

Arsitektur Geospasial API dengan *Laravel* untuk Sistem Deteksi Kesesuaian Lahan Pertanian

Geospasial API Architecture with Laravel for Agricultural Land Suitability Detection System

Yerymia Alfa Susetyo

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jalan Dr. O. Notohamidjojo 1-10, Blotongan, Sidorejo, Kota Salatiga, Jawa Tengah, Indonesia
e-mail: yerymia.alfa@uksw.edu

(*received*: 15 April 2023, *revised*: 26 April 2023, *accepted*: 3 Juli 2023)

Abstrak

Identifikasi kesesuaian lahan pertanian melibatkan berbagai variabel yang bersifat heterogen. Heterogenitas data iklim dan fisiografi berbasis spasial pada faktanya cukup kompleks untuk diselesaikan. Oleh sebab itu, diperlukan arsitektur sistem berbasis spasial yang memenuhi kriteria inklusivitas, kolaborasi, pengembangan kapasitas, dan request-response yang cepat. Penelitian ini bertujuan membangun arsitektur geospasial API dengan *Laravel* untuk Sistem Deteksi Kesesuaian Lahan Pertanian. Arsitektur geospasial API dalam penelitian ini dibangun menggunakan RESTful Web services pada Framework *Laravel*. Simulasi arsitektur melibatkan lima node sebagai server dan satu node sebagai client. Enam API utama dihasilkan pada penelitian ini. Empat services dihasilkan dari empat sever, dimana services tersebut adalah services yang berhubungan dengan data spasial (wilayah, ketinggian, kelerengan, dan curah hujan). Sementara itu, dua services lainnya, berhubungan dengan data konvensional informasi zonasi kesesuaian lahan pertanian dihasilkan oleh Server 5. Service yang dihasilkan oleh server terakhir berhasil diimplementasikan pada client application berbasis web peta interaktif.

Kata kunci: kesesuaian lahan, geospasial API, RESTful, web services, *Laravel*

Abstract

Identification of agricultural land suitability involves a variety of variables that are heterogeneous. The heterogeneity of spatially-based climate and physiographic data is in fact quite complex to solve. Therefore, a spatial-based system architecture that meets the criteria of inclusiveness, collaboration, capacity development, and quick request-response is needed. The research aims to build an API geospasial architecture with Laravel for Agricultural Land Suitability Detection Systems. The geospasial API architecture in this study was built using RESTful Web services on the Laravel Framework. Simulation architecture involves five nodes as a server and one node as a client. Six main API were produced in this study. Four services are derived from four sever, where services are services related to spatial data (area, altitude, slope, and rainfall). Meanwhile, two other services, relating to conventional information zoning of agricultural land suitability generated by Server 5. The service generated by the last server was successfully implemented on the client-based web-based interactive map application.

Keywords: land suitability, geospasial API, RESTful, web services, *Laravel*

1 Pendahuluan

Sektor pertanian memiliki peranan penting dalam membangun negara karena tidak hanya memberikan dampak ekonomi yang signifikan, tetapi juga diharapkan dapat berkontribusi dalam pembangunan lingkungan dan sosial masyarakat. Untuk mencapai tujuan positif pada ketiga sektor tersebut, yaitu ekonomi, lingkungan, dan masyarakat, maka perencanaan pembangunan pertanian harus dilakukan dengan menyeluruh dan matang. Adanya identifikasi kesesuaian lahan dalam perencanaan

sistem pertanian dan tata guna lahan akan membantu meningkatkan efektivitas produksi pangan [1]. Direktorat Jenderal Tanaman Pangan, mengidentifikasi bahwa peningkatan produksi pertanian dapat dilakukan dengan perluasan area dan optimasi lahan [2]. Optimasi lahan pertanian merupakan faktor penting dalam suksesnya sistem pertanian. Dalam perencanaan sistem pertanian diperlukan satu panduan yang jelas yaitu peta kesesuaian lahan yang diterapkan dengan bantuan teknologi informasi, sehingga memiliki dinamika yang lebih fleksibel [3]. Penyusunan peta kesesuaian lahan dapat menerapkan korelasi faktor klimatologi dan fisiologi suatu wilayah. Komponen Ada beberapa faktor yang terdiri dari jenis tanah, curah hujan, fisiografi lahan, dan ketinggian.

Perencanaan sistem pertanian perlu melihat potensi-potensi atau karakteristik suatu wilayah. Karakteristik suatu wilayah secara detail disajikan dalam bentuk peta [4], seperti peta curah hujan, peta jenis tanah, peta kelerengan, dan peta drainase. Objek-objek tersebut merupakan objek spasial, yang secara teknologi membutuhkan perlakuan khusus dalam pengelolaannya. Pemerintah Daerah memerlukan pendekatan yang inovatif dan informatif berbasis teknologi digital dalam mengelola dan menyalurkan informasi potensi kesesuaian lahan untuk perencanaan sistem pertanian kepada masyarakat luas. Sistem Informasi Geografis merupakan sistem atau infrastruktur teknologi yang memfasilitasi pemrosesan, manipulasi, dan/atau visualisasi data spasial, termasuk teknologi berbasis desktop dan web [5].

Pembangunan aplikasi berbasis spasial memerlukan dukungan sumber daya teknologi yang tinggi. Kurangnya dukungan sumber daya teknologi akan menyebabkan Sistem Informasi Geografis menjadi tidak memenuhi kriteria inklusivitas, kolaborasi, pengembangan kapasitas, *request-response* yang cepat, atau timbulnya berbagai masalah di kalangan pengguna [6]. Heterogenitas data iklim dan fisiografi berbasis spasial pada faktanya cukup kompleks untuk diselesaikan. Misalnya sumber data yang berasal dari berbagai institusi, dengan format yang berbeda-beda, ukuran data yang besar, serta data yang bersifat dinamis sering dengan berjalannya waktu. Guna mendukung pembangunan aplikasi yang lebih efektif dan efisien dalam penggunaan sumber daya teknologi serta mampu mengatasi heterogenitas data spasial, maka diperlukan adanya kemandirian *back-end processing* dengan *front-end application*. Salah satu *platform* yang mendukung kemandirian tersebut adalah *platform* yang bekerja berbasis *Application Programming Interface* (API) atau *Application Programming Interface* adalah sebuah arsitektur yang memecah layanan menjadi bagian-bagian kecil dan terpisah sehingga layanan yang dihasilkan dapat digunakan oleh aplikasi lain tanpa perlu membuatnya dari awal atau menggandakannya. Selain itu, dengan mengimplementasikan API, layanan atau modul yang digunakan menjadi lebih efisien dan hemat sumber daya [7]. Sebagai contoh, memisahkan back-end dan front-end akan memungkinkan aplikasi lain untuk menggunakan back-end yang telah dibuat dengan data dan fungsi yang telah disediakan. Dengan cara ini, beban pada server dapat dikelola dengan lebih efektif dan mencegah terjadinya penurunan performa [8].

Salah satu teknologi yang mendukung pemrograman berbasis API adalah *Representational State Transfer* (REST) atau yang biasa dikenal dengan *RESTful web services*. *RESTful web services* merupakan layanan yang memungkinkan sistem melakukan akses dan manipulasi data menggunakan *set* yang seragam dan telah ditentukan sebelumnya [9]. Teknologi *RESTful web service* ini juga telah didukung oleh *framework* Laravel. *Framework* Laravel merupakan *framework* yang ditujukan untuk pengembangan aplikasi *web* dengan mengikuti pola arsitektur *Model-View-Controller* (MVC). *Framework* ini juga menyediakan fitur *authentication*, *routing*, *session manager*, *caching*, *IoC container*, dan juga *database migration tools*, semua fitur tersebut memberikan dukungan untuk pengembangan aplikasi yang lebih kompleks.

