

PENGEMBANGAN ARSITEKTUR BACKEND WEB-GIS UNTUK ANALISIS SPASIAL KERAWANAN TUBERKULOSIS DI PROVINSI LAMPUNG DENGAN PENDEKATAN AGILE

Mahendra Pratama^{1*}, Muhammad Dzikri Rofa², Gigih Forda Nama³, Wahyu Eko Sulistiono⁴

^{1,2,3,4}Program Studi Teknik Informatika; Jl. Prof. Dr. Ir. Sumantri Brojonegoro No.1, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung 35141; Telp. (0721) 701609

Keywords:

Tuberkulosis, Web-GIS, Arsitektur Backend, REST API, Agile Development, K6.

Correspondent Email:

mahendra.pratama15@eng.unila.ac.id

Abstrak. Tuberkulosis (TBC) merupakan permasalahan kesehatan global, dengan Indonesia menempati urutan kedua sebagai negara dengan beban kasus tertinggi di dunia. Di Provinsi Lampung, pengendalian TBC menghadapi tantangan distribusi kasus yang tidak merata dan sistem pemantauan yang belum mampu menyajikan pemetaan tingkat kerawanan wilayah. Penelitian ini bertujuan merancang dan mengimplementasikan arsitektur backend Sistem Informasi Tuberkulosis berbasis Web-GIS untuk memetakan tingkat kerawanan berdasarkan parameter epidemiologi, sosiodemografi, dan lingkungan. Pengembangan sistem menggunakan metode *Agile Development* dengan kerangka kerja Scrum. Backend dibangun menggunakan Laravel dan MySQL serta menyediakan layanan *Representational State Transfer* (REST) API. Validasi dilakukan melalui *Black Box Testing*, sedangkan pengujian kinerja menggunakan K6. Hasil pengujian menunjukkan seluruh endpoint API berfungsi dengan tingkat keberhasilan 100%, serta server mampu mempertahankan waktu respons yang rendah hingga beban 300 pengguna virtual secara simultan. Arsitektur Web-GIS yang dihasilkan mampu menyajikan pemetaan interaktif sebagai pendukung pengambilan keputusan dalam mitigasi risiko TBC di tingkat daerah.



Copyright © 2026 The Author(s), Published by Jurnal Informatika dan Teknik Elektro Terapan. This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Abstract. Tuberculosis (TB) remains a major global health problem, with Indonesia ranking second among countries with the highest TB burden worldwide. In Lampung Province, TB control is challenged by uneven case distribution and monitoring systems that do not adequately support spatial vulnerability mapping. This study aims to design and implement a Web-GIS-based tuberculosis information system backend to map regional vulnerability using epidemiological, sociodemographic, and environmental parameters. The system was developed using the Agile Development methodology with the Scrum framework. The backend was implemented using Laravel and MySQL and provides *Representational State Transfer* (REST) API services. Functional validation was conducted through *Black Box Testing*, while performance evaluation employed K6. The results showed that all API endpoints achieved a 100% success rate, and the server maintained low response times under workloads of up to 300 concurrent virtual users. The proposed Web-GIS architecture provides interactive vulnerability mapping to support strategic decision-making for TB risk mitigation at the regional level.

1. PENDAHULUAN

Tuberkulosis (TB) adalah penyakit menular yang disebabkan oleh bakteri *Mycobacterium tuberculosis* dan ditularkan melalui tetesan air liur di udara [1]. Berdasarkan Laporan *Tuberkulosis Global 2024* yang diterbitkan oleh WHO, sekitar 10,7 juta orang akan jatuh sakit akibat tuberkulosis dan lebih dari 1,23 juta orang akan meninggal sepanjang tahun 2024 [2]. Indonesia berada di posisi kedua negara dengan beban TB tertinggi, dengan perkiraan insiden nasional sebesar 1,08 juta kasus per tahun atau 388,1 per 100.000 penduduk berdasarkan Studi Inventarisasi TB 2023–2024 [3].

Di tingkat provinsi, Lampung menghadapi kesenjangan yang cukup tajam dalam pengendalian TBC antar wilayah. Data evaluasi dari Dinas Kesehatan Provinsi Lampung tahun 2024 menunjukkan bahwa hanya dua dari lima belas kabupaten/kota yang memenuhi target Angka Deteksi Kasus (CDR) nasional, yaitu Kota Metro dan Kabupaten Pringsewu. Di sisi lain, delapan kabupaten/kota masih mencatat CDR di bawah 60%, dengan angka terendah di Kabupaten Tanggamus (37,3%) dan Lampung Timur (40,7%). Perbedaan ini mencerminkan masalah sistemik yang tidak hanya terkait dengan kapasitas pelayanan kesehatan, tetapi juga dipengaruhi oleh berbagai faktor lingkungan, sosial, dan demografis di setiap wilayah.

