

BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

PT XYZ merupakan perusahaan yang bergerak di industri Teknologi Informasi, yang fokus pada pengembangan solusi perangkat lunak dan layanan informatika. Didirikan pada tahun 2006, PT XYZ memiliki komitmen untuk menyediakan teknologi terkini kepada klien-klien di berbagai industri. Perusahaan ini bertujuan menciptakan solusi inovatif dan terdepan, serta melakukan analisis data secara cepat dan efisien dengan memanfaatkan mesin pengolahan yang terintegrasi. Sebagai perusahaan yang menyediakan layanan teknologi, PT. XYZ menangani data dalam jumlah besar dan terus berkembang. Namun, sistem konvensional yang digunakan saat ini seperti melakukan pencarian data menggunakan kertas yang memiliki keterbatasan dalam mencari informasi yang relevan, cepat, dan efisien diantara dokumen-dokumen digital yang tersebar dalam berbagai format, termasuk PDF. Keterbatasan ini mengakibatkan proses pencarian informasi yang memakan waktu, menghambat produktivitas, serta mengurangi efisiensi operasional. Dalam upaya memenuhi kebutuhan perusahaan akan pengelolaan data yang lebih efektif, PT XYZ membutuhkan sistem yang mampu mendukung pengambilan keputusan dengan akses informasi yang cepat dan relevan. Penelitian ini berfokus pada pengembangan dan optimasi sistem pencarian yang menggabungkan kemampuan *machine learning* untuk pemrosesan bahasa alami dengan MySQL sebagai basis data terdistribusi yang andal. Sistem ini diharapkan dapat memberikan hasil pencarian yang relevan dan akurat dengan waktu respons yang cepat, serta mampu menangani beban permintaan yang tinggi.

Web application yang akan dirancang akan memanfaatkan data yang diperoleh langsung dari PT XYZ. Data yang diberikan akan dimasukkan kedalam *database* dan akan menjadi sumber data. Hal ini dilakukan karena ketersediaan data yang luas dan kredibilitasnya, sehingga dapat memastikan keakuratan dan konsistensi dalam analisis yang dilakukan. Dalam penelitian ini, akan menggunakan model BERT. Model ini dipilih karena kemampuannya dalam melakukan analisis dengan pendekatan *text processing* yang intuitif serta mampu menangkap pola dan tren dengan baik. Hal ini memungkinkan model untuk memberikan hasil yang cukup akurat. Model BERT yang merupakan model *pre-trained* dari Google, digunakan untuk memahami konteks dalam data tekstual yang berkaitan dengan *text processing*. BERT mampu menghasilkan representasi kata yang lebih akurat dan kontekstual, sehingga dapat meningkatkan kualitas prediksi.

Penelitian ini juga akan mengintegrasikan model RAG (*Retrieval-Augmented Generation*) untuk meningkatkan kemampuan pencarian informasi. Model RAG memungkinkan sistem tidak hanya mencari informasi tetapi juga menghasilkan jawaban relevan berdasarkan konteks yang ditemukan. Dengan menggabungkan RAG dan ChatGPT-4 API, sistem dapat memberikan output responsif yang lebih interaktif dan kontekstual bagi pengguna. Implementasi RAG diharapkan dapat mempercepat proses pengambilan keputusan dan meningkatkan kepercayaan pengguna terhadap informasi yang dihasilkan, sementara ChatGPT-4 API akan memperkaya pengalaman pengguna dengan kemampuan interaksi berbasis teks yang lebih natural.

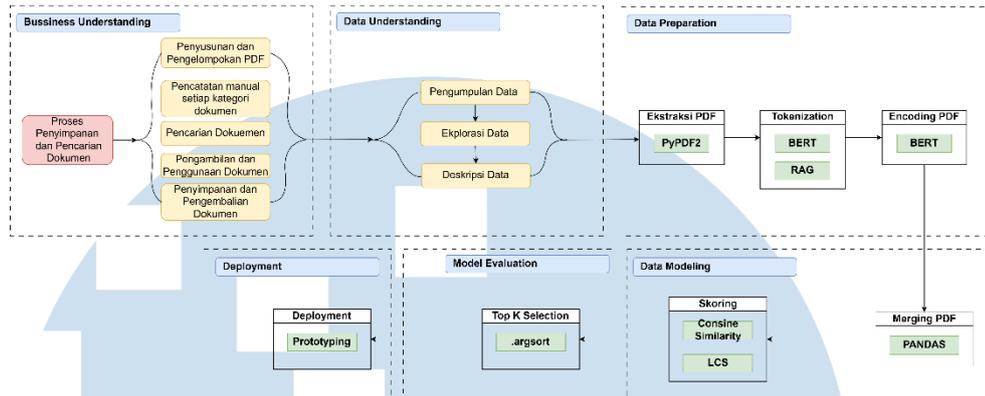
3.2 Metode Penelitian

Metode penelitian merupakan bagian penting dalam sebuah penelitian yang berfungsi sebagai panduan sistematis untuk mencapai tujuan penelitian. Dalam penelitian ini, metode yang digunakan dirancang untuk memastikan proses pengembangan *search engine* berbasis *machine learning* dapat berjalan secara terstruktur dan efisien.

