

TREND PENGGUNAAN NOSQL UNTUK BASIS DATA NON RELASIONAL

Mega Orina Fitri

Dosen Jurusan Teknik Informatika Fakultas Sains dan Teknologi
Universitas Islam Negeri Alauddin Makassar

ABSTRACT *NoSQL is a non-SQL RDBMS which derived from RDB system, developed by Walter W. Hobbs at RAND Organisation. NoSQL uses UNIX as it's operating system. A NoSQL database is exactly the type of database that can handle the sort of unstructured, messy and unpredictable data that our system of engagement requires. This database can be used as alternative rather than RDBMS and its SQL. This paper explained many things about NoSQL and how to modeling database with NoSQL. NoSQL itself has many advantages : elastic scaling, big data, eliminate DBA, economic, and flexible data model. However, it has challenges as well. The challenges are about the maturity. support, analytic and business intelligence, administration, and expertise.*

Key words : *database, non-relational, nosql, rdbms*

PENDAHULUAN

Basis data relasional, pertama kali diperkenalkan oleh seorang ahli komputer E.F. Codd tahun 1970an, dikelola dengan *software* RDBMS (*Relational Database Management System*). Basis data relasional ini diimplementasikan dalam bentuk tabel-tabel yang saling berelasi satu sama lainnya. Pertengahan tahun 1980an mulai dikenalkan bahasa standard untuk mengelola dan mengakses basis data relasional, yaitu SQL (*Standard Query Language*). Selanjutnya basis data relasional dengan *software* RDBMS yang menggunakan bahasa SQL menjadi sangat populer dan banyak digunakan dalam pengelolaan basis data selama puluhan tahun.

Saat ini dominasi basis data relasional mulai pudar karena munculnya jenis basis data baru yaitu NoSQL. Istilah NoSQL diciptakan tahun 1998, dikembangkan oleh Walter W. Hobbs dari RAND Organisation. NoSQL didefinisikan sebagai generasi baru basis data yang bersifat : *non-relational*, terdistribusi/tersebar, dan *open source*. Basis data dengan NoSQL menyediakan mekanisme yang lebih mudah dan sederhana untuk menyimpan dan mengambil data, dibanding dengan basis data relasional. Ada juga yang menyebut NoSQL sebagai “Not only SQL” (Gery Menegaz, 2012). Ini untuk menekankan bahwa beberapa sistem di NoSQL juga dapat melakukan hal-hal seperti yang biasa dilakukan oleh SQL, bukan hanya SQL yang dapat melakukannya.

KAJIAN TEORI

Apa itu NoSQL ?

NoSQL adalah suatu cara penyimpanan data (*datastore*), dimana cara menyimpan dan mengambil kembali datanya dapat dilakukan dengan cepat, seperti basis data relasional pada umumnya, tetapi tidak berdasarkan relasi matematis antar-tabel seperti pada basis data relasional (David Bolton,2013)

Seperti diketahui SQL lebih dahulu populer karena mempunyai berbagai kelebihan. Diantara kelebihan SQL adalah memiliki dasar matematik yang kuat dan adanya *declarative syntax*. Sampai saat ini pun, walaupun sudah diperkenalkan NoSQL, tetapi SQL masih belum tergantikan. SQL masih banyak digunakan terutama bagi yang sudah memiliki basis data relasional, dengan adanya tabel-tabel yang saling terkait satu sama lainnya.

Basis data NoSQL dapat bekerja lebih cepat dibandingkan basis data relasional dengan SQL nya. Tetapi karena masih merupakan teknologi baru yang masih terus berkembang, NoSQL belum setangguh SQL yang sudah berusia puluhan tahun. Pertumbuhan *website* yang sangat pesat menyebabkan berkembangnya NoSQL karena menjadi alternatif untuk mempercepat akses dibanding menggunakan basis data relasional. Umumnya untuk penyimpanan dan pengambilan kembali data *Key-Values*, dilakukan pada dokumen berukuran besar dengan format XML atau JSON.

