HIJM
HORIZON: Indonesian Journal of Multidisciplinary

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

SQL STRUCTURE EMBEDDING BERBASIS TRANSFORMER UNTUK PERANKINGAN FEW-SHOT EXAMPLE PADA SISTEM TEXT-TO-SQL

Argo Yanuar Pamudyo[1], Agung Prasetya[2]
[1, 2]Universitas Bhinneka PGRI, Jl. Mayor Sujadi Timur, Tulungagung, Jawa Timur, Indonesia
Email: argoyepe1207@gmail.com

Article History

Received: 01-07-2026

Revision: 21-07-2026

Accepted: 27-07-2026

Published: 30-07-2026

Abstract. This research develops and evaluates a Transformer encoder-based SQL Structure Embedding (SSE) to improve SQL structure representation quality in Text-to-SQL systems. Static embedding approaches such as GloVe, commonly used in few-shot learning-based Large Language Model (LLM) systems, have limited ability to capture inter-clause contextual relationships in SQL, thereby reducing the quality of few-shot example selection. The dataset used is the BIRD dev set with 1,534 Natural Language Query (NLQ) and SQL query pairs, annotated with five SQL latent intent categories and 44 SQL component labels. The SSE model is built using an encoder-only Transformer architecture initialized with 100-dimensional pretrained GloVe weights, two encoder layers with 4 attention heads, and mean pooling. Evaluation results show the SSE model achieves a Micro F1 of 0.8671, Macro F1 of 0.7920, and Samples F1 of 0.8312, improving by 9.17%, 10.31%, and 9.20% respectively over the GloVe baseline. The combination of SSE with a downstream classifier yields the best performance with a Micro F1 of 0.8821. These results confirm that the self-attention mechanism in Transformer more effectively captures relational context between SQL clauses to support few-shot example ranking in Text-to-SQL systems.

Keywords: SQL Structure Embedding; Transformer; Text-to-SQL; Few-Shot Learning; Multi-Label Classification.

Abstrak. Penelitian ini mengembangkan dan mengevaluasi *SQL Structure Embedding* (SSE) berbasis *Transformer encoder* untuk meningkatkan kualitas representasi struktur SQL pada sistem *Text-to-SQL*. Penelitian ini dilatarbelakangi oleh keterbatasan *embedding* statis, seperti GloVe, dalam merepresentasikan hubungan antarklausa SQL secara kontekstual, yang berdampak pada kualitas seleksi *few-shot example* pada sistem berbasis *Large Language Model* (LLM). Penelitian menggunakan 1.534 pasangan *Natural Language Query* (NLQ) dan *query SQL* dari *BIRD dev set* yang dianotasi berdasarkan kategori *latent intent SQL* dan komponen SQL. Model SSE dikembangkan dan dibandingkan dengan *baseline* GloVe menggunakan metrik Micro F1, Macro F1, dan Samples F1. Hasil penelitian menunjukkan bahwa SSE meningkatkan kualitas representasi struktur SQL dengan pencapaian Micro F1 sebesar 0,8671, Macro F1 sebesar 0,7920, dan Samples F1 sebesar 0,8312, masing-masing meningkat lebih dari 9% dibandingkan baseline GloVe. Performa terbaik diperoleh ketika SSE dikombinasikan dengan *downstream classifier*, dengan Micro F1 sebesar 0,8821. Temuan ini menunjukkan bahwa representasi berbasis Transformer lebih efektif dalam menangkap hubungan kontekstual antarkomponen SQL dan berkontribusi terhadap peningkatan kualitas perankingan contoh *few-shot* pada sistem *Text-to-SQL*.

Kata Kunci: *SQL Structure Embedding*; Transformer; *Text-to-SQL*; *Few-Shot Learning*; *Multi-Label Classification.*

How to Cite: Pamudyo, A. Y & Prasetya, A. (2026). *SQL Structure Embedding* Berbasis Transformer untuk Perankingan *Few-Shot Example* pada *Sistem Text-To-SQL*. *HORIZON: Indonesian Journal of Multidisciplinary*, 4 (4), 5030-5038. http://doi.org/10.54373/hijm.v4i4.6834

