Sistem Informasi Jurnalis Warga Berbasis Web Terintegrasi Asisten Virtual AI Berbasis *Retrieval-Augmented Generation* (RAG)

Agustiningtyas Dyahretno Supriyani*[1], Nur Latifah Dwi Mutiara Sari[2]
[1]Informatika, Universitas PGRI Semarang, Kota Semarang, dyahretnosupriyani@gmail.com
[2]Informatika, Universitas PGRI Semarang, Kota Semarang, nurlatifah@upgris.ac.id

Abstract.

Managing digital journalism often takes too much time. Therefore, this study created a web-based Citizen Journalist Information System equipped with an AI Virtual Assistant for a campus television newsroom. The system was built using Laravel with a Prototyping method, implementing Role-Based Access Control (RBAC) simply to separate admin and citizen journalist access. To maintain order, the system requires login authentication, logs manuscript history, and uses a basic curation workflow (pending, rejected, published). The virtual assistant relies on a Retrieval-Augmented Generation (RAG) architecture on the Laravel backend to pull local news data. Meanwhile, the Google Gemini 2.5 Flash LLM processes the text generation in the cloud. The chatbot is restricted by the Google Search Grounding feature to only read from the verified local news database, which prevents the AI from hallucinating. The final result is a news portal and analytics dashboard sufficient to speed up newsroom operations and deliver news.

Keywords: CMS; Citizen Journalism; Laravel; Retrieval-Augmented Generation; Large Language Model; Chatbot

Abstrak

Pengelolaan informasi dalam jurnalisme digital sering kali memakan waktu. Oleh karena itu, penelitian ini membuat Sistem Informasi Jurnalis Warga berbasis web yang dilengkapi Asisten Virtual AI untuk ruang redaksi televisi kampus. Sistem ini dibuat menggunakan Laravel dengan metode Prototyping dan menerapkan *Role-Based Access Control* (RBAC) sekadar untuk membatasi akses antara admin dan jurnalis warga. Untuk menjaga ketertiban, sistem mewajibkan login, mencatat riwayat naskah, dan memakai alur kurasi dasar (*pending, ditolak, published*). Asisten virtualnya menggunakan arsitektur *Retrieval-Augmented Generation* (RAG) di *backend* Laravel untuk mengambil teks berita lokal. Sementara itu, LLM Google Gemini 2.5 Flash memproses jawabannya di *cloud*. *Chatbot* ini dibatasi oleh fitur *Google Search Grounding* agar hanya mengambil data dari database lokal, sehingga mencegah AI mengarang bebas (halusinasi). Hasil akhirnya adalah portal berita dan *dashboard* analitik yang cukup untuk mempercepat kerja redaksi dan menyajikan berita.

Kata Kunci: CMS; Jurnalis Warga; Laravel; *Retrieval-Augmented Generation*; *Large Language Model*; *Chatbot*

1. Pendahuluan

Perkembangan teknologi digital telah mengubah lanskap jurnalisme secara fundamental. Masyarakat kini tidak hanya berperan sebagai konsumen informasi, tetapi juga sebagai produsen berita melalui konsep jurnalisme warga (*citizen journalism*). Namun, keterbukaan partisipasi ini memunculkan tantangan serius berupa maraknya disinformasi, hoaks, dan konten yang tidak terverifikasi di ruang publik digital [1][2]. Institusi pers, khususnya lembaga media kampus, membutuhkan platform pengelolaan informasi yang tidak hanya mampu menampung kontribusi warga secara efisien, tetapi juga mampu menjamin kredibilitas dan akuntabilitas setiap informasi yang dipublikasikan kepada publik [1]. Oleh karena itu, kehadiran platform digital yang mampu memfasilitasi partisipasi warga sekaligus menjamin validitas informasi menjadi kebutuhan yang mendesak bagi lembaga pers kampus.

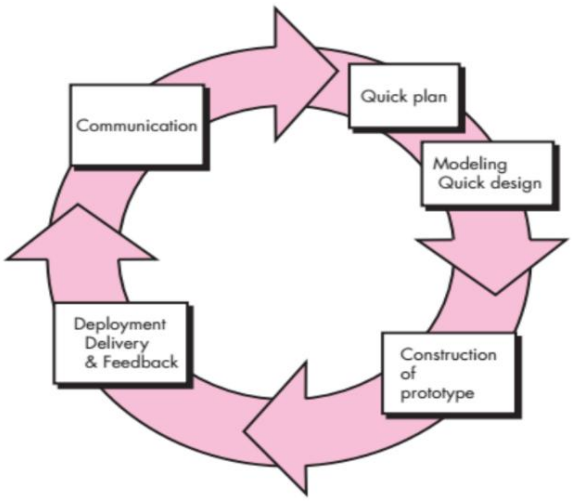
Berbagai upaya telah dilakukan untuk menjawab tantangan tersebut. Penerapan *Content Management System* (CMS) terpusat terbukti mampu meningkatkan efisiensi tata

