



Pengembangan dan Implementasi Teknologi Serverless Computing pada Sistem Informasi Data Kependudukan Desa dengan Pendekatan Metode Agile

Muhammad Dimas Haryo Saputra, Teduh Dirgahayu*

Fakultas Teknik Industri, Program Studi Informatika, Universitas Islam Indonesia, Sleman

Kaliurang St KM. 14,5, Krawitan, Umbulmartani, Ngemplak, Kabupaten Sleman, Daerah Istimewa Yogyakarta, Indonesia

Email: ¹20523207@students.uii.ac.id, ^{2,*}teduh.dirgahayu@uui.ac.id

Email Penulis Korespondensi: teduh.dirgahayu@uui.ac.id

Submitted: 16/10/2024; Accepted: 26/10/2024; Published: 27/10/2024

Abstrak–Sistem informasi merupakan elemen penting dalam pengumpulan, penyimpanan, dan pengelolaan data untuk mendukung pengambilan keputusan yang efektif. Seiring perkembangan teknologi, serverless computing muncul sebagai solusi efisien yang menghilangkan kebutuhan pengelolaan infrastruktur server secara manual, memungkinkan pengembang untuk lebih fokus pada aplikasi inti. Studi ini mengeksplorasi bagaimana teknologi serverless dapat diterapkan untuk meningkatkan efisiensi administratif dalam pengelolaan data kependudukan desa, mengidentifikasi keuntungan dan tantangan yang muncul, dan dapat menjadi acuan bagi pemerintah desa dalam memanfaatkan teknologi serverless untuk transformasi digital dan optimalisasi pelayanan publik. Metodologi penelitian menggunakan pendekatan Agile dengan mempertimbangkan penggunaan teknologi serverless untuk memastikan proses pengembangan yang iteratif dan responsif terhadap kebutuhan pengguna. Hasil penelitian memperlihatkan bahwa waktu eksekusi fungsi cloud rata-rata di bawah 1 detik, dengan penggunaan memori yang stabil berkisar antara 19.1 MB hingga 38.1 MB. Dengan hasil pengujian tersebut penerapan serverless computing mampu meningkatkan efisiensi operasional dan kualitas layanan dalam pengelolaan data kependudukan. Penggunaan Firebase sebagai basis data mendukung sinkronisasi data secara real-time tanpa skema relasional. Dengan sistem ini, perangkat desa dapat mengelola data peristiwa kependudukan, sedangkan admin dapat mengelola data pengguna. Kepala desa dapat memantau dan mendapatkan laporan dari data yang dikelola. Selain itu, pengujian sistem menunjukkan bahwa penerapan keamanan dan kinerja yang baik dapat meningkatkan kualitas dari data kependudukan desa. Kesimpulan dari penelitian ini dapat memberikan panduan bagi pemerintah desa untuk memanfaatkan teknologi serverless dalam upaya meningkatkan layanan masyarakat, meskipun tantangan yang dihadapi termasuk kebutuhan untuk memahami dan mengimplementasikan aturan keamanan yang kompleks dan potensi kesulitan dalam mengelola sumber daya cloud yang terus berubah.

Kata Kunci: Serverless Computing; Firebase; Sistem Informasi; Data Kependudukan; Agile

Abstract–Information systems are vital for the collection, storage, and management of data to support decision-making. With technological advancements, serverless computing has emerged as an efficient solution, eliminating the need for manual server infrastructure management and allowing developers to focus on core applications. This study explores the application of serverless technology to enhance administrative efficiency in managing village population data, identifying its benefits and challenges. The research methodology employs an Agile approach, ensuring iterative development and responsiveness to user needs. The results show that serverless computing improves operational efficiency and service quality in population data management. Firebase enables real-time data synchronization without a relational schema. Village officials can manage population events, admins handle user data, and the village head monitors and generates reports. Additionally, system testing confirms that good security and performance enhance data quality. The study provides insights for village governments to adopt serverless technology to improve public services, despite challenges like complex security requirements and managing evolving cloud resources.

Kata Kunci: Serverless Computing; Firebase; Information System; Population Data; Agile

1. PENDAHULUAN

Sistem informasi merupakan elemen penting dalam pengumpulan, penyimpanan, dan pengelolaan data untuk mendukung pengambilan keputusan yang efektif. Seiring perkembangan teknologi, serverless computing muncul sebagai solusi efisien yang menghilangkan kebutuhan pengelolaan infrastruktur server secara manual, memungkinkan pengembang untuk lebih fokus pada aplikasi inti, dan membayar berdasarkan penggunaan aktual sumber daya. Serverless computing adalah inovasi dalam cloud computing yang memungkinkan pengembang menjalankan aplikasi tanpa harus mengelola infrastruktur server secara manual. Teknologi ini mencakup Function-as-a-Service (FaaS) dan Backend-as-a-Service (BaaS), yang menawarkan pengelolaan fungsi pemrograman serta layanan backend seperti autentikasi dan notifikasi. Dengan serverless, pengembang hanya membayar berdasarkan penggunaan fitur tanpa perlu membayar biaya tetap untuk infrastruktur yang tidak digunakan [1], [2]. Dua fitur utama dari teknologi ini adalah skalabilitas otomatis dan sistem pembayaran berdasarkan penggunaan, yang membuat serverless computing lebih efisien dan fleksibel dibandingkan dengan pendekatan tradisional yang mengharuskan pengelolaan server secara manual [1]. Penyedia teknologi serverless diawali oleh platform AWS Lambda [3], diikuti oleh Apache OpenWhisk [4], Azure Functions [5], Google Cloud Functions [6], dan Iron.io IronFunctions [7]. Hal tersebut menunjukkan bahwa penggunaan serverless dari tahun ke tahun meningkat sejak tahun 2015. Hal ini merupakan indikasi meningkatnya perhatian terhadap komputasi tanpa server di industri, pertemuan, blog, dan komunitas pengembangan [8].

