

PENINGKATAN KINERJA DISTRIBUTED DATABASE MELALUI METODE DMQ BASE LEVEL

Untung Rahardja¹
Hidayati²
Mia Novalia³

*e-mail : untung.rahardja@faculty.rahajarja.ac.id, hidayati@faculty.rahajarja.ac.id,
mianovalia@si.rahajarja.ac.id*

Diterima :8 Februari 2011 /Disetujui : 23 Februari 2011

ABSTRACT

The purpose of the application that is web-based information system to facilitate users in accessing information wherever and whenever. But this time the user needs to information is very diverse, so that the necessary structures and ways of working too diverse to meet these demands. Especially for information that is Write Once Read Thousands (WORT), and require information systems that can provide data as fast as possible in accordance with the standard TWT (Tolerable Wait Time), required a method using multi-layered display buffer is also called the method Quey Data Mart (DMQ) Base Level. Information systems do not have to bear to always perform the update to the data buffer, because this method is only suitable to be developed for data that are relatively semi-permanent. In this article, identified the problems encountered in a distributed system, especially a problem in presenting information that is wort, definitions, database architecture, characteristics, algorithms and benefits of the method DMQ Base Level. In its implementation, is shown listing program written using ASP script. Contribution method DMQ Base Level in Distributed Database is a solution that is very helpful to user needs, particularly in terms of improving the display speed of the previous data is very slow and in the process, considering the standard TWT (Tolerable Wait Time).

Key words: DMQ base level, Write Once Read Thousands (WORT), TWT (Tolerable Wait Time)

-
- 1. Dosen Jurusan Sistem Informasi, STMIK Raharja**
Jl. jend Sudirman No. 40 Modern Cikokol-Tangerang Telp. 5529692
 - 2. Dosen Jurusan Sistem Informasi, STMIK Raharja**
Jl. jend Sudirman No. 40 Modern Cikokol-Tangerang Telp. 5529692
 - 3. Mahasiswa Jurusan Sistem Informasi, STMIK Raharja**
Jl. jend Sudirman No. 40 Modern Cikokol-Tangerang Telp. 5529692

ABSTRAKSI

Tujuan diterapkannya sistem informasi berbasis web yaitu untuk mempermudah pengguna dalam mengakses informasi dimanapun dan kapanpun. Namun saat ini kebutuhan pengguna terhadap informasi sangat beragam, sehingga diperlukan struktur dan cara kerja yang beraneka guna memenuhi permintaan-permintaan tersebut. Khususnya bagi informasi yang sifatnya *Write Once Read Thousands (WORT)*, serta membutuhkan sistem informasi yang dapat menyediakan data secepat mungkin sesuai dengan standar *TWT (Tolerable Wait Time)*, diperlukan sebuah metode menggunakan *buffer display* berlapis yang disebut juga metode *Data Mart Quey (DMQ) Base Level*. Sistem informasi tidak perlu dibebani untuk selalu melakukan *update* terhadap data *buffer*, karena metode ini hanya cocok dikembangkan bagi data yang sifatnya relatif semi permanen. Dalam artikel ini, diidentifikasi masalah yang dihadapi dalam suatu sistem terdistribusi khususnya masalah dalam menampilkan informasi yang bersifat *WORT*, definisi, arsitektur database, ciri khas, algoritma serta manfaat dari metode *DMQ Base Level*. Pada implementasinya, ditampilkan *listing program* yang ditulis menggunakan *script ASP*. Kontribusi metode *DMQ Base Level* dalam *Distributed Database* merupakan suatu solusi yang sangat membantu kebutuhan pengguna, khususnya dalam hal peningkatan kecepatan *display data* yang sebelumnya sangat lambat dan dalam prosesnya tetap memperhitungkan standar *TWT (Tolerable Wait Time)*.

Kata kunci: DMQ base level, Write Once Read Thousands (WORT), TWT (Tolerable Wait Time)

PENDAHULUAN

Database adalah kumpulan fakta-fakta sebagai representasi dari dunia nyata yang saling berhubungan dan mempunyai arti tertentu. *Database* digunakan untuk menyimpan data agar data tersebut dapat dimanipulasi dengan mudah, terjamin keakuratannya, efisien dalam penyimpanannya, dan tentu saja dapat dengan mudah untuk diakses kembali [13].

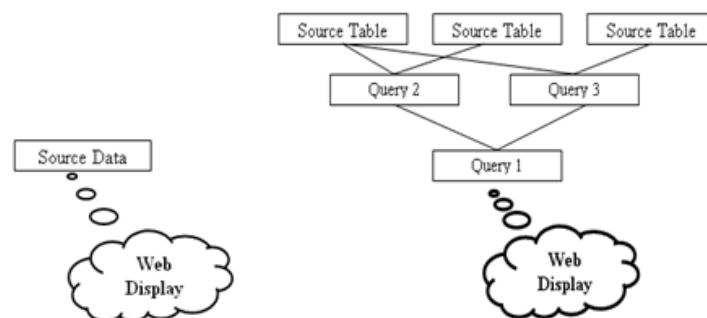
Pada dasarnya *database* dapat dibuat dan diolah dengan menggunakan suatu *software* (perangkat lunak). *Software* yang digunakan untuk mengelola dan memanggil *query database* disebut *Database Management System (DBMS)*. Perkembangan teknologi terus meningkat dengan cepat, mempengaruhi laju kebutuhan manusia atas informasi, terlebih disuatu organisasi atau perusahaan. Jika perusahaan benar-benar perlu untuk meningkatkan kinerja *query paralel*, maka *administrator* akan perlu menyesuaikan konfigurasi *hardware* yang sesuai dengan situasi sehingga *database* dapat melakukan *query paralel*, tunduk pada keterbatasan perangkat kerasnya.

