JURNAL FASILKOM
Volume 16 No. 2 | Agustus 2026: 344-351

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Pembangunan *Multi-Domain Human-Labelled* Dataset untuk Konversi Bahasa Alami ke SQL pada Bahasa Indonesia

Fiorella Asyfa Firlanda [1], Agung Prasetya [2]
[1,2] Universitas Bhinneka PGRI Tulungagung, Tulungagung, Jawa Timur, 66221, Indonesia
[1]fiorellaasyfa@gmail.com *, [2]agung@ubhi.ac.id

Abstract

This study develops a multi-domain, human-labelled Text-to-SQL dataset in Indonesian to address the absence of such resources for the language. The dataset was constructed through a corpus-based approach comprising four stages: corpus planning, corpus construction, corpus validation, and model evaluation. Database schemas represented as Entity Relationship Diagrams (ERD) were used as the structural foundation for generating pairs of Indonesian natural language questions and SELECT-type SQL queries, all created through manual annotation. Validation was performed through four sequential mechanisms: SQL syntax checking, schema conformity verification, query execution testing, and semantic alignment assessment between questions and SQL queries. The resulting dataset is Spider-compatible in JSON format, covering 40 databases, 27 domains, 1,524 questions, and 1,355 unique SQL queries distributed across four difficulty levels. Preliminary evaluation using SQLNet and TypeSQL baseline models under example split and database split scenarios confirms that the dataset provides a representative and challenging evaluation environment for Indonesian Text-to-SQL experiments, though model performance remains limited on complex queries and previously unseen database schemas. The dataset is publicly available and is intended to support future development and evaluation of Indonesian Text-to-SQL models..

Keywords: *human-labelled dataset, Indonesian language, multi-domain, NLP, text-to-SQL.*

Abstrak

Penelitian ini membangun dataset *Text-to-SQL* berbahasa Indonesia yang bersifat *multi-domain* dan dianotasi secara manual,sebagai respons atas ketiadaan sumber daya serupa untuk bahasa Indonesia. Dataset dikonstruksi melalui pendekatan berbasis korpus yang mencakup empat tahap: perencanaan korpus, konstruksi korpus, validasi korpus, dan evaluasi model. Skema basis data dalam bentuk *Entity Relationship Diagram* (ERD) dijadikan fondasi struktural untuk menghasilkan pasangan pertanyaan bahasa Indonesia dan kueri SQL bertipe *SELECT*, seluruhnya disusun melalui anotasi manual. Validasi dilakukan melalui empat mekanisme berurutan: pemeriksaan sintaks *Structured Query Language* (SQL), verifikasi kesesuaian skema, pengujian eksekusi kueri, dan penilaian keselarasan semantik antara pertanyaan dan kueri SQL. Dataset yang dihasilkan berformat JSON kompatibel dengan standar Spider, mencakup 40 basis data, 27 domain, 1.524 pertanyaan, dan 1.355 kueri SQL unik yang terdistribusi pada empat tingkat kesulitan. Evaluasi awal menggunakan model *baseline SQLNet* dan *TypeSQL* pada skenario *example split* dan *database split* mengonfirmasi bahwa dataset menyediakan lingkungan evaluasi yang representatif dan menantang untuk eksperimen *Text-to-SQL* berbahasa Indonesia, meskipun performa model masih terbatas pada kueri kompleks dan skema basis data yang belum pernah dilihat sebelumnya. Dataset tersedia secara publik dan ditujukan untuk mendukung pengembangan serta evaluasi model *Text-to-SQL* berbahasa Indonesia pada penelitian selanjutnya.

Kata kunci: dataset berlabel manusia, bahasa Indonesia, *multi-domain*, NLP, *text-to-SQL*.

1. Pendahuluan

Perkembangan *Artificial Intelligence* serta pemrosesan bahasa alami *Natural Language Processing* telah menciptakan peluang baru dalam interaksi manusia dengan sistem informasi. Salah satu aplikasi yang berkembang pesat adalah *Natural Language Interface to Databases* (NLIDB), yang memungkinkan pengguna mengakses basis data dengan bahasa sehari-hari tanpa harus menguasai sintaks SQL yang rumit[1]. Pendekatan ini memberikan nilai praktis yang signifikan, terutama bagi pengguna non-teknis seperti staf administrasi, peneliti, atau akademisi yang memerlukan akses data cepat tanpa latar belakang pemrograman. Seiring dengan kemajuan *Large Language Model* (LLM), sistem NLIDB yang berbasis *Text-to-SQL* menjadi semakin relevan dan banyak dieksplorasi [2]. Perkembangan model *Text-to-SQL* sendiri telah melewati beberapa fase, dari pendekatan berbasis aturan, model neural, hingga pemanfaatan *pre-trained language model* yang masing-masing meningkatkan kebutuhan akan dataset yang lebih beragam dan representatif [3].

Text-to-SQL adalah bentuk khusus *semantic parsing* yang mengubah pertanyaan berbahasa alami menjadi perintah SQL yang dapat dijalankan pada basis data relasional[4]. Sistem ini berperan sebagai penghubung antara representasi informal bahasa manusia dengan struktur formal SQL, sehingga pemahaman tentang skema basis data, hubungan antartabel, serta konteks pertanyaan menjadi sangat krusial. Kinerja *Text-to-SQL* sangat dipengaruhi oleh ketersediaan dataset yang representatif, konsisten, dan mencakup variasi struktur kueri yang memadai [5]. Tanpa data berkualitas, model yang dibangun cenderung mengalami *overfitting* pada

P-ISSN : 2089-3353
E-ISSN : 2808-9162

pola tertentu dan gagal menggeneralisasi ke skenario baru.