Penelitian ini bertujuan membangun skema baru arsitektur *Geospatial API* menggunakan Laravel *RESTful Web Service* untuk aplikasi sistem deteksi kesesuaian lahan pertanian di Kabupaten Semarang.

2 Tinjauan Literatur

Pentingnya membangun arsitektur spasial dalam perencanaan sistem pertanian didukung kuat oleh penelitian yang berjudul *Optimalisasi model prediksi kesesuaian lahan kelapa sawit menggunakan algoritma pohon keputusan spasial*. Pada penelitian tersebut disebutkan bahwa dalam mengevaluasi kesesuaian lahan pertanian, perlu mempertimbangkan korelasi data spasial dengan algoritma pendukung keputusan. Beberapa variabel atau faktor dan analisis data direferensikan secara geografis,

seperti kesesuaian lahan. Oleh karena itu, penting untuk mempertimbangkan korelasi data spasial, seperti posisi, kedekatan, dan orientasi [10]

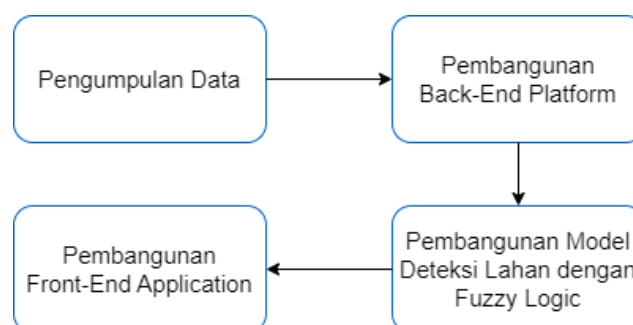
Salah satu arsitektur spasial yang dapat digunakan adalah arsitektur berbasis data GeoJSON. Sebuah penelitian menyebutkan bahwa GeoJSON memiliki kelebihan transformasi data yang sederhana dan kemudahan adaptasi dalam implementasi pada aplikasi spasial [11]. Pada penelitian dengan studi kasus pemetaan perencanaan lingkungan industri tersebut, dihasilkan transformasi peta topografi menjadi peta topografi berbasis *web* yang lebih interaktif. Disebutkan pula bahwa elemen terpenting yang mempengaruhi keberhasilan transformasi sistem adalah GeoJSON [11].

JSON dan juga GeoJSON adalah format data yang juga dihasilkan oleh *web service*. Framework Laravel adalah salah satu framework yang mendukung REST *web service*, dimana secara arsitektur terdiri dari *routing*, *middleware*, *MVC framework*, *ORM*, dan komponen *authentication*. Penelitian yang berjudul *Restful API Architecture Based on Laravel Framework* menyebutkan bahwa pada saat pembangunan RESTful API, hanya dua komponen yang berperan penting yaitu Model dan Controller. Pada akhir dari proses ini akan dihasilkan JSON sebagai *return value*. Selain itu, Laravel juga menyediakan kerangka kerja ORM yang kuat untuk manipulasi basis data relasional, dimana pada bagian Model sangat bergantung mengandalkan ORM untuk pemrosesan data [12].

Dengan membandingkan beberapa penelitian terkait, maka penelitian pembangunan Aplikasi Perencanaan Sistem Pertanian ini memiliki lingkup penelitian: (1). Berdasarkan karakteristik variabel yang digunakan, maka sistem dibangun berbasis geospasial, (2). Sistem dibangun di atas arsitektur *microservice* untuk kolaborasi heterogenitas sumber daya dan kemudahan transformasi data, dan (3). *Microservices* yang digunakan adalah Restful API pada *Framework* Laravel.

3 Metode Penelitian

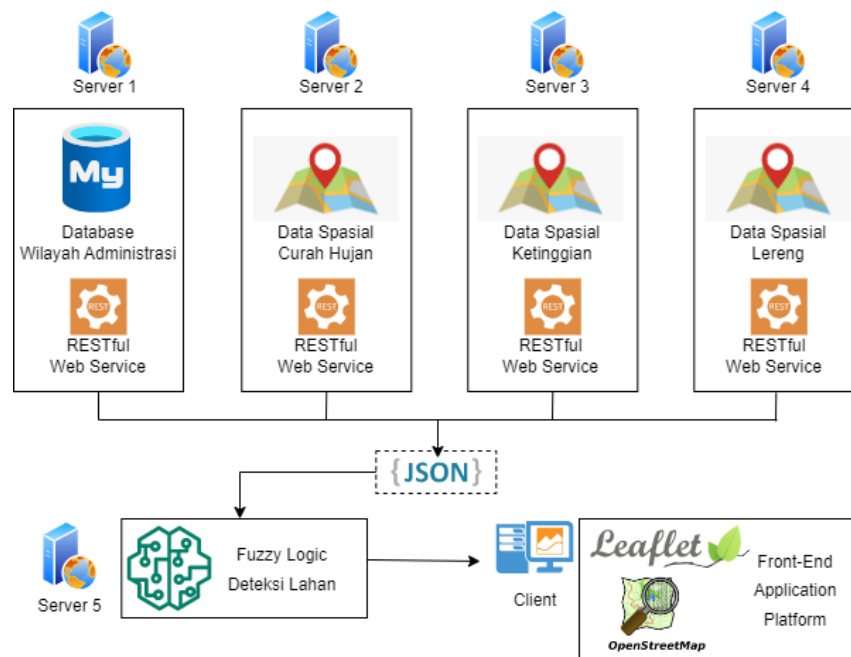
Data yang digunakan dalam penelitian ini berupa data geospasial, dimana data tersebut memiliki atribut dan meta data yang besar. Tantangan dalam perancangan sistem ini adalah bagaimana membangun dan menerapkan manajemen database dalam skala besar dan heterogen (tipe data dan sumber data) secara efisien. Penerapan arsitektur *microservices* merupakan solusi untuk mengintegrasikan berbagai sumber daya yang heterogen menjadi satu sistem pertanian yang inklusif dengan berbagai jenis user. Berdasarkan tantangan tersebut, maka penelitian ini terbagi menjadi empat tahap, yaitu: pengumpulan data, pembangunan *Back-end processing platform*, pembangunan model deteksi lahan dengan Logika Fuzzy, dan pembangunan *front-end application*. Tahap penelitian ditunjukkan pada Gambar 1.



Gambar 1. Tahap-Tahap Penelitian

Bagian pertama dari penelitian ini adalah tahap pengumpulan dan pengolahan data. Data yang diperlukan adalah Data Administrasi Kabupaten Semarang dari tingkat desa/kelurahan sampai tingkat kecamatan. Data ini diambil dari Badan Pusat Statistik (BPS) Kabupaten Semarang. Data geospasial yang digunakan adalah Peta Kabupaten Semarang tingkat desa/kelurahan dan kecamatan, Peta Curah Hujan Kabupaten Semarang, Peta Ketinggian Wilayah Kabupaten Semarang, dan Peta Kelerengan Kabupaten Semarang. Peta-peta tersebut adalah peta resmi yang dikeluarkan oleh Badan Perencanaan Daerah (Bappeda) Kabupaten Semarang. Peta-peta tersebut masih dalam format *shapefile* (Shp), sehingga selanjutnya peta tersebut dikonversi menjadi database spasial. Pada tahap kedua sampai

keempat penelitian ini, arsitektur sistem yang dibangun mengacu pada arsitektur yang ditunjukkan pada Gambar 2.



