

**PENGEMBANGAN DAN IMPLEMENTASI TEKNOLOGI  
SERVERLESS COMPUTING PADA SISTEM INFORMASI  
DATA KEPENDUDUKAN DESA**



Disusun Oleh:

N a m a : Muhammad Dimas Haryo S

NIM : 20523207

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2024**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN DAN IMPLEMENTASI TEKNOLOGI  
SERVERLESS COMPUTING PADA SISTEM INFORMASI  
DATA KEPENDUDUKAN DESA**

**TUGAS AKHIR**

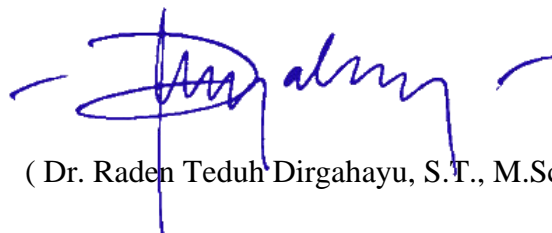


Nama : Muhammad Dimas Haryo Saputra  
NIM : 20523207

الجمهورية الإسلامية الإندونيسية

Yogyakarta, 28 Oktober 2024

Pembimbing,



( Dr. Raden Teduh Dirgahayu, S.T., M.Sc)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN DAN IMPLEMENTASI TEKNOLOGI  
SERVERLESS COMPUTING PADA SISTEM INFORMASI  
DATA KEPENDUDUKAN DESA**

**TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 11 November 2024

Tim Penguji

Dr. Raden Teduh Dirgahayu, S.T., M.Sc.

**Anggota 1**

Sheila Nurul Huda, S.Kom., M.Cs.

**Anggota 2**

Sri Mulyati, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Muhammad Dimas Haryo Saputra

NIM : 20523207

Tugas akhir dengan judul:

### **PENGEMBANGAN DAN IMPLEMENTASI TEKNOLOGI SERVERLESS COMPUTING PADA SISTEM INFORMASI DATA KEPENDUDUKAN DESA**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.



Yogyakarta, 28 Oktober 2024

( Muhammad Dimas Haryo S )

## HALAMAN PERSEMBAHAN

*Al-hamdu lillahi rabbil 'alamin*, Dengan penuh rasa syukur kepada Allah SWT atas rahmat dan petunjuk-Nya, laporan tugas akhir ini saya dedikasikan kepada kedua orang tua dan keluarga yang selalu memberikan doa serta dukungan yang selalu menyertai setiap langkah saya. Ucapan terima kasih kepada seluruh rekan-rekan yang telah memberikan semangat dan bantuan, serta semua pihak yang berkontribusi dalam penyusunan laporan tugas akhir ini. Semoga penelitian ini dapat bermanfaat.

## HALAMAN MOTO

*“Man Jadda Wa Jadda”*

“Dan jika kamu menghitung nikmat Allah, niscaya kamu tidak akan mampu menghitungnya.

Sesungguhnya Allah benar-benar Maha Pengampun, lagi Maha Penyayang.”

- QS An Nahl 18

*“The best investment you can make, is an investment on yourself. The more you learn, the more you’ll earn”*

- Warren Baffett

## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji syukur kehadiran Allah SWT yang telah memberikan rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Pengembangan dan Implementasi Teknologi Serverless Computing pada Sistem Informasi Data Kependudukan Desa” ini dengan baik. Tanpa bimbingan dan pertolongan-Nya, tugas akhir ini tidak akan dapat diselesaikan dengan lancar. Shalawat serta salam tetap tercurahkan kepada junjungan kita, Nabi Muhammad SAW, yang menjadi teladan bagi umat manusia dalam mencari ilmu dan kebenaran.

Penulis menyadari sepenuhnya bahwa banyak pihak yang telah memberikan bantuan dan dukungan, baik secara langsung maupun tidak langsung, sehingga penelitian ini dapat terlaksana dan terselesaikan dengan baik. Oleh karena itu, penulis mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Allah SWT yang telah memberikan petunjuk dan kemudahan sehingga penulis dapat menyelesaikan penelitian ini dengan baik.
2. Bapak Sunarya dan Ibu Hartatik selaku orang tua penulis, dan Keluarga Besar yang selalu memberikan dukungan, semangat, dan doa agar penulis dapat menyelesaikan tugas akhir ini.
3. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
4. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Prodi Informatika Program Sarjana Universitas Islam Indonesia.
5. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Dosen Pembimbing Penelitian sekaligus Dosen Dosen Pembimbing Akademik yang telah bersedia meluangkan waktu, tenaga, dan pemikirannya.
6. Bapak dan Ibu dosen Program Sarjana Studi Informatika, yang telah memberikan ilmu yang bermanfaat.
7. Mutiara Ananda Nuriantonno yang menjadi *support system* dan menjadi bagian penting untuk penulis selama masa perkuliahan.
8. Seluruh rekan penulis yang telah menemani dan membantu selama masa perkuliahan.
9. Seluruh pihak yang terlibat dalam penelitian yang tidak bisa disebutkan satu per satu, terima kasih atas semua dukungan dan doa-doa baik yang telah di berikan.

Semoga ilmu, bimbingan, doa yang diberikan memperoleh balasan yang lebih dari Allah SWT. Penulis menyadari sepenuhnya bahwa masih banyak kekurangan dalam pembuatan tugas akhir ini. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun. Penulis berharap penelitian ini juga dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 28 Oktober 2024

A handwritten signature in black ink, consisting of stylized, overlapping letters and lines, representing the name Muhammad Dimas Haryo S.

( Muhammad Dimas Haryo S )

## SARI

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 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, Data Kependudukan, Sistem Informasi.

## GLOSARIUM

Cloud Computing	Model penyediaan layanan teknologi informasi yang memungkinkan pengguna untuk mengakses sumber daya komputasi melalui internet tanpa harus mengelola infrastruktur fisik.
Footprint	Jejak data unik yang dibuat oleh seseorang atau bisnis saat menggunakan internet, disebut juga jejak digital atau bayangan digital.
GitHub Pages	Layanan hosting dari GitHub yang memungkinkan pengembang untuk menyimpan dan mempublikasikan situs web statis secara gratis.
IoT (Internet of Things)	Jaringan perangkat fisik yang saling terhubung melalui internet untuk bertukar data dan menjalankan tindakan tertentu.
NoSQL	Jenis basis data yang tidak menggunakan struktur tabel relasional seperti SQL, biasanya lebih fleksibel dan digunakan untuk aplikasi yang memerlukan skalabilitas tinggi.
Otentikasi	Proses untuk memastikan bahwa pengguna atau sistem adalah entitas yang sah sebelum diberi akses ke sistem atau data.
Skalabilitas	Kemampuan suatu sistem atau aplikasi untuk menangani peningkatan jumlah data atau pengguna tanpa mengorbankan kinerja.
Skema Basis Data	Rancangan atau struktur dari basis data yang menunjukkan bagaimana data diatur, dikelompokkan, dan dihubungkan satu sama lain.
UniqueID	Identifikasi unik yang digunakan untuk mengakses dokumen tertentu dalam Firebase Realtime Database, memungkinkan pengorganisasian data secara terstruktur.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING .....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI .....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR.....	vii
SARI.....	ix
GLOSARIUM .....	x
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR.....	xiv
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
1.6 Metode Penelitian .....	4
1.7 Sistematika Laporan.....	5
<b>BAB II KAJIAN PUSTAKA .....</b>	<b>6</b>
2.1 Landasan Teori.....	6
2.2 Penelitian Sebelumnya .....	8
2.2.1 <i>Serverless</i> dan E-menu Restoran.....	8
2.2.2 <i>Serverless</i> dan IoT.....	8
2.2.3 <i>Serverless</i> dan <i>Chatbot</i> .....	9
2.2.4 <i>Serverless</i> dan <i>Bisnis Cloud</i> .....	10
2.2.5 Analisis Terhadap Penerapan <i>Serverless</i> .....	10
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>12</b>
3.1 Analisis Kebutuhan .....	12
3.2 Perancangan Perilaku .....	15
3.2.1 <i>Activity Diagram</i> Admin .....	15
3.2.2 <i>Activity Diagram</i> Perangkat Desa .....	17
3.2.3 <i>Activity Diagram</i> Kepala Desa .....	25
3.3 Perancangan Sistem .....	26
3.3.1 Rancangan Antarmuka .....	26
3.3.2 Rancangan Basis Data.....	37
3.3.3 Rancangan Penerapan <i>Serverless Computing</i> .....	38
3.4 Pengujian Sistem.....	39
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>40</b>
4.1 Implementasi .....	40
4.1.1 Halaman Antar Muka Admin.....	40
4.1.2 Halaman Antar Muka Perangkat Desa.....	42
4.1.3 Halaman Antar Muka Kepala Desa .....	50
4.1.4 Implementasi <i>Database</i> .....	51
4.2 Pengujian Sistem.....	56
4.2.1 Pengujian Keamanan.....	56

4.2.2 Pengujian Kinerja.....	58
BAB V KESIMPULAN .....	60
5.1 Kesimpulan .....	60
5.2 Saran.....	60
DAFTAR PUSTAKA.....	62

**DAFTAR TABEL**

Tabel 3.1 <i>Use Case</i> oleh Perangkat Desa .....	13
Tabel 3.2 <i>Use Case</i> oleh Admin .....	14
Tabel 3.3 <i>Use Case</i> oleh Kepala Desa .....	15
Tabel 4.1 Implementasi <i>Database</i> .....	52

## DAFTAR GAMBAR

Gambar 2.1 Popularitas <i>Serverless</i> dari Tahun ke Tahun.....	7
Gambar 3.1 <i>Use Case</i> Diagram .....	12
Gambar 3.2 <i>Activity</i> Diagram Mengelola Akun oleh Admin. ....	16
Gambar 3.3 <i>Activity</i> Diagram Mengelola Data Penduduk Tetap oleh Perangkat Desa. ....	18
Gambar 3.4 <i>Activity</i> Diagram Mengelola Data Penduduk Menikah oleh Perangkat Desa.....	19
Gambar 3.5 <i>Activity</i> Diagram Mengelola Data Penduduk Bercerai oleh Perangkat Desa. ....	20
Gambar 3.6 <i>Activity</i> Diagram Mengelola Data Penduduk Meninggal oleh Perangkat Desa...21	21
Gambar 3.7 <i>Activity</i> Diagram Mengelola Data Penduduk Lahir oleh Perangkat Desa. ....	22
Gambar 3.8 <i>Activity</i> Diagram Mengelola Data Penduduk Pindah oleh Perangkat Desa.....	23
Gambar 3.9 <i>Activity</i> Diagram Mengelola Data Penduduk Datang oleh Perangkat Desa. ....	24
Gambar 3.10 <i>Activity</i> Diagram <i>Dashboard</i> oleh Perangkat Desa.....	25
Gambar 3.11 <i>Activity</i> Diagram <i>Dashboard</i> oleh Kepala Desa. ....	26
Gambar 3.12 Rancangan Kelola Halaman Pengguna .....	27
Gambar 3.13 Rancangan Kelola Halaman Penduduk Tetap.....	28
Gambar 3.14 Rancangan Kelola Halaman Penduduk Menikah.....	29
Gambar 3.15 Rancangan Kelola Halaman Penduduk Bercerai. ....	30
Gambar 3.16 Rancangan Kelola Halaman Penduduk Meninggal. ....	31
Gambar 3.17 Rancangan Kelola Halaman Penduduk Lahir. ....	32
Gambar 3.18 Rancangan Kelola Halaman Penduduk Pindah.....	33
Gambar 3.19 Rancangan Kelola Halaman Penduduk Datang. ....	34
Gambar 3.20 Rancangan Halaman <i>Dashboard</i> Perangkat Desa. ....	35
Gambar 3.21 Rancangan Halaman <i>Dashboard</i> Kepala Desa. ....	36
Gambar 3.22 Rancangan Basis Data.....	37
Gambar 3.23 Arsitektur <i>Serverless Computing</i> . ....	38
Gambar 4.1 Halaman <i>Dashboard</i> Admin. ....	41
Gambar 4.2 Halaman Tambah Akun Admin. ....	41
Gambar 4.3 Halaman <i>Dashboard</i> Perangkat Desa. ....	42
Gambar 4.4 Halaman Menikah Perangkat Desa. ....	42
Gambar 4.5 Halaman <i>Form</i> Menikah Perangkat Desa. ....	43
Gambar 4.6 Halaman Bercerai Perangkat Desa.....	43
Gambar 4.7 Halaman <i>Form</i> Bercerai Perangkat Desa. ....	44
Gambar 4.8 Halaman Meninggal Perangkat Desa. ....	44

Gambar 4.9 Halaman <i>Form</i> Meninggal Perangkat Desa. ....	45
Gambar 4.10 Halaman Kelahiran Perangkat Desa. ....	45
Gambar 4.11 Halaman <i>Form</i> Lahir Perangkat Desa.....	46
Gambar 4.12 Halaman Datang Perangkat Desa.....	46
Gambar 4.13 Halaman <i>Form</i> Datang Perangkat Desa.....	47
Gambar 4. 14 Halaman Pindah Perangkat Desa. ....	47
Gambar 4.15 Halaman <i>Form</i> Pindah Perangkat Desa. ....	48
Gambar 4.16 Halaman Penduduk Perangkat Desa. ....	48
Gambar 4.17 Halaman <i>Form</i> Penduduk Perangkat Desa. ....	49
Gambar 4.18 Halaman Kartu Keluarga Perangkat Desa. ....	49
Gambar 4.19 Halaman <i>Form</i> Kartu Keluarga Perangkat Desa.....	50
Gambar 4.20 Halaman <i>Dashboard</i> Penduduk Kepala Desa.....	51
Gambar 4.21 Halaman Kartu Keluarga Kepala Desa. ....	51
Gambar 4.22 <i>Rules False</i> . ....	56
Gambar 4.23 <i>Rules True</i> . ....	57
Gambar 4.24 <i>Rules Auth</i> . ....	57
Gambar 4.25 <i>Logs</i> . ....	58
Gambar 4.26 <i>Logs</i> . ....	59

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Sistem Informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi (Puspitasari, 2016). Sistem informasi (SI) adalah elemen kunci dalam lingkungan bisnis dan organisasi modern yang berperan penting dalam mengumpulkan, menyimpan, mengelola, dan mengolah data untuk mendukung pengambilan keputusan yang efisien dan efektif.

Perkembangan sistem informasi telah mengalami transformasi yang signifikan dalam beberapa tahun terakhir. Organisasi ataupun individu mengandalkan sistem informasi konvensional yang memerlukan investasi besar dalam infrastruktur server, manajemen teknis yang rumit, dan pemeliharaan rutin. Perkembangan teknologi informasi, terutama komputasi awan (*cloud computing*), telah mengubah cara mengakses, menyimpan, dan membagikan informasi. Hal ini telah membuka peluang besar dalam pengembangan sistem informasi yang lebih canggih dan terintegrasi.

*Cloud computing* adalah model penyediaan layanan teknologi informasi (IT) yang memungkinkan untuk mengakses sumber daya komputasi, penyimpanan, dan layanan melalui internet. Dalam model ini, penyedia layanan *cloud* mengelola infrastruktur fisik, seperti server dan jaringan, sehingga pengguna dapat fokus pada pengembangan aplikasi dan penggunaan sumber daya IT tanpa harus khawatir tentang aspek infrastruktur. Salah satu perkembangan teknologi *cloud computing* modern adalah teknologi *serverless computing*.

*Serverless computing* atau komputasi nirserver, adalah salah satu layanan *cloud computing* yang penyediaannya melakukan pemeliharaan, pembaruan keamanan, penyediaan, dan pemantauan keandalan server yang digunakan oleh pengembang (Castro et al., 2019). *Serverless computing* memungkinkan pengembang untuk fokus pada kode aplikasi mereka tanpa perlu memikirkan infrastruktur server tradisional, seperti pengaturan server, skalabilitas vertikal, atau pemantauan yang rumit. Teknologi ini memungkinkan aplikasi dijalankan dalam lingkungan yang dikendalikan oleh penyedia layanan *cloud*, yang akan menangani pemeliharaan, peningkatan, dan manajemen sumber daya komputasi. Selain itu, *serverless* memungkinkan perusahaan untuk menghemat biaya operasional karena mereka hanya membayar berdasarkan penggunaan yang sebenarnya, bukan kapasitas tetap. Apabila

pengembang tidak sedang dalam menggunakan fitur-fitur *serverless* (idle), maka pengembang tidak perlu membayar fitur tersebut (Castro et al., 2019).

Teknologi berbasis *cloud computing* yang menggunakan konsep *serverless* telah terbukti sebagai solusi yang efisien dalam berbagai situasi, salah satunya dalam pembuatan aplikasi E-Menu Restoran. Hasil penelitian sebelumnya (Oktaviani et al., 2021) menunjukkan bahwa pemanfaatan teknologi *serverless* dalam konteks ini mampu memberikan tingkat skalabilitas yang luas, hingga mencapai pasar global, dan pada saat yang sama mengurangi biaya pengelolaan. Manfaat dan potensi teknologi *serverless* dapat dimanfaatkan di berbagai bidang, termasuk dalam pengelolaan data kependudukan. Penelitian mengenai *serverless computing* memiliki peran penting dalam memberikan pemahaman yang lebih mendalam dan implementasi teknologi *serverless computing* ke dalam aplikasi yang lebih luas.

Perkembangan teknologi informasi juga mempengaruhi cara pengelolaan data kependudukan. Biasanya, pemerintah desa harus mengelola infrastruktur server yang kompleks untuk menyimpan dan mengelola data penduduk. *Serverless computing* berbeda dengan konsep komputasi awan biasa dalam arti bahwa infrastruktur dan platform tempat layanan yang dijalankan tersembunyi dari pelanggan (Shafiei et al., 2019). Dengan kemunculan konsep *serverless computing*, pendekatan pengelolaan data kependudukan diharapkan dapat menjadi lebih efisien, skalabel, dan hemat biaya.

Dalam lingkup pengelolaan data kependudukan, *serverless computing* memungkinkan pemerintah desa untuk dengan mudah mengumpulkan, menyimpan, mengelola, dan menganalisis data penduduk tanpa harus menghadapi kendala infrastruktur yang kompleks. Penerapan *serverless computing* dalam pengelolaan data kependudukan juga memungkinkan integrasi yang lebih baik dengan layanan data lainnya, termasuk basis data *serverless*, penyimpanan objek, dan layanan otentikasi. Hal ini membuka pintu bagi inovasi dalam pelayanan kepada masyarakat dan peningkatan efisiensi dalam proses administratif desa.

Walaupun *serverless computing* membawa berbagai keunggulan, penggunaannya juga menghadapi sejumlah masalah yang perlu diatasi. Isu terkait keamanan data, privasi, dan ketersediaan layanan tetap menjadi hal yang perlu diberi perhatian utama dalam pengelolaan data kependudukan menggunakan teknologi ini. Oleh karena itu, diperlukan perencanaan yang matang dan pemahaman yang mendalam untuk mengintegrasikan *serverless computing* ke dalam manajemen data kependudukan.

Berdasarkan penjelasan diatas, diperlukan penelitian untuk mengetahui konsep *serverless computing* dalam pengelolaan data kependudukan. Penelitian ini juga akan

membahas potensi keuntungan dan tantangan yang terkait dengan penggunaan *serverless computing* dalam pengelolaan data penduduk, serta memberikan pandangan tentang peran teknologi ini dalam transformasi pengelolaan data kependudukan di tingkat desa. Bagaimana *serverless* dapat digunakan dalam pengelolaan data kependudukan akan memberikan pedoman berharga bagi pemerintah dalam upaya meningkatkan efisiensi dan kualitas layanan kepada masyarakat dengan memanfaatkan teknologi *serverless*.

## 1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini, diantaranya :

1. Bagaimana konsep *serverless computing* dapat diterapkan dalam pengelolaan data kependudukan di tingkat desa?
2. Apa saja keuntungan dan tantangan yang terkait dengan penggunaan *serverless computing* dalam pengelolaan data penduduk di tingkat desa?

## 1.3 Batasan Masalah

Penelitian ini akan membatasi lingkungannya pada pengelolaan data kependudukan di tingkat desa, sehingga hasil penelitian tidak secara langsung berlaku untuk skala yang lebih besar seperti tingkat nasional.

## 1.4 Tujuan Penelitian

Beberapa tujuan penelitian ini, diantaranya :

1. Mengeksplorasi konsep *serverless computing* dan bagaimana konsep tersebut agar dapat diterapkan dalam pengelolaan data kependudukan di tingkat desa.
2. Mengidentifikasi keuntungan dan tantangan yang terkait dengan penggunaan *serverless computing* dalam pengelolaan data penduduk di tingkat desa.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Memberikan wawasan tentang potensi penerapan teknologi *serverless* dalam pengelolaan data kependudukan di tingkat desa.
2. Membantu pemerintah desa dalam memahami keuntungan dan tantangan yang terkait dengan penggunaan *serverless computing*.
3. Memberikan kontribusi dalam literatur tentang pengelolaan data kependudukan dengan pendekatan *serverless computing*.

