

MongoDB

(DBMS Document Oriented Database - NoSQL)

Not Only SQL



Basis Data II

Dosen Pengampu :

Noor Ifada

Kelompok →BD2-C6

Asadullohil Ghalib Kubat	09 04 111 00029
Abdul Basith	09 04 111 00095
Miftah Alfiansyah	09 04 111 00096
M.Dannish Koosrivanto	09 04 111 00126

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA
©Juni 2011**

Daftar Isi

Daftar Isi.....	2
Deskripsi Umum	3
Pokok Pembahasan	3
Bab 1 - Tentang MongoDB.....	4
Bab 2 - Kelebihan dan Kekurangan MongoDB	5
Bab 3 – Instalasi MongoDB di OS Windows dan Linux	6
Bab 4 – Query Dasar di MongoDB	10
Bab 5 – Manajemen MongoDB dengan PHPMoAdmin	16
Bab 6 – Dukungan dan Integrasi dengan Bahasa Pemrograman (JAVA dan PHP)	18
Bab 7 – Fitur Khusus dalam MongoDB	26
KESIMPULAN	28
DAFTAR PUSTAKA	29

DESKRIPSI UMUM

Dalam makalah ini kita akan membahas berbagai hal tentang MongoDB meliputi, pengenalan tentang MongoDB sebagai DBMS berbasis dokumen, cara instalasi mongodb di berbagai sistem operasi serta implementasi query pada mongodb. Selain itu dalam makalah ini juga disertakan bagaimana cara mengintegrasikan mongodb dengan beberapa bahasa pemrograman yakni JAVA dan PHP agar para pembaca dapat lebih memahami cara penggunaan mongoDB pada program.

Selain beberapa hal diatas, dalam makalah ini juga kami sertakan beberapa fitur unggulan dari mongoDB yang tidak dimiliki oleh beberapa DBMS yang lain sehingga akan membuat pembaca tertarik untuk mencoba dan menggunakan mongoDB.

POKOK PEMBAHASAN

1. Sekilas Tentang MongoDB
2. Kelebihan dan Kekurangan MongoDB
3. Instalasi MongoDB di Sistem Operasi Windows dan Linux
4. Query Dasar di MongoDB
5. Manajemen MongoDB dengan PHPMoAdmin
6. Dukungan dan Integrasi dalam Bahasa Pemrograman (PHP dan Java)
7. Fitur Khusus yang ada di MongoDB

A. SEKILAS TENTANG MONGODB

Merupakan sebuah sistem basis data yang berbasis dokumen (Document Oriented Database) dan termasuk sistem basis data yang menganut paham NoSQL. NoSQL itu bukan berarti anti menggunakan SQL (No-SQL), tapi NoSQL itu singkatan dari Not Only SQL. Artinya sebuah sistem basis data tidak hanya harus menggunakan perintah SQL untuk melakukan proses manipulasi data.

MongoDB tidak memiliki yang namanya tabel, kolom dan baris. Dalam MongoDB yang ada hanyalah koleksi dan dokumen. Koleksi dalam MongoDB bisa kita anggap sebuah Folder (Directory) dan Dokumen bisa kita anggap berkas (File) dalam Folder (Koleksi) tersebut. Dokumen yang terdapat dalam MongoDB dapat memiliki berbeda atribut dengan dokumen lain walaupun berada dalam satu koleksi. Hal ini tidak dapat dilakukan dalam RDBMS dimana sebuah baris dalam Tabel tidak mungkin memiliki kolom yang berbeda dengan baris yang lain jika berada dalam satu tabel.

MongoDB merupakan sistem basis data yang menggunakan konsep key-value, artinya setiap dokumen dalam MongoDB pasti memiliki key. Hal ini berbeda dalam RDBMS yang kita bisa tidak menggunakan primary key ketika membuat sebuah tabel. Sehingga walaupun kita membuat sebuah dokumen tanpa menggunakan primary key, tapi secara otomatis MongoDB memberinya sebuah key. Penggunaan konsep key-value sangat berperan penting, karena hal ini membuat MongoDB menjadi sistem basis data yang sangat cepat jika dibandingkan dengan non key-value seperti RDBMS.

MongoDB tidak menggunakan bahasa yang biasa digunakan RDBMS (SQL atau PL/SQL). MongoDB menggunakan bahasa BSON, dimana BSON merupakan singkatan dari Binary JSON. Seperti JSON, BSON mendukung embedding dokumen dan dokumen lainnya dalam array dan array.

BSON juga mengandung ekstensi yang memungkinkan representasi tipe data yang bukan merupakan bagian dari spec JSON. Misalnya, BSON memiliki jenis Tanggal dan jenis BinData. BSON dirancang untuk memiliki tiga karakteristik berikut:

1. Ringan (Lightweight)

Menjaga overhead spasial untuk minimum penting untuk format representasi data, terutama bila digunakan melalui jaringan

2. Dilalui Dengan Mudah (Traversable)

BSON dirancang untuk dilalui dengan mudah. Ini adalah properti penting dalam perannya sebagai representasi data utama untuk MongoDB.

3. Efisien (Efficient)

Encoding data untuk BSON dan decoding dari BSON dapat dilakukan sangat cepat dalam bahasa paling karena penggunaan jenis C data.

Bahasa BSON memiliki struktur bahasa yang hampir sama dengan bahasa JavaScript. Jadi, ketika sudah terbiasa menggunakan JavaScript, pasti tidak terlalu sulit untuk menggunakan mongoDB.

B. KELEBIHAN DAN KEKURANGAN MONGODB

1. Kelebihan MongoDB

- Sangat bagus untuk data list produk, karena MongoDB bisa melakukan searching lebih cepat ketimbang MySql atau RDBMS lainnya
- Kita tidak perlu membuat struktur tabel, kita hanya perlu melakukan proses insert maka MongoDB akan otomatis membuatkan struktur tabelnya
- Proses CRUD (Create, Update, Delete) terasa sangat ringan
- Banyak website besar yang sudah menggunakan MongoDB

2. Kekurangan MongoDB

- MongoDB harus diinstall di sebuah server, dan ketika kita menggunakan PHP anda juga harus merestart server anda driver MongoDB anda dapat digunakan oleh PHP
- Belum support di hosting, namun bisa diakali dengan menggunakan MongoHQ (untuk free terdapat limit sampai 16MB)

Kapan Menggunakan MongoDB ?

MongoDB merupakan basis data yang tidak relasional, hal ini membuat MongoDB sangat cepat saat melakukan proses manipulasi data dari pada sistem basis data relasional (RDBMS), selain itu MongoDB berbasis dokumen sehingga tidak memiliki struktur yang teratur seperti table. MongoDB sangat cocok untuk membangun sebuah sistem seperti blog, forum, portal berita, social networking, dan website yang tidak membutuhkan proses transaksi seperti sistem Bank

Kapan Tidak Menggunakan MongoDB ?

