ANALISIS IMPLEMENTASI MULTI-TENANCY PADA DATABASE AS A SERVICE

Miftah Rizqi Santoso[1], Kemas Rahmat Saleh Wiharja[2], Alfian Akbar Gozali[3]

[1]Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Perkembangan teknologi informasi semakin membantu perusahaan-perusahaan dalam mengelola informasi yang dimilikinya kapanpun dan dimanapun. Dan pada masa sekarang mulai banyak perusahaan yang menyediakan layanan pengelolaan informasi yang dapat dilihat kapanpun dan dimanapun. Hal tersebut dapat dilakukan dengan adanya arsitektur cloud pada database yang merupakan arsitektur dimana sebuah database dapat diakses oleh client dari cloud service. Arsitektur cloud memiliki salah satu karakter khusus, yaitu multitenancy. Multitenancy merupakan karakteristik pada cloud dimana para penyewa layanan dapat menggunakan sumber daya komputasi secara bersama-sama. Dibalik keuntungan yang diberikan dengan adanya karakteristik multitenancy pada cloud, masih ada beberapa isu yang masih menjadi penelitian pada karakteristik tersebut. Salah satunya adalah resource isolation. Oleh karena itu, pada Tugas akhir ini penulis akan mengimplementasikan bagaimana penanganan multitenancy, khususnya aspek resource isolation, yang dibangun pada Database as a Service. Adapun studi kasus yang digunakan adalah database distro. Penelitian dilakukan pada aspek resource isolation jenis Dedicate Schema/Tabel dan Shared Schema/Tabel. Hasil akhir yang didapat dari penelitian ini berupa performansi system DBaas dari jenis resource isolation yang diujikan mengunnakan parameter response time, throughput dan error rate. Dimana pada implemetasi yang dilakukan masing-masing jenis resource isolation memiliki keunggulan masing-masing dari skenario yang diujikan. Tetapi secara garis besar jenis resource isolation Dedicate schema/Tabel memberikan hasil yang lebih baik dibandingkan jenis lain yang diujikan.

Kata Kunci : multitenancy, resource isolation, DBaaS, Performansi. Response time, throughput, error rate.

Abstract

The development of the information technology helps companies manage their information, anytime and anywhere. And at the present time, there are companies that provide information management services that can be viewed anytime, anywhere. This can be done with the database in the cloud architecture which database can be accessed by a client of the cloud service. Cloud architecture has one special character, which is multitenancy. Multitenancy is a characteristic where cloud service tenant could use computing resources together. Behind the advantage that provided by multitenancy characteristic in the cloud, there are still some issues that are still to be research on these characteristics. One of them is resource isolation. Therefore, in this final project, the writer will implement how multitenancy can be handle in the Database as a Service, especially aspects of resource isolation. The case study used is the distro database. The study was conducted on aspects of resource isolation. The types of resource isolation that used in this final project is Dedicate Schema / Table and Shared Schema / Table. The results of this final project is the performance of each type of resource isolation in the DBaas system using tested parameters response time, throughput and error rate. Where performed in implementation that each type of resource isolation has the advantage based on the scenarios tested. But overall, Dedicate schema / tables give better results than other resource isolation types.

Keywords : multitenancy, resource isolation, DBaaS, Performance. Response time, throughput, error rate.

1. Pendahuluan

1.1. Latar Belakang

Perkembangan teknologi informasi membuat sebagian besar orang semakin terbiasa dengan penggunaan perangkat lunak sebagai pengelola informasi. Informasi yang dikelola disimpan dalam sebuah *database* yang merupakan komponen penting dalam teknologi informasi sebagai penyimpan data. Untuk mengatur setiap data-data yang tersimpan dalam *database* diperlukan *Database Management System*(DBMS). DBMS merupakan komponen yang sangat penting pada semua lingkungan komputasi[1] serta dalam sebuah perusahaan yang khususnya bergerak di bidang teknologi informasi(TI). Perusahaan-perusahaan pada masa sekarang sudah banyak yang mengimplementasikan DBMS untuk pencatatan data perusahaan.

