Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi
Vol. 3, No. 2, Juli 2024
ISSN: 2809-9834, DOI: 10.31284/p.semtik.2024-2.6211

Optimisasi Performa Akses Data dalam Grafana Menggunakan Indeks B-Tree MySQL

Muhammad Samsul Anwar[1], Nanang Fakhrur Rozi[2]

Institut Teknologi Adhi Tama Surabaya

E-mail: samsul.anw98@gmail.com

ABSTRACT

In today's modern digital era, data become a crucial asset for companies and organizations, facilitating strategic decision-making, predictive analysis, and operational reporting. Grafana, an open-source real-time data visualization and monitoring platform, integrates seamlessly with various data sources, such as MySQL. The performance of Grafana heavily relies on efficient data access, which often poses a challenge when using MySQL for large-scale databases. One of the techniques to enhance data access performance is indexing, where B-Tree indexing emerges as one of the most commonly used methods in MySQL due to its efficiency in data searching and sorting. This study aims to evaluate the impact of B-Tree indexing on data access performance in MySQL used within Grafana. Testing was conducted by comparing query execution times between indexed and non-indexed tables. The results show that B-Tree indexing usage significantly reduces total query time while maintaining data storage efficiency. These findings underscore the importance of indexing in enhancing database system performance, particularly in real-time applications like Grafana. The study recommends further research to develop more advanced indexing techniques to address performance challenges as data scales. Thus, this research contributes to a deeper understanding of optimizing database performance for real-time monitoring and analysis.

Kata Kunci

Basis Data;
Grafana;
Indeks B-Tree
MySQL;

ABSTRAK

Di era digital saat ini, data menjadi aset krusial bagi perusahaan dan organisasi untuk pengambilan keputusan strategis, analisis prediktif, dan pelaporan operasional. Grafana, sebagai platform open-source untuk visualisasi dan monitoring data real-time, memungkinkan integrasi dengan berbagai sumber data seperti MySQL. Performa Grafana sangat tergantung pada efisiensi akses data, yang sering kali menjadi tantangan utama ketika menggunakan MySQL untuk basis data yang besar. Salah satu teknik untuk meningkatkan performa akses data adalah pengindeksan, di mana indeks B-Tree menjadi salah satu yang paling umum digunakan dalam MySQL karena efisiensinya dalam pencarian dan pengurutan data. Penelitian ini bertujuan untuk mengevaluasi dampak pengindeksan B-Tree terhadap performa akses data di MySQL yang digunakan dalam Grafana. Pengujian dilakukan dengan membandingkan waktu eksekusi query antara tabel yang diindeks dan tidak diindeks. Hasil pengujian menunjukkan bahwa penggunaan indeks B-Tree secara signifikan mengurangi waktu total query, sementara mempertahankan efisiensi penyimpanan data. Temuan ini menegaskan pentingnya pengindeksan dalam meningkatkan kinerja sistem basis data, khususnya dalam konteks aplikasi real-time seperti Grafana. Studi ini merekomendasikan penelitian lebih lanjut untuk mengembangkan teknik pengindeksan yang lebih canggih guna mengatasi tantangan performa saat data semakin besar. Dengan demikian, penelitian ini memberikan kontribusi untuk pemahaman lebih lanjut dalam optimalisasi kinerja basis data dalam konteks monitoring dan analisis real-time.

PENDAHULUAN

Di era digital modern, data merupakan salah satu aset terpenting bagi perusahaan dan organisasi. Data digunakan untuk berbagai tujuan, mulai dari pengambilan keputusan strategis hingga analisis prediktif dan pelaporan operasional. Seiring dengan meningkatnya volume dan kompleksitas data, kebutuhan akan akses data yang cepat dan efisien juga semakin meningkat. Salah satu platform yang banyak digunakan untuk memantau data secara real-time adalah Grafana.

Grafana adalah platform open-source yang menyediakan fitur visualisasi dan pemantauan untuk berbagai jenis data. Dengan kemampuannya untuk mengintegrasikan berbagai sumber data, seperti MySQL, InfluxDB, dan Elasticsearch, Grafana menjadi pilihan populer bagi banyak organisasi untuk membangun dashboard interaktif dan informatif. Namun, performa platform ini sangat bergantung pada kecepatan dan efisiensi akses data dari sumber-sumber tersebut. MySQL adalah salah satu sistem manajemen basis data relasional (RDBMS) yang paling banyak digunakan di dunia. Popularitasnya berasal dari kemampuannya untuk menangani volume data yang besar dan mendukung berbagai aplikasi, mulai dari situs web e-commerce hingga sistem manajemen konten. Meski demikian, performa MySQL dalam hal kecepatan akses data dapat bervariasi tergantung pada berbagai faktor, termasuk struktur tabel, ukuran dataset, dan teknik optimasi yang digunakan[1].

