HIJM
HORIZON: Indonesian Journal of Multidisciplinary

p-ISSN: 2987-7776
e-ISSN: 2987-7180
Vol. 4, No. 4, 2026

DETEKSI ILL-FORMED NATURAL LANGUAGE QUERY BERBASIS RULE PADA MODEL LLM TEXT-TO-SQL BERBAHASA INDONESIA

Ricky Fajar Mukti[1], Agung Prasetya[2], Taufiq Agung Cahyono[3]
[1, 2, 3]Universitas Bhinneka PGRI Tulungagung, Jl. Mayor Sujadi No.7, Tulungagung, Jawa Timur, Indonesia
Email: 1ricokky20@gmail.com

Article History

Received: 04-07-2026

Revision: 23-07-2026

Accepted: 27-07-2026

Published: 30-07-2026

Abstract. The performance of a Large Language Model (LLM)-based Text-to-SQL system is heavily influenced by the quality of the user-supplied Natural Language Queries (NLQs). Ill-formed NLQs, such as incomplete, ambiguous, or inconsistent information within the database schema, can generate syntactically valid but semantically incorrect SQL queries. This study aims to develop a rule-based ill-formed NLQ detection system as a pre-processing stage in an Indonesian Text-to-SQL pipeline. The system is built using the Experta library with 15 IF-THEN rules that identify six categories of ill-formed NLQs: insufficient information, multiple interpretations, incomplete conditions, illogical, out-of-schema reference, and unsafe/non-SELECT. After rule-only filtering, a LLaMA 3.2 3B Instruct-based validator is optionally used to examine cases requiring deeper semantic understanding. An evaluation of 202 Indonesian NLQs showed that the Rule-only mode achieved 89.60% Recall, 78.87% Precision, 83.90% F1-Score, and 78.71% Accuracy. False positives were primarily influenced by overly strict rules, while false negatives were dominated by semantic ambiguity. These results demonstrate that the rule-based approach provides a consistent and transparent initial filtering mechanism and can be combined with an LLM-based validator to handle cases requiring more complex semantic interpretation.

Keywords: Text-to-SQL, Ill-formed NLQ, Rule-based System, Experta, LLaMA, Indonesian Language.

Abstrak. Kinerja sistem Text-to-SQL berbasis *Large Language Model* (LLM) sangat dipengaruhi oleh kualitas *Natural Language Query* (NLQ) yang diberikan pengguna. NLQ yang *ill-formed*, seperti informasi yang tidak lengkap, ambigu, atau tidak sesuai dengan skema basis data, dapat menghasilkan query SQL yang valid secara sintaksis tetapi keliru secara semantik. Penelitian ini bertujuan mengembangkan sistem deteksi *ill-formed* NLQ berbasis *rule* sebagai tahap pra-proses pada pipeline Text-to-SQL berbahasa Indonesia. Sistem dibangun menggunakan pustaka Experta dengan 15 aturan IF-THEN yang mengidentifikasi enam kategori *ill-formed* NLQ, yaitu informasi kurang, multitafsir, syarat tidak lengkap, tidak logis, *out-of-schema reference*, dan *unsafe/non-SELECT*. Setelah melalui penyaringan berbasis *rule* (*Rule-only*), validator berbasis LLaMA 3.2 3B Instruct digunakan secara opsional untuk memeriksa kasus yang memerlukan pemahaman semantik lebih mendalam. Evaluasi terhadap 202 NLQ berbahasa Indonesia menunjukkan bahwa mode *Rule-only* memperoleh *Recall* 89,60%, *Precision* 78,87%, *F1-Score* 83,90%, dan *Accuracy* 78,71%. *False Positive* terutama dipengaruhi oleh aturan yang terlalu ketat, sedangkan *False Negative* didominasi oleh ambiguitas semantik. Hasil tersebut menunjukkan bahwa pendekatan berbasis *rule* mampu memberikan mekanisme penyaringan awal yang konsisten dan transparan, serta dapat dikombinasikan dengan validator berbasis LLM untuk menangani kasus yang memerlukan interpretasi semantik yang lebih kompleks.

Kata Kunci: *Text-to-SQL, Ill-formed NLQ, Rule-based System, Experta, LLaMA,* Bahasa Indonesia.