Kerentanan suatu daerah terhadap penyebaran tuberkulosis (TB) dipengaruhi oleh interaksi dinamis antara kondisi lingkungan, sosiodemografi, dan akses terhadap layanan kesehatan, di mana faktor-faktor seperti kepadatan perumahan, kualitas sanitasi, tingkat ekonomi, dan ketersediaan fasilitas medis memainkan peran penting dalam menentukan risiko penularan [4]. Dengan kompleksitas ini, upaya pengendalian TB membutuhkan sistem informasi yang mampu mengintegrasikan dan memvisualisasikan hal ini ke dalam peta spasial. Namun, keterbatasan Sistem Informasi Tuberkulosis (SITB) saat ini dalam menyajikan analisis kerentanan berbasis peta dan mekanisme pertukaran data melalui API standar menciptakan urgensi untuk pengembangan backend berbasis Web-GIS. Inovasi ini merupakan solusi strategis karena memungkinkan para pemangku kepentingan untuk mengakses, mengelola, dan

memvisualisasikan data spasial secara langsung melalui *platform* web interaktif [5].

Penelitian sebelumnya telah menunjukkan bahwa kombinasi kerangka kerja Laravel, REST API, dan metode Pengembangan Agile telah terbukti efektif dalam membangun sistem informasi kesehatan yang efisien dan adaptif [6]. Namun, belum ada penelitian yang secara khusus mengintegrasikan ketiga komponen ini dalam konteks analisis spasial kerentanan TB di Indonesia dengan menggabungkan data epidemiologi, sosiodemografi, dan lingkungan dalam satu arsitektur terpadu. Penelitian ini hadir untuk mengisi kesenjangan tersebut dengan tiga tujuan utama: (1) merancang arsitektur backend yang mampu mengelola data kerentanan TB multidimensi secara terstruktur dan aman, (2) menerapkan Pengembangan Agile berbasis Scrum untuk menghasilkan sistem adaptif, dan (3) membangun REST API standar yang mendukung integrasi backend dan frontend secara efisien.

2. TINJAUAN PUSTAKA

2.1. Tuberkulosis

Tuberkulosis bukan hanya masalah kesehatan tetapi juga produk dari kondisi sosial dan ekonomi serta lingkungan yang ada di dalam suatu masyarakat. Faktor risiko penularan tuberkulosis di Indonesia terdiri dari berbagai aspek, yaitu adalah kepadatan perumahan, kualitas ventilasi perumahan yang buruk, tingkat pendidikan rendah, pendapatan yang tidak mencukupi, riwayat penyakit kronis, dan kesulitan mengakses fasilitas medis. Distribusi spasial dari semua faktor risiko yang disebutkan di atas perlu dianalisis untuk mengembangkan langkah-langkah efektif yang bertujuan untuk mengatasi masalah tuberkulosis.

Untuk mengklasifikasikan wilayah berdasarkan kerentanannya secara komparatif, metode klasifikasi interval sama diterapkan. Metode ini membagi seluruh interval nilai indeks kerentanan (jumlah semua parameter) menjadi empat interval dengan jarak yang sama sangat rendah, rendah, sedang, dan tinggi [7]. Keuntungan dari metode ini termasuk kemudahan penggunaan dan kemungkinan pembaruan data secara dinamis dalam aplikasi perangkat lunak backend secara real-time.

2.2. Web-GIS

Sistem informasi biasanya dianggap sebagai integrasi teknologi informasi dan interaksi manusia yang bertujuan untuk mendukung aktivitas organisasi dan pengambilan keputusan. Sistem tersebut melibatkan perangkat keras, perangkat lunak, basis data, sistem komunikasi, dan personel. Web-GIS, sebagai sebuah konsep, adalah bentuk lanjutan dari sistem informasi yang bertujuan untuk memfasilitasi visualisasi data spasial melalui peramban web. Tidak seperti program GIS tradisional yang memerlukan instalasi perangkat lunak khusus, Web-GIS memungkinkan penggunaan simultan oleh beberapa orang di berbagai tempat tanpa instalasi khusus [8]. Saat ini, sebagian besar aplikasi Web-GIS kontemporer merepresentasikan data spasial dalam format GeoJSON, sebuah struktur berbasis JSON yang ringan dan mudah ditangani oleh peramban web. Backend berfungsi sebagai alat untuk pengelolaan data GeoJSON ini dan membuatnya tersedia melalui titik akhir API; sementara itu, frontend bertanggung jawab untuk memproses data GeoJSON dan menampilkan peta dengan bantuan pustaka khusus seperti Leaflet.js atau Mapbox.

2.3. Laravel dan Backend

Laravel adalah kerangka kerja PHP sumber terbuka berdasarkan arsitektur Model-View-Controller (MVC) yang dikembangkan oleh Taylor Otwell. Kerangka kerja ini menawarkan berbagai fitur produktivitas tinggi seperti Eloquent ORM untuk manajemen basis data yang elegan, sistem routing yang ekspresif, middleware yang fleksibel, dan Artisan CLI untuk otomatisasi tugas pengembangan [9]. Perbandingan kinerja yang dilakukan oleh Chavan dan Pawar [10] menunjukkan Laravel lebih unggul daripada kerangka kerja PHP lainnya dalam hal kecepatan respons dan ketahanan di bawah beban tinggi, menjadikannya pilihan yang tepat untuk backend sistem informasi skala menengah hingga besar.

REST API (Representational State Transfer) adalah gaya arsitektur layanan web yang menggunakan protokol HTTP secara stateless, di mana setiap sumber daya diidentifikasi melalui URI unik dan diakses menggunakan metode HTTP standar (*GET*, *POST*, *PUT*,

DELETE) [17]. Format pertukaran data JSON yang digunakan oleh REST API ringan dan mudah diuraikan oleh berbagai bahasa pemrograman, sehingga mendukung interoperabilitas yang tinggi antara backend dan berbagai klien frontend serta aplikasi seluler [11].