kelola konten redaksional secara terstruktur [3]. Di sisi lain, perkembangan kecerdasan buatan, khususnya *Large Language Model* (LLM) dan arsitektur *Retrieval-Augmented Generation* (RAG), membuka peluang baru dalam pengembangan asisten virtual yang mampu menjawab pertanyaan pengguna berdasarkan sumber informasi yang terverifikasi [3][4]. Beberapa penelitian telah mengintegrasikan RAG ke dalam sistem berbasis web untuk membangun *chatbot* akademik [5] maupun asisten layanan publik [6][7]. Namun, sebagian besar implementasi tersebut belum mengintegrasikan mekanisme kurasi editorial yang ketat sebagai lapisan kontrol kualitas sumber pengetahuan AI, sehingga risiko halusinasi informasi masih menjadi celah yang belum sepenuhnya diatasi [4][5]. Kondisi ini menunjukkan perlunya pendekatan baru yang menggabungkan mekanisme kontrol kualitas konten dengan kemampuan AI secara terpadu.

Celah inilah yang menjadi landasan pengembangan sistem pada penelitian ini. Berbeda dari sistem-sistem sebelumnya, pendekatan yang ditawarkan mengintegrasikan alur kurasi redaksional berbasis RBAC (*Role-Based Access Control*) secara langsung ke dalam arsitektur RAG, sehingga LLM hanya dapat mengakses dokumen berita yang telah melewati validasi editorial berstatus *published*. Dengan demikian, basis pengetahuan asisten virtual dibangun bukan dari data mentah, melainkan dari informasi yang telah terjamin kredibilitasnya oleh redaksi. Integrasi ini menjadi kebaruan yang membedakan sistem yang dikembangkan dari penelitian-penelitian sebelumnya di bidang portal berita berbasis web maupun *chatbot* berbasis RAG.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan merancang dan mengimplementasikan Sistem Informasi Jurnalis Warga berbasis web yang terintegrasi dengan asisten virtual AI menggunakan arsitektur RAG dan LLM Google Gemini 2.5 Flash pada ruang redaksi televisi kampus. Sistem dikembangkan menggunakan kerangka kerja Laravel dengan pendekatan *System Development Life Cycle* (SDLC) model Prototyping, yang memungkinkan penyesuaian fungsional secara iteratif sesuai kebutuhan operasional redaksi.

2. Metode



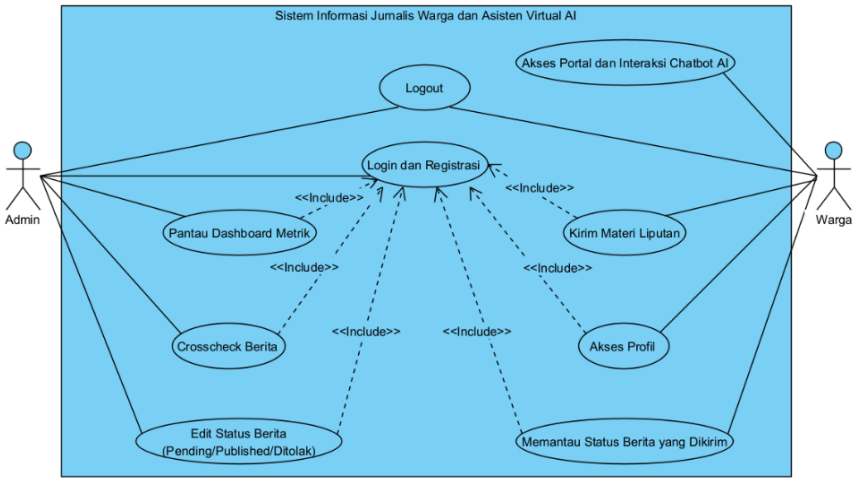
Gambar 19. Siklus Metode Prototyping [8]

Pada gambar 1 merupakan visualisi pengembangan sistem pada penelitian ini yaitu menggunakan pendekatan *System Development Life Cycle* (SDLC) dengan model Prototyping sebagai kerangka kerjanya. Model ini dipilih karena sifatnya yang iteratif dan adaptif, memungkinkan pengembang untuk melakukan penyesuaian fungsional secara bertahap berdasarkan umpan balik dari pihak redaksi kampus sebelum sistem dirilis secara penuh [9][10]. Secara operasional, proses pengembangan terbagi ke dalam lima fase utama yang dijelaskan sebagai berikut.

a. Fase 1: Pengumpulan Kebutuhan (*Requirements Gathering*)

Pada fase ini dilakukan identifikasi kebutuhan perangkat keras, perangkat lunak, dan kebutuhan fungsional dari dua entitas pengguna utama, yaitu Warga dan

Admin. Kebutuhan yang ditetapkan mencakup pembatasan hak akses berbasis peran (RBAC), perancangan basis data relasional dengan tiga status kurasi (pending, ditolak, published), serta mekanisme integrasi API kecerdasan buatan untuk memproses kueri asisten virtual.