Gambar 2. Perancangan Arsitektur Geospasial API dengan Laravel untuk Sistem Deteksi Kesesuaian Lahan Pertanian

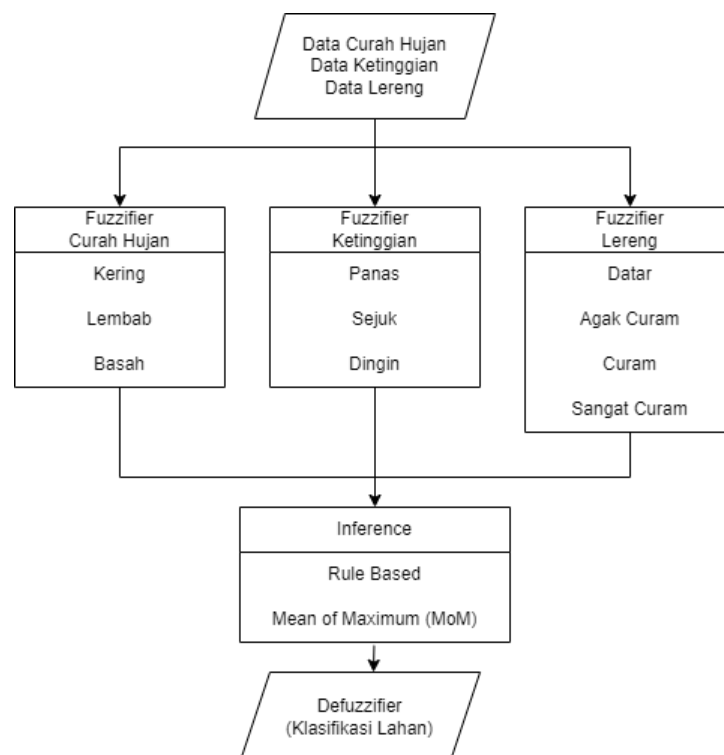
Geospatial API dibangun dengan menggunakan Restful API pada *framework* Laravel. *Representational state transfer* (REST) atau RESTful web services merupakan layanan yang memungkinkan sistem melakukan akses dan manipulasi data menggunakan *set* yang seragam dan telah ditentukan sebelumnya. Operasi pada REST dilakukan dengan *method* HTTP GET, PUT, POST, DELETE, HEADER dan OPTIONS, akan mengambil *resources* yang diminta dan selanjutnya akan dikembalikan sebagai *response* ke pemanggil [13] *Microservices* yang dibangun pada penelitian ini adalah *service* untuk mengambil Data Spasial Peta Kabupaten Semarang, Peta Curah Hujan, Peta Ketinggian, dan Peta Lereng, serta data non spasial yaitu Data Desa, Data Kecamatan, dan Data Syarat Tumbuh Tanaman. Operasi-operasi tersebut dijalankan menggunakan *method* GET. Data yang dihasilkan oleh *microservices* ini berbentuk Geo-JSON. Selain itu terdapat *service* lain untuk *insert*, *update*, dan *delete* untuk masing-masing data yang dijalankan pada *method* POST.

Pada tahap ketiga, dilakukan kalkulasi deteksi kesesuaian lahan. Pada awal tahap ini, dipersiapkan data karakteristik fisiografi dan iklim suatu wilayah. Data karakteristik fisiografi yang digunakan adalah kemiringan lereng dan drainase. Data karakteristik iklim yang digunakan adalah suhu udara (diambil dari peta ketinggian wilayah) dan rejim kelembaban yang diperoleh dari peta curah hujan tahunan.

Terbagi menjadi empat kategori, kemiringan lereng dapat diklasifikasikan menjadi Zona 1 dengan kemiringan kurang dari 8% dan fisiografi yang datar hingga agak datar, Zona 2 dengan kemiringan antara 8% hingga 15% dan fisiografi yang berombak serta lereng yang agak curam, Zona 3 dengan kemiringan antara 15% hingga 40% dan fisiografi yang berbukit serta lereng yang curam, serta Zona 4 dengan kemiringan lebih dari 40% dan fisiografi yang bergunung serta lereng yang sangat curam [13]. Klasifikasi suhu udara terdiri dari tiga kelompok, yaitu: Panas yang merujuk pada daerah dengan ketinggian 500 mdpl atau memiliki rata-rata suhu udara tahunan lebih dari 26°C; Sejuk yang merujuk pada daerah dengan ketinggian antara 500 mdpl hingga 1000 mdpl atau memiliki rata-rata suhu udara tahunan antara 26°C hingga 23°C; dan Dingin yang merujuk pada daerah dengan ketinggian lebih dari 1000 mdpl atau memiliki rata-rata suhu udara tahunan kurang dari 23°C. Terakhir, variabel Dalam klasifikasi kelembaban wilayah, terdapat tiga kelompok berdasarkan data bulan kering atau curah hujan, yaitu: Kering, jika terdapat lebih dari 7 bulan kering dalam satu tahun atau curah hujan tahunan kurang

dari 1500 mm; Lembab, jika terdapat antara 4 hingga 7 bulan kering dalam satu tahun atau curah hujan tahunan antara 1500 mm hingga 3000 mm; dan Basah, jika terdapat kurang dari 3 bulan kering dalam satu tahun atau curah hujan tahunan lebih dari 3000 mm [14].

Model yang digunakan untuk mengelompokkan wilayah berdasarkan karakteristik yang sama adalah Logika Fuzzy. Sistem Logika Fuzzy adalah model yang meniru cara berpikir manusia dan dianggap sebagai solusi yang efektif untuk menyelesaikan masalah yang mengandung ketidakpastian. Karena data karakteristik lahan cukup kompleks, maka *Fuzzy Inference System* (FIS) dapat digunakan untuk pengolahan datanya [15]. Terakhir, untuk memperoleh rekomendasi sistem pertanian yang sesuai pada lahan tersebut, hasil pengelompokan data akan dicocokkan dengan Data Syarat Tumbuh Tanaman diperoleh dari lampiran Peraturan Menteri No.47/Permentan/OT.140/10/2006 tentang Pedoman Umum Budidaya Pertanian Pada Lahan Pegunungan. Tahapan *Fuzzy Inference System* (FIS) untuk deteksi kesesuaian lahan pertanian, ditunjukkan pada Gambar 3.