## 1.6 Metode Penelitian

Metode Penelitian yang digunakan untuk membuat aplikasi data kependudukan dengan pendekatan *serverless computing* meliputi :

- A. Identifikasi Kebutuhan: Menganalisis kebutuhan dan tuntutan pengelolaan data kependudukan desa untuk perancangan sistem seperti *input*, *output*, proses, dan antarmuka.
- B. Proses perancangan sistem : Dimulai dengan identifikasi fitur yang dibutuhkan, seperti manajemen penduduk, pencatatan peristiwa kependudukan, dan pembuatan laporan. Setelah itu, analisis kebutuhan seperti membuat *Use Case Diagram*, *Activity Diagram*, Skema Basis Data, dan *User Interface*.
- C. Pengembangan sistem : melibatkan implementasi desain yang telah dibuat. Proses ini mencakup pengembangan berbagai komponen sistem, pengujian unit untuk memastikan fungsionalitas yang benar, dan integrasi sistem agar seluruh komponen dapat berinteraksi dengan baik
- D. Pengujian sistem: Pengujian sistem dalam pengembangan aplikasi data kependudukan berbasis *serverless computing* bertujuan untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan yang telah diidentifikasi dan dirancang sebelumnya. Pengujian dilakukan untuk menilai kualitas, kinerja, dan keamanan sistem.

## 1.7 Sistematika Laporan

Sistematika penulisan pada tugas akhir ini bertujuan untuk memudahkan dalam memahami isi dari laporan ini. Pada laporan ini terdiri dari berbagai pokok pembahasan, sebagai berikut :

### **BAB I PENDAHULUAN**

Bab ini berisi pembahasan mengenai latar belakang aplikasi data kependudukan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika laporan.

### **BAB II LANDASAN TEORI**

Bab ini berisi pembahasan mengenai teori teori yang berhubungan dengan penelitian yaitu mengenai *serverless computing* dan penjelasan mengenai tinjauan pustaka.

### **BAB III METODOLOGI PENELITIAN**

Bab ini berisi pembahasan tentang analisis dan perancangan sistem, seperti *use case* diagram, *activity* diagram, perancangan perilaku, dan perancangan basis data.

### **BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM**

Bab ini berisi pembahasan mengenai hasil pembuatan sistem yang telah dilakukan. Pembahasan yang dilakukan seperti implementasi sistem dan pengujian sistem, serta perancangan antarmuka untuk sistem.

### **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari hasil dan analisis membangun sistem yang telah dilakukan, serta saran untuk perbaikan dan pengembangan pada aplikasi data kependudukan ini untuk kedepannya.

## BAB II KAJIAN PUSTAKA

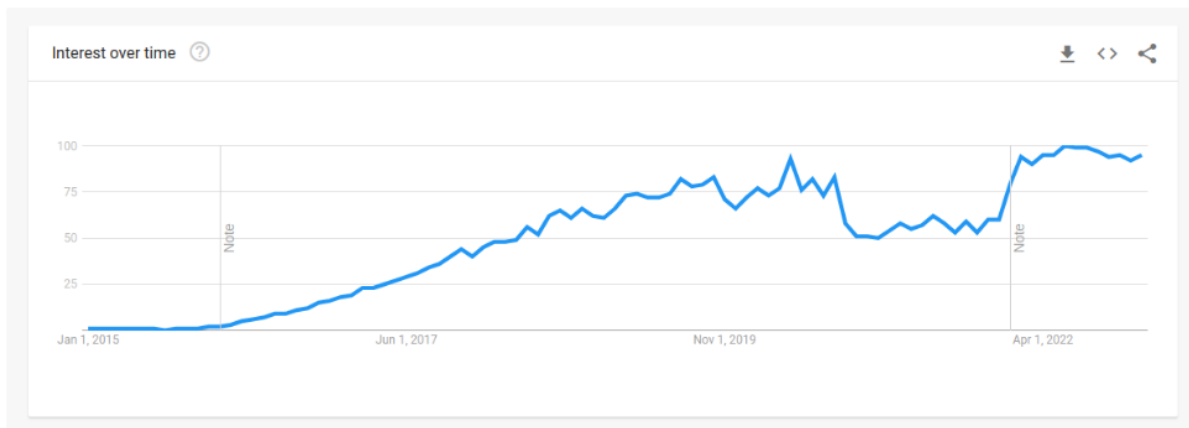
### 2.1 Landasan Teori

*Serverless computing* atau komputasi tanpa server sebenarnya tidak memiliki definisi secara pasti (Haque et al., 2022). (Castro et al., 2019) mendefinisikan *serverless* sebagai salah satu layanan *cloud computing* yang penyediannya melakukan pemeliharaan, pembaruan keamanan, penyediaan, dan pemantauan keandalan server yang digunakan oleh pengembang. *Serverless* menurut (Haque et al., 2022) adalah layanan yang terdiri atas *FaaS* (*Function-as-a-Service*) dan *BaaS* (*Backend-as-a-Service*). *FaaS* untuk mengaktifkan isolasi dan pemanggilan fungsi, sementara *BaaS* menyediakan dukungan *backend* secara keseluruhan untuk layanan *online*. Saat menggunakan *FaaS* pengembang hanya perlu melakukan pengiriman kode sebagai sinyal untuk penyedia layanan mengeksekusi kode yang dikirimkan oleh pengembang. Maka dari itu, *FaaS* dapat disebut juga dengan *serverless function* (Nupponen & Taibi, 2020). Autentikasi dan notifikasi adalah contoh dari *BaaS* layanan *online* yang menangani tugas tertentu. Baik *BaaS* dan *FaaS* tidak memerlukan pengelolaan dari pelanggan. Meskipun *FaaS* hanya menawarkan untuk menjalankan fungsi pengguna, *BaaS* menawarkan serangkaian fungsi lengkap layanan *online* (Shafiei et al., 2019).

Ada dua karakteristik agar sebuah layanan dapat disebut dengan *serverless*. Karakteristik yang pertama yakni teknologi *serverless* menghadirkan konsep skalabilitas otomatis, di mana sistem secara otomatis menangani peningkatan dan penurunan permintaan dengan menambahkan atau menghapus sumber daya komputasi sesuai kebutuhan. Pengelolaan server dilakukan sepenuhnya oleh penyedia layanan, sehingga pengembang tidak perlu mengelola atau memelihara infrastruktur server secara langsung. Ketika fitur-fitur *serverless* tidak digunakan (*idle*), pengembang tidak perlu melakukan konfigurasi atau pemantauan server tambahan (Castro et al., 2019).

Penyedia layanan atau *cloud provider* menyediakan layanan infrastruktur server, manajemen sistem, dan segala hal yang berkaitan dengan penyimpanan dan operasi sistem. Singkatnya *serverless* berarti server, memori atau *storage* yang disediakan oleh *cloud provider*. Hal ini yang membedakan dengan VPS dan *hosting* yang mengharuskan pengembang untuk membeli keseluruhan server baik memori maupun melakukan kustomisasi pada server. Penyedia teknologi *serverless* diawali oleh platform AWS Lambda, diikuti oleh Apache

OpenWhisk, Azure Functions, Google Cloud Functions (Rajeshwari B Mathapati, 2023), Iron.io IronFunctions, dan Open Lambda (McGrath & Brenner, 2017).



Gambar 2.1 Popularitas *Serverless* dari Tahun ke Tahun.

Sumber: (Saputra, 2023)

Dapat dilihat dari Gambar 2.1 di atas, 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. Sebaliknya, perhatian civitas akademika masih terbatas (Baldini et al., 2017).

Dengan adanya teknologi *serverless*, pengembang dapat mengintegrasikan dengan layanan lain dengan mudah, misalnya pada basis data dan penyimpanan objek. *Serverless* juga mendukung pengembang untuk menggunakan berbagai macam bahasa pemrograman. Python, Java script, dan Java adalah beberapa contoh bahasa pemrograman yang dapat digunakan dalam teknologi *serverless*. *Cloud Provider* berlomba-lomba menawarkan layanan basis data *serverless* baik SQL dan NoSQL. Beberapa basis data *serverless* yang disediakan oleh *cloud provider* besar yang dapat digunakan untuk membangun aplikasi dengan arsitektur *serverless*, seperti SQL yang terdiri atas Amazon Aurora Serverless (kompatibel dengan MySQL dan PostgreSQL) dan Azure SQL Database Serverless (Microsoft SQL Server). Sedangkan NoSQL terdiri atas Amazon DynamoDB, MongoDB Atlas, FaunaDB, Google Cloud Firestore.

## 2.2 Penelitian Sebelumnya

Penggunaan teknologi *serverless* pada data kependudukan belum pernah dilakukan sebelumnya. Berikut adalah beberapa pemanfaatan teknologi *serverless* yang dapat memudahkan dan dapat menjadi acuan untuk penelitian ini, penjelasan terkait dijabarkan pada poin 2.2.1 sampai 2.2.5.

### 2.2.1 *Serverless* dan E-menu Restoran

Dalam penelitian mengenai penggunaan teknologi *cloud computing* dan arsitektur *serverless* lambda dalam pengembangan E-Menu restoran, (Oktaviani et al., 2021) menyajikan argumen yang kuat terkait dengan perlunya peningkatan kualitas sistem layanan restoran di Indonesia. Mereka menunjukkan bahwa persaingan global, terutama dengan masuknya restoran internasional, memaksa restoran di Indonesia untuk meningkatkan kualitas pelayanan mereka. Dalam konteks ini, studi tersebut mengusulkan penerapan sistem informasi E-menu restoran sebagai solusi modern dan efisien. Alasan di balik pemilihan teknologi *cloud computing* dan arsitektur *serverless* lambda dijelaskan sebagai upaya untuk menciptakan sistem layanan yang tidak hanya memudahkan pelanggan dalam melakukan pembelian *online* tetapi juga meminimalkan biaya operasional restoran.

Oleh karena itu, penelitian ini memberikan landasan untuk merancang dan mengembangkan aplikasi E-Menu restoran yang menggunakan teknologi *cloud computing* dan arsitektur *serverless* lambda dengan tujuan utama meningkatkan kualitas pelayanan restoran, mempermudah pelanggan dalam pemesanan online, meminimalkan biaya restoran, dan mempercepat pengelolaan dan penyajian informasi data. Sebagai metodologi penelitian, (Oktaviani et al., 2021) mencakup studi literatur, observasi, dan wawancara. Studi literatur digunakan untuk merinci landasan teoritis dari penerapan teknologi ini, observasi dilakukan untuk memahami kebutuhan pelanggan dan kondisi operasional restoran, dan wawancara dengan pemangku kepentingan membantu dalam merinci persyaratan dan harapan mereka terhadap sistem yang akan dikembangkan.

### 2.2.2 *Serverless* dan IoT

Dalam dunia Internet of Things (IoT), perangkat memiliki keterbatasan pada kapasitas penyimpanan dan komputasi, yang dapat menyebabkan berbagai masalah seperti kehandalan, kinerja, skalabilitas, keamanan, dan privasi (Bayu et al., 2022). Salah satu solusi yang diusulkan untuk mengatasi permasalahan tersebut adalah mengintegrasikan perangkat IoT dengan teknologi *cloud computing*. Meskipun integrasi ini dapat memberikan manfaat, namun muncul masalah performa dalam komputasi akibat peningkatan jumlah perangkat yang

terhubung. Untuk mengatasi hal ini, penelitian mengusulkan pengembangan platform IoT Cloud berbasis layanan komputasi *serverless* menggunakan Google Cloud Platform (GCP) (Bayu et al., 2022).

Studi yang dilakukan oleh (Mulyani & Oktiawati, 2022) mengenai penggunaan teknologi *cloud* dan *serverless* dalam *monitoring cold chain logistic*, mereka menghadapi tantangan keterbatasan perangkat IoT seperti penyimpanan data dan komputasi. Dalam menjawab tantangan ini, penelitian ini memandang teknologi *cloud* sebagai solusi, khususnya dengan menerapkan konsep *serverless*. Tujuan penelitian ini adalah untuk menguji performa AWS IoT sebagai layanan *serverless* yang diintegrasikan dengan perangkat IoT *gateway* dalam memantau suhu pada *cold chain logistics*. Manfaat dari penelitian ini mencakup solusi yang lebih efektif dan efisien dalam pemantauan suhu pada *cold chain logistics*, penghematan waktu dan biaya, serta peningkatan efisiensi dalam pendistribusian produk *perishable*. Studi ini merinci perancangan perangkat dan alur sistem, skenario pengujian, dan implementasi kerangka kerja Layanan Berbasis Lokasi (LBS) yang terintegrasi dengan Platform Google Maps. Kesimpulan dari penelitian ini menegaskan bahwa penggunaan arsitektur tanpa server, seperti AWS IoT, dapat berhasil diimplementasikan dalam situasi nyata dan memberikan hasil yang memuaskan.

### **2.2.3 Serverless dan Chatbot**

Penelitian yang dilakukan oleh (Facta, 2018) menyoroti kebutuhan akan efisiensi waktu dalam menangani permintaan pemesanan tiket dan layanan lainnya. Mereka mengembangkan aplikasi *chatbot* untuk meningkatkan efisiensi dan efektivitas pemesanan tiket dan layanan lainnya. Tujuan penelitian ini adalah untuk memberikan solusi yang lebih efektif dan efisien dalam memantau suhu pada *cold chain logistics*. Manfaat dari penelitian ini mencakup pengembangan aplikasi *chatbot* yang dapat meningkatkan efisiensi dan efektivitas pemesanan tiket dan layanan lainnya. Sistem ini terdiri dari tiga bagian: *webhook Node JS*, layanan *Wit.AI* Layanan NLP, dan *Ticket.com Order API*.

(Yan et al., 2016) membahas tren penggunaan *chatbot* dalam komunikasi konsumen di seluruh dunia. Mereka menyoroti kepopuleran layanan bahasa alami dan kemudahan pengembangan *chatbot* dengan biaya yang terjangkau. Tujuan penelitian ini adalah untuk menyelidiki penggunaan model pemrograman *serverless* sebagai kerangka kerja *mashup* untuk memodularisasi pemrosesan *domain-specific* pada *chatbot*. Manfaat dari penelitian ini mencakup pengembangan *chatbot* yang lebih efektif dan efisien dengan menggunakan model pemrograman *serverless* sebagai kerangka kerja *mashup*. Penelitian ini mencakup tinjauan

literatur, desain arsitektur, dan pengembangan prototipe untuk menyelidiki penggunaan komputasi tanpa server sebagai kerangka kerja mashup untuk pengembangan *chatbot*.

#### **2.2.4 Serverless dan Bisnis Cloud**

Hasil dari penelitian yang dilakukan oleh (Utomo & Falahah, 2020) menyoroti tren penggunaan Function as a Service (FaaS) sebagai tren berikutnya dalam bisnis *cloud*. Mereka menunjukkan bahwa penggunaan GitHub sebagai web hosting dan pendekatan JAMstack belum begitu populer di kalangan pengembang di Indonesia. Tujuan penelitian ini adalah untuk menunjukkan bagaimana mengimplementasikan pendekatan JAMstack dalam mendesain dan membangun situs web, serta menggunakan GitHub sebagai platform *hosting*. Manfaat dari penelitian ini adalah memberikan panduan langkah demi langkah tentang cara mengembangkan dan mempublikasikan situs web menggunakan pendekatan JAMstack dan GitHub Pages. Metode penelitian ini melibatkan membangun aplikasi web menggunakan arsitektur JAMstack, dengan Hugo sebagai Static Site Generator dan GitHub Pages sebagai Content Delivery Network (CDN).

#### **2.2.5 Analisis Terhadap Penerapan Serverless**

Konsep komputasi tanpa server menawarkan efisiensi dan kemudahan penggunaan, tetapi terdapat berbagai laporan yang kontradiktif mengenai keputusan perusahaan untuk beralih ke model tanpa server. Studi yang melibatkan analisis terhadap 89 aplikasi tanpa server mencoba memberikan pemahaman yang lebih baik tentang paradigma ini, termasuk alasan praktisi memilihnya, kapan penerapannya paling sesuai, dan bagaimana implementasinya dilakukan (Eismann et al., 2021). Studi ini memberikan wawasan komprehensif dengan tujuan membantu manajer dan pengembang membuat keputusan yang tepat terkait penggunaan aplikasi tanpa server.

(Nupponen & Taibi, 2020) juga menjelaskan konsep *serverless* yang menarik perhatian industri. Dalam penelitian mereka, fokus utama adalah mengidentifikasi dan mengatasi praktik-praktik buruk dalam pengembangan aplikasi tanpa server. Studi ini didukung oleh survei dan kelompok fokus dengan partisipasi praktisi, yang memberikan informasi empiris komprehensif untuk membantu pengembang perangkat lunak membangun solusi *serverless* yang lebih baik.

Di sisi lain, (Castro et al., 2019) menyoroti tren pergeseran arsitektur aplikasi *enterprise* ke kontainer dan mikro layanan, yang mendorong munculnya *serverless computing*. Studi ini membahas manfaat penggunaan *serverless computing* di *cloud*, seperti keuntungan ekonomi dan kemudahan penggunaan, serta memberikan panduan praktis bagi pengembang dalam

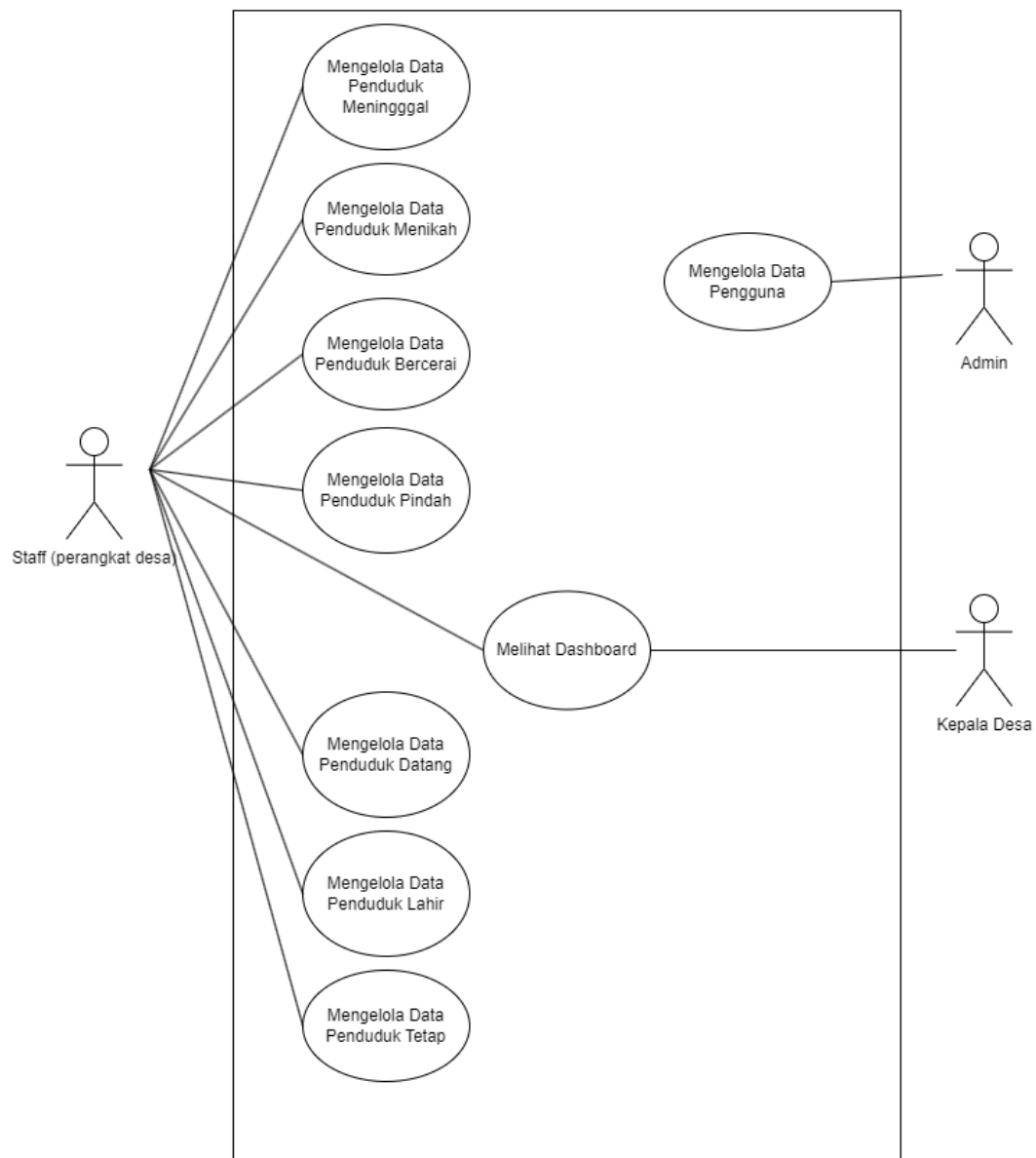
merancang dan mengimplementasikan aplikasi *serverless*. Dengan pemahaman mendalam tentang *serverless computing*, organisasi dan pengembang dapat mengoptimalkan pengembangan aplikasi *cloud* mereka.

