

SISTEM BASIS DATA TERDISTRIBUSI CLIENT-SERVER 3-TIER BERBASIS KOMPONEN

Sumarsono*

Abstract

A distributed database (DDB) is a collection of multiple, logically interrelated databases distributed over a computer network. A distributed database management system (distributed DBMS) is the software system that permits the management of the distributed database and makes the distribution transparent to the users. The term distributed database system (DDBS) is typically used to refer to the combination of DDB and the distributed DBMS. Distributed DBMSs are similar to distributed file systems (see Distributed File Systems) in that both facilitate access to distributed data. The administration of distributed data is a need that nowadays many enterprises strive to fulfill. However, it is often difficult for these enterprises to find a software product capable of providing the solution they require. This is due to the inherent complexity to distributed data access, which has caused the majority of database applications to provide distributed access to a centralized database as a simpler alternative to distributed database systems. This article demonstrates that it is possible to design a component-based reference architecture of a distributed database systems (D-DBS) with a 3-tiered client-server structure from the solution given to the data distribution problem through the implementation of a global conceptual schema. Therefore, this proposal reduces the design complexity of a D-DBS.

Keywords: distributed database system, component-based system, 3-tiered client-server system.

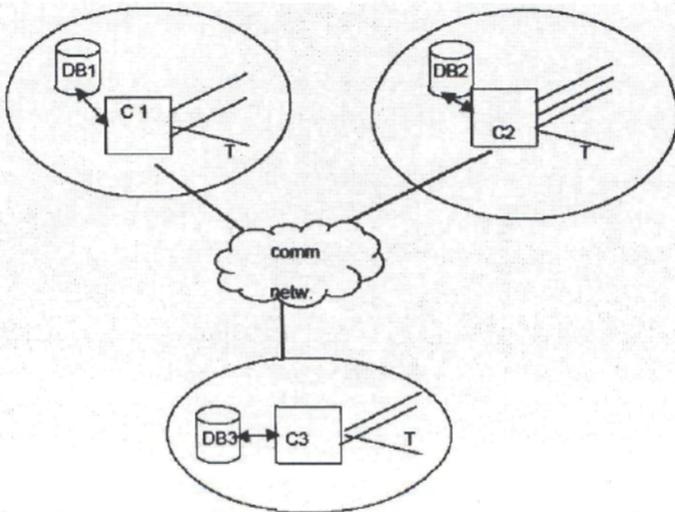
* Dosen Prodi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga

A. Pendahuluan

Dalam sebuah *database* terdistribusi, *database* disimpan pada beberapa komputer. Komputer-komputer dalam sebuah sistem terdistribusi berhubungan satu sama lain melalui bermacam-macam media komunikasi seperti *high-speed buses* atau *telephone line*. Sebuah sistem database terdistribusi berisikan sekumpulan site, di mana tiap-tiap site dapat berpartisipasi dalam pengeksekusian transaksi-transaksi yang mengakses data pada satu site atau beberapa site. Tiap-tiap site dapat memproses transaksi lokal yaitu sebuah transaksi yang mengakses data pada satu site di mana transaksi telah ditentukan. Sebuah site juga dapat mengambil bagian dalam mengeksekusi transaksi global yaitu transaksi yang mengakses data pada site yang berbeda di mana transaksi telah ditentukan, atau transaksi yang mengakses data pada beberapa site yang berbeda.

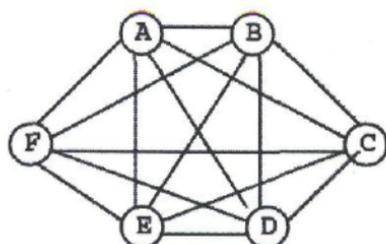
Ada dua aspek penting dari DDB :

1. Distribusi : data tidak disimpan pada tempat (prosesor) yang sama, sehingga DDB dapat dibedakan dari database tunggal, sentralisasi
2. Korelasi logika : data memiliki property yang berhubungan sehingga DDB dapat dibedakan dari sekumpulan database lokal atau file yang disimpan pada tempat yang berbeda pada jaringan komputer.

