

ANALISIS DAN IMPLEMENTASI DATABASE TUNING UNTUK SISTEM PENDETEKSIAN KOROSI PADA PIPA MINYAK (STUDI KASUS : CNOOC SES LTD.)

Canrakerta¹, Hetti Hidayati², Purba Daru Kusuma³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Berkembangnya kebutuhan aplikasi sebagai salah satu support bisnis suatu perusahaan berbanding lurus dengan kebutuhan resource. Semakin canggih sebuah aplikasi, maka akan semakin besar resource yang akan dibutuhkan. Salah satu aplikasi yang membutuhkan resource besar adalah aplikasi Pipeline Integrity Management Status sebagai salah satu sistem untuk mendeteksi korosi pada pipa minyak. Salah satu resource yang dibutuhkan dalam aplikasi tersebut adalah database sebagai tempat penyimpanan data-data statistik yang akan digunakan sebagai alat perhitungan korosi suatu pipa minyak. Besarnya resource data yang akan digunakan tersebut mengakibatkan response time yang lama pada suatu query. Khususnya query yang digunakan untuk menampilkan nilai perhitungan. Oleh karena itu perlu adanya suatu analisis yang dapat memperbaiki performansi dari suatu database. Salah satu yang bisa dilakukan untuk memperbaiki permasalahan tersebut adalah melalui skema tuning pada suatu database. Tuning database sendiri meliputi perbaikan skema data, penambahan indexing sebagai pencarian data agar lebih cepat, serta memperbaiki query yang digunakan. Selibuhnya adalah hal practical yang bisa kita temukan dari kondisi suatu database. Karena masing-masing database memiliki cara tuning yang berbeda tergantung pada kondisi penggunaannya. Output dari hasil tuning ini adalah rekomendasi-rekomendasi performansi yang akan diajukan ke pihak perusahaan terkait tempat aplikasi ini dikembangkan. Adapun parameter kesuksesan penelitian ini adalah besaran nilai response time setelah dilakukan tuning yang sesuai dengan harapan client. Perhitungan yang akan dianalisa adalah perhitungan Risk Assessment Level 1 dan Risk Assessment Level 2.

Kata Kunci : database, tuning, query, resource, korosi pipa minyak, response time

Abstract

The requirement of application development as one of business support system is directly proportional to resource needs. Increasingly sophisticated application, it requires a great resource. One of application that require big resource is Pipeline Integrity Management Status Application as one of sistem for corrosion detection in oil pipeline. One of resource who require by that application is database as a place to store statistic data which will be used for calculation corrosion tool in oil pipeline. Big resource because of the long response time in a query. Especially, query which used for showing calculation value. Therefore, we need analysis to solve performance problem in database. One of we can doing to solve this problem is tuning the database sistem. Database tuning coverage repair data schema, adding index for searching data, and repair the query which used. In this case, we will encounter many practical problem depend on the database. Because, each database have different problem. Output from tuning is all of recommendation performance which will be used by company. Parameter for successful for this research is the value of response time after tuning that suited with client expectation. The calculation which will be analyzed are Risk Assessment Level 1 and Risk Assessment Level 2.

Keywords : database, tuning, query, resource, oil pipeline corrosion, response time

BAB 1

Pendahuluan

1.1 Latar Belakang Masalah

Perkembangan teknologi yang semakin cepat sangat membantu para *developer* untuk mengembangkan sebuah sistem cerdas. Salah satunya adalah pengembangan sistem untuk mendeteksi sebuah kebocoran pada pipa minyak. Dengan memanfaatkan perkembangan teknologi yang ada, *developer* dapat mengembangkan sistem tersebut dengan menggunakan RDBMS, MapServer, serta *framework-framework* pengembangan aplikasi yang telah banyak tersedia.

Salah satu perusahaan yang mengembangkan sistem untuk mendeteksi sebuah kebocoran pada pipa minyak adalah CNOOC. CNOOC sendiri merupakan salah satu perusahaan multinasional yang memiliki *core business* dibidang perminyakan. Sistem yang akan dibuat CNOOC nantinya, akan diaplikasikan pada 131 pipa minyak yang ada di daerah kepulauan seribu. Bahkan, apabila *project* ini dinyatakan sukses, maka sistem ini akan dibawa ke Cina, negara asal perusahaan CNOOC untuk diuji cobakan pada seluruh kilang minyak yang dimiliki perusahaan tersebut.

