



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 5 No. 2 (2026) pp: 23707-23721

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Implementasi IaaS (Infrastructure as a Service) Menggunakan Amazon Web Services Untuk Membangun Website Berbasis Desa Digital

M. Adam Hawary Saputra¹, Tomi Tri Sujaka², Rifqi Hammad³

¹²Program Studi Ilmu Komputer, Fakultas Teknik, Universitas Bumigora

³Program Studi Rekayasa Perangkat Lunak, Fakultas Teknik, Universitas Bumigora

¹adamhawary08@gmail.com, ²Tomi_tri@universitasbumigora.ac.id, ³rifqi.hammad@universitasbumigora.ac.id

Abstrak

Program digitalisasi desa terus didorong pemerintah untuk pemerataan layanan publik, namun penerapannya di lapangan masih terhambat keterbatasan infrastruktur, sehingga banyak desa belum mampu membangun sistem informasi secara mandiri. Desa Bengkel, misalnya, telah memiliki website namun belum memiliki layanan administrasi online maupun mekanisme verifikasi dokumen, sehingga pengajuan surat masih dilakukan secara manual. Penelitian ini bertujuan membangun website desa digital berbasis Infrastructure as a Service (IaaS) menggunakan Amazon Web Services (AWS) yang mengintegrasikan enam layanan sekaligus, yaitu EC2, RDS, S3, Lambda, SQS, dan SNS, dengan kebaruan utama berupa fungsi AWS Lambda sqs-pdf-signer yang secara otomatis menyisipkan QR Code verifikasi ke dalam dokumen surat tanpa memerlukan tanda tangan fisik. Sistem dikembangkan menggunakan metode Software Development Life Cycle (SDLC) model Waterfall. Hasil Black Box Testing menunjukkan seluruh kebutuhan fungsional terpenuhi. User Acceptance Testing memperoleh rata-rata skor 4,45 dari masyarakat dan 5,00 dari perangkat desa, keduanya berkategori Sangat Baik. Stress Testing dengan beban hingga 500 pengguna secara bersamaan menunjukkan error rate 0,00–0,02% tanpa downtime, sementara Security Testing menunjukkan konfigurasi SSL/TLS Grade A tanpa kerentanan berisiko tinggi. Hasil ini membuktikan bahwa integrasi IaaS berbasis AWS dapat menjadi solusi infrastruktur sekaligus otomatisasi layanan administrasi publik tingkat desa yang andal dan terverifikasi secara digital.

Kata Kunci: Infrastructure as a Service, Amazon Web Services, Website Desa Digital, Cloud Computing, SDLC Waterfall.

1. Latar Belakang

Pemerintah Indonesia dalam beberapa tahun terakhir terus mendorong program digitalisasi desa sebagai bagian dari upaya pemerataan pembangunan, melalui program Smart Village yang digagas Kemendes PDTT sejak tahun 2021, termasuk pengembangan website desa untuk pelayanan administrasi dan penyampaian informasi. Namun, realisasinya di tingkat desa tidak selalu berjalan mulus. Penerapan e-government umumnya masih terhambat oleh keterbatasan infrastruktur, sumber daya manusia, hingga rendahnya literasi digital masyarakat [1], sehingga banyak desa belum mampu membangun sistem informasi secara mandiri dan proses digitalisasi berjalan lambat serta tidak merata.

Desa Bengkel, Kecamatan Labuapi, Kabupaten Lombok Barat, merupakan salah satu desa yang turut berupaya mendukung transformasi digital tersebut. Desa ini telah memiliki website sebagai sarana publikasi informasi, namun belum dilengkapi layanan administrasi online maupun pengaduan masyarakat, sehingga pengajuan surat masih dilakukan secara manual dengan masyarakat datang langsung ke kantor desa. Sistem Informasi Desa seperti OpenSID sebenarnya telah menyediakan kemudahan layanan administrasi desa [2], namun umumnya belum dilengkapi mekanisme verifikasi keaslian dokumen otomatis, sehingga keabsahannya tetap bergantung pada tanda tangan dan cap fisik. Selain dari sisi layanan, permasalahan juga ditemukan dari sisi infrastruktur, di mana pengelolaan data pada website Desa Bengkel masih bergantung pada server fisik dan perangkat lokal seperti laptop, PC, dan hard disk, sehingga berisiko terhadap kehilangan data serta memiliki keterbatasan kapasitas penyimpanan [3], [4]. Keterbatasan dana dan fasilitas pendukung turut menghambat penyediaan infrastruktur server secara mandiri di tingkat desa [5], [6].

Untuk mengatasi permasalahan tersebut, cloud computing dapat dijadikan solusi infrastruktur layanan publik. Kajian bibliometrik terhadap tren global cloud computing pada arsitektur e-government menegaskan bahwa cloud computing merupakan solusi yang tepat bagi pemerintah daerah di negara berkembang, termasuk Indonesia, karena memungkinkan penyelenggaraan layanan publik digital tanpa investasi perangkat keras yang besar [7]. Model Infrastructure as a Service (IaaS) secara khusus memungkinkan pengguna mengakses sumber daya server,

jaringan, dan penyimpanan secara virtual dengan kapasitas yang fleksibel [8], [9], dan penerapannya pada instansi pemerintah daerah di Indonesia terbukti meningkatkan efisiensi pengelolaan infrastruktur TI serta mengurangi ketergantungan terhadap perangkat keras fisik [6]. Dalam ekosistem cloud computing, terdapat beberapa platform penyedia layanan IaaS yang umum digunakan, yaitu Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), serta Virtual Private Server (VPS) [10], [11]. Amazon Web Services (AWS) dipilih sebagai platform IaaS dalam penelitian ini atas pertimbangan kelengkapan ekosistem layanan yang terintegrasi dalam satu lingkungan [12].

Penelitian terdahulu yang menerapkan layanan AWS pada konteks serupa umumnya masih terbatas pada pemanfaatan layanan komputasi dan penyimpanan dasar, seperti Amazon EC2 dan S3 sebagai infrastruktur hosting [13], atau penambahan Elastic Load Balancing dan Auto Scaling pada EC2 untuk kebutuhan skalabilitas [14]. Kedua pendekatan tersebut belum menyentuh otomatisasi proses administratif berbasis event-driven, khususnya pada aspek verifikasi keaslian dokumen digital, sehingga penelitian ini mengintegrasikan enam layanan AWS sekaligus, yaitu Amazon EC2, Amazon RDS, Amazon S3, AWS Lambda, Amazon SQS, dan Amazon SNS, untuk melengkapi hal tersebut.

