

Review: Implementasi Holap Untuk Optimasi Query Sistem Basis Data Terdistribusi Dengan Pendekatan Algoritma Genetik

Rahmad Syaifudin^{a,*}, Selo^b, Rudy Hartanto^c

^aMahasiswa S2 Teknik Elektro Universitas Gadjah Mada

^bDosen Jurusan Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada

^cDosen Jurusan Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada

Naskah Diterima : 14 Oktober 2014; Diterima Publikasi : 12 Nopember 2014

Abstract

Distributed Database is one of database that is under control of the Database Management System (DBMS) was focused on storage devices are separated from one and another. Optimization data query on distributed database system not be separated from data processing methods that used. Then for fast query optimization this database need some required methods that can optimize it. Hybrid online analytical processing (HOLAP) or often to call Hybrid-OLAP is one of technology for optimization query on distributed database. Genetic Algorithm is one of algorithm for heuristic searching was based on the mechanisms of biological evolution. Process of genetic algorithm is combining a selection process, using a crossover operator and mutation to get the best solution. From the reviews about implementation HOLAP with Genetic Algorithm approach was expected being used as a basis research on HOLAP implementation for query optimization on distributed database with genetic algorithm approach.

Keywords : Query Optimization; Distributed database; HOLAP; OLAP; Genetetic algorithm.

Abstrak

Basis data terdistribusi (*distributed database*) merupakan suatu basis data yang berada di bawah kendali *Database Management System* (DBMS) yang terpusat pada peranti penyimpanan (*storage devices*) yang terpisah-pisah satu dari yang lainnya. Optimasi *query* data pada sistem terdistribusi tentunya tidak lepas dari metode pengolahan data yang digunakan. Agar operasi *query* tidak lambat diperlukan metode yang dapat mengoptimalkan operasi *query* tersebut. Salah satu teknologi yang digunakan untuk optimalisasi *query* pada sistem basis data terdistribusi salah satunya adalah *Hybrid online analytical processing* (HOLAP) atau sering di sebut *Hybrid-OLAP*. Algoritma genetik merupakan algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Proses algoritma genetik menggabungkan proses seleksi, penggunaan operator crossover (penyilangan) dan mutasi untuk mendapatkan solusi terbaik. Dari review tentang implementasi HOLAP dengan pendekatan algoritma genetik diharapkan mampu dapat digunakan sebagai dasar penelitian tentang implementasi HOLAP untuk optimasi *query* sistem basis data terdistribusi dengan pedekatan algoritma genetik yang akan dilakukan.

Keywords: Optimasi Query; Sistem Terdistribusi; HOLAP; OLAP; Algoritma Genetik

1. Pendahuluan

Basis data terdistribusi (*distributed database*) merupakan suatu basis data yang berada di bawah kendali *Database Management System* (DBMS) yang terpusat pada peranti penyimpanan (*storage devices*) yang terpisah-pisah satu dari yang lainnya (Pankti and Vijay, 2011). Tempat penyimpanan ini dapat berada di satu lokasi yang secara fisik berdekatan misalnya dalam satu bangunan atau terpisah oleh jarak yang jauh dan terhubung melalui jaringan

internet. Penggunaan basis data terdistribusi dapat dilakukan pada server internet, intranet maupun ekstranet. Optimasi transformasi data pada sistem terdistribusi tentunya tidak lepas dari metode pengolahan data yang digunakan.

Secara teori database adalah suatu sistem yang memproses input berupa data menjadi output yaitu informasi yang diinginkan. Agar operasi *query* tidak lambat diperlukan metode yang dapat mengoptimalkan operasi *query* tersebut. Suatu metode optimasi *query* mencoba untuk menentukan

*) penulis korespondensi: rahmadsyaifudin@gmail.com

cara yang paling optimal untuk mengeksekusi query yang diberikan dengan mempertimbangkan rencana query yang mungkin (Zhang *et al.*, 2006)(Meng *et al.*, 2008).

Salah satu teknologi yang digunakan untuk optimalisasi *query* pada sistem basis data terdistribusi salah satunya adalah Hibryd online analytical processing (HOLAP) atau sering di sebut *Hybrid-OLAP*. HOLAP yang merupakan salah satu model terbaru dari online analytical processing (OLAP). Cara kerja HOLAP adalah dengan menggabungkan model *Multidimensional Online Analytical Processing* (MOLAP) dan *Relational Online Analytical Processing* (ROLAP), yang mana HOLAP merupakan jalan tengah yang menutupi kekurangan antara keduanya. MOLAP yang mampu menangani jumlah volume data yang sangat besar dan dapat memanfaatkan fungsi-fungsi yang ada pada relational database yang dipakai, dikombinasi dengan ROLAP yang mampu melakukan pengambilan data secara cepat dan optimal serta membentuk kalkulasi yang kompleks dan cepat (Theodoros, n.d.). Algoritma genetik merupakan algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Proses algoritma genetik menggabungkan proses seleksi, penggunaan operator crossover (penyilangan) dan mutasi untuk mendapatkan solusi terbaik. Metode *crossover* yang dapat digunakan untuk menyelesaikan masalah optimasi query database, diantaranya adalah *M2S crossover* dan *CHUNK crossover* (Manahan *et al.*, 2008). Dari gabungan metode HOLAP dan algoritma genetik diharapkan dapat mengoptimalkan *query* pada sistem terdistribusi.