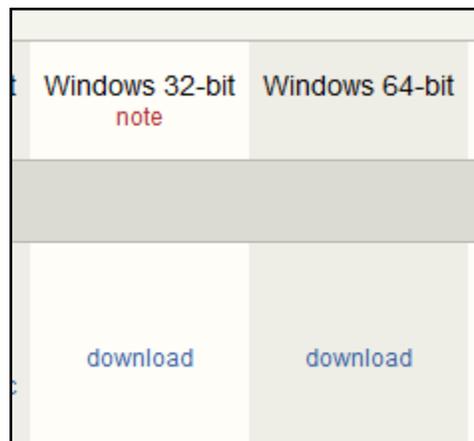
Hampir semua sistem basis data NoSQL tidak mendukung proses transaksi. Padahal hal ini sangat penting jika kita membangun sistem yang selalu memerlukan proses transaksi seperti sistem Bank misalnya. Oleh Karena itu MongoDB sangat tidak cocok jika digunakan untuk membangun sistem berbasis transaksi, karena jika sistem basis data NoSQL (seperti MongoDB) digunakan dalam kasus ini, maka bisa dipastikan akan banyak terjadi kegagalan dalam sistem

C. INSTALASI MONGODB DI OS WINDOWS DAN LINUX

Install MongoDB di Windows

Untuk menggunakan mongoDB di Windows anda harus melakukan hal-hal seperti berikut ini :

1. Download File Master MongoDB For Windows di website resminya yakni di <http://www.mongodb.org> misalkan yang saya download adalah **mongodb-win32-i386-1.8.2.zip**



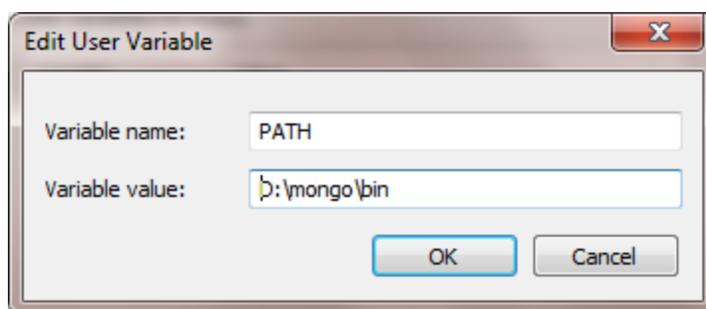
- Copy file tersebut ke dalam Drive Utama anda misalkan C: atau D: kemudian ekstrak file zipnya
- Setelah diekstrak maka akan muncul sebuah folder baru dengan nama mongoddb-win32-i386-1.8.2
- Kemudian setting PATH nya agar mudah dipanggil saat membuka command prompt, Berikut ini cara melakukan setting PATH.

Klik Kanan My Computer → Properties

Pilih Advance system settings

Pada Tab Advance Klik Environment Variables

Klik Tombol new yang paling atas kemudian isikan parameter berikut ini



Nah sekarang kita sudah berhasil melakukan setting path untuk mongoddb, Isikan **Variable Name** = PATH dan **Variable Value** = Lokasi anda meletakkan file binary mongoddb

- Sebelum kita menjalankan mongoddb, kita harus membuat folder untuk menyimpan file database yang akan kita buat, misalkan saya membuat folder dengan nama dbdata di driver D: sehingga lokasinya terlihat seperti ini D:\dbdata
- Buka command prompt dan ketikkan **mongod --dbpath=D:\dbdata** kemudian Enter

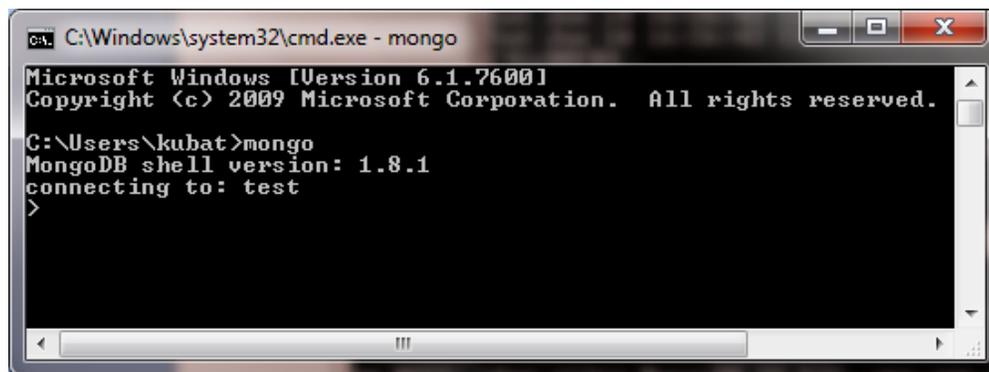
```

C:\Users\kubat>mongod --dbpath=D:\data\db
Sat Jun 18 16:26:43 [initandlisten] MongoDB starting : pid=3200 p
h=D:\data\db 32-bit
** NOTE: when using MongoDB 32 bit, you are limited to about 2 gi
**      see http://blog.mongodb.org/post/137788967/32-bit-limita
**      with --dur, the limit is lower
Sat Jun 18 16:26:43 [initandlisten] db version v1.8.1, pdfile ver
Sat Jun 18 16:26:43 [initandlisten] git version: a429cd4f535b2499
f41c00f04
Sat Jun 18 16:26:43 [initandlisten] build sys info: windows <5, 1
vice Pack 3' > BOOST_LIB_VERSION=1_35
Sat Jun 18 16:26:43 [initandlisten] waiting for connections on po
Sat Jun 18 16:26:43 [websvr] web admin interface listening on por

```

(nah pada proses ini kita sudah menjalankan mongod servernya) lokasi dbpath diatas harus anda sesuaikan dengan letak dimana anda ingin menyimpan database anda, kebetulan tadi saya membuat folder dbdata di Drive D untuk menyimpan databasenya.