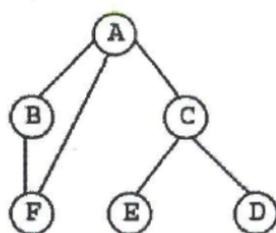


Gambar 1. Database Terdistribusi secara Geografis

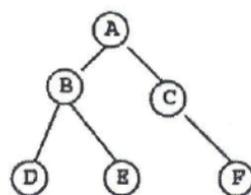
Site-site dalam database terdistribusi dihubungkan secara fisik dengan berbagai cara. Beberapa topologi digambarkan sebagai sebuah graph yang simpul simpulnya bersesuaian dengan site. Sebuah edge dari simpul A ke simpul B bersesuaian dengan sebuah hubungan langsung antara dua site. Beberapa konfigurasi (bentuk) digambarkan sebagai berikut:



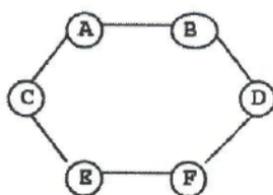
Fully Connected Network



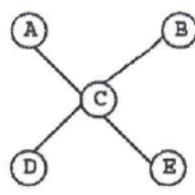
Partially Connected Network



Tree Structured Network



Ring Network



Star Network

Gambar 2. Tipologi Network

Fully Connected Network:

Keuntungan: kalau salah satu node rusak, yang lainnya masih dapat berjalan (tetapi biaya mahal).

Kerugian: *control management* tidak terjamin *Partially connected network*:

Partially Connected Network

Keuntungan: *reliability* rendah, biaya dapat ditekan

Kerugian: *control management* tidak terjamin

Tree structure network:

Keuntungan: bersifat sentral, *control management* lebih terjamin

Kerugian: kalau node pusat (A) rusak, semua akan rusak.

Ring Network (LAN):

Keuntungan: rusak satu, yang lain masih berjalan

Kerugian : Control management kurang terjamin karena bersifat desentralisasi

Star Network (LAN):

Keuntungan:

1. *control management* lebih terjamin, karena bersifat sentral
2. *reliability* rendah

Kerugian : kalau pusat rusak, yang lainnya rusak

Sistem Manajemen Database Terdistribusi (Distributed DBMS) merupakan sistem *software* yang dapat memelihara DDBS dan transparan ke user. DDBS bukan merupakan kumpulan dari file yang dapat disimpan tersendiri di setiap node dari jaringan komputer. Untuk membentuk DDBS, file tidak seharusnya berelasi secara logika saja, tetapi perlu ada struktur di antara file dan akses data bukan merupakan hal yang khusus.

Sistem manajemen basis data terdistribusi atau *distributed database management system* (DDBMS) merupakan *system* perangkat lunak yang mengatur basis data terdistribusi dan merancang distribusinya secara terbuka pada pengguna. Pernyataan *distributed database system* (DDBS) secara tipikal digunakan dengan menunjuk pada kombinasi dari DDB dan DDBMS. Distributed DBMS sebenarnya mirip dengan sistem file terdistribusi (*Distributed File Systems*) dalam hal fasilitas akses pada *distributed data*.

Pengelolaan administrasi terhadap data terdistribusi merupakan sebuah kebutuhan bagi seluruh *enterprise* dalam mencapai *powerfull*. Sering kali terdapat kesulitan bagi *enterprise* untuk menemukan produk perangkat lunak yang mampu memberikan solusi yang dibutuhkan *user*. Fakta ini kelihatannya berada di luar wilayah dimana teknologi jaringan menjadi konstan. Meskipun ketersediaan infrastruktur teknologi pemrosesan data terdistribusi terpenuhi, terdapat beberapa faktor yang menjadi penghalang dalam pengembangan perangkat lunak. Salah faktornya adalah kompleksitas yang melekat pada akses data terdistribusi. Kompleksitas ini adalah manifestasi dari serangkaian masalah seperti kebutuhan akan pemahaman yang lebih baik pada teknologi basisdata terdistribusi, pemrosesan *query* terdistribusi, model-model transaksi terdistribusi, analisa replikasi dan dampaknya pada arsitektur sistem basis data terdistribusi, strategis implementasi dari system manajemen basisdata terdistribusi, metodologi perancangan bagi

basisdata terdistribusi dan interkoneksi terhadap sistem pemrosesan data autonomus.¹