Berkembangnya sistem informasi dari waktu ke waktu telah menghasilkan banyak informasi yang semakin kompleks. Kompleksnya informasi tersebut

disebabkan oleh banyaknya permintaan, jumlah data serta tingkat iterasi perintah SQL dalam suatu program. Didalam suatu sistem jaringan banyak organisasi ataupun perusahaan yang telah menerapkan *database* terdistribusi, namun semakin bertambahnya data sehingga menimbulkan masalah ketidaknyamanan khususnya perihal *respon time* dalam melakukan *display* data tersebut.

Tapi saat ini *bandwidth* dan pengalaman pengguna telah meningkat, pedoman untuk respon halaman web pun telah berevolusi. Menurut Schneiderman yang didirikan pada tahun 1980, bahwa 15 detik adalah waktu menunggu ditoleransi. Menurut King, bahwa untuk dapat muncul sebagai Internet dan Web di tahun 1990-an, waktu tanggapan toleransi dibagi menjadi 2, yaitu 8 - 12 detik tanpa umpan balik, dan 20 - 30 detik atau lebih dengan umpan balik. Pada tahun 1996 Jacob Nielsen pun menulis bahwa 10 detik adalah waktu respon maksimum sebelum pengguna kehilangan bunga. Zona Riset menerbitkan sebuah studi yang sering dikutip untuk Akamai yang menawarkan aturan 8 detik. Pada awal 2003, King menemukan bahwa rata-rata TWT (*Tolerable Wait Time*) adalah 8,6 detik untuk kondisi sekarang [6].

Untuk mengatasi masalah *response time* ini, biasanya digunakan sebuah metode dengan nama DMQ (*Data Mart Query*). DMQ merupakan metode yang menerapkan analogi “*Waste Space for Speed*”. DMQ terbentuk karena adanya pemisahan antara “*Engine*” dan “*Display*”. Dengan kata lain metode DMQ dapat langsung menampilkan *source code* pada *display* sedangkan proses *query* dikerjakan pada *engine*. Secara umum DMQ menghasilkan sebuah *display* data yang jauh lebih cepat dibandingkan dengan menggunakan metode umum, karena DMQ tidak melakukan proses lagi dalam menampilkan data. Dan akhirnya DMQ merupakan suatu solusi yang dapat membantu kebutuhan user pada proses *display* data yang sebelumnya sangat lambat dan tidak efisien [5].

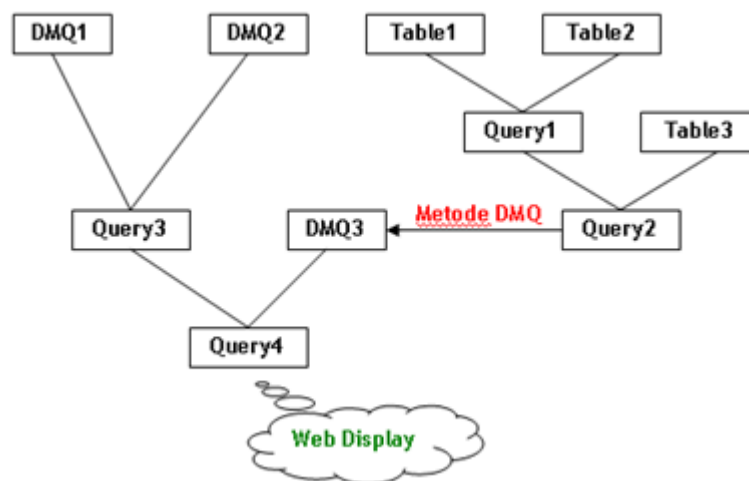


Gambar 1. Perbandingan source data konvensional dan source data dengan Data Mart Query

PERMASALAHAN

Pada awalnya metode DMQ (*Data Mart Query*) dinyatakan sebagai sebuah solusi yang dapat membantu meningkatkan kinerja *distributed database system*. Namun pada perjalanannya ternyata metode tersebut masih memerlukan pengembangan lanjutan, khususnya bagi *distributed database* yang sifatnya sangat kompleks. Salah satu contohnya adalah dalam proses view jadwal rencana study. Dikatakan kompleks, karena untuk menghasilkan jadwal rencana study ini memerlukan data hasil gabungan antara kesediaan ruangan, kesediaan hari, kesediaan waktu, serta kesediaan jadwal kelas. Sehingga untuk menghasilkan informasi tersebut, diperlukan query majemuk guna menggabungkan seluruh data yang ada.

