

IMPLEMENTASI ARSITEKTUR MICROSERVICE PADA SISTEM INFORMASI DESA

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

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

² Sistem Informasi, Institut Teknologi Adhi Tama Surabaya

*Email: ¹suryoatmojo@uwp.ac.id

ABSTRAK

Sistem informasi dapat dikatakan baik yaitu sistem dapat memberikan kemudahan dalam penyajian informasi, juga harus dapat bertahan saat mengalami gangguan dan mampu beradaptasi. Kemampuan dalam menghadapi gangguan serta beradaptasi dinamakan resiliensi. Salah satu arsitektur sistem informasi yang kurang memiliki kemampuan resiliensi yaitu arsitektur monolitik, dimana backend sistem dan frontend dijadikan satu dalam fungsi serta services yang sama, selain itu arsitektur monolitik juga menjalankan semua logika dalam satu server aplikasi. Arsitektur microservice memungkinkan pengembangan fungsi pada aplikasi atau sistem informasi dipecah menjadi beberapa bagian service yang kecil dan menjadikan service memiliki kemampuan sumber daya yang bisa diatur dengan tepat dan menjadi layanan yang kecil dan saling terhubung menjadi satu. Perancangan Sistem Informasi Desa berbasis microservice ini merupakan solusi terbaik untuk memecahkan permasalahan-permasalahan yang ada pada sistem informasi berbasis arsitektur monolitik khususnya pada sistem informasi desa. Pada penelitian ini, studi kasus berfokus pada permasalahan pada Sistem Informasi Desa Pengalangan seperti, kebutuhan sistem yang cepat berubah, kebutuhan sistem yang dapat dengan mudah diperluas, aksesibilitas yang hanya tergantung pada satu teknologi, tumpukan administrasi, file yang semakin banyak, pembuatan laporan yang membutuhkan waktu dikarenakan proses pengumpulan data masih manual, permasalahan keamanan sistem bagi desa yang sudah menggunakan sistem informasi, dan pengembangan sistem yang tidak mudah dikarenakan harus merombak sistem yang sudah ada. Dengan menggunakan arsitektur microservice, kinerja pelayanan perangkat desa akan meningkat dan menjadi lebih cepat, mudah untuk dilakukan perluasan proses bisnis pada sistem, serta keamanan sistem terjaga.

Kata Kunci: Microservice; Sistem; Informasi; Desa

PENDAHULUAN

Sistem informasi dapat dikatakan baik jika sistem tersebut dapat memberikan kemudahan dalam menyajikan informasi, selain itu, sistem tersebut juga harus dapat bertahan saat mengalami ancaman sistem dan mampu beradaptasi dari gangguan sistem. Resiliensi adalah kemampuan sistem dalam menghadapi ancaman serta kemampuan beradaptasi. Salah satu contoh arsitektur sistem informasi yang kurang baik dalam kemampuan resiliensi yaitu arsitektur monolitik, dimana backend dan frontend dijadikan satu pada fungsi atau service yang sama, serta menjalankan semua logika dalam satu server sistem atau aplikasi [1]. Hal ini dapat mempengaruhi ketergantungan antar setiap komponen. Sehingga, jika terdapat kesalahan pada salah satu komponen akan mempengaruhi keseluruhan sistem dan jika ingin melakukan perubahan teknologi, maka akan merubah keseluruhan isi dari sistem. Aplikasi yang dibangun dengan arsitektur ini dapat memiliki ukuran sangat besar

dikarenakan struktur aplikasi dibangun secara kompleks [2]. Salah satu masalah yang paling sering muncul pada arsitektur monolithic yaitu resilient challenges, yaitu kegagalan pada proses pembaharuan dan penambahan fitur pada sistem sehingga jika terjadi pada kegagalan pada proses penambahan fitur, maka seluruh fitur aplikasi akan mengalami kegagalan [3]. Akhir ini muncul istilah arsitektur microservices sebagai arsitektur pengganti monolitik. Arsitektur microservices adalah dimana sistem dirancang untuk membagi service lebih kecil dan lebih fleksibel [4]. Software akan dirancang untuk menjalankan fungsi secara independen. Artinya, setiap permasalahan teknis dapat diselesaikan dengan cara dan teknologi yang berbeda beda yang nantinya akan dihubungkan dengan Application Programming Interface (API) [5].

