

# Pengembangan Sistem Manajemen Konten Berita Berbasis *Website* Dengan Integrasi *Gemini Ai* : Studi Kasus Di Desa Kebondalem Dengan Metode *Waterfall*

1<sup>st</sup> Muhammad Firdaus Zulkarnain  
Fakultas Teknik Informatika

Telkom University  
Purwokerto, Indonesia

[firdauszull@student.telkomuniversity.ac.id](mailto:firdauszull@student.telkomuniversity.ac.id)

2<sup>nd</sup> Ariq Cahya Wardhana, S.Kom., M.Kom  
Fakultas Teknik Informatika

Telkom University  
Purwokerto, Indonesia

[ariqcahya@telkomuniversity.ac.id](mailto:ariqcahya@telkomuniversity.ac.id)

**Abstrak** — Penelitian ini bertujuan mengembangkan sistem manajemen konten berita berbasis *website* dengan integrasi *Gemini AI* untuk mendukung otomatisasi dan personalisasi konten. Hasil penelitian menunjukkan bahwa sistem berfungsi sesuai spesifikasi dan mendukung distribusi informasi secara cepat, terstruktur, serta mudah diakses. Pengujian fungsional dan acceptance testing menunjukkan bahwa sistem mudah digunakan, antarmuka sederhana, dan mampu mempercepat proses penulisan berita. Penelitian ini memberikan kontribusi dalam penerapan AI untuk pengelolaan informasi desa dan dapat dijadikan model implementasi bagi wilayah lain.

**Kata kunci**— *Artificial intelligence, content management system, Gemini AI, website desa, waterfall, V-model.*

## I. PENDAHULUAN

Di era digital saat ini, kebutuhan akan sistem informasi yang andal dan efisien semakin meningkat, terutama bagi organisasi pemerintahan di tingkat lokal seperti desa. Penyebaran informasi yang cepat, akurat, dan tepat menjadi sangat penting dalam pelayanan publik, khususnya di desa-desa yang memiliki keterbatasan akses terhadap informasi langsung. Melalui sistem pengelolaan konten berbasis *website* yang terintegrasi dengan teknologi kecerdasan buatan (AI), pemerintah desa dapat menyampaikan informasi secara lebih efisien dan relevan dengan kebutuhan masyarakat [1]. Dengan demikian, sistem ini diharapkan dapat meningkatkan partisipasi masyarakat dalam kegiatan desa sekaligus mendukung penyebaran informasi yang cepat dan terstruktur[2]. Namun, penerapan sistem pengelolaan konten di lingkungan pedesaan tidak terlepas dari tantangan, seperti infrastruktur yang terbatas dan literasi digital yang rendah. Adopsi teknologi digital ini juga diharapkan dapat meningkatkan transparansi dan akuntabilitas dalam pelayanan publik.

Informasi desa yang disampaikan secara jelas dan terstruktur akan meningkatkan kepercayaan masyarakat

terhadap pemerintah desa[3]. Secara keseluruhan, pengembangan sistem manajemen konten berita berbasis *website* dengan integrasi *Gemini AI* diharapkan dapat membawa manfaat signifikan bagi Desa Kebondalem, baik dari segi efisiensi kinerja maupun kecepatan penyebaran informasi. Dengan kemajuan teknologi yang pesat, penerapan sistem ini dapat membantu desa merespon tantangan era digital dan menjadi model adaptif bagi pengelolaan informasi di wilayah pedesaan lainnya [4].

## II. KAJIAN TEORI

Penelitian yang berkaitan dengan pengembangan Content management System (CMS) berbasis *website* dan penggunaan Kecerdasan buatan (AI) dalam otomatisasi penulisan berita telah banyak dilakukan. Banyak diantaranya mampu memberikan inovasi signifikan dalam mengelola dan menyajikan konten secara efektif.

### A. Pengembangan Sistem Informasi Desa

Penelitian oleh Fahrur Rozi dan Tomi Listiawan menunjukkan bahwa membangun sistem informasi desa penting untuk meningkatkan kemampuan aparatur desa dalam mengelola data dan meningkatkan kinerja kerja. Dengan pelatihan dan penggunaan sistem berbasis *website*, perangkat desa bisa lebih baik dalam melakukan administrasi digital. Hal ini menjadi dasar bagi penelitian ini yang bertujuan sama, yaitu memperkuat kemampuan teknologi informasi di tingkat desa[5].

### B. Sistem Manajemen Berbasis Web

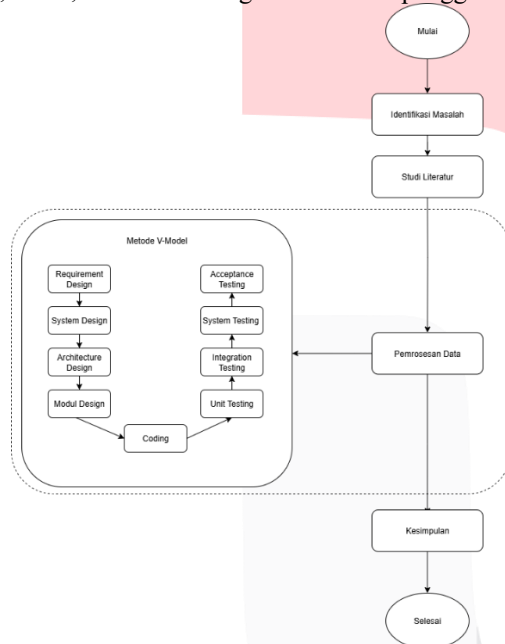
Meri Audrilia dan Arief Budiman membuat sistem informasi manajemen bengkel berbasis web untuk mengatasi masalah pengelolaan data secara manual. Mereka menggunakan metode rekayasa web dan membuktikan bahwa sistem yang mereka kembangkan efektif melalui pengujian ISO 25010. Studi ini menjadi acuan dalam pembuatan sistem yang baik dan efisien untuk mengelola data, serta menekankan pentingnya kualitas perangkat lunak dalam mendukung operasional[6].