Dalam artikel ini akan dicoba menelaah kembali dan memperbaharui konsep basis data terdistribusi yang dihilangkan oleh aplikasi basis data saat ini melalui pemahaman terhadap arsitektur berbasis komponen dari sistem basis data terdistribusi dengan struktur client server 3 tier. Arsitekturnya didasarkan pada komponen-komponen, dengan harapan mampu mengimplementasikan aplikasi basis data dan melayani serta memfasilitasi standarisasi sistem basis data terdistribusi.

Kemudian, artikel ini mencoba mengajukan sebuah referensi arsitektur untuk mengurangi kompleksitas perancangan sistem basisdata terdistribusi. Sebuah sistem basis data terdistribusi (*distributed database system* -DDBS) dapat divisualisasikan sebagai *system client-server* yang memungkinkan klien mengakses data secara simultan dan disimpan ke dalam lebih dari 1 server².

Sebuah *distributed database system* (DDBS) disusun berdasarkan pada dua hal yaitu *distributed database* (DDB) dan *distributed database management system* (DDBMS).³ Seperti produk perangkat lunak pada umumnya, DDBS disusun ke dalam 3 level perancangan, yaitu *architecture*, *code* dan *executable*. Dan ini dimungkinkan untuk mendefinisikan level arsitektur sebagai koleksi dari kolaborasi komponen-komponen.⁴

B. Arsitektur DDBS

Dalam bagian ini, akan dijabarkan dasar komponen yang menunjuk arsitektur sistem basis data terdistribusi dengan struktur *client server 3-tier* menurut tiga pendekatan yang diajukan oleh Kangassalo.⁵ Pendekatannya didasarkan pada komponen yang digunakan untuk menjabarkan model struktur dan model komponen DDBS, pendekatan

¹ M. T. Özsu, P. Valduriez. "Distributed Database Systems: Where Are We Now?" *IEEE Computer*, 24 (8): 68–78, 1991.

² L. Maciaszek. *Requirements Analysis and System Design*, (t.t.: Addison-Wesley, 2001)

³ M. T. Özsu, P. Valduriez. *Principles of Distributed Database Systems*. (USA: Prentice Hall, 1999).

⁴ M. Shaw, D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*. (USA: Prentice Hall, 1996).

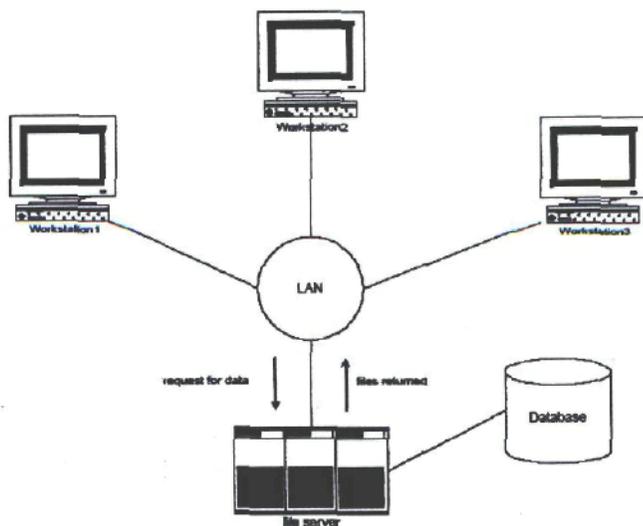
⁵ H. Kangassalo. *On the Selection of the Approach for the Development of the Reference Model for DBMS Standards*. ISO/TC 97/SC 5/WG 5 Document N104, 1983.

yang didasarkan pada data digunakan untuk menjabarkan model data logikal DDBS, dan pendekatan yang didasarkan pada fungsi digunakan untuk menjabarkan komponen *local database management system*-LDBMS.