Kemajuan penelitian *Text-to-SQL* ditandai dengan hadirnya serangkaian *benchmark* yang semakin kompleks. ATIS (*Air Travel Information System*) [6] merupakan salah satu dataset paling awal dalam bidang ini, berisi 5.418 pasangan pertanyaan-SQL pada domain tunggal informasi perjalanan udara. Keterbatasan domain tunggal tersebut kemudian dijawab oleh *GeoQueri*[7], *benchmark* klasik dengan 880 pertanyaan mengenai geografi Amerika Serikat yang diadaptasi ke format SQL. Meskipun berguna sebagai tolok ukur awal, kedua dataset tersebut dianggap terlalu sederhana karena proporsi kueri mudah yang tinggi dan tidak mendukung pengujian generalisasi lintas domain [8]. Perubahan signifikan terjadi ketika *WikiSQL* diperkenalkan dengan 80.654 pasangan pertanyaan-SQL yang tersebar di 24.241 tabel Wikipedia [9]. *WikiSQL* menjadi dataset pertama yang benar-benar menguji model pada skala besar dan cakupan domain yang luas, meskipun semua kueri hanya melibatkan satu tabel sehingga belum mencerminkan kompleksitas sistem informasi dunia nyata [10].

Kemajuan signifikan pada *benchmark* tercapai melalui *Spider* [5], yang hingga kini menjadi referensi utama dalam penelitian *Text-to-SQL* modern. *Spider* menyediakan 10.181 pertanyaan serta 5.693 kueri SQL unik yang meliputi 200 basis data multi-tabel dari 138 domain, dengan anotasi yang dilakukan oleh 11 mahasiswa. Keunikan *Spider* dibandingkan dengan versi sebelumnya terletak pada skema evaluasinya: basis data yang digunakan pada set pelatihan dan set pengujian dijamin tidak tumpang tindih, sehingga model diuji secara nyata pada kemampuan generalisasi, bukan sekadar menghafal pola. Perkembangan *benchmark* terus berlanjut dengan munculnya *benchmark* multibahasa dan lintas domain. *MultiSpider* [11] memperluas Spider ke tujuh bahasa, yakni bahasa Inggris, Jerman, Prancis, Spanyol, Jepang, Mandarin, dan Vietnam, serta mencatat penurunan akurasi absolut sebesar 6,1 % pada bahasa selain Inggris, menandakan bahwa faktor bahasa berperan besar dalam tugas Text-to-SQL. Penurunan ini turut dipengaruhi oleh keterbatasan pendekatan penerjemahan otomatis dari benchmark berbahasa Inggris, yang menurut Dou et al [11] rentan menghasilkan pertanyaan yang tidak alami dan tidak akurat sehingga memerlukan proses penerjemahan berlapis dan validasi manual agar kualitas data tetap terjaga. Hal ini mengindikasikan bahwa anotasi manual langsung dalam bahasa target berpotensi menjadi pendekatan yang lebih tepat dibandingkan sekadar menerjemahkan *benchmark* berbahasa Inggris. Temuan ini menegaskan bahwa *benchmark* multibahasa yang ada belum mencakup seluruh variasi bahasa dunia, termasuk Bahasa Indonesia yang dipakai oleh lebih dari 270 juta orang. Jika dibandingkan dengan Bahasa Inggris, Bahasa Indonesia memiliki karakteristik morfologis yang berbeda, khususnya penggunaan imbuhan (afiksasi) yang produktif dalam pembentukan kata [12] Karakteristik ini menjadi salah satu tantangan utama dalam pengembangan sistem NLP berbahasa Indonesia, termasuk pada tugas-tugas yang menyentuh morfo-sintaksis [13]. Perbedaan ini membuat model yang dilatih pada dataset berbahasa Inggris tidak dapat langsung dipindahkan ke Bahasa Indonesia tanpa penyesuaian mendasar. Situasi ini diperparah oleh keterbatasan sumber daya NLP berbahasa Indonesia yang terstandarisasi. *NusaCrowd* [14] mengumpulkan 137 dataset dan 118 data *loader* standar untuk Bahasa Indonesia serta bahasa daerah, namun di antara semua sumber daya tersebut belum ada dataset *Text-to-SQL* yang bersifat *multi-domain* dan dianotasi secara manual. Contoh anotasi manual untuk NLP Indonesia telah dibuktikan oleh *IndoNLI* [15], yang menunjukkan bahwa anotasi manusia menghasilkan data inferensi bahasa yang lebih dapat diandalkan sebagai standar evaluasi. Hal ini memperkuat argumen bahwa dataset berbahasa Indonesia untuk tugas *Text-to-SQL* sebaiknya dibangun melalui proses anotasi manusia yang terkontrol, bukan sekadar terjemahan otomatis dari *benchmark* berbahasa Inggris. Studi yang lebih luas oleh Klie et al. [16] juga menegaskan bahwa kualitas anotasi manusia secara langsung memengaruhi proses pelatihan dan evaluasi model, sehingga prosedur kontrol kualitas yang terdefinisi jelas menjadi prasyarat utama dalam pembuatan dataset yang dapat dipercaya.

Beberapa studi pendahuluan mengenai *Text-to-SQL* dalam bahasa Indonesia telah dilakukan, namun cakupannya masih sangat terbatas. Prasetya et al. [17] meneliti jenis operasi *Data Manipulation Language* (DML) dengan model *BiLSTM* pada kalimat berbahasa Indonesia, memperlihatkan potensi metode neural *network* dalam memahami perintah SQL berbahasa lokal. Keterbatasan riset semacam ini sejalan dengan pola yang lebih luas pada literatur *Text-to-SQL* secara umum: Qin et al [4] dalam survei komprehensifnya menyoroti bahwa mayoritas penelitian *Text-to-SQL* masih terpusat pada bahasa Inggris, sementara bahasa-bahasa lain masih kurang mendapat perhatian riset yang memadai. Hal ini konsisten dengan temuan *MultiSpider* [11], yang mencatat penurunan akurasi absolut hingga 6,1% ketika *benchmark* diperluas ke bahasa selain Inggris. Walaupun studi Prasetya et al. [17] menunjukkan kemajuan yang signifikan sebagai langkah awal untuk konteks Bahasa Indonesia, penelitian tersebut masih terfokus pada kasus terbatas, satu domain, atau jumlah data yang belum cukup untuk melatih dan mengevaluasi model *Text-to-SQL* secara menyeluruh.