2.4. Pengembangan Agile dengan Scrum

Asal mula Pengembangan Agile berasal dari kekurangan yang terkait dengan metode tradisional seperti pendekatan Waterfall yang linier dan tidak fleksibel. Yang membedakan Agile adalah bahwa ia melibatkan pengembangan perangkat lunak fungsional secara bertahap menggunakan iterasi sambil memasukkan umpan balik pengguna setiap saat selama proses pengembangan [12]. Sebuah studi perbandingan empiris mengungkapkan bahwa pendekatan Agile selalu memberikan tingkat kepuasan pengguna dan kualitas yang lebih tinggi dibandingkan dengan metode pengembangan tradisional, terutama dalam kasus di mana terdapat perubahan persyaratan [13]. Sebagai pendekatan paling populer untuk pengembangan Agile, Scrum melibatkan pengorganisasian pengembangan ke dalam iterasi yang dikenal sebagai sprint, yang biasanya berlangsung antara 1 hingga 4 minggu.

2.5. Black Box dan K6

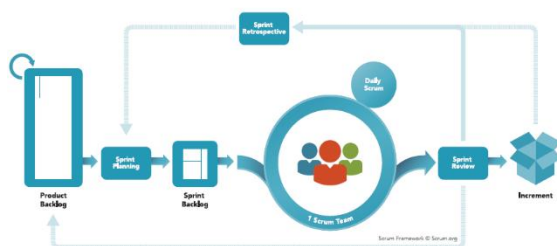
Pengujian *black box* mengevaluasi fungsionalitas suatu sistem semata-mata dari perspektif pengguna akhir tanpa memeriksa kode internal, dengan menyediakan berbagai skenario input dan memverifikasi kesesuaian output dengan spesifikasi yang ditentukan [14]. Dalam *REST API*, setiap *endpoint* diuji terhadap skenario input valid, tidak valid, dan valid menggunakan alat seperti *Postman*, yang memungkinkan pembuatan set pengujian terstruktur dan dapat direproduksi.

K6 adalah alat pengujian beban berbasis JavaScript yang mampu mensimulasikan ratusan pengguna virtual dan menghasilkan laporan metrik dengan waktu respons p50, p90, p95, p99, *throughput* (req/s), dan tingkat kesalahan [15]. Pengujian Kinerja dengan K6 bertujuan untuk mengukur kemampuan sistem dalam menangani beban permintaan yang tinggi. Hasil pengujian ini untuk memastikan sistem dapat beroperasi dengan benar.

3. METODE PENELITIAN

3.1. Siklus Pengembangan Agile Scrum

Penelitian rekayasa perangkat lunak yang telah dilakukan telah dilakukan secara terorganisir berdasarkan metodologi pengembangan Agile yang mengikuti pendekatan Scrum untuk memastikan bahwa pengembangan sistem informasi kesehatan dapat beradaptasi dengan situasi yang berubah-ubah [16], [17]. Dapat dilihat dari gambar 1 bahwa penelitian rekayasa perangkat lunak ini melibatkan serangkaian tahapan manajemen seperti persiapan *Product Backlog* untuk mengidentifikasi semua kebutuhan yang berkaitan dengan aplikasi web-gis yang mencakup modul autentikasi, manajemen GeoJSON serta algoritma pembobotan kerentanan TBC. Setelah mendefinisikan kebutuhan, tim akan beralih ke *Sprint Planning* untuk menetapkan prioritas di antara modul yang harus dikembangkan dalam satu siklus iterasi atau sprint. Dari segi pengembangan teknis, fase Eksekusi Sprint mencakup semua operasi teknis termasuk pengkodean backend berbasis Laravel. Sinkronisasi kemajuan dan penyelesaian tantangan teknis akan terjadi selama pertemuan *Daily Scrum*. Adapun aktivitas penutup, *Sprint Review* digunakan untuk menilai fungsionalitas API sedangkan *Sprint Retrospective* adalah waktu refleksi internal.



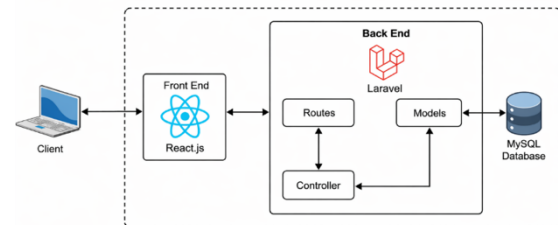
Gambar 1. Alur metode Agile Scrum.

