

MySQL *Full-Text Searching*

Didik Setiawan

di2k.setiawan@gmail.com

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Sebuah metode pencarian pada sebuah sistem aplikasi atau database harus mampu memberikan hasil sebagaimana harapan pengguna. Kecepatan dan relevansi hasil pencarian merupakan syarat mutlak yang harus dipenuhi.

Terdapat beberapa metode dalam pencarian, baik dari sisi aplikasi atau pun berdasarkan fungsionalitas yang tertanam dalam sebuah sistem database. Full-text Searching adalah salah satu fungsi yang terdapat dalam database MySQL yang dapat digunakan untuk mengoptimalkan proses dan hasil pencarian.

Pendahuluan

Penggunaan direktori dalam sebuah situs website dengan cara mengelompokkan artikel berdasarkan kategori untuk kemudian dikelompokkan pada direktori merupakan salah satu upaya untuk memudahkan pengguna (user) menemukan artikel yang dikehendaknya diantara banyaknya artikel yang ada.

Permasalahan akan timbul saat artikel yang dikehendaki tidak terdapat pada kelompok direktori tertentu yang sesuai dengan topik yang ada atau artikel yang bersesuaian tersebar di banyak direktori, disinilah fungsi pencarian dapat membantu pengguna dalam memecahkan masalah tersebut.

Salah satu metode yang dapat digunakan untuk melakukan proses pencarian adalah dengan memanfaatkan fungsi *indexing* pada database. Tulisan ini memperkenalkan fungsi *Full-Text Searching* pada database MySQL untuk proses pencarian data.

Definisi

Berikut adalah definisi atau pengertian dari beberapa istilah yang sering digunakan dalam tulisan ini :

- a. MySQL, adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang bersifat *multithread* dan *multiuser* yang dikembangkan pertama kali oleh Michael Wildenius, bersifat *free software* dengan lisensi dibawah GNU/GPL (*General Public license*).
- b. *Full-Text*, adalah fungsionalitas yang terdapat pada database (dalam hal ini database MySQL) yang memungkinkan user untuk melakukan pencarian tertentu dalam tabel dengan cara melakukan perbandingan *string*.
- c. *Boolean Search*, adalah sebuah metode pencarian yang memungkinkan pengguna untuk mempersempit hasil pencarian dengan menggunakan operator *Boolean*.
- d. *Boolean Operator*, adalah sistem logis deduktif yang digunakan untuk mempersempit hasil pencarian dengan menggunakan operator AND, OR, XOR dan lainnya.

Prasyarat

Sebelum mengenal lebih jauh mengenai *Full-Text Searching* ada beberapa hal (prasyarat) yang perlu diketahui bersama :

- a. *Full-Text Searching* pada tulisan ini mengacu pada fungsionalitas yang terdapat pada database MySQL;
- b. *Full-text index* pada MySQL adalah index dengan tipe `FULLTEXT`;
- c. *Full-text index* hanya dapat digunakan pada database `MyISAM` dan dapat digunakan untuk kolom dengan tipe `CHAR`, `VARCHAR`, atau `TEXT`;
- d. *Full-text index* dapat dibuat saat proses pembuatan tabel dengan perintah "`CREATE TABLE`" atau saat perubahan struktur table yang sudah ada dengan menggunakan perintah "`ALTER TABLE`" atau "`CREATE INDEX`".

Skenario

Tulisan ini menggunakan tabel `article` sebagai contoh untuk menggambarkan penggunaan *Full-text Searching* di database MySQL, adapun struktur tabel `article` yang digunakan adalah sebagai berikut :

```
CREATE TABLE IF NOT EXISTS article (id int(10) NOT NULL AUTO_INCREMENT, title
varchar(150) NOT NULL, body text NOT NULL, PRIMARY KEY (id))
ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

Adapun contoh data yang digunakan adalah sebagai berikut :

```
INSERT INTO `article` (`id`, `title`, `body`) VALUES
(1, 'PHP & Mysql : Tutorial Dasar', 'Pada tutorial kali ini, kita akan mencoba
melakuka...'),
(2, 'MySQL - Pengenalan', 'Tutorial MySQL di situs ini memang buat kamu-kamu
...'),
(3, 'Mysql', 'Tutorial penggunaan database masa kini...'),
(4, 'Mysql & Oracle Database', 'Database masa kini...'),
(5, 'Implementasi Oracle', 'Bagaimana menggunakan oracle...'),
(6, 'SqlLite Database', 'Penggunaan SqlLite dan cpanel...'),
(7, 'let it be', 'by The Beatles...');
```

Pencarian data secara “konvensional” biasanya dilakukan dengan menggunakan perintah SQL `LIKE`. Contoh : pencarian artikel yang mengandung kata MySQL dapat dilakukan dengan perintah :