How to Cite: Mukti, R. F., Prasetya, A., & Cahyono, T. A. (2026). *Deteksi Ill-Formed Natural Language Query* Berbasis *Rule* Pada Model *Llm Text-to-SQL* Berbahasa Indonesia. *HORIZON: Indonesian Journal of Multidisciplinary*, 4 (4), 5088-5098. http://doi.org/10.54373/hijm.v4i4.6916

PENDAHULUAN

Text-to-SQL merupakan teknologi yang memungkinkan pengguna mengakses basis data menggunakan bahasa alami tanpa harus menulis perintah SQL secara langsung. Perkembangan *Large Language Model* (LLM) telah meningkatkan kemampuan sistem Text-to-SQL dalam menerjemahkan *Natural Language Query* (NLQ) menjadi query SQL dengan akurasi yang semakin baik. Berbagai penelitian, seperti Solid-SQL (Liu et al., 2025), PICARD (Scholak et al., 2021), DIN-SQL (Ye et al., 2023), dan *Execution Consistency* (Yang et al., 2025), berfokus pada peningkatan kualitas proses generasi SQL melalui *schema linking*, *constrained decoding*, maupun *in-context learning*. Namun, penelitian-penelitian tersebut umumnya mengasumsikan bahwa NLQ yang diberikan pengguna telah tersusun dengan baik sehingga belum menyediakan mekanisme untuk mendeteksi kualitas input sebelum diproses oleh LLM.

Permasalahan tersebut menjadi penting karena dalam praktiknya pengguna sering mengajukan NLQ yang *ill-formed*, seperti ambigu, tidak lengkap, tidak logis, atau mengacu pada atribut yang tidak terdapat pada skema basis data. Kondisi ini dapat menyebabkan LLM menghasilkan query SQL yang valid secara sintaksis, tetapi tidak sesuai dengan maksud pengguna sehingga menurunkan kualitas hasil pencarian data. Dengan demikian, kualitas input menjadi faktor yang memengaruhi keberhasilan sistem Text-to-SQL, terutama pada implementasi berbasis bahasa Indonesia yang masih memiliki keterbatasan dataset dan sumber daya penelitian.

Beberapa penelitian di Indonesia telah membahas implementasi Text-to-SQL dan pemanfaatan LLM pada sistem basis data, namun sebagian besar masih berfokus pada peningkatan akurasi penerjemahan query atau pengembangan antarmuka bahasa alami tanpa melakukan deteksi terhadap NLQ yang bermasalah sebelum diproses oleh model (Haryono & Prasetya, 2025; Nugraha & Kurniawan, 2024). Penelitian lain mengenai sistem berbasis aturan menunjukkan bahwa pendekatan *rule-based* masih relevan untuk menangani permasalahan yang memiliki pola linguistik yang jelas karena menghasilkan proses pengambilan keputusan yang transparan dan mudah diinterpretasikan (Sari et al., 2023). Meskipun demikian, penerapan pendekatan *rule-based* untuk mendeteksi *ill-formed* NLQ sebagai tahap *pre-processing* pada pipeline LLM Text-to-SQL berbahasa Indonesia masih belum banyak dilaporkan.

Berdasarkan kondisi tersebut, penelitian ini mengusulkan sistem deteksi *ill-formed Natural Language Query* berbasis aturan (*rule-based*) sebagai tahap pra-proses pada pipeline LLM Text-to-SQL berbahasa Indonesia. Sistem dirancang menggunakan aturan IF–THEN untuk mengidentifikasi enam kategori kesalahan NLQ sebelum diproses oleh model LLM, sedangkan validator berbasis LLM digunakan sebagai komponen opsional untuk menangani

kasus yang memerlukan pemahaman semantik yang lebih kompleks. Pendekatan ini diharapkan dapat meningkatkan kualitas input yang diterima sistem sekaligus menyediakan mekanisme deteksi yang konsisten dan transparan. Penelitian ini bertujuan merancang, mengimplementasikan, dan mengevaluasi kinerja sistem deteksi *ill-formed* NLQ berbasis aturan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*.