Masalah utama yang dihadapi ketika menggunakan MySQL dalam platform monitoring seperti Grafana adalah waktu yang lama untuk load data dari tabel yang berukuran besar. Dalam aplikasi real-time, seperti monitoring sistem, keterlambatan dalam akses data dapat menyebabkan penundaan dalam pengambilan keputusan dan respon yang tidak efektif terhadap insiden yang terjadi.Salah satu teknik optimasi yang umum digunakan untuk meningkatkan performa query dalam MySQL adalah indexing. Indexing adalah proses membuat struktur data tambahan yang menyimpan salinan dari sebagian kolom data yang diindeks, sehingga memungkinkan query untuk menemukan data dengan lebih cepat. Indeks B-Tree adalah salah satu jenis indeks yang paling umum digunakan dalam MySQL karena efisiensinya dalam mencari dan menyortir data. Penggunaan indeks B-Tree dapat mempercepat proses pencarian data dalam tabel yang besar secara signifikan[2].

Penelitian ini relevan karena kecepatan akses data sangat kritikal dalam konteks monitoring dan analisis real-time. Dalam lingkungan di mana keputusan harus diambil dengan cepat berdasarkan data terbaru, setiap detik yang dihemat dalam proses load data dapat berdampak signifikan pada efisiensi operasional dan ketepatan pengambilan keputusan. Dengan demikian, memahami dampak indexing terhadap performa load database di platform Grafana adalah penting bagi pengembang dan administrator database yang ingin mengoptimalkan kinerja sistem.

TINJAUAN PUSTAKA

Dalam konteks pengelolaan basis data relasional seperti MySQL yang digunakan dalam aplikasi monitoring dan analisis data real-time seperti Grafana, performa akses data menjadi faktor krusial yang mempengaruhi responsivitas sistem secara keseluruhan. Dalam tinjauan pustaka ini, kami akan mengulas beberapa konsep dan penemuan terkait penggunaan indexing dalam meningkatkan kecepatan load data.

Pengenalan MySQL dan Grafana

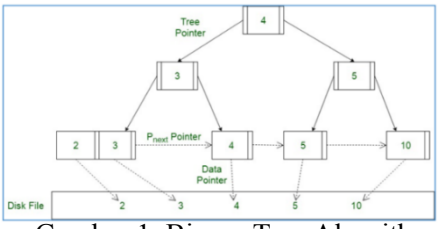
MySQL merupakan sistem manajemen basis data relasional yang populer karena kemampuannya dalam menangani transaksi tingkat tinggi dan aplikasi web skala besar. Dengan dukungan untuk indeks B-Tree, MySQL memungkinkan pengembang untuk meningkatkan performa query dengan mempercepat pencarian dan pengurutan data berdasarkan kolom tertentu [1].

Grafana adalah platform open-source yang digunakan untuk visualisasi dan monitoring data dari berbagai sumber, termasuk basis data relasional seperti MySQL. Dengan antarmuka yang intuitif dan kemampuan untuk mengintegrasikan berbagai sumber data, Grafana menjadi pilihan utama bagi organisasi dalam membangun dashboard monitoring real-time yang efektif [7].

Indeks B-Tree dalam MySQL

Indeks B-Tree merupakan struktur data yang paling umum digunakan dalam pengindeksan basis data relasional. Indeks ini dirancang untuk mendukung pencarian efisien, dengan mempertahankan keseimbangan antara waktu akses dan ruang penyimpanan. Ketika sebuah tabel diindeks menggunakan indeks B-Tree pada kolom kunci atau kolom yang sering diquery, MySQL

dapat mempercepat operasi pencarian data dengan mengurangi jumlah baris yang harus diperiksa secara langsung [1].

