

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada penyusunan penelitian ini dilakukan beberapa studi literatur untuk mempelajari penelitian yang serupa dengan memperbanyak teori dan referensi yang diambil dari penelitian terdahulu. Referensi dan rujukan hasil penelitian sebelumnya yang telah dilakukan oleh peneliti lain merupakan tujuan pada tinjauan pustaka ini.

Penelitian yang dilakukan oleh Ariq Cahya Wardhana, Sri Wahjuni dan Yani Nurhadryani pada tahun 2020 yang berjudul “*Knowledge Management System Berbasis Web tentang Budidaya Hidroponik untuk Mendukung Smart Society*” menjelaskan rancang dan bangun web menggunakan metode *Knowledge Management System Life Cycle (KMSLC)* [6]. Penelitian ini bertujuan untuk mengembangkan *Knowledge Management System (KMS)* budidaya hidroponik dengan mengadopsi metode *Knowledge Management Life Cycle* melalui identifikasi pengetahuan *tacit* maupun *explicit* dari komunitas hidroponik.

Penelitian yang dilakukan oleh Muhammad Akram Ahmad, Seif Obeid Al-Shbiel pada tahun 2019 yang berjudul “*User's Segmentation on Continued Knowledge Management System Use in the Public Sector*” menjelaskan bahwa peneliti di masa depan mungkin bekerja sama dengan pemerintah Kota Kaohsiung untuk menjangkau pengguna tidak terdaftar tersebut dan mengevaluasi persepsi mereka tentang penggunaan KMS terus-menerus. Ketiga, hasil studi ini menunjukkan bahwa varian terjelaskan dari perilaku penggunaan terus-menerus relatif rendah di dua kelompok [18].

Penelitian yang dilakukan oleh Gabriele Santoroa, Demetris Vrontis, Alkis Thrassou, Luca Dezi pada tahun 2019 yang berjudul “*The Internet of Things: Building a knowledge management system for open innovation and knowledge management capacity*” menjelaskan *Knowledge Management System* mengacu pada sistem informasi yang diterapkan untuk mengelola pengetahuan organisasi dan untuk meningkatkan pembuatan, penyimpanan, transfer, dan penerapan pengetahuan [19]. Penelitian ini bertujuan untuk menyelidiki hubungan antara *Knowledge Management System*, inovasi terbuka, kapasitas manajemen pengetahuan dan kapasitas inovasi.

Penelitian yang dilakukan oleh Muhammad Akram Ahmad, Seif Obeid Al-Shbiel pada tahun 2019 yang berjudul “*The Effect of Accounting Information System on Organizational Performance in Jordanian Industrial SMEs: The Mediating Role of Knowledge Management*” pada penelitian tersebut [20], peneliti menyajikan Efek mediator *Knowledge Management* pada hubungan antara Sistem Informasi Akuntansi dan kinerja organisasi telah diteliti dalam sampel penelitian yang terdiri dari 350 karyawan untuk mengevaluasi peran beberapa faktor termasuk kepuasan kerja, dukungan manajemen atas, kepemimpinan, dan inovasi.

Penelitian yang dilakukan oleh Entis Sutrisna pada tahun 2018 yang berjudul “Implementasi *Knowledge Management System* Berbasis *Website* Dengan Model Spiral Pada PT. Trans Retail Indonesia” pada penelitian tersebut, peneliti menyajikan rancang dan bangun *Website knowledge management system* berbasis *website* dengan model Spiral pada PT. Trans Retail Indonesia dapat dilakukan dengan tiga putaran model Spiral [21]. Gabungan dari ketiga putaran tahapan model Spiral ini menghasilkan modul *authentication, forum discussion group, e-SOP, knowledge base, search knowledge, archive project* dan *dashboard*.

Penelitian yang dilakukan oleh Condro Kartiko, Hesa Abda Arrasyid dan Ariq Cahya Wardhana pada tahun 2021 yang berjudul “*Designing a mobile user experience student knowledge management system using Lean UX*” menjelaskan manajemen pengetahuan atau *Knowledge Management System* kemahasiswaan IT Telkom Purwokerto. Perancangan *user interface* dan *user experience* dilakukan dengan menggunakan metode *Lean UX* [22]. Penelitian ini bertujuan mengevaluasi hasil *MVP* dan kemudian mencatat setiap perbaikan yang dilakukan. Proses pengujian dilakukan dengan mengulang beberapa prototipe sampai diyakini tidak ada kesalahan.

