

# Analisis Implementasi Penanganan Distributed Heterogenous Database pada Arsitektur Cloud

Ria Aniansari<sup>1</sup>, Warih Maharani<sup>2</sup>, Alfian Akbar Gozali<sup>3</sup>

Fakultas Informatika IT Telkom Bandung

Jl. Telekomunikasi No. 1 Bandung

e-mail: [riaaniansari@gmail.com](mailto:riaaniansari@gmail.com), [wrh@ittelkom.ac.id](mailto:wrh@ittelkom.ac.id), [aag@ittelkom.ac.id](mailto:aag@ittelkom.ac.id)

## Abstrak

Perkembangan teknologi informasi menjadikan pengelolaan informasi dan kemudahan pengaksesan menjadi unsur penting dalam sistem informasi. Adanya penurunan performansi pada arsitektur yang tersentralisasi menyebabkan sebagian pihak mengatasinya dengan melakukan perubahan ke arsitektur basis data yang terdistribusi. Selain itu, munculnya arsitektur cloud pada database yang merupakan suatu arsitektur dimana sebuah database dapat diakses oleh client dari cloud service. Hal ini memberikan kemudahan saat melakukan pengaksesan informasi secara on demand. Oleh karena itu, pada penelitian ini penulis mengimplementasikan bagaimana penanganan database terdistribusi heterogen yang dibangun sebagai Database as a Service (DBaaS) pada arsitektur relational cloud. Adapun studi kasus yang digunakan adalah database distro dan menggunakan tiga jenis penyimpanan data pada database terdistribusi yaitu fragmentasi vertikal, fragmentasi horizontal dan replikasi. Hasil akhir yang didapat dalam penelitian ini berupa performansi sistem DBaaS dengan menggunakan parameter uji response time, throughput dan memory usage. Dimana pada implementasi yang dilakukan database terdistribusi dengan penyimpanan data secara fragmentasi vertikal memiliki hasil performansi yang lebih baik dibandingkan penyimpanan secara fragmentasi horizontal, dan replikasi. Berdasarkan hasil pengujian, fragmentasi vertikal lebih baik daripada fragmentasi horizontal dan replikasi.

**Kata kunci:** : response time, throughput, memory usage, fragmentasi, replikasi

## Abstract

Development of information technology makes management of information and ease of accessing a database become an important element in information systems. A decrease in performance on a centralized architecture causes can be handle by changing to the architecture of a distributed database. In addition, the architecture of cloud on the database is an architecture database that can be accessed by a client with cloud service. This provides a convenience when accessing information on demand. Therefore, in this research, author implemented how to handling of heterogeneous distributed database that was built as a Database as a Service (DBaaS) on cloud relational architecture. The case study used is the distro database and using 3 different data storage in distributed database that is vertical fragmentation, horizontal fragmentation, and replication. The results in this research are the performance of DBaaS system using the test parameter of response time, throughput and memory usage. Where performed on the implementation of a distributed database by storing data in vertical fragmentation has a better performance results than horizontal fragmentation, and replication. Based on the test results, the vertical fragmentation better than horizontal fragmentation and replication.

**Keywords:** : response time, throughput, memory usage, fragmentation, replication

## 1. Pendahuluan

Seiring dengan perkembangan teknologi informasi untuk diketahui atau diolah menjadi suatu bentuk informasi lainnya baik di akses dalam lokasi yang sama ataupun di lokasi yang berbeda-beda dengan adanya kemudahan pengaksesan suatu database dan kecepatan dalam pengaksesan data menjadikannya suatu unsur penting dalam suatu sistem informasi. Oleh karena itu, performansi suatu sistem informasi haruslah diperhatikan.

Arsitektur dalam basis data terdiri dari beberapa bentuk yaitu tersentralisasi dan terdistribusi. Perbedaan antara keduanya terletak pada lokasi basis data yang digunakan. Pada arsitektur tersentralisasi, basis data yang digunakan terletak pada sebuah server saja. Sedangkan untuk arsitektur terdistribusi, basis data yang digunakan terletak pada beberapa server yang terletak di lokasi yang berbeda. Permasalahan penurunan performansi yang biasa terjadi pada arsitektur tersentralisasi dapat ditangani dengan mengubah arsitektur basis datanya menjadi arsitektur basis data yang terdistribusi. Basis data terdistribusi terdiri dari dua tipe yaitu *homogen database* dan *heterogen database*. Jenis penyimpanan data pada database terdistribusi ada dua jenis yaitu fragmentasi dan replikasi. Saat ini basis data terdistribusi dengan tipe *heterogen database* memiliki perkembangan DBMS yang berbeda dalam

kemajuan komputer dan komunikasi sehingga menyebabkan peningkatan penggunaan implementasi *heterogen database* pada skenario jaringan DBMS.