Dapat disimpulkan dari penelitian sebelumnya bahwa teknologi *serverless* menawarkan berbagai manfaat dalam efisiensi pengelolaan data, fleksibilitas integrasi dengan sistem lain, dan pengurangan biaya operasional, yang dapat diterapkan pada sistem pengelolaan data kependudukan di desa. Penerapan *serverless computing* berpotensi meningkatkan layanan kependudukan dengan memastikan proses yang cepat dan efisien tanpa memerlukan infrastruktur server fisik yang besar. Teknologi ini juga dapat menangani masalah skalabilitas dan keterbatasan komputasi, memungkinkan penambahan pengguna tanpa mengorbankan kinerja. Penerapan *serverless* dalam *chatbot* turut mendukung efisiensi dengan memberikan informasi dasar bagi pengguna. Selain itu, *serverless computing* dan Function as a Service (FaaS) mempermudah penerapan otomatisasi yang tidak rumit, sesuai untuk pengelolaan fungsi-fungsi berulang seperti pencatatan data real-time atau perubahan status kependudukan.

## BAB III METODOLOGI PENELITIAN

### 3.1 Analisis Kebutuhan

Terdapat tiga aktor utama pada *use case* sistem informasi data kependudukan desa yang dilakukan di Desa Gambasan, yaitu : Perangkat Desa, admin, dan kepala desa. Ketiga aktor tersebut dapat berinteraksi dengan sistem namun harus login terlebih dahulu. *Use case* diagram pada sistem ini dapat dilihat pada Gambar 3.1



Gambar 3.1 *Use Case* Diagram

Terdapat 9 (Sembilan) Use Case yang dilakukan oleh 3 aktor di atas yaitu : Perangkat Desa mengelola data penduduk datang (CRUD), Perangkat Desa mengelola data penduduk pindah (CRUD), Perangkat Desa mengelola data penduduk lahir (CRUD), Perangkat Desa mengelola data penduduk meninggal (CRUD), Perangkat Desa mengelola data penduduk menikah/bercerai (CRUD), Admin membuat, menampilkan, dan mengganti data pengguna (CRU), admin mengelola data penduduk tetap (CRUD), Kepala desa dan Perangkat Desa melihat dashboard data kependudukan.

Tabel 1 hingga Tabel 3 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 3.1 *Use Case* oleh Perangkat Desa

Nama UC	Keterangan
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, dengan atribut tambahan seperti tanggal pernikahan, nama pasangan, dan sebagainya.
Mengelola data penduduk bercerai	Perangkat desa dapat mengelola data penduduk yang sudah bercerai, dengan atribut tambahan seperti tanggal perceraian, alasan, dan sebagainya. Proses ini memungkinkan pengelolaan data yang lebih spesifik untuk penduduk yang telah bercerai.
	Perangkat desa dapat mengelola data penduduk yang telah meninggal, dengan atribut tambahan seperti tanggal kematian, penyebab, dan

Mengelola data penduduk meninggal	sebagainya. Proses ini memungkinkan pengelolaan data yang lebih spesifik untuk penduduk yang telah meninggal.
Mengelola data penduduk pindah	Perangkat desa dapat mengelola data penduduk yang pindah ke lokasi lain. Proses ini mencakup pembaruan alamat, tanggal pindah, dan lainnya untuk memastikan data penduduk pindah di tempat tujuan.
Mengelola data penduduk datang	Perangkat desa dapat mengelola data penduduk yang datang ke lokasi tersebut, dengan mengidentifikasi atribut yang perlu diisi seperti asal, tanggal kedatangan, dan lainnya. Proses ini memungkinkan pencatatan data penduduk yang baru datang.
Mengelola data penduduk lahir	Perangkat desa dapat mengelola data penduduk yang baru lahir, dengan atribut yang relevan seperti nama bayi, tanggal lahir, dan data orang tua. Proses ini memungkinkan pencatatan kelahiran dan pembaruan data keluarga.
Mengelola data penduduk tetap	Perangkat desa dapat mengelola data penduduk tetap, termasuk menambah, mengedit, dan menghapus data. Informasi seperti nama, alamat, nomor identitas, dan lainnya dapat dikelola melalui antarmuka pengguna.

Tabel 3.2 *Use Case* oleh Admin

<b>Nama UC</b>	<b>Keterangan</b>
Mengelola data pengguna	Admin dapat membuat, menampilkan, dan mengganti data pengguna, termasuk informasi seperti nama, alamat email, dan hak akses dalam

	sistem. Ini melibatkan pembuatan akun, pemantauan data pengguna, serta pembaharuan dan penyesuaian informasi pengguna yang terdaftar.
--	---

Tabel 3.3 *Use Case* oleh Kepala Desa

Nama UC	Keterangan
Melihat Dashboard	Kepala desa dapat melihat ringkasan dan visualisasi data dalam bentuk dashboard. Proses ini memberikan gambaran umum tentang data kependudukan desa.

### 3.2 Perancangan Perilaku

Gambar pada 3.2 hingga 3.10 akan menggambarkan proses dalam sistem manajemen kependudukan desa, dimulai dari admin yang membuat akun untuk Perangkat Desa dan kepala desa untuk keperluan *login* pengguna. Selain membuat akun untuk pengguna, admin dapat menampilkan dan mengganti data pengguna. Setelah pembuatan akun pengguna berhasil dilakukan, perangkat desa dan kepala desa dapat melakukan *login* akun. Setelah berhasil *login*, Perangkat Desa dan kepala desa akan diarahkan ke *dashboard* data kependudukan. Perbedaan mendasar antara akun Perangkat Desa dan kepala desa terletak pada hak akses pengelolaan data kependudukan. Perangkat desa dapat melakukan manajemen data untuk kategori seperti peristiwa penduduk menikah/bercerai, perpindahan/kematian, serta kelahiran/pendatang. Sebaliknya, akun kepala desa hanya memiliki akses untuk melihat *dashboard* utama. Setelah aktivitas di atas selesai dilakukan, baik perangkat desa maupun kepala desa dapat melakukan *logout* dari akun yang telah digunakan.

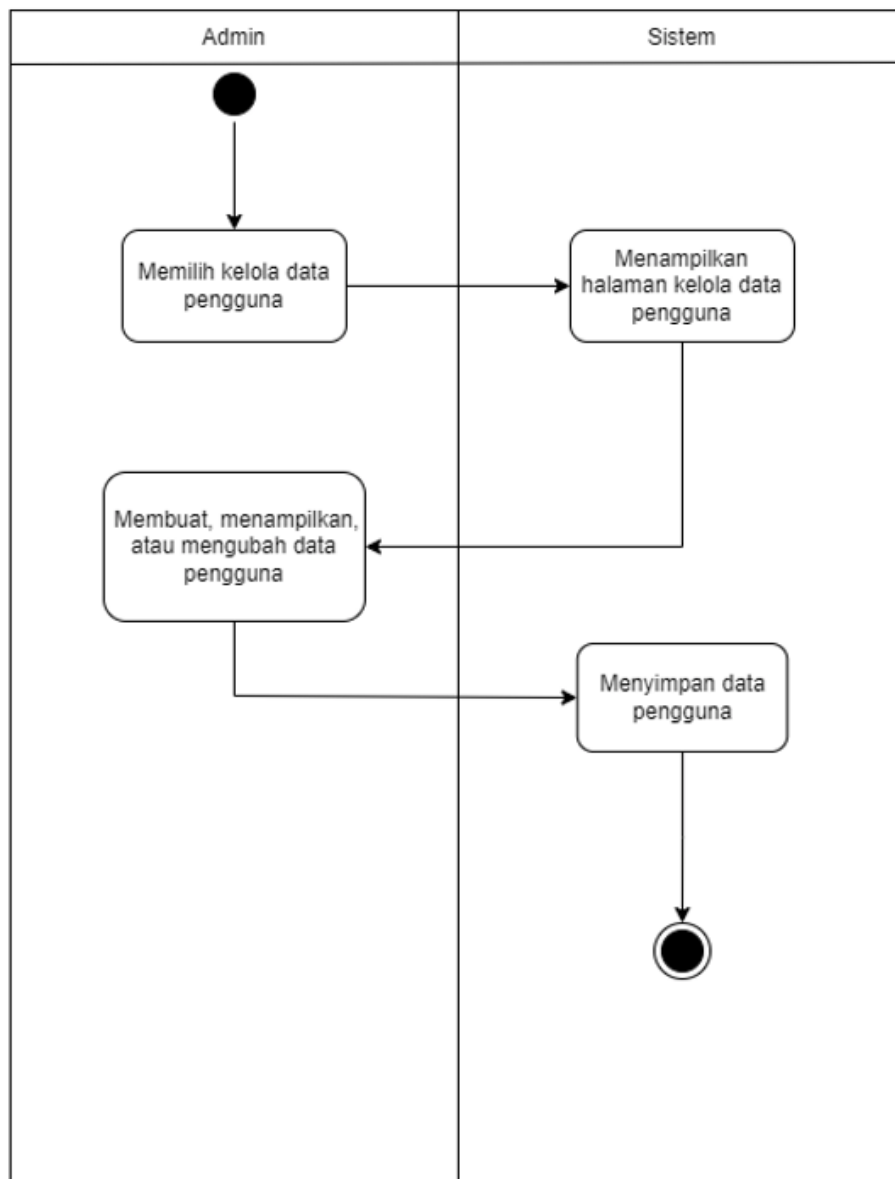
#### 3.2.1 Activity Diagram Admin

Penjelasan *Activity* diagram menggambarkan alur kerja atau proses yang dijalankan oleh admin dalam sistem. Diagram ini bertujuan untuk memberikan visualisasi yang jelas tentang alur langkah yang diambil oleh admin. Berikut adalah penjelasan activity diagram admin :

#### Mengelola Data Pengguna

Dalam pengelolaan data pengguna, Admin memiliki peran penting dalam membuat, melihat, dan mengubah informasi pengguna seperti nama, alamat email, dan hak akses di dalam sistem yang terinci dalam gambar 3.2. Ini mencakup proses membuat akun, memantau data

pengguna, serta melakukan pembaruan dan penyesuaian terhadap informasi pengguna yang sudah terdaftar di sistem.



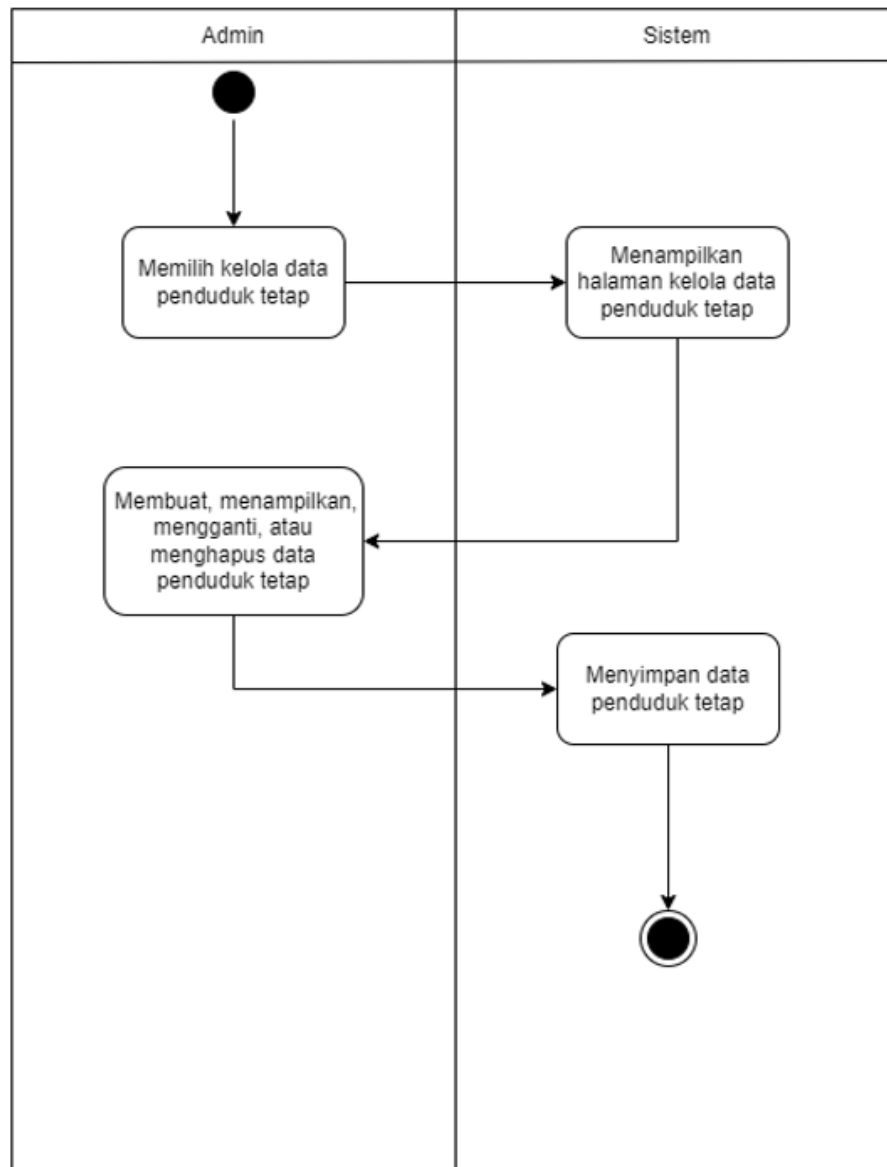
Gambar 3.2 Activity Diagram Mengelola Akun oleh Admin.

### 3.2.2 *Activity* Diagram Perangkat Desa

Penjelasan *activity* diagram menjelaskan alur proses yang dilakukan oleh perangkat desa dalam sistem pengelolaan data kependudukan. Diagram ini mengilustrasikan berbagai aktivitas yang dilakukan perangkat desa, seperti data penduduk tetap, kelahiran, kematian, pernikahan, perceraian, pindah, dan datang.

#### **Mengelola Data Penduduk Tetap**

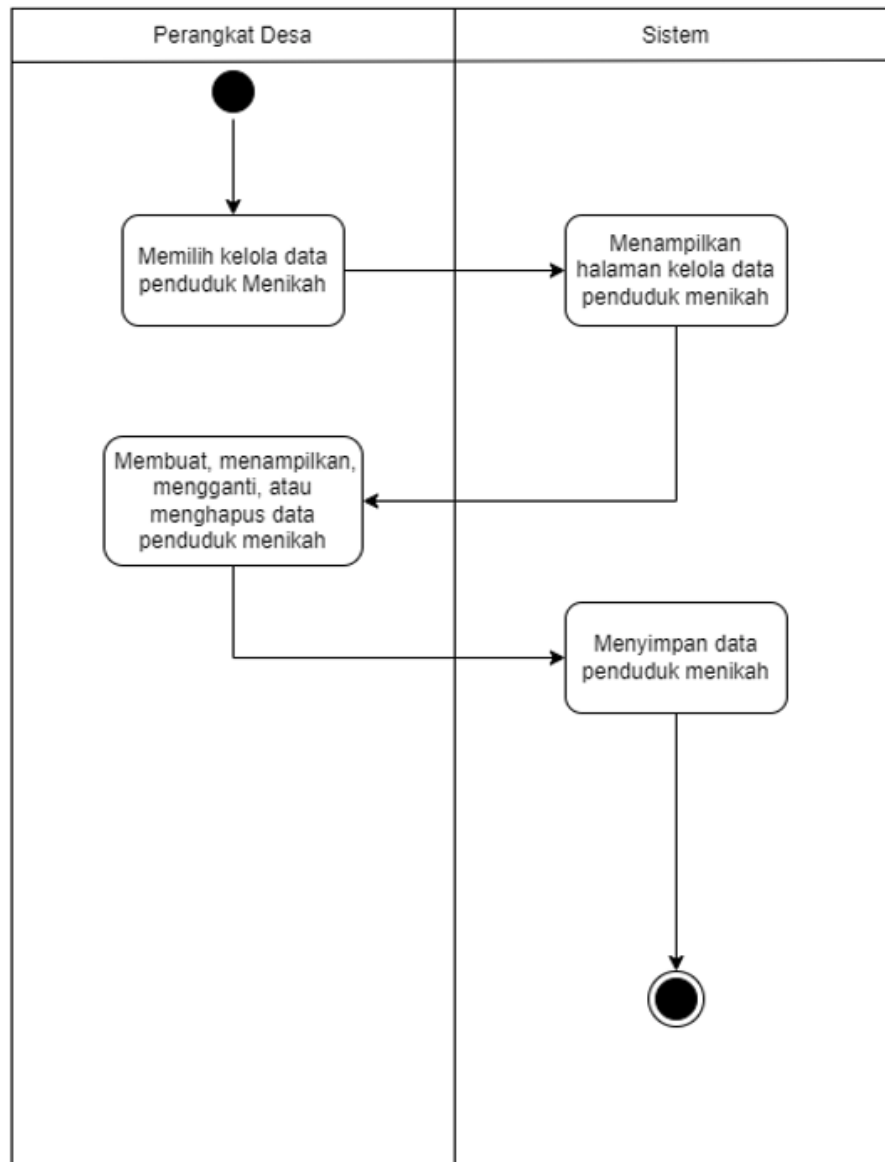
Dalam mengelola data penduduk tetap, Perangkat desa memiliki tanggung jawab untuk menambah, mengedit, dan menghapus informasi yang . Melalui aplikasi, perangkat desa dapat dengan mudah mengelola berbagai detail, seperti nama, alamat, nomor identitas, dan informasi penting lainnya terkait penduduk tetap. *Activity* Diagram dapat dilihat pada Gambar 3.3



Gambar 3.3 *Activity* Diagram Mengelola Data Penduduk Tetap oleh Perangkat Desa.

### Mengelola Data Penduduk Menikah

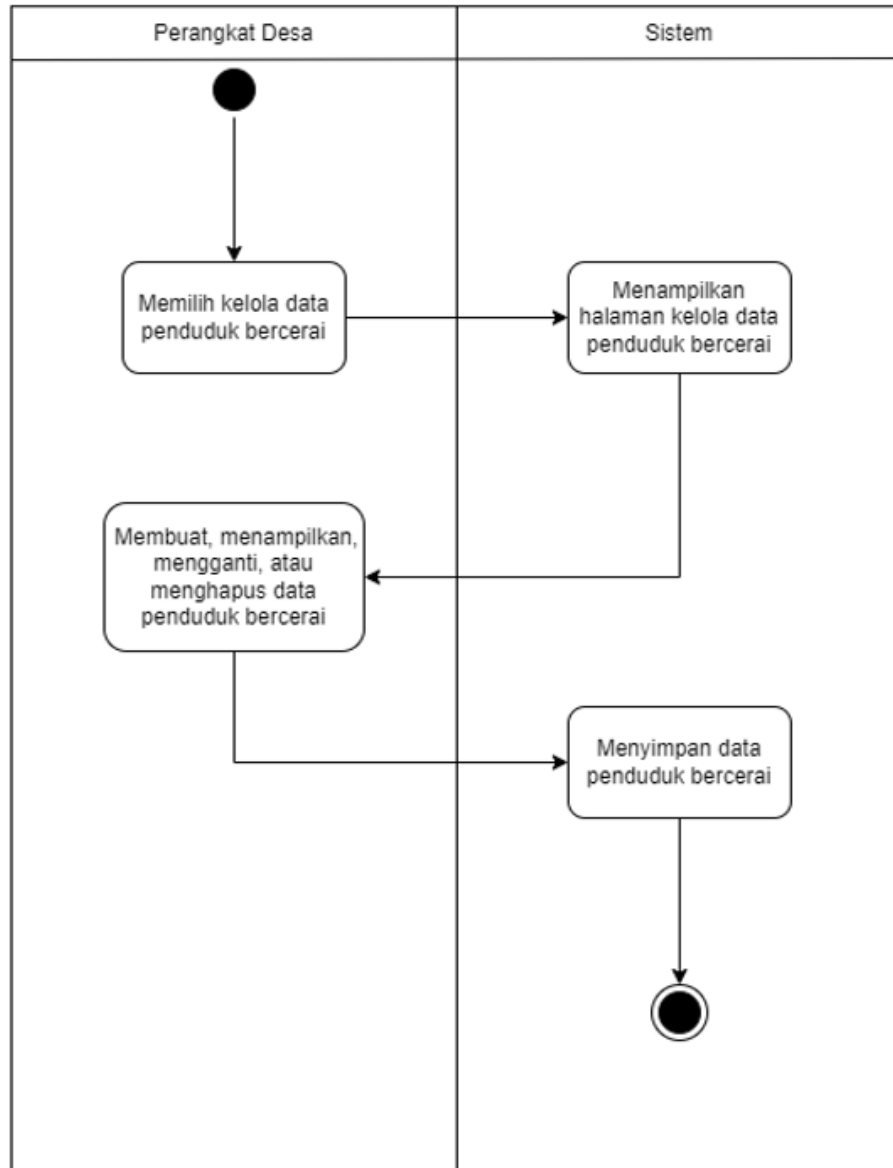
Dalam aktivitas perangkat desa, mereka dapat mengelola data penduduk yang sudah menikah, termasuk atribut tambahan seperti tanggal pernikahan, nama pasangan, dan informasi lainnya. Hal ini memungkinkan perangkat desa untuk mencatat dan memperbarui data penduduk yang telah menikah dengan lebih rinci. *Activity* Diagram dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Activity Diagram* Mengelola Data Penduduk Menikah oleh Perangkat Desa.