Tabel 2. 2 Tinjauan Pustaka

No	Peneliti	Judul	Metode	Hasil	Perbandingan
1	Ariq Cahya Wardhana, Sri Wahjuni dan Yani Nurhadryani	<i>Knowledge Management System Berbasis Web tentang Budidaya Hidroponik untuk Mendukung Smart Society</i>	<i>Knowledge Management Life Cycle</i>	Aplikasi <i>Knowledge management system</i> berbasis <i>Website</i> dengan <i>framework codeigniter</i> berbahasa pemrograman <i>PHP</i> , Hasil uji coba menunjukkan bahwa fungsi standar klasifikasi <i>KMS</i> sudah terpenuhi dan berfungsi dengan baik.	Perbedaan penelitian dengan peneliti terdapat pada model penangkapan <i>Knowledge</i> . Pada penelitian yang dilakukan peneliti menggunakan model <i>SECI</i> untuk penangkapan <i>Knowledge</i> . Sedangkan pada penelitian tersebut menggunakan proses penangkapan <i>Knowledge</i> secara observasi.
2	Ahmad, Muhannad Akram, Al-Shbiel, Seif Obeid	<i>User's Segmentation on Continued Knowledge Management System Use in the Public Sector</i>	<i>Knowledge Management System</i>	Hasil analisis pada penelitian tersebut menunjukkan bahwa Pengguna tidak terdaftar seperti itu mungkin memiliki persepsi yang berbeda tentang penggunaan <i>KMS</i> terus-menerus. Peneliti di masa depan mungkin bekerja sama dengan pemerintah Kota Kaohsiung untuk menjangkau pengguna tidak terdaftar tersebut dan mengevaluasi persepsi mereka tentang penggunaan <i>KMS</i> terus-menerus. Ketiga, hasil studi ini menunjukkan bahwa varian terjelaskan dari perilaku penggunaan terus-menerus relatif rendah di dua kelompok	Perbedaan penelitian dengan peneliti terdapat pada model <i>Knowledge mapping</i> . Pada penelitian yang dilakukan peneliti menggunakan model <i>Mapping</i> sesuai kategori pengetahuan. Sedangkan pada penelitian tersebut menggunakan <i>User's Segmentation</i> dalam mengategorikan <i>Knowledge</i> sesuai dengan pekerjaan masing-masing pekerja

No	Peneliti	Judul	Metode	Hasil	Perbandingan
3	Gabriele Santoroa, Demetris Vrontis, Alkis Thrassou dan Luca Dezi	<i>The Internet of Things: Building a knowledge management system for open innovation and knowledge management capacity</i>	<i>Knowledge Management System</i>	Hasil analisis pada penelitian tersebut menunjukkan bahwa pada sampel 298 perusahaan Italia dari berbagai sektor, dari sudut pandang manajerial, muncul bahwa perusahaan yang mengadopsi pendekatan terbuka aktif terhadap inovasi, lebih mungkin untuk mengembangkan kapasitas inovasi.	Perbedaan penelitian dengan peneliti terdapat pada metode pendekatan penangkapan <i>Knowledge</i> . Pada penelitian yang dilakukan peneliti menggunakan pendekatan <i>socialization</i> yang termasuk dalam Model <i>SECI</i> . Sedangkan pada penelitian tersebut menggunakan pendekatan dengan Analisis <i>Knowledge Management System</i> yang dipakai atau telah dikembangkan
4	Ahmad, Muhannad Akram, Al-Shbiel, Seif Obeid	<i>The Effect of Accounting Information System on Organizational Performance in Jordanian Industrial SMEs: The Mediating Role of Knowledge Management</i>	<i>Knowledge Management System, Sampling, Quisisionnaire</i>	Hasil evaluasi pada penelitian tersebut menunjukkan bahwa pengujian ada hubungan antara Sistem Informasi Akuntansi dan kinerja organisasi telah diteliti dalam sampel penelitian yang terdiri dari 350 karyawan. Temuan dari studi ini dapat digunakan oleh studi-studi di masa depan untuk mengevaluasi peran beberapa faktor termasuk kepuasan kerja, dukungan manajemen atas, kepemimpinan, dan inovasi.	Perbedaan penelitian dengan peneliti terdapat pada metode pengujian. Pada penelitian yang dilakukan peneliti menggunakan pengujian <i>BlackBox</i> terhadap <i>Knowledge</i> yang direkam dalam <i>KMS</i> . Sedangkan pada penelitian tersebut menggunakan <i>Sampling</i> melalui penyebaran <i>Quisisionnaire</i> .
5	Sutrisna, Entis	Implementasi <i>Knowledge Management System</i> Berbasis <i>Website</i> Dengan Model <i>Spiral</i> Pada PT.	<i>Knowledge Management System, Spiral</i>	<i>Website knowledge management system</i> berbasis <i>website</i> dengan model <i>Spiral</i> pada PT. Trans Retail Indonesia dapat dilakukan dengan tiga putaran model <i>Spiral</i> .	Perbedaan penelitian dengan peneliti terdapat pada model penangkapan <i>Knowledge</i> . Pada penelitian yang dilakukan peneliti menggunakan model <i>SECI</i> yang terdiri dari <i>Socialization, Externalization, Combination</i> dan

No	Peneliti	Judul	Metode	Hasil	Perbandingan
		Trans Retail Indonesia			<i>Internalization</i> . Sedangkan pada penelitian tersebut menggunakan Model Spiral yang terdiri dari <i>Objective Setting</i> , <i>Development</i> dan <i>Planning</i> .
6	Condro Kartiko, Hesa Abda Arrasyid dan Ariq Cahya Wardhana	<i>Designing a mobile user experience student knowledge management system using Lean UX</i>	<i>Lean UX, Knowledge Management Life Cycle, BlackBox Testing</i>	Prototype Aplikasi KMS kemahasiswaan IT Telkom Purwokerto dengan metode <i>Lean UX, Knowledge Management Life Cycle, BlackBox Testing</i>	Perbedaan penelitian dengan peneliti terdapat pada sistem yang dikembangkan. Pada penelitian yang dilakukan peneliti sistem sudah dikembangkan dan dapat dipakai oleh <i>Stakeholder</i> . Sedangkan pada penelitian tersebut pengembangan Sistem hanya sampai pada proses pembuatan <i>blueprint</i> yang masih berupa <i>prototype</i>

