

Implementasi Sistem-informasi Desa Berbasis Arsitektur Microservices

Implementation of Microservice Architecture in Village Information Systems

Suryo Atmojo^{1*}
Ruli Utami²
Suzana Dewi³
Nurwahyudi Widhiyanta⁴

^{1,3,4}Teknik Informatika, Universitas Wijaya Putra Suarabaya, Indonesia

²Sistem-informasi, ITATS Surabaya, Indonesia

¹ suryoatmojo@uwp.ac.id, ²ruli.utami@itats.ac.id, ³suzanadewi@uwp.ac.id,

³nurwahyudiwidhiyanta@uwp.ac.id

*Penulis Korespondensi:

Suryo Atmojo
suryoatmojo@uwp.ac.id

Riwayat Artikel:

Diterima : 27 Maret 2022

Direview : 31 Maret 2022

Disetujui : 17 Juni 2022

Terbit : 18 Juni 2022

Abstrak

Suatu sistem informasi dapat dikatakan baik, yaitu sistem tersebut dapat memberikan penyajian informasi yang sederhana, dan juga harus mampu bertahan dan beradaptasi dengan gangguan. Resiliensi adalah kemampuan suatu benda untuk menghadapi gangguan dan beradaptasi. Arsitektur sistem informasi yang kurang tahan terhadap elastisitas adalah arsitektur monolitik dimana sistem backend dan frontend digabungkan dalam satu tempat dari segi fungsionalitas dan layanan, selain itu arsitektur monolitik juga menjalankan semua logika dalam satu server aplikasi. Arsitektur microservices memungkinkan pengembangan fungsionalitas dalam sebuah aplikasi atau sistem informasi untuk didekomposisi menjadi beberapa bagian layanan kecil, memungkinkan layanan untuk mengelola sumber daya dengan baik dan menjadi layanan kecil yang saling berhubungan. Dalam studi ini, studi kasus fokus pada masalah dalam sistem informasi desa Pengalangan, seperti persyaratan sistem yang berubah dengan cepat, pembuatan laporan yang memakan waktu karena proses pengumpulan data masih manual, ketergantungan hanya pada satu teknologi untuk aksesibilitas, tumpukan Manajemen, lebih banyak file, perlunya kemudahan dalam menambah fitur sistem yang mudah, desa yang sudah menggunakan sistem informasi memiliki masalah keamanan sistem, pengembangan sistem tidak mudah karena harus merombak sistem yang ada. Perancangan sistem informasi desa dengan arsitektur microservice merupakan solusi yang tepat untuk mengatasi permasalahan yang ada pada sistem informasi berbasis arsitektur monolitik khususnya sistem informasi desa pengalangan. Dengan menggunakan arsitektur layanan mikro, dimungkinkan untuk menjaga keamanan sistem, meningkatkan kinerja layanan desa dan menjadi lebih cepat dan lebih mudah untuk menskalakan proses bisnis pada sistem.

Kata Kunci: *Microservice, Sistem, Informasi, Desa*

Abstract

Information systems can be said to be good, namely, the system can provide convenience in presenting information, must also be able to survive when experiencing disturbances, and be able to adapt. The ability to deal with disturbances and adapt is called resilience. One of the information system architectures that lack resilience capabilities is monolithic architecture, where the backend and frontend systems are combined in the same functions and services, besides that the monolithic architecture also runs all logic in one application server. The microservice architecture allows the development of functions in an application or information system to be broken down into several small service parts and makes the service have the ability to manage resources properly and become a small service that is interconnected into

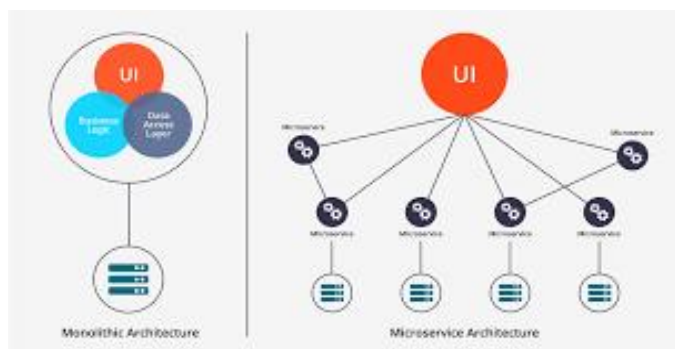
one. The design of this microservice-based Village Information System is the best solution to solve the problems that exist in monolithic architecture-based information systems, especially in village information systems. In this study, the case study focuses on problems in the Pengalangan Village Information System such as rapidly changing system requirements, system requirements that can be easily expanded, accessibility that only depends on one technology, administrative stacks, more files, report generation that requires time because the data collection process is still manual, system security problems for villages that already use information systems, and system development is not easy because they have to overhaul the existing system. By using a microservice architecture, the service performance of village apparatus will increase and become faster, it is easier to expand business processes on the system, and system security is maintained.