### Mengelola Data Penduduk Bercerai

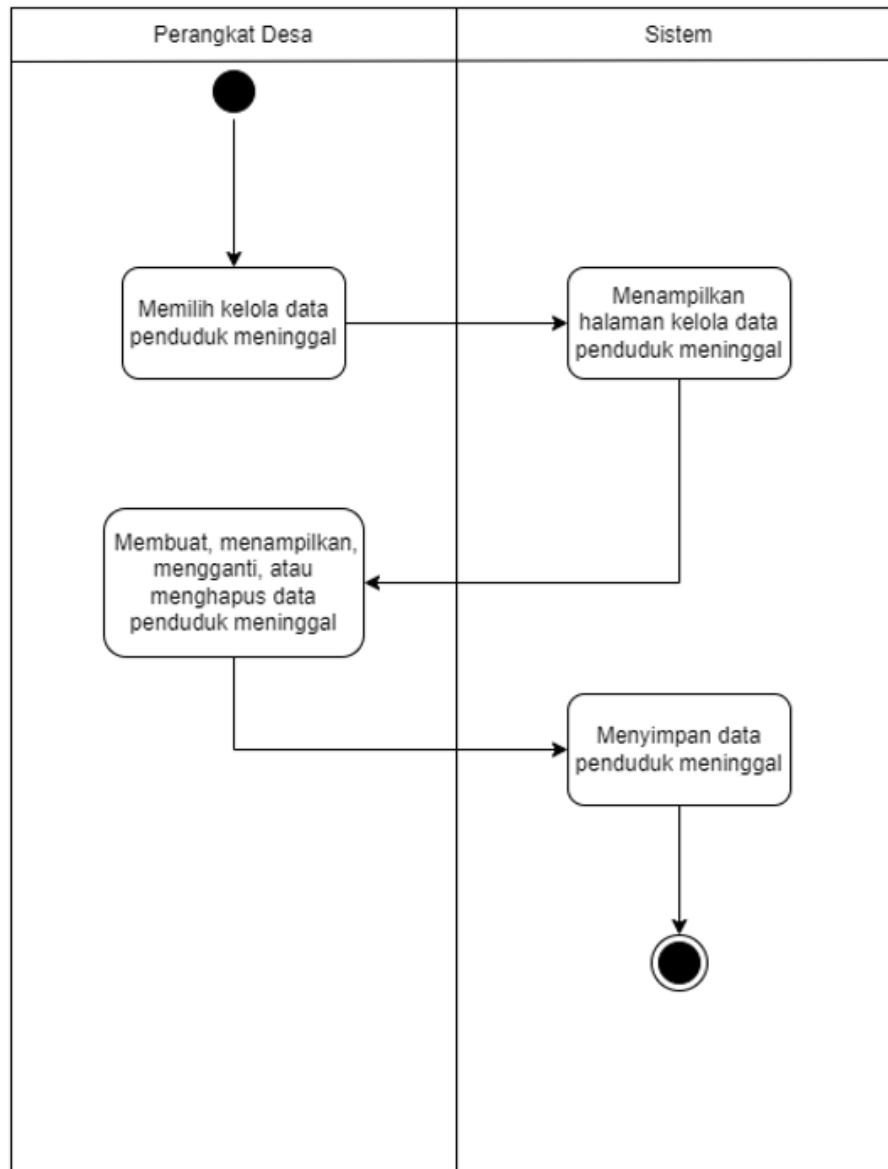
Pengelolaan data penduduk yang telah bercerai, perangkat desa dapat mengelola informasi tambahan seperti tanggal perceraian, alasan, dan detail lainnya. Proses ini memfasilitasi pengelolaan data yang lebih spesifik untuk penduduk yang telah mengalami perceraian. *Activity Diagram* dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Activity* Diagram Mengelola Data Penduduk Bercerai oleh Perangkat Desa.

### Mengelola Data Penduduk Meninggal

Aktivitas pengelolaan data penduduk yang telah meninggal, perangkat desa dapat mengelola atribut tambahan seperti tanggal kematian, penyebab, dan informasi lainnya. Proses ini memungkinkan pengelolaan data yang lebih spesifik untuk penduduk yang telah meninggal. *Activity Diagram* dapat dilihat pada Gambar 3.6.

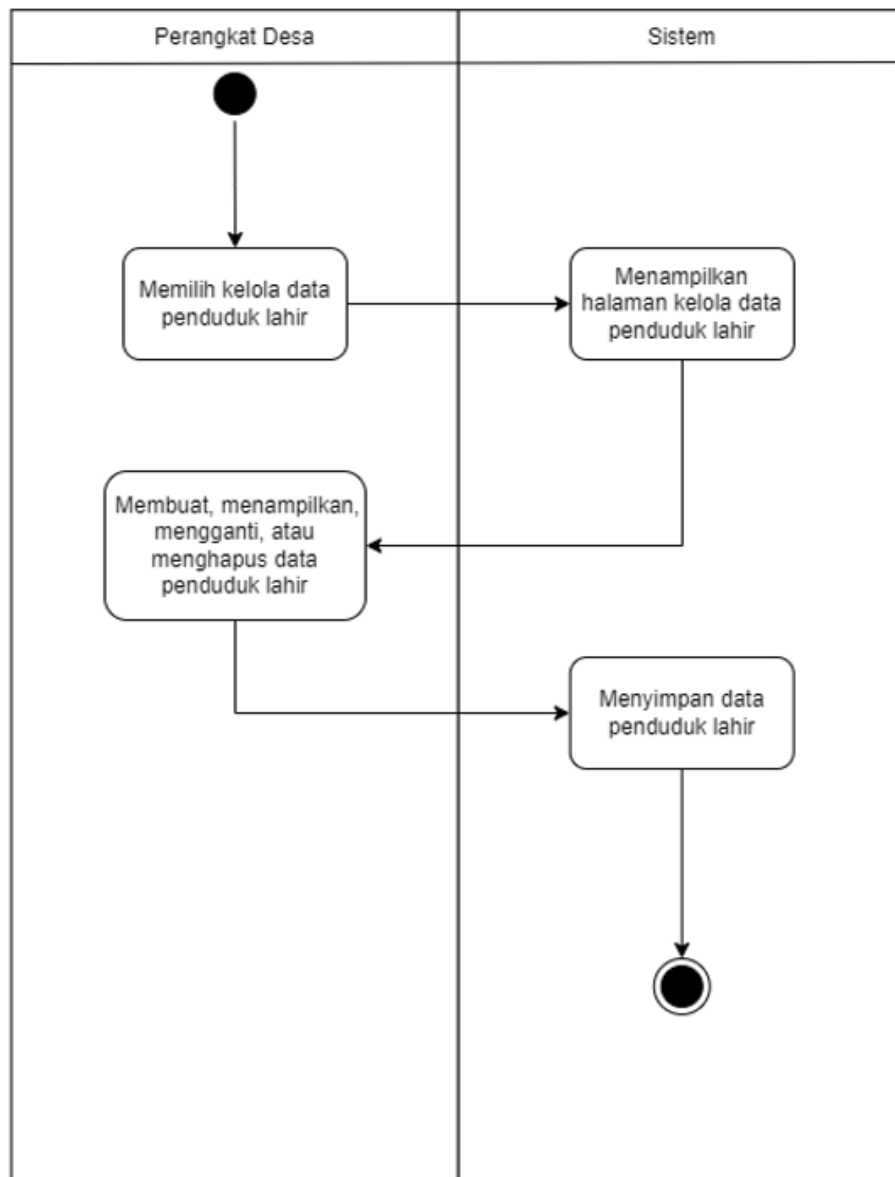


Gambar 3.6 *Activity Diagram* Mengelola Data Penduduk Meninggal oleh Perangkat Desa.

### Mengelola Data Penduduk Lahir

Dalam aktivitas pengelolaan data penduduk yang baru lahir, perangkat desa dapat mengelola atribut yang relevan seperti nama bayi, tanggal lahir, dan data orang tua. Proses ini

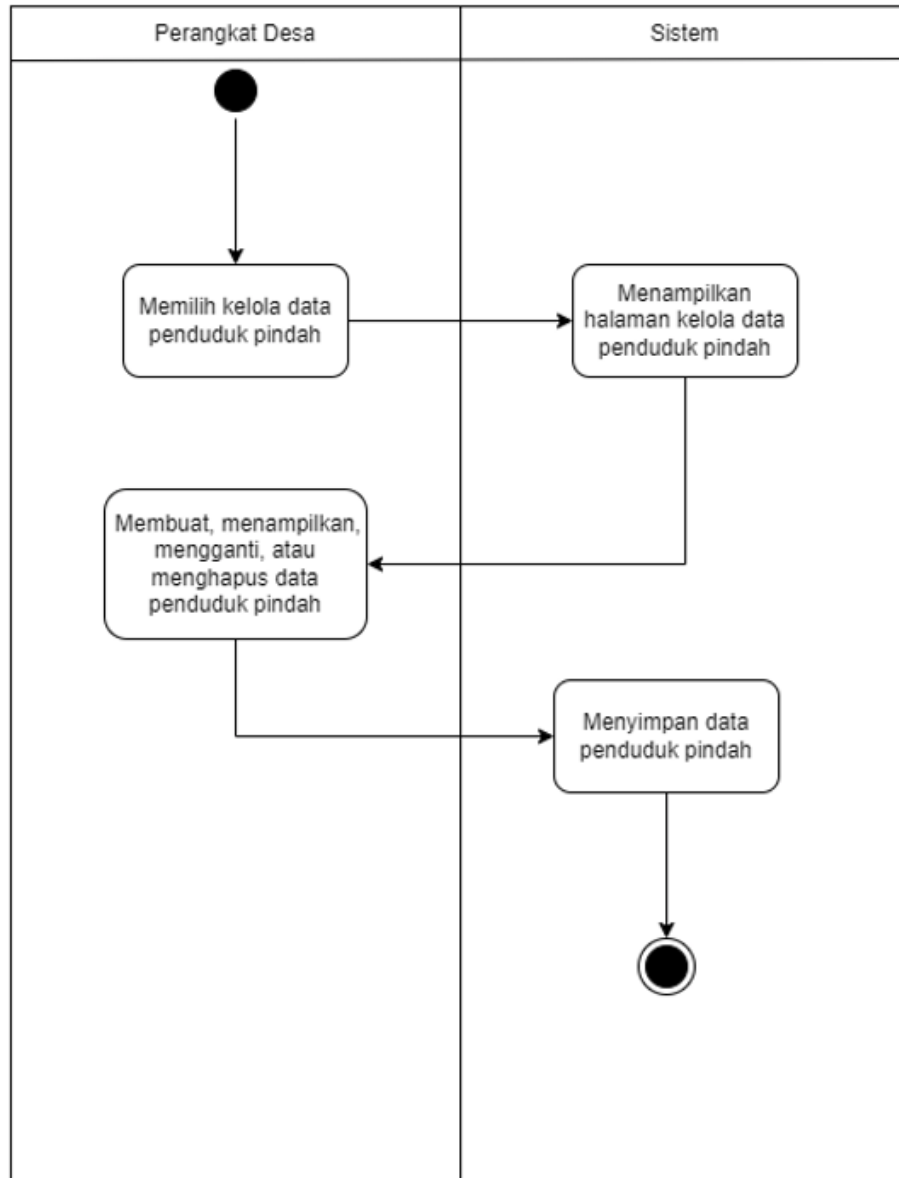
memungkinkan pencatatan kelahiran dan pembaruan data keluarga dengan lebih akurat. *Activity Diagram* dapat dilihat pada Gambar 3.7.



Gambar 3.7 *Activity Diagram* Mengelola Data Penduduk Lahir oleh Perangkat Desa.

### Mengelola Data Penduduk Pindah

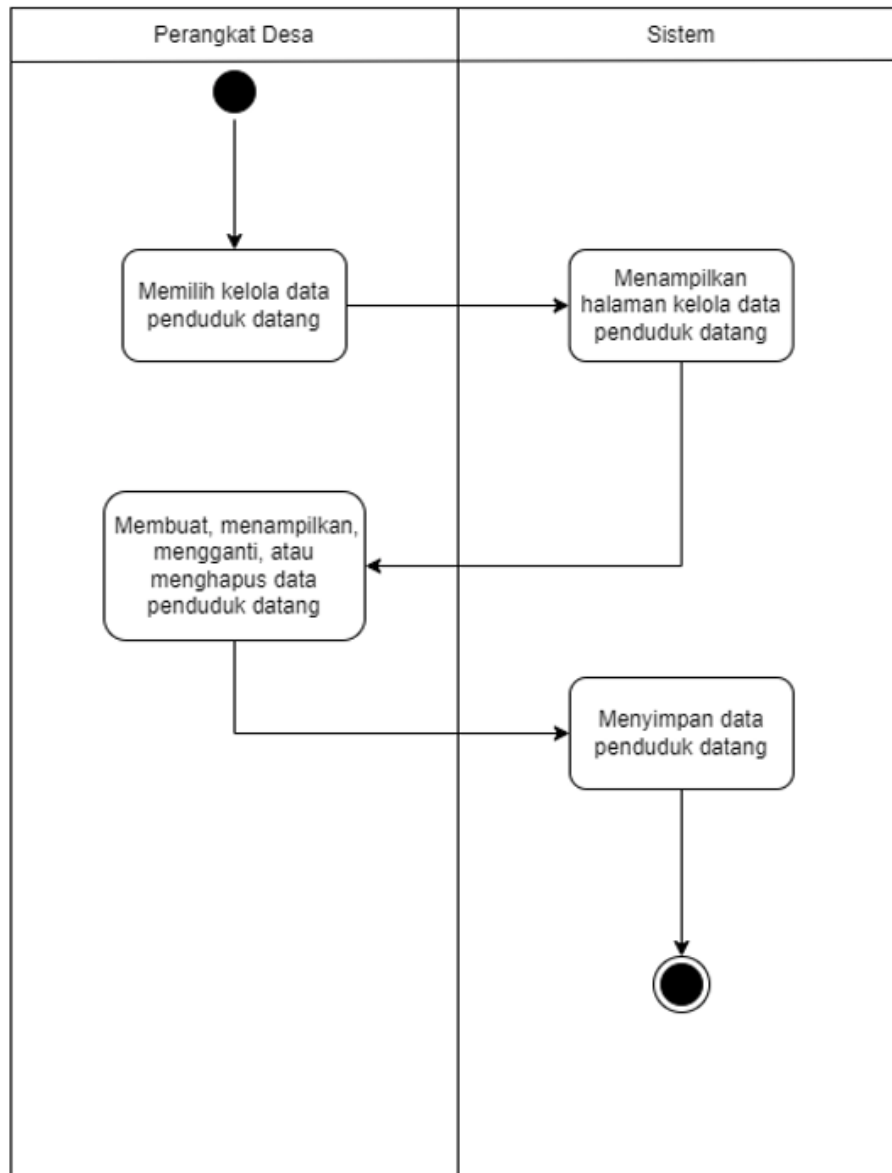
Dalam aktivitas pengelolaan data penduduk yang pindah ke lokasi lain, perangkat desa dapat mengelola proses pembaruan alamat, tanggal pindah, dan detail lainnya. Hal ini bertujuan untuk memastikan bahwa data penduduk yang pindah tercatat dengan akurat di tempat tujuan mereka. *Activity Diagram* dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Activity Diagram* Mengelola Data Penduduk Pindah oleh Perangkat Desa.

### Mengelola Data Penduduk Datang

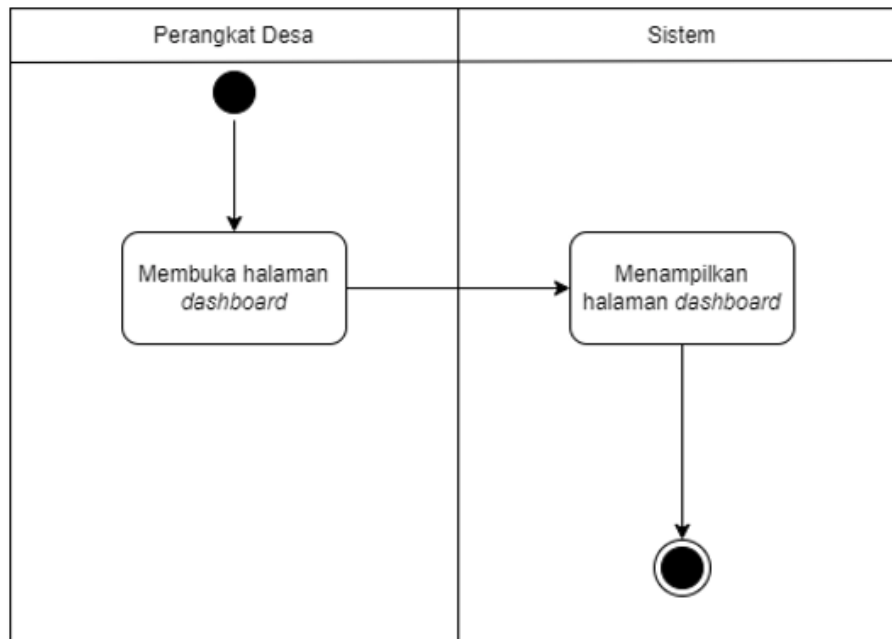
Dalam aktivitas pengelolaan data penduduk yang datang dari wilayah lain, perangkat desa dapat mengidentifikasi dan mengisi atribut seperti asal, tanggal kedatangan, dan informasi lainnya. Proses ini membantu dalam pencatatan data penduduk yang baru datang ke wilayah tersebut. *Activity Diagram* dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Activity Diagram* Mengelola Data Penduduk Datang oleh Perangkat Desa.

### ***Dashboard Perangkat Desa***

Dalam aktivitas melihat ringkasan dan visualisasi data dalam bentuk *dashboard*, perangkat desa dapat dengan mudah memahami dan memonitor informasi penting terkait kependudukan desa. Proses ini memberikan kemudahan bagi perangkat desa dalam mendapatkan gambaran komprehensif tentang statistik dan trend kependudukan secara cepat dan efisien. *Activity Diagram* dapat dilihat pada Gambar 3.10



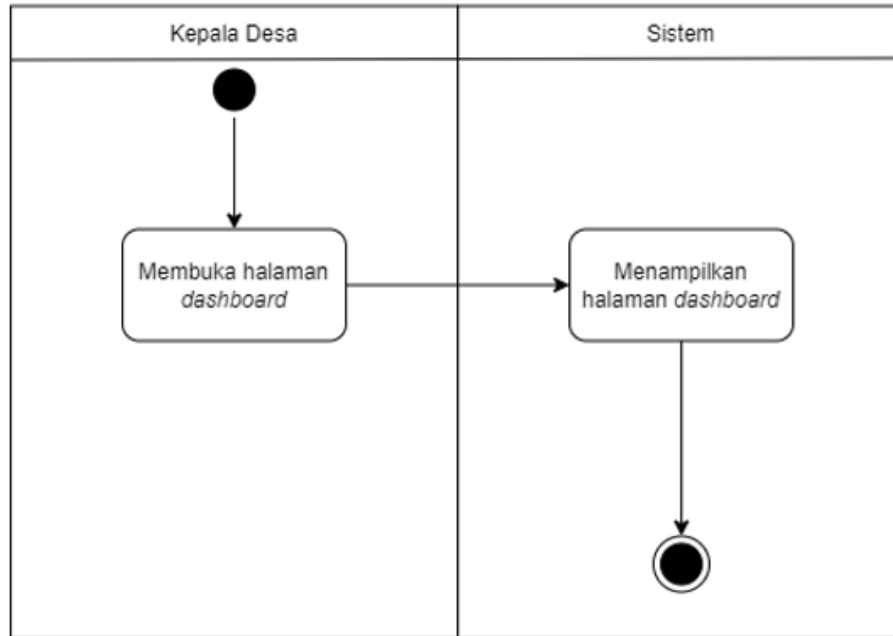
Gambar 3.10 *Activity Diagram Dashboard* oleh Perangkat Desa.

### **3.2.3 *Activity Diagram Kepala Desa***

Penjelasan *activity diagram* menjelaskan alur proses yang dilakukan oleh kepala desa dalam sistem pengelolaan data kependudukan. Diagram ini mengilustrasikan aktivitas yang dilakukan kepala desa untuk memantau data kependudukan yang telah dikelola oleh perangkat desa, serta mengakses laporan dan analisis data.

### ***Dashboard Kepala Desa***

Dalam aktivitas melihat ringkasan dan visualisasi data dalam bentuk *dashboard*, Kepala Desa dapat memperoleh gambaran umum tentang data kependudukan desa. Proses ini memberikan kemudahan bagi Kepala Desa dalam memantau dan memahami informasi terkait kependudukan secara keseluruhan. *Activity Diagram* dapat dilihat pada Gambar 3.11



Gambar 3.11 Activity Diagram *Dashboard* oleh Kepala Desa.

### 3.3 Perancangan Sistem

Perancangan sistem akan menjelaskan dua komponen utama, yaitu antarmuka dan *database*, yang saling terintegrasi untuk membangun sistem pengelolaan data yang optimal. Rancangan antarmuka dan *database* ini dibuat untuk mendukung sistem pengelolaan data yang efisien dan *user-friendly*. Antarmuka pengguna dirancang dengan mempertimbangkan kemudahan navigasi, kejelasan informasi, serta aksesibilitas untuk berbagai peran pengguna. Sementara itu, struktur database dirancang secara terorganisir dan scalable untuk memastikan data dapat diakses dan dikelola secara tepat guna, dengan fokus pada keamanan dan konsistensi data. Implementasi ini mengintegrasikan antarmuka interaktif dengan arsitektur database yang mendukung fungsi CRUD (Create, Read, Update, Delete) serta berbagai logika bisnis yang diperlukan.