Berdasarkan dari penelitian terdahulu yang sudah dipaparkan di atas maka dapat disimpulkan bahwa dalam pengembangan *Knowledge Management System* mempunyai beberapa metode sesuai dengan kebutuhan masing-masing. Tetapi dalam Penelitian ini memiliki kebaruan dengan menerapkan metode *Knowledge Management System Life Cycle* dengan Model *SECI* yang melibatkan empat proses kejadian dalam transfer pengetahuan, serta mengintegrasikan model penangkapan data yang lebih terstruktur untuk memperoleh hasil yang lebih efisien dan terukur dalam pengembangan *Knowledge Management System*, untuk pengujian *Knowledge Management System* pada penelitian ini digunakan pengujian *BlackBox* yang dikarenakan kesederhanaan tes ini mudah dilakukan, karena seseorang berfokus pada *input* dan *output*. Penguji tidak perlu tahu bagaimana sistem bekerja secara internal.

2.2 Landasan Teori

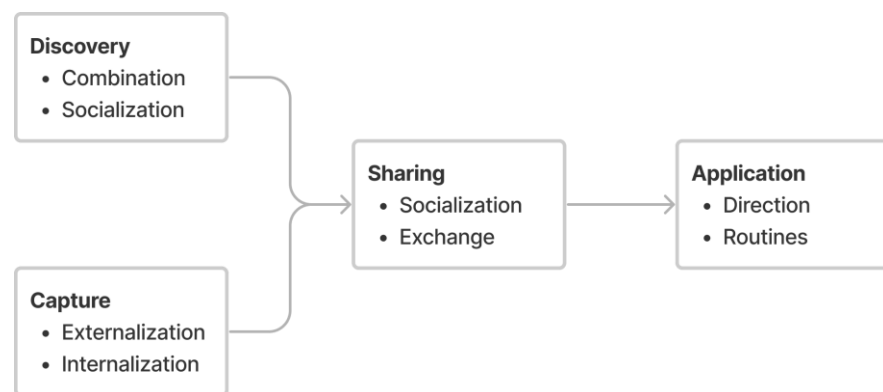
Berikut adalah kajian mengenai beberapa teori yang digunakan pada penelitian ini:

2.2.1 *Knowledge Management System*

Knowledge Management System (KMS) adalah sistem untuk mengatur dan menyimpan informasi, pengetahuan, dan data dalam sebuah organisasi dengan cara yang membuatnya dapat diakses dan digunakan. *KMS* dapat digunakan untuk menangkap, menyimpan, dan membagikan pengetahuan dan informasi dalam sebuah organisasi, dan mungkin termasuk alat seperti *database*, sistem manajemen dokumen, dan perangkat lunak kolaborasi [12], [19].

Tujuan dari *KMS* adalah untuk meningkatkan efisiensi dan efektivitas sebuah organisasi dengan memudahkan karyawan untuk mengakses dan membagikan informasi dan pengetahuan, dan dengan memungkinkan organisasi untuk lebih baik menangkap, memelihara, dan memanfaatkan pengetahuan dan keahlian kolektifnya.

KMS dapat dibagi ke dalam masing-masing proses manajemen pengetahuan, yaitu *knowledge discovery system*, *knowledge capture system*, *knowledge sharing system*, dan *knowledge application system*. *KM* yang dalam bahasa Indonesia dapat juga disebut Manajemen Pengetahuan (MP) dan dibahas langsung oleh akademisi di manajemen pengetahuan, dan untuk menjawab kebutuhan para praktisi yang ingin gambaran sederhananya terkait *KM* [23]. *KM* didefinisikan sebagai konsep pengelolaan pengetahuan yang meliputi proses *KM*. Gambar 2.1.



Gambar 2. 1 *Knowledge Management System* [23]

2.2.2 *Knowledge Management System Life Cycle*

Metode *Knowledge Management System Life Cycle (KMSLC)* memiliki tahapan sampai pada tahap verifikasi dan validasi.

a. Mengevaluasi Infrastruktur

Pada tahap ini, dilakukan evaluasi terhadap infrastruktur yang melibatkan perangkat lunak, sumber daya manusia, dan pengetahuan. Evaluasi terhadap perangkat lunak berfokus pada mengidentifikasi aplikasi yang telah digunakan dalam proses berbagi pengetahuan tentang ukiran kayu khas bali [6], [24]. Selanjutnya, evaluasi terhadap sumber daya manusia melibatkan pemilihan para ahli hidroponik dari dalam komunitas sebagai sumber pengetahuan. Setelah itu, evaluasi pengetahuan bertujuan

untuk memperoleh gambaran tentang pengetahuan apa yang telah tersedia di dalam komunitas.

b. Membentuk Tim *Knowledge Management*

Pada tahap ini, dilakukan pembentukan tim manajemen pengetahuan yang bertugas untuk mendukung pengembangan Sistem Manajemen Pengetahuan Ukiran Kayu Khas Bali. Pembentukan tim tersebut dilakukan dengan cara mengidentifikasi para pemangku kepentingan yang diperlukan, kemudian mereka terlibat dalam proses pengembangan Sistem Manajemen Pengetahuan tersebut [6], [24].