### C. Sistem Informasi Penjualan Menggunakan V-Model

Penelitian oleh Arif Alfariy, Fenand, dan Muhammad Putra Muari menggunakan pendekatan *V-Model* dalam pengembangan sistem informasi penjualan. Model ini terbukti mampu menghasilkan sistem yang baik melalui beberapa tahapan validasi. Pendekatan ini juga digunakan dalam penelitian ini untuk memastikan sistem dapat berjalan dengan baik dan teruji secara kualitas[7].

## III. METODE

Dalam konteks integrasi *Gemini AI* ke dalam *website* desa, *V-Model* digunakan untuk memastikan bahwa proses perencanaan, analisis kebutuhan, perancangan, dan implementasi sistem dilakukan dengan mempertimbangkan integrasi AI sejak awal. Setiap fase pembangunan disertai langkah evaluasi untuk menjamin bahwa AI bekerja secara optimal, aman, dan sesuai dengan kebutuhan pengguna[8].



GAMBAR 1  
DIAGRAM ALIR PENELITIAN

### A. IDENTIFIKASI MASALAH

Pada awal penelitian dilakukan langkah untuk mengidentifikasi permasalahan terkait manajemen konten informasi di Desa Kebondalem. Berdasarkan masalah yang diuraikan, penelitian ini akan fokus pada pengembangan sistem pengelolaan konten berita dengan mengintegrasikan *Gemini AI* untuk meningkatkan efisiensi dan kualitas konten.

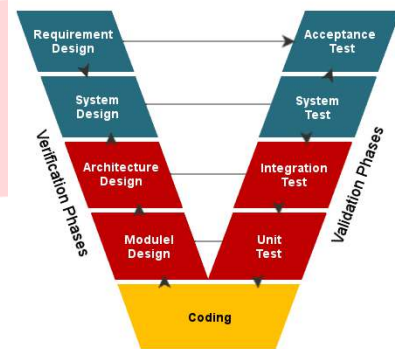
### B. STUDI LITERATUR

Pada tahap ini, Studi literatur dilakukan dengan mengumpulkan informasi dari majalah, buku, dan *website* terkait berbagai topik penting, seperti sistem manajemen konten (CMS), pengembangan *website* menggunakan Node.js dan React.js, mengintegrasikan AI ke dalam sistem manajemen konten, implementasi *Gemini AI*, metode *V-Model* dalam pengembangan perangkat lunak, dan praktik terbaik dalam pengembangan situs web modern. Tujuan dari tinjauan literatur adalah untuk memberikan pengetahuan dan wawasan yang diperlukan untuk mendukung penelitian ini. Melalui tinjauan literatur, peneliti dapat lebih memahami

konsep, teori yang mendasari, dan praktik terbaik terkait topik tersebut. Informasi dari literatur juga membantu peneliti mengidentifikasi kekuatan, kelemahan, dan tantangan yang mungkin timbul selama implementasi. Oleh karena itu, tahap tinjauan pustaka memegang peranan yang sangat penting dalam penyusunan penelitian ini, karena memberikan landasan yang kokoh bagi pengembangan sistem manajemen konten berbasis web dengan mengintegrasikan efisiensi dan efektivitas *Gemini AI*.

### C. PEMROSESAN DATA

Pada pemrosesan data ini terdapat alir diagram metode *waterfall* dengan pendekatan *V-Model*, dengan beberapa tahapan yang pertama adalah :



GAMBAR 2  
METODE V-MODEL

#### 1) Requirement Design

Tahap analisis kebutuhan sistem mencakup proses pemecahan masalah yang bertujuan untuk memahami sistem lebih dalam. Dalam analisis kebutuhan perangkat keras, fokusnya adalah pada perangkat keras yang dibutuhkan, seperti spesifikasi server dan infrastruktur jaringan. Sementara itu, dalam analisis kebutuhan perangkat lunak, perhatian lebih diberikan pada kerangka kerja, perpustakaan, database, penyimpanan, alat pengembangan, serta persyaratan *API* dan integrasi.

#### 2) System Design

Dalam penelitian ini, kami menggunakan *Unified Modeling Language* (UML) untuk memberikan gambaran mendalam tentang desain sistem manajemen konten berita. *Use Case Diagram* menunjukkan bagaimana pengguna berinteraksi dengan sistem, menunjukkan fitur yang tersedia, dan menentukan hak akses pengguna. Selain memodelkan struktur database dan hubungan antar entitas dalam sistem, *Class Diagram* menggambarkan definisi class dan method yang membentuk sistem manajemen konten. Diagram ini memberikan gambaran yang jelas tentang bagaimana data dan fitur berinteraksi satu sama lain.

Kami dapat menggambarkan dan menganalisis desain sistem manajemen konten berita secara terstruktur dengan menggunakan UML, Ini akan

mendukung perancangan dan implementasi sistem yang efisien.

### 3) *Architecture Design*

Untuk menjamin skalabilitas dan efisiensi, penelitian ini merancang arsitektur desain sistem manajemen konten berita modern. Untuk mengintegrasikan *Gemini AI*, kami juga menggunakan arsitektur *microservices*, yang memungkinkan pengembangan dan pemeliharaan sistem yang lebih terdistribusi dan fleksibel. Sistem manajemen konten berita dapat berjalan dengan baik, aman, dan mudah diperluas berkat desain yang terorganisir.

### 4) *Modul Design*

Modul desain sistem manajemen konten berita ini terdiri dari berbagai modul penting yang bekerja secara terintegrasi untuk memastikan kinerja sistem yang efisien dan aman. Terakhir, modul *Analytics* dan *Reporting* saling berinteraksi untuk membuat sistem yang terstruktur, efisien, dan mudah dikelola. Modul ini menawarkan fitur *analytics* dan *Reporting* untuk memantau performa konten serta memberikan wawasan yang berguna bagi pengelola sistem.

