

PERANGKAT LUNAK OPTIMASI INDEX BASIS DATA MENGGUNAKAN ALGORITMA HEURISTIK

Yoyok Tri Wicaksono¹, Dhinta Darmantoro², Dana Suliyo Kusumo³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Seiring pertumbuhan basis data yang menjadi semakin besar, performansi menjadi hal yang sangat penting. Salah satu faktor yang mempengaruhi performansi suatu basis data relational adalah perancangannya secara fisik. Perancangan basis data relasional secara fisik harus mempertimbangkan index - index yang dipilih dan akan digunakan.

Pada Tugas akhir ini telah dikembangkan suatu perangkat lunak untuk membantu memilih index dan mengkonfigurasikannya pada DBMS MySQL 5. Pemilihan index dilakukan dengan memanfaatkan workload dari log audit dan data definition language dari system catalog yang digunakan. Pemilihan index yang tepat diharapkan akan dapat meningkatkan performansi basis data.

Berdasarkan pengujian, menunjukkan perangkat lunak dapat melakukan pemilihan index untuk optimasi index dengan penurunan biaya query mencapai 1.611.542 % untuk workload transaksi dan 14.364.850 % untuk workload laporan setelah optimasi dengan 1 kolom index dan penggunaan aturan heuristik mengurangi jumlah kandidat index yang perlu dibuat secara real dengan rata - rata sebesar 33,2 % yang akan mengurangi waktu eksekusi.

Kata Kunci : Index, pemilihan Index, workload, system catalog, MySQL, performansi.

Abstract

Along with the growing of database became much bigger, the performance become an important issue. One of the factors influences the relational database performance is the physical design. The physical design of relational database must weigh the indexes that will be select and used. In this final task has developed a software to help selecting index and configure. The index selection is done with use workload from log audit and data definition language from system catalog which used. Selection a correct index is hoped to increase the database performance. Based on testing, showing software can do the index selection for optimization of index with the decreasing cost of query reach 1.611.542 % for the workload of transaction and 14.364.850 % for the workload of report after optimization by 1 column of index and use of heuristic rules lessen the amount of index candidate which require to be made by real with average 33,2 % to lessen the time execute.

Keywords : Index, Index Selection, workload, system catalog, MySQL, performance.

1. Pendahuluan

1.1 Latar belakang

Basis data yang banyak digunakan sampai sekarang adalah basis data model relasional (*Relational Database*), meskipun sudah muncul basis data model objek relasional (*Object Relational Database*) maupun basis data berorientasi objek (*Object Oriented Database*). Kebutuhan masyarakat dan terutama dunia industri akan basis data relasional yang masih sangat besar membuat basis data model relasional masih banyak digunakan.

Seiring pertumbuhan basis data yang menjadi semakin besar, performansi menjadi hal yang sangat penting. Salah satu faktor yang mempengaruhi performansi suatu basis data relasional adalah perancangannya secara fisik. Perancangan basis data relasional secara fisik harus mempertimbangkan *index - index* yang dipilih dan akan digunakan.

Tanggung jawab *Administrator* basis data, salah satunya adalah *tuning* basis data sehingga performansi basis data meningkat. *Tuning* yang dilakukan antara lain mengkonfigurasi *DataBase Management System* serta *hardware*-nya. Dalam hal perancangan dan perawatan, *Administrator* basis data mengkonfigurasi *index* yang digunakan. Sayangnya, vendor pembuat *RDBMS* MySQL tidak menyediakan *tool* untuk mengkonfigurasi *index* supaya *index* yang digunakan optimal.

Untuk itu pada Tugas akhir ini akan dikembangkan suatu perangkat lunak untuk membantu memilih *index* dan mengkonfigurasikannya. Pemilihan *index* dilakukan dengan memanfaatkan *workload* dari *log audit* dan *data definition language* dari *system catalog* yang digunakan. Pemilihan *index* yang tepat diharapkan akan dapat meningkatkan performansi basis data.

Tipe data yang paling banyak digunakan adalah karakter dan numerik karena dibutuhkan masyarakat dan dunia industri, sedangkan *Blob* atau *Text* dan *Spatial* jarang digunakan. Tipe data *Blob* atau *Text* digunakan untuk menyimpan data catatan atau data *binary* berupa *image*, tetapi field ini lebih sebagai *suplemen* dan tidak digunakan dalam seleksi, sehingga tidak dimasukkan dalam TA ini. Sedangkan tipe data *Spatial* digunakan khusus dalam *Geographic Information System (GIS)* dan menggunakan metode *index R-Tree*, sehingga tidak dimasukkan dalam TA ini.

Sebagai studi kasus, akan digunakan kasus “Utilisasi Nasabah Asuransi” sebagai kasus penelitian. DBMS yang akan digunakan sebagai perangkat lunak untuk penelitian adalah MySQL 5.

1.2 Perumusan masalah

Hipotesa sementara dari topik penelitian ini adalah penggunaan algoritma heuristic dapat membantu melakukan pemilihan *index* sehingga optimal yang dapat meningkatkan performansi basis data untuk *query select*.

Permasalahan yang dijadikan objek penelitian pada tugas akhir ini adalah :

1. Bagaimana memilih himpunan *index* berdasarkan *data definition language* dari *system catalog* dan *workload* dari *log audit*.
2. Bagaimana metode heuristic digunakan dalam pemilihan *index* supaya *index – indexnya* optimal
3. Bagaimana mengimplementasikan *index – index* terpilih kedalam basis data.