Keywords: *Microservice, System, Information, Village*

1. Pendahuluan

Pengalangan adalah salah satu desa yang terletak di Jalan Menganti. Lebih tepatnya letak desa ini berbatasan dengan Kota Surabaya di sebelah utara dan timur, Desa Setro di sebelah selatan, dan Desa Randupadangan di sebelah barat. Jika melihat pelayanan Desa Pengalangan Kabupaten Gresik, terlihat bahwa pemanfaatan teknologi informasi ini belum maksimal. Sistem yang digunakan selama ini masih berupa sistem informasi yang bersifat hanya sebagai media pencatatan saja [2], dan ditemukan beberapa kendala dalam penggunaan sistem informasi yang telah ada [3], seperti sulit dan lambatnya pengajuan proposal surat baru, proses perubahan atau penghapusan data, khususnya menyediakan laporan data Kependudukan [4]. Banyak juga yang percaya bahwa birokrasi pemerintah itu rumit dan sulit. Hal ini didasarkan pada apa yang mereka lihat ketika berhadapan dengan manajemen kependudukan [5]. Masalah lain yang terjadi adalah pencatatan data secara tradisional sering mengakibatkan duplikasi data [6] sehingga mengakibatkan pembukuan yang tidak efisien [7]. Selain itu, penyimpanan buku di lemari arsip juga mempengaruhi sulitnya mencari bahan yang dibutuhkan setiap saat [8]. Melihat hal tersebut maka diperlukan suatu sistem informasi yang dapat mempermudah pengelolaan data dan memberikan pengelolaan data yang efektif di Desa Pengalangan.

Suatu sistem informasi dapat dikatakan sebagai sistem informasi yang baik jika dapat memberikan ekspresi informasi yang sederhana [9], dan sistem tersebut harus mampu menahan ancaman sistem dan beradaptasi dengan gangguan sistem. Ketahanan atau resiliensi adalah kemampuan suatu sistem untuk menghadapi ancaman dan beradaptasi. Contoh arsitektur sistem informasi yang kurang kuat adalah arsitektur monolitik [10], di mana backend dan frontend digabungkan menjadi fungsi atau layanan yang sama dan menjalankan semua logika dalam satu sistem atau server aplikasi. Penggabungan frontend dan backend ini mempengaruhi ketergantungan antara setiap komponen [11]. Jadi jika salah satu komponen salah, itu akan mempengaruhi keseluruhan sistem, jika Anda membuat perubahan teknis, itu akan mengubah keseluruhan sistem. Aplikasi yang dibangun menggunakan arsitektur ini dapat berskala sangat besar karena struktur aplikasi dibangun dengan cara yang kompleks. Salah satu masalah yang sering terjadi pada arsitektur monolitik adalah resiliency challenge [12], yaitu kegagalan terjadi pada proses update dan penambahan fitur pada sistem, sehingga jika terjadi kegagalan pada proses penambahan fitur maka semua fungsi aplikasi akan gagal.



Gambar 1. Ilustrasi Perbedaan monolith dan microservices

Baru-baru ini, istilah arsitektur layanan mikro telah menjadi alternatif pengembangan dari arsitektur monolitik. Arsitektur *microservice* adalah sistem yang dirancang untuk berbagi layanan yang lebih kecil dan lebih fleksibel [13]. Perangkat lunak akan dirancang untuk melakukan fungsi secara mandiri. Artinya, setiap masalah teknis dapat diselesaikan dengan menggunakan metode dan teknik yang berbeda, yang kemudian dihubungkan ke antarmuka pemrograman aplikasi (API).

Layanan mikro dapat didefinisikan sebagai layanan yang membagi aplikasi menjadi layanan yang lebih kecil dan saling berhubungan. Arsitektur *microservice* adalah gaya arsitektur sistem atau aplikasi yang membagi fungsi dan memungkinkan setiap fungsi dalam suatu aplikasi mengalami perkembangannya sendiri [14]. Bentuk arsitektur *microservices* mempengaruhi hubungan antara aplikasi dan database. Dalam kerangka kerja layanan mikro, setiap layanan memiliki skema basis datanya sendiri [15]. Selain itu, layanan dapat menggunakan semua jenis database dan bahasa pemrograman yang mereka anggap sesuai dengan kebutuhan mereka. Dengan cara ini, setiap layanan lebih dioptimalkan. Menggunakan kerangka kerja arsitektur layanan mikro, aplikasi bisa lebih padat dan kompleks, namun tetap ringan. Sederhananya, layanan mikro adalah cara membagi layanan menjadi bagian-bagian yang lebih kecil tetapi masih saling terkait.