### 5) *Coding*

Untuk memastikan bahwa sistem manajemen konten berita ini bekerja dengan baik, proses coding melibatkan beberapa tahap pengembangan yang terorganisir. Kemudian, pengembangan frontend akan dilakukan menggunakan *React*. Pengembangan database juga penting untuk memastikan bahwa data berita dan pengguna disimpan dengan aman dan efisien, dan implementasi *API* akan membuat titik-titik akhir yang menghubungkan frontend dan backend, dan implementasi fitur sistem akan mencakup berbagai fungsi seperti manajemen berita, pengelolaan pengguna, dan laporan *analytics*.

### 6) *Unit Testing*

Unit Testing Pengujian unit akan dilakukan untuk memastikan bahwa logika server berjalan dengan baik, termasuk autentikasi, otorisasi, dan pengelolaan data, pengujian komponen frontend akan mencakup pengujian antarmuka pengguna untuk memastikan interaktivitas dan responsivitas. Selanjutnya, uji coba integrasi *Gemini AI* dilakukan untuk memastikan bahwa sistem dapat berkolaborasi dengan *Gemini AI* dan mencapai hasil yang diinginkan.

### 7) *Integrasi Testing*

Testing integrasi frontend dan backend akan memastikan data mengalir dengan lancar antara antarmuka pengguna dan server. Testing integrasi dengan *Gemini AI* akan memastikan bahwa kecerdasan buatan dapat diakses dan diterapkan dengan benar dalam proses manajemen konten. Sebelum penggunaan penuh, pengujian penerimaan

pengguna (UAT) akan dilakukan untuk memastikan bahwa sistem memenuhi kebutuhan dan harapan pengguna.

### 8) *System Testing*

Hasil dari pengujian ini menunjukkan bahwa sistem berfungsi dengan baik dan memenuhi harapan pengguna. Umpan balik dari pengujian ini menjadi dasar penting untuk melaksanakan peningkatan kecil pada antarmuka dan proses kerja sistem agar lebih selaras dengan kebiasaan serta kebutuhan operasional perangkat desa. Dengan pelaksanaan *System Testing* oleh mereka yang akan langsung menggunakan sistem ini, dapat dipastikan bahwa sistem manajemen konten berita tersebut benar-benar siap untuk diterapkan dan mampu memberikan keuntungan nyata bagi proses digitalisasi informasi di Desa Kebondalem.

### 9) *Acceptance Testing*

Tujuan dari tahap ini adalah untuk memastikan bahwa sistem yang telah dikembangkan sepenuhnya memenuhi keinginan dan kebutuhan pengguna akhir sebelum diimplementasikan sepenuhnya di lapangan.

## IV. HASIL DAN PEMBAHASAN

Hasil implementasi menunjukkan bahwa integrasi *Gemini AI* ke dalam sistem manajemen konten berita berbasis *website* berhasil diterapkan secara efektif di Desa Kebondalem. Sistem ini dirancang untuk mengotomatisasi proses penulisan berita berdasarkan input berupa judul dan poin-poin penting dari admin desa. Konten yang dihasilkan ditampilkan kembali ke frontend dan dapat disimpan ke dalam database *MongoDB* jika dinilai layak. Penerapan ini secara signifikan mempercepat proses pembuatan berita, mengurangi penulisan manual, serta menghasilkan konten yang relevan dan berkualitas. Sistem ini berperan dalam mempercepat penyebaran informasi secara efisien di lingkungan desa.

### A. *Requirement Design*

Proses ini dilakukan melalui observasi langsung serta wawancara mendalam dengan Kepala Desa Kebondalem, Bapak Purwanto, dan perangkat desa lainnya yang terlibat dalam pengelolaan informasi publik desa.

Hasil observasi menunjukkan bahwa pemerintah desa memiliki kebutuhan nyata akan sistem informasi yang mendukung proses penyebaran berita secara cepat dan efisien. Dalam wawancara yang dilakukan, responden menyampaikan bahwa desa sangat memerlukan *website* berita sebagai media informasi resmi yang mudah diakses oleh masyarakat. Kepala desa menekankan pentingnya tampilan antarmuka yang sederhana, tidak membingungkan, dan mudah dipahami oleh pengguna awam.

Sistem dinilai praktis, mudah digunakan, dan layak untuk diterapkan secara resmi sebagai sistem informasi desa.. Berikan kemungkinan pengembangan atau penelitian ke depan terkait penelitian ini.

TABLE 1  
KEBUTUHAN FUNGSIONAL

| No | Kebutuhan Fungsional                       | Deskripsi   |
|----|--|---|
| 1  | Pembuatan Berita                           | Admin dapat membuat berita dengan input judul dan isi utama.                  |
| 2  | Otomatisasi Konten dengan <i>Gemini AI</i> | Sistem menghasilkan draf berita berdasarkan poin masukan dari admin.          |
| 3  | Review dan Edit                            | Admin dapat meninjau, menyunting, dan mengoreksi hasil AI sebelum publikasi.  |
| 4  | Publikasi Berita                           | Berita yang telah disetujui dapat dipublikasikan langsung ke <i>website</i> . |
| 5  | Kategori Berita                            | Berita dapat diklasifikasikan berdasarkan kategori agar mudah dicari.         |

TABLE 2  
KEBUTUHAN NON FUNGSIONAL

| No | Kebutuhan Non-Fungsional | Deskripsi   |
|----|--------------------------|---|
| 1  | Antarmuka Sederhana      | Tampilan sistem harus intuitif, tidak memerlukan pelatihan teknis khusus. |
| 2  | Responsif                | Sistem dapat diakses baik melalui desktop maupun perangkat mobile.        |
| 3  | Keamanan Data            | Akses dan data pengguna dilindungi dari akses yang tidak sah.             |
| 4  | Kecepatan Respons        | Waktu tanggap sistem maksimal 3 detik untuk tiap permintaan pengguna.     |
| 5  | Integrasi Stabil         | Seluruh komponen (frontend, backend, dan AI) harus berjalan konsisten.    |