7. Biarkan jendela CMD sebelumnya tetap berjalan Kemudian untuk menjalankan mongod client nya buka sebuah CMD baru dan ketikkan **mongo**. Sekarang kita sudah masuk ke dalam mongo terminal dan anda sudah bisa memulai untuk membuat database dengan mongod



```
C:\Windows\system32\cmd.exe - mongo
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\kubat>mongo
MongoDB shell version: 1.8.1
connecting to: test
>
```

Install MongoDB di Linux

Untuk menggunakan mongoDB di Linux anda harus melakukan hal-hal seperti berikut ini :

1. Download File Master MongoDB For Linux di website resminya yakni di <http://www.mongodb.org> misalkan yang saya download adalah **mongodb-linux-i686-1.8.2.tgz**
2. Copy file tersebut ke dalam Komputer anda misalkan di folder /home/kubat kemudian ekstrak file tgz nya



3. Setelah diekstrak maka akan muncul sebuah folder baru dengan nama mongodb-linux-i686-1.8.2, kemudian rename folder tersebut agar gak terlalu panjang misalnya menjadi **mongodb**

4. Sebelum kita menjalankan mongod, kita harus membuat folder untuk menyimpan file database yang akan kita buat, misalkan saya membuat folder dengan nama dbdata di home sehingga lokasinya terlihat seperti ini /home/kubat/dbdata
5. Sekarang Buka Terminal (Console) kemudian masuk pada folder tempat kita meletakkan master mongod tadi dengan cara ketikkan cd /home/kubat/mongod/bin
6. Untuk menjalankan mongod ketikkan `./mongod --dbpath=/home/kubat/dbdata` kemudian Enter (nah pada proses ini kita sudah menjalankan mongod servernya) lokasi dbpath diatas harus anda sesuaikan dengan letak dimana anda ingin menyimpan database anda, kebetulan tadi saya membuat folder dbdata di /home/kubat/dbdata

```
File Edit View Search Terminal Help
kubat@kubat-laptop:~$ cd mongod/bin/
kubat@kubat-laptop:~/mongod/bin$ ./mongod --dbpath=/home/kubat/datadb
Sat Jun 18 18:07:23 [initandlisten] MongoDB starting : pid=1848 port=270
17 dbpath=/home/kubat/datadb 32-bit
The libunwind project is built on
** NOTE: when using MongoDB 32 bit, you are limited to about 2 gigabytes
of data
** see http://blog.mongodb.org/post/137788967/32-bit-limitations
** with --dur, the limit is lower
Sat Jun 18 18:07:23 [initandlisten] db version v1.8.1, pdfile version 4.
5
Sat Jun 18 18:07:23 [initandlisten] git version: a429cd4f535b2499cc4130b
06ff7c26f41c00f04
Sat Jun 18 18:07:23 [initandlisten] build sys info: Linux bs-linux32.10g
en.cc 2.6.21.7-2.fc8xen #1 SMP Fri Feb 15 12:39:36 EST 2008 i686 BOOST_L
IB_VERSION=1_37
```

7. Kemudian untuk menjalankan mongod client nya ketikkan `./mongo` lalu ENTER. Sekarang kita sudah masuk ke dalam mongo terminal dan anda sudah bisa memulai untuk membuat database dengan mongod

```
File Edit View Search Terminal Help
kubat@kubat-laptop:~$ cd mongod/bin
kubat@kubat-laptop:~/mongod/bin$ ./mongo
MongoDB shell version: 1.8.1
connecting to: test
>
Sat Jun 18 18:07:23 [initandlisten] db version v1.8.1, pdfile version
Sat Jun 18 18:07:23 [initandlisten] git version: a429cd4f535b2499cc41
06ff7c26f41c00f04
Sat Jun 18 18:07:23 [initandlisten] build sys info: Linux bs-linux32.10g
```

D. QUERY PERINTAH DASAR MONGODB

Dalam melakukan query pada mongoDB tidak menggunakan perintah SQL seperti pada RDBMS. MongoDB memiliki style tersendiri untuk melakukan penyimpanan data, update data , delete dan pembuatan database. Mongoddb menyimpan data dalam bentuk **collection** berupa struktur tipe data seperti array. Collection dalam MongoDB menggunakan struktur data J-SON. Berikut ini adalah perintah dasar yang biasa digunakan dalam mongoDB :

Perintah yang sering digunakan dalam mongoDB

Perintah	Keterangan
Use <nama_db>	Untuk CREATE DATABASE baru atau memilih database yang sudah ada untuk digunakan
Db.mycoll.save(object)	Untuk INSERT data baru kedalam collection, jika sebelumnya namacollection belum ada maka akan dibuatkan secara otomatis
Db.mycoll.update(kondisi)	Untuk UPDATE data yang ada dalam collection
Db.mycoll.remove(kondisi)	Untuk DELETE data yang ada dalam collection
Db.mycoll.find(kondisi)	Untuk SELECT data yang ada dalam collection, perintah ini juga digunakan untuk melakukan pencarian data dalam collection

Berikut ini akan kita berikan contoh penerapan beberapa query dasar pada MongoDB yakni **Save(Input Data)** , **Update(Edit)**, **Remove(Hapus)** dan **find(Tampilkan atau Cari)**

I. Proses Input Data

Proses input data menggunakan mongoDB tidak terikat dengan yang namanya kolom, field maupun tabel. Karena mongoDB tidak mengenal adanya tabel, field, maupun kolom. MongoDB mengenal adanya koleksi (collection) dan obyek yang ada dalam koleksi tersebut. Untuk membuat suatu koleksi, tidak diperlukan suatu syntax khusus. Kita tinggal menambahkan saja suatu data ke dalam suatu nama koleksi. Jika nama koleksi yang dituju sudah ada, maka data yang dimasukkan akan secara otomatis dimasukkan ke dalam koleksi tersebut. Namun, jika nama koleksi yang dituju belum ada, maka mongoDB secara otomatis membuatkan koleksi tersebut dengan data yang dimasukkan tadi.

Contoh syntax yang digunakan untuk menginputkan data adalah sebagai berikut :

```
db.mahasiswa.save(  
  {  
    nama : "Miftah Alfian Syah",  
    nrp : "090411100096"  
  }  
);
```

Maksud dari syntax diatas adalah kita menyimpan data "Miftah Alfian Syah" ke dalam sebuah key nama dan data "090411100096" ke dalam sebuah key nrp dalam sebuah koleksi (collection) yang bernama mahasiswa. Ketika kita ingin memasukkan data lagi, kita bisa menyimpan data dengan nama key yang berbeda dari key yang telah diinputkan tadi. Oleh karena itu, proses input data menggunakan mongoDB akan terasa sangat mudah.

II. Proses Update Data

Proses update data pada mongoDB pun juga sangat mudah. Kita tinggal menentukan saja key dengan nama apa yang diubah dan berdasarkan apa. Contoh syntax yang digunakan untuk mengupdate data dalam koleksi (collection) mongoDB adalah sebagai berikut :

```
db.mahasiswa.update(  
  {nrp:"090411100029"},  
  {$set:{nilai:100}},  
  false,true  
);
```

Maksud dari syntax diatas adalah akan diupdate suatu key bernama nilai dengan nilai 100, dimana key bernama nrp mempunyai data "090411100029" yang berada dalam koleksi (collection) mahasiswa.

III. Proses Delete Data

Proses delete data pada mongoDB pun juga sangat mudah. Kita tinggal menentukan data yang dihapus berdasarkan key apa dengan nilai apa. Contoh syntax yang digunakan untuk menghapus data dalam koleksi (collection) mongoDB adalah sebagai berikut :

```
db.mahasiswa.remove(
  {nrp:"090411100126"}
);
```

Maksud dari syntax di atas adalah akan dihapus data dalam koleksi (collection) mahasiswa dimana key yang bernama nrp bernilai "090411100126".