Seiring dengan meningkatnya era internet saat ini menjadikan munculnya suatu arsitektur *cloud* yang merupakan suatu paradigma baru yang saat ini populer dalam bidang teknologi informasi dengan model yang nyaman, dapat diakses dimana-mana dan *on-demand* akses melalui jaringan untuk berbagai *resource*. [1] Arsitektur *cloud* pada *database* sendiri merupakan suatu arsitektur dimana sebuah *database* dapat diakses oleh *client*. [5]

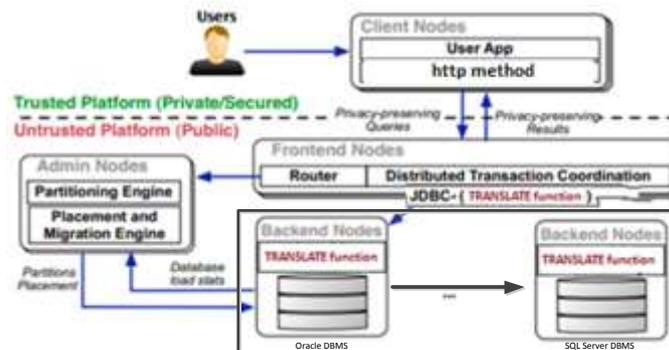
Berdasarkan dengan karakteristik *distributed heterogenous database* dan perkembangan arsitektur *cloud* dan keuntungan yang dimilikinya, maka penulis akan melakukan suatu analisa dan implementasi bagaimana penanganan *distributed heterogenous database* yang dibangun sebagai DBaaS (*Database as a Service*) pada arsitektur *cloud* berdasarkan performansi sistem dengan parameter uji *response time*, *throughput* dan *memory usage*.

## 2. Metode Penelitian

Tahapan penelitian yang dilakukan adalah melakukan studi literatur mengenai topik *distributed database*, untuk kemudian dibangun rancangan sistem yang dapat diimplementasikan pada arsitektur *cloud*.

### 2.1 Arsitektur Relational Cloud

Sistem ini di desain dan di rancang sesuai dengan topik penelitian yang dilakukan mengenai analisis dan implementasi penanganan *heterogenous distributed database* pada arsitektur *cloud*. Berikut ini adalah gambaran penelitian yang dilakukan :



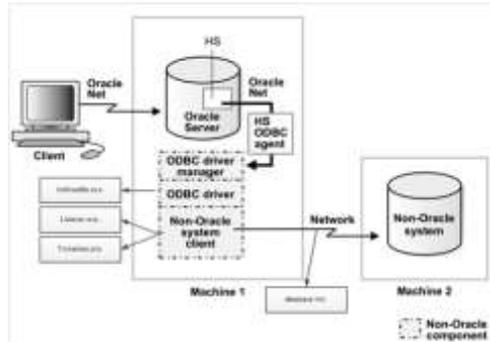
Gambar 1 : Modifikasi arsitektur Relational Cloud

Pada penelitian ini, penulis membangun *database* distribusi yang bersifat heterogen yang nantinya akan dibangun sebagai DBaaS pada layer *untrusted platform*.

Rancangan sistem basis data terdistribusi dengan jenis heterogen diimplementasikan dengan menggunakan dbms oracle 10g dan sql server 2008. Pengaksesan kedua *database* yang saling terdistribusi ini diatur oleh Oracle sebagai penyedia layanan distribusinya. Pada saat *user* akan melakukan pengaksesan atau *request* terhadap sistem maka *user* tidak perlu mengetahui bahwa *database* yang diakses berada di mana atau *database* apa yang digunakan ataupun apakah sifat *database* yang digunakan terdistribusi atau tidak, *user* hanya menerima hasil sesuai dengan *request* yang diakses dengan menggunakan arsitektur *cloud*. Dalam hal ini *cloud* berperan sebagai suatu fasilitas pengaksesan *request* yang diberikan kepada sistem dalam merequest data yang disediakan oleh *cloud provider*. Dimana hasil dari *request* tersebut akan didistribusikan kepada *client* atau *user*.