Microservices berarti membagi aplikasi menjadi layanan yang lebih kecil dan saling

terhubung. Arsitektur *microservices* merupakan gaya arsitektur sistem atau aplikasi yang membagi fitur dan memungkinkan tiap fitur pada aplikasi mengalami pengembangan tersendiri [6]. Bentuk arsitektur *microservices* memengaruhi hubungan antara aplikasi dan basisdata. Pada kerangka *microservices*, masing-masing *services* atau layanan memiliki skema basisdata tersendiri. Selain itu, *services* dapat menggunakan jenis basisdata dan bahasa pemrograman yang dinilai tepat sesuai dengan keperluan. Dengan begitu, tiap-tiap layanan akan lebih optimal. Dengan kerangka arsitektur *microservices* memungkinkan aplikasi menjadi lebih padat dan kompleks namun tetap ringan. Kesimpulannya, *microservices* adalah metode dengan membagi *services* menjadi bagian yang lebih kecil namun tetap saling berkaitan. Selain itu, pada setiap *services* yang dibuat dapat menggunakan teknologi yang berbeda. Pada implementasinya, *service* yang dibangun memiliki interface yang berfungsi memproses dan mengirim pesan dalam bentuk XML atau JSON. *Service* yang diberikan sebagai output menggunakan konsep *object oriented* pada proses pengiriman *request* maupun responsnya [7].

Pada Penelitian sebelumnya dengan judul

“Arsitektur *Microservice* untuk Resiliensi Sistem Informasi” dihasilkan bahwa implementasi *microservices* berhasil meningkatkan resiliensi sistem informasi[5]. Berikutnya pada penelitian dengan judul “Pengorganisasian kerja sistem parkir menggunakan arsitektur *microservice* disimpulkan bahwa setiap modul sistem dapat beroperasi menggunakan Bahasa pemrograman yang berbeda-beda namun dapat dikoneksikan dan dapat beroperasi sesuai dengan fungsi kerja masing-masing *service* [8]. Pada penelitian yang berjudul “refactoring arsitektur *microservice* pada aplikasi absensi PT. Graha Usaha Teknik” dihasilkan kesimpulan arsitektur *microservice* dapat mengungguli arsitektur *monolithic* dalam mengembalikan *response time* [9].

Berdasarkan uraian tersebut peneliti termotifasi untuk membuat Implementasi Arsitektur *Microservices* pada Sistem Informasi Desa. Pada dasarnya, dengan memanfaatkan sistem informasi, perangkat desa bisa berinteraksi dan bertukar informasi dengan warga, sebagai bentuk langkah

proses kontrol dalam pengambilan sebuah keputusan, juga digunakan sebagai laporan kepada instansi yang terkait [10] sehingga perangkat desa mampu memperlancar jalannya pemerintah desa didalam memberikan pelayanan dan informasi kepada masyarakat [11]. Salah satu indikator yang dapat digunakan untuk mengukur sebuah daerah atau Desa dikatakan maju jika secara data administratif terekam dengan baik, meliputi data kependudukan, pendidikan, ekonomi, penggunaan lahan [12].

Sistem informasi desa berbasis *microservice* ini merupakan pengembangan dari sistem yang secara umum telah banyak di buat dan mampu membantu pekerjaan perangkat desa dalam mengelola administrasi kependudukan yang secara detail adalah kegiatan dalam penyusunan, penataan dan penertiban data dan dokumen penduduk yang diperoleh melalui pencatatan sipil, pendaftaran penduduk, pengelolaan informasi kependudukan, dan penggunaan hasilnya untuk pelayanan publik dan pembangunan sektor lainnya [13].

METODE PENELITIAN

Penelitian ini menggunakan metode pengembangan *Web Service Implementation Methodology (WSIM)*. Tahapan pengembangan sistem informasi desa berbasis *microservice*, dimulai dengan tahapan *requirements* hingga tahap *deployment*[14].

1. Requirements

Pada penelitian ini, tahap *requirements* dilakukan wawancara kepada perangkat desa Pengalangan. Wawancara digunakan untuk mengidentifikasi kebutuhan bisnis pada Sistem Informasi Desa.

2. Analysis

Pada *analysis* hal hal yang dilakukan adalah menentukan arsitektur dari *web service* yang akan dibangun, memilih platform yang akan digunakan, memilih teknologi untuk hosting *web service*, dan memilih bahasa pemrograman untuk implementasi.

3. Design

Tahap *design* adalah tahap yang dilakukan dengan cara membuat desain URI, membuat *modelling REST API*, dan membuat ERD.

4. Coding

Tahap coding merupakan hasil dari tahap design dikembangkan menjadi web service. Web service dibuat dengan aplikasi Visual Studio Code, framework laravel dan framework nuxt.js.

