PENENTUAN FUNGSI AGREGASI QUERY BERBASIS RULE BASED APPROACHES PADA MASALAH TEXT-TO-SQL

PROPOSAL SKRIPSI

Oleh :

Rizal Efendi

NPM. 21161562014

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS BHINNEKA PGRI

OKTOBER 2024

PENENTUAN FUNGSI AGREGASI QUERY BERBASIS RULE BASED APPROACHES PADA MASALAH TEXT-TO-SQL

PROPOSAL SKRIPSI

Diajukan untuk Memenuhi Salah Satu Persyaratan dalam Menyelesaikan Program Sarjana

Oleh :
Rizal Efendi
NPM. 21161562014

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS BHINNEKA PGRI
OKTOBER 2024

LEMBAR PERSETUJUAN PEMBIMBING

Proposal Skripsi karya **Rizal Efendi** ini telah diperiksa dan disetujui untuk diuji oleh pembimbing pada tanggal, 11 Desember 2024

Tulungagung, 11 Desember 2024
Pembimbing

Agung Prasetya, M.Kom
NIDN. 0725078503

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga proposal penelitian ini dapat terselesaikan dengan baik. Proposal penelitian ini berjudul "**Penentuan Fungsi Agregasi Query Berbasis Rule based approaches Pada Masalah Text-To-Sql**". Proposal ini disusun sebagai salah satu persyaratan untuk melaksanakan Seminar Proposal pada Program Studi Informatika, Universitas Bhinneka PGRI. Dalam penyusunan proposal ini, penulis banyak menerima bantuan dan dukungan dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Dr. Imam Sujono, S.Pd., M.M., selaku Rektor Universitas Bhinneka PGRI.
2. Ibu Vertika Panggayuh, S.Pd., M.Pd , selaku Dekan Fakultas Sains dan Teknologi Universitas Bhinneka PGRI.
3. Bapak Agung Prasetya, M.Kom selaku Dosen Pembimbing dan sekaligus penguji I yang telah memberikan arahan, bimbingan, serta saran berharga dalam penulisan proposal ini.
4. Keluarga dan teman-teman yang selalu memberikan dukungan moral dan motivasi selama proses penulisan ini.

Penulis menyadari bahwa proposal ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi penyempurnaan penelitian ini.

Tulungagung, 11 Desember 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

A. Latar Belakang

Dalam era digital saat ini, data menjadi salah satu aset paling berharga di berbagai sektor, baik pemerintahan, bisnis, pendidikan, maupun kesehatan. Kemampuan untuk mengelola, menganalisis, dan mengambil informasi dari data yang tersedia menjadi kunci utama dalam pengambilan keputusan yang tepat. Menurut McKinsey Global Institute (2016), *"Data has become the lifeblood of economic growth and decision-making, driving innovation and improving operational efficiency across industries."* Namun, seiring dengan pertumbuhan data yang begitu cepat, muncul tantangan bagaimana mengakses dan memanfaatkan data secara efisien dan tepat guna.

Salah satu solusi dalam pengelolaan data adalah dengan **database** atau basis data. Database menjadi sarana penyimpanan dan pengelolaan data secara terstruktur, yang memungkinkan data dapat disimpan, diakses, dan dimanipulasi dengan lebih mudah. Menurut **Connolly dan Begg (2015)**, *"A database provides a structured way to store, retrieve, and manipulate data, ensuring data integrity and accessibility in a controlled environment."* Untuk mengambil atau memanipulasi data di dalam database, dibutuhkan **query**, yaitu perintah dalam sintaks tertentu seperti **Structured Query Language (SQL)**. SQL menjadi bahasa standar dalam pengelolaan data di database relasional, dan memiliki kemampuan ekstensif, seperti pemfilteran, penggabungan data, hingga penerapan fungsi agregasi (**SUM, AVG, COUNT, MAX, dan MIN**) yang berguna untuk meringkas atau menghitung data secara spesifik.

Namun, penggunaan SQL memerlukan **pemahaman teknis** yang tidak semua orang miliki. Pengguna non-teknis sering kali mengalami kesulitan dalam menulis query yang tepat untuk mengekstrak data yang mereka butuhkan. Akibatnya, aksesibilitas terhadap informasi yang terkandung dalam database menjadi terbatas, dan ketergantungan terhadap tim teknis pun meningkat. Hal ini senada dengan pernyataan Iyer dan Subramanian (2014) yang menyatakan bahwa,

"Technical barriers in querying databases hinder data-driven decision-making for non-technical users."

Untuk menjawab tantangan ini, teknologi **Text-to-SQL** muncul sebagai solusi inovatif. Teknologi ini memanfaatkan pemrosesan bahasa alami (NLP) untuk menerjemahkan perintah dalam **bahasa manusia** menjadi query SQL. Dengan Text-to-SQL, pengguna hanya perlu memberikan perintah sederhana seperti "Tampilkan total penjualan bulan ini" atau "Hitung rata-rata pendapatan tahunan". Sistem kemudian akan secara otomatis menerjemahkan perintah tersebut ke dalam query SQL yang sesuai. Menurut Zhong dkk. (2017), *"Text-to-SQL bridges the gap between natural language and database querying, enabling non-experts to interact with data seamlessly."*

Metode Text-to-SQL berbasis rule based approaches merupakan pendekatan yang memanfaatkan aturan dan pola tertentu untuk menerjemahkan bahasa alami ke dalam kueri SQL. Salah satu teknik yang telah diusulkan adalah metode berbasis sketch, di mana pola-pola bahasa spesifik dalam teks, seperti "berapa banyak" atau "jumlah", dikenali dan diterjemahkan ke dalam fungsi SQL, seperti COUNT atau SUM. Metode ini sederhana namun efektif untuk kueri dengan pola serupa, sehingga sangat cepat dan efisien tanpa memerlukan pelatihan yang mendalam.

Pendekatan lainnya adalah *Syntax Parsing*, yang menggunakan parser untuk menguraikan kalimat dan mengidentifikasi elemen-elemen penting seperti subjek, objek, dan kata kerja. Elemen-elemen ini kemudian dihubungkan dengan komponen SQL yang relevan, seperti tabel, kolom, atau fungsi. *Syntax Parsing* mampu menangani kalimat yang lebih kompleks karena mengenali struktur kalimat secara lebih mendalam, memungkinkan konversi yang lebih akurat untuk kueri dengan susunan yang variatif.

Selanjutnya, ada *Keyword Mapping*, yang memetakan kata-kata kunci tertentu dari bahasa alami ke fungsi SQL, misalnya, "rata-rata" ke AVG dan "total" ke SUM. Dengan pemetaan kata kunci ini, metode ini mampu menerjemahkan istilah-istilah umum langsung ke dalam SQL dengan cepat dan cocok untuk sistem sederhana yang tidak membutuhkan kompleksitas tinggi.

Terakhir, metode *Rule based approaches-Based Conditional Transformation* menggunakan aturan-aturan khusus berdasarkan kondisi tertentu untuk menangani variasi kalimat yang lebih rumit, seperti memilih antara *WHERE* atau *HAVING* tergantung konteks. Dengan demikian, metode ini dapat menghasilkan kueri dengan filter atau kondisi yang lebih detail, sangat cocok untuk konversi dengan struktur kalimat yang bervariasi.

Meskipun metode Text-to-SQL berbasis *rule based approaches* memiliki keunggulan dalam hal efisiensi dan kesederhanaan, metode ini juga memiliki beberapa keterbatasan yang signifikan. Salah satu kelemahan utamanya adalah ketergantungan pada aturan yang eksplisit dan terbatas, sehingga metode ini sulit beradaptasi dengan variasi bahasa yang lebih kompleks atau pertanyaan yang tidak terstruktur dengan pola yang telah ditentukan. Akibatnya, ketika pengguna mengajukan pertanyaan dengan struktur yang berbeda atau menggunakan sinonim yang tidak dikenali, metode ini sering kali gagal menerjemahkan teks secara akurat ke dalam kueri SQL.

Selain itu, metode berbasis *rule based approaches* ini juga kurang fleksibel dan memerlukan pemeliharaan intensif karena setiap perubahan dalam pola bahasa alami atau kebutuhan kueri baru mengharuskan pembaruan aturan yang ada. Metode-metode ini juga belum mempertimbangkan kalimat berbahasa Indonesia, yang strukturnya berbeda dengan bahasa Inggris, misalnya pada susunan subjek, predikat, dan objek yang dapat memengaruhi hasil terjemahan kueri. Dampak dari kelemahan ini adalah sistem sering kali gagal memahami maksud pertanyaan pengguna dalam bahasa Indonesia dan menghasilkan kueri yang tidak akurat. Hal ini menyebabkan keterbatasan dalam penerapan metode berbasis *rule based approaches* untuk pengguna non-Inggris, terutama di lingkungan yang membutuhkan keakuratan tinggi dalam beragam bahasa.

Penelitian ini mengusulkan solusi kombinasi metode berbasis *rule based approaches* dengan model pembelajaran mesin untuk meningkatkan ketepatan Text-to-SQL. Solusi ini memanfaatkan metode berbasis *rule based approaches* untuk pola-pola sederhana, sementara model pembelajaran mesin menangani variasi bahasa yang lebih kompleks dan sinonim dalam pertanyaan pengguna.

Metode berbasis *rule based approaches* unggul dalam efisiensi dan penerjemahan cepat untuk pola yang konsisten. Dampak positif dari solusi ini adalah peningkatan akurasi dan fleksibilitas sistem, yang memungkinkan Text-to-SQL menangani pertanyaan pengguna dengan lebih baik tanpa kehilangan presisi dalam menghasilkan kueri SQL.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan dapat diketahui rumusan masalah penelitian ini sebagai berikut :

1. Bagaimana pendekatan berbasis *rule based approaches* dapat secara akurat mengidentifikasi fungsi agregasi seperti SUM, AVG, dan COUNT dari pertanyaan yang diajukan dalam bahasa alami?
2. Seberapa efektif metode ini dalam mengkonversi pertanyaan berbahasa Indonesia menjadi *query* SQL yang tepat, terutama dalam hal fungsi agregasi?

C. Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah dijelaskan dapat diketahui tujuan dari penelitian ini adalah sebagai berikut :

1. Mengembangkan pendekatan berbasis *rule-based approaches* yang mampu secara akurat mengidentifikasi fungsi agregasi seperti **SUM, AVG, dan COUNT** dari perintah atau pertanyaan dalam bahasa alami.
2. Peningkatan Akurasi: Mengidentifikasi dan mengembangkan metode untuk meningkatkan akurasi konversi kalimat bahasa Indonesia menjadi query SQL yang tepat, terutama untuk fungsi agregasi seperti SUM, AVG, dan COUNT.
3. Pemahaman Struktur Kalimat: Meneliti bagaimana variasi struktur kalimat dalam bahasa Indonesia mempengaruhi kemampuan sistem dalam menginterpretasikan dan menerjemahkan pertanyaan pengguna menjadi query SQL yang sesuai.

D. Manfaat Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah dijelaskan dapat diketahui manfaat dari penelitian ini sebagai berikut :

1. Manfaat Teoritis

    a. Pengembangan Ilmu Pengetahuan, penelitian ini berkontribusi pada pengembangan teori dalam bidang *Natural Language Processing* (NLP) dan sistem Text-to-SQL dengan menyoroti bagaimana pendekatan berbasis *rule based approaches* dapat diadaptasi untuk bahasa Indonesia. Ini membantu memperluas pemahaman tentang tantangan dan solusi yang spesifik untuk bahasa non-Inggris.

2. Manfaat Praktis

    a. Peningkatan *User Experience*, Dengan sistem yang mampu menerjemahkan bahasa alami ke dalam query SQL secara akurat, pengguna yang tidak memiliki pengetahuan teknis dapat lebih mudah mengakses dan menganalisis data. Ini meningkatkan pengalaman pengguna dan memperluas adopsi teknologi di kalangan pengguna non-teknis.

    b. Aksesibilitas Data yang Lebih Baik: Penelitian ini memungkinkan lebih banyak orang untuk berinteraksi dengan sistem *database* menggunakan bahasa sehari-hari mereka, meningkatkan aksesibilitas dan pemanfaatan data di berbagai bidang, seperti pendidikan, bisnis, dan penelitian.

E. Asumsi dan Keterbatasan

1. **Asumsi**

    a. Pemahaman Dasar Pengguna: Penelitian ini mengasumsikan bahwa pengguna memiliki pemahaman dasar tentang konsep pertanyaan yang ingin mereka ajukan dalam bahasa alami, meskipun mereka mungkin tidak memiliki latar belakang teknis dalam SQL atau pemrograman.

    b. Struktur Bahasa Indonesia: Penelitian ini mengasumsikan bahwa meskipun terdapat variasi dalam penggunaan bahasa Indonesia, pola umum dalam struktur kalimat dapat diidentifikasi dan diadaptasi oleh sistem berbasis *rule based approaches* yang dikembangkan.

c. Ketersediaan Data: Penelitian ini juga mengasumsikan bahwa data yang diperlukan untuk pengujian dan validasi sistem dapat diakses dan dikumpulkan dari sumber yang relevan, seperti survei atau database publik

2. Keterbatasan

a. Keterbatasan Bahasa: Metode yang diusulkan mungkin tidak sepenuhnya mampu menangani semua variasi dan kompleksitas dalam bahasa Indonesia, terutama dalam konteks penggunaan kata-kata yang tidak baku atau dialek lokal yang berbeda.

b. Ketergantungan pada Data yang Tersedia: Validitas hasil penelitian sangat bergantung pada kualitas dan kuantitas data yang digunakan. Jika data tidak representatif, hasil konversi dan evaluasi akurasi mungkin tidak mencerminkan kinerja sistem di dunia nyata.

c. Keterbatasan Fungsi Agregasi: Penelitian ini difokuskan pada fungsi agregasi tertentu (seperti SUM, AVG, dan COUNT), sehingga mungkin tidak mencakup seluruh jenis *query* atau pertanyaan yang lebih kompleks yang melibatkan beberapa fungsi atau kondisi.