Berdasarkan kajian literatur tersebut, terungkap kesenjangan riset yang penting: belum ada kumpulan data *Text-to-SQL* berbahasa Indonesia yang bersifat multi-domain, beranotasi manual (*human-labelled*), dan sesuai dengan format standar benchmark internasional seperti *Spider* [5]. Kekurangan ini menjadi hambatan utama bagi pengembangan sistem

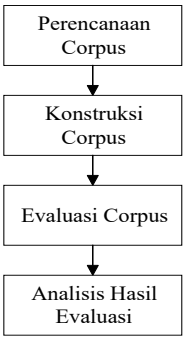
P-ISSN : 2089-3353
E-ISSN : 2808-9162

Text-to-SQL modern berbahasa Indonesia, termasuk pendekatan berbasis LLM, teknik *prompt engineering*, maupun *Retrieval-Augmented Generation* (RAG), yang semuanya memerlukan dataset berkualitas sebagai dasar evaluasi dan pelatihan [3], [4] .

Penelitian ini bertujuan menjawab kesenjangan riset dengan menciptakan dataset *Text-to-SQL* berbahasa Indonesia yang bersifat multi-domain, disusun lewat anotasi manual, dan disajikan dalam format JSON yang mengikuti standar *Spider*. Kontribusi utama penelitian meliputi: (1) penyediaan dataset multi-domain dalam bahasa Indonesia untuk tugas *Text-to-SQL*; (2) pengadaptasian metodologi pembangunan korpus berbasis *Entity-Relationship Diagram* (ERD) yang mengacu pada pendekatan *Spider* [5], dengan proses validasi berlapis meliputi pengecekan sintaks, kecocokan skema, eksekusi kueri, dan konsistensi semantik; serta (3) evaluasi awal menggunakan model *baseline SQLNet* [18] dan *TypeSQL* [5] untuk menguji kelayakan dataset pada skenario *example split* dan *database split*. Diharapkan hasil penelitian ini menjadi dasar bagi pengembangan model *Text-to-SQL* berbahasa Indonesia pada penelitian selanjutnya.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan konstruksi korpus yang diadaptasi dari metodologi *Spider* [5], dengan fokus pada pembangunan dataset *Text-to-SQL* berbahasa Indonesia yang bersifat multi-domain dan dianotasi secara manual. Keseluruhan proses dilaksanakan melalui empat tahap utama yang saling berkaitan: perencanaan korpus, konstruksi korpus, validasi korpus, dan evaluasi model. Diagram alur tahapan penelitian ditampilkan pada Gambar 1.



Gambar 1. Diagram Alur Tahapan Penelitian

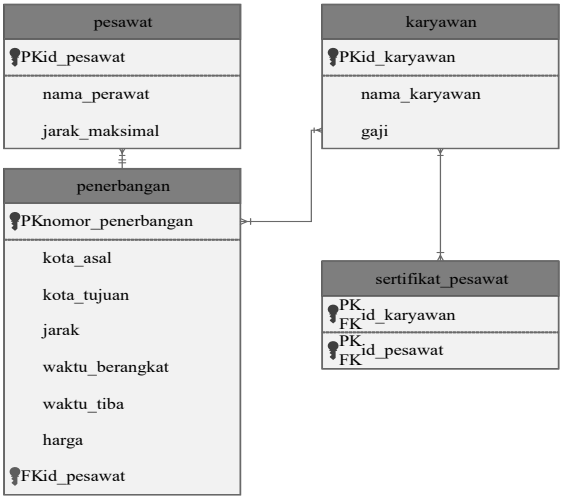
2.1 Perencanaan Korpus

Korpus dirancang untuk mendukung tugas konversi bahasa alami ke SQL dalam Bahasa Indonesia pada skenario multidomain. Setiap entri korpus memuat tiga unsur: kalimat pertanyaan berbahasa Indonesia, pernyataan SQL tipe *SELECT* yang bersesuaian, serta skema basis data yang meliputi nama tabel, nama kolom, tipe data, *primary key*, dan *foreign key*. Data disusun dalam format JSON standar *Spider* [5], yang memungkinkan kompatibilitas langsung dengan skrip evaluasi yang telah tersedia secara publik. Struktur folder korpus mengikuti pola *train.json, dev.json, tables.json,* serta direktori *databases/* yang masing-masing berisi *file.sqlite* dan *schema.sql.*

Korpus dibangun dari 40 basis data yang mencakup 27 domain yang berbeda. Sumber basis data meliputi repositori *GitHub* (27 basis data), platform *Dataedo* (2 basis data), dokumentasi resmi *MySQL* (1 basis data), serta konversi *file CSV* dari *W3Schools* dan sumber publik lain (10 basis data). Semua basis data diubah ke format *SQLite* dan sebagian nama tabel serta kolom disesuaikan ke Bahasa Indonesia untuk menjaga keselarasan dengan pertanyaan yang telah diberi anotasi. Proses anotasi dilakukan oleh satu annotator (*single annotator setting*), yakni peneliti sendiri yang menguasai Bahasa Indonesia serta konsep basis data relasional. Pendekatan ini dipilih untuk mengurangi variasi subjektif antar anotator (*inter-annotator variability)* sehingga gaya bahasa dan pola penulisan kueri SQL tetap seragam di seluruh korpus. Untuk mengurangi risiko bias akibat penggunaan *single annotator*, diterapkan dua langkah mitigasi: (1) jeda waktu selama lebih dari satu minggu antara proses pembuatan pasangan pertanyaan-kueri dan proses validasi ulang, sehingga anotator tidak lagi mengandalkan ingatan terhadap maksud awal saat memvalidasi data; dan (2) validasi dilakukan secara bertahap dan terpisah dari proses pembuatan, mencakup empat mekanisme berurutan (pemeriksaan sintaks SQL, verifikasi kesesuaian skema, pengujian eksekusi kueri, dan penilaian keselarasan semantik), yang berfungsi sebagai bentuk *self-audit* sistematis terhadap konsistensi data.

