

NOSQL TECHNOLOGY IN ANDROID BASED MOBILE CHAT APPLICATION USING MONGODB

Yovita Tunardi; Rita Layona

Computer Science Department, School of Computer Science, Binus University
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
ytunardi@binus.edu, rlayona@binus.edu

ABSTRACT

Along with the development of data storage technology which previously using relational concept began to change to the non-relational concept or sometimes referred to as the term NoSQL technology. Data storage in NoSQL is no longer based on the relations between tables but using another methods, one of them is document-oriented. This method was applied specifically in MongoDB. This concept brings new hope because of their superiority that can handle very large data with promising performance and is perfect for agile system development. The purpose of this research was to measure the performance NoSQL, especially MongoDB with implemented it in a Android based mobile chat application. This research uses three methods, analysis method including literature study, analysis of similar application, questionnaires, design method using Agile Software Development, and evaluation method including eight golden rules, analysis of similar application, questionnaires, and interview. The results of this research is a mobile chat application that uses MongoDB as the data storage technology. Through this research can be drawn the conclusion that NoSQL technology implementation, MongoDB, give special advantages like lighter data storage and faster data access.

Keywords: NoSQL, Android, Chat, MongoDB

ABSTRAK

Seiring dengan berkembangnya teknologi penyimpanan data yang sebelumnya bersifat relational mulai berubah ke arah non-relational atau yang sering dikenal dengan istilah teknologi NoSQL. Penyimpanan data pada NoSQL tidak lagi didasarkan pada relasi antar tabel tapi menggunakan metode lain yang salah satunya adalah document-oriented. Metode tersebut diaplikasikan secara khusus pada MongoDB. Konsep tersebut membawa harapan baru karena keunggulannya yang dapat menangani data yang sangat besar dengan performa yang menjanjikan dan sangat cocok untuk pengembangan sistem yang agile. Tujuan dari penelitian ini adalah mengukur performa dari NoSQL khususnya MongoDB dengan cara mengimplementasikannya ke dalam mobile chat application yang berbasis Android. Metode penelitian yang digunakan terdiri dari tiga yaitu metode analisis yang meliputi studi literatur, analisis aplikasi sejenis, dan kuesione, metode perancangan yang menggunakan metode Agile Software Development, dan metode evaluasi yang meliputi evaluasi berdasarkan delapan aturan emas, aplikasi sejenis, kuesioner, dan wawancara. Hasil dari penelitian ini adalah sebuah mobile chat application yang menggunakan MongoDB sebagai teknologi penyimpanan datanya. Melalui penelitian ini dapat ditarik simpulan bahwa penerapan teknologi NoSQL khususnya MongoDB memberikan keunggulan khusus dalam hal penyimpanan data yang lebih ringan dan pengaksesan data yang lebih cepat.

Kata kunci: NoSQL, Android, Chat, MongoDB

PENDAHULUAN

Teknologi penyimpanan data selalu berkembang sesuai dengan zaman dan kebutuhan. Mulai dari penyimpanan dengan hierarchical database yang berbentuk seperti sebuah struktur pohon yang saling terkait, *network database* yang terdiri dari *record* yang saling berhubungan dengan menggunakan *pointer*, sampai *relational database* yang sudah cukup modern di mana data sudah tertata dengan rapi dalam bentuk tabel yang saling berhubungan satu sama lain. Teknologi *relational database* sangat sering digunakan dalam pengembangan aplikasi karena keunggulannya yang terstruktur dan memungkinkan dimanipulasi dengan mudah. Namun, seiring dengan perkembangan zaman, teknologi tersebut masih dirasa kurang karena data yang semakin banyak dan data harus dapat diakses dengan cepat sehingga kemudian muncul teknologi baru yaitu non-relational database atau yang sering dikenal dengan istilah database NoSQL.

Penyimpanan data pada NoSQL tidak lagi didasarkan pada relasi antar tabel tapi menggunakan metode lain yaitu *key-value*, *big table*, *document-oriented*, dan *graph*. Salah satu jenis dari teknologi NoSQL adalah MongoDB yang berbasis pada metode *document-oriented*. NoSQL dan MongoDB secara khusus membawa harapan baru bagi masyarakat karena keunggulannya yang dapat menangani data yang sangat besar dengan performa yang menjanjikan dan sangat cocok untuk pengembangan sistem yang *agile* (Tiwari, 2011). Menurut Idrees (2012), ada beberapa fitur utama dari MongoDB yang membuatnya menjadi pilihan yang baik, yaitu penanganan data yang sangat besar, mendukung operasi *asynchronous insert*, dan pengolahan data dengan MapReduce sehingga MongoDB dijadikan pilihan *database* yang unggul dalam segi kecepatan dan besar penyimpanan.

Oleh karena itu, pada penelitian ini akan diteliti bagaimana NoSQL, khususnya MongoDB, dapat diimplementasikan dalam bentuk aplikasi yang dapat menjawab kebutuhan dari masyarakat tersebut. Aplikasi yang akan dikembangkan sebagai bahan penelitian adalah sebuah *mobile chat application* yang berbasis sistem operasi Android. Aplikasi jenis ini dipilih karena merupakan salah satu aplikasi yang berdasarkan memori penyimpanan yang dimiliki relatif kecil sehingga hanya dapat menyimpan data yang besarnya terbatas dan untuk akses data dibutuhkan cepat. Sistem operasi Android dipilih dalam penelitian ini melihat dari data prediksi market share dari tahun 2013 sampai tahun 2017, Android menempati posisi teratas dengan market share lebih dari 50%.