Serverless computing telah menunjukkan potensi yang besar dalam berbagai skenario aplikasi. Salah satu contoh penerapan sukses adalah dalam pengembangan chatbot dan sistem pemesanan online. Handoyo [9] mengembangkan chatbot berbasis arsitektur serverless untuk meningkatkan efisiensi layanan pemesanan tiket. Teknologi ini memungkinkan modularisasi pemrosesan domain-specific, yang membantu meningkatkan performa dan menurunkan biaya. Selain itu, serverless computing juga telah digunakan dalam pengembangan aplikasi Internet of Things (IoT). Menurut Mulyani [10], penerapan AWS IoT dalam cold chain logistics memungkinkan pemantauan suhu secara real-time, yang meningkatkan efisiensi operasional secara signifikan. Untuk mengatasi keterbatasan pada kapasitas penyimpanan dan komputasi, yang dapat menyebabkan berbagai masalah seperti kehandalan, kinerja, skalabilitas, keamanan, dan privasi dalam (IoT) Bayu [11] menggunakan Cloud berbasis layanan komputasi serverless menggunakan Google Cloud Platform (GCP) untuk mengatasi hal tersebut.

Utomo [12] menyoroti tren penggunaan FaaS dalam bisnis cloud, terutama dalam pengembangan situs web dengan pendekatan JAMstack. Dengan memanfaatkan FaaS, pengembang dapat membangun aplikasi berbasis web secara efisien dengan otomatisasi layanan backend, seperti penyimpanan data dan otentikasi. Salah satu contoh yang menonjol adalah penerapan arsitektur serverless yang dilakukan oleh Oktaviani [13] pada aplikasi EMenu Restoran, yang terbukti meningkatkan skalabilitas secara global sambil mengurangi biaya operasional. Implementasi serverless pada berbagai aplikasi ini menunjukkan bahwa teknologi ini mampu mendukung pengembangan sistem yang lebih modular, adaptif, dan hemat biaya.

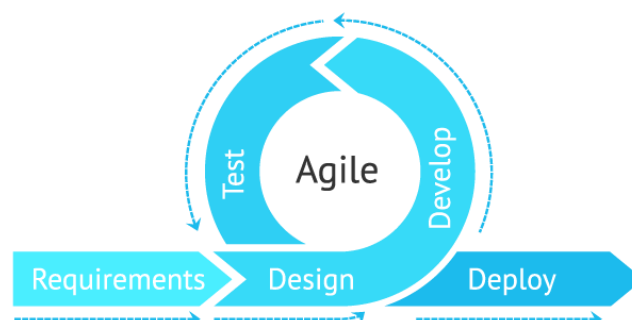
Dalam konteks pengelolaan data kependudukan, serverless computing menawarkan solusi yang lebih efisien bagi pemerintah desa. Teknologi ini memungkinkan pemerintah mengelola data penduduk tanpa harus berinvestasi dalam infrastruktur server yang rumit dan mahal. Penggunaan serverless memfasilitasi integrasi dengan layanan lain seperti basis data, penyimpanan objek, dan layanan otentikasi pengguna, yang sangat berguna dalam mengumpulkan, menyimpan, serta menganalisis data penduduk. Shafiei [14] mengemukakan bahwa penerapan serverless dalam pengelolaan data kependudukan dapat meningkatkan efisiensi administratif desa, mengurangi beban kerja petugas, serta meningkatkan kecepatan dan akurasi pengolahan data. Dalam studi Eisman [15], mereka mengidentifikasi beberapa alasan utama mengapa praktisi memilih serverless, termasuk pengurangan biaya operasional dan kemudahan dalam skalabilitas aplikasi. Nuppenon [16] juga menunjukkan bahwa penggunaan serverless membantu mencegah praktik buruk dalam pengembangan aplikasi, karena arsitekturnya yang mendukung modifikasi dan pembaruan secara lebih cepat tanpa gangguan pada layanan.

Penelitian ini bertujuan untuk mengeksplorasi bagaimana serverless computing dapat diterapkan dalam pengelolaan data kependudukan di tingkat desa. Selain itu, penelitian ini juga bertujuan mengidentifikasi keuntungan dan tantangan dari penggunaan serverless computing, serta memberikan panduan bagi pemerintah desa dalam meningkatkan efisiensi dan kualitas layanan masyarakat. Pemahaman lebih dalam tentang konsep serverless computing memungkinkan pemerintah desa memanfaatkan teknologi ini untuk mengoptimalkan proses administratif dan pelayanan kepada warga secara lebih efektif. Penelitian ini juga diharapkan memberikan kontribusi dalam literatur pengelolaan data kependudukan dengan pendekatan serverless computing, serta membantu pemerintah desa memahami bagaimana teknologi ini dapat meningkatkan efisiensi dan efektivitas sistem informasi yang mereka kelola. Secara keseluruhan, serverless computing berpotensi besar untuk mendukung transformasi digital di lingkungan pemerintahan, terutama dalam konteks pengelolaan data penduduk yang membutuhkan fleksibilitas, efisiensi biaya, dan skalabilitas yang tinggi.

2. METODOLOGI PENELITIAN

2.1 Metode Agile

Metode penelitian ini menggunakan pendekatan Agile yang diintegrasikan dengan serverless computing. Pada penelitian ini, penulis memilih menggunakan metode ini karena dianggap sebagai konsep penting dalam membangun perangkat lunak yang dapat disesuaikan dengan kebutuhan pengguna [17]. Metode Agile berfungsi sebagai pedoman untuk pengembangan sistem dari tahap awal sampai akhir, dengan tahapan proses seperti yang terlihat pada Gambar 1.



