

ABSTRAK

Dokumen teks tergolong dalam data tidak terstruktur. Jika dibandingkan dengan informasi dalam bentuk yang terstruktur seperti dalam tabel pada sebuah database. Maka data tidak terstruktur cukup sulit dalam proses pengelolaan, penyimpanan dan pencarian. Salah satu metode untuk memudahkan dalam proses pencarian teks adalah *Inverted Index*. Secara intuitif, indexing mirip dengan indeks yang terdapat pada buku. Pada buku yang cukup tebal sering dijumpai halaman “Indeks” yang cara kerjanya sama dengan inverted index. Tujuan menggunakan *Inverted Index* adalah untuk meningkatkan kecepatan dan efisiensi dalam proses pencarian pada sekumpulan dokumen. Dengan mengindeks dokumen yang akan dicari, maka pencarian dokumen dengan query tertentu tidak perlu dilakukan secara sekuensial atau diperiksa satu persatu. Proses *Indexing* terdiri dari beberapa tahapan yaitu, *cleansing*, lower case kalimat, stopword removal, *stemming*, dan pembentukan index. Proses dilakukan pada data Twitter yang diambil setiap detik oleh sistem pada regional Indonesia. Indeks yang dihasilkan digunakan dalam proses pencarian dokumen teks. Baik hasil pencarian maupun waktu yang dibutuhkan semua di uji dan dicatat untuk mengukur performa sistem. Hasil menunjukkan bahwa setelah di indexing proses pencarian menjadi lebih cepat dan efisien. Untuk menampung data yang besar diperlukan sebuah sistem yang mumpuni. Pada umumnya, Perusahaan besar seperti Twitter tidak menggunakan Relational DataBase Management System (RDBMS) dalam pengolahan datanya, melainkan menggunakan NoSQL. Kelebihan utama dari NoSQL adalah untuk menangani Big Data dimana data terus-menerus berkembang, dimana data tersebut sangat kompleks sehingga sebuah database relational tidak lagi bisa mengakomodir. Harus ada cara lain untuk menanggulangi masalah ini, dan NoSQL lah solusinya. NoSQL menyederhanakan proses yang terjadi dalam sistem basis data relasional. Dimana hal-hal yang menyebabkan redundansi, dihilangkan sehingga trafik server akan seimbang. Salah satu produk NoSQL yang mumpuni adalah MongoDB. dilakukan juga pengujian dan dicatat mengenai perbedaan performa dalam penggunaan MySQL untuk RDBMS dan MongoDB untuk NoSQL. Hasil menunjukkan penggunaan NoSQL lebih cepat dibandingkan RDBMS.

Kata Kunci: *Inverted Index*, MongoDB, MySQL, NoSQL, RDBMS

ABSTRACT

Text documents classified in unstructured data. When compared with information in a form that is structured like a table in a database. Then the unstructured data is quite difficult in process management, storage and search. One method to simplify the search process is the text of Inverted Index. Intuitively, indexing similar to the index contained in the book. In the book thick enough often encountered the "Index" which works the same way with the inverted index. The purpose of using Inverted Index is to improve the speed and efficiency in the process of searching in a collection of documents. By indexing the documents to be searched, then the search documents with certain query does not need to be done sequentially or checked one by one. Indexing process consists of several stages, namely, cleansing, lower case sentence, stopword removal, stemming, and the formation index. The process is done on the Twitter data is taken every second by the system at the regional Indonesia. The resulting index is used in the process of searching text documents. Both the search results and the time required all tested and recorded to measure system performance. Results showed that after indexing search process becomes faster and more efficient. To accommodate the large data needed a system that qualified. In general, a large company like Twitter do not use the Relational DataBase Management System (RDBMS) in data processing, but instead use NoSQL. The main advantages of NoSQL is to handle Big Data where data is constantly evolving, where the data is so complex that a relational database can no longer accommodate. There must be another way to tackle these problems, and NoSQL is a solution. NoSQL simplify the processes that occur in a relational database system. Where are the things that cause redundancies, eliminated so that traffic will be balanced server. One product is the MongoDB NoSQL qualified. do also test and note the difference in performance in the use of MySQL to MongoDB's RDBMS and NoSQL. The results showed use of NoSQL faster than RDBMS.