Gambar 3. Tahapan Fuzzy Inference System (FIS) untuk Deteksi Kesesuaian Lahan Pertanian

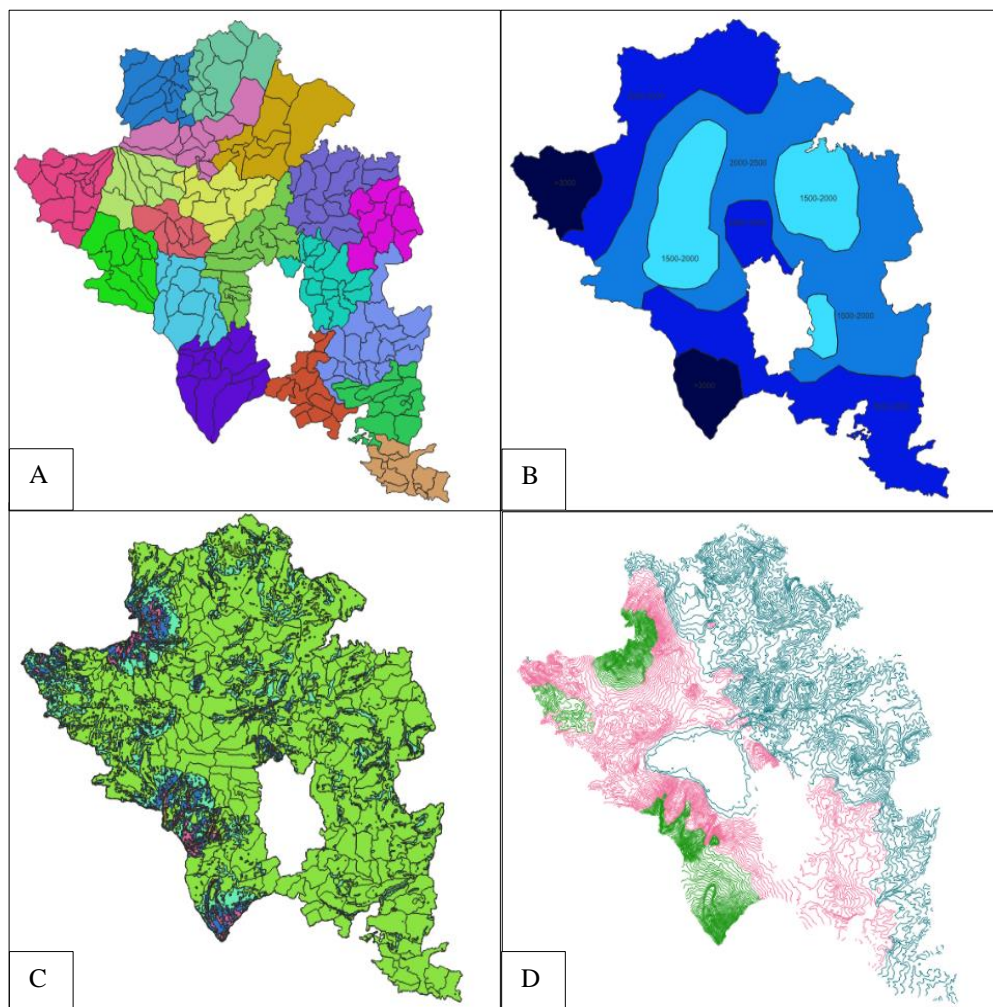
Tahap terakhir dari penelitian ini adalah membangun *User Interface* (UI), dimana sistem disajikan dalam bentuk peta interaktif menggunakan *framework* Leaflet dan *Open Street Map*. Leaflet merupakan *open-source JavaScript library* untuk *mobile-friendly interactive maps*. Leaflet mendukung penyajian peta secara *responsive* dimana peta dapat dibuka pada aplikasi berbasis *web* maupun *mobile*. Selain itu, Leaflet juga mengizinkan terjadinya interaksi user dengan peta digital [16].

4 Hasil dan Pembahasan

Data spasial yang digunakan pada penelitian ini adalah berupa Peta Kabupaten Semarang tingkat desa/kelurahan dan kecamatan, Peta Curah Hujan Kabupaten Semarang, Peta Ketinggian Wilayah Kabupaten Semarang, dan Peta Kelerengan Kabupaten Semarang seperti yang ditunjukkan pada Gambar 3. Data spasial tersebut diproyeksikan dalam *Coordinate Reference System* (CRS) EPSG:4267 dengan model koordinat *latitude* dan *longitude*. Format data yang digunakan adalah GeoJSON.

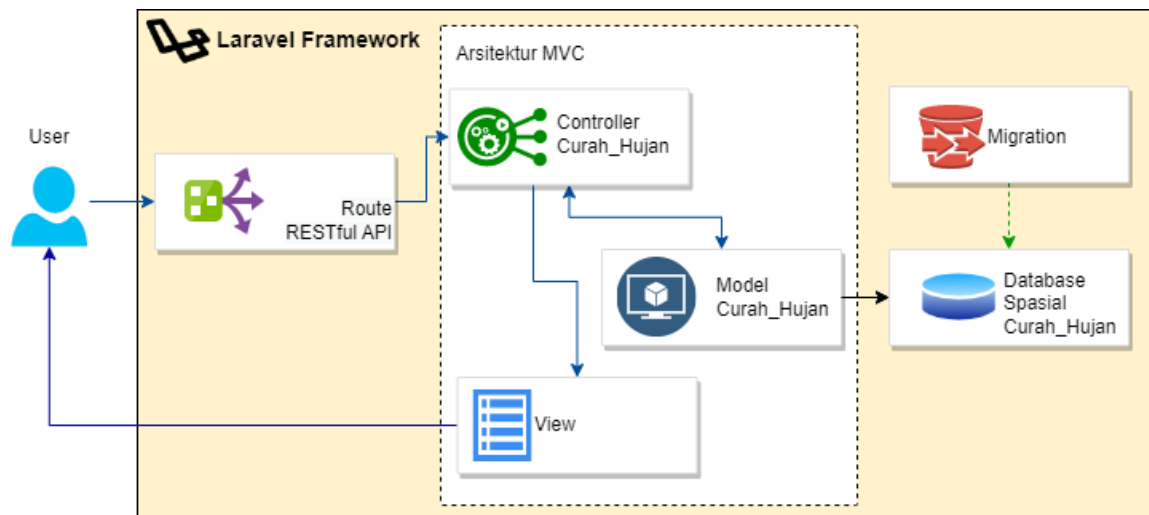
GeoJSON (*Geographic JavaScript Object Notation*) merupakan format data yang menggambarkan setiap elemen dari suatu titik, garis, atau poligon dalam tabel data geometris (*longitude and latitude*, atau satu set koordinat) [17]. Tipe data yang digunakan dalam membentuk sistem koordinat ini menggunakan tipe data geometri *polygon*. Tipe data *multipolygon string* adalah tipe data

geometri berupa kumpulan yang berisi *polygon*. Sedangkan *polygon* itu sendiri adalah kumpulan titik-titik yang membentuk himpunan bidang ruang yang terhubung. Tipe data ini biasanya digunakan untuk memetakan objek tertentu seperti wilayah, pulau, dan danau [18].



Gambar 4. Peta Choropleth: (A) Peta Kabupaten Semarang Tingkat Desa, (B) Peta Curah Hujan (C) Peta Kelerengan, dan (D) Peta Ketinggian Kabupaten Semarang

Framework Laravel menyertakan secara *default* ORM yaitu teknik untuk konversi data di antara tipe yang tidak kompatibel menggunakan bahasa pemrograman berorientasi objek yang disebut sebagai *Eloquent*. Operasinya bersifat sederhana, dimana setiap tabel dalam *database* akan terhubung dengan kelas di PHP, yang mencakup semua logika yang diperlukan untuk berinteraksi dengan *database*. Kelas-kelas ini merupakan *Model* dalam pola MVC yang juga mendukung operasi dan relasi antar tabel [19]. Laravel Framework menerapkan pola MVC (*Model-View-Controller*) dengan tugas yang terdefinisi untuk setiap bagian. Bagian *Controller* memiliki *Class*, *method*, dan *attribute*, sementara Model berperan sebagai basis pembuatan RESTful *web service* pada sistem. *Class* Model tergabung dalam *library Eloquent* yang mempermudah pengolahan data. *Controller* bertindak sebagai penghubung antara *request client* dari *View* ke *Class* Model. Pola MVC dalam Laravel memiliki struktur yang berbeda dari pola MVC umumnya, terutama dalam penghubungan *route* antara *request* dari *user* dan *Controller*. Dengan struktur ini, *Controller* tidak langsung menerima request dari user [20]. Sebagai contoh, arsitektur MVC pada *Server* Curah Hujan ditunjukkan pada Gambar 5.