Sistem pendeteksian korosi pada pipa minyak ini memiliki cara kerja yang sederhana tetapi memiliki nilai komputasi yang besar. Sistem tersebut dapat menghitung berbagai kemungkinan yang terjadi selama pengiriman minyak dari satu pipa ke pipa lainnya untuk mendeteksi kemungkinan waktu korosi pada sebuah pipa. Dalam kasus ini, dibutuhkan komputasi data yang sangat besar, karena perhitungannya dilakukan di dalam *database*. Sehingga, sebuah *query plan* yang digunakan dalam melakukan *retrieve* data memiliki *response time* yang cukup lama. Dapat dikatakan pula bahwa lingkungan komputasi yang sudah modern memiliki tingkat karakteristik yang lebih tinggi^[1].

Lewat *tuning database* inilah diharapkan permasalahan komputasi pada *database* sistem pendeteksian korosi pada pipa minyak dapat terselesaikan. Sehingga secara performansi terutama kecepatan akses data bisa lebih baik. Langkah yang dilakukan untuk mencapai hal tersebut dalam penelitian adalah dengan melakukan studi literatur keseluruhan sistem pada aplikasi tersebut, deteksi permasalahannya, diagnosa langkah yang harus dilakukan, dan yang terakhir adalah *resolve* permasalahannya^[2].

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan di atas, terdapat beberapa pembahasan yang akan diteliti dalam penelitian kali ini. Diantaranya yaitu :

1. Apakah desain database yang selama ini *existing* di dalam sistem merupakan desain database yang baik?
2. Bagaimana query yang ada pada sistem yang *existing* bila dibandingkan dengan query baru yang didapatkan dari hasil query evaluation.
3. Bagaimana kemampuan DBMS yang *existing* pada sistem secara *response time*, kemudian akan dibandingkan hasilnya dengan DBMS lainnya.
4. Seberapa besar table scan yang terjadi pada query yang digunakan.

Batasan masalah yang akan dibatasi dalam penelitian tugas akhir ini adalah :

1. Penelitian terfokus pada *tuning* performansi *database*
2. *Query* yang diujikan sebanyak dua *query*, yaitu untuk *query risk assessment level 1* dan *risk assessment level 2*
3. DBMS yang digunakan selama penelitian adalah MySQL, Microsoft SQL, dan PostgreSQL
4. Sistem operasi yang digunakan selama penelitian adalah Windows Server 2008

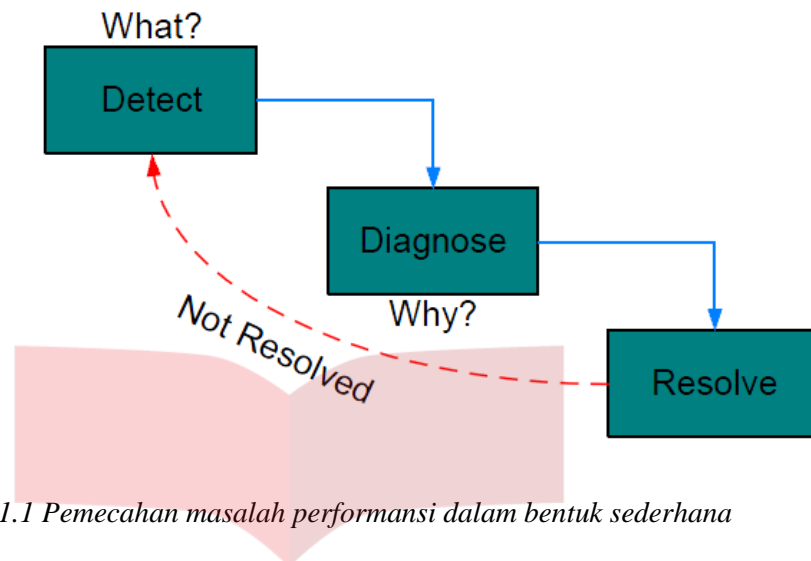
1.3 Tujuan

Tujuan yang akan dicapai dalam penelitian ini adalah :

1. Menghasilkan desain *database* yang lebih baik
2. Menemukan *query-query* yang optimal untuk setiap eksekusi
3. Penentuan DBMS yang digunakan
4. Menemukan perbandingan table scan yang terjadi pada setiap perubahan proses tuning
5. Rekomendasi perancangan desain sistem secara keseluruhan bagi perusahaan terkait

1.4 Metodologi Penyelesaian Masalah

Secara sederhana, dalam menyelesaikan isu-isu tentang performansi, metode penyelesaian dapat dilakukan dengan cara deteksi permasalahan, diagnose, lalu selesaikan permasalahan tersebut^[2].