Gambar 1. Tahapan Pengembangan Metode Agile



Dalam pengembangan sistem dengan metode Agile, proses dilakukan secara iteratif dan berkelanjutan. Identifikasi kebutuhan (requirements) menjadi tahap awal untuk memahami input, output, antarmuka pengguna, dan alur kerja sistem, serta mengidentifikasi fitur serverless seperti Function-as-a-Service (FaaS) dan Backend-as-a-Service (BaaS). Selanjutnya, perancangan sistem (design) mencakup pengembangan fitur utama, seperti manajemen penduduk dan pencatatan peristiwa kependudukan, serta pemilihan layanan cloud untuk integrasi komponen. Pada tahap pengembangan (develop), kode program diimplementasikan berdasarkan rancangan, dengan pemanfaatan teknologi serverless yang mengotomatisasi integrasi tanpa pengelolaan server manual. Pengujian sistem (test) dilakukan untuk memastikan keamanan, dan kinerja sesuai spesifikasi. Perilisan (deploy) menjadi langkah akhir setelah evaluasi menyeluruh, memastikan sistem siap diluncurkan dengan kualitas dan kinerja optimal.

2.2 Identifikasi Kebutuhan

Tahap ini melibatkan analisis kebutuhan untuk memahami proses pengelolaan data kependudukan, meliputi input, output, antarmuka pengguna, serta alur kerja sistem. Pada metode Agile, proses identifikasi kebutuhan dilakukan secara iteratif dengan melibatkan pengguna untuk memberikan umpan balik di awal dan sepanjang pengembangan. Selain itu, pada tahap ini dilakukan identifikasi fitur serverless yang relevan, seperti Function-as-a-Service (FaaS) dan Backend-as-a-Service (BaaS) untuk pemrosesan data, autentikasi, dan penyimpanan.

2.3 Perancangan Sistem

Dimulai dengan perancangan fitur utama seperti manajemen penduduk, pencatatan peristiwa kependudukan (kelahiran, kematian, pindah, datang, menikah, bercerai), dan pembuatan laporan. Dalam konteks serverless, desain arsitektur sistem mencakup pemilihan layanan cloud yang akan digunakan, seperti AWS Lambda atau Google Cloud Functions, serta pengaturan integrasi antar komponen yang berbeda.

2.4 Pengembangan Sistem

Pada tahap ini, dilakukan implementasi sistem dengan menuliskan kode program berdasarkan rancangan yang telah disusun pada tahap perancangan sistem sebelumnya [18]. Proses ini melibatkan implementasi desain yang telah dibuat, di mana setiap fungsi yang diimplementasikan menggunakan teknologi serverless. Proses ini mencakup pengembangan berbagai komponen sistem, pengujian unit untuk memastikan fungsionalitas yang benar, serta pemanfaatan layanan serverless untuk pengelolaan data dan event-triggered functions. Penggunaan serverless computing memungkinkan pengembang untuk mengotomatisasi integrasi antar modul tanpa perlu mengelola infrastruktur server secara manual.

2.5 Pengujian Sistem

Pada tahap ini, dilakukan pengujian untuk mengevaluasi apakah sistem beroperasi sesuai dengan harapan yang telah ditetapkan sebelumnya [19]. Pengujian sistem bertujuan memastikan bahwa semua fungsi bekerja sesuai spesifikasi dan memenuhi kebutuhan pengguna melalui pengujian keamanan dan kinerja. Proses ini melibatkan pengumpulan umpan balik dari pengguna untuk perbaikan lebih lanjut sebelum sistem diluncurkan secara penuh. Pengujian layanan seperti Function-as-a-Service (FaaS) dan Backend-as-a-Service (BaaS) juga menjadi bagian dari proses ini untuk memastikan bahwa sistem dapat merespons beban kerja yang bervariasi dengan optimal.

2.6 Perilisan

Pada tahap ini, sistem yang telah melewati proses pengujian sebelumnya akan disiapkan untuk dirilis. Proses ini melibatkan pengevaluasian menyeluruh terhadap semua fitur dan fungsi sistem untuk memastikan bahwa kualitas dan kinerja sistem telah mencapai standar yang diharapkan [20].

3. HASIL DAN PEMBAHASAN

3.1 Identifikasi Kebutuhan

Terdapat tiga aktor utama pada use case sistem informasi data kependudukan Desa Gambasan, yaitu Perangkat Desa, admin, dan kepala desa. Ketiga aktor tersebut dapat berinteraksi dengan sistem namun harus login terlebih dahulu. Gambar 2 menunjukkan diagram use case yang menggambarkan interaksi antara tiga aktor utama pada sistem informasi data kependudukan Desa Gambasan, yaitu Perangkat Desa, Admin, dan Kepala Desa.



Gambar 2. Use Case Diagram

Tabel 1 merincikan berbagai Use Case yang dapat dilakukan oleh masing masing pengguna dalam sistem. Mulai dari Kepala Desa dan Perangkat Desa yang melihat dashboard hingga mengelola data penduduk yang menikah, bercerai, meninggal, pindah, datang, dan lahir, serta Admin yang dapat mengelola data pengguna dan penduduk tetap. Hal ini diharapkan memberikan gambaran menyeluruh terkait penggunaan aplikasi Kependudukan Desa sesuai dengan peran pengguna yang bersangkutan.

