

**SISTEM INFORMASI PEMETAAN DAN DIAGNOSA DINI
PENYAKIT TBC MENGGUNAKAN METODE *NAIVE BAYES*
DI KOTA PANGKALPINANG**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Sarjana Terapan/Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Andika Saputra

NIM: 1062203

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2025**

LEMBAR PENGESAHAN

SISTEM INFORMASI PEMETAAN DAN DIAGNOSA DINI PENYAKIT TBC MENGGUNAKAN METODE *NAIVE BAYES* DI KOTA PANGKALPINANG

Oleh:

Andika Saputra NIM: 1062203

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan/Diploma IV Politeknik Manufaktur Negeri Bangka
Belitung

Menyetujui,

Pembimbing 1



Riki Afriatasyah, S.T., M.T.
NIP. 199004042019031013

Pembimbing 2



Ocsirendi, S.ST., M.T.
NIP. 198710192024211014

Penguji 1



Yang Agita Rindri, S.Kom., M.Eng.
NIP. 198609282022032003

Penguji 2



Better Swengky, M.Kom.
NIP. 199301222024061001

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa : Andika Saputra NIM: 1062203

Dengan Judul : SISTEM INFORMASI PEMETAAN DAN DIAGNOSA
DINI PENYAKIT TBC MENGGUNAKAN METODE *NAIVE BAYES* DI KOTA
PANGKALPINANG

Menyatakan bahwa laporan akhir ini adalah hasil kerja saya sendiri dan bukan merupakan plagiat. Pernyataan ini saya buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, saya bersedia menerima sanksi yang berlaku.

Sungailiat, 21 Juli 2025

Nama Mahasiswa

Tanda Tangan

Andika Saputra



ABSTRAK

Penyakit Tuberkulosis (TBC) masih menjadi adapun satu dari sekian banyaknya masalah kesehatan utama di Indonesia, termasuk di Kota Pangkalpinang. Tantangan seperti keterlambatan diagnosis dan kurangnya penyebaran informasi menjadi faktor penghambat utama dalam penanganan kasus TBC. Penelitian ini mengembangkan sebuah sistem informasi yang memadukan pemetaan geografis dan sistem pakar berbasis metode Naive Bayes untuk membantu proses diagnosa dini TBC. Sistem ini dirancang agar masyarakat dapat lebih mudah mengetahui persebaran kasus TBC dan melakukan pemeriksaan gejala secara mandiri melalui antarmuka web yang ramah pengguna. Berdasarkan hasil pengujian, model Bernoulli Naive Bayes yang diterapkan mampu mencapai tingkat akurasi sebesar 95,24% dengan precision 1.00 dan recall 0.93. Selain itu, hasil UAT menunjukkan bahwa sistem diterima dengan sangat baik oleh pengguna dengan nilai kepuasan mencapai 91,56%. Dengan adanya sistem ini, diharapkan proses pendeteksian dini dan pemantauan penyebaran TBC dapat dilakukan dengan lebih cepat dan akurat.

Kata Kunci: Tuberkulosis, Diagnosa Dini, Naive Bayes, Sistem Pakar, Sistem Informasi Geografis

ABSTRACT

Tuberculosis (TB) remains a major public health concern in Indonesia, particularly in Pangkalpinang City. Issues such as delayed diagnosis and lack of information dissemination have become major barriers in managing TB cases. This study developed an information system that integrates geographic mapping and an expert system based on the Naive Bayes method to assist early diagnosis of TB. The system is designed to help the public identify the spread of TB cases and self-assess symptoms through a user-friendly web interface. Based on testing results, the implemented Bernoulli Naive Bayes model achieved an accuracy of 95.24%, with a precision of 1.00 and recall of 0.93. Furthermore, UAT results indicated strong user acceptance with a satisfaction rate of 91.56%. It is expected that this system can improve the speed and accuracy of early detection and monitoring of TB spread.

Keywords: Tuberculosis, Early Diagnosis, Naive Bayes, Expert System, Geographic Information System

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh,

Alhamdulillah segala puji dan syukur penulis panjatkan ke hadirat Allah SWT. atas limpahan rahmat, karunia, serta hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan Laporan Akhir dengan judul “Sistem Informasi Pemetaan dan Diagnosa Dini Penyakit TBC menggunakan Metode *Naive Bayes* di Kota Pangkalpinang”.

Laporan akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan jenjang Diploma IV pada Program Studi Teknologi Rekayasa Perangkat Lunak Politeknik Manufaktur Negeri Bangka Belitung. Penulis menyadari bahwa pencapaian ini tidak terlepas dari doa, dukungan dan bimbingan dari berbagai pihak yang telah tulus dan ikhlas membantu selama proses pengerjaan proyek akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
2. Bapak Irwan, M.Sc, Ph.D selaku Wakil Direktur I Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Muhammad Subhan, M.T selaku Wakil Direktur II Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Eko Sulistyono, M.T selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung.
5. Ibu Yang Agita Rindri, M.Eng selaku Ka. Jurusan Informatika dan Bisnis Politeknik Manufaktur Negeri Bangka Belitung.
6. Bapak Sidhiq Andriyanto, M.Kom selaku Ka. Program Studi D4 Teknologi Rekayasa Perangkat Lunak Politeknik Manufaktur Negeri Bangka Belitung.

7. Bapak Riki Afriansyah, S.T., M.T., selaku Dosen Pembimbing I, yang telah menyediakan waktu, tenaga, dan pikiran untuk membantu penulis dalam penyusunan laporan proyek akhir ini.
8. Bapak Ocsirendi, S.ST., M.T., selaku Dosen Pembimbing II, yang telah menyediakan waktu, tenaga, dan pikiran untuk membantu penulis dalam penyusunan laporan proyek akhir ini.
9. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral kepada penulis.
10. Teman dan seluruh pihak yang telah banyak membantu penulis dalam menyelesaikan laporan proyek akhir ini.

Penulis menyadari bahwa penyusunan laporan proyek akhir ini masih jauh dari kata sempurna, maka dari itu penulis memohon maaf apabila terdapat kesalahan kata, pengetikan, dan kekeliruan. Penulis berharap laporan ini dapat bermanfaat bagi kepentingan bersama dan dapat membawa manfaat bagi pengembangan ilmu. Akhir kata, penulis mengucapkan terima kasih.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Sungailiat, 21 Juli 2025



Andika Saputra

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN BUKAN PLAGIAT	i
ABSTRAK	iii
<i>ABSTRACT</i>	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN.....	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan Proyek Akhir.....	3
BAB 2 DASAR TEORI	4
2.1 Penelitian Terdahulu.....	4
2.2 Dasar Teori.....	7
2.2.1 Sistem Informasi Geografis.....	7
2.2.2 Sistem Pakar.....	8
2.2.3 Metode <i>Prototype</i>	8
2.2.4 <i>Unified Modeling Language (UML)</i>	10
2.2.5 Metode <i>Naive Bayes</i>	12
2.2.6 <i>Laplace Smoothing</i>	14
2.2.7 <i>User Acceptance Testing (UAT)</i>	15
BAB 3 METODE PELAKSANAAN	17
3.1 Diagram Alir.....	17
3.2 Identifikasi Masalah	18
3.3 Pengumpulan Data.....	18

3.4	Analisis Kebutuhan	18
3.4.1	Analisis Kebutuhan Perangkat Lunak.....	19
3.4.2	Analisis Kebutuhan Fungsional	19
3.4.3	Analisis Kebutuhan Nonfungsional	19
3.5	Perancangan Sistem.....	20
3.5.1	<i>Use Case Diagram</i>	20
3.5.2	<i>Activity Diagram</i>	21
3.5.3	<i>Entity Relationship Diagram (ERD)</i>	30
3.6	<i>Prototyping</i>	30
3.7	Pengkodean.....	35
3.8	Pengujian	35
3.9	Laporan.....	36
BAB 4 PEMBAHASAN		37
4.1	Hasil Analisa dan Perancangan Sistem	37
4.2	Perancangan <i>Database</i>	37
4.2.1	<i>Table Users</i>	38
4.2.2	<i>Table Chat_Messages</i>	38
4.2.3	<i>Table Patients</i>	39
4.2.4	<i>Table Districts</i>	39
4.2.5	<i>Table Diagnose_Histories</i>	40
4.2.6	<i>Table Articles</i>	40
4.2.7	<i>Table Diseases</i>	40
4.3	Hasil Tampilan Antarmuka	41
4.3.1	Tampilan Halaman Utama	41
4.3.2	Tampilan Halaman <i>Login</i>	42
4.3.3	Tampilan Halaman <i>Register</i>	42
4.3.4	Tampilan Halaman Berita	43
4.3.5	Tampilan Halaman Detail Berita	43
4.3.6	Tampilan Pemetaan.....	44
4.3.7	Tampilan Diagnosa Dini	44
4.3.8	Tampilan Hasil Diagnosa Dini.....	45

4.3.9	Tampilan Riwayat Diagnosa	45
4.3.10	Tampilan <i>Chatbot</i>	46
4.3.11	Tampilan Halaman Konsultasi.....	46
4.3.12	Tampilan <i>Dashboard Admin</i>	47
4.3.13	Tampilan <i>Dashboard</i> Kelola Pasien	47
4.3.14	Tampilan <i>Dashboard</i> Pemetaan.....	48
4.3.15	Tampilan <i>Dashboard</i> Kelola <i>User</i>	48
4.3.16	Tampilan <i>Dashboard</i> Kelola Berita	48
4.3.17	Tampilan <i>Dashboard</i> Kelola Gejala	49
4.3.18	Tampilan <i>Dashboard</i> Kelola Kecamatan.....	49
4.4	Pengujian Model <i>Naive Bayes</i>	50
4.4.1	Data Pasien TBC	53
4.4.2	Pembagian Data (<i>Split Validation</i>).....	54
4.4.3	Evaluasi Model.....	56
4.4.4	Hasil Pengujian	61
4.5	Pengujian Sistem	64
4.5.1	Pengujian Metode <i>User Acceptance Testing (UAT)</i>	64
BAB 5 KESIMPULAN DAN SARAN.....		73
5.1	Kesimpulan.....	73
5.2	Saran.....	73
DAFTAR PUSTAKA		75

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	4
Tabel 2. 2 Simbol <i>Use Case Diagram</i>	11
Tabel 2. 3 Simbol <i>Activity Diagram</i>	12
Tabel 2. 4 Kriteria Skor UAT	15
Tabel 2. 5 Persentase Jawaban UAT	16
Tabel 4. 1 Kode Gejala TBC.....	54
Tabel 4. 2 Rasio Pembagian Data	55
Tabel 4. 3 Data Gejala <i>False Negative</i>	58
Tabel 4. 4 Gejala Pengujian Data Uji.....	61
Tabel 4. 5 Kuesioner UAT	64
Tabel 4. 6 Hasil Kuesioner Aspek Tingkat Kemudahan Pengguna.....	66
Tabel 4. 7 Hasil Kuesioner Aspek Tingkat Kepuasan Pengguna.....	67
Tabel 4. 8 Hasil Kuesioner Aspek Fungsionalitas Sistem	69
Tabel 4. 9 Hasil Kuesioner Aspek Kinerja Sistem.....	71

DAFTAR GAMBAR

Gambar 2. 1 Metode <i>Prototype</i>	9
Gambar 3. 1 Diagram Alir	17
Gambar 3. 2 <i>Use Case Diagram</i>	20
Gambar 3. 3 <i>Activity Diagram Register</i>	21
Gambar 3. 4 <i>Activity Diagram</i> Diagnosa.....	22
Gambar 3. 5 <i>Activity Diagram</i> Kelola Gejala	23
Gambar 3. 6 <i>Activity Diagram</i> Kelola Artikel	24
Gambar 3. 7 <i>Activity Diagram</i> Kelola Kecamatan.....	25
Gambar 3. 8 <i>Activity Diagram</i> Kelola Pasien	26
Gambar 3. 9 <i>Activity Diagram</i> Kelola User	27
Gambar 3. 10 <i>Activity Diagram</i> Konsultasi <i>Admin</i>	28
Gambar 3. 11 <i>Activity Diagram</i> Konsultasi <i>Chatbot</i>	29
Gambar 3. 12 <i>Entity Relationship Diagram</i>	30
Gambar 3. 13 Rancangan Halaman Utama	31
Gambar 3. 14 Rancangan Halaman Login	31
Gambar 3. 15 Rancangan Halaman Register	32
Gambar 3. 16 Rancangan Halaman Berita	32
Gambar 3. 17 Rancangan Halaman Detail Berita	33
Gambar 3. 18 Rancangan Halaman Diagnosa Dini	33
Gambar 3. 19 Rancangan Halaman Hasil Diagnosa	34
Gambar 3. 20 Rancangan Halaman Riwayat Diagnosa	34
Gambar 3. 21 Rancangan Halaman <i>Dashboard Admin</i>	35
Gambar 4. 1 Hasil Rancangan Database	38
Gambar 4. 2 <i>Table Users</i>	38
Gambar 4. 3 <i>Table Chat_Messages</i>	39
Gambar 4. 4 <i>Table Patients</i>	39
Gambar 4. 5 <i>Table Districts</i>	39

Gambar 4. 6 <i>Table Diagnose_Histories</i>	40
Gambar 4. 7 <i>Table Articles</i>	40
Gambar 4. 8 <i>Table Diseases</i>	40
Gambar 4. 9 Tampilan Halaman Utama	41
Gambar 4. 10 Tampilan Halaman <i>Login</i>	42
Gambar 4. 11 Tampilan Halaman <i>Register</i>	42
Gambar 4. 12 Tampilan Halaman Berita	43
Gambar 4. 13 Tampilan Halaman Detail Berita.....	43
Gambar 4. 14 Tampilan Peta Penyebaran	44
Gambar 4. 15 Tampilan <i>Form</i> Diagnosa.....	44
Gambar 4. 16 Tampilan Hasil Diagnosa	45
Gambar 4. 17 Tampilan Halaman Riwayat Diagnosa.....	45
Gambar 4. 18 Tampilan <i>Chatbot</i>	46
Gambar 4. 19 Tampilan Halaman Konsultasi Admin	46
Gambar 4. 20 Tampilan Halaman <i>Dashboard Admin</i>	47
Gambar 4. 21 Tampilan Halaman Kelola Pasien	47
Gambar 4. 22 Tampilan Halaman Peta Penyebaran <i>Admin</i>	48
Gambar 4. 23 Tampilan Halaman Kelola <i>User</i>	48
Gambar 4. 24 Tampilan Halaman Kelola Berita.....	49
Gambar 4. 25 Tampilan Halaman Kelola Gejala	49
Gambar 4. 26 Tampilan Halaman Kelola Kecamatan	50
Gambar 4. 27 Tahap <i>Cleaning</i> Data.....	50
Gambar 4. 28 Tahap Pembagian Data.....	51
Gambar 4. 29 Tahap Melatih Data	51
Gambar 4. 30 Tahap Pengujian Data	52
Gambar 4. 31 Tahap Pelatihan dan Pengujian Model.....	52
Gambar 4. 32 Tahap Evaluasi Model.....	52
Gambar 4. 33 Potongan <i>Dataset</i>	53
Gambar 4. 34 Alur <i>Split Validation</i>	54
Gambar 4. 35 Potongan <i>Data Training</i>	55
Gambar 4. 36 Potongan Data Testing	56

Gambar 4. 37 Hasil <i>Confusion Matrix</i>	57
Gambar 4. 38 Hasil <i>Classification Report</i>	57
Gambar 4. 39 Hasil Pengujian Data Gejala <i>False Negative</i>	58
Gambar 4. 40 Korelasi Fitur Dengan Klasifikasi.....	60
Gambar 4. 41 Hasil Perhitungan Program	61
Gambar 4. 42 Grafik Persentase <i>Index</i> Tingkat Kemudahan Pengguna	67
Gambar 4. 43 Grafik Persentase <i>Index</i> Tingkat Kepuasan Pengguna.....	69
Gambar 4. 44 Grafik Persentase <i>Index</i> Fungsionalitas Sistem	70
Gambar 4. 45 Grafik Persentase <i>Index</i> Kinerja Sistem.....	72



DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup

Lampiran 2 *Form* Bimbingan

Lampiran 3 *Form Monitoring*

Lampiran 4 Wawancara Pasien TBC

Lampiran 5 Bukti Kuesioner *User Acceptance Testing* (UAT)



BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Tuberkulosis (TBC) merupakan salah satu penyakit infeksi yang sangat mudah menular dan hingga kini masih menjadi tantangan serius dalam dunia kesehatan di Indonesia. Penyakit ini tidak hanya menjadi salah satu penyebab utama kematian di Indonesia, tetapi juga di seluruh dunia. Selama dua tahun terakhir, kasus TBC terus mengalami peningkatan, menciptakan persoalan yang kompleks baik dari sisi medis maupun sosial [1].

Menurut *Global TB Report 2023*, Indonesia menempati peringkat kedua dunia dalam jumlah kasus TBC, dengan estimasi sekitar 1.060.000 kasus dan 134.000 kematian setiap tahunnya akibat penyakit ini [2]. Tuberkulosis paru, atau yang lebih dikenal sebagai TB paru, adalah penyakit menular yang disebabkan oleh bakteri *Mycobacterium Tuberculosis*. Penyakit ini lebih mudah menyerang individu dengan daya tahan tubuh lemah, seperti penderita HIV/AIDS atau mereka yang mengalami kekurangan gizi. Penularan TB paru terjadi melalui udara, khususnya saat penderita batuk, bersin, atau bahkan berbicara. Dalam satu kali batuk, penderita bisa melepaskan sekitar 3.000 partikel dahak (*aerosol*) yang mengandung bakteri ke udara, sehingga orang di sekitarnya berisiko terinfeksi. Jika pengobatan tidak dijalani dengan benar dan rutin selama minimal 6 bulan, TB paru bisa menjadi penyakit yang mematikan. Dampaknya pun tak hanya dirasakan oleh pasien, tetapi juga oleh keluarganya. Secara psikologis, penderita bisa mengalami kecemasan, penurunan rasa percaya diri, dan kurangnya dukungan dari lingkungan sekitar [3].

TBC bisa menyerang siapa saja, namun paling rentan dialami oleh mereka yang masih berada di usia produktif maupun anak-anak. Sekitar 75% penderita TBC berasal dari kelompok usia 15 hingga 50 tahun, yaitu usia yang secara ekonomi paling aktif. Seorang penderita TBC dewasa diperkirakan akan

kehilangan waktu kerja selama 3 hingga 4 bulan. Jika tidak tertolong dan meninggal, maka potensi kehilangan pendapatan bisa mencapai hingga 15 tahun. Selain dampak ekonomi yang besar, penderita TBC juga kerap menghadapi tekanan sosial, termasuk risiko dikucilkan oleh lingkungan sekitar [4].

Di Kota Pangkalpinang, tingkat kesembuhan TBC masih jauh dari target nasional. Berdasarkan data Dinas Kesehatan Kota Pangkalpinang tahun 2018, angka kesembuhan TBC hanya mencapai 67,9%, lebih rendah dari target nasional sebesar 85%. Kurangnya pengetahuan dan akses informasi dapat menyebabkan peningkatan penyebaran serta keterlambatan dalam proses diagnosis dan pengobatan. Hingga saat ini, belum tersedia sistem informasi yang dapat memetakan pasien TBC di Kota Pangkalpinang.

Sudah ada beberapa penelitian dengan topik yang kurang lebih sama, seperti yang dilakukan oleh [1] dengan judul “Implementasi Metode *Naive Bayes* pada Sistem Pakar Diagnosa Penyakit *Tuberculosis*”, penelitian yang dilakukan oleh [5] dengan judul “Pemetaan Penyakit Tuberkulosis dengan Sistem Informasi Geografis di Wilayah Bantul” dan penelitian yang dilakukan oleh [6] dengan judul “Implementasi *Data Mining* Untuk Diagnosa Prediksi Penyakit *Tuberculosis* dengan Menggunakan Algoritma *Naive Bayes*”. Oleh karena itu, Penulis berencana untuk mengembangkan topik tersebut dengan judul “Sistem Informasi Pemetaan dan Diagnosa Dini Penyakit TBC Menggunakan Metode *Naive Bayes* di Kota Pangkalpinang”. Namun, belum ada sistem yang menggabungkan kedua pendekatan tersebut dalam satu *platform* yang terintegrasi.

Oleh karena itu, penulis mengembangkan proyek akhir dengan judul “Sistem Informasi Pemetaan dan Diagnosa Dini Penyakit TBC Menggunakan Metode *Naive Bayes* di Kota Pangkalpinang”. Sistem ini bertujuan untuk membantu proses diagnosis dini secara digital melalui sistem pakar berbasis gejala, sekaligus menyajikan peta sebaran pasien berbasis SIG. Dengan sistem ini, diharapkan informasi mengenai penyebaran dan deteksi dini penyakit TBC dapat disampaikan secara lebih cepat, akurat, dan mudah diakses oleh masyarakat maupun pihak kesehatan.

1.2 Perumusan Masalah

Sebagaimana pemaparan latar belakang yang telah dipaparkan sebelumnya, berikut ini rumusan yang menjadi *focus* dalam penelitian ini:

1. Bagaimana merancang sistem informasi pemetaan untuk memetakan penyebaran penyakit TBC di Kota Pangkalpinang?
2. Bagaimana merancang sistem pakar untuk mendiagnosa dini penyakit TBC?
3. Bagaimana merancang sistem untuk mempermudah dalam memberikan edukasi dan informasi terkait penyakit TBC di Kota Pangkalpinang?

1.3 Tujuan Proyek Akhir

Adapun tujuannya dari proyek akhir dengan judul tersebut yakni:

1. Mempermudah dalam memberikan edukasi dan informasi terkait penyebaran dan juga segala informasi terkait penyakit TBC.
2. Membuat sistem untuk memetakan pasien penyakit TBC yang ada di Kota Pangkalpinang guna melihat penyebaran penyakit TBC.
3. Membuat sistem pakar untuk mendiagnosa atau mendeteksi dini penyakit TBC berdasarkan gejala-gejala yang dialami oleh pasien.

BAB 2 DASAR TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu sebagai kajian terhadap studi-studi sebelumnya yang mempunyai keterkaitannya dengan topik penelitian ini. Adapun hasilnya dari penelitian tersebut dirangkum dan disajikan dalam Tabel 2.1 berikut:

Tabel 2. 1 Penelitian Terdahulu

No	Judul	Metode	Hasil
1	Implementasi Pakar Diagnosa Penyakit <i>Tuberculosis</i> [1]	Metode Sistem Pakar Diagnosa Penyakit <i>Tuberculosis</i> [1]	Penelitian ini mengembangkan sistem pakar diagnosa penyakit <i>Tuberculosis</i> berbasis web untuk membantu pasien mengenali gejala secara mandiri. Sistem menggunakan metode <i>Naive Bayes</i> dalam proses klasifikasi berdasarkan data gejala yang diinput. Dengan tingkat akurasi 81,81%, sistem terbukti efektif membantu diagnosa awal dan memberikan informasi yang cepat dan efisien sesuai kebutuhan pasien.
2	Perancangan Sistem Pakar Diagnosis Penyakit TBC Paru dengan Metode <i>Certainty Factor</i> dan <i>Dempster Shafer</i> [2]	Metode Sistem Pakar Diagnosis Penyakit TBC Paru dengan Metode <i>Certainty Factor</i> dan <i>Dempster Shafer</i> [2]	Penelitian ini merancang sistem pakar berbasis web untuk mendiagnosis TBC paru menggunakan metode <i>Certainty Factor</i> dan <i>Dempster Shafer</i> . Sistem ini memproses input gejala dari pengguna dengan perhitungan tingkat keyakinan dan menggabungkan nilai kepercayaan setiap gejala untuk menentukan hasil

			diagnosis. Dari 85 data yang diuji, sistem mencapai akurasi sebesar 88,2%, menunjukkan potensi metode ini dalam membantu deteksi dini TBC secara efisien dan akurat.
3	Pemetaan Tuberkulosis Sistem Geografis di Bantul [5]	Penyakit Dengan Informasi Wilayah	<p>Penelitian ini memanfaatkan Sistem Informasi Geografis (SIG) untuk memetakan sebaran kasus Tuberkulosis (TBC) di wilayah Kabupaten Bantul. Proses analisis dilakukan menggunakan aplikasi <i>Quantum GIS</i>, dengan hasil yang divisualisasikan dalam bentuk peta dan grafik. Dari hasil pemetaan, diketahui bahwa kasus TBC terbanyak ditemukan di Kecamatan Bantul, Banguntapan, dan Pandak. Beberapa faktor lingkungan, seperti curah hujan, tingkat kelembaban, dan kepadatan penduduk, turut memengaruhi pola persebaran penyakit. Penggunaan SIG dalam penelitian ini terbukti efektif untuk menyajikan data secara visual serta mendukung pengambilan keputusan berdasarkan wilayah yang terdampak.</p>
4	Penerapan Teorema <i>Bayes</i> Mendiagnosa Tuberculosis [7]	Metode Penyakit	<p>Penelitian ini mengembangkan sistem pakar untuk mendiagnosis penyakit Tuberculosis (TBC) menggunakan metode Teorema <i>Bayes</i>. Sistem bertujuan membantu masyarakat dalam</p>

mengenali gejala TBC secara lebih cepat dan efisien, khususnya untuk mengatasi hambatan biaya pemeriksaan. Metode Teorema *Bayes* digunakan untuk menghitung probabilitas diagnosa berdasarkan gejala yang dialami pasien. Hasil penelitian menunjukkan bahwa sistem mampu mendiagnosis TBC Paru dengan tingkat keakuratan sebesar 87%, membuktikan efektivitas sistem pakar berbasis probabilistik dalam mendukung deteksi dini penyakit menular ini.

5	Sistem Geografis Penyebaran TBC di Provinsi Jambi Tahun 2022 [8].	Informasi Pemetaan TBC di Tahun	Penelitian ini memanfaatkan Sistem Informasi Geografis (SIG) untuk memetakan persebaran kasus <i>Tuberculosis</i> (TBC) di Provinsi Jambi pada tahun 2022. Data dianalisis menggunakan aplikasi <i>Quantum GIS</i> (QGIS) dan disajikan dalam bentuk peta spasial dan grafik statistik. Hasil pemetaan menunjukkan bahwa kasus tertinggi terjadi di Kabupaten Merangin dan Batanghari. Sebaliknya, kasus terendah tercatat di Kabupaten Tanjung Jabung Timur, Tanjung Jabung Barat, dan Kerinci. Faktor-faktor seperti kepadatan penduduk, akses layanan kesehatan, dan kondisi sosial ekonomi memengaruhi persebaran kasus. SIG terbukti efektif dalam visualisasi
---	---	---------------------------------	---

distribusi penyakit, mendukung perencanaan intervensi yang lebih efisien, dan pengambilan keputusan berbasis wilayah oleh pembuat kebijakan.

Berdasarkan penelitian sebelumnya, penerapan teknologi informasi di bidang kesehatan telah terbukti meningkatkan efisiensi dan ketepatan dalam proses diagnosis serta pengelolaan data penyakit. Namun, sistem yang ada masih memiliki keterbatasan, khususnya dalam hal deteksi dini dan visualisasi sebaran kasus yang dibuat pada satu sistem. Oleh karena itu, beberapa penelitian mengembangkan sistem pakar dan Sistem Informasi Geografis (SIG) yang mampu mendeteksi Tuberkulosis (TBC) secara lebih akurat serta menyajikan informasi secara spasial. Inovasi ini mencakup penggunaan metode *Naive Bayes* untuk klasifikasi, serta pemanfaatan SIG untuk pemetaan wilayah terdampak. Dengan sistem tersebut, proses diagnosis dan pemantauan TBC menjadi lebih cepat, informatif, dan mampu mendukung pengambilan keputusan yang tepat bagi tenaga medis maupun instansi kesehatan.

2.2 Dasar Teori

Dasar teori berisi kumpulan konsep, definisi, dan prinsip ilmiah yang menjadi landasan dalam pelaksanaan penelitian. Teori-teori ini digunakan untuk memperkuat argumen dan mendukung proses perancangan sistem. Pemilihan teori yang tepat membantu dalam memahami permasalahan, menentukan pendekatan yang sesuai, serta menjelaskan keputusan teknis yang diambil selama pengembangan sistem.