Gambar 1.1 Pemecahan masalah performansi dalam bentuk sederhana

Dalam penelitian kali ini, ada empat tahapan atau proses yang akan dilakukan, yaitu studi literatur, evaluasi sistem, proses tuning, kemudian yang terakhir adalah *execute baseline or benchmark test*. Secara garis besar, yang dilakukan dalam proses tersebut akan dipaparkan di bawah ini.

1. Studi literatur

Mempelajari literatur-literatur yang berkaitan dengan permasalahan yang ada, seperti konsep tuning serta sistem pendeteksian korosi itu sendiri.

2. Evaluasi sistem

Pada tahap evaluasi pada sistem, penelitian mencoba untuk mengamati bagaimana sistem berjalan untuk mengambil banyak *sample* dari permasalahan yang ada pada sistem pendeteksian korosi pada pipa minyak tersebut. *Sample-sample* yang didapat nantinya akan diperbaiki pada tahap *tuning database*.

3. *Tuning database*

Tahap *tuning database* merupakan tahapan dilakukannya perbaikan performansi pada *database*. Pelaksanaan tuning sendiri mengacu pada nilai-nilai *error* yang didapat setelah tahapan evaluasi sistem dilakukan. Output dari hasil tuning database sendiri adalah perbaikan yang telah dilakukan agar bisa memperbaiki performansi sistem.

4. *Execute baseline or benchmark test*

Tahap *execute baseline or benchmark test* ini adalah posisi saat sistem telah dituning. Kegiatan yang dilakukan pada tahap ini merupakan kegiatan *testing* sistem. Peneliti melakukan *testing* terhadap tahapan yang sudah diperbaiki dari tahap sebelumnya. Bila hasil dari tahap ini memiliki performansi yang lebih baik, maka penelitian bisa dikatakan berhasil, akan tetapi bila ternyata performansi masih di bawah atau bahkan sama dengan hasil sebelumnya, maka akan dilakukan tahap *tuning* kembali dengan proses yang hampir sama pada tahap evaluasi sistem. Hanya saja pada tahap *tuning* ini, peneliti lebih fokus pada mencari sesuatu yang bisa dilakukan *tuning* kembali.

1.5 Jadwal Kegiatan

Table 1-1 : Jadwal Kegiatan Penelitian

No.	Kegiatan	Bulan 1	Bulan 2	Bulan 3	Bulan 4	Bulan 5
1	Studi Literatur					
2	Evaluasi sistem					
3	Execute baseline or Benchmark test					
4	Tune <i>database</i>					
5	Penyusunan laporan tugas akhir					



Bab 5

Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil analisis yang didapat, dapat disimpulkan bahwa :