3.2. Pemodelan Sistem

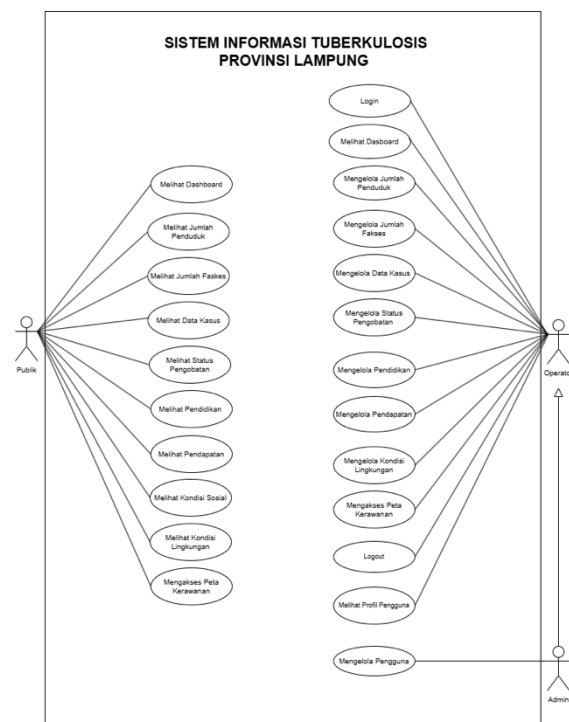
Seperti yang ditunjukkan pada Gambar 2, backend ini telah dibuat dengan menggunakan bahasa PHP bersama dengan *framework* Laravel, dengan *database* MySQL bertindak sebagai sistem manajemen database backend.

Laravel dipilih karena ketersediaan Eloquent ORM, yang memudahkan untuk membangun hubungan antar data medis. Selain itu, mekanisme routing API Laravel

menjadikannya pilihan ideal untuk mengarahkan lalu lintas data spasial secara efisien [18]. Semua logika yang terlibat dalam menghitung indeks kerentanan dan memproses data spasial terpusat di lapisan backend ini.



Gambar 2. Arsitektur Sistem Backend Berbasis Web-GIS.



Gambar 3. Use Case Diagram Sistem Informasi Tuberculosis Provinsi Lampung.

Use Case Diagram pada Gambar 3 menunjukkan sistem yang melibatkan tiga aktor utama yaitu Publik, Operator, dan Admin. Aktor Publik memiliki akses untuk melihat informasi yang disajikan oleh sistem, termasuk data kasus tuberkulosis, populasi, jumlah fasilitas kesehatan, faktor risiko sosial dan lingkungan, dan peta kerentanan regional berbasis Web-GIS. Pengguna juga dapat mengakses hasil analisis kerentanan, yang ditampilkan dalam visualisasi spasial.

Aktor Operator bertanggung jawab untuk mengelola data operasional sistem. Hak akses mereka meliputi penambahan, modifikasi, dan penghapusan data kasus tuberculosis, status pengobatan, populasi, jumlah fasilitas kesehatan, pendidikan, pendapatan, kondisi lingkungan, dan data analisis kerentanan regional. Operator juga dapat mengakses dasbor dan memperbarui data yang digunakan dalam proses analisis spasial.

Aktor Admin memiliki semua hak akses Operator, ditambah wewenang tambahan untuk mengelola pengguna. Fungsi ini meliputi penambahan, modifikasi, dan penghapusan akun pengguna, serta pengaturan hak akses sistem. Mekanisme ini memungkinkan kontrol terpusat atas keamanan dan manajemen sistem.

3.3. Pemodelan Algoritma Indeks Kerentanan

Untuk membuat peta kerentanan, tidak cukup hanya memetakan jumlah kasus; hal ini melibatkan perhitungan indeks kerentanan (*Vulnerability Index*) dengan faktor multivariabel. Logika untuk perhitungan ini dilakukan melalui *level Controller* dalam arsitektur Laravel.

Faktor-faktor yang dikuantifikasi adalah sebagai berikut yang pertama Tingkat Notifikasi Kasus TB, kedua Kepadatan Penduduk, ketiga Proporsi Rumah Sehat, dan keempat Status Sosial Ekonomi/Pendapatan. Untuk setiap faktor, bobot signifikansi tertentu (*W*) dihitung sesuai dengan tinjauan literatur medis [19]. Rumus akhir untuk indeks kerentanan adalah sebagai berikut:

$$I = \sum_{i=1}^n (W_i \times X_i)$$

Di mana W_i adalah bobot yang diberikan pada kriteria ke- i , sedangkan X_i menunjukkan nilai standar dari kriteria ke- i di wilayah tersebut. Indeks Kerentanan dihasilkan melalui proses penggabungan kriteria standar menggunakan bobot [20]. Output selanjutnya diklasifikasikan ke dalam tingkat kerentanan (Rendah, Sedang, Tinggi) melalui klasifikasi interval yang sama.

4. HASIL DAN PEMBAHASAN

4.1. Konstruksi Backend dan Integrasi Spasial

Melalui API berbasis Laravel, sistem melakukan agregasi data, penilaian lingkungan, dan pembuatan GeoJSON, memungkinkan visualisasi grafis dan peta interaktif yang efisien di bagian depan. Sistem backend berhasil mengintegrasikan data penentu *tuberculosis* dengan data administrasi spasial di Provinsi Lampung.

```

10 Schema::create('lingkungan', function (Blueprint $table) {
11     $table->softDeletes();
12     $table->uuid('id')->primary();
13     $table->uuid('id_wilayah');
14     $table->uuid('id_status');
15     $table->integer('presentase');
16     $table->timestamp('tanggal_ditemukan')->nullable();
17     $table->integer('tahun_ditemukan');
18
19     // audit
20     $table->uuid('id_creator');
21     $table->uuid('id_updater')->nullable();
22     $table->timestamp('tanggal_create')->nullable();
23     $table->timestamp('last_update')->nullable();
24
25     $table->unique(
26         ['id_wilayah', 'tahun_ditemukan', 'id_status'],
27         'uniq_lingkungan_full'
28     );
29
30     // FK
31     $table->foreign('id_wilayah')->references('id')->on('wilayah');
32     $table->foreign('id_status')->references('id')->on('status');
33     $table->foreign('id_creator')->references('id')->on('users');
34     $table->foreign('id_updater')->references('id')->on('users');
35 });
36