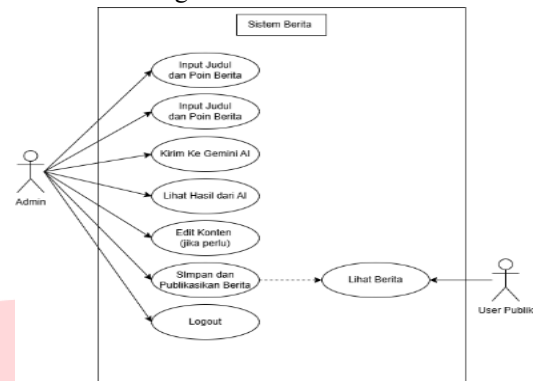
Kebutuhan pengguna ini menjadi landasan dalam proses perancangan, pengembangan, dan pengujian sistem ke tahap selanjutnya.

### B. System Design

Tahap perancangan sistem bertujuan untuk mengidentifikasi kebutuhan pengguna dan menyusunnya ke dalam bentuk struktur sistem yang bisa dikembangkan secara teknis.

#### 1) Use Case

Diagram Penggunaan menggambarkan fungsionalitas yang diharapkan dari sistem serta interaksi antara sistem dan aktor. Aktor dapat berupa manusia atau sistem eksternal yang berinteraksi dengan sistem utama

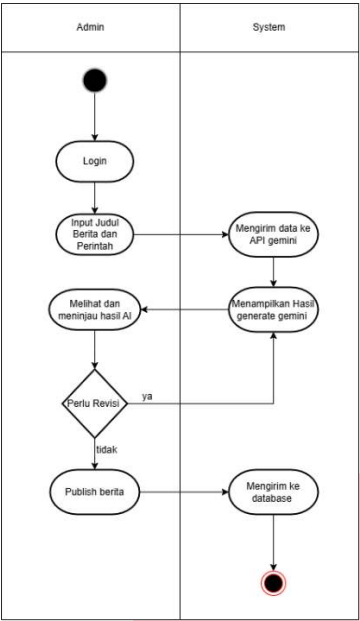


GAMBAR 3  
DIAGRAM USE CASE

Diagram *Use Case* ini menjelaskan bagaimana sistem berinteraksi dengan aktor utama, yaitu admin, dalam mengelola konten berita melalui *website*. Admin menjadi pusat dari seluruh proses, mulai dari membuat hingga menerbitkan berita. Proses dimulai dengan admin melakukan login, lalu memasukkan judul dan poin utama yang akan menjadi dasar artikel. Data tersebut dikirim ke layanan *Gemini AI* untuk diproses menjadi tulisan lengkap. Setelah itu, admin bisa melihat hasil dari AI, lalu melakukan perbaikan jika diperlukan. Jika konten dianggap sudah sesuai, admin akan menyimpan dan menerbitkan berita ke halaman publik. Proses selesai dengan admin melakukan logout. *Use Case* ini menunjukkan bahwa sistem dirancang agar admin bisa mengelola berita dengan lebih efisien dan terorganisir, dengan bantuan teknologi kecerdasan buatan.

#### 2) Activity Diagram

Diagram aktivitas berfungsi untuk menggambarkan perjalanan suatu proses bisnis, seperti langkah-langkah dalam menyusun berita baru, tahapan pemvalidasian oleh kepala desa, hingga penerbitan di situs web. Diagram ini sangat berguna untuk memahami runtutan tindakan dan proses pengambilan keputusan dalam sistem.

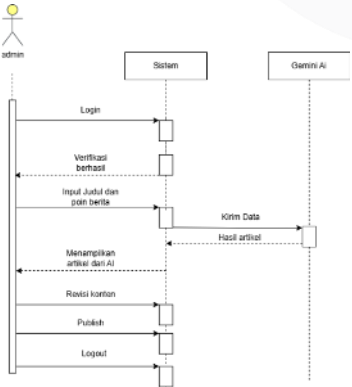


GAMBAR 4  
ACTIVITY DIAGRAM

Pada diagram tersebut menggambarkan alur bagaimana admin berinteraksi dengan sistem, yang pertama admin melakukan login dan memasukkan judul atau point – point, ketika sudah selesai memasukkan kemudian sistem akan mengirimkan hal tersebut ke *API Gemini AI* untuk memproses pembuatan artikel. Setelah itu sistem menghasilkan artikel atau berita tersebut yang bisa di tinjau atau dilihat oleh admin, admin diberi pilihan, perlu di revisi atau tidak, jika tidak maka admin bisa melakukan post berita atau publish artikel tersebut.

3) *Sequence Diagram*

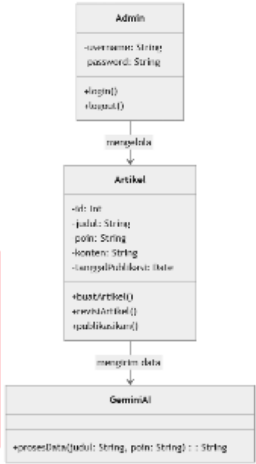
Diagram *Sequence* menunjukkan bagaimana objek (seperti pengguna, sistem, dan modul AI) berinteraksi satu sama lain dalam suatu urutan waktu yang spesifik. Alat ini sangat berharga untuk merepresentasikan proses yang dinamis, seperti cara pengguna berinteraksi ketika memanfaatkan fitur *Gemini AI* untuk secara otomatis menyusun berita.



