

# Fulltext-Search

12 December 2018 13:18

Cari sebuah nama product yang mengandung kata "Car" dari dalam database classicmodels

Select productName from products where productName LIKE '%Car';

```
MariaDB [classicmodels]> select productName from products
-> where productName like '%Car';
```

productName
1911 Ford Town Car
1917 Maxwell Touring Car
1950's Chicago Surface Lines Streetcar
1962 City of Detroit Streetcar

Select productName from products where productName LIKE '%Car%';

```
MariaDB [classicmodels]> select productName from products
-> where productName like '%Car%';
```

productName
1911 Ford Town Car
1999 Indy 500 Monte Carlo SS
18th Century Vintage Horse Carriage
1917 Maxwell Touring Car
1950's Chicago Surface Lines Streetcar
1962 City of Detroit Streetcar

6 rows in set (0.13 sec)

Kelemahan cara mencari teks seperti contoh di atas adalah:

1. Performance (kurang bagus jika jumlah baris data semakin banyak), karena cara ini harus membandingkan/mengevaluasi setiap ekspresi like / regular expressions (re) di seluruh baris data dalam tabel yg dicari.
2. Kurang flexible untuk mencari sebuah kata, misal mencari kata 'car' tetapi tida mengandung kata 'classic'
3. Relevance ranking: tidak mungkin memberikan bobot/ranking pada baris data yg ditemukan, susunan data yg diperoleh berdasarkan metode pengurutan data umum.

Oleh karena itu disediakan fasilitas pencarian "fulltext-search" dg beberapa keuntungan sbb:

1. Native SQL-like interface: cara pencarian teks mirip dengan sql dg ekspresi LIKE.
2. Fully dynamic index: MySQL akan mengupdate isi index, setiap kali ada perubahan data pada kolom yg dipilih ketika membuat index saat definisi fulltext-search.
3. Moderate index size: tidak memerlukan ukuran memory besar untuk menyimpan index.
4. Jauh lebih cepat untuk mencari teks menggunakan perintah sql yg kompleks.

Note: fulltext-search hanya tersedia pada MySQL 5.6 dan setelahnya.

Cara mengaktifkan fasilitas fulltext-search:

1. Diaktifkan pada saat membuat tabel:

Contoh:

```
1 CREATE TABLE posts (  
2   id int(4) NOT NULL AUTO_INCREMENT,  
3   title varchar(255) NOT NULL,  
4   post_content text,  
5   PRIMARY KEY (id),  
6   FULLTEXT KEY post_content (post_content)  
7 );
```

**FULLTEXT KEY post\_content** (post\_content)

**post\_content** yg pertama digunakan untuk memberi file index post\_content di dalam adalah nama kolom pada tabel yg akan dicari menggunakan fulltext-search, jumlah kolom dalam kurung boleh lebih dari satu kolom, dipisahkan dengan tanda koma.

**FULLTEXT KEY postidx** (title, post\_content)

2. Modifikasi tabel yg sudah ada, dengan perintah ALTER TABLE ....

Contoh:

```
1 ALTER TABLE products  
2 ADD FULLTEXT(productDescription,productLine)
```

Menambah fasilitas fulltext-search pada table products untuk kolom productDescription, productLine.

3. Membuat index dg perintah "Create fulltext index ...."

Contoh:

```
1 CREATE FULLTEXT INDEX address  
2 ON offices(addressLine1,addressLine2)
```

Contoh mencari kata 'Truck' pada kolom productName yang tidak mengandung kata 'Pickup'. Ada operator BOOLEAN (-) di depan kata 'Pickup'.

```
1 SELECT productName, productline  
2 FROM products  
3 WHERE MATCH(productName) AGAINST( 'Truck -Pickup' IN BOOLEAN MODE )
```

productName	productline
1940s Ford truck	Trucks and Buses

Daftar operator yg bisa digunakan:

Operator	Description
+	Include, the word must be present.
-	Exclude, the word must not be present.
>	Include, and increase ranking value.
<	Include, and decrease the ranking value.
()	Group words into subexpressions (allowing them to be included, excluded, ranked, and so forth as a group).
~	Negate a word's ranking value.
*	Wildcard at the end of the word.
""	Defines a phrase (as opposed to a list of individual words, the entire phrase is matched for inclusion or exclusion).