Tabel 1. Use Case Diagram Aktor

Aktor	Nama Use Case	Keterangan
Perangkat Desa	Melihat Dashboard	Perangkat desa dapat melihat ringkasan dan visualisasi data dalam bentuk dashboard.
	Mengelola Data Penduduk Menikah	Perangkat desa dapat mengelola data penduduk yang sudah menikah.
	Mengelola Data Penduduk Bercerai	Perangkat desa dapat mengelola data penduduk yang sudah bercerai.
	Mengelola Data Penduduk Meninggal	Perangkat desa dapat mengelola data penduduk yang telah meninggal.
	Mengelola Data Penduduk Pindah	Perangkat desa dapat mengelola data penduduk yang pindah ke lokasi lain.
	Mengelola Data Penduduk Datang	Perangkat desa dapat mengelola data penduduk yang datang ke lokasi tersebut.
	Mengelola Data Penduduk Lahir	Perangkat desa dapat mengelola data penduduk yang baru lahir.
	Mengelola Data Penduduk Tetap	Perangkat desa dapat mengelola data penduduk tetap.
Admin	Mengelola Data Pengguna	Admin dapat membuat, menampilkan, dan menghapus akun, termasuk informasi alamat email, dan hak akses dalam sistem.

Aktor	Nama Use Case	Keterangan
Kepala Desa	Melihat Dashboard	Kepala desa dapat melihat ringkasan dan visualisasi data dalam bentuk dashboard.

3.2 Perancangan Sistem

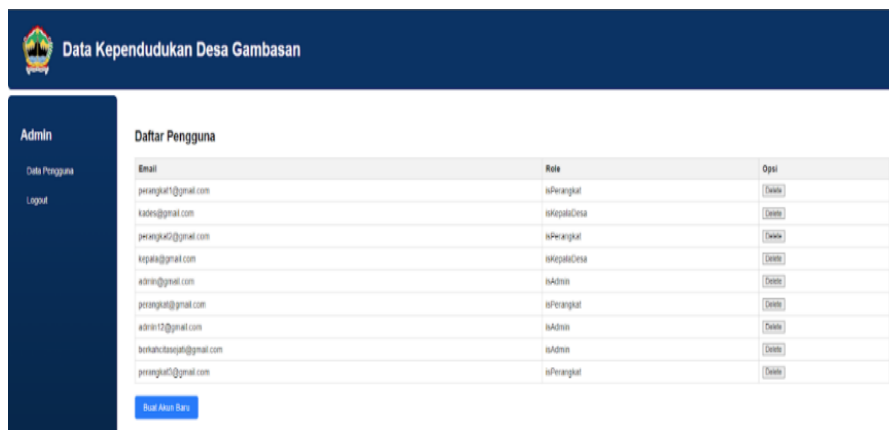
Pengelolaan data kependudukan desa mencakup tiga peran utama yang tersimpan pada role, yaitu: admin, perangkat desa, dan kepala desa. Masing-masing peran memiliki hak akses berbeda. Admin memiliki akses penuh untuk mengelola data pengguna dan penduduk, termasuk membuat, menampilkan, dan mengubah informasi. Perangkat desa berfokus pada pengelolaan data peristiwa kependudukan seperti pernikahan, perceraian, kematian, perpindahan, kedatangan, dan kelahiran. Sistem ini menggunakan Firebase sebagai basis data, yang memungkinkan sinkronisasi data secara realtime tanpa skema relasional karena menggunakan konsep NoSQL dengan model tree berbasis key value.

3.3 Pengembangan Sistem

Gambar 4 hingga 6 akan menjadi desain rancangan antarmuka halaman dashboard pada use case tabel 1 hingga tabel 3. Desain antarmuka akan dirancang dengan mempertimbangkan kebutuhan pengguna pada setiap use case, memprioritaskan kejelasan informasi dan kemudahan penggunaan untuk meningkatkan efisiensi dalam operasional sehari-hari.

3.3.1 Dashboard Admin

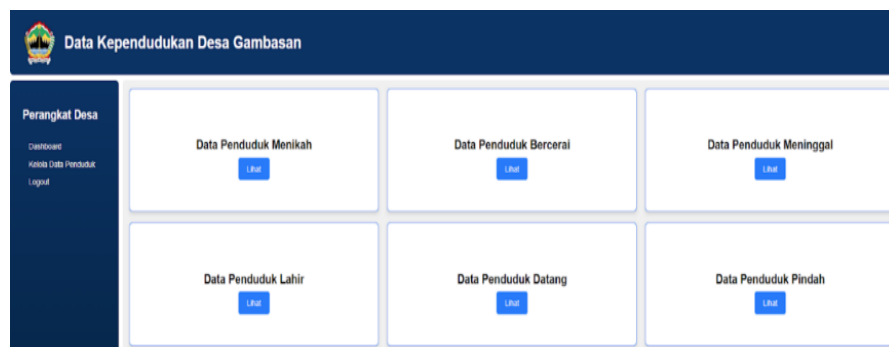
Dalam desain antarmuka, fokus utama adalah memastikan bahwa admin dapat dengan mudah melakukan pembuatan, penampilan, dan penggantian data pengguna. Fitur-fitur ini mencakup pengisian informasi seperti nama, alamat email, dan hak akses dalam sistem. Proses ini melibatkan pembuatan akun baru, pemantauan data pengguna yang terdaftar. Dengan antarmuka pada Gambar 4 yang dirancang dengan baik, diharapkan admin dapat menjalankan tugasnya dengan efisien dan tanpa kesulitan.