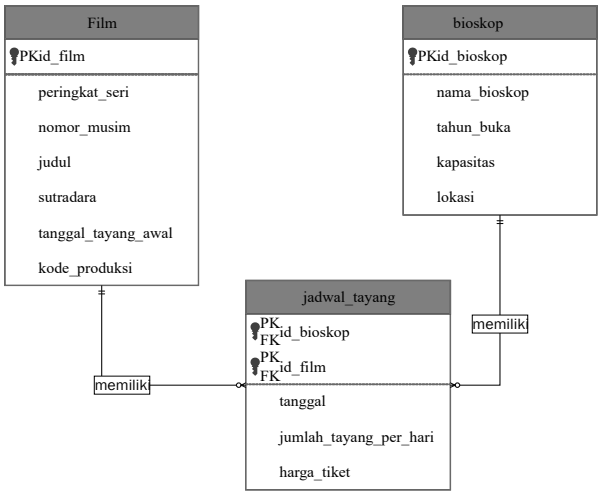
2.2 Konstruksi Korpus

Konstruksi korpus dilakukan melalui lima tahapan berurutan. Langkah pertama, setiap basis data dianalisis strukturnya melalui metadata skema dan kemudian divisualisasikan sebagai *Entity Relationship* Diagram (ERD), seperti yang ditunjukkan pada Gambar 2 dan Gambar 3. ERD berperan sebagai pedoman bagi anotator untuk memahami hubungan antar tabel sebelum membuat kueri SQL yang memakai operasi JOIN.



Gambar 2. Contoh ERD Penerbangan_1

P-ISSN : 2089-3353
E-ISSN : 2808-9162



Gambar 3. Contoh ERD Bioskop

Langkah kedua, merumuskan pertanyaan dalam Bahasa Indonesia dengan memperhatikan dua syarat utama: (1) tidak boleh ambigu, artinya pertanyaan harus secara jelas menunjuk ke atribut yang ada dalam skema, dan (2) tergantung pada skema, sehingga pertanyaan dapat dijawab sepenuhnya menggunakan data dalam basis tanpa memerlukan pengetahuan di luar. Tahap ketiga, setiap pertanyaan diubah menjadi kueri SQL tipe *SELECT* yang memiliki makna semantik yang setara. Langkah keempat, kueri dijalankan secara langsung lewat antarmuka *SQLite* Web untuk memeriksa kebenaran sintaks dan hasilnya. Tahap kelima, dilakukan pengecekan kembali kesesuaian semantik antara pertanyaan dan kueri guna memastikan tidak terdapat perbedaan arti.

Ruang lingkup pola SQL dalam korpus mencakup *SELECT* dengan banyak kolom, fungsi agregasi seperti *COUNT, SUM, AVG, MIN, MAX*, klausa *WHERE* baik dengan kondisi tunggal maupun gabungan, serta perintah *GROUP BY, HAVING, ORDER BY, LIMIT*, operasi *JOIN* antar tabel, serta kueri bersarang dan operator himpunan seperti *UNION*, *INTERSECT, EXCEPT*. Setiap tabel pada masing-masing basis data diusahakan muncul minimal dalam satu kueri demi memastikan representasi skema secara menyeluruh.

2.3 Validasi Korpus

Proses validasi dilakukan secara internal oleh peneliti dengan empat langkah yang diterapkan secara berurutan pada tiap pasangan pertanyaan-kueri: (1) pemeriksaan sintaks SQL, (2) pencocokan nama tabel dan kolom dengan skema basis data, (3) eksekusi kueri pada basis data *SQLite* untuk memastikan output yang benar, dan (4) peninjauan kesesuaian makna antara maksud pertanyaan dengan hasil eksekusi kueri. Pasangan yang tidak lolos pada salah satu langkah akan diperbaiki sebelum dimasukkan ke kumpulan data akhir. Contoh hasil validasi ditampilkan pada Tabel 1.

Tabel 1. Contoh Hasil Validasi

Pertanyaan	Kueri SQL	Status
Tampilkan semua nama depan dan nama belakang mahasiswa	`SELECT nama_depan, nama_belakang FROM mahasiswa;`	Valid
Tampilkan semua nama mahasiswa	`SELECT nama FROM mahasiswa;`	Tidak Valid
Berapa jumlah mahasiswa?	`SELECT COUNT(*) FROM mahasiswa;`	Valid
Tampilkan id mahasiswa yang berusia di atas 20 tahun	`SELECT id_mahasiswa FROM mahasiswa WHERE umur > 20;`	Valid
Tampilkan id mahasiswa yang berusia di atas 20 tahun	`SELECT id_mahasiswa FROM mahasiswa WHERE usia > 20;`	Tidak Valid

Catatan: Kueri tidak valid disebabkan oleh penggunaan nama atribut yang tidak sesuai skema.

2.4 Evaluasi model

Dataset dievaluasi menggunakan dua model baseline Text-to-SQL yang sudah teruji, yaitu *SQLNet* [18] dan *TypeSQL* [5], pada dua skema pemisahan data. Pada skema *example split*, data dibagi berdasarkan pertanyaan sehingga basis data yang sama dapat muncul di set pelatihan dan set pengujian; skema ini menilai kemampuan model dalam menangani variasi linguistik.

Pada skema *database split*, pembagian dilakukan pada tingkat basis data sehingga basis data di set pengujian tidak pernah muncul di set pelatihan; skema ini menguji kemampuan model untuk menggeneralisasi terhadap skema yang belum dikenal. Rancangan pembagian data disajikan pada Tabel 2.