METODE

Penelitian ini merupakan penelitian rekayasa sistem (*engineering research*) dengan evaluasi eksperimental komputasional. Penelitian berfokus pada perancangan, implementasi, dan evaluasi kinerja sistem deteksi *ill-formed Natural Language Query* (NLQ) berbasis aturan (*rule-based*) sebagai tahap pra-proses pada pipeline LLM Text-to-SQL berbahasa Indonesia. Evaluasi dilakukan melalui pengujian sistem menggunakan dataset NLQ berlabel untuk mengukur kemampuan klasifikasi model berdasarkan metrik *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*. Tahapan penelitian meliputi studi literatur, pembentukan dataset, perancangan aturan, implementasi sistem, pengujian dan evaluasi, serta penyusunan laporan.

Pembentukan Dataset

Dataset uji disusun secara sintetis dengan mengadaptasi prosedur pembangkitan data pada penelitian *SQL-GEN: Bridging the Dialect Gap for Text-to-SQL via Synthetic Data and Model Merging*. Proses pembentukan dataset dilakukan melalui empat tahap. Pertama, **Seed** Template Extraction, yaitu menyusun kumpulan template NLQ *well-formed* berbahasa Indonesia berdasarkan skema basis data akademik dan kebutuhan informasi yang umum diajukan pengguna. Kedua, Template Expansion, yaitu menghasilkan variasi sinonim, struktur kalimat, dan bentuk pertanyaan untuk meningkatkan keragaman linguistik. Ketiga, Template Filling, yaitu mengisi template menggunakan nama tabel, atribut, dan nilai literal sesuai skema database, kemudian memodifikasinya untuk menghasilkan NLQ *ill-formed* melalui penghilangan informasi penting, penggunaan syarat yang tidak lengkap, penambahan ambiguitas, penggunaan referensi di luar skema, atau penyusunan pertanyaan yang tidak logis. Keempat, Quality Check, yaitu proses pemeriksaan ulang dan pelabelan akhir untuk memastikan kesesuaian kategori *well-formed* maupun *ill-formed* berdasarkan definisi operasional yang telah ditetapkan.

Skema database yang digunakan berupa skema akademik simulasi yang terdiri atas lima tabel relasional, yaitu mahasiswa (*nim, nama, jurusan, ipk, angkatan*), matakuliah (*kode_mk, nama_mk, sks, semester*), nilai (*nim, kode_mk, nilai_angka, nilai_huruf*), dosen (*nip,*

nama_dosen, jurusan, jabatan), dan jadwal (*kode_mk, hari, jam, ruang, nip*). Dataset akhir terdiri atas 202 NLQ berbahasa Indonesia yang mencakup data *well-formed* dan *ill-formed*. Seluruh sampel diberi label secara manual berdasarkan enam kategori kesalahan yang digunakan dalam penelitian, yaitu informasi kurang, multitafsir, syarat tidak lengkap, tidak logis, *out-of-schema reference*, dan *unsafe/non-SELECT*.

Perancangan Sistem Berbasis Rule

Ill-formed NLQ dikategorikan ke dalam enam kategori kesalahan sesuai batasan penelitian, yaitu informasi kurang, multitafsir, syarat tidak lengkap, tidak logis, *out-of-schema reference*, dan *unsafe/non-SELECT*. Setiap kategori dipetakan ke dalam aturan berformat *IF–THEN*, di mana kondisi merepresentasikan pola kesalahan NLQ dan aksi merepresentasikan keputusan klasifikasi beserta jenis kesalahan yang terdeteksi. Sebelum aturan dijalankan, setiap NLQ diproses oleh modul *Feature Extractor* yang mengekstraksi enam fitur utama: *token_count*, *has_entity*, *has_attribute*, *has_condition*, *has_sql_hint*, dan *is_ambiguous*. Fitur-fitur ini dikemas ke dalam objek *NLQFact* sebagai input bagi *Rule Engine*.

