

Pendekatan Berbasis Rule Untuk Mengidentifikasi Pertanyaan Tak Terdefinisi Pada Masalah *Text-to-SQL*

Mustopa Nurkholik Gifari^{1*}, Agung Prasetya²

^{1,2}Universitas Bhinneka PGRI Tulungagung, Tulungagung, Jawa Timur, 66221, Indonesia

e-mail: ^{*}1mustopagifari@gmail.com, ²agung@ubhi.ac.id

Diterima
17-07-2025

Direvisi
13-08-2025

Disetujui
09-09-2025

Abstract: *This study aims to evaluate the performance of a rule-based approach in detecting undefined questions within the Text-to-SQL problem, which involves translating natural language into SQL queries automatically. The research is motivated by the growing demand for intelligent systems capable of understanding user input and generating accurate database queries. A rule-based expert system was developed using a forward chaining inference strategy, implemented through the Experta library in Python. The knowledge base was constructed based on information collected from 15 domain experts, and the system was tested using 20 evaluation sentences relevant to an inventory database schema. The experimental results indicate that the system achieved a precision of 100%, but the recall was limited to 55.56%, resulting in an overall accuracy of 60%. These findings suggest that strict rule-based approaches are highly effective in recognizing predefined query patterns but remain less capable of handling unseen variations. Future work recommends combining rule-based methods with machine learning to improve performance.*

Keywords: *Text-to-SQL; Rule-Based System; Undefined Questions; Sentence Classification; Experta*

Abstrak: Penelitian ini bertujuan mengevaluasi kinerja pendekatan berbasis aturan dalam mendeteksi pertanyaan tak terdefinisi pada permasalahan Text-to-SQL, yaitu proses penerjemahan bahasa alami menjadi perintah SQL secara otomatis. Penelitian ini dilatarbelakangi oleh meningkatnya kebutuhan sistem yang mampu memahami bahasa pengguna dan menghasilkan kueri yang akurat pada basis data. Sistem pakar dikembangkan dengan strategi inferensi forward chaining yang diimplementasikan melalui pustaka Experta pada bahasa pemrograman Python. Basis pengetahuan disusun berdasarkan hasil akuisisi informasi dari 15 pakar domain, sedangkan pengujian dilakukan menggunakan 20 kalimat uji yang relevan dengan skema basis data inventori. Hasil eksperimen menunjukkan bahwa sistem mencapai tingkat presisi 100%, tetapi nilai recall hanya 55,56%, dengan akurasi keseluruhan 60%. Temuan ini menunjukkan bahwa aturan yang ketat sangat efektif dalam mengenali pola pertanyaan terdefinisi tanpa menghasilkan kesalahan positif, namun masih kurang optimal dalam menangani variasi struktur kalimat. Implikasi penelitian ini membuka peluang pengembangan sistem hybrid yang menggabungkan pendekatan berbasis aturan dengan pembelajaran mesin untuk meningkatkan kinerja sistem secara keseluruhan.

Kata kunci: *Text-to-SQL, Sistem Berbasis Aturan, Pertanyaan Tak Terdefinisi, Klasifikasi Kalimat, Experta*

I. PENDAHULUAN

Structured Query Language (SQL), atau Text-to-SQL, merupakan salah satu bagian penting dalam Pemrosesan Bahasa Alami (NLP). Tujuan utama dari proses ini adalah untuk memungkinkan pengguna yang tidak terbiasa dengan sintaks SQL untuk mengakses dan memanipulasi data yang

tersimpan dalam basis data relasional menggunakan bahasa sehari-hari. Dengan demikian, Text-to-SQL dapat menciptakan antarmuka yang lebih intuitif dan secara signifikan mengurangi hambatan bagi para profesional maupun pengguna awam saat berhadapan dengan repositori data yang kompleks (Katsogiannis-Meimarakis and Koutrika 2023).