```

Gambar 4. Skema Basis Data untuk Data Lingkungan

Gambar 4 adalah struktur basis data yang digunakan untuk menyimpan data kondisi lingkungan dan hubungannya dengan wilayah administratif dan kategori status.

```

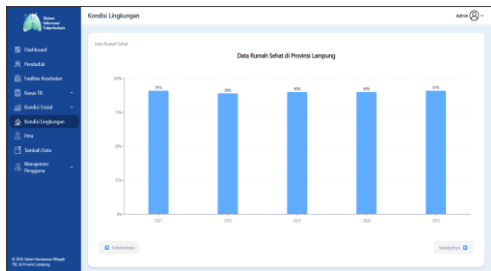
21 public function tahun()
22 {
23     $keys = [
24         'jml_kasus',
25         'jml_penduduk',
26         'jml_rumah_sehat',
27         'jml_pendapatan'
28     ];
29     $weight = 100 / count($keys);
30     $data = $this->lingkungan->query()
31         ->select('wilayah as kab, lingkungan.id, wilayah as kab, kab.id')
32         ->where('kab.level', 'Kabupaten')
33         ->select()
34         ->orderBy('YEAR(lingkungan.tanggal_ditemukan) as tahun',
35             'kab.id',
36             'lingkungan.id_status' => '1-001' THEN lingkungan.presentase ELSE 0 END) as jml_kasus,
37             'lingkungan.id_status' => '1-002' THEN lingkungan.presentase ELSE 0 END) as jml_penduduk,
38             'lingkungan.id_status' => '1-003' THEN lingkungan.presentase ELSE 0 END) as jml_rumah_sehat,
39             'lingkungan.id_status' => '1-004' THEN lingkungan.presentase ELSE 0 END) as jml_pendapatan,
40             'lingkungan.id_status' => '1-005' THEN lingkungan.presentase ELSE 0 END) as persentase);
41     $grouped = $data->groupBy('kab.id', DB::raw('YEAR(lingkungan.tanggal_ditemukan)'));
42     $grouped->get();
43
44     $scores = $grouped->map(function ($rows) use ($weight) {
45         $return = [
46             'jml_kasus' => round($rows->jml_kasus * $weight) / 100,
47             'jml_penduduk' => round($rows->jml_penduduk * $weight) / 100,
48             'jml_rumah_sehat' => round($rows->jml_rumah_sehat * $weight) / 100,
49             'jml_pendapatan' => round($rows->jml_pendapatan * $weight) / 100,
50             'persentase' => round($rows->persentase * $weight) / 100;
51         ];
52         return $return;
53     });
54     $return = [
55         'tahun' => $rows->first()->tahun,
56         'total' => round($scores->avg()),
57     ];
58     return response()->json($return);
59 }

```

Gambar 5. Proses Agregasi Data Lingkungan

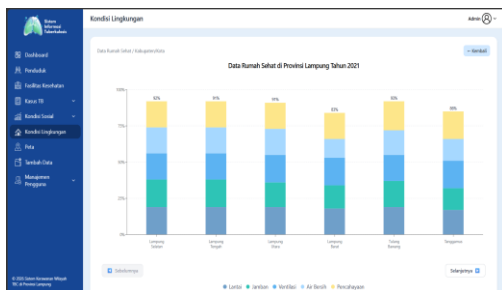
Gambar 5 adalah algoritma backend yang digunakan untuk menghitung persentase

kondisi lingkungan dan ringkasan tahunan untuk analisis.



Gambar 6. Grafik Tren Kondisi Lingkungan

Gambar 6 menampilkan skor kondisi lingkungan tahunan di Provinsi Lampung yang dihasilkan dari data API yang diagregasi.



Gambar 7. Grafik Distribusi Indikator Lingkungan

Gambar 7 mengilustrasikan proporsi setiap komponen indikator lingkungan di berbagai tahun menggunakan grafik batang bertumpuk.

```

35  * =====
36  * MODE PROGRESSIVE (DB)
37  * =====
38  */
39  if ($mode === 'progressive') {
40      $tahun = request()->get('tahun', date('Y'));
41      $this->getKerawananWilayah($tahun); Too many
42  }
43  $limit = request()->get('limit', 10);
44  $page = request()->get('page', 1);
45  $offset = ($page - 1) * $limit;
46
47  $rows = DB::select("
48      SELECT
49          p.id,
50          ST_AsGeoJSON(p.geom) AS geometry,
51          kab.uid AS kabupaten_id,
52          kab.name AS kabupaten
53      FROM peta_kabupaten p
54      JOIN wilayah_kab ON kab.uid = p.wilayah_id
55      ORDER BY p.id
56      LIMIT ? OFFSET ?
57      ", [$limit, $offset]);
58
59  $features = array_map(fn($row) => [
60      "type" => "Feature",
61      "geometry" => json_decode($row->geometry),
62      "properties" => [
63          "id" => $row->id,
64          "kabupaten_id" => $row->kabupaten_id,
65          "kabupaten" => $row->kabupaten,
66      ],
67      $rows);
    
```

Gambar 8. Layanan Pembentukan Data GeoJSON

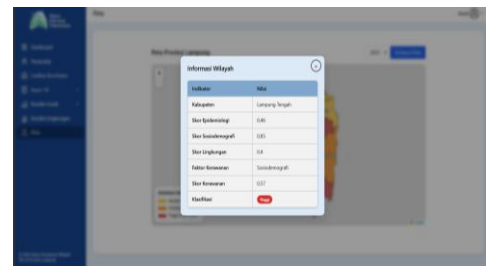
Gambar 8 menunjukkan layanan backend yang mengkonversi data spasial dari basis data

ke format GeoJSON untuk visualisasi Web-GIS.



