

DESAIN BASIS DATA NON RELASIONAL NOSQL MONGODB PADA WEBSITE SISTEM INFORMASI AKADEMIK

Mohamad Farozi, S.Kom., M.Kom
Dosen STMIK Lembah Dempo
Jalan Sidik Adim No. 98 Jembatan Beringin Pagar Alam
Pos-el : farozi@research@gmail.com

ABSTRACT

Non relational data bases differ from the relational database model that has been used so far. The NoSQL database is very useful in continuously developing data, where the data is so complex that a relational database can no longer accommodate.

The Academic System aims to support the implementation of education, so that universities can provide better and more effective information services to their environment, both inside and outside the university through the internet. Various needs in the field of education as well as regulations that surround it are so high that the management of academic data is a work that is very time-consuming, energy and mind.

MongoDB does not have tables, columns, and rows. In MongoDB there are only collections and documents. Documents contained in mongoDB can have different attributes than other documents even though they are in one collection.

Keywords : *System, NoSQL, MongoDB*

ABSTRAK

Basis data non relasional berbeda dengan model basis data relasional yang selama ini digunakan, Basis data NoSQL sangat berguna pada data - data yang terus - menerus berkembang, dimana data tersebut sangat kompleks sehingga sebuah basis data relasional tidak lagi bisa mengakomodir.

Sistem Akademik bertujuan untuk mendukung penyelenggaraan pendidikan, sehingga perguruan tinggi dapat menyediakan layanan informasi yang lebih baik dan efektif terhadap lingkungannya, baik didalam maupun diluar perguruan tinggi tersebut melalui internet. Berbagai kebutuhan dalam bidang pendidikan maupun peraturan yang melingkupinya yang demikian tinggi sehingga pengelolaan data akademik menjadi pekerjaan yang sangat menguras waktu, tenaga dan pikiran.

MongoDB tidak memiliki tabel, kolom, dan baris. Pada MongoDB yang ada hanyalah koleksi dan dokumen. Dokumen yang terdapatdalam mongoDB dapat memiliki atribut yang berbeda dengan dokumen lain walaupun berada dalam satu koleksi.

Kata Kunci : *Sistem, NoSQL, MongoDB*

I. PENDAHULUAN

Generasi terbaru dari perangkat lunak internet sangat tergantung pada sistem basis data yang digunakan. Kenyataannya untuk meningkatkan kemampuan dan kecepatan dalam operasi, pengembang perangkat lunak dapat melonggarkan beberapa aturan ketat yang terdapat pada basis data seperti konsistensi dan otomatisasi.

Basis data *non relasional* berbeda dengan model basis data relasional yang selama ini digunakan, basis data *non relasional* menggunakan beberapa metode yang berbeda-beda. Metode ini bergantung dari jenis basis data yang digunakan seperti Basis data NoSQL.

Basis data NoSQL sangat berguna pada data - data yang terus - menerus berkembang, dimana data tersebut sangat kompleks sehingga sebuah basis data relasional tidak lagi bisa mengakomodir. Salah satu bentuknya adalah ketika suatu data saling berhubungan satu sama lain maka akan muncul proses duplikasi data. Data tersebut saling memanggil ke beberapa permintaan, tambahan data baru, perubahan data dan lain-lain dengan *key* yang sama karena

faktor hubungan antar data yang sama terjadi terus - menerus, mendorong faktor redundansi data, data menjadi berlipat-lipat dan pada akhirnya akan menyebabkan *crash* pada basis data berkonsep RDBMS.

Dalam basis data NoSQL meliputi hirarki, graf dan basis data berorientasi objek. Penggunaan basis data *non relasional* kembali marak seiring bertambahnya aplikasi berbasis web yang menuntut skalabilitas tinggi meskipun memiliki kelemahan pada redundansi dan konsistensi data, namun basis data *non relasional* dapat menyelesaikan permasalahan yang berkaitan dengan *availability* dan *partition tolerance*.