Berbagai penelitian terbaru telah menyoroti pentingnya deteksi pertanyaan tak terdefinisi (*undefined questions*) dalam sistem Text-to-SQL untuk meningkatkan keandalan dan akurasi penerjemahan kueri. (Wang et al. 2023) menekankan bahwa keberhasilan sistem *Text-to-SQL* sangat bergantung pada kemampuannya mengenali pertanyaan yang ambigu atau tidak memiliki jawaban yang valid. *PRACTIQ* (Dong et al. 2025) juga memperkenalkan dataset percakapan yang secara eksplisit menguji model terhadap pertanyaan yang sulit atau tidak terjawab, menunjukkan bahwa pendekatan deteksi dini mampu mengurangi kesalahan penerjemahan. Selain itu, (Zeng et al. 2020) melalui sistem Photon mengimplementasikan mekanisme berbasis aturan yang dapat memflag input yang tidak dapat diterjemahkan menjadi kueri SQL, sementara (Somov and Tutubalina 2025) mengusulkan metode *confidence estimation* untuk mengukur tingkat ketidakpastian model sebelum proses penerjemahan dilakukan. Penelitian-penelitian ini menjadi dasar penting bagi pengembangan pendekatan berbasis aturan yang diajukan dalam studi ini.

Dalam perkembangannya, berbagai metode telah dikembangkan untuk tugas *Text-to-SQL*, mencakup pendekatan berbasis aturan/*rule-based*, berbasis *template*, serta pembelajaran mesin seperti jaringan saraf tiruan dan model transformer. Pendekatan berbasis aturan memanfaatkan struktur formal dari bahasa SQL untuk mendefinisikan aturan-aturan pemrosesan yang mudah diinterpretasikan, namun seringkali kurang fleksibel dalam menghadapi variasi bahasa. Di sisi lain, metode berbasis pembelajaran mesin mampu menangani pertanyaan yang lebih rumit, namun seringkali kesulitan dalam menangani pertanyaan yang ambigu atau sangat bervariasi, yang dapat menyebabkan kesalahan dalam penerjemahan atau kegagalan sistem dalam menghasilkan kueri yang sesuai (Jeong dkk. 2023) (Zeng dkk. 2020).

Kelemahan ini pada akhirnya dapat meningkatkan jumlah pertanyaan yang tidak dapat diproses dengan benar oleh sistem. Menangani pertanyaan tak terdefinisi yaitu input yang tidak dapat diterjemahkan menjadi kueri SQL yang valid menjadi tantangan krusial (Wang dkk. 2023).

Penelitian ini mengusulkan sebuah pendekatan berbasis aturan yang secara spesifik berfokus untuk mengidentifikasi "pertanyaan tak terdefinisi" sebelum sistem mencoba menerjemahkannya ke SQL. Dengan mendeteksi pertanyaan-pertanyaan problematik ini di tahap awal, diharapkan sistem Text-to-SQL secara keseluruhan dapat menjadi lebih andal dan efisien. Oleh karena itu, tujuan dari penelitian ini adalah untuk menganalisis dan mengevaluasi performa pendekatan berbasis aturan dalam mengidentifikasi pertanyaan tak terdefinisi, menilai sejauh mana akurasinya, serta mengidentifikasi tantangan yang dihadapi dalam menangani variasi bahasa alami.

II. METODE PENELITIAN

Penelitian ini menggunakan rancangan yang terdiri dari enam tahapan utama: studi literatur, pengumpulan data, perancangan pendekatan berbasis aturan, implementasi sistem, uji coba, dan penyusunan laporan. Tahapan-tahapan ini disusun secara sistematis untuk memastikan pendekatan yang diusulkan dapat dikembangkan dan dievaluasi dengan baik. Prosedur pengumpulan data dilakukan dengan teknik analisis dokumen.