Selain itu, setiap layanan yang dibuat dapat menggunakan teknologi yang berbeda. Dalam implementasinya, layanan yang dibangun memiliki antarmuka untuk memproses dan mengirim pesan dalam bentuk XML atau JSON [16]. Layanan yang diberikan sebagai output menggunakan konsep berorientasi objek dalam mengirimkan permintaan dan tanggapan.

Pada penelitian sebelumnya yang berjudul “*Microservice Architecture for Information System Resilience*”, ditemukan bahwa implementasi *microservices* berhasil meningkatkan ketahanan sistem informasi. Selanjutnya penelitian yang berjudul “*Menyelenggarakan Kerja Sistem Parkir Menggunakan Arsitektur Microservice*” menyimpulkan bahwa setiap modul sistem dapat beroperasi dengan menggunakan bahasa pemrograman yang berbeda, namun dapat dihubungkan dan diimplementasikan sesuai dengan fungsi kerja masing-masing operasi layanan. Dalam penelitian yang berjudul “*Refactoring Arsitektur Microservice Pada PT*”. Graha Usaha Teknik” menyimpulkan bahwa arsitektur *microservices* dapat mengungguli arsitektur monolitik dalam hal waktu respon pemulihan[17].

Berdasarkan uraian tersebut, peneliti termotivasi untuk mengimplementasikan arsitektur *microservice* pada sistem informasi desa. Pada dasarnya dengan menggunakan sistem informasi berbasis *microservices* berarti membagi aplikasi menjadi layanan yang lebih kecil dan saling terhubung. *Microservices* memungkinkan tiap fitur pada aplikasi mengalami pengembangan tersendiri. Pola arsitektur *microservices* secara signifikan memengaruhi hubungan antara aplikasi dan database. Di dalam *microservices*, masing-masing *services* atau layanan memiliki skema database tersendiri. Selain itu, *services* dapat menggunakan jenis database dan bahasa pemrograman yang paling sesuai dengan keperluan sistem informasi desa.

Dengan begitu, tiap-tiap layanan desa akan lebih optimal. Lebih lanjut adanya *microservices* memungkinkan sistem informasi desa menjadi lebih padat dan kompleks namun tetap ringan [18]. Intinya, *microservices* adalah metode dengan membagi *services* ke bagian yang lebih kecil namun tetap berkaitan. Selain itu, dalam setiap *services* yang dibuat bisa menggunakan teknologi yang berbeda pula.

Sistem informasi tingkat desa berbasis *microservice* ini merupakan pengembangan dari sistem yang pada dasarnya telah terbentuk dan dapat membantu kader desa dalam mengelola pengelolaan kependudukan khususnya kegiatan penyusunan, pengumpulan dan pengendalian data dan dokumen kependudukan. Melalui pencatatan sipil, pencatatan kependudukan,

pengelolaan informasi kependudukan dan pemanfaatan hasilnya untuk pelayanan publik dan pembangunan sektoral lainnya [19].

2. Metode Penelitian



Gambar 2. Alur Penelitian

Penelitian ini mengadopsi metode pengembangan Web Service Implementation Methodology (WSIM). Tahapan pengembangan sistem-informasi desa berbasis microservice, dari tahap requirements hingga tahap deployment [14].

Analisa Kebutuhan

Dalam penelitian ini wawancara dengan kepala desa Pengalangan dilakukan pada tahap kebutuhan. Wawancara digunakan untuk mengidentifikasi kebutuhan bisnis dalam sistem-informasi desa.

Analisis Arsitektur

Dalam analisis, hal-hal yang harus dilakukan adalah menentukan arsitektur layanan Web yang akan dibangun, memilih platform yang akan digunakan, memilih teknologi untuk meng-host layanan Web, dan memilih bahasa pemrograman yang akan diterapkan.

Desain

Desain adalah fase yang dilakukan dengan melakukan desain URI, pemodelan REST API, dan ERD.

Pengkodean

Tahap coding merupakan hasil dari tahap design yang dikembangkan menjadi web service. Web service ini dibangun menggunakan aplikasi Visual Studio Code, framework laravel dan framework nuxt.js.

