

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Menurut Wildan dkk. (2025), penerapan teknologi kecerdasan buatan (*AI*) dalam layanan publik memiliki potensi besar dalam meningkatkan efisiensi dan kualitas pelayanan melalui otomasi serta penyediaan informasi yang lebih cepat dan akurat. Namun, penelitian ini juga menunjukkan adanya berbagai tantangan dalam implementasi *AI*, seperti keterbatasan infrastruktur digital, kesiapan sumber daya manusia yang belum optimal, serta perlunya perhatian terhadap aspek keamanan dan privasi data. Hal ini menunjukkan bahwa keberhasilan penerapan *AI* tidak hanya ditentukan oleh teknologi yang digunakan, tetapi juga oleh kesiapan organisasi dan kebijakan pendukung. Temuan tersebut relevan dengan penelitian tentang penerapan *chatbot* pada layanan administrasi kependudukan (*Adminduk*), karena menggambarkan bahwa pemanfaatan *AI* dalam pelayanan publik masih memerlukan pengembangan dan penyesuaian agar dapat berjalan secara optimal [10].

Menurut Agustian dkk. (2024), penelitian ini mengembangkan aplikasi berbasis *website* yang dilengkapi *chatbot* layanan publik untuk meningkatkan efisiensi penyampaian informasi dan pengaduan di Sekretariat DPRD Kota Cimahi. *Chatbot* dirancang untuk memberikan informasi layanan serta menampung aspirasi masyarakat secara otomatis sehingga dapat mengurangi beban kerja petugas. Sistem dikembangkan menggunakan metode *Waterfall* dan diuji melalui *User Acceptance Test* (UAT) serta *Black Box Testing*

dengan hasil UAT sebesar 93,12% yang menunjukkan tingkat penerimaan pengguna yang tinggi. Namun, penelitian ini masih berfokus pada penyediaan informasi dan pengaduan secara umum, sehingga menunjukkan adanya peluang pengembangan lebih lanjut, khususnya dalam peningkatan kemampuan *chatbot* agar dapat memberikan layanan yang lebih adaptif dan terintegrasi dengan data layanan secara menyeluruh [11].

Menurut Muhammad Vicky Al Hasri dan Endah Sudarmilah (2023), sistem informasi administrasi kependudukan berbasis *website* mampu meningkatkan efisiensi pelayanan yang sebelumnya dilakukan secara manual. Sistem ini memungkinkan masyarakat mengajukan permohonan layanan secara daring dengan mengisi formulir dan mengunggah dokumen persyaratan, sedangkan petugas dapat melakukan verifikasi serta pengelolaan data permohonan secara terpusat. Hasil pengujian menunjukkan bahwa sistem berjalan dengan baik dan dapat meningkatkan efisiensi pelayanan administrasi kependudukan. Namun, penelitian ini masih berfokus pada pengelolaan layanan berbasis formulir dan belum membahas penerapan fitur interaktif seperti *chatbot* atau layanan otomatis berbasis kecerdasan buatan, sehingga interaksi dengan pengguna masih bergantung pada proses verifikasi oleh petugas [12].

Menurut Wahyono dkk. (2022), implementasi pelayanan administrasi kependudukan di Kota Malang telah berjalan cukup baik, khususnya pada layanan KTP dan KK, yang ditunjukkan oleh tingkat kepuasan masyarakat yang tinggi. Keberhasilan implementasi dipengaruhi oleh faktor komunikasi,

disposisi aparatur, dan struktur birokrasi yang telah didukung oleh Standar Operasional Prosedur (SOP). Namun, penelitian ini juga menemukan adanya kendala pada aspek sumber daya, terutama keterbatasan kualitas sumber daya manusia serta belum optimalnya sarana prasarana dan jaringan internet. Kondisi tersebut menunjukkan bahwa pemanfaatan teknologi informasi dalam pelayanan administrasi kependudukan masih perlu ditingkatkan agar proses pelayanan dapat berlangsung lebih cepat, efisien, dan merata [13].

Hasil penelitian terdahulu menunjukkan bahwa pemanfaatan teknologi informasi dan *chatbot* mampu meningkatkan kualitas layanan publik, namun penerapannya masih terbatas pada penyampaian informasi dan belum terintegrasi dengan *dataset* resmi serta proses pengajuan layanan administrasi kependudukan. Oleh karena itu, diperlukan sistem layanan administrasi kependudukan berbasis *website* yang dilengkapi *chatbot* menggunakan metode *Retrieval Augmented Generation (RAG)* agar informasi yang disajikan lebih akurat dan terintegrasi dengan pengajuan layanan tertentu.