Kebutuhan akan pencatatan data perusahaan pada *database* semakin meningkat tiap tahunya. Hal itu membuat perusahaan mengalokasikan banyak *resource* baik perangkat maupun Sumber daya manusia. Perkembangan TI pun menjawab permasalahan tersebut dengan adanya paradigma *cloud computing*. Menurut *National Institute of Technology*(NIST) *cloud computing* merupakan model yang nyaman, dapat diakses dimana-mana dan *on-demand* akses melalui jaringan untuk berbagi *resource*[7]. Cloud computing dianggap dapat mengatasi permasalahan alokasi *resource* dan biaya yang besar dalam memberikan layanan TI pada perusahaan. Menurut IBM, layer *service* pada *cloud computing* terbagi menjadi *platform as a Service* (PaaS), *Infrastructure as a Service* (IaaS), *Business as a Service* (BPaaS) dan *Software as a Service*(SaaS)[5]. Pada SaaS terdapat beberapa Jenis layanan yang dapat digunakan dalam lingkungan *cloud computing* seperti *storage*, *database* dan *operating system*.

Dalam perkembanganya telah dikembangkan *Database as a Service*(DbaaS) pada layer SaaS, sebagai bentuk layanan *database* pada layer SaaS. Salah satu karakteristik kunci pada SaaS adalah *multitenancy*[12]. *Multi-tenancy* adalah desain *cloud* untuk berbagi sumber daya komputasi yang digunakan diantara penyewa yang menggunakan *cloud*[2]. *Multi-tenancy* memungkinkan memberikan *service* berdasarkan *request user* secara bersamaan dari satu atau lebih *host*. Dibalik banyak keuntungan yang dapat diambil dari karakteristik *multi-tenancy*, terdapat isu yang masih menjadi kekurangan pada karakteristik tersebut yaitu *resource isolaiton*, *security*, *customization*, *scalability*, dan performansi.Berawal dari masalah tersebut, pada tugas ini penulis tertarik untuk melakukan implementasi dan analisis multi tenancy pada *Database* as a *Service*(DBaaS).

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, dapat dirumuskan beberapa masalah diantaranya:

1. Bagaimana implementasi aspek *resource isolation Multi tenancy* pada *Database as a Service*(DBaaS).
2. Bagaimana performansi dari tiap jenis *resource isolation* yang diimplementasikan pada *Database as a Sevice*(DBaaS) berdasarkan parameter uji *response time*, *throughput* dan *error rate*.

Berdasarkan perumusan masalah diatas, pada penelitian ini, terdapat beberapa batasan masalah yang akan dilakukan, diantanranya:

1. Pada impelementasi aplikasi ini tidak menangani masalah jaringan.
2. *Database Management System(DBMS)* yang digunakan tunggal(homogen).

1.3. Tujuan

Tujuan yang ingin dicapai pada penelitian ini adalah:

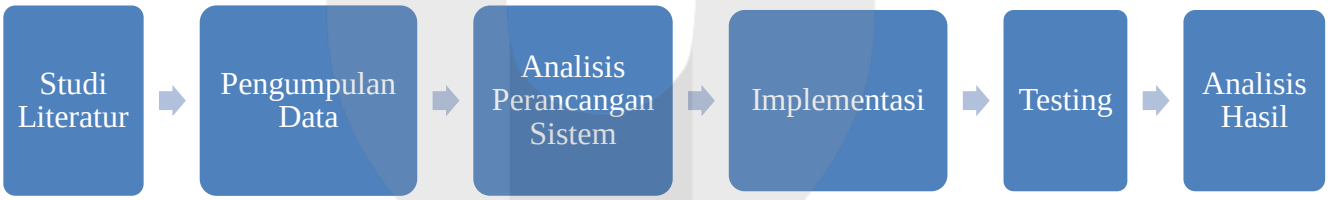
1. Mengimplementasikan aspek *resource isolation multitenancy* pada *Database as a Service*(DaaS).
2. Menganalisis performansi dari berbagai jenis *resource isolation* pada *Database as a Service*(DaaS) berdasarkan parameter uji response time, throughput dan error rate.

1.4. Hipotesa

Berdasarkan penelitian yang dilakukan Zhi Hu Wang dkk[12], jenis *resource isolation* yang mempunyai performansi terbaik adalah jenis *resource isolation dedicate tabel/schema*. Hal itu dikarenakan memiliki tampilan indeks yang lebih *simple* sehingga memberikan performansi yang baik dimana mengurangi masalah ketika jumlah penyewa ditingkatkan.

