1990.
- [14] S. Izzat and N. N. Saleem, "Software Testing Techniques and Tools: A Review," *Journal of Education and Science*, vol. 32, pp. 31–40, Feb. 2023, Accessed: Feb. 19, 2026. [Online]. Available: https://www.researchgate.net/publication/371214561_Software_Testing_Techniques_and_Tools_A_Review
- [15] M. E. Khan, "Different approaches to white box testing technique for finding errors," *International Journal of Software Engineering and its Applications*, vol. 5, no. 3, pp. 1–14, 2011.
- [16] I. Sommerville, *Software engineering*, 9th ed. Boston, MA: Pearson Education, 2011.
- [17] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed. New York: McGraw-Hill Education, 2014.
- [18] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*. Reading, MA, USA: Addison-Wesley, 1999.
- [19] I.-Y. Song, M. Evans, and U. E. K. Park, "A Comparative Analysis of Entity-Relationship Diagrams 1," *Journal of Computer and Software Engineering*, vol. 3, no. 4, pp. 427–459, 1995.
- [20] P. P.-S. Chen, "The entity-relationship model—toward a unified view of data," *ACM Trans. Database Syst.*, vol. 1, no. 1, pp. 9–36, Mar. 1976.
- [21] Postman, "Postman API Platform." Accessed: Oct. 20, 2025. [Online]. Available: <https://www.postman.com>
- [22] ZAP, "Introducing ZAP", Accessed: Oct. 20, 2025. [Online]. Available: <https://www.zaproxy.org>
- [23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2016, pp. 785–794.
- [24] E. Astha Triyono *et al.*, "Early Detection of Tuberculosis Application (E-TIBI): A New Paradigm to Detect New Cases of Tuberculosis," *Jurnal Berkala Epidemiologi*, vol. 11, no. 3, pp. 267–276, Sep. 2023.
- [25] J. S. Pramono, D. Hendriani, D. Ardyanti, and N. A. Chifdillah, "Implementasi Aplikasi Deteksi Dini Suspek Tuberkulosis Berbasis mHealth di antara Kontak Serumah: Tinjauan Sistematis," *Jurnal Kesehatan Vokasional*, vol. 8, no. 3, p. 163, Sep. 2023.

- [26] S. R. Rahayu *et al.*, "Pengembangan Aplikasi SIKRIBO (Sistem Skrining Tuberkulosis)," *Bookchapter Kesehatan Masyarakat Universitas Negeri Semarang*, no. 2, pp. 1–30, May 2022.
- [27] N. A. Damayanti *et al.*, "Capacity Building Skrining TBC Anak di Kabupaten Jember," *Jurnal Pengabdian Kepada Masyarakat*, vol. 30, Mar. 2024.
- [28] N. Y. Ratnasari and S. Handayani, "Tuberculosis Research Trends in Indonesian Health Scientific Journals: From Research Design to Data Analysis," *Kemas*, vol. 18, no. 4, pp. 483–491, 2023.
- [29] S. Kumar, J. Ojha, M. M. Tripathi, and K. Garg, "Integrated e-healthcare management system using machine learning and flask," *Int. J. Electron. Healthc.*, vol. 13, no. 1, pp. 71–90, 2023.
- [30] T. Thesing, C. Feldmann, and M. Burchardt, "Agile versus Waterfall Project Management: Decision model for selecting the appropriate approach to a project," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 746–756.
- [31] C. Fagarasan, O. Popa, A. Pisla, and C. Cristea, "Agile, waterfall and iterative approach in information technology projects," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1169, no. 1, p. 012025, Aug. 2021, [Online]. Available: <https://iopscience.iop.org/article/10.1088/1757-899X/1169/1/012025>
- [32] J. Brooke, "SUS: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, Mar. 1995.