```
SELECT * FROM article WHERE title like'%mysql%';
```

Penggunaan operator `like` pada MySQL untuk jumlah data yang besar akan menyebabkan turunnya performa.

Implementasi

Full-Text Searching dapat menjadi pilihan untuk melakukan fungsi pencarian pada database MySQL. Syarat untuk dapat menjalankan fungsi *Full-Text Searching* adalah dengan menambahkan index dengan tipe `FULLTEXT` pada field yang ingin dicari. Berikut perintah untuk menambahkan *Full-Text Indexing* pada field `title` dan `body` pada tabel `article` :

```
ALTER TABLE article ADD FULLTEXT(title);
ALTER TABLE article ADD FULLTEXT(body, title);
```

Berikut beberapa tipe penggunaan *Full-Text Searching* dengan menggunakan data contoh pada tabel article :

a. *Natural Language*

Secara default penggunaan *Full-Text Searching* termasuk dalam tipe *Natural Language*.

Berikut beberapa contoh penggunaan *Full-Text Searching* tipe *Natural Language* :

Contoh 1 :

```
SELECT * FROM article WHERE MATCH (title) AGAINST ('tutorial');
```

Query diatas menghasilkan data pada field title yang mengandung kata “tutorial”.

id	title	body
1	PHP & Mysql : Tutorial Dasar	Pada tutorial kali ini, kita akan mencoba melakuka...

Contoh 2 :

```
SELECT * FROM article WHERE MATCH (title) AGAINST ('tutor')
```

Perintah diatas tidak akan menghasilkan data, karena pencarian “tutor” tidak sama dengan pencarian “tutorial”.

Contoh 3 :

Pencarian pada 2 field (title dan body) dapat dilakukan dengan menggunakan perintah :

```
SELECT * FROM article WHERE MATCH (title, body) AGAINST ('tutorial');
```

Perintah diatas menghasilkan data sebagai berikut :

id	title	body
1	PHP & Mysql : Tutorial Dasar	Pada tutorial kali ini, kita akan mencoba melakuka...
3	Mysql	Tutorial penggunaan database masa kini...
2	MySQL - Pengenalan	Tutorial MySQL di situs ini memang buat kamu-kamu ...

3 rows in set

Contoh 4 :

```
SELECT *, MATCH (title, body) AGAINST ('tutorial') as Score FROM article WHERE MATCH (title, body) AGAINST ('tutorial');
```

Perintah diatas menghasilkan data sebagai berikut :

id	title	body	Score
1	PHP & Mysql : Tutorial dasar	Pada tutorial kali ini, kita akan mencoba melakuka...	0.40983861684799
3	Mysql	Tutorial penggunaan database masa kini...	0.26911327242851
2	MySQL - Pengenalan	Tutorial MySQL di situs ini memang buat kamu-kamu ...	0.22223679721355

3 rows in set

Data yang diperoleh diurutkan berdasarkan urutan skor relevansi pencarian.

Contoh 5 :

```
SELECT * FROM article WHERE MATCH (title) AGAINST ('mysql');
```

Perintah diatas tidak menghasilkan data karena MySQL mengimplementasikan “*Stopword*” atau “*Skipword*” untuk setiap hasil pencarian yang tampil pada 50% atau lebih dari total seluruh data.

Stopword adalah mekanisme untuk membatasi pencarian dengan tidak menampilkan hasil pencarian berdasarkan daftar atau list kata-kata yang sering digunakan dalam bahasa inggris. Pengaturan *Stopword* dapat dilihat pada file konfigurasi MySQL (*my.ini*).

b. *Boolean*

Tipe *Boolean* mengijinkan penggunaan operator pada *keyword* pencarian seperti – atau +.

Berikut daftar operator yang dapat digunakan :

- [non operator] secara default memiliki arti “atau” / “or” ;
- + *keyword* harus terdapat pada semua hasil yang didapat;
- - *keyword* tidak boleh ada pada hasil yang didapat;
- > *keyword* termasuk pada hasil yang didapat dengan urutan berdasar ranking secara *increase*;
- < *keyword* termasuk pada hasil yang didapat dengan urutan berdasar ranking secara *decrease*;
- () digunakan untuk melakukan *grouping*/pengelompokan dan dapat menambahkan operator lain untuk meningkatkan relevansi hasil pencarian;
- ~ berfungsi sebagai operator *negasi*, jika sebuah *keyword* diberikan tanda ~, mengakibatkan relevansi/score minus;
- * merupakan operator *wildcard*;
- “ digunakan pada awal dan akhir *keyword* yang terdiri dari dua kata atau lebih.

Contoh 1 :

```
SELECT * FROM articles WHERE MATCH (title,body) AGAINST ('+mysql -database' IN BOOLEAN MODE);
```

Perintah diatas menghasilkan data sebagai berikut :

```
+-----+-----+-----+
| id | title                | body                |
+-----+-----+-----+
| 2  | How To Use MySQL Well | After you went through a ... |
| 3  | Optimizing MySQL     | In this tutorial we will show ... |
| 4  | 1001 MySQL Tricks    | 1. Never run mysqld as root. 2. ... |
| 6  | MySQL Security       | When configured properly, MySQL ... |
+-----+-----+-----+
4 rows in
```

Hasil yang diperoleh harus mengandung kata `mysql` dan tidak boleh mengandung kata `database` pada field `title` dan `body`.

Contoh 2 :