#### 3.3.1 Rancangan Antarmuka


Antarmuka sistem ini dirancang dengan pendekatan prototipe menggunakan Figma, untuk menciptakan tampilan yang menarik, intuitif, dan *user-friendly*. Dengan desain yang responsif dan modern, antarmuka ini diharapkan dapat meningkatkan keterlibatan pengguna serta memperkuat efisiensi operasional sistem.

Gambar 3.12 hingga 3.15 akan menjadi desain rancangan antarmuka pada *use case* tabel 1 hingga tabel 3. Desain antarmuka akan dirancang dengan mempertimbangkan kebutuhan

pengguna pada setiap setiap *use case*, memprioritaskan kejelasan informasi dan kemudahan penggunaan untuk meningkatkan efisiensi dalam operasional sehari-hari.

### Mengelola Data Pengguna

Dalam perancangan antarmuka, fokus utama adalah memastikan bahwa admin dapat dengan mudah melakukan pembuatan, penampilan, dan penggantian data pengguna, seperti yang ditunjukkan pada gambar 3.12. 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, serta kemampuan untuk melakukan pembaharuan dan penyesuaian informasi pengguna sesuai kebutuhan. Dengan antarmuka yang dirancang dengan baik, diharapkan admin dapat menjalankan tugasnya dengan efisien dan tanpa kesulitan.



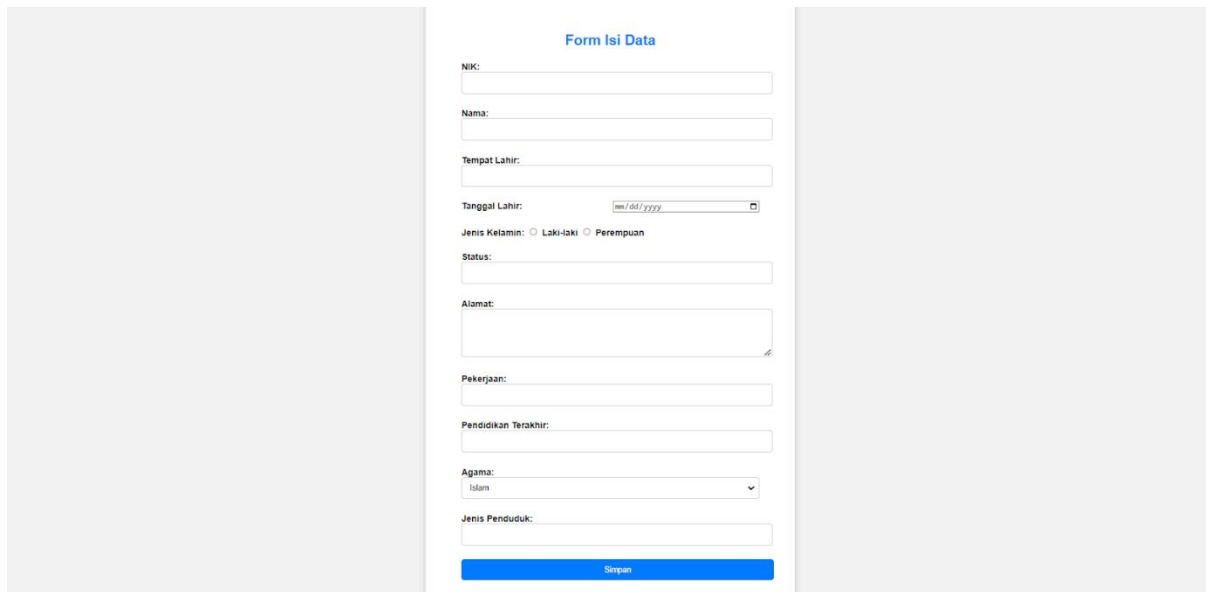
Data Kependudukan Desa Gambasan			
Username	Password	Jabatan	Aksi
Akun 1	Password 1	Kepala Desa	Edit
Akun 2	Password 1	Perangkat Desa	Edit
Akun 3	Password 1	Perangkat Desa	Edit
Akun 4	Password 1	Perangkat Desa	Edit

[Tambah Staf](#)

Gambar 3.12 Rancangan Kelola Halaman Pengguna

### 3.3.1 Mengelola Data Penduduk Tetap

Admin dapat dengan mudah menambah, mengedit, dan menghapus data, termasuk informasi vital seperti nama, alamat, nomor identitas, dan lainnya. Antarmuka pengguna dirancang untuk memberikan admin kontrol penuh terhadap informasi yang terkait dengan penduduk tetap, sehingga memungkinkan proses pengelolaan data berjalan lancar dan efisien, seperti yang ditunjukkan dalam Gambar 3.13.



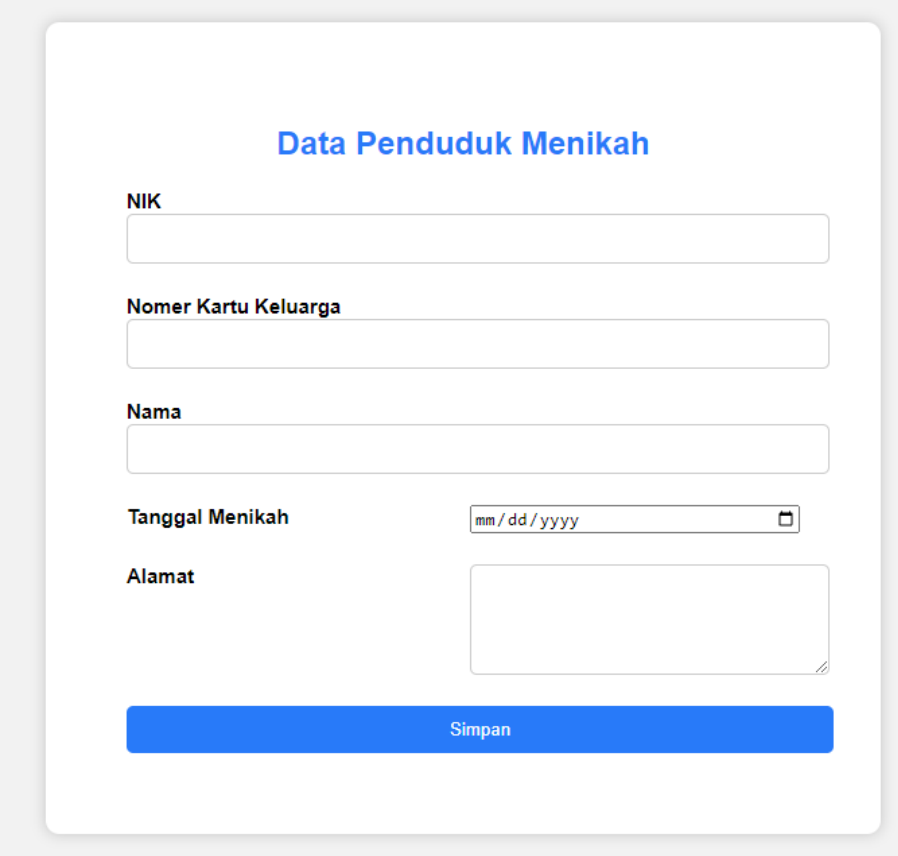
The image shows a web form titled "Form Isi Data" for managing permanent residents. The form contains the following fields and controls:

- NIK: Text input field
- Nama: Text input field
- Tempat Lahir: Text input field
- Tanggal Lahir: Date picker with a calendar icon and a date format "mm/dd/yyyy"
- Jenis Kelamin: Radio buttons for "Laki-laki" and "Perempuan"
- Status: Text input field
- Alamat: Text input field with a small icon at the end
- Pekerjaan: Text input field
- Pendidikan Terakhir: Text input field
- Agama: Dropdown menu with "Islam" selected
- Jenis Penduduk: Text input field
- Simpan: Blue button at the bottom

Gambar 3.13 Rancangan Kelola Halaman Penduduk Tetap

### 3.3.2 Mengelola Data Penduduk Menikah

Antarmuka pada 3.14 gambar dirancang agar perangkat desa dapat dengan mudah mengelola informasi, seperti tanggal pernikahan, nama pasangan, dan atribut tambahan lainnya. Hal ini bertujuan untuk memastikan efisiensi dan akurasi dalam pengelolaan data pernikahan di tingkat desa.



The image shows a web form titled "Data Penduduk Menikah" (Married Resident Data). The form is contained within a light gray rounded rectangle. At the top center, the title "Data Penduduk Menikah" is displayed in blue. Below the title, there are five input fields arranged vertically. The first field is labeled "NIK" and is a single-line text input. The second field is labeled "Nomer Kartu Keluarga" and is a single-line text input. The third field is labeled "Nama" and is a single-line text input. The fourth field is labeled "Tanggal Menikah" and is a date picker with a calendar icon and a placeholder "mm / dd / yyyy". The fifth field is labeled "Alamat" and is a multi-line text area. At the bottom of the form, there is a prominent blue button with the white text "Simpan" (Save).

Gambar 3.14 Rancangan Kelola Halaman Penduduk Menikah.

### 3.3.3 Mengelola Data Penduduk Bercerai

Antarmuka pada gambar 3.15 dirancang untuk memfasilitasi input dan pembaruan informasi, seperti tanggal perceraian, alamat, dan atribut lainnya. Proses ini bertujuan untuk memberikan pengelolaan data yang lebih spesifik terkait penduduk yang telah mengalami perceraian di tingkat desa.

**Data Penduduk Bercerai**

**NIK**

**Nomer Kartu Keluarga**

**Nama**

**Tanggal Bercerai**

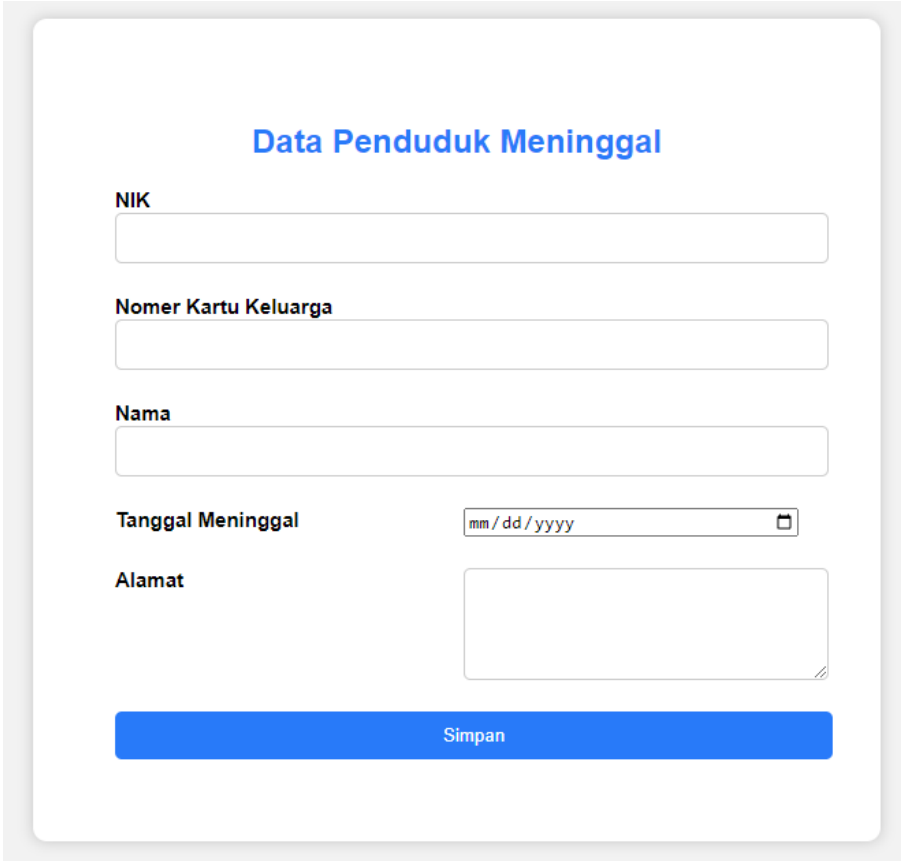
**Alamat**

**Simpan**

Gambar 3.15 Rancangan Kelola Halaman Penduduk Bercerai.

### 3.3.4 Mengelola Data Penduduk Meninggal

Rancangan antarmuka pada gambar 3.16 untuk data penduduk meninggal mengandung informasi pribadi dari warga yang telah meninggal, seperti Nomor Induk Kependudukan (NIK), Nomor Kartu Keluarga (no KK), Nama, dan parameter lainnya. Tujuan utama dari antarmuka ini adalah untuk mempermudah proses pencatatan data penduduk yang telah meninggal.



The image shows a web form titled "Data Penduduk Meninggal" in blue text. The form contains the following fields:

- NIK**: A single-line text input field.
- Nomer Kartu Keluarga**: A single-line text input field.
- Nama**: A single-line text input field.
- Tanggal Meninggal**: A date input field with a placeholder "mm/dd/yyyy" and a calendar icon.
- Alamat**: A multi-line text input field.

At the bottom of the form is a blue button labeled "Simpan".

Gambar 3.16 Rancangan Kelola Halaman Penduduk Meninggal.

### 3.3.5 Mengelola Data Penduduk Lahir

Rancangan antarmuka pada gambar 3.17 untuk data penduduk lahir mencakup informasi pribadi dari bayi yang baru lahir, termasuk Nomor Induk Kependudukan (NIK), Nomor Kartu Keluarga (no KK), Nama, dan atribut lainnya yang relevan. Tujuan dari antarmuka ini adalah untuk memudahkan pencatatan data kelahiran penduduk.

### Data Penduduk Lahir

**NIK**

**Nomer Kartu Keluarga**

**Nama Orang Tua**

**Nama Anak**

**Tempat Lahir**

**Tanggal Lahir**

**Jenis Kelamin**  Laki-laki  Perempuan

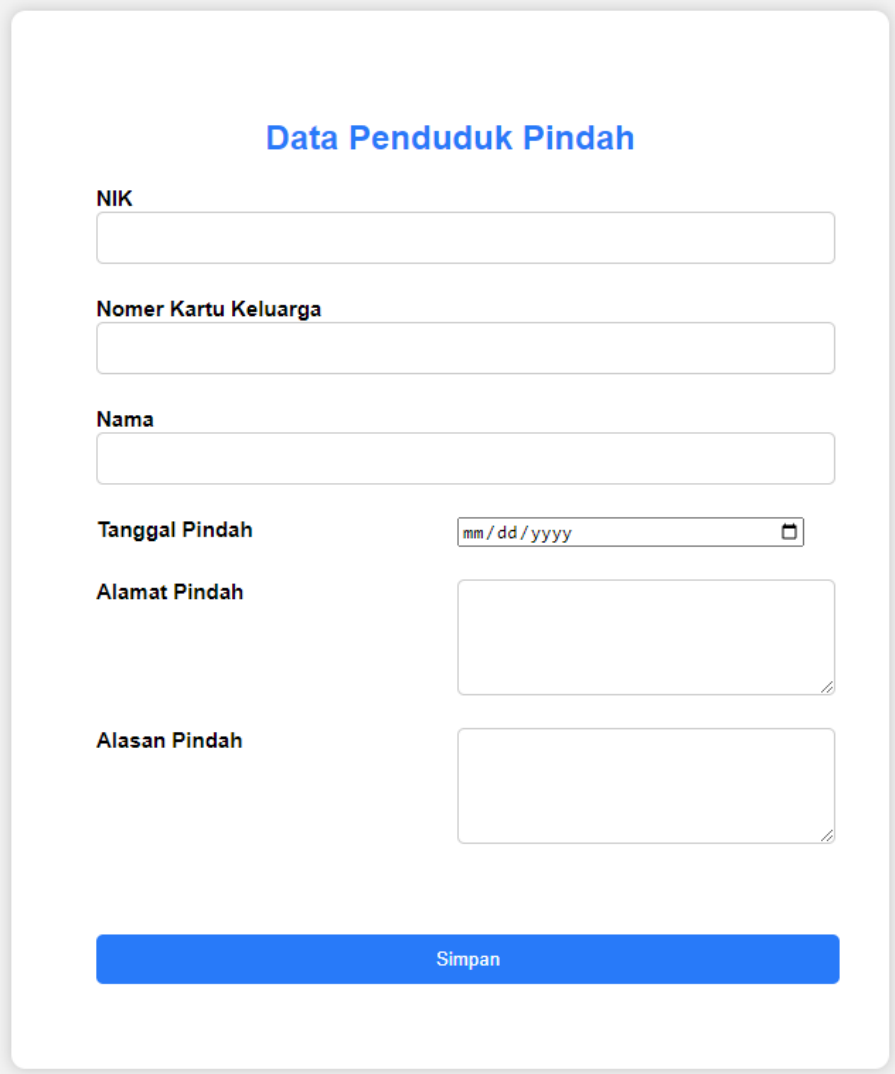
**Alamat**

**Simpan**

Gambar 3. 17 Rancangan Kelola Halaman Penduduk Lahir.

### 3.3.6 Mengelola Data Penduduk Pindah

Rancangan antarmuka pada gambar 3.18 untuk data penduduk pindah mencakup informasi perpindahan penduduk ke lokasi baru, termasuk pembaruan alamat, tanggal pindah, dan data relevan lainnya. Hal ini dirancang untuk menyediakan kemudahan dalam mencatat perpindahan penduduk ke tempat tujuan baru.



**Data Penduduk Pindah**

NIK

Nomer Kartu Keluarga

Nama

Tanggal Pindah

Alamat Pindah

Alasan Pindah

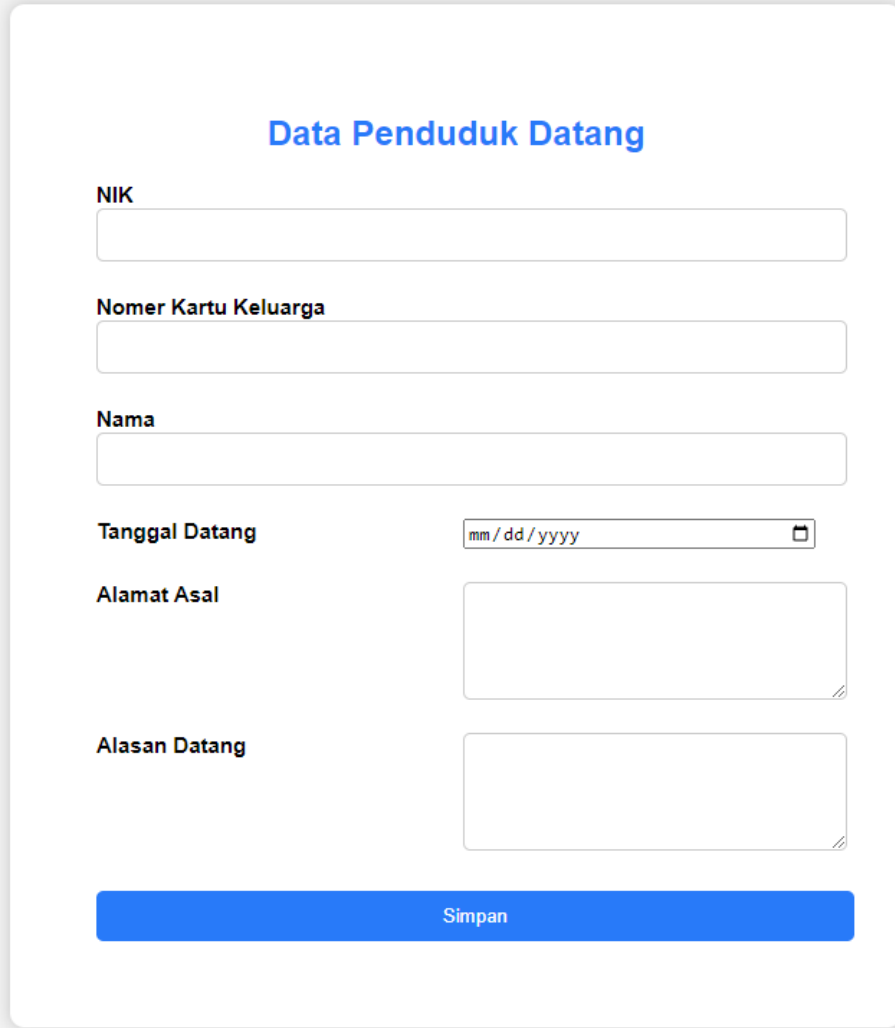
**Simpan**

Gambar 3.18 Rancangan Kelola Halaman Penduduk Pindah.

### 3.3.7 Mengelola Data Penduduk Datang

Rancangan antarmuka pada gambar 3.19 untuk data penduduk datang melibatkan pengelolaan informasi penduduk yang baru datang ke suatu lokasi. Proses ini mencakup pengidentifikasian atribut yang perlu diisi, seperti asal, tanggal kedatangan, dan informasi

lainnya. Tujuannya adalah untuk memfasilitasi pencatatan data penduduk yang baru datang dengan lebih efisien.

