

## Analisis Perbandingan Unjuk Kerja Database SQL dan Database NoSQL Untuk Mendukung Era Big Data

Ari Fadli\*, Mulki Indana Zulfa, Azis Wisnu Widhi Nugraha, Acep Taryana, dan Muhammad Syaiful Aliim

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman, Purwokerto, Jawa Tengah, Indonesia

\*Corresponding author, e-mail: arifadli@unsoed.ac.id

**Abstrak** - Baru-baru ini, perkembangan data di dunia digital telah menghasilkan tantangan baru. Tantangannya adalah meningkatnya jumlah dan kompleksitas data dan variasi data (terstruktur dan tidak terstruktur). Sistem manajemen basis data konvensional (RDBMS) tidak dapat menjawab tantangan ini secara efektif karena dalam hal ini RDBMS menggunakan pendekatan tabel terstruktur sebagai konsep dalam menyimpan informasi. Saat ini, telah hadir sistem manajemen basis data yang disebut NoSQL, Sistem NoSQL memiliki kemampuan menangani data yang tidak terstruktur yang jumlah dan kompleksitas terus meningkat. Dalam penelitian ini, kami menyajikan hasil analisis perbandingan unjuk kerja Sistem Manajemen Basis Data konvensional (RDBMS) dengan NoSQL pada kemampuannya dalam menangani pertumbuhan data yang terus meningkat. Metode yang digunakan dalam melakukan analisis perbandingan unjuk kerja RDBMS dengan NoSQL adalah waktu eksekusi yang dibutuhkan untuk melakukan operasi dasar pada sistem basis data yaitu *create*, *read*, *delete* dan *update*. RDBMS yang digunakan pada penelitian kali ini adalah MySQL sedangkan pada sistem NoSQL digunakan oleh Redis Database. Hasil penelitian menunjukkan bahwa Redis Database memiliki waktu eksekusi yang lebih baik daripada MySQL Database, terdapat peningkatan kecepatan waktu eksekusi tersebut sebesar 87.58% pada operasi *create*, 85.53% pada operasi *update*, dan pada operasi *delete* sebesar 86.40% dan sedangkan pada operasi *read* peningkatan kecepatan waktu eksekusi yang diperoleh sebesar 57.09%, sehingga secara rata-rata *database* Redis memiliki unjuk kerja yang lebih baik dari *database* MySQL yaitu sebesar 79.15%.

*Kata Kunci* : *MySql Database, Redis Database, Pemrograman PHP*

**Abstract** - Recently, the development of data in the digital world has produced new challenges. The challenge is increasing number and complexity of data and variations in data (structured and unstructured). Conventional database management system (RDBMS) is not able to answer these challenges effectively because in RDBMS uses a related structured table approach to store information. Currently present, a database management system called NoSQL. This NoSQL system was introduced to provide quick scalability (amount and complexity) and unstructured data. In this study, we present the performance analysis of RDBMS and NoSQL database management systems in the case of increasing the number of data. We use the simple php page to test database systems, experiments are done with use four data sets of different sizes from academic information systems. The database NoSQL approach is used by Redis and the SQL database approach is used by MySQL. The results show that the noSql database has a execution time better than the MySQL database, there is an increase in the speed of the execution time by 87.58% in the create operation, 85.53% in the update operation, and in the delete operation by 86.40% and while in the read operation the increase in execution time speed is obtained by 57.09%, so that on average the Redis database has a show work better than the MySQL database at 79.15%..

*Keywords* : *MySql Database, Redis Database, PHP Programming*

### 1. Pendahuluan

Pada perkembangan teknologi aplikasi berbasis web, dikenal beberapa bahasa pemrograman yang dapat digunakan sebagai tools untuk membangun aplikasi tersebut. Beberapa bahasa pemrograman tersebut yaitu PHP, ASP.Net, Ruby, Python, dan Java. Dari beberapa bahasa pemrograman tersebut PHP (*Hypertext Preprocessor*) merupakan yang paling banyak digunakan dalam mengembangkan aplikasi berbasis web, seperti ditunjukkan pada Gambar 1, tampak bahwa 81,7% bahasa pemrograman PHP dipilih sebagai bahasa yang

digunakan untuk mengembangkan aplikasi berbasis web [1].