Sistem Informasi Akademik dirancang untuk keperluan pengelolaan data - data akademik dengan penerapan teknologi komputer baik *hardware* maupun *software* sehingga seluruh proses kegiatan akademik dapat dikelola menjadi informasi yang bermanfaat dalam pengelolaan manajemen perguruan tinggi dan pengambilan keputusan - keputusan bagi pengambil keputusan di lingkungan perguruan tinggi.

Sistem akademik bertujuan untuk mendukung penyelenggaraan pendidikan, sehingga perguruan tinggi dapat menyediakan layanan informasi yang lebih baik dan efektif terhadap lingkungannya, baik didalam maupun diluar perguruan tinggi tersebut melalui internet.

Berbagai kebutuhan dalam bidang pendidikan maupun peraturan yang melingkupinya yang demikian tinggi sehingga pengelolaan data akademik menjadi pekerjaan yang sangat menguras waktu, tenaga dan pikiran.

Dengan dirancangnya sistem akademik, diharapkan pengolahan data antar sistem, seperti pengguna dengan bagian akademik yang menerima *entry* data mahasiswa serta aktivitas kegiatan perkuliahan dan kegiatan administrasi antara mahasiswa dan perguruan tinggi tersebut dapat berlangsung dengan baik.

Permasalahan yang sering terjadi pada perguruan tinggi terhadap kegiatan perkuliahan diantaranya :

1. Banyaknya proses pengolahan data yang harus dilakukan dalam waktu yang singkat. Contohnya : pengolahan data mahasiswa yang

mengambil mata kuliah dan kelas dengan dosen atau tenaga kependidikan.

2. Pada pengolahan data tersebut akan didapatkan pula data - data transaksi berupa, data absensi, data mata kuliah, data transkrip nilai, data jadwal ujian, data biaya pembayaran (status pembayaran mahasiswa), data dosen beserta data penggajian yang merupakan bagian dari sistem keuangan yang diterapkan pada sisi keuangan.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang sudah disebutkan diatas, maka penulis membuat sebuah rumusan masalah penelitian. Rumusan masalah tersebut sebagai berikut :

1. Bagaimana merancang sistem informasi akademik dengan menggunakan basis data non relasional nosql mongodb untuk keperluan pengelolaan data - data akademik agar seluruh proses kegiatan akademik dapat dikelola menjadi informasi yang bermanfaat guna pengambilan keputusan – keputusan di lingkungan perguruan tinggi ?

1.2 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menghasilkan rancangan sistem informasi akademik yang dapat digunakan untuk pengolahan data yang dilakukan dalam waktu yang singkat.
2. Mempermudah dan mempercepat proses pengalokasian kelas, mata kuliah dan dosen pengampu pada periode perkuliahan yang akan berjalan sehingga tidak ada jadwal yang terjadi secara bersamaan.

II. LANDASAN TEORI

2.1 Konsep Dasar NoSQL

Bagian ini akan menjelaskan fitur yang dimiliki oleh basis data NoSQL terutama dalam hal distribusi data dan *query* dalam beberapa server yang sama. Telah banyak produk basis data NoSQL dengan berbagai keunggulannya masing – masing. Eric brewer [1] menyatakan bahwa NoSQL didasarkan pada teori CAP yaitu pemilihan dua dari tiga aspek yang ada yang harus dipenuhi oleh basis data yaitu *consistency*, *availability* dan *partition tolerance*.

Dalam basis data NoSQL penerapan konsep tersebut diterjemahkan dalam empat konsep dasar yaitu :

1. *Non Relational*,

Konsep Non-Relational dalam basis data NoSQL meliputi hirarki, graf, dan basis data berorientasi obyek yang sudah terlebih dahulu ada sejak tahun 1960 sebelum akhirnya basis data relasional muncul pada tahun 1970 [2].

Penggunaan basis data *non relational* kembali marak seiring dengan bertambahnya aplikasi berbasis web yang memerlukan banyak penyimpanan data.

Meskipun memiliki kelemahan pada konsistensi dan redundansi data, namun basis data *non relasional* dapat menyelesaikan beberapa permasalahan terkait dengan *availability* dan *partition tolerance*.