Data mengenai kesediaan ruangan, kesediaan hari, kesediaan waktu, serta kesediaan jadwal kelas tersebut dalam prakteknya telah diproses menggunakan metode DMQ, sehingga dihasilkan banyak DMQ. Kemudian DMQ-DMQ tersebut diproses kembali seperti halnya cara kerja query majemuk, dan dihasilkanlah jadwal rencana study. Karena itu response time menjadi lambat. Karena setiap kali proses display dieksekusi, maka terjadi proses besar-besaran diantara DMQ-DMQ tersebut, dimana datanya pun bukanlah data yang sedikit. Berikut adalah alur prosesnya yang ditampilkan pada gambar 2.



Gambar 2. Cara kerja metode Data Mart Query (DMQ) Classic

Dari gambar 2 diatas, [5] kita bisa lihat bahwa untuk menghasilkan suatu tampilan pada *web display*, perlu dilakukan *query* bertingkat, meskipun data-data

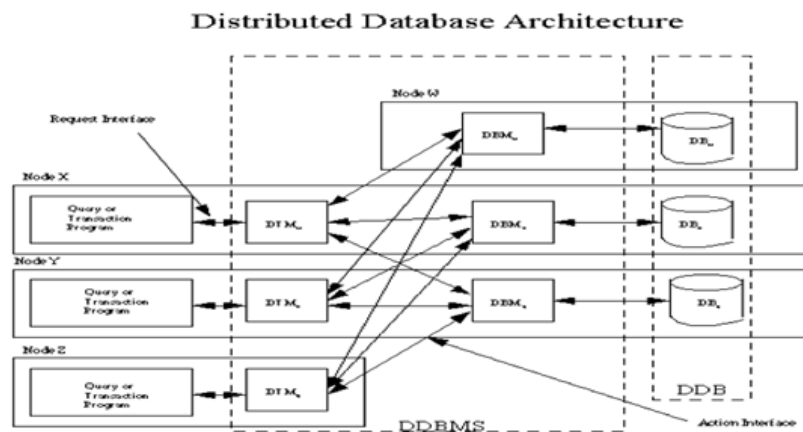
sebelumnya sudah menggunakan metode DMQ. *Source data* dilakukan mulai dari DMQ yang satu kemudian ke DMQ yang lain lalu dari *query* yang satu ke *query* yang lain. Bayangkan jika ada ratusan atau ribuan DMQ dan *query* didalam suatu *database*, kemudian *database* itu terdistribusi sehingga terjadi hubungan antara *database* yang satu dengan yang lain. Berapa lama waktu yang dibutuhkan hanya untuk memberikan satu tampilan *web*?

LITERATURE REVIEW

Banyak penelitian yang sebelumnya dilakukan mengenai *distributed database*. Dalam upaya pengembangan *distributed database* ini, perlu dilakukan studi pustaka sebagai salah satu dari penerapan metode penelitian. Diantaranya adalah mengidentifikasi kesenjangan (*identify gaps*), menghindari pembuatan ulang (*reinventing the wheel*), mengidentifikasi metode yang pernah dilakukan, meneruskan penelitian sebelumnya, serta mengetahui orang lain yang spesialisasi dan area penelitiannya sama dibidang ini. Beberapa *Literature review* tersebut adalah sebagai berikut :

1. Penelitian ini dilakukan oleh Jun Lin Lin dan Margaret H. Dunham dari Southern Methodist University dan Mario A. Nascimento berjudul “*A Survey of Distributed Database Checkpointing*”. Penelitian ini membahas mengenai *checkpointing* pada *database* terdistribusi dan pendekatan-pendekatan yang digunakan. Penelitian ini bermula dari adanya banyak survey yang dilakukan berkenaan dengan proses *recovery database*, dan banyak teknik yang diusulkan untuk mengatasinya. Dengan *distributed database checkpointing*, dapat mengurangi waktu proses *recovery* suatu kegagalan didalam *database* terdistribusi. *Checkpointing* dapat digambarkan sebagai suatu aktivitas menulis informasi ke penyimpanan yang stabil selama operasi normal dalam rangka mengurangi jumlah pekerjaan pada saat restart. Penelitian ini membantah bahwa sedikit batasan dan sedikit sumber daya menjadi masalah dalam pendekatan *database* terdistribusi, serta Membantah bahwa *checkpointing* hanya dapat digunakan untuk sistem distribusi yang *multidatabase*. Meskipun penelitian ini telah banyak dilakukan namun cukup rumit dalam implementasinya. Dengan penelitian ini kita dapat mengembangkan *database* terdistribusi dengan *checkpointing* untuk mempercepat proses *recovery database*[1].
2. Penelitian ini dilakukan oleh Carolyn Mitchell dari Norfolk State University berjudul “*Components of a Distributed Database*” tahun 2004. Penelitian ini membahas tentang komponen-komponen didalam *database*. Salah satu komponen utama dalam DDBMS adalah *Database Manager*. “Sebuah

Database Manager adalah perangkat lunak yang bertanggung jawab untuk memproses segmen data yang didistribusikan. Komponen utama lainnya adalah *Query User Interface*, yang merupakan sebuah program klien yang bertindak sebagai sebuah antarmuka untuk Transaksi Manager yang terdistribusi..” Sebuah Transaksi Manager terdistribusi adalah program yang menterjemahkan permintaan dari pengguna dan mengkonversi mereka ke *query database manager*, yang biasanya didistribusikan. Sebuah sistem *database* yang terdistribusi terbuat dari kedua manajer yaitu *Database Manager* dan Transaksi Manager Terdistribusi[2].



