**Basis Data Non Relational ( NO SQL )**



Disusun Oleh:

Nama : Maulana Yusup

Nim : 161240000597

Mata Kuliah : Basis Data Non Relational

Program Studi : Teknik Informatika

Dosen Pengampu : Nadia Anisa Maori M.Kom.,

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NAHDATUL ULAMA’ (UNISNU) JEPARA**

**TAHUN 2019**

Basis Data NO SQL

Muhammad Khairul Faqih

Universitas Islam Nahdlatul Ulama’ Jepara

[161240000597@unisnu.ac.id](mailto:161240000597@unisnu.ac.id)

# Abstrak

Saat ini jumlah penelitian pada topik basis data NoSQL masih banyak didiskusikan. Penelitian tentang isu terkini pada basis data NoSQL dilakukan berdasarkan tinjauan pustaka secara sistematik. Tujuan tinjauan pustaka ini adalah untuk melakukan analisis konsep, arsitektur dan pemodelan data dari NoSQL database, melakukan identifikasi isu-isu terkini dalam NoSQL database, dan menyimpulkan beberapa isu yang mungkin dapat digunakan untuk penelitian - penelitian selanjutnya. Langkah dalam tinjauan pustaka ini dengan mengumpulkan dan menganalisis bermacam jurnal dan penelitian dengan topik yang berhubungan dengan basis data NoSQL yang telah dipublikasikan antara waktu 2010 – 2018. Kesimpulan dari tinjauan pustaka ini akan menampilkan kemungkinan-kemungkinan pengembangan lebih lanjut mengenai NoSQL Databases.

**Kata kunci**: NoSQL Database; Tinjauan Pustaka; Isu terkini dalam database

1. **Pendahuluan**

Seiring perkembangan generasi aplikasi-aplikasi saat ini, seperti business intelligence, Web 2.0 dan social networking yang membutuhkan pengolahan data yang sangat besar dengan kapasitas terabyte atau bisa jadi petabyte. Dalam pengolahan data yang sebesar itu, dibutuhkan proses data yang terdistribusi (distributed processing). Relational database sudah tidak cukup tepat dalam distributed processing jika melibatkan data yang sangat besar tersebut. [1] Sehubungan dengan hal tersebut, teknologi database yang terdistribusi (distributed database) mulai popular dikarenakan kemampuannya yang lebih baik daripada teknologi database yang tradisional. [2] Terdapat beberapa alasan diperlukannya distributed processing. Pada satu sisi, sebuah program harus scalable dan harus dapat memanfaatkan beberapa sistem dan mampu memaksimalkan pemakaian arsitektur multi-core CPU. Di sisi lain, server sebuah website harus didistribusikan secara global untuk mengurangi latency dan failover.[1]

Dalam beberapa tahun terakhir, banyak perusahaan yang mengadopsi beberapa tipe NoSQL database dan mulai memunculkan banyak aplikasi menggunakan database tersebut, dan mendapat minat pasar yang luas. Setiap tipe NoSQL database memiliki pendekatan yang berbeda-beda. Salah satu keuntungan penggunaan NoSQL database, tidak seperti relational databases, NoSQL database dapat menangani unstructured data seperti dokumen, email, multimedia dan social media secara efisien.[3]

Fitur umum dari NoSQL database dapat diringkas dengan : high scalabilty and reliablitiy, data model yang sederhana, query language yang sederhana, kurangnya mekanisme penanganan dan pengelolaan konsistensi data dan pemeliharaan kendala integritas (contoh foreign key) dan hampir tidak ada dukungan untuk keamanan di tingkat database. [3]

tPentingnya dari pembelajaran ini untuk menampilkan kembali penelitian-penelitian baru yang berhubungan dengan lingkup penelitian NoSQL database, yang saat ini lebih banyak membahas tentang perbandingan antara SQL dengan NoSQL Database[4]–[6], Isu keamanan dalam NoSQL Database[3], perbandingan performa antara SQL dan NoSQL[7]–[9], perbandingan performa antara tipe-tipe NoSQL database[10], Penggunaan NoSQL dalam Big Data [11]–[13].

Tujuan utama dari penelitian ini untuk mengumpulkan, merinci, dan mensintesiskan sebuah tinjauan pustaka dari isu terkini mengenai NoSQL database. Selain itu perbandingan antara SQL dan NoSQL database cukup menarik untuk diteliti lebih lanjut.Struktur dalam penelitian ini dibagi kedalam 5 bagian, seperti pendahuluan, latar belakang teori, metodologi, hasil, dan kesimpulan dan saran.