NoSQL menjadi populer karena tidak semua data berbentuk relasional., sehingga NoSQL dapat menjadi alternatif untuk menangani dan mengelola data-data tersebut. Contoh dokumen data yang tidak disimpan dalam bentuk relasional dapat dilihat pada : Hadoop/Hbase, Cloudata *Google's big table clone*, Amazone SimpleDB, Facebook, Twitter, Digg, CouchDB, dan LinkedIn (Tim Perdue, 2013). Dari berbagai contoh tersebut, Cloudata dari Google, dengan data yang sangat besar dan *user/customer* yang bersifat *mobile*, merupakan contoh yang familiar dan menarik dari basis data NoSQL.

Alasan penggunaan basis data NoSQL oleh Google sebagai *search engine* terpopuler saat ini, dapat dijelaskan sebagai berikut. Google memerlukan suatu *storage layer* untuk melakukan peng-indeks-an pencarian terbalik (*inverted search index*). Akan tetapi hal tersebut tidak dapat ditangani oleh RDBMS. BigTable berada di bagian atas pada GFS *file system* milik Google, *user/customer* yang *mobile*, dan yang paling penting, terdapatnya ribuan perangkat keras yang menyediakan *speed* dan *redundancy*. Selain itu, akibat data yang berantakan dan tidak terstruktur, *software* tidak dapat menyimpan data dalam bentuk SQL. Dengan berbagai alasan tersebut kemudian Google menggunakan basis data NoSQL.

Mengapa dikembangkan NoSQL?

NoSQL dikembangkan karena adanya berbagai kebutuhan akan, yaitu kebutuhan akan:

1. **Data Store** (Penyimpanan Data): Dimana data disimpan dalam ukuran exabyte. Sebuah exabyte sama dengan satu miliar gigabytes (GB) data. Menurut Internet.com, jumlah tambahan data yang disimpan pada tahun 2006 adalah 161 exabyte. Hanya 4 tahun kemudian pada tahun 2010, jumlah data yang disimpan hampir mencapai 1.000 exabyte yang merupakan peningkatan lebih dari 500%. Dengan kata lain, ada banyak data yang disimpan di dunia dan yang hanya akan terus tumbuh.

2. **Interconnected Data** (Data yang saling berhubungan): sehingga data di seluruh dunia menjadi lebih terhubung. Pembuatan web didukung dengan adanya *hyperlink*, blog memiliki *pingbacks* dan setiap sistem jaringan utama sosial memiliki *tag* yang mengaitkan sesuatu dengan yang lainnya. Sistem utama yang dibangun untuk saling berhubungan.

3. **Complex Data Structure** (Struktur data yang rumit): dalam hal ini berarti NoSQL dapat dengan mudah menangani struktur hirarkis data bersarang. Sedangkan pada SQL pada RDBMS, hal ini rumit dan sulit dilakukan, karena butuh beberapa tabel relasional dengan berbagai macam kunci. Selain itu, ada hubungan antara kinerja dan kompleksitas data. Kinerja dapat menurun dalam RDBMS tradisional seperti yang kita menyimpan sejumlah besar data yang dibutuhkan dalam aplikasi jejaring sosial dan web semantik.

NoSQL memiliki banyak kelebihan, yang diantaranya adalah :

- a. Mudah digunakan pada *conventional load-balanced clusters*
- b. Data yang tetap (Persistent data), tidak sekedar caches yang selalu di-*update*
- c. Dapat dilakukan pengukuran terhadap ketersediaan memori
- d. Tidak ada skema tetap dan memungkinkan migrasi skema tanpa *downtime*
- e. Memiliki *system query* sendiri
- f. Bersifat ACID (*Atomicity, Consistency, Isolation, Durability*) dalam suatu *node* pada *cluster* dan konsisten di semua *cluster* (Robert Rees, 2010).