Mengacu pada uraian tersebut, penelitian ini mengusulkan implementasi IaaS menggunakan AWS sebagai infrastruktur pendukung website Desa Bengkel, dengan kebaruan utama berupa fungsi AWS Lambda "sqs-pdf-signer" yang secara otomatis menyisipkan QR Code verifikasi ke dalam dokumen surat yang disetujui, sehingga penerbitan surat dapat berlangsung sepenuhnya secara online tanpa tanda tangan fisik. Dengan demikian, penelitian ini bertujuan membangun website desa digital berbasis IaaS AWS sekaligus menjadi referensi teknis penerapan infrastruktur cloud pada pelayanan publik di tingkat desa.

2. Metode Penelitian

2.1. Metode Pengumpulan Data

1. Wawancara

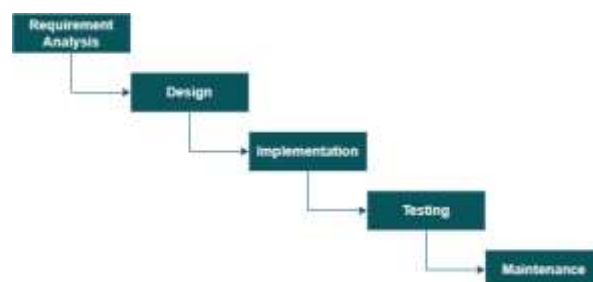
Wawancara dilakukan sebagai metode pengumpulan data primer untuk mengetahui kebutuhan informasi, pola interaksi antara pemerintah desa dan masyarakat, serta persepsi perangkat desa terhadap penggunaan teknologi dalam pelayanan publik. Wawancara dilakukan secara langsung kepada Kepala Desa Bengkel beserta perangkat desa untuk memperoleh masukan terkait kebutuhan sistem yang akan dibangun.

2. Observasi

Observasi dilakukan secara langsung di Kantor Desa Bengkel untuk mengamati proses layanan administrasi yang sedang berjalan, hambatan yang sering muncul, serta cara pengelolaan data dan informasi yang selama ini diterapkan.

2.2. Metode Pengembangan Sistem

Metode perancangan sistem yang digunakan dalam penelitian ini adalah Software Development Life Cycle (SDLC) dengan model Waterfall, yang mengikuti pendekatan bersifat linier dan bertahap, di mana setiap tahap harus diselesaikan secara menyeluruh sebelum melanjutkan ke tahap berikutnya [15]. Model Waterfall dalam penelitian ini terdiri dari lima tahapan sebagai berikut:



Gambar 1. Tahapan SDLC Waterfall

1. Requirement Analysis (Analisis Kebutuhan)

Tahapan ini berfokus pada pengumpulan dan analisis kebutuhan sistem yang akan dibangun berdasarkan hasil wawancara dan observasi yang dilakukan, guna mengidentifikasi kebutuhan utama serta kebutuhan pendukung yang diperlukan dalam proses implementasi.

2. Design (Perancangan Sistem)

Pada tahap ini disusun rancangan sistem yang akan dibangun, meliputi Use Case Diagram dan Activity Diagram untuk menggambarkan interaksi antara aktor dengan sistem, Entity Relationship Diagram

(ERD) untuk merancang struktur basis data, serta rancangan arsitektur sistem berbasis cloud computing yang mengintegrasikan layanan Amazon Web Services (AWS).

3. Implementation (Implementasi)

Pengembangan website dilakukan menggunakan framework Laravel dengan bahasa pemrograman PHP untuk membangun sisi backend dan frontend sistem. Implementasi infrastruktur dilakukan menggunakan layanan Amazon Web Services (AWS) sebagai penyedia Infrastructure as a Service (IaaS), mencakup konfigurasi server pada Amazon EC2, penyimpanan data terstruktur pada Amazon RDS, penyimpanan objek pada Amazon S3, pemrosesan otomatis berbasis event menggunakan AWS Lambda, manajemen antrian pesan melalui Amazon SQS, serta notifikasi pengguna melalui Amazon SNS, yang seluruhnya terintegrasi dalam satu lingkungan cloud.

4. Testing (Pengujian)

Pengujian dilakukan mencakup empat aspek utama, yaitu pengujian fungsionalitas menggunakan metode Black Box Testing, pengujian penerimaan pengguna menggunakan kuesioner berbasis Skala Likert, pengujian performa menggunakan Stress Testing dengan Apache JMeter, serta pengujian keamanan menggunakan SSL Labs, OWASP ZAP, dan Mozilla HTTP Observatory.

5. Maintenance (Pemeliharaan)

Pemeliharaan sistem dilakukan melalui mekanisme version control menggunakan GitHub sebagai repositori pusat kode program, sehingga setiap perubahan dapat terdokumentasi dan dilacak secara terstruktur. Pemeliharaan juga mencakup pemantauan ketersediaan server pada Amazon EC2 serta pengelolaan backup data pada Amazon RDS dan Amazon S3 untuk mengantisipasi kehilangan data.

3. Hasil dan Diskusi

3.1. Requirement Analysis (Analisis Kebutuhan)

Berdasarkan hasil wawancara dengan kepala desa beserta perangkat desa dan observasi terhadap proses pelayanan administrasi di Kantor Desa Bengkel, diperoleh sejumlah kebutuhan yang menjadi dasar pengembangan website desa digital. Sistem dirancang untuk menyediakan layanan informasi desa, meliputi profil desa, berita, dan pengumuman, serta mendukung digitalisasi pelayanan administrasi melalui fitur pengajuan surat, pengelolaan data kependudukan, dan autentikasi pengguna sesuai hak akses masing-masing. Selain kebutuhan fungsional, sistem juga dirancang untuk memenuhi kebutuhan nonfungsional, seperti ketersediaan layanan, keamanan data, dan kemudahan akses melalui pemanfaatan infrastruktur Amazon Web Services (AWS). Seluruh kebutuhan tersebut menjadi acuan pada tahap perancangan dan implementasi sistem sehingga arsitektur yang dibangun mampu mendukung proses pelayanan administrasi secara lebih efektif dibandingkan proses manual yang sebelumnya diterapkan. Spesifikasi perangkat keras dan perangkat lunak yang digunakan selama proses pengembangan serta implementasi sistem disajikan pada Tabel 1 dan Tabel 2.