Implementasi Sistem

Sistem diimplementasikan menggunakan bahasa pemrograman Python dengan *library Experta*, yaitu implementasi Python dari CLIPS *(C Language Integrated Production System)* yang mendukung mekanisme inferensi *forward chaining*. Rule Engine mendeklarasikan objek *NLQFact* ke dalam *working memory* (*declare()*), menjalankan *run()* untuk pencocokan pola terhadap 15 rule yang telah dirancang, kemudian menghasilkan fakta *IllFormedFact* berisi kode rule, *severity*, deskripsi, dan saran perbaikan apabila suatu kondisi terpenuhi. Sebagai komponen opsional, model LLaMA 3.2 3B Instruct dari Meta (via HuggingFace Transformers) diintegrasikan sebagai *LLM Validator* untuk menangani kasus ambiguitas semantik yang tidak tercakup oleh pola rule. Kedua komponen digabungkan melalui mekanisme *Hybrid Decision* untuk menentukan status akhir NLQ. Implementasi dijalankan pada lingkungan Google Colaboratory dengan akselerator GPU T4, menggunakan library *experta, transformers, accelerate, bitsandbytes, pandas,* dan *openpyxl*.

Evaluasi Kinerja

Evaluasi kinerja detektor dilakukan menggunakan *confusion matrix* sebagai dasar perhitungan metrik *accuracy, precision, recall,* dan *F1-score* terhadap 202 sampel dataset pada mode *Rule-only*, untuk mengukur kemampuan sistem dalam membedakan NLQ *well-formed* dan *ill-formed* secara tepat

HASIL

Arsitektur dan Cara Kerja Sistem

Sistem detektor terdiri atas dua komponen utama yang saling melengkapi, yaitu *Rule Engine* berbasis Experta dan *LLM Validator* berbasis LLaMA 3.2 3B Instruct, yang digabungkan melalui mekanisme *Hybrid Decision*. Alur kerja dimulai dari input NLQ berbahasa Indonesia, diproses oleh *Feature Extractor* untuk mengidentifikasi fitur linguistik dan semantik, kemudian dicocokkan terhadap 15 aturan *IF-THEN* pada *Rule Engine.* Seluruh proses *Rule Engine* berjalan secara deterministik tanpa komputasi *neural network,* sehingga setiap keputusan dapat ditelusuri langsung ke rule yang terpicu. Dari 15 rule tersebut, enam rule utama (R01, R03, R05, R06, R10, R12) memetakan langsung enam kategori ill-formed NLQ yang didefinisikan, sedangkan sembilan rule tambahan berfungsi sebagai perluasan yang menangkap variasi pola kesalahan linguistik bahasa Indonesia. Daftar lengkap rule disajikan pada Tabel 1.

Tabel 1. Daftar rule sistem detektor Ill-formed NLQ

ID	Nama Rule	Kondisi IF	Severity	Kategori
R01	TOO_SHORT	< 3 token	High	Informasi Kurang
R02	TOO_LONG	> 60 token	Medium	Informasi Berlebih
R03	NO_ENTITY	Tidak ada entitas DB	High	Informasi Kurang
R04	NO_ATTRIBUTE	Tidak ada nama kolom	Medium	Informasi Kurang
R05	AMBIGUOUS	Kata ambigu + no entity	High	Multitafsir
R06	INCOMPLETE	Fragment kalimat	High	Informasi Kurang
R07	ONLY_STOPWORDS	Semua token stopword	High	Informasi Kurang
R08	DUPLICATE_WORDS	Token berulang berturutan	Low	Kesalahan Struktur
R09	SPECIAL_CHARS	> 3 karakter khusus	Medium	Kesalahan Struktur
R10	RAW_SQL	Diawali keyword SQL	Medium	Unsafe/Non-SELECT
R11	DOUBLE_NEGATION	Negasi ganda	Medium	Tidak Logis
R12	UNKNOWN_COLUMN	Referensi kolom di luar schema	Medium	Out-of-Schema Ref
R13	NOISE_WORDS	Kata polite marker/slang	Low	Informasi Kurang
R14	MIXED_LANGUAGE	Campur ID-EN tidak konsisten	Low	Kesalahan Struktur
R15	NO_CONDITION	Ada entitas+atribut, tanpa kondisi	Low	Syarat Tidak Lengkap

Karakteristik Dataset Pengujian

Dataset pengujian yang digunakan terdiri dari 202 sampel NLQ berbahasa Indonesia, dengan distribusi 76 sampel *well-formed* (37,6%) dan 126 sampel *ill-formed* (62,4%), sebagaimana disajikan pada Tabel 2.