5. Test

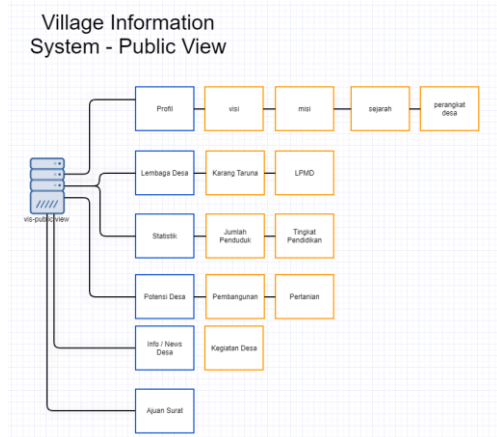
Tahap test merupakan tahap dilakukan pengujian web service menggunakan software postman. Tes dilakukan menggunakan metode black box testing dimana pengujian sistem dari segi spesifikasi fungsional[15].

HASIL DAN PEMBAHASAN

Berikut adalah hasil dan pembahasan berdasarkan metode penelitian yang dilakukan :

1. Analisa kebutuhan

Pada tahapan ini didapati beberapa modul yang dibutuhkan berdasarkan hasil wawancara.



Gambar 1. Kebutuhan Sistem Desa

Berdasarkan gambar 1, kebutuhan sistem desa dapat dijabarkan menjadi beberapa menu yaitu profil, Lembaga desa, statistic, potensi desa, info news dan ajuan surat. Pada menu profil secara detail terdapat beberapa submenu yaitu visi, misi, sejarah, dan perangkat desa. Pada menu Lembaga desa terdapat submenu karang taruna dan LPMD. Pada menu statistik terdapat submenu jumlah penduduk dan tingkat Pendidikan. Pada menu potensi desa terdapat submenu pembangunan dan pertanian.

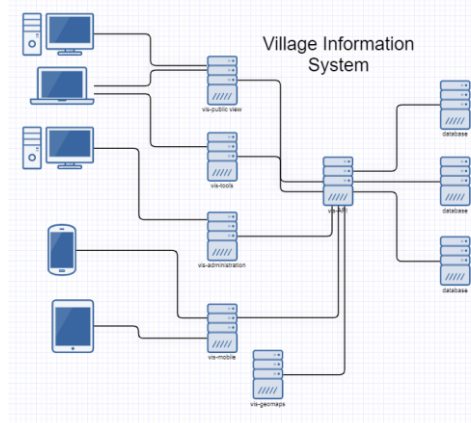
Analisa kebutuhan secara detail dapat dilihat pada Tabel 1 :

Tabel 1. Analisa Kebutuhan

ID	Aktor	Functional Requirement
1	Admin	Memasukkan data desa
2		Menghapus dan mengedit data desa
3		Memasukkan data berita desa
4		Menghapus dan mengedit data berita desa
5		Memasukkan data perangkat desa
6		Menghapus dan mengedit data perangkat desa
7		Memasukkan data user
8		Menghapus dan mengedit data user
9	Perangkat Desa	Memasukkan data kartu keluarga
10		Menghapus dan mengedit data kartu keluarga
11		Memasukkan data jenis Lembaga desa
12		Menghapus dan mengedit data jenis Lembaga desa
13		Memasukkan data Lembaga desa
14		Menghapus dan mengedit data Lembaga desa
15		Memasukkan data jenis potensi desa
16		Menghapus dan mengedit data jenis potensi desa
17		Memasukkan data jenis surat
18		Menghapus dan mengedit data jenis surat
19		Memasukkan data master surat
20		Menghapus dan mengedit data master surat
21	Memasukkan data surat keluar	
22	Menghapus dan mengedit data surat keluar	
23	Memasukkan data surat masuk	
24	Menghapus dan mengedit data surat masuk	
25	Memasukkan data penduduk	
26	Menghapus dan mengedit data penduduk	
27	warga	Mengakses profil desa
28		Mengakses berita desa
29		Mengakses data penduduk
30		Mengakses ajuan surat

