

DATA MART QUERY (DMQ) SOLUSI MEMPERCEPAT DISPLAY DATA DALAM DISTRIBUTED DATABASE SISTEM

Untung Rahardja¹, Retantyo Wardoyo², Shakinah Badar³

Jurusan Sistem Informasi, Perguruan Tinggi Raharja., Jurusan Ilmu Komputer, Universitas Gajah Mada
Jl. Jend. Sudirman No. 40 Modern Cikokol Tangerang 15117
Telp. (021) 5529692, 5529586 Faks. (021) 5529742

E-mail: untung.rahardja@faculty.raharja.ac.id, rw@ugm.ac.id, shakinahbadar@si.raharja.ac.id

ABSTRAKS

Perkembangan IPTEK yang semakin pesat mendorong timbulnya berbagai pertumbuhan teknologi jaringan. Dengan menggunakan suatu jaringan komputer, suatu informasi dapat tersebar dengan lebih cepat. Informasi disalurkan antara satu database dengan database yang lain sehingga memungkinkan untuk saling berhubungan dalam suatu sistem database yang terdistribusi. Banyaknya database yang tersebar membuat suatu sistem kadang berjalan lambat. Sehingga loading yang berlarut-larut dan sering terjadi error karena proses pencarian yang terlalu lama. Dengan Data Mart Query, proses display data dapat lebih cepat. Dalam artikel ini, dilakukan identifikasi permasalahan yang timbul dari suatu database yang terdistribusi, berbagai penelitian mengenai distributed database serta manfaat dari metode ini. Metode Data Mart Query dalam distributed database merupakan solusi tepat dalam proses display data yang efektif.

Kata Kunci: distributed database, data mart query

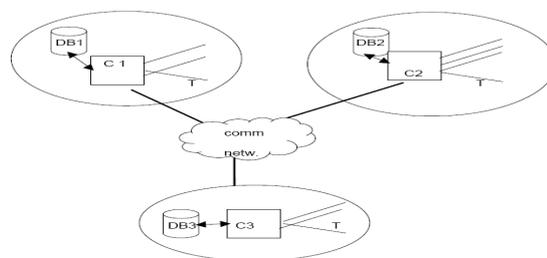
1. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat, mendorong munculnya ide-ide baru. System yang awalnya berjalan secara manual perlahan-lahan mulai tergeser. Teknologi kini menjadi suatu keharusan, terlebih disuatu organisasi atau perusahaan. Informasi terus mengalir dan jumlahnya semakin lama semakin meningkat seiring dengan jumlah permintaan, serta jumlah data yang semakin banyak. Selain itu penggunaan *database* dalam suatu perusahaan dan organisasi pun semakin banyak terlebih dengan adanya sistem jaringan. *Database* dapat didistribusikan dari satu komputer ke komputer lain. Jumlah arus pemakai pun meningkat seiring besarnya organisasi atau perusahaan.

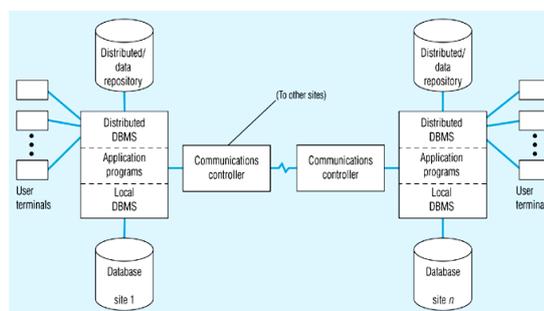
Organisasi maupun perusahaan membutuhkan sistem informasi untuk mengumpulkan, mengolah dan menyimpan data serta menyalurkan suatu informasi. Berkembangnya sistem informasi dari waktu ke waktu telah menghasilkan banyak informasi yang semakin kompleks. Kompleksnya informasi tersebut disebabkan oleh banyaknya permintaan, jumlah data serta tingkat iterasi perintah SQL dalam suatu program..

Pemanfaatan teknologi informasi oleh organisasi atau perusahaan secara garis besar bertujuan untuk memudahkan pelaksanaan proses bisnis dan meningkatkan kemampuan kompetitif. Melalui teknologi informasi, diharapkan proses bisnis perusahaan dapat dilaksanakan lebih mudah, cepat, efisien dan efektif. Penggunaan teknologi jaringan didalam suatu organisasi ataupun perusahaan menjadi hal yang biasa. Didalam suatu sistem jaringan sekarang ini banyak organisasi ataupun perusahaan yang telah menerapkan *database* terdistribusi untuk sistem *datasenya*.