Tabel 2. Rancangan Skenario Pembagian Data Evaluasi

Skenario	Data latih	Data validasi	Data uji
Example split	70,0 % data	10,0 % data	20,0 % data
Database split	70,0 % basis data	10,0 % basis data	20,0 % basis data

Kinerja model dievaluasi dengan dua metrik utama. *Exact Match* menghitung proporsi kueri yang diprediksi identik secara struktural dengan kueri referensi setelah normalisasi. *Component Matching* menilai kesesuaian tiap komponen *SQL SELECT, WHERE, GROUP BY, ORDER BY*, serta kata kunci dan dilaporkan sebagai skor F1, mengikuti protokol evaluasi *Spider* [5]. Selain itu, tingkat kesulitan tiap kueri dikategorikan ke dalam empat level berdasarkan kombinasi tiga komponen: komponen 1 (jumlah klausa struktural seperti *WHERE, JOIN, GROUP BY*), komponen 2 (jumlah kueri bersarang atau operator himpunan), dan lainnya (jumlah kolom SELECT, kondisi *WHERE*, serta fungsi agregasi). Klasifikasi ini menghasilkan empat kategori: mudah, sedang, sulit, dan sangat sulit, sebagaimana diringkas pada Tabel 3.

Tabel 3. Komponen Penentu *Hardness* Pada Dataset

Komponen	**Deskripsi**	**Jenis Kueri**
Component1	Merepresentasikan jumlah komponen struktural dalam kueri SQL	*WHERE, GROUP BY, ORDER BY, LIMIT, JOIN, OR, LIKE*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Component2	Merepresentasikan jumlah nested kueri atau subkueri	Subkueri, *UNION, INTERSECT, EXCEPT*
Others	Merepresentasikan kompleksitas tambahan dalam kueri	Jumlah kolom *SELECT*, jumlah kondisi *WHERE*, jumlah agregasi, jumlah atribut *GROUP BY*

Untuk memberikan gambaran konkret mengenai masing-masing kategori, Tabel 4 menyajikan contoh *kueri* representatif beserta analisis komponen penentu kesulitannya

Tabel 4. Contoh Kueri Per Kategori Kesulitan

Kategori	Contoh Kueri	Analisis Komponen
Mudah	`SELECT nama FROM pelanggan WHERE kota = 'Bandung';`	*component1*=1 (*WHERE*), *component2*=0, *others*=0
Sedang	`SELECT nama, kota FROM pelanggan WHERE umur > 20 ORDER BY nama;`	component1=2 (*WHERE+ORDER BY*), *component2*=0, *others*=1 (multi-kolom)
Sulit	`SELECT id_pelanggan, COUNT(*) FROM transaksi GROUP BY id_pelanggan;`	*component1*=1 (*GROUP BY*), *component2*=0, *others* tinggi (agregasi)
Sangat Sulit	`SELECT AVG(total) FROM pelanggan WHERE id IN (SELECT id FROM transaksi GROUP BY id HAVING COUNT(*) > 3);`	*component1*≥2, *component2*≥1 (*nested*), *others* tinggi

Secara keseluruhan, tahapan perencanaan, konstruksi, dan validasi korpus yang telah diuraikan dirancang untuk menghasilkan dataset yang tidak hanya valid secara sintaksis dan struktural, tetapi juga merepresentasikan variasi tingkat kesulitan kueri secara proporsional. Kombinasi prosedur tersebut dengan skema evaluasi *example split* dan *database split* memungkinkan evalusi dataset dari dua perspektif berbeda, yaitu kemampuan model dalam mengelola variasi bahasa serta kemampuan generalisasinya terhadap skema basis data baru.

3. Hasil dan Pembahasan

Bagian ini menyajikan hasil penelitian yang diperoleh melalui tahapan yang telah dijelaskan pada bagian sebelumnya, mencakup hasil konstruksi dataset, karakteristik dan distribusi data, serta hasil evaluasi model. Hasil disajikan secara berurutan untuk menunjukkan bagaimana dataset yang dibangun dapat dimanfaatkan dalam eksperimen *Text-to-SQL* berbahasa Indonesia, diikuti dengan pembahasan yang menjelaskan makna dan implikasi dari temuan tersebut.

3.1. Hasil Konstruksi Dataset

Penelitian ini menghasilkan dataset *Text-to-SQL* berbahasa Indonesia dalam format JSON yang sesuai dengan standar *Spider*, mencakup 40 basis data, 27 domain, 1.524 pertanyaan berbentuk bahasa alami, dan 1.355 kueri SQL unik. Ringkasan hasil konstruksi dipaparkan pada Tabel 5.

Tabel 5. Ringkasan Hasil Konstruksi Dataset

Komponen	Jumlah
Jumlah Basis Data	40
Jumlah Domain	27
Jumlah Pertanyaan	1.524
Jumlah Kueri SQL unik	1.355
Format Dataset	JSON

Perbedaan antara total pertanyaan (1.524) dan kueri unik (1.355) menunjukkan bahwa sebagian kueri SQL direpresentasikan oleh lebih dari satu variasi pertanyaan bahasa alami. Hal ini umum terjadi pada dataset Text-to-SQL, karena satu kebutuhan informasi dapat diekspresikan lewat berbagai cara penyusunan kalimat. Rata-rata pertanyaan per basis data mencapai 38,1, sementara rata-rata kueri unik per basis data sekitar 33,88, menandakan kontribusi data yang cukup merata di seluruh basis data.

Sampel pasangan pertanyaan dan kueri SQL dari beberapa domain disajikan pada Tabel 6. Sampel tersebut menunjukkan bahwa dataset mencakup ragam kebutuhan informasi, mulai dari kueri agregasi yang sederhana hingga kueri yang memerlukan operasi JOIN serta pengelompokan data.