1. Model Struktural DBMS dan DDBS

a. Model Struktural DBMS

Di lingkungan *file-server*, pemrosesan didistribusikan ke jaringan yang Local Area Network (LAN). File-Server menunjang kebutuhan file dengan aplikasi-aplikasi dan DBMS. Aplikasi dan DBMS bekerja pada masing-masing workstation. Permintaan file dari file-server diilustrasikan pada gambar di atas. File-server bertindak sebagai sebuah drive hard-disk yang dapat dipakai bersama. DBMS pada masing - masing workstation mengirimkan permintaan ke file-server untuk seluruh data yang dibutuhkan DBMS yang disimpan di disk seperti terlihat pada gambar berikut ini.

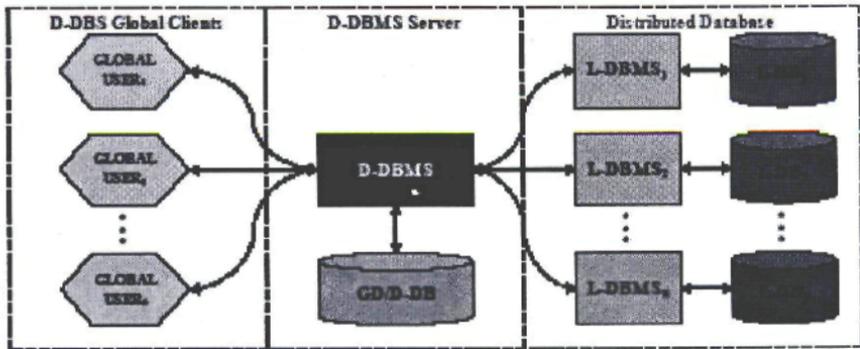


Gambar

b. Model Struktural DDBS

Model struktural dari DDBS adalah client-server 3-tier dengan server tunggal. Model ini memberikan keuntungan arsitektur yang di-layer-layer, kendali yang terpusat, "light clients" dan akses data terdistribusi. Sedangkan kerugiannya, model ini memasukkan point tunggal dari kegagalan sistem. Terdapat 3

layer dari model struktural ini yaitu *DDBS global client*, *Server D-DBMS* dan *Distributed Database*. Selengkapnya dapat dilihat pada gambar 1 di bawah ini:



Gambar 1. Arsitektur D-DBS client server 3-tier

Pada gambar 1 terdapat 3 layer dari arsitektur D-DBS, dengan penjelasan sebagai berikut (dari sisi kiri ke sisi kanan):

1. *D-DBS Global Clients*.

Layer ini dirangkai oleh *global users* atau klien system, memiliki kemampuan dalam pemrosesan sebuah perangkat lunak dalam menampilkan query ke layer server DDBMS untuk kepentingan aplikasi basisdata.

2. *D-DBMS Server*.

Layer ini menyimpan node server D-DBS. Layer ini dirangkai oleh proses D-DBMS dan sebuah basisdata dengan direktori *global directory/dictionary information (GD/D-DB)*. Layer ini menyediakan klien global dengan berbagai penjelasan, manipulasi dan mekanisme kontrol pada basisdata terdistribusi.

3. *Distributed Database*.

Layer ini berisi *distributed database (D-DB)* yang menyimpan data terdistribusi. Setiap node dari D-DB dirangkai oleh *local database management system (LDBMS)* dan *local database (L-DB)*.