NoSQL dapat dibagi atas 4 (empat) kategori dasar, yaitu:

1) **Key-values Stores**

Adalah suatu *hash table* yang memiliki sebuah *key* unik dan pointer ke item tertentu dari data. Pemetaan ini biasanya diikuti oleh mekanisme *cache* untuk memaksimalkan kinerja. Contohnya pada Tokyo Cabinet/Tyrant, Redis, Voldemort, Oracle BDB. Jenis aplikasi yang dapat menggunakannya adalah aplikasi *content caching*. Sedangkan contoh penerapan aplikasinya adalah pada pembuatan *forum software*, dimana terdapat sebuah *home-profile page* yang memiliki statistik apa yang telah dilakukan oleh *user/pengguna-nya*, seperti berapa jumlah pesan yang telah dikirim, dll dan juga apa saja 10 pesan terakhir yang dikirim/diterima penggunanya. *Home-profile page* ini membaca dari *key* berdasarkan *user_id* dan me-*retrieve string* JSON yang mewakili semua informasi yang relevan. Hal ini diulangi kembali setiap 15 menit dan dicatat di *key-store*.

2) **Column Family Stores**

Adalah tempat yang dibuat untuk menyimpan dan memproses jumlah data yang sangat besar, yang tersebar di berbagai lokasi. Di sini masih terdapat

beberapa *key*, tapi untuk mengacu ke *multiple columns*. Dalam hal ini, BigTable (model dari Column Family NoSQL milik Google), baris diidentifikasi oleh suatu baris *key* dengan data yang terurut dan disimpan oleh *key* tersebut. Kolom-kolom itu diatur oleh *column family*.

3) Document Databases

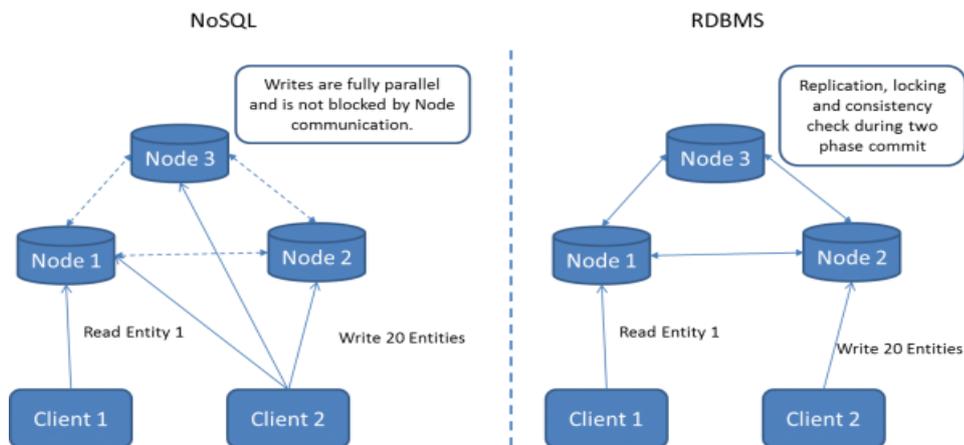
Diinspirasi oleh Lotus Notes dan mirip dengan *key-value stores*. Model ini pada dasarnya merupakan dokumen yang menyimpan kumpulan *key-value collections*. Dokumen yang semi-terstruktur disimpan dalam format seperti JSON. Contohnya adalah: CouchDB dan MongoDB. Jenis aplikasi yang menggunakannya adalah aplikasi web.

4) Graph Databases

Adalah suatu basis data yang dibangun oleh kumpulan *node*, relasi antara *notes* dan karakteristik *node*. Pada basis data ini tidak digunakan tabel yang terdiri dari baris dan kolom serta struktur SQL yang kaku, tetapi memakai model graf yang dapat diterapkan di berbagai mesin (lokasi). Contoh dari basis data graf (*Graph Database*) ini adalah Neo4J, InfoGrid, dan Infinite Graph. Aplikasi yang menggunakannya biasanya jenis aplikasi yang bersifat *social networking* seperti Facebook dan Twitter (Tim Perdue, 2013).