IV. Proses Pencarian dan Menampilkan Data

Proses pencarian data dalam mongoDB pun juga sangat mudah. Kita tinggal mengetikkan nama koleksi beserta nama key yang akan ditampilkan ataupun yang dicari. Berikut ini, terdapat beberapa contoh syntax yang digunakan untuk menampilkan atau mencari data dengan mongoDB

```
db.mahasiswa.find();
```

Syntax diatas digunakan untuk menampilkan seluruh data yang berada dalam koleksi (collection) mahasiswa. Hasil yang ditampilkan dalam console akan seperti berikut :

```
< "_id" : ObjectId<"4dfafc5c9c88f94146b19207">, "nama" : "Miftah Alfian Syah", "
nrp" : "090411100096" >
< "_id" : ObjectId<"4dfafc739c88f94146b19208">, "nama" : "M. Dannish Koosrivanto
", "nrp" : "090411100126" >
< "_id" : ObjectId<"4dfafd859c88f94146b19209">, "nilai" : 99 >
< "_id" : ObjectId<"4dfafdb09c88f94146b1920a">, "nrp" : "090411100029", "nilai"
: 100 >
```

```
db.mahasiswa.find({}, {nama:1, nrp:1});
```

Syntax diatas digunakan untuk menampilkan data-data dalam key nama dan nrp dalam koleksi (collection) mahasiswa. Hasil yang ditampilkan dalam console akan seperti berikut :

```
< "_id" : ObjectId<"4dfafc5c9c88f94146b19207">, "nama" : "Miftah Alfian Syah", "
nrp" : "090411100096" >
< "_id" : ObjectId<"4dfafc739c88f94146b19208">, "nama" : "M. Dannish Koosrivanto
", "nrp" : "090411100126" >
< "_id" : ObjectId<"4dfafd859c88f94146b19209"> >
< "_id" : ObjectId<"4dfafdb09c88f94146b1920a">, "nrp" : "090411100029" >
```

```
db.mahasiswa.count();
```

Syntax diatas digunakan untuk menampilkan jumlah data yang berada dalam koleksi (collection) mahasiswa. Hasil yang ditampilkan dalam console akan seperti berikut :

```
4
> _
```

Perintah Dasar Lengkap MongoDB berdasarkan Kategori

MONGOCLIENT

Perintah	Keterangan
Use <nama_db>	Untuk membuat database baru atau memilih database yang sudah ada untuk digunakan
Show dbs	Menampilkan seluruh nama database yang sudah dibuat di mongodb
Show collections	Menampilkan seluruh nama collection dalam tiap database beserta nama database nya
Show users	Menampilkan namauser pada database
Db.help()	Menampilkan manual untuk manipulasi database
db.namacollection.help()	Menampilkan manual untuk manipulasi collection
exit	Keluar dari Mongo Shell

DATABASE

Perintah	Keterangan
Db.addUser(user,password)	Menambah user baru pada database
Db.auth()	Memberikan hak akses pada user
Db.copyDatabase(FromDB, todb, FromHost)	Menyalin isi database ke database lain
Db.createCollection()	Membuat sebuah collection baru pada database
Db.currentOp()	Menampilkan operasi / proses terkini dari database
Db.dropDatabase()	Menghapus database yang sedang digunakan sekarang
Db.getCollection(cname)	Mengambil objek collection tertentu dari database yang sedang aktif sekarang
Db.getCollectionNames()	Mengambil seluruh nama collection yang terdapat dalam database yang sedang aktif
Db.getName()	Menampilkan nama dari database yang sedang aktif
Db.printCollectionStats()	Menampilkan seluruh Collection beserta atributnya pada database
Db.printReplicationInfo()	Menampilkan info dari replikasi database
Db.printShardingStatus()	Menampilkan status dari fitur sharding pada database

Db.removeUser(username)	Menghapus user tertentu pada database
Db.repairDatabase()	Melakukan repair pada database yang sedang aktif sekarang

COLLECTION

Perintah	Keterangan
Db.mycoll.find().help()	Menampilkan manual (panduan) untuk menggunakan perintah find
Db.mycoll.count()	Menampilkan jumlah data pada collection mycoll
Db.mycoll.dataSize()	Menampilkan ukuran data pada collection mycoll
Db.mycoll.distinct(key)	Menampilkan seluruh data pada collection mycoll dan jika ada isi data yang sama pada key maka yang ditampilkan hanya 1 saja
Db.mycoll.drop()	Menghapus collection mycoll pada database
Db.mycoll.dropIndex(name)	Menghapus index pada collection mycoll
Db.mycoll.find([query],[field])	Menampilkan seluruh data pada collection
Db.mycoll.find(.....).count()	Menampilkan jumlah data pada collection
Db.mycoll.find(.....).limit(n)	Menampilkan data pada collection berdasarkan limit tertentu
Db.mycoll.find(.....).skip(n)	Menampilkan data pada collection dengan pengecualian beberapa data
Db.mycoll.find(.....).sort(.....)	Menampilkan data pada collection dan disorting dengan ASCENDING maupun DESCENDING
Db.mycoll.getDB()	Mengambil object data dari sebuah collection
Db.mycoll.remove(query)	Menghapus data yang ada pada collection dengan query tertentu
Db.mycoll.renameCollection(newName, <dropTarget>)	Mengubah nama suatu collection
Db.mycoll.save(obj)	Tambah data baru pada collection
Db.mycoll.stats()	Menampilkan statistik dari collection mycoll
Db.mycoll.totalIndexSize()	Ukuran (Bytes) dari semua index
Db.mycoll.totalSize()	Total ukuran seluruh data
Db.mycoll.update(query, object[, upsert_bool, multi_bool])	Melakukan update data pada suatu collection

Db.mycoll.getShardVersion()	Menampilkan versi dari fitur sharding yang digunakan pada collection
-----------------------------	--

Pada tabel dibawah ini adalah contoh **perbandingan syntax sql dan syntax yang digunakan pada mongoDB** untuk menampilkan data maupun untuk pencarian data dalam suatu database.