Tabel 2. Distribusi dataset pengujian

Kategori	Jumlah	Persentase	Keterangan
Well-formed	76	37,6%	NLQ valid dengan struktur tepat; mencakup query normal, filter, JOIN, agregasi, dan sorting.
Ill-formed	126	62,4%	NLQ bermasalah, distribusi merata pada 15 kategori kesalahan.
Total	**202**	**100%**	**Dataset lengkap yang digunakan untuk evaluasi sistem detektor.**

Hasil Evaluasi Kinerja Detektor

Evaluasi dilakukan menggunakan mode *Rule Engine* saja (use_llm=False) terhadap 202 sampel dataset. Hasil evaluasi menghasilkan matriks konfusi pada Tabel 3.

Tabel 3. Confusion Matrix Hasil Evaluasi Detektor (Rule-only)

Aktual \ Prediksi	Pred: Ill-formed	Pred: Well-formed
Aktual: Ill-formed	TP = 112	FN = 13
Aktual: Well-formed	FP = 30	TN = 47

Berdasarkan matriks konfusi tersebut diperoleh 112 *True Positive*, 47 *True Negative*, 30 *False Positive*, dan 13 *False Negative*. Nilai metrik evaluasi disajikan pada Tabel 4.

Tabel 4. Metrik evaluasi kinerja detektor

Metrik	Nilai	Formula	Keterangan
Accuracy	78,71%	(112+47)/(112+47+30+13)	Proporsi prediksi benar keseluruhan
Precision	78,87%	112/(112+30)	Ketepatan saat memprediksi ill-formed
Recall	89,60%	112/(112+13)	Kemampuan mendeteksi semua ill-formed
F1-Score	83,90%	2×(P×R)/(P+R)	Harmonik precision dan recall

Hasil evaluasi menunjukkan bahwa detektor berbasis aturan memiliki *Recall* sebesar 89,60%, yang mengindikasikan kemampuan tinggi dalam mengenali NLQ bermasalah. Hanya 13 dari 125 NLQ ill-formed yang tidak terdeteksi (*False Negative*), sehingga sistem relatif efektif sebagai mekanisme penyaringan awal (*pre-filter*) untuk mencegah pertanyaan bermasalah diproses oleh model Text-to-SQL. Karakteristik ini penting karena kesalahan yang tidak terdeteksi berpotensi menghasilkan SQL yang secara sintaksis benar, tetapi keliru secara semantik. Di sisi lain, *Precision* sebesar 78,87% menunjukkan masih terdapat 30 *False Positive*, yaitu NLQ yang sebenarnya valid tetapi diklasifikasikan sebagai ill-formed. Temuan ini mengindikasikan bahwa beberapa aturan masih bersifat konservatif sehingga lebih memilih menandai query sebagai bermasalah daripada berisiko meloloskan query yang ambigu.

Pendekatan tersebut meningkatkan sensitivitas deteksi, tetapi berdampak pada berkurangnya spesifisitas sistem.

Nilai *Accuracy* sebesar 78,71% dan *F1-Score* sebesar 83,90% memperlihatkan bahwa sistem mampu mempertahankan keseimbangan antara kemampuan mendeteksi NLQ ill-formed dan meminimalkan kesalahan klasifikasi. Meskipun demikian, hasil ini belum menunjukkan bahwa pendekatan berbasis aturan telah menyelesaikan seluruh permasalahan ill-formed NLQ. Kinerja sistem masih dipengaruhi oleh keterbatasan aturan dalam menangani ambiguitas semantik, konteks implisit, dan variasi bahasa alami yang tidak dapat direpresentasikan secara eksplisit. Oleh karena itu, hasil penelitian mendukung penggunaan detektor berbasis aturan sebagai lapisan validasi awal yang transparan dan deterministik, yang dapat dikombinasikan dengan validator berbasis LLM untuk menangani kasus-kasus yang memerlukan penalaran semantik lebih mendalam.

Analisis False Positive (FP = 30)

Analisis terhadap 30 kasus *False Positive* menunjukkan bahwa seluruh kesalahan berasal dari tiga penyebab utama sebagaimana disajikan pada Tabel 5.