3.1 Alur Penelitian

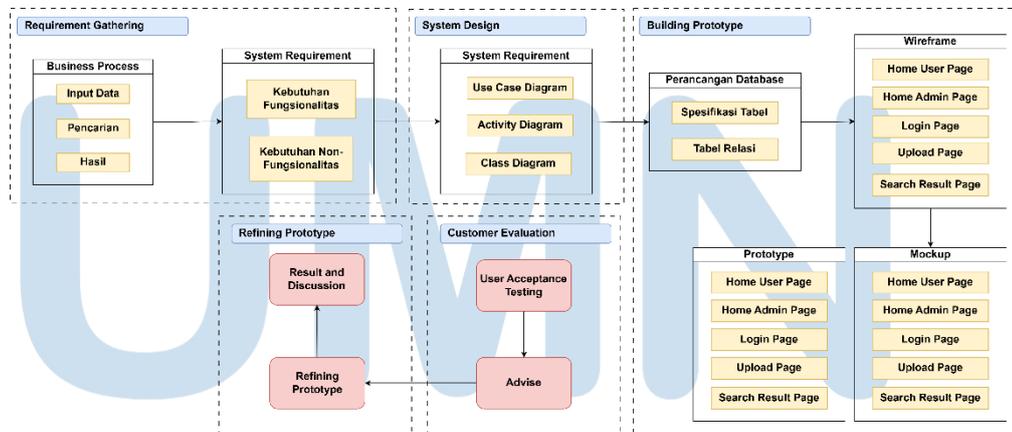
Proses penelitian dalam studi ini dijelaskan melalui *diagram* alur yang divisualisasikan menggunakan *flow chart*. Gambar 3.2 menampilkan *diagram* yang merangkum tahapan penelitian secara sistematis. *Diagram* ini mencakup langkah-langkah yang akan dilakukan dalam penelitian ini. Pengembangan sebuah sistem, terdapat beberapa tahap yang saling berkaitan. Pada Gambar 3.4 merupakan tahap tahap yang akan dilakukan untuk mencapai maksud dan tujuan dari penelitian. Alur penelitian dibagi menjadi lima fase yaitu fase *bussines understanding*, *data understanding*, *data preparation*, *data modeling*, *model evaluation*, dan *deployment*. Pada tahap *deployment*, sistem akan dikembangkan menggunakan *framework prototyping*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3. 1 Alur Penelitian Data Mining

Tahapan pertama dalam proses penelitian ini mengikuti metodologi CRISP-DM dengan fokus pada *Business Understanding*, yaitu analisis kebutuhan berdasarkan proses bisnis. Selanjutnya, pada tahap *Data Understanding*, pemilihan variabel dalam dataset dilakukan untuk memahami karakteristik data yang relevan. Pada tahap *Data Preparation*, data dipersiapkan dengan melakukan pembersihan dan transformasi agar siap digunakan dalam pembuatan model. Pada tahap *Data Modeling*, model dikembangkan menggunakan algoritma BERT untuk meningkatkan akurasi pencarian informasi. Tahap *Model Evaluation* bertujuan untuk menguji akurasi setiap model yang dihasilkan.



Gambar 3. 2 Alur Penelitian *Prototyping*

Terakhir, tahap *Deployment* berfokus pada implementasi model yang sudah dibangun agar dapat diakses melalui aplikasi web. Pada fase *Deployment*, sistem dikembangkan menggunakan metodologi *prototyping*, yang dimulai dengan pengembangan sistem dan dilanjutkan dengan pengujian untuk memastikan

fungsionalitasnya. Di akhir tahap ini, evaluasi terhadap sistem dilakukan untuk memastikan kualitas dan efektivitasnya. Pada fase *Hasil*, seluruh hasil penelitian disajikan, termasuk komparasi dengan penelitian terdahulu menggunakan tabel klasifikasi untuk menunjukkan relevansi dan perbandingan. Setelah tahap *Result and Discussion*, kesimpulan ditarik berdasarkan temuan-temuan yang ditemukan selama fase penelitian ini.