```
SELECT * FROM articles WHERE MATCH (title,body) AGAINST ('tutor*' IN
BOOLEAN MODE);
```

Perintah diatas menghasilkan data sebagai berikut :

```
+-----+-----+-----+
| id | title                | body                |
+-----+-----+-----+
| 1  | MySQL Tutorial       | DBMS stands for DataBase ... |
| 3  | Optimizing MySQL     | In this tutorial we will show ... |
+-----+-----+-----+
2 rows in set
```

Penambahan asterisk (*) pada awal atau akhir keyword berfungsi seperti perintah `LIKE` `'tutor%'`.

Contoh 3 :

```
SELECT * FROM articles WHERE MATCH (title,body) AGAINST ('+mysql
(>tutorial <database)' IN BOOLEAN MODE);
```

Perintah diatas menghasilkan data sebagai berikut :

```
+-----+-----+-----+
| id | title                | body                |
+-----+-----+-----+
| 1  | MySQL Tutorial       | DBMS stands for DataBase... |
| 3  | Optimizing MySQL     | In this tutorial we will show... |
| 5  | MySQL vs. YourSQL    | In the following database comparison... |
+-----+-----+-----+
3 rows in set
```

Penggunaan operator `>` dan `<` menghasilkan data yang mengandung kata `mysql` dan `tutorial` memiliki score relevansi yang lebih tinggi dibandingkan data yang mengandung

kata `mysql` dan `database`, dan seluruh hasil harus mengandung kata `mysql+tutorial` atau `database`.

Contoh 4 :

```
SELECT * FROM articles WHERE MATCH (title,body) AGAINST ('"mysql tutorial"  
IN BOOLEAN MODE);
```

Perintah diatas menghasilkan data sebagai berikut :

```
+-----+-----+-----+  
| id | title                | body                |  
+-----+-----+-----+  
| 1  | MySQL Tutorial      | DBMS stands for DataBase ... |  
+-----+-----+-----+  
1 row in set
```

Hasil pencarian harus mengandung kata “`mysql`” yang kemudian diikuti kata “`tutorial`”.

c. *Query Expansion*

Query Expansion merupakan salah satu fitur yang bias digunakan ketika pencarian hanya mengandung *keyword* yang sedikit (hanya satu kata).

Contoh : ketika user mencari *keyword* `database`, kemungkinan hasil yang diinginkan adalah yang berkaitan dengan `MySQL` atau `Oracle` atau `SQLite`, dengan menggunakan *Query Expansion*, *result* yang dihasilkan diproses dua kali dengan mempelajari *result* pencarian pertama kemudian digabungkan dengan *result* pencarian kedua dengan tambahan *keyword* yang paling relevan terhadap pencarian pertama.

Berikut perbandingan hasil pencarian dengan menggunakan tipe *Natural Language* dengan hasil pencarian dengan menggunakan tipe *Query Expansion* :

1. Pencarian dengan menggunakan tipe *Natural Language* :

```
SELECT * FROM articles WHERE MATCH (title,body) AGAINST ('database');
```

Diperoleh hasil sebagai berikut :

```
+-----+-----+-----+  
| id | title                | body                |  
+-----+-----+-----+  
| 5  | MySQL vs. YourSQL  | In the following database comparison ... |  
| 1  | MySQL Tutorial      | DBMS stands for DataBase ... |  
+-----+-----+-----+  
2 rows in set
```

2. Pencarian dengan menggunakan tipe *Query Expansion* :

```
SELECT * FROM articles WHERE MATCH (title,body) AGAINST ('database' WITH  
QUERY EXPANSION);
```

Diperoleh hasil sebagai berikut :

```
+-----+-----+-----+  
| id | title           | body                                     |  
+-----+-----+-----+  
| 1 | MySQL Tutorial  | DBMS stands for DataBase ...         |  
| 5 | MySQL vs. YourSQL | In the following database comparison ... |  
| 3 | Optimizing MySQL | In this tutorial we will show ...     |  
+-----+-----+-----+  
3 rows in set
```

Tampak bahwa result pertama *keyword* `mysql` selalu muncul, maka *keyword* `mysql` dianggap sebagai *keyword* dengan relevansi tinggi pada pencarian `database` sehingga hasil yang diperoleh juga memunculkan semua data yang mengandung kata `mysql`.

Catatan : penggunaan *Query Expansion* pada *keyword* yang panjang akan mengakibatkan penambahan hasil yang tidak relevan.

Referensi

1. Jim Ferrara, "Using MySQL Full-text Searching", <http://devzone.zend.com>
2. "MySQL Fulltext Search", <http://logcode.net/content/mysql-fulltextsearch>
3. <http://www.mysqltutorial.org>

Biografi Penulis



Didik Setiawan
Pranata Komputer Kementerian Keuangan RI