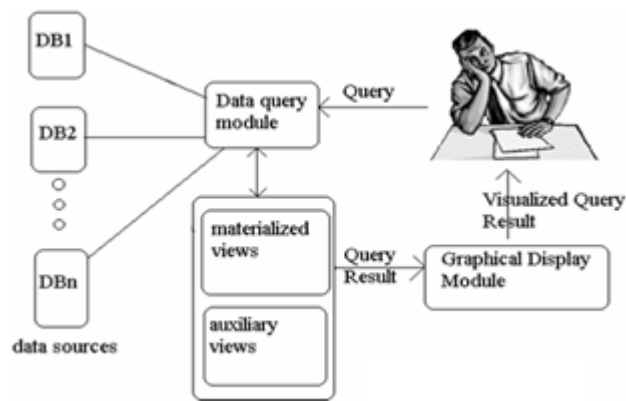
Gambar 3. Arsitektur Distributed Database dan Komponennya

3. Penelitian yang dilakukan oleh Hamidah Ibrahim, “*Deriving Global Integritas Dan Local Rules For Distributed Database*”. Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Putra Malaysia, 43400 UPM Serdang. Ibrahim mengatakan bahwa tujuan terpenting didalam *database* sistem adalah menjamin konsistensi data, yang berarti bahwa data yang terdapat dalam *database* harus baik dan akurat. Didalam pelaksanaannya untuk menjaga konsistensi perubahan data sangat sulit, khususnya untuk didistribusikan dalam *database*. Dalam tulisan ini, menjelaskan sebuah algoritma penegakan aturan berdasarkan mekanisme untuk didistribusikan *database* yang bertujuan meminimalisir jumlah data yang harus ditransfer atau diakses diseluruh jaringan yang menjaga konsistensi dari *database* di satu situs, yaitu di situs mana pembaruan perlu dilakukan. Teknik ini disebut sebagai tes integritas generasi, yang berasal dari lokal dan global integritas, dan aturan yang telah efektif dapat mengurangi biaya kendala dalam memeriksa suatu data yang telah

didistribusikan dalam lingkungan. Didalam penelitian ini telah berhasil menghasilkan sebuah sistem sentralistik yang besar dengan tingkat kehandalan yang tinggi untuk integritas data[3].

4. Penelitian yang dilakukan oleh Steven P. Coy dari *University of Maryland* berjudul “*Security Implication of the Choice of Distributed Database Management System Model: Relational Vs Object Oriented*”. Penelitian ini menjelaskan bahwa keamanan data harus dibenahi ketika mengembangkan *database* dan diantaranya memilih antara *relational* dan *object oriented model*. Banyak faktor yang harus dipertimbangkan, terutama dari segi efektifitas dan efisiensi, juga apakah sekuritas dan integritas ini memakan sumber daya yang terlalu besar tidak semata mata fitur keamanan. Kedua pilihan ini akan mempengaruhi kekuatan dan kelemahan dari *database* tersebut. Untuk *centralized database* kedua model ini bisa dikatakan sama baiknya. Namun untuk *distributed database*, *relational model* lebih unggul dibidang sekuritas. Ini lebih banyak disebabkan karena *object oriented model database* masih kurang maturitasnya. Sehingga didalam lingkungan heterogenous, proses integritasnya masih menimbulkan banyak masalah. OODBMS tetap saja masih perlu perkembangan teknologi lebih lanjut, namun di lingkungan homogenous, OODBMS dapat menjadi pilihan yang baik[4].
5. Penelitian ini dilakukan oleh Untung Rahardja dan Shakinah Badar dari Perguruan Tinggi Raharja Indonesia yang berjudul “*Penerapan metode Data Mart Query (DMQ) dalam Distributed Database System*” tahun 2009. Penelitian ini menjelaskan *Distributed database* memiliki banyak keunggulan terlebih untuk struktur organisasi saat ini. Namun, karena banyaknya database yang tersebar dan jumlah data yang banyak dan terus meningkat didalam suatu organisasi maupun perusahaan. Jika suatu database memiliki sejumlah data yang tersimpan dengan banyak query dan tabel, suatu permintaan mengakibatkan proses pencarian data atau source data menjadi lambat. Selain itu banyaknya user yang dapat mengakses suatu tampilan Web display suatu sistem informasi juga menjadi lambat. Dengan adanya *Data Mart Query (DMQ)* merupakan metode yang tepat untuk mempercepat waktu proses pada suatu sistem informasi dengan database yang terdistribusi. DMQ ini digunakan untuk menghindari penggunaan Query majemuk. Dengan demikian DMQ akan mengorbankan besarnya kapasitas penyimpanan data (*space hard disk*) untuk meningkatkan kecepatan (*increase speed*) dalam pengaksesan. Hal ini pun telah dibuktikan baik secara logik, secara grafik dengan perhitungan regresi linear dan korelasi linear dan juga melalui implementasi [5].