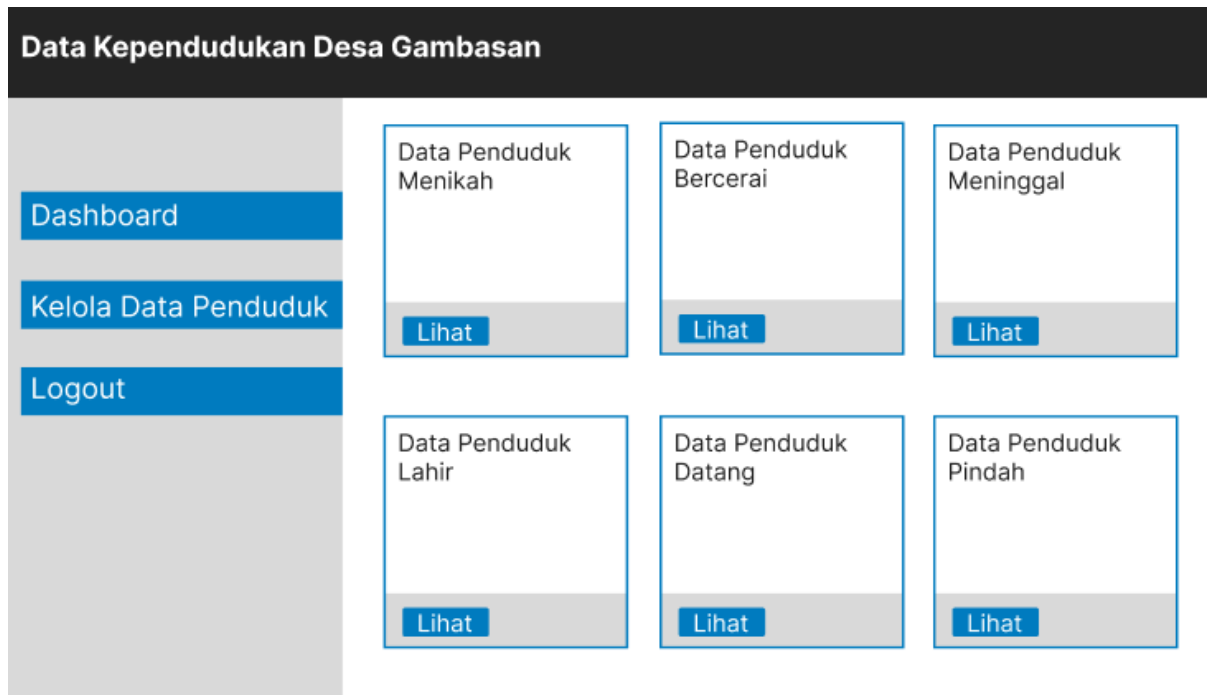


The image shows a web form titled "Data Penduduk Datang" (New Resident Data). The form is contained within a light gray border and has a white background. At the top center, the title "Data Penduduk Datang" is written in a bold, blue font. Below the title, there are six input fields arranged vertically. The first field is labeled "NIK" and is a simple text input. The second field is labeled "Nomer Kartu Keluarga" and is also a simple text input. The third field is labeled "Nama" and is a simple text input. The fourth field is labeled "Tanggal Datang" (Date of Arrival) and is a date picker with a calendar icon on the right; the placeholder text "mm/dd/yyyy" is visible. The fifth field is labeled "Alamat Asal" (Original Address) and is a larger text area with a small diagonal icon in the bottom right corner. The sixth field is labeled "Alasan Datang" (Reason for Arrival) and is also a larger text area with a small diagonal icon in the bottom right corner. At the bottom of the form, there is a prominent blue button with the white text "Simpan" (Save).

Gambar 3.19 Rancangan Kelola Halaman Penduduk Datang.

### 3.3.8 Dashboard Perangkat Desa

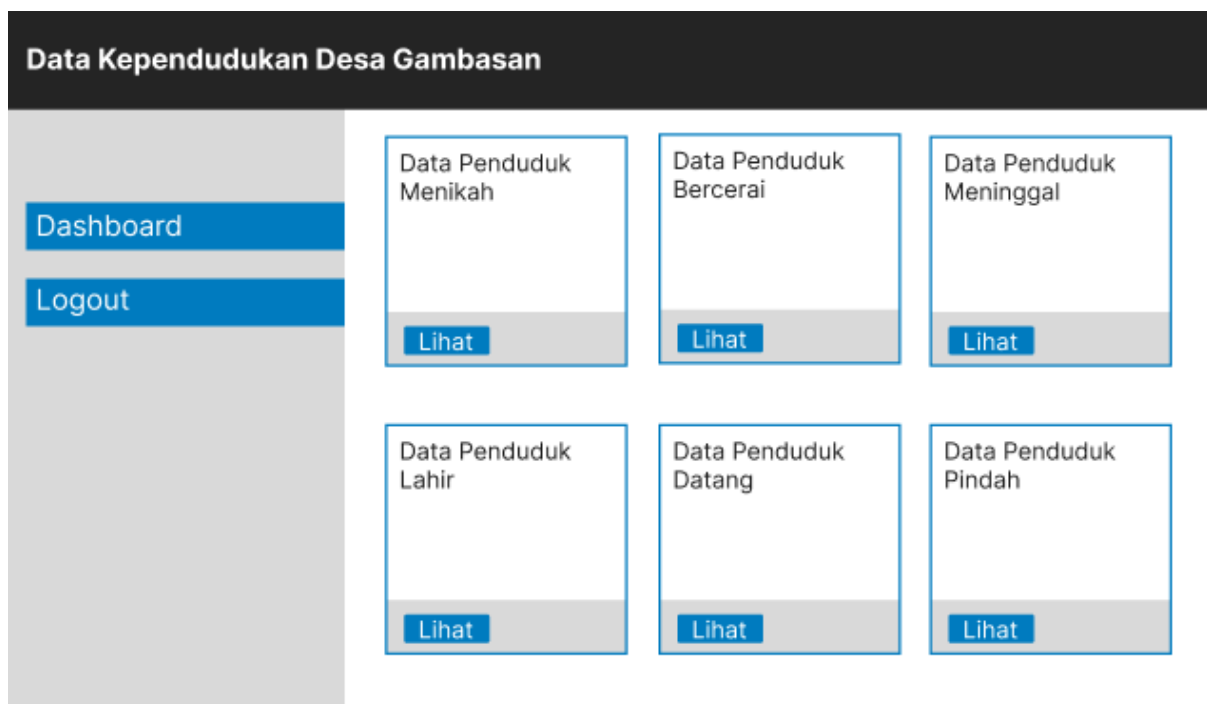
Rancangan antarmuka *dashboard* perangkat desa pada gambar 3.20 disusun untuk memberikan akses kepada perangkat desa untuk melihat ringkasan dan visualisasi data dalam bentuk *dashboard*. Dengan fitur ini, perangkat desa dapat dengan cepat memahami kondisi demografis dan statistik penduduk desa. Proses 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 3.20 Rancangan Halaman *Dashboard* Perangkat Desa.

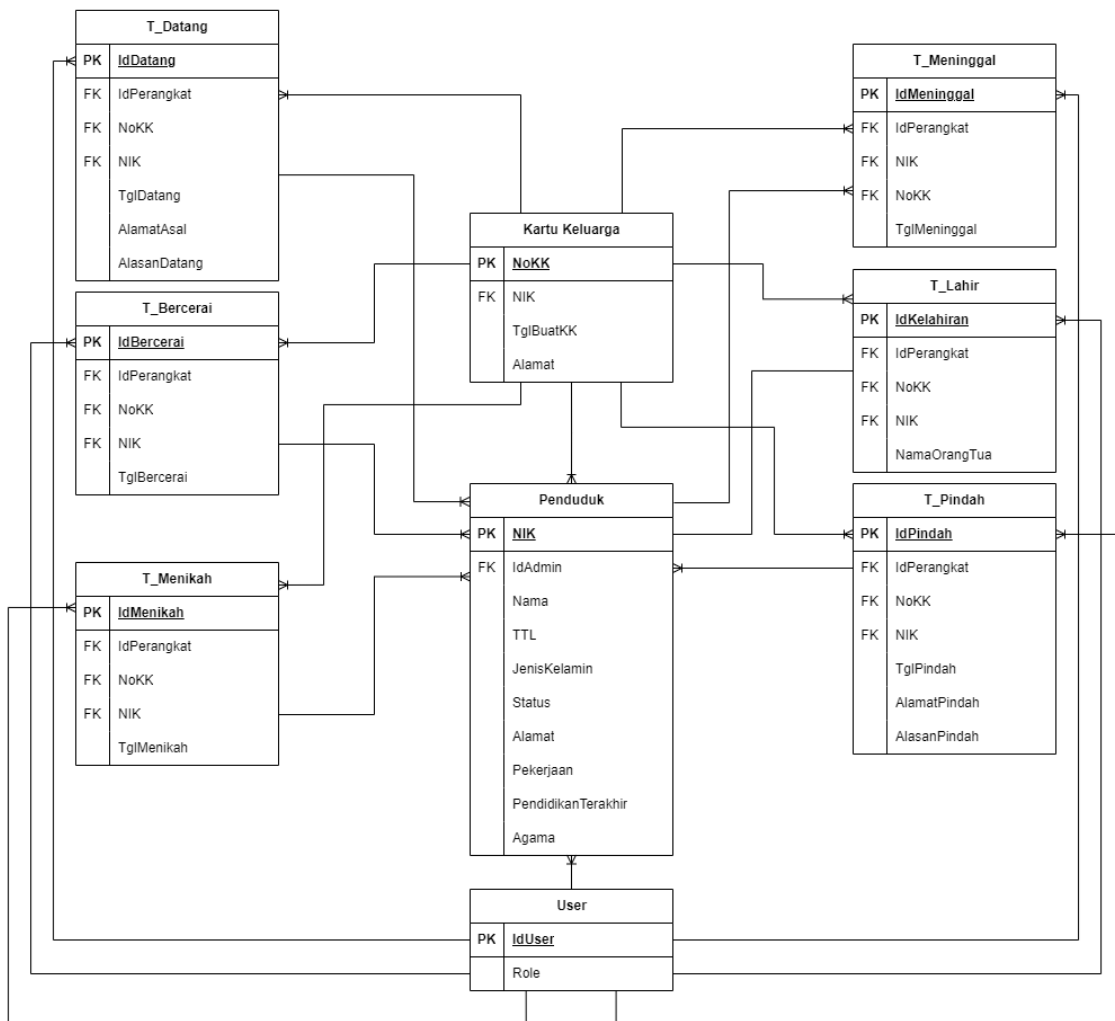
### 3.3.9 Dashboard Kepala Desa

Rancangan antarmuka *dashboard* Kepala Desa pada gambar 3.21 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.



Gambar 3.21 Rancangan Halaman *Dashboard* Kepala Desa.

### 3.3.2 Rancangan Basis Data



Gambar 3.22 Rancangan Basis Data

Gambar 3.22 merupakan rancangan pengelolaan data kependudukan desa, terdapat tiga “Role” utama, yaitu admin, perangkat desa, dan kepala desa. Tiga “Role” dalam “User” memiliki peran yang berbeda dalam mengakses dan mengelola informasi. Admin memiliki akses penuh terhadap sistem dan bertanggung jawab untuk membuat, menampilkan, dan mengganti data pengguna. Selain itu, admin dapat mengelola data penduduk untuk penyesuaian informasi pengguna yang terdaftar. Fungsionalitas ini memastikan bahwa pengelolaan akses ke sistem tetap terjaga dan terkendali.

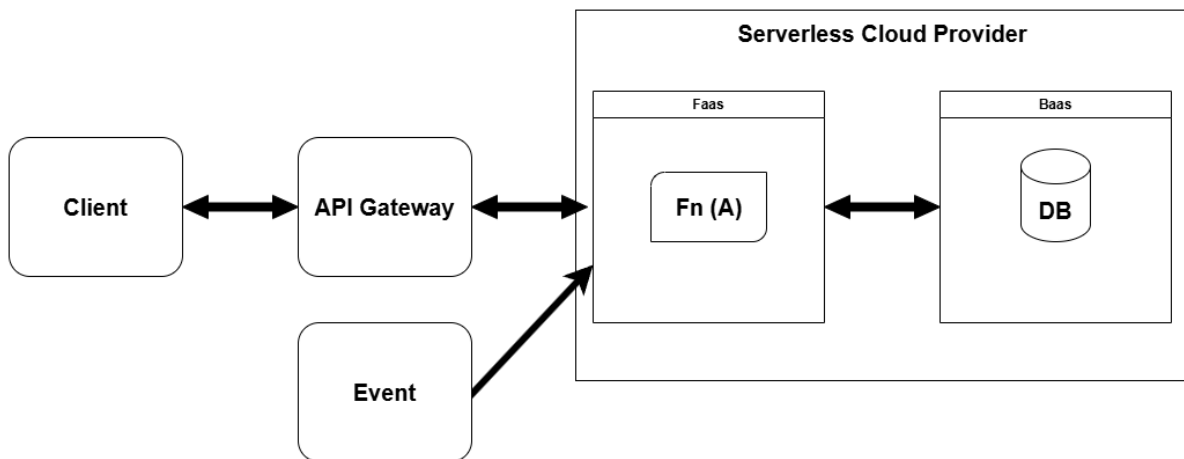
Perangkat desa, sebagai pengguna yang terlibat langsung dalam kegiatan administratif desa, memiliki kemampuan untuk mengelola data penduduk dengan fokus pada aktivitas yang berhubungan dengan peristiwa kependudukan. Dalam hal ini, perangkat desa dapat mengelola data penduduk yang menikah, bercerai, meninggal, pindah, datang, atau lahir. Hal ini

memungkinkan perangkat desa untuk menjalankan tugas administratif terkait perubahan status kependudukan di tingkat desa.

Untuk penyimpanan data pada aplikasi ini, peneliti akan mengintegrasikan aplikasi dengan basis data Firebase. Dengan menggunakan Firebase, semua data pengguna dapat disinkronisasikan kepada perangkat desa lain maupun kepala desa dan dapat diakses secara realtime. Dengan menggunakan Firebase, peneliti tidak perlu lagi menggunakan model skema basis data yang berelasi karena menggunakan konsep NoSQL. Firebase menggunakan skema model tree dimana database akan menggunakan key value yang unik sehingga dapat di akses secara realtime.

### 3.3.3 Rancangan Penerapan Serverless Computing

Berikut adalah rancangan sistem informasi kependudukan desa yang menerapkan teknologi *serverless computing*. Arsitektur ini dirancang untuk meningkatkan efisiensi pengelolaan data kependudukan, meminimalkan kebutuhan infrastruktur, dan mendukung pengalaman pengguna yang lebih baik. Gambar 3.23 menjelaskan bagaimana berbagai komponen sistem, seperti Client, API Gateway, BaaS, dan FaaS, saling berinteraksi untuk memastikan pengelolaan data yang efektif dan aman menggunakan teknologi *serverless computing*.



Gambar 3.23 Arsitektur *Serverless Computing*.

Rancangan sistem informasi kependudukan desa dengan penerapan *serverless computing* memanfaatkan dua komponen utama, yaitu Backend-as-a-Service (BaaS) dan Function-as-a-Service (FaaS). Firebase Realtime Database digunakan sebagai BaaS untuk menyimpan dan mengelola data pengguna serta data kependudukan secara real-time, menggunakan struktur data berbasis *tree* yang memungkinkan sinkronisasi cepat antar perangkat tanpa memerlukan skema basis data relasional. Hal ini memungkinkan pengelolaan data yang efisien dan mudah

diakses oleh berbagai peran pengguna, seperti admin, perangkat desa, dan kepala desa, dengan hak akses yang terstruktur sesuai dengan peran masing-masing.

Pada sistem ini, penggunaan Faas ketika admin berhasil mendaftar akun pengguna, maka Firebase Authentication akan memicu *event* *onCreate* yang otomatis menjalankan Cloud Function. Penggunaan Faas juga Fungsi ini kemudian memproses data pengguna baru dan menggunakan layanan SMTP untuk mengirimkan email sambutan kepada pengguna. Email ini berisi pesan selamat datang serta informasi penting terkait akun pengguna, yang mempercepat proses pengenalan pengguna baru terhadap sistem. Selain itu, untuk proses penghapusan akun, fungsi lain akan menangani penghapusan data pengguna dari database dan Firebase Authentication, serta mengirim email konfirmasi kepada pengguna.

Sistem ini mengadopsi arsitektur *serverless* yang menghilangkan kebutuhan untuk pengelolaan infrastruktur server, memungkinkan Firebase untuk menangani pengelolaan secara otomatis. Hal ini memastikan bahwa aplikasi dapat menangani lonjakan jumlah pengguna dan permintaan tanpa penurunan performa. Firebase Security Rules juga diterapkan untuk mengamankan data dan membatasi akses sesuai dengan peran pengguna. Dengan demikian, pengelolaan data dilakukan dengan aman, dan hanya pengguna yang berhak yang dapat mengakses data informasi.

### **3.4 Pengujian Sistem**

Pengujian sistem ini dilakukan untuk memastikan keamanan dan kinerja aplikasi yang menggunakan Firebase Realtime Database dan Cloud Functions. Tujuan utama pengujian keamanan adalah mengevaluasi kontrol akses terhadap data, memastikan bahwa hanya pengguna yang memiliki hak akses dapat melihat dan mengubah informasi sensitif. Pengujian dilakukan menggunakan Firebase Security Rules Playground dengan berbagai skenario, mulai dari aturan yang sepenuhnya membatasi akses hingga yang hanya mengizinkan akses bagi pengguna yang telah terautentikasi. Sementara itu, pengujian kinerja difokuskan pada pemantauan waktu eksekusi dan penggunaan memori dari fungsi *cloud*, seperti dalam proses pengiriman email saat penambahan akun pengguna dan penghapusan akun pengguna, untuk memastikan sistem dapat menangani beban kerja dengan efisien.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi

Proses ini melibatkan penerapan rancangan yang telah direncanakan, dengan setiap fungsi dikembangkan menggunakan teknologi *serverless*. Tahapan ini mencakup pembangunan berbagai komponen sistem, pengujian unit untuk memastikan setiap fungsi berjalan dengan baik, serta pemanfaatan layanan *serverless* dalam pengelolaan data dan fungsi berbasis pemicu (*event-triggered functions*). Dengan menggunakan *serverless computing*, pengembang dapat mengotomatiskan integrasi antar modul tanpa harus mengelola infrastruktur server secara manual.

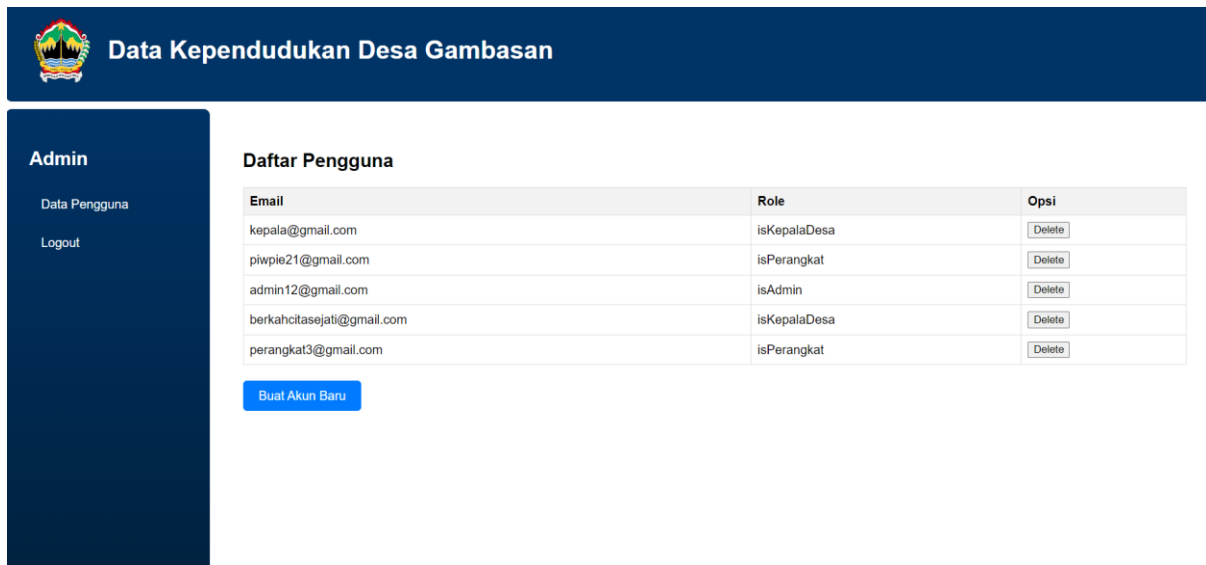
Pada arsitektur *serverless* yang menggunakan *cloud functions* dan Firebase Realtime Database, sistem kependudukan desa dapat dirancang untuk memenuhi kebutuhan otomatisasi serta penyimpanan data secara real-time. Saat admin menambah pengguna baru, *cloud function* secara otomatis memproses penambahan melalui Firebase Authentication yang tersinkroniasi dengan Realtime Database dan memicu pengiriman email selamat datang (*welcome email*) kepada pengguna tersebut, memberi notifikasi bahwa mereka telah bergabung dalam sistem. Sebaliknya, ketika admin menghapus pengguna, *cloud function* juga mengirim email perpisahan (*bye email*) secara otomatis, menyampaikan bahwa akun pengguna telah dihapus dari sistem. Automatisasi ini membuat pengelolaan pengguna lebih cepat dan efisien tanpa memerlukan pemrosesan manual dari pihak admin.