Gambar 5. Arsitektur MVC pada Server Curah Hujan

Kode Program 1 adalah kode program bagian *Model*. Model merupakan modul yang mendefinisikan MySQL *Spatial Data Fields* yang telah terbentuk di *Migration*. Pada Kode Program 1 ditunjukkan model Peta Curah Hujan Kabupaten Semarang. Model ini merupakan warisan dari Super Kelas Grimzy yang memetakan dengan ORM sebagai Eloquent. Pada baris tiga, *SpatialTrait* diterapkan pada model ini untuk mengenali *field* yang bertipe data spasial. Baris lima sampai baris sembilan merupakan data atribut yang menempel pada data spasial atau peta. Adapun baris sebelas adalah *array SpatialFields* yang digunakan untuk menampung data *polygon* peta curah hujan.

Kode Program 1. Model Peta Curah Hujan Kabupaten Semarang

```

1   class model_curah_hujan extends Model
2   {
3       use SpatialTrait;
4
5       protected $fillable = [
6           'kode_hujan',
7           'nilai_curah_hujan'
8
9       ];
10
11      protected $spatialFields = [
12          'coordinates'
13      ];
14  }
```

Sementara itu, *query* pengambilan data sebaran curah hujan dilakukan pada bagian *controller*, seperti yang ditunjukkan pada Kode Program 2. Nama kelas *controller* yang digunakan pada Kode Program 2 adalah *curah_hujan_Controller* seperti yang ditunjukkan pada baris satu. Variabel *\$all_data* digunakan untuk mengambil data berupa *String* dan data berupa *polygon geometry* dari Peta Curah Hujan. Baris sembilan sampai baris dua puluh menampung data hasil *query* ke dalam bentuk *array* asosiatif yang dipersiapkan sesuai dengan format *GeoJSON*. Terakhir, *controller* ini akan mengembalikan nilai dalam bentuk *GeoJSON* seperti yang ditunjukkan pada baris dua puluh dua.

Kode Program 2. Controller Peta Curah Hujan Kabupaten Semarang

```
1 class curah_hujan_Controller extends Controller
2 {
3     public function get_peta_CH(Request $request){
4         $semua_data = model_curah_hujan::select('kode_hujan',
5         'nilai_curah_hujan', 'coordinates');
6         $semua_data = $semua_data->groupBy('kode_hujan')->get();
7         ...
8         $tmp = [];
9         $tmp['type'] = "Feature";
10        $tmp['properties'] = [
11            'KODE_LOKASI' => $ak->kode_hujan,
12            'CURAH_HUJAN' => $ak->nilai_curah_hujan,
13        ];
14        ...
15        $tmp['geometry'] = array(
16            'type' => 'MultiPolygon',
17            'coordinates' => [$longlat]
18        );
19        array_push($hasil,$tmp);
20        ...
21        return response()->json($hasil,200,array());
22    }
23 }
```

API didefinisikan sebagai jembatan dengan koneksi aman antara data eksternal dan platform, yang memungkinkan sistem untuk mengirim data dari perangkat eksternal. Salah satu model API yang dapat digunakan adalah REST atau *Model Representational State Transfer*, dimana pada model ini terdapat beberapa method yang disediakan seperti GET, POST, PUT, dan DELETE [21]. *Controller* yang telah dibangun pada tahap sebelumnya akan memasuki tahap *deploy* ke dalam REST API. Kode Program 3 adalah kode program yang menunjukkan API *route* yang didefinisikan pada framework *Laravel*.

Kode Program 3. API Route Pada Server Curah Hujan untuk curah_hujan_Controller

```
1 Route::get('/getCH',[...\curah_hujan_Controller::class, 'get_peta_CH']);
2 Route::get('/getCH/{id}',[...\curah_hujan_Controller::class,'getCH_byId']);
3 Route::post('/setCH',[...\controller_cud::class, 'set_CH']);
4 Route::post('/updateCH/{id}',[...\curah_hujan_Controller::class, 'update_CH']);
5 Route::post('/deleteCH/{id}',[...\curah_hujan_Controller::class, 'delete_CH']);
```

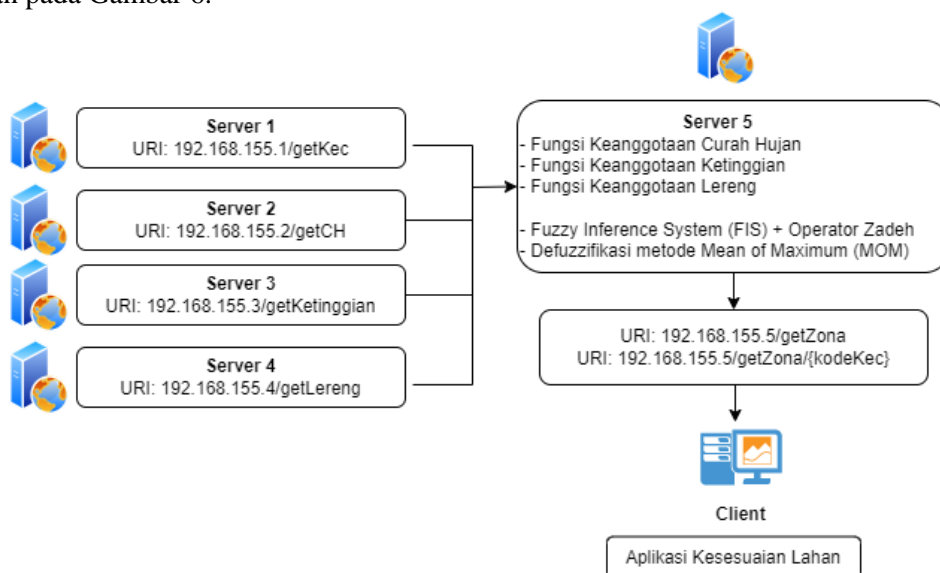
Baris satu sampai tiga menunjukkan *controller* yang bernama *curah_hujan_Controller* telah masuk tahap *deploy* ke dalam API dengan method GET dan nama *path* URI */getCH* untuk mendapatkan data curah hujan semua wilayah, */getCH/{id}* untuk mendapatkan nilai curah hujan suatu wilayah, dan */setCH* untuk memasukkan data curah hujan. Daftar API *route* yang dihasilkan dalam penelitian ini terdapat tiga API dengan *method* POST yaitu *setCH*, *updateCH*, dan *deleteCH*. Dua API yang lain menggunakan *method* GET yaitu *getCH*, dan *getCH/{id}*.

Return value yang dihasilkan dari beberapa API dalam penelitian ini adalah dalam bentuk GeoJSON. GeoJSON (*Geographic JavaScript Object Notation*) adalah format data yang menggambarkan setiap elemen dari suatu titik, garis, atau poligon dalam tabel data geometris (*longitude and latitude*, atau satu set koordinat) [22]. Salah satu keuntungan yang diperoleh dengan menggunakan GeoJSON ini, objek spasial berupa *polygon* akan lebih mudah direpresentasikan pada *front-end application* [11]. Format GeoJSON yang untuk Peta Curah Hujan (*/getCH*) ditunjukkan pada Kode Program 4. Pada Kode Program 4 diperlihatkan bahwa satu poligon curah hujan memiliki fitur *properties* dan *geometry*. Pada struktur tersebut terkandung dua bagian utama yaitu *properties* dan *geometry*. *Properties* peta berisi identitas dari suatu poligon yaitu: kode lokasi, curah hujan, dan luas poligon. Sementara itu *geometry* berisi tipe data dan kumpulan koordinat.