Gambar 9. Visualisasi Peta Kerawanan Wilayah

Gambar 9 menunjukkan distribusi spasial tingkat kerawanan tuberkulosis di berbagai kabupaten di provinsi Lampung.

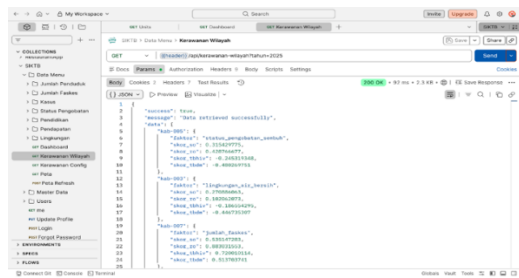


Gambar 10. Popup Informasi Wilayah

Gambar 10 menyajikan informasi detail kabupaten yang ditampilkan ketika pengguna memilih suatu wilayah pada peta interaktif.

4.2. Validasi Fungsionalitas REST API

Uji verifikasi yang dilakukan melalui pengujian *BlackBox* menggunakan *Postman* memberikan hasil yang sukses dan memenuhi persyaratan desain sistem. Modul token otentikasi yang disebut sebagai *middleware* telah dikonfirmasi mampu mengamankan lalu lintas apa pun dengan membatasi akses tidak sah ke dalamnya. Implementasi operasi CRUD untuk semua modul yang mengelola faktor risiko serta pengaturan bobot di wilayah tersebut juga berhasil dan tidak menyebabkan kerusakan sistem selama proses pengujian.



Gambar 11. Hasil Pengujian Fungsionalitas API menggunakan Postman.

TABEL I. Hasil pengujian fungsionalitas API.

Deskripsi Pengujian	Data Pengujian	Hasil yang Diharapkan	Status
Pengujian menampilkan data kerawanan wilayah	GET /api/kerawanan-wilayah Year valid	200 "Data kerawanan wilayah berhasil ditampilkan"	Valid
Pengujian menampilkan konfigurasi legenda dinamis	GET /api/kerawanan-config	200 "Konfigurasi legenda berhasil ditampilkan"	Valid
Pengujian klasifikasi kerawanan menggunakan metode <i>Equal Interval</i>	GET /api/kerawanan-config	200 "Klasifikasi <i>Equal Interval</i> berhasil ditampilkan"	Valid
Pengujian menampilkan faktor dominan kerawanan	GET /api/kerawanan-wilayah	200 "Faktor dominan berhasil ditampilkan"	Valid
Pengujian integrasi <i>heatmap</i> dengan hasil analisis kerawanan	GET /api/kerawanan-wilayah	200 " <i>Heatmap</i> kerawanan berhasil ditampilkan"	Valid
Pengujian sinkronisasi data legenda dengan hasil klasifikasi	GET /api/kerawanan-config	200 "Legenda dinamis berhasil menyesuaikan data kerawanan"	Valid

Hasil evaluasi yang disajikan dalam Tabel I dan Gambar 11 menunjukkan bahwa semua modul fungsional backend memiliki tingkat kepatuhan arsitektur 100%. Endpoint yang dilindungi melalui Laravel Sanctum dapat secara efektif memblokir akses tidak sah ke aplikasi (HTTP 401 *Unauthorized*) dan dengan demikian menjamin perlindungan data pasien TB.

Penanganan kesalahan juga seragam. Semua input data yang tidak valid akan secara otomatis ditolak melalui lapisan Validasi Laravel,

memberikan respons HTTP yang sesuai (422 *Unprocessable Entity*) dan pesan terperinci mengenai masalah ini.

4.3. Analisis Stabilitas Kinerja

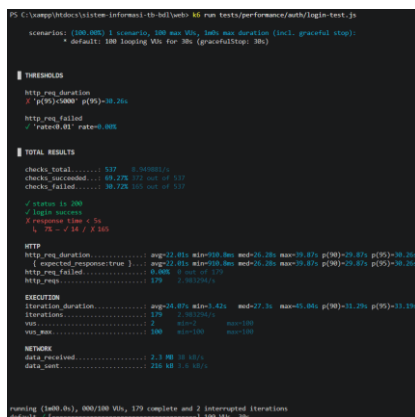
Analisis ketahanan sistem berdasarkan beban kerja eksponensial juga dinilai melalui pengujian praktis menggunakan modul alat pengujian beban K6. Simulasi dilakukan dengan mengirimkan lalu lintas secara bersamaan melalui 100, 200, dan 300 pengguna virtual (VU) terhadap titik akhir yang bertugas menyiapkan peta kerentanan regional. Hasil pengujian ini menunjukkan stabilitas sistem yang sangat baik sejauh menyangkut produksi tanpa kegagalan atau kehilangan permintaan HTTP dengan tingkat keberhasilan sempurna 100%. Waktu respons server juga efisien dalam kisaran milidetik.