2.2 Landasan Teori

2.2.1 Website

Website disebut juga *site*, situs, *situs web* atau *portal*.

Merupakan kumpulan halaman *web* yang berhubungan antara satu dengan lainnya, halaman pertama sebuah *website* adalah *home page*, sedangkan halaman demi halamannya secara mandiri disebut *web page*, dengan kata lain *website* adalah situs yang dapat diakses dan dilihat oleh para pengguna internet diseluruh dunia. *Website* adalah

situs yang dapat diakses dan dilihat oleh para pengguna Internet. Pengguna Internet semakin hari semakin bertambah banyak, sehingga hal ini adalah potensi pasar yang berkembang terus. Lebih jelasnya, *website* merupakan halaman-halaman yang berisi informasi yang ditampilkan oleh *browser* seperti *Mozilla*, *Firefox*, *Google Chrome* atau yang lainnya [14].

2.2.2 Chatbot

Chatbot merupakan teknologi intelegensia buatan berbasis teks atau audio yang dapat melakukan interaksi percakapan antara mesin dengan *user*, respon yang diberikan *chatbot* saat berinteraksi dengan *user* didapatkan berdasarkan *input* yang diberikan. Komponen *chatbot* terdiri dari program utama (*bot*) dan *brain file*. *Input* yang masuk akan di *parsing* ke *brain file* untuk mendapatkan data pemetaan respon yang ditampilkan oleh *chatbot* [15].

2.2.3 Visual Studio Code

Visual Studio Code adalah sebuah editor kode sumber yang dikembangkan oleh *Microsoft*. Perangkat lunak gratis dan *open-source* yang dirancang untuk membantu para pengembang dalam menulis dan mengedit kode dengan berbagai bahasa pemrograman. Selain pengeditan dan penelusuran kesalahan kode, *Visual Studio* mencakup kompilator, alat penyelesaian kode, kontrol sumber, ekstensi, dan banyak lagi fitur untuk meningkatkan setiap tahap proses pengembangan perangkat lunak. *Visual Code* dapat dipakai untuk

membuat serta mengedit *source code* dari berbagai bahasa pemrograman Contohnya, seperti *Html*, *CSS*, Dan *JavaScript* [16].

2.2.4 Laragon

Laragon adalah sebuah *portable development environment* untuk sistem operasi *Windows* yang sangat ringan, cepat, dan mudah digunakan, dirancang khusus untuk membantu para pengembang *web* dalam membangun, mengelola, dan menjalankan aplikasi *web* secara lokal. *Laragon* mendukung berbagai teknologi populer seperti *PHP*, *Apache/Nginx*, *MySQL/MariaDB*, *Node.js*, dan *Python*, serta memiliki fitur otomatisasi seperti *auto virtual host*, *isolated environment*, dan manajemen *database* yang efisien [17].

2.2.5 HTML

HTML merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman *web*. *HTML* berfungsi untuk mempublikasi dokumen *online*. *Statement* dasar dari *HTML* disebut *tags*. Sebuah *tag* dinyatakan dalam sebuah kurung siku (<>). *Tags* yang ditujukan untuk sebuah dokumen atau bagian dari suatu dokumen haruslah dibuat berupa pasangan. Terdiri dari *tag* pembuka dan *tag* penutup. Dimana *tag* penutup menggunakan tambahan tanda garis miring (/) di awal nama *tag* [18].

2.2.6 CSS

CSS adalah bahasa-bahasa yang merepresentasikan halaman *web*. Seperti warna, *layout*, dan *font*. Dengan menggunakan *CSS*, seorang *web developer* dapat membuat halaman *web* yang dapat

beradaptasi dengan berbagai macam ukuran layar. Pembuatan *CSS* biasanya terpisah dengan halaman *HTML*. Meskipun *CSS* dapat disisipkan di dalam halaman *HTML*. Hal ini ditujukan untuk memudahkan pengaturan halaman *HTML* yang memiliki rancangan yang sama [19].

2.2.7 FastAPI

FastAPI adalah *web framework* yang digunakan untuk mengembangkan *RESTful API* dalam bahasa pemrograman *Python*. *FastAPI* merupakan didesain agar mudah untuk digunakan, selain itu *FastAPI* juga didesain mirip dengan *framework Flask* yang merupakan salah satu *framework* yang banyak digunakan. *FastAPI* juga merupakan salah satu *framework* tercepat dalam bahasa pemrograman *Python* dan setara dengan *NodeJS* dan *Go* [20].