3.2 Metodologi Data Mining

Terdapat beberapa metodologi yang umum digunakan dalam proses data mining [59]. Pada penelitian ini menggunakan framework CRISP-DM. Alasan memilih *framework* tersebut berdasarkan kelebihan dan kekurangan yang ditampilkan pada Tabel 3.1. Setiap metodologi memiliki karakteristik unik serta keunggulan masing-masing dalam mendukung analisis data. Tabel 3.1 adalah perbandingan dari berbagai metodologi *data mining* yang sering diterapkan, termasuk karakteristik dan keunggulan masing-masing.

Tabel 3. 1 Perbandingan Metodologi Data Mining

Indikator	CRISP-DM [60]	KDD [44]	SEMMA [59]
Definisi	Metodologi data mining yang banyak digunakan di industri, dengan tahapan jelas dan terstruktur.	<i>Framework</i> yang mencakup seluruh proses data mining dari ekstraksi hingga penemuan pengetahuan.	<i>Framework</i> yang fokus pada proses pembuatan model prediktif melalui tahap-tahap eksplorasi data.
Kelebihan	<ol style="list-style-type: none"> 1. Dokumentasi lengkap dan umum digunakan 2. Struktur yang jelas dan berfokus pada kebutuhan bisnis. 	<ol style="list-style-type: none"> 1. Fokus pada proses penemuan pengetahuan (knowledge discovery). 2. Mencakup seluruh proses data mining dan memungkinkan iterasi untuk hasil lebih baik. 	<ol style="list-style-type: none"> 1. Iteratif dengan pendekatan trial dan error yang fleksibel. 2. Fleksibel untuk eksplorasi data yang mendalam.
Kekurangan	<ol style="list-style-type: none"> 1. Kurang menangani outlier pada tahap preparation 2. Memakan waktu yang cukup lama pada tahap persiapan data 	<ol style="list-style-type: none"> 1. Iteratif yang berulang-ulang dapat memakan waktu lebih lama. 2. Kompleksitasnya tinggi. 	<ol style="list-style-type: none"> 1. Tidak mencakup tahap Business Understanding, sehingga kurang memahami kebutuhan bisnis. 2. Kurang kuat pada tahapan evaluasi dan deployment.

Berdasarkan analisis kelebihan dan kekurangan dalam tabel 3.3, CRISP-DM (*Cross-Industry Standard Process for Data Mining*) menawarkan pendekatan yang terstruktur dan sistematis dalam proses data mining. Dibandingkan dengan metode lain seperti KDD dan SEMMA, CRISP-DM memiliki keunggulan dalam fokusnya pada evaluasi dan penerapan model. Dalam konteks penelitian ini, yang berfokus pada optimalisasi pencarian informasi di *Knowledge Management System* menggunakan machine learning dan SQL, CRISP-DM menyediakan kerangka kerja yang lebih terstruktur dan efisien untuk memastikan bahwa solusi yang dikembangkan dapat memenuhi kebutuhan operasional. Ada beberapa langkah yang akan dilalui saat menerapkan metodologi CRISP-DM. Berikut adalah langkah-langkah tersebut:

1. *Business Understanding*

Tahap ini mencakup pemahaman menyeluruh terhadap konteks bisnis untuk menilai sumber daya yang tersedia dan menentukan tujuan utama dari proses data mining yang akan dilakukan.

2. *Data Understanding*

Pada fase ini, data dikumpulkan dari berbagai sumber, dieksplorasi, dan dideskripsikan. Pemeriksaan kualitas data juga dilakukan untuk memastikan data yang akan diproses sudah sesuai.

3. *Data Preparation*

Fase ini mencakup pemilihan data berdasarkan kriteria tertentu serta pembersihan data yang memiliki kualitas kurang baik, sehingga data yang diolah memiliki validitas yang tinggi.

4. *Data Modeling*

Dalam fase pemodelan, teknik pemodelan yang tepat dipilih dan diuji. Test case dibuat, dan model dibangun menggunakan metode yang paling sesuai untuk keperluan analisis data dalam penelitian ini.

5. *Model Evaluation*

Model yang telah dibangun kemudian dievaluasi berdasarkan tujuan bisnis awal untuk memastikan kesesuaian dan efektivitasnya.

6. *Deployment*

Fase ini melibatkan implementasi model dalam sistem prototipe yang telah dikembangkan, sehingga integrasi solusi data mining dengan sistem produksi berjalan lancar. Integrasi ini akan menjadikan proses implementasi lebih

terstruktur dan mengurangi potensi risiko kesalahan atau kendala yang mungkin muncul selama proses tersebut.