PENDAHULUAN

Perkembangan kecerdasan buatan dalam bidang pemrosesan bahasa alami (*Natural Language Processing*/NLP) mendorong pemanfaatan *Large Language Model* (LLM) dalam sistem *Text-to-SQL*, yaitu sistem yang menerjemahkan permintaan informasi dalam bahasa alami menjadi *query SQL* untuk mengakses basis data relasional (Liu et al., 2024; Prasetya et al., 2024). Dalam perkembangan tersebut, Transformer dengan mekanisme *self-attention* menjadi arsitektur penting karena mampu menangkap hubungan kontekstual dan dependensi antarelemen secara lebih efektif (Vaswani et al., 2017). Salah satu komponen penting dalam sistem *Text-to-SQL* berbasis LLM adalah representasi embedding yang digunakan untuk mengubah pertanyaan bahasa alami dan *query SQL* ke dalam representasi vektor. Kualitas embedding berpengaruh terhadap proses retrieval dan perankingan, contoh pada skema *few-shot learning*, yang selanjutnya dapat memengaruhi kualitas konteks yang diberikan kepada LLM (Sun et al., 2023; Nan et al., 2023).

Penelitian terdahulu telah mengembangkan beberapa pendekatan untuk meningkatkan relevansi contoh few-shot. Pendekatan pertama menggunakan embedding semantik umum untuk mengukur kemiripan antara pertanyaan dan contoh. Pendekatan kedua memanfaatkan representasi struktural, seperti *Abstract Syntax Tree* (AST), untuk menangkap struktur sintaksis query SQL (Ji & Pan, 2024). Pendekatan lainnya menggunakan schema link graph untuk memperkuat hubungan antara pertanyaan, skema basis data, dan elemen query dalam proses *in-context learning* (Lee et al., 2025). Meskipun pendekatan tersebut memperkaya representasi semantik atau struktural, masih terdapat keterbatasan dalam mengintegrasikan informasi struktur SQL ke dalam satu representasi embedding yang kontekstual dan dapat digunakan secara langsung untuk perankingan contoh. Sebaliknya, embedding statis seperti GloVe dan Word2Vec merepresentasikan token secara relatif tetap sehingga belum mampu menangkap perubahan makna dan hubungan antarkomponen SQL berdasarkan konteks klausa dalam query.

Berdasarkan pemetaan tersebut, penelitian ini mengembangkan *SQL Structure Embedding* (SSE) berbasis *Transformer encoder*. Perbedaan utama SSE dibandingkan dengan pendekatan terdahulu terletak pada fokusnya terhadap representasi struktur SQL secara kontekstual pada tingkat *encoder*. SSE tidak hanya mengukur kemiripan semantik antara pertanyaan dan *query*, serta tidak merepresentasikan struktur SQL sebagai struktur sintaksis eksternal seperti AST atau *graph*, tetapi memanfaatkan mekanisme *self-attention* untuk mempelajari hubungan antarkomponen SQL dalam satu representasi *embedding*. Dengan demikian, kebaruan penelitian ini terletak pada pengembangan *embedding* khusus struktur SQL yang

mengintegrasikan informasi semantik *Natural Language Query* (NLQ) dan pola struktural query melalui representasi Transformer untuk mendukung perankingan *few-shot example*. Penelitian ini tidak membangun sistem *Text-to-SQL* secara *end-to-end*, tetapi berfokus pada peningkatan kualitas komponen representasi yang menjadi dasar proses retrieval dan seleksi contoh.

Tujuan penelitian ini adalah: (1) menyusun dataset berbasis BIRD untuk menguji kemampuan representasi struktur SQL; (2) mengembangkan dan mengimplementasikan SSE berbasis *Transformer encoder*; (3) mengevaluasi kinerja SSE menggunakan metrik Micro F1, Macro F1, dan Samples F1; serta (4) membandingkan kinerja SSE dengan *baseline embedding* statis GloVe untuk mengetahui efektivitas representasi kontekstual dalam menangkap struktur SQL.

