Implementasi Sistem Database Terdistribusi Dengan Metode Multi-Master Database Replication

Ahmad Heryanto^{1*}, Albert²

¹Fakultas Ilmu Komputer, Jurusan Sistem Komputer, Universitas Sriwijaya, Palembang, Indonesia

²Fakultas Ilmu Komputer, Jurusan Teknik Komputer, Universitas Sriwijaya, Palembang, Indonesia

Email: ^{1*}hery@unsri.ac.id, ²albert1992@hotmail.com

Abstrak

Database merupakan kebutuhan utama pada setiap aplikasi komputer untuk menyimpan, mengolah dan memodifikasi data. Salah satu permasalahan penting yang dihadapi dalam database adalah tersedianya infrastruktur teknologi informasi yang memadai dalam mengelola dan melakukan pengamanan terhadap data yang terdapat pada database. Data yang tersimpan pada database harus memiliki proteksi terhadap ancaman dan gangguan. Ancaman dan gangguan dapat diakibatkan dari berbagai macam hal, seperti maintenance, kerusakan data dan bencana alam. Untuk mengantisipasi kehilangan dan kerusakan data perlu dilakukan replikasi terhadap sistem database. Mekanisme replikasi yang digunakan oleh peneliti adalah multi-master replication. Teknik replikasi mampu membentuk cluster database dengan waktu replikasi kurang dari 0.2 detik.

Kata kunci: database, multi-master, cluster, replikasi, informasi

Abstract

Databases are the main need for every computer application to store, process and modify data. One important problem faced in databases is the availability of adequate information technology infrastructure in managing and securing data contained in the database. Data stored on the database must have protection against threats and disturbances. Threats and disruptions can result from a variety of things, such as maintenance, data damage, and natural disasters. To anticipate data loss and damage, replication of the database system needs to be done. The replication mechanism used by researchers is multi-master replication. The replication technique is able to form a database cluster with replication time of fewer than 0.2 seconds.

Keywords: database, multi-master, cluster, replication, information

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi semakin hari semakin berkembang pesat. Perkembangan pesat teknologi telah memberi pengaruh besar bagi segala aspek kehidupan, banyak sekali manfaat yang dihasilkan oleh penggunaan teknologi. Teknologi saat ini telah memberi banyak kemudahan dalam penyampaian suatu informasi. Informasi disimpan, diolah dan dihapus menjadi bagian utama dari ketersediaan informasi. Salah satu permasalahan penting yang dihadapi sebuah perusahaan atau instansi yang berskala besar maupun kecil adalah tersedianya infrastruktur teknologi informasi yang memadai dalam mengelola dan melakukan pengamanan terhadap ribuan bahkan jutaan data penting yang dimiliki perusahaan. Pengelolaan data oleh suatu instansi atau perusahaan biasanya menggunakan suatu sistem database.

Kebutuhan database yang semakin meningkat menjadikan data sebagai aset yang bernilai tinggi. Penyimpanan data dalam jumlah besar akan meningkatkan kemungkinan data akan mengalami kerusakan, data tidak dapat diakses atau bahkan kemungkinan terburuk yang dapat terjadi adalah data tersebut akan hilang (loss). Gangguan tersebut bisa diakibatkan dari berbagai macam hal, seperti gangguan maintenance, kerusakan database, kerusakan media serta data corruption dan bencana yang tak terduga (bencana alam).

Tentu hilangnya data yang dianggap penting dapat menyebabkan kerugian bagi pemilik, untuk mengatasi hal ini biasanya digunakan cara manual yaitu dengan cara melakukan backup data pada media penyimpanan lain, namun hal itu masih dianggap belum efektif karena data tidak dapat ter-update secara langsung pada media penyimpanan cadangan[1]. Sistem backup tersebut akan memiliki permasalahan ketika ada perubahan data yang dilakukan pada media penyimpanan utama. Data yang telah dirubah tersebut tidak secara otomatis mengubah data yang ada pada media penyimpanan cadangan. Oleh karena itu, administrator harus melakukan update secara manual pada data data yang mengalami perubahan.