SQL	MongoDB
CREATE DATABASE akademik	use rumahsakit;
INSERT INTO mahasiswa (nrp, nama, jurusan) VALUES ('09126', 'DANNIS', 'Informatika')	db.mahasiswa.save({nrp: "09126", nama: "DANNIS", jurusan: "Informatika"});
SELECT * FROM mahasiswa	db.mahasiswa.find();
UPDATE mahasiswa SET nama = "Budi" WHERE nrp = "090411100029"	db.mahasiswa.update({nrp: "090411100029"}, {\$set: {nilai: "Budi"}}, false, true);
DELETE mahasiswa WHERE Nrp = "090411100029"	db.mahasiswa.remove({nrp: "090411100029"});
SELECT * FROM users WHERE age=33	db.users.find({age:33});
SELECT * FROM users WHERE age>33	db.users.find({age:{\$gt:33}});
SELECT * FROM users WHERE age<33	db.users.find({age:{\$lt:33}});
SELECT * FROM users WHERE age<33 AND age>=17	db.users.find({age:{\$lt:33,\$gte:17}});
SELECT * FROM users WHERE a=1 OR b=2	db.users.find({\$or:[{a:1},{b:2}]});
SELECT name, age FROM users WHERE age=33	db.users.find({age:33}, {name:1, age:1});
SELECT * FROM users WHERE age=33 ORDER BY name	db.users.find({age:33}).sort({name:1});
SELECT * FROM users WHERE age=33 ORDER BY name DESC	db.users.find({age:33}).sort({name:-1});
SELECT * FROM users WHERE name LIKE "%Basith%"	db.users.find({name:/Basith/});

SELECT * FROM users WHERE name LIKE "Kubat%"	db.users.find({name:/^Kubat});
SELECT DISTINCT last_name FROM users	db.users.distinct('last_name');
SELECT COUNT(*) FROM users	db.users.count();

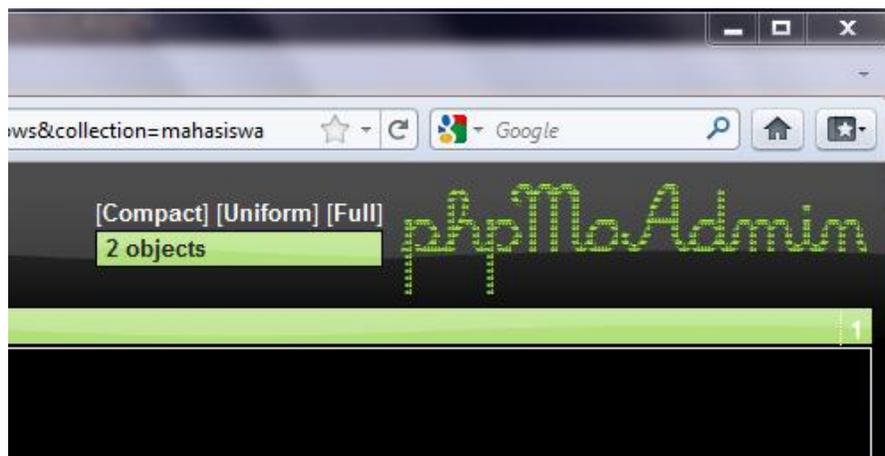
E. MANAJEMEN MONGODB DENGAN PHPMOADMIN

Untuk memudahkan para pengguna dalam merancang database serta collection di dalamnya, terdapat sebuah software manajemen database berbasis GUI untuk MongoDB yakni **PHPMoAdmin**, aplikasi ini dibuat dengan PHP dan fungsinya mirip dengan PHPMyAdmin yakni sama-sama digunakan untuk memudahkan dalam manajemen database.

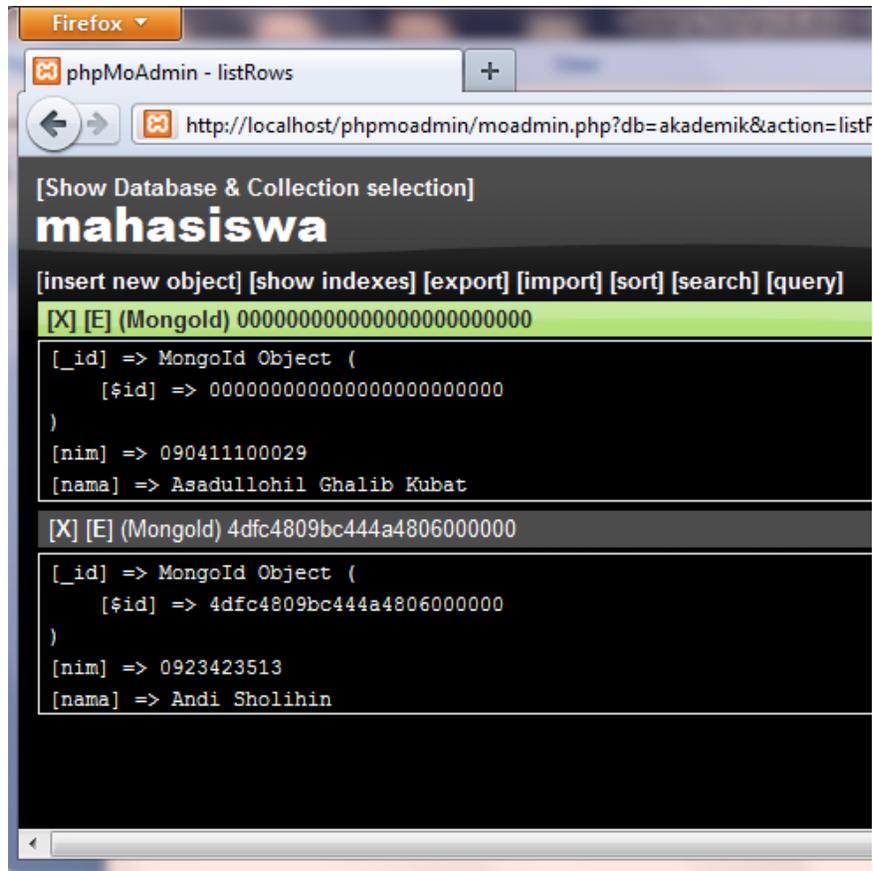
Untuk dapat menggunakannya silahkan unduh filenya di situs resminya di <http://www.phpmoadmin.com/>

Setelah itu berikut langkah memasangnya :

1. Pastikan di computer anda sudah terinstall Web Server Apache (misalnya XAMPP atau LAMPP), jika sudah terinstall langsung aktifkan web server anda
2. Setelah itu copykan phpmoadmin.zip yang telah anda download ke dalam folder htdocs dalam XAMPP atau LAMPP anda kemudian ekstrak.
3. Buka Web Browser misalnya Firefox dan ketikkan pada address bar <http://localhost/phpmoadmin/moadmin.php>.



4. Sekarang anda sudah dapat membuat database serta collection dengan mudah bersama PHPMoAdmin.



F. DUKUNGAN DAN INTEGRASI DALAM BAHASA PEMROGRAMAN

Hampir semua bahasa pemrograman modern mendukung MongoDB dan untuk mengaplikasikan MongoDB dalam bahasa pemrograman yang di gunakan, masing – masing memerlukan driver MongoDB yang berbeda-beda bahasa pemrograman yang mendukung MongoDB ialah JAVA, C/C++, Python, Ruby, PHP, C# dan lain – lain. Driver untuk setiap bahasa pemrograman bisa didownload di situs resmi mongoDB (<http://mongodb.org/>).