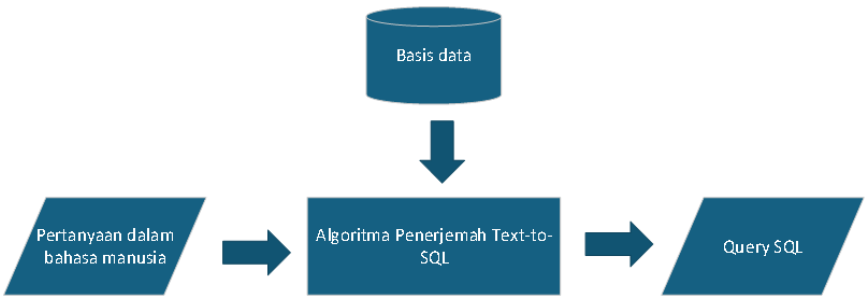
BAB II
KAJIAN PUSTAKA

A. Landasan Teori

1. Masalah Text-To-SQL

Basis data relasional memiliki keunggulan dalam manajemen data, yaitu konsistensi dan integritas data. Konsistensi merujuk pada apa yang berubah di sebuah record dalam satu tabel akan berubah di tabel-tabel lain yang terkait. Sedangkan integritas merujuk pada adanya kepastian bahwa data yang tersimpan dapat ditemukan dan dihubungkan dengan data lainnya (*Silberschatz et al., 2011*). Walaupun demikian, melakukan query data dari sebuah basis data relasional memerlukan pengetahuan tentang sintaks *Structured Query Language* (SQL) dan struktur tabel beserta relasinya. Pengguna awam seringkali mengalami kesulitan dalam menerjemahkan pertanyaan dalam bahasa manusia ke query SQL, karena mereka harus memahami sintaks SQL serta struktur tabel yang kompleks (*Gupta et al., 2020*).

Teknologi Text-to-SQL membantu menjawab tantangan ini dengan menerjemahkan pertanyaan dalam bahasa manusia menjadi query SQL secara otomatis. Text-to-SQL menerima dua *input*, yaitu pertanyaan dalam bahasa alami dan struktur basis data relasional, lalu menghasilkan output berupa query SQL yang valid dan sesuai untuk mengeksekusi data (*Yu et al., 2018*). Jika query ini dieksekusi pada basis data relasional, data yang dihasilkan harus menjawab apa yang dimaksudkan dalam pertanyaan pengguna. Hal ini memungkinkan pengguna non-teknis untuk berinteraksi dengan basis data secara lebih mudah tanpa perlu memahami sintaks SQL (*Zhong et al., 2017*). Sebagai ilustrasi, Text-to-SQL diilustrasikan pada diagram di bawah ini:



Gambar 1 : Diagram Ilustrasi Text-To-SQL

Sebuah basis data relasional yang menjadi masukan Text-To-SQL terdiri dari dua hal, yaitu: struktur tabel dan relasi antar tabel. Beberapa metode Text-To-SQL mensyaratkan adanya *record-record* dalam tabel. Namun penelitian ini berasumsi bahwa *record* tidak dapat diakses secara sembarangan karena adanya data *privacy*. Berikut adalah contoh sebuah basis data "Sistem Manajemen Perpustakaan" yang terdiri dari struktur tabel dan relasi antar tabel

Id_pelanggan	Nama_pelanggan	Alamat_pelanggan	Telepon
101	Rizal Efendi	Kel. Karangwaru, Kec. Tulungagung	082331723701
102	Bramasta Albatio Haryono	Ds. Kamulan, Kab. Trenggalek	0895347042844
103	Khabib Khoirul Yazid	Kel. Bago, Kec. Tulungagung	087840581879

Table 1 : Tabel Pelanggan

Id_buku	judul_buku	penulis	Tahun_terbit	kategori
201	Dasar Pemrograman PHP	Budi Santoso	2021	Pendidikan
202	Belajar Database MySQL	Siti Aisyah	2022	Teknologi
203	Kumpulan Cerita Rakyat	Ahmad Fauzi	2022	Sastra

Table 2 : Tabel Buku

Id_peminjaman	Id_pelanggan	Id_buku	Tanggal_pinjam	Tanggal_kembali
301	101	201	2024-10-01	2024-10-15
302	103	202	2024-10-02	2024-10-16
303	110	203	2024-10-03	2024-10-17

Table 3 : Tabel Peminjaman

Id_detail_peminjaman	Id_peminjaman	Id_buku	jumlah	status
401	301	201	1	Sudah Dikembalikan
402	302	202	2	Belum Dikembalikan
403	303	203	1	Sudah Dikembalikan

Table 4 : Tabel Detail Peminjaman

Gambar 2 : Relasi Antar Tabel

Berikut ini adalah beberapa contoh pertanyaan dalam bahasa manusia :

1. Pertanyaan tentang Peminjaman:
   a. "Buku apa saja yang dipinjam oleh Iwan Setiawan pada bulan Oktober 2024?"
   b. "Berapa banyak buku yang belum dikembalikan oleh pelanggan dengan ID 101?"

c. "Daftar semua peminjaman yang belum dikembalikan sampai hari ini."

2. Pertanyaan tentang Buku:
   a. "Buku apa saja yang tersedia di kategori Teknologi?"
   b. "Siapa penulis dari buku yang berjudul 'Dasar Pemrograman PHP'?"
   c. "Berapa jumlah total buku yang ada di perpustakaan ini?"
3. Pertanyaan tentang Pelanggan:
   a. "Berapa kali Rudi Harahap meminjam buku di perpustakaan ini?"
   b. "Siapa saja pelanggan yang meminjam buku pada bulan Oktober 2024?"
   c. "Alamat lengkap dari pelanggan yang bernama Soni Ananta?"
4. Pertanyaan Gabungan (Peminjaman dan Buku):
   a. "Buku apa saja yang dipinjam pada transaksi peminjaman ID 301?"
   b. "Berapa jumlah total buku yang dipinjam dalam setiap transaksi?"
   c. "Tampilkan semua detail peminjaman untuk buku berjudul 'Belajar Database MySQL'."

Berikut ini adalah beberapa contoh hasil penerjemah pertanyaan ke query SQL

1. SELECT judul_buku FROM buku WHERE kategori = 'Teknologi';
2. SELECT buku.judul_buku
   FROM peminjaman
   JOIN detail_peminjaman ON peminjaman.id_peminjaman = detail_peminjaman.id_peminjaman JOIN buku ON detail_peminjaman.id_buku = buku.id_buku JOIN pelanggan ON peminjaman.id_pelanggan = pelanggan.id_pelanggan WHERE pelanggan.nama_pelanggan = 'Iwan Setiawan';
3. SELECT DISTINCT pelanggan.nama_pelanggan
   FROM peminjaman

JOIN pelanggan ON peminjaman.id_pelanggan pelanggan.id_pelanggan WHERE MONTH(peminjaman.tanggal_pinjam) = 10 AND YEAR(peminjaman.tanggal_pinjam) = 2024;

2. Fungsi Agregasi pada SQL

Fungsi agregasi SQL digunakan untuk melakukan perhitungan pada data dalam tabel dan mengembalikan hasil tunggal. Berikut adalah lima fungsi agregasi dasar:

1. SUM: Menjumlahkan total denda dari peminjaman buku.
   SELECT SUM(denda) AS total_denda FROM peminjaman;
2. AVG: Menghitung rata-rata jumlah buku yang dipinjam per transaksi.
   SELECT AVG(jumlah_buku) AS rata_rata_buku_dipinjam FROM detail_peminjaman;
3. COUNT: Menghitung jumlah total pelanggan.
   SELECT COUNT(id_pelanggan) AS jumlah_pelanggan FROM pelanggan;
4. MAX: Menemukan jumlah denda tertinggi pada suatu peminjaman.
   SELECT MAX(denda) AS denda_tertinggi FROM peminjaman;
5. MIN: Menemukan jumlah denda terendah pada suatu peminjaman.
   SELECT MIN(denda) AS denda_terendah FROM peminjaman;

3. Penentuan Fungsi Agregasi Pada Kalimat Bahasa Alami

Berikut adalah beberapa indikator dalam kalimat bahasa alami yang dapat menunjukkan penggunaan fungsi agregasi tertentu dalam SQL:

1. SUM: Kata seperti "total," "jumlah keseluruhan," atau "akumulasi" menunjukkan adanya agregasi SUM.
   Contoh: "Berapa total denda dari semua peminjaman?"
2. AVG: Kata seperti "rata-rata" atau "rerata" menunjukkan penggunaan fungsi AVG.

Contoh: "Berapa rata-rata jumlah buku yang dipinjam per transaksi?"

3. COUNT: Istilah seperti "jumlah," "banyaknya," atau "berapa kali" mengindikasikan penggunaan fungsi COUNT.
   Contoh: "Berapa banyak pelanggan yang meminjam buku pada bulan ini?"
4. MAX: Kata "tertinggi," "maksimal," atau "nilai paling tinggi" menunjukkan fungsi MAX.
   Contoh: "Apa denda tertinggi yang dikenakan dalam peminjaman?"
5. MIN: Kata "terendah," "terkecil," atau "nilai paling rendah" menunjukkan penggunaan MIN.
   Contoh: "Berapa denda terendah yang pernah dikenakan?"

4. Pendekatan Berbasis Rule Based Approaches

a. Pengenalan

Rule Based Approaches (juga dikenal sebagai sistem produksi atau sistem pakar) adalah bentuk paling sederhana dari kecerdasan buatan. *Rule based approaches* menggunakan aturan sebagai representasi pengetahuan untuk pengetahuan yang dikodekan ke dalam sistem. Definisi dari *rule based approaches* hampir sepenuhnya bergantung pada sepenuhnya bergantung pada sistem pakar, yang merupakan sistem yang meniru penalaran manusia ahli dalam memecahkan masalah yang bersifat pengetahuan (*Giarratano & Riley, 2005*).

Alih-alih merepresentasikan pengetahuan secara deklaratif dan statis sebagai sekumpulan hal yang benar, sistem berbasis aturan merepresentasikan pengetahuan dalam bentuk sekumpulan *rule* yang memberitahukan apa yang harus dilakukan atau apa yang harus disimpulkan dalam situasi yang berbeda. menyimpulkan dalam situasi yang berbeda. Aturan ini biasanya dinyatakan dalam bentuk **IF-THEN** atau aturan produksi (*Jackson, 1998*).

Sebuah sistem berbasis aturan dapat secara sederhana dibuat dengan menggunakan sekumpulan pernyataan dan sekumpulan aturan yang

menentukan bagaimana bertindak pada set pernyataan. Hal ini memungkinkan sistem untuk bekerja secara sistematis dalam memecahkan masalah berdasarkan aturan yang telah ditetapkan sebelumnya (*Durkin, 1994*). Aturan dinyatakan sebagai seperangkat pernyataan jika-maka (disebut IF-THEN disebut aturan IF-THEN atau aturan produksi):

JIKA P MAKA Q

yang juga setara dengan:

P⇒Q.

b. Element dari Rule based approaches-Based System

Setiap *rule based approaches* terdiri dari beberapa elemen dasar dan sederhana sebagai berikut:

1. Satu set fakta. Fakta-fakta ini sebenarnya adalah pernyataan dan harus berupa sesuatu yang relevan dengan kondisi awal sistem.
2. Seperangkat aturan. Ini berisi semua tindakan yang harus diambil dalam ruang lingkup masalah yang menentukan bagaimana cara bertindak pada kumpulan pernyataan. Sebuah aturan berhubungan dengan fakta-fakta di bagian IF ke beberapa tindakan di bagian THEN. Sistem harus berisi hanya aturan-aturan yang relevan dan menghindari aturan-aturan yang tidak relevan karena jumlah aturan dalam sistem akan mempengaruhi kinerjanya.
3. Kriteria penghentian. Ini adalah kondisi yang menentukan bahwa sebuah solusi telah ditemukan atau tidak ada solusi. Hal ini diperlukan untuk menghentikan beberapa sistem berbasis aturan yang berada dalam perulangan tak terbatas.

c. Rule based approachess

Sebuah aturan terdiri dari dua bagian: bagian IF dan bagian THEN. Bagian IF disebut anteseden atau premis (atau kondisi) dan bagian MAKA disebut konsekuen atau kesimpulan (atau tindakan).

Dengan demikian, aturan sederhana dapat dinyatakan sebagai:

IF = antecedent

THEN = consequent.

Contoh:

No.	Antecedent IF	Consequent THEN
1.	The season is winter	It is cold

Aturan ini menguji ekspresi logika dalam premis, dan, jika ekspresi tersebut bernilai benar, maka aturan ini menyatakan bahwa sebuah fakta mengenai suatu hal atau kelas dari suatu hal adalah benar.

Sebuah aturan umum dapat memiliki beberapa anteseden yang digabungkan dengan salah satu operator logika AND, OR (atau dengan campuran keduanya).