Gambar 3. Halaman Dashboard Admin

3.3.2 Dashboard Perangkat Desa

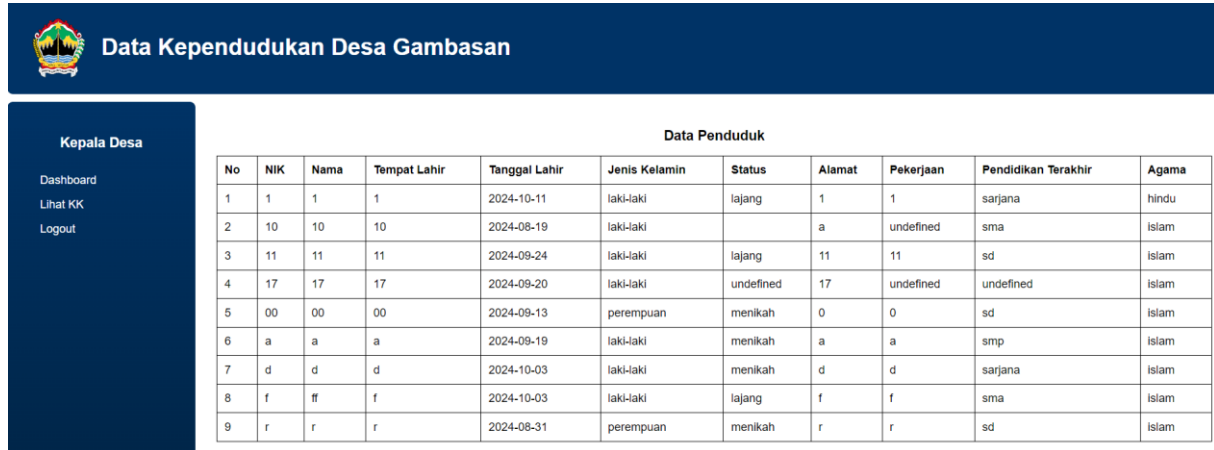
Gambar 5 menunjukkan antarmuka dashboard perangkat desa disusun untuk memberikan akses kepada perangkat desa untuk melihat ringkasan dan visualisasi data dalam bentuk dashboard. Dengan fitur ini, perangkat desa dapat dengan cepat memahami kondisi demografis dan statistik penduduk desa. Proses ini bertujuan untuk mempermudah perangkat desa dalam memantau dan mengelola data penduduk dengan lebih efisien, memungkinkan mereka untuk memberikan pelayanan yang lebih baik kepada masyarakat desa.



Gambar 4. Halaman Dashboard Perangkat Desa.

3.3.3 Dashboard Kepala Desa

Pada gambar 5 dirancang untuk memberikan gambaran umum tentang data kependudukan desa. Melalui tampilan ringkasan dan visualisasi data dalam bentuk dashboard, Kepala Desa dapat dengan mudah memantau perkembangan dan kondisi penduduk desa secara menyeluruh. Proses ini bertujuan untuk memberikan informasi yang relevan dan memudahkan Kepala Desa dalam mengambil keputusan terkait kebijakan atau tindakan yang diperlukan.



No	NIK	Nama	Tempat Lahir	Tanggal Lahir	Jenis Kelamin	Status	Alamat	Pekerjaan	Pendidikan Terakhir	Agama
1	1	1	1	2024-10-11	laki-laki	lajang	1	1	sarjana	hindu
2	10	10	10	2024-08-19	laki-laki		a	undefined	sma	islam
3	11	11	11	2024-09-24	laki-laki	lajang	11	11	sd	islam
4	17	17	17	2024-09-20	laki-laki	undefined	17	undefined	undefined	islam
5	00	00	00	2024-09-13	perempuan	menikah	0	0	sd	islam
6	a	a	a	2024-09-19	laki-laki	menikah	a	a	smp	islam
7	d	d	d	2024-10-03	laki-laki	menikah	d	d	sarjana	islam
8	f	ff	f	2024-10-03	laki-laki	lajang	f	f	sma	islam
9	r	r	r	2024-08-31	perempuan	menikah	r	r	sd	islam

Gambar 5. Halaman Dashboard Kepala Desa.

Pengembangan Implementasi ke dalam Firebase Realtime Database dan Firebase Authentication, data kependudukan akan terstruktur dalam bentuk koleksi yang mencerminkan peran dan tanggung jawab masing-masing pengguna. Berikut adalah penjelasan relasi antar data yang ada dalam sistem:

3.3.4 Admin

Dengan integrasi Firebase Authentication dan Realtime Database(BaaS), admin dapat membuat atau menghapus data pengguna menggunakan fitur Cloud Functions (FaaS) yang secara otomatis memvalidasi dan mengirimkan notifikasi email setiap kali ada perubahan data pengguna.

3.3.5 Perangkat Desa

Perangkat desa bertanggung jawab untuk mengelola data peristiwa kependudukan, seperti Penduduk, Menikah, Bercerai, Lahir, Meninggal, Pindah, dan Datang. Dengan menggunakan Firebase Realtime Database, perangkat desa dapat memastikan bahwa seluruh perubahan status dan informasi penduduk tercatat secara real-time.

3.3.6 Kepala Desa

Kepala Desa memiliki akses untuk meninjau dan mendapatkan laporan dari data yang dikelola oleh admin dan perangkat desa, tetapi tidak memiliki hak untuk mengubah data. Dalam Firebase Realtime Database semua data ini disimpan dalam struktur hierarkis yang tidak terikat oleh skema relasional. Masing-masing koleksi (seperti users, Penduduk, Menikah, Lahir) dapat diakses secara langsung melalui ID unik, dan relasi antar koleksi dapat dilakukan dengan menggunakan NIK sebagai kunci penghubung.