Tabel II. Daftar Endpoint API dalam Pengujian Kinerja.

No	Endpoint	Method	Fungsi
1	/api/login	POST	Proses autentikasi pengguna
2	/api/dashboard	GET	Mengambil data <i>dashboard</i> sistem
3	/api/peta	GET	Mengambil data <i>GeoJSON</i> peta kerawanan
4	/api/kerawanan-wilayah	GET	Mengambil data kerawanan wilayah
5	/api/kerawanan-config	POST	Mengambil data kerawanan config
6	/api/kasus-chart/detail	POST	Mengambil detail grafik kasus <i>TBC</i>
7	/api/pendapatan-chart/detail	POST	Mengambil detail grafik pendapatan
8	/api/lingkungan-chart/detail	POST	Mengambil detail grafik kondisi lingkungan
9	/api/jumlah-penduduk	POST	Menambahkan data jumlah penduduk
10	/api/jumlah-faskes/chart/detail	POST	Mengambil detail grafik jumlah faskes
11	/api/jumlah-penduduk/chart/detail	POST	Mengambil detail grafik jumlah penduduk
12	/api/kasus	POST	Menambahkan data kasus <i>TBC</i>

13	/api/pendapatan	<i>POST</i>	Menambahkan data pendapatan
14	/api/lingkungan	<i>POST</i>	Menambahkan data kondisi lingkungan
15	/api/users	<i>GET</i>	Mengambil data pengguna

Dari analisis data di atas yang terdapat pada Tabel II, terlihat jelas bahwa arsitektur backend yang dirancang berkinerja lebih baik dibandingkan arsitektur lainnya. Dengan beban puncak 300 VU, server dapat melayani hingga 610,8 RPS sambil memastikan 0,00% kegagalan dalam transaksi data. Kinerja ini disebabkan oleh manajemen memori server yang efisien oleh PHP-FPM serta penggunaan optimasi indeks untuk basis data relasional MySQL. Meskipun terjadi peningkatan waktu respons server dari 112 ms menjadi 298 ms, penting untuk dicatat bahwa kecepatan 298 ms tersebut cukup cepat dan jauh lebih rendah dari 2000 ms. Latensi P95 tercatat 340 ms pada 300 VU, memastikan bahwa 95% dari semua pengguna menerima data peta dalam sepertiga detik.



Gambar 12. Hasil Pengujian Kinerja menggunakan alat uji K6.

Seperti yang ditunjukkan pada Gambar 12, terdapat cukup banyak manfaat operasional yang dihasilkan dari integrasi sistem tersebut. Secara khusus, penggunaan visualisasi peta kerentanan interaktif yang dimungkinkan berkat keandalan backend yang disebutkan telah berhasil menyederhanakan pemahaman data kompleks tentang penyakit menular dan mengubahnya menjadi alat spasial yang berharga. Akibatnya, Layanan Kesehatan dapat memiliki peluang yang jauh lebih baik dalam

hal menganalisis area yang membutuhkan intervensi medis, mengidentifikasi tempat-tempat di mana terdapat titik-titik kosong kejadian penyakit, dan mendistribusikan sumber daya dengan cara yang lebih terencana.

5. KESIMPULAN

Penelitian ini secara empiris menunjukkan bahwa perancangan arsitektur backend Web-GIS menggunakan pendekatan Agile Scrum dapat menghilangkan keterbatasan fungsional sistem pelaporan TB konvensional. Sistem ini berhasil mengorkestrasi dimensi epidemiologi, lingkungan, dan sosiodemografi ke dalam format data spasial terstruktur yang didistribusikan melalui REST API standar GeoJSON. Algoritma pembobotan multivariabel secara akurat merepresentasikan tingkat kerentanan regional. Pengujian kinerja K6 memvalidasi ketahanan server di bawah beban kerja berat tanpa kegagalan layanan. Visualisasi kerentanan spasial platform memberikan perspektif analitis yang unggul untuk pengambilan keputusan eliminasi TB di Provinsi Lampung.

Sebagai arah strategis untuk penelitian lebih lanjut, integrasi arsitektur komputasi awan dengan penskalaan otomatis direkomendasikan untuk mengantisipasi akumulasi data kesehatan yang besar. Implementasi modul kecerdasan buatan berbasis pembelajaran mesin untuk peramalan deret waktu sangat direkomendasikan untuk meningkatkan kemampuan sistem dari instrumen deskriptif menjadi prediktif.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] W. S. Budi, M. Raharjo, Nurjazuli, dan S. Poerwati, "Hubungan Kualitas Lingkungan Fisik Rumah dengan Kejadian Tuberkulosis di Kecamatan Panekan" *Media Publikasi Promosi Kesehatan Indonesia.*, vol. 7, no. 4, pp. 1012–1018, 2024.
<https://doi.org/10.56338/mppki.v7i4.5106>
- [2] World Health Organization, *Global Tuberculosis Report 2024*. Geneva: WHO, 2024.

