

# 1 PENDAHULUAN

## 1.1 Latar Belakang

Saat ini, besarnya lingkup komputasi dan data dalam suatu organisasi memicu timbulnya penumpukan sumber daya (*islands of computing*). Penumpukan sumber daya ini justru membawa penurunan efisiensi penggunaan sumber daya – sumber daya tersebut [13]. Sumber daya tersebut terikat secara fisik sementara penggunaannya terkadang justru terpusat ke satu komponen / *site* saja. Sehingga akan sangat sulit dan merepotkan dalam pengelolaannya, terutama saat terjadi *traffic* (penumpukan beban kerja) atau ada komponen / *site* yang berhenti beroperasi. Salah satu solusi yang diajukan untuk menangani isu ini ialah dengan *grid computing* yang di dalamnya juga mencakup *grid database*.

*Grid database* merupakan suatu bidang penelitian baru yang mengimplementasikan teknik komputasi grid dengan *database*. Dengan teknik komputasi grid, modularisasi perangkat keras [14] memungkinkan untuk skalabilitas, *information sharing*, dan alokasi sumber daya secara lebih luas serta fleksibilitas penanganan *failure* (*failover*) [6].

Node-node fisik dalam grid sendiri antara lain memiliki karakteristik terdistribusi, dan dinamis [5]. Dengan sifatnya yang dinamis, maka dalam pemrosesan *query* tidak dapat menggunakan teknik *query processing* tradisional, melainkan harus dapat menyesuaikan dengan kebutuhan-kebutuhan *query processing grid database* yang dinamis pula [2].

Lebih lanjut, karena tiap node memiliki karakteristik yang berbeda satu sama lain [13], maka biaya yang dibutuhkan tiap node untuk menyelesaikan suatu *query* akan berbeda. Sehingga, salah satu poin utama dalam *grid database* terletak pada bagaimana menentukan pemberian *query* ke node yang tepat [8] (baik dari segi waktu pengaksesan, ketersediaan dan akurasi data sekalipun saat terjadi *failure*).

Salah satu skema pemrosesan queri yang ada yaitu VO (*Virtual organizations*). VO merupakan sebuah *global metadata view*, yakni sebuah skema metadata yang diekstrak dari atribut-atribut dalam *database* di node-node grid. Sehingga *query* yang masuk ke dalam grid akan didekomposisi menjadi *query* lokal (atau disini disebut sebagai *subquery*) berdasarkan global metadata tersebut [10].

Pertanyaannya adalah apakah VO ini mampu menjawab tantangan dalam *grid database* itu sendiri yakni dalam hal *failover* dan *availability* saat *downtime* sekalipun. Seberapa cepat skema ini dalam menyelesaikan *subquery* yang masuk ke node-node grid? Apakah skema ini masih mampu menyelesaikan suatu *subquery* ketika terjadi node mati? Jika ya, apakah akurasi datanya masih terjaga?

Hal inilah yang selanjutnya akan menjadi kajian penulis dalam tugas akhir ini. Harapannya VO dapat menjadi solusi untuk diimplementasikan dalam *subquery processing* di *grid database*, mengingat *subquery* sendiri merupakan salah satu teknik penting dalam pemrosesan data khususnya dalam *grid database* yang pada umumnya menggunakan *query* kompleks.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah yang menjadi fokus penulis di sini adalah:

1. Apakah VO dapat menangani masalah *failover node* dengan tetap menjaga *availability data* (termasuk akurasinya) dalam pemrosesan *subquery* di *grid database*?
2. Seberapa toleran VO terhadap *failure node* termasuk bagaimana kinerjanya di *grid database* pada kondisi tersebut?

## 1.3 Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini, meliputi :

1. Skema VO dapat diimplementasikan pada *grid database*.
2. VO menjadi solusi pemrosesan subquery di *grid database* yang dapat menjaga data *availability data* untuk setiap node yang tergabung di *grid infrastruktur* sehingga layanan akan selalu tersedia bagi semua unsur yang berkolaborasi di dalamnya.