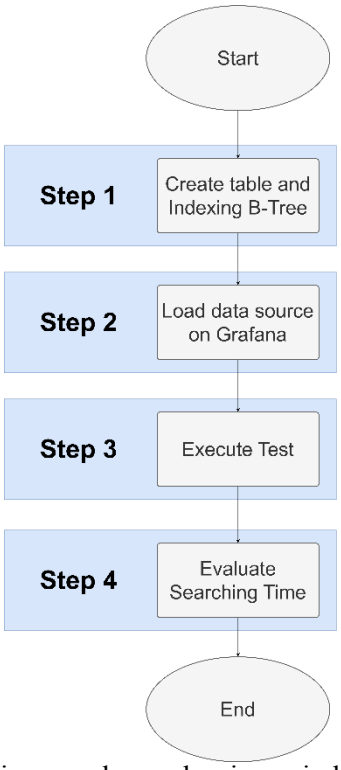


Gambar 1. Binary Tree Algorithm

Penggunaan indeks B-Tree secara signifikan meningkatkan performa query di MySQL, terutama pada tabel-tabel dengan baris yang banyak baris data. Mereka menemukan bahwa query yang menggunakan indeks B-Tree dapat mempercepat waktu eksekusi dibandingkan dengan query tanpa indeks[2].

METODE

Proses ini dapat dibagi menjadi empat fase. Pertama, sebuah tabel dibuat dengan indeks B-Tree pada fase desain basis data. Kedua, sumber data dengan berbagai ukuran data, misalnya 10.000 baris, disiapkan. Ketiga, pengujian dilakukan dengan merekam dan mengevaluasi waktu pencarian pada setiap query. Keempat, adalah fase analisis di mana semua data yang dikumpulkan pada fase sebelumnya dianalisis dan dibahas. Akhirnya, berdasarkan semua bukti dan hasil, sebuah kesimpulan dapat dibuat.



Gambar 2. Diagram alur evaluasi pengindeksan B-Tree

Bagian ini menunjukkan implementasi teknik pengindeksan B-Tree. Teknik ini kemudian diterapkan pada MySQL. Selain itu, proses implementasi akan dijelaskan dan ditunjukkan secara rinci, mulai dari cara membuat basis data, membuat indeks, memuat data ke dalam basis data, menjalankan query dan mendapatkan hasilnya, hingga tahap akhir yaitu membuat analisis komparatif berdasarkan hasil tersebut. Contoh proses penelitian di mana semua proses ditunjukkan dalam Gambar 2, yang menyajikan dan menjelaskan gambaran umum dari penelitian. Hasil uji

berdasarkan kecepatan yang diperkirakan berdasarkan model matematika komputasi atau diekstrapolasikan dari eksperimen lain akan diidentifikasi dengan jelas. Bagian berikutnya akan membahas: Langkah 1: Membuat tabel dan pengindeksan, Langkah 2: Memuat data, Langkah 3: Melaksanakan pengujian, Langkah 4: Mengevaluasi hasil.

Tahap pertama adalah menjelaskan cara membuat tabel dan indeks B-Tree. Sebuah tabel dibuat menggunakan pernyataan SQL seperti yang ditunjukkan pada Gambar 3. Jika tidak ada indeks yang dibuat, sistem basis data secara otomatis akan menetapkan teknik pengindeksan biner setelah tabel dibuat. Kunci utama diidentifikasi sebagai ID dan diatur menjadi NOT NULL. Setelah tabel dibuat, pengindeksan dapat diterapkan menggunakan sintaks pernyataan SQL CREATE-INDEX-INDEX_NAME-ON-TABLE_NAME-USING-INDEX_TYPES. Berdasarkan Gambar 3, ditunjukkan contoh cara membuat indeks sederhana menggunakan sintaks SQL.

Kemudian, setiap tabel yang menggunakan teknik pengindeksan diperiksa dengan menggunakan perintah \D TABLE_NAME. Bagian berikutnya akan membahas cara memuat data ke dalam basis data.

```
CREATE INDEX b_table ON b_table USING b tree (id);
```

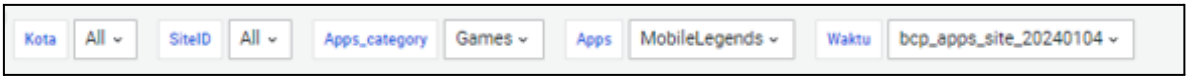
Gambar 3. Pembuatan sintaks SQL Indexing