Sebagai contohnya kampus merupakan salah satu instansi yang memiliki jumlah data yang sangat besar. Penggunaan sistem basis data dapat memiliki peranan penting dalam menunjang kegiatan suatu instansi. Sebagai instansi yang besar seharusnya memiliki sistem basis data yang menyimpan data-data penting yang berhubungan dengan mahasiswa dan instansi- instansi terkait. Pada penelitian yang pernah dilakukan oleh (Chandramitasari *et al.*, 2014) bahwa saat ini, Sistem Informasi Akademik Mahasiswa Universitas Brawijaya (SIKAD UB) Malang masih menerapkan sistem basis data yang mengakses tabel dasar dan masih memanfaatkan *online transaction processing* (OLTP), sehingga proses analisis masih memanfaatkan penggunaan join untuk mengambil data dari berbagai tabel. Waktu eksekusi yang dibutuhkan adalah 1,975668 detik dengan jumlah sebanyak 27768 baris. Untuk mengurangi waktu eksekusi, maka penulis mencoba menawarkan model optimasi query menggunakan HOLAP dengan pendekatan algoritma genetik.

Berdasarkan uraian diatas, maka optimasi *query* dibutuhkan untuk efektifitas transformasi data. Maka akan dilakukan penelitian yang berkenaan dengan tema tersebut, yakni implementasi HOLAP untuk

optimasi query basis data terdistribusi dengan pendekatan algoritma genetik.

2. Pembahasan

OLAP merupakan metode pendekatan untuk menyajikan jawaban dari permintaan proses analisis yang bersifat dimensional secara cepat, yaitu desain dari aplikasi dan teknologi yang dapat mengoleksi, menyimpan, memanipulasi suatu data multidimensi untuk tujuan analisis (Loukopoulos and Ahmad, 2006). Menurut (Berson and Smith, 1997) ada tiga kategori utama peralatan OLAP antara lain *Multidimensional Online Analytical Processing* (MOLAP) dan *Relational Online Analytical Processing* (ROLAP) dan *Hybrid On-Line Analytical Processing* (HOLAP).

Aturan- aturan OLAP didefinisikan oleh (Codd, 1995) menjadi 12, antara lain :

- a. *Multidimensional model*
Mendukung EIS (*Executive Information System*) operasi slice dan dice yang biasanya diperlukan dalam pemodelan keuangan.
- b. *Transparency of the server*
Merupakan bagian dari *open system* yang mendukung heterogeneous data sources, sehingga end user tidak harus mempedulikan akses detail data atau konversi.
- c. *Accessibility*
Penyajian pada user dengan *metode single logical schema* dari data, mesin OLAP bertindak sebagai middleware yang berada di antara *heterogeneous data sources* dan *OLAP front-end*.
- d. *Stable access performance*
Performance sistem seharusnya tidak menurun karena jumlah dimensi dalam model meningkat.
- e. *Client/server architecture*
Memerlukan sistem modular terbuka, karena tidak hanya produk harus bersifat client / server tetapi komponen server OLAP harus memungkinkan bahwa berbagai clien dapat terpasang dengan minimum effort dan programming for integration
- f. *Generic Dimensionality*
Tidak terbatas pada dimensi tertentu, sebuah fungsi diterapkan pada sebuah dimensi tetapi juga harus dapat diterapkan ke dimensi yang lain.
- g. *Management of data sparsity*
Menghubungkan *ide nulls* dalam relational database dan konsep kompresi *file* besar, sistem OLAP harus mengakomodir berbagai penyimpanan dan pilihan data-handling.
- h. *Multi-user*
Mendukung berbagai user secara bersamaan, termasuk pandangan individu atau slice dari database umum.
- i. *Operation on dimension*

Semua dimensi dibuat sama, sehingga semua bentuk perhitungan harus diizinkan di semua dimensi.

j. *Intuitive manipulation of data*

User tidak harus menggunakan menu atau melakukan operasi berbagai langkah yang kompleks ketika tindakan intuitif drag and drop akan dilakukan.

k. *Flexible posting and editing*

User dapat hanya mencetak apa yang di butuhkan dan setiap perubahan pada model secara otomatis tercermin dalam laporan.

l. *Multiple dimensions and levels*

Mendukung setidaknya 15 atau 20 dimensi.