Berdasarkan sumber yang dijadikan sebagai acuan dikatakan bahwa pihak operator atau *developer* pada AaaS (*Application as a Service*) atau yang biasa disebut dengan SaaS (*Software as a Service*) mengembangkan aplikasi dalam bentuk *web service*. Maka pada analisis arsitektur disini yaitu *front-end* adalah antarmuka komunikasi antara aplikasi *server* dan aplikasi *client*. Sehingga *front-end* dibangun menggunakan *web service* yang nantinya akan menampilkan format data berupa JSON. Adapun *back-end* merupakan layer *database* yang digunakan pada DBaaS. Dimana DBaaS (*Database as a Service*) yang dibangun menggunakan *database relational* yang terdistribusi secara heterogen

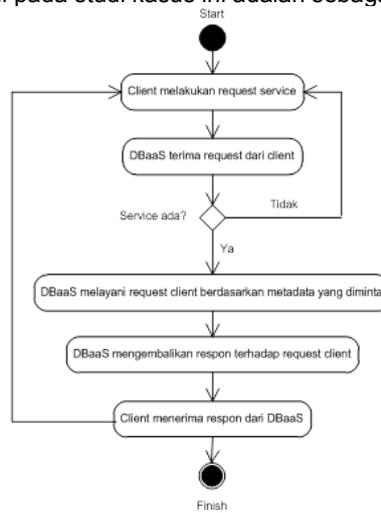
Aplikasi klien akan berkomunikasi dengan arsitektur *relational cloud* yang dibangun dengan menggunakan *http request*. Ketika *front-end* menerima *request* dari *client*, maka DBaaS akan memanggil *router* yang akan menganalisis kemana *request* harus dilanjutkan untuk suatu eksekusi di *database* yang terdistribusi heterogen tersebut berdasarkan metadata yang diminta oleh klien. Dimana penanganan data pada *database* yang terdistribusi antara DBMS Oracle dan DBMS SQL Server tersebut diatur oleh DBMS pada oracle yang dihubungkan dengan menggunakan *database link*.



Gambar 2 : arsitektur komunikasi heterogen database

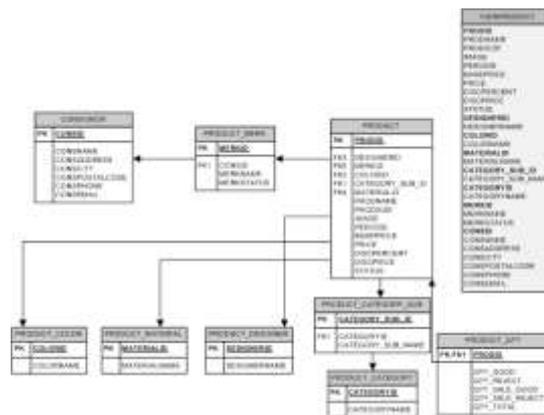
## 2.2 Alur Sistem DBaaS

Proses bisnis yang terjadi pada studi kasus ini adalah sebagai berikut:



Gambar 3: Alur Sistem DBaaS

Adapun studi kasus yang digunakan adalah kasus database store. Berikut ini adalah skema tabel relasinya :



Gambar 4 : Tabel relasi

## 3. Analisis Hasil Pengujian

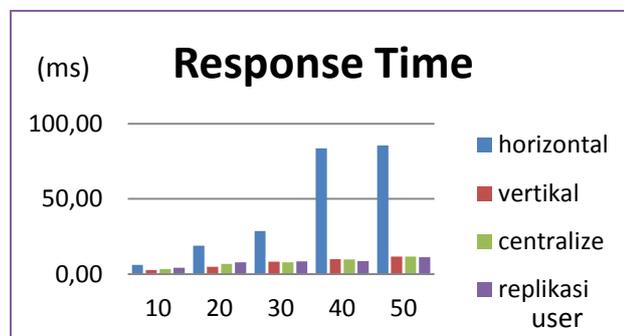
Pengukuran performansi yang di lakukan pada umumnya dilakukan dengan menggunakan parameter pengukuran *throughput*, *response time* dan *memory usage*. Dimana parameter *response time* merupakan waktu yang dibutuhkan DBaaS untuk melayani satu *service* yang diminta atau di *request* oleh *client*, sedangkan parameter *throughput* merupakan jumlah *service* yang bisa dilayani oleh DBaaS dalam satuan waktu tertentu. Sedangkan *memory usage* merupakan penggunaan memory pada saat melakukan

satu kali request. Berikut ini adalah skenario pengujian yang dilakukan terhadap sistem yang diimplementasikan :