Kemudian, data tersebut diunggah menggunakan Grafana ke dalam tabel yang telah dibuat. Misalnya, sebuah tabel bernama $Waktu dibuat. Berdasarkan Gambar 3, ditunjukkan bagaimana dataset disalin ke dalam tabel $Waktu. Perlu dicatat bahwa proses yang sama diterapkan untuk B-Tree dalam berbagai ukuran data. Bagian berikutnya akan membahas cara menjalankan pengujian.

```
---Main Query Grafana
SELECT
  app,
  site_id,
  total_volume,
  kabupaten,
  latitude,
  longitude
FROM
 $Waktu
WHERE
   kabupaten in ($Kota )
  and app = '$Apps'
  and site_id in ($SiteID )
```

Gambar 4. Main Query Dashboard Grafana

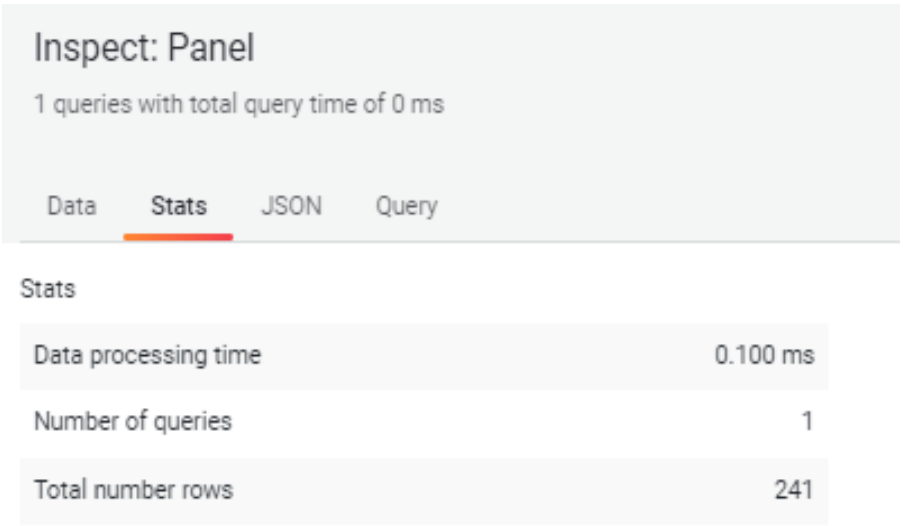
Gambar 5. Query Variabel Dashboard Grafana



Gambar 6. Variabel Dashboard Grafana

Untuk main query pembuatan dashboard dapat dilihat pada Gambar 4 dan refrensi query $Waktu , $SiteID, $Kota, $Apps, $Apps_categoty dapat dilihat pada Gambar 5, jadi ketika pengguna ingin mengganti untuk menampilkan variable tertentu dapat memilih menu Gambar 6.

Untuk pelaksanaan, setiap pengujian harus diterapkan menggunakan B-Tree. Setiap query pada setiap tabel harus dipindai. Dengan menggunakan pernyataan SELECT sederhana dari SQL, dapat diidentifikasi setiap parameter termasuk biaya, kecepatan, dan baris pada setiap catatan. Gambar 7 menunjukkan contoh pemindaian query menggunakan SELECT dan semua detailnya. Waktu pencarian untuk B-Tree dicatat. Pemindaian query SELECT dijalankan dan dianalisis pada tabel dengan baris tertentu. Semua hasilnya dicatat. Selanjutnya, akan dibahas hasil untuk B-Tree.



Gambar 7. Hasil Pengujian Indexing MySQL Dashboard Grafana

Bagian ini menjelaskan bagaimana pengindeksan B-Tree diterapkan dalam pengujian kinerja query. Dari hasil inspeksi panel, terlihat bahwa terdapat satu query dengan waktu total eksekusi 0 ms. Proses pemrosesan data memakan waktu 0,100 ms dengan total 241 baris data yang diproses. Hal ini menunjukkan efisiensi tinggi dari teknik pengindeksan B-Tree dalam menangani query tersebut. Statistik ini mendukung kesimpulan bahwa teknik pengindeksan, seperti B-Tree, memiliki dampak signifikan terhadap peningkatan kinerja pengambilan data dari basis data, memudahkan pencarian dan pemrosesan data dengan kecepatan yang optimal.