2.2.8 Python

Python adalah sebuah bahasa pemrograman berbasis objek yang dapat diinteraksi secara interaktif. Bahasa ini memiliki struktur data tingkat tinggi. *Python* merupakan sebuah bahasa pemrograman interpretatif yang memiliki banyak fungsi, dan didesain dengan fokus pada kejelasan dan kemudahan pemahaman kode. *Python* dianggap sebagai bahasa yang menggabungkan kemampuan dan kejelasan sintaks kode. Bahasa pemrograman *Python* dirancang khusus untuk memudahkan *programmer* dalam membuat program dengan efisiensi waktu, kemudahan pengembangan, dan kompatibilitas dengan sistem.

Python dapat digunakan untuk membuat aplikasi mandiri atau pemrograman *script* [21].

2.2.9 MySQL

MySQL adalah salah satu dari aplikasi (*DBMS*) *Data Base Management System* yang sering digunakan untuk mengelola basis data, dengan kemampuan keamanan yang cukup handal. *MySQL* juga dapat terhubung dengan aplikasi seperti *PHP* dan lainnya. Jika ingin belajar banyak ada beberapa kendala ketika menggunakan *MySQL* sudah tersedia di forum yang mengkhususkan pembahasan tentang *MySQL* sehingga memudahkan bagi *user* baru, dengan aplikasi *DBSM* yang handal *software* yang selalu diupdate dan juga gratis sehingga memudahkan proses instalasi *MySQL* [22].

2.2.10 Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG) merupakan arsitektur yang menggabungkan proses pencarian informasi dengan generasi teks. Memperkenalkan konsep ini sebagai solusi untuk meningkatkan kualitas jawaban yang dihasilkan oleh *model* bahasa alami dengan memanfaatkan sumber informasi eksternal. Teknik ini memungkinkan *model* bahasa untuk mengambil informasi yang relevan dari dokumen sebelum menghasilkan jawaban, sehingga meningkatkan akurasi dan relevansi jawaban [23].

2.2.11 API Gemini

API AI Gemini adalah sebuah *API* yang dirancang untuk memudahkan integrasi teknologi kecerdasan buatan (*AI*) dalam

aplikasi *web* dan *mobile*. *API* ini menyediakan berbagai fitur canggih, termasuk analisis data, rekomendasi, dan pemrosesan bahasa alami *Natural Language Processing* (NLP). Dengan menggunakan *API AI Gemini*, pengembang dapat dengan mudah menambahkan fungsi *AI* ke dalam aplikasi mereka tanpa perlu membangun model *AI* dari awal [24].

2.2.12 Javascript

Javascript merupakan suatu bahasa yang perkembangannya lambat di bandingkan dengan *Java* yang berkembang sangat cepat. Di *Javascript* kita tidak mungkin menyembunyikan kode *skript* yang kita tulis, kode langsung di tulis di dalam dokumen *HTML* dan sangat mudah terlihat, sedangkan di *Java*, kode sudah berbentuk setengah terkompilasi (dalam bentuk *applet*) dan tidak mungkin terlihat dari dalam dokumen *HTML*, satu mesin virtual di sisi *user* yang bertanggung jawab untuk menterjemahkan program di dalam *applet* tersebut setiap kali halaman *HTML* yang memuat *applet* tersebut dipanggil oleh *browser*. Dibandingkan dengan *applet java* yang cukup lambat dibuka oleh *browser*, bisa kita katakan bahwa *Javascript* cukup cepat di panggil (di *load*) oleh navigator [25].

2.2.13 Domain

Domain Name atau Nama Domain adalah nama unik yang digunakan untuk mengidentifikasi alamat sebuah *website* di *internet*. Nama domain berfungsi untuk mempermudah pengguna internet pada saat melakukan akses ke *server*, selain itu juga dipakai untuk

mengingat nama *server* yang dikunjungi tanpa harus mengenal deretan angka yang rumit (*IP address*) [26].

2.2.14 Hosting

Hosting adalah tempat fisik dimana semua isi atau *content* sebuah *website* disimpan di dalamnya. *Hosting* membutuhkan *harddisk* yang terletak pada pemilik penyedia *hosting*. *Hosting* dapat diakses dan digunakan oleh pemilik *website*. *Hosting* digunakan oleh pemilik *website* untuk mengelola *CMS* atau *Content Management Systems* [26].