Sumber data berupa dokumen ilmiah seperti jurnal, artikel konferensi, dan laporan tugas akhir yang memuat masalah riil, telah melalui proses review, serta menggunakan bahasa Indonesia baku. Kriteria pemilihan dokumen meliputi aksesibilitas publik, keberadaan *Entity Relationship Diagram* (ERD), serta deskripsi sistem atau proses bisnis dalam domain yang relevan (hotel, inventori, akademik, dll.). Data yang dikumpulkan dari dokumen-dokumen ini adalah ERD dan kalimat-kalimat bahasa alami. ERD yang terkumpul kemudian digambar ulang menggunakan notasi *Crow's Foot* (Furqan, Risawandi, and Rosnita 2022) dengan melakukan standarisasi nama (misalnya mengubah singkatan dan menghilangkan kata tidak relevan). Sementara itu, kalimat diturunkan secara natural dari deskripsi kebutuhan atau proses bisnis yang ada pada dokumen untuk memastikan

keselarasan antara kalimat dan skema basis data. Analisis data dan perancangan sistem dilakukan melalui pendekatan berbasis aturan (*rule-based*) yang terdiri dari tiga komponen utama: antarmuka pengguna, mesin inferensi, dan basis pengetahuan (Rahayu et al. 2025). Alur sistem dimulai saat kalimat masukan dari pengguna diproses oleh komponen antarmuka pengguna. Komponen ini tidak memiliki tampilan grafis, melainkan berfungsi untuk melakukan pemrosesan bahasa alami yang mencakup empat sub-proses:

1. filtering untuk memastikan masukan adalah kalimat utuh,
2. Text cleaning untuk mengubah teks menjadi huruf kecil dan menghapus tanda baca
3. Tokenisasi untuk memecah kalimat menjadi daftar kata,
4. Stemming untuk mengubah kata berimbuhan menjadi kata dasar.

Selanjutnya, mesin inferensi akan memanipulasi pengetahuan yang tersimpan dalam basis pengetahuan untuk menghasilkan kesimpulan apakah kalimat tersebut terdefinisi atau tidak. Basis pengetahuan ini dibangun melalui proses akuisisi pengetahuan dari 15 pakar domain (5 akademisi dan 10 profesional) untuk mengurangi bias dan meningkatkan kredibilitas. Teknik akuisisi yang digunakan adalah *Consensus Decision-Making (Voting)*, di mana setiap pakar mengklasifikasikan kalimat sebagai "terdefinisi" atau "tidak terdefinisi" dan memberikan alasannya. Hasil dari pakar kemudian dianalisis dan direpresentasikan sebagai aturan *IF-THEN* dalam bentuk *First Order Logic (FOL)* untuk mengurangi ambiguitas (Pratiwi 2024).

Basis pengetahuan yang terbentuk kemudian diverifikasi untuk memastikan tidak ada masalah konsistensi (aturan redundan, konflik, atau melingkar) dan kelengkapan menggunakan matriks elemen-aturan dan matriks elemen-elemen. Untuk proses penarikan kesimpulan, mesin inferensi menerapkan strategi *forward chaining* dengan resolusi konflik *first applicable* (Efendy, Alfeno, and Christianto 2016). Tahap terakhir adalah uji coba, di mana data yang telah dikumpulkan dibagi menjadi 80% data latih dan 20% data uji secara acak. Data latih digunakan untuk membangun basis pengetahuan, sedangkan data uji digunakan untuk mengevaluasi kinerja sistem guna menghindari bias. Performa pendekatan yang diusulkan diukur menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-Score* yang dihitung berdasarkan *confusion matrix*.

III. HASIL

Pengujian sistem dilakukan terhadap 20 kalimat uji yang telah dilabeli secara manual untuk menghasilkan ground-truth. Setiap kalimat dikirim ke *endpoint* API yang telah diimplementasikan, dan hasil prediksi dari sistem dibandingkan dengan label yang sebenarnya. Hasil dari komparasi tersebut disajikan dalam bentuk *Confusion Matrix* pada Tabel 1.