Pada tahap pengujian, dilakukan perbandingan waktu eksekusi query antara tabel yang diindeks dan tabel yang tidak diindeks. Pengujian ini dilakukan dengan mengambil data dari sepuluh percobaan untuk masing-masing jenis tabel. Setiap percobaan melibatkan pengukuran total waktu eksekusi query dalam satuan detik, jumlah baris tiap tabel, serta ukuran tabel dalam gigabyte.

Data yang dihasilkan menunjukkan bahwa tabel yang diindeks memiliki waktu eksekusi query yang secara konsisten lebih cepat dibandingkan tabel yang tidak diindeks. Misalnya, pada percobaan pertama, tabel yang diindeks memiliki waktu eksekusi query sebesar 6.79 detik, sementara tabel yang tidak diindeks membutuhkan 24.70 detik. Tren serupa terlihat pada

percobaan-percobaan berikutnya, dimana waktu eksekusi query untuk tabel yang diindeks selalu lebih rendah.

Berikut adalah tabel pengujian, dengan membandingkan waktu total permintaan query antara tabel yang menggunakan indeks dan tabel yang tidak menggunakan indeks :

Tabel 1. Data Hasil Percobaan

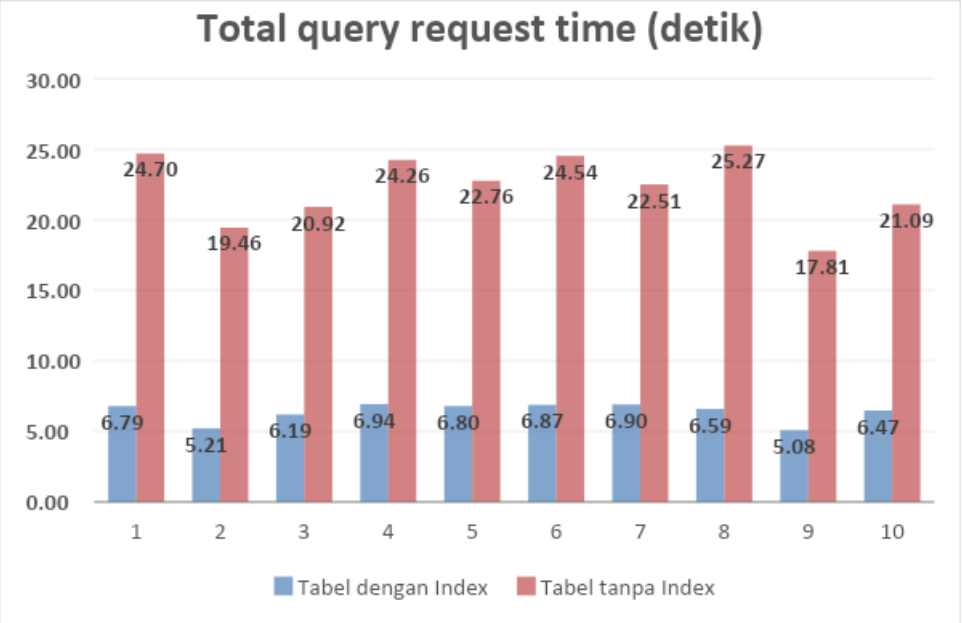
Data Percobaan ke-	Total query request time (detik)		Jumlah baris tiap Tabel	Ukuran Tabel (Gigabyte)
	Tabel dengan Index	Tabel tanpa Index		
1	6,79	24,70	8.451.882	1,7
2	5,21	19,46	8.077.248	1,6
3	6,19	20,92	8.472.111	1,7
4	6,94	24,26	8.496.625	1,7
5	6,80	22,76	8.296.611	1,7
6	6,87	24,54	8.418.450	1,7
7	6,90	22,51	8.420.379	1,7
8	6,59	25,27	8.459.862	1,7
9	5,08	17,81	8.095.159	1,6
10	6,47	21,09	8.261.586	1,7

Selain itu, meskipun ukuran tabel dan jumlah baris tiap tabel hampir sama untuk kedua jenis tabel, perbedaan signifikan dalam waktu eksekusi query menegaskan pentingnya penggunaan indeks dalam meningkatkan efisiensi akses data pada database yang besar. Pengujian ini membuktikan bahwa indeks sangat berperan dalam mempercepat waktu respon query, yang sangat krusial dalam aplikasi dengan data yang besar dan kompleks.