c. Menangkap *Knowledge*

Pada tahap ini, dilakukan pembentukan tim manajemen pengetahuan yang bertugas mendukung pengembangan Sistem Manajemen Pengetahuan dalam budidaya hidroponik. Proses pembentukan tim ini dilakukan dengan mengidentifikasi pemangku kepentingan yang relevan, kemudian mereka terlibat aktif dalam pengembangan Sistem Manajemen Pengetahuan tersebut.

d. Mendesain *Blueprint Knowledge Management System*

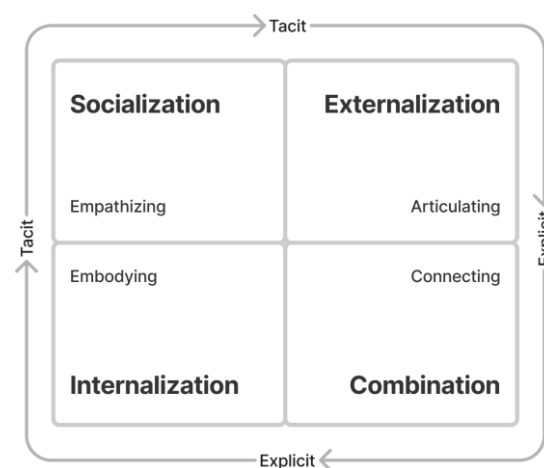
Dalam tahap ini, dilakukan proses perancangan kodifikasi pengetahuan dengan memanfaatkan *knowledge map* yang bertujuan untuk mewakili pengetahuan secara visual. Selanjutnya, dilakukan perancangan aplikasi dengan pendekatan berorientasi objek menggunakan *Unified Modeling Language (UML)* [6], [24]. Kemudian, dilakukan proses konversi rancangan tersebut ke dalam kode program untuk membentuk aplikasi berbasis web.

e. *Verifikasi dan Validasi Knowledge Management System*

Tahap berikutnya adalah melakukan pengujian pengetahuan, yang terdiri dari pengujian logika dan pengujian penerimaan pengguna. Pengujian logika bertujuan untuk memverifikasi hasil dari proses kodifikasi pengetahuan melalui keterlibatan pakar [6], [24]. Sementara itu, pengujian penerimaan pengguna menggunakan metode pengujian *blackbox* pada pakar untuk menguji fungsi-fungsi aplikasi dengan memeriksa keluaran data yang dihasilkan.

2.2.3 Model SECI

Model *SECI* telah menjadi landasan penciptaan pengetahuan dan mentransfer teori. Model ini membedakan dua dimensi pengetahuan sebagai pengetahuan *tacit* dan *explicit*, dan mengusulkan sebuah proses penciptaan pengetahuan melalui interaksi sosial untuk mengkonversi pengetahuan antara dua dimensi. *Knowledge* dibagi ke dalam bentuk *tacit knowledge* dan *explicit knowledge* sebagai dua jenis utama *knowledge* manusia [5], [12] memiliki tahapan seperti pada Gambar 2.2:



Gambar 2. 2 *SECIModel* [5], [12]

a. *Socialization*

Socialization adalah bentuk pembuatan pengetahuan yang cukup terbatas, Yang di mana SDM belajar keterampilan Pendiri. Namun baik SDM maupun Pendiri tidak mendapatkan pemahaman sistematis tentang pengetahuan keterampilan mereka. Karena pengetahuan mereka tidak pernah menjadi *explicit* yang tidak dapat dengan mudah digunakan oleh seluruh organisasi.

Dari *eksplisit* ke *eksplisit*. Seorang individu juga dapat menggabungkan potongan pengetahuan eksplisit yang terpisah menjadi kesatuan yang baru. Misalnya, ketika seorang *Controller* dari sebuah perusahaan mengumpulkan informasi dari seluruh organisasi dan menyusunnya dalam laporan keuangan, itu adalah pengetahuan baru dalam arti bahwa itu mengintegrasikan informasi dari banyak sumber yang berbeda [5].

b. *Externalization*

Externalization (mengubah pengetahuan *tacit* menjadi pengetahuan *eksplisit*) dan internalisasi (menggunakan pengetahuan *eksplisit* tersebut untuk memperluas dasar pengetahuan *tacit* sendiri) adalah langkah-langkah penting dalam spiral pengetahuan ini.

Pengetahuan *tacit* termasuk model-model mental dan keyakinan di samping *know-how*, berpindah dari *tacit* ke *eksplisit* sesungguhnya adalah proses mengungkapkan visi seseorang tentang dunia-apa yang ada dan seharusnya ada. Ketika karyawan menciptakan pengetahuan baru, mereka juga menciptakan ulang diri mereka sendiri, perusahaan, dan bahkan dunia [5].

c. *Combination*

Dari *eksplisit* ke *eksplisit*. Seorang individu juga dapat menggabungkan potongan pengetahuan eksplisit yang terpisah menjadi kesatuan yang baru. Misalnya, ketika seorang *Controller* dari sebuah perusahaan mengumpulkan informasi dari seluruh organisasi dan menyusunnya dalam laporan keuangan, itu adalah pengetahuan baru dalam arti bahwa itu mengintegrasikan informasi dari banyak sumber yang berbeda. Namun kombinasi ini sebenarnya tidak benar-benar memperluas dasar pengetahuan perusahaan yang ada [5].