1. Skema database *existing* yang digunakan selama ini perlu diperbaiki. Perbaikan skema database secara relasi dan normalisasi berpengaruh pada hasil *response time*. Hal tersebut dikarenakan perbedaan yang terjadi pada pengalamanan data pada block-block yang disediakan. Semakin sedikitnya block yang digunakan maka akan mempengaruhi banyaknya *table scan* yang terjadi. Sehingga nilai *response time* dapat lebih cepat.
2. Kesalahan penggunaan view yang dapat menyebabkan nilai *response time* yang lebih lambat. Penggunaan view yang berjenjang untuk melakukan suatu perhitungan butuh waktu atau proses yang lebih lama. Bila satu view harus melakukan perhitungan ke dalam 131 pipa minyak, maka perhitungan *table scan* bisa mencapai 131xjenjang view. Sehingga dalam *query evaluation* ditetapkan penggunaan function sebagai pengganti view yang *existing* karena function memiliki proses yang lebih cepat ketimbang view.
3. *Query existing* yang digunakan pada sistem masih bisa diperbaiki lagi berdasarkan hasil *evaluation plan*. Dengan query yang baru, nilai *response time*, *block*, serta *table scan* memiliki hasil yang lebih baik seperti yang tertulis pada pengujian.
4. Perbandingan hasil *response time* pada PostgreSQL memiliki nilai *response time* yang lebih cepat secara signifikan untuk Risk Assessment Level 1 ketimbang DBMS lainnya, yaitu 0.177 second. Hal tersebut dikarenakan PostgreSQL memiliki *scope management attribute* yang lebih baik. Misalnya saja, PostgreSQL membatasi tipe data int ke dalam tiga jenis, yaitu int2, int4, dan int8. Hal tersebut berpengaruh pada komputasi yang terjadi pada suatu *query select* yang berkaitan dengan *data stored*. Sementara, untuk Risk Assessment Level 2, didapatkan bahwa masing-masing DBMS memiliki nilai yang tidak terlalu jauh berbeda, nilai *response time* berada di angka 0.018 second – 0.025 second.
5. Dalam menentukan index pada database, tidak bisa dilakukan sembarang. Hal tersebut dikarenakan *over indexing* justru dapat membuat proses pencarian data lebih lama. Pemanfaatan *index* tidak perlu dilakukan dengan memberikan semua *index* pada atribut yang digunakan, akan tetapi cukup pada beberapa atribut dalam satu tabel yang sering digunakan dalam query.

6. Dengan spesifikasi komputer yang sama, sistem masih bisa dilakukan *tuning* untuk mendapatkan nilai *response time* yang lebih baik dari yang sebelumnya dengan nilai yang diinginkan *client*.

5.2 Saran

Kebutuhan *resource* yang besar pada *database* tidak selalu harus diikuti perangkat yang lebih canggih. Sehingga *database tuning* sangat mampu dikembangkan dalam kondisi permasalahan tersebut sebagai bahan *research*. Sesuai dengan tujuan dari penelitian kali ini, maka hasil penelitian di atas akan digunakan sebagai saran terhadap perusahaan CNOOC SES Ltd. Selaku perusahaan terkait untuk menggunakan DBMS MySQL dengan perbaikan diberbagai sisi yang telah dilakukan *tuning*.



Daftar Pustaka

- [1] Kon, Fabio., H. Campbell, Roy., Mickunas, M. Dennis., Nahrstedt, Klara., and J.Ballesteros, Francisco. *2K: A Distributed Operating Systems for Dynamic Heterogeneous Environments*. *IEEE conference*, 201-208, 6 August 2002
- [2] Barber, Scott. . 2006, *A Performance Engineering Strategy*, PerfTestPlus Inc.
- [3] Stuhler, Julian. *A perspective on Database Performance Tuning*. <http://www.databasejournal.com/features/db2/article.php/3876521/A-Perspective-on-Database-Performance-Tuning.htm> (diakses tanggal 4 Desember 2011)
- [4] Mohamed, Azeem. *A Successful Performance Tuning Methodology*, California, Quest Software, Inc.
- [5] Ramakrishnan, Raghu., Gehrke, Johannes. 2000, *Database Management Systems*, United States of America, McGraw-Hill Companies, Inc.
- [6] Shasha, Dennis., Bonnet, Philippe. *Database Tuning : Principles, Experiments, And Troubleshooting Techniques*, Morgan Kaufmann Publisher
- [7] Chan, Immanuel. July 2008, *Oracle Database Performance Tuning Guide, 11g release 1 (11.1)*, Oracle Database
- [8] Dumke, Reiner., Rautenstrauch, Claus., Schmietendorf, Andreas., Scholz, Andre. *Performance Engineering : State of the Art and Current Trends*, Springer.
- [9] Teradata. December 2003, *Performance Optimazion*, United State of America, NCR Corporation
- [10] Pipes Jay. *Performance Tuning Best Practices, Community Relations Manager*, North America.
- [11] Escape, A. Keith. *Piping and Pipelines Assessment Guide*, Burlington, Gulf Proffesional Publishing.
- [12] Fraiteur, Gael. *SQL Tuning Tutorial – Understanding a Database Execution Plan*. <http://www.codeproject.com/Articles/9990/SQL-Tuning-Tutorial-Understanding-a-Database-Execu> (diakses tanggal 3 Juli 2012)

Telkom
University