Gambar 20. Use Case Diagram Sistem Informasi Jurnalis Warga

Pada gambar 2 merupakan visualisasi batasan fungsionalitas dan interaksi yang dapat dilakukan oleh pengguna Warga dan Admin di dalam sistem.

b. Fase 2: Membangun Prototipe (*Building Prototyping*)

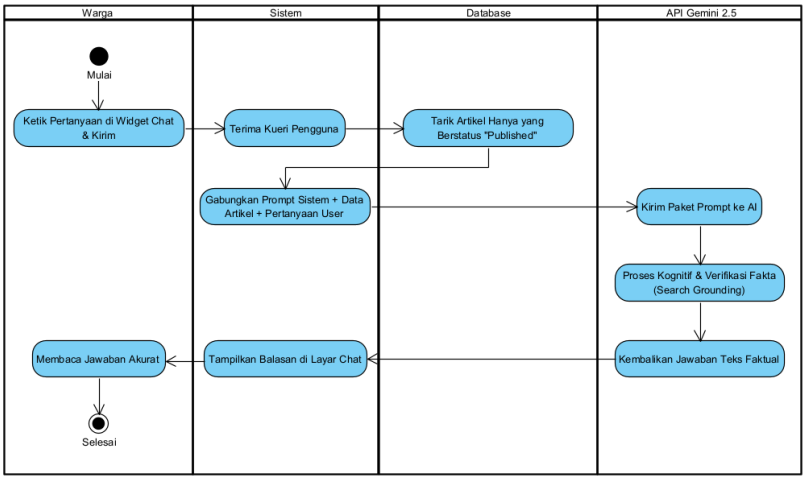
Rancangan awal antarmuka pengguna dibangun menggunakan *framework* utilitas Tailwind CSS untuk memastikan tampilan portal berita dan *dashboard* CMS bersifat responsif di berbagai ukuran perangkat. Pada fase ini juga dirancang alur navigasi sistem secara keseluruhan sebelum pengkodean inti dimulai.

c. Fase 3: Evaluasi Prototipe (*Evaluating Prototyping*)

Rancangan awal dievaluasi untuk memastikan kesesuaian alur autentikasi, mekanisme kurasi redaksional, dan integrasi antarmuka asisten virtual dengan kebutuhan yang telah ditetapkan. Temuan pada fase ini digunakan sebagai dasar perbaikan sebelum memasuki tahap pengkodean.

d. Fase 4: Mengodekan Sistem (*Coding the System*)

Implementasi sistem dibangun menggunakan *framework* Laravel berbasis PHP dengan MySQL sebagai sistem manajemen basis data relasional. Pada fase inilah logika RAG diimplementasikan secara komputasional. *Controller* Laravel diprogram untuk menyaring artikel berstatus *published* dari basis data, menggabungkannya dengan instruksi sistem melalui teknik *Prompt Engineering* [11][12], lalu mengirimkan keseluruhan paket tersebut ke API Google Gemini 2.5 Flash yang dikonfigurasi dengan fitur *Google Search Grounding* untuk verifikasi fakta secara langsung [5]. Arsitektur RAG pada sistem ini beroperasi dalam tiga fase kerja internal, yaitu fase Retrieval (penyaringan dokumen *published* dari MySQL), fase *Augmented* (perakitan konteks berita dengan pertanyaan pengguna), dan fase Generation (pengiriman *augmented prompt* ke API LLM untuk menghasilkan jawaban akhir) [3][4]. Alur kerja lengkap arsitektur ini digambarkan pada Gambar 3.



Gambar 21. Activity Diagram alur tanya jawab Asisten Virtual AI berbasis RAG

Pada gambar 3 merupakan alur sistematis yang berjalan dari kueri awal pengguna, penyaringan konteks berita lokal yang terverifikasi, hingga proses generasi jawaban oleh asisten virtual.

e. Fase 5: Pengujian Sistem (*Testing the System*)

Pengujian sistem menggunakan metode *Black Box Testing*, yaitu pendekatan yang menitikberatkan pada fungsionalitas antarmuka tanpa menelaah struktur kode internal. Tiga skenario diuji: proses registrasi dan autentikasi sesi pengguna, operasi CRUD pada pengelolaan status berita oleh admin, serta akurasi dan objektivitas jawaban asisten virtual AI terhadap pertanyaan berbasis konteks berita lokal. Untuk asisten virtual, dilakukan simulasi prompt testing dengan berbagai variasi pertanyaan untuk memastikan skema RAG dan *Google Search Grounding* efektif mencegah halusinasi AI. Hasilnya, seluruh fitur platform jurnalis warga dan integrasi asisten virtual terbukti berjalan lancar serta memenuhi kebutuhan operasional redaksi. Spesifikasi perangkat lunak yang digunakan pada penelitian ini disajikan pada Tabel 1.