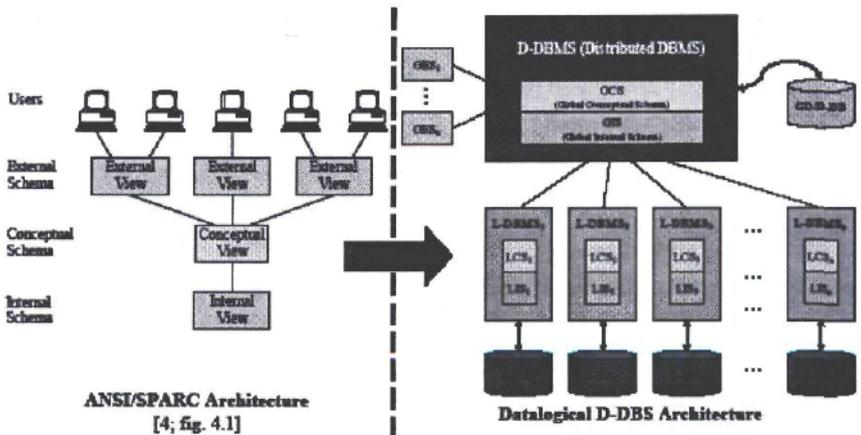
2. Model D-DBS Datalogical

Contoh klasik sebuah arsitektur berbasis data adalah model ANSI/SPARC yang di ajukan oleh Tsichritzis dan Klug.⁶ Arsitektur

⁶ M. Shaw, D. Garlan. *Software Architecture: Perspectives on an Emerging Discipline*, (USA: Prentice Hall, 1996).

ANSI/SPARC mengusulkan 3 gambaran informasi yaitu *external*, *conceptual* dan *internal*, dapat disamakan dengan user system, enterprise dan sistem informasi. Masing-masing gambaran tersebut direalisasikan dengan sebuah definisi skema. Perubahan data D-DBS hingga gambaran informasi ini menandakan pemetaan spesifik bagaimana menghasilkan definisi level 1 dari definisi level yang lain.⁷

Penyederhanaan model ANSI/SPARC tersebut menyediakan aplikasi dalam arsitektur D-DBS [4]. Oleh karena itu, dimungkinkan mengadaptasi model data logikai ANSI/SPARC kedalam model klien server 3-tier dengan sebuah server tunggal. Adaptasi ini ditunjukkan pada gambar 2.



Gambar 2. Pemetaan model ANSI/SPARC menjadi model struktural D-DBS.

Dalam gambar 2 dilihat bahwa model ANSI/SPARC (sisi kiri) dan adaptasinya ke dalam model klien server 3 tier dengan model struktural server tunggal (sisi kanan). Adaptasi tersebut mencakup elemen-elemen sebagai berikut :

1. *GES (Global External Schema)*. Merupakan serangkaian skema eksternal umum yang menghubungkan ke skema eksternal dalam arsitektur ANSI/SPARC;
2. *D-DBMS (Distributed DBMS)*. Merupakan suatu *Database Management System (DBMS)* dari *distributed database*. D-DBMS

⁷ M. T. Özsu, P. Valduriez. *Principles of Distributed Database Systems*. (USA: Prentice Hall, 1999).

mencakup skema konseptual global dan skema internal global. Informasi skema ini disimpan dalam *global directory/dictionary database* (GD/D-DB);

3. *GCS* (*Global Conceptual Schema*) dari basisdata terdistribusi
4. *GIS* (*Global Internal Schema*) dari basisdata terdistribusi. Termasuk pula mekanisme interkoneksi yang memungkinkan berkomunikasi dengan system manajemen basisdata lokal (*local database management systems - LDBMS*)
5. *GD/D-DB* (*Global directory/dictionary database*) yang menyimpan skema external global dan skema internal global agar supaya memungkinkan transformasi data antara kedua skema tersebut.⁸
6. *L-DBMSn*. Merupakan DBMS dari *local database*
7. *LCSn* (*local conceptual schema*) dari *local database*
8. *LISn* (*The local internal schema*) dari *local database*
9. *L-DBn*. (*local database*)

3. Model L-DBMS Berbasis Fungsi

Sebuah cara untuk mendefinisikan arsitektur berbasis fungsi dari komponen L-DBMS adalah dengan memvisualisasikan implementasi remote API dalam mengakses basisdata. Ide ini dapat dilihat pada gambar 3.