3.3 Metode Pengembangan Sistem

Penelitian ini menggunakan metode kuantitatif karena fokus utamanya adalah pada pengembangan sistem. Ada berbagai model dalam system development life cycle (SDLC) yang bisa diterapkan, namun dalam penelitian ini, model prototyping dipilih sebagaimana dijelaskan dalam Tabel 3.2 [61]. *Prototyping* adalah metode pengembangan sistem yang bertujuan untuk menciptakan sebuah model yang dapat diuji oleh pengguna untuk mendapatkan umpan balik. Dengan demikian, peneliti dapat memperbaiki kualitas model tersebut pada tahap revisi, dengan mempertimbangkan pencapaian tujuan sistem [62]. Metode *prototyping* ini akan dibandingkan dengan dua model SDLC lainnya, yaitu *waterfall* dan *Rapid Application Development (RAD)*. Model *waterfall* adalah prosedur pengembangan sistem yang dilakukan secara berurutan, di mana setiap proses berjalan seperti air terjun yang mengalir ke bawah melalui beberapa fase, seperti perencanaan, pemodelan, implementasi, dan pengujian [63]. Sementara itu, RAD adalah metode pengembangan sistem yang bersifat linier berurutan dan menekankan pada pengurangan waktu yang dibutuhkan untuk pembuatan system [64]. Perbandingan antara ketiga model SLDC ini disajikan dalam bentuk Tabel 3.2 [62].

Tabel 3.2 Perbandingan Metode *Waterfall*, *Prototyping* dan RAD

Parameter	<i>Prototyping</i>	<i>RAD</i>	<i>Waterfall</i>
Pendekatan	Iteratif dan incremental	Linear sequential, namun cepat dan iterative	Linear sequential
Tujuan utama	Menghasilkan model untuk diuji oleh <i>user</i>	Mengurangi durasi pengembangan	Melalui semua fase secara berurutan
Keterlibatan Pengguna	Tinggi	Tinggi	Rendah
Fleksibilitas	Relatif cepat, tergantung feedback	Sangat cepat, fokus pada pengurangan durasi	Lambat, karena setiap fase harus diselesaikan sebelum lanjut
Biaya	Bervariasi	Tinggi	Tinggi
Kualitas hasil	Tinggi	Tinggi	Tinggi
Resiko kesalahan	Rendah	Rendah	Tinggi

Dokumentasi	Minimal pada tahap awal	Lengkap namun dilakukan secara cepat dan tepat	Sangat lengkap dan terperinci di setiap fase
-------------	-------------------------	--	--

Berdasarkan Tabel 3.2 yang telah disusun, keputusan untuk memilih model *prototyping* sebagai metode pengembangan sistem diperkuat oleh alasan bahwa *prototyping* memberikan solusi yang seimbang antara dua model lainnya, yaitu *Waterfall* dan RAD. Mengingat waktu yang terbatas untuk pembangunan sistem, model *Waterfall* lebih menekankan pada prosedur perancangan yang terstruktur dan mekanisme perancangan sistem yang diharapkan, karena sistem yang dikembangkan harus berfokus pada kebutuhan, arahan, dan petunjuk dari pengguna.

Mekanisme seperti ini dapat dihadirkan oleh model *prototyping* dan RAD. *Prototyping* dipilih sebagai pilihan utama dibandingkan RAD karena, meskipun *prototyping* bersifat fleksibel dan mendukung pengembangan sistem dalam waktu singkat, model ini tetap menjaga keseimbangan antara pemenuhan kebutuhan pengguna dan waktu yang tersedia, sehingga waktu yang ada dapat dimanfaatkan secara optimal. Selain itu, model RAD biasanya diterapkan pada proyek yang melibatkan beberapa tim pengembang aplikasi secara modular, yang tidak sesuai dengan proyek ini karena dikerjakan secara individu tanpa kolaborasi tim.

Metode ini dipilih karena sangat sesuai untuk pengembangan sistem yang dapat disesuaikan (*customizable*), di mana sistem dapat diubah sesuai dengan kebutuhan dan permintaan spesifik. Alasan lain yang mendasari pemilihan metode *prototyping* dalam penelitian ini adalah:

1. Calon pengguna dari aplikasi dapat melakukan pengujian terhadap sistem yang sedang dikembangkan serta memberikan masukan yang bermanfaat untuk penyempurnaan sistem.
2. Metode ini memungkinkan identifikasi terhadap fungsi-fungsi yang belum tersedia pada sistem.
3. Mempermudah deteksi kesalahan pada sistem sehingga perbaikan dapat dilakukan sebelum sistem melangkah ke tahap pengembangan lebih lanjut.