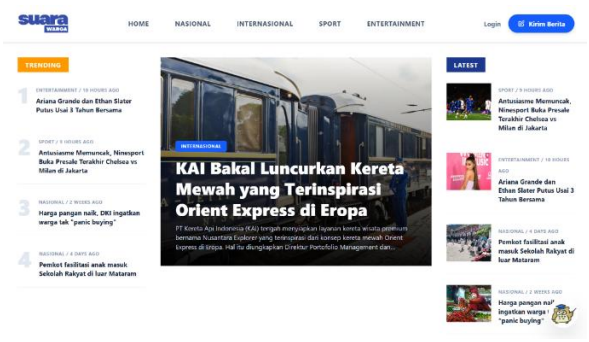
Tabel 6. Spesifikasi perangkat lunak pengembangan sistem

Komponen	Teknologi
Backend Framework	Laravel 10 (PHP)
Database	MySQL
Frontend	Tailwind
AI Model	Google Gemini 2.5 Flash API
Arsitektur AI	*Retrieval-Augmented Generation* (RAG)
Fitur Verifikasi	*Google Search Grounding*

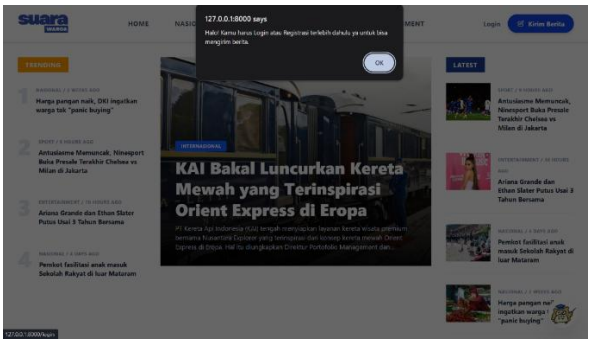
3. Hasil dan Pembahasan

3.1. Hasil Implementasi Sistem

Pengembangan sistem menghasilkan sebuah portal berita jurnalis warga berbasis web yang terintegrasi dengan asisten virtual AI. Sistem dapat diakses oleh dua entitas pengguna dengan hak akses yang berbeda, yaitu Warga dan Admin, sesuai dengan mekanisme RBAC yang telah dirancang. Hasil implementasi antarmuka sistem disajikan pada gambar-gambar berikut.



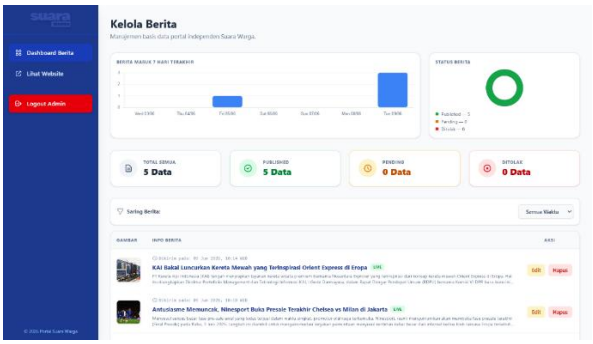
Gambar 22. Tampilan halaman utama portal berita jurnalis warga



Gambar 23. Tampilan restriksi akses dan formulir pengiriman materi liputan

Pada gambar 4 merupakan antarmuka halaman beranda portal berita bagi publik yang menampilkan menu navigasi kategori dan sorotan informasi terkini. Halaman utama portal dirancang menggunakan tata letak *grid* responsif yang menampilkan berita terkini dan berita trending, dilengkapi menu navigasi berdasarkan kategori seperti Nasional, Internasional, Sport, dan Entertainment. Sementara, gambar 5 merupakan halaman validasi keamanan yang memastikan hanya pengguna terdaftar dan berstatus login yang diizinkan untuk mengirimkan naskah liputan. Fitur pengiriman materi liputan hanya dapat diakses oleh pengguna yang telah terautentikasi. Apabila pengunjung yang belum login mencoba mengakses fitur ini, sistem secara otomatis menampilkan peringatan dan mengarahkan pengguna ke halaman registrasi atau login.

Gambar 24. Tampilan interaksi asisten virtual AI berbasis RAG



Gambar 25. Tampilan dashboard admin dan analitik visual

Pada gambar 6 merupakan *widget* interaktif bagi pengunjung untuk melakukan tanya jawab dan memperoleh informasi faktual secara langsung dari kecerdasan buatan. Asisten virtual AI tersedia sebagai *widget* mengambang yang dapat diakses oleh pengunjung tanpa perlu login. Pengguna dapat mengetikkan pertanyaan menggunakan bahasa sehari-hari dan sistem akan merespons dengan jawaban faktual berbasis konteks berita lokal yang telah terverifikasi. Sementara gambar 7 merupakan halaman dasbord bagi pihak redaksi untuk memantau analitik portal sekaligus mengevaluasi kelayakan naskah berita dari jurnalis warga. *Dashboard* manajerial CMS menyediakan panel analitik visual berupa grafik batang intensitas berita harian dan grafik cincin proporsi status berita, serta tabel manajemen berita yang dilengkapi fitur penyaringan berdasarkan rentang waktu.