Pada bagian ini akan kami jelaskan langkah untuk melakukan integrasi mongoDB dengan beberapa bahasa pemrograman, pada makalah ini akan kami beri contoh 2 saja yakni menggunakan PHP dan JAVA.

Integrasi MongoDB dengan PHP

Berikut ini merupakan langkah langkah dalam integrasi mongoDB dengan PHP

1. Sebelum anda mulai membuat aplikasi dengan PHP dan MongoDB pastikan bahwa di komputer anda sudah terinstall Apache Server dan PHP (Contoh XAMPP atau LAMPP) agar kode PHP bisa dijalankan
2. Download Driver koneksi MongoDB PHP di <http://www.mongodb.org>, misalkan yang kita download adalah ***mongo-latest-php5.3vc6ts.zip***
3. Ekstrak file zip tersebut dan akan muncul sebuah file yakni ***php_mongo.dll*** (ini merupakan file driver koneksi mongodb dengan php)
4. Copykan ***php_mongo.dll*** ke dalam folder ekstension php, jika anda menggunakan XAMPP maka copykan ke dalam C:\xampp\php\ext
5. Edit File php.ini pada server PHP anda (jika anda menggunakan XAMPP maka php.ini berada di C:\xampp\php\php.ini), tambahkan perintah ***extension=php_mongo.dll*** pada php.ini kemudian simpan
6. Jalankan Server Apache anda (XAMPP), jika tidak ada pesan error maka pemasangan driver php_mongo tadi berhasil dan kita sudah bisa membuat aplikasi dengan PHP MongoDB.

Sekarang kita akan mencoba membuat aplikasi untuk simpan, update, hapus dan menampilkan data dengan PHP dan MongoDB.

TAMBAH DATA

Buka Editor kesayangan anda dan ketikkan kode berikut ini kemudian simpan di C:\xampp\htdocs\cobamongo dengan nama *insert.php*

```
<?php
$m = new Mongo();
$db = $m->rumahsakit;
$koleksi = $db->pasien;

$objek = array(
    "idpasien" => "082134125",
    "nama" => "Abdul Basith",
    "alamat" => "Jakarta"
);

$koleksi->insert($objek);
?>
```

insert.php

EDIT DATA

Buka Editor kesayangan anda dan ketikkan kode berikut ini kemudian simpan di C:\xampp\htdocs\cobamongo dengan nama *update.php*

```
<?php
$m = new Mongo();
$db = $m->rumahsakit;
$koleksi = $db->pasien;

$kriteria = array("idpasien" => "082134125");

$objek = array(
    "idpasien" => "082134125",
    "nama" => "Dannish",
    "alamat" => "Surabaya"
);

$koleksi->update($kriteria,$objek);
?>
```

update.php

HAPUS DATA

Buka Editor kesayangan anda dan ketikkan kode berikut ini kemudian simpan di C:\xampp\htdocs\cobamongo dengan nama *remove.php*

```
<?php
$m = new Mongo();
$db = $m->rumahsakit;
$koleksi = $db->pasien;

$kriteria = array("idpasien" => "082134125");
$koleksi->remove($kriteria);
?>
```

remove.php

TAMPILKAN DATA

Buka Editor kesayangan anda dan ketikkan kode berikut ini kemudian simpan di C:\xampp\htdocs\cobamongo dengan nama *find.php*

```
<?php
$m = new Mongo();
$db = $m->rumahsakit;
$koleksi = $db->pasien;

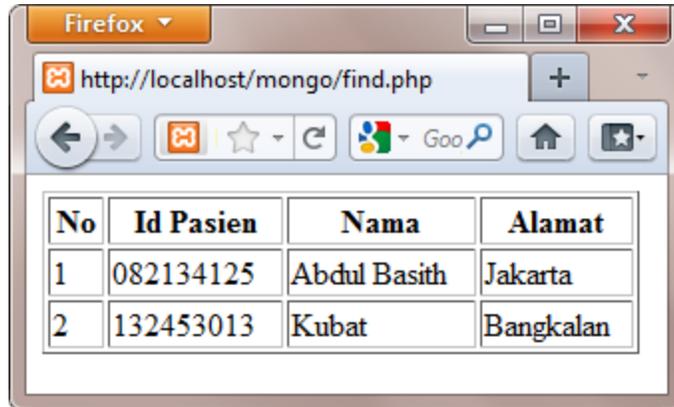
$cursor = $koleksi->find();
$no = 0;
echo "
    <table border=1 width=300>
        <th>No</th><th>Id Pasien</th><th>Nama</th><th>Alamat</th>";

foreach($cursor as $obj){
    $no++;
    echo "<tr>
        <td>$no</td>
        <td>$obj[idpasien]</td>
        <td>$obj[nama]</td>
        <td>$obj[alamat]</td>
    </tr>
    ";
}
?>
```

find.php

Untuk menjalankan 4 kode PHP diatas, buka Browser kesayangan anda kemudian ketikkan pada address bar <http://localhost/cobamongo/<namafile>.php>

Jika file find.php dijalankan maka akan menampilkan seluruh data yang ada pada koleksi pasien pada database rumahsakit seperti berikut



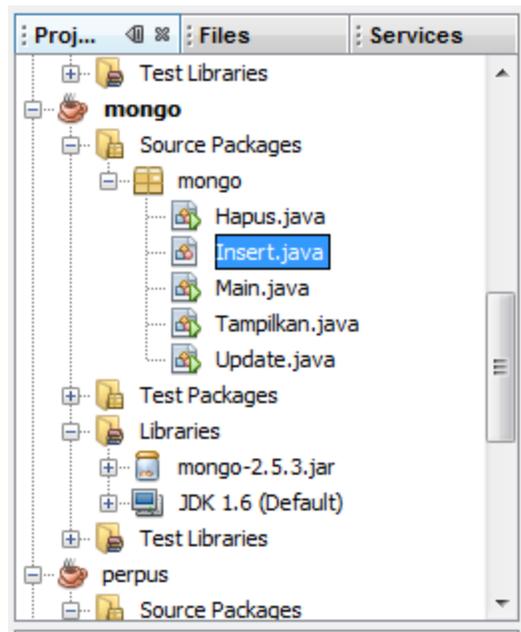
Penjelasan dari kode diatas

Sintaks	Keterangan
<code>\$m = new Mongo()</code>	: Membuat sebuah variabel objek baru dengan nama \$m dari class Mongo. Nantinya file objek \$m inilah yang akan mengkoneksikan PHP dengan MongoDB
<code>\$db = \$m->rumahsakit</code>	: Membuat variabel objek \$db yang akan digunakan sebagai objek dari database rumahsakit yang dipilih
<code>\$koleksi = \$db->pasien</code>	: Membuat variable objek koleksi yang akan digunakan sebagai variable dari collection yang dipilih yakni pasien
<code>\$koleksi->insert(\$object)</code> <code>\$koleksi->update(\$object)</code> <code>\$koleksi->remove(\$object)</code> <code>\$koleksi->find(\$object)</code>	: Perintah untuk melakukan Query Insert, update, remove dan find pada koleksi, parameter \$object di isi sesuai dengan perintahnya