Metode *prototyping* berfokus pada pengembangan aplikasi berbasis web yang dapat memenuhi kebutuhan pengguna. Berikut adalah tahapan-tahapan yang dilakukan dengan metode ini:

1. *Requirement Gathering*

Pada tahap ini, analisis terhadap kebutuhan dilakukan dengan mengidentifikasi kebutuhan bisnis pada aplikasi berbasis web ini.

2. *Quick Design*

Pada tahap *Quick Design*, perancangan awal dilakukan dengan membuat *wireframes* dan *mockups* menggunakan Figma. Desain awal aplikasi ini disesuaikan dengan kebutuhan pengguna yang diperoleh dari survei pada tahap sebelumnya, memberikan gambaran awal antarmuka pengguna dari aplikasi.

3. *Building Prototype*

Setelah kebutuhan pengguna dan antarmuka pengguna awal ditetapkan, prototipe aplikasi dibangun menggunakan *framework* yang sesuai, seperti Django.

4. *Customer Evaluation*

Setelah prototipe selesai, pengguna akhir yang menjadi target aplikasi akan diberikan akses untuk mencoba prototipe. Masukan yang diberikan oleh pengguna akan menjadi dasar untuk pengembangan lebih lanjut.

5. *Refining Prototype*

Pada tahap ini, peneliti memperbaiki prototipe berdasarkan masukan dari pengguna akhir yang telah mencoba aplikasi, sehingga sistem yang dihasilkan lebih sesuai dengan kebutuhan dan harapan pengguna.

3.4 *Data Modelling*

Tabel 3.5 menyajikan perbandingan antara BERT dan LSTM, yang mencakup karakteristik dan keunggulannya dalam pemrosesan bahasa alami (*Natural Language Processing*). Kedua algoritma ini sering digunakan dalam berbagai penelitian dan aplikasi yang berfokus pada pencarian informasi cerdas. Pada penelitian ini menggunakan model BERT. Selain berdasarkan penelitian terdahulu, BERT dipilih berdasarkan kelebihan dan kekurangan yang disajikan pada Tabel 3.3.

Tabel 3. 3 Tabel perbandingan metode *problem solving*

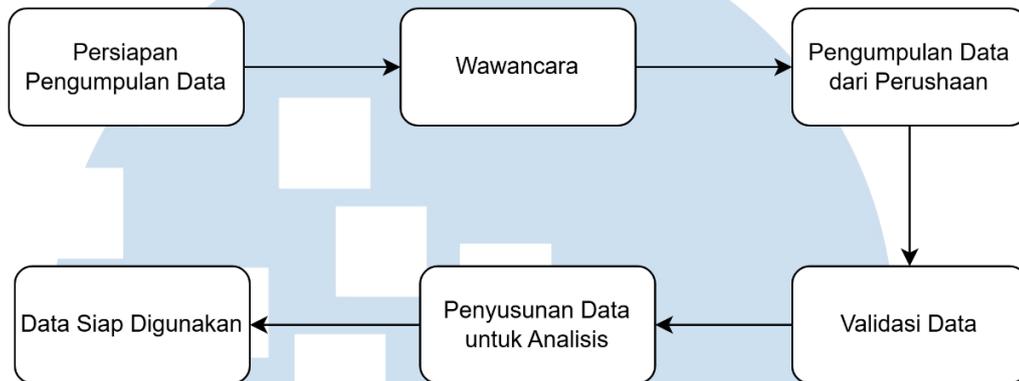
Indikator	BERT [65]	LSTM [66]
Jenis model	Pretrained language model berbasis Transformer yang memproses input secara bidirectional.	Jenis Recurrent Neural Network (RNN) yang dirancang untuk mengatasi masalah vanishing gradient.
Karakteristik	1. Memproses teks secara bidirectional, memungkinkan pemahaman konteks yang lebih dalam.	1. Memproses data secara urut, dengan informasi yang dibawa sepanjang waktu oleh cell memory. 2. Dirancang khusus