No.	Operator Logika	Antecedent IF	Consequent THEN
1.	AND	The season is winter	The weather is cold
		The temperatures is <0 degrees	
		It is windy	
2.	OR	The season is winter	It is cold
		The temperature is <0 degrees	
		It is windy	
3.	AND dan OR	The season is winter	It is cold
		The temperature is <0 degrees	
		The weather is windy (digabungkan dengan **OR**)	

5. Struktur Sistem Pakar Berbasis Rule based approaches

Sebuah sistem pakar berbasis aturan memiliki struktur yang diberikan dalam diagram pada Gambar 2.3 dan terdiri dari elemen-elemen utama berikut

ini (lima elemen yang paling penting ditandai yang paling penting ditandai dengan huruf tebal pada gambar):

a. Basis pengetahuan

Berisi pengetahuan domain yang direpresentasikan sebagai aturan (aturan IF-THEN) tentang subjek yang sedang dibahas [5][19].

b. Basis Data

Terdiri dari fakta kalkulus predikat yang cocok dengan bagian IF dari aturan dalam basis pengetahuan.

c. Mesin Inferensi

Terdiri dari semua proses yang memanipulasi basis pengetahuan untuk menyimpulkan informasi yang diminta oleh pengguna dan membawa penalaran yang dibutuhkan oleh sistem pakar untuk mencapai solusi.

d. Penjelasan Subsistem

Menganalisis struktur penalaran yang dilakukan oleh sistem dan menjelaskannya kepada pengguna, memberikan kemungkinan kepada pengguna untuk bertanya kepada sistem tentang cara bagaimana suatu kesimpulan dicapai atau tentang fakta yang digunakan.

e. Antarmuka Pengguna

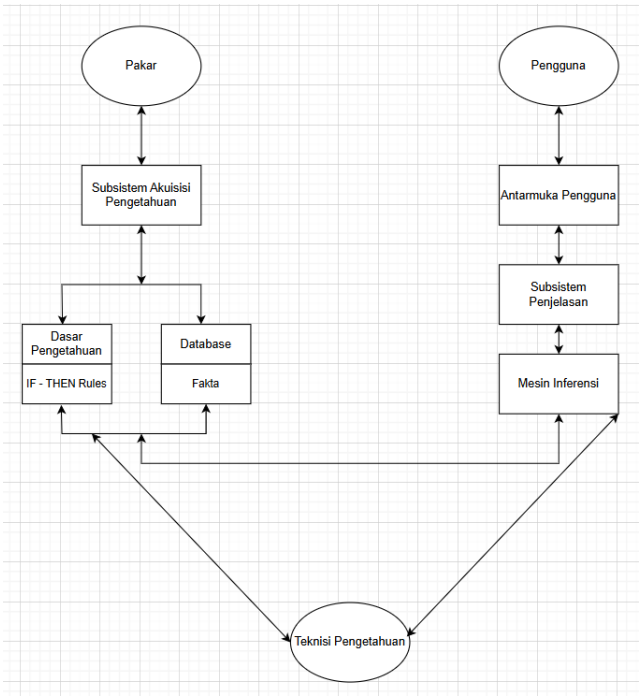
Mengacu pada komunikasi antara pengguna yang mencari solusi dan sistem pakar dan terdiri dari beberapa jenis sistem pemrosesan bahasa alami atau antarmuka pengguna grafis dengan menu.

f. Insinyur Pengetahuan

Biasanya seorang ilmuwan komputer dengan pelatihan AI yang bekerja dengan seorang ahli di bidang aplikasi untuk merepresentasikan pengetahuan yang relevan dari *expert* dalam bentuk yang dapat dimasukkan ke dalam basis pengetahuan.

g. Subsistem akuisisi pengetahuan

Memeriksa dan memperbarui basis pengetahuan yang terus berkembang untuk mengetahui kemungkinan adanya ketidakkonsistenan dan informasi yang tidak lengkap.



Gambar 3 Struktur sistem pakar berbasis aturan

Sistem berbasis aturan bekerja dengan cara yang sangat sederhana: sistem ini dimulai dengan basis aturan, yang berisi semua pengetahuan yang sesuai yang dikodekan ke dalam aturan IF-THEN, dan memori kerja, yang pada awalnya mungkin berisi data, pernyataan, atau informasi yang sudah diketahui. Sistem memeriksa semua kondisi aturan (JIKA) dan menentukan subset, set konflik, dari aturan yang kondisinya terpenuhi berdasarkan memori kerja. Dari kumpulan konflik ini, salah satu aturan dipicu (dipecat). Yang mana yang dipilih didasarkan pada strategi penyelesaian konflik. Ketika aturan dijalankan, tindakan apa pun yang ditentukan dalam klausa THEN akan dijalankan. Tindakan ini dapat memodifikasi memori kerja, basis aturan itu sendiri, atau melakukan apa saja yang diputuskan oleh pemrogram sistem untuk disertakan. Perulangan penembakan aturan dan pelaksanaan tindakan ini terus berlanjut sampai kriteria penghentian terpenuhi. Kriteria penghentian ini dapat diberikan oleh fakta bahwa tidak ada lagi aturan yang kondisinya terpenuhi atau sebuah aturan ditembakkan yang tindakannya menentukan bahwa program harus dihentikan. Penalaran

adalah cara di mana aturan-aturan digabungkan untuk mendapatkan pengetahuan baru. Penalaran adalah cara manusia bekerja dengan pengetahuan, fakta, dan strategi pemecahan masalah untuk menarik kesimpulan.

h. Jenis - jenis sistem pakar berbasis rule based approaches

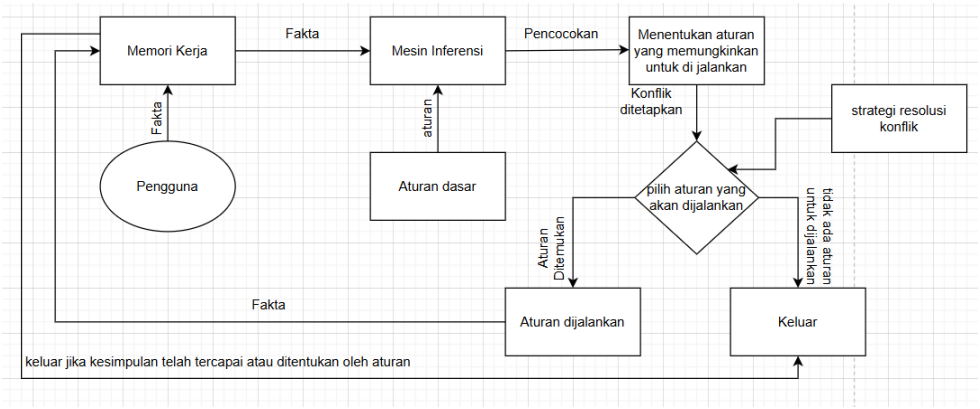
Sistem pakar berbasis aturan bekerja sebagai berikut: mesin inferensi membandingkan setiap aturan dalam basis pengetahuan dengan fakta dalam basis data. Jika bagian IF dari sebuah aturan cocok dengan fakta maka bagian THEN dijalankan dan aturan tersebut dijalankan. Dengan menjalankan sebuah aturan, sebuah hasil baru (fakta baru) dapat diperoleh dan ini akan ditambahkan ke dalam basis data. Dengan menembakkan aturan, rantai inferensi diperoleh. Rantai inferensi menunjukkan bagaimana sistem pakar menerapkan aturan-aturan untuk mencapai kesimpulan atau tujuan. Ada dua cara utama di mana aturan dijalankan dan hal ini menyebabkan adanya dua sistem aturan utama: - sistem rantai maju *(forward chaining)*. Sistem *forward chaining* dimulai dengan fakta-fakta awal dan terus menggunakan aturan-aturan untuk menarik kesimpulan baru (atau mengambil tindakan tertentu) berdasarkan fakta-fakta tersebut. - sistem perunutan ke belakang *(backward chaining)*. Sistem *backward chaining* dimulai dengan beberapa hipotesis (atau tujuan) untuk dibuktikan, dan terus mencari aturan yang memungkinkan untuk menyimpulkan hipotesis tersebut, dengan menetapkan sub-tujuan baru untuk dibuktikan seiring dengan berjalannya proses. Sistem *forward chaining* pada dasarnya digerakkan oleh data, sedangkan sistem *backward chaining* digerakkan oleh tujuan.

i. *Forward Chaining Systems*

Cara kerja *forward chaining* adalah sebagai berikut: diberikan sekumpulan fakta tertentu dalam memori kerja, gunakan aturan untuk menghasilkan fakta-fakta baru sampai tujuan yang diinginkan tercapai.

Langkah-langkah di bawah ini diikuti (lihat Gambar 4 untuk ilustrasi *forward chaining*):

1. Cocokkan bagian IF dari setiap aturan dengan fakta-fakta yang ada di memori kerja.
2. Jika ada lebih dari satu aturan yang dapat digunakan (lebih dari satu aturan yang dijalankan), pilih aturan mana yang akan digunakan dengan menggunakan resolusi konflik (dijelaskan pada bagian berikut).
3. Terapkan aturan tersebut. Jika fakta baru diperoleh, tambahkan fakta tersebut ke memori kerja.
4. Berhenti (atau keluar) ketika kesimpulan ditambahkan ke memori kerja atau jika ada aturan yang menentukan untuk mengakhiri proses.



Gambar 4 Forward chaining diagram.

Contoh sekarang mari kita pertimbangkan sebuah contoh praktis. Diberikan sekumpulan fakta yang berisi berbagai informasi tentang bunga dan diberikan sekumpulan aturan, tugasnya adalah untuk menghasilkan solusi, yang mengindikasikan bunga mana yang merupakan bunga lili putih. *Variabel linguistik* (objek) dan nilai yang mungkin diizinkan oleh sistem pakar dan dimasukkan ke dalam basis data diberikan pada Tabel 1. Basis pengetahuan terdiri dari aturan-aturan berikut

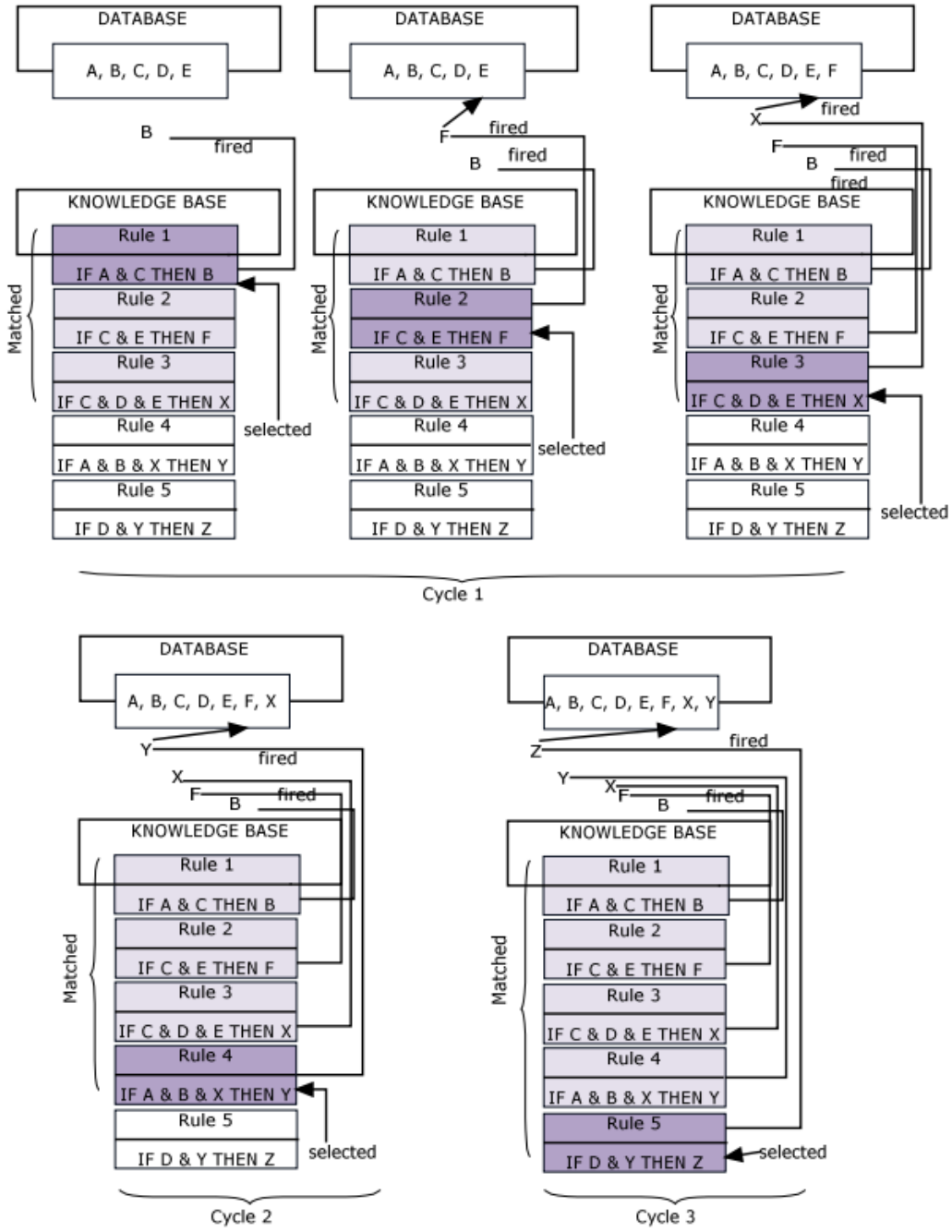
No	Aturan	Kondisi	Hasil
1.	Aturan 1	JIKA ukuran > 10 DAN ukuran < 50 ATAU ukuran > 150	tinggi adalah tinggi
2.	Aturan 2	JIKA ukuran > 50 DAN ukuran < 150	tinggi badan sedang
3.	Aturan 4	JIKA ukuran > 150	tinggi badan tinggi
4.	Aturan 5	JIKA siklus hidup satu tahun	jenis hidup tahunan
5.	Aturan 6	JIKA siklus hidup lebih dari satu tahun	jenis hidup abadi
6.	Aturan 7	JIKA musim adalah musim panas DAN (warna biru ATAU warna ungu ATAU warna kuning ATAU jenis hidup abadi ATAU jenis akar umbi ATAU nama bunga iris)	-
7.	Aturan 8	JIKA musim adalah musim gugur ATAU warna putih ATAU warna merah jambu ATAU warna merah jambu-merah	nama bunga anemone
8.	Aturan 9	JIKA musim adalah musim gugur DAN tinggi adalah sedang DAN (warna kuning ATAU warna sementara ATAU warna ungu ATAU warna merah)	nama bunga Krisan
9.	Aturan 10	JIKA musim adalah musim semi DAN (jenis akar umbi ATAU warna putih ATAU warna kuning ATAU warna jingga ATAU warna ungu ATAU warna merah ATAU warna biru ATAU beraroma benar ATAU wangi benar)	bunga adalah Freesia
10.	Aturan 11	JIKA jenis hidup abadi ATAU tinggi sedang ATAU jenis akar umbi ATAU musim panas ATAU nama bunga Dahlia	-
11.	Aturan 12	JIKA musim adalah musim semi DAN jenis akar umbi DAN (warna kuning ATAU warna putih)	nama bunga Narcissus