METODE

Dataset *Text-to-SQL*

Penelitian ini menggunakan dataset BIRD (*Benchmarking Intermediate Reasoning for Database*) *dev set* yang terdiri dari 1.534 pasangan *Natural Language Query* (NLQ) dan *query SQL ground truth* berbahasa Inggris dengan cakupan *Data Query Language* (DQL) (Hui et al., 2023). Dataset dipilih karena menyediakan variasi pola struktur *SQL* yang representatif untuk pengujian SSE. Setiap *query* dianotasi dalam dua format label *multi-label*, yaitu lima kategori *latent intent SQL* (*aggregation*, *filtering*, *relational*, *ranking*, *grouping*) dan 44 label komponen *SQL* terperinci yang mencakup klausa dasar (SELECT, FROM, WHERE), klausa relasional (JOIN, INNER JOIN, LEFT JOIN), klausa agregasi (COUNT, SUM, AVG, MAX, MIN), serta operator dan fungsi *SQL* lainnya. Anotasi dilakukan secara otomatis melalui *parsing SQL* dan diverifikasi secara terbatas untuk menjaga konsistensi label.

Tahap pra-pemrosesan meliputi *lowercasing*, penghapusan tanda baca menggunakan ekspresi reguler, normalisasi spasi berulang, serta tokenisasi berbasis *whitespace*. *Vocabulary* dibangun dengan menyaring berkas *GloVe* 6B berdimensi 100 berdasarkan kata yang muncul dalam dataset, menghasilkan 2.090 kata yang ditemukan dalam *GloVe* sehingga total *vocabulary* menjadi 2.092 token setelah penambahan token <PAD> dan <UNK>. Dataset dibagi dengan rasio 80:20 menggunakan *seed* tetap (SEED=42), menghasilkan 1.227 data latih dan 307 data validasi.

Tabel 1. Distribusi label utama dataset

Komponen SQL	Kategori	Jumlah	Persentase
SELECT	Klausa Dasar	1.534	100%
FROM	Klausa Dasar	1.534	100%
WHERE	Filtering	1.385	90,3%
JOIN / INNER JOIN	Relational	1.140	74,3%
COUNT	Agregasi	622	40,6%
ORDER BY	Ranking	305	19,9%
GROUP BY	Grouping	120	7,8%

Berdasarkan Tabel 1, distribusi label memperlihatkan ketidakseimbangan kelas (*class imbalance*) yang signifikan; label SELECT dan FROM muncul pada seluruh data, sedangkan GROUP BY hanya muncul pada 7,8% data. Kondisi ini merupakan karakteristik alami dataset *Text-to-SQL* yang mencerminkan pola penggunaan *SQL* pada dunia nyata, di mana *query* sederhana jauh lebih banyak dibandingkan *query* dengan agregasi kompleks.

Perancangan *SQL Structure Embedding* Berbasis Transformer

Model SSE dirancang menggunakan arsitektur *Transformer encoder-only* yang diinisialisasi dengan bobot *embedding GloVe* 6B 100 dimensi yang dibekukan (*frozen*). Arsitektur terdiri atas lapisan *embedding* dengan *positional encoding* sinusoidal (Vaswani, 2017), dua lapisan *Transformer encoder* dengan mekanisme *multi-head self-attention* (4 *attention head*), *feed-forward network* berdimensi 256, serta *mean pooling* untuk menghasilkan representasi global berdimensi 100 dari setiap *query*. Sebagai pembanding, disusun model *baseline GloVe* yang hanya menggunakan rata-rata *embedding* token (*average pooling*) tanpa mekanisme *self-attention.*

Tabel 2. Arsitektur dan konfigurasi model

Komponen	Model Encoder (SSE)	Baseline GloVe
Embedding	GloVe 6B 100d (freeze)	GloVe 6B 100d (freeze)
Positional Encoding	Sinusoidal	—
Encoder Layer	Transformer (2 layer)	Mean pooling langsung
Attention head	4 head	—
Feed-Forward Dim	256	—
d_model	100	100
Dropout	0,1	0,1
Optimizer	Adam (lr=1e-3)	Adam (lr=1e-3)
Loss Function	BCEWithLogitsLoss	BCEWithLogitsLoss
Epoch / Batch Size	10 / 32	10 / 32

Encoder SSE dilatih pada tugas klasifikasi lima *latent intent SQL* menggunakan fungsi *loss* BCEWithLogitsLoss dan *optimizer* Adam (*learning rate* 1e-3) selama 10 *epoch* dengan *batch size* 32. Setelah pelatihan, seluruh parameter *encoder* dibekukan dan *embedding* yang

dihasilkan digunakan sebagai fitur tetap untuk melatih *downstream classifier* berupa jaringan linear sederhana yang memetakan vektor *embedding* berdimensi 100 ke 44 label komponen *SQL*. Pendekatan dua tahap ini memungkinkan evaluasi kualitas *embedding* secara independen dari performa *classifier downstream*.