PHP dipilih sebagai tools pengembang aplikasi berbasis web dikarenakan memiliki banyak keuntungan, beberapa diantaranya yaitu dukungan untuk berbagai jenis *database* serta tersedianya banyak dokumentasi yang dapat dipelajari, selain itu PHP adalah bahasa pemrograman *open source* yang dapat digunakan untuk mengembangkan aplikasi berbasis web [2][3][4][5].

*Database* didefinisikan sebagai pengumpulan data yang terorganisir. selain itu, basis data juga

Received date 2020-04-28, Revised date 2020-10-31, Accepted date 2020-11-01

<https://doi.org/10.25077/jnte.v9n3.774.2020>

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

dapat merujuk ke sistem *datastore* atau juga disebut kumpulan dataset yang disimpan dalam sistem *datastore*. Sistem *datastore* ini yang menangani semua dataset dan organisasinya, mengambil, menyimpan, dan memperbarui disebut Sistem Manajemen Basis Data (DBMS) [6].

DBMS digunakan sebagai perangkat untuk melakukan penyimpanan dan manajemen data. Saat ini DBMS diklasifikasikan ke dalam dua kategori, yaitu DBMS Relasional yang menggunakan *Structured Query Languages* (SQL) dan dikenal sebagai *SQL Database*, dan DBMS Non-Relasional dikenal sebagai NoSQL dan singkatan dari *Not Only Structured Queries Languages*.

Bahasa pemrograman PHP dan ASP.Net memiliki API yang dapat digunakan untuk menangani koneksi basis data. Oleh karena itu, PHP dapat digunakan untuk pada database relasional dan NoSQL *Database* termasuk Oracle, MS SQL Server, MySQL, dan MongoDB. Tetapi MySQL lebih disukai oleh pengembang PHP sebagai pilihan *database* untuk mengembangkan aplikasi berbasis web [7]. Saat ini PHP menawarkan API yang berbeda untuk terhubung ke MySQL dengan ekstensi *mysqli* dan *PDO* [2].

*Database* relasional dan *database* NoSQL menyediakan mekanisme yang berbeda untuk menyimpan dan mengambil data [8]. *Database* relasional menyimpan data dalam bentuk struktur tabel. Setiap tabel relasional terdiri dari baris dan kolom. Secara luas *database* relasional [9] telah banyak digunakan untuk menyimpan informasi seperti dokumen bisnis, informasi keuangan, dan data akademik [10][11]. *Database* relasional tidak mendukung data yang tidak terstruktur [12]. Fitur utama dari sistem basis data relasional mengatur data ke dalam hubungan dan menyediakan ACID (*Atomicity, Consistency, Isolation, Durability*).

Saat ini aplikasi, seperti intelijen bisnis, Web 2.0 dan jejaring sosial membutuhkan pemrosesan data yang sangat besar hingga kapasitas *terabyte*. *cloud computing* adalah sebuah model komputasi di mana prosesor, *storage*, perangkat jaringan, dan perangkat lunak menjadi abstrak dan hanya disediakan sebagai layanan. Untuk memenuhi kebutuhan tersebut diperlukan pemrosesan data secara terdistribusi, dan dalam hal ini *database* relasional dan tidak mengakomodasinya [8]. Solusinya adalah model *database* lain yang saat ini dikenal istilah *database* NoSQL. Salah satu kelebihan menggunakan *database* NoSQL, tidak seperti *database* relasional, *database* NoSQL dapat menangani data yang tidak terstruktur seperti

dokumen, email, multimedia dan media sosial secara efisien [13]. *Database* NoSQL telah banyak diterapkan dalam aplikasi big data dan aplikasi web waktu nyata [14].

Dalam konsep *database* NoSQL ada empat jenis skema dalam melakukan penyimpanan datanya yaitu *key-value storage, document stores, family column stores, and graphic databases* [14]. Dari beberapa skema tersebut, skema *key-value storage* merupakan bentuk paling sederhana untuk menyimpan data, dimana setiap kunci dipetakan ke nilai yang berisi data arbitrer.

Prinsip kerja *key-value storage* mirip dengan peta atau kamus di mana data ditangani oleh sebuah kunci unik. Kunci unik mengidentifikasi nilai dan mengekstrak data yang disimpan. Nilainya dalam bentuk blok data yang memiliki struktur tidak tetap. Skema *key-value storage* cepat dalam proses membaca dan menulis data, namun lambat dalam *updating* data. *Redis Database* yang merupakan *database* NoSQL bekerja berdasarkan skema *key-value storage* [15].