- <https://www.who.int/publications/i/item/9789240101531>
- [3] O. S. Simarmata et al., *Laporan Hasil Studi Inventori Tuberkulosis Indonesia 2023–2024*. Jakarta: Kemenkes RI, 2024. <https://tbindonesia.or.id/>
 - [4] R. Singh, P. Singh, K. Srivastava, A. Kumar, A. Sharma, and S. Kant, "Environmental and socioeconomic factors affecting tuberculosis," *Infection*, vol. 11, p. 12, 2025. <https://doi.org/10.18231/j.ijirm.12670.1764326>
 - [5] S. Banerjee, C. E. Fullerton, S. S. Gaharwar, and E. J. Jaselskis, "Strategic Web-Based Data Dashboards as Monitoring Tools for Promoting Organizational Innovation," *Buildings*, vol. 15, no. 13, p. 2204, 2025. <https://doi.org/10.3390/buildings15132204>
 - [6] P. C. Dhage, R. A. Thakker, and K. K. Warhade, "Laravel Technology based Maternal e-Healthcare Systems with Improved Response Time and Stability," *International Research Journal of Multidisciplinary Technovation*, vol. 7, no. 3, pp. 326–344, 2025. <https://doi.org/10.54392/irjmt25324>
 - [7] F. Huang, Y. Yang, B. Jiang, Z. Chang, C. Zhou, S. H. Jiang, et al., "Effects of Different Division Methods of Landslide Susceptibility Levels on Regional Landslide Susceptibility Mapping," *Bulletin of Engineering Geology and the Environment*, vol. 84, no. 6, p. 276, 2025. <https://doi.org/10.1007/s10064-025-04281-4>
 - [8] Q. Geng, X. Yao, J. Sun, S. Jia, F. Xu, L. Zhang, et al., "Open-Source GIS Software and Tools: A Systematic Review," *Big Earth Data*, pp. 1–48, 2026. <https://doi.org/10.1080/20964471.2026.2666947>
 - [9] P. P. Ray, "A Survey on Model Context Protocol: Architecture, State-of-the-Art, Challenges and Future Directions," *Authorea Preprints*, 2025. <https://doi.org/10.36227/techrxiv.174495492.2752319/v1>
 - [10] P. R. Chavan dan S. Pawar, "Comparison Study Between Performance of Laravel and Other PHP Frameworks," *IJRESM*, vol. 4, no. 10, pp. 27–29, 2021. <https://www.ijresm.com>
 - [11] F. Prasetyo et al., "Trends and Best Practices in API-Based Web Development Using Laravel and React," *Brilliance: Research of Artificial Intelligence*, vol. 5, no. 1, pp. 223–233, 2025. <https://doi.org/10.47709/brilliance.v5i1.5971>
 - [12] H. Fawareh and S. I. Ishaq, "Design Thinking Integrated Within Agile Software Development Approach," *Artificial Intelligence for Sustainable Innovation Management and Risk Management*, 2026, vol. 227, pp. 1999–2010. https://doi.org/10.1007/978-3-031-95310-1_145
 - [13] S. G. Tetteh, "Empirical Study of Agile Software Development Methodologies: A Comparative Analysis," *Asian Journal of Research in Computer Science*, vol. 17, no. 5, pp. 30–42, 2024. <https://doi.org/10.9734/ajrcos/2024/v17i5436>
 - [14] Z. M. Alshamaa and N. N. Saleem, "Black Box Software Testing Techniques: A Literature Review," *Passer Journal of Basic and Applied Sciences*, vol. 7, no. 2, pp. 1001–1011, 2025. <https://doi.org/10.24271/psr.2025.508216.1963>
 - [15] I. Nakamura, K. Horikawa, B. Reid, Y. Kashiwa, and H. Iida, "An Empirical Investigation into Maintenance of Load Testing Scripts," in *2025 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2025, pp. 331–337. <http://doi.org/10.1109/ESEM64174.2025.00031>
 - [16] M. Riesener et al., "Framework for an Agile, Databased Development," *Procedia CIRP*, vol. 100, pp. 343–348, 2023. <https://doi.org/10.1016/j.procir.2021.05.084>
 - [17] K. Singh, "Agile Methodology for Product Development: A Conceptual Study," *International Journal of Recent Technology and Engineering (IJRTE)*, *Eng.*, no. 1, pp. 209–215, 2021. <https://doi.org/10.35940/ijrte.A5899.0510121>
 - [18] A. Murod et al., "Rancang Bangun Sistem Informasi Manajemen Persediaan Barang Menggunakan Framework Laravel," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, pp. 2210–2219, 2024. <http://dx.doi.org/10.23960/jitet.v12i3.4706>
 - [19] S. M. Aveda, and R. Anggraini "Analisis Spasial Kasus Tuberkulosis di Kabupaten Lampung Barat Tahun 2023," *Jurnal Kesehatan Masyarakat*, vol. 11, no. 2, pp. 122–133, 2023, doi: <http://dx.doi.org/10.31602/ann.v11i2.16621>
 - [20] N. S. Sauti, M. E. Daud, M. Kaamin, and S. Sahat, "Development of an Exposure Vulnerability Index Map Using GIS Modeling for Preliminary Seismic Risk Assessment in Sabah, Malaysia," *International Journal of Design & Nature and Ecodynamics*, vol. 16, no.

1, pp. 111–119, 2021. [Online]. Available:
<https://doi.org/10.18280/ij dne.160115>