3.2. Pembahasan

3.2.1 Arsitektur Sistem dan Tata Kelola Redaksi

Penerapan *framework* Laravel dengan pola *Model-View-Controller* (MVC) berhasil menciptakan pemisahan yang jelas antara logika basis data, antarmuka pengguna, dan proses komputasi *backend* [13][3]. Kombinasi Laravel dan Tailwind CSS menghasilkan antarmuka yang responsif dan adaptif di berbagai ukuran perangkat [6], sekaligus mempersingkat waktu respons halaman ketika memuat aset dinamis [14]. Dengan demikian, kombinasi Laravel dan

Tailwind CSS terbukti menjadi fondasi teknis yang solid untuk membangun antarmuka sistem yang efisien dan mudah digunakan.

Dari sisi tata kelola redaksi, sistem berhasil mengimplementasikan mekanisme RBAC yang membatasi hak akses secara tegas antara Warga dan Admin [2]. Setiap naskah yang dikirimkan warga secara otomatis berstatus pending dan wajib melewati proses kurasi admin sebelum dapat dipublikasikan. Penerapan tiga status kurasi (*pending, published, ditolak*) ini sejalan dengan standar kontrol kualitas jurnalistik yang berlaku pada platform CMS terpusat [2][15][16], sekaligus menjadi pembeda utama dibanding sistem portal berita berbasis web yang telah dikembangkan sebelumnya [12]. Hal ini menegaskan bahwa mekanisme RBAC dan kurasi editorial merupakan komponen kritis yang tidak dapat dipisahkan dari sistem informasi jurnalisme warga yang bertanggung jawab.

3.2.2 Integrasi Arsitektur RAG dan LLM

Inovasi utama sistem ini terletak pada implementasi asisten virtual yang ditopang arsitektur RAG dan LLM Google Gemini 2.5 Flash. Penting untuk dipahami bahwa RAG dan LLM merupakan dua entitas berbeda yang saling melengkapi dalam satu alur arsitektur [3]. RAG beroperasi sepenuhnya di sisi *backend* Laravel lokal melalui tiga fase kerja, yaitu fase Retrieval yang menyaring dokumen *published* dari basis data MySQL [17], fase *Augmented* yang merakit konteks berita lokal dengan pertanyaan pengguna melalui teknik *Prompt Engineering* [11][12], dan fase *Generation* yang mengirimkan augmented prompt ke API LLM untuk diolah menjadi jawaban akhir. Sementara itu, LLM Google Gemini 2.5 Flash berposisi di sisi *cloud* sebagai mesin penalaran bahasa alami yang mengeksekusi instruksi dan menghasilkan respons yang luwes dan mudah dipahami pengguna [15]. Pemisahan peran yang tegas antara RAG dan LLM ini menjadikan sistem lebih transparan, dapat dikontrol, dan tidak bergantung sepenuhnya pada pengetahuan bawaan model AI.

Pemisahan peran ini memberikan keunggulan yang signifikan dibanding *chatbot* konvensional yang menjawab berdasarkan pengetahuan bawaannya semata. Dengan membatasi sumber pengetahuan RAG hanya pada dokumen yang telah melewati kurasi editorial, sistem ini secara efektif mengikat LLM agar hanya menjawab berdasarkan konteks berita lokal yang kredibel [4]. Keandalan arsitektur ini semakin diperkuat dengan aktifnya fitur *Google Search Grounding* yang memungkinkan verifikasi silang terhadap basis data internet terkini sebelum jawaban akhir dirumuskan [5][7]. Hasil pengujian menunjukkan bahwa alur logika ini secara nyata mampu menekan tingkat halusinasi AI, sehingga informasi yang disajikan kepada pengunjung bersifat faktual dan relevan [4][5]. Hal ini sejalan dengan temuan Vidivelli et al. yang menyatakan bahwa integrasi RAG secara signifikan meningkatkan akurasi dan relevansi respons *chatbot* berbasis LLM [4], serta temuan Husain et al. yang membuktikan efektivitas RAG dalam membangun *chatbot* layanan akademik berbasis sumber terverifikasi [5]. Temuan-temuan ini memperkuat validitas pendekatan yang digunakan dalam penelitian ini dan menunjukkan bahwa arsitektur RAG yang terintegrasi dengan kurasi editorial merupakan solusi yang efektif untuk membangun asisten virtual berbasis berita lokal yang andal.