Tugas pengecekan konsistensi dan redundansi data diserahkan pada sisi aplikasi, sedangkan basis data *non relational* hanya bertugas memanipulasi penyimpanan saja.

2. *MapReduce*

merupakan model pemrograman yang diadaptasi dari pemrograman fungsional yang diimplementasikan

untuk mengolah dataset yang sangat besar.

Tujuan dari *MapReduce* adalah merancang suatu abstraksi baru yang memungkinkan pengguna untuk membuat antarmuka pemrograman sederhana dan menyembunyikan detail yang rumit dari paralelisasi, *fault-tolerance*, distribusi data, dan *load balancing* dalam pustaka pemrogramannya.

Hasilnya menunjukkan bahwa penerapan *MapReduce* dapat menyederhanakan antarmuka pemrograman yang dapat mendukung paralelisasi dan distribusi komputasi skala besar secara otomatis [3].

Pemrograman dengan *MapReduce* telah sukses diterapkan penggunaannya oleh Google untuk berbagai tujuan, salah satunya adalah *Google indexing*.

Dalam kinerjanya, Google menerapkan ribuan mesin yang bekerja pada ratusan terabytes data dengan lokasi server yang tersebar di beberapa lokasi.

Jenis arsitektur seperti ini dapat mengurangi waktu kinerja yang diperlukan. Pembangunan arsitektur Google dengan menggunakan

MapReduce memerlukan waktu hanya dalam beberapa jam saja dibandingkan dengan tidak menerapkan *MapReduce* yang memerlukan waktu selama berbulan-bulan.

Penggunaan *e-library* dalam *MapReduce* memiliki beberapa keuntungan seperti *load balancing*, optimasi perangkat penyimpanan yang nantinya akan meningkatkan keefisienan sistem dan kemudahan dalam penggunaannya.

Mudahnya, *MapReduce* bekerja dengan membagi proses menjadi dua fase, yaitu tahap *map* dan tahap *reduce*.

3. Schema Free

NoSQL dan RDBMS mempunyai perbedaan dalam hal penerapan skema basis data. Dalam basis data relasional, sebuah tabel didesain dengan peraturan skema yang ketat. NoSQL menyimpan data dengan aturan yang lebih longgar, artinya tidak seperti basis data yang berdasarkan SQL tradisional, NoSQL tidak memiliki tabel, kolom, primary dan *foreign key*, *join*, dan relasi .

Dalam pengembangan basis data relasional, developer / database

administrator harus berhati-hati dalam menentukan bagaimana tabel saling berelasi dan field yang ada didalam setiap tabel. Karena perubahan skema dalam RDBMS dapat menimbulkan masalah ketergantungan dan integritas, seperti timbulnya kolom null dan relasi kunci yang tidak cocok. Hal ini bukan masalah dalam NoSQL karena adanya penerapan schemafree. Setiap dokumen bertanggung jawab terhadap isinya sendiri, maksudnya null value dapat dihilangkan dalam beberapa baris, dan field baru dapat didefinisikan dalam setiap dokumen secara independen [17].

Salah satu produk basis data NoSQL yang menerapkan fitur schema-free adalah Cassandra. Dalam Cassandra, pengembang hanya perlu mendefinisikan keyspace sebagai pertimbangan container dan contains column families. Keyspace hanya digunakan sebagai logical namespace yang dapat dimasukkan dalam configurations dan hold column families [18]. Column families artinya sebuah nama untuk relasi data dan hirarki kolom (memungkinkan penyisipan tabel dalam kolom). Disamping itu, kita

hanya perlu menambahkan data dalam tabel, menggunakan kolom, tanpa menentukan kolomnya terlebih dahulu. Dalam Cassandra kita hanya perlu mendefinisikan tabel namespace dan hirarki kolom didalamnya, memodelkan dan mendesain tabel join adalah masalah dalam basis data relasional bukan NoSQL.

Manfaat lain dalam penggunaan schema-free adalah penghematan dalam media penyimpanan. Dalam basis data relasional, setiap field yang ada dalam tabel harus mempunyai nilai, walaupun nilai itu null. Model data schema free artinya setiap baris memungkinkan memiliki nilai sebanyak yang telah didefinisikan dalam tiap fields, dan tidak perlu menggunakan nilai yang memang tidak diperlukan [18].