Tabel 1. Kebutuhan perangkat keras

Nama Perangkat	Spesifikasi
Laptop	- Processor: 11 th Gen Intel® Core™ i5-1140H @ 2.70GHz (2.69GHz)
	- Memory: 8 GB RAM
	- Storage: 447 GB SSD
	- Sistem Operasi: Windows 11
Server Cloud (Amazon EC2)	- vCPU: 2 Core
	- Memory: 1 GiB
	- Storage: 20 GB
	- Sistem Operasi: Ubuntu Server 22.04 LTS

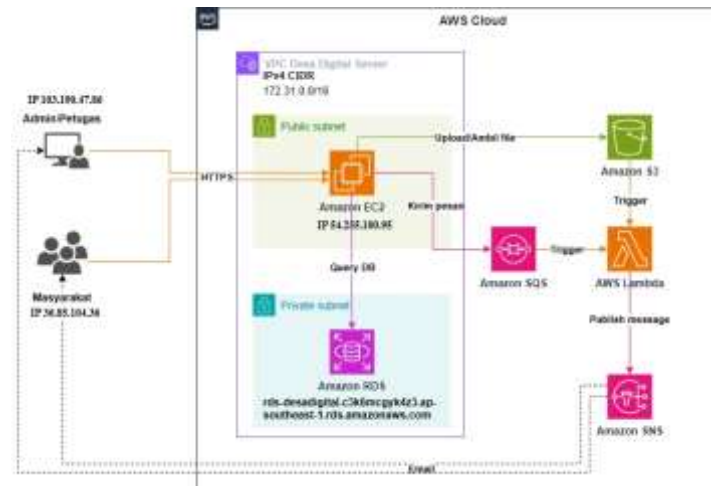
Tabel 2. Kebutuhan perangkat lunak

Nama Software	Fungsi
Windows 11	Sistem operasi yang digunakan untuk menjalankan seluruh aplikasi pengembangan sistem.
Ubuntu Server 22.04 LTS	Sistem operasi server yang digunakan pada instance Amazon EC2 sebagai lingkungan deployment website dan layanan backend
Amazon Web Services	Digunakan sebagai platform cloud computing untuk implementasi layanan infrastructure as a Service (IaaS). Seperti EC2, RDS, S3, Lambda, SQS, dan SNS.
Visual Studio Code	Digunakan sebagai text editor untuk menulis dan mengedit kode program website.
MySQL Workbench	Digunakan untuk mengatur data dari database Amazon RDS.
Google Chrome/Microsoft Edge	Digunakan untuk melakukan pengujian tampilan dan fungsionalitas website.

3.2. Design (Perancangan Sistem)

Tahap perancangan menghasilkan model sistem yang menjadi acuan dalam proses implementasi website desa digital. Perancangan meliputi Use Case Diagram, Activity Diagram, Entity Relationship Diagram (ERD), serta rancangan arsitektur sistem berbasis Amazon Web Services (AWS). Keempat komponen tersebut digunakan untuk menggambarkan kebutuhan fungsional, alur proses bisnis, struktur basis data, dan integrasi infrastruktur cloud yang mendukung operasional sistem.

Desain Arsitektur Sistem



Gambar 5. Desain Arsitektur Sistem

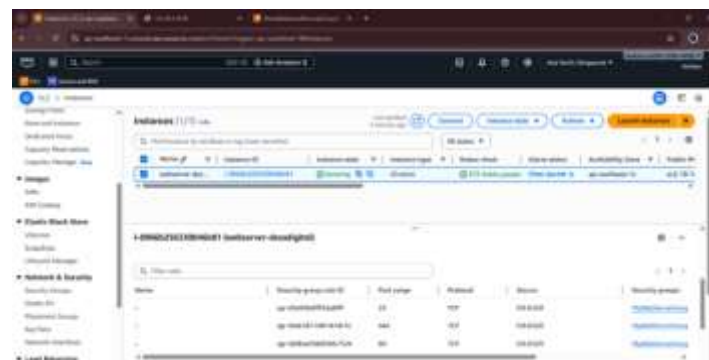
Rancangan arsitektur sistem menggambarkan integrasi layanan infrastruktur berbasis model Infrastructure as a Service (IaaS), di mana sumber daya server, jaringan, dan penyimpanan diakses secara virtual dan dapat disesuaikan secara fleksibel sesuai kebutuhan [8]. Arsitektur sistem dirancang dengan mengintegrasikan enam layanan Amazon Web Services (AWS) dalam satu lingkungan Virtual Private Cloud (VPC). Amazon EC2 berfungsi sebagai web server yang menerima permintaan pengguna melalui protokol HTTPS, kemudian berkomunikasi dengan Amazon RDS sebagai penyimpanan data terstruktur dan Amazon S3 sebagai media penyimpanan objek. Proses yang memerlukan eksekusi asynchronous dikelola menggunakan Amazon SQS, sedangkan AWS Lambda menjalankan pemrosesan berbasis event, seperti penandatanganan dokumen digital dan pengiriman notifikasi melalui Amazon SNS. Integrasi tersebut membentuk arsitektur Infrastructure as a Service (IaaS) yang mendukung layanan administrasi desa secara terpusat, otomatis, dan mudah dikembangkan.

3.3. Impelentation (Implementasi)

Tahap implementasi merupakan penerapan dari hasil perancangan sistem yang telah dijelaskan pada subbab sebelumnya. Pembangunan aplikasi website desa digital dilakukan menggunakan framework Laravel, sementara implementasi infrastruktur dilakukan menggunakan layanan cloud computing Amazon Web Services (AWS) sebagai penyedia Infrastructure as a Service (IaaS), meliputi Amazon EC2 sebagai web server, Amazon RDS sebagai database server, Amazon S3 sebagai media penyimpanan objek, AWS Lambda sebagai layanan komputasi serverless, Amazon SQS sebagai layanan antrean pesan, serta Amazon SNS sebagai layanan notifikasi, yang seluruhnya terintegrasi dalam satu lingkungan cloud.

Amazon EC2 (Elastic Compute Cloud)

Amazon EC2 diimplementasikan sebagai web server utama yang menjalankan aplikasi website desa digital. Instance menggunakan tipe t3.micro dengan sistem operasi Ubuntu Server 22.04 LTS dan ditempatkan pada public subnet di dalam Virtual Private Cloud (VPC) “Desa Digital Server”. Untuk memastikan server dapat diakses secara konsisten melalui internet, instance dikaitkan dengan Elastic IP sehingga alamat IP publik tidak berubah meskipun instance dihentikan atau dijalankan kembali.