Integrasi MongoDB dengan JAVA

Berikut ini merupakan langkah langkah dalam integrasi MongoDB dengan JAVA

1. Sebelum kita mulai membuat aplikasi dengan JAVA pastikan Java Development Kit (JDK) sudah terinstal di computer anda (Jika anda familier dengan membuat aplikasi JAVA dengan IDE seperti Netbeans atau Eclipse anda juga bisa memakainya).
2. Download Driver koneksi MongoDB JAVA di situs resmi nya <http://www.mongodb.org> , misalkan yang saya download adalah **mongo-2.5.3.jar** (file ini adalah class library yang akan dapat anda gunakan untuk mengkoneksikan MongoDB dengan JAVA)
3. Buka Editor Kesayangan anda, disini kita menggunakan Netbeans.
4. Buatlah sebuah Project Baru, disini projectnya kita beri nama mongo
5. Import Library **mongo-2.5.3.jar** pada project anda dengan cara klik kanan pada Libraries dan Pilih Add JAR/Folder kemudian pilih file jarnya (Setelah library di import sekarang anda sudah bisa menggunakan beberapa class yang anda untuk melakukan koneksi dengan MongoDB)



6. Buatlah 4 sebuah Class baru bernama **Insert.java** , **Tampilkan.java** , **Update.java** , **Hapus.java** dan ketikkan kode berikut

```

import com.mongodb.BasicDBObject;
import com.mongodb.Mongo;
import com.mongodb.DB;
import com.mongodb.DBCollection;

public class Insert {

    public static void main(String[] args) {
        try{
            Mongo mongo = new Mongo();
            DB db = mongo.getDB("rumahsakit");
            DBCollection koleksi = db.getCollection("pasien");

            BasicDBObject object = new BasicDBObject();
            object.put("idpasien", "090411100029");
            object.put("nama", "Asadullohil Ghalib Kubat");
            object.put("alamat", "Bangkalan");

            koleksi.insert(object);
        }
        catch(Exception e){
            System.out.println(e.toString());
        }
    }
}

```

Insert.java

```

import com.mongodb.Mongo;
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;

public class Tampilkan {

    public static void main(String[] args){
        try{
            Mongo mongo = new Mongo();
            DB db = mongo.getDB("rumahsakit");
            DBCollection koleksi = db.getCollection("pasien");

            DBCursor cursor = koleksi.find();

            while(cursor.hasNext()){
                System.out.println(cursor.next());
            }
        }
        catch(Exception e){
            System.out.println(e.toString());
        }
    }
}

```

Tampilkan.java

```
import com.mongodb.BasicDBObject;
import com.mongodb.Mongo;
import com.mongodb.DB;
import com.mongodb.DBCollection;
```

Update.java

```
public class Update {

    public static void main(String[] args){
        try{
            Mongo mongo = new Mongo();
            DB db = mongo.getDB("rumahsakit");
            DBCollection koleksi = db.getCollection("pasien");

            BasicDBObject kriteria = new BasicDBObject();
            kriteria.put("idpasien", "090411100029");

            BasicDBObject object = new BasicDBObject();
            object.put("idpasien", "090411100029");
            object.put("nama", "Miftah Alfiansyah");
            object.put("alamat", "Bandung");

            koleksi.update(kriteria, object);
        }
        catch(Exception e){
            System.out.println(e.toString());
        }
    }
}
```

```
import com.mongodb.Mongo;
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.BasicDBObject;
```

Hapus.java

```
public class Hapus {

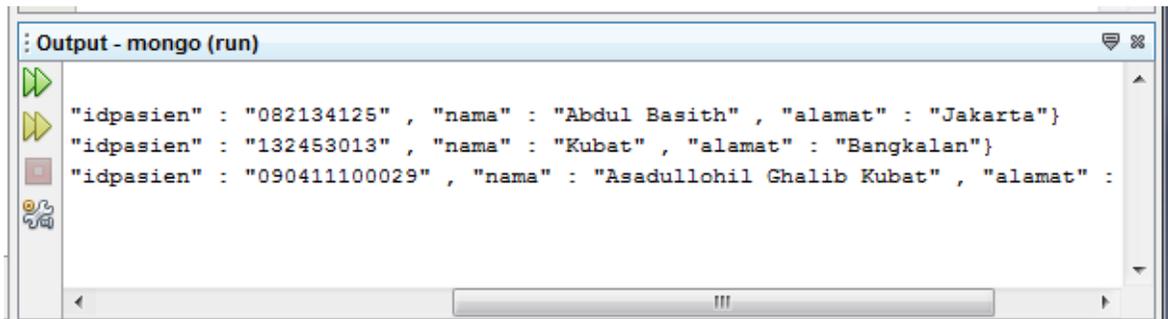
    public static void main(String[] args){
        try{
            Mongo mongo = new Mongo();
            DB db = mongo.getDB("rumahsakit");
            DBCollection koleksi = db.getCollection("pasien");

            BasicDBObject kriteria = new BasicDBObject();
            kriteria.put("idpasien", "090411100029");

            koleksi.remove(kriteria);
            mongo.close();
        }
        catch(Exception e){

        }
    }
}
```

Jika **Tampil.java** dijalankan maka akan muncul tampilan seperti berikut ini dengan



```
Output - mongo (run)
"\"idpasien\" : \"082134125\" , \"nama\" : \"Abdul Basith\" , \"alamat\" : \"Jakarta\"}
\"idpasien\" : \"132453013\" , \"nama\" : \"Kubat\" , \"alamat\" : \"Bangkalan\"}
\"idpasien\" : \"090411100029\" , \"nama\" : \"Asadullohil Ghalib Kubat\" , \"alamat\" :
```