Selain itu, *redis database* merupakan salah satu *database* NoSQL yang paling populer [16], bersifat *open source* dan *server single-threaded* [17]. *Redis Database* memiliki banyak opsi penyimpanan data termasuk *strings, lists, sets, hashes dan lists*, beberapa *functions* seperti *disk persistence dan scripts*. Dalam *Redis Database* data yang disimpan sebagai teks biasa dan tidak didukung oleh enkripsi data [18].

Penelitian lain yang membahas perbandingan unjuk kerja *database* SQL dengan *database* NoSQL adalah [19][20][21][22], penelitian lainnya membahas sisi keamanan yang ada pada *database* NoSQL [13], perbandingan kinerja antar *database* NoSQL [23], penerapan *database* NoSQL dalam Big Data [15][18].

Fokus kajian yang dilakukan pada penelitian ini adalah analisis unjuk kerja dari *database* MySQL dan *database* Redis pada waktu eksekusi untuk melakukan operasi dasar *database* yaitu CRUD (*Create, Read, Update, Delete*). Operasi dasar ini dipilih dikarenakan dalam konteks fungsi manajemen basis data operasi dasar CRUD merupakan fungsi yang dapat menggambarkan kemampuan *database* dalam manajemen data didalam media penyimpanan [24].

## 2. Metoda

Dalam melakukan analisis unjuk kerja antara *database* SQL dan *database* NoSQL pada waktu

eksekusi yang diperlukan untuk melakukan operasi dasar yaitu (*Create, Read, Update, Delete*), dipilih Redis versi 3.0.503 sebagai *database* NoSQL dan MySQL versi 5.5.32 sebagai *database* SQL. Data yang digunakan sebagai objek penelitian ini berasal dari tabel *krs\_khs* yang terdiri dari kolom *no*, *nama*, *nim*, *ipk*, *jml\_sks*, *semester*, dan *program studi*. Jumlah baris yang digunakan pada tabel tersebut dibuat bervariasi yaitu adalah 250, 500, 750 dan 1000 baris.

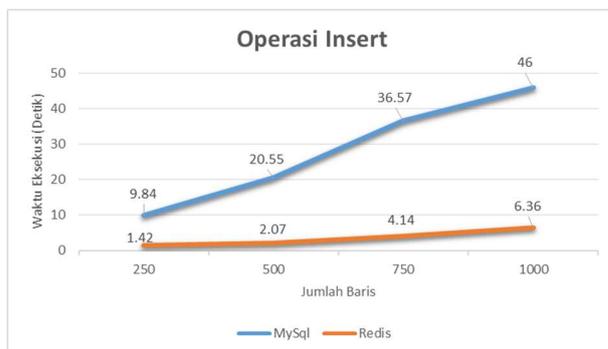
Untuk merekam waktu eksekusi masing-masing operasi dasar tersebut digunakan fungsi PHP *microtime*. Fungsi ini akan mencatat waktu dari awal *runtime* skrip dan sampai selesai. Selanjutnya untuk menjalankan empat operasi dasar, kami menggunakan sebuah halaman PHP dan menggunakan server web XAMPP.

### 3. Hasil dan Pembahasan

Hasil yang diperoleh pada masing-masing operasi yang dijalankan, akan menunjukkan unjuk kerja yang dimiliki oleh *database* SQL dan *database* NOSQL. Semakin singkat waktu eksekusi yang dibutuhkan untuk menjalankan operasi tersebut maka *database* tersebut memiliki unjuk kerja yang lebih baik

#### 3.1. Operasi Create (Insert)

Gambar 1 menunjukkan waktu eksekusi pada operasi *create* dalam *Database* Redis dan *Database* MySQL. Berdasarkan Gambar 1 tersebut tampak bahwa waktu eksekusi meningkat seiring dengan pertumbuhan data. Tetapi dalam hal ini *Database* Redis memberikan kinerja yang jauh lebih baik daripada *Database* MySQL, pada operasi *create*.