	<ol style="list-style-type: none"> Menggunakan mekanisme self-attention untuk menangkap hubungan antar kata dalam kalimat. Di-training dengan berbagai jenis data dan dapat diterapkan pada berbagai tugas NLP. 	<ul style="list-style-type: none"> untuk menangani dependensi jangka panjang dalam data urutan. Sering digunakan untuk masalah berbasis waktu atau urutan.
Kelebihan	<ol style="list-style-type: none"> Pemahaman kontekstual yang lebih mendalam berkat pemrosesan bidirectional. Efektif untuk tugas NLP seperti analisis sentimen, penerjemahan bahasa, dan ekstraksi informasi. Transfer learning yang kuat, dapat diterapkan ke banyak domain tanpa pelatihan ulang dari awal. 	<ol style="list-style-type: none"> Memiliki kemampuan untuk menangkap dependensi jangka panjang dalam urutan data. Lebih baik dalam menangani masalah yang melibatkan urutan atau data berbasis waktu, seperti teks berurutan atau analisis urutan kejadian. Memiliki mekanisme pembelajaran yang efektif untuk tugas-tugas berbasis urutan dengan data terbatas.
Kekurangan	<ol style="list-style-type: none"> Membutuhkan banyak sumber daya komputasi, terutama untuk pelatihan model besar. Pemrosesan lebih lambat pada model yang lebih besar. 	<ol style="list-style-type: none"> Tidak efektif untuk menangkap konteks global dalam teks yang lebih panjang. Memerlukan banyak data untuk menghindari overfitting dan mengoptimalkan hasil. Proses pelatihan lebih lama dan sulit diatur untuk masalah yang lebih kompleks.

Perbandingan Tabel 3.3 mengungkapkan bahwa BERT unggul dalam memahami konteks teks secara lebih mendalam. Keunggulan BERT terletak pada kemampuannya yang fleksibel dalam menangani berbagai tugas NLP yang kompleks. Oleh karena itu, BERT lebih cocok digunakan dalam pengembangan sistem pencarian informasi dan aplikasi berbasis bahasa alami yang membutuhkan pemahaman kontekstual yang lebih kuat.

3.3 Teknik Pengumpulan Data

Pada penelitian ini, terdapat beberapa tahap teknik pengumpulan data. Sementara itu, data perusahaan diperoleh berupa dokumen dan informasi teknis yang relevan untuk mendalami implementasi sistem.



Gambar 3. 3 Alur Pengumpulan Data

Pada penelitian ini, proses pengumpulan data dilakukan dengan cara terstruktur pada Gambar 3.2. Tahap pertama adalah persiapan pengumpulan data, yang dimulai dengan menyusun instrumen wawancara berupa daftar pertanyaan yang relevan dengan penelitian. Pertanyaan ini dirancang untuk menggali informasi mengenai penggunaan sistem pencarian informasi dalam. Selanjutnya, sumber data yang akan diwawancarai, seperti manajer, staff IT, atau pengguna sistem, diidentifikasi, dan jadwal wawancara disusun sesuai dengan ketersediaan responden.

Tahap kedua adalah pengumpulan data melalui wawancara, di mana wawancara dilakukan dengan pihak-pihak yang telah ditentukan. Selama wawancara, peneliti mencatat atau merekam percakapan untuk memastikan bahwa data dapat dianalisis dengan baik. Selain itu, catatan lapangan dibuat untuk mencatat poin-poin penting yang dibahas, termasuk reaksi responden terhadap pertanyaan. Setelah wawancara, data yang diperoleh dikonfirmasi kembali kepada responden untuk menghindari kesalahan interpretasi.

Tahap ketiga melibatkan pengumpulan data dari perusahaan, yang dilakukan melalui koordinasi dengan pihak perusahaan untuk mendapatkan akses ke dokumen yang relevan, yaitu file PDF atau data pengguna. Data yang diminta mencakup laporan, file, atau log aktivitas sistem yang mendukung analisis.

Tahap keempat adalah validasi data, di mana kualitas dan relevansi data yang diperoleh dari wawancara dan perusahaan diverifikasi. Data juga dibandingkan untuk memastikan konsistensi dan kesesuaian informasi yang diterima. Tahap kelima adalah penyusunan data untuk analisis, di mana data dikelompokkan berdasarkan kategori tertentu, seperti

pola penggunaan sistem pencarian dan kendala yang dihadapi. Hasil wawancara yang direkam ditranskripsikan, dan semua data disimpan dalam format yang terstruktur untuk mempermudah analisis lebih lanjut.

Terakhir, data yang telah dikumpulkan dan disusun dilaporkan dalam tahap penyusunan laporan, yang merinci seluruh proses pengumpulan data, temuan dari wawancara, dan data teknis dari perusahaan. Laporan ini menjadi dasar untuk analisis lebih lanjut mengenai sistem pencarian informasi yang digunakan dan area yang dapat ditingkatkan dalam penelitian ini. Dengan mengikuti langkah-langkah ini, proses pengumpulan data dapat dilakukan secara menyeluruh dan mendukung tujuan penelitian dalam pengembangan sistem pencarian informasi.