Kode Program 4. Struktur GeoJSON Peta Curah Hujan Kabupaten Semarang

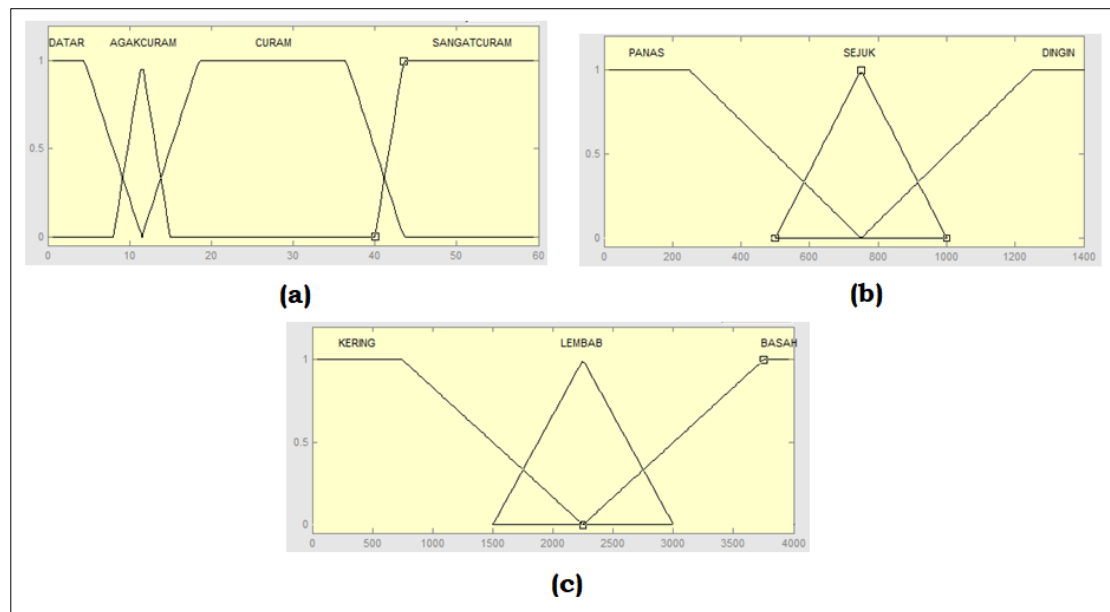
```
{
  "type": "Feature",
  "properties": {
    "KODE_LOKASI": "SMG01",
    "CURAH_HUJAN": "2500-3000",
    "HECTARES": 15852.883
  },
  "geometry": {
    "type": "MultiPolygon",
    "coordinates": [
      [
        [
          110.420723149679958,
          -7.098268860567361
        ],
        [
          110.420762424276219,
          -7.098103168881597
        ],
        [
          110.420806619107012,
          -7.097908235391677
        ],
        ...
      ]
    ]
  }
}
```

Tahap terakhir dari pembangunan *Back-end Processing Platform* ini adalah menguji API yang telah dibuat menggunakan Postman. Hal ini dilakukan untuk memastikan API yang dihasilkan dapat diterapkan pada *layer application* pada Server 5. Hasil pengujian pada Postman menunjukkan *response* yang dikembalikan oleh API sesuai dengan format GeoJSON sehingga dapat digunakan oleh Server 5. Pada Gambar 2 mengenai Arsitektur sistem, ditunjukkan semua API dari Server 1 sampai Server 4 akan ditarik oleh Server 5. Server 5 merupakan mesin deteksi kesesuaian lahan menggunakan Logika Fuzzy. Logika Fuzzy pada sistem deteksi kesesuaian lahan ini bekerja dengan mengambil data Wilayah, Curah Hujan, Ketinggian, dan Kelerengan dari masing-masing server. Arsitektur Server 5 yang terbentuk ditunjukkan pada Gambar 6.



Gambar 6. Arsitektur Server 5 – Mesin Deteksi Kesesuaian Lahan dengan Logika Fuzzy

Logika Fuzzy digunakan sebagai algoritma dalam menentukan kesesuaian lahan di masing-masing desa di Kabupaten Semarang. Tiga variabel yang digunakan meliputi curah hujan, ketinggian, dan kemiringan. Terdapat 6.895 titik atau sampel yang tersebar di seluruh wilayah yang diproses menggunakan Logika Fuzzy tersebut. Proses dimulai dengan pengambilan nilai curah hujan, ketinggian, dan kemiringan dari API Server 1 hingga Server 4. Kemudian, dilakukan penghitungan derajat keanggotaan dengan menggunakan fungsi-fungsi tertentu. Kurva fungsi keanggotaan masing-masing variabel terlihat pada Gambar 7.



Gambar 7. Kurva Fungsi Keanggotaan: (a) Kelerengan, (b) Ketinggian, dan (c) Curah Hujan

Sistem menghasilkan 36 aturan yang didasarkan pada fungsi keanggotaan. Setelah menghitung derajat keanggotaan, nilai-nilai tersebut akan dimasukkan ke dalam proses Fuzzy Inference System dengan menggunakan operator Zadeh MIN. Operator ini berfungsi untuk mencari nilai terkecil dari nilai-nilai tersebut. Aturan dalam *fuzzy inference system* menggunakan operator AND, yang mengambil nilai keanggotaan terkecil antara elemen pada himpunan yang relevan. Metode defuzzifikasi yang digunakan adalah MOM (*Mean of Maximum*). Crisp value didapatkan dengan cara mencari nilai rata-rata dari domain yang memiliki nilai keanggotaan maksimum. Setelah zona kesesuaian lahan pada suatu wilayah diketahui, sistem akan menampilkan hasil zona kesesuaian lahan serta penjelasan tentang kesesuaian lahan yang sesuai dengan komoditas pertanian yang dapat dibudidayakan.

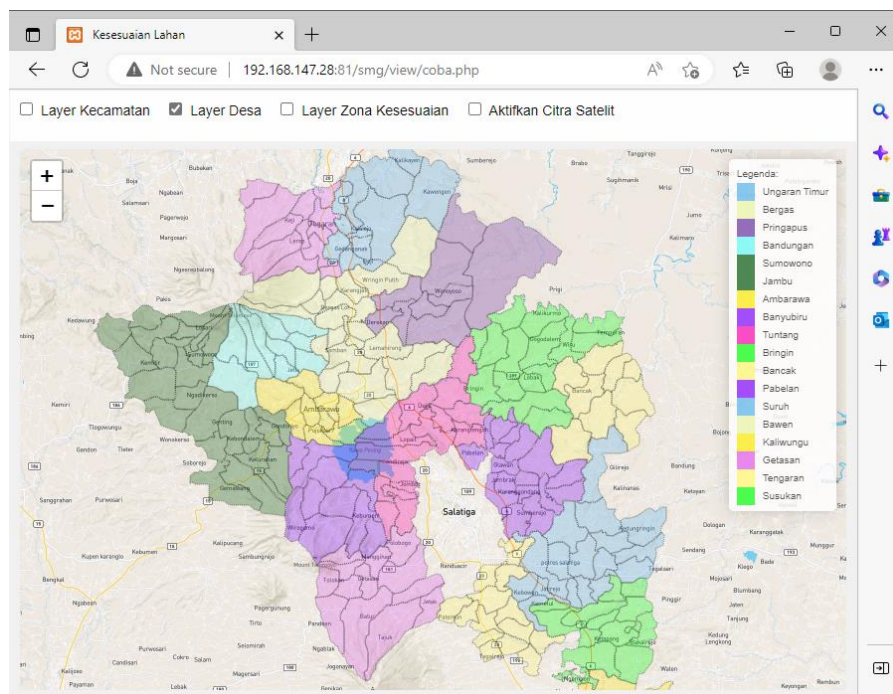
Inovasi yang diberikan dalam penelitian ini yaitu kolaborasi teknologi dengan kemandirian antar aplikasi yang diwujudkan melalui *microservices*. GeoJSON *output* dari *microservices* merupakan bentuk kolaborasi data spasial dengan data non spasial, juga sangat memungkinkan untuk kemandirian antara layanan *back-end* dengan berbagai tipe aplikasi *front-end* seperti *Mobile Apps* dan *Web Based*. Pembagian layanan ke dalam bagian yang lebih kecil dan terpisah telah dilakukan melalui arsitektur *microservices* yang telah dibangun. GeoJSON atau modul yang dihasilkan membuat aplikasi ini berjalan lebih ringan [5].