Selain itu, Firebase Realtime Database memainkan peran utama dalam menyimpan dan memperbarui data penduduk desa. Ketika perangkat desa menambahkan, memperbarui, atau menghapus data penduduk, Firebase Realtime Database memperbarui informasi tersebut secara langsung untuk semua pengguna yang terhubung, termasuk kepala desa yang memiliki hak akses. Hal ini memungkinkan perubahan data dapat dilihat secara instan oleh pengguna lain yang berwenang, tanpa menunggu sinkronisasi. Arsitektur ini memberikan respons cepat dan otomatis pada setiap interaksi, mendukung pengelolaan data kependudukan desa dengan efisien dan akurat tanpa beban pengelolaan server tambahan.

##### 4.1.1 Halaman Antar Muka Admin

Implementasi antarmuka halaman admin pada gambar 4.1 dan 4.2 bertujuan untuk menyediakan antarmuka yang mudah digunakan bagi administrator dalam mengelola data dan fungsi penting dari sistem. Antarmuka ini dirancang dengan fokus pada kemudahan navigasi,

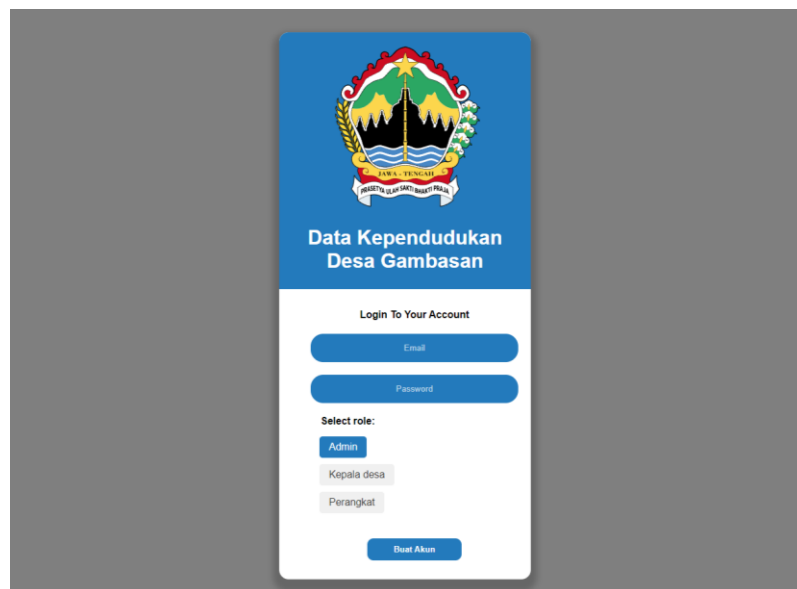
akses cepat ke berbagai fitur, dan tampilan data yang informatif, sehingga administrator dapat menjalankan tugas-tugas seperti manajemen pengguna, pemantauan aktivitas, serta pengelolaan data kependudukan dengan efisien. Melalui implementasi ini, diharapkan admin dapat lebih responsif dalam menjalankan tanggung jawabnya, memastikan kelancaran operasional sistem, dan meningkatkan kualitas layanan bagi pengguna lainnya.



Email	Role	Opsi
kepala@gmail.com	isKepalaDesa	Delete
piwpie21@gmail.com	isPerangkat	Delete
admin12@gmail.com	isAdmin	Delete
berkahcitasajat@gmail.com	isKepalaDesa	Delete
perangkat3@gmail.com	isPerangkat	Delete

Buat Akun Baru

Gambar 4.1 Halaman *Dashboard* Admin.



Data Kependudukan  
Desa Gambasan

Login To Your Account

Email

Password

Select role:

Admin

Kepala desa

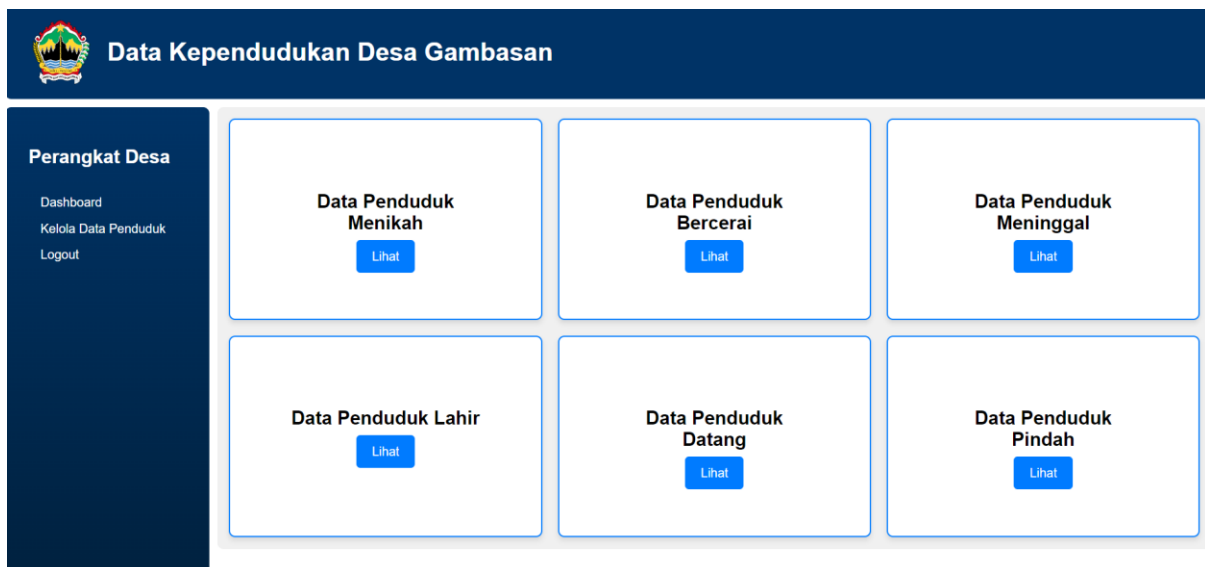
Perangkat

Buat Akun

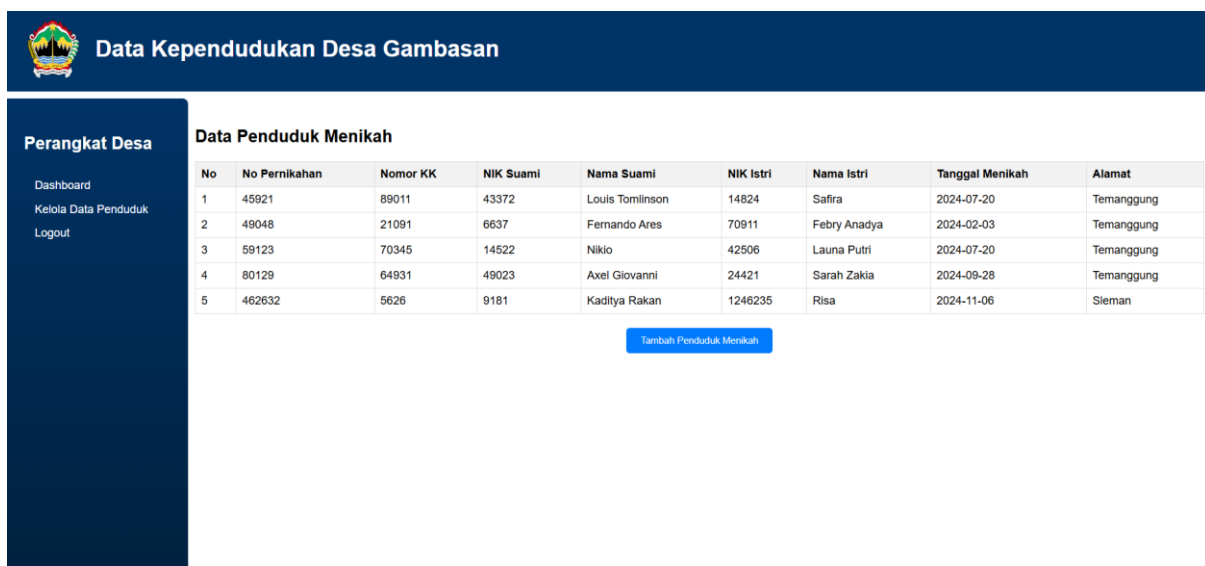
Gambar 4.2 Halaman Tambah Akun Admin.

#### 4.1.2 Halaman Antar Muka Perangkat Desa

Implementasi antarmuka perangkat desa pada gambar 4.3 sampai dengan bertujuan untuk memfasilitasi pengelolaan data kependudukan secara efisien. Antarmuka ini dirancang dengan mempertimbangkan kebutuhan perangkat desa dalam mengakses dan memanipulasi informasi terkait peristiwa kependudukan, seperti pencatatan kelahiran, kematian, dan pernikahan. Dengan fitur yang intuitif dan responsif, perangkat desa dapat melakukan tugasnya dengan lebih mudah, memastikan bahwa semua data yang diperlukan tercatat secara akurat dan *real-time*.



Gambar 4.3 Halaman *Dashboard* Perangkat Desa.



Gambar 4.4 Halaman Menikah Perangkat Desa.

**Data Penduduk Menikah**

Nomor Pernikahan

Nomor KK

NIK Suami

NIK Istri

Tanggal Menikah

Alamat

**Simpan**

Gambar 4.5 Halaman *Form* Menikah Perangkat Desa.

**Data Kependudukan Desa Gambasan**

**Perangkat Desa**

- Dashboard
- Kelola Data Penduduk
- Logout

**Data Penduduk Bercerai**

No	No Nikah	NIK Suami	Nama Suami	NIK Istri	Nama Istri	Tanggal Bercerai	Opsi
1	123	585867	Ozik	5466234	Maya	2024-11-13	<a href="#">Hapus</a>
2	532523	14522	Nikio	235253	Niken	2024-11-28	<a href="#">Hapus</a>
3	12745	71865	Niall Horan	57900	Maudy Putri	2024-11-02	<a href="#">Hapus</a>

**Tambah Penduduk Bercerai**

Gambar 4.6 Halaman Bercerai Perangkat Desa.

**Data Penduduk Bercerai**

**Nomor Pernikahan**

**Tanggal Bercerai**

**Simpan**

Gambar 4.7 Halaman *Form Bercerai* Perangkat Desa.

**Data Kependudukan Desa Gambasan**

**Perangkat Desa**

- Dashboard
- Kelola Data Penduduk
- Logout

**Data Penduduk Meninggal**

No	Nomor Kematian	NIK	Nama	Tanggal Meninggal	Opsi
1	123	25523	Cinta	2024-11-13	<a href="#">Delete</a>
2	551	78934	Nayla	2023-12-09	<a href="#">Delete</a>
3	762	64281	Liam Payne	2024-10-28	<a href="#">Delete</a>
4	764554	389050350	Rusael	2024-11-20	<a href="#">Delete</a>
5	981238912	9181	Kaditya Rakan	2024-11-20	<a href="#">Delete</a>

[Tambah Data Meninggal](#)

Gambar 4.8 Halaman *Meninggal* Perangkat Desa.

**Data Penduduk Meninggal**

**Nomor Kematian**

**NIK**

**Tanggal Meninggal**  📅

**Simpan**

Gambar 4.9 Halaman *Form* Meninggal Perangkat Desa.

**Data Kependudukan Desa Gambasan**

**Perangkat Desa**

- Dashboard
- Kelola Data Penduduk
- Logout

**Data Kelahiran**

No	NIK	Nama Anak	No KK	Nama Orang Tua	Tempat Lahir	Tanggal Lahir	Jenis Kelamin	Alamat	Opsi
1	31515	Nanda	234	Ozik	Jogja	2024-11-21	laki-laki	Jogja	Delete
2	8312838	Vitalik	234	Ozek	Bantul	2024-11-12	laki-laki	Bantul	Delete
3	34500	Chloe Li	89011	Louis Tomlinson	Temanggung	2024-11-01	laki-laki	Temanggung	Delete
4	50125	Anissa	64931	Axel Giovanni	Temanggung	2024-11-03	perempuan	Temanggung	Delete

**Tambah Penduduk**

Gambar 4.10 Halaman Kelahiran Perangkat Desa.

**Data Penduduk Lahir**

NIK

Nama Anak

Nomer Kartu Keluarga

Nama Orang Tua

Tempat Lahir

Tanggal Lahir

Jenis Kelamin  Laki-laki  Perempuan

Alamat

Agama

**Simpan**

Gambar 4.11 Halaman *Form* Lahir Perangkat Desa.

**Data Kependudukan Desa Gambasan**

**Perangkat Desa**

- Dashboard
- Kelola Data Penduduk
- Logout

**Data Penduduk Datang**

No	NIK	Nama	Tanggal Datang	Alamat Asal	Alasan Datang	Opsi
1	514512	Jesse	2024-11-05	Jerman	Pindah Negara	<input type="button" value="Delete"/>
2	905690	Johnson	2024-11-18	Malang	Menikah	<input type="button" value="Delete"/>
3	14824	Safira	2024-11-01	Kebumen	Kerja	<input type="button" value="Delete"/>

Gambar 4.12 Halaman Datang Perangkat Desa.

**Data Penduduk Datang**

NIK

Nama

Tempat Lahir

Tanggal Lahir

Jenis Kelamin  Laki-laki  Perempuan

Tanggal Datang

Alamat Asal

Alasan Datang

Agama

[Simpan](#)

Gambar 4.13 Halaman *Form* Datang Perangkat Desa.

**Data Kependudukan Desa Gambasan**

**Perangkat Desa**

- Dashboard
- Kelola Data Penduduk
- Logout

**Data Penduduk Pindah**

No	NIK	Nama	Tanggal Pindah	Alamat Pindah	Alasan Pindah	Opsi
1	95694	Hanif	2024-11-21	Banyuwangi	Pulang Kampung	<a href="#">Delete</a>
2	78563	Amalia Putri	2024-04-23	Jakarta	Kerja	<a href="#">Delete</a>

[Tambah Data Pindah](#)

Gambar 4. 14 Halaman Pindah Perangkat Desa.

Gambar 4.15 Halaman *Form* Pindah Perangkat Desa.

No	NIK	Nama	Tempat Lahir	Tanggal Lahir	Jenis Kelamin	Status	Alamat	Pekerjaan	Pendidikan Terakhir	Agama	Opsi
1	6637	Fernando Ares	Jember	1998-05-09	laki-laki	menikah	Temanggung	Dokter	sarjana	kristen	Edit Delete
2	11288	Sandy Kristian	Cianjur	2003-08-17	laki-laki	lajang	Temanggung	Karyawan Swasta	sarjana	hindu	Edit Delete
3	14522	Nikio	Mgl	2024-11-07	laki-laki	menikah	Mgl	Dokter	sarjana	islam	Edit Delete
4	14824	Safira	Kebumen	2000-03-04	perempuan	menikah	Temanggung	Apoteker	sarjana	kristen	Edit Delete
5	23769	Zayn Malik	Jakarta	1992-01-12	laki-laki	lajang	Temanggung	Buruh Pabrik	sma	katolik	Edit Delete
6	24421	Sarah Zakia	Temanggung	1995-12-28	perempuan	menikah	Temanggung	Karyawan Swasta	sarjana	katolik	Edit Delete
7	42506	Launa Putri	Makassar	2000-01-04	perempuan	menikah	Temanggung	Karyawan Swasta	sarjana	hindu	Edit Delete
8	43372	Louis Tomlinson	Tangerang	1991-12-24	laki-laki	menikah	Temanggung	Guru SD	sarjana	kristen	Edit Delete

Gambar 4.16 Halaman Penduduk Perangkat Desa.

**Form Isi Data**

NIK:

Nama:

Tempat Lahir:

Tanggal Lahir:

Jenis Kelamin:  Laki-laki  Perempuan

Status:

Alamat:

Pekerjaan:

Pendidikan Terakhir:

Agama:

[Simpan](#)

Gambar 4.17 Halaman *Form* Penduduk Perangkat Desa.

**Data Kependudukan Desa Gambasan**

No	NO KK	Anggota Keluarga	Tanggal Buat KK	Alamat	Opsi
1	234	585867-Ozik, 5466234-Maya, 31515-Nanda, 8312838-Vitalik	2024-11-20	Bantul	<a href="#">Edit</a> <a href="#">Delete</a>
2	4545	14522-Nikio, 235253-Niken	undefined	Mgi	<a href="#">Edit</a> <a href="#">Delete</a>
3	5626	9181-Kaditya Rakan, 1246235-Risa	undefined	Sleman	<a href="#">Edit</a> <a href="#">Delete</a>
4	21091	6637-Fernando Ares, 70911-Febry Anadya	undefined	Temanggung	<a href="#">Edit</a> <a href="#">Delete</a>
5	48361	71865-Niall Horan, 57900-Maudy Putri	undefined	Temanggung	<a href="#">Edit</a> <a href="#">Delete</a>
6	64931	49023-Axel Giovanni, 24421-Sarah Zakia, 50125-Anissa	undefined	Temanggung	<a href="#">Edit</a> <a href="#">Delete</a>
7	70345	14522-Nikio, 42506-Launa Putri	undefined	Temanggung	<a href="#">Edit</a> <a href="#">Delete</a>
8	89011	43372-Louis Tomlinson, 14824-Safira, 34500-Chloe Li	undefined	Temanggung	<a href="#">Edit</a> <a href="#">Delete</a>

[Buat KK](#)

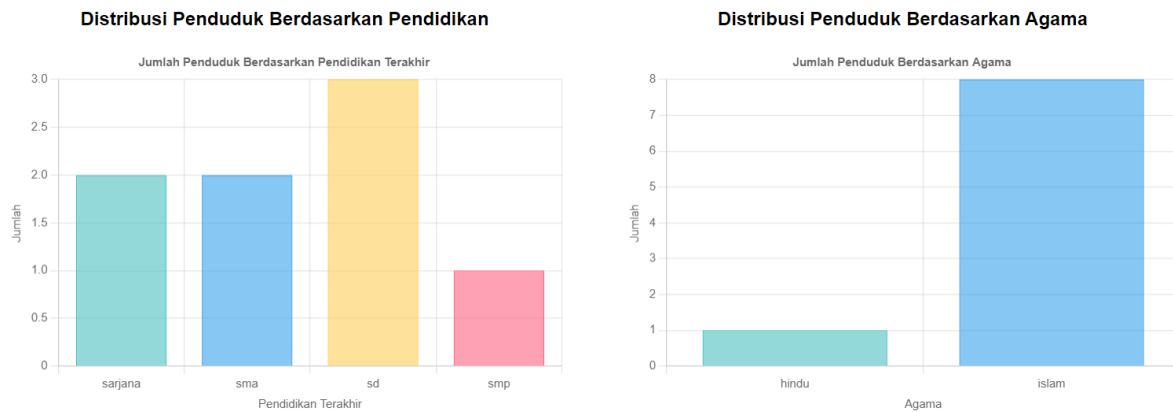
Gambar 4.18 Halaman Kartu Keluarga Perangkat Desa.

Gambar 4.19 Halaman *Form* Kartu Keluarga Perangkat Desa.

### 4.1.3 Halaman Antar Muka Kepala Desa

Implementasi Antarmuka untuk kepala desa pada gambar 4.20 dan 4.21 dirancang dengan tujuan memberikan akses yang informatif dan terpusat untuk memantau data kependudukan yang dikelola oleh perangkat desa. Berbeda dengan perangkat desa yang berfokus pada pengelolaan data langsung, kepala desa memiliki fungsi lebih administratif dan strategis.

Data Kependudukan Desa Gambasan										
Kepala Desa	Data Penduduk									
No	NIK	Nama	Tempat Lahir	Tanggal Lahir	Jenis Kelamin	Status	Alamat	Pekerjaan	Pendidikan Terakhir	Agama
1	6637	Fernando Ares	Jember	1998-05-09	laki-laki	menikah	Temanggung	Dokter	sarjana	kristen
2	11288	Sandy Kristian	Cianjur	2003-08-17	laki-laki	lajang	Temanggung	Karyawan Swasta	sarjana	hindu
3	14522	Niklo	Mgl	2024-11-07	laki-laki	menikah	Mgl	Dokter	sarjana	islam
4	14824	Safira	Kebumen	2000-03-04	perempuan	menikah	Temanggung	Apoteker	sarjana	kristen
5	23769	Zayn Malik	Jakarta	1992-01-12	laki-laki	lajang	Temanggung	Buruh Pabrik	sma	katolik
6	24421	Sarah Zakia	Temanggung	1995-12-28	perempuan	menikah	Temanggung	Karyawan Swasta	sarjana	katolik
7	42506	Launa Putri	Makassar	2000-01-04	perempuan	menikah	Temanggung	Karyawan Swasta	sarjana	hindu
8	43372	Louis Tomlinson	Tangerang	1991-12-24	laki-laki	menikah	Temanggung	Guru SD	sarjana	kristen
9	49023	Axel Giovanni	Surakarta	2003-03-05	laki-laki	menikah	Temanggung	Karyawan Swasta	sarjana	katolik
10	50125	Anissa	Temanggung	2024-11-03	perempuan	undefined	Temanggung	undefined	undefined	budha
11	57900	Maudy Putri	Ambarawa	1990-01-31	perempuan	lajang	Semarang	Ibu Rumah Tangga	sarjana	kristen
12	66611	Baskara Hindia	Semarang	1990-08-14	laki-laki	lajang	Semarang	Dokter Hewan	sarjana	budha
13	70911	Febry Anadya	Depok	1998-03-11	perempuan	menikah	Depok	Karyawan Swasta	sarjana	kristen

Gambar 4.20 Halaman *Dashboard* Penduduk Kepala Desa.