2.2.1 Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) adalah sistem berbasis komputer yang dirancang untuk menggabungkan data spasial (seperti peta) dengan informasi tambahan yang terkait (data atribut). SIG memungkinkan pengguna untuk mengumpulkan, mengelola, memproses, menganalisis, dan menampilkan data

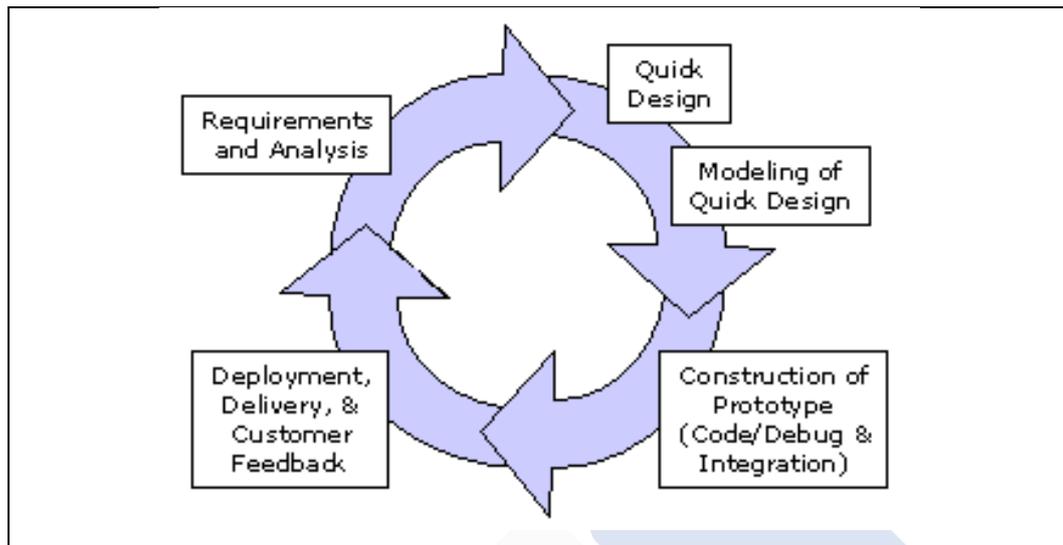
geografis secara visual. Sistem ini sangat membantu dalam kegiatan perencanaan, pengambilan keputusan, dan penelitian terhadap berbagai permasalahan yang berkaitan dengan lokasi atau wilayah tertentu. Dalam kehidupan sehari-hari, SIG digunakan untuk mempermudah pencarian lokasi, karena menyajikan informasi dalam bentuk visual seperti peta digital, sehingga memudahkan pengguna memahami letak dan kondisi suatu tempat [9].

2.2.2 Sistem Pakar

Sistem pakar merupakan salah satu cabang dari kecerdasan buatan yang dirancang untuk menyelesaikan masalah dengan cara meniru pemikiran seorang ahli di bidang tertentu. Sistem ini memanfaatkan pengetahuan khusus yang dimasukkan ke dalam program untuk membantu pengambilan keputusan yang kompleks. Dalam praktiknya, sistem pakar banyak diterapkan di berbagai bidang, salah satunya di dunia medis, khususnya untuk membantu proses diagnosis penyakit [10].

2.2.3 Metode *Prototype*

Dalam mengembangkan sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang, metode penelitian yang digunakan adalah metode *prototype*. Metode *prototype* merupakan pendekatan dalam pengembangan sistem yang diawali dengan pembuatan model awal (*prototype*) untuk diuji dan dievaluasi sebelum sistem versi final dikembangkan. Metode ini sangat membantu karena memungkinkan pengembang menemukan kelemahan sejak dini dan melakukan perbaikan berdasarkan masukan dari pengguna. Dengan cara ini, proses pengembangan menjadi lebih efisien dan hasil akhirnya pun bisa lebih sesuai dengan kebutuhan. Melalui tahapan ini, pengembang dapat menguji apakah fungsi-fungsi utama sistem berjalan dengan baik, menilai desain antarmuka, serta mengumpulkan *feedback* langsung dari calon pengguna. Masukan tersebut sangat berharga untuk meningkatkan kualitas dan performa sistem secara keseluruhan. Oleh karena itu, metode *prototype* berperan penting dalam menciptakan sistem yang lebih matang, sekaligus mengurangi risiko kesalahan saat sistem benar-benar diterapkan [11].



Gambar 2. 1 Metode *Prototype*

Tahapan-tahapan dalam metode *prototype* yang ditampilkan pada Gambar 2.1 mencakup beberapa langkah berikut:

- ***Requirement and Analysis***: Pada tahapan awal ini, dilaksanakan dengan menghimpun maupun menganalisa kebutuhan sistem berdasarkan perspektif pengguna akhir. Tujuannya adalah guna mengetahui apa yang sebenarnya diperlukan pengguna.
- ***Quick Design***: Setelah kebutuhan dianalisis, langkah selanjutnya adalah membuat desain awal secara cepat sebagai gambaran kasar dari sistem yang akan dikembangkan.
- ***Modeling of Quick Design***: Desain awal tersebut kemudian dimodelkan lebih lanjut ke dalam bentuk perancangan sistem yang lebih terstruktur, menggunakan model atau metode desain tertentu.
- ***Construction of Prototype (Code/Debug & Integration)***: Tahap ini merupakan proses pembuatan *prototype* berdasarkan desain yang telah dimodelkan. Meliputi kegiatan *coding*, pengujian awal, serta integrasi komponen agar *prototype* dapat dijalankan.
- ***Deployment, Delivery, & Customer Feedback***: *Prototype* yang telah selesai kemudian diserahkan kepada pengguna untuk diuji dan dievaluasi. Pengguna dapat memberikan masukan atau permintaan revisi jika *prototype* belum sesuai kebutuhan. Jika sudah sesuai, maka *prototype* dianggap siap untuk

digunakan atau dilanjutkan ke tahap pengembangan akhir. Bila revisi diperlukan, proses akan kembali ke siklus awal hingga menghasilkan sistem yang benar-benar sesuai harapan [12].

2.2.4 *Unified Modeling Language (UML)*

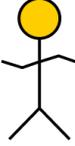
UML merujuk pada metode pemodelan visual yang diadopsi dalam proses perancangan maupun pengembangan perangkat lunak berbasis objek. UML berfungsi sebagai bahasa standar yang memungkinkan para pengembang guna memberikan gambaran struktur, perilaku, maupun interaksi sistem dalam bentuk diagram yang mudah dipahami. UML telah distandarisasi dan sering dianggap sebagai "*blueprint*" atau cetak biru dari perangkat lunak yang akan dibangun. Melalui pemodelan UML, berbagai aspek sistem dapat dijelaskan secara visual, mulai dari alur bisnis, struktur kelas, hingga cara sistem berinteraksi dengan pengguna atau komponen lain [13].

Pemodelan ini sangat membantu dalam menjembatani komunikasi antara tim teknis dan non-teknis, serta memastikan bahwasanya desain sistem sesuai dengan kebutuhan yang telah ditentukan. Dalam praktiknya, UML menyediakan berbagai jenis diagram yang dapat digunakan sesuai dengan kebutuhan pengembangan sistem, seperti *use case diagram* dan *activity diagram*. Yang semuanya memiliki peran penting dalam proses dokumentasi dan perancangan sistem [13]. Ada beberapa diagram UML yang umumnya dipergunakan dalam pengembangan sistem, yakni:

a. *Use Case Diagram*

Use case diagram merupakan gambaran abstrak dari interaksi antara sistem dan aktor yang terlibat, seperti *admin* atau pengguna. Karena itu, penting untuk menentukan bentuk abstraksi yang sesuai dengan kebutuhan sistem. Diagram ini menunjukkan bagaimana sistem digunakan melalui skenario interaksi yang umum terjadi antara aktor dan sistem [13]. *Use case diagram* tersusun dengan teratur maupun menggunakan berbagai simbol sebagai berikut dijelaskan pada Tabel 2.2 berikut:

Tabel 2. 2 Simbol *Use Case Diagram*

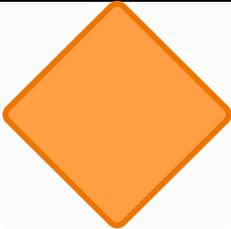
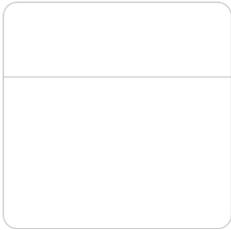
Simbol	Keterangan
	<i>Actor</i> : Individu atau pihak yang merepresentasikan sebuah entitas dalam suatu sistem.
	<i>Use Case</i> : memberikan gambaran interaksi antara pengguna (aktor) dan sistem, serta menjelaskan fungsi atau pelayanan yang dapat disediakan oleh sistem bergantung pada permintaan dan preferensi pengguna.
	<i>Include</i> : Mengindikasikan bahwasanya suatu <i>use case</i> mempunyai fungsi turunan atau sebagai bagian dari fungsionalitas yang dimiliki <i>use case</i> lain.
	<i>Extend</i> : Merepresentasikan bahwasanya suatu <i>use case</i> mempunyai peranannya sebagai pelengkap fitur dari <i>use case</i> yang lain.
	<i>Association</i> : Penghubung diantara <i>actor</i> maupun <i>use case</i> .

b. *Activity Diagram*

Dalam *activity diagram*, setiap simbol mewakili jenis aktivitas tertentu. Proses biasanya dimulai dari sisi kiri sebagai *input* (sumber daya), lalu menunjukkan hasil atau *output* di sisi kanan. Diagram ini menjelaskan alur kerja (*workflow*) secara menyeluruh — mulai dari titik awal, aktivitas-aktivitas yang terjadi, urutannya, hingga proses berakhir. Dengan begitu,

activity diagram membantu menggambarkan bagaimana suatu sistem informasi berjalan secara fungsional [13]. Diagram aktivitas disusun secara sistematis dengan memanfaatkan sejumlah simbol yang mempunyai makna tertentu dan dijelaskan pada Tabel 2.3 sebagai berikut:

Tabel 2. 3 Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Initial State</i> : menunjukkan dimulainya suatu aktivitas.
	<i>Final State</i> : menunjukkan diakhirnya suatu aktivitas.
	<i>Activity</i> : aktivitas atau pekerjaan yang dilakukan pada sistem.
	<i>Decision</i> : menunjukkan sebuah pilihan kondisi dimana ada kemungkinan perbedaan transisi.
	<i>Line Connector</i> : berfungsi sebagai garis penghubung yang menunjukkan alur atau hubungan antara satu simbol dengan simbol lainnya dalam sebuah diagram.
	<i>Swimlane</i> : digunakan untuk menunjukkan pihak atau peran yang bertanggung jawab dalam menjalankan suatu aktivitas dalam proses bisnis atau alur kerja.

2.2.5 Metode *Naive Bayes*

Naive Bayes merupakan salah satu metode dalam teknik klasifikasi yang didasarkan pada pendekatan probabilistik dan statistik. Metode ini bekerja dengan

cara memprediksi kemungkinan suatu kejadian di masa depan berdasarkan pola atau data yang pernah terjadi sebelumnya. Dalam konteks pengambilan keputusan, *Naive Bayes* menghitung probabilitas dari setiap kemungkinan berdasarkan kondisi yang tersedia, sehingga keputusan yang dihasilkan bersifat lebih objektif dan terukur secara matematis.

Keunggulan utama dari metode ini terletak pada kemampuannya dalam bekerja secara efisien, bahkan ketika hanya tersedia jumlah data latih yang terbatas. Hal tersebut menjadikan *Naive Bayes* sebagai salah satu metode yang mudah dipahami, mudah diimplementasikan, serta efektif dalam proses pelatihan model. Tidak hanya itu, metode ini juga terbukti memberikan tingkat akurasi yang cukup tinggi dalam berbagai studi klasifikasi data.

Secara historis, *Naive Bayes* dikembangkan berdasarkan teori probabilitas bersyarat yang diperkenalkan oleh Thomas Bayes, seorang ilmuwan asal Inggris. Teori tersebut kemudian menjadi dasar dalam pengembangan pendekatan probabilistik yang digunakan hingga saat ini dalam metode *Naive Bayes* [1].

a. Teorema Bayes

Teorema Bayes menyatakan hubungan antara probabilitas *posterior* dan probabilitas *prior* suatu hipotesis. Secara matematis ditulis dalam rumus menurut [6] adalah sebagai berikut:

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

Penjelasan:

- $P(H|X)$ adalah prob. posterior dari hipotesis H berdasarkan data X,
- $P(X|H)$ adalah likelihood data X diberikan hipotesis H,
- $P(H)$ adalah prob. prior dari hipotesis H,
- $P(X)$ adalah prob. evidence atau data X secara keseluruhan,

b. Jenis-Jenis *Naive Bayes*

- *Gaussian Naive Bayes*

Digunakan pada data numerik yang diasumsikan berdistribusi normal (*Gaussian*).

- *Multinomial Naive Bayes*

Digunakan untuk data frekuensi atau count, seperti klasifikasi teks berdasarkan jumlah kata.

- *Bernoulli Naive Bayes*

Digunakan untuk data biner (ya/tidak, 0/1).

2.2.6 *Laplace Smoothing*

Laplace Smoothing, yang juga dikenal sebagai *add-one smoothing*, merupakan salah satu metode pemulusan (*smoothing*) tertua yang digunakan dalam bidang statistik dan pembelajaran mesin. Teknik ini digunakan untuk mengatasi masalah probabilitas nol, yaitu kondisi di mana suatu fitur atau token tidak muncul dalam data pelatihan, sehingga menghasilkan nilai probabilitas nol. Nilai nol ini dapat menyebabkan kesalahan dalam proses perhitungan probabilitas gabungan, khususnya pada algoritma seperti *Naive Bayes*, yang sangat bergantung pada perkalian antar probabilitas. Konsep dasar dari *Laplace Smoothing* adalah dengan menambahkan nilai 1 pada setiap frekuensi kemunculan fitur (token), sehingga tidak ada probabilitas yang bernilai nol. Penambahan ini menjamin bahwa semua kemungkinan tetap memiliki nilai probabilitas yang kecil namun tidak nol, sehingga perhitungan klasifikasi tetap berjalan dengan baik dan tidak terdistorsi oleh satu fitur yang tidak muncul [6].

Dalam penelitian ini, metode *Laplace Smoothing* diterapkan untuk menghindari nilai probabilitas nol pada perhitungan dalam algoritma *Naive Bayes*. Dengan menerapkan teknik ini, hasil klasifikasi menjadi lebih akurat dan stabil karena seluruh fitur memiliki kontribusi dalam proses perhitungan, meskipun frekuensi kemunculannya rendah atau tidak ada sama sekali dalam data pelatihan. Perhitungan dengan metode *Laplace Smoothing* dapat dijelaskan melalui rumus menurut [6] adalah sebagai berikut:

$$P(Wi|class) = \frac{freq(Wi, class) + 1}{N_{class} + V_{class}}$$

Penjelasan dari rumus di atas adalah sebagai berikut:

$P(Wi|class)$ = Probabilitas kata W_i pada kelas tertentu.

$freq(Wi, class)$ = Jumlah kemunculan kata W_i pada kelas tersebut.

N_{class} = Jumlah total kata dalam kelas tersebut.

V_{class} = Jumlah kemungkinan nilai kata pada kelas tersebut.

2.2.7 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah tahapan pengujian yang dilakukan langsung oleh pengguna akhir (*end user*) yang akan menggunakan sistem tersebut dalam aktivitas sebenarnya. Tujuan utama dari UAT adalah untuk memastikan bahwa sistem yang dikembangkan sudah berjalan sesuai dengan kebutuhan dan harapan pengguna, bukan sekadar memenuhi spesifikasi teknis yang telah dirancang di awal.

Pengujian ini biasanya dilakukan pada tahap akhir dari proses pengembangan perangkat lunak, sebelum sistem resmi dirilis atau diluncurkan ke publik. Oleh karena itu, UAT sering dianggap sebagai salah satu proses pengujian yang paling krusial dan menentukan. Hasil dari UAT menjadi bukti bahwa sistem tidak hanya layak dari sisi teknis, tetapi juga benar-benar siap digunakan oleh pengguna akhir. UAT juga dikenal dengan beberapa istilah lain seperti *beta testing*, pengujian aplikasi, atau pengujian pengguna akhir, dan menjadi langkah penting sebelum perangkat lunak secara resmi dipasarkan atau digunakan secara luas [14].

Berikut pada tabel 2.4 dibawah ini sebagai implementasi dari kriteria skor [15].

Tabel 2. 4 Kriteria Skor UAT

Jawaban	Bobot
“Sangat Setuju”	“5”
“Setuju”	“4”

Jawaban	Bobot
“Netral”	“3”
“Tidak Setuju”	“2”
“Sangat Tidak Setuju”	“1”

Dari nilai tabel di atas, hasil jawaban kuesioner dianalisa dengan mengadopsi rumus menurut [15] adalah sebagai berikut:

$$P = \frac{\sum R}{N} \times 100\%$$

Penjelasan dari rumus di atas adalah sebagai berikut:

P = Nilai Persentase Respon Siswa

$\sum R$ = Jumlah Jawaban Yang Diberikan

N = Jumlah Skor Maksimal

Adapun interpretasi persentasenya ini meliputi:

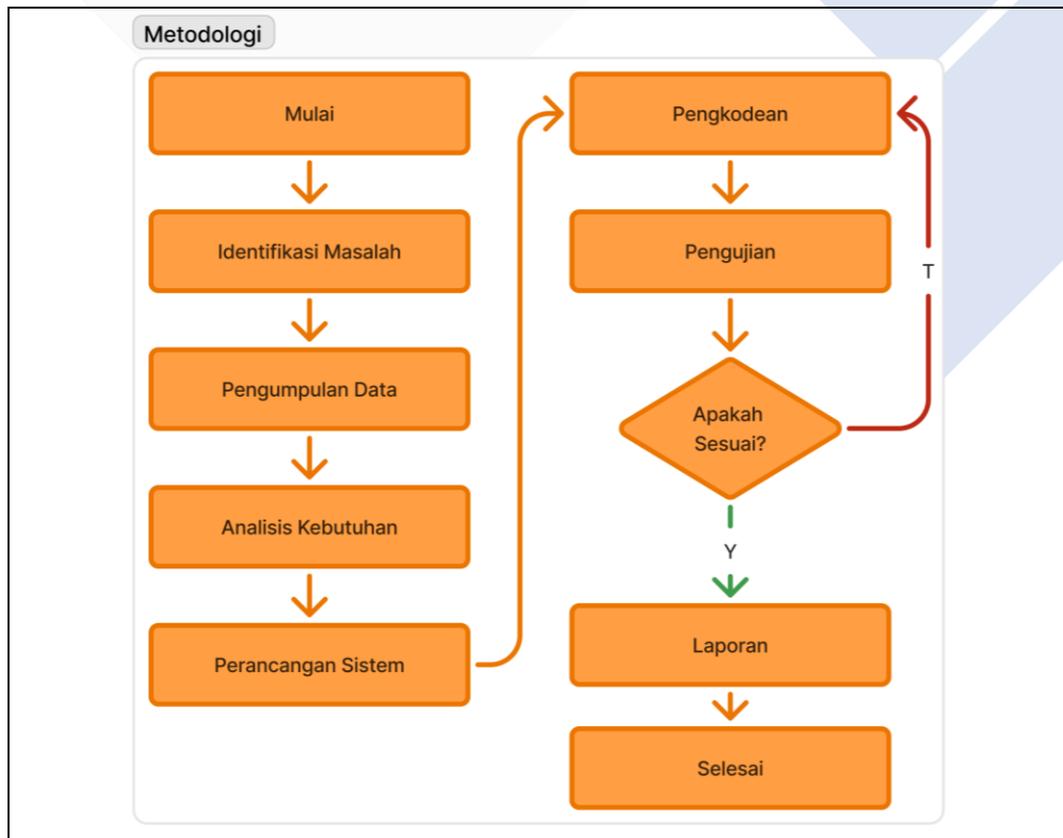
Tabel 2. 5 Persentase Jawaban UAT

Interval Persentase	Nilai	Kualifikasi
“0% - 20%”	“1”	“Sangat Tidak Setuju”
“21% - 40%”	“2”	“Tidak Setuju”
“41% - 60%”	“3”	“Netral”
“61% - 80%”	“4”	“Setuju”
“81% - 100%”	“5”	“Sangat Setuju”

BAB 3 METODE PELAKSANAAN

3.1 Diagram Alir

Diagram alir sebagai diagram yang menggambarkan alur proses atau intruksi dalam sistem secara berurutan. Analisis sistem menggunakannya untuk mendokumentasikan dan menjelaskan rancangan logis kepada *programmer*. Dengan simbol-simbol khusus dan garis penghubung, *flowchart* mempermudah pemahaman alur kerja serta memudahkan penyesuaian jika ada perubahan. Setelah *flowchart* selesai, programmer akan menerjemahkannya ke dalam kode program sesuai bahasa yang disepakati [16].



Gambar 3. 1 Diagram Alir

3.2 Identifikasi Masalah

Identifikasi masalah pada studi kasus adalah sebagai berikut:

1. Pembuatan sistem pemetaan yang bisa memetakan pasien terduga TBC di Kota Pangkalpinang
2. Pembuatan sistem pakar untuk diagnosa dini penyakit TBC
3. Pembuatan sistem yang memudahkan edukasi terkait informasi penyakit TBC

3.3 Pengumpulan Data

Dalam proses penghimpunan data, pendekatan yang dipakai mempunyai tujuannya guna memperoleh informasi yang dibutuhkan dalam perancangan sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di kota pangkalpinang meliputi:

a. Wawancara

Teknik penghimpunan data dengan wawancara dilaksanakan melalui dialog langsung dengan responden guna menggali informasi terkait kebutuhan sistem yang akan dikembangkan. Responden yang dimaksud adalah masyarakat yang pernah terkena gejala TBC.

b. Studi Pustaka

Studi pustaka merupakan analisis pada berbagai pustaka terdahulu yang sesuai dengan proyek akhir penulis. Penulis sudah menganalisis dan mencari berbagai sumber pustaka yang bisa digunakan untuk kebutuhan sistem yang diperlukan oleh penulis.

3.4 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi dan mendefinisikan fitur-fitur yang harus dimiliki oleh sistem agar dapat berjalan sesuai harapan. Proses ini mencakup kebutuhan fungsional, yaitu layanan utama yang harus disediakan oleh sistem, serta kebutuhan nonfungsional seperti keamanan, performa, dan kemudahan penggunaan. Hasil analisis ini menjadi acuan dalam proses desain dan implementasi sistem selanjutnya.

3.4.1 Analisis Kebutuhan Perangkat Lunak

- Laragon
- Visual Studio Code
- MySQL
- *Framework* Laravel
- *Framework* ReactJS
- Google Colab
- FastAPI
- Google Chrome, Microsoft Edge, Firefox, Safari

3.4.2 Analisis Kebutuhan Fungsional

- Autentikasi Pengguna (*login, logout, registrasi*)
- Manajemen data pengguna
- Manajemen data artikel
- Manajemen data gejala
- Manajemen data pasien
- Menampilkan data TBC dalam bentuk grafik, *table*
- Menampilkan pemetaan TBC
- Sistem diagnosa TBC
- Sistem konsultasi

3.4.3 Analisis Kebutuhan Nonfungsional

Kebutuhan nonfungsional berfokus pada aspek teknis yang mendukung agar sistem dapat berjalan dengan baik dan efisien. Salah satu hal penting yang harus diperhatikan adalah keamanan sistem, yang mencakup mekanisme autentikasi pengguna serta enkripsi data untuk mencegah akses dari pihak yang tidak berwenang.

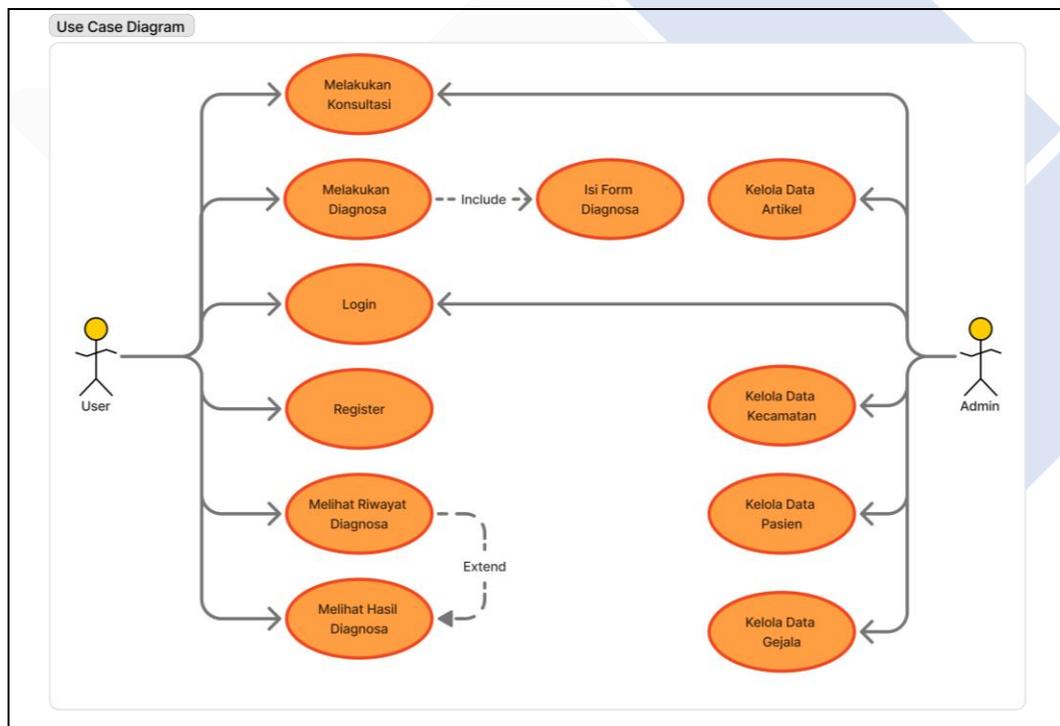
Dari segi kinerja, sistem harus mampu melayani banyak pengguna secara bersamaan tanpa mengalami penurunan performa atau gangguan. Selain itu, sistem perlu dirancang agar responsif terhadap berbagai perangkat, termasuk menyesuaikan tampilan dengan ukuran layar yang berbeda, sehingga tetap nyaman digunakan baik di *desktop* maupun perangkat *mobile*.

Stabilitas sistem juga menjadi poin penting, yaitu kemampuan sistem untuk tetap berjalan dengan baik meskipun digunakan dalam waktu lama atau dalam kondisi beban tinggi. Di samping itu, sistem harus mudah untuk dipelihara dan dikembangkan di kemudian hari, agar dapat menyesuaikan diri dengan kebutuhan baru atau perbaikan yang mungkin diperlukan di masa depan.

3.5 Perancangan Sistem

Perancangan sistem yang akan dibuat dalam pembangunan sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan *use case diagram*, *activity diagram* dan *entity relationship diagram* adalah sebagai berikut:

3.5.1 Use Case Diagram



Gambar 3. 2 Use Case Diagram

Pada *use case diagram* Gambar 3.2 sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang memiliki 2 aktor, yaitu:

- *User*

User dapat melakukan *login* dan juga *register*. *User* dapat melakukan diagnosa dan harus melakukan *isi form* diagnosa setelah itu *user* bisa melihat

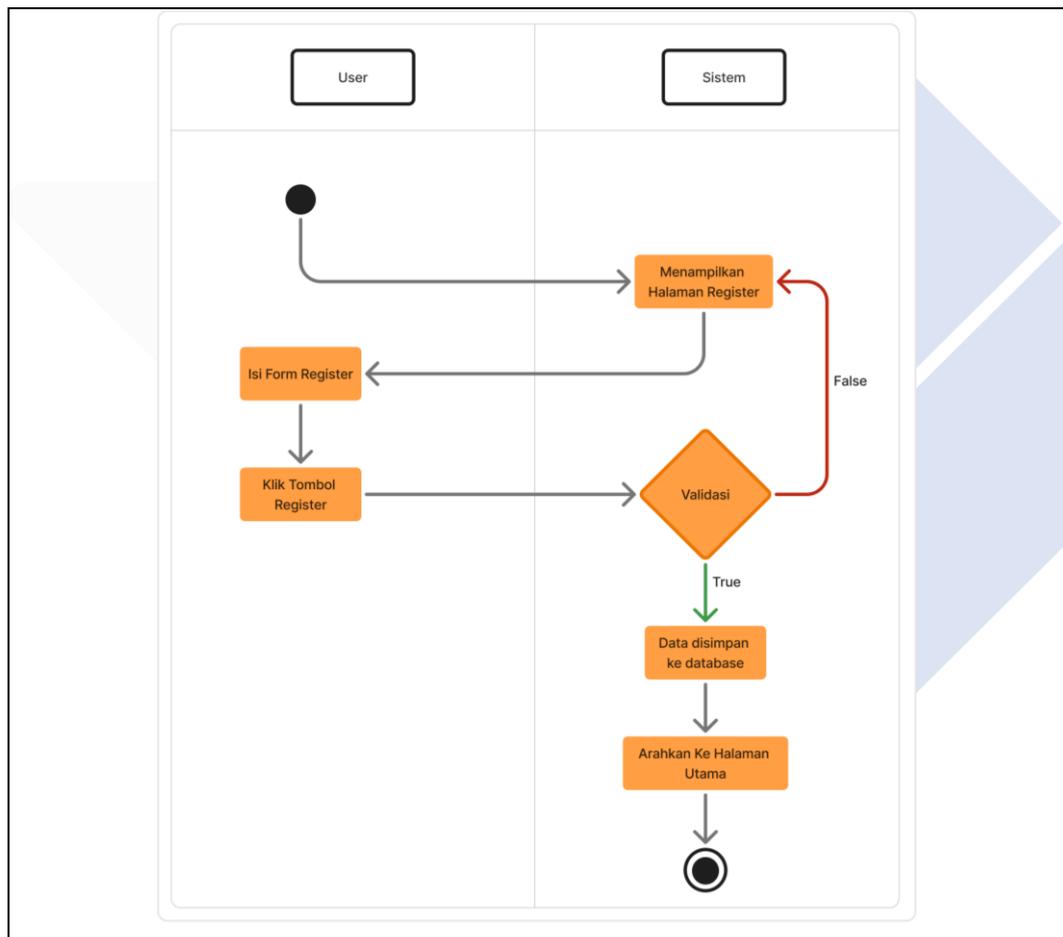
hasil diagnosa yang sudah dilakukan. *User* mampu melakukan konsultasi terkait penyakit TBC kepada *admin*. *User* memiliki riwayat diagnosa penyakit TBC yang sudah dilakukan sebelumnya.