12.	Aturan 13	JIKA tanah asam DAN (warna putih ATAU warna pink ATAU warna merah ATAU jenis hidup abadi DAN jenis akar akar)	nama bunga Camelia
13.	Aturan 14	JIKA musim adalah musim semi DAN (jenis akar umbi ATAU wangi benar ATAU tinggi kecil ATAU jenis hidup abadi ATAU nama bunga Lily)	-
14.	Aturan 15	JIKA musim semi ATAU musim panas DAN tinggi badan kecil DAN tipe hidup tahunan DAN (tanah kaya ATAU tanah gembur ATAU tanah subur)	nama bunga Begonia
15.	Aturan 16	JIKA musim adalah musim dingin DAN (warna putih ATAU warna merah muda ATAU warna merah ATAU nama bunga Azalea)	-
16.	Aturan 17	JIKA tipe hidup tahunan DAN tipe akar akar ATAU warna putih ATAU warna merah ATAU warna biru ATAU warna kuning ATAU nama bunga Anemon	-
17.	Aturan 18	JIKA tipe hidup tahunan ATAU tipe akar akar ATAU (warna putih ATAU warna merah muda ATAU warna merah jambu ATAU warna merah ATAU warna kuning ATAU wangi benar ATAU tanah dikeringkan dengan baik	bunga adalah mawar
18.	Aturan 19	JIKA nama bunga Lily ATAU wangi benar ATAU nama bunga Lili putih	-

Object	Value	Object	Value
Nama bunga	Iris Anemone Chrysanthemums Freesia Dahlia Narcissus Camellias Lily Begonia Azaleas Anemone Roses White lily	Warna	biru ungu kuning merah putih merah muda oranye ungu violet merah muda-merah
Musim	Gugur Panas Semi Dingin	Ukuran	10-50 cm 50-150 cm >150 cm
Jenis akar	Akar bohlam	Parfum	Benar Salah
Siklus hidup	Satu tahun Lebih dari satu tahun	Tinggi tumbuhan	Kecil Sedang Tinggi

Misalkan kita memiliki fakta-fakta berikut dalam basis data: musim: musim semi, jenis akar: umbi, wangi: benar, ukuran: 16-18 cm, siklus hidup lebih dari satu tahun, warna: oranye, merah, putih, merah muda. Kami ingin menyimpulkan bunga lily putih. Proses penalaran ke depan dilakukan sebagai berikut:

Siklus	Proses	Aturan yang berlaku	Aturan yang dipilih	Penerapan aturan	Memori kerja
1	Pencocokan	Aturan 1 dan Aturan 6	Aturan 1	Tinggi kecil ditambahkan ke memori kerja	Musim: musim semi, jenis akar: umbi, wangi: benar, ukuran: 16-18 cm, siklus hidup lebih dari satu tahun, warna: oranye, merah, putih, merah muda, tinggi: kecil.
	Pencocokan	Aturan 1 dan Aturan 6	Aturan 6	Jenis hidup abadi ditambahkan ke memori kerja	Musim: musim semi, jenis akar: umbi, wangi: benar, ukuran: 16-18 cm, siklus hidup lebih dari satu tahun, warna: oranye, merah, putih, merah muda, tinggi: kecil, jenis hidup: abadi.
2	**Pencocokan**	Aturan 1, Aturan 6, Aturan 10, Aturan 12, Aturan 14	Aturan 10	Nama bunga Freesia ditambahkan ke memori kerja	Memori kerja sebelumnya + Nama bunga: Freesia.
	Pencocokan	Aturan 1, Aturan 6, Aturan 10, Aturan 12, Aturan 14	Aturan 12	Nama bunga Narcissus ditambahkan ke memori kerja	Memori kerja sebelumnya + Nama bunga: Narcissus.
	Pencocokan	Aturan 1, Aturan 6, Aturan 10, Aturan 12, Aturan 14	Aturan 14	Nama bunga Lily ditambahkan ke memori kerja	Musim: musim semi, jenis akar: umbi, wangi: benar, ukuran: 16-18 cm, siklus hidup lebih dari satu tahun, warna: oranye, merah, putih, merah muda, tinggi: kecil, jenis hidup: abadi, nama bunga: Freesia, Narcissus, Lily.
3	Pencocokan	Aturan 1, Aturan 6, Aturan 10, Aturan 12, Aturan 14, Aturan 19	Aturan 19	Nama bunga Lili Putih ditambahkan ke memori kerja. **Tujuan tercapai**	Musim: musim semi, jenis akar: umbi, wangi: benar, ukuran: 16-18 cm, siklus hidup lebih dari satu tahun, warna: oranye, merah, putih, merah muda, tinggi: kecil, jenis hidup: abadi, nama bunga: Freesia, Narcissus, Lily, Lili Putih.



Gambar 2.5 *Forward chaining* untuk contoh yang dipertimbangkan. Tujuannya adalah untuk mencapai Z.

1. Conflict Resolution

Penting untuk mendefinisikan cara atau urutan penembakan aturan selama proses inferensi. Ada beberapa strategi yang berbeda seperti [2][11]:

a. Penerapan pertama: Jika aturan-aturan berada dalam urutan tertentu, menembakkan aturan yang berlaku pertama adalah cara termudah untuk mengontrol urutan penembakan aturan. Dari perspektif praktis, urutan dapat dibuat dengan mengurutkan aturan-aturan dalam basis pengetahuan dengan menempatkannya dalam urutan yang diinginkan (tetapi ini hanya bekerja untuk sistem kecil hingga 100 aturan).

Ini adalah strategi yang paling sederhana dan memiliki potensi masalah yang besar: yaitu perulangan yang tak terbatas pada aturan yang sama. Jika basis data tetap tidak berubah (tidak ada hal baru yang dimasukkan atau tidak ada yang dihapus), seperti halnya basis aturan, maka 7.4 Jenis-jenis Sistem Pakar Berbasis Aturan 173 kondisi dari aturan pertama tidak berubah dan akan dijalankan berulang kali. Untuk mengatasi hal ini, merupakan praktik umum untuk menangguhkan aturan yang telah ditembakkan dan mencegahnya untuk menembakkan kembali hingga data yang memenuhi kondisi aturan, telah berubah.

6. Metric Precision and Accuracy

Presisi dan akurasi adalah metrik penting dalam mengevaluasi sistem pembelajaran mesin dan klasifikasi, terutama ketika menganalisis efektivitas model prediktif.

a. Presisi: Presisi, yang sering diterapkan dalam klasifikasi biner, mengukur ketepatan prediksi positif. Ini adalah rasio positif yang benar (prediksi positif yang benar) terhadap total prediksi positif (baik positif yang benar maupun positif yang salah). Rumus untuk presisi adalah:

$$\text{Precision} = \frac{\text{True Positive}}{}$$

True Positive + False Positives

Misalnya, jika pengklasifikasi dengan benar mengidentifikasi 8 sampel positif dari 10 total prediksi (termasuk 2 prediksi positif yang salah), maka presisinya adalah 80%. Ketepatan berguna dalam aplikasi di mana biaya positif palsu tinggi, seperti diagnosis medis atau deteksi spam.

b. **Akurasi**: Akurasi adalah rasio prediksi yang benar (baik positif maupun negatif) terhadap total prediksi yang dibuat. Akurasi mencerminkan efektivitas model secara keseluruhan di kedua kelas. Rumus untuk akurasi adalah:

$$\text{Accuracy} = \frac{\text{True Positives + True Negatives}}{\text{Total Predictions}}$$

Misalnya, jika model mengklasifikasikan 90 sampel dengan benar dari 100 sampel, akurasinya adalah 90%. Akurasi bermanfaat dalam set data yang seimbang tetapi dapat menyesatkan dalam set data yang tidak seimbang, karena tidak membedakan antara biaya positif palsu dan negatif palsu.

7. Teknik Voting untuk Pembentukan Basis Pengetahuan

Teknik voting adalah pendekatan kolektif untuk menentukan aturan yang akan dimasukkan ke dalam basis pengetahuan. Pendekatan ini digunakan ketika terdapat beberapa ahli atau sumber yang memberikan masukan, namun memiliki sudut pandang yang berbeda.

Tahapan dalam Teknik Voting

1. **Pengumpulan Aturan**: Semua aturan yang diusulkan dikumpulkan dari berbagai sumber atau ahli.
2. **Evaluasi Individual**: Setiap ahli memberikan penilaian terhadap aturan berdasarkan relevansi, keakuratan, dan manfaatnya.
3. **Proses Voting**: Aturan yang memiliki jumlah suara terbanyak dianggap sebagai aturan yang valid. Dalam beberapa kasus, bobot suara dapat diberikan berdasarkan tingkat keahlian atau pengalaman ahli.

4. **Konsolidasi Aturan**: Aturan yang disetujui dimasukkan ke dalam basis pengetahuan, sementara aturan yang tidak memenuhi kriteria dieliminasi atau direvisi.

Teknik voting memastikan bahwa basis pengetahuan mencerminkan konsensus kolektif, sehingga meningkatkan validitas dan keandalan sistem.

8. Verifikasi Basis Pengetahuan

Verifikasi basis pengetahuan adalah proses untuk memastikan bahwa aturan *(rule based approachess)* yang digunakan dalam sistem berbasis pengetahuan bebas dari kesalahan logika dan dapat mendukung pengambilan keputusan secara efektif. Proses ini bertujuan untuk mengidentifikasi dan mengatasi berbagai masalah yang muncul dalam pengelolaan aturan, seperti aturan redundan, konflik logis, dan kesimpulan yang tidak tercapai.

Masalah dalam Verifikasi Basis Pengetahuan

1. **Redundant Rule based approaches**

   Aturan yang menghasilkan kesimpulan yang sama meskipun tanpa aturan tersebut dianggap redundan. Keberadaan aturan seperti ini dapat meningkatkan kompleksitas sistem dan memperlambat proses inferensi. Redundansi dapat diidentifikasi dengan menggunakan algoritma optimasi berbasis graf atau teknik minimisasi aturan.

2. **Conflicting Rule based approaches**

   Konflik terjadi ketika dua aturan menghasilkan kesimpulan yang bertentangan dari kondisi yang sama. Misalnya, jika satu aturan menyatakan "A adalah benar" dan aturan lain menyatakan "A adalah salah" untuk premis yang sama, konflik ini harus dipecahkan melalui mekanisme resolusi, seperti penentuan prioritas atau penilaian berbobot (*weighted rule based approaches evaluation*).

3. **Subsumed Rule based approaches**

   Aturan yang terlalu umum dapat digantikan oleh aturan spesifik yang mencakup kasus yang sama. Contohnya, aturan "Jika X, maka

Y" dapat digantikan oleh aturan "Jika X dan Z, maka Y" jika Z selalu bernilai benar dalam konteks X.

4. **Unnecessary If Condition**

   Kondisi yang tidak relevan dalam aturan hanya meningkatkan kompleksitas tanpa menambah nilai. Identifikasi kondisi ini dapat dilakukan melalui analisis statistik terhadap frekuensi aktivasi kondisi tersebut.

5. **Circular Rule based approaches**

   Siklus logika muncul ketika aturan merujuk pada kesimpulan dirinya sendiri. Siklus ini dapat menyebabkan sistem tidak mampu menyimpulkan hasil karena terus berputar dalam lingkaran yang tidak berujung. Teknik seperti analisis graf digunakan untuk memutus siklus ini.

6. **Dead-End If Condition**

   Aturan yang memiliki kondisi awal yang tidak pernah terpenuhi menjadi tidak berguna. Misalnya, aturan "Jika X > 1000, maka Y" dalam konteks data di mana nilai X tidak pernah lebih dari 1000.

7. **Dead-End Goal**

   Tujuan yang tidak dapat dicapai karena tidak ada aturan pendukung dalam sistem disebut tujuan buntu. Ini menunjukkan bahwa sistem membutuhkan aturan tambahan untuk mendukung pencapaian tujuan.

8. **Unreachable Conclusion**

   Kesimpulan yang tidak dapat dicapai meskipun teorinya diperlukan menunjukkan adanya celah dalam aturan pendukung. Hal ini biasanya terjadi akibat kurangnya jalur logis yang menghubungkan premis dengan kesimpulan.

Metode Penanganan Masalah

Masalah-masalah di atas dapat diatasi melalui berbagai teknik, seperti:

1. **Analisis Dependensi Aturan**: Mengidentifikasi relasi antara aturan untuk menemukan redundansi atau konflik.
2. **Uji Cakupan dan Validasi Logis**: Memastikan semua aturan diuji terhadap skenario data nyata.
3. **Optimasi Basis Pengetahuan**: Menggunakan algoritma pengurangan aturan untuk menghilangkan aturan tidak perlu.
4. **Visualisasi Graf**: Membantu mendeteksi siklus dan kesimpulan yang tidak tercapai.

Relevansi Kajian

Masalah dalam verifikasi basis pengetahuan memiliki implikasi signifikan, terutama dalam sistem berbasis aturan yang digunakan untuk pengambilan keputusan kritis, seperti diagnosis medis, sistem rekomendasi, atau inferensi logika dalam kecerdasan buatan.

9. First Order Logic

First Order Logic (FOL), atau logika predikat tingkat pertama, adalah kerangka formal dalam logika matematis yang digunakan untuk merepresentasikan pengetahuan tentang dunia nyata secara sistematis. FOL menggunakan istilah seperti predikat, fungsi, variabel, kuantor eksistensial (∃), dan kuantor universal (∀). Predikat merepresentasikan hubungan antar entitas atau atribut dari suatu entitas, sementara fungsi mengembalikan hasil tertentu berdasarkan masukan tertentu.

Komponen Utama FOL

1. **Konstanta**: Representasi dari entitas spesifik, seperti angka, objek, atau individu tertentu. Misalnya, `John`, `5`, atau `Library`.
2. **Predikat**: Fungsi yang menguji sifat atau hubungan antar entitas. Contohnya, `Peminjam(John, BukuX)`, yang berarti

First Order Logic (FOL), atau logika predikat tingkat pertama, adalah kerangka kerja formal dalam logika matematis yang digunakan untuk menyatakan fakta-fakta tentang dunia nyata dengan cara yang terstruktur dan logis. FOL terdiri dari komponen utama seperti konstanta, predikat,

variabel, fungsi, serta operator logika seperti kuantor universal (∀) dan eksistensial (∃). Kerangka ini sangat berguna untuk memodelkan relasi dan atribut entitas dalam suatu domain.