ROLAP merupakan tipe OLAP yang bergantung kepada *Relational Database Management System* (RDBMS) sebagai media penyimpanan (*storage*) data yang akan diolah. Dengan strategi tersebut maka OLAP Server terhindar dari masalah pengelolaan *data storage* dan hanya menerjemahkan proses *query analysis* atau *Multidimensional Expressions* (MDX) ke *relational query* atau *Structure query language* (SQL). Otomatis proses optimasi ROLAP akan sangat ditentukan di sisi produk RDBMS yang digunakan misalkan dari sisi penanganan jumlah data dan strategi *indexing* (Codd, 1995).

Menurut Fauzi (2014) cara kerja ROLAP secara umum adalah sebagai berikut:

- OLAP *client* mengirimkan query analisis ke OLAP *Server*.
- OLAP server akan melakukan pemeriksaan di *cache* apakah sudah bisa melayani permintaan query dari client tersebut, jika sudah akan dikirimkan.
- Jika pada *cache* belum terdapat data diminta, akan dilakukan *query SQL* ke *data mart* dan hasil eksekusinya disimpan di *cache* dan dikirimkan kepada client, demikian seterusnya.
- Cache* akan disimpan selama periode waktu tertentu dan akan dibersihkan total jika *server* ke dalam dua tahapan yaitu:
 - Tahap konstruksi dan populasi data, dalam tahapan ini aplikasi akan membaca dan melakukan perhitungan *agregasi* atau *summary* dari sumber data. Perhitungan dilakukan untuk pada berbagai level dimensi, dan hasilnya akan disimpan pada *database*.
 - Tahap *query* atau layanan permintaan data analisis, dalam tahapan ini OLAP *server* akan melayani permintaan *query* dari *client* dan membaca dari database MOLAP. Table yang akan dibaca adalah suatu fragmen yang akan disesuaikan dengan permintaan dari client.

Karakteristik HOLAP yang merupakan kombinasi dari ROLAP dan MOLAP biasanya menyimpan data dalam *Relational Database* (RDB) dan *Multidimensional Database* (MDDB)

selanjutnya menggunakan mana yang paling cocok dengan jenis pengolahan yang diinginkan. *Database* merupakan *tools* yang paling sering digunakan untuk menyimpan data dalam cara yang paling fungsional. Untuk pengolahan data yang besar lebih efisien disimpan dalam RDB, sedangkan untuk pengolahan spekulatif, data lebih efektif disimpan dalam MDDB (Doherty, n.d.). Menurut (Fauzi, 2014) perbandingan ROLAP, MOLAP dan HOLAP diilustrasikan pada Tabel 1.

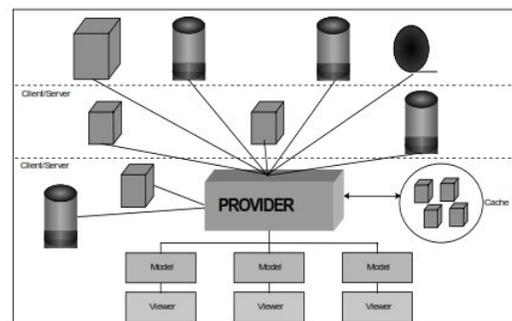
Tabel 1. Perbandingan ROLAP, MOLAP dan HOLAP (Fauzi, 2014)

Type	Penyimpanan RDBMS	Penyimpanan Internal	Performa Pembacaan	Pre-Komputasi
ROLAP	Ya	Tidak	Tergantung RDBMS	Tidak
MOLAP	Tidak	Ya	Sangat Baik	Ya
HOLAP	Ya	Ya	Sangat Baik	Ya

Pada makalah yang diterbitkan oleh *SUN Institute* (Weinberger and Ender, n.d.) disebutkan spesifikasi *HOLAP* antara lain :

a. *Multiple Storage Formats*

MDDB Server memungkinkan untuk menangani data dalam format dan lokasi yang berbeda sebagai salah satu *logical unit*. Secara rinci, dapat mengakses kumpulan data disimpan dalam MDDBS, perangkat *output*, *Relational data* dan *Star Schemas*. Bahkan *HOLAP* juga memungkinkan OLAP *reporting* setiap data source yang muncul pada sistem.