- *Admin*

Admin bisa *login* maupun mengelola berbagai data *website* seperti data artikel, data gejala, data user, data kecamatan dan juga data pasien.

3.5.2 Activity Diagram

- *Activity Diagram Register*

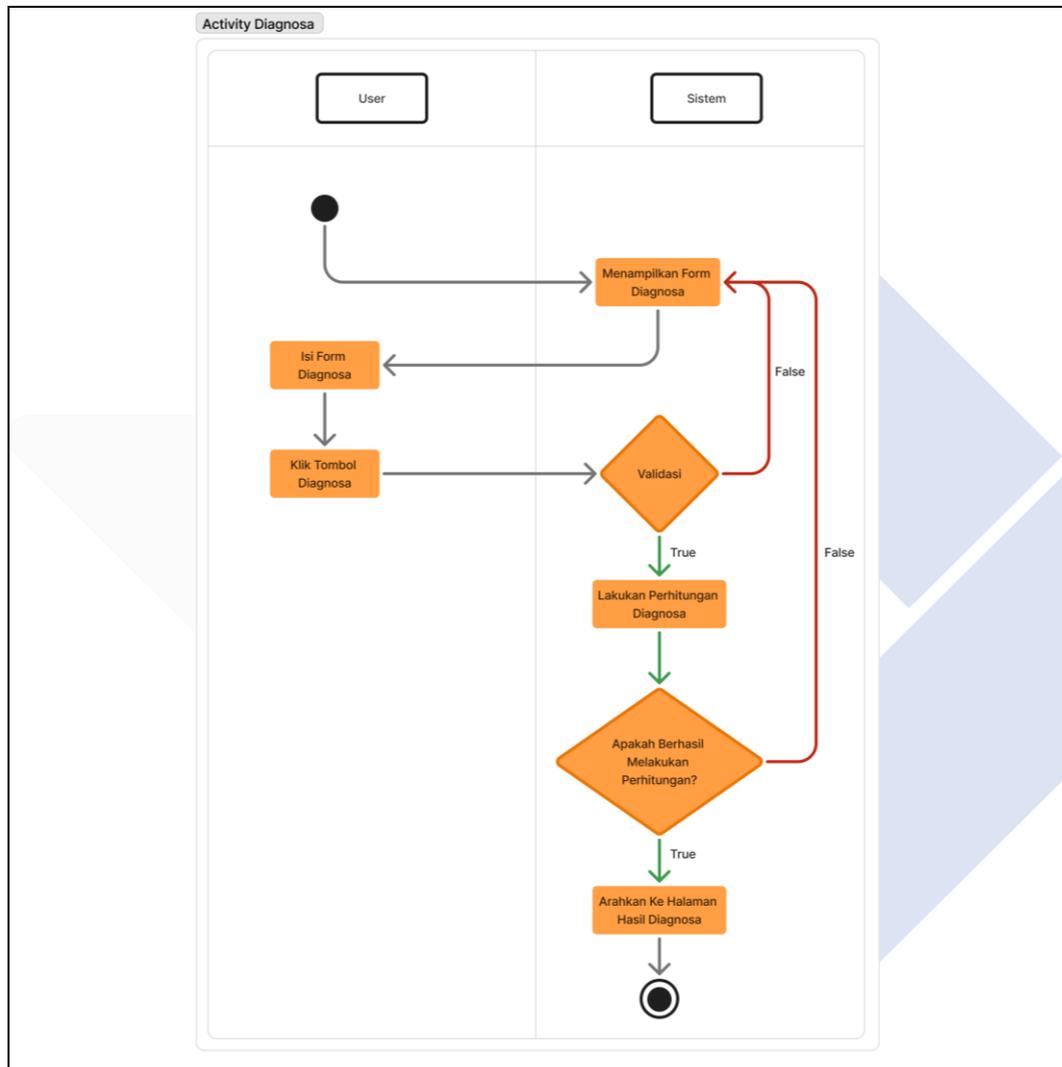


Gambar 3. 3 Activity Diagram Register

Gambar 3.3 di atas adalah *activity diagram* untuk melakukan *register* akun. Pertama sistem akan menampilkan halaman *register*, lalu user akan mengisi *form register* dan menekan tombol *register*, lalu sistem akan melakukan validasi misalnya apakah email sudah terdaftar ataupun belum dan lainnya. Apabila

validasi gagal, maka sistem akan memberikan tampilan kembali halaman *register* dan *user* harus mengisi kembali form *register*, dan jika validasi berhasil maka *user* akan diarahkan ke halaman utama.

- *Activity Diagram Diagnosa*

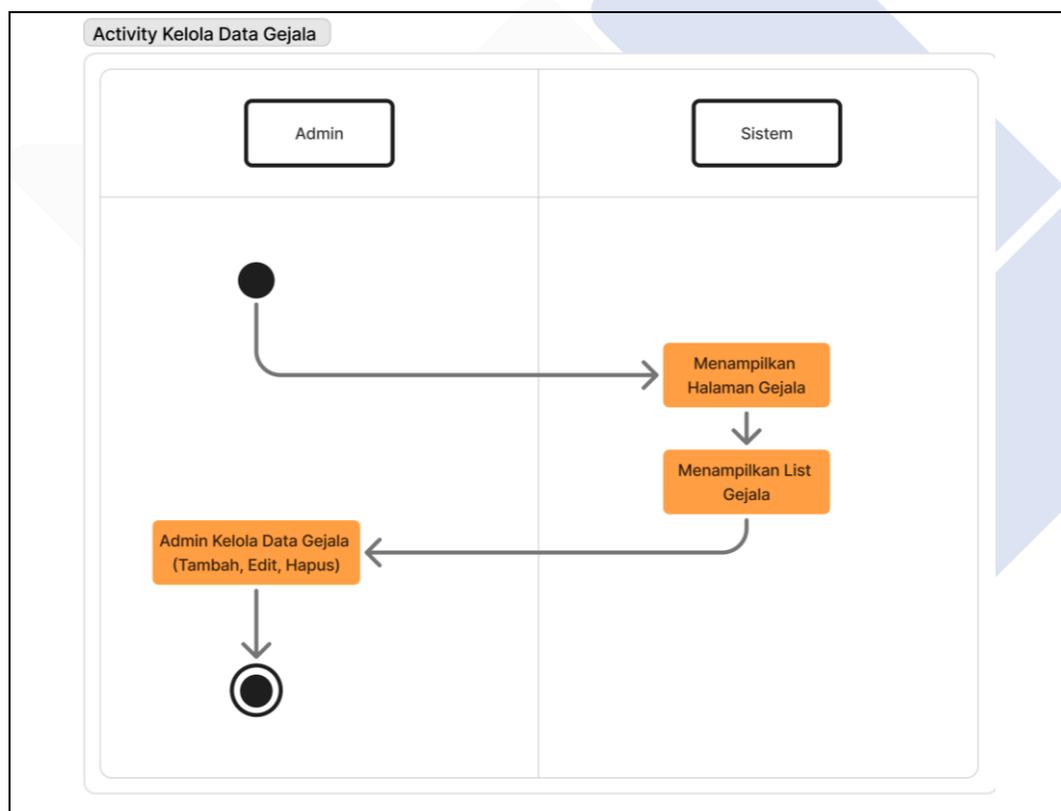


Gambar 3. 4 *Acitivity Diagram Diagnosa*

Gambar 3.4 di atas adalah *activity diagram* untuk proses diagnosa penyakit. Pertama sistem akan menampilkan form diagnosa lalu *user* akan mengisi *form* diagnosa tersebut maupun menekan tombol diagnosa, lalu sistem akan melakukan validasi apakah semua *form* sudah diisi atau sebaliknya. Apabila validasi gagal maka sistem akan memperlihatkan kembali dan *user* harus mengisi kembali *form* diagnosa, dan jika validasi benar maka akan dilakukan perhitungan,

lalu sistem akan mengecek apakah perhitungan berhasil dilakukan. Karena perhitungan menggunakan API yang dibuat sendiri menggunakan *Framework FastAPI* jadi ada kemungkinan perhitungan gagal karena koneksi yang lambat sehingga apabila gagal selanjutnya sistem akan menampilkan kembali *form* diagnosa dan *user* harus mengisi kembali *form* tersebut. Dan jika perhitungan berhasil maka sistem akan diarahkan ke halaman hasil diagnosa untuk *user* bisa melihat hasil diagnosa yang sudah dilakukan perhitungan dari *form* yang sudah diinput sebelumnya.

- *Activity Diagram* Kelola Gejala

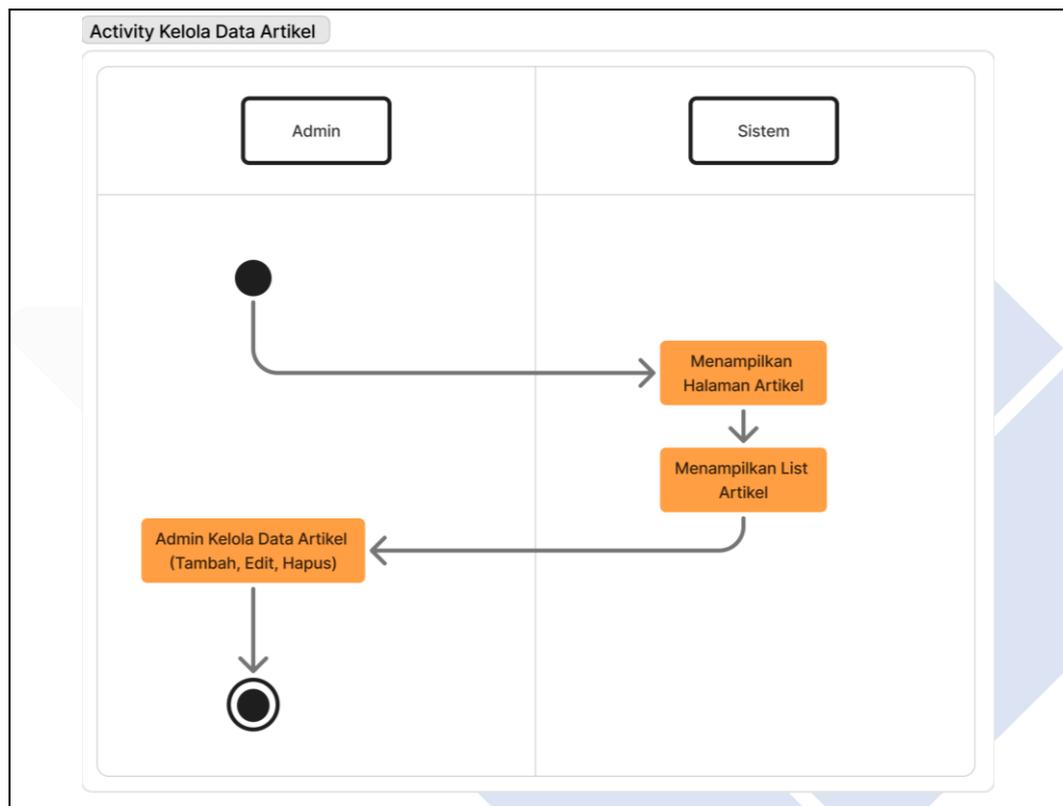


Gambar 3. 5 *Activity Diagram* Kelola Gejala

Pada gambar 3.5 diatas yaitu *activity diagram* untuk melakukan kelola data gejala. Pertama sistem akan memperlihatkan tampilan halaman gejala maupun juga *list* gejala, lalu *admin* akan memilih apakah mau melakukan tambah, *edit* maupun hapus data gejala. Jika memilih tambah data gejala maka *admin* akan mengisi data gejala dan jika validasi berhasil berarti data gejala akan disimpan. Jika *edit* maka *admin* akan menentukan data gejala yang ingin *edit* lalu isi *form*

edit dan jika validasi berhasil maka data gejala akan disimpan. Apabila hapus maka *admin* akan memilih juga data gejala yang ingin dihapus lalu validasi apakah *admin* memilih ingin dihapus ataupun tidaknya. Jika tidak maka akan kembali dan apabila dihapus maka data gejala yang dipilih sebelumnya akan dihapus.

- *Activity Diagram* Kelola Artikel

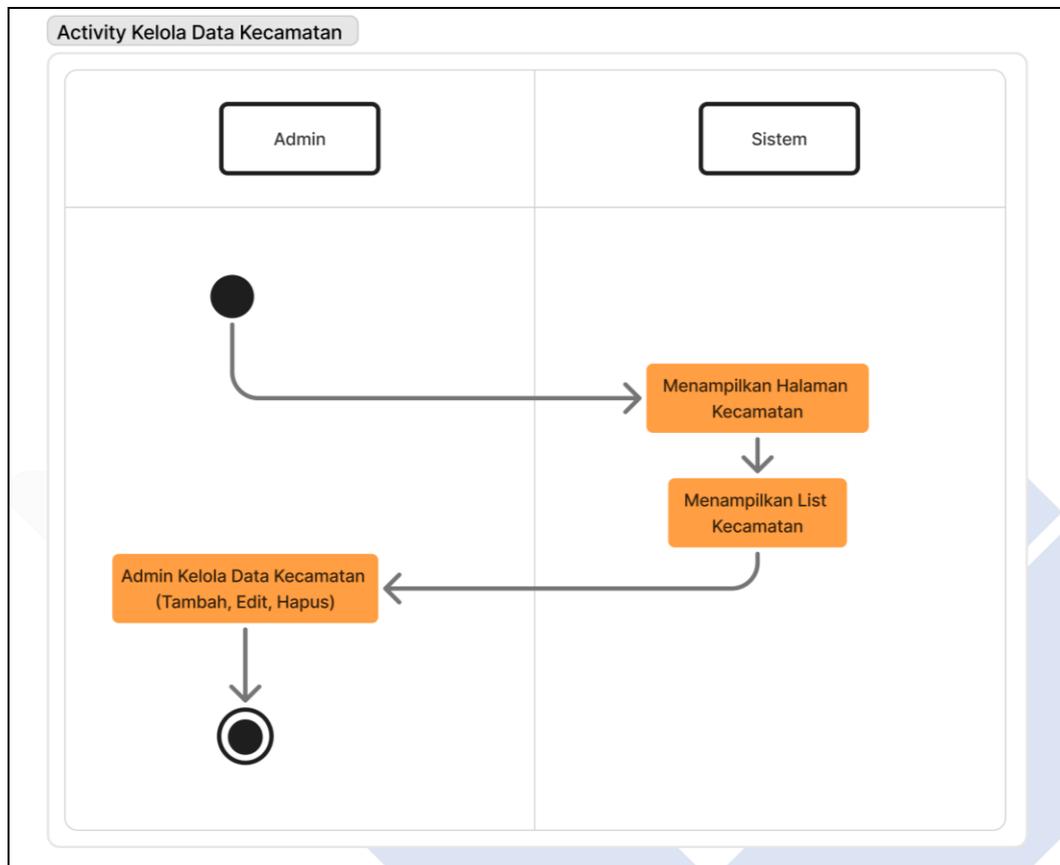


Gambar 3. 6 *Activity Diagram* Kelola Artikel

Pada gambar 3.6 diatas yaitu *activity diagram* untuk melakukan kelola data artikel. Pertama sistem akan memperlihatkan tampilan halaman artikel maupun juga *list* artikel, lalu *admin* akan memilih apakah ingin melakukan tambah, *edit* maupun hapus data artikel. Jika memilih tambah data artikel maka *admin* akan mengisi data artikel dan jika validasi berhasil berarti data artikel akan disimpan. Jika *edit* maka *admin* akan menentukan data artikel yang ingin *edit* lalu isi *form edit* dan jika validasi berhasil maka data artikel akan disimpan. Apabila hapus maka *admin* akan memilih juga data artikel yang ingin dihapus lalu validasi apakah *admin* memilih ingin dihapus ataupun tidaknya. Jika tidak maka

akan kembali dan apabila dihapus maka data artikel yang dipilih sebelumnya akan dihapus.

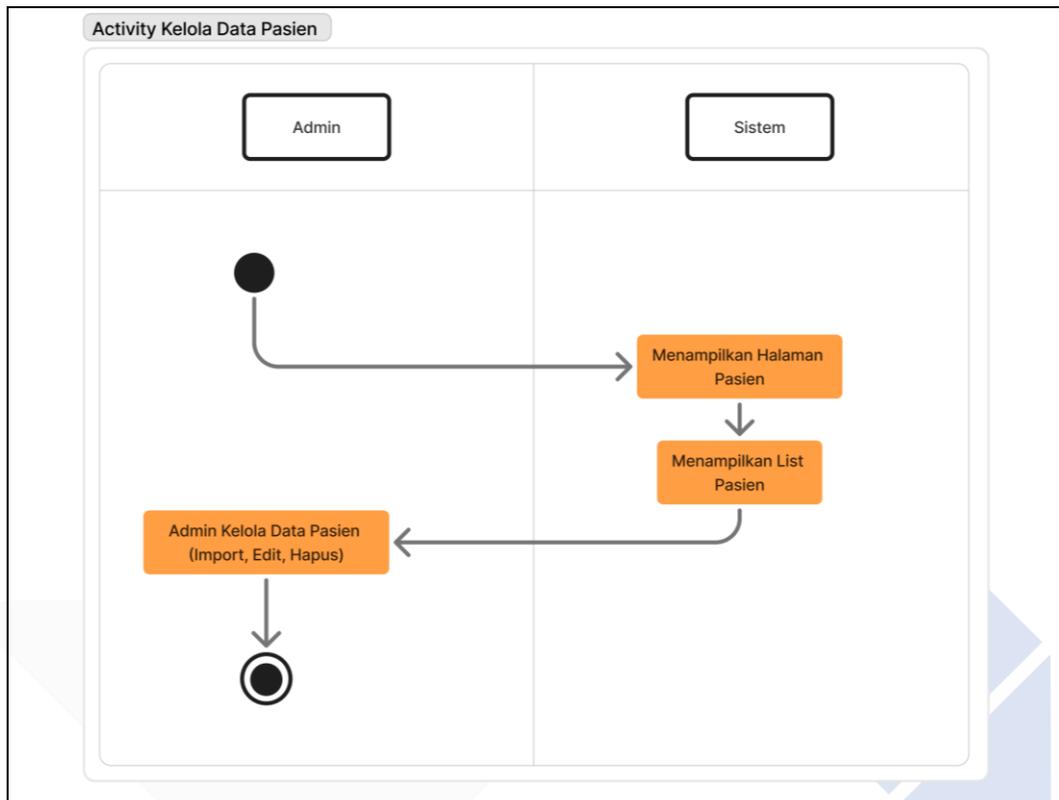
- *Activity Diagram* Kelola Kecamatan



Gambar 3. 7 *Activity Diagram* Kelola Kecamatan

Pada gambar 3.7 diatas yaitu *activity diagram* untuk melakukan kelola data kecamatan. Pertama sistem akan memperlihatkan tampilan halaman kecamatan maupun juga *list* kecamatan, lalu *admin* akan memilih apakah mau melakukan tambah, *edit* maupun hapus data kecamatan. Jika memilih tambah data kecamatan maka *admin* akan mengisi data kecamatan dan jika validasi berhasil berarti data kecamatan akan disimpan. Jika *edit* maka *admin* akan menentukan data kecamatan yang ingin diedit lalu isi *form edit* dan jika validasi berhasil maka data kecamatan akan disimpan. Apabila hapus maka *admin* akan memilih juga data kecamatan yang ingin dihapus lalu validasi apakah *admin* memilih ingin dihapus ataupun tidaknya. Jika tidak maka akan kembali dan apabila dihapus maka data kecamatan yang dipilih sebelumnya akan dihapus.

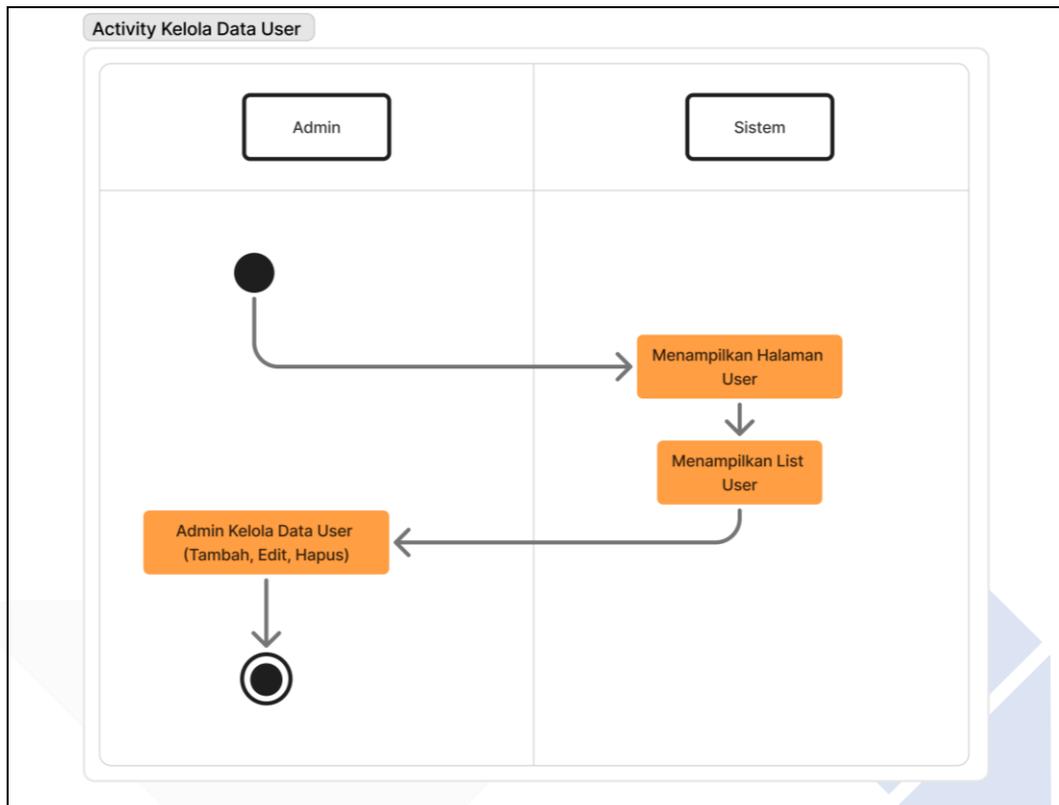
- *Activity Diagram* Kelola Pasien



Gambar 3. 8 *Activity Diagram* Kelola Pasien

Pada gambar 3.8 diatas yaitu *activity diagram* untuk melakukan kelola data pasien. Pertama sistem akan memperlihatkan tampilan halaman pasien maupun juga *list* pasien, lalu *admin* akan memilih apakah mau melakukan *import*, *edit* maupun hapus data pasien. Jika memilih *import* data pasien maka admin akan *download template excel*, setelah *download template* barulah *admin* mengisi data pasien pada *template* tersebut, dan jika sudah semua maka *template* tadi akan diunggah dan data pasien akan ditambah. Jika *edit* maka *admin* akan menentukan data pasien yang ingin diedit lalu isi *form edit* dan jika validasi berhasil maka data pasien akan disimpan. Apabila hapus maka *admin* akan memilih juga data pasien yang ingin dihapus lalu validasi apakah *admin* memilih ingin dihapus ataupun tidaknya. Jika tidak maka akan kembali dan apabila dihapus maka data pasien yang dipilih sebelumnya akan dihapus.

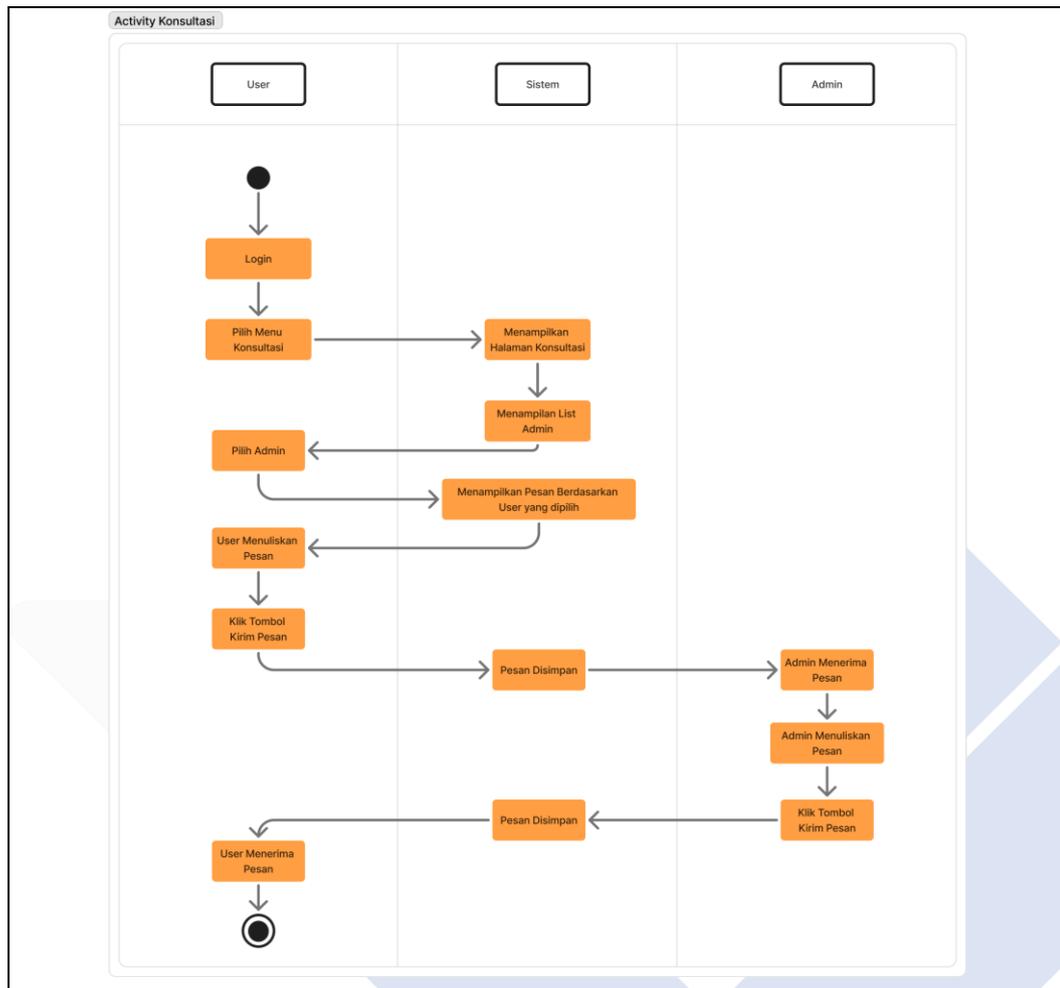
- *Activity Diagram Kelola User*



Gambar 3. 9 *Activity Diagram Kelola User*

Pada gambar 3.9 diatas yaitu *activity diagram* untuk melakukan kelola data user. Pertama sistem akan memperlihatkan tampilan halaman user maupun juga *list user*, lalu *admin* akan memilih apakah mau melakukan tambah, *edit* maupun hapus data *user*. Jika memilih tambah data user maka *admin* akan mengisi data user dan jika validasi berhasil berarti data *user* akan disimpan. Jika *edit* maka *admin* akan menentukan data *user* yang ingin *diedit* lalu isi *form edit* dan jika validasi berhasil maka data *user* akan disimpan. Apabila hapus maka *admin* akan memilih juga data *user* yang ingin dihapus lalu validasi apakah *admin* memilih ingin dihapus ataupun tidaknya. Jika tidak maka akan kembali dan apabila dihapus maka data *user* yang dipilih sebelumnya akan dihapus.