d. *Internalization*

Dari eksplisit ke *tacit*. Lebih jauh lagi, saat pengetahuan eksplisit baru dibagikan di seluruh organisasi, karyawan lain mulai mencernanya, yaitu mereka menggunakannya untuk memperluas, memperpanjang, dan menyusun ulang pengetahuan *tacit* mereka sendiri. Proposal *Controller* menyebabkan revisi sistem pengendalian keuangan perusahaan. Karyawan lain menggunakan inovasi tersebut dan akhirnya datang untuk menganggapnya sebagai bagian dari latar belakang alat dan sumber daya yang diperlukan untuk melakukan pekerjaan mereka [5].

2.2.4 *Artificial Intelligence*

Artificial Intelligence atau *AI* adalah simulasi kecerdasan manusia yang diterapkan ke dalam sistem komputer atau perangkat mesin lain, sehingga perangkat tersebut punya cara berpikir seperti manusia [8]. Tujuan diciptakannya *AI* adalah untuk meniru aktivitas kognitif manusia, seperti cara belajar (*learning*), melakukan penalaran (*reasoning*), mengambil keputusan (*decision making*), dan mengoreksi diri (*self-correction*).

2.2.5 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) merupakan bidang dalam kecerdasan buatan yang berfokus pada pemahaman, analisis, dan generasi bahasa alami oleh komputer. *NLP* bertujuan untuk memungkinkan komputer berinteraksi dengan bahasa manusia secara lebih alami dan memahami konteks serta makna yang terkandung dalam teks atau ucapan [8].

NLP melibatkan penggunaan algoritma dan teknik komputasi untuk memproses, menganalisis, dan memodelkan teks dan percakapan. Beberapa tugas yang dapat dilakukan oleh *NLP* meliputi pemrosesan teks, pemahaman bahasa, terjemahan mesin, dan pembelajaran mesin berbasis teks [7], [8].

a. Pemrosesan Teks

Melibatkan langkah-langkah seperti *tokenisasi* (membagi teks menjadi unit-unit yang lebih kecil seperti kata atau frasa), pemenggalan (menghilangkan imbuhan dan menganalisis akar kata), identifikasi entitas (mengenali entitas seperti orang, tempat, tanggal, dll.), dan analisis sintaksis (memahami struktur kalimat).

b. Pemahaman Bahasa

Melibatkan pengenalan dan pemahaman makna kata, frasa, dan kalimat. Ini mencakup tugas seperti analisis sentimen (mengidentifikasi sentimen positif, negatif, atau netral dalam teks), pemahaman pertanyaan (menafsirkan pertanyaan dan memberikan jawaban yang relevan), dan pemahaman konteks (memahami konteks dan makna yang terkandung dalam teks).

c. Terjemahan Mesin

Melibatkan penerjemahan teks dari satu bahasa ke bahasa lain secara otomatis. Metode yang digunakan meliputi penggunaan model statistik dan pembelajaran mesin untuk memahami dan menerjemahkan teks dengan akurasi yang lebih tinggi.

d. Pembelajaran Mesin Berbasis Teks

Melibatkan penggunaan teks sebagai *input* untuk pelatihan model pembelajaran mesin. Metode ini memungkinkan komputer untuk mempelajari pola dan hubungan dalam teks, seperti klasifikasi teks (mengkategorikan teks ke dalam kategori yang tepat) dan pemodelan bahasa (membangun model statistik untuk memprediksi kata berikutnya dalam sebuah kalimat).

2.2.6 *OpenAPI ChatGPT*

OpenAPI, singkatan dari *Open Application Programming Interface*, adalah sebuah spesifikasi yang digunakan untuk mendefinisikan dan menggambarkan antarmuka yang terbuka untuk berinteraksi dengan suatu aplikasi atau layanan. Dalam konteks *ChatGPT* [7], [25], *OpenAPI* dapat digunakan untuk menjelaskan antarmuka atau *API* yang memungkinkan pengembang untuk berkomunikasi dengan *ChatGPT* secara programatik.

OpenAPI untuk *ChatGPT* dapat menyediakan deskripsi tentang *endpoint-endpoint* API yang tersedia, format data yang diharapkan dan dikembalikan oleh API, serta operasi-operasi yang dapat dilakukan oleh pengguna [7]. Hal ini memungkinkan pengembang untuk memahami secara jelas cara berinteraksi dengan *ChatGPT*, termasuk cara mengirimkan permintaan dan menerima respon dari model.

2.2.7 ChatBOT

Implementasi *chatbot* dengan sistem manajemen pengetahuan (*knowledge management system*) melibatkan penggunaan sistem untuk mengelola dan menyimpan pengetahuan yang diperlukan oleh *chatbot* untuk menjawab pertanyaan dan memberikan informasi kepada pengguna [8], [26].