Database terdistribusi merupakan sebuah *database* yang berada dibawah kontrol DBMS sentral dimana tempat penyimpanan tidak terpusat ke suatu CPU tetapi mungkin disimpan di *multiple* komputer dalam lokasi fisik yang sama atau disebarakan melalui jaringan komputer yang saling terkoneksi.



Gambar 1. *Distributed Database* secara geografis



Gambar 2. Arsitektur *Distributed Database*

Gambar diatas, merupakan gambaran arsitektur *database* yang terdistribusi. Dimana didalam sistem *database* terdistribusi ini memungkinkan beberapa terminal terkoneksi dalam suatu sistem *database*.

Dan masing-masing terminal ini bisa mengakses atau memperoleh data dari *database* baik yang ada dikomputer pusat maupun dikomputer lokal ataupun *database* yang satu dengan *database* yang lain. *Database* terdistribusi juga mempunyai keunggulan seperti dapat merefleksikan struktur organisasi, otonomi lokal. Kesalahan dalam satu fragmen tidak akan mempengaruhi *database* keseluruhan. Adanya *balancing database* didalam *server* dan sistem dapat dimodifikasi tanpa mempengaruhi modul lain.

Untuk itu, penyimpanan data pada tabel SQL *server* dalam sebuah *distributed database*, merupakan langkah praktis yang dilakukan banyak kalangan saat ini. Bagi instansi tertentu proses penyajian data haruslah cepat, misalnya pada google.com dimana penyajian data yang diinginkan oleh kosumen haruslah dengan cepat berada pada tampilan *display*, padahal tidak sedikit data yang harus dikeluarkan. Pada dasarnya penyajian data dari tabel memerlukan waktu proses yang sesuai dengan banyaknya data yang akan di tampilkan. Penggunaan cara konvensional pada dasarnya adalah cara praktis, karena tidak membutuhkan pengeditan bila data di *database* bertambah, namun apakah kecepatan tampilan akan lama bila data yang ditampilkan banyak.

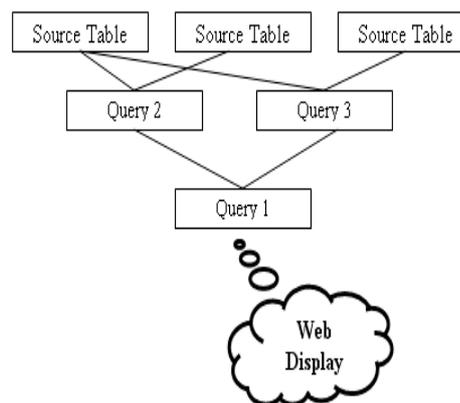
2. PERMASALAHAN

Distributed database memang memiliki banyak keunggulan terlebih untuk struktur organisasi saat ini, diantaranya yaitu:

1. Pengawasan distribusi dan pengambilan data
Jika beberpa site yang berbeda dihubungkan, seorang pemakai yang berada pada satu site dapat mengakses data pada site lain.
2. Reliability dan availability
Sistem distribusi dapat terus menerus berfungsi dalam menghadapi kegagalan dari site sendiri atau mata rantai komunikasi antar site.
3. Otonomi lokal
Pendistribusian sistem mengijinkan sekelompok individu dalam sebuah perusahaan untuk melatih pengawasan lokal melalui data mereka sendiri. Dengan kemampuan ini dapat mengurangi ketergantungan pada pusat pemrosesan.
4. Efisiensi dan fleksibel
Data dalam sistem distribusi dapat disimpan dekat dengan titik diman data tersebut dipergunakan. Data dapat secara dinamik bergerak atau disain, atau salinannya dapat dihapus.

Namun diantara keunggulan itu, *distributed database* juga memungkinkan suatu sistem menjadi lebih kompleks, 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* atau *Web display* suatu sistem informasi juga menjadi lambat.. Berikut ini disampaikan tampilan sumber data konvensional yang memiliki *query* bertingkat:



Gambar 3. *Source Data* Konvensional

Dari gambar diatas, kita bisa lihat bahwa untuk menghasilkan suatu tampilan pada *web display*, pada *source data* konvensional perlu dilakukan *query* bertingkat. *Source data* dilakukan mulai dari tabel yang satu kemudian tabel yang lain lalu ke *query* yang satu ke *query* yang lain. Bayangkan jika ada ratusan atau ribuan tabel 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*?