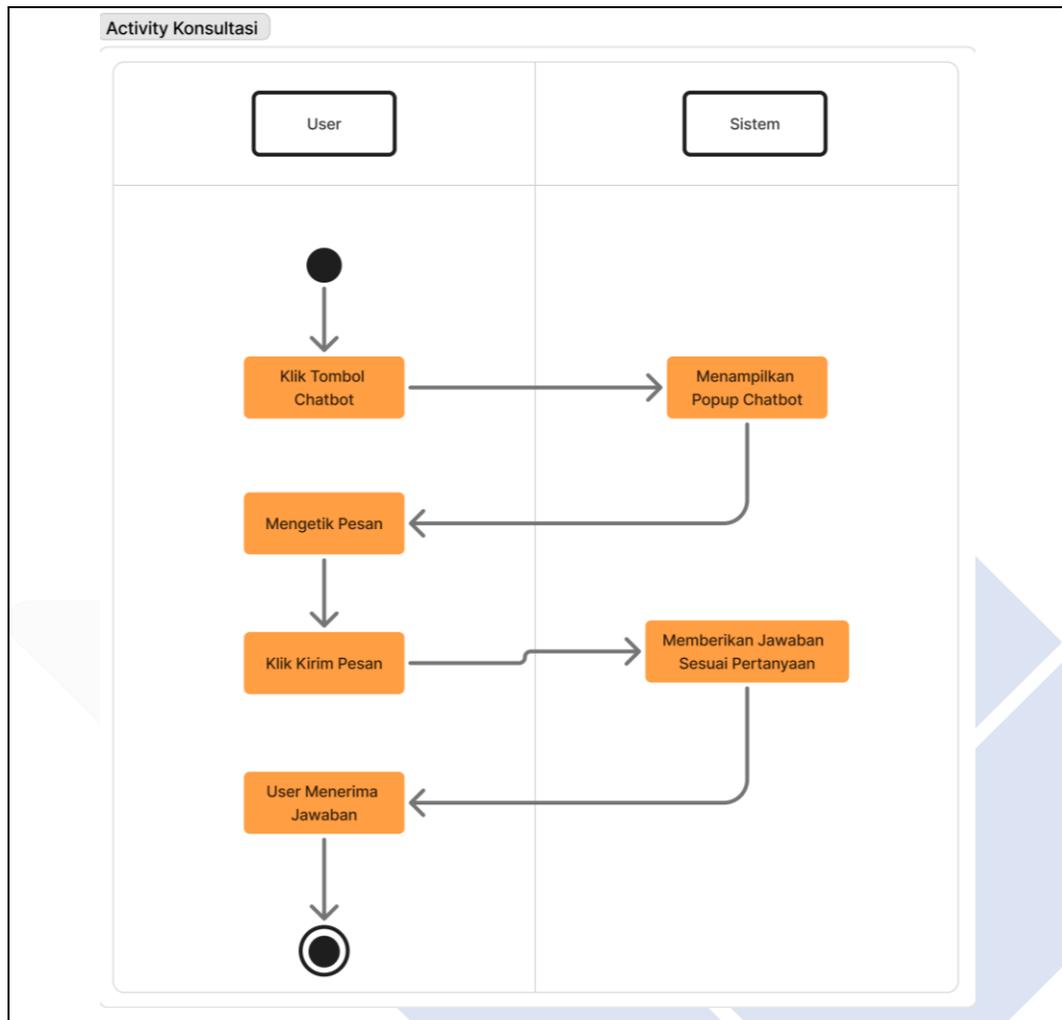
- *Activity Diagram Konsultasi Admin*



Gambar 3. 10 *Activity Diagram Konsultasi Admin*

Pada Gambar 3.10 di atas adalah *activity diagram* untuk *user* melaksanakan konsultasi dengan *admin*. Pertama *user* harus *login* terlebih dahulu dan memilih menu konsultasi, lalu sistem akan menampilkan halaman konsultasi dan juga menampilkan list *admin* yang bisa dilakukan konsultasi. Lalu *user* memilih *admin* tersebut dan sistem akan menampilkan pesan berdasarkan *admin* yang dipilih, lalu *user* bisa menuliskan pesan dan menekan tombol kirim pesan. Lalu pesan akan disimpan dan *admin* akan menerima pesan tersebut serta menuliskan pesan untuk membalas pesan tersebut dan menekan tombol kirim pesan, lalu pesan akan disimpan dan diteruskan kepada *user* tersebut.

- *Activity Diagram Chatbot*

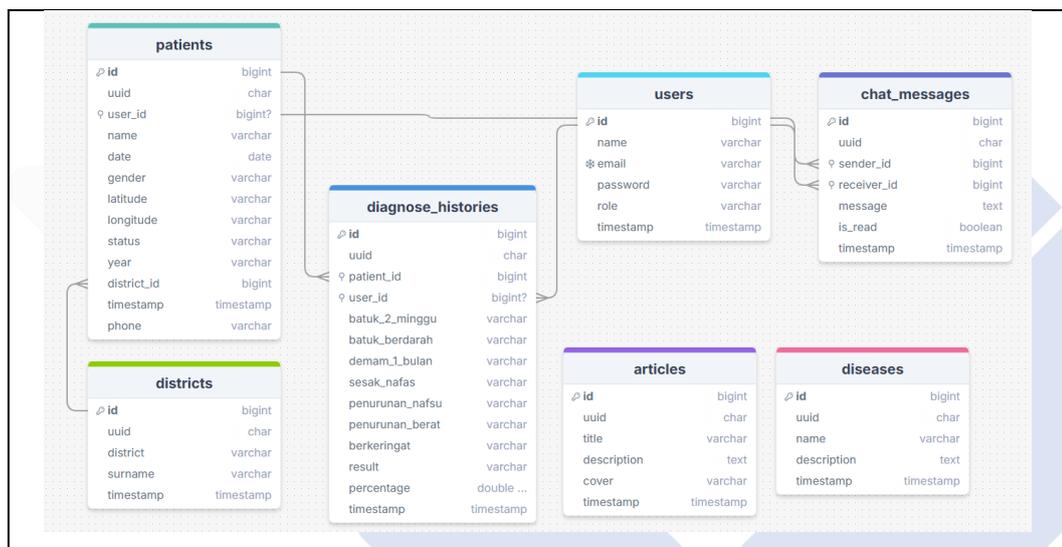


Gambar 3. 11 *Activity Diagram* Konsultasi *Chatbot*

Pada Gambar 3.11 di atas adalah *activity diagram* untuk *user* melakukan konsultasi secara cepat melalui *chatbot*. Pesan akan dikirimkan ke pihak ketiga yaitu *OpenAI* dan *server* akan memberikan jawaban yang relevan dengan apa yang dipertanyakan oleh *user*. *Chatbot* juga sudah bisa memfilter apakah pertanyaan yang dikirimkan oleh *user* terkait dengan TBC atau tidak, jika tidak terkait dengan TBC maka *chatbot* akan memberikan jawaban penolakan secara halus bahwasanya *chatbot* hanya bisa memberikan jawaban terkait pertanyaan tentang TBC saja.

3.5.3 Entity Relationship Diagram (ERD)

ERD atau *Entity Relationship Diagram* adalah diagram yang dipergunakan dalam merancang struktur basis data dengan menunjukkan hubungan antar entitas ataupun objek beserta berbagai atributnya. ERD biasanya digunakan dalam pembuatan *database* relasional, mencakup elemen-elemen seperti nama label, atribut, dan jenis hubungan antar tabel. Dengan merancang ERD secara tepat, maka proses pembuatan *database* juga akan lebih akurat dan sesuai kebutuhan sistem. Diagram ini membantu menggambarkan bagaimana data saling terhubung dan digunakan dalam sistem [17].



Gambar 3. 12 *Entity Relationship Diagram*

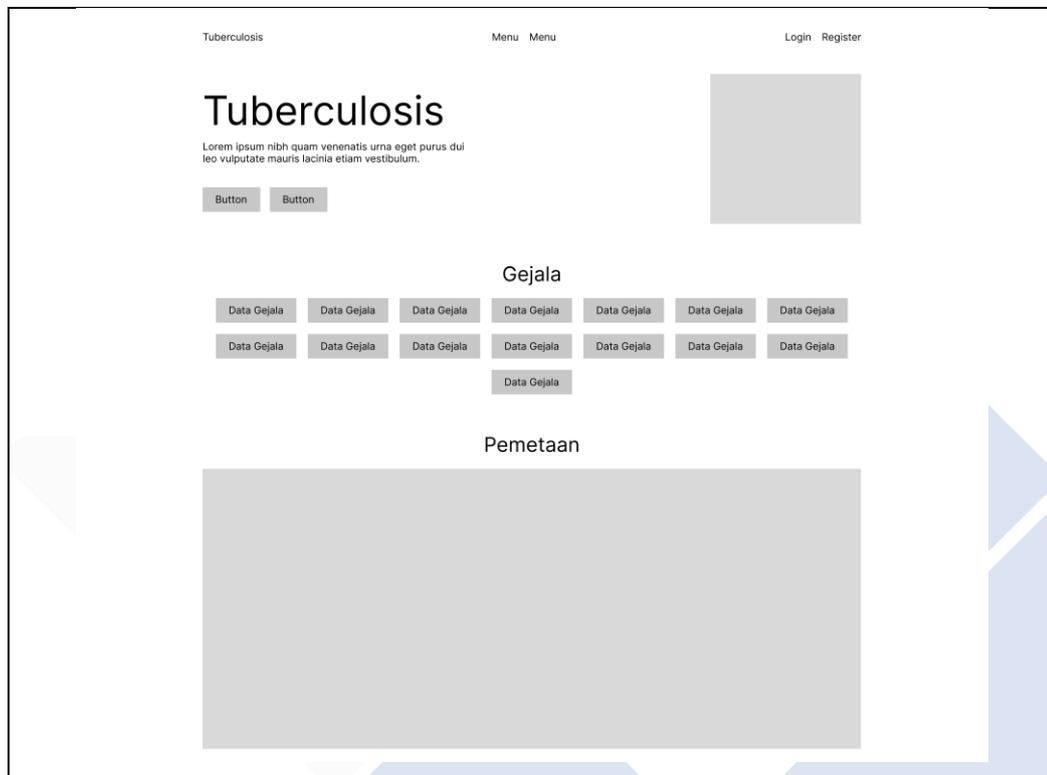
Pada Gambar 3.12 diatas adalah *ERD* dari sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di kota pangkalpinang yaitu ada *users*, *patients*, *diagnose_histories*, *diseases*, *articles*, *districts*, *chat_messages*.

3.6 Prototyping

Dalam pembuatan sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di kota pangkalpinang terdapat rancangan tampilan desain dengan menyajikan visual dan bagaimana sistem dibuat.

- Rancangan Halaman Utama

Pada halaman utama dirancang untuk menampilkan halaman utama *user* dan akan menampilkan gejala, pemetaan juga *form* untuk melakukan diagnosa dini.



Gambar 3. 13 Rancangan Halaman Utama

- Rancangan Halaman *Login*

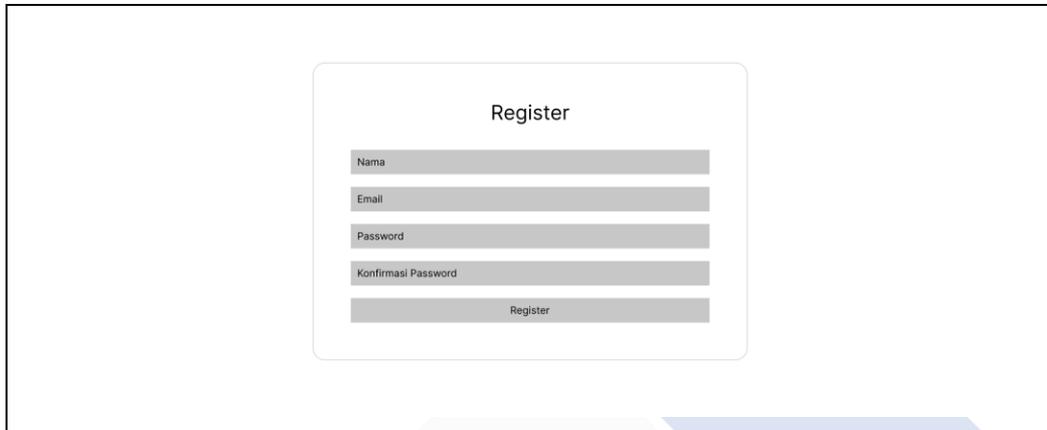
Pada halaman *login* dirancang untuk *user* mengisi *form* untuk melakukan *login* dan masuk ke dalam sistem.



Gambar 3. 14 Rancangan Halaman Login

- Rancangan Halaman *Register*

Pada halaman *register* dirancang untuk *user* mengisi *form* untuk melakukan proses pendaftaran akun.

A wireframe of a registration form titled "Register". The form is centered within a larger rectangular frame. It contains five input fields stacked vertically: "Nama", "Email", "Password", "Konfirmasi Password", and a "Register" button at the bottom.

Gambar 3. 15 Rancangan Halaman Register

- Rancangan Halaman Berita

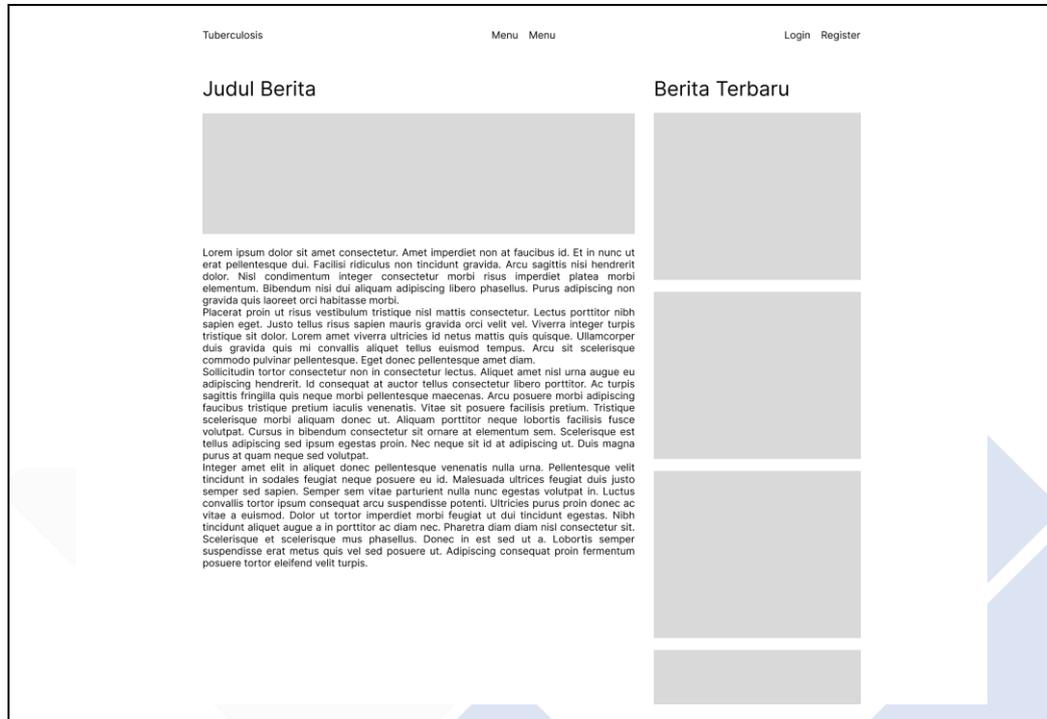
Pada halaman berita dirancang untuk menampilkan semua berita yang nantinya bisa dilihat atau dibaca oleh *user*.



Gambar 3. 16 Rancangan Halaman Berita

- Rancangan Halaman Detail Berita

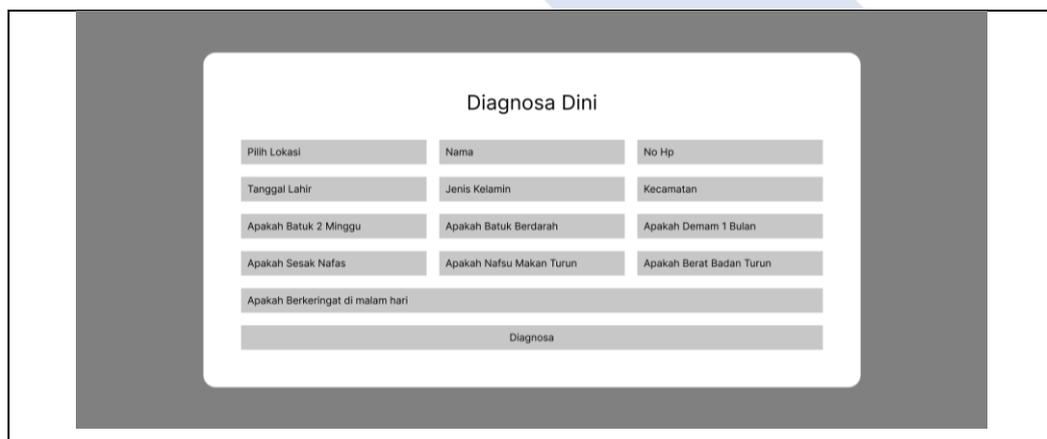
Pada halaman detail berita dirancang untuk menampilkan detail dari berita yang akan dipilih oleh *user*.



Gambar 3. 17 Rancangan Halaman Detail Berita

- Rancangan Diagnosa Dini

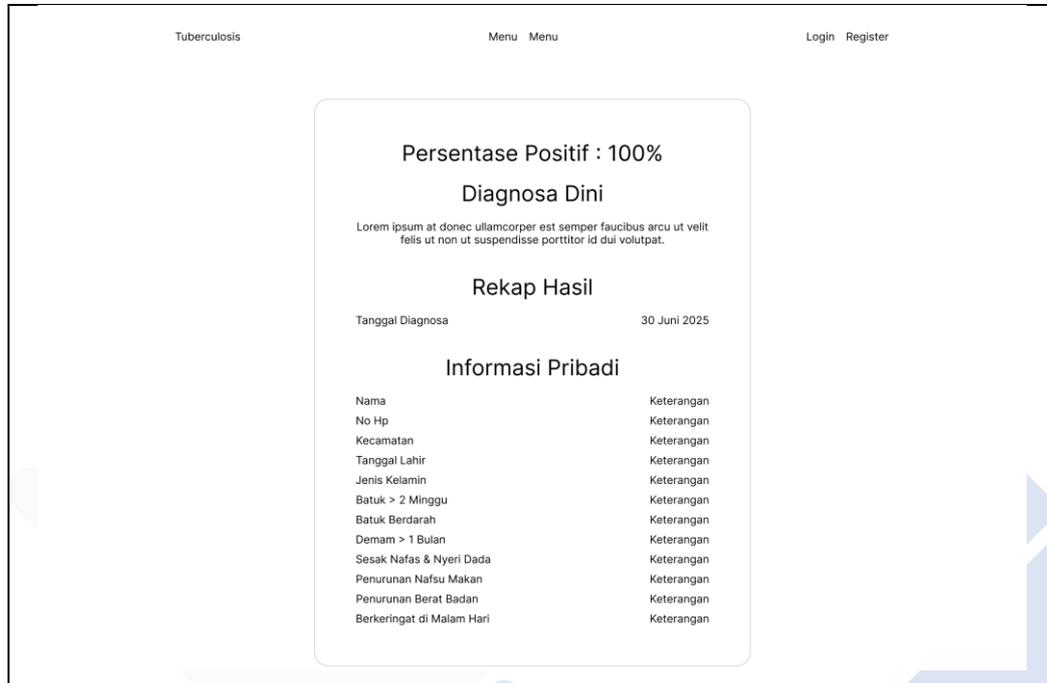
Pada halaman diagnosa dini dirancang agar *user* bisa mengisi *form* yang nantinya akan diproses untuk melakukan diagnosa dini apakah *user* tersebut terduga atau negatif TBC.



Gambar 3. 18 Rancangan Halaman Diagnosa Dini

- Rancangan Hasil Diagnosa Dini

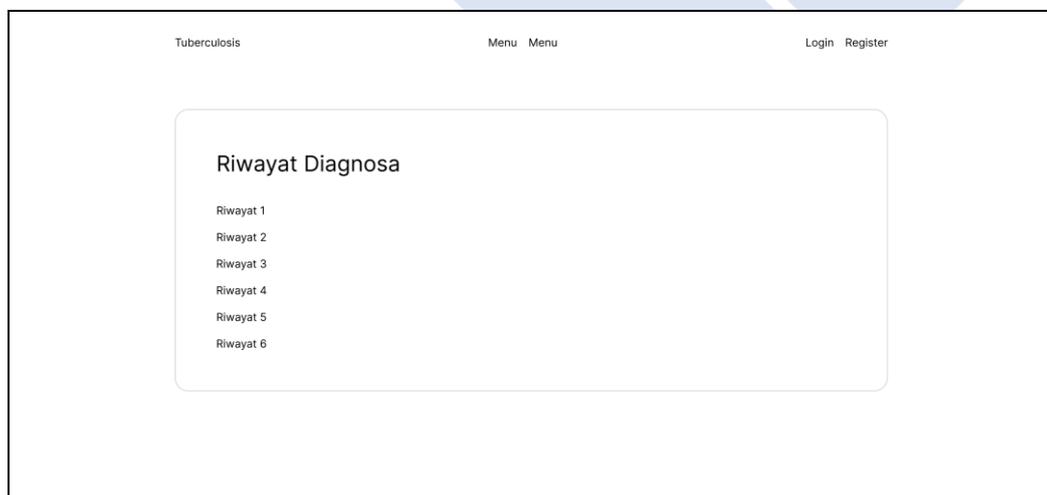
Pada halaman hasil diagnosa dini dirancang untuk *user* bisa melihat hasil dari diagnosa dini yang sudah dilakukan sebelumnya.



Gambar 3. 19 Rancangan Halaman Hasil Diagnosa

- Rancangan Riwayat Diagnosa

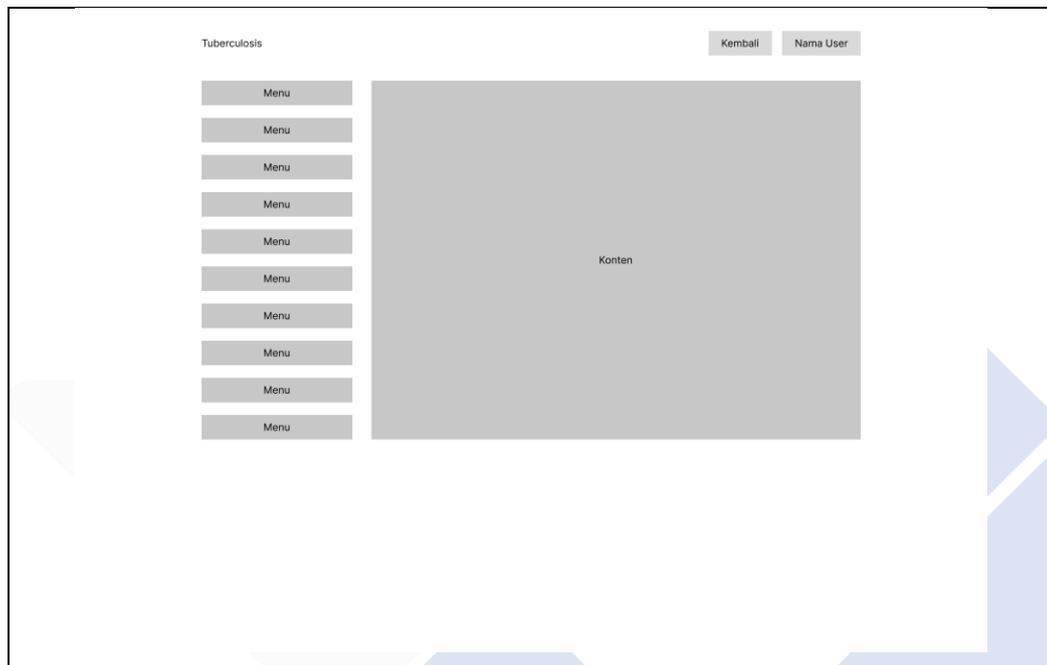
Pada halaman riwayat diagnosa dirancang untuk *user* agar bisa melihat riwayat dari diagnosa yang pernah dilakukan sebelumnya.



Gambar 3. 20 Rancangan Halaman Riwayat Diagnosa

- Rancangan *Dashboard Admin*

Pada halaman *dashboard admin* dirancang untuk *admin* bisa melihat dan mengelola data-data *website*, seperti data gejala, data pasien, data *user*, data berita, data kecamatan, data riwayat diagnosa semua *user*, dan data statistik pasien TBC di Kota Pangkalpinang.



Gambar 3. 21 Rancangan Halaman *Dashboard Admin*

3.7 Pengkodean

Tahap pengkodean adalah proses penerjemahan dari rancangan sistem menjadi bentuk kode program menggunakan bahasa pemrograman yang telah ditentukan. Dalam proyek ini, *backend* dikembangkan menggunakan *Laravel 12* dan *frontend* menggunakan *React* melalui *Inertia.js*.

3.8 Pengujian

Pada tahap berikutnya, sistem akan menjalani proses pengujian untuk memastikan bahwa sistem yang telah dibangun benar-benar dapat diakses dan berfungsi sebagaimana mestinya. Pengujian ini menjadi langkah akhir dalam proses pengembangan sistem. Tujuannya adalah untuk menilai apakah sistem sudah siap digunakan. Saat ini, banyak peneliti yang melakukan pendekatan

pengujian fungsional serta UAT atau *User Acceptance Testing* sebagai metode utama dalam mengevaluasi kesiapan dan kinerja sistem.

3.9 Laporan

Penyusunan laporan mengenai sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang menjadi tahap akhir dalam penyelesaian proyek akhir ini. Laporan ini berisi penjelasan lengkap dan menyeluruh mengenai proses pengembangan sistem, mulai dari perancangan hingga implementasi. Di dalamnya juga dijabarkan alur kerja sistem, khususnya pada pemetaan TBC, diagnosa TBC, dan konsultasi. Dengan demikian, laporan ini disusun sebagai bentuk rangkuman dari seluruh hasil yang telah dicapai selama pelaksanaan proyek akhir.

BAB 4

PEMBAHASAN

4.1 Hasil Analisa dan Perancangan Sistem

Berikut merupakan sistem pada sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang untuk masing-masing *user* antara lain:

a. Admin

- Mengelola Data Pengguna.
- Mengelola Data Pasien.
- Mengelola Data Gejala.
- Mengelola Data Artikel.
- Mengelola Data Kecamatan.
- Melihat Dashboard Statistik Penyakit TBC di Kota Pangkalpinang.
- Melihat Pemetaan Dengan Status Positif & Terduga TBC di Kota Pangkalpinang.

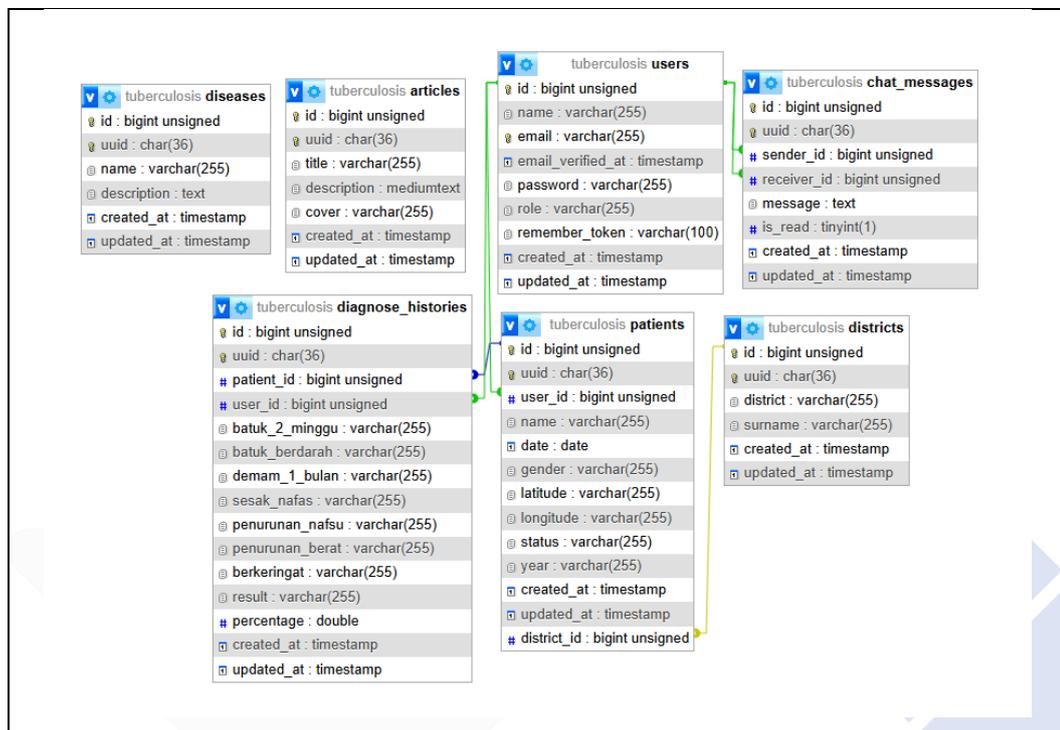
b. User

- Melihat Pemetaan Dengan Status Positif TBC di Kota Pangkalpinang.
- Melakukan Diagnosa Dini Penyakit TBC.
- Memiliki Riwayat Diagnosa.
- Melakukan Konsultasi Dengan Ahli terkait Penyakit TBC.
- Melakukan Konsultasi Cepat dengan Fitur *Chatbot* terkait Penyakit TBC.

4.2 Perancangan Database

Pada tahap desain atau perancangan sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang ini menggunakan *database* berbasis SQL yaitu MySQL.