3.3.1 Populasi dan Sampel

Pada penelitian ini terdapat sumber data yang akan digunakan pada penelitian ini dan tempat dimana data tersebut diperoleh. Data terbagi menjadi dua jenis yaitu data primer dan juga data sekunder.

1. Data Primer

Data primer pada penelitian ini merupakan hasil wawancara yang telah dilakukan sebelumnya. Selain wawancara, data primer dapat didapatkan dengan berbagai metode lain seperti survei maupun observasi.

2. Data Sekunder

Data sekunder pada penelitian ini merupakan hasil data yang telah dikumpulkan oleh orang lain. Hasil data tersebut dapat dikumpulkan dari berbagai sumber seperti data perusahaan, buku, jurnal, maupun situs web. Pada penelitian ini, data sekunder digunakan sebagai dasar dari topik penelitian ini.

3.3.2 Periode Pengambilan Data

Dalam penelitian ini, data dikumpulkan melalui dua periode yang terstruktur. Periode pertama melibatkan wawancara dengan pihak terkait untuk memperoleh wawasan mengenai penggunaan sistem pencarian informasi dalam. Periode kedua difokuskan pada pengumpulan data teknis dari perusahaan, seperti dokumen dan log aktivitas sistem, yang mendukung analisis lebih mendalam:

1. Periode pertama

Periode pertama dilaksanakan pada tanggal 1 Agustus 2024 Pada periode ini dilakukan wawancara dengan pihak yang terkait pada PT. XYZ serta kebutuhan-kebutuhan pertama yang diinginkan untuk memuat sistem. Setelah menerima *list*, kebutuhan yang diinginkan perusahaan, maka hal ini mendukung proses pembuatan laporan dan sistem.

2. Periode kedua

Periode kedua dilaksanakan pada tanggal 1 Oktober 2024 Pada periode ini dilakukan pengecekan apakah kebutuhan yang sebelumnya sudah memenuhi dan menyelesaikan masalah atau belum. Setelah berdiskusi maka akan dilakukan pengamatan pada sistem dan menambahkan serta mempermudah identifikasi yang akan dibuat pada laporan dan sistem.

3.4 Teknik Analisis Data

Terdapat beberapa alat yang akan digunakan untuk mendukung proses perancangan aplikasi *web* dalam penelitian ini. Setiap *tools* memiliki kelebihan dan kekurangan yang disesuaikan dengan kebutuhan integrasi sistem *Knowledge Management*. Berikut perbandingan dari opsi alat yang digunakan dalam penelitian yang ditampilkan pada Tabel 3.4.

Tabel 3. 4 Pebandingan *code editor*

Indikator	Visual Studio Code [67]	Atom [68]
Harga	Gratis (<i>open source</i>)	Gratis (<i>open source</i>)
Performa	Lebih cepat dibanding editor berbasis Electron	Lebih lambat dari Visual Studio Code
<i>Setting Up</i>	Memiliki beberapa fitur <i>built-in</i> yang dapat digunakan untuk memulai menulis kode tanpa harus menginstal ekstensi	Untuk menikmati fitur dari Atom, pengguna harus melakukan instalasi ekstensi terlebih dahulu
Ukuran	Sekitar 70 MB	Sekitar 190 MB
Kostumisasi	Dapat melakukan kustomisasi seperti tema, font, dll.	<i>Fully hackable</i> , memungkinkan kustomisasi pada Atom dilakukan dengan mudah

Berdasarkan perbandingan indikator pada tabel 3.4, Visual Studio Code dipilih sebagai code editor dalam penelitian ini karena beberapa alasan. Visual Studio Code memiliki performa yang lebih ringan dan cepat dibandingkan Atom, dengan ukuran aplikasi yang

lebih kecil, yaitu sekitar 70 MB dibandingkan 190 MB pada Atom. Selain itu, Visual Studio Code mempunyai fitur-fitur bawaan yang lengkap, sehingga pengguna dapat langsung memulai penulisan kode tanpa perlu menginstal tambahan ekstensi. Di sisi ekosistem, Visual Studio Code juga didukung oleh berbagai extension yang memperluas fungsionalitas editor, serta menyediakan opsi kustomisasi tema, font, dan elemen lainnya. Hal ini membuat Visual Studio Code lebih fleksibel dan mudah disesuaikan dengan kebutuhan pengguna, menjadikannya pilihan yang tepat untuk penelitian ini.