Berikut adalah langkah-langkah umum dalam mengimplementasikan *chatbot* dengan sistem manajemen pengetahuan:

a. Identifikasi dan pencatatan pengetahuan

Tentukan jenis pengetahuan yang akan dikelola oleh *chatbot*, seperti *FAQ*, dokumentasi produk, panduan pengguna, atau sumber informasi lainnya. Kumpulkan dan kategorikan pengetahuan tersebut dalam sistem manajemen pengetahuan.

b. Desain struktur pengetahuan

Rancang struktur pengetahuan yang akan digunakan dalam sistem manajemen pengetahuan. Ini dapat mencakup pembuatan kategori, subkategori, atau *tag* untuk membantu mengorganisir dan mengelompokkan informasi.

c. Integrasi sistem manajemen pengetahuan

Hubungkan *chatbot* dengan sistem manajemen pengetahuan sehingga *chatbot* dapat mengakses dan mencari informasi yang relevan. Ini dapat melibatkan penggunaan API atau metode lain untuk berkomunikasi antara *chatbot* dan sistem manajemen pengetahuan.

d. Proses pertanyaan pengguna

Ketika pengguna mengajukan pertanyaan kepada *chatbot*, *chatbot* harus mampu memproses dan mencari jawaban yang tepat dari sistem manajemen pengetahuan. Hal ini melibatkan kemampuan *chatbot* untuk menganalisis dan memahami pertanyaan pengguna, serta mencocokkan dengan informasi yang ada dalam sistem manajemen pengetahuan.

e. Logika pemrosesan bahasa (*natural language processing*)

Implementasikan logika pemrosesan bahasa alami dalam *chatbot* untuk memahami dan menafsirkan pertanyaan pengguna dengan lebih baik. Ini melibatkan penggunaan algoritma dan teknik *NLP* untuk mengenali entitas, mengurai kalimat, dan mengekstrak informasi yang relevan.

f. Peningkatan sistem

Terus tingkatkan sistem *chatbot* dengan menerapkan umpan balik pengguna. Melalui umpan balik, Anda dapat mengidentifikasi dan memperbaiki kesalahan, mengisi kekosongan pengetahuan, atau menambahkan informasi baru ke dalam sistem manajemen pengetahuan.

Dengan mengintegrasikan *chatbot* dengan sistem manajemen pengetahuan, Anda dapat memanfaatkan pengetahuan yang ada untuk memberikan respon yang akurat dan relevan kepada pengguna, meningkatkan pengalaman pengguna, dan mengurangi kebutuhan interaksi manusia dalam menjawab pertanyaan rutin.

2.2.8 *Gradio*

Gradio adalah sebuah *framework Python* yang memungkinkan pengembang untuk membuat antarmuka pengguna interaktif untuk model pembelajaran mesin dengan cepat dan mudah. *Gradio*

memfasilitasi visualisasi dan pengujian model *ML* secara *real-time* dengan antarmuka yang dapat diakses melalui *browser* [26].

Dengan *Gradio*, pengembang dapat dengan mudah menghubungkan model *ML* yang telah dibuat sebelumnya dengan *input* dari pengguna dan menampilkan *output* model secara interaktif. *Gradio* menyediakan komponen antarmuka yang dapat digunakan untuk membuat *input* seperti teks, gambar, atau audio, dan menampilkan *output* yang dihasilkan oleh model [8], [26]. Pengguna dapat berinteraksi langsung dengan model melalui antarmuka ini dan melihat hasil prediksi atau *output* yang dihasilkan.

2.2.9 *HuggingFace*

HuggingFace adalah perusahaan dan komunitas yang terkenal dalam bidang pemrosesan bahasa alami (*NLP*) dan kecerdasan buatan. Mereka terkenal karena kontribusi mereka dalam pengembangan dan distribusi berbagai alat dan model *NLP* yang inovatif [25].

Salah satu kontribusi terpenting *HuggingFace* adalah *Transformers*, sebuah perpustakaan *Python* yang menyediakan berbagai model *NLP* yang telah dilatih sebelumnya, termasuk model *transformer* terkenal seperti *BERT*, *GPT*, dan *RoBERTa*. Melalui *Transformers*, *HuggingFace* memungkinkan para pengembang untuk dengan mudah menggunakan model-model *NLP* yang canggih dalam aplikasi mereka sendiri [8], [25].

Selain itu, *HuggingFace* juga menyediakan *API* yang memungkinkan pengguna untuk mengakses model-model *NLP* tersebut secara *online*. Ini mempermudah pengguna untuk mengintegrasikan kemampuan pemrosesan bahasa yang canggih ke dalam aplikasi mereka tanpa perlu melatih dan mengimplementasikan model-model tersebut secara mandiri.

2.2.10 Website

Sebuah situs web adalah kumpulan halaman *web* yang terhubung secara elektronik dan dapat diakses melalui jaringan seperti *Internet*. Halaman *web* tersebut bisa berisi teks, gambar, audio, video, atau kombinasi dari semua itu [6], [21]. Situs *web* digunakan untuk berbagai tujuan, seperti memberikan informasi, berbagi konten, melakukan transaksi bisnis, dan berinteraksi dengan pengguna.

Situs web umumnya memiliki tautan atau *hyperlink* yang memungkinkan pengguna untuk berpindah antara halaman-halaman yang berbeda [6]. Tautan ini membentuk struktur navigasi yang memungkinkan pengguna untuk menjelajahi dan menemukan konten yang diinginkan. Situs web juga bisa memiliki elemen interaktif, seperti formulir kontak, fitur pencarian, atau komentar pengguna.