Gambar 1. Waktu Eksekusi Operasi Insert

Dari hasil yang diperoleh pada Gambar 1, terdapat *persentase* peningkatan kecepatan waktu eksekusi pada operasi *create*, yaitu sebesar 85.56 % untuk 250 baris data, 89.20% untuk 500 baris data,

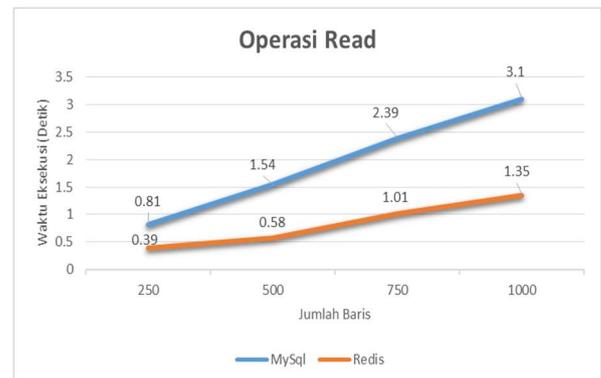
pada 750 baris data sebesar 88.67%, dan 86.17% untuk 1000 baris data, sehingga rata-rata peningkatannya adalah sebesar 87.58%.

#### 3.2. Operasi Read

Gambar 2 menunjukkan waktu eksekusi pada operasi *read* dari *Database* MySQL dan *Database* Redis.

Berdasarkan hasil ini menunjukkan bahwa waktu eksekusi meningkat seiring dengan pertumbuhan data. Tetapi dalam hal ini *Database* Redis memberikan kinerja yang sedikit lebih baik daripada *Database* MySQL, pada operasi *read*.

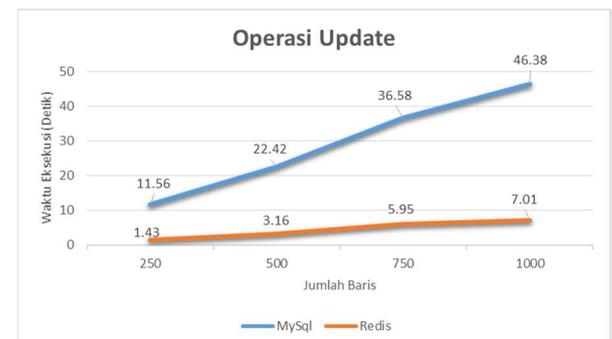
Dari hasil yang diperoleh pada Gambar 2, terdapat *persentase* peningkatan kecepatan waktu eksekusi pada operasi *read*, yaitu sebesar 51.85 % untuk 250 baris data, 62.33% untuk 500 baris data, pada 750 baris data sebesar 57.74%, dan 56.45% untuk 1000 baris data, sehingga rata-rata peningkatannya adalah sebesar 57.09%.



Gambar 2. Waktu Eksekusi Operasi Read

#### 3.3. Operasi Update

Gambar 3 menunjukkan waktu eksekusi pada operasi *update* dalam *Database* Redis dan *Database* MySQL.



Gambar 3. Waktu Eksekusi Operasi Update

Berdasarkan hasil yang diperoleh menunjukkan bahwa waktu eksekusi meningkat seiring dengan pertumbuhan data. Tetapi dalam hal ini *Database*

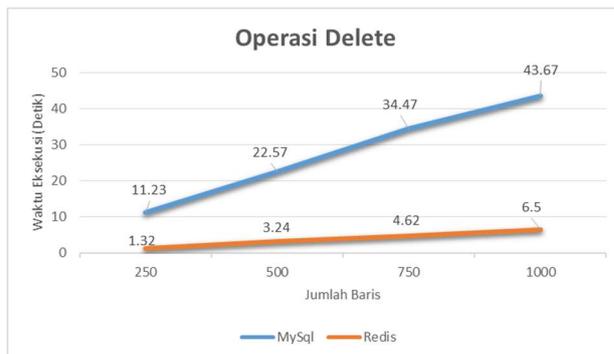
Redis memberikan kinerja yang jauh lebih baik daripada *Database MySQL*, pada operasi *update*.

Berdasarkan Gambar 3, dapat dihitung *persentase* peningkatan kecepatan waktu eksekusi pada operasi *update*, yaitu sebesar 87.62 % untuk 250 baris data, 85.90% untuk 500 baris data, pada 750 baris data sebesar 83.73%, dan 84.88% untuk 1000 baris data, sehingga rata-rata peningkatannya adalah sebesar 85.53%.