GAMBAR 5  
SEQUENCE DIAGRAM

4) *Class Diagram*

*Class Diagram* menggambarkan organisasi data dalam sistem, mencakup kelas-kelas penting seperti Pengguna, Konten, Kategori, dan Modul Kecerdasan Buatan. Diagram ini juga menyertakan properti, fungsi, dan keterkaitan antar kelas seperti pewarisan dan asosiasi.



GAMBAR 6  
CLASS DIAGRAM

Pada *Class Diagram* ini menggambarkan struktur statis sistem berupa kelas yang terhubung dari beberapa kelas mulai dari kelas admin, artikel yang didalamnya terdapat beberapa point untuk mengirimkan data ke *Gemini AI* melalui sistem.

5) *Component Diagram*

Diagram komponen menunjukkan susunan modular dari suatu sistem. Diagram ini mengilustrasikan hubungan antara berbagai modul, seperti antarmuka pengguna, modul pengelolaan konten, serta integrasi *Gemini AI* dalam sistem tersebut.



GAMBAR 7  
COMPONENT DIAGRAM

TABLE 3  
PENJELASAN *DIAGRAM COMPONENT*

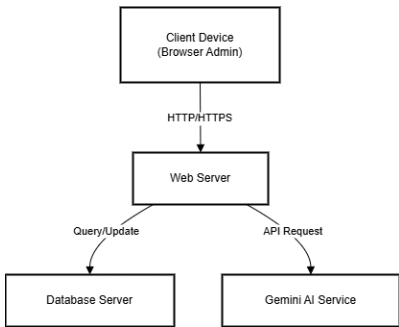
| Komponen        | Teknologi/Fungsi                            | Komponen        |
|-----------------|---|-----------------|
| <i>React.js</i> | Frontend web yang digunakan oleh admin desa | <i>React.js</i> |



| Komponen                     | Teknologi/Fungsi   | Komponen                     |
|------------------------------|--|------------------------------|
| <i>Express.js API</i>        | Backend berbasis Node.js yang menangani permintaan dari frontend     | <i>Express.js API</i>        |
| <i>Kontroler Autentikasi</i> | Mengelola login/logout pengguna                                      | <i>Kontroler Autentikasi</i> |
| <i>Kontroler Berita</i>      | Menangani input berita, revisi, dan publikasi                        | <i>Kontroler Berita</i>      |
| <i>Service Gemini AI</i>     | Mengirim permintaan ke Gemini AI dan menerima hasil artikel otomatis | <i>Service Gemini AI</i>     |

6) Component Diagram

Diagram penyebaran menunjukkan lokasi elemen sistem dan koneksinya dengan pengguna.



GAMBAR 8  
COMPONENT DIAGRAM

| TABLE 4<br>PENJELASAN DIAGRAM KOMPONEN |  |
|--|--|
| Node Fisik                             | Deskripsi  |
| <b>Client Device</b>                   | Perangkat admin desa yang mengakses aplikasi melalui browser (laptop, PC, atau smartphone) |
| <b>Web Server</b>                      | Server yang menjalankan backend Node.js dan Express.js, memproses permintaan dari client   |
| <b>Database Server</b>                 | Server yang menjalankan <i>MongoDB</i> untuk menyimpan data aplikasi                       |
| <b>Gemini AI Service</b>               | Layanan eksternal berbasis cloud yang menyediakan AI untuk pembuatan konten berita         |

7) Architecture Diagram

Desain arsitektur sistem ini menjelaskan cara komponen utama aplikasi manajemen konten berita berbasis web dirancang dan saling berinteraksi. Sistem menggunakan pola arsitektur client-server dengan empat komponen utama.

React.js digunakan di frontend untuk antarmuka pengguna, sementara Express.js berfungsi sebagai backend yang mengelola logika bisnis dan komunikasi. *MongoDB* menyimpan data penting, dan sistem terhubung dengan *Gemini AI API* untuk otomatisasi penulisan artikel berita. Integrasi ini meningkatkan kemudahan dan efisiensi pengelolaan berita desa.

TABLE 5  
TRANSFORMTION MAPPING AKTIFITAS SISTEM

| Proses Fungsional                    | Transformasi Desain                              |
|--------------------------------------|--|
| Login                                | Validasi kredensial dan sesi                     |
| Menginput Judul dan Poin Berita      | Form input dan validasi                          |
| Mengirim Data ke Gemini AI           | Menghubungkan dan mengirim data ke API Gemini AI |
| Melihat dan Meninjau Hasil AI        | Menampilkan hasil artikel dari AI                |
| Melakukan Revisi Konten              | Menyediakan editor dan menyimpan revisi          |
| Menyimpan dan Mempublikasikan Berita | Penyimpanan ke database dan update status        |
| Menampilkan Berita ke Publik         | Render halaman publik dengan konten berita       |
| Logout                               | Menghapus sesi dan mengarahkan ke halaman login  |

8) Modul Design

Modul integrasi dalam sistem ini menghubungkan antarmuka administrasi dengan layanan kecerdasan buatan *Gemini AI*. Modul ini memproses permintaan pembuatan konten berita secara otomatis berdasarkan input admin, seperti judul dan poin penting berita. Proses dimulai saat admin mengisi formulir dan mengirim informasi melalui antarmuka. Data dikirim ke backend menggunakan metode *HTTP POST*. Proses ini menggunakan Express.js dan pustaka *@google/generative-ai* untuk menghasilkan artikel berita yang kemudian dikembalikan ke frontend dalam format *JSON*. System ini meningkatkan efisiensi dan memudahkan admin dalam menyajikan informasi.