Gambar 6. Detail instance EC2

Hasil implementasi menunjukkan bahwa instance berhasil dijalankan dengan status Running, sehingga dapat digunakan sebagai lingkungan deployment aplikasi. Website kemudian dikonfigurasi menggunakan Nginx sebagai web server, PHP sebagai runtime aplikasi, serta Composer untuk pengelolaan dependensi Laravel. Selanjutnya dilakukan konfigurasi DNS sehingga domain “desabengkel.site” mengarah ke Elastic IP instance, kemudian dipasang sertifikat SSL/TLS menggunakan Certbot agar seluruh komunikasi antara pengguna dan server berlangsung melalui protokol HTTPS.

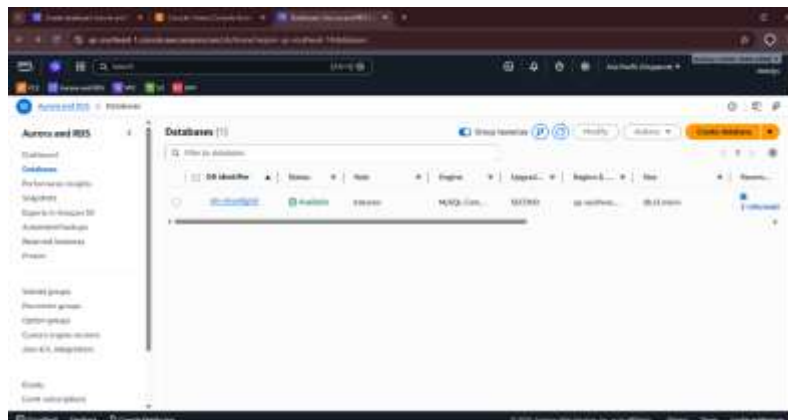


Gambar 7. Website berhasil diakses via HTTPS

Website berhasil diakses menggunakan nama domain melalui koneksi HTTPS, yang menunjukkan bahwa layanan Amazon EC2 telah berfungsi sebagai infrastruktur komputasi utama bagi aplikasi website desa digital. Implementasi ini juga menjadi fondasi integrasi dengan layanan AWS lainnya, seperti Amazon RDS untuk penyimpanan data, Amazon S3 untuk penyimpanan berkas, serta AWS Lambda, Amazon SQS, dan Amazon SNS untuk mendukung otomatisasi proses layanan administrasi.

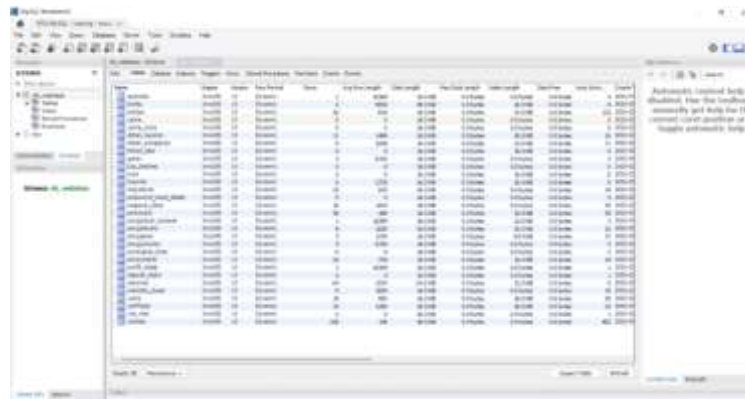
Amazon RDS (Relational Database Service)

Amazon RDS diimplementasikan sebagai server basis data menggunakan MySQL Community 8.4.7 yang ditempatkan pada private subnet sehingga tidak dapat diakses secara langsung melalui internet. Pengaturan tersebut bertujuan meningkatkan keamanan penyimpanan data dengan membatasi akses hanya dari instance Amazon EC2 melalui security group yang telah dikonfigurasi.



Gambar 8. Instance database Amazon RDS

Integrasi antara aplikasi Laravel dan Amazon RDS dilakukan melalui parameter koneksi pada berkas environment. Hasil implementasi menunjukkan bahwa basis data “db_webdesa” berhasil digunakan sebagai media penyimpanan seluruh data aplikasi. Struktur basis data terdiri atas 20 tabel utama yang mendukung proses bisnis sistem serta 10 tabel bawaan Laravel yang digunakan untuk mendukung operasional framework.

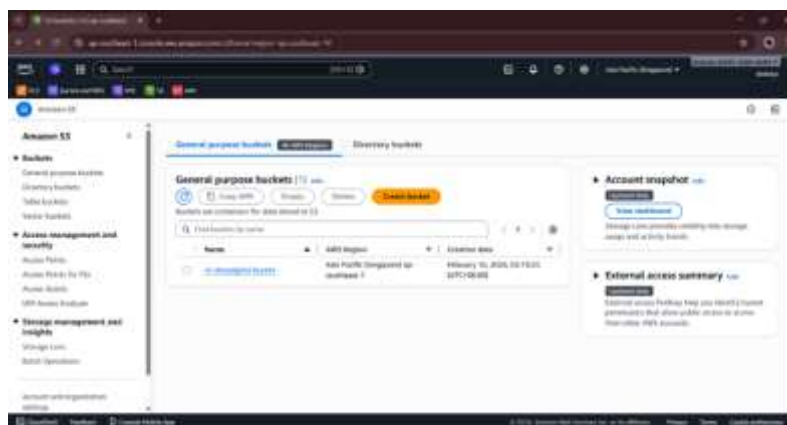


Gambar 9. Tampilan Database pada MySQL Workbench

Implementasi Amazon RDS memungkinkan pengelolaan data dilakukan secara terpusat pada layanan cloud sehingga aplikasi dapat mengakses basis data secara konsisten tanpa bergantung pada server lokal.

Amazon S3 (Simple Storage Service)

Amazon S3 digunakan sebagai media penyimpanan objek untuk berbagai berkas yang diunggah pengguna, meliputi foto profil, gambar berita, dokumen persyaratan administrasi, serta dokumen surat yang telah diproses. Bucket dikonfigurasi dengan server-side encryption (SSE-S3) serta pengaturan hak akses yang membedakan objek publik dan objek privat sesuai kebutuhan sistem.



Gambar 10. Bucket Amazon S3

Aplikasi mengakses bucket menggunakan IAM User dengan kebijakan akses terbatas (least privilege) sehingga hanya memiliki izin yang diperlukan untuk mengunggah, mengambil, menghapus, dan mengatur visibilitas objek. Implementasi tersebut memungkinkan penyimpanan dokumen dilakukan secara terpusat, aman, dan terintegrasi dengan proses bisnis aplikasi.