Untuk membuat sebuah situs web, biasanya diperlukan pemrograman web dan pengetahuan tentang bahasa pemrograman seperti *HTML (HyperText Markup Language)*, *CSS (Cascading Style Sheets)*, dan *JavaScript* [27]. Namun, ada juga platform dan alat yang memungkinkan pembuatan situs web tanpa perlu pengetahuan pemrograman yang mendalam, seperti *WordPress*, *Wix*, atau *Squarespace*.

Situs web bisa diakses menggunakan perangkat seperti komputer, laptop, *smartphone*, atau tablet yang terhubung ke *Internet*. Pengguna bisa mengakses situs web dengan menggunakan peramban *web (web browser)* seperti *Google Chrome*, *Mozilla Firefox*, *Safari*, atau *Microsoft Edge*.

a. Keterjangkauan dan aksesibilitas global

Situs web memungkinkan akses global bagi pengguna di seluruh dunia. Hal ini memungkinkan bisnis atau individu untuk

mencapai audiens yang lebih luas dan potensial dari berbagai negara dan wilayah.

b. Penyediaan informasi

Situs web memungkinkan penyedia informasi untuk menampilkan konten secara terperinci dan mendalam. Hal ini memungkinkan pengguna untuk mencari, mempelajari, dan mengakses informasi dengan mudah. Situs web juga memungkinkan untuk menyajikan berbagai jenis media seperti gambar, audio, dan video untuk memperkaya pengalaman pengguna.

c. Kepraktisan dan kenyamanan

Dibandingkan dengan metode komunikasi tradisional seperti surat atau telepon, situs web menyediakan cara yang lebih cepat dan praktis untuk berinteraksi dengan pengguna. Pengguna dapat mengakses informasi atau melakukan transaksi kapan pun dan di mana pun mereka berada, selama terhubung ke Internet.

2.2.11 Headless CMS (Content Management System)

Headless CMS (Content Management System) adalah sistem manajemen konten yang terpisah antara *backend* dan *frontend*. Pada umumnya, *CMS* memiliki komponen *backend* yang mengatur konten dan *frontend* yang menampilkan konten kepada pengguna. Namun, pada model *headless CMS*, kedua komponen ini dipisahkan [28].

Dalam *headless CMS*, *backend* berperan sebagai tempat penyimpanan dan manajemen konten. Konten dapat diedit, diatur, dan disimpan dalam satu lokasi pusat. *Backend* ini menyediakan API (*Application Programming Interface*) yang memungkinkan konten diambil dan digunakan oleh berbagai platform dan perangkat, seperti

situs web, aplikasi seluler, aplikasi desktop, atau *perangkat Internet of Things (IoT)* [28].

Dalam model *headless CMS*, *frontend* memiliki fleksibilitas yang lebih besar dalam menampilkan konten. Pengguna situs atau pengembang memiliki kendali penuh terhadap tampilan dan pengalaman pengguna. *Frontend* dapat dikembangkan menggunakan teknologi seperti *React*, *Angular*, atau *Vue.js*, dan konten dapat diambil dari *backend* melalui *API* untuk ditampilkan sesuai kebutuhan [9].

2.2.12 Strapi.io

Strapi adalah sebuah *CMS headless* sumber terbuka yang memberikan kebebasan kepada pengembang untuk memilih alat dan kerangka kerja favorit mereka, serta memungkinkan editor untuk mengelola dan mendistribusikan konten melalui panel admin aplikasi mereka [9], [29]. Berdasarkan sistem *plugin*, *Strapi* adalah *CMS* yang fleksibel dengan panel admin dan *API* yang dapat diperluas - dan setiap bagian dari sistem ini dapat disesuaikan untuk memenuhi berbagai kasus penggunaan [9]. *Strapi* juga dilengkapi dengan sistem pengguna bawaan yang memungkinkan pengelolaan detail terhadap hak akses administrator dan pengguna akhir.

Strapi merupakan sistem manajemen konten yang tidak terikat dengan tampilan atau antarmuka pengguna tertentu. *Strapi* memberikan kebebasan kepada pengembang untuk memilih dan menggunakan alat serta kerangka kerja yang mereka sukai dalam mengembangkan aplikasi [28]. Para editor, di sisi lain, dapat dengan mudah mengelola dan mendistribusikan konten melalui panel admin yang terintegrasi dalam aplikasi mereka.

2.2.13 Next.js

Next.js adalah sebuah kerangka kerja (*framework*) sisi *server* (*server-side*) yang digunakan untuk pengembangan aplikasi web dengan menggunakan bahasa pemrograman *JavaScript*. Dikembangkan oleh tim *Vercel*, *Next.js* memungkinkan pengembang untuk membuat aplikasi web dengan mudah dan efisien [9].

Next.js menggunakan konsep rendah *server-side rendering* (*SSR*) yang memungkinkan pembuatan halaman *web* yang dapat di-*render* di sisi server sebelum dikirim ke browser pengguna. Dengan *SSR*, pengguna dapat melihat konten dasar halaman dengan lebih cepat, meningkatkan kecepatan dan pengalaman pengguna [9]. Selain *SSR*, *Next.js* juga mendukung pembuatan aplikasi dengan *rendering* di sisi klien (*client-side rendering*) dan generasi *statis* (*static generation*) untuk mengoptimalkan performa aplikasi.