Selain keempat kategori tersebut, menurut seorang ahli yang bernama Robert Rees, 2010, ada tambahan kategori lain dari NoSQL, yaitu Basis data XML, *Distributed Peer Stores*, dan *Object Stores*. Contoh dari basis data XML adalah Exist, Oracle, dan MarkLogic. Basis data XML ini biasanya digunakan pada aplikasi yang bersifat *publishing*, seperti pada perusahaan penerbitan buku, Sedangkan contoh dari *Distributed Peer Stores* adalah Cassandra, HBase, Riak. Cara ini umumnya dipakai pada system yang tersebar (terdistribusi). Dan yang terakhir, contoh dari *Object Stores* adalah : Oracle Coherence, db4o, ObjectStore, GemStone, Polar. Umum *object stores* digunakan pada jenis aplikasi keuangan.

Perbedaan Cara Kerja NoSQL Dengan RDBMS



Gambar 1. Perbedaan NoSQL dengan RDBMS dalam hal pendistribusian entitas

Pada Gambar 1 terlihat bahwa dengan menggunakan NoSQL, data dapat didistribusikan dan secara otomatis melalui *database nodes*, sekaligus juga dapat mencatat data tersebut. Jika kita ingin mencatat dan menyimpan data ke-20 entitas pada suatu *database cluster* dengan 3 *nodes*, maka mungkin saja data tersebut dicatat di semua *nodes*. *Database* tidak perlu melakukan sinkronisasi antar-*node* terlebih dahulu, juga tidak memerlukan 2 fase *commit*, dan *Client* 1 dapat melihat perubahan pada *Node* 1 sebelum *Client* 2 selesai menulis ke-20 entitas. Sedangkan pada RDBMS, hal tersebut tidak dapat dilakukan.

Keuntungan NoSQL

Berbagai keuntungan jika menggunakan NoSQL dalam pengelolaan basis data yaitu:

1. Pola pikir yang fleksibel (*Elastic Scaling*), dimana hal ini dapat dijelaskan sebagai berikut.

Selama berpuluh tahun, DBA (*Database Administrator*) selalu berpikir untuk membeli *server* yang lebih besar kemampuannya karena peningkatan volume basis data, daripada memikirkan penyebaran basis data melalui *multiple hosts* karena peningkatan jumlah beban. Padahal jika terjadi peningkatan transaksi dan permintaan, basis data dapat disimpan tersebar dengan menggunakan teknik *cloud computing* atau dengan kata lain menggunakan lingkungan yang virtual. Hal ini akan lebih ekonomis dibanding membeli server dengan kemampuan yang lebih besar dan canggih. Basis data seperti ini dapat dengan mudah diterapkan dengan menggunakan NoSQL dibanding dengan SQL pada RDBMS. NoSQL didesain untuk menekan biaya *hardware*.

2. Data yang dikelola volumenya besar (*Big Data*), yang disebabkan oleh hal-hal berikut ini.

Transaksi selama berpuluh tahun, yang semakin lama semakin pesat pertumbuhan dan perkembangannya, mengakibatkan volume data yang meningkat tajam pula. O'Reilly menyebut hal ini sebagai suatu "revolusi industri dari data". Ketidakmampuan RDBMS menangani peningkatan volume data yang sangat besar akibat pertambahan jumlah transaksi yang pesat, tidak dapat ditolerir lagi oleh perusahaan-perusahaan besar. Sehingga perusahaan-perusahaan tersebut beralih menggunakan system pengelolaan basis data NoSQL, seperti Hadoop, yang dapat menangani jumlah data yang sangat besar dengan mudah.

3. DBA (*Database Administrator*) berkurang peranannya pada sistem basis data dengan NoSQL.

Seperti diketahui, walaupun selama bertahun-tahun pada *vendor* mengklaim sudah melakukan berbagai perbaikan, tetap saja sistem RDBMS tercanggih sekalipun memerlukan para DBA yang terlatih. Sedangkan pada NoSQL, peranan DBA menjadi berkurang karena NoSQL memang didesain untuk mengurangi peranan manusia seperti perbaikan otomatis, adanya penyebaran/distribusi data sehingga data yang harus dikelola tidak terlalu besar, dan model datanya lebih sederhana. Tetapi untuk data yang kritis, tetap diperlukan DBA.