Gambar 1. Model HOLAP (Weinberger and Ender, n.d.)

b. *Stacking*

Merupakan kemampuan penyimpanan *individual aggregation levels* pada *file* yang terpisah dan bisa juga dalam server yang terpisah jika diinginkan. *Stacking* dapat digunakan untuk meningkatkan kinerja dengan cara menyimpan agregasi yang paling banyak dicari pada MDDB jaringan lokal dan agregasi lainnya yang disimpan dalam perangkat data atau tabel relasional di server yang terpisah. Misalnya data penduduk suatu daerah disimpan dalam MDDB lokal dan file utama disimpan dalam tabel relasional di server pusat.

c. *Remote Computer Service*

Layanan *remote* komputer memungkinkan *rollups* dinamis, perhitungan kolom *derived* dan aplikasi melakukan *sorting* dan *cutting criteria*. *Rollup dinamis* adalah kemampuan untuk melakukan kalkulasi pada saat *query time* level agregasi yang tidak disimpan yaitu dengan meringkas data dengan menaikkan konsep hierarki (Usman *et al.*, 2010).

Sedangkan fitur- fitur yang ditawarkan oleh HOLAP adalah :

a. *Caching*

Merupakan kemampuan untuk menyimpan hasil query sehingga dapat digunakan kembali di lain waktu. *Cache* dapat dikontrol dengan menetapkan jumlah absolut dan ukuran maksimum permintaan untuk terus dalam *cache* (Usman *et al.*, 2010) dan (Goil and Choudhary, 1999).

b. *Logging*

Logging menciptakan sebuah *file* yang memuat informasi tentang setiap permintaan yang dibuat. *File* kemudian dapat dianalisis untuk membantu dengan mengoptimalkan struktur kelompok data HOLAP. Contohnya, mungkin jika terjadi request yang memerlukan dinamis *roll-up* sementara permintaan sedikit yang mengakses maka agregasi disimpan (Usman *et al.*, 2010) (Goil and Choudhary, 1999).

c. *Extensibility*

Komponen HOLAP yang muncul sebagai *software* berorientasi objek menyediakan cara mudah bagi *user* untuk memperluas dan memodifikasi perilaku data *provider* secara terprogram (Usman *et al.*, 2010) (Goil and Choudhary, 1999).

OLAP diintegrasikan dengan proses data mining dalam database dengan menerapkan aturan asosiasi yang mengakibatkan fleksibilitas aturan dalam proses penambangan multidimensi dan multi-level association (Usman *et al.*, 2010) dan (Fong *et al.*, 2007). Penggabungan *OLAP* dan *Data Mining* dalam sebuah database dengan cara yang berbeda dapat digunakan untuk menemukan pola dari sebuah data. Dalam *data mining* sendiri memiliki dua teknik yang sering digunakan yaitu *clustering* dan *decision tree*. *Clustering* digunakan untuk mengelompokkan data dalam database sesuai dengan jumlah data yang ada. Sedangkan *decision tree* dan *OLAP* digunakan untuk memeriksa pengelompokkan yang dihasilkan dan mencari korelasi antara pola-pola, populasi dan gambaran dari interfertilitasnya (Dzeroski *et al.*, 2000) dan (Usman *et al.*, 2010).

Data warehouse telah menjadi salah satu point paling penting dalam mendukung keputusan suatu organisasi. Selain itu, bukti empiris menunjukkan bahwa pengguna dapat meningkatkan kinerja dengan menerapkan data warehouse. Dalam proses untuk mendapatkan pengetahuan dari data yang disimpan dalam sebuah gudang data, pengambilan keputusan

biasanya menggunakan *OLAP*, *query* dan pelaporan maupun proses *data mining* (Zhijuan *et al.*, 2012). Oleh karena itulah penggunaan *OLAP* yang diintegrasikan dengan data *warehouse* dapat sangat membantu proses pengambilan keputusan.

Kualitas metadata dalam *OLAP* memiliki pengaruh yang luar biasa pada stabilitas dan kehandalan *OLAP*. Pendekatan integrasi metadata dalam Model-driven memperkenalkan konsep manajemen metadata berdasarkan paradigma berorientasi objek untuk pemodelan dan query metadata *OLAP* untuk multidimensional data (Zhao and Huang, 2010).

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan pada mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom dalam individu organisme. Variasi *kromosom* ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup (Kristanto, 2004) (Jiunn-Chin Wang *et al.*, 1996). Pada dasarnya ada empat kondisi yang sangat mempengaruhi proses evaluasi, yaitu :

1. Kemampuan organisme untuk melakukan reproduksi.
2. Keberadaan populasi organisme yang bisa melakukan reproduksi.
3. Keberagaman organisme dalam suatu populasi.
4. Perbedaan kekuatan dan kemampuan organisme untuk bertahan hidup.