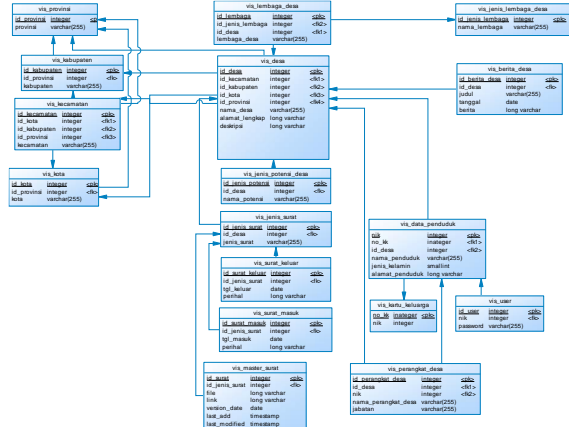
2. Analisa arsitektur

Analisa arsitektur bertujuan untuk memberikan gambaran secara umum tentang cara kerja sistem. Kerangka kerja sistem informasi desa secara utuh dibagi menjadi dua komponen yaitu aplikasi public view pada web atau mobile android untuk masyarakat desa serta perangkat desa dan sistem berbasis web sebagai backend. Secara sederhana arsitektur sistem informasi desa dapat dilihat pada gambar 2



Gambar 2. Rancangan Sistem Informasi Desa

3. Desain



Gambar 3. ERD Sistem Informasi Desa

Pada gambar 3 merupakan entity relationship diagram dimana menggambarkan rancangan tabel beserta hubungan relasi entitasnya.

4. 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.

5. 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.

API gateway yang berhasil dibuat tersaji pada table 2

Tabel 2. Gateway API

N o.	method	uri
1	GET HEAD	/
2	POST	api/beritadesa
3	GET HEAD	api/beritadesa
4	GET HEAD	api/beritadesa/{beritadesa}
5	DELETE	api/beritadesa/{beritadesa}
6	PUT PATCH	api/beritadesa/{beritadesa}
7	POST	api/datapenduduk
8	GET HEAD	api/datapenduduk
9	DELETE	api/datapenduduk/{datapenduduk}
10	GET HEAD	api/datapenduduk/{datapenduduk}
11	PUT PATCH	api/datapenduduk/{datapenduduk}
12	GET HEAD	api/desa
13	POST	api/desa
14	GET HEAD	api/desa/{desa}
15	DELETE	api/desa/{desa}
16	PUT PATCH	api/desa/{desa}
17	GET HEAD	api/jenislembagadesa
18	POST	api/jenislembagadesa
19	DELETE	api/jenislembagadesa/{jenislembagadesa}
20	GET HEAD	api/jenislembagadesa/{jenislembagadesa}
21	PUT PATCH	api/jenislembagadesa/{jenislembagadesa}
22	GET HEAD	api/jenispotensidesa

23	POST	api/jenispotensidesa
	GET HE	api/jenispotensidesa/{jenispo
24	AD	tensidesa}
	PUT PA	api/jenispotensidesa/{jenispo
25	TCH	tensidesa}
	DELETE	api/jenispotensidesa/{jenispo
26		tensidesa}
27	POST	api/jenissurat
	GET HE	api/jenissurat
28	AD	
	PUT PA	api/jenissurat/{jenissurat}
29	TCH	
	GET HE	api/jenissurat/{jenissurat}
30	AD	
	DELETE	api/jenissurat/{jenissurat}
31		
32	POST	api/kabupaten
	GET HE	api/kabupaten
33	AD	
	PUT PA	api/kabupaten/{kabupaten}
34	TCH	
	GET HE	api/kabupaten/{kabupaten}
35	AD	
	DELETE	api/kabupaten/{kabupaten}
36		
37	POST	api/kartukeluarga
	GET HE	api/kartukeluarga
38	AD	
	DELETE	api/kartukeluarga/{kartukelu
39		arga}
	PUT PA	api/kartukeluarga/{kartukelu
40	TCH	arga}
	GET HE	api/kartukeluarga/{kartukelu
41	AD	arga}
	POST	api/kecamatan
42		
	GET HE	api/kecamatan
43	AD	
	DELETE	api/kecamatan/{kecamatan}
44		
	PUT PA	api/kecamatan/{kecamatan}
45	TCH	
	GET HE	api/kecamatan/{kecamatan}
46	AD	
	POST	api/kota
47		
	GET HE	api/kota
48	AD	
	DELETE	api/kota/{kotum}
49		
	PUT PA	api/kota/{kotum}
50	TCH	
	GET HE	api/kota/{kotum}
51	AD	
	GET HE	api/lembagadesa
52	AD	
53	POST	api/lembagadesa