3.2.3 Kinerja Chatbot

Implementasi logika kecerdasan buatan dan eksekusi arsitektur RAG pada sistem ini dikelola secara terpusat di dalam lapisan *Controller* pada kerangka kerja Laravel. Secara spesifik, seluruh proses kognitif asisten virtual dieksekusi di dalam berkas *ChatbotController.php* melalui fungsi *sendMessage()*. Pada tahap *Retrieval*, sistem diinstruksikan untuk menarik maksimal 5 berita terbaru yang secara tegas memiliki status published dari basis data MySQL.

Data yang ditarik tersebut kemudian digabungkan dengan profil asisten bernama Sawa, penanda waktu aktual (*real-time*), serta lima aturan format balasan wajib ke dalam sebuah variabel konteks (*promptContext*). Potongan kode implementasi perakitan instruksi tersebut dapat dilihat pada sintaks berikut:

```php
$beritaDb = Berita::where('status', 'published')->orderBy('created_at', 'desc')->limit(5)->get();
        $teksBerita = "";

        if($beritaDb->count() > 0) {
            foreach($beritaDb as $b) {
                $teksBerita .= "- Judul: " . $b->judul . " | Ringkasan: " . $b->ringkasan . " | URL Baca: " . url('/berita/' . $b->id) . "\n";
            }
        } else {
            $teksBerita = "Belum ada berita di database lokal Suara Warga.";
        }

        $promptContext = "Instruksi Sistem: Kamu adalah Sawa, asisten virtual cerdas dari portal jurnalisme independen 'Suara Warga'. Hari ini: " . $tanggalHariIni . " (" . $tahunIni . ").\n\n";
        $promptContext .= "DATABASE LOKAL (Prioritaskan referensi dari sini jika ditanya berita):\n" . $teksBerita . "\n\n";

        $promptContext .= "ATURAN FORMAT BALASAN (WAJIB DIIKUTI):\n";
        $promptContext .= "1. Gaya Bahasa: Jawab LANGSUNG ke intinya. Gunakan gaya penulisan yang natural, humanis, dan ramah, layaknya seorang mahasiswa atau teman diskusi yang profesional.\n";
        $promptContext .= "2. Pencarian: Jika jawaban tidak ada di Database Lokal, cari di Google. Prioritaskan keakuratan fakta.\n";
        $promptContext .= "3. Format Link: Tuliskan URL aslinya secara polosan/mentah di akhir kalimat agar bisa diklik. DILARANG menggunakan markdown kurung siku.\n";
        $promptContext .= "4. Relevansi: Pastikan berita yang kamu sampaikan relevan dengan tahun " . $tahunIni . ".\n";
        $promptContext .= "5. TUGAS UTAMA (KIRIM BERITA): Jika ada user yang bertanya 'Bagaimana cara mengirim berita?', 'Syarat ngirim berita', atau semacamnya, kamu WAJIB menjawab:
                        'Halo! Untuk mengirim materi liputan, kamu wajib memiliki akun Jurnalis Warga ya. Silakan Daftar/Login terlebih dahulu, baca Ketentuan Pengiriman, lalu isi formnya. Kamu bisa
                        langsung klik tombol Kirim Berita di pojok kanan atas atau akses link ini: " . url('/kirim-berita') . " '\n\n";
        $promptContext .= "Pertanyaan User: " . $pesanUser;
```

Gambar 26. Kode implementasi perakitan instruksi (prompt context) asisten virtual

Potongan kode implementasi perakitan instruksi dan penyisipan data lokal pada sistem dapat dilihat pada Gambar 8. Setelah variabel *prompt* diperkaya (*augmented prompt*) dengan data lokal dan aturan sistem, aplikasi melanjutkan ke tahap Generation dengan mengirimkan paket data tersebut menuju endpoint API Google Gemini menggunakan permintaan HTTP. Pada pengiriman ini, fitur *Google Search Grounding* turut diaktifkan sebagai lapisan validasi faktual tambahan apabila data lokal tidak mencukupi.

3.2.4 Waktu Respons Chatbot

Dari segi kinerja operasional dan metrik pengalaman pengguna, asisten virtual terbukti memiliki tingkat efisiensi yang sangat baik. Pemilihan varian model Gemini 2.5 Flash memberikan dampak signifikan terhadap kecepatan pemrosesan *Natural Language Processing* (NLP).