Perancangan *database* ini mempunyai fungsinya dalam mengelola data yang nantinya akan diintegrasikan dengan *website*.



Gambar 4. 1 Hasil Rancangan *Database*

Lalu berikut adalah detail dari setiap *table* diantara lain:

4.2.1 *Table Users*

Tabel *users* menyimpan data terkait akun pengguna, seperti nama, email, dan peran. Adapun struktur tabel tersebut adalah sebagai berikut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	email_verified_at	timestamp			Yes	NULL			Change Drop More
5	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	role	varchar(255)	utf8mb4_unicode_ci		No	user			Change Drop More
7	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
8	created_at	timestamp			Yes	NULL			Change Drop More
9	updated_at	timestamp			Yes	NULL			Change Drop More

Gambar 4. 2 *Table Users*

4.2.2 *Table Chat Messages*

Tabel *chat_messages* menyimpan data terkait riwayat *chat* pengguna. Adapun struktur tabel tersebut adalah sebagai berikut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 uuid 🔑	char(36)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 sender_id 🔑	bigint		UNSIGNED	No	None			Change Drop More
<input type="checkbox"/>	4 receiver_id 🔑	bigint		UNSIGNED	No	None			Change Drop More
<input type="checkbox"/>	5 message	text	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	6 is_read	tinyint(1)			No	0			Change Drop More
<input type="checkbox"/>	7 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	8 updated_at	timestamp			Yes	NULL			Change Drop More

Gambar 4. 3 Table Chat_Messages

4.2.3 Table Patients

Tabel *patients* menyimpan data terkait data pasien, seperti nama, tanggal lahir, jenis kelamin, dan lainnya. Adapun struktur tabel tersebut adalah sebagai berikut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 uuid 🔑	char(36)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 user_id 🔑	bigint		UNSIGNED	Yes	NULL			Change Drop More
<input type="checkbox"/>	4 name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 date	date			No	None			Change Drop More
<input type="checkbox"/>	6 gender	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	7 latitude	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	8 longitude	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	9 status	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	10 year	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	11 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	12 updated_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	13 district_id 🔑	bigint		UNSIGNED	Yes	NULL			Change Drop More

Gambar 4. 4 Table Patients

4.2.4 Table Districts

Tabel *districts* digunakan untuk menyimpan data kecamatan. Adapun struktur dari *table districts* sebagai berikut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 uuid 🔑	char(36)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 district	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 surname	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	6 updated_at	timestamp			Yes	NULL			Change Drop More

Gambar 4. 5 Table Districts

4.2.5 Table Diagnose_Histories

Tabel *diagnose_histories* digunakan untuk menyimpan data riwayat diagnosa dini dari pasien. Adapun struktur dari *table diagnose_histories* sebagai berikut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2 uuid	char(36)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	3 patient_id	bigint		UNSIGNED	No	None			
<input type="checkbox"/>	4 user_id	bigint		UNSIGNED	Yes	NULL			
<input type="checkbox"/>	5 batuk_2_minggu	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	6 batuk_berdarah	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	7 demam_1_bulan	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	8 sesak_nafas	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	9 penurunan_nafsu	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	10 penurunan_berat	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	11 berkeringat	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	12 result	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	13 percentage	double			No	None			
<input type="checkbox"/>	14 created_at	timestamp			Yes	NULL			
<input type="checkbox"/>	15 updated_at	timestamp			Yes	NULL			

Gambar 4. 6 Table Diagnose_Histories

4.2.6 Table Articles

Tabel *articles* dipergunakan menyimpan data berita. Adapun struktur dari *table articles* yaitu:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2 uuid	char(36)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	3 title	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	4 description	mediumtext	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	5 cover	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	6 created_at	timestamp			Yes	NULL			
<input type="checkbox"/>	7 updated_at	timestamp			Yes	NULL			

Gambar 4. 7 Table Articles

4.2.7 Table Diseases

Tabel *diseases* dipergunakan sebagai penyimpanan data gejala TBC. Adapun struktur dari *table diseases* meliputi:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2 uuid	char(36)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	3 name	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	4 description	text	utf8mb4_unicode_ci		Yes	NULL			
<input type="checkbox"/>	5 created_at	timestamp			Yes	NULL			
<input type="checkbox"/>	6 updated_at	timestamp			Yes	NULL			

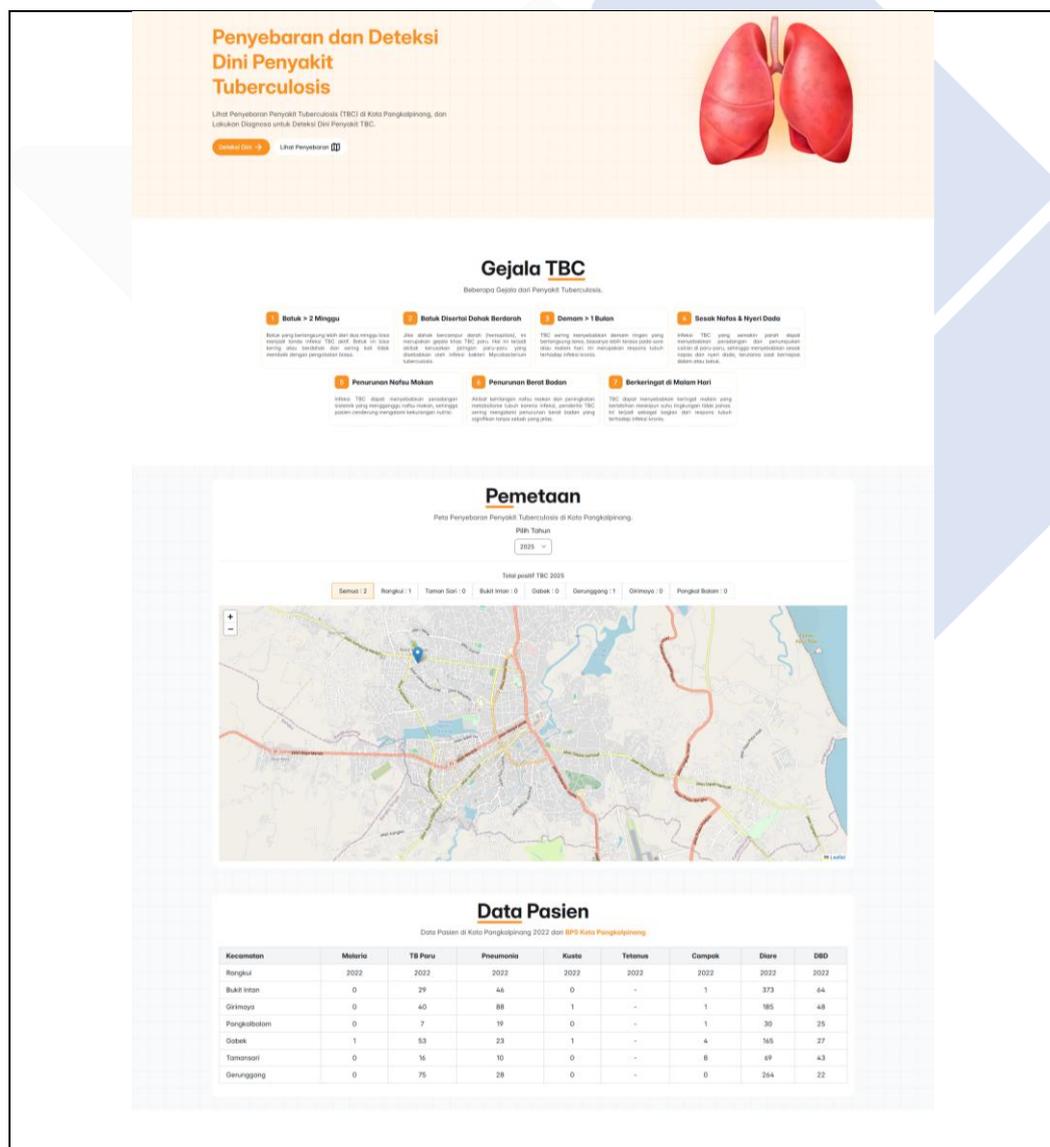
Gambar 4. 8 Table Diseases

4.3 Hasil Tampilan Antarmuka

Hasil tampilan antarmuka menampilkan desain visual dari aplikasi yang telah dikembangkan. Antarmuka dirancang agar mudah digunakan oleh pengguna dan merepresentasikan fungsi sistem secara intuitif. Pada bagian ini ditampilkan beberapa contoh tampilan utama dari aplikasi untuk memberikan gambaran menyeluruh mengenai pengalaman pengguna.

4.3.1 Tampilan Halaman Utama

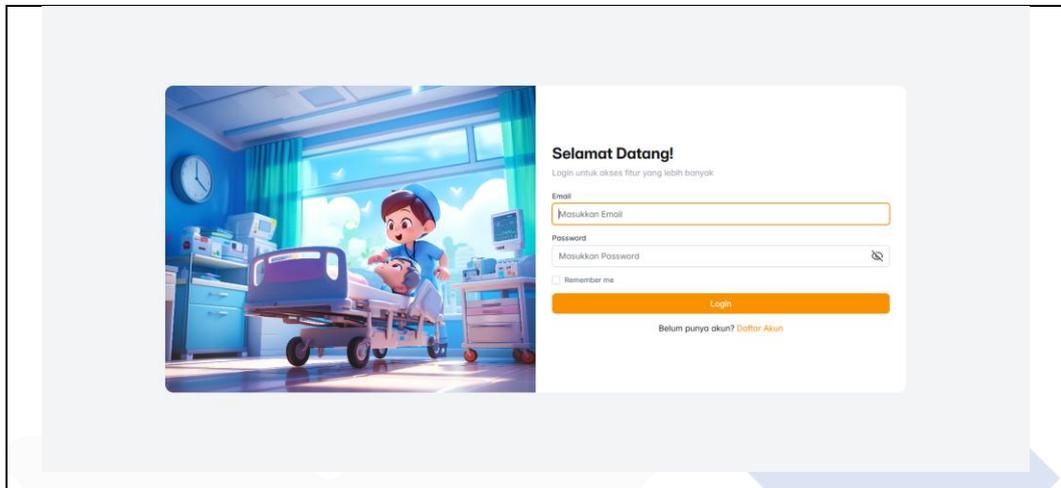
Halaman utama berisikan pemetaan, diagnosa dini, tombol untuk konsultasi, 4 berita terbaru.



Gambar 4. 9 Tampilan Halaman Utama

4.3.2 Tampilan Halaman *Login*

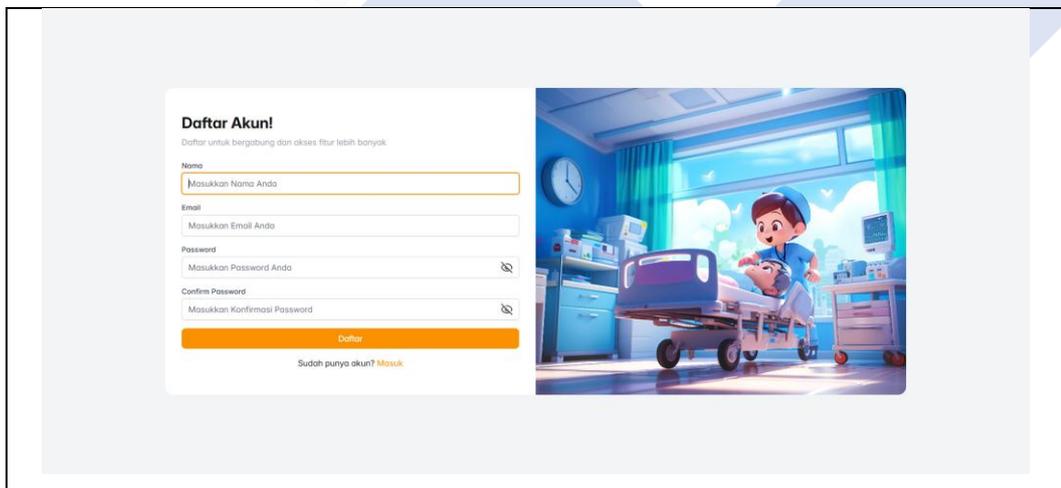
Halaman *login* yakni untuk masuk *login* baik sebagai *admin* dan *user*. *User* agar bisa melakukan konsultasi dengan ahli dan memiliki riwayat diagnosa dini.



Gambar 4. 10 Tampilan Halaman *Login*

4.3.3 Tampilan Halaman *Register*

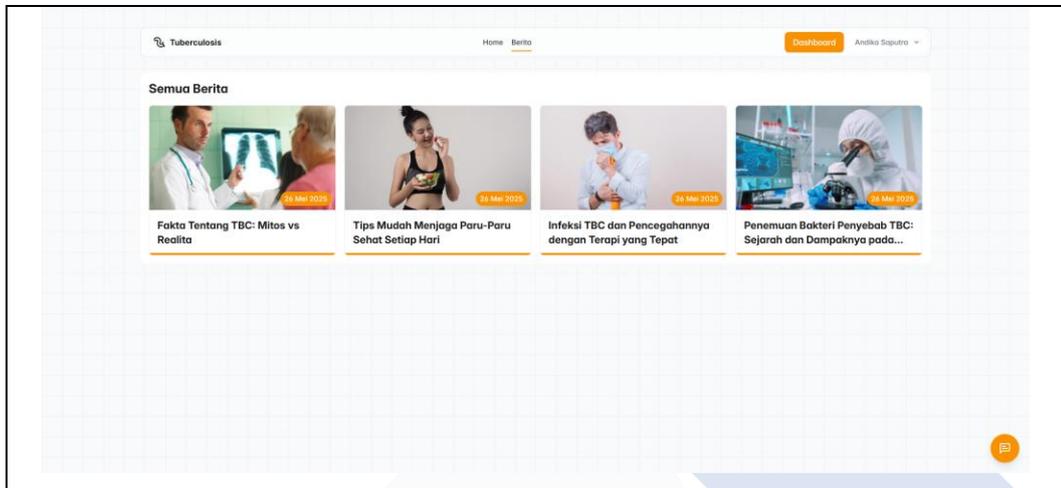
Halaman *register* digunakan untuk mendaftarkan ke dalam sistem sebagai *user* biasa, baru setelah itu pengguna bisa *login* dan menggunakan fitur-fitur yang diharuskan *login*.



Gambar 4. 11 Tampilan Halaman *Register*

4.3.4 Tampilan Halaman Berita

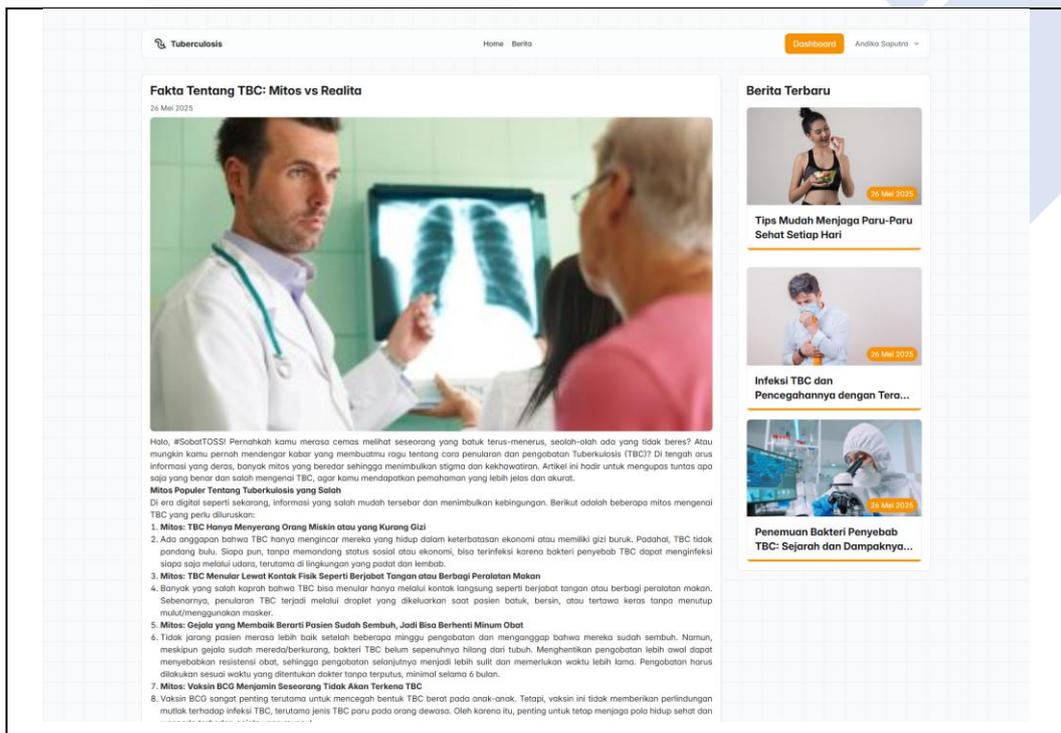
Halaman berita ini berisikan semua data berita yang sudah dibuat oleh *admin*.



Gambar 4. 12 Tampilan Halaman Berita

4.3.5 Tampilan Halaman Detail Berita

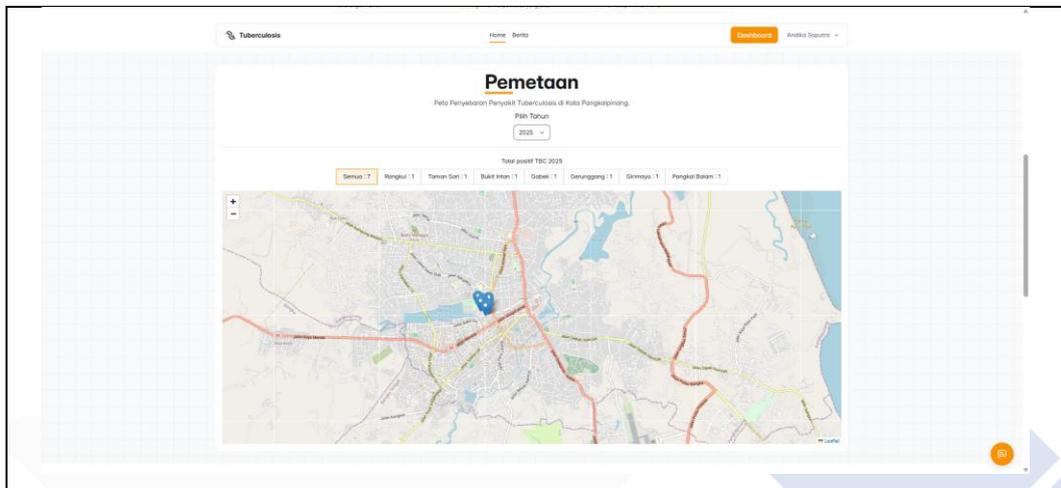
Halaman detail berita adalah detail dari berita yang dipilih oleh pengguna.



Gambar 4. 13 Tampilan Halaman Detail Berita

4.3.6 Tampilan Pemetaan

Pada tampilan pemetaan *user* terdapat peta untuk melihat kasus TBC dengan status positif di Kota Pangkalpinang, pada tampilan pemetaan untuk *user* terdapat *filter* tahun dan juga kecamatan.



Gambar 4. 14 Tampilan Peta Penyebaran

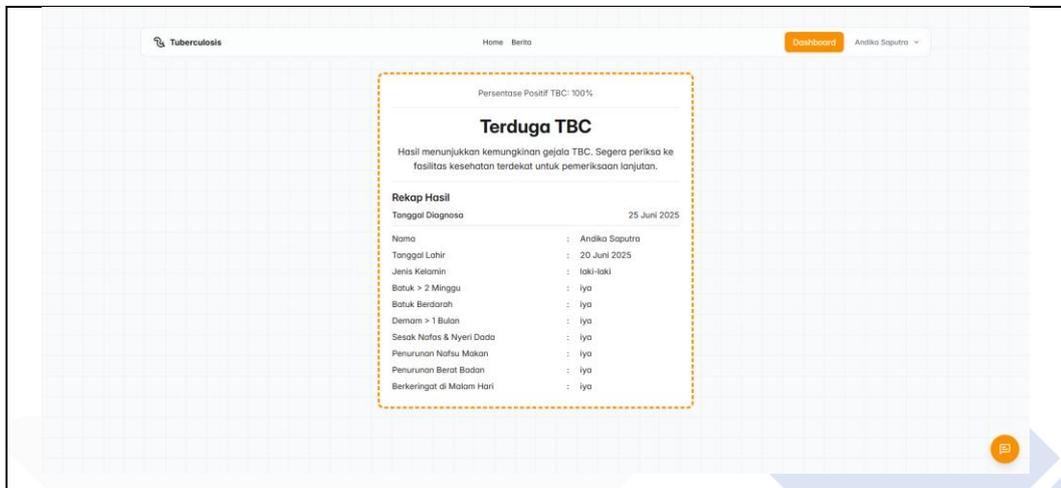
4.3.7 Tampilan Diagnosa Dini

Pada tampilan diagnosa dini digunakan *user* untuk melakukan diagnosa dini penyakit TBC, *user* akan mengisi data yang diperlukan dan sistem akan melakukan perhitungan menggunakan *AI model Naive Bayes* yang sudah dirancang sebelumnya.

Gambar 4. 15 Tampilan *Form* Diagnosa

4.3.8 Tampilan Hasil Diagnosa Dini

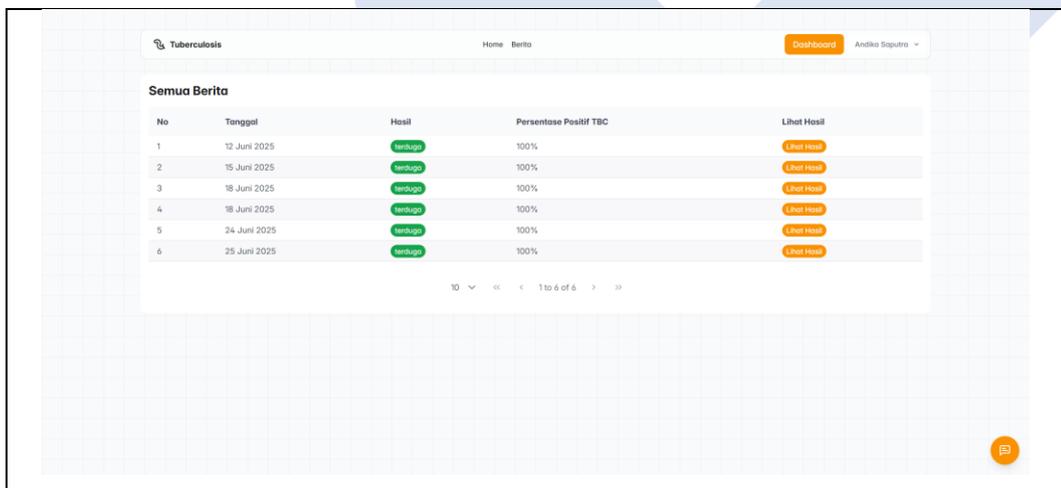
Pada tampilan hasil diagnosa dini digunakan *user* guna melihat perolehan diagnosa yang sudah dilaksanakan diagnosa sebelumnya. Pada hasil terdapat persentase kemungkinan positif TBC.



Gambar 4. 16 Tampilan Hasil Diagnosa

4.3.9 Tampilan Riwayat Diagnosa

Pada tampilan riwayat diagnosa dipergunakan *user* guna melihat riwayat diagnosa yang sudah dilaksanakan oleh *user* tersebut. Riwayat diagnosa hanya bisa diakses jika *user* sudah *login*.



Gambar 4. 17 Tampilan Halaman Riwayat Diagnosa

4.3.10 Tampilan *Chatbot*

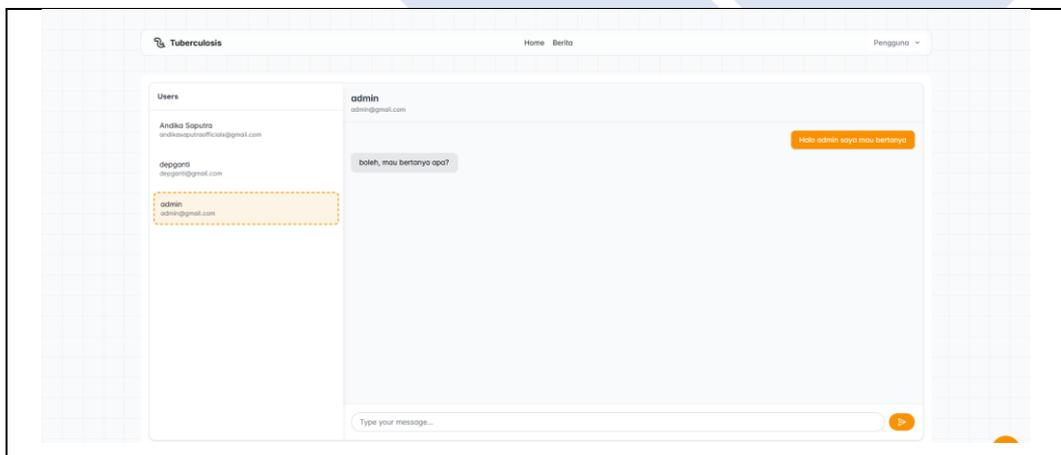
Pada tampilan *chatbot* digunakan *user* untuk melakukan konsultasi secara cepat, dimana *chatbot* terhubung dengan *server OpenAI* dan hanya menjawab pertanyaan seputar TBC. *Chatbot* bisa diakses pada kanan bawah halaman.



Gambar 4. 18 Tampilan *Chatbot*

4.3.11 Tampilan Halaman Konsultasi

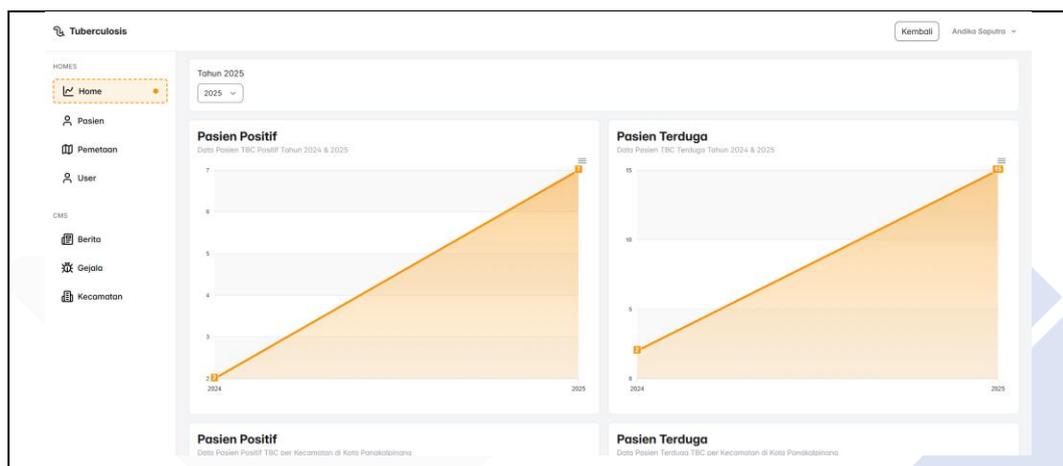
Pada halaman konsultasi dipergunakan *user* dalam berkonsultasi atau bertanya dengan *admin* terkait penyakit TBC. Setelah *user* bertanya, *admin* akan menerima pesan dari *user* tersebut dan *admin* bisa membalas pertanyaan dari *user* tersebut.



Gambar 4. 19 Tampilan Halaman Konsultasi Admin

4.3.12 Tampilan *Dashboard Admin*

Pada tampilan *dashboard admin* digunakan *admin* untuk melihat statistik pasien TBC seperti grafik pasien positif dan terduga tahun yang dipilih dan dibandingkan dengan tahun sebelumnya, grafik pasien positif dan terduga per kecamatan dan dibandingkan dengan tahun sebelumnya, dan total semua pasien dan dibagi ke beberapa kategori yaitu pasien positif, negatif, terduga dan dibandingkan dengan jumlah dari tahun sebelumnya.



Gambar 4. 20 Tampilan Halaman *Dashboard Admin*

4.3.13 Tampilan *Dashboard Kelola Pasien*

Pada tampilan *dashboard* kelola pasien digunakan *admin* untuk mengelola data pasien, *admin* juga bisa *import* data pasien menggunakan *file excel* dimana *admin* bisa mengunduh *template* yang diperlukan.