Tabel 1. Perbandingan sistem dengan label

Prediksi/Label	Terdefinisi	Tak Terdefinisi
Terdefinisi	10 TP	0
Tak Terdefinisi	8 FN	8 TN

Berdasarkan nilai dari confusion matrix di atas, metrik performa utama sistem dihitung dan disajikan pada Tabel 2.

Tabel 2. Hasil Metrik

Metrik	Rumus	Nilai
Akurasi	$(TP + TN) / (TP + FP + FN + TN)$	60.00%
Presisi	$TP / (TP + FP)$	100.00%
Recall	$TP / (TP + FN)$	56.56%
F1-Score	$2(Presisi \cdot Recall) / (Presisi + Recall)$	71.43%

IV. PEMBAHASAN

Untuk memberikan gambaran mengenai cara kerja *knowledge base* dan mengapa hasil performa (khususnya presisi dan *recall*) menjadi seperti yang telah dipaparkan, berikut adalah beberapa contoh aturan kunci yang diimplementasikan dalam sistem.

```
@Rule( AND( Kata(value="tampil"), Kata(value="semua"),
Kata(adalah_nama_tabel=True) ) ) def rule_tampilkan_semua(self):
self.declare(JenisKalimat(value="TERDEFINISI_TAMPIL_SEMUA"))
```

Gambar 1. Rule Dasar

Rule pada Gambar 1 adalah contoh paling fundamental dalam sistem. Sifatnya yang sangat spesifik—membutuhkan tiga fakta terpisah (Kata(value="tampil"), Kata(value="semua"), dan Kata(adalah_nama_tabel=True)) untuk aktif—menjelaskan mengapa nilai Presisi bisa mencapai 100%. Ketika sebuah kalimat cocok persis dengan pola ini, hampir tidak mungkin terjadi kesalahan klasifikasi.

```
@Rule(
    AND(
        Kata(value="cari"),
        SyaratFilter()
    )
)
def rule_filter_kondisi(self):
    self.declare(JenisKalimat(value="TERDEFINISI_FILTER"))
```

Gambar 2. Deklarasi Aturan Deteksi menggunakan filter

Aturan pada Gambar 2 menunjukkan kemampuan sistem yang lebih maju. Aturan ini tidak hanya mencocokkan kata, tetapi juga sebuah *Fact* turunan yaitu Syarat Filter(), yang dideklarasikan saat sistem menemukan pola kondisi (misalnya, "stok kurang dari 10"). Meskipun lebih canggih, aturan ini masih kaku. Kalimat uji seperti "Tampilkan barang yang paling banyak dikembalikan" tidak memicu aturan ini karena tidak mengandung kata cari secara eksplisit, yang berkontribusi pada rendahnya nilai *Recall*.

```
@Rule(OpiniPertanyaan())
def rule_opini(self):
    self.declare(JenisKalimat(value="OPINI_PERTANYAAN"))
```

Gambar 3. Deklarasi Aturan Identifikasi Kalimat tidak terdefinisi

Aturan pada Gambar 3 sangat krusial sesuai dengan tujuan penelitian. Aturan ini secara spesifik menangani kalimat tak terdefinisi dengan cara mendeteksi *Fact* Opini_Pertanyaan() yang dideklarasikan jika kalimat mengandung kata tanya eksploratif seperti "apakah", "bagaimana", atau "mengapa". Ini adalah mekanisme utama sistem untuk menyaring kalimat yang tidak bisa dijawab oleh *database*.

V. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa pendekatan berbasis aturan yang dikembangkan mampu mengidentifikasi pertanyaan tak terdefinisi dengan karakteristik performa yang sangat spesifik. Sistem ini menunjukkan efektivitas yang sangat tinggi dalam hal keandalan, yang dibuktikan dengan pencapaian nilai Presisi sebesar 100.00%. Hal ini menjawab tujuan penelitian bahwa aturan yang ketat dapat secara akurat mengklasifikasikan pola kalimat yang telah dikenal tanpa menghasilkan *false positive*. Namun, temuan penelitian juga menyoroti tantangan

utama dari pendekatan ini, yaitu keterbatasan dalam generalisasi yang tercermin dari nilai *Recall* rendahnya *recall* disebabkan oleh basis pengetahuan yang belum cukup luas untuk mencakup semua variasi linguistik dalam data uji. Untuk pengembangan selanjutnya, disarankan agar fokus utama adalah memperluas *knowledge base* secara signifikan dengan menambahkan aturan-aturan yang dapat menangani pola kalimat kompleks untuk meningkatkan *recall*, serta mempertimbangkan implementasi pendekatan hibrida dengan *machine learning* untuk meningkatkan fleksibilitas sistem.

REFERENSI

- Dong, Mingwen, Nischal Ashok Kumar, Yiqun Hu, Anuj Chauhan, Chung-Wei Hang, Shuaichen Chang, Lin Pan, Wuwei Lan, Henghui Zhu, Jiarong Jiang, Patrick Ng, and Zhiguo Wang. 2025. "PRACTIQ: A Practical Conversational Text-to-SQL Dataset with Ambiguous and Unanswerable Queries." (August 2023):255–73. doi: 10.18653/v1/2025.naacl-long.13.
- Efendy, Muhammad Yusuf, Sandro Alfeno, and Aditya Christianto. 2016. "Penerapan Backward Chaining Sebagai Model Criminal Investigation Expert System (CRIES) Untuk Menangani Kasus Pembunuhan." *Jurnal Sisfotek Global* 6(2).
- Furqan, Hafizul, Risawandi Risawandi, and Lidya Rosnita. 2022. "Sistem Pendukung Keputusan Pemilihan Laptop Pada E-Commerce Menggunakan Metode Simple Multi Attribute Rating Technique." *Jurnal Teknologi Terapan and Sains 4.0* 3(1):651–62.
- Jeong, Geunyeong, Mirae Han, Seulgi Kim, Yejin Lee, Joosang Lee, Seongsik Park, and Harksoo Kim. 2023. "Improving Text-to-SQL with a Hybrid Decoding Method." *Entropy* 25(3):1–20. doi: 10.3390/e25030513.
- Katsogiannis-Meimarakis, George, and Georgia Koutrika. 2023. "A Survey on Deep Learning Approaches for Text-to-SQL." *The VLDB Journal* 32(4):905–36. doi: 10.1007/s00778-022-00776-8.
- Pratiwi, Heny. 2024. *Buku Ajar Kecerdasan Buatan: Disertai Praktik Baik Pemanfaatannya*. Asadel Liamsindo Teknologi.
- Rahayu, Prastyadi Wibawa, Loso Judijanto, Hasbu Naim Syaddad, Apriyanto Apriyanto, and Rabiah Adawiyah. 2025. *Pengantar Sistem Pakar: Teori Dan Implementasi*. PT. Green Pustaka Indonesia.
- Somov, Oleg, and Elena Tutubalina. 2025. "Confidence Estimation for Error Detection in Text-to-SQL Systems." *Proceedings of the AAAI Conference on Artificial Intelligence* 39(23):25137–45. doi: 10.1609/aaai.v39i23.34699.
- Wang, Bing, Yan Gao, Zhoujun Li, and Jian Guang Lou. 2023. "Know What I Don't Know: Handling Ambiguous and Unanswerable Questions for Text-to-SQL." *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 1:5701–14. doi: 10.18653/v1/2023.findings-acl.352.
- Zeng, Jichuan, Xi Victoria Lin, Caiming Xiong, Richard Socher, Michael R. Lyu, Irwin King, and Steven C. H. Hoi. 2020. "PHOTON: A Robust Cross-Domain Text-to-SQL System." *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 204–14. doi: 10.18653/v1/2020.acl-demos.24.