Tabel 6. Sampel Pasangan Pertanyaan-SQL

Nama basis data	Pertanyaan	Kueri
binaragawan	Berapa jumlah binaragawan yang ada?	`SELECT count(*) FROM atlet_angkat_besi`
bioskop	Di lokasi mana saja terdapat lebih dari satu bioskop dengan kapasitas di atas 300 tempat duduk?	`SELECT lokasi FROM bioskop WHERE kapasitas > 300 GROUP BY lokasi HAVING count(*) > 1`
penerbangan_2	Berapa jumlah penerbangan untuk setiap maskapai?	`SELECT t2.nama_maskapai, COUNT(*) FROM penerbangan t1 JOIN maskapai t2 ON t1.id_maskapai = t2.id_maskapai GROUP BY t2.id_maskapai;`
perguruan_tinggi_1	Sebutkan nama mata kuliah yang diajarkan pada lebih dari satu kelas.	`SELECT t2.deskripsi_mk FROM kelas t1 JOIN mata_kuliah t2 ON t1.kode_mk = t2.kode_mk GROUP BY t2.kode_mk HAVING COUNT(*) > 1;`
perguruan_tinggi_2	Tampilkan nama mahasiswa yang berasal dari departemen Psychology.	`SELECT nama_mahasiswa FROM mahasiswa WHERE nama_departemen = 'Psychology';`

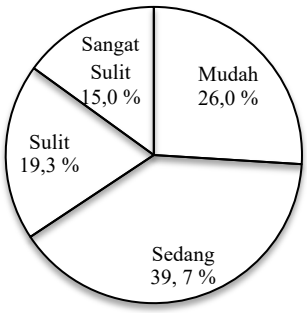
Dataset lengkap tersedia secara publik dan dapat diakses melalui repositori *HuggingFace* pada tautan berikut:

https://huggingface.co/datasets/fiorellasyf/text-to-sql-indonesia/tree/main

P-ISSN : 2089-3353
E-ISSN : 2808-9162

3.2. Karakteristik dan Distribusi Dataset

Distribusi tingkat kesulitan kueri mengindikasikan bahwa dataset tidak didominasi oleh kueri yang mudah. Kategori sedang memiliki jumlah terbanyak, yaitu 605 kueri, diikuti oleh kategori mudah dengan 396 kueri, kategori sulit dengan 294 kueri, serta kategori sangat sulit dengan 229 kueri. Persebaran tingkat kesulitan dataset disajikan pada Gambar 4.



Gambar 4. Persebaran Tingkat Kesulitan Dataset

Distribusi ini mengindikasikan bahwa 34,3 % dari semua kueri termasuk kategori sulit dan sangat sulit. Komposisi tersebut penting untuk memastikan dataset dapat dipakai menguji kemampuan model tidak hanya pada operasi dasar, tetapi juga pada struktur SQL yang rumit, seperti JOIN antar tabel, *nested kueri*, fungsi agregasi majemuk, dan kombinasi beberapa kondisi logis secara bersamaan.

3.3 Hasil Evaluasi Model

Evaluasi menggunakan metrik *Exact Match* sesuai dengan tingkat kesulitan kueri ditampilkan di Tabel 7. Secara umum, *TypeSQL* mencatat tingkat akurasi yang lebih tinggi daripada *SQLNet* di kedua skenario. Pada skenario *example split*, *TypeSQL* meraih akurasi 11,8 % sementara *SQLNet* hanya 5,9 %. Pada skenario database split, *TypeSQL* memperoleh akurasi 7,8 % dan *SQLNet 5*,6 %.

Tabel 7. Akurasi Exact Match Berdasarkan Tingkat Kesulitan Kueri

	Test				
	Mudah	Sedang	Sulit	Sangat Sulit	Semua
		Example Split			
SQLNet	12,3	5,8	1,7	2,1	5,9
TypeSQL	23,1	10,0	6,7	4,3	11,8
		Database Split			
SQLNet	10,3	5,2	1,6	2,2	5,6
TypeSQL	13,6	6,8	6,6	2,2	7,8

Hasil *Component Matching* per komponen SQL disajikan pada Tabel 8. Pada skenario *example split*, skor *TypeSQL* berada di atas *SQLNet* pada komponen *SELECT, GROUP BY*, dan *ORDER B*Y, sedangkan *SQLNet* memperoleh skor lebih tinggi pada komponen *WHERE* dan *keyword*. Pada skenario *database split*, *TypeSQL* tetap unggul pada sebagian besar komponen namun mengalami penurunan signifikan pada komponen *WHERE*.

Tabel 8. Skor F1 dari *Component Matching* Pada Semua Kueri SQL Pada *Test Set*

Model	*Select*	*Where*	Group By	Order By	Keyword
		Example Split			
SQLNet	25,2	12,3	3,4	6,0	60,3
TypeSQL	30,5	9,7	18,8	26,3	52,2
		Database Split			
SQLNet	20,7	9,6	6,1	7,1	53,0
TypeSQL	233	2,4	14,5	23,0	43,8

Nilai *Exact Match* yang masih rendah pada kedua model menunjukkan bahwa *dataset* memiliki kompleksitas yang cukup untuk menguji model *baseline*. Hal ini sejalan dengan sifat dataset multi-domain seperti *Spider*[5], di mana model *baseline* tradisional tidak dirancang untuk mengatasi variasi domain yang luas. *TypeSQL* memiliki kelebihan pada kebanyakan kategori karena teknik *type-aware encoding* memudahkan model mengidentifikasi tipe atribut serta entitas dalam pertanyaan, sehingga lebih tepat dalam menghubungkan frasa ke kolom basis data yang sesuai [5].