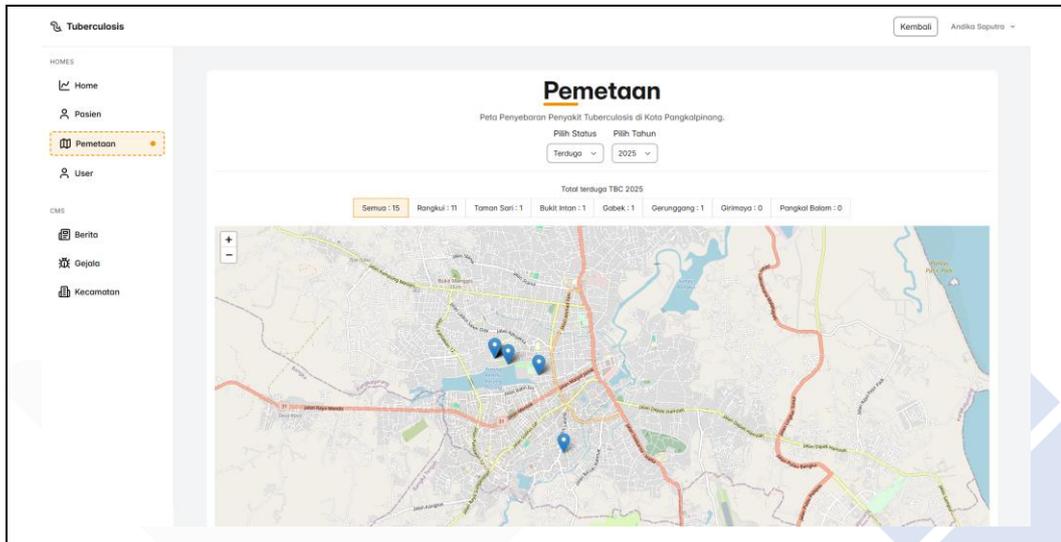
The screenshot shows the 'Data pasien' section of the Tuberculosis Admin Dashboard. It features a sidebar with navigation options: Home, Pasien, Pemetaan, User, Berita, Gejala, and Kecamatan. The main content area displays a table of patient data. The table has columns for No, Nama, Tanggal Lahir, Jenis Kelamin, Kecamatan, Status, Tahun, and Action. There are 5 rows of data, each with an 'Edit' button and a 'Delete' button. A '+ Import Pasien' button is located above the table. The table is filtered by 'Semua Tahun', 'Semua Status', and 'Semua Jenis Kelamin'. A search bar is also present.

No	Nama	Tanggal Lahir	Jenis Kelamin	Kecamatan	Status	Tahun	Action
1	Andika Saputra	7 Juni 2025	laki-laki	Gabek	terduga	2025	Edit Delete
2	Putra Angkasa	16 Mei 2025	laki-laki	Gerunggang	terduga	2024	Edit Delete
3	gmamon	29 Mei 2025	laki-laki	Girimaya	terduga	2023	Edit Delete
4	gmamon	23 Mei 2025	laki-laki	Pangkal Balam	terduga	2024	Edit Delete
5	Putra Angkasa	15 Mei 2025	laki-laki	Rangkul	terduga	2025	Edit Delete

Gambar 4. 21 Tampilan Halaman Kelola Pasien

4.3.14 Tampilan *Dashboard* Pemetaan

Pada tampilan *dashboard* pemetaan digunakan *admin* untuk melihat pemetaan atau penyebaran pasien TBC di Kota Pangkalpinang. Berbeda dengan pemetaan di *user*, pemetaan di *admin* bisa melihat pasien positif dan juga terduga, dimana pada pemetaan *user* hanya melihat pasien positif saja.



Gambar 4. 22 Tampilan Halaman Peta Penyebaran *Admin*

4.3.15 Tampilan *Dashboard* Kelola *User*

Pada tampilan *dashboard* kelola *user* dipergunakan *admin* untuk melakukan pengelolaan pada data *user*.

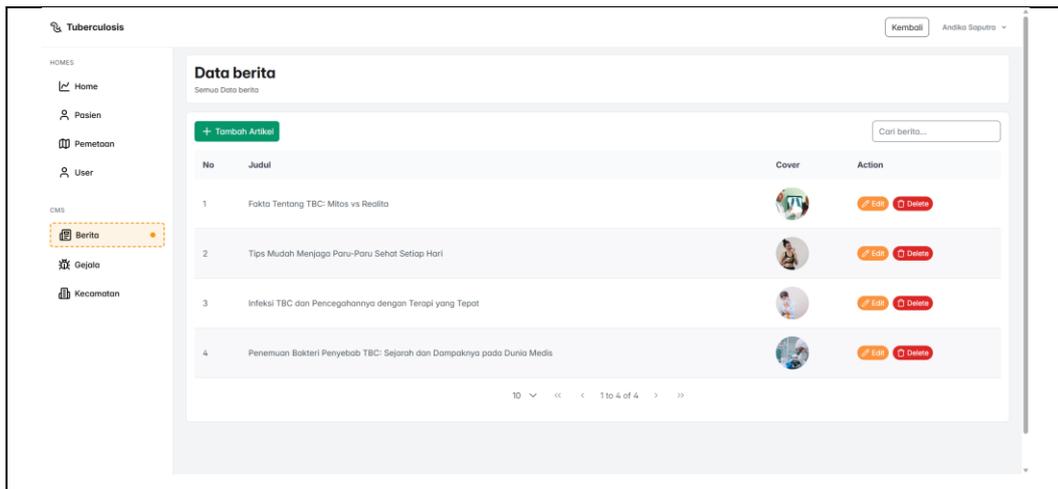
The screenshot shows the 'Data User' management page in the Tuberculosis Admin Dashboard. The page title is 'Data User' and the subtitle is 'Semua Data User'. There is a '+ Tambah User' button and a search bar labeled 'Cari User...'. Below is a table with 5 rows of user data.

No	Nama	Email	Role	Action
1	Andika Saputra	andikasaputraofficials@gmail.com	admin	Edit Delete
2	depganti	depganti@gmail.com	admin	Edit Delete
3	Putra Angkasa	putraangkasa@gmail.com	user	Edit Delete
4	Gita	gita@gmail.com	user	Edit Delete
5	admin	admin@gmail.com	admin	Edit Delete

Gambar 4. 23 Tampilan Halaman Kelola *User*

4.3.16 Tampilan *Dashboard* Kelola Berita

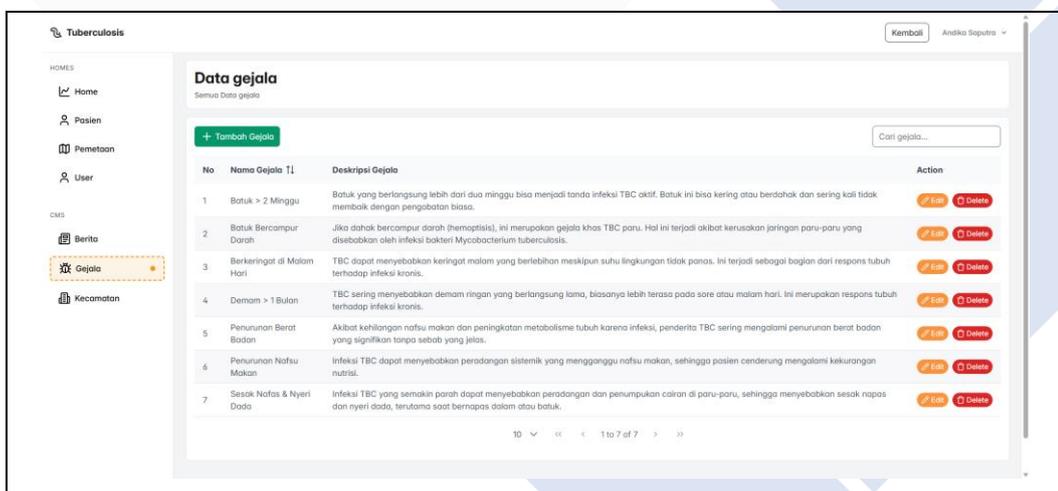
Pada tampilan *dashboard* kelola berita dipergunakan *admin* untuk melakukan pengelolaan pada data berita.



Gambar 4. 24 Tampilan Halaman Kelola Berita

4.3.17 Tampilan *Dashboard* Kelola Gejala

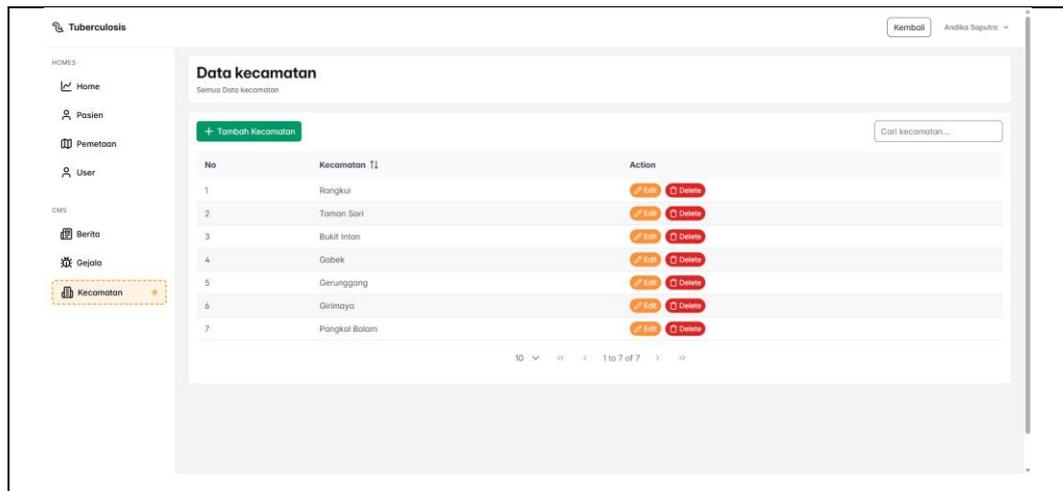
Pada tampilan *dashboard* kelola gejala dipergunakan *admin* untuk melakukan pengelolaan pada data gejala.



Gambar 4. 25 Tampilan Halaman Kelola Gejala

4.3.18 Tampilan *Dashboard* Kelola Kecamatan

Pada tampilan *dashboard* kelola kecamatan dipergunakan *admin* untuk melakukan pengelolaan pada data kecamatan.



Gambar 4. 26 Tampilan Halaman Kelola Kecamatan

4.4 Pengujian Model *Naive Bayes*

Pengujian model *Naive Bayes* dilakukan untuk mengevaluasi kinerja algoritma dalam melakukan prediksi terhadap data yang tersedia. Pada tahap ini, dilakukan dua jenis perhitungan, yaitu perhitungan menggunakan program komputer dan perhitungan manual di atas kertas. Tujuannya adalah untuk memastikan bahwa implementasi algoritma telah berjalan sesuai dengan rumus dan logika dasar metode *Naive Bayes*. Hasil dari kedua pendekatan ini kemudian dibandingkan untuk menjamin keakuratan model serta validitas hasil prediksi yang dihasilkan oleh sistem.

```
data = pd.read_csv('dataset.csv', sep=';')
data = data.drop(columns=['Timestamp', 'Nama'])
data = data.replace({'Iya': 1, 'Tidak': 0})
```

Gambar 4. 27 Tahap *Cleaning Data*

Pada gambar 4.27 dilakukan proses *cleaning* atau pembersihan data, dimana data akan membaca file “dataset.csv” dengan pemisah “;”, lalu kolom *Timestamp* dan Nama akan dibuang karena tidak diperlukan, lalu data akan dikonversi ke dalam bentuk biner yaitu “Iya” menjadi 1 dan “Tidak” menjadi 0.

```
X = data.drop(columns=['result'])
y = data['result']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
```

Gambar 4. 28 Tahap Pembagian Data

Pada gambar 4.28 dilakukan tahap pembagian data, dimana data dipisahkan menjadi fitur (X) dan label (y), lalu data dibagi menjadi 2 bagian yaitu *data training* dan *data testing* dengan rasio pembagian data yaitu 0.3 atau 70% *data training* dan 30% *data testing*.

```
def train_naive_bayes_bernoulli(X, y, alpha=1):
    classes = y.unique()
    model = {}

    for c in classes:
        X_c = X[y == c]
        prior = len(X_c) / len(X)

        probs = {}
        for col in X.columns:
            # Bernoulli NB: P(feature=1 | class) dan P(feature=0 | class) dari frekuensi
            feature_present = (X_c[col] == 1).sum()
            prob_feature_given_class = (feature_present + alpha) / (len(X_c) + 2 * alpha)
            probs[col] = prob_feature_given_class

        model[c] = {
            'prior': prior,
            'probs': probs
        }

    return model
```

Gambar 4. 29 Tahap Melatih Data

Pada gambar 4.29 dilakukan tahap melatih data dengan fungsi *train_naive_bayes_bernoulli* dengan cara menghitung nilai *prior* dan *conditional probability* berdasarkan frekuensi kemunculan setiap fitur pada masing-masing kelas. Nilai probabilitas dihitung dengan metode *Laplace Smoothing* agar menghindari nilai nol.

```

def predict_naive_bayes(model, X):
    predictions = []
    for _, row in X.iterrows():
        class_scores = {}
        for c in model:
            log_prob = np.log(model[c]['prior']) # log prior
            for feature in X.columns:
                prob = model[c]['probs'][feature]
                if row[feature] == 1:
                    log_prob += np.log(prob)
                else:
                    log_prob += np.log(1 - prob)
            class_scores[c] = log_prob
        predicted_class = max(class_scores, key=class_scores.get)
        predictions.append(predicted_class)
    return predictions

```

Gambar 4. 30 Tahap Pengujian Data

Pada gambar 4.30 dilakukan tahap pengujian data dengan fungsi *predict_naive_bayes* dengan cara menghitung nilai *posterior probability* untuk setiap kelas menggunakan logaritma natural dari *prior* dan *likelihood*. Kelas dengan probabilitas tertinggi kemudian dipilih sebagai hasil prediksi.

```

model_manual = train_naive_bayes_bernoulli(X_train, y_train)
y_pred_manual = predict_naive_bayes(model_manual, X_test)

```

Gambar 4. 31 Tahap Pelatihan dan Pengujian Model

Pada gambar 4.31 dilakukan proses pelatihan dan pengujian model dari *data training* dan *data testing* pada fungsi yang sudah dibuat sebelumnya yaitu *train_naive_bayes_bernoulli* dan *predict_naive_bayes*.

```

print("Akurasi Manual: {:.2f}%".format(accuracy_score(y_test, y_pred_manual) * 100))
print("Confusion Matrix (Manual):\n", confusion_matrix(y_test, y_pred_manual))
print("Classification Report (Manual):\n", classification_report(y_test, y_pred_manual))

```

Gambar 4. 32 Tahap Evaluasi Model

Tabel 4. 1 Kode Gejala TBC

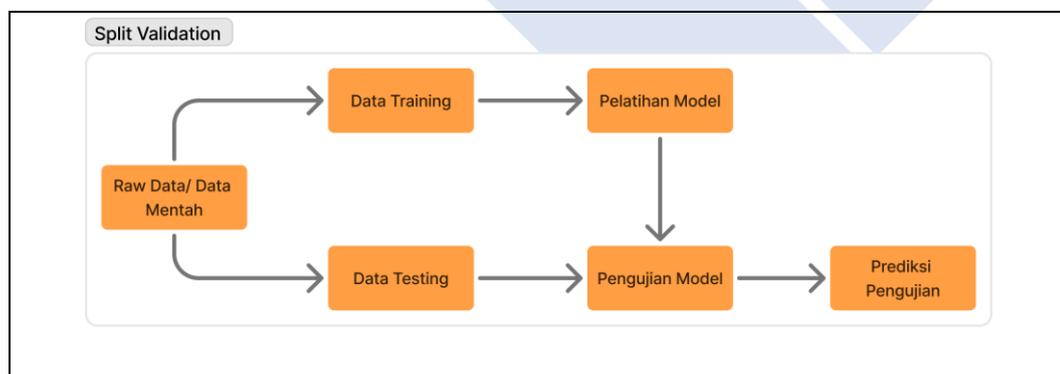
Kode Gejala	Gejala Penyakit
“G1”	“Batuk yang berlangsung lebih > 2 minggu”
“G2”	“Munculnya darah saat batuk”
“G3”	“Demam yang tidak kunjung reda selama > 1 bulan”
“G4”	“Mengalami kesulitan bernapas”
“G5”	“Hilangnya selera makan”
“G6”	“Berat badan yang terus menurun”
“G7”	“Berkeringat di malam hari meskipun tanpa aktivitas fisik”

4.4.2 Pembagian Data (*Split Validation*)

Data pasien yang telah dikumpulkan sebelumnya akan melalui tahap pengolahan data. Pada tahap ini, digunakan metode *Split Validation*, di mana data dibagi menjadi dua bagian yaitu *data training* dan *data testing*.

- *Data training* berfungsi sebagai sarana pembelajaran bagi model kecerdasan buatan (AI) guna mengenali pola dari kumpulan data yang tersedia.
- Sedangkan *data testing* dimanfaatkan untuk menilai kemampuan model dalam melakukan prediksi terhadap data yang sebelumnya tidak dikenalnya.

Pembagian ini penting untuk memastikan bahwasanya model tidak hanya menghafal data, tetapi juga mampu melakukan prediksi yang akurat terhadap kasus nyata.



Gambar 4. 34 Alur *Split Validation*

Untuk rasio pembagian data ada beberapa yang biasanya digunakan, namun penulis hanya mencoba 4 rasio pembagian data, sebagaimana pada tabel 4.2 berikut:

Tabel 4. 2 Rasio Pembagian Data

<i>Data Training</i>	<i>Data Testing</i>	Akurasi
“90%”	“10%”	“85.71%”
“80%”	“20%”	“92.86%”
“70%”	“30%”	“95.25%”
“60%”	“40%”	“92.86%”

Pengolahan data dilakukan menggunakan *platform Google Colab*, dengan metode *split validation* menggunakan rasio 70:30. Rasio ini dipilih karena memberikan hasil akurasi terbaik, yaitu 95,25%. Proses pembagian data dilakukan menggunakan fungsi *train_test_split* dari *library scikit-learn*, dengan menambahkan parameter *stratify=y* agar distribusi label tetap seimbang antara data pelatihan (*training*) dan pengujian (*testing*). Tujuan dari pembagian ini adalah untuk melatih model pada sebagian data, lalu menguji kemampuannya dalam memprediksi data yang belum pernah dikenalnya sebelumnya.

batuk_2_minggu	batuk_berdarah	demam_1_bulan	sesak_nafas	penurunan_nafsu	penurunan_berat	berkeringat	result
1	1	1	1	1	1	0	1
1	0	0	0	0	1	1	1
1	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	0
1	1	0	0	1	1	1	1
0	0	0	0	0	0	0	1
1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	1

Gambar 4. 35 Potongan *Data Training*

Pada Gambar 4.35 di atas adalah potongan dari *data training* yang nantinya data tersebut akan digunakan untuk melatih model AI *Naive Bayes*.

batuk_2_minggu	batuk_berdarah	demam_1_bulan	sesak_nafas	penurunan_nafsu	penurunan_berat	berkeringat	result
1	0	0	1	1	1	0	1
0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	1
1	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1
1	0	1	1	1	0	1	1
0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1
1	0	1	1	1	1	1	1
1	0	0	1	1	1	1	1
0	1	0	0	1	0	0	1
1	0	1	0	1	1	0	1
0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1
1	0	1	1	1	1	1	1

Gambar 4. 36 Potongan Data Testing

Pada Gambar 4.36 diatas adalah potongan dari *data testing* yang nantinya data tersebut akan digunakan untuk menguji atau evaluasi model AI yang sudah dilatih sebelumnya.

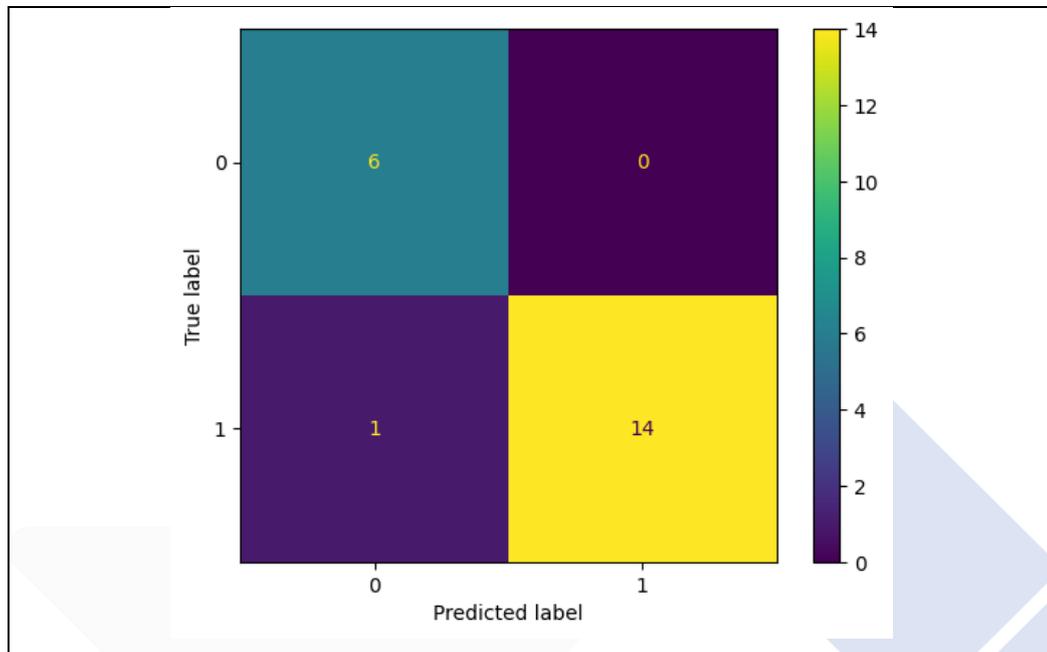
4.4.3 Evaluasi Model

Evaluasi performa model dimulai dengan menyusun *confusion matrix*, yaitu tabel dua dimensi yang menunjukkan jumlah prediksi yang tepat dan tidak tepat yang dilakukan oleh model. Tabel ini membandingkan antara label sebenarnya (y_{true}) dengan hasil prediksi model (y_{pred}), sehingga kita bisa memahami di mana model membuat keputusan yang benar dan di mana terjadi kesalahan.

Confusion matrix terdiri dari empat komponen utama:

- TP atau *True Positive*: Model berhasil memberikan prediksi kasus sebagai positif, maupun kenyataannya memang benar positif.
- TN atau *True Negative*: Model memberikan prediksinya atas kasus sebagai negatif, maupun ternyata memang benar negatif.
- FP atau *False Positive*: Model memberikan prediksi kasus sebagai positif, padahal negatif (juga dikenal sebagai kesalahan tipe I).
- FN atau *False Negative*: Model memberikan prediksi atas kasus sebagai negatif, padahal sebenarnya positif (dikenal juga sebagai kesalahan tipe II).

Berdasarkan pengujian terhadap data uji, diperoleh *confusion matrix* sebagai berikut:



Gambar 4. 37 Hasil *Confusion Matrix*

Penjelasan dari hasil *table confusion matrix* diatas adalah sebagai berikut:

- Sebanyak 6 data negatif (kelas 0) berhasil diprediksi dengan benar (TN)
- Sebanyak 14 data positif (kelas 1) juga diprediksi dengan benar (TP)
- Terdapat 1 kesalahan prediksi, yaitu 1 data positif diklasifikasikan sebagai negatif (FN)
- Tidak terdapat kesalahan prediksi positif palsu (FP = 0)

Dari *confusion matrix* ini, dapat dihitung berbagai metrik evaluasi lain yang dirangkum dalam *classification report*, yaitu:

Classification Report (Manual):				
	precision	recall	f1-score	support
0	0.86	1.00	0.92	6
1	1.00	0.93	0.97	15
accuracy			0.95	21
macro avg	0.93	0.97	0.94	21
weighted avg	0.96	0.95	0.95	21

Gambar 4. 38 Hasil *Classification Report*

Penjelasan dari hasil *classification report* pada gambar 4.38 di atas adalah sebagai berikut:

- **Precision** mengukur seberapa tepat model dalam memprediksi kelas tertentu. Pada kelas 1 (TB), nilai *precision* mencapai 1.00, yang berarti semua prediksi positif yang dibuat oleh model benar adanya.
- **Recall** menunjukkan seberapa baik model dalam menemukan semua data yang benar-benar positif. Dengan nilai *recall* 0.93, artinya model berhasil mendeteksi hampir semua kasus TB, kecuali satu kasus yang terlewat.
- **F1-Score** sebagai rerata harmonik dari *precision* dan *recall*, yang memberikan cerminan pada keseimbangan antara keduanya. Di kelas 1, nilai *F1-Score* sebesar 0.97 menandakan performa model yang sangat baik.
- **Accuracy** menunjukkan tingkat keseluruhan prediksi yang benar, yaitu sebesar 95,24%, berdasarkan seluruh data uji yang digunakan.

Dari percobaan yang sudah penulis lakukan pada semua data uji sebanyak 21 data, ditemukan data gejala kesalahan yang masuk ke dalam FN atau *false negative*, dimana program memberikan prediksi sebagai negatif padahal sebenarnya data tersebut adalah data dengan label positif, yaitu sebagai berikut:

Tabel 4. 3 Data Gejala *False Negative*

G1	G2	G3	G4	G5	G6	G7	HASIL
0	1	0	0	1	0	0	1

Lalu data tersebut akan diuji dengan menggunakan perhitungan program, dan hasilnya adalah sebagai berikut:

```

Peluang Positif TBC (manual): 7.32%
Peluang Negatif TBC (manual): 92.68%
Hasil Deteksi (threshold 85%): Tidak TBC
Hasil Deteksi (manual): Negatif
  
```

Gambar 4. 39 Hasil Pengujian Data Gejala *False Negative*

Jadi, dari gambar 4.39 diatas bisa dilihat bahwa data tersebut diprediksi sebagai data dengan label negatif padahal data sebenarnya adalah data dengan label positif. Meskipun model mencapai presisi sempurna (100%), keberadaan

satu data *false negative* menyebabkan penurunan pada metrik *recall*, yang turut memengaruhi *F1-Score* dan akurasi keseluruhan. Hal ini mengindikasikan bahwa model sangat baik dalam menghindari prediksi positif yang salah, namun masih memiliki kekurangan dalam mengenali seluruh data positif. Secara keseluruhan, model menunjukkan performa yang sangat baik dengan akurasi sekitar 95,24% dan *F1-Score* sebesar 96,55%, yang menandakan keseimbangan antara ketepatan dan kelengkapan dalam klasifikasi.

Berdasarkan hasil evaluasi ini, dapat disimpulkan bahwasanya model *Bernoulli Naive Bayes* yang diterapkan memiliki performa tinggi dalam mengklasifikasikan data pasien ke dalam kategori TB dan Non-TB, dengan tingkat akurasi dan ketepatan prediksi yang sangat memuaskan.

Untuk menghitung *precision*, *recall*, *f1-score* dan *accuracy*, langkah-langkahnya adalah sebagai berikut:

- *True Negative* (TN) = 6
- *False Positive* (FP) = 0
- *False Negative* (FN) = 1
- *True Positive* (TP) = 14

Untuk kelas 0 (negatif)

$$\text{Precision} = \frac{TN}{TN+FN} = \frac{6}{6+1} = \frac{6}{7} = 0,857$$

$$\text{Recall} = \frac{TN}{TN+FP} = \frac{6}{6+0} = \frac{6}{6} = 1$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{recall}} = 2 \cdot \frac{0,857 \cdot 1}{0,857 + 1} = 2 \cdot \frac{0,857}{1,857} = 0,922$$

Untuk kelas 1 (positif)

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{14}{14+0} = \frac{14}{14} = 1$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{14}{14+1} = \frac{14}{15} = 0,933$$

$$F1\text{-Score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot \frac{1 \cdot 0,933}{1 + 0,933} = 2 \cdot \frac{0,933}{1,933} = 0,965$$

Untuk *Accuracy*

$$Accuracy = \frac{TP+TN}{Total} = \frac{16+4}{21} = \frac{20}{21} = 0,9524 = 95,24\%$$

Dengan menggunakan fungsi *data.corr()*, kita dapat melihat sejauh mana hubungan atau korelasi antara setiap fitur dengan label hasil (*result*). Nilai korelasi ini berada dalam rentang -1 hingga 1, di mana:

- Nilai yang hampir mencapai 1 mencerminkan korelasi positif yang sangat kuat,
- Nilai yang mendekati -1 mencerminkan adanya korelasi negatif yang kuat, dan
- Nilai yang berada di sekitar 0 mengindikasikan bahwasanya terdapat korelasi yang lemah, bahkan mungkin tidak ada keterkaitan antara fitur dengan hasil.

Hasil analisis korelasi antara fitur-fitur dan label *result* dapat dilihat pada gambar berikut:

<code>result</code>	<code>1.000000</code>
<code>batuk_2_minggu</code>	<code>0.898519</code>
<code>penurunan_nafsu</code>	<code>0.897616</code>
<code>sesak_nafas</code>	<code>0.668437</code>
<code>penurunan_berat</code>	<code>0.668437</code>
<code>demam_1_bulan</code>	<code>0.611844</code>
<code>berkeringat</code>	<code>0.465710</code>
<code>batuk_berdarah</code>	<code>0.309426</code>

Gambar 4. 40 Korelasi Fitur Dengan Klasifikasi

Dari gambar di atas, terlihat bahwa fitur batuk lebih dari 2 minggu dan penurunan nafsu makan memiliki korelasi tertinggi terhadap status pasien (positif TB), dengan nilai korelasi masing-masing 0.89. Hal ini menunjukkan bahwa gejala tersebut sangat relevan dalam mendeteksi kemungkinan TB. Sebaliknya, fitur batuk disertai darah memiliki korelasi paling rendah (0.30), yang menunjukkan bahwa meskipun gejala ini penting secara klinis, dalam *dataset* ini kemunculannya tidak selalu berasosiasi langsung dengan hasil positif TB.