Kelemahan dalam penggunaan schema free adalah memunculkan lemahnya pendefinisian struktur yang memungkinkan terjadinya penggunaan basis data yang tidak konsisten. Jika tujuan pembangunan basis data didasarkan pada konsistensi yang ketat, seperti wiki, document management systems, discussion forums, blogs, dan

support management systems, basis data relasional masih merupakan pilihan yang tepat [17].

2.2 Sistem Informasi, PHP dan MongoDB

Sistem informasi adalah pengaturan orang, data, proses dan teknologi informasi yang berinteraksi untuk mengumpulkan, memproses, menyimpan dan menyediakan sebagai *output* informasi yang diperlukan untuk mendukung sebuah organisasi [3].

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diberikan [4]

Menurut [5] **PHP (Hypertext Preprocessor)** merupakan sebuah bahasa pemrograman yang berbentuk scripting dimana sistem kerja dari program ini adalah sebagai interpreter bukan sebagai compiler. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin

(Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui *console* serta juga dapat menjalankan perintah - perintah sistem.

MongoDB merupakan sebuah sistem basis data yang berbasis dokumen (Document Oriented Database) dan termasuk sistem basis data yang menganut paham NoSQL. NoSQL singkatan dari Not Only SQL, artinya sebuah sistem basis data tidak hanya harus menggunakan perintah SQL untuk melakukan proses manipulasi data.

MongoDB tidak memiliki tabel, kolom, dan baris. Pada MongoDB yang ada hanyalah koleksi dan dokumen. Dokumen yang terdapat dalam mongoDB dapat memiliki atribut yang berbeda 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.

Pertama kali dikembangkan oleh perusahaan asal New York City, 10gen (sekarang MongoDB Inc.) pada bulan Oktober 2007 sebagai

bagian dari platform yang direncanakan sebagai produk jasa, perusahaan bergeser ke model pembangunan sumber terbuka pada tahun 2009, dengan 10gen menawarkan dukungan komersial dan layanan lainnya. Sejak itu, MongoDB telah diadopsi sebagai perangkat lunak backend oleh sejumlah situs dan layanan, termasuk Craigslist, eBay, Foursquare, SourceForge, dan The New York Times. MongoDB adalah sistem basis data NoSQL yang paling populer.

MongoDB merupakan sistem basis data yang menggunakan konsep key-value, artinya setiap dokumen dalam mongodb pasti memiliki key. Hal ini berbeda dalam RDBMS yang 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.

Kelebihan MongoDB :

1. Performa yang ditawarkan MongoDB lebih cepat disebabkan oleh memcached dan format dokumennya yang berbentuk seperti JSON
2. Kita tidak perlu membuat struktur tabel, karena MongoDB akan otomatis membuat struktur tabelnya pada saat proses insert (fleksibel skema)

Kekurangan MongoDB :

1. Belum banyak hosting yang support.
2. Fleksibilitas dalam query (sebagai contoh tidak adanya JOIN).
3. Selain itu, karena bervariasinya produk dan format penyimpanan, berpindah antar satu produk database ke produk noSQL lainnya perlu waktu untuk belajar.

Contohnya ketika anda pindah dari MongoDB ke Cassandra, maka anda harus belajar lagi dari awal, berbeda dengan database RDMS.

SQL	Mongo DB
database	Database
table	Collection
row	Document atau BSON document
kolom	Field
index	Index
table join	embeded document dan link

Gambar 1. Terminologi dan konsep mongodb dan perbandingan dengan konsep yang ada di sql database.

III. ALUR (TAHAPAN) PERANCANGAN SISTEM DENGAN MONGODB

A. TAHAP PENDAHULUAN

1. Instalasi dan Konfigurasi Database MongoDB

Pada tahap awal ini, membahas bagaimana cara instalasi mongoDB di Windows. Dalam instalasi ini saya menggunakan Windows 7 32 bit, Instalasi Mongo Akan diletakan di:

```
d:\xampp\mongodb.
```

Langkah instalasi :

1. Download software mongo dari <http://www.mongodb.org/download>.

versi terakhir saat tulisan ini dibuat adalah 4.0.5. MongoDB juga mendukung platform Linux, Mac dan Solaris, akan tetapi jurnal ini membatasi pembahasan pada Instalasi di Windows.