Penjelasan dari kode diatas

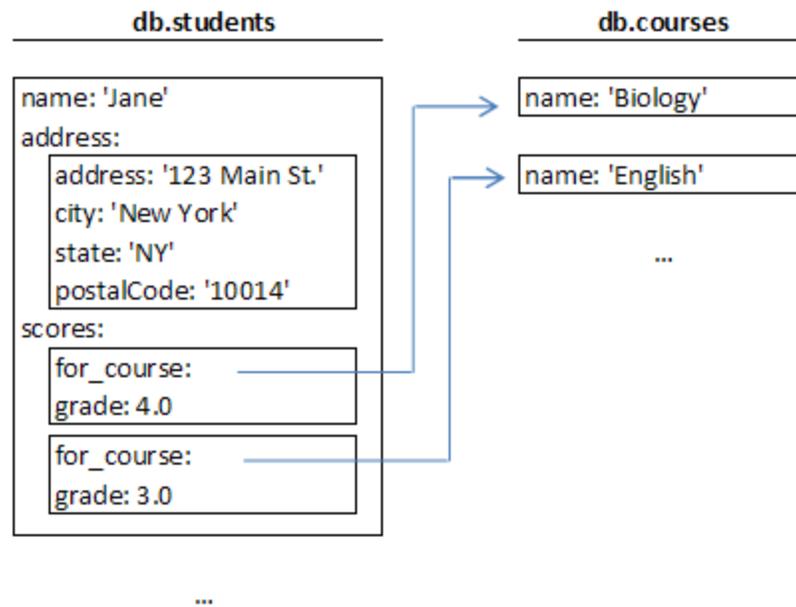
Sintaks	Keterangan
<pre>import com.mongodb.Mongo () import com.mongodb.DB ()</pre>	: Melakukan import beberapa class yang dibutuhkan untuk koneksi JAVA dan MongoDB
<pre>Mongo mongo = new Mongo ()</pre>	: Membuat sebuah objek dengan nama mongo, syntak ini digunakan untuk melakukan koneksi ke MongoDB
<pre>DB db = mongo.getDB ("rumahsakit");</pre>	: Memilih database rumahsakit
<pre>DBCollection koleksi = db.getCollection ("pasien");</pre>	: Memilih Koleksi Pasien
<pre>BasicDBObject object = new BasicDBObject (); object.put ("idpasien", "090411100029");</pre>	Membuat object baru dari class DBObject serta menambahkan data baru pada object
<pre>koleksi.insert () koleksi.update () koleksi.find ()</pre>	Untuk melakukan query INSERT, UPDATE , FIND pada koleksi yang dipilih

G. FITUR KHUSUS MONGODB

MongoDB memiliki beberapa fitur khusus yang tidak dimiliki oleh Database yang lain, hal inilah yang membuat MongoDB dari beberapa sisi lebih diunggulkan dari RDBMS seperti MySQL ataupun ORACLE. Berikut beberapa fitur khusus yang dimiliki oleh MongoDB

- **Document-Oriented Storage**

Dengan MongoDB anda tidak perlu melakukan normalisasi desain database seperti pada saat anda melakukan desain di RDBMS (Manajemen Database Relasional), jika anda ingin melakukan embedded (referensi) suatu collection dengan collection yang lain anda dapat melakukannya dengan cara berikut ini. Sebagai contoh, pada diagram di bawah ini



terdapat 2 buah collection, Students dan courses. Di dalam dokumen students terdapat dokumen address dan score, dan dokumen score memiliki referensi ke dokumen courses

Embed Vs Reference

Ini adalah 2 buah metode yang berbeda yang digunakan dalam referensi nilai dari 2 buah table (collection), jika pada RDBMS seperti MySQL kita biasa menggunakan Reference maka dalam Database berbasis dokumen seperti MongoDB kita dapat menggunakan cara Embed, Cara Embed lebih cepat dalam proses pencarian dari pada cara Reference, sebagai

contoh jika saya ingin mengakses nama course dari sebuah grade score pertama maka saya cukup mengetikkan perintah berikut ini

```
print(student.scores[0].for_course.name)
```

Perintah diatas dirasa lebih cepat dalam proses penggunaan memory dalam komputer

- **Full Index Support**

Index adalah sebuah struktur data yang mengumpulkan informasi tentang nilai yang spesifik (unik) di dalam dokumen. Index digunakan untuk mempercepat pencarian dan pengurutan data. Index yang ada di MongoDB hampir sama dengan Index yang kita kenal pada RDBMS seperti MySQL.

Untuk membuat Index pada MongoDB kita dapat menggunakan perintah `ensureIndex()`. Misalkan kita ingin menandai bahwa field `j` pada collection `things` menjadi sebuah index maka perintahnya adalah

```
db.things.ensureindex({j:1});
```

Jika sebuah collection sudah memiliki Index maka kita dapat melakukan pencarian cepat dengan memanfaatkan index tersebut sebagai contoh :

```
db.things.find({j:2});           → lebih cepat karena j merupakan index  
db.things.find({x:3});           → lebih lambat karena x bukan index
```

Dalam MongoDB terdapat sebuah field yang digunakan sebagai index default pada setiap collection yakni field “**_id**” nilai dari field `_id` ini bermacam macam dan random

- **Auto-Sharding**

Jika dalam RDBMS seperti MySQL kita sangat kesulitan bahkan sangat sulit untuk melakukan Sharding, maka dalam database NoSQL seperti MongoDB sharding dapat kita lakukan dengan mudah. Nah sebelumnya apa sih Sharding itu ? Sharding adalah metode pencarian data pada database, pencarian ini bukan pencarian biasa dengan sharding hasil pencarian dapat lebih akurat karena hasil pencarian dikelompokkan berdasarkan country (Negara) atau daerah tempat si pencari sehingga hasilnya bisa lebih optimal. Metode sharding ini dipake pada pencarian inbox dan friend pada kotak pencarian Facebook.

Jika kita menggunakan Database MongoDB, kita akan dimudahkan dalam melakukan proses sharding.

Fitur Lainnya

Sebenarnya masih banyak fitur-fitur menarik yang disediakan khusus oleh MongoDB seperti **Replication & High Availability, Fast In-Place Updates, Map/Reduce, GridFS** dan lain-lain, inilah yang menjadi alasan kenapa MongoDB cukup menarik untuk kita coba dan gunakan.

KESIMPULAN

MongoDB ini merupakan teknologi baru dalam penyimpanan data (database) yang sangat baik dan cocok digunakan jika aplikasi yang kita buat tidak ada proses transaksi, karena proses pencarian akan terasa lebih cepat dan akurat dengan MongoDB. Akan tetapi jika pada aplikasi yang kita buat terdapat proses transaksi maka MongoDB sangat tidak cocok digunakan karena akan membuat kesalahan yang fatal.

DAFTAR PUSTAKA

BSON, <http://bsonspec.org/> , 18 Juni 2011

Manual Help Mongo Shell , 18 Juni 2011

MongoDB, <http://www.mongodb.org/display/DOCS/Home> , 18 Juni 2011

Java Tutorial, <http://www.mongodb.org/display/DOCS/Java+Tutorial#JavaTutorial-GettingACollection>, 18 Juni 2011

Tutorial, <http://www.php.net/manual/en/mongo.tutorial.php>, 18 Juni 2011

Eko Kurniawan, 2010, Pengenalan MongoDB untuk Penggunaan RDBMS, <http://eecchoo.wordpress.com/2010/10/22/pengenalan-mongodb-untuk-pengguna-relational-dbms/>, 10 Juni 2011

Indexes, <http://www.mongodb.org/display/DOCS/Indexes> , 19 Juni 2011

Schema Design, <http://www.mongodb.org/display/DOCS/Schema+Design> , 19 Juni 2011

Sharding, <http://www.mongodb.org/display/DOCS/Sharding> , 19 Juni 2011