- a. Pengujian dilakukan dengan menggunakan bantuan aplikasi pengujian dari titik *client* yang dilakukan sebanyak 3 kali pengujian berdasarkan 6 *request* yang diberikan yang akan kemudian akan diambil rata-rata *response time*, *throughputnya*, dan *memory usage* yang nantinya akan dianalisis.
- b. Pengujian dilakukan dengan 10,20,30,40,50 *user* secara bersamaan sebagai perbandingan konkurensi terhadap *request* yang dilakukan
- c. Pengujian dilakukan terhadap :
  - i. Sistem pengaksesan dua basis data yang terdistribusi secara heterogen dengan fragmentasi horizontal
  - ii. Sistem pengaksesan dua basis data yang terdistribusi secara heterogen dengan fragmentasi vertikal
  - iii. Sistem pengaksesan dua basis data yang terdistribusi secara heterogen dengan replikasi
  - iv. Pengaksesan basis data yang terpusat (*centralize database*)

### 3.1. Analisis terhadap Response Time

Berikut ini merupakan hasil analisis dari setiap skenario berdasarkan *request* yang dilakukan terhadap pengujian DBaaS. Dimana pada pengujian ini client melakukan request untuk mengambil seluruh data product yang tersimpan pada database.

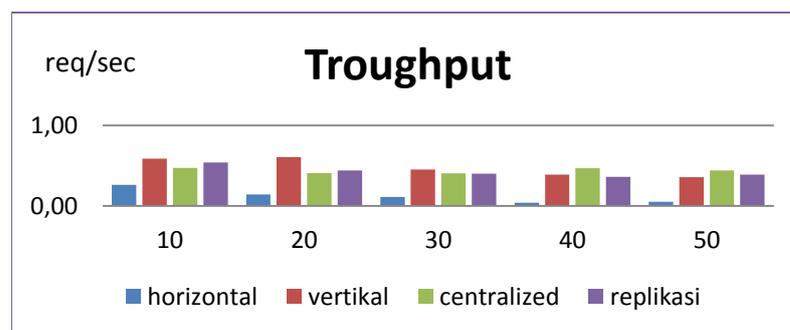


Gambar 5: Grafik response time view all product

Berdasarkan grafik gambar 4-1 dapat disimpulkan bahwa pada *database* terdistribusi heterogen dengan menggunakan fragmentasi horizontal memiliki peningkatan *response time* yang sangat signifikan berdasarkan banyaknya *user* sedangkan distribusi *database* heterogen dengan fragmentasi vertikal, replikasi dan *database* yang terpusat memiliki *response time* yang tidak jauh berbeda. Hal ini dikarenakan pada skenario View All Product ini dilakukan *join* sebanyak 8 tabel sehingga proses komputasi yang dibutuhkan untuk fragmentasi horizontal ini lebih besar karena proses yang dipakai terdiri dari proses *join* dan juga *union* untuk menggabungkan isi dari dua buah *database* yang terdistribusi dan juga menggunakan *database link*. Sedangkan pada fragmentasi vertikal serta replikasi proses komputasi yang perlu dikeluarkan hanya proses *join*. Dan untuk menggabungkan antara tabel yang terbagi/terfragmentasi hanya perlu menggunakan *database link* saja. Pada *database* terpusat *response time* cenderung lebih stabil hampir sama dengan vertikal dan replikasi.

### 3.2. Analisis terhadap Throughput

Berdasarkan hasil pengujian, diperoleh grafik di bawah ini :

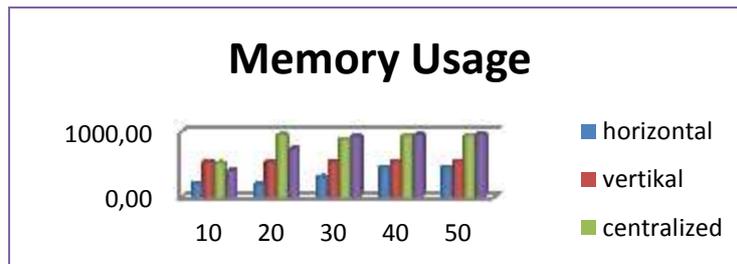


Gambar 6: Grafik throughpt view all product

Pada grafik *throughput* dapat disimpulkan bahwa semakin banyak *user* yang mengakses *request* secara bersamaan maka *throughput* yang diperoleh akan semakin kecil. Semakin cepat *response time* pada setiap *request* yang dilakukan maka nilai akan semakin besar. Pada *database* terdistribusi baik dengan fragmentasi horizontal, vertikal, maupun replikasi mengalami perubahan *throughput* yang menurun seiring dengan banyaknya *user* yang mengakses sistem, hal ini dikarenakan pengaruh jaringan pada saat pengaksesan dari sisi client ke DBaaS, serta adanya pengaruh link komunikasi antara *database* terdistribusi dari sisi oracle dan sql server. Sedangkan pada *database* terpusat perubahan *throughput* cenderung lebih stabil.