Individu yang lebih kuat (*fit*) akan memiliki tingkat *survival* atau tingkat daya bertahan hidup yang lebih tinggi. Selain itu individu yang semakin kuat akan memiliki tingkat reproduksi yang lebih tinggi jika dibandingkan dengan individu yang kurang *fit*. Pada kurun waktu tertentu (sering dikenal dengan istilah generasi), populasi secara keseluruhan akan memuat lebih banyak organisme yang *fit*. Pada algoritma ini teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin, dikenal dengan istilah populasi. Di dalam populasi tersebut terdapat individu yang disebut dengan istilah kromosom. Kromosom- kromosom tersebut merupakan suatu solusi yang masih berbentuk simbol, biasanya adalah bilangan biner. Kromosom-kromosom ini akan mengalami evolusi melalui sejumlah iterasi yang disebut dengan generasi. Dalam setiap generasi kromosom akan mengalami proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Dalam algoritma genetik, istilah kromosom merujuk pada kandidat solusi dari suatu masalah, sering dilambangkan sebagai sebuah string yang terdiri dari bit. Gen adalah sebuah bit tunggal atau sebuah blok yang terdiri dari bit-bit yang berdampingan yang melambangkan elemen tertentu dari kandidat solusi. *Crossover* adalah pertukaran material genetik antara kromosom dari parent. Mutasi adalah menukar sebuah *allele* pada *locus* (posisi) random dengan dengan *allele* lainnya, misal 0 menjadi 1. Setiap generasi akan menghasilkan

kromosom-kromosom baru yang dibentuk dari generasi sebelumnya dengan menggunakan operator reproduksi (*reproduction*), kawin silang (*crossover*), dan juga mutasi (*mutation*). Nilai *fitness* dalam suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) yang terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Generasi-generasi baru dibentuk dengan cara:

1. Melakukan proses seleksi sesuai dengan nilai obyektif dari kromosom *parent* dan juga kromosom *offspring*.
2. Membuang beberapa kromosom sehingga jumlah populasi akan kembali menjadi kromosom.

Demikian generasi yang baru terus dihasilkan sesuai dengan besar generasi yang ditentukan dan setelah melalui beberapa generasi maka algoritma ini akan konvergen ke kromosom terbaik (Kristanto, 2004)

Dalam penyelesaian masalah dengan menggunakan algoritma genetik, akan dilakukan dengan cara mencari hasil optimum dari algoritma genetik yang digunakan pada *Left-Deep Strategy*. Dalam menyelesaikan permasalahan dengan menggunakan algoritma genetik, ada beberapa hal dasar yang harus diperhatikan, yaitu representasi kromosom, inialisasi populasi, fungsi evaluasi, seleksi, jenis operator genetik yang digunakan (*crossover* dan mutasi), dan penentuan parameter. Proses algoritma genetik yang dilakukan adalah Representasi Kromosom dimana dalam algoritma genetik untuk masalah optimasi *query database* ini adalah suatu gen yang terurut, dimana pada masing-masing gen terdiri dari relasi, join dan metode join (Meng *et al.*, 2008). Penggunaan dua metode *crossover* tersebut diuji dan dianalisa hasilnya, untuk mengetahui metode *crossover* apa yang terbaik yang dapat digunakan untuk menyelesaikan masalah optimasi *query database*, dengan mencari nilai minimum.

Pemanfaatan algoritma genetik untuk proses clustering basis data yang dikombinasikan dengan metode *TSP (travelling salesman problem)* dilakukan oleh (Cheng *et al.*, 2002) dimana proses yang dilakukan adalah dengan menerapkan pendekatan *genetic search-based clustering* ke dalam partisi basis data. Diusulkan sebuah pendekatan baru, yang merumuskan masalah partisi sebagai *TSP* untuk mengubah urutan atribut / transaksi. Sistem partisi di bagi menjadi dua, yaitu *horizontal partitioning* dan *vertical partitioning*. Dilakukan pemotongan pada *tour* atribut yang memiliki *cost* tertinggi untuk mendapatkan fragment. Sebuah metodologi berbasis algoritma genetik digunakan untuk mendapatkan solusi. Proses *vertical partitioning* dan *horizontal partitioning* awalnya menggunakan dua model dan perhitungan yang berbeda. Dapat dibuktikan bahwa

algoritma genetik dapat dimanfaatkan untuk melakukan seleksi dalam proses partisi. Hasil dari penelitian yang dilakukan adalah dengan menggunakan algoritma genetik model pada *vertical partitioning* dapat diimplementasikan pada *horizontal partitioning* tanpa harus membuat dua model partisi yang berbeda..