TABLE 6  
STRUKTUR MODUL API GEMINI AI

| Fungsi          | Deskripsi  |
|-----------------|--|
| <i>POST "/"</i> | Endpoint utama yang menerima prompt dari frontend, memproses permintaan, dan mengembalikan hasil dari <i>Gemini AI</i> |

| Fungsi                         | Deskripsi   |
|--------------------------------|---|
| <i>GoogleGenerativeAI()</i>    | Inisialisasi objek <i>API</i> Gemini menggunakan kunci <i>API</i> dari variabel lingkungan (.env)           |
| <i>generateContent(prompt)</i> | Langkah proses dalam fungsi route yang mengirim prompt ke model <i>Gemini AI</i> dan mendapatkan hasil teks |
| <i>Res.status().JSON()</i>     | Mengirimkan hasil keluaran Kembali ke pengguna dalam format <i>JSON</i>                                     |

Dengan pendekatan modular seperti ini, sistem menjadi lebih mudah dikembangkan dan dipelihara, serta memungkinkan fleksibilitas jika ingin mengganti atau mengembangkan layanan AI dimasa depan.

9) Coding

Pada tahap implementasi, kode dari modul layanan kecerdasan buatan *Gemini AI* diintegrasikan ke dalam arsitektur backend yang menggunakan Express.js. Sistem menerima input dari frontend, mengolahnya dengan *Gemini AI*, dan mengembalikan hasilnya kepada pengguna. File .env digunakan untuk menyimpan kunci *API* dengan aman. File *gemini.js* mendefinisikan route *POST/* sebagai endpoint utama, sementara *server.js* mengatur middleware dan routing. Layanan AI diintegrasikan dengan modul *genAI.getGenerative()* yang memproses input prompt menggunakan fungsi *generateContent(prompt)*. Hasil keluaran dikembalikan ke pengguna dalam format yang sesuai. Struktur kode yang baik mendukung keterbacaan dan perawatan sistem.

```
1 const express = require('express');
2 const { GoogleGenerativeAI } = require('@google/generative-ai');
3 const dotenv = require('dotenv').config();
4
5 const router = express.Router();
6 const genAI = new GoogleGenerativeAI(process.env.GEMINI_API_KEY);
7
8 router.post('/', async (req, res) => {
9   const { title, date, prompt } = req.body;
10
11   if (!title || !date || !prompt) {
12     return res.status(400).json({ error: 'Missing title, date, or prompt' });
13   }
14
15   try {
16     const model = genAI.getGenerativeModel({ model: 'gemini-1.5-flash' });
17     const result = await model.generateContent([
18       {
19         role: 'user',
20         parts: [
21           { text: `Buatlah artikel dengan judul "${title}" dan tanggal "${date}" berdasarkan prompt berikut: ${prompt}` },
22         ],
23       },
24     ]);
25     const response = result.response.candidates[0].content.parts[0].text || '';
26
27     res.status(200).json({ generatedText: response });
28   } catch (error) {
29     console.error('Error in generateContent:', error);
30     return res.status(500).json({ error: 'Failed to get response from Gemini', details: error.message });
31   }
32 });
33
34 module.exports = router;
```

GAMBAR 9  
CODE POST LAYANAN GEMINI AI

10) Coding

Pengujian unit dilakukan untuk memastikan bahwa modul *API* dengan *Gemini AI* dapat berfungsi secara mandiri. Pengujian ini melibatkan endpoint *API* dan fungsi utama yang berkomunikasi antara frontend dan layanan AI,

menggunakan Postman untuk mengirim permintaan dan mengecek respon.

TABLE 7  
PENGUJIAN HASIL INTEGRASI MODUL *GEMINI AI*

| Fungsi                         | Deskripsi   | Hasil   |
|--------------------------------|---|---|
| <i>POST "/API/gemini"</i>      | Menerima data dari frontend dan memproses permintaan  | Respon HTTP 200 dengan teks hasil dari gemini dalam format <i>JSON</i>        |
| <i>GoogleGenerativeAI()</i>    | Inisialisasi objek <i>API</i> Gemini menggunakan kunci <i>API</i> dari variabel lingkungan (.env) | Berhasil membuat instance model <i>AI</i> Gemini                              |
| <i>generateContent(prompt)</i> | Mengirim prompt kelayanan <i>Gemini AI</i>  | Output teks yang dihasilkan sesuai  |
| <i>Res.status().JSON()</i>     | Mengirimkan hasil keluaran Kembali ke pengguna dalam format <i>JSON</i>                           | Format <i>JSON</i> dikembalikan dengan struktur {response : "..."} tanpa eror |

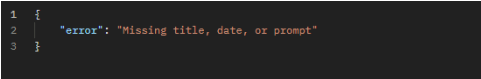
11) Coding

Pengujian *API* dilakukan untuk memastikan modul integrasi antara sistem backend dan layanan *Gemini AI* bekerja dengan baik. Pengujian pertama berhasil dengan menggunakan metode *POST* untuk mengirim data dari frontend ke backend, menghasilkan kode status 200 OK dan output *JSON*.



GAMBAR 10  
RESPONS *API* GEMINI AI

Pengujian kedua gagal karena input yang tidak lengkap, menghasilkan kode status 404 Bad Request.



GAMBAR 11  
RESPONS GAGAL

Pengujian terakhir mensimulasikan kegagalan sistem akibat kesalahan konfigurasi, yang memunculkan respons 500 Internal Server Error.



GAMBAR 12  
RESPONS KESALAHAN KONFIGURASI *API*  
Seluruh skenario menunjukkan bahwa sistem dapat melakukan validasi dan penanganan kesalahan dengan baik.