Komponen Utama dalam FOL

1. **Konstanta**: Simbol yang merepresentasikan entitas spesifik, seperti nama atau objek unik. Misalnya, `John`, `BukuA`.
2. **Predikat**: Menyatakan properti atau relasi antar entitas, misalnya `Peminjam(John, BukuA)` berarti John meminjam BukuA.
3. **Variabel**: Elemen yang dapat menggantikan entitas dalam suatu domain, misalnya `x` atau `y`.
4. **Fungsi**: Merepresentasikan hubungan antar elemen yang menghasilkan suatu nilai, seperti `Judul(BukuA)` mengembalikan judul dari BukuA.
5. **Operator Logika**:
   - **Kuantor Universal (∀)**: Menyatakan bahwa suatu fakta berlaku untuk semua elemen, misalnya, `∀x Peminjam(x, BukuA)` berarti semua orang meminjam BukuA.
   - **Kuantor Eksistensial (∃)**: Menyatakan bahwa ada setidaknya satu elemen yang memenuhi fakta tertentu, misalnya, `∃x Peminjam(x, BukuA)` berarti ada seseorang yang meminjam BukuA.

Keunggulan FOL

FOL menawarkan fleksibilitas dan kekuatan representasi yang luas karena mampu menangkap hubungan kompleks antar entitas. FOL sangat cocok untuk domain yang membutuhkan pemodelan relasi yang rumit, seperti sistem manajemen perpustakaan.

B. Literatur Review

No	Judul	Masalah Yang Diselesaikan	Penelitian Terdahulu
1,	TypeSQL: Knowledge-based Type-Aware Neural Text-to-SQL Generation	Mengusulkan pendekatan baru yang disebut TYPESQ: yang memandang masalah Text-To-SQL sebagai pengisian slot. Selain itu, metode usulan menggunakan informasi type untuk bisa lebih memahami entitas dan angka yang langka dalam pertanyaan bahasa alami.	Penelitian yang akan dilakukan berfokus pada optimalisasi pemrosesan data dengan model yang lebih sederhana namun akurat, tanpa terlalu bergantung pada pengisian slot berbasis tipe.
2.	A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions	Menyajikan tinjauan komprehensif tentang pendekatan pembelajaran mendalam untuk penguraian Text-To-SQL. Pertama memperkenalkan korpus penguraian Text-To-SQL yang dapat dikategorikan sebagai satu putaran dan banyak putaran. Kedua, memberikan gambaran sistematis mengenai model bahasa yang telah dilatih sebelumnya dan metode-metode yang ada untuk penguraian Text-To-SQL. Ketiga menyajikan kepada pembaca tantangan-tantangan yang dihadapi oleh penguraian Text-To-SQL dan mengeksplorasi beberapa potensi arah masa depan dalam bidang ini.	Penelitian ini akan mempersempit cakupan pada pengembangan metode baru yang lebih relevan untuk skenario praktis di dunia nyata, dibandingkan hanya melakukan tinjauan literatur.
3.	Improving Text-to-SQL Evaluation Methodology	Mengidentifikasi keterbatasan dan mengusulkan perbaikan sistem Text-To-SQL. Dengan cara membandingkan pertanyaan yang dibuat oleh manusia dan pertanyaan yang dibuat secara otomatis, mengkarakterisasi sifat-sifat pertanyaan yang diperlukan untuk aplikasi dunia nyata. Selain itu menunjukkan bahwa pembagian data saat ini ke dalam set pelatihan dan pengujian mengukur ketahanan terhadap variasi dalam cara pertanyaan diajukan, tetapi hanya sebagian menguji seberapa baik sistem menggeneralisasi ke kueri baru.	Penelitian mendatang akan fokus pada peningkatan akurasi dalam implementasi Text-To-SQL menggunakan pendekatan hibrida tanpa hanya mengevaluasi generalisasi.
4.	RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers	Menyajikan kerangka kerja terpadu, berdasarkan mekanisme perhatian-diri yang sadar akan relasi, untuk menangani pengkodean skema, penautan skema, dan representasi fitur dalam penyandi Text-To-SQL. Selain	Akan dikembangkan pendekatan yang mengurangi kompleksitas implementasi sambil tetap mempertahankan kinerja tinggi dalam pengkodean skema.

		itu, mengamati peningkatan kualitatif dalam pemahaman model mengenai skema penautan dan penyelarasan.	
5.	Structure-Grounded Pretraining for Text-to-SQL	Menyajikan kerangka kerja prapelatihan Structure-Grounded (STRUG) baru yang diawasi secara lemah untuk Text-To-SQL yang dapat secara efektif belajar untuk menangkap perataan teks-tabel berdasarkan korpus teks-tabel paralel. Serta mengidentifikasi serangkaian tugas prapelatihan baru: landasan kolom, landasan nilai, dan pemetaan nilai-kolom, dan memanfaatkannya untuk melatih penyandian tabel-teks.	Fokus penelitian adalah menciptakan model yang dapat langsung digunakan tanpa memerlukan banyak prapelatihan dengan efisiensi waktu dan sumber daya.
6.	Recent Advances in Text-to-SQL: A Survey of What We Have and What We Expect	Meninjau kemajuan terbaru pada text-to-SQL untuk dataset, metode, dan evaluasi serta menyediakan survei sistematis ini, yang membahas tantangan-tantangan dan mendiskusikan arah potensial di masa depan.	Penelitian ini akan memperkenalkan metode implementasi langsung tanpa hanya membahas tantangan dan arah masa depan.
7.	DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction	Menunjukkan bahwa memecah masalah pembangkitan menjadi sub-masalah dan memasukkan solusi dari sub-masalah tersebut ke dalam Large Language Models (LLM) dapat menjadi pendekatan yang efektif untuk meningkatkan kinerjanya secara signifikan.	Penelitian mendatang akan memanfaatkan decomposed learning tetapi dengan metode yang lebih efisien dan terintegrasi untuk aplikasi yang lebih luas.
8.	Using Database Rule based approaches for Weak Supervised Text-to-SQL Generation	Menyajikan sebuah cara sederhana untuk melakukan tugas dari masalah Text-To-SQL dengan pengawasan yang lemah dengan menyebutnya Rule based approaches-SQL. Dengan merancang beberapa aturan untuk mengurangi ruang pencarian eksplorasi.	Penelitian akan mengkombinasikan pendekatan berbasis aturan dengan pembelajaran mesin untuk memberikan hasil yang lebih adaptif pada dataset baru.
9.	DuoRAT: Towards Simpler Text-to-SQL Models	Membangun DuoRAT, sebuah implementasi ulang dari model RAT-SQL yang canggih yang tidak seperti RAT-SQL yang hanya menggunakan relasi-sadar atau vanilla transformer sebagai blok pembangunnya. Serta melakukan beberapa percobaan ablasi menggunakan DuoRAT sebagai model dasar. Eksperimen ini mengkonfirmasi kegunaan beberapa teknik dan menunjukkan redundansi teknik lainnya, termasuk fitur SQL struktural dan fitur yang menghubungkan pertanyaan dengan skema1 .	Penelitian ini akan fokus pada simplifikasi lebih lanjut model, tetapi dengan memastikan tidak ada penurunan kinerja dalam menangani data dunia nyata.

10.	Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation	Melakukan perbandingan sistematis dan ekstensif terhadap metode-metode rekayasa cepat yang ada, termasuk representasi pertanyaan, pemilihan contoh, dan pengorganisasian contoh, dan dengan hasil eksperimen ini, serta menguraikan kelebihan dan kekurangannya. Berdasarkan temuan ini, dapat mengusulkan solusi terintegrasi baru, bernama DAIL-SQL, yang menyegarkan papan peringkat Spider dengan akurasi eksekusi 86,6% dan menetapkan standar baru.	Penelitian ini akan membangun sistem yang lebih ringan dan fleksibel, mengutamakan integrasi dengan aplikasi praktis tanpa hanya berfokus pada benchmark dataset.

Metode-metode dalam daftar di atas, meskipun memberikan kemajuan signifikan dalam tugas **Text-to-SQL**, memiliki kelemahan utama sebagai berikut:

1. **Belum Menyelesaikan Masalah Identifikasi Fungsi Agregasi**:

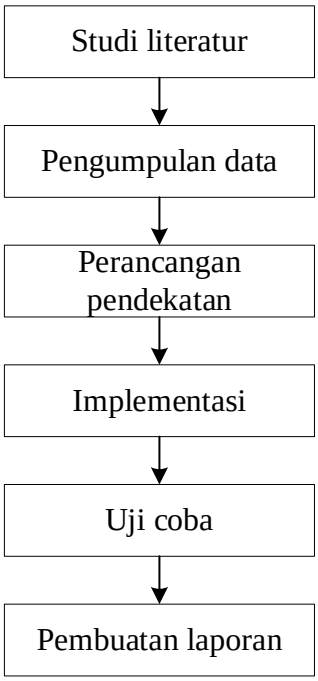
   Semua pendekatan yang dijelaskan dalam metode-metode tersebut masih belum secara eksplisit berfokus pada identifikasi jenis fungsi agregasi seperti SUM, AVG, MAX, dll., terutama dalam kasus penggunaan di mana fungsi agregasi memainkan peran kritis. Hal ini menjadi tantangan dalam domain yang memerlukan kemampuan pengenalan jenis fungsi agregasi dengan tingkat presisi yang tinggi.
2. **Belum Diterapkan untuk Struktur Bahasa Indonesia**:

   Pendekatan yang dipresentasikan sebagian besar berfokus pada bahasa Inggris dan dataset terkait, seperti *Spider*. Tidak ada bukti kuat bahwa metode ini telah diuji atau diadaptasi untuk struktur sintaksis dan semantik Bahasa Indonesia, yang memiliki perbedaan signifikan dalam pola pertanyaan dan hubungan kata dengan skema basis data. Hal ini membuat penerapannya dalam konteks lokal, seperti sistem basis data di Indonesia, masih belum optimal.

BAB III
METODE PENELITIAN

A. Tahapan Penelitian

Penelitian ini terdiri dari 6 tahapan yaitu studi literatur, pengumpulan data, perancangan sistem, implementasi, uji coba dan pembuatan laporan. Gambar 5 menunjukkan alur tahapan penelitian. Penelitian ini memerlukan tiga jenis data. Data yang pertama adalah basis data. Data yang kedua adalah kumpulan pertanyaan. Data yang ketiga adalah query dalam sintaks *Struktur Query Language* (SQL). Ketiga data ini selanjutnya digunakan untuk dua hal yaitu pembentukan dan pengujian sistem berbasis *rule based approaches* untuk Penentuan Fungsi Agregasi Query. Berikut ini akan diuraikan bagaimana mengumpulkan kedua jenis data tersebut.



Gambar 5 Alur Tahapan Penelitian

B. Pengumpulan Data

Fase kedua dari penelitian ini adalah pengumpulan data. Fase ini sebenarnya berupaya mengumpulkan data awal yang berikutnya akan dibentuk ulang ketika akuisi pengetahuan. Fase ini menggunakan teknik pengumpulan data yaitu teknik analisis dokumen. Alasan dipilihnya teknik analisis dokumen pada penelitian ini adalah karena data berupa kalimat dan basis data tidak selalu tersedia ketika digali secara langsung melalui observasi dan juga basis data tidak dapat dipublikasikan terkait privacy.

Dokumen-dokumen yang menjadi sumber data pada penelitian ini adalah berupa jurnal ilmiah, artikel konferensi, laporan tugas akhir dan laporan tesis. Jenis dokumen ini dipertimbangkan sebagai sumber data karena mengandung dokumentasi suatu masalah riil, dokumen tersebut sudah melalui pengujian maupun review oleh akademisi, dan menggunakan kosakata dan struktur bahasa baku.