Evaluasi Kinerja

Kinerja model dievaluasi menggunakan tiga metrik klasifikasi *multi-label*, yaitu *Micro F1*, *Macro F1*, dan *Samples F1*. *Micro F1* mengukur performa secara agregat berdasarkan total *true positive*, *false positive*, dan *false negative* seluruh label sehingga memberi bobot lebih besar pada kelas mayoritas. *Macro F1* menghitung rata-rata tidak tertimbang F1-*score* per label sehingga lebih sensitif terhadap kelas minoritas. *Samples F1* menghitung rata-rata F1-*score* per sampel sehingga relevan untuk evaluasi pada level instans *multi-label*. Ketiga metrik dirumuskan sebagai:

$$Micro\ F1 = 2 \times (Micro\ Precision \times Micro\ Recall) / (Micro\ Precision + Micro\ Recall) \quad ...(1)$$

$$Macro\ F1 = (1/N)\ \Sigma\ F1i\ ,\ untuk\ i = 1\ sampai\ N \quad ...(2)$$

$$Samples\ F1 = (1/|S|)\ \Sigma\ (2 \times Precisions \times Recalls) / (Precisions + Recalls) \quad ...(3)$$

Evaluasi dilakukan pada data validasi dengan *threshold* 0,5 untuk mengonversi probabilitas *sigmoid* menjadi prediksi biner pada setiap label

HASIL

Hasil Pelatihan Model

Model *encoder* SSE dilatih selama 10 *epoch* dan mencapai *validation loss* terbaik sebesar 0,2987 pada *epoch* ke-5, sebelum mengalami *overfitting* ringan pada *epoch* berikutnya. Sebagai pembanding, model *baseline GloVe* menunjukkan konvergensi yang lebih lambat dengan *validation loss* terbaik sebesar 0,3531 pada *epoch* ke-8. *Downstream classifier* yang dilatih menggunakan *embedding frozen* dari *encoder* SSE mencapai *validation loss* terbaik sebesar 0,2193 pada *epoch* ke-6. Perbedaan *validation loss* ini mengindikasikan bahwa mekanisme *self-attention* pada *Transformer* membentuk representasi yang lebih baik dibandingkan rata-rata *embedding* statis dalam meminimalkan kesalahan prediksi pada data yang tidak dilihat selama pelatihan.

Hasil Evaluasi Model

Rangkuman perbandingan performa ketiga model yang dievaluasi, yaitu *baseline GloVe*, *encoder* SSE berbasis *Transformer*, dan *downstream SQL component classifier* yang memanfaatkan *embedding* SSE adalah sebagai berikut.

Tabel 3. Perbandingan performa keseluruhan model

Metrik	Baseline GloVe	SSE Encoder	SSE + Downstream
Micro F1	0,7943	0,8671	0,8821
Macro F1	0,7180	0,7920	0,8100
Samples F1	0,7612	0,8312	0,8487
Val Loss (terbaik)	0,3531	0,2987	0,2193
Epoch terbaik	8	5	6
Peningkatan Micro F1 vs Baseline	—	+9,17%	+8,78%*

Catatan: persentase peningkatan dihitung relatif terhadap baseline GloVe.

Model *encoder* SSE mencapai Micro F1 sebesar 0,8671, Macro F1 sebesar 0,7920, dan Samples F1 sebesar 0,8312. Dibandingkan dengan baseline GloVe, performa tersebut meningkat masing-masing sebesar 9,17%, 10,31%, dan 9,20%. Kombinasi SSE dengan *downstream classifier* menghasilkan performa tertinggi dengan Micro F1 sebesar 0,8821, Macro F1 sebesar 0,8100, dan Samples F1 sebesar 0,8487. Pada tingkat per label, performa terbaik encoder SSE dicapai pada label *filtering* (F1 = 0,97), sedangkan performa terendah terdapat pada label *grouping* (F1 = 0,60), yang memiliki jumlah sampel paling terbatas. Hasil perbandingan tersebut disajikan pada Tabel 4.