Kedua model menunjukkan penurunan kinerja pada skenario *database spl*it dibandingkan *example split*, dan hal ini konsisten di seluruh kategori kesulitan. Pola tersebut mengindikasikan bahwa model masih sangat tergantung pada kemiripan pola data pelatihan dan belum mampu melakukan generalisasi semantik yang sejati ketika dihadapkan pada skema basis data baru. Hal ini sejalan dengan temuan pada *benchmark* multi-domain sebelumnya yang menyatakan bahwa generalisasi lintas skema tetap menjadi tantangan utama bagi model baseline [5], [8].

Pada *dataset multi-domain* berbahasa Indonesia, tantangan tersebut menjadi lebih signifikan karena setiap domain memiliki kosakata entitas yang berbeda contohnya “nama pelanggan” pada satu domain bisa berubah menjadi “nama anggota” atau “nama pengguna” pada domain lain dengan fungsi yang sama. Fenomena ini teramati langsung pada beberapa kasus gagal *SQLNet*. Misalnya, untuk pertanyaan "Siapa nama dosen yang mengajar mata kuliah dengan kode 274?", model memprediksi `select count(*) from mahasiswa where nama_mahasiswa = "value"`, padahal kueri yang benar adalah `SELECT t1.nama_dosen FROM dosen AS t1 JOIN mengajar AS t2 ON t1.id_dosen = t2.id_dosen WHERE t2.kode_mk = '274'` model keliru memetakan entitas "nama" ke kolom `nama_mahasiswa` alih-alih `nama_dosen`. Pola serupa terjadi pada pertanyaan "Pemain mana saja yang pernah mendapatkan kartu kuning?", di mana model memprediksi `SELECT COUNT(*), nama_kampus FROM seleksi_pemain` alih-alih `SELECT nama_pemain FROM pemain WHERE kartu_kuning = 'ya'`. Kasus-kasus ini menegaskan bahwa variasi penamaan entitas antar domain menjadi salah satu sumber kegagalan model dalam menghasilkan kueri yang tepat.

Analisis *Component Matching* memperlihatkan pola yang lebih detail. Bagian *WHERE* menjadi bagian

P-ISSN : 2089-3353
E-ISSN : 2808-9162

paling menantang bagi kedua model, khususnya pada skenario *database split* di mana skor *TypeSQL* menurun tajam dari 9,7 menjadi 2,4. Kesulitan ini disebabkan oleh karakteristik komponen WHERE yang biasanya melibatkan banyak kondisi logika, operator perbandingan, serta nilai yang harus disesuaikan dengan atribut skema baru yang belum dikenali model. Sebaliknya, komponen *keyword* memperoleh nilai tertinggi pada kedua model karena kata kunci SQL bersifat konstan dan tidak terpengaruh oleh variasi skema atau bahasa. Variasi performa antara *TypeSQL* dan *SQLNet* pada komponen *GROUP BY* dan *ORDER BY* menunjukkan bahwa pendekatan *type-aware encoding* efektif untuk kueri yang mencakup pengurutan dan pengelompokan, namun masih kurang optimal untuk kondisi penyaringan yang kompleks.

Karakteristik Bahasa Indonesia juga berkontribusi pada rendahnya akurasi model. Variasi morfologi seperti imbuhan, sinonim, dan fleksibilitas susunan kalimat menghasilkan representasi pertanyaan yang lebih bervariasi dibandingkan dataset berbahasa Inggris. Pertanyaan seperti "film yang paling sering diputar", "film terfavorit", dan "film dengan jumlah penayangan tertinggi" memiliki arti yang sama namun disusun dengan struktur yang berbeda, sehingga model yang dilatih dengan data bahasa Inggris mengalami kesulitan dalam melakukan pemetaan yang konsisten. Temuan ini sejalan dengan laporan *MultiSpider* [11] yang mencatat penurunan akurasi pada bahasa non-Inggris, serta menegaskan kebutuhan akan model yang secara khusus dikembangkan atau disesuaikan untuk Bahasa Indonesia.

Meskipun akurasi masih belum tinggi, hasil evaluasi menunjukkan bahwa dataset yang dibuat dapat menyajikan skenario pengujian yang representatif. Empat kategori kesulitan yang ada memungkinkan analisis kinerja model secara bertahap, sementara pembagian *database split* memberikan penilaian generalisasi yang lebih realistis dibandingkan hanya menguji pada basis data yang sudah familiar.

Meskipun evaluasi memperlihatkan bahwa dataset dipakai untuk eksperimen Text-to-SQL dalam bahasa Indonesia, ada beberapa batasan yang harus dicatat secara objektif. Pertama, jumlah pertanyaan hanya 1.524, yang masih kecil jika dibandingkan dengan standar internasional seperti *Spider* yang mempunyai lebih dari 10.000 pertanyaan, sehingga kemampuan model untuk belajar beragam pola masih dibatasi oleh ukuran data. Kedua, anotasi yang dilakukan oleh satu orang annotator saja membatasi penilaian kesepakatan antar-annotator sebagai ukuran keandalan, walaupun hal ini diusahakan melalui validasi internal berlapis. Ketiga, evaluasi hanya memakai model *baseline* tradisional sehingga belum mampu menunjukkan potensi penuh dataset bila dipadukan dengan model yang lebih canggih. Batasan-batasan ini menjadi dasar bagi penelitian selanjutnya untuk memperluas dan memperkuat sumber daya yang telah dibangun.

4. Kesimpulan

Penelitian ini berhasil membangun dataset *Text-to-SQL* berbahasa Indonesia yang mencakup berbagai domain, diberi anotasi manual, dan sejalan dengan format standar *Spider*. Dataset tersebut meliputi 40 basis data, 27 domain, 1.524 pertanyaan dalam bahasa alami, serta 1.355 kueri SQL unik yang tersebar dalam empat tingkat kesulitan. Metode pembangunan berbasis ERD dengan pemeriksaan berlapis memastikan kumpulan data yang konsisten secara sintaks, struktur, dan makna, sehingga cocok dipakai sebagai bahan pelatihan maupun evaluasi dalam riset *Text-to-SQL* berbahasa Indonesia.