	GET HE	api/lembagadesa/{lembagade
54	AD	sa}
	DELETE	api/lembagadesa/{lembagade
55		sa}
	PUT PA	api/lembagadesa/{lembagade
56	TCH	sa}
	GET HE	api/mastersurat
57	AD	
	POST	api/mastersurat
58		
	GET HE	api/mastersurat/{mastersurat
59	AD	}
	PUT PA	api/mastersurat/{mastersurat
60	TCH	}
	DELETE	api/mastersurat/{mastersurat
61		}
	GET HE	api/perangkatdesa
62	AD	
	POST	api/perangkatdesa
63		
	GET HE	api/perangkatdesa/{perangka
64	AD	tdesa}
	DELETE	api/perangkatdesa/{perangka
65		tdesa}
	PUT PA	api/perangkatdesa/{perangka
66	TCH	tdesa}
	GET HE	api/provinsi
67	AD	
	POST	api/provinsi
68		
	DELETE	api/provinsi/{provinsi}
69		
	PUT PA	api/provinsi/{provinsi}
70	TCH	
	GET HE	api/provinsi/{provinsi}
71	AD	
	POST	api/suratkeluar
72		
	GET HE	api/suratkeluar
73	AD	
	GET HE	api/suratkeluar/{suratkeluar}
74	AD	
	PUT PA	api/suratkeluar/{suratkeluar}
75	TCH	
	DELETE	api/suratkeluar/{suratkeluar}
76		
	POST	api/suratmasuk
77		
	GET HE	api/suratmasuk
78	AD	
	DELETE	api/suratmasuk/{suratmasuk}
79		
	PUT PA	api/suratmasuk/{suratmasuk}
80	TCH	
	GET HE	api/suratmasuk/{suratmasuk}
81	AD	
	POST	api/user
82		
	GET HE	api/user
83	AD	
84	GET HE	api/user/{user}

	AD	
	PUT PA	
85	TCH	api/user/{user}
86	DELETE	api/user/{user}

KESIMPULAN

Kesimpulan yang dihasilkan dari penelitian ini yaitu :

1. Pada penelitian ini telah dibangun layanan API berbasis microservice untuk sistem informasi desa.
2. Layanan yang dikembangkan sudah sesuai dengan Analisa kebutuhan dasar sistem informasi desa.
3. Arsitektur mikroservis dapat memecah servis dari sistem menjadi independent dimana dapat dipanggil melalui API gateway.
4. Sistem informasi desa memiliki 86 endpoint yang dapat dikembangkan dengan menambahkan endpoint disesuaikan dengan service yang dibutuhkan
5. Framework Laravel dapat mendukung pengembangan RESTFUL API.

DAFTAR PUSTAKA

- [1] I. Redbooks, *Microservices from Theory to Practice*. 2015.
- [2] N. Dragoni *et al.*, "Microservices: Yesterday, today, and tomorrow," in *Present and Ulterior Software Engineering*, 2017.
- [3] K. Katuwal, "Microservices : A Flexible Architecture for the Digital Age Version 1 . 1," *Am. J. Comput. Sci. Eng.*, vol. 3, no. 4, 2016.
- [4] G. Munawar and A. Hodijah, "Analisis Model Arsitektur Microservice Pada Sistem Informasi DPLK," *Sink. J. dan Penelit. Tek. Inform.*, vol. 3, no. 1, 2018.
- [5] A. Garindra, T. Wati, and I. W. Widi P, "Perancangan Arsitektur Microservices Untuk Resiliensi Sistem Informasi Perpustakaan Pusat (Studi Kasus UPN Veteran Jakarta)," *Format J. Ilm. Tek. Inform.*, vol. 9, no. 2, 2021, doi: 10.22441/format.2020.v9.i2.001.
- [6] A. Messina, R. Rizzo, P. Storniolo, and A. Urso, "A Simplified Database Pattern for the Microservice Architecture," *Eighth Int. Conf. Adv. Databases, Knowledge, Data Appl.*, no. June, 2016.
- [7] C. Setya Budi and A. M. Bachtiar, "Implementasi Arsitektur Microservices pada Backend Comrades," *Progr. Stud. Tek. Inform. Univ. Komput. Indones.*, 2018.
- [8] S. Dharma Handayani and U. Uminingsih, "Pengorganisasian Kerja Sistem Parkir Menggunakan Arsitektur Microservice," *J. Teknol.*, vol. 13, no. 1, 2020.
- [9] 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.
- [10] J. A. H. D. D. A. Anggi Elanda, "SISTEM INFORMASI PELAYANAN ADMINISTRASI KEPENDUDUKAN DESA (SIPAKDE) BERBASIS WEB DENGAN FRAMEWORK CODEIGNITER," *J. Interkom*, 2020, doi: 10.35969/interkom.v14i4.58.
- [11] Paryanta, Sutariyani, and S. Desi, "Sistem Informasi Administrasi Kependudukan Berbasis Web Desa Sawahan," *IJSE - Indones. J. Softw. Eng.*, 2017.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.