### 3.4. Operasi Delete

Gambar 4 menunjukkan waktu eksekusi pada operasi *delete* dalam *Database Redis* dan *Database MySQL*.

Berdasarkan hasil ini menunjukkan bahwa waktu eksekusi meningkat seiring dengan pertumbuhan data. Tetapi dalam hal ini *Database Redis* memberikan kinerja yang jauh lebih baik daripada *Database MySQL*, pada operasi *delete*.



Gambar 4. Operasi *Delete* (Satuan Detik).

Dari hasil yang diperoleh pada Gambar 4, terdapat *persentase* peningkatan kecepatan waktu eksekusi pada operasi *delete*, yaitu sebesar 88.24 % untuk 250 baris data, 85.64% untuk 500 baris data, pada 750 baris data sebesar 86.59%, dan 85.11% untuk 1000 baris data, sehingga rata-rata peningkatannya adalah sebesar 86.40%.

Berdasarkan hasil yang diperoleh dari operasi dasar CRUD yang ada pada Gambar 1 - Gambar 4, tampak bahwa seiring bertambahnya jumlah baris data, *Database Redis* memiliki kemampuan kecepatan eksekusi yang lebih cepat jika dibandingkan dengan *Database MySQL*, sehingga dengan hal ini tampak bahwa *Redis Database* memiliki kinerja yang lebih baik dalam hal mengatasi pertumbuhan data di Era Big Data saat ini, jika dibandingkan dengan *MySQL database*.

## 4. Kesimpulan

Berdasar pada hasil penelitian yang dilakukan diperoleh bahwa waktu eksekusi *database Redis* lebih cepat dari *database MySQL* untuk melakukan operasi dasar yaitu *create, read, delete* dan *update*. Peningkatan kecepatan waktu eksekusi tersebut sebesar 87.58% pada operasi *create*, 85.53% pada operasi *update*, dan pada operasi *delete* sebesar 86.40% dan sedangkan pada operasi *read* peningkatan kecepatan waktu eksekusi diperoleh sebesar 57.09%, sehingga secara rata-rata *Database Redis* memiliki untuk kerja yang lebih baik dari *Database MySQL* yaitu sebesar 79.15%.

Berdasarkan hal tersebut, penggunaan *Redis Database* ini secara efektif dapat menangani pertumbuhan data pada Era Big Data. Unjuk kerja *Redis Database* yang dikaji saat ini baru sebatas terkait dengan variasi pertumbuhan data namun belum dikaitkan variasi sumber data yang populer pada Era Big Data, misalkan data dari hasil sensing sebuah sensor, transaksi pada toko online / media sosial dan lainnya sehingga untuk penelitian berikutnya dapat dirancang mekanisme untuk menguji unjuk kerja *Database Redis* dan *Database MySQL* pada kaitannya dengan variasi data tersebut

## Ucapan Terima Kasih (Acknowledgement)

Terima kasih kepada LPPM Universitas Jenderal Soedirman yang telah membiayai penelitian ini dalam skim Penelitian Riset Dosen Pemula

## Daftar Pustaka

- [1] N. Prokofyeva and V. Boltunova, "Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems," *Procedia Comput. Sci.*, vol. 104, no. December 2016, pp. 51–56, 2016, doi: 10.1016/j.procs.2017.01.059.
- [2] A. Mishra, "Critical Comparison Of PHP And ASP . NET For Web Development," *Int. J. Sci. Technol. Res.*, vol. 3, no. 7, pp. 331–333, 2014.
- [3] W.-M. Chen and Y.-C. Chen, "Web design and implementation for remote control. In Intelligent Control and Automation (WCICA)," *IEEE*, p. 2012, 2012.
- [4] S. Gabarro, *The Different Approaches of Web Programming. In Web Application Design and Implementation: Apache 2, PHP5, MySQL, JavaScript, and Linux/UNIX. 1st ed.* 2007.