4.4.4 Hasil Pengujian

Data Pengujian:

Tabel 4. 4 Gejala Pengujian Data Uji

G1	G2	G3	G4	G5	G6	G7	HASIL
1	1	0	0	1	1	0	1

Pada perhitungan menggunakan program, didapat hasil:

<p>Peluang Positif TBC (manual): 99.64%</p> <p>Peluang Negatif TBC (manual): 0.36%</p> <p>Hasil Deteksi (threshold 85%): Positif TBC</p> <p>Hasil Deteksi (manual): Positif</p>

Gambar 4. 41 Hasil Perhitungan Program

Sedangkan pada perhitungan secara manual dari kertas, langkah-langkahnya adalah sebagai berikut:

Langkah 1: Menghitung *Prior Probability*

<p>Jumlah Data Training = 47</p> <p>Jumlah Data Training Positif = 32</p> <p>Jumlah Data Training Negatif = 15</p> $P(\text{Positif}) = \frac{32}{47} = 0,6808$ $P(\text{Negatif}) = \frac{15}{47} = 0,3191$
--

Langkah 2: Menghitung *Likelihood* ($P(\text{Fitur}|\text{Kelas})$) atau persentase setiap fitur atau gejala

<p>Persentase Positif</p> $P(W_i \text{positif}) = \frac{\text{freq}(W_i, \text{positif}) + 1}{N_{\text{positif}} + V_{\text{positif}}}$ $[G1] = \frac{31}{32} = \frac{32}{34} = 0,9411$
--

$$[G2] = \frac{10}{32} = \frac{11}{34} = 0,3235$$

$$[G3] = \frac{21}{32} = \frac{22}{34} = 0,6470$$

$$[G4] = \frac{24}{32} = \frac{25}{34} = 0,7352$$

$$[G5] = \frac{32}{32} = \frac{33}{34} = 0,9705$$

$$[G6] = \frac{21}{32} = \frac{22}{34} = 0,6470$$

$$[G7] = \frac{20}{32} = \frac{21}{34} = 0,6176$$

Persentase Negatif

$$P(Wi|negatif) = \frac{freq(Wi,negatif) + 1}{N_{negatif} + V_{negatif}}$$

$$[G1] = \frac{1}{15} = \frac{2}{17} = 0,1176$$

$$[G2] = \frac{0}{15} = \frac{1}{17} = 0,0588$$

$$[G3] = \frac{0}{15} = \frac{1}{17} = 0,0588$$

$$[G4] = \frac{0}{15} = \frac{1}{17} = 0,0588$$

$$[G5] = \frac{2}{15} = \frac{3}{17} = 0,1764$$

$$[G6] = \frac{0}{15} = \frac{1}{17} = 0,0588$$

$$[G7] = \frac{2}{15} = \frac{3}{17} = 0,1764$$

Langkah 3: Menghitung *Posterior Probability*

Yaitu mengalikan semua likelihood dengan syarat, jika data gejala pasien baru itu 1 maka gunakan hasil likelihood asli, tetapi jika 0 maka kurangi hasil likelihood dengan 1.

Likelihood Positif

$$[G1] \text{ bernilai } 1 = 0,9411$$

$$[G2] \text{ bernilai } 1 = 0,3235$$

$$[G3] \text{ bernilai } 0 = 1 - 0,6470 = 0,3529$$

$$[G4] \text{ bernilai } 0 = 1 - 0,7352 = 0,2647$$

$$[G5] \text{ bernilai } 1 = 0,9705$$

$$[G6] \text{ bernilai } 1 = 0,6470$$

$$[G7] \text{ bernilai } 0 = 1 - 0,6176 = 0,3823$$

$$\text{Jadi, } 0,9411 \times 0,3235 \times 0,3529 \times 0,2647 \times 0,9705 \times 0,6470 \times 0,3823 = 0,00683$$

$$\text{Lalu kalikan dengan prior positif} = 0,00683 \times 0,6808 = 0,004650$$

Likelihood Negatif

$$[G1] \text{ bernilai } 1 = 0,1176$$

$$[G2] \text{ bernilai } 1 = 0,0588$$

$$[G3] \text{ bernilai } 0 = 1 - 0,0588 = 0,9411$$

$$[G4] \text{ bernilai } 0 = 1 - 0,0588 = 0,9411$$

$$[G5] \text{ bernilai } 1 = 0,1764$$

$$[G6] \text{ bernilai } 1 = 0,0588$$

$$[G7] \text{ bernilai } 0 = 1 - 0,1764 = 0,8235$$

$$\text{Jadi, } 0,1176 \times 0,0588 \times 0,9411 \times 0,9411 \times 0,1764 \times 0,0588 \times 0,8235 = 0,0000524$$

$$\text{Lalu kalikan dengan prior negatif} = 0,0000524 \times 0,3191 = 0,00001672$$

Langkah 4: Normalisasi

Normalisasi dilakukan untuk mendapatkan persentase hasil positif dan juga negatif.

$$P(\text{total}) = \text{posterior positif} + \text{posterior negatif} \\ = 0,004650 + 0,00001672 = 0,00467$$

$$\text{Persentase Positif atau } P(\text{Positif}) = \frac{0,00465}{0,00467} = 0,9957 = 99,57\%$$

Jadi, berdasarkan hasil dari perhitungan menggunakan program yaitu 99,64% dan perhitungan secara manual yaitu 99,57%, hasil tersebut bisa

dikategorikan mirip dan hasil diagnosa untuk pasien baru dengan gejala tersebut diklasifikasikan sebagai pasien positif TBC.

4.5 Pengujian Sistem

Selanjutnya, tahapan terakhir dari pengembangan sistem ialah tahapan pengujian sistem. Pada pengujian ini, metode pengujian *User Acceptance Testing* (UAT) digunakan guna memastikan bahwa sistem telah memenuhi kebutuhan dan harapan pengguna akhir. Pengujian UAT dilakukan dengan melibatkan pengguna langsung untuk mencoba seluruh fungsi utama dari sistem. Pengujian ini bertujuan untuk mengevaluasi apakah sistem layak digunakan dan siap diterapkan secara operasional. Proses pengujian dilakukan untuk memperoleh umpan balik dari berbagai sudut pandang pengguna. Adapun berikut merupakan hasil dari pengujian yang telah dilakukan.

4.5.1 Pengujian Metode *User Acceptance Testing* (UAT)

Proses UAT atau *User Acceptance Testing* dilakukan dengan melibatkan pengguna langsung untuk menguji apakah sistem yang dikembangkan sudah sesuai kebutuhan. Hasil dari pengujian ini digunakan sebagai bukti apakah sistem tersebut diterima atau perlu perbaikan lebih lanjut. Dalam pelaksanaannya, UAT dilakukan melalui kuesioner *online* yang berisi 17 pertanyaan, dan disebarakan kepada masyarakat umum menggunakan *Google Form*. Pertanyaan dari UAT dijelaskan pada tabel 4.5 berikut:

Tabel 4. 5 Kuesioner UAT

No.	Pertanyaan	Skala Penilaian				
		1	2	3	4	5
Tingkat Kemudahan Pengguna						
1	Tampilan antarmuka sistem mudah dipahami.					
2	Informasi yang disajikan mudah dimengerti.					
3	Sistem mudah digunakan tanpa pelatihan					

No.	Pertanyaan	Skala Penilaian				
		1	2	3	4	5
	khusus.					
4	Fitur konsultasi dengan <i>admin</i> mudah diakses dan digunakan.					
5	Lokasi <i>chatbot</i> mudah dikenali dan ditemukan.					
6	<i>Chatbot</i> mudah digunakan dan membantu dalam proses konsultasi awal.					
Fungsionalitas Sistem						
7	Sistem dapat melakukan diagnosa dini TBC dengan benar.					
8	Peta lokasi pasien TBC ditampilkan dengan akurat.					
9	Hasil diagnosa berdasarkan input gejala dapat dipercaya.					
10	Sistem dapat menampilkan data peta dan diagnosa dengan baik.					
11	<i>Chatbot</i> mampu memberikan jawaban yang relevan.					
Kinerja Sistem						
12	Sistem responsif terhadap setiap input pengguna.					
13	Sistem berjalan lancar tanpa error saat digunakan.					
14	Waktu respon sistem cukup cepat.					
Tingkat Kepuasan						
15	Saya merasa terbantu dengan adanya sistem ini.					
16	Sistem ini bermanfaat dalam mendeteksi					

No.	Pertanyaan	Skala Penilaian				
		1	2	3	4	5
	dini penyakit TBC.					
17	Saya puas dengan keseluruhan fungsi dari sistem ini.					

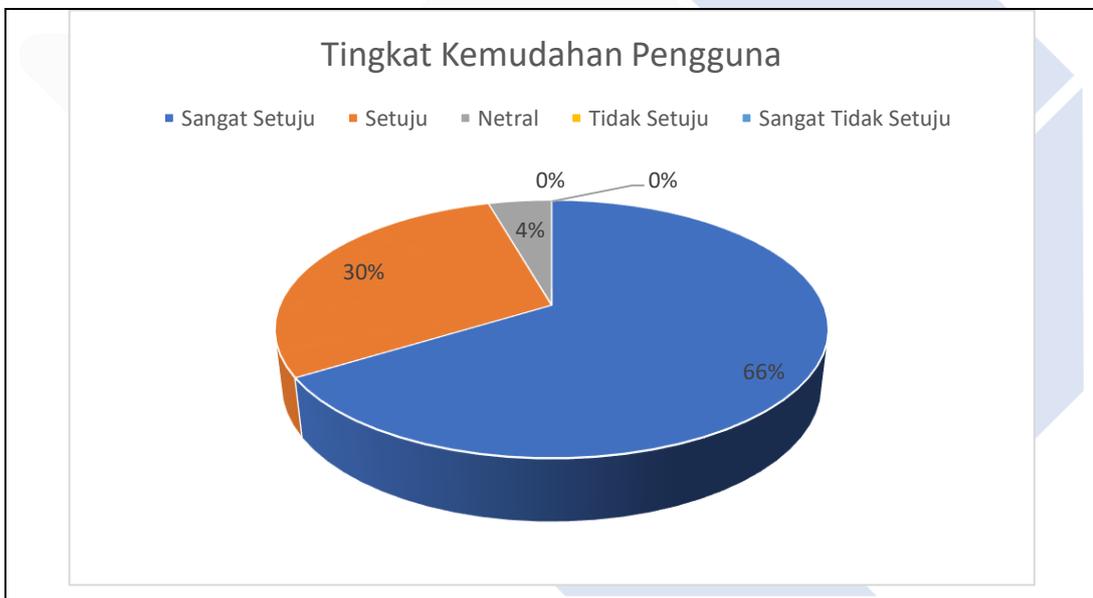
Lalu dari pertanyaan-pertanyaan tersebut akan dilakukan perhitungan dari kuesioner yang sudah dilakukan secara *online* sebelumnya, dan hasil dari perhitungan akan dibagi menjadi 4 aspek kategori, dimana perhitungan-perhitungan tersebut dijelaskan sebagai berikut:

- Aspek Tingkat Kemudahan Pengguna

Tabel 4. 6 Hasil Kuesioner Aspek Tingkat Kemudahan Pengguna

Tingkat Kemudahan Pengguna				
Poin		Jumlah	Skor	%Frekuensi
5	X	162	810	$\frac{810}{1224} \times 100\% = 66,18\%$
4	X	90	360	$\frac{360}{1224} \times 100\% = 29,41\%$
3	X	18	54	$\frac{54}{1224} \times 100\% = 4,41\%$
2	X	0	0	0%
1	X	0	0	0%
Total			1224	100%
Jumlah Responden	45			
Jumlah Pertanyaan	6			
Skor Maksimal	5 X 45 X 6	1350		
%Index	$\frac{1224}{1350} \times 100\%$	90,67%		

Berdasarkan Tabel 4.6, mayoritas responden memberikan tanggapan positif terhadap sistem yang dikembangkan. Sebanyak 66,18% responden menyatakan sangat setuju, 29,41% setuju, 4,41% netral, 0% tidak setuju, dan 0% yang sangat tidak setuju. Dari hasil kuesioner ini, diperoleh total skor 1224 dari 45 responden dan 6 pertanyaan. Untuk mengetahui tingkat kemudahan pengguna, perhitungan dilakukan dengan mempertimbangkan skor tertinggi (maksimal) yaitu 1350 dan total skor yang diperoleh. Berdasarkan rumus *index* persentase, didapatkan nilai sebesar 90,67% untuk aspek tingkat kemudahan pengguna. Nilai ini menunjukkan bahwa tingkat kemudahan pengguna pada sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang termasuk dalam kategori layak.



Gambar 4. 42 Grafik Persentase *Index* Tingkat Kemudahan Pengguna

Dan dari gambar 4.42 diatas dapat dilihat grafik persentase *index* dari aspek tingkat kemudahan pengguna, terlihat bahwa yang paling dominan adalah poin Sangat Setuju dengan persentase sebesar 66%.

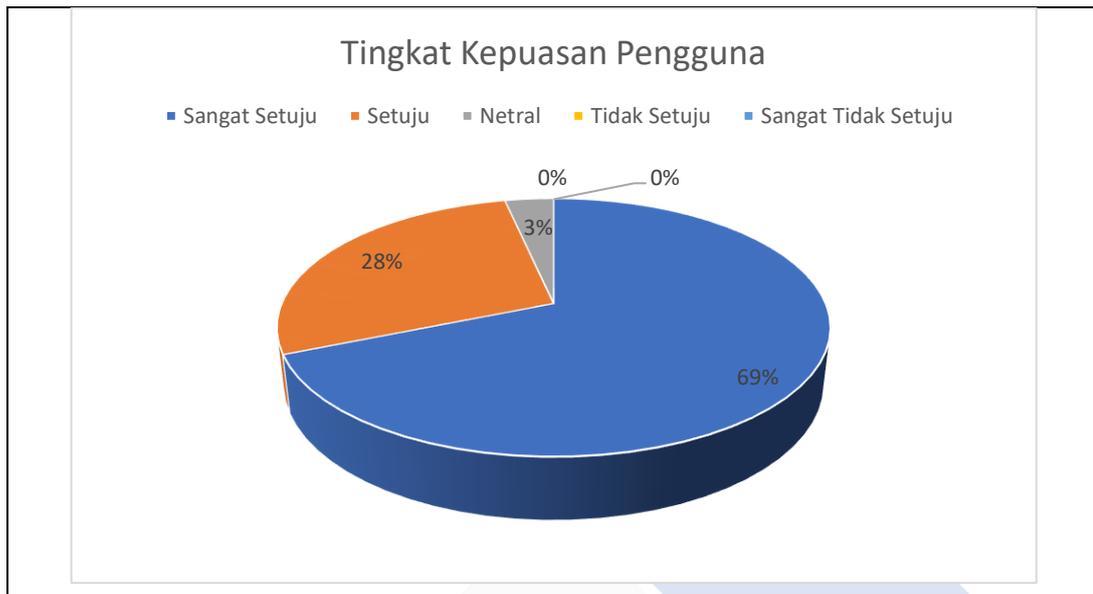
- Aspek Tingkat Kepuasan Pengguna

Tabel 4. 7 Hasil Kuesioner Aspek Tingkat Kepuasan Pengguna

Tingkat Kepuasan Pengguna				
Poin		Jumlah	Skor	%Frekuensi

Tingkat Kepuasan Pengguna				
Poin		Jumlah	Skor	%Frekuensi
5	X	85	425	$\frac{425}{618} \times 100\% = 68,77\%$
4	X	43	172	$\frac{172}{618} \times 100\% = 27,83\%$
3	X	7	21	$\frac{21}{618} \times 100\% = 3,40\%$
2	X	0	0	0%
1	X	0	0	0%
Total			618	100%
Jumlah Responden	45			
Jumlah Pertanyaan	3			
Skor Maksimal	5 X 45 X 3	675		
%Index	$\frac{618}{675} \times 100\%$	91,56%		

Berdasarkan Tabel 4.7, mayoritas responden memberikan tanggapan positif terhadap sistem yang dikembangkan. Sebanyak 68,77% responden menyatakan sangat setuju, 27,83% setuju, 3,41% netral, 0% tidak setuju, dan 0% yang sangat tidak setuju. Dari hasil kuesioner ini, diperoleh total skor 618 dari 45 responden dan 3 pertanyaan. Untuk mengetahui tingkat kepuasan pengguna, perhitungan dilakukan dengan mempertimbangkan skor tertinggi (maksimal) yaitu 675 dan total skor yang diperoleh. Berdasarkan rumus *index* persentase, didapatkan nilai sebesar 91,56% untuk aspek tingkat kepuasan pengguna. Nilai ini menunjukkan bahwa tingkat kepuasan pengguna pada sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang termasuk dalam kategori layak.



Gambar 4. 43 Grafik Persentase *Index* Tingkat Kepuasan Pengguna

Dan dari gambar 4.43 diatas dapat dilihat grafik persentase *index* dari aspek tingkat kepuasan pengguna, terlihat bahwa yang paling dominan adalah poin Sangat Setuju dengan persentase sebesar 69%.

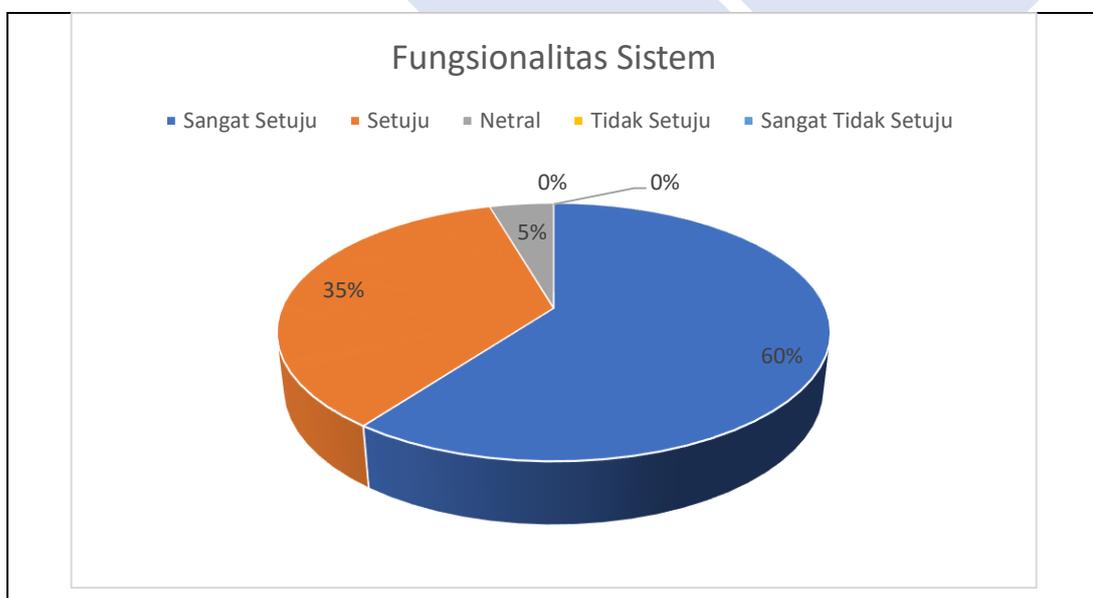
- Aspek Fungsionalitas Sistem

Tabel 4. 8 Hasil Kuesioner Aspek Fungsionalitas Sistem

Fungsionalitas Sistem				
Poin		Jumlah	Skor	%Frekuensi
5	X	121	605	$\frac{605}{1006} \times 100\% = 60,14\%$
4	X	89	356	$\frac{356}{1006} \times 100\% = 35,39\%$
3	X	15	45	$\frac{45}{1006} \times 100\% = 4,47\%$
2	X	0	0	0%
1	X	0	0	0%
Total			1006	100%
Jumlah Responden	45			

Fungsionalitas Sistem				
Poin		Jumlah	Skor	%Frekuensi
Jumlah Pertanyaan	5			
Skor Maksimal	5 X 45 X 5	1125		
%Index	$\frac{1006}{1125} \times 100\%$	89,42%		

Berdasarkan Tabel 4.8, mayoritas responden memberikan tanggapan positif terhadap sistem yang dikembangkan. Sebanyak 60,14% responden menyatakan sangat setuju, 35,39% setuju, 4,47% netral, 0% tidak setuju, dan 0% yang sangat tidak setuju. Dari hasil kuesioner ini, diperoleh total skor 1006 dari 45 responden dan 5 pertanyaan. Untuk mengetahui aspek fungsionalitas sistem, perhitungan dilakukan dengan mempertimbangkan skor tertinggi (maksimal) yaitu 1125 dan total skor yang diperoleh. Berdasarkan rumus *index* persentase, didapatkan nilai sebesar 89,42% untuk aspek fungsionalitas sistem. Nilai ini menunjukkan bahwa tingkat fungsionalitas sistem pada sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang termasuk dalam kategori layak.



Gambar 4. 44 Grafik Persentase *Index* Fungsionalitas Sistem

Dan dari gambar 4.44 diatas dapat dilihat grafik persentase *index* dari aspek fungsionalitas sistem, terlihat bahwa yang paling dominan adalah poin Sangat Setuju dengan persentase sebesar 60%.

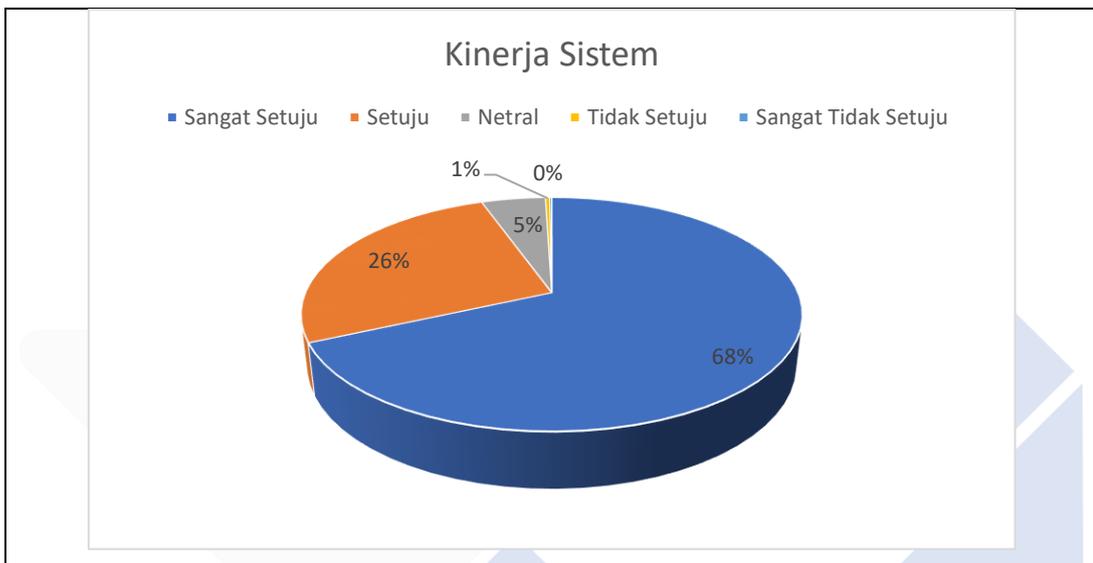
- Aspek Kinerja Sistem

Tabel 4. 9 Hasil Kuesioner Aspek Kinerja Sistem

Kinerja Sistem				
Poin		Jumlah	Skor	%Frekuensi
5	X	83	415	$\frac{415}{608} \times 100\% = 68,26\%$
4	X	40	160	$\frac{160}{608} \times 100\% = 26,32\%$
3	X	10	30	$\frac{30}{608} \times 100\% = 4,93\%$
2	X	1	2	$\frac{2}{608} \times 100\% = 0,33\%$
1	X	1	1	$\frac{1}{608} \times 100\% = 0,16\%$
Total			608	100%
Jumlah Responden	45			
Jumlah Pertanyaan	3			
Skor Maksimal	5 X 45 X 3	675		
%Index	$\frac{608}{675} \times 100\%$	90,07%		

Berdasarkan Tabel 4.9, mayoritas responden memberikan tanggapan positif terhadap sistem yang dikembangkan. Sebanyak 68,26% responden menyatakan sangat setuju, 26,32% setuju, 4,93% netral, 0,33% tidak setuju, dan hanya 0,16% yang sangat tidak setuju. Dari hasil kuesioner ini, diperoleh total skor 608 dari 45 responden dan 3 pertanyaan. Untuk mengetahui aspek kinerja

sistem, perhitungan dilakukan dengan mempertimbangkan skor tertinggi (maksimal) yaitu 675 dan total skor yang diperoleh. Berdasarkan rumus *index* persentase, didapatkan nilai sebesar 90,07% untuk aspek kinerja sistem. Nilai ini menunjukkan bahwa tingkat kinerja sistem pada sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang termasuk dalam kategori layak.



Gambar 4. 45 Grafik Persentase *Index* Kinerja Sistem

Dan dari gambar 4.45 diatas dapat dilihat grafik persentase *index* dari aspek kinerja sistem, terlihat bahwa yang paling dominan adalah poin Sangat Setuju dengan persentase sebesar 68%.

BAB 5

KESIMPULAN DAN SARAN

Sebagai penutup dari laporan akhir ini, Bab 5 memuat kesimpulan maupun saran yang disusun berlandaskan perolehan analisa yang telah dilakukan sebelumnya. Berikut ini poin-poin kesimpulan serta saran yang dapat disampaikan:

5.1 Kesimpulan

Sebagaimana perolehan perancangan maupun implementasi sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang, dapat disimpulkan hal-hal berikut:

1. Sistem berhasil dikembangkan dengan dua fitur utama, yaitu:
Pemetaan penyebaran kasus TBC berdasarkan wilayah kecamatan menggunakan Sistem Informasi Geografis (SIG) dan Diagnosa dini penyakit TBC menggunakan metode klasifikasi *Naive Bayes* dengan dukungan algoritma *Laplace Smoothing*.
2. Sistem dilengkapi fitur tambahan seperti *chatbot* berbasis *OpenAI* dan konsultasi langsung dengan *admin*, yang meningkatkan interaksi dan pelayanan kepada pengguna.
3. Hasil pengujian model menunjukkan tingkat akurasi yang tinggi, yaitu hingga 95.25% dengan split *data training* 70:30, serta hasil perhitungan manual yang konsisten dengan hasil sistem.
4. Berdasarkan hasil *User Acceptance Testing*, sistem mendapatkan *index* kepuasan pengguna sebesar 91,56%, yang mengindikasikan bahwasanya sistem diterima dan dianggap layak digunakan oleh masyarakat.

5.2 Saran

Dalam proses pengembangan sistem informasi pemetaan dan diagnosa dini penyakit TBC menggunakan metode *Naive Bayes* di Kota Pangkalpinang, penulis

sadar jika terdapat berbagai kekurangan maupun keterbatasan yang membuat sistem ini belum sepenuhnya sempurna. Sehingga, penulis sangat berharap agar sistem ini mampu terus dikembangkan dan disempurnakan ke depannya. Adapun saran yang mampu menjadi rujukan pertimbangan guna pengembangan selanjutnya meliputi:

1. Penambahan fitur notifikasi atau reminder pengobatan bagi pasien positif agar lebih bermanfaat secara medis.
2. Disarankan untuk melakukan evaluasi lebih lanjut dengan mencoba metode klasifikasi lain, seperti *Random Forest*, *Support Vector Machine* (SVM), atau algoritma lainnya, guna membandingkan performa model dan memperoleh hasil yang lebih optimal.
3. Gunakan *dataset* yang lebih banyak agar hasil diagnosa lebih akurat dan lebih bagus lagi.
4. Pengembangan sistem ke-*platform mobile* agar lebih mudah diakses oleh masyarakat luas.