Dari penjelasan di atas, dapat dirumuskan beberapa permasalahan yaitu sebagai berikut :

1. Apakah proses *view* karena *query* bertingkat pada *Database* terdistribusi sudah ada penelitiannya?
2. Berapa dampak dari lambatnya sebuah proses akibat *query* bertingkat?
3. Metode apa yang dapat digunakan untuk mempercepat proses display pada suatu sistem *database* terdistribusi?
4. Apa kelemahan dan kelebihan dengan metode baru yang diusulkan ini?

3. LITERATURE REVIEW

Telah ada berbagai penelitian yang sebelumnya dilakukan mengenai *distributed database*. Dalam upaya pengembangan *distributed database* ini perlu dilakukan studi pustaka sebagai salah satu dari penerapan metode penelitian yang akan dilakukan. Diantaranya adalah mengidentifikasi

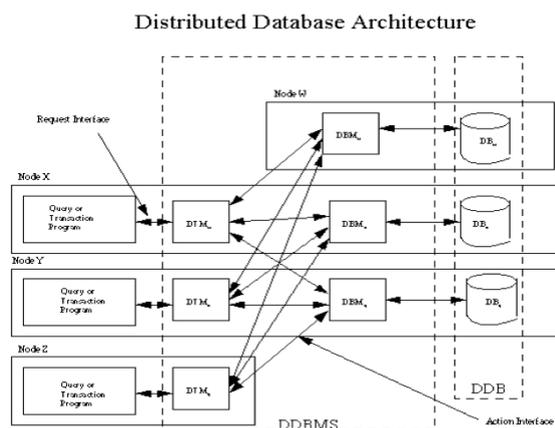
kesenjangan (*identify gaps*), menghindari pembuatan ulang (*reinventing the wheel*), mengidentifikasi metode yang pernah dilakukan, meneruskan penelitian sebelumnya, serta mengetahui orang lain yang spesialisasi dan area penelitiannya sama dibidang ini. Beberapa *Literature review* tersebut adalah sebagai berikut :

1. Penelitian ini dilakukan oleh Jun Lin Lin dan Margaret H. Dunham (1997) 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*.

2. Penelitian ini dilakukan oleh David J. DeWitt dari Universitas Wisconsin dan Jim Gray (1992) berjudul “*Parallel Database Systems: The Future of High Performance Database Processing*”. Penelitian ini dilakukan dengan Konsep *database* terdistribusi yang merupakan *database* yang disimpan pada beberapa komputer yang terdistribusi satu sama lain. Pada penelitian ini, dijelaskan Sistem *database* paralel mulai menggantikan *Mainframe* komputer besar untuk pengolahan data dan transaksi tugas. Paralel *database* komputer memiliki arsitektur yang berkembang dari penggunaan perangkat lunak yang eksotik untuk perangkat keras yang paralel. Seperti kebanyakan aplikasi, user menginginkan *hardware* sistem *database* yang murah, cepat. Ini menyangkut tentang prosesor, memori dan disk. Akibatnya, konsep *hardware database* yang eksotis tidak sesuai untuk teknologi saat

ini. Di lain sisi, ketersediaan *microprocessors* cepat, murah dan kecil menjadi paket standar murah tapi cepat sehingga menjadi *platform* yang ideal untuk sistem *database* paralel. Stonebraker mengusulkan rancangan sederhana untuk spektrum disain yaitu *shared memory*, *shared disk* dan *shared nothing*. Dan bahasa yang digunakan dalam *database* adalah SQL sesuai dengan standar ANSI dan ISO. Dengan penelitian ini, kita dapat mengembangkan sistem *database* agar dapat digunakan diberbagai ruang lingkup.

3. Penelitian ini dilakukan oleh Carolyn Mitchell (2004) dari Norfolk State University berjudul “*Components of a Distributed Database*”. 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



Gambar 4. Arsitektur *Distributed Database* dan Komponennya

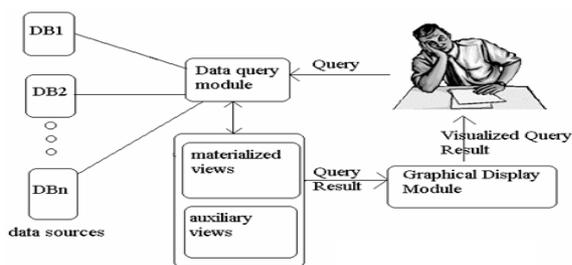
4. Penelitian yang dilakukan oleh Hamidah Ibrahim (2001), “*Deriving Global Integrity and Local Rules For Distributed Database*”. Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Putra Malaysia, 43400 UPM Serdang. Ibrahim mengatakan bahwa tujuan terpenting didalam *database* sistem