Salah satu keuntungan yang diperoleh dalam membangun aplikasi berbasis *microservice* adalah kebebasan pemilihan *platform* yang digunakan pada *front end application*. Hal ini menyebabkan tidak bergantungnya *front end* dengan jenis *database*, *framework*, maupun bahasa pemrograman pada sisi *back end*. Pada penelitian ini, *front end application* dibangun menggunakan *framework* Leaflet dan OpenStreetMap berbasis Web. GeoJSON yang dihasilkan oleh *microservices* merupakan satu-satunya format JSON yang digunakan sebagai standar pemetaan digital [18]. Selain memiliki ukuran yang kecil, GeoJSON juga memiliki kemampuan transformasi dan adaptasi data yang lebih sederhana ketika diterapkan pada *framework* Leaflet. Ada beberapa *framework* lain yang juga dapat dikombinasikan dengan Leaflet untuk membangun peta interaktif, diantaranya D3, Google Maps API, OpenLayer atau MapBox [19].

Kode Program 5. Pemanggilan Services pada Leaflet

```
1 var LayerDesa = new L.GeoJSON.AJAX("http://192.168.155.5/getZona", {  
2 style: function (feature){  
3 kode = feature.properties.kode_lokasi;  
4 ...
```

Kode Program 5 adalah kode program yang menggambarkan pemanggilan *service* pada Leaflet di aplikasi *client*. Fungsi tersebut digunakan untuk mendapatkan data GeoJSON yang selanjutnya ditransformasi menjadi peta digital zona kesesuaian lahan. Pada baris satu ditunjukkan sebagai perintah untuk mengambil API dari *back end*, dalam hal ini Server 5. Selanjutnya, pada baris dua dan seterusnya merupakan perintah untuk menggambar peta digital wilayah desa atau kelurahan. Salah satu contoh peta desa/kelurahan di Kabupaten Semarang ditunjukkan pada Gambar 8.

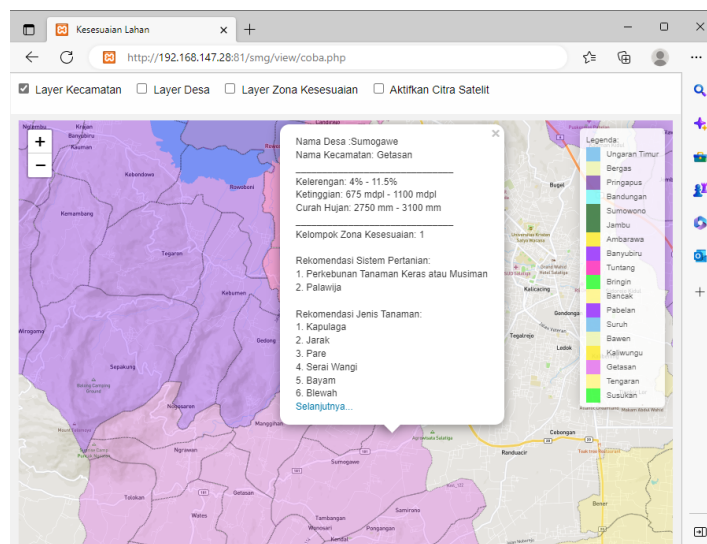


Gambar 8. Poligon Desa/Kelurahan Kabupaten Semarang di Sisi Aplikasi *Client*

Aplikasi pemetaan zona kesesuaian lahan Kabupaten Semarang ini menggabungkan data spasial dan data non spasial dimana kedua jenis data tersebut diperoleh dari *microservices*. Aplikasi ini memiliki fitur menggabungkan layer peta desa di Kabupaten Semarang dengan data fisiografi lahan, ketinggian wilayah, curah hujan, dan kelompok zona, serta layer OpenStreetMap, sehingga dapat diketahui kesesuaian lahan pertanian daerah di masing-masing desa. Hasil *overlay* layer peta dan data informasi zona kesesuaian lahan pertanian suatu desa tersebut ditunjukkan pada Gambar 9.

Overlay layer peta dengan informasi zona kesesuaian lahan ini mampu memberikan kemudahan ke berbagai *user*, termasuk *user* yang tidak memahami peta tematik pun, akan dapat mudah menginterpretasikan peta. Hal ini terjadi karena *layer* OpenStreetMap memberikan gambaran kondisi nyata dengan penyajian jalan dan beberapa penanda lokasi yang mudah dikenali oleh masyarakat luas. Selain menerapkan *layer overlay* untuk menyajikan peta desa, aplikasi ini juga dapat memberikan informasi secara detail mengenai kesesuaian lahan dan rekomendasi tanaman yang cocok dibudidayakan. Data detail mengenai satu kesesuaian lahan, diperoleh dari *services*, yang selanjutnya data tersebut dihubungkan dengan peta digital.

Inovasi yang dihasilkan pada *front end application* sisi *client*, menunjukkan kolaborasi dari beberapa *resources* data yang diberikan oleh API *services*, dapat menghasilkan aplikasi yang dapat mengintegrasikan peta spasial dengan data kesesuaian lahan non spasial. Apabila hal ini dibangun menggunakan arsitektur *monolithic* maka penggabungan dua *resources* data yang berbeda adalah hal yang tidak mudah dilakukan.



Gambar 9. Layer Overlay antara Peta Desa dan Informasi Zona Kesesuaian Lahan

5 Kesimpulan

Pemetaan kesesuaian lahan pertanian daerah Kabupaten Semarang berupa peta desa dan informasi kesesuaian lahan pertanian memiliki titik tumpu pada inovasi kolaborasi berbagai sumber daya data dan kemudahan penggunaan di berbagai jenis *user*. Penelitian ini telah membuktikan *microservices* adalah arsitektur yang dapat digunakan untuk menangani kolaborasi sumber daya data yang heterogen. Data spasial fisiografi lahan, ketinggian wilayah, dan curah hujan, serta informasi kesesuaian lahan mampu dikolaborasikan dengan RDBMS melalui Restful *web services* pada Laravel. Enam API utama dihasilkan pada penelitian ini, empat API dihasilkan dari empat *sever*, *services* tersebut adalah *services* yang berhubungan dengan data spasial (wilayah, ketinggian, kelerengian, dan curah hujan), sementara dua *services* lainnya, berhubungan dengan data konvensional informasi zonasi kesesuaian lahan pertanian (dihasilkan oleh *Server Fuzzy/Server 5*). JSON/GeoJSON sebagai format data yang dikeluarkan oleh masing-masing *services* dapat dengan sederhana dan mudah diterapkan pada *front end application* yang berbasis web menggunakan Leaflet *framework*. Sebagai wujudnya, pada *front-end application* terdapat fitur pemetaan desa yang digabungkan (*overlay*) dengan OpenStreetMap yang dapat memberikan gambaran kondisi nyata kesesuaian lahan yang mudah dikenali oleh masyarakat luas. Dalam penelitian yang lebih jauh lagi, arsitektur *microservices* dapat ditambahkan dengan standar keamanan yang lebih kuat menggunakan JWT (*JSON Web Tokens*). JWT sangat diperlukan terutama untuk aplikasi-aplikasi yang menyangkut data rahasia yang akan dilampirkan dalam sistem.