Tabel 1 Data dan Forecast Market Share Sistem Operasi Smartphone, 2013–2017
(Sumber: www.idc.com, 2013)

Smartphone OS	2013 Market Share	2017 Market Share
Android	75.3%	68.3%
iOS	16.9%	17.9%
Windows Phone	3.9%	10.2%
BlackBerry OS	2.7%	1.7%
Others	1.2%	1.9%
Totals	100.0%	100.0%

Hasil penelitian ini diharapkan dapat memberi hasil gambaran implementasi teknologi NoSQL, secara khusus MongoDB, untuk *mobile chat application* dilihat bukan hanya dari sisi fitur yang disediakan tapi juga dari segi besar penyimpanan data yang dibutuhkan lebih sedikit dan kecepatan akses yang dalam hal ini adalah kecepatan pengiriman dan penerimaan pesan menjadi lebih cepat dibandingkan aplikasi lainnya yang tidak menggunakan NoSQL, khususnya MongoDB.

METODE

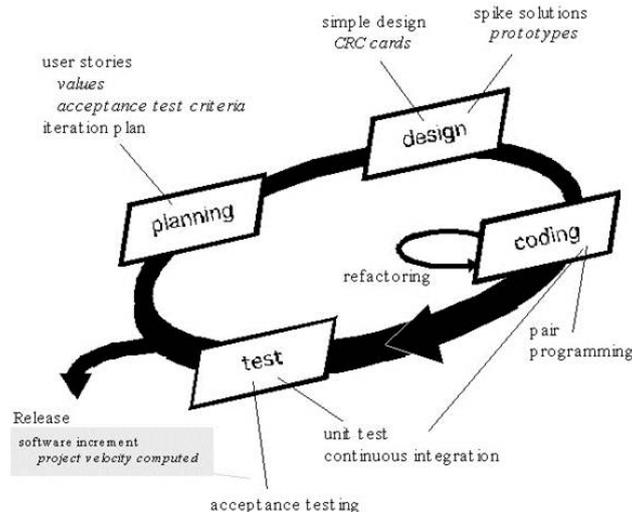
Metodologi penelitian yang digunakan terdiri dari tiga metode utama yaitu metode analisis, metode perancangan, dan metode evaluasi.

Metode Analisis

Metode analisis yang dilakukan dalam penelitian ini terdiri dari studi literatur, analisis aplikasi sejenis, dan kuesioner. (1) Studi literatur – Studi literatur dilakukan sebagai tahap awal sebelum memulai penelitian dengan mengumpulkan teori-teori yang sudah dikemukakan oleh para pakar dalam bentuk buku dan artikel. Selain itu dilakukan juga studi literatur akan penelitian yang sudah pernah dilakukan sebelumnya untuk mendapatkan data metode yang sudah pernah digunakan sebelumnya dan hasil dari penelitian tersebut, yang didapatkan dari paper dan jurnal nasional maupun internasional. Hasil dari studi literatur tersebut dianalisis dan dijadikan acuan awal dalam penelitian ini. (2) Analisis aplikasi sejenis – Analisis aplikasi sejenis dilakukan untuk mendapatkan informasi pendukung dari aplikasi sejenis lainnya sebagai perbandingan. Perbandingan dilakukan berdasarkan besar penyimpanan data dan kecepatan akses. Hal tersebut juga tidak lepas dari fitur-fitur pendukung pada aplikasi yang tentu akan sangat mempengaruhi dua hal tersebut. Aplikasi sejenis yang akan dianalisis adalah Blackberry Messenger, WhatsApp Messenger, dan Line Messenger. (3) Kuesioner – Kuesioner dilakukan dengan cara membagikan kuesioner kebutuhan *user* untuk membantu dalam proses implementasi MongoDB dalam bentuk *mobile chat application* yang sesuai dengan kebutuhan *user* bukan hanya dalam hal fitur tetapi juga dalam penyimpanan data dan kecepatan akses. Analisis ini dilaksanakan dengan menyebarkan kuesioner yang berisi 10 pertanyaan ke pelajar, mahasiswa, dan karyawan. Adapun beberapa hal yang ditanyakan dalam kuesioner tersebut adalah umur, jenis kelamin, profesi, jenis *smartphone* yang dimiliki, cara berkomunikasi, masalah-masalah yang selama ini dirasakan dalam kegiatan *chatting* khususnya dalam penyimpanan data dan kecepatan akses, dan harapan akan fitur lainnya.

Metode Perancangan

Metode perancangan yang digunakan adalah salah satu metode *Agile Software Development* yaitu *Extreme Programming*. *Mobile chat application* dirancang dan dikembangkan melalui empat tahap sesuai dengan metode *agile* yaitu: (1) Tahap Perancangan (*Planning*). (2) Tahap Desain (*Design*). (3) Tahap Pengembangan (*Coding*). (4) Tahap Pengujian (*Testing*). Metode ini dipilih karena berdasarkan teori dan penelitian yang sebelumnya dilakukan, diketahui bahwa NoSQL khususnya MongoDB sangat cocok dan unggul jika diterapkan untuk pengembangan sistem yang *agile*. Metode *agile* ini sendiri juga sangat cocok dalam pengembangan *mobile chat application* karena memungkinkan proses pengembangan aplikasi untuk kembali ke proses sebelumnya (*refactoring*) dan juga menjamin *real-time problem solving* dan *real-time quality assurance* (Pressman, 2010).