Keunggulan utama dari *Next.js* adalah kemudahan penggunaan dan fleksibilitasnya. Dengan menggunakan *Next.js*, pengembang dapat membuat aplikasi *web* dengan cepat dan efisien karena kerangka kerja ini menyediakan struktur dan alat bantu yang kuat. *Next.js* juga mengintegrasikan *routing* yang sederhana dan otomatis, memudahkan navigasi antar halaman dalam aplikasi [9].

2.2.14 TailwindCSS

Tailwind CSS adalah sebuah kerangka kerja *CSS* yang inovatif yang dikembangkan dengan tujuan mempermudah proses pengembangan antarmuka pengguna (*UI*) dalam pengembangan *web*. Dalam pendekatan yang berbeda dengan kebanyakan kerangka kerja *CSS* lainnya, *Tailwind CSS* tidak menyediakan komponen *UI* pra-dibuat, melainkan menawarkan berbagai utilitas kelas *CSS* yang dapat digunakan untuk membangun komponen *UI* sesuai kebutuhan peneliti [10].

Salah satu kelebihan *Tailwind CSS* adalah pendekatan "*Utility-First*" yang diusungnya. Dengan menggunakan kelas-kelas utilitas CSS yang tersedia, peneliti dapat dengan mudah mengatur properti-properti seperti ukuran, warna, margin, dan *padding* untuk elemen-elemen *UI* tanpa harus menulis CSS khusus untuk setiap elemen tersebut [9], [10]. *TailwindCSS* memberikan fleksibilitas yang lebih besar dalam merancang tampilan sesuai dengan kebutuhan penelitian.

2.2.15 NodeJS

Node.js adalah *environment runtime JavaScript open-source, cross-platform* yang mengeksekusi kode *JavaScript* di luar *web browser* yang dibangun dengan *engine JavaScript Chrome V8* dan memungkinkan para pengembang menjalankan *JavaScript* di sisi *server*, menciptakan aplikasi sisi *server* dengan *JavaScript* [30]. *Node.js* sering digunakan untuk membangun program jaringan, seperti *server web*, dan untuk menciptakan aplikasi *web real-time*, seperti aplikasi *chat* dan *game online*. Selain itu *Node.js* juga populer untuk membangun *command-line tools* dan untuk menciptakan *script* otomatisasi tugas.

2.2.16 Yarn

Yarn adalah sebuah alat manajemen paket yang banyak digunakan dalam pengembangan perangkat lunak dengan fokus pada bahasa pemrograman *JavaScript*. Dikembangkan oleh beberapa perusahaan teknologi terkemuka, *Yarn* dirancang untuk meningkatkan efisiensi, kehandalan, dan keamanan dalam pengelolaan dan instalasi paket-paket *JavaScript* [31].

Salah satu keunggulan *Yarn* bagi peneliti adalah kecepatannya. Dalam proses instalasi paket-paket, *Yarn* memanfaatkan *caching* dan penggunaan paralelisme untuk meningkatkan kecepatan [31]. Hal ini sangat bermanfaat ketika peneliti bekerja dengan proyek yang

memiliki banyak dependensi, karena *Yarn* dapat mempercepat proses instalasi dan pengelolaan paket-paket.

2.2.17 VPS

VPS (Virtual Private Server) adalah sebuah layanan *hosting* yang menyediakan lingkungan *server* virtual terisolasi di dalam satu server fisik [29]. Dalam hal ini, *server* fisik dibagi menjadi beberapa bagian yang berfungsi sebagai *server* virtual independen.

Setiap *VPS* memiliki sumber daya seperti *CPU*, *RAM*, dan penyimpanan yang didedikasikan untuk pengguna atau pemilik *VPS*. Dalam konteks ini, *VPS* dapat dianggap sebagai komputer virtual yang berjalan di atas *server* fisik [29].

Salah satu keuntungan utama menggunakan *VPS* adalah fleksibilitas dan kontrol yang lebih besar daripada *shared hosting*. Pengguna *VPS* memiliki akses penuh sebagai *root* atau administrator ke lingkungan server virtual mereka, memungkinkan mereka untuk meng-*instal*, mengonfigurasi, dan mengelola perangkat lunak dan layanan sesuai kebutuhan mereka [29]. Berbeda dengan *shared hosting* di mana pengguna harus berbagi sumber daya dengan pengguna lain di server yang sama.

2.2.18 BlackBox Testing

Blackbox testing merupakan metode pengujian perangkat lunak yang penting dalam rangka menguji fungsionalitas dan kualitas sistem tanpa memerhatikan detail implementasinya. Dalam *blackbox testing*, peneliti memperlakukan sistem sebagai kotak hitam atau entitas yang tidak diketahui secara internal, dan fokus pada perilaku eksternal dan persyaratan yang diharapkan [27].

Dalam *blackbox testing*, peneliti tidak memiliki akses langsung ke kode sumber atau *rincian* implementasi internal sistem yang sedang diuji. Sebaliknya, mereka hanya menggunakan *input* dan *output* yang terlihat untuk memeriksa apakah sistem berperilaku sesuai dengan harapan dan persyaratan yang ditentukan sebelumnya [27]. Hal ini memungkinkan peneliti untuk memeriksa fungsionalitas sistem dari sudut pandang pengguna akhir atau pengamat eksternal..