Ketersediaan data yang baik secara realtime terkadang menjadi kebutuhan yang rumit pada masalah tertentu. Masalah tersebut berkaitan dalam hal kinerja sistem, jumlah pengaksesan data, serta lalu lintas jaringan data yang dalam perkembangannya semakin bertambah padat. Hal tersebut memicu timbulnya teknologi replikasi data pada masa ini.

Proses replikasi data adalah sebuah proses yang sering dilakukan pada sistem database terdistribusi. Sistem database terdistribusi merupakan kumpulan dari node-node komputer. Setiap node tersebut dapat berpartisipasi dalam proses transaksi yang melakukan pengaksesan data pada suatu site atau beberapa site. Terdapat beberapa hal yang dapat dijadikan alasan dalam membangun database terdistribusi, seperti pemakaian bersama (share), kehandalan (reliability), ketersediaan (availability) dan kecepatan pemrosesan query[2]. Keuntungan utama yang dapat kita peroleh dari database terdistribusi adalah kemampuan untuk pemakaian dan pengaksesan data secara bersamaan dengan handal serta efisien[3]. Replikasi berguna dalam hal peningkatan kinerja sistem serta dapat melindungi ketersediaan data yang akurat.

2. TEORITIS

2.1. Database

Database adalah suatu sistem hardware dan software yang di gunakan dalam menyimpan, menghapus, mengolah dan mengalirkan data. Data dapat menjadi sebuah informasi dengan pengelolaan dari suatu program komputer. Perangkat lunak yang dipakai untuk mengelolah serta memangil database dikenal dengan *database management system* (DBMS). Pada DBMS digunakan *query* untuk berinteraksi dengan data yang tersimpan di dalam database. Pada database, data disimpan dengan cara terstruktur. Struktur data memiliki skema khusus dalam menyimpan data. Skema menggambarkan objek yang diwakili suatu database, serta hubungan diantara objek tersebut. Teknik yang digunakan adalah menggunakan model data relasional. Model data relasional membuat semua informasi yang tersimpan di dalam data membentuk bentuk tabel-tabel yang saling berhubungan.

2.2. Replikasi

Replikasi adalah teknik penggandaan data pada beberapa lokasi fisik yang berbeda untuk satu data *logic* yang sama. Kegiatan ini bertujuan apabila terjadi kerusakan atau kegagalan pada satu lokasi fisik tidak akan mempengaruhi kinerja seluruh sistem. Replikasi *database* juga dapat digunakan dalam hal meningkatkan availabilitas sistem database dengan cara membagi beban pekerjaan pada tiap *database* server[4-5].

Jenis-jenis replikasi yang digunakan pada database adalah sebagai berikut:

1. Replikasi *Synchronous*
 - a. Proses replikasi terjadi secara real time
 - b. Sinkronisasi data menyediakan *recovery* data yang konsisten
 - c. Proses penulisan pada *master* dan *slave* harus selesai terlebih dahulu sebelum beralih ke transaksi berikutnya.
2. Replikasi *Asynchronous*
 - a. Proses replikasi tidak berjalan secara realtime. Data akan diletakan dalam sebuah buffer terlebih dahulu setelah itu dalam jangka waktu tertentu akan di replikasi ke *slave*.
 - b. Apabila terjadi *crash* pada salah satu *node* saat replikasi belum selesai, data hasil replikasi tidak dapat dipastikan telah identik.
 - c. Setelah transaksi di master selesai barulah proses replikasi berlangsung.

Replikasi database dapat dilakukan dengan tiga cara yang berbeda, antara lain;

1. *Snapshot Replication* : Data pada satu server database disalin keserver database lainnya.
2. *Merging Replication* : Data dari dua atau lebih database digabungkan menjadi satu *database*.
3. *Transactional Replication* : Server replikasi mendapatkan salinan awal database yang lengkap dan kemudian mendapatkan pembaruan berkala seiring perubahan data.