2. Selanjutnya, extract file :

```
mongodb-win32-x86_64-008plus-ssl-4.0.5.zip
```

ke folder XAMPP. rename foldernya menjadi mongodb.

Buatlah folder data di :

```
d:\xampp\mongodb\data sebagai lokasi penyimpanan database.
```

2. Cara Menjalankan Server MongoDB

Berikut konfigurasi untuk menjalankan Server MongoDB :

1. Buka command prompt dan masuk ke :

```
d:\xampp\mongodb\bin
```

dan gunakan perintah dibawah ini untuk menjalankan server mongodb.

```
mongod.exe --dbpath=d:\xampp\mongodb\data
```

2. Jika perintah diatas berhasil dijalankan maka akan muncul teks panjang yang diakhiri dengan baris seperti dibawah ini :

```
...
...
un Mar 02 16:42:10.749 [initandlisten] waiting
for connections on port 27017
```

3. Install MongoDB sebagai Windows Service.

1. MongoDB juga bisa diinstall sebagai Windows service sehingga anda tidak perlu repot repot menjalankannya secara manual.
2. Caranya buka kembali Command prompt dengan akses super user (run as Administrator). Lalu masuk ke folder :

```
d:\xampp\mongodb\bin
```

3. Jalankan perintah ini.

```
mongod.exe -install -journal -
logpath d:\xampp\mongodb\
mongo.log
```

```
dbpath=d:\xampp\mongodb\data
```

4. Jika perintah diatas sukses, maka di service windows (services.msc) sudah ada Mongoddb.

4. Manajemen Windows Service MongoDB

Jika Mongo dijalankan sebagai service windows, maka berikut ini adalah perintah untuk :

start, stop, restart dan uninstall mongoddb dari windows service :

1. Start Service :

```
net start mongoddb
```

2. Stop Service

```
net stop mongoddb
```

3. Restart Service

```
net restart mongoddb
```

4. Uninstall mongoddb dari service windows

```
d:\xampp\mongodb\mongod.exe -
remove
```

5. Cara Mengakses MongoDB Server

1. Mengakses mongoddb server, bukalah satu lagi Command Prompt dan masuk ke folder :

```
d:\xampp\mongodb\bin
```

Jalankan perintah dibawah ini :

```
D:\xampp\mongodb\bin>mongo 127.0.0.1
MongoDB shell version: 2.4.8
connecting to: 127.0.0.1/test
>
```

B. TAHAP PENGEMBANGAN

1. Analisis Kebutuhan Sistem (System Requirements)

Analisis kebutuhan sistem sangat dibutuhkan dalam mendukung kinerja sistem, apakah sistem yang dibuat telah sesuai dengan kebutuhan atau belum karena kebutuhan sistem akan mendukung tercapainya tujuan.

Kebutuhan sistem ini bertujuan untuk memetakan apa saja yang dibutuhkan untuk merancang basis data NoSQL MongoDB pada sistem informasi akademik. Kebutuhan sistem dalam penelitian ini dibagi menjadi dua jenis yaitu kebutuhan fungsional dan non fungsional yaitu :

1. Kebutuhan Fungsional

Menurut [9] kebutuhan fungsional merupakan jenis kebutuhan yang berisi proses - proses apa saja yang nantinya dilakukan oleh sistem.

Kebutuhan fungsional juga berisi informasi - informasi apa saja yang harus ada dan dihasilkan oleh sistem.

Salah satu kebutuhan fungsional sebagai berikut :

1. Dosen

- a. Sistem layanan harus dapat melakukan *login* ke sistem dan hal yang berhubungan dengan *activity* yang dapat dilakukan oleh dosen setelah *login*.

- b. *User* dosen dapat memasukkan *username* dan *password login* melalui sistem.