Tabel 5. Analisis penyebab false positive

Rule Penyebab	Jumlah	Penjelasan
R15_NO_CONDITION	16	Mendeteksi query valid tanpa kondisi WHERE eksplisit sebagai ill-formed, padahal query tersebut valid untuk SELECT seluruh data.
R04_NO_ATTRIBUTE	12	Menganggap query tidak menyebutkan nama kolom secara eksplisit, meskipun maksud query sudah jelas dari konteks.
R03_NO_ENTITY + R05	2	Query menyebutkan entitas secara tidak langsung (sinonim/singkatan) sehingga tidak terdeteksi sebagai entitas valid.

Penyebab utama FP adalah rule R15_NO_CONDITION (16 kasus) dan R04_NO_ATTRIBUTE (12 kasus). Rule R15 mendeteksi query yang memiliki entitas dan atribut namun tidak memiliki kondisi filter eksplisit sebagai *ill-formed*, padahal dalam banyak kasus nyata, query *SELECT* seluruh data tanpa filter adalah valid. Misalnya, query "Tampilkan nama dan IPK mahasiswa jurusan Teknik Informatika" dianggap *ill-formed* oleh R15 karena tidak ada kondisi angka eksplisit, padahal "jurusan Teknik Informatika" sendiri sudah merupakan kondisi filter. Temuan ini mengindikasikan perlunya penyempurnaan threshold pada rule R15 dan R04.

Analisis False Negative (FN = 13)

Analisis terhadap 13 kasus *False Negative* mengidentifikasi empat tipe kesalahan sebagaimana disajikan pada Tabel 6.

Tabel 6. Analisis Penyebab False Negative

Tipe FN	Jml	Analisis
Ambiguitas Subjektif	8	Contoh: "cari dosen yang galak", "tampilkan nilai yang memuaskan". Kata sifat subjektif tidak terdeteksi rule berbasis pola tetap dan memerlukan pemahaman semantik.
Terlalu Panjang Lolos Threshold	2	Query sangat panjang (>60 token) namun memiliki entitas valid sehingga R02 tidak terpicu.
Karakter Noise Tidak Terdeteksi	2	Pola noise (mis. komentar SQL, karakter berulang) tidak tercakup dalam daftar regex yang ada.
Query No_Condition Lolos	1	Memiliki dua entitas sekaligus sehingga R15 tidak terpicu.

Penyebab terbesar FN adalah ambiguitas semantik berbasis kata sifat subjektif (8 dari 13 kasus), seperti “cari dosen yang galak” atau “tampilkan nilai yang memuaskan”. Query-query tersebut secara struktur memiliki entitas dan atribut yang valid, namun mengandung kondisi yang tidak dapat dievaluasi secara objektif terhadap skema database, sehingga memerlukan pemahaman semantik yang lebih dalam — area di mana *LLM Validator* berpotensi memberikan kontribusi signifikan apabila diaktifkan.

DISKUSI

Sistem deteksi *ill-formed* NLQ berbasis *rule* menunjukkan bahwa pendekatan deterministik masih relevan sebagai mekanisme validasi awal pada pipeline *Text-to-SQL* berbasis LLM, khususnya untuk bahasa Indonesia yang masih memiliki keterbatasan dataset beranotasi. Berbeda dengan penelitian *Text-to-SQL* sebelumnya yang berfokus pada peningkatan kemampuan LLM dalam menghasilkan SQL melalui *schema linking* (Liu et al., 2025), *execution consistency* (Yang et al., 2025), maupun *constrained decoding* (Scholak et al., 2020), penelitian ini menempatkan proses validasi kualitas masukan sebagai tahap prapemrosesan. Pendekatan tersebut penting karena kualitas keluaran LLM sangat dipengaruhi oleh kualitas *Natural Language Query* (NLQ) yang diterima.

Temuan penelitian menunjukkan bahwa sebagian besar kesalahan NLQ dapat direpresentasikan dalam aturan eksplisit sehingga mampu dideteksi tanpa memerlukan proses pelatihan model. Karakteristik ini memberikan keunggulan berupa transparansi, kemudahan interpretasi, dan konsistensi keputusan, yang sulit diperoleh pada pendekatan berbasis LLM murni. Namun demikian, masih terdapat keterbatasan pada kasus yang melibatkan ambiguitas semantik atau makna implisit, karena interpretasi konteks semacam ini memerlukan

kemampuan penalaran yang berada di luar cakupan sistem berbasis aturan. Kondisi tersebut menunjukkan bahwa pendekatan *rule-based* lebih tepat digunakan sebagai mekanisme penyaringan awal (*first-layer validation*), bukan sebagai pengganti kemampuan inferensi semantik LLM.