12) Integration Testing

Pengujian integrasi adalah langkah penting dalam pengembangan sistem untuk memastikan semua komponen dapat bekerja sama. Pengujian ini melibatkan antarmuka admin dengan React.js, backend menggunakan Express.js, layanan *Gemini AI*, dan basis data *MongoDB*. Proses dilakukan dengan skenario nyata, seperti pembuatan berita otomatis oleh admin. Hasilnya menguji fungsi masing-masing elemen dalam sistem. Berikut merupakan tabel skenario pengujian integrasi:

TABLE 8  
PENGUJIAN INTEGRASI BLACKBOX

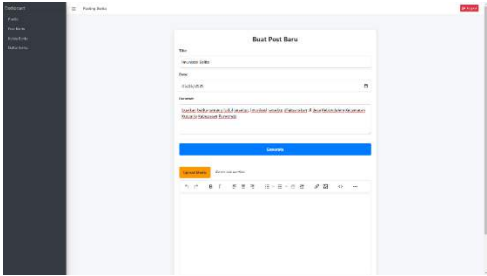
| Skenario                        | Langkah  | Hasil yang Diharapkan   | Status |
|---------------------------------|--|---|--------|
| Alur Pembuatan Berita Lengkap   | Admin mengisi form (judul+topik+tanggal), kirim ke backend, hasil dari <i>Gemini AI</i> di tampilkan ke frontend | Artikel otomatis ditampilkan di frontend dan dapat disimpan ke database         | Lulus  |
| Kesalahan pada Input Kosong     | Admin mengirim form tanpa mengisi judul atau poin  | Backend memberikan respons eror, dan frontend menampilkan validasi input        | Lulus  |
| Simpan Artikel ke Database      | Setelah hasil dari <i>Gemini AI</i> ditampilkan, admin klik tombol publish                                       | Data artikel berhasil disimpan ke <i>MongoDB</i> dan muncul dalam daftar berita | Lulus  |
| Respons AI tidak Aktif/Time out | Admin mengirim prompt saat koneksi ke <i>Gemini AI</i> terputus  | Sistem menampilkan pesan kesalahan dan tetap stabil (tidak crash)               | Lulus  |

Dari hasil pengujian, integrasi sistem berfungsi baik pada sebagian besar kasus, ditandai dengan kode 200 untuk permintaan yang berhasil. Terdapat satu kasus dengan kode 500 akibat kesalahan server. Pengujian menunjukkan sistem mampu menangani kesalahan dan memberikan feedback yang jelas.

13) System Testing

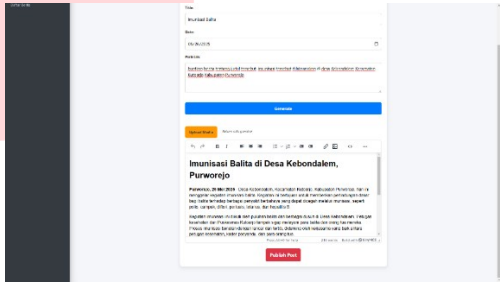
Dokumentasi ini menunjukkan bagaimana antarmuka sistem berfungsi selama pengujian integrasi. Dimulai dari pengisian berita di antarmuka admin, sistem menunjukkan aliran dari judul dan konten berita hingga hasil artikel dari layanan *Gemini AI*. Validasi sistem tercermin dari

pesan kesalahan untuk input kosong dan notifikasi sukses saat menyimpan artikel di database.

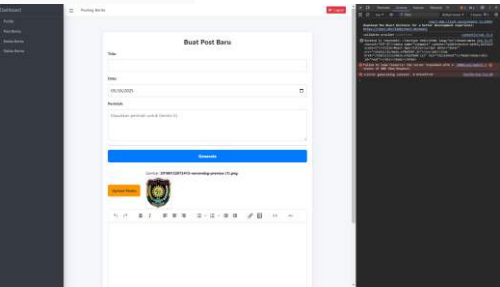


GAMBAR 13  
TAMPILAN FORM INPUT BERITA DI ANTARMUKA ADMIN

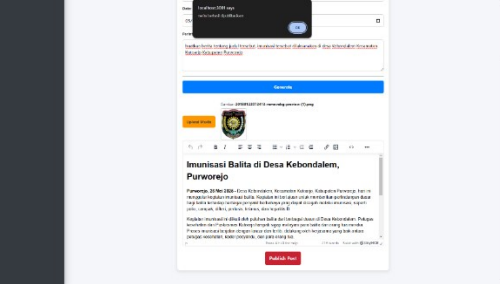
Gambar diatas memperlihatkan form input berita bagi admin untuk memasukkan judul dan poin penting yang akan diproses.



GAMBAR 14  
HASIL ARTIKEL OTOMATIS DI HALAMAN ADMIN  
Setelah data dikirim, artikel yang dihasilkan secara otomatis oleh layanan *Gemini AI* ditampilkan kembali pada halaman admin. Tampilan ini memungkinkan admin meninjau hasil teks sebelum melanjutkan proses penyimpanan ke basis data.



GAMBAR 15  
TAMPILAN VALIDASI INPUT KOSONG  
Gambar 15 menunjukkan validasi saat form dikirim tanpa data, dengan pesan kesalahan muncul.



GAMBAR 16  
KONFIRMASI PENYIMPANAN ARTIKEL KE DATABASE



Tampilan menunjukkan notifikasi bahwa artikel sudah disimpan di *MongoDB* setelah admin menekan tombol Publish. Proses integrasi berjalan baik. Survei oleh Kepala Desa dan perangkat desa menunjukkan sistem mudah digunakan, fitur AI membantu pembuatan berita, dan fungsi utama memenuhi kebutuhan operasional. Kepala Desa, Bapak Purwanto, mendukung penyebaran informasi tanpa pelatihan teknis. Sistem siap diimplementasikan di Desa Kebondalem.

#### 14) Acceptance Testing

*Acceptance Testing* dilakukan untuk memastikan sistem manajemen konten berita berbasis web dengan *Gemini AI* memenuhi kebutuhan pengguna. Metode yang digunakan adalah *Blackbox Testing* untuk menguji fungsionalitas dan kemudahan sistem.