- c. *User* dosen dapat *logout* sistem dan *login* kembali dengan *username* dan *password* yang sesuai datanya dengan *database* dosen.

- d. *User* dosen dapat melihat *profil* dosen berdasarkan akses *login user* dan memperoleh gambaran singkat tentang sistem layanan informasi akademik tersebut.

- e. *User* dosen dapat meng-*input* data mengenai mata kuliah, jadwal dan fasilitas yang disediakan, setelah *login* kedalam sistem.

- f. *User* dosen dapat melihat (*view*) daftar mahasiswa melalui sistem layanan informasi mahasiswa.

2. Admin

Sistem harus dapat memberikan hak akses kepada *admin* sebagai *super user* :

1. *User admin* dapat *login* dengan meng-*input username* dan *password login* untuk dapat masuk kedalam sistem.

2. *User admin* dapat memilih menu yang ada di pilihan daftar menu.
 3. *User admin* layanan informasi dapat menambahkan, meng-*update*, menghapus dan melihat data *user*.
 4. *User admin* dapat menambahkan data mahasiswa, dosen dan menentukan jadwal perkuliahan dan kelas.
 5. *User admin* dapat menambahkan, memperbaiki data yang di-*input* oleh dosen dan mahasiswa melalui sistem.
3. Mahasiswa
- Sistem harus dapat menampilkan data yang telah diolah oleh *admin* untuk dapat dilihat oleh mahasiswa :
1. Mahasiswa dapat *login* dengan meng-*input* *username* dan *password* *login* untuk dapat masuk kedalam sistem.
 2. Mahasiswa dapat memilih menu yang ada di pilihan daftar menu.
 3. Mahasiswa dapat melihat data informasi mata kuliah, penjadwalan melalui sistem layanan akademik.
 4. Mahasiswa melalui menu dapat menambahkan, mengubah dan menghapus data *profil* mahasiswa setelah *login* kedalam sistem.
- ## 2. Kebutuhan *Non Fungsional*
1. Operasional

Untuk menjalankan sistem ini baik sebagai *user* dosen, mahasiswa dan *admin* membutuhkan komputer dengan spesifikasi minimum sebagai berikut :

 1. Processor Intel Celeron, Pentium IV atau Higher Processor.
 2. 1 – 2 Gb of RAM
 3. Harddisk minimum kapasitas 20 Gb atau lebih.
 4. Monitor dengan resolusi minimal 1024 x 768.
 5. VGA Card 8 MB, Mouse dan keyboard.
 2. Keamanan

Dilengkapi dengan *password* pada saat *login* bagi *user* yang akan menggunakan sistem.
 3. Informasi
 1. Digunakan untuk menginformasikan apabila *username* dan *password*

login yang dimasukkan pengguna sistem tersebut salah.

2. Digunakan untuk menampilkan informasi mengenai layanan akademik dan informasi penting lainnya bagi mahasiswa dan dosen.

2. Hasil

Berdasarkan analisis dan desain yang telah dilakukan, maka diperoleh Rancangan Basis Data Non Relasional NoSQL MongoDB pada Sistem Informasi Akademik dengan akses data informasi lebih cepat, tepat dan terstruktur dan dalam pencarian data menjadi lebih cepat, aktual dan optimal.

3. Implementasi antarmuka

Implementasi antarmuka dalam penelitian ini bertujuan untuk dapat memberikan gambaran dan sebagai mekanisme komunikasi antara pengguna (*user*) dengan sistem. Antarmuka pemakai (*user interface*) dapat menerima informasi dari pengguna (*user*) dan memberikan informasi kepada pengguna untuk membantu mengarahkan ditemukan suatu solusi.

Berikut implementasi antarmuka tersebut

1. Rancangan Halaman Utama Sistem Informasi Akademik



Gambar 2 : Halaman Utama Sistem

Pada gambar 2 diatas, menampilkan halaman utama sistem informasi akademi yang akan digunakan untuk *user* sistem.