1. **Pembahasan**
2. Basis Data

Basis Data atau *Database* adalah sekumpulan relasi data logika, dan deskripsi dari data yang dirancang untuk memenuhi kebutuhan informasi organisasi*. Database* memungkinkan tempat penyimpanan data yang besar dan dapat digunakan secara bersamaan oleh banyak departemen dan pengguna. *Database* mewakili entitas, atribut dan hubungan logis antara entitas. ( Connolly dan Begg, 2005, p15 )

Menurut Inmon (2005, 493), database adalah sekumpulan data yang saling berhubungan yang disimpan (biasanya dengan redudansi yang terkontrol dan terbatas) berdasarkan skema. Sebuah database dapat melayani single atau multiple applications.

Menurut Gottschalk dan Saether dalam jurnal (2010, 41), database adalah sekumpulan data yang terorganisir untuk mendukung banyak aplikasi secara efisien dengan memusatkan data dan mengontrol data redundant.

Dari pengertian di atas dapat dinyatakan bahwa database yaitu kumpulan-kumpulan data yang berisi informasi yang terhubung satu sama lain yang diorganisasikan dengan struktur tertentu serta dapat ditemukan dengan mudah dan cepat menggunakan bantuan komputer.

1. **MongoDB**

MongoDB adalah DBMS *open source* yang mendukung *Document Oriented Database* yang menyediakan performansi tinggi, *availability* tinggi dan *automatic scaling*. Performansi tinggi karena MongoDB didukung dengan *indexing* sehingga mendukung *query* menjadi lebih cepat dan mencakup *key* dari *document* dan *array*. Ketersediaan tinggi karena MongoDB didukung oleh replikasi. Satu set replika adalah sekelompok *server* MongoDB yang menyimpan sekumpulan data yang sama, menyediakan redudansi untuk meningkatkan ketersediaan data. *Automatic scaling* karena MongoDB didukung oleh *horizontal specification* sebagai bagian dari fungsi inti. *Sharding* otomatis ini mendistribusikan data ke sekelompok *node* dalam cluster. Namun set replika ini mengakibatkan “*eventual-consistency*” jadi data yang dimiliki bisa jadi saat yang sama antara user A dan B melihat *value* yang berbeda padahal dokumennya sama.

Format penulisan pada MongoDB adalah BSON(Binary JSON). Format BSON mirip dengan format JSON namun BSON memiliki beberapa kelebihan yaitu mampu menyimpan tipe data *date* dan tipe BinData. MongoDB memiliki *query* dasar yang mirip dengan *query* SQL sehingga memudahkan kita yang terbiasa menggunakan SQL untuk mempelajari sintaknya. *Query* pada MongoDB dibagi menjadi dua kelompok besar yaitu *read operation* dan *write operation*.

1. **No SQL**

NoSQL adalah sebuah tren yang berkembang dewasa ini sebagai wujud adanya kebutuhan untuk menangani data yang besar dan kebutuhan akan *cluster*. Pramod menyatakan bahwa NoSQL *is very ill-defined* meliputi *schemaless* data, jalan pada *cluster*, dan memiliki kemampuan menjual *traditional consistency* untuk *property* berguna yang lain. Mereka mengklaim NoSQL dapat membangun sistem yang lebih performatif, *scalability* yang lebih baik dan lebih mudah untuk diprogram. Selain itu, NoSQL memiliki beberapa jenis model data yang dapat dipilih sesuai dengan kebutuhan sistem. Beberapa kelebihan NoSQL jika dibandingkan dengan SQL adalah schemaless yaitu NoSQL tidak terikat dengan *schema*. Jadi dalam mendefinikan struktur data tidak harus diawal dan antar *record* tidak harus memiliki *field* yang sama. Hal ini cocok untuk data-data yang *nonuniform* seperti data ClikKB. Dan *large-scale* data Sebuah yaitu dapat menangkap lebih banyak data dan memprosesnya dengan lebih cepat. NoSQL sengaja didesain untuk berjalan pada sistem *cluster* sehingga ini akan lebih baik untuk skenario big data.