Uji Coba

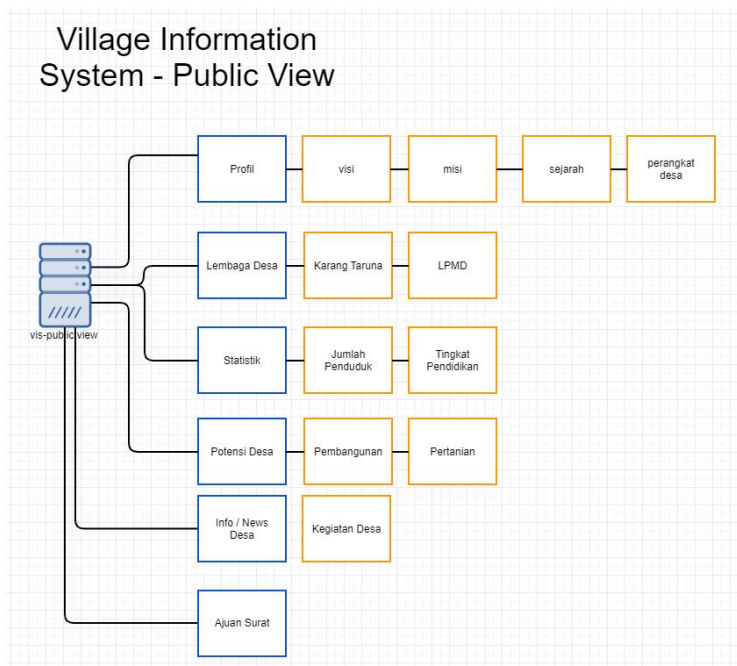
Tahap uji coba adalah tahap dimana endpoint dari web service diuji menggunakan software Postman. Pengujian dilakukan dengan menggunakan pendekatan pengujian black-box, dimana sistem diuji terhadap spesifikasi fungsional [15].

3. Hasil dan Pembahasan

Berikut adalah hasil dan pembahasan berdasarkan metode penelitian yang dilakukan :

Analisa kebutuhan

Hasil dari tahapan ini yaitu beberapa modul yang dibutuhkan berdasarkan hasil wawancara.



Gambar 3. *Kebutuhan Sistem Desa*

Berdasarkan Gambar 3 kebutuhan sistem desa dapat dijabarkan menjadi beberapa menu yaitu Pendahuluan, Kelembagaan Desa, Statistika, Potensi Desa, Informasi Berita dan Penyampaian Surat [20]. Terdapat beberapa submenu pada menu arsip detail yaitu Visi, Misi, Sejarah, dan Perangkat Desa. Pada menu Kelembagaan Desa terdapat submenu Karang Taruna dan LPMD. Pada menu Statistik terdapat submenu Kependudukan dan Pendidikan. Terdapat submenu Pembangunan dan Pertanian pada menu Potensi Desa. Analisa kebutuhan secara detail dapat dilihat pada tabel 1:

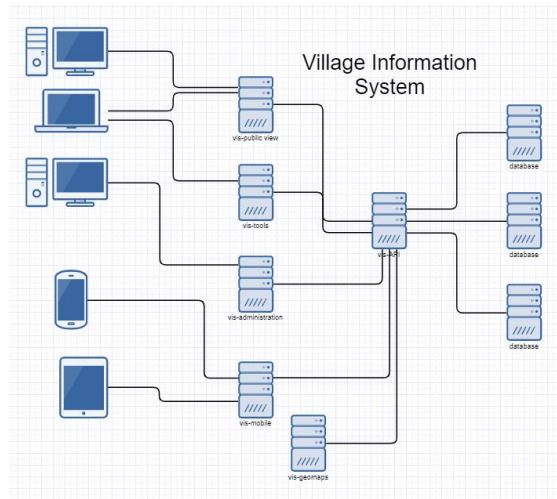
Tabel 1. *Analisa Kebutuhan*

ID	Aktor	Functional Requirement
1	Admin	Insert data desa
2		Edit & Delete data desa
3		Insert data berita desa
4		Edit & Delete data berita desa
5		Insert data perangkat desa
6		Edit & Delete data perangkat desa
7		Insert data user
8	Perangkat Desa	Edit & Delete data user
9		Insert data kartu keluarga
10		Edit & Delete data kartu keluarga
11		Insert data jenis Lembaga desa
12		Edit & Delete data jenis Lembaga_desa
13		Insert data Lembaga desa
14		Edit & Delete data Lembaga_desa
15		Insert data jenis potensi desa
16		Edit & Delete data jenis potensi desa
17		Insert data jenis surat
18		Edit & Delete data jenis surat
19		Insert data master surat
20		Edit & Delete data master surat
21		Insert data surat keluar
22		Edit & Delete data surat keluar
23		Insert data surat masuk
24		Edit & Delete data surat masuk
25		Insert data penduduk

26		Edit & Delete data penduduk
27	warga	Akses data profil desa
28		Akses data berita desa
29		Akses data data penduduk
30		Akses data ajuan surat