Tabel 3. 5 Perbandingan *framework* python

Indikator	Django [69]	Flask [70]
Harga	<i>Open source</i>	<i>Open source</i>
Arsitektur	Menggunakan arsitektur <i>MVC (Model-View-Controller)</i> .	Menggunakan arsitektur <i>MVC (Model-View-Controller)</i> , namun lebih fleksibel dan dapat menggunakan arsitektur lain seperti <i>MVT (Model-View-Template)</i> .
<i>Database</i>	Memiliki fitur ORM (<i>Object-Relational Mapping</i>) bawaan yang mempermudah pengembang untuk berinteraksi dengan <i>database</i> menggunakan objek Python.	Tidak memiliki fitur ORM bawaan, tetapi dapat menggunakan library ORM seperti SQLAlchemy.
<i>Admin Interface</i>	Django menyediakan halaman <i>admin</i> bawaan (<i>Django Admin</i>) yang memudahkan kustomisasi data.	Flask tidak memiliki fitur <i>admin</i> bawaan, namun dapat menggunakan ekstensi seperti <i>Flask-Admin</i> untuk fitur <i>admin</i>
Komunitas	Django memiliki komunitas yang besar dengan dokumentasi, tutorial, dan library yang lengkap.	Flask memiliki komunitas yang lebih kecil dibandingkan Django, namun tetap menyediakan dokumentasi dan ekstensi yang cukup lengkap.

Berdasarkan tabel 3.5, Django dan Flask merupakan dua *framework* web berbasis Python yang populer dengan keunggulan masing-masing. Dalam penelitian ini, Flask dipilih sebagai *framework* utama untuk pengembangan Web Application karena beberapa alasan penting. Pertama, fleksibilitas arsitektur Flask memungkinkan penyesuaian yang lebih mudah sesuai dengan kebutuhan spesifik sistem yang dikembangkan. Kedua, kemampuan Flask untuk diintegrasikan dengan berbagai ekstensi seperti SQLAlchemy

dan Flask-Admin memberikan kemudahan dalam pengelolaan *database* dan *administrasi* tanpa membebani sistem dengan fitur yang tidak diperlukan. Selain itu, ukuran aplikasi Flask yang lebih kecil dan performa yang lebih cepat mendukung efisiensi dalam pengembangan dan penerapan model BERT untuk optimasi pencarian informasi. Dengan demikian, Flask menyediakan kerangka kerja yang ringan namun kuat, sesuai dengan kebutuhan penelitian ini yang mengutamakan fleksibilitas dan efisiensi dalam pengelolaan data dan informasi.

Tabel 3. 6 Perbandingan *database system*

Indikator	SQLite [71]	MySQL [56]
Kelebihan	<ol style="list-style-type: none"> 1. Ringan dan mudah untuk di-deploy 2. Performa cepat untuk pembacaan data (Low overhead) 3. <i>Open source</i> 4. Ideal untuk proyek kecil hingga menengah 	<ol style="list-style-type: none"> 1. <i>Open source</i> 2. Dirancang untuk proyek berskala besar 3. Memiliki performa baik untuk <i>query</i> kompleks dan dataset besar
Kekurangan	<ol style="list-style-type: none"> 1. Tidak ideal untuk proyek berskala besar 2. Lebih lambat untuk <i>query</i> kompleks 3. Fitur keamanan minim 	<ol style="list-style-type: none"> 1. Membutuhkan web-server terpisah 2. Lebih kompleks untuk di-deploy 3. Membutuhkan biaya lebih besar untuk hardware dan infrastruktur

Berdasarkan tabel 3.6, MySQL dan SQLite memiliki keunggulan yang berbeda sesuai dengan kebutuhan proyek dan skala aplikasi. Penelitian ini menggunakan MySQL karena memiliki keunggulan dalam mengelola data dalam skala besar, sesuai dengan kebutuhan sistem yang akan dibangun. MySQL menawarkan performa tinggi untuk menangani *query* yang kompleks dan dapat diandalkan dalam menjaga keamanan data, yang sangat penting untuk melindungi informasi sensitif. Selain itu, MySQL memiliki komunitas pengguna dan dokumentasi yang luas, memudahkan dalam pemecahan masalah serta mendapatkan dukungan teknis selama proses pengembangan. Dengan kemampuannya dalam menangani akses multi-*user* secara efisien, MySQL menjadi pilihan ideal untuk memenuhi kebutuhan sistem yang akan melayani banyak pengguna secara bersamaan.