1.5. Metodologi Penyelesaian Masalah

Metodologi yang digunakan dalam memecahkan masalah diatas adalah dengan menggunakan langkah-langkah sistematis berikut:



Gambar 1-1 : Langkah-Langkah Penyelesaian Masalah

1. Studi Literatur
   Mempelajari referensi-referensi, seperti buku literatur ataupun jurnal ilmiah, terutama yang berhubungan dengan *multitenancy* pada *Database as a Service*(DBaaS).
2. Pengumpulan Data
   Pengumpulan data yang diperlukan untuk mendukurng penyelesaian penelitian ini.
3. Analisis Perancangan Sistem
   Melakukan analisis kebutuhan yang diperlukan kemudian dilakukan perancangan sistem sesuai analisis kebutuhan yang telah dilakukan.
4. Implementasi
   Membangun sistem sesuai dengan rancangan yang telah dibuat pada tahap sebelumnya.
5. Testing
   Melakukan pengujian terhadap sistem yang telah dibanguan menurut skenario yang telah dirancang

6. Analisis Hasil
   Analisis terhadap hasil pengujian yang didapat dari pengujian berdasarkan parameter uji yang telah ditentukan untuk setiap skenario pengujian.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis terhadap uji performansi *multi-tenancy* pada DBaaS, Maka dapat diambil beberapa poin kesimpulan sebagai berikut:

a. Secara garis besar peningkatan jumlah *client* dan jumlah *tenant* menimbulkan peningkatan nilai *response time*.
b. Peningkatan jumlah *client* dan jumlah *tenant* juga menimbulkan penurunan nilai *throughput*.
c. Hampir di seluruh skenario percobaan menunjukan bahwa DBaaS akan mulai mengalami *error* pada saat jumlah *tenant* enam dan jumlah *client* per *tenant* dua puluh.
d. Jenis *resource isolation dedicate database* memberikan performansi yang cukup baik ketika diberikan skenario yang kompleks seperti pada saat uji skenario untuk *service view all product*, *view all product* menggunakan *viewproduct*, dan *insert*. Dimana pada skenario tersebut jenis *resource isolation dedicate database* memberikan rata-rata nilai *throughput* tertinggi dan nilai *response time* terendah.
e. Jenis *resource isolation shared schema* memberikan performansi yang terbaik untuk skenario *update* dan *delete* dimana memberikan nilai *response time* terendah dan *throughput* tertinggi.
f. Jenis *resource isolation shared schema with index* memberikan nilai *throughput* tertinggi dan *response time* terendah untuk skenario *view product by ID* menggunakan *viewproduct*.
g. Secara umum jenis *resource isolation* yang memberikan *error rate* terkecil dihampir seluruh skenario uji yang dilakukan adalah jenis *resource isolation shared schema*.
h. Dari hasil pengujian dapat disimpulkan bahwa *multi-tenancy* dapat diimplementasikan pada *Database as a Service*(DBaaS)

5.2. Saran

Adapun saran untuk penelitian *multi-tenancy* pada DBaaS selanjutnya adalah sebagai berikut :

a. Pada penelitian ini digunakan dua jenis *resource isolation* yaitu *dedicate schema/table* dan *shared schema/table*, pada penelitian selanjutnya dapat digunakan jenis *resource isolation* lain dan lebih dari dua jenis *resource isolation*.
b. Penelitian ini hanya menganalisis *multi-tenancy* pada aspek *resource isolation*, selanjutnya dapat digunakan aspek *security*, *customization*, dan *scalability*.
c. Penelitian ini fokus pada performansi *database*, performansi dari jaringan juga dapat dijadikan fokus penelitian selanjutnya.

Daftar Pustaka

[1] Agrawal, Divyakant dkk.2011. *Database Scalability, Elasticity, and Autonomy in the Cloud*.

[2] Bobrowski, Steve. 2011. Optimal Multi*tenant* Designs for Cloud Apps.

[3] Curino, Carlo dkk. 2010. Relational Cloud: A *Database*-as-a-*Service* for the Cloud.

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

[5] IBM. 2011. IBM Cloud Computing Reference Architecture 2.0.

[6] Louvel, Jerome dkk. 2010. Restlet in Action. Manning Publications.

[7] Mell, Peter dan Timothy Grance.2011. The NIST Definition of Cloud Computing.

[8] Ray, Randy J dan Pavel Kulchenko. 2002. Programming Web Services with Perl.

[9] Rodriguez, Alex. 2008. *Restful Web services: The basics.*

[10] Sandoval, Jose. 2009. RESTful Java Web Services. Birmingham: Packt Publishing

[11] Sudrajat, Febriyana. 2012. Implementasi dan Analisis Performansi Pada *Database* as a *Service* Dengan Menggunakan Arsitektur Relational Cloud. Institut Teknologi Telkom : Bandung.

[12] Zhi Hu Wang dkk. 2008. A Study and Performance Evaluation of The Multi-*Tenant* Data Tier Design Patterns for *Service* Oriented Computing