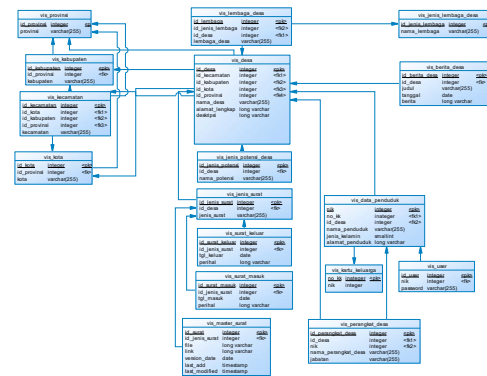
Analisa arsitektur

Tujuan dari tahap ini adalah untuk menguraikan cara kerja sistem. Kerangka sistem informasi desa secara keseluruhan dibagi menjadi dua modul, yaitu tampilan publik dari aplikasi web dan sistem berbasis web sebagai backend. Arsitektur sistem informasi desa ditunjukkan pada Gambar 4.



Gambar 4. Modul Sistem-informasi Desa

Desain



Gambar 5. Entity Relationship Diagram Sistem-informasi Desa

Pada gambar 5 merupakan entity relationship diagram dimana menggambarkan rancangan tabel beserta hubungan relasi entitasnya.

Coding

Pada tahapan ini, dilakukan pembuatan endpoint API menggunakan framework Laravel. Proses ini dimulai dari pembuatan service-service yang berkaitan dengan sistem desa hingga membuat API gateway.

Testing

Pengujian dilakukan dalam bentuk functional testing guna dilakukan Analisa pada endpoint yang telah dibuat. Functional testing merupakan bagian dari black box testing dimana dalam

prosesnya yaitu memastikan sistem dapat berfungsi sesuai model. Pengujian endpoint menggunakan tools postman. Black box testing adalah pengujian perangkat lunak dalam hal spesifikasi fungsional, tanpa pengujian desain dan kode program". Pengujian dirancang untuk mengetahui apakah fungsi, input, dan output perangkat lunak memenuhi spesifikasi yang diperlukan. Beberapa API gateway yang berhasil dibuat tersaji pada tabel 2

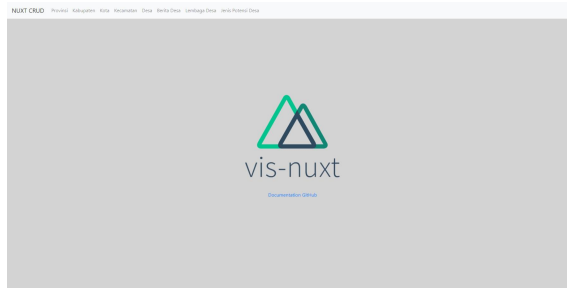
Tabel 2. Gateway API

No.	method	uri
1	GET	/
2	POST	api/berita_desa
3	GET	api/berita_desa
4	GET	api/berita_desa/{berita_desa}
5	DELETE	api/berita_desa/{berita_desa}
6	PUT PATCH	api/berita_desa/{berita_desa}
7	POST	api/data_penduduk
8	GET	api/data_penduduk
9	DELETE	api/data_penduduk/{data_penduduk}
10	GET	api/data_penduduk/{data_penduduk}
11	PUT PATCH	api/data_penduduk/{data_penduduk}
12	GET	api/desa
13	POST	api/desa
14	GET	api/desa/{desa}
15	DELETE	api/desa/{desa}
16	PUT PATCH	api/desa/{desa}
17	GET	api/jenis_lembaga_desa
18	POST	api/jenis_lembaga_desa
19	DELETE	api/jenis_lembaga_desa/{jenis_lembaga_desa}
20	GET	api/jenis_lembaga_desa/{jenis_lembaga_desa}
21	PUT PATCH	api/jenis_lembaga_desa/{jenis_lembaga_desa}
22	GET	api/jenis_potensi_desa
23	POST	api/jenis_potensi_desa
24	GET	api/jenis_potensi_desa/{jenis_potensi_desa}
25	PUT PATCH	api/jenis_potensi_desa/{jenis_potensi_desa}
26	DELETE	api/jenis_potensi_desa/{jenis_potensi_desa}

Untuk pengujian endpoint, dilakukan pembuatan aplikasi frontend menggunakan Nuxt framework.