Gambar 1 Metode Pendekatan *Extreme Programming*
 (Sumber: Software Engineering: A Practitioners Approach, Pressman, 2010)

Metode Evaluasi

Metode evaluasi yang dilakukan dalam penelitian ini terdiri dari evaluasi berdasarkan delapan aturan emas, aplikasi sejenis, kuesioner, wawancara, lima faktor manusia terukur, dan sepuluh prinsip perancangan *mobile application*. (1) Delapan aturan emas – Aplikasi dievaluasi berdasarkan delapan aturan emas, yaitu konsistensi, fungsi yang universal, umpan balik informatif, dialog untuk penutupan, pencegahan kesalahan sederhana, pengembalian ke aksi sebelumnya, pengendalian internal, dan mengurangi beban ingatan jangka pendek. (2) Aplikasi sejenis – Evaluasi terhadap aplikasi sejenis dilakukan berdasarkan fitur, besar penyimpanan data, dan kecepatan akses dari Blackberry Messenger, WhatsApp Messenger, dan Line Messenger dibandingkan dengan aplikasi yang dikembangkan. (3) Kuesioner – Kuesioner dilakukan dengan cara membagikan kuesioner evaluasi untuk mengevaluasi implementasi MongoDB dalam bentuk *mobile chat application*, bukan hanya dalam hal fitur tetapi juga dalam penyimpanan data dan kecepatan akses. Analisis ini dilaksanakan dengan menyebarkan kuesioner yang berisi 12 pertanyaan ke pelajar, mahasiswa, dan karyawan. Adapun beberapa hal yang ditanyakan dalam kuesioner tersebut adalah mengenai kelengkapan fitur aplikasi dan kecepatan pengiriman pesan. (4) Wawancara – Wawancara dilakukan kepada para pakar dalam bidang *Information Technology* (IT) dan dijadikan sebagai salah satu metode evaluasi aplikasi yang dikembangkan khususnya mengacu pada *database* yang digunakan yaitu MongoDB. Wawancara dilakukan pada 10 responden yang merupakan pakar IT yang meliputi para pekerja IT dan staff pengajar bidang IT. Melalui wawancara diperoleh kelebihan dan kekurangan dari aplikasi, fungsionalitasnya, performa seperti kecepatan dan besar penyimpanannya, desain, dan saran untuk penelitian selanjutnya.

Tinjauan Pustaka

Android

Menurut Darcey dan Conder (2012), Android merupakan sebuah *mobile platform* pertama yang lengkap, *opensource*, dan *gratis* yang dikembangkan dengan menggunakan *Software Development Kit* yang *comprehensive* untuk mengembangkan aplikasi yang *powerful* dan kaya akan fitur. Menurut Gargenta (2011), Android merupakan sebuah *comprehensive open-source platform* yang didesain untuk perangkat *mobile*. Android dimiliki oleh *Open Handset Alliance* dan dipelopori

oleh Google. Menurut Meier (2011), Android adalah gabungan dari tiga komponen, komponen tersebut yaitu: sebuah *open-source operating system* yang digunakan untuk perangkat *mobile*, sebuah *open-source platform* yang digunakan untuk membuat aplikasi *mobile* dan sebuah perangkat, terutama *mobilephone*, yang menjalankan *Android operating system* dan aplikasi yang dibuat di sistem operasi itu.

Keunggulan Android

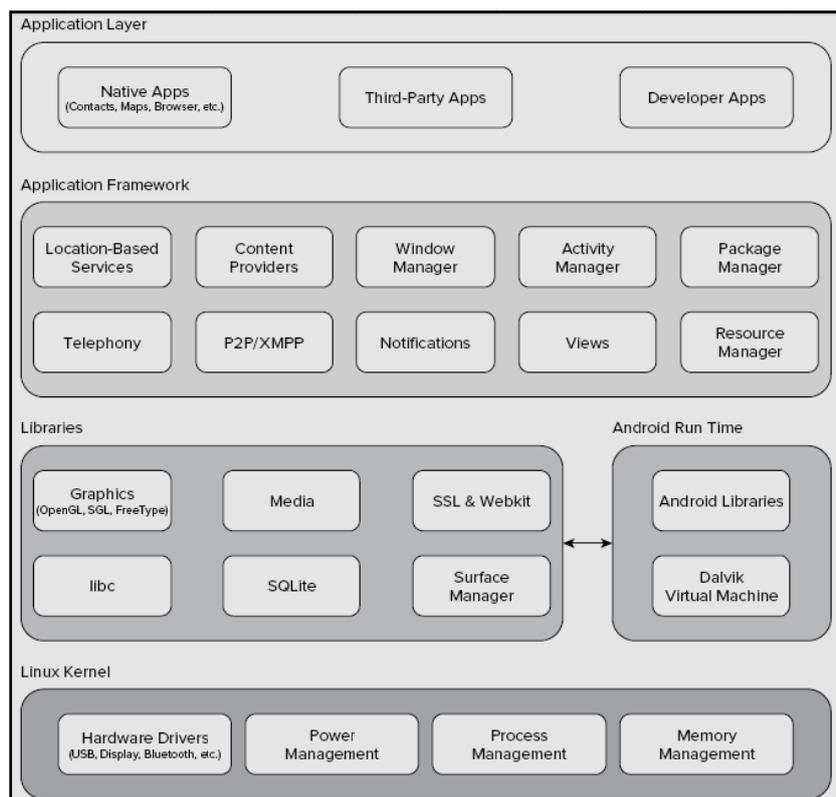
Menurut Meier (2011), terdapat beberapa keunggulan dan fitur-fitur yang dimiliki oleh android Android, yaitu: (1) Aplikasi Google Maps – *Google Maps* yang disediakan untuk *mobile* sangat populer dan Android menawarkan aplikasi *Google Maps* yang dapat digunakan untuk aplikasi lain. (2) Background services and applications – *Background services* memungkinkan pengembang aplikasi untuk membuat sebuah aplikasi yang menggunakan *event*, di mana aplikasi berjalan tanpa terlihat bahwa aplikasi tersebut sedang berjalan ketika aplikasi lain sedang dijalankan. (3) Sharing data dan InterProcess Communication (IPC) – Android memungkinkan pertukaran pesan, melakukan pengolahan, dan berbagi data dengan menggunakan *Content Provider*. (4) Semua aplikasi yang dibuat sama – Android tidak membedakan aplikasi yang asli dengan aplikasi yang dikembangkan oleh pihak ketiga/*third parties*.

