

BAB 1. PENDAHULUAN

1.1 Latar belakang

Dengan meningkatnya perkembangan teknologi informasi yang semakin cepat saat ini, maka banyak sekali aplikasi – aplikasi yang berbasis *online* bermunculan, dan sangat membutuhkan sebuah sistem database yang menjadi pondasi dari sebuah aplikasi tersebut. Sehingga aplikasi yang berkembang secara cepat tentu memiliki sistem *database* yang sangat besar. *Database* adalah sekumpulan *data store* (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya (Al-Bahra, 2013). *Database* biasanya disimpan pada mesin server – server yang bekerja secara terus menerus tanpa henti, dan terkadang server tersebut mengalami sistem *crash / hang*, sehingga data mengalami *corrupt*.

Pada situasi *crash* tersebut, sangat diperlukan suatu teknik yang dapat melakukan proses *recovery database* di bawah waktu 1 jam untuk dapat mengembalikan *database server* tersebut pada posisi *online* kembali untuk menerima dan mengirim request *query*. Kendala yang sering dihadapi adalah jika *database* yang dimiliki sangat besar dan tidak memiliki *backup* pada saat hari / waktu terjadinya *crash* tersebut. Kasus ini terjadi pada server Portal Pendidikan Indonesia, Portal Pendidikan Indonesia adalah suatu sistem manajemen ijazah secara online yang sedang di kembangkan oleh Kementerian Pendidikan dan Kebudayaan dan telah di lakukan uji coba terhadap sekolah SMA / SMK di daerah Tangerang. Sistem *database* pada Portal Pendidikan Indonesia pernah terjadi *crash* akibat terjadinya kegagalan pada sistem fisik yakni berupa rusaknya komponen pendingin pada *motherboard*, dan sistem tiba - tiba melakukan shutdown secara paksa, pada saat insiden terjadi mekanisme backup belum berjalan ketika terjadinya *crash*. Maka data yang diharapkan hanya terdapat pada sistem yang sedang berjalan saat itu. Dan tidak adanya teknisi khusus yang dapat menghandle *database server* yang *corrupt* tersebut. Teknologi yang diperlukan dalam kasus ini adalah, suatu perintah *command line interface* yang dapat mengambil kembali data tersebut tanpa

merusak data yang ada. Dengan diasumsikan bahwa hanya sistem *database* saja yang rusak, namun data – data aplikasi portal pendidikan masih ada, sehingga diperlukan teknik pengambilan data yang cepat, aman dan efisien.

Salah satu teknik *Database Recovery* yakni pada jurnal (Patel, 2015) Metode yang digunakan adalah sistem log, yakni mereview ulang log terakhir terjadinya *error* dan melakukan pengembalian data satu per satu untuk table yang mengalami *crash* / rusak. Waktu yang dibutuhkan untuk tahap ini memiliki durasi lebih dari 1 jam, karena harus mereview ulang setiap liog dan transaksi *database* sebelum terjadinya *crash*. Apabila data yang dimiliki sangat besar tentu menjadi kendala dan dibutuhkan lebih dari 1 orang teknisi khusus untuk melakukan penanganan terhadap *recovery database* tersebut sehingga ini menjadikan kelahmahannya. Selain itu, desain *recovery database* dapat menggunakan basis *cloud service*, pada jurnal (Challagidad, Dalawai, & Birje, 2017), dapat menggunakan *cloud service* yang disediakan oleh *Google Cloud Computing* ataupun *Amazon Web Service*. Penyedia *Cloud Computing* memiliki beberapa banyak pilihan, seperti *failover*, maupun penyimpanan data terdistribusi. Pengguna, dapat membuat beberapa pengaturan sesuai kebutuhannya untuk penyimpanan *database* sebagai *backup*. Sehingga sangat dimungkinkan untuk melakukan proses *recovery* secara *cloud*, tanpa diperlukan teknisi yang melakukan proses *recovery on site* atau hadir dilokasi untuk mengambil data .

Proses pengamanan replikasi *database* juga diperlukan, yang dijelaskan pada jurnal (Herman, 2014) menggunakan metode *SSL* pada proses replikasi *database*, proses pengamanan juga perlu dilakukan agar tercipta kemanan tanpa terjadinya penyadapan duplikasi *database* antar jaringan. Pengamanan pada sistem *database* juga diperlukan untuk menjaga kerahasiaan dari data yang diperlukan, sehingga dapat terhindar dari penyadapan. Jenis – jenis penyebab data *loss* terdapat beberapa hal, jurnal yang ditulis oleh (Gharat, 2015) menjelaskan hal tersebut, beberapa penyebab terjadinya *loss data* yang menempati urutan pertama adalah terjadi bencana yang disebabkan oleh alam atau biasa disebut *natural disaster*, hal ini menyebabkan terjadinya *loss data* sebesar dua persen, dan jenis bencana ini biasanya bervariasi, mengakibatkan terjadinya kehilangan data. Yang kedua adalah,

terjadinya kerusakan pada aplikasi yang menyebabkan terjadinya rusak bahkan hilangnya data. Jenis penyebab data loss yang ketiga adalah gagalnya pada sistem jaringan untuk berfungsi secara normal, pada sistem *cloud* yang sangat bergantung pada sistem jaringan antar sistem, apabila terjadi *network failure* atau kegagalan pada sistem jaringan, dapat menyebabkan terjadinya *crash* pada sistem jaringan *cloud*, dan hal ini dapat juga menyebabkan terjadinya *loss* pada data. Dan yang terakhir adalah adanya serangan pada jaringan, jenis serangan ini diakibatkan oleh pihak lain yang memiliki tujuan untuk menyerang suatu sistem *database*, yang dapat mengakibatkan sistem *database* gagal berfungsi dengan baik, bahkan dapat mengakibatkan sistem *database* mengalami *crash* dan hilangnya beberapa data.

Untuk solusi paling baik adalah dengan melakukan *recovery database* dengan menggunakan sistem based on directory, yakni melakukan backup keseluruhan pada sistem database dengan memindahkannya di temporary folder, lalu melakukan *reinstall* ulang terhadap sistem mysql, dan mengganti file *ibdata1* yang baru dengan *ibdata1* dari temporary folder. Lalu, dilanjutkan dengan memindahkan keseluruhan file database dari temporary folder menuju sistem mysql yang baru. Langkah ini lebih efektif, karena data kembali pulih hingga 100 % serta dapat menerima query dan mengirim query yang dibutuhkan setelah sistem kembali normal.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, dapat dirumuskan beberapa pokok permasalahan sebagai berikut :

- a. Bagaimana tahap awal dalam proses penyelamatan database ?
- b. Bagaimana tahap penyelamatan database menggunakan CLI pada Linux ?
- c. Bagaimana dampak terhadap kinerja database setelah dilakukan recovery dengan menggunakan metode yang ditawarkan ?
- d. Bagaimana sistem monitoring kesehatan database dapat berperan mengetahui kerusakan yang terjadi pada database ?

1.3 Batasan Masalah

Pada proses recovery database ini, dibatasi beberapa batasan masalah sebagai berikut :

- a. Proses *recovery* telah digunakan pada sistem database Portal Pendidikan Indonesia
- b. Proses *recovery* hanya mengembalikan data yang *crash* agar dapat berjalan secara normal kembali.
- c. Monitoring Sistem dapat mengenali error log tertentu saat terjadinya *crash*.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah dapat melakukan *recovery* terhadap database sistem yang mengalami *crash* atau gagal fungsi yang dikarenakan faktor hardware.

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah dapat membantu para teknisi database untuk mempercepat proses penanganan terhadap database yang mengalami *corrupt*.