Halaman Antarmuka Utama

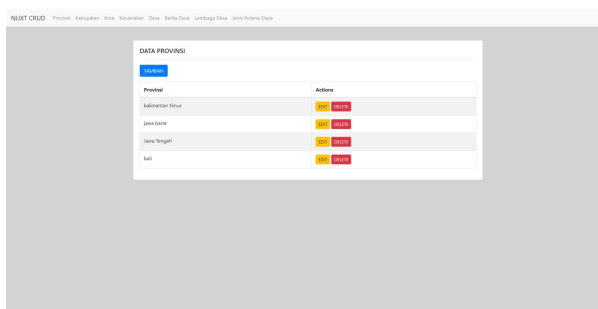
Halaman ini merupakan halaman utama pengujian API. Pada antarmuka ini memuat beberapa link untuk menuju pada link antarmuka lainnya.



Gambar 6. Halaman Antarmuka Utama

Halaman provinsi

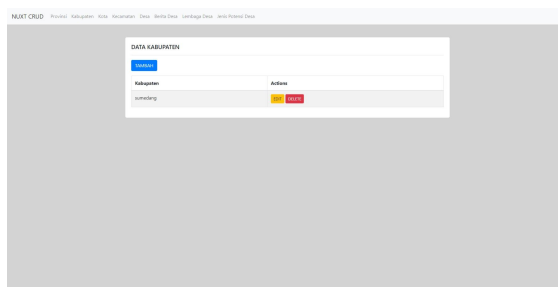
Pada antarmuka ini dilakukan pengujian endpoint yang berfungsi untuk GET, POST, PUT dan DELETE data provinsi



Gambar 7. Halaman Provinsi

Halaman Antarmuka Kabupaten

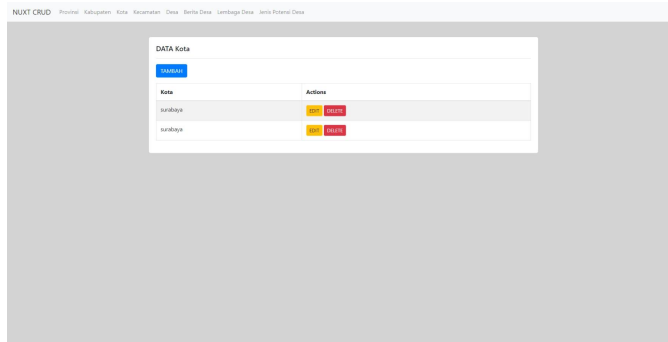
Pada antarmuka ini dilakukan pengujian endpoint yang berfungsi untuk GET, POST, PUT dan DELETE data Kabupaten



Gambar 8. Halaman Antarmuka Kabupaten

Halaman Antarmuka Kota

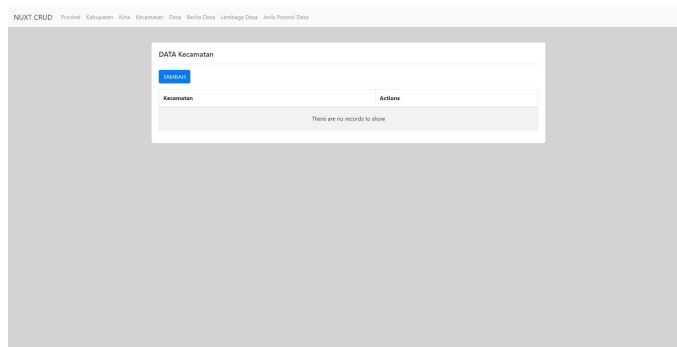
Pada antarmuka ini dilakukan pengujian endpoint yang berfungsi untuk GET, POST, PUT dan DELETE data Kota



Gambar 9. Halaman Antarmuka Kota

Halaman Antarmuka Kecamatan

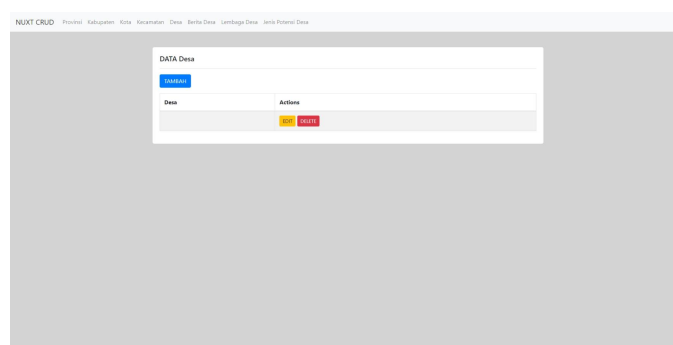
Pada antarmuka ini dilakukan pengujian endpoint yang berfungsi untuk GET, POST, PUT dan DELETE data Kecamatan