- [5] Fadli, A., Sugiyanto, G. Zulfa, M.I. "Upaya Mereduksi Jumlah Kecelakaan Lalu Lintas Melalui Penggunaan Sistem Informasi Geografis", *Jurnal Warta LPM*, 23(2) September 2020.
- [6] Fadli A., "Konsep Dasar In Memory Database," pp. 1–6, 2019.
- [7] M. E. Davis and J. A. Phillips, *Learning PHP & MySQL: Step-by-Step Guide to Creating Database Driven Web Sites 2nd ed.* O'REILLY, 2007.
- [8] H. Zhang, G. Chen, B. C. Ooi, K. L. Tan, and M. Zhang, "In-memory big data management and processing: A survey. *IEEE Transactions on Knowledge and Data Engineering*," vol. 27, no. 7, pp. 1920–1948, 2015, doi: 10.1590/s1809-98232013000400007.
- [9] M. A. Mohamed, M. A. Mohamed, O. G. Altrafi, and M. O. Ismail, "Relational Vs. NoSQL databases: A survey," *Int. J. Comput. Inf. Technol.*, vol. 03, no. 03, pp. 2279–0764, 2014.
- [10] A. Fadli, M. I. Zulfa, and Y. Ramadhani, "Performance Comparison of Data Mining Classification Algorithms for Early Warning System of Students Graduation Timeliness," *J. Teknol. dan Sist. Komput.*, vol. 6, no. 4, p. 158, 2018, doi: 10.14710/jtsiskom.6.4.2018.158-163.
- [11] M. I. Zulfa, A. Fadli, and Y. Ramadhani, "Classification model for graduation on time study using data mining techniques with SVM algorithm," *AIP Conf. Proc.*, vol. 2094, 2019, doi: 10.1063/1.5097475.
- [12] V. D. Jogi and A. Sinha, "Performance evaluation of MySQL, Cassandra and HBase for heavy write operation.," *Recent Adv. Inf. Technol.*, p. 2016, 2016.
- [13] and J. A. L. Okman, N. Gal-Oz, Y. Gonen, E. Gudes, "Security issues in NoSQL databases," *Proc. 10th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2011, 8th IEEE Int. Conf. Embed. Softw. Syst. ICES 2011, 6th Int. Conf. FCST 2011*, vol. 541–547, p. 2011, 2011.
- [14] S. Heripracoyo and K. R., "Big Data Analysis with MongoDB for Decision Support System," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 14, no. 3, 2016, doi: <https://doi.org/10.3929/ethz-b-000238666>.
- [15] A. B. M. Moniruzzaman and S. A. Hossain, "NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison," vol. 6, no. 4, pp. 1–13, 2013.
- [16] J. L. Carlson, *Redis in Action*. 2013.
- [17] R. Lubis and A. Sagala, "Multithread performance on a single thread in-memory database.," in *Information Technology and Electrical Engineering (ICITEE), 7th International Conference on IEEE.*, 2015, pp. 571–575.
- [18] E. Sahafizadeh and M. A. Nematbakhsh, "A Survey on Security Issues in Big Data and NoSQL," *Adv. Comput. Sci. an Int. J.*, vol. 4, no. 4, pp. 68–72, 2015.
- [19] Y. Li and S. Manoharan, "A performance comparison of SQL and NoSQL databases," *IEEE Pacific RIM Conf. Commun. Comput. Signal Process. - Proc.*, no. November, pp. 15–19, 2013, doi: 10.1109/PACRIM.2013.6625441.
- [20] A. Raje and A. Jagdale, "Sql Vs NoSql : NewSql The Solution For Big Data," pp. 45–51.
- [21] C. Györödi, R. Gyorodi, G. Pecherle, and A. Olah, "A Comparative Study: MongoDB vs. MySQL Energetical sustainability of a local community using air flows View project Convergence of university practical training for integration with success in the labor market View project," no. June, 2015, doi: 10.13140/RG.2.1.1226.7685.
- [22] V. N. Gudivada, D. Rao, and V. V. Raghavan, "NoSQL Systems for Big Data Management," *2014 IEEE World Congr. Serv.*, pp. 190–197, 2014, doi: 10.1109/SERVICES.2014.42.
- [23] T. Patel and T. Eltaieb, "Relational Database vs NoSQL," *J. Multidiscip. Eng. Sci. Technol.*, vol. 2, no. 4, pp. 3159–40, 2015.
- [24] Bakar, M. A., Ismail, S., Idris, S., & Shukur, Z. seMeja API Design Based on CRUD+N Concept. Selangor, Malaysia, 2015.