Menurut Eric Brewer, NoSQL didasarkan pada teori Consistency, Availability, Partition-Tolerance (CAP). Consistency Availability (CA) berseberangan dengan Partition-Tolerance dan berhubungan dengan replikasi. Consistency Partition- Tolerance (CP) berseberangan dengan Availability dalam penyimpanan data. Availability Partition-Tolerance (AP) dimana system mencapai kondisi eventual consistency melalui replikasi dan verifikasi yang konsisten dalam node yang telah terbagi – bagi. Dalam basis data NoSQL, penerapan konsep tersebut diterjemahkan dalam empat konsep dasar, yaitu Non-Relational, MapReduce, Schema Free, dan Horizontal Scaling, berikut adalah penjelasan dari ke empat konsep tersebut :

1. **Non-Relational**

Konsep Non-Relational dalam basis data NoSQL meliputi hirarki, graf, dan basis data berorientasi obyek. Penggunaan basis data non-relasional kembali merebak seiring bertambahnya aplikasi berbasis web yang menuntut skalabilitas tinggi. Meskipun memiliki kelemahan pada redundansi dan konsistensi data, namun basis data non-relasional dapat menyelesaikan permasalahan yang berkaitan dengan availability, dan partition-tolerance. Basis data non-relasional bertugas untuk memanipulasi kapasitas dan kapabilitas penyimpanan.

1. ***MapReduce***

MapReduce merupakan mode pemrograman yang diadaptasi dari pemrograman fungsional yang diimplementasikan mengolah dataset yang besar. Tujuan dari MapReduce adalah merancang suatu abstraksi baru yang memungkinkan pengguna untuk membuat antarmuka pemrograman sederhana dan menyembunyikan detail yang rumit dari paralelisasi, fault-tolerance, distribusi data, dan load-balancing dalam pustaka pemrogramannya. Hasilnya menunjukkan bahwa MapReduce dapat menyederhanakan antarmuka pemrograman yang dapat mendukung paralelisasi dan distribusi komputasi skala besar secara otomatis. MapReduce bekerja dengan membagi proses menjadi dua fase, yaitu tahap map dan tahap reduce. Seorang programmer dapat memanfaatkan dua fungsi ini bersama fungsi key-value pairs sebagai input dan output untuk mencapai semua fase.

1. ***Schema-Free***

NoSQL dan RDBMS mempunyai perbedaan dalam hal penerapan skema basis data. Dalam RDBMS, sebuah table didesain dengan peraturan skema yang ketat, sedangkan pada NoSQL tidak diharuskan memiliki table, kolom, primary key, foreign key, join, dan relasi. Dalam pengembangan RDBMS, developer/database administrator harus berhati – hati dalam menentukan bagaimana table saling berelasi dan field yang ada di dalam setiap tabel. Karena perubahan skema dapat menimbulkan masalah ketergantungan dan integritas. Permasalahan ini tidak terjadi pada NoSQL karena adanya penerapan schema-free. Setiap dokumen bertanggungjawab terhadap isinya sendiri, null value dapat dihilangkan dalam beberapa baris dan field baru dapat didefinisikan dalam setiap dokumen secara independen.

Manfaat lain dalam penggunaan schema-free adalah penghematan dalam media penyimpanan. Model data schema free artinya setiap baris memungkinkan memiliki nilai sebanyak yang telah didefinisikan dalam tiap fields, dan tidak perlu menggunakan nilai yang memang tidak diperlukan. Kelemahan dalam schemafree adalah memunculkan lemahnya pendefinisian struktur yang memungkinkan terjadinya penggunaan basis data yang tidak konsisten. Jika tujuan pembangunan basis data didasarkan pada aplikasi yang membutuhkan konsistensi ketat, seperti wiki, support management systems, discussion forums, blogs, maka RDBMS masih merupakan pilihan yang tepat.

1. **Horizontal Scaling**

Horizontal Scaling memungkinkan basis data dijalankan pada beberapa server untuk meningkatkan kemampuan perangkat penyimpanan dan meningkatkan efisiensi waktu. Hal ini berarti memungkinkan dilakukannya penambahan server dalam satu jaringan dan user tidak sadar jika terdapat hardware yang diganti dari sisi server. Untuk mencapai hal tersebut, dibutuhkan kemampuan dinamis pemartisian data dalam serangkaian node dalam suatu cluster pada server. Kemampuan untuk meningkatkan kemampuan dengan menambahkan beberapa computer sangatlah penting untuk data yang jumlahnya banyak, karena vertical scaling dilakukan dengan meningkatkan kemampuan spesifikasi single server (missal penambahan prosesor, memori, dan peralatan penyimpanan) terbatas dan mahal.