2.3. MYSQL Galera Cluster

Galera Cluster adalah replikasi *synchronous* multi-master yang menggunakan format data InnoDB. Pada replikasi ini, sebuah aplikasi tidak perlu tahu server mana yang bisa mereka tulis (Master) dan server mana yang mereka bisa baca datanya (Slave). Semua node dalam kluster galera memiliki kedudukan yang sama.

MULTI-MASTER REPLIKASI

Gambar 1. Cluster Multi-Master Database

Perbedaan antara MySQL Replikasi dan Galera Replikasi, Replikasi MySQL merupakan bagian dari database standar, serta bersifat *asynchronous*. Proses tulis dan modif selalu dilakukan pada satu master, lalu akan disebarkan ke *slave*. Konsep seperti ini tidak disarankan karena sangat rentan bagi node server untuk tidak sinkron jika terjadi kegagalan *master*. Tidak ada failover otomatis atau sinkronisasi ulang pada kasus ini. Sedangkan, Galera

Replication adalah plug-in ke MySQL serta memungkinkan setup *master-master* yang sinkron semua node. Dalam kluster Replikasi Galera, semua node adalah master serta aplikasi dapat membaca dan menulis dari *node* manapun. Proses yang dilakukan oleh galera ditampilkan oleh gambar 1. Transaksi dilakukan secara sinkron pada semua *node*, apabila terjadi kegagalan *node*, *node* lainnya akan beroperasi dan selalu *realtime*. Saat *node* yang gagal akan muncul lagi, *node* ini secara otomatis melakukan sinkronisasi dengan *node* lain sebelum diizinkan kembali ke kluster. Konsep ini seperti ini tidak ada data yang hilang saat sebuah *node* gagal [5].

Manfaat Galera Replikasi adalah sebagai berikut:

1. Solusi untuk ketersediaan data yang tinggi dengan replikasi yang sinkron, failover dan resinkronisasi.
2. Tidak akan ada data yang hilang.
3. Semua server, memiliki data yang mutakhir.

Kekurangan Galera Replikasi;

1. Hanya mendukung InnoDB.
2. Dengan meningkatnya jumlah master yang dapat ditularkan menyebabkan *latency* dalam transaksi menjadi tinggi.

2.4. Linux Virtual Server

Linux Virtual Server membuat perangkat lunak yang mengarahkan koneksi jaringan ke beberapa *server* yang memiliki beban kerja masing-masing, yang dapat digunakan untuk membangun layanan berskala tinggi dan selalu tersedia. Aplikasi berinteraksi dengan *cluster* seolah-olah itu *server* tunggal. Pengunjung tidak terpengaruh oleh interaksi dengan *cluster* dan tidak perlu modifikasi. Skalabilitas dicapai secara transparan dengan menambahkan atau menghapus sebuah *node* di *cluster*. *High availability* disediakan dengan mendeteksi kegagalan *node* atau daemon dan sistem LVS akan dikonfigurasi ulang secara tepat dan cepat [6].

3. ANALISA DAN PEMBAHASAN

Pada penelitian ini akan diimplementasikan *Replikasi Database Pada Multi node server*. Secara umum alur penelitian dijelaskan pada gambar berikut ini:

Gambar 2. Metodologi Penelitian

1. Tahap Pertama (Studi Pustaka / *Literature*)
Tahap ini dilakukan dengan cara mengkaji dan mempelajari sumber-sumber referensi berupa *literature* yang terdapat pada naskah ilmiah, buku, internet, atau lainnya.
2. Tahap Kedua (Observasi)

Tahap ini dilakukan pengamatan dan pencatatan terhadap data yang diperoleh. Observasi dilakukan dengan menggunakan software simulator jaringan komputer konvensional dan dilakukan dokumentasi terhadap paket data yang diperoleh. Data tersebut akan digunakan sebagai data awal pada teknologi Database.