Optimasi *query* yang melibatkan kumpulan operasi untuk komunikasi data dengan *cost* transmisi rendah diperlukan untuk efektifitas transformasi data terdistribusi.pada penelitian yang dilakukan oleh (Jiunn-Chin Wang *et al.*, 1996) yang menerapkan algoritma genetik meminimalisir *cost* pemrosesan transmisi *query* terdistribusi. Pembahasan yang dilakukan lebih spesifik terhadap model penggabungan kromosom. Dalam proses penelitiannya dibuktikan bahwa pendekatan genetik memberikan kontribusi praktis terhadap *RDDBS* serta dapat diterapkan pada banyak aplikasi. Penerapan algoritma genetik pada penelitian ini adalah merepresentasikan kromosom kedalam bentuk *binary vector*. Selanjutnya ditentukan populasi awal dan strategi seleksi sesuai dengan *rule GA*. Operator genetik yang digunakan sebagai penekatan adalah *one-point crossover* dan mutasi. Terdapat tiga strategi yang digunakan yaitu strategi menentukan populasi awal menggunakan probabilitas yang disebut AG-BIAS, teori membagi generasi kedalam 2 tingkatan yang disebut GA-ESBI dan GA-1 yang hanya menggunakan model *cross-over*. Ekperimen yang dilakukan adalah melakukan perbandingan dari metode yang digunakan dengan AH (*Authentication Header*), hasil perbandingan kualitas solusi seperti pada Tabel 2 dan waktu komputasi pada Tabel 3.

Tabel 2 . Perbandingan kualitas solusi

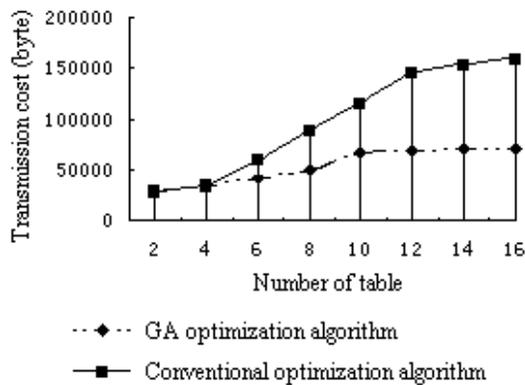
NODE	AH	GA-1	GA-ESBI	GA-BIAS
40	10241	10159	10159	10159
100	29647	29585	29162	29387
200	66230	67002	66869	67333
300	96634	98480	99099	99478
400	136165	141704	141741	143004
500	166097	171875	176136	176984
600	206046	212643	221256	222321
700	233874	240194	258304	259023

Tabel 3. Waktu komputasi (detik)

NODE	AH	GA-1	GA-ESBI	GA-BAIS
40	0.93	0.85	2.02	1.53
100	52.45	40.88	9.39	6.53
200	688.22	470.47	36.33	21.56
300	3592.32	1192.12	173.38	63.89
400	11091.64	2653.91	351.55	177.46
500	31008.10	6365.41	1073.53	368.95
600	66108.53	15037.23	1956.52	693.80
700	102344.76	31384.45	2710.69	1645.27

Sebuah metodologi berbasis *algoritma genetik* digunakan untuk mendapatkan solusi. Penelitian tentang optimasi *query* basis data terdistribusi yang menerapkan pengkodean dengan *tree-structure* berdasarkan posisi dan nilai seperti yang dilakukan oleh (Li and Luo, 2008). Selain itu operator genetik, yaitu *reproduksi*, *crossover* dan *mutasi* digunakan untuk proses pengkodean. Perbaikan *Crossover* perlu dilaksanakan dalam dua langkah dan perbaikan, proses mutasi terdiri dari mutasi nilai dan mutasi posisi. Dalam penelitian yang dilakukan dimasukkan 70 nodes dan 20 tabel ke dalam database terdistribusi yang digunakan sebagai model. Algoritma genetik dengan *tree-structure* yang didasarkan pada posisi dan nilai digunakan untuk optimasi *query*.

Dilakukan juga perbandingan dengan metode tanpa menggunakan *dicission tree*. Hasil dari eksperimen yang dilakukan bahwa *transmission cost* data turun dari 150.000 menjadi 50.000 seperti ditunjukkan dalam Gambar 2 berikut ini :



Gambar 2. Perbandingan hasil penggunaan metode konvensional dengan AG(Li and Luo, 2008).

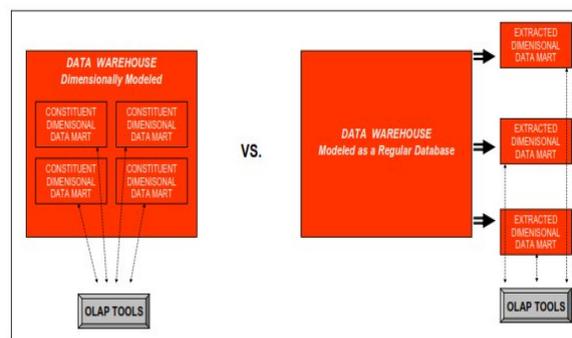
Penelitian tentang optimasi *query* basis data system terdistribusi pernah dilakukan oleh (Manahan *et al.*, 2008), pada penelitian tersebut digunakan algoritma genetik. Proses yang digunakan adalah dengan menggabungkan proses seleksi, penggunaan operator *crossover* (penyilangan) dan mutasi untuk mendapatkan solusi terbaik. Dalam penyelesaian masalah optimasi *query* database ada dua metode *crossover* yang dapat digunakan, yaitu M2S *crossover* dan CHUNK *crossover*.