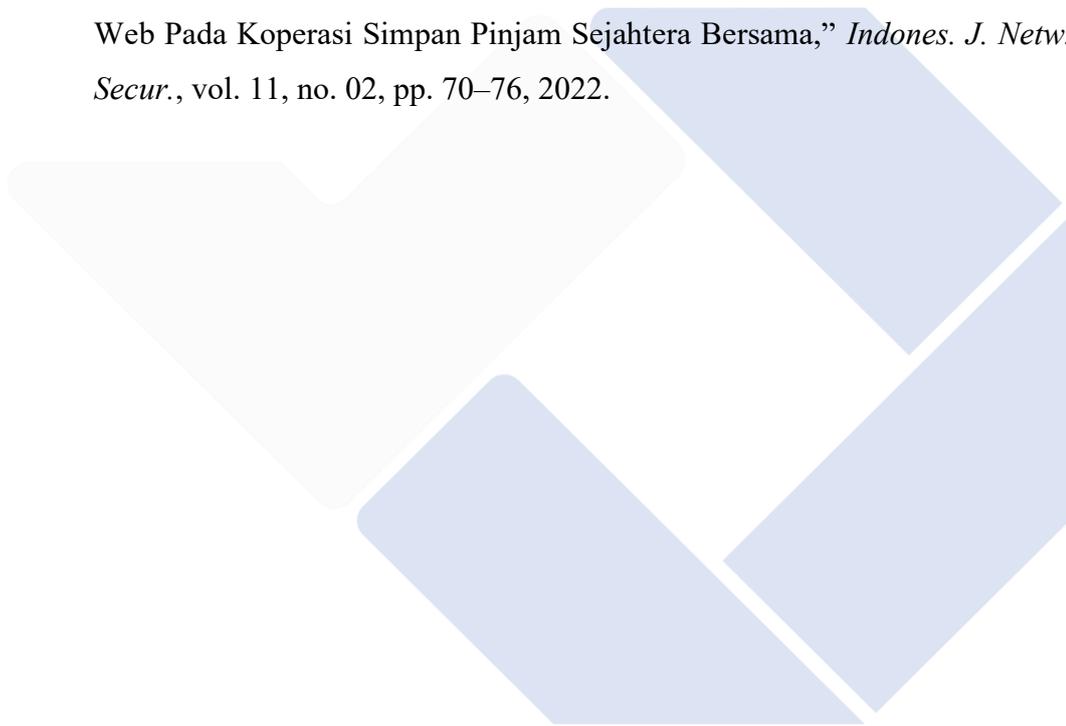
DAFTAR PUSTAKA

- [1] D. S. Wulandari and M. G. Rohman, "Implementasi Metode Naïve Bayes Pada Sistem Pakar Diagnosa Penyakit Tuberculosis," *Gener. J.*, vol. 7, no. 3, pp. 64–76, 2023, doi: 10.29407/gj.v7i3.21054.
- [2] A. A. Prameswaty, M. H. P. Swari, and W. S. J. Saputra, "Perancangan Sistem Pakar Diagnosis Penyakit Tbc Paru Dengan Metode Certainty Factor Dan Dempster Shafer," vol. 8, no. 5, pp. 8658–8663, 2024.
- [3] Z. Indra and M. R. P. Harahap, "Pengembangan Sistem Pakar Untuk Diagnosa Penyakit Tbc Di Masa Pandemi Dengan Menggunakan Metode Forward Chaining Di Puskesmas Medan Johor," vol. 2, no. 1, pp. 40–46, 2015.
- [4] A. A. Nabila and A. Desiani, "Sistem Pakar Diagnosa Penyakit Tuberculosis Dengan Metode Certainty," *J. Artif. Intell. Softw. Eng.*, vol. 3, no. 1, p. 1, 2023, doi: 10.30811/jaise.v3i1.3901.
- [5] A. Sulistyono, N. H. Nariswaria, and H. Rohman, "Pemetaan Penyakit Tuberculosis Dengan Sistem Informasi Geografis Di Wilayah Bantul Mapping of Tuberculosis Disease with Geographic Information System in Bantul Region," *J. Ilmu Kesehat. Bhakti Setya Med.*, vol. 7, no. tuberculosis, pp. 26–37, 2022.
- [6] E. Pramana E and A. Saifuddin, "Implementasi Data Mining Untuk Diagnosa Prediksi Penyakit Tuberculosis (Tbc) Menggunakan Algoritma Naïve Bayes," *J. Mater. Process. Technol.*, vol. 1, no. 1, pp. 1–8, 2023, [Online]. Available: <http://dx.doi.org/10.1016/j.cirp.2016.06.001><http://dx.doi.org/10.1016/j.powtec.2016.12.055><https://doi.org/10.1016/j.ijfatigue.2019.02.006>

[Ahttps://doi.org/10.1016/j.matlet.2019.04.024](https://doi.org/10.1016/j.matlet.2019.04.024)[Ahttps://doi.org/10.1016/j.matlet.2019.127252](https://doi.org/10.1016/j.matlet.2019.127252)[Ahttp://dx.doi.o](http://dx.doi.o)

- [7] G. W. Nyipto Wibowo, S. Widiastuti, M. Muratno, E. Lolang, and S. Soraya, "Penerapan Metode Teorema Bayes Dalam Mendiagnosa Penyakit Tuberculosis," *Build. Informatics, Technol. Sci.*, vol. 4, no. 4, pp. 1782–1788, 2023, doi: 10.47065/bits.v4i4.3035.
- [8] Y. Anzari, M. Mutmainnah, S. Wibowo, and C. Cecilia, "Sistem Informasi Geografis Pemetaan Penyebaran TBC di Provinsi Jambi Tahun 2022," vol. 4, no. 1, pp. 44–52, 2025.
- [9] N. Sahrnun and S. Sularno, "Rancang Bangun Sistem Informasi Geografis Menemukan Lokasi Dokter Hewan Berbasis Android," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 1, pp. 21–32, 2023, doi: 10.47233/jteksis.v5i1.732.
- [10] R. Tarigan, D. Saripurna, and J. Halim, "Sistem Pakar Mendiagnosa Penyakit TB Paru Pada Anak Menggunakan Teorema Bayes," *J. Cyber Tech*, vol. 5, no. 1, p. 36, 2023, doi: 10.53513/jct.v5i1.7856.
- [11] N. Andini, R. Taufiq, D. Y. Priyanggodo, and Y. Sugiyani, "Penggunaan Metode Prototype Pada Pengembangan Sistem Informasi Imunisasi Posyandu," *JIKA (Jurnal Inform.*, vol. 7, no. 4, p. 431, 2023, doi: 10.31000/jika.v7i4.9329.
- [12] R. Trisudarmo, "Penerapan Metode Prototype dalam Sistem E-Government pada Pelayanan Administrasi Kependudukan," *J. Inform. dan Teknol. Pendidik.*, vol. 2, no. 2, pp. 64–71, 2022, doi: 10.25008/jitp.v2i2.35.
- [13] S. W. Ramdany, S. A. Kaidar, B. Aguchino, C. A. A. Putri, and R. Anggie, "Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web," *J. Ind. Eng. Syst.*, vol. 5, no. 1, 2024, doi: 10.31599/2e9afp31.
- [14] I. Wahyudi, Fahrullah, F. Alameka, and Haerullah, "Analisis Blackbox Testing Dan User Acceptance Testing Terhadap Sistem Informasi

- Solusimedsosku,” *J. Teknosains Kodepena* |, vol. 04, no. 01, pp. 1–9, 2023.
- [15] S. Purwanti and R. Z. Anggraini Putri, “Pengembangan Modul Berbasis Hots Pada Tema 6 Materi Membandingkan Siklus Makhluk Hidup Kelas Iv Sekolah Dasar,” *Elem. Sch. J. Pendidik. dan Pembelajaran ke-SD-an*, vol. 8, no. 1, pp. 155–160, 2021, doi: 10.31316/esjurnal.v8i1.1080.
- [16] R. Rosaly and A. Prasetyo, “Flowchart Beserta Fungsi dan Simbol-Symbol,” *J. Chem. Inf. Model.*, vol. 2, no. 3, pp. 5–7, 2020.
- [17] B. Simare Mare and A. A. Yana, “Perancangan Sistem Informasi Berbasis Web Pada Koperasi Simpan Pinjam Sejahtera Bersama,” *Indones. J. Netw. Secur.*, vol. 11, no. 02, pp. 70–76, 2022.





LAMPIRAN 1
DAFTAR RIWAYAT HIDUP

Lampiran 1 Daftar Riwayat Hidup

1. Data Pribadi

Nama Lengkap : Andika Saputra
NIM : 1062203
Tempat & Tanggal Lahir : Pangkalpinang, 10 Mei 2004
Alamat : Jl. A. Yani GG. Sahabat
Email : andikasaputraofficials@gmail.com
No. Hp : 085709962535
Jenis Kelamin : Laki-Laki
Agama : Islam



2. Riwayat Pendidikan

SDN 43 Pangkalpinang : 2009 - 2015
SMPN 5 Pangkalpinang : 2015 - 2018
SMKN 2 Pangkalpinang : 2018 - 2021
Politeknik Manufaktur Negeri Bangka Belitung : 2022 – Sekarang

Sungailiat, 21 Juli 2025

Tanda Tangan,

Andika Saputra



LAMPIRAN 2
FORM BIMBINGAN

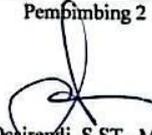


LAMPIRAN 3
FORM MONITORING

Lampiran 3 Form Monitoring

		FORM MONITORING PROYEK AKHIR TAHUN AKADEMIK 2024/2025	
		JUDUL SISTEM INFORMASI PEMETAAN DAN DIAGNOSA DINI PENYAKIT TBC MENGGUNAKAN METODE NAIVE BAYES DI KOTA PANGKALPINANG	
Nama Mahasiswa		1. ANDIKA SAPUTRA / NIM : 1062203	
Monitoring ke	Tanggal	Progress Alat	Paraf Pembimbing
	17/09 2025	Progress Altim 50%	✓ Rika
	23/05 2025	Progress Altim Pakar, Locam 20%	✓ Rika

KESIAPAN ALAT UNTUK SIDANG : SIAP ~~ABELUM~~ (coret yang tidak terpenuhi)

Mengetahui	
Pembimbing 1  (Riki Afrizyah, S.T., M.T.)	Pembimbing 2  (Ocsirendi, S.ST., M.T.)

Silahkan diatur kolom baru jika jumlah pembimbing lebih dari yang tersedia.



LAMPIRAN 4
WAWANCARA PASIEN TBC

Lampiran 4 Wawancara Pasien TBC



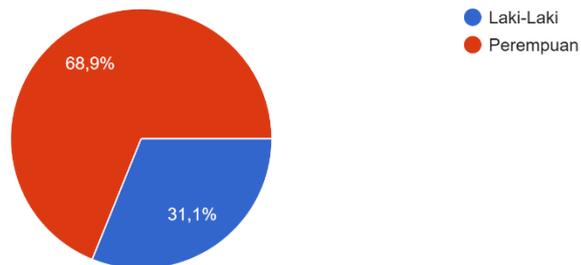


LAMPIRAN 5
BUKTI KUESIONER *USER ACCEPTANCE TESTING* (UAT)

Lampiran 5 Bukti Kuesioner *User Acceptance Testing* (UAT)

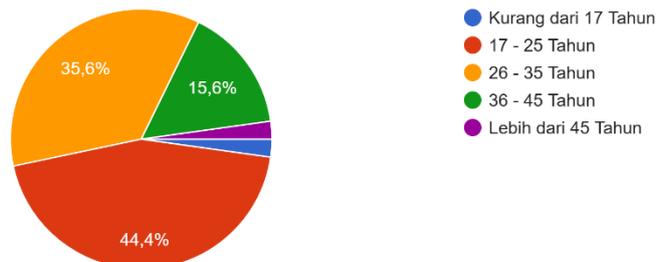
• Jenis Kelamin

Jenis Kelamin
45 jawaban



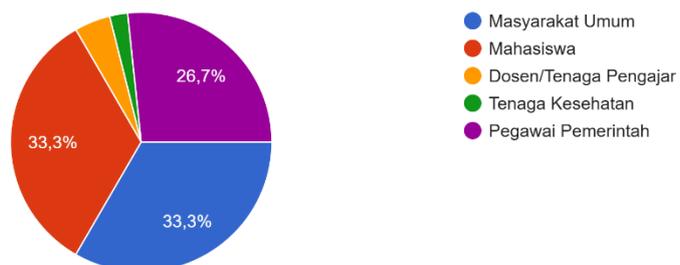
• Umur

Umur
45 jawaban



• Profesi/Latar Belakang

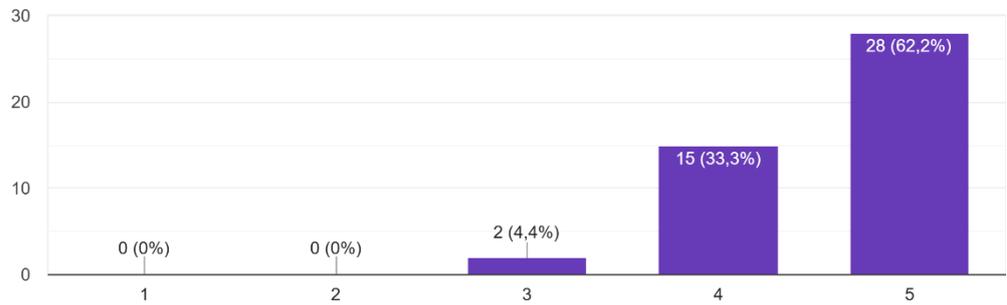
Profesi/Latar Belakang
45 jawaban



- **Pertanyaan Ke-1**

Tampilan antarmuka sistem mudah dipahami

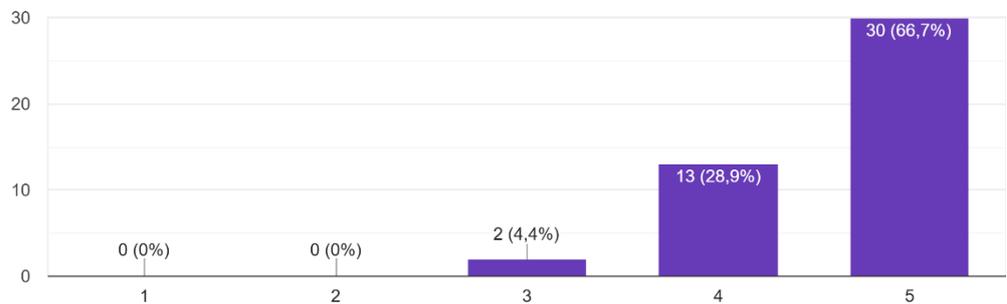
45 jawaban



- **Pertanyaan Ke-2**

Informasi yang disajikan mudah dimengerti

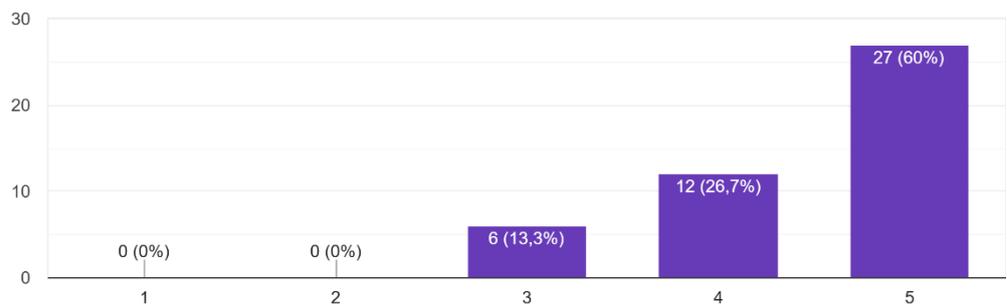
45 jawaban



- **Pertanyaan Ke-3**

Sistem mudah digunakan tanpa pelatihan khusus

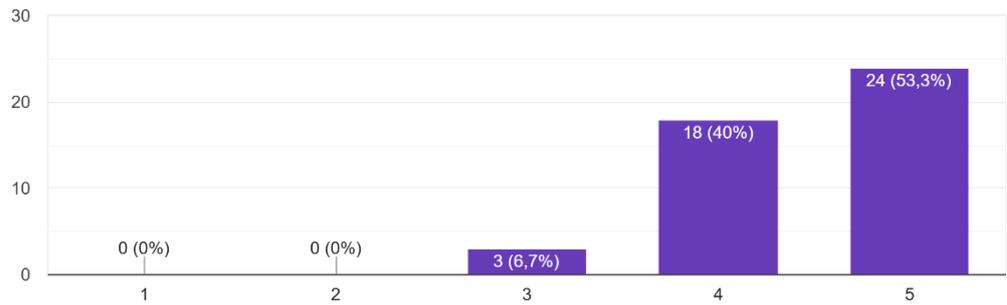
45 jawaban



- **Pertanyaan Ke-4**

Sistem dapat melakukan diagnosa dini TBC dengan benar

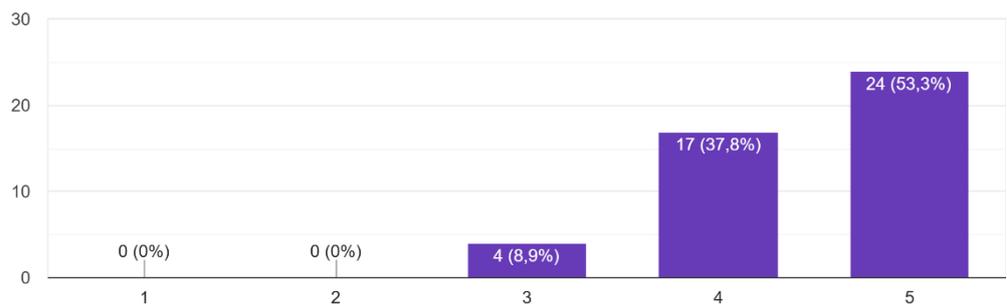
45 jawaban



- **Pertanyaan Ke-5**

Peta lokasi pasien TBC ditampilkan dengan akurat

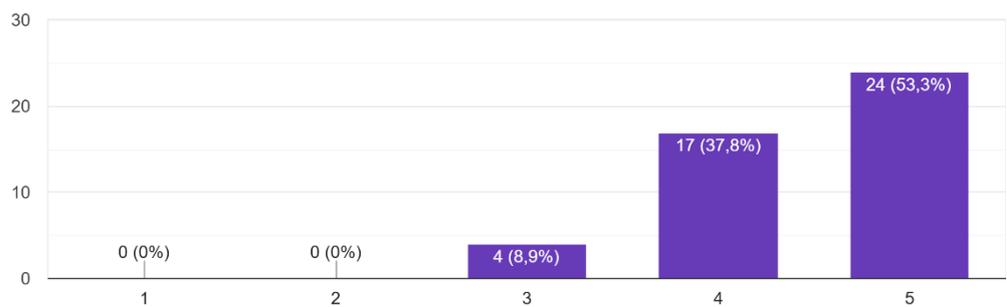
45 jawaban



- **Pertanyaan Ke-6**

Hasil diagnosa berdasarkan input gejala dapat dipercaya

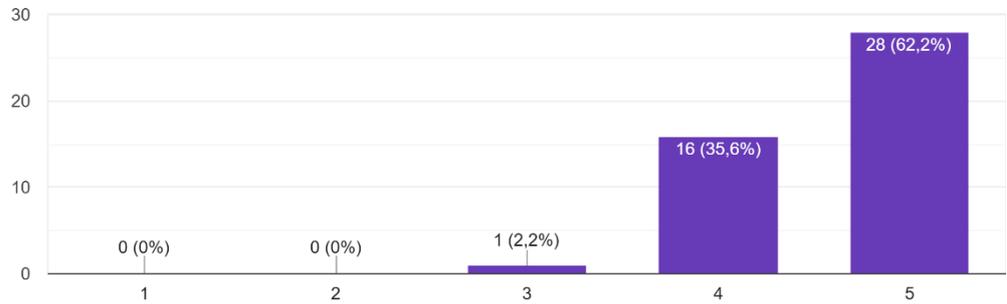
45 jawaban



- **Pertanyaan Ke-7**

Sistem responsif terhadap setiap input pengguna

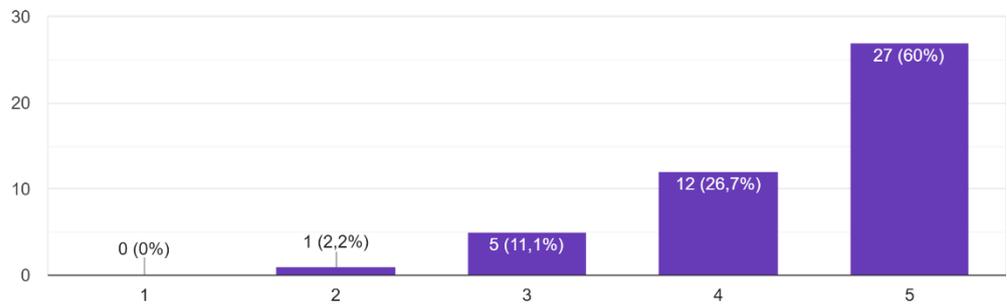
45 jawaban



- **Pertanyaan Ke-8**

Sistem berjalan lancar tanpa error saat digunakan

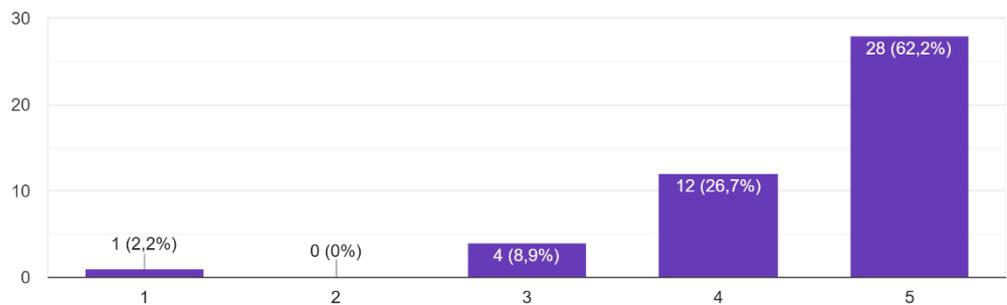
45 jawaban



- **Pertanyaan Ke-9**

Waktu respon sistem cukup cepat

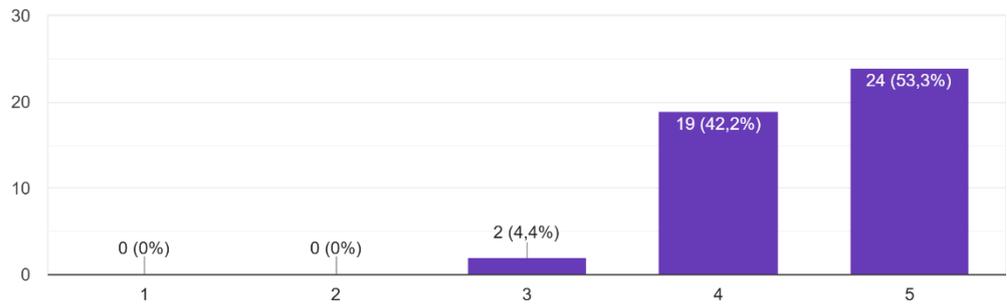
45 jawaban



- **Pertanyaan Ke-10**

Sistem dapat menampilkan data peta dan diagnosa dengan baik

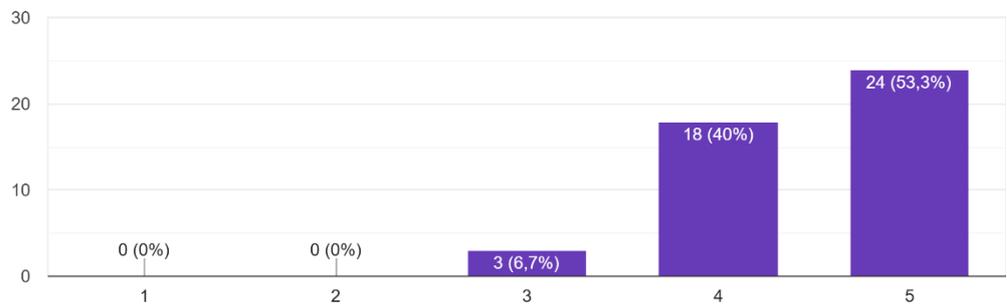
45 jawaban



- **Pertanyaan Ke-11**

Fitur konsultasi dengan admin mudah diakses dan digunakan

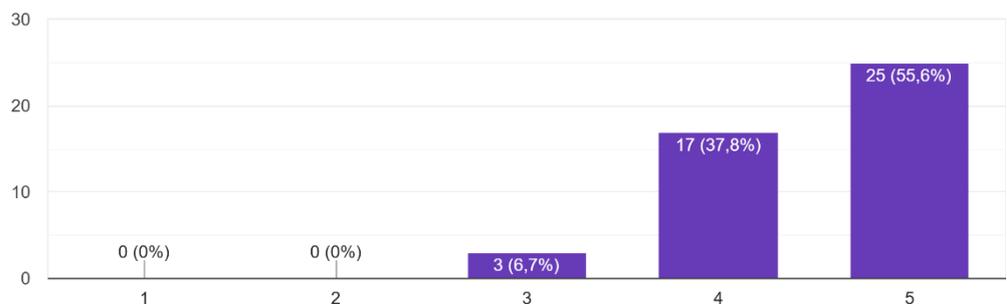
45 jawaban



- **Pertanyaan Ke-12**

Lokasi chatbot mudah dikenali dan ditemukan

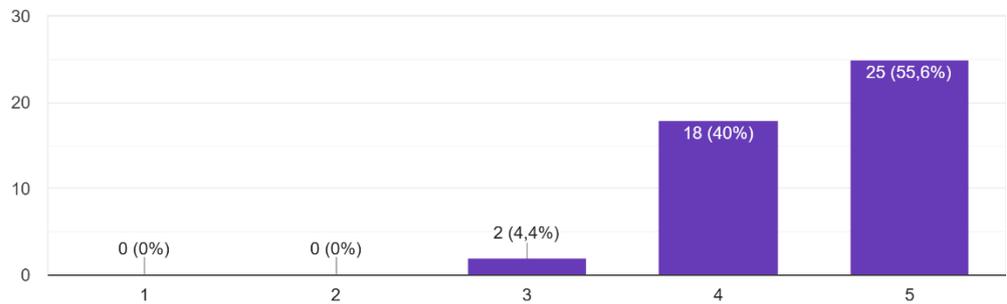
45 jawaban



- **Pertanyaan Ke-13**

Chatbot mampu memberikan jawaban yang relevan

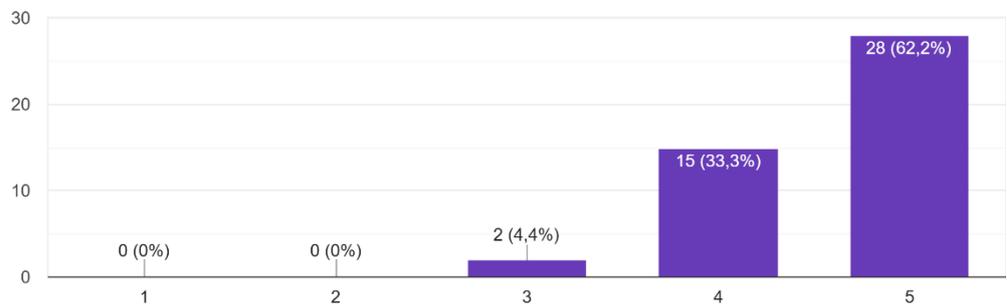
45 jawaban



- **Pertanyaan Ke-14**

Chatbot mudah digunakan dan membantu dalam proses konsultasi awal

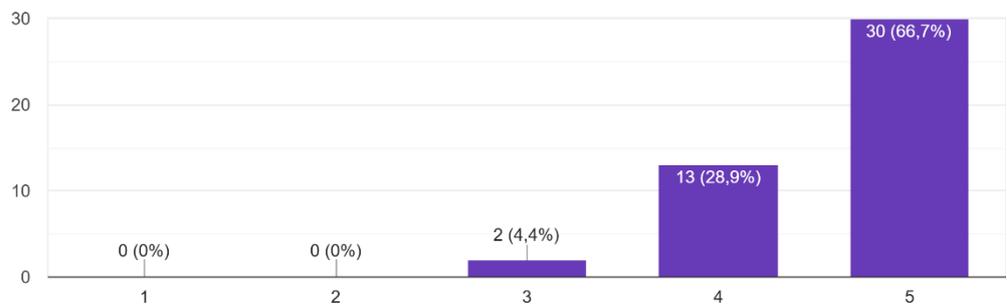
45 jawaban



- **Pertanyaan Ke-15**

Saya merasa terbantu dengan adanya sistem ini

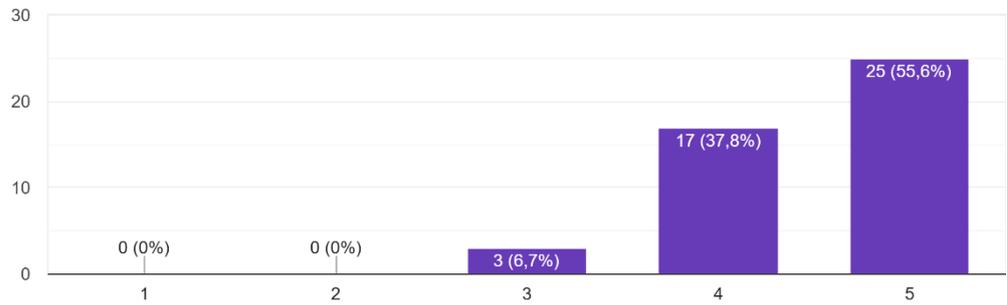
45 jawaban



- **Pertanyaan ke-16**

Sistem ini bermanfaat dalam mendeteksi dini penyakit TBC

45 jawaban



- **Pertanyaan Ke-17**

Saya puas dengan keseluruhan fungsi dari sistem ini

45 jawaban