Tabel 3. Custom policy bucket “s3-desadigital-bucket”

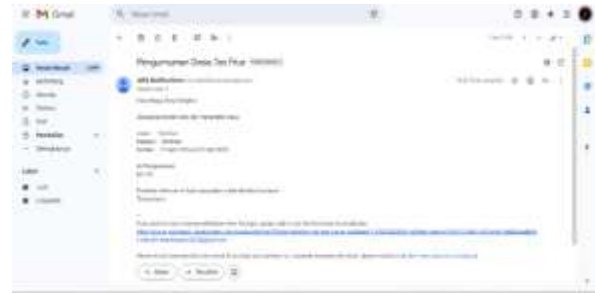
Kelompok Izin	Aksi	Fungsi
Operasi pada objek di dalam bucket	s3:PutObject	Mengunggah file ke bucket
	s3:GetObject	Mengambil file dari bucket
	s3>DeleteObject	Menghapus file dari bucket
	s3:PutObjectAcl	Mengatur visibilitas file
Membaca daftar isi bucket	s3:ListBucket	Mengetahui daftar file yang tersimpan di dalam bucket

Amazon SNS (Simple Notification Service)

Amazon SNS dimanfaatkan sebagai layanan pengiriman notifikasi email kepada admin dan masyarakat. Implementasi menggunakan dua topic, yaitu “notifikasi-admin” untuk pemberitahuan pengajuan surat dan pengaduan baru, serta “notifikasi-desa” untuk penyampaian pengumuman kepada masyarakat.



Gambar 11. Notifikasi email yang diterima admin



Gambar 12. Notifikasi email yang diterima warga

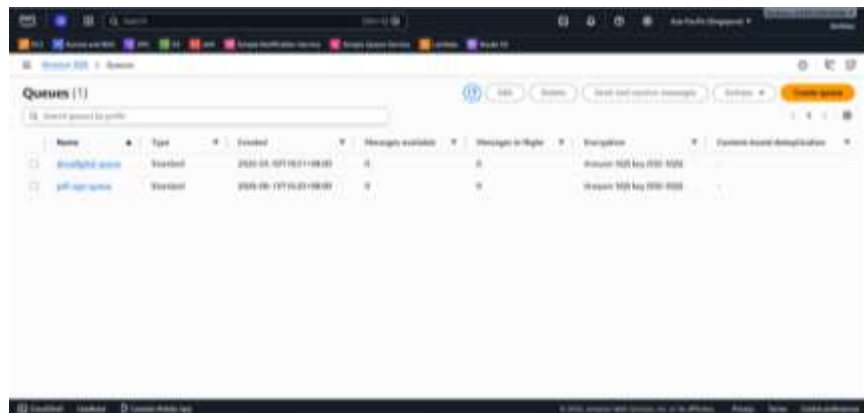
Seluruh proses komunikasi dengan Amazon SNS dikelola melalui kelas layanan “SnsService” sehingga pengiriman notifikasi maupun pendaftaran subscriber dilakukan secara terpusat. Pendekatan ini mempermudah pemeliharaan kode sekaligus memastikan proses distribusi informasi berjalan secara otomatis sesuai jenis notifikasi.

Tabel 4. Method pada kelas SnsService

Method	Fungsi
publishPengumuman()	Mengirim pesan broadcast kepada seluruh subscriber topic notifikasi-desa, digunakan untuk keperluan pengumuman desa dan agenda kegiatan
daftarkanEmail()	Mendaftarkan alamat email warga sebagai subscriber pada topic notifikasi-desa
daftarkanEmailAdmin()	Mendaftarkan alamat email admin sebagai subscriber pada topic notifikasi-admin

Amazon SQS (Simple Queue Service)

Amazon SQS diimplementasikan sebagai layanan antrian pesan untuk mendukung pemrosesan asynchronous sehingga proses yang memerlukan waktu lebih lama tidak menghambat respons aplikasi kepada pengguna. Dua antrian digunakan pada penelitian ini, yaitu “desadigital-queue” untuk pengiriman notifikasi dan “pdf-sign-queue” untuk proses penandatanganan dokumen digital.



Gambar 13. Antrian Amazon SQS

Interaksi aplikasi dengan Amazon SQS dikelola melalui kelas “SqsService” yang menyediakan fungsi pengiriman pesan sesuai konteks layanan. Pemisahan antrian berdasarkan fungsi tersebut mendukung penerapan prinsip single responsibility, sehingga setiap antrian hanya menangani satu jenis proses.

Tabel 5. Method pada kelas SqsService

Method	Fungsi
kirimPesan()	Mengirim pesan ke antrian SQS dengan payload berisi judul, isi pesan, tipe pengiriman (admin atau broadcast), dan waktu pengiriman
kirimSignPdf()	Mengirim pesan ke antrian “pdf-sign-queue” dengan payload berisi path file PDF di Amazon S3, nomor surat, dan verification hash untuk proses penandatanganan digital

Tabel 6. Tipe pesan pada antrian Amazon SQS.

Nama Antrian	Tipe Pesan	Kondisi Pemicu	Aksi
--------------	------------	----------------	------

desadigital-queue	admin	Warga mengajukan surat atau mengirimkan pengaduan baru	Notifikasi diteruskan ke seluruh admin melalui Amazon SNS
desadigital-queue	broadcast	Admin membuat berita, pengumuman, atau agenda kegiatan baru	Notifikasi diteruskan ke seluruh warga yang terdaftar sebagai subscriber
pdf-sign-queue	sign	Petugas menyetujui pengajuan surat	Lambda functions sqs-pdf-signer menyematkan QR Code verifikasi pada dokumen surat

Amazon Lambda

AWS Lambda digunakan sebagai layanan komputasi serverless untuk menjalankan proses berbasis event-driven tanpa memerlukan pengelolaan server tambahan. Implementasi terdiri atas tiga function, yaitu “sqs-to-sns-handler” untuk meneruskan pesan dari Amazon SQS ke Amazon SNS, “s3-event-handler” untuk memproses unggahan berkas pada Amazon S3, serta “sqs-pdf-signer” yang bertugas menyisipkan QR Code verifikasi ke dalam dokumen surat secara otomatis.