Ada beberapa teknik partisi yang digunakan dalam basisdata untuk melakukan horizontal scaling, salah satunya adalah consistent. Kunci dari penerapanconsistent hashing adalah membuat suatu lingkaran atau “ring”.Setiap node dalam sistem yang ditandai dengan random value dalam suatu space yang merepresentasikan posisi dariring. Suatu kunci item ditandai dengan sebuah node untuk memperoleh posisinya pada ring, kemudian berpindahmenuju node selanjutya yang sudah ditandai. Node memerankan peranan penting sebagai koordinator untuk kunciyang akan digunakan dalam route request. Kemudian, setiap node menjadi tanggung jawab dari daerah yang ada diring diantara noder dan node dalam ring sebelumnya.

NoSQL merupakan basis data non-relasional dengan schema-free yang memunculkan pertanyaan bagaimana NoSQL dapat melakukan partisi untuk data yang berukuran besar, melakukan query, replikasi data, dan mendukung adanya konsistensi. NoSQL memiliki teknik dan pola pengelolaan data, yaitu column-oriented, documentoriented, object-oriented, graph-oriented.

Berikut adalah penjelasan dari pola tersebut, antara lain :

1. **Column-oriented**

Terdapat beberapa contoh penerapan column-oriented, salah satunya adalah Cassandra yang menggunakan distribusi multidimensional map indexed dengan sebuah key. Baris kunci yang sering digunakan adalah string dengan panjang 16- 36 byte. Setiap kolom digabungkan menjadi sebuah column families. Contoh lainnya diterapkan oleh Wei Kang, dimana suatu kolom atomic dari informasi didukung oleh Cassandra yang diekspresikan dengan nama value. Super-column merupakan gabungan dari kolom dengan nama yang umum dan digunakan untuk pemodelan tipe data yang kompleks. Baris secara unik mengidentifikasikan data yang terdapat dalam column dan super-column. Dalam Cassandra, baris dapat dikenali dengan sebuah kunci.

Column Family merupakan bagian dari suatu unit abstraksi yang berisi baris kunci yang tergabung dalam column dan super column yang memiliki struktur data yang tinggi. Keyspace merupakan level tertinggi dari unit informasi yang terdapat dalam Cassandra. Kumpulan column-families sebenarnya merupakan subordinat dari satu keyspace. Pada intinya, model data column-oriented memungkinkan suatu aplikasi secara bebas untuk mengembangkan bagaimana informasi disusun berdasarkan desain schema.

1. **Document-oriented**

Contoh penerapan model data document-oriented terdapat dalam CouchDB yang dibangun oleh IBM. Basis data yang memiliki model data document-oriented sangat bermanfaat untuk suatu domain yang bentuk masukannya dokumen yang tidak terstruktur seperti web pages, wikis, discussion forums, dan blogs. Data tersimpan dalam basis data CouchDB yang mencakup serangkaian dokumentasi yang berisi beberapa atribut dan nilai dengan masing-masing id yang unik dan metadata. CouchDB tidak pernah melakukan overwrite document, melainkan menambahkandokumen baru ke basis data bila diperlukan seperti ketika terjadi proses update.

1. **Object-oriented**

Basis data berorientasi objek adalah model basis data dimana informasi direpresentasikan dalam bentuk objek yang digunakan dalam pemrograman berorientasi obyek. OrientDB adalah contoh dari basis data object-oriented. Dokumen OrientDB yang ada dalam suatu cluster, dapat berupa fisik, logis atau in-memory, yang digunakan untuk menyimpan link ke dalam data. Cluster adalah cara yang sangat umum untuk mengelompokkan record, hal ini merupakan suatukonsep yang tidak ada dalam basis data relasional. Cara ini dapat mengelompokkan semua record pada jenis tertentu,atau dengan nilai-nilai tertentu.

OrientDB menggunakan segmen data untuk menyimpan isi record. Segmen datamirip dengan file physical cluster yang menggunakan dua atau lebih file, yaitu satu atau beberapa file dengan ekstensi"oda" (Orient Data) dan hanya satu file dengan ekstensi "odh" (Orient data Holes).

1. **Graph-oriented**

Basis data grafik (GraphDB) adalah basis data yang menggunakan struktur grafik yang berisi node, edge, dan properti untuk mewakili dan menyimpan informasi. GraphDB diperlukan untuk data grafik yang berskala besar, terutama yangdipergunakan oleh para peneliti biologi jaringan dan situs jaringan sosial, seperti Facebook, dan Twitter. GraphDBmemetakan secara langsung objek ke aplikasi dan lebih intuitif untuk menggambarkan dataset asosiatif.