Gambar 3. Konsep L-DBMS sebagai Database Remote API

Dari gambar 3 dapat dijelaskan sebagai berikut: bahwa seorang user mengakses dengan remote dengan serangkaian fungsi (API) yang diimplementasikan oleh L-DBMS. Implementasi fungsi L-DBMS menurut *interface* basis data lokal (L-DB), di mana *interface* basis data lokal terdiri dari dukungan SQLnya [2]. API L-DBMS memiliki fungsionalitas untuk membaca dan menulis data dalam basisdata (DATA API) dalam memodifikasi skema basisdata (Schema API). Dalam arsitektur yang digambarkan dalam gambar 1 dijelaskan bahwa D-

⁸ M. T. Özsu, P. Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, 1999.

DBMS berperan sebagai user dalam L-DBMS. Ketika D-DBMS membutuhkan informasi basisdata local untuk menyelesaikan query global, D-DBMS mengirimkan request pada L-DBMS yang menterjemahkan petisi kedalam pernyataan yang dapat dimengerti ke dalam basisdata lokal menurut dukungan SQL basisdata local. L-DBMS kemudian menerima jawaban dari basisdata lokal, mengkonversi jawaban kedalam format data oleh D-DBMS dan akhirnya mengirim respon ke D-DBMS.

4. Model D-DBS Berbasis Komponen

Fungsi utama D-DBMS adalah menyediakan akses transparan ke data terdistribusi bagaimanapun lokasi data dalam basisdata terdistribusi. Dalam mencapai tujuan ini, perlu mengimplementasikan skema global konseptual (Global Conceptual Schema - GCS) sebagai **nexus** antara gambaran user dan gambaran terdistribusi dari system. dengan GCS sebagai poin referensi kami dapat menjelaskan kembali gambar 2 dan merancang komponen yang mampu menjabarkan arsitektur datalogikal sebelumnya. Implementasi distribusi data oleh GCS menjadi obyektif bagi arsitektur berbasis komponen, dan ini dimungkinkan untuk menyederhanakan perancangan arsitektur dengan mengenyampingkan skema eksternal global (global external schemas -GESn) dan konseptual serta skema internal dari basisdata lokal (LCSn and LISn). Dalam pada itu skema internal global (GIS) bertugas untuk merancang termasuk juga informasi yang dibutuhkan untuk mengakses data terdistribusi dan system manajemen basisdata lokal.

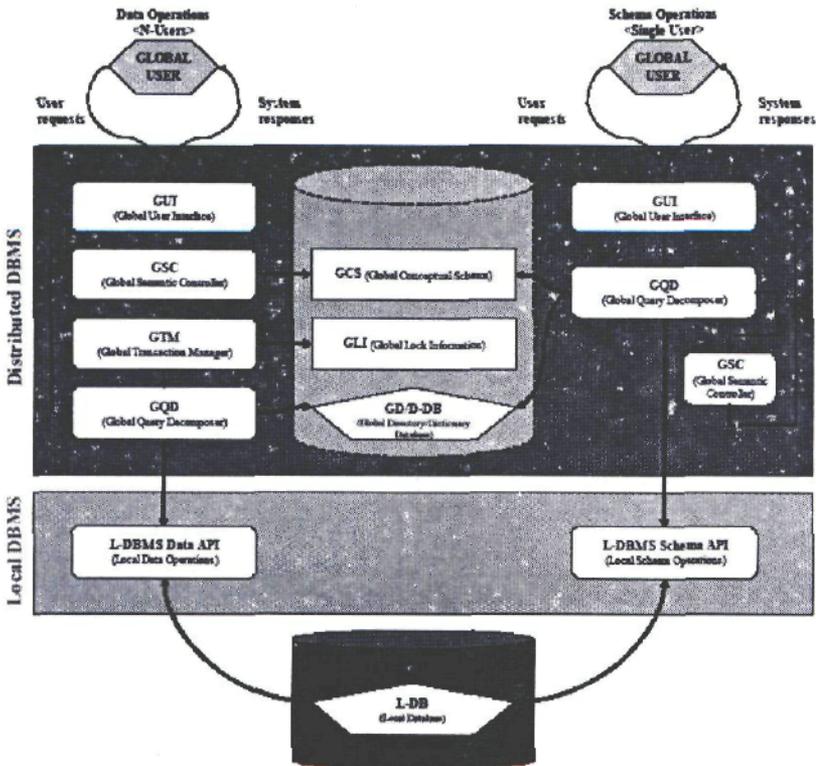
Dengan maksud untuk menyederhanakan arsitektur yang diajukan, GIS diintegrasikan sebagai bagian dari Global Conceptual Schema (GCS) sebab kedua skema dibutuhkan oleh system dan mereka dapat digambarkan sebagai unit. Diawali dengan observasi dan dengan arsitektur D-DBMS yang pernah diajukan oleh M.Tamer Özsu and Valduriez sebagai panduan, berikut ini komponen D-DBMS yang diusulkan.⁹