Keywords: Inverted Index, MongoDB, MySQL, NoSQL, RDBMS

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS LAPORAN PENELITIAN.....	ii
PERNYATAAN PUBLIKASI LAPORAN PENELITIAN	iii
PRAKATA.....	iv
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR NOTASI/ LAMBANG.....	xiii
DAFTAR SINGKATAN	xvi
DAFTAR ISTILAH	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Pembahasan	2
1.4 Ruang Lingkup.....	3
1.5 Sumber Data.....	3
1.6 Sistematika Penyajian	3
BAB 2 KAJIAN TEORI	5
2.1 Scalability.....	5
2.2 Inverted Index.....	5
2.3 Twitter	5
2.4 Twitter API	6

2.5 Mongo Database	6
2.6 ERD(Entity Relationship Diagram).....	7
2.7 Unified Modelling Language(UML).....	7
BAB 3 ANALISIS DAN RANCANGAN SISTEM.....	8
3.1 Rancangan Eksperimen	8
3.1.1 <i>Crawling</i> / Pengumpulan Data	8
3.1.2 Langkah Pendaftaran Aplikasi Twitter	8
3.1.3 Aturan twitter	9
3.1.4 Batasan pada Twitter API	9
3.1.5 <i>Indexing</i>	10
3.2 Rancangan Basis Data.....	13
3.2.1 <i>Entity Relationship Diagram</i>	13
3.2.2 Transformasi ERD ke dalam Table	13
3.3 Desain Perangkat Lunak	14
3.3.1 Pemodelan Perangkat Lunak.....	14
3.3.1.1 <i>Use Case Diagram</i>	15
3.3.1.1.1 Rancangan Use Case Diagram	15
3.3.1.1.2 Deskripsi <i>Use Case Diagram</i>	15
3.3.1.2 Activity Diagram.....	16
3.3.1.2.1 Setting <i>Crawling</i>	16
3.3.1.2.2 Setting <i>Indexing</i>	17
3.3.1.2.3 Start <i>Crawling</i>	18
3.3.1.2.4 Start <i>Indexing</i>	19
3.3.1.2.5 Stop <i>Crawling</i>	20
3.3.1.2.6 Stop <i>Indexing</i>	21
3.3.2 Rancangan Antarmuka	23

BAB 4 IMPLEMENTASI.....	25
4.1 <i>Class Diagram</i>	25
4.2 Implementasi Antarmuka <i>Search</i>	25
4.3 Implementasi Antarmuka <i>Crawling</i>	26
4.4 Implementasi Antarmuka <i>Indexing</i>	26
BAB 5 PENGUJIAN	28
5.1 Pengujian <i>Search Inverted Index</i>	29
5.2 Pengujian Perbandingan MongoDB dan MySQL dalam konteks <i>Scalability</i>	29
5.2.1 Select Data	30
5.2.2 Insert Data	32
5.2.3 Perbandingan Pengambilan Data Setelah <i>Indexing</i> dan Sebelum <i>Indexing</i>	33
BAB 6 SIMPULAN DAN SARAN.....	34
6.1 Simpulan	34
6.2 Saran.....	34
DAFTAR PUSTAKA	1

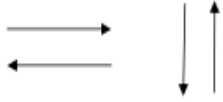
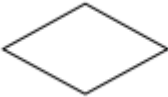
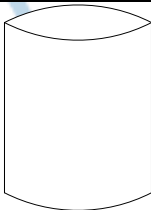
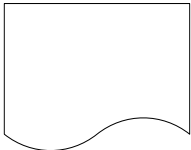
DAFTAR GAMBAR

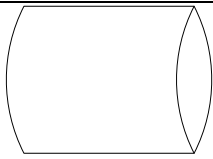
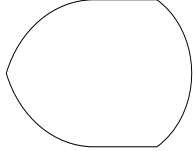
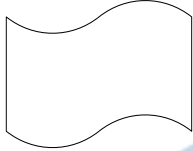
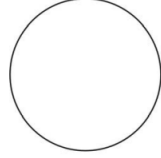
Gambar 3.1 Tahap indexing	10
Gambar 3.2 Entity Relationship Diagram	13
Gambar 3.3 Use Case Diagram	15
Gambar 3.4 Activity Diagram Setting <i>Crawling</i>	17
Gambar 3.5 Activity Diagram Setting <i>Indexing</i>	18
Gambar 3.6 Activity Diagram Start <i>Crawling</i>	19
Gambar 3.7 Activity Diagram Start <i>Indexing</i>	20
Gambar 3.8 Activity Diagram Stop <i>Crawling</i>	21
Gambar 3.9 Activity Diagram Stop <i>Indexing</i>	22
Gambar 3.10 Rancangan Antarmuka Search	23
Gambar 3.11 Rancangan Antarmuka <i>Crawling</i>	24
Gambar 3.12 Rancangan Antarmuka <i>Indexing</i>	24
Gambar 4.1 <i>Class Diagram</i>	25
Gambar 4.2 Implementasi Antarmuka Search	26
Gambar 4.3 Implementasi Antarmuka <i>Crawling</i>	26
Gambar 4.4 Implementasi Antarmuka <i>Indexing</i>	27
Gambar 5.1 Contoh <i>Crawling</i>	28
Gambar 5.2 Contoh <i>Indexing</i>	29
Gambar 5.3 Contoh Pengujian Select Data MySQL dan MongoDB	30
Gambar 5.4 Contoh Pengujian Indexing	33