Proses yang digunakan adalah dengan menggabungkan proses seleksi, penggunaan operator *crossover* (penyilangan) dan mutasi untuk mendapatkan solusi terbaik. Ukuran populasi = 1 – 100 dan maksimum generasi = 1 - 1000. Sedangkan untuk probabilitas *crossover* (P_c) dan probabilitas mutasi (P_m) nilainya disesuaikan pada saat pengujian. Dua metode *crossover* yang dapat digunakan untuk menyelesaikan masalah optimasi *query database*, yaitu M2S *crossover* dan CHUNK *crossover*. Penggunaan dua metode *crossover*

tersebut diuji dan dianalisa hasilnya, untuk mengetahui metode *crossover* apa yang terbaik yang dapat digunakan untuk menyelesaikan masalah optimasi *query database*, dengan mencari nilai minimum (Sontya *et al.*, 2007). Hasil dari pengujian yang dilakukan adalah dengan metode M2S *crossover* dengan *probabilitas crossover* (P_c) = 100 maka eksekusi yang mampu dilakukan adalah yang dilakukan adalah 12.328 dalam waktu 5,62 detik, sedangkan dengan CHUNK *crossover* adalah 12.328 dalam waktu 5,633 detik. Semakin besar nilai (P_c , P_{psize} dan P_{maxgen}) maka hasilnya akan semakin akurat. Sebaliknya nilai P_m tidak perlu terlalu besar karena akan membuat hasil akhir kurang akurat. Namun demikian, hasil pengujian selanjutnya dapat saja berbeda karena komponen algoritma genetik berbasis pada fungsi random.

Optimasi *query* pernah dilakukan oleh (Chandramitasari *et al.*, 2014) adalah menerapkan OLAP dengan pendekatan *Materialized Query Table (MQT)* yang diimplementasikan pada SIAKAD UB. Perancangan data warehouse menggunakan *Kimball Nine-Step Methodology* yang dikemukakan oleh *Ralph Kimball* dan untuk perancangan *data flow* dan *control flow* menggunakan *tools IBM Design Studio* dan untuk menampilkan hasil OLAP menggunakan *IBM Cognos Insight*. Hasil dari eksperimen yang dilakukan adalah dari 117.642 baris yang awalnya membutuhkan waktu eksekusi 5.398107 detik turun menjadi 2.915688 detik.

Penggunaan OLAP dapat memungkinkan pengguna untuk melakukan *query* terhadap table fakta dan dimensi dengan menggunakan titik sederhana dan aplikasi permintaan pembangunan data. Ada beberapa pendekatan arsitektur untuk mengembangkan sebuah data warehouse, akan tetapi semua aritektur tersebut memberikan hasil akhir sebuah dimensi data mart yang dimodelkan sebagai antarmuka konseptual (Jukic *et al.*, 2013).



Gambar 3. Interface OLAP untuk arsitektur data warehouse yang berbeda (Chandramitasari *et al.*, 2014).

3. Kesimpulan

Berdasarkan pembahasan review diatas dapat ditarik kesimpulan bahwa metode OLAP dengan pendekatan algoritma genetik dapat digunakan untuk optimasi *query* sistem basis data terdistribusi. Sehingga diharapkan dapat membantu penelitian yang akan dilakukan.

Ucapan Terima Kasih

Ucapan terima kasih kami sampaikan kepada Tuhan YME yang selalu memberikan rahmat dan hidayah-Nya. Kepada ayah dan ibunda tercinta, yang dengan tulus dan penuh kasih senantiasa selalu memberikan do'a, materi dan motivasinya. Kepada bapak Selo, ST, MT, M.Sc, Ph.D dan bapak Ir. Rudy Hartanto, MT yang selalu mengarahkan dan membimbing kami sehingga makalah ini dapat tersusun. Seluruh sahabat dan teman-teman MTI UGM 2013 serta semua pihak yang telah memberikan banyak masukan, bantuan dan motiasi.