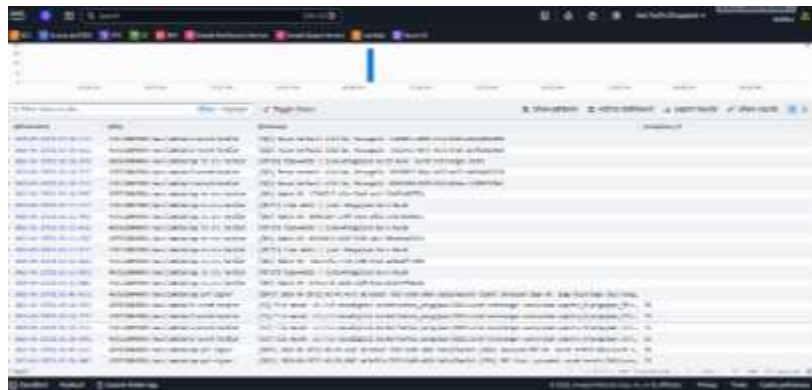


Gambar 14. Dokumen Surat dengan QR Code



Gambar 15. Hasil Pemindaian/Halaman Verifikasi QR Code

Dokumen yang telah disetujui diproses secara otomatis oleh fungsi “sqs-pdf-signer”, kemudian diunggah kembali ke Amazon S3 dalam bentuk dokumen yang telah dilengkapi QR Code. Ketika QR Code dipindai, pengguna diarahkan ke halaman verifikasi yang menampilkan informasi keaslian dokumen berdasarkan verification hash yang tersimpan pada sistem. Mekanisme tersebut memungkinkan proses penerbitan dan verifikasi surat dilakukan secara digital tanpa memerlukan tanda tangan maupun cap fisik.



Gambar 16. Hasil eksekusi Lambda functions pada CloudWatch Logs.

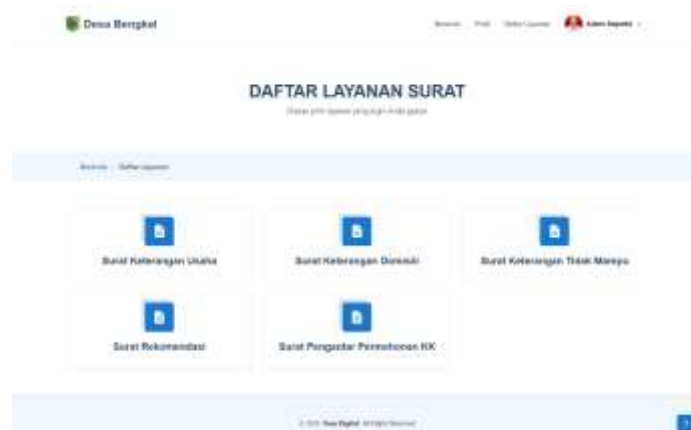
Keberhasilan setiap fungsi dalam memproses event turut dipantau melalui Amazon CloudWatch. Hasil pemantauan menunjukkan bahwa setiap trigger berhasil menjalankan fungsi sesuai perannya, baik dalam proses pengiriman notifikasi maupun penandatanganan dokumen digital.

Implementasi Frontend

Antarmuka aplikasi dikembangkan menggunakan Laravel Blade Template, dengan MediLab (Bootstrap 5.3.3) digunakan pada halaman publik dan layanan mandiri masyarakat, sedangkan digunakan pada dashboard administrator dan petugas. Pemisahan antarmuka tersebut bertujuan menyesuaikan kebutuhan masing-masing pengguna, sehingga masyarakat memperoleh akses yang sederhana terhadap layanan administrasi, sementara administrator memiliki fasilitas pengelolaan data yang lebih lengkap.



Gambar 17. Tampilan Halaman Utama Website Desa Bengkel



Gambar 18. Halaman daftar layanan



Gambar 19. Halaman form pengajuan surat



Gambar 20. Tampilan Dashboard Admin

3.4. Testing (Pengujian)

Tahap pengujian dilakukan untuk mengevaluasi sistem yang telah dibangun dari aspek fungsionalitas, penerimaan pengguna, performa, dan keamanan. Pengujian meliputi Black Box Testing, User Acceptance Testing (UAT), Stress Testing, dan Security Testing.

Black Box Testing

Black Box Testing dilakukan untuk memverifikasi kesesuaian fungsi sistem dengan kebutuhan yang telah ditetapkan pada tahap analisis. Pengujian mencakup fitur utama, seperti registrasi akun, autentikasi pengguna, pengajuan surat, pengelolaan pengaduan, proses verifikasi administrasi, hingga pengunduhan dokumen yang telah disetujui.

Tabel 7. Pengujian Black Box Testing

Fitur	Skenario Pengujian	Keluaran yang Diharapkan	Hasil
Registrasi	Registrasi dengan data yang valid	Sistem mengirimkan kode OTP ke alamat email yang didaftarkan	Sesuai
Registrasi	Registrasi dengan format email tidak valid	Sistem menampilkan pesan kesalahan format email tidak valid	Sesuai
Login	Login dengan kredensial yang valid	Pengguna berhasil masuk ke halaman sesuai role	Sesuai
Pengajuan Surat	Mengajukan surat dengan data yang lengkap	Permohonan berhasil tersimpan dan tercatat dalam riwayat pengajuan	Sesuai
Pengajuan Surat	Mengunggah berkas dengan format yang tidak didukung	Sistem menampilkan pesan bahwa format berkas tidak didukung	Sesuai
Verifikasi Permohonan	Menyetujui permohonan surat	Status permohonan berubah menjadi disetujui	Sesuai
Unduh Surat	Mengunduh surat yang telah disetujui	Berkas surat (dengan QR Code) berhasil diunduh	Sesuai
Pengaduan	Mengajukan pengaduan disertai file lampiran	Pengaduan beserta file lampiran berhasil tersimpan	Sesuai

Hasil pengujian menunjukkan seluruh skenario menghasilkan keluaran sesuai dengan yang diharapkan tanpa ditemukan kegagalan fungsi. Seluruh fitur utama, termasuk mekanisme pengajuan surat yang terintegrasi dengan Amazon SQS, AWS Lambda, dan Amazon S3, dapat berjalan sebagaimana rancangan sistem. Hasil tersebut menunjukkan bahwa kebutuhan fungsional yang telah diidentifikasi pada tahap analisis berhasil diimplementasikan pada sistem.

User Acceptance Testing (UAT)

User Acceptance Testing dilakukan untuk mengetahui tingkat penerimaan pengguna terhadap website desa digital menggunakan kuesioner berbasis skala Likert. Pengujian melibatkan dua kelompok responden, yaitu masyarakat sebagai pengguna layanan dan perangkat desa sebagai pengelola sistem.