Jenis Aplikasi Android

Menurut Meier (2011), sebagian besar aplikasi yang dibuat menggunakan Android akan termasuk dalam salah satu kategori berikut: (1) *Foreground* – *Foreground* merupakan sebuah aplikasi yang berguna ketika ada pada *foreground* dan tidak efektif ketika tidak terlihat. Contohnya adalah *games* dan *map mashups*. (2) *Background* – *Background* merupakan sebuah aplikasi dengan interaksi yang terbatas, Contohnya aplikasi *call screening* dan *Short Message Service (SMS) auto-responder*. (3) *Intermittent* – *Intermittent* merupakan sebuah aplikasi yang dapat jalan di *foreground* dan *background*, aplikasi ini biasanya dibentuk dan kemudian dijalankan secara tersembunyi (*background*) dan memberitahu pengguna di saat yang tepat. Contohnya adalah *media player*. (4) *Widget* – *Widget* adalah aplikasi yang dibuat hanya untuk ditampilkan sebagai *widget* pada *screenhome*.

Menurut Helal *et al.* (2012), terdapat empat jenis komponen yang ada dalam aplikasi Android, Komponen tersebut adalah sebagai berikut: (1) *Activity* – Sebuah *activity* mewakili *user interface* aplikasi yang didefinisikan pada *activity class*. Contoh: pada *music player application* memiliki satu *activity* untuk memutar lagu dan satu *activity* lainnya untuk memilih album atau *audio file*. (2) *Service* – *Service* akan berjalan secara tersembunyi dan tidak secara langsung berinteraksi dengan *user*. Komponen ini berjalan di *background* dan tidak memiliki *user interface*. (3) *Content Provider* – *Content provider* adalah tempat untuk menyimpan dan membagikan data dengan aplikasi lain. (4) *Broadcast Receiver* – *Broadcast receiver* bertugas untuk merespon pengumuman sistem siaran.

Gambar berikut merupakan arsitektur Android yang terdiri dari komponen-komponen utama dari sistem operasi Android yang dibagi menjadi beberapa bagian (*layer*)



Gambar 2 Arsitektur Android
(Sumber : Professional Android 2 Application Development, Meier, 2011)

Komponen-komponen utama dari arsitektur sistem operasi Android adalah: (1) Linux Kernel – Linux Kernel merupakan komponen paling bawah dari arsitektur Android. Linux Kernel menangani *service* inti seperti perangkat keras, proses dan *memory management*, sekuritas, jaringan, dan *power management*. (2) *Libraries* – Libraries berjalan di atas dari komponen Kernel. Android memasukkan berbagai *library* inti dari C / C++ termasuk *libc* dan *SSL*, serta *MediaLibrary* untuk memutar kembali *media* audio dan *video*, *Surface manager* untuk menyediakan pengaturan tampilan, *Graphics libraries* yang mencakup *SGL* dan *OpenGL* untuk 2D dan 3D *graphics*, *SQLite* untuk dukungan pada *database* aslinya serta *SSL* dan *WebKit* untuk integrasi *web browser* dan sekuritas *internet*. (3) *Android run time* – *Android run time* merupakan *engine* yang berfungsi untuk menjalankan aplikasi Android dan bersama dengan *libraries*. (4) *ApplicationFramework* – *Application Framework* menyediakan *class* yang digunakan untuk membuat aplikasi Android. (5) *Application layer* – Semua aplikasi dibuat di *application layer* dengan menggunakan *library API*. *Application layer* berjalan di dalam *Android run time* dengan menggunakan *class-class* dan *service* yang dibuat dari *application framework*.

MongoDB

MongoDB merupakan salah satu jenis *database* NoSQL, yang merupakan sebuah konsep penyimpanan *non-relational data*. Ada empat metode berbeda yang dipakai oleh *database* jenis ini untuk menyimpan data, yaitu *key-value*, *big table*, *document-oriented*, dan *graph*. Metode yang dipakai oleh MongoDB adalah *document-oriented* yang menyimpan dokumen seperti *JavaScript Object Notation* (JSON) (Tiwari, 2011).