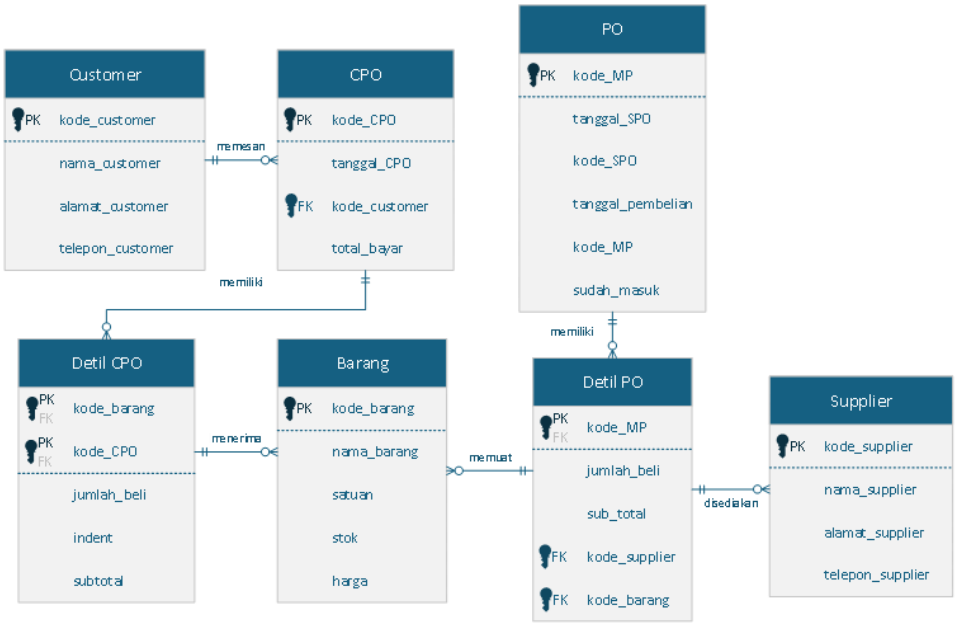
Namun demikian, tidak semua dokumen yang disebutkan diatas dipertimbangkan pada peneletian ini. Beberapa kriteria yang menentukan apakah suatu dokumen menjadi sumber data adalah:

1. Dokumen tergolong salah satu dari jurnal ilmiah, artikel konferensi, laporan tugas akhir/tesis/disertasi
2. Dokumen tertulis dalam kosakata dan struktur bahasa Indonesia
3. Dokumen dapat diakses secara publik
4. Dokumen memuat *Entity Relationship Diagram* dalam notasi tertentu
5. Dokumen memuat deskripsi sistem, hasil analisis kebutuhan sistem, hasil pendefinisian fitur atau diagram proses bisnis
6. Dokumen memuat hasil pengembangan sistem atau hasil penyelesaian masalah riil
7. Dokumen menyelesaikan masalah di salah satu domain. Penelitian ini mempertimbangkan sejumlah domain masalah yang jumlahnya terbatas meliputi: hotel, inventori, akademik, kepegawaian, akuntansi, pemesanan dan penjualan. Alasan dipilihnya domain ini adalah ketersediaan pakar atau

ahli pada domain tersebut sehingga hasil pengumpulan basis data dapat terverifikasi oleh pakar domain

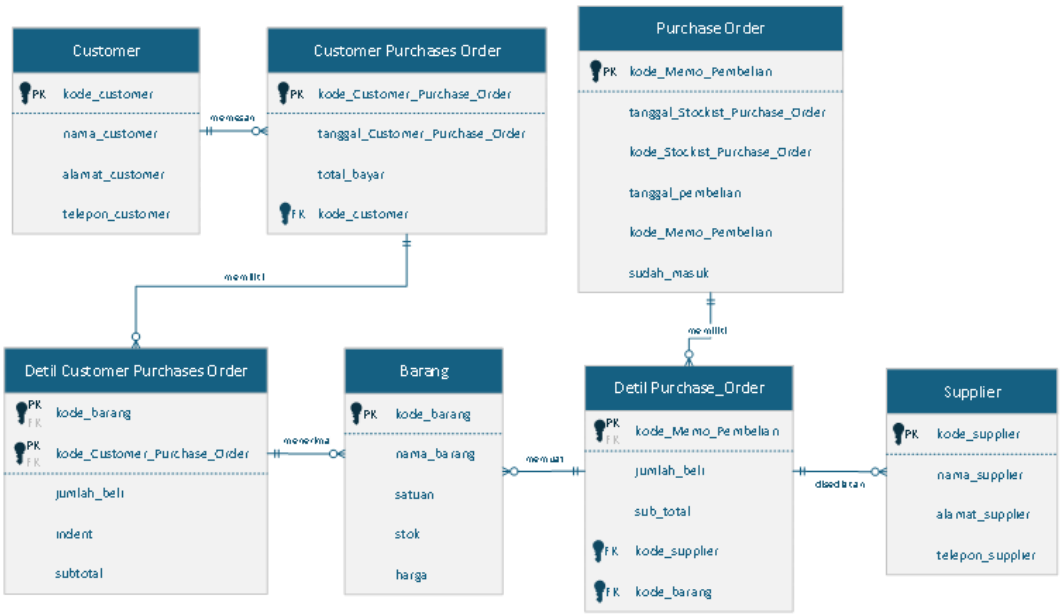
Pada penelitian ini, dokumen-dokumen yang memenuhi keseluruhan kriteria diatas digali untuk dua jenis data. Data pertama yang dikumpulkan adalah *Entity Relationship Diagram*. Setiap ERD didalam dokumen digambar ulang menggunakan notasi *Crow's Foot*. Ada beberapa pemrosesan dalam menggambar ulang ERD, yaitu:

1. Mengubah setiap singkatan yang tidak umum yang muncul di nama kolom, tabel dan relasi menjadi bentuk lengkap. Untuk menentukan bentuk lengkap dari suatu singkatan, penelitian ini menggunakan konteks yang tertuang didalam dokumen itu sendiri, tanpa menggunakan pengetahuan eksternal di luar dokumen. Berikut ini adalah ERD sebelum perubahan:



Gambar 6 Tabel ERD

Pada ERD diatas, sejumlah singkatan seperti CPO, PO dan MP diubah menjadi bentuk lengkap. Singkatan ini diubah ke bentuk lengkap mengikuti informasi yang ada di dalam dokumen. Hasil pengubahan singkatan pada sampel ERD diatas adalah:



Ketika sebuah singkatan tidak dapat diketahui bentuk lengkapnya dari dokumen, penelitian ini tidak mengubah singkatan tersebut dan tetap membiarkan dalam bentuk singkatan. Penelitian selanjutnya dapat mempertimbangkan sebuah mekanisme atau metode untuk menentukan bentuk lengkap singkatan.

2. Mengurangi kata-kata yang tidak relevan yang muncul di nama kolom, nama tabel dan nama relasi. Beberapa ERD memiliki kata-kata yang tidak terlalu berpengaruh terhadap pendekatan usulan. Kata-kata yang tidak relevan tersebut adalah kata “tabel”, ”tbl”, “table”, “column”, “kolom”, ”fk”, dan mengubah kata jamak menjadi bentuk tunggal. Berikut ini adalah ERD sebelum perubahan:

   Gambar ERD sebelum penghapusan kata-kata tidak relevan

   Berikut ini adalah hasil penghapusan kata-kata tidak relevan dari ERD diatas:

Selain ERD, data yang ingin diperoleh pada penelitian ini adalah kalimat-kalimat yang akan ditentukan jenis fungsi aggregasi-nya. Kalimat-kalimat ini diperoleh dengan cara menurunkan dari hasil pendefinisian fitur atau hasil pengumpulan kebutuhan atau hasil pemodelan proses bisnis. Selain itu,

pembentukan kalimat didasarkan pada ERD. Alasan kalimat-kalimat diturunkan dari fitur atau kebutuhan atau proses bisnis adalah sinkronisasi antara kalimat dengan ERD. Pada konteks masalah text-to-SQL, kalimat haruslah dapat diterjemahkan ke query yang selaras dengan ERD.

Penelitian ini tidak menggunakan template dalam pembentukan kalimat. Pembentukan kalimat didasarkan pada kemampuan manusia tanpa mengikuti pola tertentu sehingga lebih natural. Adapun sampel hasil pembentukan kalimat ditunjukkan pada tabel berikut ini:

No	Kalimat	Fitur/Kebutuhan/Proses Bisnis
1	Berapa jumlah total pesanan sampai pada tanggal 11 Desember 2023?	Administrasi melakukan input sekaligus mengecek daftar pesanan barang dari Database. Bila produk atau nutrisi ada, Administrasi akan mencetak Customer Purchase Order 3 rangkap. Rangkap 1 diarsipkan, rangkap 2 diserahkan pada Customer sambil melakukan pembayaran dan pengambilan produk atau nutrisi, rangkap 3 diserahkan ke Logistik untuk diarsipkan.
2	Tampilkan daftar pesanan barang !	
3	Berapa rata-rata barang terjual untuk semua produk hingga 20 Januari 2024?	
4	Tampilkan pelanggan yang jumlah pesanan di bulan Februari 2024 paling banyak	
5	Tampilkan pesanan barang yang statusnya "belum selesai"!	Administrasi memeriksa status pesanan dan memastikan barang yang belum selesai diproses. Jika status berubah, admin memperbarui sistem untuk menghindari keterlambatan pengiriman.
6	Tambahkan pesanan baru untuk pelanggan dengan nama "Ahmad Fadillah".	Administrasi menginput data pesanan baru dari pelanggan, mencatat detail barang, kuantitas, dan tanggal pesanan ke dalam sistem.
7	Hapus data pelanggan bernama "Fitri Handayani".	Administrasi menghapus informasi pelanggan tertentu dari basis data setelah konfirmasi terkait ketidakaktifan atau permintaan dari pelanggan.
8	Berapa jumlah barang yang telah habis terjual hingga saat ini?	Logistik memeriksa stok barang yang terjual dari laporan penjualan harian untuk analisis permintaan produk dan restocking.
9	Tampilkan barang yang	Sistem membantu logistik memprioritaskan pengisian ulang barang

	stoknya kurang dari 10 unit! yang hampir habis dengan memberi laporan secara otomatis.	
10	Berapa total pesanan pelanggan pada bulan Januari 2024?	Sistem menghitung jumlah total pesanan berdasarkan periode waktu tertentu yang diminta oleh bagian administrasi atau keuangan untuk laporan bulanan.

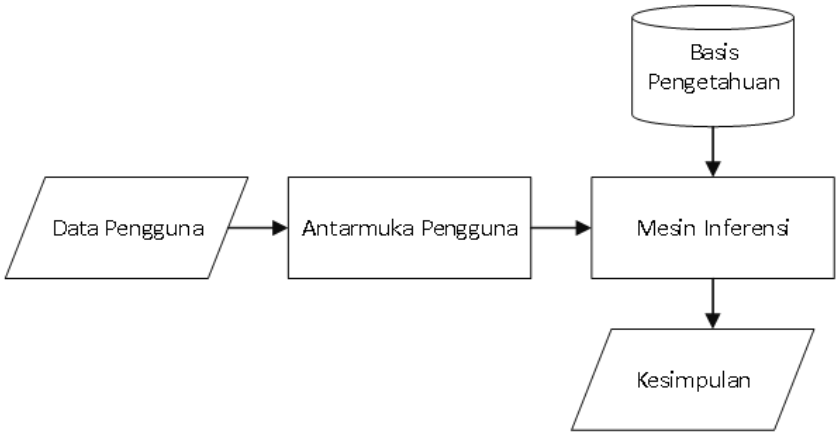
Table 5 Sample Hasil Pembentukan Kalimat

Pada sampel diatas, beberapa kalimat diturunkan dari satu prosedur dalam proses bisnis. Seperti yang sudah dijelaskan, selain proses bisnis, penelitian ini juga mempertimbangkan fitur dan deskripsi kebutuhan dalam pembentukan kalimat.

Hasil pembentukan kalimat akan diverifikasi dan diproses ulang untuk membentuk pengetahuan oleh pakar domain. Proses verifikasi dan pembentukan pengetahuan diuraikan pada subbab berikutnya.

C. Perancangan pendekatan berbasis rule based approaches

Pendekatan berbasis *rule based approaches* pada penelitian ini tersusun atas sejumlah komponen. Rancangan pendekatan pada penelitian ini ditunjukkan pada gambar 7 berikut ini:



Gambar 7 Rancangan pendekatan berbasis *rule based approaches*

Komponen-komponen tersebut meliputi Antarmuka pengguna, Mesin inferensi, dan Basis Pengetahuan. Selain itu, pendekatan memiliki masukan berupa data pengguna dan luaran berupa kesimpulan. Secara singkat, alur proses mulai dari masukan hingga luaran diuraikan sebagai berikut.

Berdasarkan rancangan diatas, masukan data pengguna berupa kalimat yang akan diidentifikasi jenis fungsi aggregasi-nya memasuki komponen antarmuka

pengguna. Komponen ini hanya mengerjakan satu fungsi utama yaitu pemrosesan bahasa ilmiah terhadap kalimat. Setelah pemrosesan diharapkan kalimat dapat dipahami oleh mesin inferensi. Selanjutnya, mesin inferensi memanipulasi pengetahuan yang ada di basis pengetahuan dan kalimat, dan pada akhirnya melakukan reasoning untuk mendapatkan kesimpulan. Pendekatan usulan tidak memiliki subsistem penjelasan karena pendekatan ini merupakan bagian dari metode text-to-sql berbasis template sehingga tidak memerlukan penjelasan pada pengguna. Detil mengenai komponen-komponen pendekatan diuraikan pada sub bab berikut ini.

1. Data Pengguna

Masukan utama pendekatan berbasis rule based approaches adalah kalimat. Lebih spesifiknya, kalimat yang menjadi masukan adalah kalimat dalam bahasa Indonesia. Kalimat ini dapat berupa pertanyaan, perintah maupun kalimat deklaratif. Berikut ini adalah beberapa sampel kalimat yang ada pada penelitian ini:

No	Kalimat	Jenis
1	Berapa jumlah total pesanan pada bulan Desember 2023?	Pertanyaan
2	Apa rata-rata barang yang terjual untuk produk merk "ABC" bulan ini?	Pertanyaan
3	Tampilkan semua data pelanggan yang melakukan transaksi pada Januari 2024.	Pertanyaan
4	Tambahkan pesanan baru untuk pelanggan bernama "Siti Aminah" dengan jumlah barang 3 unit!	Perintah
5	Hapus data pesanan dengan ID "P12345"!	Perintah
6	Update status pesanan ID "P12346" menjadi "Selesai"!	Perintah
7	Semua pesanan pelanggan pada bulan ini telah tercatat di sistem.	Deklaratif/pernyataan
8	Basis data kami mencatat bahwa stok barang "XYZ" telah habis.	Deklaratif/pernyataan
9	Transaksi yang dilakukan pada tanggal 1 Februari 2024 mencapai rekor tertinggi.	Deklaratif/pernyataan

Table 6 Sample Data Pengguna

Pendekatan pada penelitian ini mempertimbangkan masukan berupa kalimat lengkap tunggal. Asumsi yang ada pada penelitian ini adalah proses identifikasi jenis fungsi agregasi hanya dapat dikerjakan jika teks yang dimasukkan adalah kalimat utuh. Selain itu penelitian ini berasumsi bahwa

kalimat yang menjadi masukan adalah tertulis dengan kosakata dan struktur bahasa Indonesia.

Pendekatan pada penelitian ini tidak mempertimbangkan masukan berupa unit teks yang lebih luas daripada kalimat seperti paragraf. Selain itu, pendekatan juga tidak mempertimbangkan pemrosesan unit teks yang lebih pendek daripada kalimat, misalnya frasa. Keterbatasan ini dapat dipertimbangkan untuk penelitian selanjutnya. Termasuk pula bagaimana mendeteksi kalimat atau unit teks yang tidak lengkap.