4. Lebih ekonomis, karena sistem basis data NoSQL biasanya menggunakan server yang murah.

Sedangkan RDBMS cenderung menggunakan server dan berbagai peralatan yang mahal. Hal ini menyebabkan biaya yang dibutuhkan untuk menyimpan dan memproses data pada NoSQL jauh lebih murah dibanding RDBMS.

5. Model data fleksibel pada NoSQL akan memudahkan jika harus mengalami perubahan data.

Pada RDBMS, hal ini tidak dapat dilakukan sembarangan, harus sangat hati-hati. Perubahan sedikit saja pada model data dapat menyebabkan berkurangnya kinerja pelayanannya. Karena itu model data untuk RDBMS dianggap tidak fleksibel. Berbeda halnya dibanding dengan NoSQL.

Tantangan Pada NoSQL

Walaupun banyak yang antusias terhadap NoSQL, tetapi pada kenyataannya masih terdapat hambatan dalam penggunaannya. Berikut ini adalah tantangan dan hambatan pada NoSQL.

1. Dalam hal kesiapan (*Maturity*), karena merupakan metode baru dalam pengelolaan basis data.

Sedangkan sistem RDBMS sudah berjalan berpuluh-puluh tahun, sehingga lebih matang dan lebih siap mengatasi berbagai hambatan dan tantangan. NoSQL masih terhitung sedikit yang menggunakan dibanding RDBMS. Bagi para pendukung NoSQL, *maturity* hanya merupakan tanda mulai kadaluarsanya suatu produk. Tetapi kebanyakan CIOs menganggap bahwa *maturity* RDBMS merupakan jaminan. Sebagian besar RDBMS stabil dan banyak memiliki fungsi/fitur. Sebaliknya, NoSQL sejauh ini NoSQL masih merupakan produk versi *pre-production* dan banyak fitur penting yang belum diimplementasikan.

2. Masalah dukungan (*support*) bagi perusahaan yang menggunakannya.

Perusahaan-perusahaan menginginkan adanya jaminan bahwa sistem yang penting tidak akan gagal, bekerja sesuai waktunya dan kompeten. Dalam hal ini semua *vendor* RDBMS memberikan apa yang diinginkan perusahaan-perusahaan tersebut. Sebaliknya, NoSQL merupakan proyek *open-source*, dan walaupun ada perusahaan yang memberikan dukungan, tetapi biasanya perusahaan kecil dengan jangkauan global yang kecil pula, kurang dukungan sumber daya, dan tidak ada kredibilitas dari Oracle, Microsoft, atau IBM.

3. Analisis dan *Business Intelligence* (BI)

Basis data NoSQL telah berkembang untuk memenuhi tuntutan aplikasi modern Web 2.0. Akibatnya sebagian besar fitur NoSQL berorientasi untuk memenuhi tuntutan tersebut. Di lain pihak aplikasi yang banyak berperan dalam bisnis memiliki data yang melebihi siklus *insert-read-update-delete* yang merupakan ciri khas dari aplikasi Web. Bisnis dapat menghasilkan informasi di

basis data perusahaan untuk meningkatkan efisiensi dan daya saing, dan *Business Intelligent* (BI) memegang peranan penting bagi perusahaan-perusahaan menengah dan besar. Basis data NoSQL menawarkan berbagai fasilitas untuk *query* dan analisis. Sebuah *query* sederhana sekalipun memerlukan keahlian tertentu, dan umumnya BI *tools* tidak menyediakan koneksi ke NoSQL. Sebagai solusinya, diberikan fungsi HAVE atau PIG yang dapat mempermudah akses ke data di *cluster* Hadoop dan juga basis data NoSQL lainnya.

4. Masalah administrasi

Dimana tujuan utama dari NoSQL adalah memberikan solusi “No Admin” atau “Tanpa Admin”. Kenyataannya saat ini adalah NoSQL memerlukan banyak keahlian untuk menginstal dan *maintain*-nya. Ini tentunya akan menjadi tantangan/hambatan pada pengembangan NoSQL.