Tabel 2. Implementasi Database menjadi NoSql Database

Koleksi	Dokumen
Bercerai	"Bercerai": { "uniqueID": { "namaIstri": "", "namaSuami": "", "nikIstri": "", "nikSuami": "", "noNikah": "", "tanggalCerai": "" } },
Datang	"Datang": { "{NIK}": { "alamatAsal": "", "alasanDatang": "",



Koleksi	Dokumen
Kartu Keluarga	<pre>"nama": "", "nik": "", "tanggalDatang": "" }, "KK":{ "alamat": "", "nik": ["{nik1}", "{nik2}", "{nik3}", "{nik4}", "{nik5}", "{nik6}", "{nik7}", "{nik8}"], "noKK": "", "tanggalBuat": "" },</pre>
Lahir	<pre>"Lahir": { "{NIK}": { "alamat": "", "jenisKelamin": "", "namaAnak": "", "namaOrangTua": "", "nik": "", "nomorKK": "", "tanggalLahir": "", "tempatLahir": "" } },</pre>
Menikah	<pre>"Menikah": { "{uniqueID}": { "alamat": "", "namaIstri": "", "namaSuami": "", "nikIstri": "", "nikSuami": "", "noKK": "", "noNikah": "", "tanggalMenikah": "" } },</pre>
Meninggal	<pre>"Meninggal": { "{NIK}": { "nama": "", "nik": "", "nomorKematian": "", "tanggalMeninggal": "" } },</pre>
Penduduk	<pre>"Penduduk": { "{NIK}": { "agama": "", "alamat": "", "jenisKelamin": "", "nama": "", "nik": "", "pekerjaan": "",</pre>

Koleksi	Dokumen
Pindah	<pre> "pendidikan": "", "status": "", "tempat": "", "ttl": "" } }, "Pindah": { "{NIK}": { "alamatPindah": "", "alasanPindah": "", "nik": "", "tanggalPindah": "" } } </pre>
Users	<pre> }, "users": { "{userID}": { "email": "", "isAdmin": false, "isKepalaDesa": false, "isPerangkat": false } } </pre>

3.4 Pengujian Sistem

Pengujian sistem ini bertujuan untuk memastikan keamanan dan kinerja aplikasi berbasis Firebase Realtime Database dan Cloud Functions. Tujuan utama dari pengujian keamanan adalah untuk mengevaluasi kontrol akses terhadap data, memastikan bahwa hanya pengguna yang berhak yang dapat mengakses dan memodifikasi informasi sensitif. Pengujian menggunakan Firebase Security Rules Playground, dengan beberapa skenario untuk menguji kontrol akses, mulai dari aturan yang sepenuhnya membatasi akses hingga aturan yang hanya mengizinkan pengguna terotentikasi. Pengujian kinerja bertujuan untuk memantau waktu eksekusi dan penggunaan memori fungsi cloud, seperti pengiriman email dan penghapusan akun, untuk memastikan bahwa sistem dapat menangani beban kerja secara efisien.

3.4.1 Keamanan

Pengujian sistem keamanan dilakukan dengan Firebase Security Rules Playground. Rules pada Firebase Realtime Database adalah lapisan keamanan yang dirancang untuk mengatur dan mengelola siapa yang dapat membaca atau menulis data pada database. Firebase Security Rules menggunakan sintaksis JSON yang memungkinkan untuk menetapkan kontrol akses berbasis identitas pengguna (auth), struktur data, dan logika kondisi tertentu untuk setiap node dalam database.

a. Skenario Pengujian 1

Ketika mencoba mengakses data di Firebase (baik membaca maupun menulis) menggunakan akun terotentikasi maupun anonim, permintaan ditolak dengan pesan kesalahan PERMISSION_DENIED. Semua permintaan data dari aplikasi menghasilkan error, dan tidak ada data yang dapat ditampilkan di antarmuka pengguna. Tidak ada perubahan data yang bisa dilakukan, memastikan bahwa keamanan data sepenuhnya terjaga. Hal ini menunjukkan bahwa aturan False sudah berfungsi dengan benar, memblokir akses apapun ke database.

```

{
  "rules": {
    ".read": false,
    ".write": false
  }
}

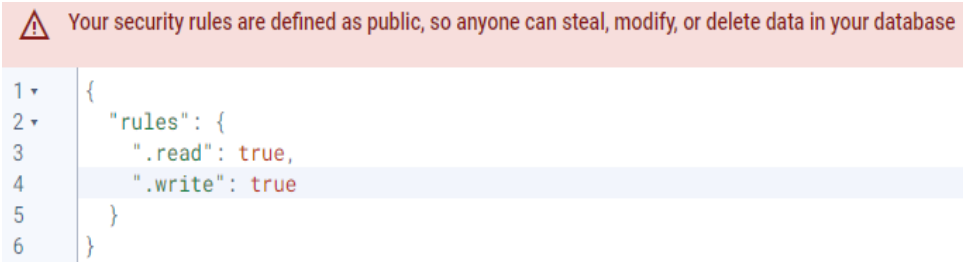
```

Gambar 6. Rules False

b. Skenario Pengujian 2

Semua pengguna, baik yang anonim, ter-autentikasi, maupun tidak terdaftar di sistem, dapat membaca dan menulis data ke database tanpa batasan. Data dapat dibaca, ditampilkan, dan dimodifikasi secara bebas di antarmuka pengguna. Semua permintaan data, termasuk permintaan yang biasanya diblokir (misalnya, perubahan data pengguna lain), akan berhasil dilakukan. Pengaturan ini menimbulkan resiko besar bagi keamanan data, karena tidak ada batasan akses yang diterapkan. Siapa pun yang mengetahui URL database dapat melihat, mengubah, dan menghapus data. Data dapat diubah atau dihapus oleh pengguna yang tidak

berhak, sehingga integritas data menjadi tidak terjamin. Pengaturan ini berbahaya jika diterapkan dalam lingkungan produksi, terutama untuk aplikasi yang menyimpan data sensitif atau pribadi.