2. Komponen antarmuka pengguna

Kalimat dari pengguna akan masuk pertama kali ke komponen antarmuka pengguna. Walaupun komponen ini bernama antarmuka pengguna, komponen ini tidak memiliki tampilan grafis. Komponen ini hanya melakukan satu fungsi yaitu melakukan pemrosesan bahasa alami pada kalimat. Tujuan pemrosesan ini adalah agar kalimat dapat diolah oleh komponen selanjutnya yaitu mesin inferensi.

Pemrosesan kalimat pada komponen antarmuka pengguna terdiri dari beberapa sub proses, yaitu:

a. Filtering

Sub proses filterring memeriksa apakah data pengguna berupa kalimat atau tidak. Pemeriksaaanya menggunakan mekanisme sederhana yaitu dengan mempertimbangkan jumlah kata dan menghitung jumlah huruf untuk setiap kata. Aturan filtering adalah:

- Untuk data pengguna berupa kalimat deklaratif, jika jumlah kata dari data pengguna adalah kurang dari 3 kata, maka data pengguna dianggap bukan kalimat karena sebuah kalimat deklaratif lengkap minimal terdiri dari dari subyek, predikat dan obyek. Untuk mengidentifikasi apakah data pengguna adalah kalimat deklaratif atau tidak, penelitian ini mengidentifikasi tanda titik di akhir kalimat.
- Untuk data pengguna berupa kalimat tanya, jika jumlah kata dari data pengguna adalah kurang dari 3 kata, maka data pengguna

dianggap bukan kalimat karena sebuah kalimat tanya lengkap minimal terdiri dari kata tanya, subyek dan predikat. Untuk mengidentifikasi apakah data pengguna adalah kalimat tanya atau tidak, penelitian ini mengidentifikasi tanda tanya di akhir kalimat.

- Untuk data pengguna berupa kalimat perintah, jika jumlah kata dari data pengguna adalah kurang dari 2 kata, maka data pengguna dianggap bukan kalimat karena sebuah kalimat perintah lengkap minimal terdiri dari predikat dan subyek dan predikat. Untuk mengidentifikasi apakah data pengguna adalah kalimat tanya atau tidak, penelitian ini mengidentifikasi tanda seru di akhir kalimat.

b. Text cleaning

Subproses ini melakukan proses pengubahan teks kalimat menjadi huruf kecil semua. Pada penelitian ini, kalimat perlu distandarisasi kedalam huruf kecil karena identifikasi jenis fungsi agregasi didasarkan secara leksikal, memperhatikan bentuk huruf, bukan semantik. Selain itu, text cleaning melakukan penghapusan sejumlah tanda baca yang meliputi:

‘!”#$%&'()*+,-./:;?@[\]^_`{|}~'

dan juga URL.

Tanda baca-tanda baca ini dapat dihapus karena pendekatan pada penelitian ini hanya mengidentifikasi kata. Proses penghapusan dilakukan menggunakan regular expression.

c. Tokenisasi

Sub proses ini mengubah kalimat yang telah melalui text cleaning menjadi daftar kata. Pemisahan kalimat menjadi daftar kata dilakukan berdasarkan adanya spasi atau karakter sejenis spasi meliputi tab, spasi ganda dan baris baru.

d. Stemming

Sub proses stemming mengubah kata berimbuhan menjadi kata dasar. Namun kata jamak pada penelitian ini tidak diubah ke kata dasar. Setiap kata dari hasil tokenisasi diubah menjadi kata dasar karena

pendekatan pada penelitian ini tidak membedakan antara satu bentuk kata dengan bentuk yang lain dan tidak memperhitungkan kelas kata. Sebagai ilustrasi, kata 'menambahkan' dan 'tambah' dianggap sama.

3. Komponen basis pengetahuan

Komponen basis pengetahuan berfungsi untuk menampung pengetahuan atau knowledge. Komponen ini bersama dengan komponen basis data berperan ketika proses reasoning yang dikerjakan oleh mesin inferensi. Pendekatan usulan pada penelitian ini merepresentasikan pengetahuan sebagai kumpulan production *rule based approaches* atau IF-THEN *rule based approaches*. Setiap *rule based approaches* terdiri atas antecedent dan conclusion. Sebuah *rule based approaches* bisa juga memiliki lebih dari satu antecedent. Antar satu antecedent dengan antecedent lain dalam rule based approaches dihubungkan dengan penghubung logis yaitu OR, AND, NOT, dan XOR. Struktur sebuah rule based approaches ditunjukkan pada gambar 3.2 berikut ini:

IF [NOT] antecedent ([AND|OR|XOR] [NOT] antecedent)*

THEN consequent

Pada struktur rule based approaches diatas, notasi [] bermakna penghubung logis didalamnya bersifat opsional. Notasi ()* berarti antecedent didalamnya bisa lebih dari satu. Sedangkan notasi […|…|…] berarti bersifat opsional dan memilih salah satu dari penghubung logis.

Pembentukan basis pengetahuan dilakukan melalui proses knowledge acquisition atau mengakuisisi pengetahuan. Metodologi akuisisi pengetahuan yang digunakan pada penelitian ini terdiri dari beberapa fase, yaitu:

a. Perencanaan akuisisi pengetahuan

Pada fase perencanaan, ada dua langkah yang diterapkan di penelitian ini yaitu: penentuan pakar domain dan penentuan teknik akuisisi. Untuk langkah pertama, ada sejumlah atribut yang dipertimbangkan dari seseorang agar bisa sebagai pakar domain, yaitu:

1) Seorang profesional di industri yang terkait dengan membuat atau menganalisis query SQL minimal 2 tahun, atau seorang akademisi yang mengajar topik query SQL pada tingkat pendidikan tinggi minimal 2 tahun secara berturut-turut
2) Sesorang memiliki ketersedian waktu, kemampuan kerjasama dan komitmen dalam proses akuisisi pengetahuan
3) Seseorang mampu memahami teks tertulis dalam bahasa Indonesia
4) Seseorang mampu berkomunikasi untuk menyampaikan hasil reasoning atau penalaran.

Alasan adanya atribut tersebut untuk memastikan bahwa pengetahuan yang diakuisisi adalah pengetahuan yang kredibel. Penelitian ini menggunakan multi pakar domain dengan atribut yang disebutkan diatas. Jumlah total pakar domain pada penelitian ini adalah sebanyak 5 akademisi dan 10 profesional. Alasan penggunaan multi pakar dalam akuisisi pengetahuan adalah untuk mengurangi ketidakcukupan waktu akuisisi, mengurangi adanya bias, mengurangi ketidaklengkapan (incmpleteness) kepakaran dan mengurangi penalaran yang sempit

Langkah kedua adalah penentuan teknik akuisisi. Karena penelitian ini melibatkan lebih dari satu pakar domain, teknik akuisisi pengetahuan yang digunakan adalah Consensus Decision-Making (Voting). Alasan digunakanya teknik ini adalah memiliki efisiensi ketika terjadi adanya ketidaksepakatan terhadap solusi suatu masalah.

Lebih konkret, teknik Voting yang digunakan dalam penelitian ini dimulai dari pembentukan sekumpulan pertanyaan. Pembentukan pertanyaan ini dilakukan oleh knowledge engineer dengan cara mengolah kalimat-kalimat pada fase pengumpulan data. Setiap kalimat akan ditentukan oleh pakar domain apakah kalimat

ini termasuk didalam fungsi agregasi: SUM, AVG, MIN, MAX, COUNT atau NO_AGG.

Pendekatan usulan hanya mendeteksi fungsi agregasi: SUM, AVG, MIN, MAX, COUNT. Sedangkan kalimat-kalimat yang tidak termasuk salah satu kelimanya dimasukkan sebagai NO_AGG. Penelitian ini berasumsi pula bahwa setiap kalimat tidak terjadi ambiguitas, artinya tidak ada satu kalimat yang termasuk 2 atau lebih fungsi agregasi. Penelitian selanjutnya dapat mempertimbangkan hal ini.

Selanjutnya kalimat akan dipisah menjadi beberapa kata. Setiap kata akan diberi tanda oleh pakar domain jika kata tersebut memiliki kaitan atau menjadi karakteristik kemunculan suatu fungsi agregasi. Penelitian ini hanya mempertimbangkan karakteristik fungsi agregasi berdasarkan kata, bukan frase. Penelitian selanjutnya dapat mempertimbangkan frase.

Berdasarkan penjelasan diatas, struktur pertanyaan yang harus dijawab oleh pakar ditunjukkan pada tabel berikut ini:

ID	Domain DB	DB ID	Kalimat	Jenis DML
1	pemesanan	5	Berapa jumlah total pesanan sampai hari ini	SUM
2	pemesanan	5	Tampilkan daftar pesanan barang	NO_AGG
3	inventori	2	Berapa rata-rata penjualan per produk hingga sekarang	AVG
4	pemesanan	5	Tampilkan pelanggan Yang jumlah pesananya Terbanyak 2024	MAX
5	akademik	3	Berapa jumlah Total pesanan Sampai hari ini	NO_AGG
6	inventori	2	Berapa jumlah stok barang minimum pada semua kategori?	MIN
7	akademik	3	Tampilkan daftar mahasiswa yang memperoleh nilai maksimum	MAX
8	pemesanan	5	Berapa rata-rata jumlah pesanan harian bulan ini?	AVG
9	inventori	2	Tampilkan produk dengan stok paling sedikit	MIN

Table 7 Tabel Kalimat

Seorang pakar domain akan mengisi jenis fungsi aggregasi dari setiap kalimat dan memberi tanda pada kata jika kata tersebut berkaitan dengan jenis fungsi aggregasi. Pada sampel diatas, kata

"jumlah" dan "total" berkaitan dengan SUM sehingga diberi tanda bahwa kata tersebut berkaitan dengan jenis fungsi aggregasi SUM untuk domain *database* pemesanan. Namun, kata "jumlah" dan "total" untuk domain akademik tidak termasuk SUM meskipun kata "jumlah" dan "total" terkait SUM pada domain pemesanan. Hal ini karena kalimat ber-ID 5 tidak relevan dengan domain database akademik.

Selanjutnya teknik Voting yang diterapkan pada penelitian ini akan melakukan pengambilan keputusan berdasarkan jumlah terbanyak jika terjadi ketidakseragaman antar pakar dalam menilai sesuatu.

b. Ekstraksi pengetahuan pakar

Ekstraksi pengetahuan pakar merupakan proses mengisi pertanyaan-pertanyaan yang telah didefinisikan diatas. Untuk memudahkan proses pengisian, tabel diatas dimodelkan sebagai sebuah form berbasis web. Pada saat mengisi jawaban pertanyaan, pakar domain disertai dengan ERD sesuai DB ID. Hal ini bertujuan untuk memastikan apakah sebuah kalimat adalah relevan dengan domain *database* atau tidak. Kalimat yang tidak relevan termasuk NO_AGG.

c. Analisis pengetahuan

Fase ini terdiri dari satu proses yaitu menerjemahkan hasil ekstraksi pengetahuan ke representasi pengetahuan. Seperti yang telah diuraikan sebelumnya, penelitian ini menggunakan kumpulan IF-THEN *rule based approaches* sebagai representasi pengetahuan. Setiap rule based approaches ditulis dalam *First Order Logic* (FOL). Berikut ini adalah sampel *rule based approaches* yang diperoleh pada saat analisis pengetahuan.

ID RULE BASED APPROACHES	RULE BASED APPROACHES	FOL
R1	IF kalimat mengandung kata "jumlah" dan "total" THEN jenis fungsi aggregasi adalah SUM	$\forall x(kalimat(x) \wedge mengandung(x, jumlah) \wedge mengandung(x, total) \rightarrow SUM(x))$
R2	IF kalimat mengandung kata "rata-rata" THEN fungsi aggregasi adalah AVG	$\forall x(mengandung(x, rata - rata) \rightarrow AVG(x))$
R3	IF kalimat tidak cocok dengan domain basis data THEN kalimat tidak relevan	$\forall x, \exists y(\neg domaincocok(x, y) \rightarrow \neg relevan(x))$
R4	IF kalimat tidak relevan THEN fungsi aggregasi adalah NO_AGG	$\forall x(\neg relevan(x) \rightarrow NO\_AGG(x))$
R5	IF kalimat mengandung kata "terbanyak" THEN fungsi agregasi adalah MAX	∀x (contains(x, "terbanyak") → aggregationFunction(x) = MAX)
R6	IF kalimat mengandung kata "paling sedikit" THEN fungsi agregasi adalah MIN	∀x (contains(x, "paling sedikit") → aggregationFunction(x) = MIN)
R7	IF kalimat mengandung kata "hapus" dan "database" THEN fungsi adalah DROP_DATABASE	∀x (contains(x, "hapus") ∧ contains(x, "database") → databaseAction(x) = DROP_DATABASE)
R8	IF kalimat mengandung kata "tambahkan" THEN fungsi adalah INSERT	∀x (contains(x, "tambahkan") → databaseAction(x) = INSERT)
R9	IF kalimat mengandung kata "tampilkan" THEN fungsi adalah SELECT	∀x (contains(x, "tampilkan") → databaseAction(x) = SELECT)

Table 8 Tabel Sample Rule based approaches

Seperti yang ditunjukkan pada sampel diatas, setiap rule based approaches yang berbentuk FOL terdiri dari *quantifier*, *predicate*, penghubung logis, variabel, konstanta. Pada penelitian ini, jenis fungsi agregasi menjadi predicate. Alasan penggunaan FO pada penelitian adalah memudahkan dalam proses verifikasi basis pengetahuan karena FOL sudah berbentuk baku sehingga mengurangi adanya ambiguitas.

d. Verifikasi basis pengetahuan

Fase berikutnya setelah analisis pengetahuan adalah verifikasi basis pengetahuan. Ada dua aktifitas dalam fase ini yaitu pemeriksaan konsistensi dan pemeriksaan keterlengkapan *(completeness).* Aktifitas pertama memastikan bahwa basis pengetahuan tidak terjadi masalah konsistensi yang meliputi *redundant rule based approachess, conflicting rule based approachess, subsumed rule based approachess, Unnecessary IF Conditions, circular rule based approaches.*

Penelitian ini menggunakan matrix element-*rule based approaches* untuk mengidentifikasi terjadinya ketidakkonsistenan, yaitu redundant rule based approaches, circular *rule based approaches*, *Unnecessary* IF *Conditions* dan *conflicting rule based approachess*. Pada matrix ini, baris matriks merepresentasikan elemen-elemen pada rule based approaches. Sedangkan kolom merepresentasikan *rule based approaches*. Jika sebuah elemen *rule based approaches* bertindak sebagai antecedent pada suatu rule based approaches, maka elemen matriks diisi dengan label A. Jika sebuah elemen *rule based approaches* bertindak sebagai conclusion, maka elemen matriks diisi dengan label C. Setiap elemen matriks memungkinkan berisi lebih dari satu label.