DAFTAR TABEL

Tabel 3.1 Contoh Kasus	10
Tabel 3.2 Hasil <i>Cleansing</i>	11
Tabel 3.3 Hasil <i>Lower Case</i> Kalimat	11
Tabel 3.4 Hasil Stop Word <i>Removal</i> / Filtering	11
Tabel 3.5 Hasil <i>Stemming</i>	12
Tabel 3.6 Contoh Sample Hasil Pembangunan <i>Index</i>	12
Tabel 3.7 Tabel User	13
Tabel 3.8 Tabel Tweet	14
Tabel 3.9 Tabel <i>Index</i>	14
Tabel 3.10 Use Case Diagram Setting <i>Crawling</i>	15
Tabel 3.11 Use Case Setting <i>Indexing</i>	15
Tabel 3.12 Use Case Start <i>Crawling</i>	16
Tabel 3.13 Use Case Start <i>Indexing</i>	16
Tabel 3.14 Use Case Stop <i>Crawling</i>	16
Tabel 3.15 Stop <i>Indexing</i>	16
Tabel 5.1 Pengujian <i>Search Inverted Index</i>	29
Tabel 5.2 Pengujian Select Data	31
Tabel 5.3 Pengujian Pengambilan Data Dengan Query	31
Tabel 5.4 Pengujian Insert Data Dengan Query	32
Tabel 5.5 Perbandingan Sebelum dan Sesudah <i>Indexing</i>	33

DAFTAR NOTASI/ LAMBANG

Jenis	Notasi/Lambang	Nama	Arti
Flowchart		Simbol arus / <i>flow</i>	Menyatakan jalannya arus suatu proses
Flowchart		Simbol <i>process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
Flowchart		Simbol manual	Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
Flowchart		Simbol <i>decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan
Flowchart		Simbol terminal	Menyatakan permulaan atau akhir suatu program
Flowchart		Simbol Manual <i>input</i>	Memasukkan data secara manual dengan menggunakan <i>online keyboard</i>
Flowchart		Simbol hard disk	Menunjukkan <i>input/output</i> menggunakan <i>harddisk</i> .
Flowchart		Simbol Penghubung	Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.
Flowchart		Simbol Dokumen	Menunjukkan dokumen input atau output, untuk proses manual maupun mekanik.

Jenis	Notasi/Lambang	Nama	Arti
Flowchart		Simbol drum magnetik	Menunjukkan <i>input / output</i> menggunakan drum magnetik
Flowchart		Simbol Display	Menunjukkan <i>output</i> yang ditampilkan di monitor.
Flowchart		Simbol pita kertas berlubang	Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang.
ERD		Entitas	Menunjukkan sebuah objek yang dapat dibedakan dengan objek lainnya
ERD		Atribut	Mendeskripsikan karakter entitas
ERD		Relasi	Menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda
ERD		Garis (<i>one to many</i>)	Penghubung antar relasi dan entitas dimana satu entitas dapat memiliki lebih dari satu hubungan.
ERD		total participation constraint	Keberadaan suatu entity tergantung pada hubungannya dengan entity lain.
DFD		Alur Data	Menunjukkan alur data (informasi/objek) yang mengalir. Nama alur data menunjukkan nama dari data yang mengalir tersebut, dan bisa lebih dari satu.
DFD		Proses	Menunjukkan tugas atau proses yang dilakukan baik secara manual atau otomatis dan menunjukkan angka referensi dari proses tersebut

Jenis	Notasi/Lambang	Nama	Arti
DFD		Entitas Eksternal	Merupakan simbol entitas eksternal untuk menunjukkan tempat asal data (sumber) atau tempat tujuan data (Tujuan). Nama entitas eksternal (terminator) ditulis dalam bentuk tunggal.
DFD		Penyimpanan Data	Simbol ini menunjukkan gudang informasi atau data. Sangat sering terjadi bahwa unsur-unsur data tidak berjalan dari suatu proses ke proses berikutnya secara langsung, melainkan disimpan terlebih dahulu, sementara operasi lainnya atau penyusunan ulang unsur-unsur data berlangsung.

DAFTAR SINGKATAN

CSS	<i>Cascading Style Sheet</i>
ERD	<i>Entity Relationship Diagram</i>
HTML	<i>Hyper Text Markup Language</i>
PHP	<i>Hypertext Preprocessor</i>



DAFTAR ISTILAH

<i>Aplikasi</i>	suatu perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan sesuatu tugas.
<i>Form</i>	tempat untuk mengontrol proses masukan data, menampilkan data, memeriksa, dan memperbaharui data.
<i>System</i>	suatu kesatuan yang terdiri komponen yang dihubungkan bersama untuk memudahkan aliran informasi.
<i>Admin</i>	pengguna yang memiliki hak akses untuk pengelolaan suatu data dan dapat memberikan akses kepada pengguna tertentu.
<i>User</i>	pengguna.
<i>Field</i>	Sebuah baris dalam tabel pada database yang diisi oleh <i>user</i> dengan melakukan pengisian data.