1. *GUI (Global User Interface)*. Komponen ini menerima perintah user dan format mengembalikan respon ke user. komponen ini terdiri dari akses basisdata API dalam penggunaan L-DBMS API.
2. *GSC (Global Semantic Controller)*. Komponen ini memverifikasi validitas perintah user menurut isi informasi dalam skema konseptual global (global conceptual schema)

⁹ M. T. Özsu, P. Valduriez. *Principles of Distributed Database Systems*. (USA: Prentice Hall, 1999).

3. *GTM (Global Transaction Manager)*. Komponen ini menjamin sinkronisasi akses ke basisdata terdistribusi supaya menjaga integritas data terdistribusi menggunakan kontrol konkurensi pesimistis menggunakan teknik penguncian (*locking technique*)¹⁰
4. *GQD (Global Query Decomposer)*. Komponen menterjemahkan perintah global kedalam serangkaian perintah lokal menggunakan koleksi informasi di dalam GCS.
5. *GD/D-DB (Global Directory/ Dictionary Database)*. Komponen ini menyimpan GCS termasuk informasi alokasi data dan *global lock information (GLI)*, dimana informasi ini berisi operasi akses data yang konkuren. Informasi ini dibutuhkan oleh komponen GTM dalam menampilkan tugas-tugasnya.

Arsitektur D-DBS yang berisi detail komponen D-DBMS dan fungsi-fungsi L-DBMS dapat dilihat dalam gambar 4.



Gambar 4. Arsitektur D-DBS Berbasis Komponen

¹⁰ Ibid.

Dalam gambar 4, dijelaskan bahwa ada perbedaan antara proses eksekusi pada **data operations** dengan **schema operations**. Operasi data disebut dengan L-DBMS Data API selama operasi skema disebut dengan L-DBMS Schema API. Eksekusi terhadap operasi data terjadi di mode multiuser atau mode n-user. Eksekusi yang terjadi akan sinkronisasi operasi oleh komponen GTM. Eksekusi operasi skema terjadi dalam mode single user dimana sejak semula hanya sebuah user, setelah itu sinkronisasi operasi skema tidak dibutuhkan lagi. Operasi skema yang diajukan dalam mode single user dengan tujuan untuk menyederhanakan perancangan arsitektur. Penyederhanaan ini sudah sesuai dari sisi perspektif praktis dimana perancangan skema basisdata harus dilaksanakan dalam waktu yang cukup dan meminimalkan frekuensi.¹¹

Urutan eksekusi dari operasi data sebagai berikut :

1. **GUI** (Global User Interface). Permintaan user mengeksekusi operasi data.
2. **GSC** (Global Semantic Controller). Validitas perintah user diverifikasi menurut isi skema konseptual global (GCS. Contoh validitas : verifikasi eksistensi dalam GCS nama table dan kolom disediakan oleh user sebagai parameter bagi operasi SELECT.
3. **GTM** (Global Transaction Manager). Jika operasi tersebut valid, parameter pembanding GTM dari operasi request dengan isi informasi kunci global. Berdasarkan hasil perbandingan ini, bantuan GTM sebuah kunci global atau operasi blok jika terdapat konflik akses. Jika kunci global dibantu, informasi dari kunci baru akan disimpan dalam GLI.
4. **GQD** (Global Query Decomposer). Jika kunci operasi ditanggung, maka GQD menterjemahkan perintah global kedalam serangkaian perintah lokal menggunakan koleksi informasi yang berada dalam GCS. Setelah itu kumpulan hasil eksekusi operasi lokal dimasukkan kedalam respon global.
5. **GTM** (Global Transaction Manager). Jika kunci operasi di Bantu, kunci global akan dihapus dan informasinya di hapus dari GLI.