Batasan masalah pada penelitian ini adalah :

1. DBMS yang digunakan adalah MySQL 5.
2. *Index* yang digunakan adalah *B-Tree* dan *Hash*
3. Perintah SQL dalam *workload* yang dioptimasi adalah *Select*
4. Perintah implementasi *index* pada basis data menggunakan SQL standar 99
5. Tipe data yang digunakan untuk *index* bukan *Blob* dan *spatial*
6. Arsitektur DBMS yang digunakan bukan basis data terdistribusi.

1.3 Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah :

1. Membuat aplikasi pemilihan *index* pada basis data berdasarkan *data definition language* yang diambil dari *system catalog* dan *workload* yang diambil dari *log audit* atau *log query* menggunakan algoritma heuristic supaya *indexnya* optimal.
2. Aplikasi dapat mengimplementasikan himpunan *index* pilihan ke dalam basis data.

1.4 Metodologi penyelesaian masalah

Metodologi penyelesaian masalah yang dilakukan dalam tugas akhir adalah:

1. Studi Literatur
Mempelajari dasar teori dan literature mengenai *index*, *index selection*, algoritma heuristic, serta MySQL 5.
2. Analisa permasalahan dan melakukan penelitian tentang MySQL 5
Melakukan analisa masalah dan kebutuhan yang harus dipenuhi serta melakukan penelitian bagaimana persoalan tersebut dapat dipecahkan, khususnya yang berkaitan dengan MySQL 5.
3. Analisa dan perancangan perangkat lunak
Membuat analisa dan perancangan perangkat lunak dengan teknik orientasi objek dengan bahasa pemodelan *Unified Modeling Language (UML)*.
4. Implementasi perancangan perangkat lunak dan pengujian
Mengimplementasikan perancangan perangkat lunak dalam aplikasi pemilihan *index*.
5. Analisa hasil pengujian
Menganalisa apakah aplikasi yang dibuat sudah dapat mengatasi setiap poin permasalahan yang ada

6. Penutup

6.1 Kesimpulan

Berdasarkan pengujian dan analisa terhadap perangkat lunak, maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan pengujian menunjukkan perangkat lunak dapat melakukan pemilihan *index* untuk optimasi *index* dengan penurunan biaya *query* mencapai 1.611.542 % untuk workload transaksi dan 14.364.850 % untuk workload laporan setelah optimasi dengan 1 kolom *index*.
2. Berdasarkan waktu eksekusi *workload* ternyata sebagian besar waktu digunakan untuk *create index* secara *real* karena MySQL tidak mendukung *virtual index*
3. Penggunaan aturan heuristik mengurangi jumlah kandidat *index* dengan rata – rata sebesar 33,2 % yang perlu dibuat secara *real* yang akan mengurangi waktu eksekusi
4. Implementasi *index* menurunkan performansi *query update* seiring dengan jumlah *index* yang dibuat dalam sebuah tabel dan juga meningkatkan panjang data *index* basis data.

6.2 Saran

Saran – saran untuk pengembangan perangkat lunak berikutnya adalah:

1. Perlu diperhatikan dukungan DBMS terhadap *virtual index* yang akan didukung perangkat lunak.

Daftar Pustaka

- [1] Sri Kusuma Dewi, 2003, "Artificial Intelligence (Teknik dan Aplikasinya)", Graha Ilmu.
- [2] Craig Larman, 1998, "Applying UML and Pattern : an Introduction to Object Oriented Analysis and Design", Prentice Hall.
- [3] Raghu Ramakrishnan, Johannes Gehrke, 2003, "Database Management System 3ed", McGraw Hill.
- [4] Sudharsan, Henry F. Korth, Abraham Silberschatz, 2002, "Database System Concepts 4ed", McGraw Hill.
- [5] Peter Gultzan, Trudy Pelzer, 2002, "SQL Performance Tuning", Addison Wesley.
- [6] Jozef Kratica, Ivana Ljubic, Dusan Tomic, A Genetic Algorithm for the Index Selection Problem, 2006, <http://citeseer.ist.psu.edu/568873.html>, didownload pada tanggal 21 Agustus 2006.
- [7] Surajit Cauduri, Vivek Narasayy, an efficient cost-driven index selection tools for microsoft sql server, 2006, <http://citeseer.ist.psu.edu/chaudhuri97efficient.html>, didownload pada tanggal 21 Agustus 2006.
- [8] ___, MySQL Reference Manual, 2004, <http://www.mysql.org>, didownload pada tanggal 8 September 2004.
- [9] Sunil Choenni, Henk Blanken, Thiel Chang, index selection in relational database, 2006, didownload pada tanggal 21 Agustus 2006.
- [10] Paul Dubois, MySQL Query Optimization, 2006, <http://www.informit.com/articles/article.asp>, didownload pada tanggal 6 September 2006.
- [11] Paul Dubois, Enhancing MySQL Query Efficiency, 2006, <http://www.devshed.com/c/a/MySQL/Enhancing-MySQL-Query-Efficiency/article.asp>, didownload pada tanggal 6 September 2006.
- [12] Martin R. Frank, Edward R. Omiecinski, Shamkant B., Adaptive and Automated Index Selection in RDBMS, 2006, <http://citeseer.ist.psu.edu/frank92adaptive.html>, didownload pada tanggal 21 Agustus 2006.
- [13] Alberto Caprara, Jose Juan Salazar, A Branch-and-Cut Algorithm for the Index Selection Problem, 2006, <http://citeseer.ist.psu.edu/56600.html>, didownload pada tanggal 21 Agustus 2006.
- [14] Harry Chrismanaria, 2001, "TA : Simulasi Query Optimizer", STT TELKOM.
- [15] Grady Booch, James Rumbaugh, Ivar Jacobson, 1999, "The Unified Modeling Language User Guide", Addison Wesley.