Beberapa keuntungan dari GraphDB adalah:

1. Intuitive, dimengerti oleh pikiran manusia, yaitu menggambarkan entitas dan hubungan sebagai grafikmasalah umum yang akrab dengan manusia
2. Elemental untuk ilmu komputer, yaitu grafik,terutama grafik pohon (seperti binary-tree, B+ tree, red-black tree) berfungsi sebagai struktur data dasar dalamilmu komputer dan berbagai masalah (shortest path dan max-flow) dapat diubah dan diselesaikan dengan algoritma grafik.
3. Ubiquitous, yaitu pemodelan ER ke model jejaring sosial selalu dikelilingi oleh grafik baik di komputer ataupun dalam kenyataan.

## 3. Kesimpulan

NoSQL adalah sistem manajemen database non-relasional, pengambilan informasi yang cepat dari database dan portabel . NoSQL pada dasarnya berasal dari sistem database RDB . Database ini  biasanya berinteraksi dengan sistem operasi UNIX . Database NoSQL database adalah mereka yang non - relasional ,open source , didistribusikan di dunia,  serta memiliki kinerja tinggi dengan cara linear yang terukur .

            NoSQL merupakan paradigma baru dalam Sistem Manajemen Basis Data yang memiliki keunggulan dalam hal scalability dan kecepatan dalam pencarian data. Terdapat beberapa teknik pengelolaan data pada NoSQL, document-oriented, columnoriented, graph-oriented, dan object-oriented. MongoDB merupakan salah satu DBMS yang menggunakan teknik pengelolaan data document-oriented. MongoDB tidak menggunakan skema, tabel, dan atribut dalam pengorganisasian basis data, melainkan koleksi, dokumen, dan key-value.

            Karakteristik NoSQL adalah NoSQL tidak menggunakan model data relasional dengan demikian tidak menggunakan bahasa SQL . NoSQL menggunakan volume data yang besar . Dalam lingkungan terdistribusi (data menyebar ke mesin yang berbeda ) , kita menggunakan NoSQL tanpa ketidaksesuaian .  Jika ada kesalahan atau kegagalan ada di setiap mesin , maka dalam hal ini tidak akan ada penghentian pekerjaan . NoSQL adalah database open source, yaitu kode sumbernya tersedia untuk semua orang dan bebas menggunakannya tanpa overhead . Memungkinkan data NoSQL untuk menyimpan dalam catatan yang tidak memiliki apapun skema tetap. NoSQL tidak menggunakan konsep ACID properti .  NoSQL adalah horizontal scalable menyebabkan kinerja tinggi dalam cara linear . Hal ini memiliki struktur yang lebih fleksibel.

4. Saran

Dalam pengolahan data yang besar, dibutuhkan proses data yang terdistribusi (distributed processing). Relational database sudah tidak cukup tepat dalam distributed processing jika melibatkan data yang sangat besar tersebut. Sehubungan dengan hal tersebut, teknologi database yang terdistribusi (distributed database) mulai popular dikarenakan kemampuannya yang lebih baik daripada teknologi database yang tradisional. Sehingga database non relational lebih rekomendasi untuk data yg besar

## Daftar Pustaka

R. P. Padhy, M. R. Patra, and S. C. Satapathy, “RDBMS to NoSQL: Reviewing Some NextGeneration Non-Relational Database’s,” Int. J. Adv. Eng. Sci. Technol., vol. 11, no. 11, pp. 15–30, 2011.

S. Unal and Y. Zheng, “Distributed Databases : SQL vs NoSQL,” pp. 1–5, 2017.

L. Okman, N. Gal-Oz, Y. Gonen, E. Gudes, and J. Abramov, “Security issuesin NoSQLdatabases,” Proc. 10th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2011, 8th IEEE Int. Conf. Embed. Softw. Syst. ICESS 2011, 6th Int. Conf. FCST 2011, no. December, pp. 541–547, 2011.

P. Sareen, A. Professor, and P. Kumar, “NOSQL Database and Its Comparison With SqQLDatabase,” Int. J. Comput. Sci. Commun. Networks, vol. 5, no. 5, pp. 293–298, 2015.

438-943-1-SM.pdf