6. Penelitian yang dilakukan oleh Lubomir Stanchev dari University of Waterloo tahun 2001 berjudul “*Semantic Data Control In Distributed Database Environment*”. Penelitian ini menyatakan bahwa ada tiga tujuan utama dalam *semantic data control* yaitu: *view managemen*, *data security* dan *semantic integrity control*. Dalam sebuah relasi, fungsi-fungsi ini dapat mencapai keseragaman dengan menegakkan aturan-aturan manipulasi kontrol data. Solusinya adalah dengan sentralisasi ataupun terdistribusi. Dua hal utama yang efisien untuk melakukan kontrol adalah definisi data dan penyimpanan aturan (situs pilihan) dan penegakan desain algoritma yang meminimalkan biaya komunikasi. Masalahnya adalah sulit, karena peningkatan fungsi (dan umum) cenderung meningkatkan komunikasi situs. Solusi untuk semantik data kontrol terdistribusi adalah eksistensi dari sentralisasi solusi. Masalahnya adalah sederhana jika aturan kontrol sepenuhnya direplikasi di semua situs dan sulit jika situs otonomi dipatenkan. Selain itu, khusus optimasi dapat dilakukan untuk meminimalkan biaya kontrol data tetapi dengan tambahan *overhead* seperti pengelolaan data *snapshot*. Dengan demikian, spesifikasi kontrol data terdistribusi harus disertakan pada desain *database* sehingga biaya kontrol update untuk program-program ini juga dipertimbangkan[7].



Gambar 4. Data Visualization dengan materialized and auxiliary views

7. Penelitian ini dilakukan oleh Nadezhda Filipova dan Filcho Filipov 2008 dari *University of Economics*. Varna, Bul. Kniaz BorisI berjudul “*Development of database for distributed information measurement and control system*”. Penelitian ini menjelaskan mengenai pengembangan *database* dari pengukuran informasi yang didistribusikan dan sistem kontrol yang menerapkan metode optik untuk plasma spectroscopy fisika dan penelitian atom collisions dan

menyediakan akses untuk mendapat informasi dan sumber daya perangkat keras di jaringan Intranet/Internet, berdasarkan *database* pada sistem manajemen *database* Oracle9i. Perangkat lunak klien yang diwujudkan adalah dalam *Java Language*. Perangkat lunak ini dikembangkan dengan menggunakan model arsitektur, yang memisahkan aplikasi data dari komponen grafis presentasi dan masukan pengolahan logika. Berikut grafis presentasi telah dilaksanakan, pengukuran radiasi dari Spectra beam plasma dan benda, perangsangan fungsi *non-elastic collisions* dari berat partikel dan analisis data yang diperoleh dalam percobaan sebelumnya. Berikut grafis klien yang memiliki fungsi interaksi dengan *database browsing* informasi tentang percobaan dari jenis tertentu, pencarian data dengan berbagai kriteria, dan memasukkan informasi tentang validasi percobaan sebelumnya[8].

8. Penelitian ini dilakukan oleh Allison L. Powell James C.dkk, Perancis Departemen Ilmu Komputer Universitas Virginia, berjudul "*The Impact of Database Selection on Distributed Searching*". Penelitian ini menjelaskan bahwa *distributed searching* terdiri dari 3 bagian yaitu *database selection*, *query processing*, dan *results merging*. Cukup beberapa *database* yang dijadikan *database* seleksi (tidak semuanya) dan performa akan meningkat cukup signifikan. Bila seleksi *database* dilakukan dengan baik, pencarian secara *distributed* akan berkinerja lebih baik dibandingkan pencarian secara sentralisasi. Pencarian *database* juga ditambahkan proses seleksi dan ranking sehingga secara potensial meningkatkan efektifitas pencarian data[9].
9. Penelitian ini dilakukan oleh Ashwini Nangia School of Chemistry, University of Hyderabad. India berjudul "*Database research in crystal engineering*". Penelitian ini menjelaskan bahwa Database Cambridge Struktural merupakan gudang lebih dari 250 000 molekul kecil struktur kristal organik dan organometalik. Analisis statistik data struktural memberikan pemahaman yang lebih baik dari sifat ikatan hidrogen dan interaksi antarmolekul dan memfasilitasi dalam identifikasi sering terjadi pola interaksi dan synthons supramolekul. Dengan pengetahuan yang disempurnakan synthons ikatan hidrogen dalam kristal, strategi novel mungkin dirancang untuk perakitan-diri arsitektur supramolekul dari molekul multifungsi. Fenomena kembar polimorfisme dan pseudopolymorphism telah dipelajari oleh penggalian informasi mengenai konformasi molekul dan inklusi pelarut dari database. Geometri pendekatan donor ikatan hidrogen untuk berbagai kelompok fungsional yang berasal dari struktur kristal molekul kecil yang digunakan sebagai model dalam desain obat berbasis struktur dan pemetaan pharmacophore. Dengan demikian, implikasi dari rekayasa kristal hari ini jauh melampaui desain kristal organik

dan logam-organik menjadi bahan supramolekul, nanoteknologi, ligan-prediksi struktur protein mengikat dan kristal. Sorot ini survei pentingnya penelitian database dalam beberapa daerah [10].