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

5. Penelitian yang dilakukan oleh Steven P. Coy (2008) 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 pengembangan lebih lanjut, namun di lingkungan homogenous, OODBMS dapat menjadi pilihan yang baik.
6. Penelitian yang dilakukan oleh Stephane Gançarski, Claudia León, Hubert Naacke, Marta Rukoz and Pablo Santini (2006) yang berjudul "*Integrity Constraint Checking in Distributed Nested Transactions over a Database Cluster*" adalah sebuah solusi untuk memeriksa integritas dan kendala global dalam berhubungan multi *database* sistem. Penelitian ini juga menyajikan hasil eksperimental yang diperoleh atas solusi PC *cluster* dengan Oracle9i DBMS. Tujuan adalah melakukan eksperimentasi untuk mengukur waktu yang dihabiskan dalam memeriksa kendala global dalam sistem yang terdistribusi. Alhasil menunjukkan bahwa *overhead* berkurang hingga 50% dibandingkan dengan pemeriksaan integritas yang terpusat. Studi menunjukkan bahwa sistem berkemungkinan besar melanggar *referential integrity* dan *global conjunctive constraints*. Namun dengan cara *distributed nested transactions*, dengan adanya eksekusi dan parallelism, integritas dapat lebih terjamin.
7. Penelitian ini dilakukan oleh Allison L. Powell James C.dkk, (2000) Perancis Departemen Ilmu Komputer Universitas Virginia, berjudul 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.
8. Penelitian ini dilakukan oleh Yin-Fu Huang dan HER JYH-CHEN (2001) dari Universitas Nasional Sains dan Teknologi Yunlin Taiwan, berjudul "*Fragment Allocation in Distributed Database Design*". Pada penelitian ini menjelaskan mengenai *Wild Area Network (WAN)*, fragmen alokasi adalah isu utama dalam distribusi *database* desain karena kekhawatiran kinerja keseluruhan didistribusikan pada system *database*. Disini system yang diusulkan sederhana dan modelnya yang komprehensif mencerminkan aktivitas transaksi yang didistribusikan dalam *database*. Berdasarkan model dan informasi transaksi, dua bentuk algoritma dikembangkan untuk mendapatkan alokasi yang optimal seperti total biaya komunikasi yang sebisa mungkin diminimalkan. Hasilnya menunjukkan bahwa alokasi fragmentasi ditemukan dengan menggunakan algoritma yang tepat akan menjadi lebih optimal.
9. 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.

- 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.



Gambar 5. *Data Visualization* dengan *materialized and auxiliary views*

Dari berbagai *literature review* yang ada, telah banyak penelitian mengenai *checkpointing*, *parallel database system*, pembahasan *component database system*, juga mengenai *security*. Disamping itu juga ada pembahasan mengenai *nested transaction*, *distributed searching*, *view management* dan juga *fragment allocation*. Namun dapat disimpulkan pula bahwa belum ada peneliti yang secara khusus membahas atau mengatasi masalah proses *view* yang lambat akibat *query* bertingkat.

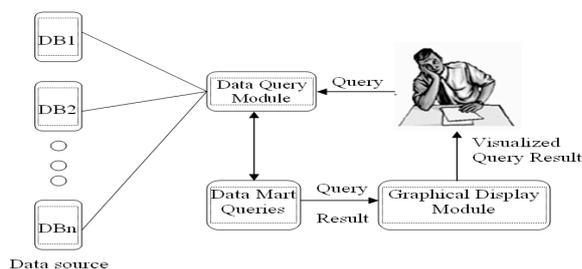
4. PEMECAHAN MASALAH

Untuk mengatasi berbagai masalah diatas, maka diperlukan proses yang cepat dan efisien dalam mengakses seluruh data yang banyak dan tidak teratur di *database*, terlebih untuk suatu sistem *database* yang terdistribusi. Saat ini programmer lebih memilih menggunakan Ms. Acces dan fungsi *query* untuk mengerjakan seluruh *script* perintah. Alhasil proses *query* besar-besaran terjadi setiap membutuhkan data. Penggunaan *SQL server* bukanlah hal yang baru dalam hal ini, maka dari itu diusulkan untuk dibentuknya suatu system yang lebih menuju ke proses pada saat loading penyajian data dan memiliki kecepatan yang secara linear lebih cepat dibandingkan dengan cara konvensional. DMQ (*Data Mart Query*) merupakan metode yang menerapkan analogi "*Waste Space for Speed*". DMQ juga merupakan salah satu metode yang berbentuk terhadap pemisahan antara "*Engine*" dan "*Display*". Dengan kata lain metode DMQ 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 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.