Untuk membentuk matrix element *rule based approaches*, setiap *rule based approaches* yang teridentifikasi pada analisis

pengetahuan dibagi menjadi dua elemen dan diberi label E diikuti dengan nomor urut. Pelabelan elemen dari *rule based approaches* pada tabel 3.5 Tabel penamaan elemen ditunjukkan pada tabel berikut:

Label	Elemen
E1	$kalimat(x)$^$mengandung(x, jumlah)$^$mengandung(x, total)$
E2	$SUM(x)$
E3	$mengandung(x, rata - rata)$
E4	$AVG(x)$
E5	$domaincocok(x, y)$
E6	$relevan(x)$
E7	$NO\_AGG(x)$
E8	$contains(x,$"terbanyak")
E9	$aggregationFunction(x) = MAX$
E10	$contains(x,$"paling sedikit")
E11	$aggregationFunction(x) = MIN$
E12	$contains(x,$hapus$)$ $contains(x,$"database")
E13	$databaseAction(x) = DROP\ DATABASE$
E14	$contains(x,$"tambahkan")
E15	$databaseAction(x) = INSERT$
E16	$contains(x,$"tampilkan")
E17	$databaseAction(x) = SELECT$

Struktur matriks element *rule based approaches* yang dibentuk *rule based approaches* hasil analisis pengetahuan pada penelitian ini ditunjukkan pada tabel berikut:

	R1	R2	R3	R4	R5	R6	R7	R8	R9
E1	A								
E2	C								
E3		A							
E4		C							
E5			A						
E6			C	A					
E7				C					
E8					A				
E9					C				
E10						A			
E11						C			
E12							A		
E13							C		
E14								A	
E15								C	
E16									A
E17									C

Table 9 Tabel Struktur Matriks

Aktifitas kedua pada fase verifikasi adalah memastikan bahwa basis pengetahuan tidak mengalami masalah ketidaklengkapan meliputi *unreferenced attribute values, illegal attribute values, unreachable conclusions*, Dead-End IF Condition dan Dead-End Goal. Pada aktifitas kedua ini selain menggunakan matriks element-rule based approaches, penelitian ini menggunakan matrix element-element.

Matrix element-element menangkap *dependency* antar sepasang elemen. Sepasang elemen dikatakan terjadi dependency apabila satu elemen menjadi antecedent dan elemen lain menjadi conclusion pada sebuah rule based approaches. Baris matrix merepresentasikan antecedent dan kolom matrix merepresentasikan conclusion. Sedangkan elemen matrix diisi dengan ID *Rule based approaches*. Karena matrix elemen-element berisi dependency, matriks element-element adalah matriks tidak simetris. Struktur matrix element-element pada penelitian ini ditunjukkan pada tabel berikut:

		Conclusion																
		E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17
Antecedent	E1	-	R1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E2	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E3	-	-	-	R2	-	-	-	-	-	-	-	-	-	-	-	-	-
	E4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E5	-	-	-	-	-	R3	-	-	-	-	-	-	-	-	-	-	-
	E6	-	-	-	-	-	-	R4	-	-	-	-	-	-	-	-	-	-
	E7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E8	-	-	-	-	-	-	-	-	R5	-	-	-	-	-	-	-	-
	E9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E10	-	-	-	-	-	-	-	-	-	-	R6	-	-	-	-	-	-
	E11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E12	-	-	-	-	-	-	-	-	-	-	-	-	R7	-	-	-	-
	E13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R8	-	-
	E15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R9
	E17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Matrix diatas menunjukkan adanya *dependency* antar elemen yang terlibat dalam *rule based approaches*. Matrix diatas juga

menunjukkan adanya rantai *dependency* yang terjadi akibat keterlibatan E5 sebagai antecedent untuk R4 sekaligus conclusion untuk R3, yaitu E4→E5→E6 dengan menotasikan dependency.

4. Komponen mesin inferensi

   Komponen mesin inferensi memanipulasi basis pengetahuan untuk menyimpulkan informasi dan melakukan penalaran yang diperlukan untuk mencapai solusi. Strategi pada mesin inferensi yang diadopsi untuk pendekatan usulan adalah *forward chaining*. Strategi ini dipilih pada penelitian ini karena sebagian besar data dan fakta diberikan di awal. Selain menggunakan *forward chaining*, pendekatan juga menggunakan strategi *first applicable* untuk penyelesaian konflik ketika reasoning. Strategi ini dipilih karena paling sederhana dan sesuai untuk jumlah rule based approaches yang tidak terlalu besar.

D. Implmenetasi

Implementasi Text-To-SQL berbasis rule-based approaches dengan fokus pada fungsi agregasi query menggunakan bahasa pemrograman Python dan library PyKE dirancang untuk menangani kebutuhan khusus dalam menghasilkan query SQL yang melibatkan operasi agregasi seperti SUM, AVG, MAX, MIN, dan COUNT. Library PyKE memberikan kemampuan untuk mendefinisikan aturan logika berbasis Prolog, yang memungkinkan inferensi berbasis fakta dan aturan untuk menghasilkan sintaks SQL secara otomatis.

Proses dimulai dengan mendefinisikan fakta-fakta yang relevan dengan fungsi agregasi. Fakta ini mencakup informasi seperti nama tabel, kolom yang tersedia, serta jenis data dalam kolom tersebut. Misalnya, fakta dapat menunjukkan bahwa kolom tertentu mengandung data numerik yang cocok untuk agregasi seperti SUM atau AVG, sementara kolom lain cocok untuk agregasi seperti COUNT.

Langkah berikutnya adalah mendefinisikan aturan logika yang mengidentifikasi pola dari pertanyaan dalam bahasa alami yang mengindikasikan kebutuhan agregasi. Aturan ini dapat mencakup frasa seperti "berapa total", "rata-rata", "jumlah", atau "nilai maksimum". Aturan juga memastikan bahwa fungsi agregasi yang dipilih sesuai dengan jenis data pada kolom yang dimaksud. Misalnya,

permintaan untuk "rata-rata" hanya akan diterapkan pada kolom dengan data numerik.

Ketika pertanyaan diajukan, sistem akan mem-parsing input pengguna untuk mengekstrak elemen-elemen penting, seperti jenis fungsi agregasi yang diminta, kolom target, dan kemungkinan filter tambahan. Mesin inferensi PyKE kemudian mencocokkan input ini dengan aturan yang telah ditentukan. Berdasarkan hasil pencocokan, sistem akan membentuk query SQL yang melibatkan fungsi agregasi sesuai permintaan pengguna.

Pendekatan ini memberikan fleksibilitas dalam menangani berbagai kebutuhan agregasi secara spesifik dan memastikan query yang dihasilkan sesuai dengan struktur basis data yang ada. Selain itu, aturan dapat diperluas untuk mendukung fungsi agregasi tambahan atau pola pertanyaan yang lebih kompleks, sehingga sistem dapat berkembang sesuai dengan kebutuhan pengguna.

E. Rancangan Uji Coba

Uji coba dilakukan untuk menguji performa pendekatan usulan. Adapun skenario dan evaluasi untuk menguji pendekatan usulan diuraikan sebagai berikut

1. Skenario Uji Coba

   Pada penelitian ini, data-data yang dikumpulkan dari tahap pengumpulan data dibagi menjadi 2, yaitu: 80% sebagai data latih dan 20% sebagai data uji. Pembagian ini mengikuti praktik baik sejumlah penelitian.

   Data-data yang termasuk data latih berfungsi untuk membentuk basis pengetahuan. Data-data ini melalui proses hingga pembentukan basis pengetahuan selesai. Dengan demikian data-data latih memiliki label jenis fungsi aggregasi dan kata-kata yang telah ditandai.

   Data-data yang termasuk data uji berfungsi untuk menguji performa pendekatan usulan. Data-data ini tidak digunakan untuk membentuk basis pengetahuan. Hal ini dilakukan untuk mengurangi bias. Karena tidak untuk pembentukan basis pengetahuan, data uji

hanya melalui proses pemberian label jenis fungsi agregasi tanpa ada kata-kata yang ditandai.

Pemilihan data untuk latih dan uji dilakukan dengan menerapkan strategi random. Setiap data memiliki probabilitas yang sama untuk menjadi data latih atau uji. Pada penelitian ini, untuk mengurangi ketidakseimbangan pendekatan usulan dalam memprediksi satu jenis fungsi aggregasi, data latih memiliki jumlah data yang sama per jenis fungsi aggregasi.

Penelitian ini hanya menggunakan satu skenario tunggal untuk menguji performa pendekatan usulan yaitu strategi inferensi adalah forward chaining dan strtaegi reasoning adalah first applicable.

2. Evaluasi

Performa pendekatan usulan diukur menggunakan beberapa metric yaitu accuracy, precision, recall dan F1-Score. Keempat metric tersebut dapat dihitung dengan bantuan Confusion Matrix. Confusion Matrix yang digunakan pada penelitian ini terdiri dari 5 baris dan 5 kolom sesuai jumlah jenis fungsi aggregasi. Tabel 10 menunjukkan Confusion Matrix yang digunakan pada penelitian ini.

		sebenarnya						
		SUM	**AVG**	**COUNT**	**MAX**	**MIN**	**NO_AGG**	**Total prediksi**
Prediksi	**SUM**	TP						
	AVG		TP					
	COUNT			TP				
	MAX				TP			
	MIN					TP		
	NO_AGG							
	Total sebenarnya							

Table 10 Confusion matrix untuk 6 label jenis fungsi agregasi.

Pada confusion matrix diatas, baris matrix menunjukkan label-label jenis fungsi agregasi yang diprediksi oleh pendekatan

usulan. Sedangkan kolom matrix menunjukkan jenis fungsi agregasi yang sebenarnya. Diagonal utama Confusion Matrix diisi dengan jumlah data per jenis fungsi agregasi yang diprediksi benar oleh pendekatan usulan. Jumlah ini disebut dengan True Positive (TP). Sedangkan elemen-elemen lain diisi dengan banyaknya data per jenis fungsi agregasi yang diprediksi salah oleh pendekatan usulan.

Recall kelas k dapat dihitung menggunakan Confusion Matrix berdasarkan rumus:

$$recall\ kelas\ k\ = \frac{TP\ kelas\ k}{TP\ kelas\ k + FN\ kelas\ k} \quad 3.1$$

$$TP\ kelas\ k + FN\ kelas\ k = total\ sebenarnya\ kelas\ k \quad 3.2$$

False Negative (FN) menunjukkan banyaknya data yang diprediksi salah oleh pendekatan usulan untuk suatu kelas. Maka, FN dihitung dengan menjumlahkan nilai kolom-kolom untuk satu baris kelas k.

Recall keseluruhan pada penelitian ini menggunakan *macro-averages* yaitu:

$$recall\ keseluruhan\ = \frac{1}{K}\sum_{i=1}^{K} recall_i \quad 3.2$$

Precision per kelas sama halnya denga *recall* per kelas dapat dihitung menggunakan Confusion Matrix berdasarkan rumus:

$$precision\ kelas\ k\ = \frac{TP\ kelas\ k}{TP\ kelas\ k + FPkelas\ k} \quad 3.3$$

$$TP\ kelas\ k + FP\ kelas\ k = total\ prediksi\ kelas\ k \quad 3.4$$

False Positive (FP) sama halnya dengan FN yaitu menunjukkan banyaknya data yang diprediksi salah oleh pendekatan usulan untuk suatu kelas. Namun, FP dihitung dengan menjumlahkan baris-baris untuk satu kolom kelas k.

Precision keseluruhan untuk semua *K*-kelas dapat dihitung dengan *macro-averages* yaitu:

$$precision\ keseluruhan\ = \frac{1}{K}\sum_{i=1}^{K} precision_i \quad 3.5$$

F1-Score adalah gabungan dari precision dan recall untuk keseluruhan kelas. F1-score dapat dihitung dengan:

$$F1\ = 2\,\frac{precision\ keseluruhan * recall\ keseluruhan}{precision\ keseluruhan + recall\ keseluruhan} \quad 3.6$$

DAFTAR PUSTAKA

Deng, N., Chen, Y., & Zhang, Y. (2022). Recent Advances in Text-to-SQL: A Survey of What We Have and What We Expect. *Proceedings - International Conference on Computational Linguistics, COLING*, *29*(1), 2166–2187.

Deng, X., Awadallah, A. H., Meek, C., Polozov, O., Sun, H., & Richardson, M. (2021). Structure-Grounded Pretraining for Text-to-SQL. *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 1337–1350. https://doi.org/10.18653/v1/2021.naacl-main.105

Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., & Zhou, J. (2024). Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *Proceedings of the VLDB Endowment*, *17*(5), 1132–1145. https://doi.org/10.14778/3641204.3641221

Katsogiannis-Meimarakis, G., & Koutrika, G. (2023). A survey on deep learning approaches for text-to-SQL. *VLDB Journal*, *32*(4), 905–936. https://doi.org/10.1007/s00778-022-00776-8

Pourreza, M. (2023). *DIN-SQL : Decomposed In-Context Learning of Text-to-SQL with Self-Correction*. *NeurIPS*, 1–10.

Qin, B., Hui, B., Wang, L., Yang, M., Li, J., Li, B., Geng, R., Cao, R., Sun, J., Si, L., Huang, F., & Li, Y. (2022). *A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions*. 1–19. http://arxiv.org/abs/2208.13629