10. Penelitian ini dilakukan oleh Guoray Cai, School of Information Sciences and Technology Penn State University, University Park. Berjudul *Contextualization of Geospatial Database Semantics for Mediating Human-GIS Dialogues*. Penelitian ini menjelaskan bahwa database ini meluas semantik data spasial untuk memasukkan tidak hanya isi dan skema, tetapi juga konteks penggunaannya. Kita tentukan seperti model semantik dalam tiga komponen terkait: konteks pengetahuan, basis ontologi konteks, dan interpretasi konteks-sensitif. Seperti kontekstualisasi data spasial semantik memungkinkan data yang mendasari sama untuk mengambil formulir beberapa semantik, dan disambiguasi spasial konsep yang didasarkan pada konteks lokal. Kami menunjukkan bagaimana seperti model semantik mendukung dikontekstualisasikan interpretasi konsep ruang samar-samar selama interaksi manusia-GIS. Kami menggunakan dialog percakapan sebagai mekanisme untuk melakukan diagnosis kolaboratif konteks dan untuk mengkoordinasikan berbagi makna di seluruh agen dan konteks [12].

Dari sepuluh *literature review* yang ada, telah banyak penelitian mengenai *checkpointing*, *Components of a Distributed Database*, pembahasan *component database system*, juga mengenai *Data Mart Query*. Namun belum adanya penelitian mengenai kekurangan dari metode DMQ, sehingga perlu dilakukan penelitian lanjutan perihal DMQ Base Level ini.

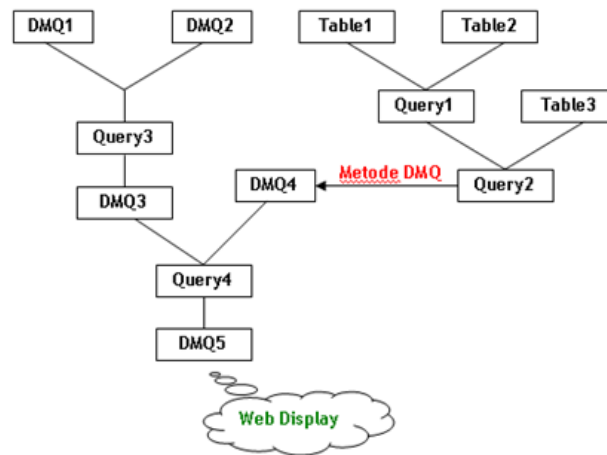
HASIL DAN PEMBAHASAN

1. Pemecahan Masalah

Untuk mengatasi masalah seperti yang telah dijelaskan diatas, dapat dilakukan melalui penerapan metode DMQ base level. Berikut merupakan 4 ciri khas dari metode DMQ base level yang diterapkan pada proses view jadwal rencana study (JRS) :

1. Informasi yang dibutuhkan bersifat WORT (*Write Once Read Thousand*).
2. Sesuai dengan standar TWT (*Tolerable Wait Time*).
3. Adanya pemisahan antara “*Engine*” dan “*Display*”.
4. Mengorbankan space hardisk untuk meningkatkan kecepatan.

Metode DMQ base level yang diterapkan pada proses view jadwal rencana study (JRS) ini cara kerjanya yaitu digambarkan seperti pada bagan dibawah ini :



Gambar 5. Cara kerja metode *Data Mart Query (DMQ) Base Level*

Pada gambar diatas, terdapat proses pembentukan DMQ bertingkat dari query majemuk yang dapat menyebabkan borosnya penggunaan space hardisk database server. Hal ini sesuai dengan ciri khas metode DMQ base level point 4. Karena itu, metode ini tidak cocok untuk menghasilkan informasi yang selalu update dalam hitungan detik, namun lebih cocok untuk informasi yang bersifat relatif semi permanen, yang perubahannya sangat jarang sekali, bahkan mungkin perubahannya hanya terjadi satu semester sekali. Informasi seperti ini bisa disebut juga dengan istilah WORT (Write Once Read Thousand). Hal ini sesuai dengan ciri khas DMQ base level point 1.

DMQ base level adalah metode lanjutan dari DMQ, dimana DMQ merupakan salah satu metode yang terbentuk dengan adanya pemisahan antara “*Engine*” dan “*Display*”. Dengan kata lain metode ini dapat langsung menampilkan *source code* pada *display* dan proses *query* yang dikerjakan pada *engine*. Secara umum DMQ menghasilkan sebuah *display data* yang jauh lebih cepat dibandingkan dengan menggunakan metode pada umumnya, karena DMQ tidak melakukan proses lagi dalam menampilkan data. Dan akhirnya DMQ merupakan suatu solusi yang dapat membantu kebutuhan *user* pada proses *display data* yang sebelumnya sangat lambat dan tidak efisien, menjadi sangat cepat serta sesuai dengan standar TWT. Hal ini sesuai dengan ciri khas DMQ base level point 2 dan 3.

2. Implementasi

Peningkatan kinerja distributed database melalui metode DMQ base level sudah diimplementasikan pada Perguruan Tinggi Raharja, yakni pada sistem informasi SIS O-OJRS (Otomatisasi Online Jadwal Rencana Studi). *Students Information Services* atau yang biasa disingkat SIS, merupakan sistem yang dikembangkan oleh Perguruan Tinggi Raharja dengan tujuan sebagai sistem pelayanan informasi kepada mahasiswa secara optimal. Pengembangan SIS juga merupakan akses publikasi bagi Perguruan Tinggi Raharja di bidang ilmu komputer dan dunia IT khususnya.