Gambar 10. Halaman Antarmuka Kecamatan

Halaman Antarmuka Desa

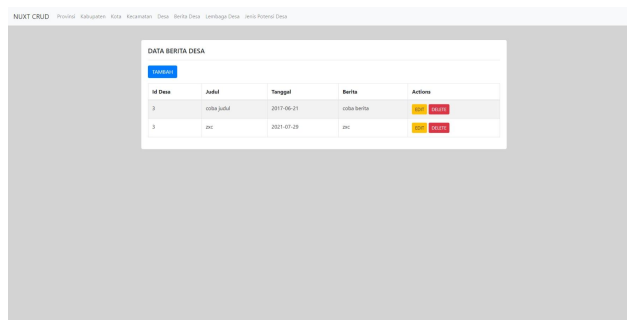
Pada antarmuka ini dilakukan pengujian endpoint yang berfungsi untuk GET, POST, PUT dan DELETE data Desa



Gambar 11. Halaman Antarmuka Desa

Halaman Antarmuka Berita Desa

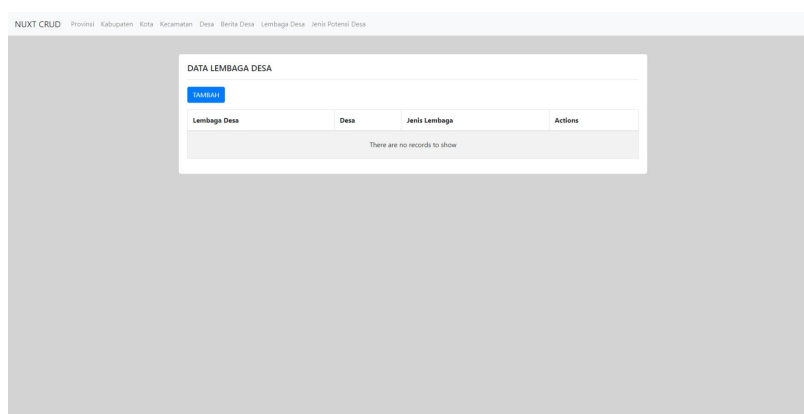
Pada antarmuka ini dilakukan pengujian endpoint yang berfungsi untuk GET, POST, PUT dan DELETE data Berita Desa



Gambar 12. Antarmuka Berita Desa

Halaman Antarmuka Lembaga Desa

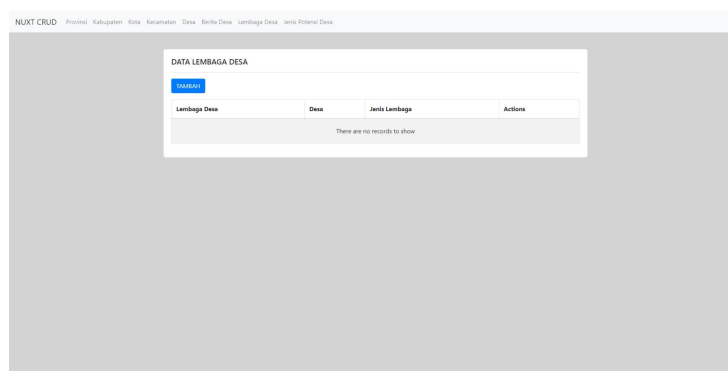
Pada antarmuka ini dilakukan pengujian endpoint yang berfungsi untuk GET, POST, PUT dan DELETE data Lembaga Desa



Gambar 13. Halaman Antarmuka Lembaga Desa

Halaman Antarmuka Jenis Potensi Desa

Pada antarmuka ini dilakukan pengujian endpoint yang berfungsi untuk GET, POST, PUT dan DELETE data Jenis Potensi Desa



Gambar 14. Halaman Antarmuka Jenis Potensi Desa

4. Penutup

Hasil penelitian ini menyimpulkan bahwa penelitian ini membangun API berbasis microservice untuk sistem informasi pedesaan. Menganalisis dan mengembangkan pelayanan berdasarkan kebutuhan dasar sistem informasi desa. Arsitektur layanan mikro dapat menguraikan layanan sistem menjadi layanan independen dan memanggilnya melalui gateway API. Sistem Informasi Desa memiliki 86 URL dan dapat digunakan untuk mengembangkan sistem informasi desa berbasis mobile. Kerangka kerja Laravel dapat mendukung pengembangan RESTFUL API.