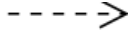
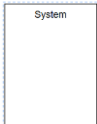
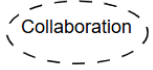
2.2.15 Unified Modeling Language

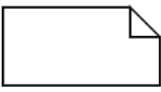
Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. *UML* merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik [27]. Terdapat beberapa diagram *UML* yang sering digunakan dalam pengembangan sebuah sistem, yaitu:

1. Use Case Diagram

Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat *actor* yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

Tabel 2. 1 Simbol Use Case Diagram

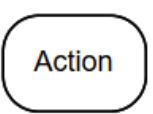
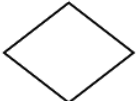
No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan Ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan Dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3.		<i>Generalization</i>	Hubungan Dimana objek anak (<i>descendent</i>) berbagai perilaku dan struktur data dari objek yang ada diatasnya objek induk (<i>ancestor</i>).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Merepresentasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antar obyek satu dengan obyek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		<i>Use Case</i>	Mendeskripsikan urutan aksi yang memberikan hasil bernilai bagi aktor.
9.		<i>Collaboration</i>	Interaksi antar elemen atau aturan untuk mencapai suatu fungsi tertentu.

No	Gambar	Nama	Keterangan
10.		<i>Note</i>	Catatan tambahan untuk menjelaskan bagian tertentu dari diagram.

2. Activity Diagram

Merupakan gambaran alur dari aktivitas-aktivitas didalam sistem yang berjalan.

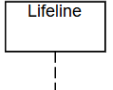
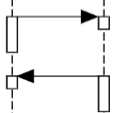
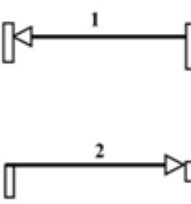
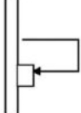
Tabel 2. 2 Simbol Activity Diagram

No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Sekelompok <i>action</i> yang membentuk satu proses lengkap. Dapat terdiri dari beberapa <i>action</i> di dalamnya.
2.		<i>Action</i>	Langkah tunggal / aksi sederhana yang dilakukan sistem. Simbol: persegi panjang dengan sudut tumpul.
3.		<i>Initial Node</i>	Titik awal aktivitas; digambarkan sebagai lingkaran hitam penuh.
4.		<i>Final Node</i>	Titik akhir aktivitas; digambarkan sebagai lingkaran dengan lingkaran lain di dalamnya.
5.		<i>Fork Node</i>	Memecah satu alur menjadi beberapa alur paralel; simbol garis tebal.
6.		<i>Decision</i>	Titik percabangan berdasarkan kondisi tertentu; simbol belah ketupat.
7.		<i>Rake</i>	Menunjukkan adanya dekomposisi
9.		<i>Send</i>	Tanda pengiriman

3. Sequence Diagram

Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

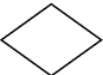
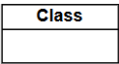
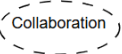
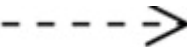
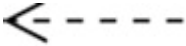
Tabel 2. 3 Simbol Sequence Diagram

No	Gambar	Nama	Kesimpulan
1.		<i>Lifeline</i>	Objek <i>entity</i> , antara muka yang saling berinteraksi.
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktivitas yang terjadi.
3.		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan system.
4.		<i>Boundary Class</i>	Menggambarkan penggambaran dari form.
5.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
6.		<i>Control Class</i>	Menggambarkan penghubung antara <i>Boundary</i> dengan label.
7.		<i>Activation</i>	Sebagai sebuah objek yang akan melakukan sebuah aksi.
8.		<i>Message</i>	Mengindikasikan komunikasi antar objek dengan objek.
9.		<i>SelfMessage</i>	Mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.

4. Class Diagram

Merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan *objek* yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

Tabel 2. 4 Simbol Class Diagram

No	Gambar	Nama	Keterangan
1.		<i>Generalization</i>	Hubungan dimana objekanan (<i>descendent</i>) berbagai perilaku dan struktur dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi yang ditampilkan dalam sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		<i>Dependency</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan Dimana perubahan yang terjadi pada suatu elemen (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7.		<i>Association</i>	Apa yang menghubungkan

No	Gambar	Nama	Keterangan
			antara objek satu dengan objek lainnya.