Tabel 8. Hasil Kuesioner Masyarakat

No.	Pertanyaan	SS (5)	S (4)	N (3)	TS (2)	STS (1)	Rata-rata
1.	Saya dapat menggunakan website ini tanpa merasa kebingungan	7	8	0	0	0	4,47
2.	Masuk ke akun (<i>login</i>) dan mendaftar (registrasi) pada <i>website</i> ini mudah dilakukan	10	5	0	0	0	4,67
3.	Mengajukan permohonan layanan surat melalui <i>website</i> ini mudah dilakukan	7	5	3	0	0	4,27
4.	Tampilan pada <i>website</i> ini terlihat menarik dan profesional	7	6	2	0	0	4,33
5.	Informasi yang ditampilkan tersusun dengan rapi dan mudah dibaca	8	7	0	0	0	4,53
6.	Informasi profil desa, berita, dan pengumuman tersedia dengan lengkap dan mudah ditemukan	9	5	1	0	0	4,53
7.	Informasi persyaratan pengajuan layanan administrasi tersedia dengan jelas	9	3	3	0	0	4,40
8.	<i>Website</i> ini membantu saya mendapatkan informasi tentang desa dengan lebih mudah	9	5	1	0	0	4,53
9.	Mengajukan permohonan surat lewat <i>website</i> ini lebih praktis dibanding datang langsung ke kantor desa	7	5	3	0	0	4,27
10.	Secara keseluruhan, <i>website</i> ini bermanfaat bagi masyarakat	9	4	2	0	0	4,47
Rata-rata Keseluruhan							4.45

Hasil pengujian menunjukkan rata-rata penilaian sebesar 4,45 dari kelompok masyarakat dan 5,00 dari perangkat desa, yang keduanya termasuk dalam kategori Sangat Baik. Nilai tertinggi pada kelompok masyarakat diperoleh pada aspek kemudahan registrasi dan login (4,67), sedangkan nilai terendah berada pada aspek kemudahan pengajuan surat dan kepraktisan dibandingkan layanan manual (4,27). Meskipun demikian, seluruh indikator memperoleh nilai di atas 4,00 sehingga menunjukkan bahwa sistem mudah digunakan dan dapat diterima dengan baik oleh pengguna.

Stress Testing

Stress Testing dilakukan menggunakan Apache JMeter untuk mengevaluasi kemampuan sistem dalam menangani peningkatan jumlah pengguna yang mengakses layanan secara bersamaan.

Tabel 9. Rekapitulasi Hasil Stress Test

Skenario	Total Request	Avg Response (ms)	90% Line (ms)	Max (ms)	Throughput	Error Rate
100 user	1.000	2.193	3.630	9.577	23,4/dtk	0,00%
200 user	2.000	4.538	6.766	12.549	23,3/dtk	0,00%
300 user	3.000	6.833	10.100	13.953	23,3/dtk	0,00%
400 user	4.000	8.460	12.540	14.940	23,9/dtk	0,00%
500 user	5.000	10.453	16.411	18.388	23,9/dtk	0,02%

Pengujian dilakukan dengan beban bertahap mulai dari 100 hingga 500 pengguna. Dari total 15.000 request, sistem hanya mengalami 1 kegagalan pada skenario 500 pengguna, sehingga menghasilkan error rate maksimum sebesar 0,02% tanpa menyebabkan downtime. Nilai throughput tetap berada pada kisaran 23 request per detik, sedangkan rata-rata waktu respons meningkat secara bertahap seiring bertambahnya jumlah pengguna. Hasil tersebut menunjukkan bahwa infrastruktur berbasis Amazon EC2 mampu mempertahankan stabilitas layanan meskipun menerima beban akses yang tinggi selama pengujian.

Security Testing

Security Testing dilakukan untuk mengevaluasi keamanan sistem dari sisi konfigurasi komunikasi, kerentanan aplikasi web, dan penerapan HTTP security header menggunakan SSL Labs, OWASP ZAP, dan Mozilla HTTP Observatory.

Tabel 10. Hasil pengujian Security testing

Tools	Aspek yang Diuji	Hasil
SSL Labs	Konfigurasi SSL/TLS	Grade A (mendukung TLS 1.2/1.3, tanpa kerentanan Heartbleed)
OWASP ZAP	Kerentanan aplikasi web	16 temuan (0 High, 3 Medium, 7 Low, 6 Informational)
Mozilla HTTP Observatory	Konfigurasi HTTP security header	Skor 25/100 (D-)

Hasil pengujian menunjukkan konfigurasi SSL/TLS memperoleh Grade A, yang menandakan komunikasi data telah menggunakan konfigurasi enkripsi yang baik. Pengujian menggunakan OWASP ZAP tidak menemukan

kerentanan dengan tingkat risiko High, meskipun masih terdapat beberapa temuan berkategori Medium, Low, dan Informational. Sementara itu, Mozilla HTTP Observatory memberikan skor 25/100 (D-), yang menunjukkan masih terdapat beberapa HTTP security header yang belum dikonfigurasi, seperti Content Security Policy (CSP) dan HTTP Strict Transport Security (HSTS). Temuan tersebut menjadi rekomendasi pengembangan untuk meningkatkan keamanan aplikasi pada implementasi berikutnya.

3.5. Maintenance (Pemeliharaan)

Tahap pemeliharaan dilakukan untuk menjaga keberlangsungan operasional aplikasi dan infrastruktur setelah sistem diimplementasikan. Pengelolaan kode program dilakukan menggunakan GitHub sebagai repositori utama sehingga setiap perubahan terdokumentasi melalui riwayat commit dan dapat diterapkan ke server menggunakan mekanisme git pull melalui koneksi SSH ke instance Amazon EC2.

Selain pemeliharaan aplikasi, pengelolaan infrastruktur dilakukan melalui AWS Management Console untuk memantau kondisi layanan, mengelola sumber daya cloud, serta melakukan pencadangan data pada Amazon RDS dan Amazon S3. Pendekatan tersebut memudahkan proses pemeliharaan karena administrasi infrastruktur dapat dilakukan secara terpusat tanpa bergantung pada keberadaan server fisik di lingkungan kantor desa.