Tabel 4. Perbandingan F1-Score per Label: Baseline GloVe vs Encoder SSE

Label	F1 Baseline GloVe	F1 SSE Encoder	Support
aggregation	0,71	0,80	808
filtering	0,95	0,97	1.496
relational	0,80	0,87	1.140
ranking	0,63	0,72	310
grouping	0,50	0,60	120

Secara keseluruhan, hasil penelitian menunjukkan bahwa SSE berbasis Transformer mampu menghasilkan representasi struktur SQL yang lebih efektif dibandingkan embedding statis GloVe. Peningkatan performa pada seluruh metrik evaluasi menunjukkan bahwa mekanisme *self-attention* memberikan kontribusi terhadap kemampuan model dalam menangkap hubungan kontekstual antarkomponen SQL. Performa terbaik yang diperoleh melalui kombinasi SSE dan *downstream classifier* juga menunjukkan bahwa embedding struktur SQL yang dihasilkan tidak hanya efektif sebagai representasi, tetapi dapat memberikan

kontribusi lebih lanjut pada proses klasifikasi komponen SQL. Dengan demikian, kontribusi utama penelitian ini terletak pada pengembangan representasi embedding khusus struktur SQL yang dapat meningkatkan kualitas pemodelan dan berpotensi mendukung proses retrieval serta perankingan *few-shot example* pada sistem Text-to-SQL berbasis LLM.

DISKUSI

Peningkatan performa encoder SSE dibandingkan baseline GloVe terutama menunjukkan bahwa representasi kontekstual lebih efektif dalam menangkap hubungan antarkomponen pada Natural Language Query (NLQ) dan struktur SQL. Mekanisme *self-attention* memungkinkan model mempertimbangkan konteks antartoken secara lebih menyeluruh, sehingga informasi seperti *how many* yang mengindikasikan *aggregation* atau *for each* yang berkaitan dengan *grouping* tidak dipahami secara terpisah dari konteks kalimat. Temuan ini sejalan dengan penelitian Berkeley et al. (2018), Qi et al. (2021), dan Zhang et al. (2023) yang menunjukkan bahwa representasi berbasis Transformer dapat memperkuat pemodelan hubungan semantik dan struktural dalam tugas Text-to-SQL. Dengan demikian, keunggulan SSE dibandingkan embedding statis menunjukkan bahwa representasi struktur SQL yang lebih kontekstual berpotensi menghasilkan pengukuran kemiripan yang lebih informatif dalam proses seleksi dan perankingan *few-shot example*.

Performa SSE + *downstream classifier* yang lebih tinggi dibandingkan encoder SSE menunjukkan bahwa representasi yang dihasilkan mampu ditransfer dari tugas klasifikasi lima kategori *latent intent* ke tugas yang lebih spesifik, yaitu klasifikasi 44 komponen SQL. Hal ini mengindikasikan bahwa embedding SSE menangkap informasi struktural yang cukup kaya untuk mendukung berbagai tugas lanjutan dalam domain Text-to-SQL. Namun, performa yang lebih rendah pada kelas minoritas, khususnya *grouping* dan *ranking*, menunjukkan bahwa ketidakseimbangan distribusi kelas masih membatasi kemampuan generalisasi model. Meskipun penggunaan *BCEWithLogitsLoss* membantu mengurangi dampak ketidakseimbangan pada proses pelatihan, distribusi data yang tidak merata tetap menjadi tantangan.

Secara keseluruhan, hasil penelitian menunjukkan bahwa peningkatan kualitas representasi SQL melalui SSE memiliki implikasi penting terhadap potensi peningkatan proses retrieval, seleksi, dan perankingan *few-shot example* karena contoh dapat dibandingkan berdasarkan representasi semantik dan struktural yang lebih kontekstual. Namun, implikasi tersebut masih bersifat potensial karena penelitian ini berfokus pada evaluasi kualitas embedding dan klasifikasi komponen SQL, bukan pengujian performa sistem Text-to-SQL

secara *end-to-end*. Oleh karena itu, penelitian selanjutnya perlu mengintegrasikan SSE ke dalam pipeline retrieval dan menguji secara langsung pengaruhnya terhadap kualitas seleksi *few-shot example* serta akurasi query SQL yang dihasilkan oleh LLM.