3. Tahap Ketiga (Perancangan Sistem)

Tahap ini merupakan tahap dimana menentukan perangkat keras maupun perangkat lunak yang *suitable* dalam merancang, membuat topologi yang sesuai dengan sistem yang akan dibangun dan mengkonfigurasi *devices* yang dibutuhkan untuk membangun sebuah sistem yaitu sistem cluster terdistribusi pada teknologi Database. Langkah selanjutnya, melakukan pengembangan sistem replikasi dengan menerapkan metode atau algoritma yang telah ditentukan sebelumnya. Adapun topologi yang akan di bangun pada penelitian ini adalah seperti gambar 3 dan gambar 4 berikut ini:

Gambar 3. Konsep Replikasi Database

Gambar 4. Topologi penelitian

Pada gambar 4 di atas adalah cluster Replikasi Galera yang akan di rancang pada penelitian ini, semua node adalah master serta aplikasi dapat membaca dan menulis dari *node* manapun. Transaksi dilakukan secara sinkron pada semua *node*, apabila terjadi kegagalan *node*, *node* lainnya akan beroperasi dan selalu *realtime*. Saat *node* yang gagal muncul lagi, ia secara otomatis melakukan sinkronisasi dengan *node* lain sebelum diizinkan kembali ke cluster. Tidak ada data yang hilang saat sebuah *node* gagal melakukan fungsinya dengan baik.

4. Tahap Keempat (Pengujian)

Setelah semua sistem selesai dibangun, kemudian dilakukan pengujian sesuai dengan batasan masalah dengan mengukur parameter-parameternya seperti (1) *delay*, (2) *jitter*, dan (3) *packet loss*.

5. Tahap Kelima (Analisa)

Hasil dari pengujian pada tahap sebelumnya kemudian akan dianalisa dengan tujuan untuk mengetahui kekurangan pada hasil perancangan dan faktor penyebabnya, sehingga dapat dilakukan pengembangan pada penelitian selanjutnya.

6. Tahap Keenam (Kesimpulan dan Saran)

Pada tahap ini, akan dilakukan penarikan kesimpulan berdasarkan studi pustaka, hasil perancangan sistem, dan hasil analisa sistem, dan kemudian dihadirkan pula beberapa poin saran dari penulis untuk penelitian selanjutnya.

4. IMPLEMENTASI

Percobaan penelitian Implementasi Sistem Database Terdistribusi Dengan Metode Multi-Master Database Replication dilakukan di Fakultas Ilmu Komputer. Adapun komponen pendukung dari penelitian terdiri dari hardware dan software komputer.

Komponen hardware yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Personal Komputer
2. Komputer Server
3. Switch
4. Router
5. Proxy & NAT

Komponen software yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Database Server
2. Web Server
3. Mysql Server
4. Mysql Client
5. Database Stress Tester

Penggabungan terhadap komponen hardware & software membentuk sistem database mysql yang dapat digunakan untuk mengambil data penelitian, tool database stress tester digunakan untuk membangkitkan proses komputasi maksimal dari sistem database yang telah dibangun. Insfrastruktur server dan jaringan dapat dilihat pada gambar .

Gambar 5. Insfrastruktur Jaringan Penelitian

Penggunaan tools stress tester dapat membangkitkan query-query yang diperlukan untuk menguji kemampuan server database. Pada gambar 6, dihasilkan resource komputasi yang sangat tinggi pada server virtualisasi yang menjalankan sistem operasi pada node-node database server.

Gambar 6. Resource Komputasi Pada Percobaan

Pada saat dilakukan komputasi pada masing-masing node, setiap database server akan mengeksekusi query mysql, seperti baca, tulis, hapus, dan modifikasi. Gambar 7 berikut ini menunjukkan data statistik query dari database server.

Gambar 7. Beban dari Cluster Database

Setiap query yang dibangkitkan tentu akan berdampak pada performa komputasi, setiap node memiliki resource komputasi yang berbeda-beda, hal ini ditunjukkan pada gambar 8 berikut ini.