Urutan eksekusi dari operasi skema sebagai berikut:

1. **GUI** (Global User Interface). Permintaan eksekusi user atas operasi skema.

¹¹ C. J. Date. *An Introduction to Database Systems*. (t.k.: Addison-Wesley, 2001).

2. **GD/D-GCS** (Global Directory/Dictionary Global Conceptual Schema). Validitas perintah user diverifikasi menurut isi GCS. Sebagai contoh validasi yang diverifikasi atas eksistensi dalam GCS nama table disediakan oleh iser sebagai parameter untuk operasi DROP TABLE.
3. **GQD** (Global Query Decomposer). Jika operasi valid, penterjemahan perintah global GQD kedalam serangkaian perintah global menggunakan informasi alokasi lainnya yang dimasukkan kedalam GCS atau dalam parameter user menurut tipe dari operasi itu sendiri. Akhirnya komponen GQD mengembalikan jawaban berupa sinyal jika operasi lokal ada atau ada tapi tidak dalam kondisi baik.
4. **GD/D-GCS** (Global Directory/Dictionary Global Conceptual Schema). Jika operasi GQD ditampilkan dengan benar, maka informasi GCS akan diupdate.
5. **GSC** (Global Semantic Controller). Jika operasi lokal ditampampilkan dengan benar, maka informasi atas komponen GSC juga akan diupdate.

C. Kesimpulan

Kajian ini merupakan salah satu alternatif yang mencoba menyusun arsitektur berbasis komponen dari sistem basis data terdistribusi dengan model struktur klien server 3 tier. Arsitektur ini menjabarkan secara detail level sistem, level komponen dan dari perspektif implementasi. Dalam kajian ini, aspek-aspek statis dan dinamik suatu arsitektur telah dijabarkan dengan suatu kesimpulan urutan eksekusi bagi kedua operasi (operasi data dan skema). Arsitektur telah divaliditas oleh dengan cara mengimplementasikan *prototype software* dalam J2SE platform versi 1.4.1_02 [11]. Implementasi ini menunjukkan kreasi terhadap system basis data terdistribusi berdasarkan arsitektur dan dengan harapan artikel dapat memudahkan tugas perancangan *software* sebagai bagian dari sistem.

DAFTAR PUSTAKA

- A. Geppert, K. R. Dittrich. *Component Database Systems*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2000.
- C. J. Date. *An Introduction to Database Systems*. Addison-Wesley, 2001.
- Ceri, Stefano & Pelagatti G, *Distributed Databases : Principles & Systems*, McGraw-Hill, Singapore, 1984
- D. Tsichritzis, A. Klug. *The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems*. Information Systems, 1:173–191, 1978.
- F. Álvarez. *Distribución de Datos para Bases de Datos Distribuidas, Una Arquitectura basada en Componentes de Software*. Master Thesis. ITESM, Campus Monterrey, May 2003.
- H. Kangassalo. *On the Selection of the Approach for the Development of the Reference Model for DBMS Standards*. ISO/TC 97/SC S/WG 5 Document N104, 1983.
- Korth, H.F & Siferschatz, *Database System Concepts*, McGraw-Hill, USA, 1986
- L. Maciaszek. *Requirements Analysis and System Design*, Addison-Wesley, 2001.
- M. T. Özsu, P. Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, 1999.
- M. Shaw, D. Garlan. *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall, 1996.
- Özsu, M.T & Valduriez, *Principles of Distributed Database Systems*, New Jersey: Prentice-Hall, 1991
- Özsu, M.T & Valduriez. Distributed Database Systems: Where are we now? *IEEE Computer*, 24 (8): 68–78, 1991.
- S. Birnam. *Distributed Java 2 Platform Database Development*, Prentice Hall, 2001.
- Sun Microsystems, Inc. Java 2 Platform, Standard Edition (J2SE™). <http://java.sun.com/>.