5. Keahlian

Sekarang ini terdapat jutaan *developer* di seluruh dunia, dan hampir di setiap segmen bisnis menggunakan konsep RDBMS. Hal sebaliknya terjadi pada NoSQL. Kebanyakan yang menggunakan NoSQL masih pemula, dalam tahap belajar, sehingga lebih sulit menemukan seseorang yang ahli dalam bidang NoSQL dibanding dengan ahli RDBMS yang notabene lebih mudah ditemukan. Dan masalah keahlian ini juga menjadi hambatan tersendiri pada NoSQL (Guy Harrison,2010)

Selain itu hampir tidak ada batasan pada model data yang menggunakan NoSQL. NoSQL *Key Value Stores* dan basis data dokumen mengizinkan aplikasi menyimpan secara *virtual* pada elemen data, struktur apapun yang diinginkan . Bahkan dalam pendefinisian BigTable yang menggunakan NoSQL (seperti Cassandra, HBase) memungkinkan adanya penambahan kolom baru tanpa banyak masalah ataupun persyaratan. Dengan demikian tidak perlu suatu unit khusus yang menangani perubahan basis data ataupun skemanya, dan iterasi pun menjadi lebih cepat.

Masa Depan NoSQL

Organisasi atau perusahaan yang memiliki kebutuhan penyimpanan data yang besar sedang serius mendalami NoSQL. Tetapi perusahaan kecil belum tertarik menggunakan konsep ini. Hal ini disebabkan karena mereka masih dapat mengandalkan RDBMS. Selain itu kebutuhan untuk penyimpanan data mereka tidak terlalu besar.

Pada survei yang dilakukan oleh *Information Week*, 44% dari para profesional di bidang bisnis IT belum pernah mendengar mengenai NoSQL. Hanya 1% responden yang mengatakan bahwa NoSQL sudah menjadi bagian dari strategi mereka. Dengan demikian, terlihat bahwa sebenarnya NoSQL telah menjadi bagian dari bisnis IT di dunia ini, tetapi masih harus terus dikembangkan sehingga banyak yang tertarik untuk menggunakannya. Jadi masa depan NoSQL cukup menjanjikan, sebagai alternatif untuk hal yang sulit ditangani oleh RDBMS.

KESIMPULAN

Penyimpanan data pada tabel-tabel dan *spreadsheet* masih menjadi cara favorit dalam pemodelan data. RDBMS dengan SQL-nya masih belum dapat tergantikan. Sampai saat ini sebagian besar ahli basis data professional masih bekerja dalam bentuk basis data relasional.

NoSQL sebagai basis data non-relasional, menawarkan cara berpikir yang berbeda dalam penyimpanan dan pengelolaan data. Basis data NoSQL mulai berkembang menjadi bagian dari dunia basis data. Jika NoSQL digunakan secara tepat, akan memberikan keuntungan. Tetapi perusahaan harus berhati-hati terhadap masalah hambatan/tantangan dan isu-isu yang berkembang pada basis data ini.

Saat ini NoSQL dengan berbagai keterbatasannya belum dapat menggantikan peranan besar dan penting dari basis data relasional (RDBMS). Tetapi masa depan NoSQL cukup menjanjikan. Tidaklah mengejutkan jika suatu saat NoSQL akan dapat mengambil alih peranan RDBMS.

DAFTAR RUJUKAN

- About, <http://cplus.about.com/od/n/g/Definition-Of-Nosql.htm>, 29 Maret 2013.
- About, <http://newtech.about.com/od/databasemanagement/a/Nosql.htm>, 29 Maret 2013.
- Compuware, <http://apmblog.compuware.com/2011/10/05/nosql-or-rdbms-are-we-asking-the-right-questions/>, 30 Maret 2013.
- Gery Menegaz, 2012, <http://www.zdnet.com/what-is-nosql-and-why-do-you-need-it-7000004989/>, 12 April 2013.
- Guy Harrison, 2010, <http://www.techrepublic.com/blog/10things/10-things-you-should-know-about-nosql-databases/1772>, 10 April 2013.
- Robert Rees, 2010, <http://www.thoughtworks.com/insights/articles/nosql-comparison>, 5 April 2013.
- Slashdot, <http://slashdot.org/topic/bi/sql-vs-nosql-which-is-better/>, 5 April 2013.