Fitur dan Kelebihan MongoDB

Menurut Idrees (2012), ada beberapa fitur utama dari MongoDB yang membuatnya menjadi pilihan yang baik, yaitu sebagai berikut: (1) Penanganan data yang sangat besar – Sistem *database* relasional mulai menjadi terlalu mahal dalam hal sumber daya sistem. Data yang sangat besar memerlukan pengolahan yang lebih dan akan menghabiskan banyak waktu dan juga tempat. MongoDB dapat menjadi alternatif yang lebih baik. (2) Mendukung operasi *asynchronousinsert* – Cara kerja dari MongoDB dalam proses *insert* adalah dengan memasukkan dokumen dan melanjutkan ke tugas berikutnya tanpa menunggu respon dari *server*. Hal tersebut membebaskan aplikasi untuk melakukan tugasnya tanpa terjebak pada satu operasi *database* yang lama dan dapat meningkatkan respons pengguna. (3) Pengolahan data dengan *MapReduce* – *MapReduce* adalah sebuah pendekatan untuk pengolahan data yang memiliki dua manfaat dibanding solusi tradisional lainnya, yaitu performa yang lebih baik dan penulisan *realcode* untuk mengolah data. *MapReduce* juga membuat operasi *aggregate* dan *query* data menjadi lebih mudah.

Selain fitur utama tersebut, menurut Jayathiloha *et al.* (2012) terdapat beberapa alasan MongoDB dijadikan pilihan dibanding produk *database* NoSQL lainnya, yaitu: (1) *Schema free* – MongoDB tidak memiliki *schema* sehingga biasa disebut *schema-free database*. Hal tersebut memungkinkan penyimpanan data dengan struktur yang berbeda dalam satu *collection*. (2) *Agile development* – MongoDB merupakan pilihan yang sangat tepat untuk pengembangan *software* secara *Agile*. MongoDB dapat mengatasi banyaknya perubahan dalam *agile development* karena sifatnya yang *schema-free* sehingga *schema* dapat berubah sesuai dengan perubahan persyaratan. (3) *Flexible document* – Penyimpanan data MongoDB berbeda dari *database* lainnya yaitu dalam bentuk dokumen yang dapat berupa *array* dan *hash*. Dokumen MongoDB disimpan sebagai objek JSON yang biasa disebut *binary JSON* (BSON). (4) *Cloud ready* – MongoDB siap untuk dijalankan pada *commodity hardware*, *virtualized environments*, dan *cloud*. MongoDB dapat berjalan dengan baik pada *hardware* jenis apapun. (5) *High performance* – MongoDB tidak menggunakan *join* seperti *database* umumnya karena MongoDB dapat berisi gabungan *field* seperti *array* dan *hash*. MongoDB juga mendukung *indexing* yang akan meningkatkan performa dari *query*. (6) *Horizontally scalable* – Berbeda dengan RDMS yang melakukan *vertical scaling*, MongoDB termasuk *horizontally scalable* yang berarti skalabilitas data dimungkinkan dengan menambahkan beberapa *server* dan tidak perlu melakukan *upgrade server*.

Konsep Dasar MongoDB

Seguin (2012) mengatakan bahwa ada enam konsep dasar yang perlu diketahui mengenai MongoDB, yaitu sebagai berikut: (1) MongoDB memiliki konsep yang sama dengan *database* pada umumnya seperti MySQL dan Windows SQL Server. MongoDB dapat memiliki nol atau lebih *database*. (2) Sebuah *database* dapat memiliki nol atau lebih *collection* yang dapat disetarakan dengan *table* pada *database* umumnya. (3) Sebuah *collection* terdiri dari nol atau lebih dokumen yang dapat disetarakan dengan baris pada *database* umumnya. (4) Sebuah dokumen terdiri dari satu atau lebih *fields* yang dapat disetarakan dengan kolom pada *database* umumnya. (5) MongoDB memiliki indeks yang memiliki fungsi sama seperti indeks pada *database* umumnya. (6) Data dari MongoDB akan dikembalikan dalam bentuk kursor.

MongoDB bersifat *client-server*. Sisi *server* sebagai tempat proses data dan untuk penulisan instruksi pada sisi *client*. Pengaksesan *database* dilakukan dengan instruksi *use DATABASE_NAME* seperti *use learn*. Instruksi tersebut akan mengakses *database learn* yang sudah ada atau membuat baru jika belum ada ketika proses *insert* pertama. Instruksi yang akan dieksekusi pada *database* dituliskan dengan menggunakan objek *db* seperti *db.help()* atau *db.stats()*. Instruksi yang akan dieksekusi pada *collection* dituliskan dengan menggunakan objek *db.COLLECTION_NAME* seperti *db.unicorns.help()* atau *db.unicorns.count()*.

Select Data

Menurut Seguin (2012), Select Data menampilkan data pada *database* MongoDB dapat dilakukan dengan menggunakan instruksi *find*, misalnya `db.unicorns.find()`. Instruksi *find* juga dapat memiliki *parameter* tambahan untuk pemberian syarat-syarat khusus dan pemilihan *field* yang ditampilkan.

Insert Data

Menurut Seguin (2012), proses *insert* data pada MongoDB dapat dilakukan dengan menggunakan instruksi *insert* atau *save*. Penggunaan *insert* akan langsung menambahkan data baru, sedangkan penggunaan *save* akan menyesuaikan dengan kondisi. Jika data yang ditambahkan belum ada, maka data akan ditambahkan, tetapi jika sudah ada maka data akan diubah. Sebagai contoh dapat dilihat seperti di bawah ini:

```
db.unicorns.insert({name: 'Aurora', gender: 'f'})
object={name: 'Aurora'}
db.unicorns.save(object)
```

Update Data

Menurut Seguin (2012), proses *update data* pada MongoDB masih sering digunakan dengan tidak tepat karena memiliki konsep yang jauh berbeda dengan *database* pada umumnya seperti SQL. Instruksi yang digunakan adalah *update* dengan 2 *parameter*, syarat seleksi yang digunakan dan *field* mana yang akan diubah. Berikut adalah contoh penggunaannya:

```
db.unicorns.update({name: 'Rooodooles'}, {weight: 590})
```