KESIMPULAN

Penelitian ini berhasil menyusun dataset *Text-to-SQL* berbasis BIRD *dev set* yang terdiri dari 1.534 entri data dengan dua format anotasi *multi-label* (lima *latent intent* dan 44 komponen *SQL*), serta merancang dan mengimplementasikan *SQL Structure Embedding* (SSE) berbasis *Transformer encoder* menggunakan PyTorch. Evaluasi menunjukkan bahwa *encoder* SSE mengungguli *baseline GloVe* pada seluruh metrik, dengan peningkatan *Micro F1*, *Macro F1*, dan *Samples F1* masing-masing sebesar 9,17%, 10,31%, dan 9,20%. Kombinasi SSE dengan *downstream classifier* menghasilkan performa terbaik dengan *Micro F1* sebesar 0,8821. Hasil ini membuktikan bahwa mekanisme *self-attention* pada *Transformer* mampu menangkap konteks relasional antarklausa *SQL* secara lebih efektif dibandingkan pendekatan *embedding* statis, sehingga berpotensi mendukung perankingan *few-shot example* yang lebih relevan dalam sistem *Text-to-SQL* berbasis LLM.

Penelitian selanjutnya disarankan untuk meningkatkan kapasitas arsitektur *Transformer encoder*, menerapkan strategi penanganan *class imbalance* yang lebih sistematis (misalnya *weighted loss* atau *focal loss*), memperluas penelitian ke konteks bahasa Indonesia, mengintegrasikan SSE ke dalam *pipeline Text-to-SQL* secara *end-to-end*, serta mengeksplorasi strategi *contrastive learning* secara eksplisit (misalnya *TripletLoss* atau *InfoNCE*) untuk menghasilkan *embedding* yang lebih terstruktur dalam ruang vektor

REFERENSI

Berkeley, U. C., Polozov, O., & Richardson, M. (2018). RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers.

Dewi, B. E. S. (2025). Pengukuran Kemiripan Kalimat Bahasa Indonesia Menggunakan Representasi Word Embedding FastText. Jurnal Teknologi Informasi Dan Digital (TRIDI), 2(2), 20–29.

Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Geng, R., Huo, N., Zhou, X., Ma, C., Li, G., Chang, K. C. C., Huang, F., Cheng, R., & Li, Y. (2023). Can LLM Already Serve as A Database Interface? A Big Bench for Large-Scale Database Grounded. NeurIPS, 1–28.

Ji, Y., & Pan, J. Z. (2024). Improving Retrieval-augmented Text-to-SQL with AST-based Ranking and Schema Pruning.

Lee, J., Lee, J. S., Lee, J., Choi, Y. S., & Lee, J. H. (2025). DCG-SQL: Enhancing In-Context Learning for Text-to-SQL with Deep Contextual Schema Link Graph. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1, 15397–15412. https://doi.org/10.18653/v1/2025.acl-long.748

Liang Shi, Zhengju Tang, Nan Zhang, Xiaotong Zhang, Z. Y. (2025). A Survey on Employing Large Language Models for Text-to-SQL Tasks. ACM Computing Surveys, 1(1), 1–36. https://doi.org/10.1145/3737873

Liu, G., Tan, Y., Zhong, R., Xie, Y., Zhao, L., Wang, Q., Hu, B., & Li, Z. (2024). Solid-SQL: Enhanced Schema-linking based In-context Learning for Robust Text-to-SQL. http://arxiv.org/abs/2412.12522

Nan, L., Zhao, Y., Zou, W., Ri, N., Tae, J., Zhang, E., Cohan, A., & Radev, D. (2023). Enhancing Text-to-SQL Capabilities of Large Language Models: A Study on Prompt Design Strategies. Findings of the Association for Computational Linguistics: EMNLP 2023, Icl, 14935–14956. https://doi.org/10.18653/v1/2023.findings-emnlp.996

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.

Qi, J., Tang, J., He, Z., Wan, X., Cheng, Y., Zhou, C., Wang, X., Zhang, Q., & Lin, Z. (2021). RASAT: Integrating Relational Structures into Pretrained Seq2Seq Model for Text-to-SQL.

Sun, R., Arik, S. Ö., Nakhost, H., Dai, H., Sinha, R., Yin, P., & Pfister, T. (2023). SQL-PaLM: Improved Large Language Model Adaptation for Text-to-SQL. arXiv:2306.00739.

Vaswani, A. (2017). Attention Is All You Need. Nips.

Zhang, K., Lin, X., Wang, Y., Cen, J., Jiang, X., Tan, H., & Shen, H. (2023). ReFSQL: A Retrieval-Augmentation Framework for Text-to-SQL Generation. 664–673