Gambar 7. Rules True

c. Skenario Pengujian 3

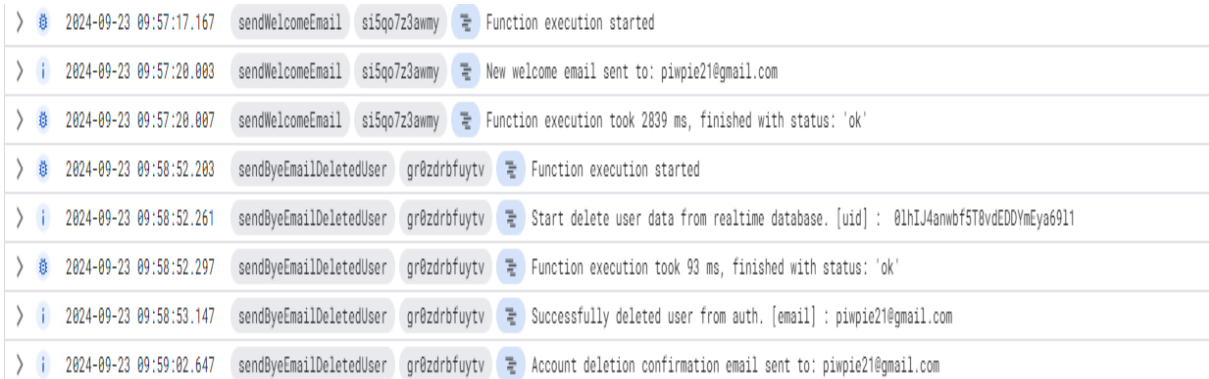
Aturan ini mengizinkan semua pengguna yang terautentikasi untuk membaca dan menulis data pada database. Kondisi `auth != null` berarti selama pengguna sudah login, mereka dapat mengakses dan memodifikasi data dengan ID pengguna (`uid`). Skenario ini adalah skenario yang paling aman karena data dapat dikendalikan oleh pengguna yang berhak mengakses data.



Gambar 8. Rules Auth

3.4.2 Kinerja

Pengujian kinerja dilakukan dengan menggunakan fitur Cloud Monitoring (logs), Cloud Function - Execution Time (Sum), dan Cloud Function - Memory Usage (Sum). Pada pengujian pertama kinerja dilakukan pada logs dari fungsi `sendWelcomeEmail` dan `sendByeEmailDeletedUser`.



Gambar 9. Logs

a. Pengiriman Welcome Email

Pada 2024-09-23 pukul 09:57, fungsi `sendWelcomeEmail` dijalankan untuk pengguna dengan UID `si5qo7z3awmy`. Email berhasil dikirim ke alamat `piwpie21@gmail.com` dalam waktu sekitar 2839 ms, dengan status 'ok'. Ini menunjukkan bahwa sistem mampu memproses pembuatan akun baru dan mengirim email selamat datang tanpa masalah. Waktu eksekusi yang relatif wajar menunjukkan performa yang baik dalam pengiriman email.

b. Pengiriman Bye Email

Pada pukul 09:58:52 hingga 09:59:02, fungsi `sendByeEmailDeletedUser` dijalankan untuk pengguna dengan UID `gr8zdrbfuytv`. Proses ini berhasil menghapus data pengguna dari basis data Realtime dan dari Firebase Authentication, serta mengirim email konfirmasi penghapusan ke `piwpie21@gmail.com`. Semua operasi terkait penghapusan pengguna ini selesai dengan status 'ok', dan masing-masing tahap dijalankan dengan sangat cepat (sekitar 93 ms untuk penghapusan dari database). Ini menunjukkan bahwa sistem dapat menangani proses penghapusan akun dengan efisien. Kecepatan penghapusan dan pengiriman email konfirmasi menunjukkan bahwa sistem dapat menangani operasi penting dengan responsivitas tinggi.

**Gambar 10.** Logs

c. Execution Times (Waktu Eksekusi)

Grafik di sisi kiri menunjukkan waktu eksekusi fungsi cloud dalam satuan waktu (milidetik hingga detik). Mayoritas eksekusi fungsi berlangsung cukup cepat, dengan sebagian besar eksekusi berada di bawah 1 detik. Ada fluktuasi waktu eksekusi yang tampaknya terjadi di bagian akhir, yang dapat menunjukkan beban tambahan pada sistem atau beberapa fungsi yang memerlukan waktu lebih lama untuk diselesaikan.

d. Memory Usage (Penggunaan Memori)

Grafik di sisi kanan menunjukkan penggunaan memori dari fungsi cloud yang dieksekusi. Penggunaan memori berkisar antara 19.1 MB hingga 38.1 MB. Ini menunjukkan bahwa fungsi yang dieksekusi memiliki footprint memori yang relatif kecil dan stabil, tanpa lonjakan penggunaan memori yang signifikan.

4. KESIMPULAN

Kesimpulan dari penelitian ini adalah bahwa penerapan serverless computing dalam pengelolaan data kependudukan di tingkat desa dapat memberikan efisiensi yang tinggi serta meningkatkan kualitas layanan yang diberikan. Keuntungan dari penggunaan serverless computing mencakup peningkatan keamanan data, kemudahan dalam melakukan penskalaan dan pemeliharaan, serta peningkatan responsivitas aplikasi dalam menghadapi permintaan pengguna. Namun, ada tantangan yang harus dihadapi, seperti kebutuhan akan pemahaman yang mendalam mengenai implementasi aturan keamanan yang kompleks serta kesulitan dalam mengelola sumber daya cloud yang bersifat dinamis dan terus berubah. Pengujian sistem keamanan yang dilakukan menggunakan Firebase Security Rules berhasil menunjukkan berbagai skenario akses data, mulai dari penolakan akses bagi semua pengguna hingga pemberian akses terbatas bagi pengguna yang terautentikasi, yang menggarisbawahi pentingnya konfigurasi yang tepat untuk menjaga integritas data. Selain itu, pengujian kinerja melalui fitur Cloud Monitoring menunjukkan bahwa sistem mampu menangani proses pengiriman email selamat datang dan penghapusan akun dengan cepat dan efisien, dengan waktu eksekusi yang sesuai dan penggunaan memori yang stabil. Hal ini menegaskan bahwa serverless computing tidak hanya mampu meningkatkan responsivitas aplikasi, tetapi juga memastikan keamanan data melalui pengaturan akses yang ketat. Penelitian ini memberikan panduan yang berguna bagi pemerintah desa untuk memanfaatkan teknologi serverless dalam meningkatkan efisiensi operasional serta memperbaiki layanan kepada masyarakat.