## 1.4 Batasan Masalah

Dalam implementasi tugas akhir ini dibatasi oleh beberapa hal, antara lain :

1. Skema pemrosesan query dalam *grid database* menggunakan skema VO dengan *vertical partitioning*.
2. Studi kasus yang digunakan yakni seputar proses bisnis pada *Endangered Species Conservation (ESC)*.
3. Data yang digunakan merupakan data *dummy*, hasil *generate* otomatis dengan menggunakan tool pembantu, Data Generator.
4. Jenis query yang dipakai yakni query *selection* dengan subquery *centralized node subquery* dan *distributed node subquery*.
5. Analisis yang dilakukan yakni terhadap kinerja waktu pemrosesan (*response time*), *availability* (termasuk akurasinya) pada kondisi normal (tanpa *failure*) dan terjadi *failure*.
6. Uji *failure* bermain di sisi *grid storage* (ketersediaan *resource* di sisi *grid storage* masih lebih banyak dibanding di sisi *grid komputasi*).
7. Skenario test *availability* dilakukan dengan meng-online offline-kan node-node *virtual storage* (ASM Disk) yang diasumsikan sebagai *physical disk storage*.
8. DG (*Disk Group*) merupakan sekumpulan ASM Disk (AD), dimana 1 AD = 1 LUN = 1 VD (*Volume Disk*).

## 1.5 Metodologi Penyelesaian Masalah

Penelitian ini akan menunjukkan bahwa VO dapat diterapkan untuk pemrosesan *subquery* pada *grid database*. Dengan parameter kajian dari sisi *availability* data dan waktu pemrosesan *subquery*.

Adapun alur metodologi penyelesaian masalah yang akan digunakan yaitu :

- a. Studi Literatur :  
Mempelajari literatur-literatur yang ada sesuai dengan permasalahan meliputi konsep dari *subquery processing*, *Virtual organization (VO)*, dan *grid database*.
- b. Perancangan sistem :  
Membuat desain skema *environment grid database* yang akan dibangun, serta skenario pengujian.
- c. Implementasi :  
Membangun infrastruktur *grid environment* berikut *database grid* dengan DBMS Oracle.
- d. Testing dan analisis hasil, meliputi :  
Melakukan pengujian kinerja terhadap sistem yang telah dibangun menurut skenario yang dirancang sebelumnya.
- e. Analisa hasil :  
Analisa terhadap hasil pengujian yang didapat, dengan melakukan analisa dari segi waktu pemrosesan (*execution time*) serta ketersediaan data saat node mati (*failover node*) untuk setiap skenario pengujian.
- f. Pembuatan laporan :  
Pengambilan kesimpulan dari hasil analisa yang didapat untuk selanjutnya dituangkan dalam laporan hasil pengerjaan Tugas Akhir.

## 1.6 Sistematika Penulisan

Tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

- BAB I PENDAHULUAN**  
Berisi pemaparan mengenai latar belakang permasalahan, tujuan yang ingin dicapai melalui tugas akhir kali ini, perumusan masalah, batasan masalah, metodologi tugas akhir, dan sistematika penulisan.
- BAB II LANDASAN TEORI**  
Berisi uraian seputar landasan teori yang digunakan, meliputi teori tentang *Grid database*, *Skema Virtual organization* dan teori-teori lain yang berkaitan dengan penelitian tugas akhir ini.
- BAB III ANALISIS PERANCANGAN SISTEM**  
Berisi perancangan sistem dan infrastruktur *Grid database Environment* yang akan dibangun serta skenario pengujian yang dilakukan.
- BAB IV ANALISIS IMPLEMENTASI DAN PENGUJIAN SISTEM**  
Berisi implementasi dari skenario yang telah dibuat, termasuk analisa hasil pengujiannya.
- BAB V PENUTUP**  
Berisi kesimpulan dan saran-saran untuk pengembangan lebih lanjut terhadap hasil penelitian tugas akhir ini.