Referensi

- [1] A. Nurkholis and I. S. Sitanggang, "Optimization for prediction model of palm oil land suitability using spatial decision tree algorithm," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 192–200, Jul. 2020, doi: 10.14710/jtsiskom.2020.13657.
- [2] Kementerian Pertanian, *Outlook komoditas pertanian tanaman pangan ubi kayu*. Jakarta: Pusat Data Dan Sistem Informasi Pertanian Kementerian Pertanian, 2016.
- [3] BBSDLP, *Atlas peta kesesuaian lahan dan arahan komoditas pertanian pertanian, Kabupaten Bogor, Provinsi Jawa Barat*, 2nd ed. Bogor: Badan Penelitian dan Pengembangan Pertanian Kementerian Pertanian, 2016.
- [4] F. R. Fauzi, S. H. Abdullah, and A. Priyati, "Evaluasi Kesesuaian Lahan untuk Komoditas Padi dengan Memanfaatkan Aplikasi Sistem Informasi Geografis (SIG) di Kabupaten Lombok Tengah," *Jurnal Ilmiah Rekayasa Pertanian dan Biosistem*, vol. 6, no. 2, pp. 131–140, Sep. 2018, doi: 10.29303/jrpb.v6i2.87.
- [5] J. B. Ahmed and S. Mansor, "Overview of the application of geospatial technology to groundwater potential mapping in Nigeria," *Arabian Journal of Geosciences*, vol. 11, no. 17. Springer Verlag, Sep. 01, 2018. doi: 10.1007/s12517-018-3852-4.

- [6] S. Kramer, T. Amos, S. Lazarus, and M. Seedat, "The Philosophical Assumptions, Utility and Challenges of Asset Mapping Approaches to Community Engagement," *Journal of Psychology in Africa*, vol. 22, no. 4, pp. 537–544, Jan. 2012, doi: 10.1080/14330237.2012.10820565.
- [7] Siti Rochimah and Bintang Nuralamsyah, "Decomposing Monolithic to Microservices: Keyword Extraction and BFS Combination Method to Cluster Monolithic's Classes," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 2, pp. 263–270, Mar. 2023, doi: 10.29207/resti.v7i2.4866.
- [8] B. P. Putra and Y. A. Susetyo, "Implementasi API Master Store Menggunakan Flask, REST, dan ORM di PT XYZ," *Jurnal Sistemasi*, vol. 9, no. 3, p. 543, Sep. 2020, doi: 10.32520/stmsi.v9i3.899.
- [9] X. Chen, Z. Ji, Y. Fan, and Y. Zhan, "Restful API Architecture Based on Laravel Framework," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Nov. 2017. doi: 10.1088/1742-6596/910/1/012016.
- [10] A. Pratama, A. Adiyatma Wicaksana, and A. Razi, "Analisa Kesesuaian Lahan Tanah untuk Tanaman Padi (*Oryza sativa* L.) dengan Metode Decision Tree Berbasis Web (Studi Kasus: Kabupaten Aceh Utara)," *JIK*, vol. 6, no. 1, 2022.
- [11] T. Horbiński and D. Lorek, "The use of Leaflet and GeoJSON files for creating the interactive web map of the preindustrial state of the natural environment," *Journal of Spatial Science*, vol. 67, no. 1. Mapping Sciences Institute Australia, pp. 61–77, 2022. doi: 10.1080/14498596.2020.1713237.
- [12] O. M. A. AL-atraqchi, "A Proposed Model for Build a Secure Restful API to Connect between Server Side and Mobile Application Using Laravel Framework with Flutter Toolkits," *Cihan University-Erbil Scientific Journal*, vol. 6, no. 2, pp. 28–35, Aug. 2022, doi: 10.24086/cuesj.v6n2y2022.pp28-35.
- [13] S. J. Kim *et al.*, "Developing spatial agricultural drought risk index with controllable geo-spatial indicators: A case study for South Korea and Kazakhstan," *International Journal of Disaster Risk Reduction*, vol. 54, Feb. 2021, doi: 10.1016/j.ijdr.2021.102056.
- [14] U. P. Jawang *et al.*, "Evaluasi Zona Agroekologi Kabupaten Sumba Tengah Berbasis Sistem Informasi Geografis", *Konser Karya Ilmiah Magister Ilmu Pertanian UKSW*, 2017.
- [15] Y. W. Baly Woda, I. Hermadi, and M. Marimin, "Sistem Pendukung Keputusan Cerdas Kesesuaian Lahan dengan Jenis Tanaman Pangan: Studi Kasus Kabupaten Sikka," *Jurnal Teknologi Industri Pertanian*, vol. 29, no. 1, pp. 62–71, Apr. 2019, doi: 10.24961/j.tek.ind.pert.2019.29.1.62.
- [16] R. Ariyanto *et al.*, "A web and mobile GIS for identifying areas within the radius affected by natural disasters based on openstreetmap data," *International journal of online and biomedical engineering*, vol. 15, no. 15, pp. 80–95, 2019, doi: 10.3991/ijoe.v15i15.11507.
- [17] R. G. Donohue, C. M. Sack, and R. E. Roth, "Time Series Proportional Symbol Maps with Leaflet and jQuery," *Cartographic Perspectives*, no. 76, pp. 43–66, Oct. 2014, doi: 10.14714/CP76.1248.
- [18] R. Pu, "A special issue of geosciences: Mapping and assessing natural disasters using geospatial technologies," *Geosciences (Switzerland)*, vol. 7, no. 1. MDPI AG, Mar. 01, 2017. doi: 10.3390/geosciences7010004.
- [19] D. A. Morales *et al.*, "Designing of an enterprise resource planning for the optimal management of agricultural plots regarding quality and environmental requirements," *Agronomy*, vol. 10, no. 9, Sep. 2020, doi: 10.3390/agronomy10091352.
- [20] O. Dwi Arianto and Y. A. Susetyo, "Penerapan RESTful Web Service dengan Framework Laravel untuk Pembangunan Sistem Informasi Manajemen Sumber Daya Manusia." *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 2, pp. 522–532, May 2022, doi: 10.29100/jupi.v7i2.2870.
- [21] R. Sturm, C. Pollard, and J. Craig, in *Application Performance Management (APM) in the Digital Enterprise, Chapter 11—Application Programming Interfaces and Connected Systems*. Boston, MA, USA, 2019.
- [22] R. G. Donohue, C. M. Sack, and R. E. Roth, "Time series proportional symbol maps with Leaflet and jQuery," *Cartographic Perspectives*, vol. 76, pp. 43–65, 2013.