REFERENCES

- [1] P. Castro, V. Ishakian, V. Muthusamy, and A. Slominski, "The rise of serverless computing," *Commun ACM*, vol. 62, no. 12, pp. 44–54, Dec. 2019, doi: 10.1145/3368454.
- [2] S. Haque, Z. Eberhart, A. Bansal, and C. McMillan, "Semantic Similarity Metrics for Evaluating Source Code Summarization," in *IEEE International Conference on Program Comprehension*, IEEE Computer Society, 2022, pp. 36–47. doi:10.48550/arXiv.2204.01632.
- [3] Administrator, "AWS Lambda Run code without thinking about servers or clusters," Accessed: 2024. [Online]. Available: <https://aws.amazon.com/lambda/>.
- [4] Administrator, "Open Source Serverless Cloud Platform," Accessed: 2024. [Online]. Available: <https://openwhisk.apache.org/>.
- [5] Administrator, "Azure serverless Go serverless—build apps faster without managing infrastructure," Accessed: 2024. [Online]. Available: <https://azure.microsoft.com/en-us/solutions/serverless>.
- [6] Rajeshwari B Mathapati, "Building Serverless Applications with GCP's Cloud Functions," Accessed: 2023. [Online]. Available: <https://www.cloudthat.com/resources/blog/building-serverless-applications-with-gcps-cloud-functions#steps-to-build-a-serverless-application-with-cloud-functions>.
- [7] Administrator, "Open Source Serverless Computing," Accessed: 2024. [Online]. Available: <https://open.iron.io/>.
- [8] Saputra D, "Kelola Kode Serverless dengan Serverless Framework," Accessed: 2023. [Online]. Available: <https://www.dicoding.com/blog/kelola-kode-serverless-dengan-serverless-framework/>.
- [9] Handoyo, E., Arfan, M., Soetrisno, Y. A. A., Somantri, M., Sofwan, A., & Sinuraya, E. W., "Ticketing chatbot service using serverless NLP technology". 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE) (pp. 325-330). IEEE. 2018, DOI:10.1109/ICITACEE.2018.8576921
- [10] A. Mulyani and U. Y. Oktiawati, "Implementasi Arsitektur Serverless Internet of Things pada Monitoring Cold Chain," *Journal of Internet and Software Engineering*, vol. 3, no. 1, 2022, Accessed: Oct. 16, 2024. [Online]. Available:



<https://download.garuda.kemdikbud.go.id/article.php?article=3249530&val=28494&title=Implementation%20of%20Serverless%20Internet%20of%20Things%20Architecture%20in%20Cold%20Chain%20Monitoring>

- [11] R. Bayu, A. Pradana, and A. Bhawiyuga, “Pengembangan Platform IoT Cloud berbasis Layanan Komputasi Serverless Google Cloud Platform (GCP),” 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [12] P. Utomo and Falahah, “Building Serverless Website on GitHub Pages,” in IOP Conference Series: Materials Science and Engineering, IOP Publishing Ltd, Aug. 2020. doi: 10.1088/1757-899X/879/1/012077.
- [13] D. Oktaviani, F. S. Papilaya, and F. Tanaem, “Perancangan Aplikasi E-Menu Restaurant dengan Menggunakan Cloud Computing dan Serverless Architecture Lambda,” vol. 12, 2021, Accessed: Oct. 16, 2024. [Online]. Available: <https://pdfs.semanticscholar.org/f159/a725166307a4a2cf2572406f7c22ad31cb92.pdf>
- [14] H. Shafiei, A. Khonsari, and P. Mousavi, “Serverless Computing: A Survey of Opportunities, Challenges and Applications,” ACM Computing Surveys, Volume 54, Issue 11s, No.: 239, Pages 1 - 32 Nov. 2019, doi:10.1145/3510611.
- [15] S. Eismann et al., “Serverless Applications: Why, When, and How?,” IEEE Softw, vol. 38, no. 1, pp. 32–39, Jan. 2021, doi: 10.1109/MS.2020.3023302.
- [16] J. Nupponen and D. Taibi, “Serverless: What it Is, What to Do and What Not to Do,” in Proceedings - 2020 IEEE International Conference on Software Architecture Companion, ICSA-C 2020, Institute of Electrical and Electronics Engineers Inc., Mar. 2020, pp. 49–50. doi: 10.1109/ICSA-C50368.2020.00016.
- [17] T. Ayunita Pertiwi, “Perancangan Dan Implementasi Sistem Informasi Absensi Berbasis Web Menggunakan Metode Agile Software Development”, JTISI, vol. 1, no. 1, pp. 53-66, Mar. 2023.
- [18] F. Reza, I. K. Dewi, and M. Ropianto, “Perancangan Dan Implementasi Institutional Repository Dengan Metadata Dublin Core,” Jurnal KomtekInfo, pp. 125–132, Dec. 2022, doi: 10.35134/komtekinfo.v9i4.318.
- [19] I. P. Dewi and R. Fikri, “Optimalisasi Keamanan Rumah dengan Implementasi Sistem Notifikasi Gerbang Cerdas Berbasis Internet of Things (IoT),” Journal of Computer System and Informatics (JoSYC), vol. 4, no. 4, pp. 816–829, Aug. 2023, doi: 10.47065/josyc.v4i4.4004.
- [20] S. B. Atim, “Permodelan Sistem Informasi Penjualan Barang Berbasis Website Menggunakan Metode Agile,” Journal of Artificial Intelligence and Technology Information (JAITI), vol. 2, no. 1, 2024, doi: 10.58602/jaiti.v2i1.104.