Waktu respons (*response time*) keseluruhan, dihitung sejak kueri dikirimkan oleh pengguna pada antarmuka widget hingga balasan teks muncul di layar membutuhkan durasi rata-rata yang berkisar antara 1,5 hingga 3 detik. Kecepatan durasi waktu respons ini dipengaruhi oleh optimalisasi pada tiga titik proses utama:

a. Latensi Basis Data Lokal, yaitu waktu eksekusi kueri MySQL sangat singkat karena komputasi hanya dibatasi untuk mengambil 5 data berita teratas.
b. Pengiriman Muatan (Payload) API, yaitu ukuran teks konteks yang dikirim ke peladen *cloud* Google relatif ringan karena sistem hanya menyertakan ringkasan (*summary*) berita, bukan teks keseluruhan artikel.
c. Pemrosesan Kognitif LLM, yaitu model arsitektur Flash secara spesifik dirancang untuk mengeksekusi tugas-tugas dialog dengan latensi yang sangat rendah.

Melalui optimalisasi pada alur kerja operasional tersebut, sistem ini tidak hanya berhasil menekan risiko halusinasi AI berkat pembatasan sumber data, tetapi juga berhasil mempertahankan latensi percakapan yang seminimal mungkin. Hal ini memastikan pengguna tetap merasakan pengalaman interaksi yang dinamis, organik, dan seketika (*real-time*) tanpa jeda tunggu yang mengganggu.

3.2.5 Hasil Pengujian Fungsional

Pengujian dilakukan terhadap tiga skenario utama: keberhasilan proses registrasi dan autentikasi sesi pengguna, kelancaran operasi CRUD pada pengelolaan status berita oleh admin, serta keakuratan dan objektivitas respons asisten virtual AI dalam menjawab pertanyaan berdasarkan konteks berita lokal yang tersedia. Pengujian ini dieksekusi menggunakan metode *Black Box Testing* untuk mengevaluasi fungsionalitas antarmuka. Rangkuman hasil pengujian disajikan pada Tabel 2.

Tabel 7. Hasil pengujian fungsional (Black Box)

No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Akses fitur kirim berita tanpa login.	Sistem mewajibkan login.	Akses ditolak, dialihkan ke halaman autentikasi.	Valid

2.	Pengiriman naskah liputan oleh warga.	Mencatat riwayat naskah dengan alur kurasi dasar.	Data tersimpan dengan status "*pending*".	Valid
3.	Pengelolaan status berita oleh admin.	Status berita dapat diubah menjadi *published*.	Berita berhasil dipublikasikan ke portal utama.	Valid
4.	Tes respons asisten AI terkait berita.	Mengambil data dari database lokal.	Jawaban faktual sesuai konteks berita lokal.	Valid
5.	Tes asisten AI di luar topik/halusinasi.	Mencegah AI mengarang bebas (halusinasi).	Bot mematuhi batasan dan tidak berhalusinasi.	Valid

Berdasarkan Tabel 2, penerapan *Role-Based Access Control* (RBAC) berhasil membatasi akses antara admin dan jurnalis warga. Selain itu, alur kurasi dasar (*pending*, *ditolak*, *published*) terbukti menjaga ketertiban pengelolaan naskah. Pada sisi asisten virtual, pembatasan dengan fitur *Google Search Grounding* agar hanya mengambil data dari database lokal terbukti efektif mencegah AI mengarang bebas atau berhalusinasi.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan Sistem Informasi Jurnalis Warga berbasis web terintegrasi asisten virtual AI untuk ruang redaksi televisi kampus, dengan kesimpulan: Pertama, kombinasi *framework* Laravel (MVC) dan *Tailwind* CSS terbukti meningkatkan efisiensi waktu respons serta mendukung pengelolaan aset digital secara dinamis [13][3][14]. Kedua, implementasi RBAC dan alur kurasi (*pending, published, ditolak*) berhasil membentuk standar kontrol kualitas jurnalistik yang terukur [2][15][16], sekaligus mencegah tayangnya konten yang tidak terverifikasi [1]. Ketiga, integrasi arsitektur RAG dan LLM Google Gemini 2.5 Flash terbukti menekan risiko halusinasi AI secara signifikan [4][5], di mana dukungan *Google Search Grounding* memperkuat keandalan informasi faktual yang disajikan [5][7]. Keempat, metode Prototyping memberikan fleksibilitas iteratif dalam menyempurnakan logika API dan antarmuka agar selaras dengan kebutuhan operasional redaksi [10][9]. Pengembangan selanjutnya disarankan berfokus pada peningkatan skalabilitas basis data seiring bertambahnya volume liputan [18], penambahan fitur verifikasi keaslian video, serta penjajakan model AI terbaru guna mengoptimalkan latensi penalaran.

5. Referensi

[1] Maryam and A. Ramadhan, “Rancang Bangun Sistem Informasi Manajemen Pada Lembaga Pers Mahasiswa Cendekia,” 2023.

[2] A. Dwiyanti, M. R. Syahputra, and D. Widhiandono, “Bagaimana Penerapan CMS Platform Content Management dalam Penulisan Berita Di Tribun News Solo,” *J. Penelit. Komun.*, vol. 03, no. 04, pp. 45–50, 2023.