4. Kesimpulan

Penelitian ini berhasil membangun website desa digital berbasis IaaS menggunakan enam layanan Amazon Web Services (EC2, RDS, S3, Lambda, SQS, dan SNS), dilengkapi mekanisme penandatanganan digital otomatis melalui AWS Lambda yang menyisipkan QR Code verifikasi ke dokumen surat tanpa memerlukan tanda tangan fisik. Hasil pengujian menunjukkan seluruh kebutuhan fungsional terpenuhi, penerimaan pengguna berada pada kategori Sangat Baik, sistem mampu menangani beban hingga 500 pengguna secara bersamaan dengan tingkat kegagalan sangat rendah, serta konfigurasi keamanan yang baik meski header keamanan HTTP masih perlu disempurnakan. Temuan ini menunjukkan bahwa arsitektur IaaS berbasis AWS dapat menjadi rujukan teknis bagi pemerintah desa lain dalam mengotomatisasi layanan administrasi publik, dengan saran pengembangan lebih lanjut berupa penguatan keamanan aplikasi dan kajian efisiensi biaya operasional untuk keberlanjutan jangka panjang.

Referensi

- [1] A. Arief and M. Y. Abbas, "Kajian Literatur (Systematic Literature Review): Kendala Penerapan Sistem Pemerintahan Berbasis Elektronik (SPBE)," *PROtek J. Ilm. Tek. Elektro*, vol. 8, no. 1, pp. 1–6, 2021.
- [2] I. F. Ambarsari, N. Azizah, A. Ansori, Y. F. Al-faruq, and K. K. Fahrozi, "Digitalisasi Informasi dan Peningkatan Kualitas Pelayanan Publik Transformasi Desa Digital Melalui Pengembangan Website Desa Klatakan," *I-Com Indones. Community J.*, vol. 4, no. 1, pp. 396–405, 2024, doi: 10.33379/icom.v4i1.4041.
- [3] S. Alviana and Y. Sutisnawati, "Implementasi Penyimpanan Data Menggunakan Nextcloud Berbasis Komputasi Awan," *INTEK J. Inform. dan Teknol. Inf.*, vol. 7, no. 2, pp. 12–17, 2024, doi: 10.37729/intek.v7i2.5627.
- [4] H. S. Jelita Ritonga, Suendri, "IMPLEMENTASI CLOUD COMPUTING PADA SISTEM INFORMASI MANAJEMEN PELAYANAN KANTOR DESA ULUMAMIS SITUNGGALING," *J. Sist. Inf. Bisnis*, vol. 6, no. 1, pp. 99–111, 2025, doi: <https://doi.org/10.55122/junsibi.v6i1.1588>.
- [5] R. L. Hakim, "TANTANGAN IMPLEMENTASI DIGITALISASI DESA DI INDONESIA : TINJAUAN LITERATUR," *J. PUBLIC Adm. POLICY*, vol. 1, no. 2, pp. 50–61, 2026, [Online]. Available: <https://journal.sti pancamargapalu.ac.id/index.php/patriot/article/view/90>
- [6] M. Purnandi, A. Ariska, and M. B. Wibawa, "Implementasi Cloud Computing Dengan Metode Infrastructure as a Service Pada Dinas Komunikasi, Informatika dan Persandian Kabupaten Pidie," *J. Informatics Comput. Sci.*, vol. 10, no. 1, pp. 1–8, 2024, [Online]. Available: <https://jurnal.uui.ac.id/index.php/jics/article/view/3887>
- [7] M. Younus, E. P. Purnomo, A. Nurmandi, D. Mutiarin, and H. A. Manaf, "Analyzing the trend of government support for cloud computing usage in e - government architecture," *J. Cloud Comput.*, 2025, doi: 10.1186/s13677-025-00735-y.
- [8] M. Syarifuddin, A. M. A. K. Parewe, J. Simarmata, and I. F. Ashari, *Cloud Computing : Konsep dan Implementasi*, no. May. 2023. [Online]. Available: <https://www.researchgate.net/publication/370815388>
- [9] A. Zaki *et al.*, "Implementasi Cloud Computing Berbasis Software as a Service (SaaS) Menggunakan OwnCloud Untuk Pengolahan Data Mahasiswa Sistem Informasi UINSU," *JUKTISI J. Komput. Teknol. Inf. Sist. Komput.*, vol. 1, no. 3, pp. 179–184, 2023.
- [10] I. U. Sari, M. Tahir, and E. Vincentius, "Analisis Komparatif Layanan Cloud : Microsoft Azure , Aws , Dan Google Cloud Platform (GCP)," *J. Informasi, Sains dan Teknol.*, vol. 6, no. 2, pp. 30–43, 2023.
- [11] C. B. Gea, K. J. D. Lase, and M. Syamsudin, "Implementasi Virtual Private Server untuk Mini Hosting," *J. InFact Sains dan Komput.*, vol. 7, no. 02, pp. 5–9, 2023, doi: <https://doi.org/10.61179/jurnalinfact.v7i01.402> Implementasi.
- [12] N. Sasongko and I. Afrianto, "Tinjauan Literatur: Performa Komputasi Awan Amazon Web Services (AWS)," no. February, pp. 1–6, 2023.
- [13] Y. Harimurti and D. Udariansyah, "Implementasi Service EC2 & S3 Amazon Web Service Pada Niche Blog Menggunakan Metode SDLC," vol. 4, no. 2, pp. 675–685, 2023, doi: 10.30865/klik.v4i2.1192.
- [14] A. Febrian Kasmar, W. Wahyuna, F. Sukma, and S. Amalia, "Implementasi Sistem Keamanan Dan High Availability Pada Cloud Server Menggunakan Amazon Web Services (Aws)," *J. Teknoif Tek. Inform. Inst. Teknol. Padang*, vol. 13, no. 1, pp. 40–47, 2025, doi: 10.21063/jtif.2025.v13.1.40-47.
- [15] C. W. Kusuma, D. G. Lukito, and I. M. Suraharta, "Perancangan Sistem ETLE Berbasis Web dengan Metode SDLC (System Development Life Cycle) Waterfall (Studi Kasus: Kota Tegal)," *J. Sos. dan Teknol.*, pp. 586–594, 2024.
- [16] E. O. W. Susanti, I. Ummami, and Winarti, "RANCANG BANGUN SISTEM INFORMASI JURNAL PERKULIAHAN BERBASIS WEB GUNA MENINGKATKAN EFEKTIVITAS PEMBELAJARAN," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 4, no. 1, pp. 386–393, 2022, doi: <https://doi.org/10.47233/jteksis.v4i2.556>.

- [17] S. W. Ramdany, S. A. Kaidar, B. Aguchino, C. A. A. Putri, and R. Anggie, "Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web," *J. Ind. Eng. Syst. (JIES)*, vol. 5, no. 1, 2024.
- [18] M. P. Hasanah, Fitria Nur and M. P. Rahmania Sri Untari, *BUKU AJAR REKAYASA PERANGKAT LUNAK*. UMSIDA Press, 2020.