Implikasi penelitian ini adalah tersedianya arsitektur validasi yang dapat mengurangi masuknya NLQ bermasalah ke dalam proses generasi SQL sehingga berpotensi meningkatkan keandalan sistem *Text-to-SQL* secara keseluruhan. Dengan demikian, kontribusi utama penelitian ini bukan pada peningkatan kemampuan LLM menghasilkan SQL, melainkan pada penyediaan mekanisme validasi masukan yang transparan, ringan secara komputasi, dan mudah dikembangkan sesuai karakteristik bahasa Indonesia maupun skema basis data yang digunakan.

KESIMPULAN

Penelitian ini berhasil mengembangkan sistem deteksi *ill-formed Natural Language Query* (NLQ) berbasis *rule* sebagai tahap pra-proses pada pipeline LLM Text-to-SQL berbahasa Indonesia. Sistem diimplementasikan menggunakan library Experta melalui 15 aturan IF–THEN yang merepresentasikan enam kategori kesalahan, yaitu informasi kurang, multitafsir, syarat tidak lengkap, tidak logis, *out-of-schema reference*, dan *unsafe/non-SELECT*. Selain itu, penelitian ini juga menghasilkan dataset pengujian berisi 202 NLQ berbahasa Indonesia yang dapat digunakan untuk mengevaluasi kinerja sistem deteksi. Hasil evaluasi menunjukkan bahwa pendekatan berbasis *rule* mampu mendeteksi sebagian besar NLQ bermasalah dengan kinerja yang baik, ditunjukkan oleh nilai *Recall* sebesar 89,60%, *Precision* 78,87%, *F1-Score* 83,90%, dan *Accuracy* 78,71%. Temuan ini menunjukkan bahwa pendekatan berbasis aturan layak digunakan sebagai mekanisme penyaringan awal (*pre-filter*) yang bersifat deterministik, transparan, dan tidak memerlukan proses pelatihan data berskala besar. Analisis kesalahan memperlihatkan bahwa *false positive* terutama disebabkan oleh aturan yang masih terlalu ketat dalam menginterpretasikan kondisi implisit, sedangkan *false negative* didominasi oleh kasus yang memerlukan pemahaman semantik dan konteks yang lebih kompleks.

Penelitian ini memberikan kontribusi berupa mekanisme validasi masukan yang dapat melengkapi sistem Text-to-SQL berbasis LLM dengan meningkatkan kualitas NLQ sebelum proses generasi SQL dilakukan. Meskipun demikian, kemampuan sistem masih terbatas pada pola kesalahan yang dapat diformalkan ke dalam aturan eksplisit. Oleh karena itu, penelitian selanjutnya perlu mengevaluasi secara eksperimental pendekatan *hybrid* yang mengombinasikan *rule engine* dengan validator berbasis LLM untuk menangani kasus-kasus

yang melibatkan ambiguitas semantik dan konteks yang lebih kompleks, serta menguji sistem pada dataset yang lebih besar dan lebih beragam.

REKOMENDASI

Berdasarkan hasil penelitian dan keterbatasan yang masih ditemukan, beberapa rekomendasi dapat dipertimbangkan untuk pengembangan penelitian selanjutnya. Pertama, jumlah dan variasi dataset *Natural Language Query* (NLQ) berbahasa Indonesia perlu diperluas, khususnya dengan menambahkan contoh-contoh yang mengandung ambiguitas semantik, makna implisit, dan struktur bahasa yang lebih beragam agar evaluasi sistem lebih merepresentasikan kondisi penggunaan nyata. Kedua, aturan deteksi, terutama R15_NO_CONDITION dan R04_NO_ATTRIBUTE, perlu disempurnakan agar mampu mengakomodasi kondisi filter yang dinyatakan secara implisit sehingga dapat mengurangi jumlah *false positive*. Selain itu, penambahan aturan R16_SUBJECTIVE_ADJECTIVE juga direkomendasikan untuk mendeteksi penggunaan kata sifat evaluatif yang bersifat subjektif, yang pada penelitian ini menjadi salah satu penyebab utama *false negative*.