**Data Kependudukan Desa Gambasan**

Kepala Desa  
Logout

**Data KK**

No	NO KK	Anggota Keluarga	Tanggal Buat KK	Alamat
1	234	685867-Ozik, 5466234-Maya, 31515-Nanda, 8312838-Vitalik	2024-11-20	Bantul
2	4545	14522-Nikio, 235253-Niken	undefined	Mgi
3	5626	9181-Kadilya Rakan, 1246235-Risa	undefined	Sieman
4	21091	6637-Fernando Ares, 70911-Febry Anadya	undefined	Temanggung
5	48361	71865-Niall Horan, 57900-Maudy Putri	undefined	Temanggung
6	64931	49023-Axel Giovanni, 24421-Sarah Zakia, 50125-Anissa	undefined	Temanggung
7	70345	14522-Nikio, 42506-Launa Putri	undefined	Temanggung
8	89011	43372-Louis Tomlinson, 14824-Safira, 34500-Chloe Li	undefined	Temanggung

Gambar 4.21 Halaman Kartu Keluarga Kepala Desa.

#### 4.1.4 Implementasi *Database*

Implementasi pengembangan ke dalam Firebase Realtime Database dan Firebase Authentication akan membuat data kependudukan terstruktur dalam bentuk koleksi yang mencerminkan peran dan tanggung jawab setiap pengguna. Berikut ini penjelasan mengenai hubungan antar data dalam sistem:

##### **Admin**

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

### Perangkat Desa

Perangkat desa bertugas mengelola data peristiwa kependudukan, seperti Kartu Keluarga, Penduduk, Menikah, Bercerai, Lahir, Meninggal, Pindah, dan Datang. Melalui Firebase Realtime Database, perangkat desa dapat memastikan semua perubahan terkait status dan informasi penduduk tercatat secara *real-time*.

### Kepala Desa

Kepala desa memiliki hak akses untuk memeriksa dan mendapatkan laporan dari data yang dikelola oleh admin dan perangkat desa, namun tidak diizinkan untuk melakukan perubahan pada data.

Dalam Firebase Realtime Database pada Tabel 4.1, semua data ini disimpan dalam struktur hierarki, tidak terikat dengan skema relasional. Setiap koleksi (seperti Penduduk, Menikah, Lahir) dapat diakses melalui ID unik, sementara relasi antar koleksi dapat dibuat dengan menggunakan NIK sebagai kunci penghubung.

Tabel 4.1 Implementasi *Database*

Koleksi	Dokumen
Bercerai	<pre>"Bercerai": {   "{IdBercerai}": {     "namaIstri": "",     "namaSuami": "",     "NIKistri": "",     "NIKSuami": "",     "IdMenikah": "",     "tanggalCerai": ""     "IdPerangkat": ""     "IdKK": "",   } },</pre>
Datang	<pre>"Datang": {   "{IdDatang}": {     "agama": "",     "alamat": "",     "alamatAsal": "",     "alasanDatang": "",</pre>

	<pre> "jenisKelamin": "", "nama": "", "NIK": "", "pekerjaan": "", "pendidikan": "", "status": "", "tanggalDatang": "" "tempat": "", "ttl": "" "IdKK": "", "IdPerangkat": "" } }, </pre>
Kartu Keluarga	<pre> "KK":{   "{IdKK}": {     "alamat": "",     "nik": [       "{nik1}",       "{nik2}",       "{nik3}",       "{nik4}",       "{nik5}",       "{nik6}",       "{nik7}",       "{nik8}"     ],     "tanggalBuat": ""     "IdPerangkat": ""   } }, </pre>
Lahir	<pre> "Lahir": {   "{IdLahir}": {     "agama": "", </pre>

	<pre>"alamat": "", "jenisKelamin": "", "namaAnak": "", "namaOrangTua": "", "NIK": "", "IdKK": "", "tanggalLahir": "", "tempatLahir": "" "IdPerangkat": "" } },</pre>
Menikah	<pre>"Menikah": {   "{IdMenikah}": {     "alamat": "",     "namaIstri": "",     "namaSuami": "",     "NIKistri": "",     "NIKSuami": "",     "IdKK": "",     "tanggalMenikah": ""     "IdPerangkat": ""   } },</pre>
Meninggal	<pre>"Meninggal": {   "{IdMeninggal}": {     "nama": "",     "NIK": "",     "nomorKematian": "",     "tanggalMeninggal": ""     "IdKK": "",     "IdPerangkat": ""   } },</pre>

Penduduk	<pre>"Penduduk": {   "{NIK}": {     "agama": "",     "alamat": "",     "jenisKelamin": "",     "nama": "",     "pekerjaan": "",     "pendidikan": "",     "status": "",     "tempat": "",     "ttl": ""     "IdKK": "",     "IdPerangkat": ""   } },</pre>
Pindah	<pre>"Pindah": {   "{IdPindah}": {     "alamatPindah": "",     "alasanPindah": "",     "NIK": "",     "tanggalPindah": ""     "IdKK": "",     "IdPerangkat": ""   } },</pre>
Users	<pre>"users": {   "{IdUser}": {     "email": "",     "isAdmin": false,     "isKepalaDesa": false,     "isPerangkat": false   } },</pre>

## 4.2 Pengujian Sistem

Pengujian sistem ini dilakukan untuk memastikan keamanan dan kinerja aplikasi yang berbasis Firebase Realtime Database dan Cloud Functions. Pengujian keamanan difokuskan pada evaluasi kontrol akses terhadap data, memastikan bahwa hanya pengguna yang memiliki izin yang dapat mengakses dan mengubah informasi sensitif. Firebase Security Rules Playground digunakan untuk menguji berbagai skenario kontrol akses, mulai dari aturan yang sepenuhnya membatasi akses hingga aturan yang mengizinkan hanya pengguna yang terotentikasi. Sementara itu, pengujian kinerja bertujuan untuk memantau waktu eksekusi dan penggunaan memori pada fungsi *cloud*, seperti pengiriman email dan penghapusan akun, guna memastikan sistem mampu menangani beban kerja dengan efisien.

### 4.2.1 Pengujian Keamanan

Pengujian keamanan sistem dilakukan menggunakan Firebase Security Rules Playground dengan berbagai skenario pada gambar 4.22 sampai dengan 4.24. Firebase Realtime Database memiliki aturan keamanan yang berfungsi sebagai lapisan perlindungan untuk mengontrol siapa yang diizinkan membaca atau menulis data di dalam database. Firebase Security Rules memanfaatkan sintaks JSON, yang memungkinkan penerapan kontrol akses berdasarkan identitas pengguna (*auth*), struktur data, serta logika kondisi tertentu untuk setiap *node* dalam *database*.

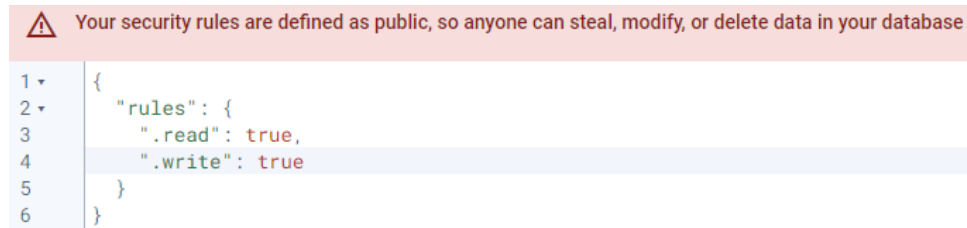
#### Skenario Pengujian 1

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

Gambar 4.22 *Rules False*.

Saat mencoba mengakses data di Firebase, baik untuk membaca maupun menulis, menggunakan akun terotentikasi atau anonim, semua permintaan ditolak dengan pesan kesalahan `PERMISSION_DENIED`. Setiap permintaan data dari aplikasi menghasilkan error, sehingga tidak ada data yang dapat ditampilkan di antarmuka pengguna. Selain itu, tidak ada perubahan data yang bisa dilakukan, memastikan bahwa keamanan data tetap terjaga. Hal ini menegaskan bahwa aturan `False` sudah berfungsi dengan baik, sepenuhnya memblokir akses ke *database*.

## Skenario Pengujian 2



The screenshot shows a warning message at the top: "Your security rules are defined as public, so anyone can steal, modify, or delete data in your database". Below the warning is a code editor showing the following JSON rule configuration:

```

1 {
2   "rules": {
3     ".read": true,
4     ".write": true
5   }
6 }

```

Gambar 4.23 *Rules True.*

Setiap pengguna, baik anonim, terautentikasi, maupun yang tidak terdaftar di sistem, dapat membaca dan menulis data ke database tanpa pembatasan apa pun. Data dapat diakses, ditampilkan, dan dimodifikasi secara bebas di antarmuka pengguna. Semua permintaan data, termasuk yang biasanya akan diblokir (misalnya, perubahan data milik pengguna lain), dapat berhasil dilakukan. Pengaturan ini menimbulkan risiko serius bagi keamanan data, karena tidak ada pembatasan akses yang diberlakukan. Siapa pun yang mengetahui URL *database* dapat melihat, mengubah, atau menghapus data. Akibatnya, data dapat diubah atau dihapus oleh pihak yang tidak berwenang, yang dapat mengancam integritas data. Pengaturan semacam ini sangat berbahaya jika digunakan dalam lingkungan produksi, terutama untuk aplikasi yang menyimpan data sensitif atau pribadi.

## Skenario Pengujian 3



The screenshot shows a code editor with the following JSON rule configuration:

```

1 {
2   "rules": {
3     ".read": "auth != null",
4     ".write": "auth != null"
5   }
6 }

```

Gambar 4.24 *Rules Auth.*

Aturan ini memungkinkan semua pengguna yang terautentikasi untuk membaca dan menulis data di *database*. Dengan kondisi `auth != null`, selama pengguna sudah *login*, mereka dapat mengakses dan memodifikasi data menggunakan ID pengguna (`uid`). Skenario ini adalah skenario yang paling aman karena data dapat dikendalikan oleh pengguna yang berhak mengakses data.

## Hubungan antara User dalam Tabel Users dengan Skenario Pengujian Keamanan

Keamanan akses data Users pada tabel 4.1 dengan skenario pengujian keamanan dapat dikelola secara efisien dengan Firebase Security Rules yang disesuaikan berdasarkan identitas dan peran pengguna. Ketika akun pengguna dibuat, Firebase Authentication menghasilkan `userID` unik yang digunakan sebagai kunci utama dalam struktur data users. Berdasarkan role

yang dipilih pada saat pembuatan akun, atribut seperti `isAdmin`, `isKepalaDesa`, atau `isPerangkat` yang sebelumnya **false** diatur ke **true** sesuai dengan peran pengguna yang dipilih.

Dalam Firebase Security Rules, aturan dapat diterapkan untuk membatasi akses, baik secara keseluruhan maupun per pengaturan spesifik dari `userID`. Misalnya, dalam Skenario Pengujian 1, dengan pengaturan aturan `false`, semua permintaan akan ditolak dan keamanan data sepenuhnya terjamin, meski pengguna telah login atau anonim. Sebaliknya, Skenario Pengujian 2 memberikan akses tanpa batas, yang memungkinkan siapa saja membaca dan mengubah data, membawa risiko besar untuk integritas data. Dalam Skenario Pengujian 3, aturan hanya mengizinkan pengguna terautentikasi (`auth != null`) untuk membaca atau menulis data, namun akses bisa diperjelas sesuai dengan peran yang telah diatur (`isAdmin`, `isKepalaDesa`, `isPerangkat`). Dengan demikian, aturan keamanan dan struktur data users bekerja sama untuk menjaga agar setiap pengguna hanya memiliki akses sesuai otoritasnya.

#### 4.2.2 Pengujian 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`.

>	2024-09-23 09:57:17.167	sendWelcomeEmail	si5qo7z3awmy	Function execution started
>	2024-09-23 09:57:20.003	sendWelcomeEmail	si5qo7z3awmy	New welcome email sent to: piwpie21@gmail.com
>	2024-09-23 09:57:20.007	sendWelcomeEmail	si5qo7z3awmy	Function execution took 2839 ms, finished with status: 'ok'
>	2024-09-23 09:58:52.203	sendByeEmailDeletedUser	gr8zdrbfuytv	Function execution started
>	2024-09-23 09:58:52.261	sendByeEmailDeletedUser	gr8zdrbfuytv	Start delete user data from realtime database. [uid] : 0lHIJ4anwbf5T8vdEDDYmEya6911
>	2024-09-23 09:58:52.297	sendByeEmailDeletedUser	gr8zdrbfuytv	Function execution took 93 ms, finished with status: 'ok'
>	2024-09-23 09:58:53.147	sendByeEmailDeletedUser	gr8zdrbfuytv	Successfully deleted user from auth. [email] : piwpie21@gmail.com
>	2024-09-23 09:59:02.647	sendByeEmailDeletedUser	gr8zdrbfuytv	Account deletion confirmation email sent to: piwpie21@gmail.com

Gambar 4.25 Logs.

#### 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.

#### 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 4.26 Logs.

### Execution Times (Waktu Eksekusi)

Grafik di sisi kiri pada gambar 4.26 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.

### Memory Usage (Penggunaan Memori)

Grafik di sisi kanan pada gambar 4.26 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.

Melalui analisis *logs* dari fungsi `sendWelcomeEmail` dan `sendByeEmailDeletedUser`, mengevaluasi kemampuan sistem untuk memproses akun baru dan penghapusan akun pengguna. Hasilnya menunjukkan bahwa email berhasil dikirim dan operasi penghapusan berlangsung cepat dan responsif, masing-masing dalam rentang waktu yang wajar. Dengan Cloud Function - Execution Time dan Memory Usage, penelitian ini menilai efisiensi waktu eksekusi dan penggunaan memori dalam skenario yang mencakup pengiriman email dan penghapusan data. Grafik eksekusi menunjukkan mayoritas waktu fungsi di bawah 1 detik, sedangkan grafik memori menunjukkan stabilitas tanpa lonjakan signifikan. Keseluruhan ini menunjukkan bahwa sistem berfungsi efisien dan menjaga performa layanan berkualitas tinggi.

## **BAB V**

### **KESIMPULAN**

#### **5.1 Kesimpulan**

Kesimpulan dari penelitian ini adalah bahwa penerapan *serverless computing* dalam pengelolaan data kependudukan di tingkat desa mampu memberikan efisiensi serta meningkatkan kualitas layanan. Keuntungan dari penggunaan *serverless computing* meliputi keamanan data yang lebih baik, kemudahan dalam skalabilitas, serta peningkatan kecepatan respon aplikasi. Namun, tantangan yang muncul mencakup perlunya memahami dan menerapkan aturan keamanan yang rumit serta tantangan dalam mengelola sumber daya *cloud* yang dinamis. Pengujian keamanan sistem melalui Firebase Security Rules berhasil menunjukkan berbagai skenario kontrol akses, mulai dari penolakan total hingga akses terbatas bagi pengguna yang terautentikasi, yang menyoroti pentingnya pengaturan akses yang tepat untuk menjaga integritas data. Pengujian kinerja menggunakan Cloud Monitoring mengungkapkan bahwa sistem dapat menangani pengiriman email selamat datang dan penghapusan akun dengan efisien, mempertahankan waktu eksekusi yang cepat serta penggunaan memori yang stabil. Ini menegaskan bahwa *serverless computing* tidak hanya meningkatkan kecepatan respon aplikasi, tetapi juga mampu menjamin keamanan data dengan pengaturan akses yang ketat. Secara keseluruhan, penelitian ini memberikan arahan bagi pemerintah desa dalam memanfaatkan teknologi *serverless* untuk meningkatkan efisiensi operasional dan kualitas layanan kepada masyarakat.

#### **5.2 Saran**

Berikut saran yang dapat dipertimbangkan dari pengembangan Sistem Informasi dengan Pendekatan Serverless Computing adalah :

- a. Pemerintah desa sebaiknya mengadakan pelatihan bagi pegawai dan pengguna sistem mengenai konsep *serverless computing*, keamanan data, dan penggunaan Firebase. Pemahaman yang baik akan teknologi ini akan membantu dalam implementasi dan pengelolaan sistem yang lebih efektif.
- b. Melakukan *monitoring* secara berkelanjutan terhadap kinerja sistem dan penggunaan sumber daya *cloud*. Ini akan membantu dalam mendeteksi dan mengatasi masalah secara proaktif, serta memastikan bahwa sistem tetap efisien dan responsif.

- c. Mengumpulkan umpan balik dari pengguna sistem untuk mengevaluasi fungsionalitas dan kinerja aplikasi. Hal ini dapat memberikan wawasan berharga untuk perbaikan berkelanjutan dan peningkatan pengalaman pengguna.
- d. Mengidentifikasi dan mengembangkan fitur tambahan yang dapat meningkatkan layanan kepada masyarakat, seperti integrasi dengan aplikasi lain atau penambahan fungsi analitik untuk pengambilan keputusan yang lebih baik.
- e. Melakukan pengujian keamanan dan kinerja sistem secara berkala untuk memastikan bahwa semua aspek aplikasi berfungsi dengan baik dan untuk mengidentifikasi potensi risiko yang mungkin muncul.

## DAFTAR PUSTAKA

- Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A., & Suter, P. (2017). Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing* (pp. 1–20). Springer Singapore. [https://doi.org/10.1007/978-981-10-5026-8\\_1](https://doi.org/10.1007/978-981-10-5026-8_1)
- Bayu, R., Pradana, A., & Bhawiyuga, A. (2022). *Pengembangan Platform IoT Cloud berbasis Layanan Komputasi Serverless Google Cloud Platform (GCP)* (Vol. 6, Issue 4). <http://j-ptiik.ub.ac.id>
- Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The rise of serverless computing. *Communications of the ACM*, 62(12), 44–54. <https://doi.org/10.1145/3368454>
- Eismann, S., Scheuner, J., Van Eyk, E., Schwinger, M., Grohmann, J., Herbst, N., Abad, C. L., & Iosup, A. (2021). Serverless Applications: Why, When, and How? *IEEE Software*, 38(1), 32–39. <https://doi.org/10.1109/MS.2020.3023302>
- Facta, Mochammad. (2018). *Ticketing Chatbot Service using Serverless NLP Technology*. Department of Electrical Engineering, Diponegoro University. [https://docpak.undip.ac.id/id/eprint/22340/1/Artikel\\_ICITACEE\\_2018.pdf](https://docpak.undip.ac.id/id/eprint/22340/1/Artikel_ICITACEE_2018.pdf)
- Haque, S., Eberhart, Z., Bansal, A., & McMillan, C. (2022). The Serverless Computing Survey: A Technical Primer for Design Architecture. *IEEE International Conference on Program Comprehension, 2022-March*, 36–47. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>
- McGrath, G., & Brenner, P. R. (2017). Serverless Computing: Design, Implementation, and Performance. *Proceedings - IEEE 37th International Conference on Distributed Computing Systems Workshops, ICDCSW 2017*, 405–410. <https://doi.org/10.1109/ICDCSW.2017.36>
- Mulyani, A., & Oktawati, U. Y. (2022). Implementasi Arsitektur Serverless Internet of Things pada Monitoring Cold Chain. *Journal of Internet and Software Engineering*, 3(1). <https://download.garuda.kemdikbud.go.id/article.php?article=3249530&val=28494&title=Implementation%20of%20Serverless%20Internet%20of%20Things%20Architecture%20in%20Cold%20Chain%20Monitoring>
- Nupponen, J., & Taibi, D. (2020). Serverless: What it Is, What to Do and What Not to Do. *Proceedings - 2020 IEEE International Conference on Software Architecture Companion, ICSCA-C 2020*, 49–50. <https://doi.org/10.1109/ICSCA-C50368.2020.00016>

- Oktaviani, D., Papilaya, F. S., & Tanaem, F. (2021). *Perancangan Aplikasi E-Menu Restaurant dengan Menggunakan Cloud Computing dan Serverless Architecture Lambda*. 12. <https://pdfs.semanticscholar.org/f159/a725166307a4a2cf2572406f7c22ad31cb92.pdf>
- Puspitasari. (2016). *SISTEM INFORMASI PERPUSTAKAAN SEKOLAH BERBASIS WEB*. <https://ejournal.nusamandiri.ac.id/index.php/pilar/article/view/277>
- Rajeshwari B Mathapati. (2023). *Building Serverless Applications with GCP's Cloud Functions*. <https://www.cloudthat.com/resources/blog/building-serverless-applications-with-gcps-cloud-functions#steps-to-build-a-serverless-application-with-cloud-functions>.
- Saputra, D. (2023). *Kelola Kode Serverless dengan Serverless Framework*. <https://www.dicoding.com/blog/kelola-kode-serverless-dengan-serverless-framework/>.
- Shafiei, H., Khonsari, A., & Mousavi, P. (2019). *Serverless Computing: A Survey of Opportunities, Challenges and Applications*. <http://arxiv.org/abs/1911.01296>
- Utomo, P., & Falahah. (2020). Building Serverless Website on GitHub Pages. *IOP Conference Series: Materials Science and Engineering*, 879(1). <https://doi.org/10.1088/1757-899X/879/1/012077>
- Yan, M., Castro, P., Cheng, P., & Ishakian, V. (2016, December 12). Building a chatbot with serverless computing. *Proceedings of the 1st International Workshop on Mashups of Things and APIs, MOTA 2016*. <https://doi.org/10.1145/3007203.3007217>