5. Referensi

- [1] R. Akbar, "Pembangunan Aplikasi Web dengan Fitur Mobile untuk Sistem Informasi Administrasi Kependudukan Pada Kantor Wali Nagari Pagaruyung," *Semin. Nas. Sains dan Teknol.* 2017, 2017.
- [2] R. Efrianza, Hendrawan, and A. Sunoto, "Perancangan Sistem Informasi Kependudukan pada Kantor Kepala Desa Tanjung Putra," *J. Ilm. Mhs. Sist. Inf.*, 2019.
- [3] W. D. Prasetyo, "Sistem Informasi Inventaris Desa berbasis Web," *Electron. These Diss. Univ. Muhammadiyah Surakarta*, 2017.
- [4] A. Kusumawati, T. Hendro Pudjiantoro, and D. Nursantika, "Sistem Informasi Kependudukan Pada Kecamatan Kadungora Kabupaten Garut," *J. Sisfotek Glob.*, 2017.
- [5] T. Prasetyo and R. P. Dhaniawaty, "Sistem Informasi Administrasi Pemerintahan Desa pada Desa Cilayung Kabupaten Kuningan," *J. Teknol. dan Inf.*, 2020.
- [6] S. Hidayat and S. Noor, "Pengembangan Sistem Informasi Desa Terintegrasi," *Global*, 2018.
- [7] R. Guntari and R. Setiawan, "Rancang Bangun Aplikasi Pengelolaan Surat di Desa Tanjung Kamuning," *J. Algoritma.*, 2017, doi: 10.33364/algoritma/v.13-2.269.
- [8] S. Syaharuddin and M. Ibrahim, "Aplikasi Sistem Informasi Desa Sebagai Teknologi Tepat Guna Untuk Pendataan Penduduk Dan Potensi Desa," *JMM (Jurnal Masy. Mandiri)*, 2017, doi: 10.31764/jmm.v1i1.14.
- [9] R. Kurniati, J. Jaroji, and A. Agustiawan, "Sistem Layanan Mandiri Di Kantor Desa Berbasis Web," *INOVTEK Polbeng - Seri Inform.*, 2018, doi: 10.35314/isi.v3i1.326.
- [10] C. Setya Budi and A. M. Bachtiar, "Implementasi Arsitektur Microservices pada Backend Comrades," *Progr. Stud. Tek. Inform. Univ. Komput. Indones.*, 2018.
- [11] Anggi Elanda, Jaka Abdul Haris, Darmansyah, Donny Apdian, "Sistem Informasi Pelayanan Administrasi Kependudukan Desa (SIPAKDE) Berbasis Web Dengan Framework Codeigniter," *J. Interkom*, 2020, doi: 10.35969/interkom.v14i4.58.
- [12] I. Budiana and I. Wibiyanti, "Perancangan Arsitektur Sistem Tiket Elektronik Kereta Api Menggunakan Kerangka Service Oriented Enterprise Architecture (Studi Kasus: Pt Railink)," *Inf. (Jurnal Inform. dan Sist. Informasi)*, vol. 11, no. 2, 2019, doi: 10.37424/informasi.v11i2.19.
- [13] R. Mufrizal and D. Indarti, "Refactoring Arsitektur Microservice Pada Aplikasi Absensi PT. Graha Usaha Teknik," *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 1, 2019, doi: 10.25077/teknosi.v5i1.2019.57-68.
- [14] G. Munawar and A. Hodijah, "Analisis Model Arsitektur Microservice Pada Sistem Informasi DPLK," *Sink. J. dan Penelit. Tek. Inform.*, vol. 3, no. 1, 2018.
- [15] S. Dharma Handayani and U. Uminingsih, "Pengorganisasian Kerja Sistem Parkir Menggunakan Arsitektur Microservice," *J. Teknol.*, vol. 13, no. 1, 2020.
- [16] M. D. Rafiqi, E. Subyantoro, and D. K. W, "Implementasi Arsitektur Microservice Pada Aplikasi Online Travel Tourinc," *Karya Ilm. Mhs. Manaj. Inform.*, vol. 1, no. 1, 2019.
- [17] F. Tsukamoto and L. Processing, "Pengolah Bahasa Alami Sebagai Query Fuzzy Tes Darah," *Semin. Nas. Teknol. Inf. dan Multimed.*, 2016.
- [18] C. Kesuma and M. D. Juniati, "Sistem Informasi Administrasi Kependudukan Desa (SIAKSA) Berbasis Web pada Desa Alangamba Kabupaten Cilacap," *J. Speed-Sentra Penelit. Eng. dan Edukasi*, 2020.

- [19] Paryanta, Sutariyani, and S. Desi, "Sistem Informasi Administrasi Kependudukan Berbasis Web Desa Sawahan," *IJSE - Indones. J. Softw. Eng.*, 2017.
- [20] S. R. S. Siregar and P. Sundari, "Rancangan Sistem Informasi Pengelolaan Data Kependudukan Desa (Studi Kasus di Kantor Desa Sangiang Kecamatan Sepatan Timur)," *Sisfotek Glob.*, 2016.