Evaluasi dengan model dasar *SQLNet* dan TypeSQL menunjukkan bahwa kumpulan data ini *dapat* menyajikan tantangan evaluasi yang representatif pada dua skema pembagian data. *TypeSQL* mencatat nilai *Exact Match* 11,8 % pada *example split* dan 7,8 % pada *database split*, sedangkan *SQLNet* memperoleh 5,9 % dan 5,6 % untuk masing-masing skenario. Penurunan kinerja pada skema *database split* menegaskan bahwa dataset ini berhasil memberikan penilaian generalisasi lintas domain yang realistis. Temuan ini sekaligus mengindikasikan bahwa model standar belum cukup fleksibel menghadapi variasi skema basis data baru serta karakteristik bahasa Indonesia. Dataset yang dihasilkan berpotensi menjadi dasar pengembangan antarmuka basis data berbahasa Indonesia yang dapat diakses oleh pengguna non-teknis, sekaligus menjadi tolok ukur lokal untuk mengukur performa beragam model NLP pada tugas Text-to-SQL.

Pada penelitian selanjutnya, disarankan memperluas ukuran kumpulan data dengan menambah jumlah basis data dan domain, memperkaya variasi struktur kueri SQL terutama pada kategori sangat sulit, serta melakukan evaluasi menggunakan model yang lebih modern seperti LLM dengan pendekatan *fine-tuning* atau *prompt engineering.*

Daftar Rujukan

[1] Quamar A., Efthymiou V., Lei C., and Özcan F., 2022. Natural Language Interfaces to Data. Foundations and Trends in Databases, 11(4), pp. 319-414. doi: 10.1561/1900000078.

[2] Hong Z., Yuan Z., Zhang Q., Chen H., and Dong J., 2025. Next-Generation Database Interfaces: A Survey of LLM-based Text-to-SQL. arXiv preprint arXiv:2406.08426, pp. 1-20. doi: 10.48550/arXiv.2406.08426.

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

[4] Qin B., Hui B., Wang L., Yang M., Li J., Li B., Geng R., Cao R., Sun J., Si L., Huang F., and Li Y., 2022. A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions. arXiv preprint arXiv:2208.13629, pp. 1-19. Tersedia di: http://arxiv.org/abs/2208.13629 [Accessed 5 September 2025].

[5] Yu T., Zhang R., Yang K., Yasunaga M., Wang D., Li Z., Ma J., Li I., Yao Q., Roman S., Zhang Z., and Radev D., 2018. Spider: A Large-scale Human-labeled Dataset for Complex and Cross-domain Semantic Parsing and Text-to-SQL Task. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). pp. 3911-3921.

P-ISSN : 2089-3353
E-ISSN : 2808-9162

doi: 10.18653/v1/d18-1425.

[6] Hemphill C. T., Godfrey J. J., and Doddington G. R., 1990. The ATIS Spoken Language Systems Pilot Corpus. In: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania. [Online] Tersedia di: https://aclanthology.org/H90-1021/ [Accessed 5 September 2025].

[7] Zelle J. M., and Mooney R. J., 1996. Learning to Parse Database Queries Using Inductive Logic Programming. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), Vol. 2. Portland, OR, August 1996. AAAI Press, pp. 1050-1055.

[8] Mitsopoulou A., and Koutrika G., 2025. Analysis of Text-to-SQL Benchmarks: Limitations, Challenges and Opportunities. In: Proceedings of the 28th International Conference on Extending Database Technology (EDBT 2025).

[9] Zhong V., Xiong C., and Socher R., 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. arXiv preprint arXiv:1709.00103, pp. 1-12. doi: 10.48550/arXiv.1709.00103.

[10] Katsogiannis-Meimarakis G., and Koutrika G., 2023. A Survey on Deep Learning Approaches for Text-to-SQL. VLDB Journal, 32(4), pp. 905-936. doi: 10.1007/s00778-022-00776-8.

[11] Dou L., Gao Y., Pan M., Wang D., Che W., Zhan D., and Lou J.-G., 2022. MultiSpider: Towards Benchmarking Multilingual Text-to-SQL Semantic Parsing. AAAI Technical Track on Speech and Natural Language Processing, 37(11), pp. 12745-12753. doi: 10.48550/arXiv.2212.13492.

[12] Pisceldo F., Mahendra R., Manurung R., and Arka I. W., 2008. A Two-Level Morphological Analyser for the Indonesian Language.

[13] Koto F., Rahimi A., Lau J. H., and Baldwin T., 2020. IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP. In: Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020). Barcelona, Spain (Online), pp. 757-770. doi: 10.18653/v1/2020.coling-main.66.

[14] Cahyawijaya S., et al., 2023. NusaCrowd: Open Source Initiative for Indonesian NLP Resources. In: Findings of the Association for Computational Linguistics: ACL 2023. Toronto, Canada, pp. 13745-13818.

[15] Mahendra R., Fikri A., Samuel A., Rahman F., and Vania C., 2021. IndoNLI: A Natural Language Inference Dataset for Indonesian. In: Proceedings of the Association for Computational Linguistics, pp. 10511-10527.

[16] Klie J.-C., de Castilho R. E., and Gurevych I., 2024. Analyzing Dataset Annotation Quality Management in the Wild. Computational Linguistics, 50(3). doi: 10.1162/coli_a_00516.

[17] Prasetya A., Sari Y. K., Iskandar J., and Ansor M. K., 2024. Identifikasi Jenis Operasi Data Manipulation Language Berbasis BiLSTM pada Kalimat Berbahasa Indonesia. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 9(4), pp. 2552-2557. doi: 10.29100/jipi.v9i4.8695.

[18] Xu X., Liu C., and Song D., 2017. SQLNet: Generating Structured Queries from Natural Language without Reinforcement Learning. arXiv preprint arXiv:1711.04436, pp. 1-13. doi: 10.48550/arXiv.1711.04436.