[3] D. Oreški and D. Vlahek, “Retrieval Augmented Generation in Large Language Models: Development of AI Chatbot for Student Support,” *CEUR Workshop Proc.*, vol. 3938, pp. 12–23, 2024.

[4] S. Vidivelli, M. Ramachandran, and A. Dharunbalaji, “Efficiency-Driven Custom Chatbot Development: Unleashing LangChain, RAG, and Performance-Optimized LLM Fusion,” *Comput. Mater. Contin.*, vol. 80, no. 2, pp. 2423–2442, 2024, doi: 10.32604/cmc.2024.054360.

[5] M. L. Husain, Y. Wibisono, and A. Anisyah, “Development of an Academic Services Chatbot Based on Retrieval-Augmented Generation (RAG),” *Brill. Res. Artif. Intell.*, vol. 5, no. 2, pp. 727–735, 2025, [Online]. Available: https://jurnal.itscience.org/index.php/brilliance/article/view/6719

[6] A. M. Nuzul, M. Nur, Y. Utomo, and T. Indrabulan, “Web-Based Public Relations Chatbot Using Large Language Models and the Retrieval-Augmented Generation,” *J. Informatics Comput. Eng. Res.*, vol. 2, no. 2, pp. 51–57, 2025, [Online]. Available:

https://poliupg.ac.id

[7] W. Scraping and S. Technology, "A Practical Application of Retrieval- Augmented Generation for Website-Based *Chatbot*s : Combining Web Scraping ," vol. 6, pp. 424–442, 2025.

[8] H. Angriani, Y. Saharaeni, and H. Hasniati, "Implementasi Metode Prototype pada Rancang Bangun Sistem Pendukung Keputusan Pemilihan Mahasiswa Berprestasi Berbasis Web," *J. INSYPRO Inf. Syst. Process.*, vol. 8, no. 1, pp. 1–7, 2023, [Online]. Available: http://journal.uinalauddin.ac.id/index.php/insypro

[9] M. Fauzan Ardiansyah, "Pengembangan Sistem Informasi Keanggotaan Online Berbasis Web Menggunakan Framework Laravel dengan Metode Prototype pada Asosiasi Inkindo," vol. 1, no. 2, pp. 266–271, 2023.

[10] A. P. Muhammad Rifky Akbar, Aidah Nur Fadhilah, "Pengembangan Sistem Informasi Csirt Berbasis Web Menggunakan Framework Laravel Dengan Metode Prototyping Pada Diskominfo Purwakarta," *J. Inform. Terpadu*, vol. 8, no. 1, pp. 1–7, 2026.

[11] V. K. Sikha, "Mastering Prompt Engineering: Optimizing Interaction with Generative AI Agents," *J. Eng. Appl. Sci. Technol.*, vol. 5, no. 6, pp. 1–8, 2023, doi: 10.47363/jeast/2023(5)e117.

[12] Agil Sirat, "Pengembangan Sistem Informasi Portal Berita Di Radar Jember Berbasis Website Menggunakan Framework Laravel 9," *Acta Univ. Agric. Silvic. Mendelianae Brun.*, vol. 53, no. 1, pp. 1–19, 2023, [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/245180/245180.pdf

[13] S. K. Rongali, "Natural Language Processing (NLP) in Artificial Intelligence," *World J. Adv. Res. Rev.*, vol. 25, no. 1, pp. 2515–2519, 2025, doi: 10.30574/wjarr.2025.25.1.0277.

[14] P. Surya, J. Permana, N. Luh, and P. Ika, "Implementasi *Chatbot* Berbasis RAG pada Sistem Informasi Rawat Jalan Klinik," vol. 5, no. April, 2026.

[15] A. A. Zaveri, J. Jaafar, E. Yafi, and S. Sarama, "Evolution of Requirements Engineering in Agile Methodology – Literature Review," *J. Eng. Technol. Appl. Phys.*, vol. 6, no. 2, pp. 57–65, 2024, doi: 10.33093/jetap.2024.6.2.9.

[16] R. Ahmad Fathurrohim, "Pengembangan Sistem *Content Management System* (Cms) Website Pondok Pesantren Menggunakan Framework Laravel," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3, 2025, doi: 10.23960/jitet.v13i3.7070.

[17] I. Radeva, I. Popchev, L. Doukovska, and M. Dimitrova, "Web Application for *Retrieval-Augmented Generation*: Implementation and Testing," *Electron.*, vol. 13, no. 7, 2024, doi: 10.3390/electronics13071361.

[18] M. Y. S. Umi Hasanah Purba, "Penggunaan AI Dalam Jurnalisme Dalam Peluang Dan Tantangan Bagi Jurnalis Era Web 3.0," *J. Nomosleca*, vol. 11, no. April, pp. 58–68, 2025, [Online]. Available: https://jurnal.unmer.ac.id/index.php/n/article/view/15657/7396