2. Halaman *entry user* mahasiswa sebagai pengguna sistem.



Gambar 3 : Form *entry* mahasiswa sebagai pengguna sistem

Pada gambar 3 diatas, menampilkan halaman *form entry* mahasiswa sebagai pengguna sistem. Halaman ini digunakan untuk mendata mahasiswa yang nantinya dapat masuk ke dalam sistem layanan.

3. Halaman *Entry* Mata Kuliah



Gambar 4 : Halaman *Entry* Mata Kuliah

Pada gambar 4 diatas, menampilkan halaman *entry* mata kuliah. Halaman ini digunakan oleh admin untuk menambahkan mata kuliah yang akan ditampilkan melalui sistem.berikan kepada pencari kost yang telah melakukan registrasi dan *login* kedalam sistem.

4. Halaman *Form Entry* Kelas Mahasiswa.



Gambar 5 : *Form Entry* Kelas Mahasiswa

Pada gambar 5 diatas, menampilkan halaman *form entry* kelas mahasiswa. Halaman ini digunakan untuk meng-*entry* kelas mahasiswa agar mahasiswa dapat mengelompokkan mahasiswa setiap kelasnya.

IV. Kesimpulan dan Saran

4.1 Kesimpulan

Adapun kesimpulan dari penelitian ini sebagai berikut:

1. Rancangan Basis Data Non Relasional NoSQL MongoDB diharapkan dapat menyelesaikan semua permasalahan - permasalahan yang berkaitan dengan pelayanan akademik melalui sistem informasi akademik.
2. Rancangan Basis Data Non Relasional NoSQL MongoDB ini

diharapkan dapat membantu proses penyampaian informasi kepada seluruh *user* (pengguna) pada perguruan tinggi.

4.2 Saran

Saran yang dapat diberikan sehubungan dengan penulisan dari pembahasan ini, penulis mencoba memberikan saran – saran diantaranya sebagai berikut :

1. Rancangan Basis Data Non Relasional NoSQL MongoDB ini hendaknya dapat diimplementasikan dan dievaluasi, diperbarui dan dikontrol untuk membantu dalam memberikan informasi kepada *user* (pengguna) layanan akademik di Perguruan Tinggi.
2. Pada pengembangan berikutnya, untuk melanjutkan Rancangan Basis Data Non Relasional NoSQL MongoDB ini dapat ditambahkan fitur tambahan pendukung pembelajaran akademik antara dosen dan mahasiswa, disediakan *form* kritik dan saran sehingga diharapkan dapat menghasilkan layanan informasi akademik yang lebih baik lagi.

DAFTAR PUSTAKA

- Al fatta, Hanif, 2007, Analisis dan perancangan system informasi untuk keunggulan perusahaan dan organisasi kelas dunia, Andi offset – STMIK AMIKOM Yogyakarta, Yogyakarta
- Jogiyanto 2005, *Analisis dan Desain Analisis Sistem Informasi Pendekatan Terstruktur dan Praktek Aplikasi Bisni, Edisi 3*, Andi Offset, Yogyakarta.
- Nugroho, Bunafit 2004, *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL*, Gaya Media, Yogyakarta.
- Pratikto, HS, *et all.* 2014. *Jurnal : SCRIPT, Sistem Pencarian dan Pemesanan Rumah Kos Menggunakan Sistem Informasi Geografi.* Yogyakarta.
- Suryana, Taryana, *SMS Gateway Kannel sebagai sarana penunjang informasi akademik*”, in *Proc. KOMPUTA Volume I*, pp.17-30, Oktober, 2014
- Susano, Adi, dkk, *Terapan Sistem Kecerdasan Buatan pada Sistem Informasi Akademik Berbasis SMS Gateway menggunakan Metode Breadth First Search*”, in *Proc. Semnasteknomedia 2014*, pp.12-26, Februari 8, 2014.
- Whitten, Jeffery L 2006, *Metode Desain dan Desain Analisis*

Sistem, Edisi 6, Andi Offset,
Yogyakarta.

Wibisono, B. 2011. *Perbedaan E-Commerce dan E-Bisnis Informasi*. Yogyakarta

Zin Myint Naung, Phyoe Mon Mon, *Information Systems Requirement Gathering using FAST Framework : Critical Analysis*.