Pengembangan berikutnya juga disarankan mengombinasikan pendekatan berbasis *rule* dengan model bahasa atau metode pembelajaran mesin yang lebih ringan, seperti IndoBERT, sehingga sistem tidak hanya mampu mendeteksi kesalahan yang bersifat struktural, tetapi juga memahami ambiguitas makna dan konteks bahasa Indonesia. Di samping itu, penelitian lanjutan perlu mengevaluasi detektor pada pipeline *Text-to-SQL* secara utuh menggunakan dataset yang lebih besar dan beragam untuk mengukur pengaruhnya terhadap kualitas SQL yang dihasilkan oleh model LLM melalui metrik Execution Accuracy (EX) dan Exact Match Accuracy (EM).

REFERENSI

Akbulut, O., Cavus, M., Cengiz, M., Allahham, A., & Giaouris, D. (2024). Hybrid Intelligent Control System for Adaptive Microgrid Optimization: Integration of Rule-Based Control and Deep Learning.

Deng, Y., Xia, C. S., & Zhang, S. D. (2022). Large Language Models are Edge-Case Fuzzers: Testing Deep Learning Libraries via FuzzGPT. Proceedings of ACM Conference (Conference'17), 1(1). Association for Computing Machinery.

Ding, Z., & Lin, Y. (n.d.). AmbiSQL: Interactive Ambiguity Detection and Resolution for Text-to-SQL.

Haryono, B. A., & Prasetya, A. (2025). Identifikasi Kuantitas Berbasis Rule Pada Masalah Text-to-SQL. *Jurnal Borneo Informatika dan Teknik Komputer*, *5*(1), 26-39.

Islam, S., Haque, M., Naser, A., & Rezaul, M. (2024). A rule-based machine learning model for financial fraud detection. 14(1), 759–771. https://doi.org/10.11591/ijece.v14i1.pp759-771

Liu, G., Tan, Y., Zhong, R., Xie, Y., Zhao, L., Wang, Q., Hu, B., & Li, Z. (2025). Solid-SQL: Enhanced Schema-linking based In-context Learning for Robust Text-to-SQL. Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025), 9793–9803. https://aclanthology.org/2025.coling-main.654/

Nascimento, E. R., Garc, G. M., Feij, L., Garcia, R. L. S., Paes, L. A. P., & Casanova, M. A. (2024). Text-to-SQL Meets the Real-World. 1(Iceis), 978–989. https://doi.org/10.5220/0012555200003690

Nugraha, D. A., & Kurniawan, T. (2024). Implementasi Natural Language Interface to Database Menggunakan Large Language Model. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *8*(2), 356–365.

Prasetya, A., Sari, Y. K., Iskandar, J., & Ansor, M. K. (2024). Identifikasi jenis operasi data manipulation language berbasis BiLSTM pada kalimat berbahasa Indonesia. 9(4), 2552–2557.

Sari, D. P., Rahman, A., & Wibowo, S. (2023). Penerapan Rule-Based System untuk Deteksi Kesalahan Bahasa Indonesia pada Sistem Cerdas. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, *12*(4), 401–410.

Scholak, T., Schucher, N., & Bahdanau, D. (2021). PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. *In Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 9895-9901).

Scholak, T., Schucher, N., & Bahdanau, D. (2021). *PICARD: Parsing incrementally for constrained auto-regressive decoding from language models*. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9895–9901. https://doi.org/10.18653/v1/2021.emnlp-main.779

Sujon, K. M., Hassan, R., Choi, K., & Samad, A. (2025). Empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models.

Yang, K., Wang, D., & Li, Z. (2018). Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task.

Yang, Y., Wang, Z., Xia, Y., Wei, Z., Ding, H., Piskac, R., Chen, H., & Li, J. (2025). Automated Validating and Fixing of Text-to-SQL Translation with Execution Consistency. Proceedings of the ACM on Management of Data, 3(3/SIGMOD). https://doi.org/10.1145/3725271

Ye, X., Iyer, S., Celikyilmaz, A., & Stoyanov, V. (2023). Complementary Explanations for Effective In-Context Learning. 4469–4484.