Pada *Data Mart Query* sumber data berasal dari tabel. Jadi pada proses DMQ ini, mengalokasikan seluruh data yang dipilih ke dalam suatu tabel. Sehingga *user* tidak perlu memikirkan pembuatan struktur tabel tujuan, yang perlu dipikirkan hanyalah dimana data tersebut berada. DMQ ini digunakan untuk menghindari penggunaan *Query* majemuk. DMQ akan mengorbankan besarnya kapasitas penyimpanan data (*space harddisc*) untuk meningkatkan kecepatan (*increase speed*). DMQ membutuhkan *trigger update data* untuk menghasilkan data yang mutakhir.

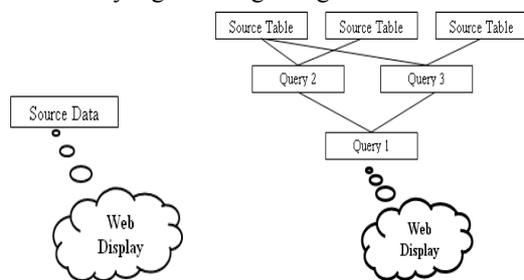
Berikut ini adalah gambaran mengenai permintaan data dari *user*. Dimana seorang *user* melakukan permintaan akan suatu tampilan, kemudian *data query module* melakukan pencarian pada db1, db2 sampai dbn. Dengan menggunakan *Data Mart Query* atau DMQ dari *data query module*

langsung menghasilkan *query* yang diinginkan dalam bentuk *graphical display module* yang dapat dilihat oleh *user*.



Gambar 6. *Data Visualization* dengan DMQ

Dengan *Data Mart Query* (DMQ) proses pencarian data lebih singkat, karena tidak seperti *source data* konvensional yang harus mencari dari tabel. *Data Mart Query* (DMQ) bisa memotong waktu proses karena proses pencarian data hanya ke satu tabel yang telah digabungkan.



Gambar 7. Perbandingan *source data* konvensional dan *source data* dengan *Data Mart Query*

Jika dibandingkan maka tampilannya akan seperti gambar diatas. Dimana dengan DMQ bisa membuat tampilan *web* lebih cepat dilakukan karena tidak memerlukan proses pencarian yang rumit.

5. KESIMPULAN

Berdasarkan uraian diatas, disimpulkan bahwa *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 harddisc*) untuk meningkatkan kecepatan (*increase speed*) dalam pengaksesan.

PUSTAKA

- Coy P. Steven. (2008). *Security Implications of the Choice of Distributed Database Management System Model: Relational Vs Object Oriented*. University of Maryland.
- DeWitt. D.J., Gray.J. *Parallel Database Systems: The Future of High Performance Database Processing*. (1992). San Francisco: Computer Sciences Department, University of Wisconsin.
- Filipova Nadezhda dan Filipov Filcho. *Development Of Database For Distributed Information*

Measurement And Control System University of Economics. Varna, Bul. Kniaz Boris I. 2008.

- Gangarski, Stephane., Leon, Claudia., Hurbert Naacke, Marta Rukoz and Pablo Santini. (2006). *Integrity Constraint Checking In Distributed Nested Transactions Over A Database Clustur*. Laboratoire de l'Information Paris 6. University Pierre et Marie Curie 8 rue du Capitaine Scott, 75015, Paris. Centro de Computacion Paralela Y Distribuida, Universidad Central de Venezuela. Apdo. 47002, Los Chaguaramos, 1041 A, Caracas, Venezuela.
- Huang Yin-Fu dan JYH-CHEN HER. (2001). *Fragment Allocation in Distributed Database Design*. Nasional Yunlin Universitas Sains dan Teknologi Yunlin. Taiwan 640, R.O.C.
- Ibrahim Hamidah. (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.
- Lin. J. L., Dunham M. H. and Nascimento M. A. (1997). *A Survey of Distributed Database Checkpointing*. Texas: Department of computer science and engineering, Shouthern Methodist University.
- Mitchell Carolyn. (2004). *Component of a distributed database*. Department of Computer science, Norfolk state University.
- Powell L Allison., James C. French, Jamie Callan, Margaret Connell and Charles L. Viles. (2000). *The Impact of Database Selection on Distributed Searching*. 23rd ACM SIGIR Conference on Information Retrieval (SIGIR'00), pages 232-239.
- Stanchev Lubomir. (2001). *Semantic Data Control In Distributed Database Environment*. University of Waterloo.