TABEL 9  
HASIL ACCEPTANCE TEST

| Deskripsi Pengujian                                    | Input / Prosedur                                       | Hasil Responden                     | Status |
|--|--|-------------------------------------|--------|
| Desa membutuhkan <i>website</i> untuk publikasi berita | Bapak Purwanto menjawab kuesioner                      | Mayoritas responden setuju          | Lulus  |
| Tampilan <i>website</i> sederhana dan mudah digunakan  | Bapak Aan menjawab kuesioner kebutuhan tampilan sistem | Semua responden setuju              | Lulus  |
| Fitur AI membantu menulis isi berita                   | Ibu Dewi menjawab kuesioner tentang kebutuhan fitur AI | Sebagian besar responden setuju     | Lulus  |
| Sistem bisa digunakan tanpa pelatihan teknis khusus    | Bapak Wibi menjawab apakah sistem mudah digunakan      | Mayoritas responden setuju          | Lulus  |
| Pengguna berhasil login ke sistem                      | 5 responden mencoba login                              | Semua berhasil login                | Lulus  |
| Pengguna bisa menambahkan berita baru                  | 5 responden mengisi form dan klik submit               | Semua berhasil menambahkan berita   | Lulus  |
| Sistem menampilkan artikel otomatis dari AI            | Kirim data ke <i>Gemini AI</i> , tampilkan hasil       | Hasil tampil di mayoritas perangkat | Lulus  |
| Berita berhasil disimpan dan muncul di daftar berita   | Simpan dan buka daftar berita                          | Semua berita tersimpan dan tampil   | Lulus  |

| Deskripsi Pengujian                  | Input / Prosedur                                  | Hasil Responden            | Status |
|--------------------------------------|---|----------------------------|--------|
| Sistem mudah dipahami saat digunakan | Responden menjawab kuesioner fungsi dan antarmuka | Mayoritas menyatakan mudah | Lulus  |

Sistem berhasil melewati pengujian teknis dan *UAT*, siap diimplementasikan untuk perangkat desa dan masyarakat.

#### V. KESIMPULAN

Penelitian ini bertujuan mengembangkan sistem manajemen konten berita berbasis *website* dengan teknologi kecerdasan buatan *Gemini AI* untuk memperbaiki penyebaran informasi di Desa Kebondalem. Hasil pengembangan menunjukkan bahwa sistem berhasil dibangun dengan metode *Waterfall* dan pendekatan V-Model. Sistem ini membantu pemerintah desa mengelola dan menyebarkan informasi secara digital, dengan cepat dan terstruktur, melalui *website*.

Integrasi *Gemini AI* memungkinkan proses pembuatan konten berita menjadi efisien, di mana artikel dapat dihasilkan berdasarkan poin berita dari admin desa. Sistem juga memungkinkan revisi sebelum publikasi, meningkatkan relevansi konten. Uji coba menunjukkan kinerja sesuai spesifikasi dengan tanggapan positif dari responden mengenai kemudahan penggunaan dan fitur AI yang mempercepat penulisan. Secara keseluruhan, sistem ini mampu meningkatkan literasi digital, partisipasi warga, dan transparansi informasi publik, serta mencapai tujuan penelitian.

#### REFERENSI

- [1] N. Purwati, H. B. Pradana, and D. Iswahyuni, "Perancangan Sistem Informasi Berita Berbasis *Website* Pt Garda Revolusi Tv Madiun," *CONTEN Comput. Netw. Technol.*, vol. 2, no. 2, pp. 108–117, 2022, doi: 10.31294/conten.v2i2.1670.
- [2] E. Kurniawan, "Pengembangan Content Management System sebagai Media Informasi di Masa Pandemi: Studi Kasus di Perpustakaan Perguruan Tinggi Muhammadiyah/Aisyiyah," *JIPI (Jurnal Ilmu Perpust. dan Informasi)*, vol. 7, no. 1, p. 19, 2022, doi: 10.30829/jipi.v7i1.10067.
- [3] N. Hartatik, N. L. Azizah, and S. Busono, "Sistem Informasi Desa Berbasis Web Dengan Menggunakan Metode *Waterfall*," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 9, no. 1, pp. 264–271, 2024, doi: 10.29100/jipi.v9i1.4428.
- [4] H. Hermansyah, S. Wahyuni, and A. Akbar, "Perancangan Sarana Media Informasi Berbasis Web Desa Klambir Lima Menggunakan Metode *Waterfall*," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 2, p. 515, 2022, doi: 10.30865/jurikom.v9i2.3803.
- [5] F. Rozi, T. Listiawan, and Y. Hasyim, "Pengembangan *Website* Dan Sistem Informasi Desa Di Kabupaten Tulungagung," *JIPI (Jurnal Ilm. Penelit. dan*

Pembelajaran Inform., vol. 2, no. 2, pp. 107–112, 2017, doi: 10.29100/jipi.v2i2.366.

- [6] M. Audrilia and A. Budiman, “Perancangan Sistem Informasi Manajemen Bengkel Berbasis Web (Studi Kasus: Bengkel Anugrah),” J. Madani Ilmu Pengetahuan, Teknol. dan Hum., vol. 3, no. 1, pp. 1–12, 2020, doi: 10.33753/madani.v3i1.78.
- [7] A. Alfariy, F. Fenando, and M. S. Muarie, “Sistem Informasi Penjualan dan Persediaan Barang Berbasis Web Menggunakan Metode *V-Model* pada Toko Arif Gorden,” J. Inf. Technol. Ampera, vol. 2, no. 1, pp. 1–16, 2021, doi: 10.51519/journalita.volume2.issue1.year2021.page1-16.
- [8] A. D. Herlambang et al., “*V-Model* Untuk Pengembangan *V-Model* for Meeting Room Managemnt Ruang Rapat,” J. Teknol. Inf. dan Ilmu Komput., vol. 7, no. 2, pp. 313–322, 2020, doi: 10.25126/jtiik.202071893.