Daftar Pustaka

- Berson, A., Smith, S.J., 1997. Data Warehousing, Data Mining, and OLAP, Mcgraw-Hill Series on Data Warehousing and Data Management. Computing Mcgraw-Hill.
- Chandramitasari, W., Wicaksono, S.A., Muristy, Y.T., 2014. Optimasi online analytical processing (OLAP) Pada data warehouse dengan pendekatan Materialized query table (Studi Kasus: Basis Data Siakad UB). PTIIK Univ. Brawijaya 1–6.
- Cheng, C.-H., Lee, W.-K., Wong, K.-F., 2002. A genetic algorithm-based clustering approach for database partitioning. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* 32, 215–230.
- Codd, E.A., 1995. Twelve Rules for Online Analytical Processing. *Computer World*.
- Doherty, R., n.d. Hybrid OLAP an Introduction. SAS Inst., 1998.
- Dzeroski, S., Hristovski, D., Peterlin, B., 2000. Using data mining and OLAP to discover patterns in a database of patients with Y-chromosome deletions. *Proc. AMIA Annu. Symp. AMIA Symp.* 215–219.
- Fauzi, M.C., 2014. Pentaho Data Analysis (Schema Workbench) With Mondrian.
- Fong, J., Huang, S.-M., Hsueh, H.-Y., 2007. Online analytical mining association rules using Chi-square test. *Int. J. Bus. Intell. Data Min.* 2, 311–327.
- Goil, S., Choudhary, A., 1999. A parallel scalable infrastructure for OLAP and data mining. *IEEE Comput. Soc.* pp. 178–186. doi:10.1109/IDEAS.1999.787266
- Jiunn-Chin Wang, Jorng-Tzong Horng, Yi-Ming Hsu, Baw-Jhiune Liu, 1996. A genetic algorithm for set query optimization in distributed database systems. *IEEE*, pp. 1977–1982. doi:10.1109/ICSMC.1996.565428
- Jukic, N., Jukic, B., Malliaris, M., 2013. Online Analytical Processing (OLAP) for Decision Support. *Dep. Comput. Sci. CITY Lib. Stud. Affil. Inst. Univ. Sheff.*
- Kristanto, A., 2004. Jaringan Syaraf Tiruan (Konsep Dasar, Algoritma, dan Aplikasinya). Gava Media, Yogyakarta.
- Li, H., Luo, B., 2008. A tree-based genetic algorithm for distributed database. Presented at the Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2008, pp. 2614–2618.
- Loukopoulos, T., Ahmad, I., 2006. Policies for caching OLAP queries in internet proxies. *IEEE Trans. Parallel Distrib. Syst.* 17, 1124–1135.
- Manahan, S., Kania Sabariah, M., Sontya, M., 2008. Optimasi Query Database Menggunakan Algoritma Genetik. *Semin. Nas. Apl. Teknol. Inf. 2008 SNATI 2008*.
- Meng, M., Cui, X., Cui, H., 2008a. The approach for optimization in watermark signal of relational databases by using genetic algorithms. Presented at the Proceedings of the International Conference on Computer Science and Information Technology, ICCSIT 2008, pp. 448–452.
- Meng, M., Cui, X., Cui, H., 2008b. The approach for optimization in watermark signal of relational databases by using genetic algorithms. Presented at the Proceedings of the International Conference on Computer Science and Information Technology, ICCSIT 2008, pp. 448–452.
- Pankti, D., Vijay, R., 2011. Review of dynamic query optimization strategies in distributed database. *Electron. Comput. Technol. ICECT 2011 3rd Int. Conf.* 145–149.
- Sontya, M., Siallagan, M., Sabariah, M.K., 2007. Pengoptimasian Query Database Menggunakan Algoritma Genetik. *Libr. UNIKOM*.
- Theodoros, C., n.d. On-Line Analytical Processing. *Dep. Comput. Sci. CITY Lib. Stud. Affil. Inst. Univ. Sheff.*
- Usman, M., Asghar, S., Fong, S., 2010. Data mining and automatic OLAP schema generation. Presented at the 2010 5th International Conference on Digital Information Management, ICDIM 2010, pp. 35–43.
- Weinberger, A., Ender, M., n.d. The Power of Hybrid OLAP in a Multidimensional World. *SAS Inst. Inc* 133–25.
- Zhang, S., Ou, J., Wang, G., Liu, S., 2006. Data mining and its applications for high-rise structure intelligent form-optimization based on genetic algorithm, in: Proceedings of the World Congress on Intelligent Control and Automation (WCICA). pp. 8779–8783.

Zhao, X., Huang, Z., 2010. A quality evaluation approach for OLAP metadata of multidimensional OLAP data. IEEE, pp. 357–361. doi:10.1109/ICIME.2010.5477583.

Zhijuan, W., Hongchang, W., Xuefang, W., 2012. A Data Warehouse Design Method. IEEE, pp. 2063–2066. doi:10.1109/CSSS.2012.513