Host	Ping(μs)	Failed SSH	CPU Util / Steal (%)	Loadavg 1 / 5 / 15
✓ 10.100.200.250	680	0	90.51 / 0.0	37.02 / 35.43 / 35.08
✓ 10.100.200.251	245	0	90.14 / 0.0	36.78 / 35.41 / 35.06
✓ 10.100.200.252	250	0	87.33 / 0.0	35.44 / 35.02 / 34.92
✓ 10.100.200.253	459	0	87.42 / 0.0	35.44 / 35.02 / 34.92
✓ 10.100.200.254	292	0	87.34 / 0.0	35.44 / 35.02 / 34.92
✓ 10.100.200.255	464	0	87.69 / 0.0	35.44 / 35.02 / 34.92
✓ 10.100.57.253	36	0	0.40 / 0.0	0.6 / 0.47 / 0.49

Gambar 8. CPU masing-masing node

Gambar 9. Mekanisme replikasi master-master

Proses replikasi di ilustrasikan seperti gambar 9. Waktu yang diperlukan untuk melakukan replikasi adalah proses pengiriman data dari salah satu node (insert, delete, modif) kepada member kluster pada infrastruktur database server. Waktu yang dibutuhkan untuk melakukan replikasi dengan 6 node adalah kurang dari 0.2 detik.

Setiap ada perubahan data dari salah satu node, maka perubahan data tersebut akan langsung direplikasi kepada node-node server lainnya. Waktu rata-rata yang diperlukan oleh masing-masing node untuk melakukan replikasi adalah 0.08-0.11 detik.

Tabel 1. Waktu Replikasi Master-Master

No	Kueri	Node1	Node2	Node3	Node4	Node5	Node6	Rata-Rata
----	-------	-------	-------	-------	-------	-------	-------	-----------

1	Node1	0	0.2	0.1	0.1	0.1	0.1	0.1
2	Node2	0.2	0	0.1	0.1	0.1	0.1	0.1
3	Node3	0.1	0.2	0s	0.1	0.1	0.1	0.1
4	Node4	0.1	0.1	0.1	0s	0.1	0.1	0.08
5	Node5	0.1	0.1	0.1	0.1	0s	0.1	0.08
6	Node6	0.2	0.1	0.1	0.2	0.2	0s	0.13
Rata-Rata		0.11	0.11	0.08	0.1	0.1	0.08	0.09

5. KESIMPULAN

Kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Sistem kluster database mampu mengatasi kekurangan resource komputasi pada engine database.
2. Aplikasi master-slave mampu menyediakan backup terhadap data yang tersimpan di dalam database.
3. Aplikasi slave dapat memperoleh *update* dari perangkat master kurang dari 0.2 detik.

REFERENCES

- [1] T. Zhou and Y. Wei, "Database replication technology having high consistency requirements," in *2013 IEEE Third International Conference on Information Science and Technology (ICIST)*, 2013, pp. 793–797.
- [2] J. Grov, L. Soares, A. J. Correia, J. Pereira, R. Oliveira, and F. Pedone, "A Pragmatic Protocol for Database Replication in Interconnected Clusters," in *2006 12th Pacific Rim International Symposium on Dependable Computing (PRDC'06)*, 2006, pp. 230–237.
- [3] R. Ramakrishnan and J. Gehrke, *Database Management Systems, 3rd Edition*, 3rd edition. Boston: McGraw-Hill, 2002.
- [4] B. Aditya and T. Juhana, "A high availability (HA) MariaDB Galera Cluster across data center with optimized WRR scheduling algorithm of LVS - TUN," in *2015 9th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 2015, pp. 1–5.
- [5] M. C. Mazilu, "Database Replication," *Database Syst. J.*, vol. 1, no. 2, pp. 33–38, 2010.
- [6] Wensong Zhang, "Linux Virtual Server for Scalable Network Services." [Online]. Available: <http://www.linuxvirtualserver.org/>. [Accessed: 10-May-2018].