HASIL DAN PEMBAHASAN

Pembahasan

Pada tahap pembahasan, hasil pengujian dari 10 percobaan menunjukkan perbedaan yang signifikan dalam total waktu eksekusi query antara tabel yang diindeks dan tabel yang tidak diindeks. Dari data yang telah dikumpulkan, terlihat bahwa tabel yang diindeks consistently menghasilkan waktu eksekusi yang jauh lebih cepat. Berikut untuk hasil percobaan bisa dilihat pada Gambar 8



Gambar 8. Grafik Hasil Pengujian Indexing MySQL Dashboard Grafana

Pada percobaan pertama, waktu eksekusi query untuk tabel yang diindeks adalah 6,79 detik, sedangkan tabel yang tidak diindeks memerlukan 24,70 detik untuk menyelesaikan query yang sama. Perbedaan ini mencerminkan pengurangan waktu sebesar lebih dari 70%. Selain itu, hasil pengujian ini menunjukkan bahwa rata-rata waktu eksekusi untuk tabel yang diindeks adalah sekitar 6,28 detik, sedangkan untuk tabel yang tidak diindeks adalah 22,33 detik. Penggunaan indeks berhasil menurunkan waktu eksekusi query secara konsisten di setiap percobaan, bahkan ketika ukuran tabel dan jumlah baris hampir sama. Ini mengindikasikan bahwa indeks sangat efektif dalam meningkatkan efisiensi pencarian dan pengambilan data dalam database.

Dari hasil ini, dapat disimpulkan bahwa penerapan indeks pada tabel dalam database yang besar sangat krusial untuk meningkatkan performa query. Indeks membantu dalam mengurangi waktu respon query secara signifikan, yang pada gilirannya meningkatkan efisiensi operasional dan pengalaman pengguna. Implementasi indeks menjadi sangat penting, terutama pada sistem yang menangani volume data besar dan membutuhkan akses data yang cepat dan real-time. Oleh karena itu, penggunaan indeks harus dipertimbangkan sebagai praktik standar dalam desain dan pengelolaan database yang efisien.

KESIMPULAN

Penelitian ini membahas tentang teknik pengindeksan dengan algoritma yang digunakan. Melalui eksperimen, penelitian ini mengimplementasikan indeks B-Tree untuk basis data. Dengan menganalisis hasil dari penelitian ini, menunjukkan bahwa penggunaan indeks B-Tree memiliki dampak signifikan dalam sistem basis data dalam hal kinerja. Penelitian ini menyarankan untuk melakukan investigasi lebih lanjut, seperti pengembangan algoritma yang ditingkatkan dari B-Tree yang dapat mengatasi penurunan kinerja saat ukuran data semakin besar. Studi kasus ini diharapkan dapat memberikan landasan untuk penelitian masa depan dalam pengembangan teknik-teknik pengindeksan baru yang dapat meningkatkan performa kueri data secara lebih efisien.

DAFTAR PUSTAKA

[1] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th ed. McGraw-Hill, 2020.

[2] A. Aminuddin, M. Z. Saringat, S. A. Mostofa, A. Mustapha, and M. H. Hassan, “A Case Study on B-Tree Database Indexing Technique,” Journal of Soft Computing and Data Mining, vol. 1, no. 1, pp. 27-35, 2020. DOI: https://doi.org/10.30880/jscdm.2020.01.01.004

[3] Jung S.,Kim C.S., Lee J., and Hong B., “Hybrid Indexing Consisting of TPR*-Tree and Hash Map for Real-Time Update and Query of Tactical Moving Objects,” in IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 2018.

[4] Jantkal B. A. and Deshpande S. L., “Hybridization of B-Tree and HashMap for Optimized Search Engine Indexing,” in International Conference on Smart Technologies for Smart Nation (SmartTechCon), Bangalore, India: IEEE, 2017.

[5] Minock M., “C-Phrase: A system for building robust natural language interfaces to databases,” Data & Knowledge Engineering, vol. 69, no. 3, pp. 290-302, 2010.

[6] Li F. And Jagadish H. V., “Constructing an interactive natural language interface for relational databases,” Proceedings of the VLDB Endowment, vol. 8, no. 1, pp. 73-84, 2014.

[7] Grafana Labs. Introduction to Grafana: Features and Benefits of Using Grafana for Data Visualization. Retrieved from Grafana Labs. (2020)