### 3.3. Analisis terhadap Memory Usage

Berdasarkan pengujian terhadap memory usage, diperoleh hasil sebagai berikut :



Gambar 7 : Grafik memory usage view all product

Penggunaan memory usage pada *database* yang terpusat juga cenderung lebih tinggi dibandingkan dengan *database* yang terdistribusi hal ini dikarenakan penggunaan memori yang digunakan untuk mengakses DBaaS. Pengaksesan data pada DBaaS yang memiliki sistem terpusat akan jauh menghabiskan memori karena memori yang digunakan terpusat pada satu *database* sedangkan pada *database* yang terdistribusi pemakaian memori akan terbagi merata diantara kedua *database*.

### 4. Kesimpulan

Berdasarkan analisis dan pengujian terhadap uji performansi DBaaS yang dibangun dengan *database* yang terdistribusi baik dengan penyimpanan data dengan fragmentasi secara horizontal, fragmentasi secara vertikal, maupun pada replikasi pada arsitektur cloud. Maka dapat diambil beberapa poin kesimpulan sebagai berikut :

- Berdasarkan hasil pengujian sesuai dengan studi kasus yang diimplementasikan *database* terdistribusi akan memiliki performansi yang lebih baik jika dilakukan proses penyimpanan data secara fragmentasi vertikal karena memiliki keunggulan mencapai  $\pm 48,2\%$  lebih dibandingkan fragmentasi horizontal, dan  $\pm 2,68\%$  dibandingkan replikasi pada skenario view all product, view product by id dan view product by category.
- Proses eksekusi pada pembagian data secara horizontal yang dilakukan pada studi kasus ini memiliki performansi *response time* dan *throughput* yang kurang bagus dikarenakan proses komputasi yang dibutuhkan untuk pengaksesan datanya jauh lebih besar karena harus melakukan proses *join* dan proses *union*.
- Penggunaan memory usage pada sistem basis data terdistribusi cenderung lebih kecil yaitu sebesar  $\pm 12,16\%$  dibandingkan dengan sistem basis data yang terpusat sebesar  $\pm 13,66\%$  hal ini dikarenakan proses query pada sistem basis data terdistribusi tidak hanya terjadi pada satu komputer saja, namun pada dua buah komputer tidak seperti *database* terpusat yang hanya menggunakan satu buah komputer.
- Berdasarkan pengujian yang dilakukan DBaaS yang dibangun berdasarkan *database* yang terdistribusi memiliki performansi yang baik jika proses fragmentasi yang dipilih sesuai dengan kebutuhan studi kasus.

Untuk pengembangan selanjutnya, dapat dilakukan pengimplementasian pada *database* yang bersifat homogen serta menggunakan jenis *database* yang berbeda misalnya network dbms, object oriented dbms, atau hierarchical dbms.

### Daftar Pustaka

#### Journal:

- [1] Hou, Zhengxioung dkk. 2010. ASAAS: Application Software as a Service for High Performance Cloud Computing.

**Textbooks:**

- [2] Brantner, Mathias and friend. Building a Database in the Cloud
- [3] Cardenas, Alfonso. 1987. Heterogeneous Distributed Database Management : The HD- DBMS
- [4] Curino, Carlo dkk. 2010. Relational Cloud: A Database-as-a-Service for the Cloud
- [5] Delaney, Kalen. 2006. Inside Mocoosoft SQL SERVER 2005 : The Storage Engine.
- [6] Gremper, Fabian. 2011. Relational Cloud: A Database-as-a-Service for the Cloud
- [7] IBM. 2011. IBM Cloud Computing Reference Architecture 2.0.
- [8] Oracle. 2011. Oracle Database Administrator Guide : Distributed Database Architecture
- [9] Silberschatz-Korth-Sudarshan. 2001. Database System Concept. Mc Graw-Hill

**Thesis/Disertation:**

- [10] Fathir, Abdul. 2011. Implementasi dan Analisis Basis Data Terdistribusi Heterogen. Institut Teknologi Telkom: Bandung
- [11] Martha Ragil, Firdaus Yanuar. ANALISA PERBANDINGAN RESPONSE TIME DAN THROUGHPUT PADA XML DAN DBMS SEBAGAI MEDIA PENYIMPANAN DATA. Intitut Teknologi Telkom : Bandung
- [12] Sudrajat, Febriyana. 2012. Implementasi dan Analisis Performansi Pada Database as a Service Dengan Menggunakan Arsitektur Relational Cloud. Institut Teknologi Telkom : Bandung