SIS sudah dikembangkan ke dalam beberapa versi, dimana masing-masing merupakan kelanjutan dari SIS versi sebelumnya. SIS O-OJRS (Otomatisasi Online Jadwal Rencana Studi) merupakan SIS versi 4.2. Sesuai namanya, SIS O-OJRS dibuat untuk kebutuhan batal tambah KST mahasiswa secara online.

Hasil akhir yang ingin dicapai dari SIS O-OJRS ini yaitu dihasilkanlah sebuah sistem yang secara flexibel melayani segala proses batal tambah KST mahasiswa secara online dan dapat dilakukan secara mandiri. Prosesnya terdiri dari view jrs, view kst dan kstf, tambah kelas, pindah kelas, serta hapus kelas. Karena itulah, untuk dapat meningkatkan pelayanan terhadap mahasiswa, diterapkanlah metode *DMQ base level* ini.



Gambar 6. Biodata Mahasiswa pada SIS

Gambar 6 diatas merupakan tampilan biodata mahasiswa. Dari halaman tersebut dapat diketahui segala informasi mengenai seorang mahasiswa, baik mengenai data pribadinya maupun mengenai data akademiknya. Untuk melihat jadwal rencana study, silahkan masuk pada menu O-OJRS. Akan ditampilkan tombol-tombol berbeda sesuai dengan status mahasiswa tersebut. Jika seorang mahasiswa dinyatakan belum membayar registrasi perkuliahan, maka pada menu O-OJRS hanya akan ditampilkan 1 tombol yaitu RBR (Rincian Biaya Registrasi). Namun jika seorang mahasiswa dinyatakan sudah registrasi, maka akan ada 3 tombol berbeda yang diklasifikasikan kembali berdasarkan status KST nya. Jika KST nya dinyatakan belum final, akan ditampilkan tombol JRS Aktif, tombol KST Aktif, dan tombol O-

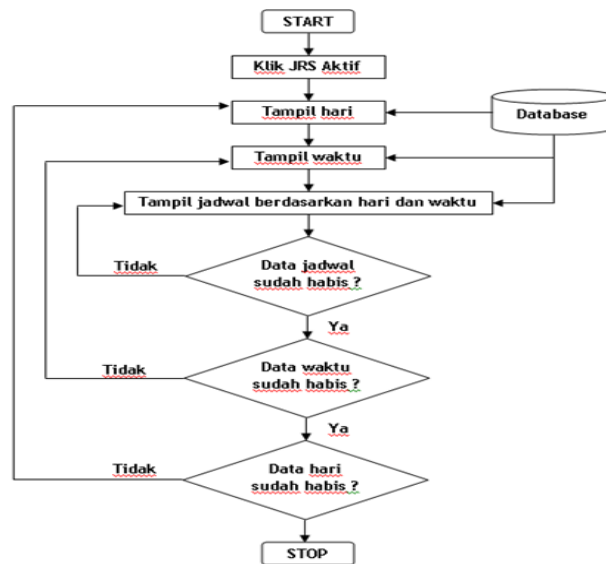
OJRS. Sementara jika mahasiswa tersebut dinyatakan bahwa status KST nya sudah final, maka akan ditampilkan 3 tombol yang berbeda pula meliputi tombol JRS Aktif, tombol KSTF, dan tombol RBK. Untuk melihat jadwal rencana study, dapat dilakukan dengan mengklik tombol JRS Aktif. Berikut adalah tampilannya:

Jadwal Rencana Studi (JRS) Semester: Genap TA: 2010/2011

HARI	JAM	RUANG																													
		RUANG 1	RUANG 2	RUANG 3	RUANG 4	RUANG 5	RUANG 6	RUANG 7	RUANG 8	RUANG 9	RUANG 10	RUANG 11	RUANG 12	RUANG 13	RUANG 14	RUANG 15	RUANG 16	RUANG 17	RUANG 18	RUANG 19	RUANG 20	RUANG 21	RUANG 22	RUANG 23	RUANG 24	RUANG 25	RUANG 26	RUANG 27	RUANG 28		
Dini	08:00	RUANG 1	RUANG 2	RUANG 3	RUANG 4	RUANG 5	RUANG 6	RUANG 7	RUANG 8	RUANG 9	RUANG 10	RUANG 11	RUANG 12	RUANG 13	RUANG 14	RUANG 15	RUANG 16	RUANG 17	RUANG 18	RUANG 19	RUANG 20	RUANG 21	RUANG 22	RUANG 23	RUANG 24	RUANG 25	RUANG 26	RUANG 27	RUANG 28		
	Siang	12:00	RUANG 1	RUANG 2	RUANG 3	RUANG 4	RUANG 5	RUANG 6	RUANG 7	RUANG 8	RUANG 9	RUANG 10	RUANG 11	RUANG 12	RUANG 13	RUANG 14	RUANG 15	RUANG 16	RUANG 17	RUANG 18	RUANG 19	RUANG 20	RUANG 21	RUANG 22	RUANG 23	RUANG 24	RUANG 25	RUANG 26	RUANG 27	RUANG 28	
		Malam	18:00	RUANG 1	RUANG 2	RUANG 3	RUANG 4	RUANG 5	RUANG 6	RUANG 7	RUANG 8	RUANG 9	RUANG 10	RUANG 11	RUANG 12	RUANG 13	RUANG 14	RUANG 15	RUANG 16	RUANG 17	RUANG 18	RUANG 19	RUANG 20	RUANG 21	RUANG 22	RUANG 23	RUANG 24	RUANG 25	RUANG 26	RUANG 27	RUANG 28