Instruksi tersebut akan menyebabkan *replace* data bukan *update*. Dokumen yang memiliki name `Rooodooles` akan diganti datanya dengan `weight 590`. Perubahan data untuk satu atau lebih *field* yang bukan bersifat *replace* dilakukan dengan menggunakan operator `$set`. Berikut adalah cara penggunaannya:

```
db.unicorns.update({weight: 590}, {$set: {name: 'Rooodooles', dob: new Date (1979, 7, 18, 18, 44), loves: ['apple'], gender: 'm', vampires: 99}})
```

Delete Data

Menurut Seguin (2012), menghapus data pada MongoDB dapat dilakukan dengan menggunakan instruksi *remove*. Seperti instruksi *find* dan *update*, *remove* juga dapat memiliki *parameter* sebagai syarat data mana yang akan dihapus. Berikut adalah cara penggunaannya:

```
db.unicorns.remove()
db.unicorns.remove({name: 'Rooodooles'})
```

Berbeda lagi untuk penghapusan *collection* dari *database*. Hal tersebut dapat dilakukan dengan instruksi `db.COLLECTION_NAME.drop()` menjadi `db.unicorns.drop()`.

Data Modelling

Menurut Seguin (2012), MongoDB tidak menggunakan *JOIN* seperti yang biasa dimiliki oleh *database* pada umumnya, tetapi dapat tetap dihubungkan secara *manual* seperti penggunaan *foreign key* di SQL. Langkah yang dapat dilakukan adalah dengan menyimpan `_id` dari satu dokumen dalam dokumen yang lain. Contohnya adalah sebagai berikut:

```

db.employees.insert(
{
  _id:ObjectId("4d85c7039ab0fd70a117d730"),
  name: 'Leto'
})
db.employees.insert(
{
  _id: ObjectId("4d85c7039ab0fd70a117d731"),
  name: 'Duncan',
  manager: ObjectId("4d85c7039ab0fd70a117d730")
})

```

Pilihan lainnya adalah dengan menggunakan DBRef yang sudah disediakan oleh MongoDB. DBRef dapat digunakan untuk menghubungkan antara lebih dari satu dokumen dalam *collection* atau *database* yang berbeda. DBRef memiliki beberapa *field* sebagai berikut: (1) **\$ref** – *Field* \$ref akan berisi nama *collection* dimana dokumen yang akan dihubungkan berada. (2) **\$id** – *Field* \$id akan berisi nilai *_id* dari dokumen yang akan dihubungkan. (3) **\$db** – *Field* \$db bersifat *optional* dan akan berisi nama *database* dimana dokumen yang akan dihubungkan berada.

Dokumen DBRef akan terbentuk menjadi { \$ref : <value>, \$id : <value>, \$db : <value> }.

Fitur Tambahan

Menurut Seguin (2012), dalam MongoDB juga disediakan beberapa fitur tambahan lagi yang mendukung performa. Berikut akan dibahas beberapa hal yang akan sering digunakan pada *database* MongoDB:

Indexes

Indeks pada MongoDB berfungsi sama dengan indeks pada *database* relasional pada umumnya yang membantu performa *query* dan *sorting* data. Penambahan indeks dapat dilakukan dengan menggunakan instruksi *ensureIndex*. Penghapusan indeks dapat dilakukan dengan menggunakan instruksi *dropIndex*. Berikut adalah contoh penggunaannya:

```

db.unicorns.ensureIndex({name: 1});
db.unicorns.dropIndex({name: 1});

```

Pembuatan *unique index* dapat dilakukan dengan menambahkan *parameter* kedua. Pengurutan indeks secara *ascending* ditandai dengan angka 1 dan secara *descending* ditandai dengan angka -1 seperti pada contoh sebagai berikut:

```

db.unicorns.ensureIndex({name: 1}, {unique: true});
db.unicorns.ensureIndex({name: 1, vampires: -1});

```

Explain

Pada MongoDB dapat dilakukan pengecekan apakah suatu *collection* dan *query* menggunakan indeks. Instruksi yang digunakan adalah *explain*. Penggunaan instruksi *db.unicorns.find().explain()* akan menampilkan kursor yang digunakan adalah *BasicCursor* yang menandakan tidak adanya penggunaan indeks. Penggunaan instruksi *db.unicorns.find({name: 'Pilot'}).explain()* akan menampilkan kursor yang digunakan adalah *BtreeCursor* adanya penggunaan indeks.

Backup dan Restore

Backup dan *restore* pada MongoDB dapat dilakukan dengan menggunakan *fileexecutable* yang sudah disediakan oleh MongoDB. *Backup* dilakukan dengan menjalankan *mongodump* dengan diikuti pilihan *database*, *collection*, dan nama *filebackup*. Sedangkan untuk *restore* dengan menjalankan *mongorestore* dengan pilihan yang sama. Berikut adalah contoh penggunaannya:

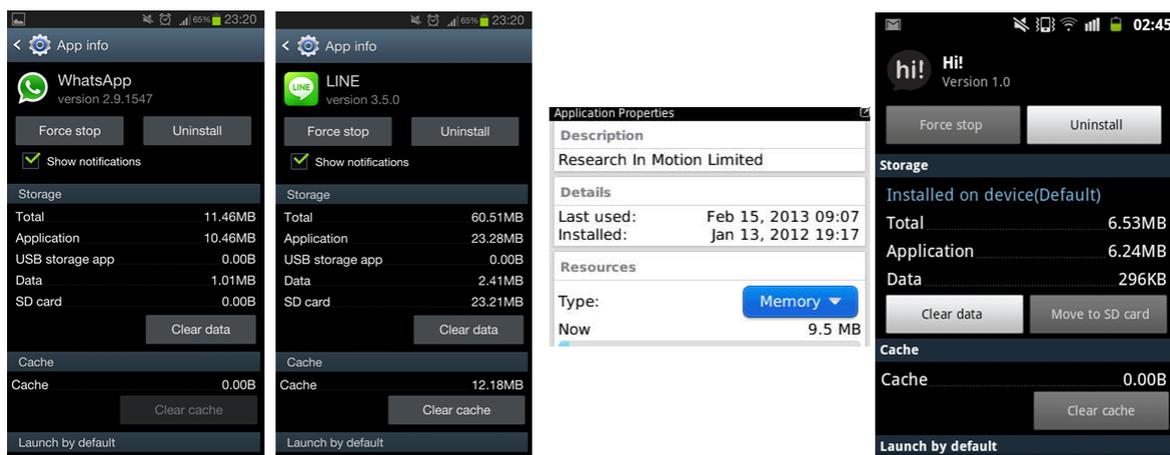
```
mongodump --db learn --collection unicorns --out backup
mongorestore --collection unicorns backup/learn/
unicorns.bson
```

Selain *backup* dan *restore*, MongoDB juga memungkinkan untuk proses *import* dan *export*. *Fileexecutable* yang digunakan adalah *mongoimport* dan *mongoexport*. Berikut adalah contoh penggunaannya:

```
mongoexport --db learn -collection unicorns
mongoexport --db learn -collection unicorns --csv -fields name,weight,vampires
```

HASIL DAN PEMBAHASAN

Hasil dari penelitian ini didapat bahwa dengan menggunakan MongoDB untuk *mobile chat application*, aplikasi ini dapat menangani data yang besar dan memiliki pengaksesan data dan performa yang lebih baik dikarenakan mongoDB memiliki fitur operasi *asynchronous insert* dan pengolahan data menggunakan MapReduce. Berikut merupakan perbandingan memori penyimpanan aplikasi lain dengan aplikasi yang dibuat dengan mongoDb: (1) Memori penyimpanan WhatsApp Messenger adalah sebesar 11.46 MB. (2) Memori penyimpanan Line Messenger adalah sebesar 60.51 MB. (3) Memori penyimpanan Blackberry Messenger adalah sebesar 9.5 MB. (4) Memori penyimpanan Aplikasi dengan MongoDB adalah sebesar 6.53 MB.

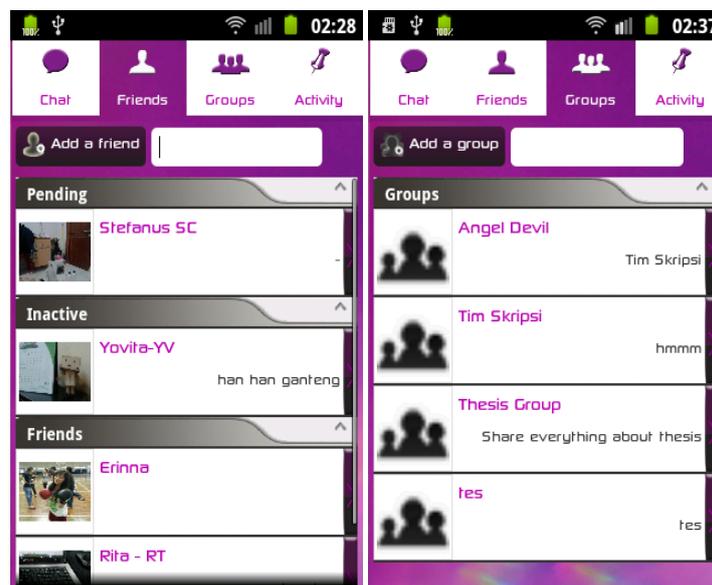


Gambar 3 Perbandingan Memori Penyimpanan

Pada penelitian ini juga dilakukan enam jenis evaluasi untuk *mobile chat application*. Evaluasi tersebut meliputi evaluasi delapan aturan emas, evaluasi analisis aplikasi sejenis, evaluasi melalui kuesioner dan wawancara, evaluasi lima faktor manusia terukur, serta sepuluh prinsip perancangan *mobile application* berdasarkan Stark (2012).

Evaluasi Terhadap Delapan Aturan Emas

Mobile chat application sudah memenuhi delapan aturan emas, yaitu: berusaha untuk konsisten, menyediakan fungsi yang bersifat universal atau umum, memberikan umpan balik yang informatif, merancang dialog untuk menghasilkan penutupan, memberikan pencegahan terhadap kesalahan yang sederhana, memungkinkan pengembalian aksi yang sebelumnya, mendukung pengendalian internal dan mengurangi beban ingatan jangka pendek



Gambar 4 Contoh Konsistensi Aplikasi

Evaluasi Terhadap Aplikasi Sejenis

Analisis dilakukan dengan membandingkan fitur yang dimiliki dengan aplikasi lainnya, seperti Blackberry Messenger, WhatsApp Messenger, dan Line Messenger.

Berdasarkan analisis aplikasi terhadap aplikasi sejenis, *mobile chat application* yang dirancang memiliki kelebihan yang dimiliki yakni dapat menangani data yang besar dan memiliki pengaksesan data dan performa yang lebih baik serta memiliki beberapa fitur tambahan seperti *sharefile* pada *group*, *block broadcast message*, pengaturan status keaktifan *user*, penerjemahan bahasa dari bahasa Inggris ke bahasa Indonesia dan sebaliknya serta pemberian *Feedback*.

