

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 penyimpanan 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 tahunnya. 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 perkembangannya 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 isolation*, *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 Service*(DBaaS) berdasarkan parameter uji *response time*, *throughput* dan *error rate*.

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

1. Pada implemetasi 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:

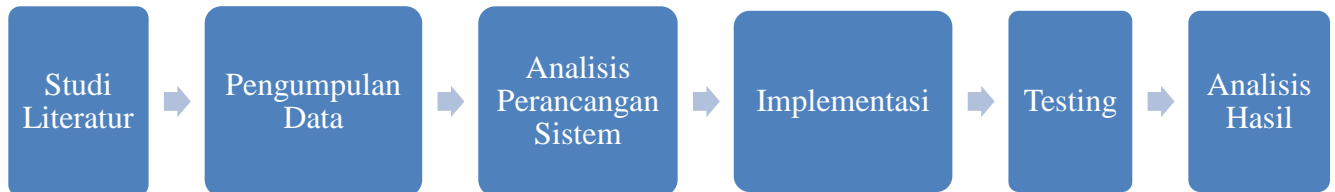
1. Mengimplementasikan aspek *resource isolation multitenancy* pada *Database as a Service(DaaS)*.
2. Menganalisis performansi dari berbagai jenis *resource isolation* pada *Database as a Service(DaaS)* berdasarkan parameter uji response time, throughput dan error rate.

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 mendukung 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 dibangun 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.