Gambar 7. Jadwal Rencana Study (JRS) pada SIS

Pada gambar 7 diatas, metode DMQ base level diterapkan pada proses view jadwal rencana study. Proses view jadwal rencana study ini hanya memerlukan waktu display ± 6 detik, karena hanya membaca dari 1 DMQ untuk display data, tanpa adanya proses pembacaan lagi dari tabel-tabel ataupun query-query lainnya.



Gambar 8. Flowchart view Jadwal Rencana Study (JRS) pada SIS

b. *Source Code*

```

<!--#include file="database.asp"-->
<%
Sql="select * from BT_Hari whereurut<4 order by Urut"
set rs=conn.execute(Sql)
Sql3="select * from BT_Akademik"
set rs3=conn.execute(Sql3)
While not rs.eof
    Sql2="select * from cT_Jam2 order by No"
    set rs2=conn.execute(Sql2)
    While not rs2.eof
        Sql7a="select * from DMQ_JRS_AKTIF_OOJRS where
Mulai="&rs2("Masuk")&" and Hari="&trim(rs("Nama_Hari"))&"
        set rs7a=conn.execute(Sql7a)
        rs2.movenext
    rs.movenext
wend
%>
  
```

KESIMPULAN

Berdasarkan uraian diatas, disimpulkan bahwa *Data Mart Query* (DMQ) *base level* sangat tepat digunakan untuk mempercepat waktu proses view data bagi informasi yang sifatnya WORT (*Write One Read Thousand*). Metode ini digunakan untuk menghindari penggunaan *Query* majemuk karena banyaknya data yang tersebar dalam suatu sistem database yang terdistribusi, sehingga terjadinya query besar-besaran pada saat setiap kali membutuhkan data. Dengan menggunakan DMQ *base level* ini, memungkinkan sebuah display data dapat ditampilkan dengan sangat cepat. Sehingga telah dibuktikan bahwa metode ini sesuai dengan standar TWT (*Tolerable Wait Time*), yang dapat memenuhi kenyamanan pengguna dari segi *response time*.

PUSTAKA

1. Petr Homola (2008), Proceedings of the ACL-08. *A Distributed Database for Mobile NLP Applications*. USA : Institute of Formal and Applied Linguistics Charles University.
2. Filipova Nadezhda dan Filipov Filcho (2008). *Development Of Database For Distributed Information Measurement And Control System*. University of Economics. Varna, Bul. Kniaz Boris I.
3. Hamidah Ibrahim (2001). *Deriving Global And Local Integrity Rules For A Distributed Database*. Departement of Computer Science Faculty of Computer Science and Information Technology, University Putra Malaysia 43400 UPM Serdang..
4. Steven P Coy (2008). *Security Implications of the Choice of Distributed Database Management System Model: Relational Vs Object Oriented*. University of Maryland.
5. Rahardja, Untung and Badar, Shakinah (2009). *Penerapan metode Data Mart Query (DMQ) dalam Distributed Database System*. CCIT Journal. Perguruan Tinggi Raharja of Indonesian
6. Rahardja Untung, Hidayati and Ardyanti, Reny (2010). *Write Validation*. CCIT Journal. Perguruan Tinggi Raharja of Indonesian
7. Stanchev Lubomir (2001). *Semantic Data Control In Distributed Database Environment*. University of Waterloo.
8. Desmet Lieven, Jacobs Bart, Piessens Frank, and Joosen Wouter (2004). *A Generic Architecture for Web Applications to Support Threat Analysis of Infrastructural Components*. Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS).. Mitchell Carolyn.

Component of a distributed database. Department of Computer science, Norfolk state University.

9. Anonim (2010). *Database* : Dari Wikipedia Indonesia, ensiklopedia bebas berbahasa Indonesia (2010). dari :<http://id.wikipedia.org/>
10. Huang, Y, Huang, S, Lin, T, and Tsai, C (2003), Proceedings of the 12th international conference on World Wide Web pp. 148–159. *Web application security assessment by fault injection and behavior monitoring.* New York, NY, USA, ACM Press.
11. Nangia, Ashwini (2002). *Database research in crystal engineering.* India : School of Chemistry, University of Hyderabad.
12. Cai, Guoray (2005), Journal of Database Management. *Contextualization of Geospatial Database Semantics for Mediating Human-GIS Dialogues.* Dari : School of Information Sciences and Technology Penn State University, University Park. Robi'in, Bambang (2005). *Manajemen dan Administrasi Databaser menggunakan SQL Server 2000.* Yogyakarta : Andi