Tabel 2 Tabel Analisis Aplikasi Terhadap Aplikasi Sejenis

	Fitur	Blackberry <i>Messenger</i>	WhatsApp <i>Messenger</i>	Line <i>Messenger</i>	Using MongoDB
<i>Private Chat</i>	Pengiriman Gambar	√	√	√	√
	Pengiriman Suara	√	√	√	√
	Pengiriman <i>Video</i>	×	√	√	√
	Pengiriman <i>File</i>	√	×	×	√
<i>Conference Chat</i>	Pengiriman Gambar	√	×	√	√
	Pengiriman Suara	√	×	√	√
	Pengiriman <i>Video</i>	×	×	√	√
	Pengiriman <i>File</i>	√	×	×	√
<i>Group</i>	<i>Sharing</i> Gambar	√	√	√	√
	<i>Sharing</i> Suara	×	√	√	√
	<i>Sharing</i> <i>Video</i>	×	√	√	√
	<i>Sharing</i> <i>File</i>	×	×	×	√
Pengaturan Pesan	<i>Block Broadcast Message</i>	×	×	×	√
	Pengaturan Suara	√	√	√	√
	Pengaturan Getaran	√	√	√	√
Pengaturan Teman	<i>Block Friends</i>	√	√	√	√
	Pengaturan Status Keaktifan <i>User</i>	×	×	×	√
	Penerjemahan Bahasa	×	×	×	√
Lainnya	Pemberian <i>Feedback</i>	×	×	×	√
	<i>Free Call</i>	×	×	√	√
	Pemberian Respons pada Aktivitas <i>User</i>	×	×	√	√
	Mengetahui Lokasi Teman	√	√	√	√
	<i>Emoticon</i>	√	√	√	√

Evaluasi Berdasarkan Kuesioner

Hasil yang diperoleh berdasarkan hasil kuesioner kepada beberapa responden yang mencoba *demo mobile chat application* adalah dapat disimpulkan bahwa responden sudah merasa puas untuk fitur baru, kecepatan, dan *user interface* yang dimiliki. Selain itu juga, dari hasil kuesioner ini, didapatkan komentar & saran untuk pengembangan selanjutnya.

Evaluasi Berdasarkan Hasil Wawancara

Evaluasi dilakukan dengan cara melakukan wawancara terhadap pakar *Information Technology* (IT). Hasil yang didapat adalah jika dilihat dari segi kecepatannya, aplikasi ini sudah memiliki performa yang cukup baik, respon dan pengirimannya cepat, tidak terlalu berat dan belum ditemukan masalah seperti terlalu berat dalam mengirim data. Untuk kedepannya dapat dioptimalisasi lagi untuk memperhatikan faktor eksternal seperti koneksi *internet* dari masing masing *user*.

Evaluasi Terhadap Lima Faktor Manusia Terukur

Aplikasi memenuhi lima faktor manusia terukur, yang meliputi waktu untuk belajar, kecepatan kinerja, tingkat kesalahan *user*, daya ingat, dan kepuasan subjektif

Evaluasi Terhadap Sepuluh Prinsip Perancangan Mobile Application

Aplikasi sudah memenuhi sepuluh prinsip perancangan *mobile application*, yang meliputi: *Mobile Mindset, Mobile Contexts, Global Guidelines, Navigation Models, User Input, Gestures, Orientation, Communications, Launching* dan *First Impressions*

SIMPULAN

Berdasarkan analisis terhadap hasil evaluasi, maka hal-hal yang dapat disimpulkan dari penelitian ini adalah: (1) NoSQL Technology seperti MongoDB dapat digunakan pada *mobile chat application* berbasis Android dengan kecepatan akses yang lebih cepat. (2) *Mobile chat application* ini dikembangkan sesuai dengan delapan aturan emas, memenuhi lima faktor manusia terukur, dan prinsip perancangan *mobile application* berdasarkan Stark (2012).

Berdasarkan simpulan yang diperoleh, maka ada beberapa saran yang dipertimbangkan untuk pengembangan aplikasi selanjutnya yaitu: penggunaan NoSQL Technology seperti MongoDB pada sistem operasi selain Android seperti sistem operasi Blackberry dan IOS (*iPhone Operating System*), serta digunakan untuk pengaksesan data secara *real-time*.

DAFTAR PUSTAKA

- Darcey, L., Conder, S. (2012). *Learning Android™ Application Programming for the Kindle Fire™*. USA: Addison-Wesley.
- Gargenta, M. (2011). *Learning Android*. USA: O'Reilly Media, Inc.
- Idrees, M. (2012). MongoDB. *Software Developer's New Ideas & Solutions for Professional Programmers*, 3-4.
- Jayathiloha, D., Fernando, A., Sooriyaarachchi, C. (2012). Thinking Big to Deal with Big Data – A Practical Insight into MongoDB. *Software Developer's New Ideas & Solutions for Professional Programmers*, 44.
- Meier, R. (2010). *Professional Android™ 2 Application Development*. Indianapolis: Wiley Publishing, Inc.
- Seguin, K. (2012). *The Little MongoDB Book*. San Francisco: GitHub Inc.
- Stark, J. (2012). *10 Principles Mobile Interface Design*. Diakses 1 November 2012, dari <http://www.creativebloq.com/mobile/10-principles-mobile-interface-design-4122910>.
- Tiwari, S. (2011). *Professional NoSQL*. Indianapolis: John Wiley & Sons, Inc.