

Replikasi

Ridzky Novasandro, 32349
Alvian Edo Kautsar, 32647
Jurusan Teknik Elektro FT UGM,
Yogyakarta

I. KONSEP DASAR REPLIKASI

Replikasi adalah suatu teknik untuk melakukan copy dan pendistribusian data dan objek-objek database dari satu database ke database lain dan melaksanakan sinkronisasi antara database sehingga konsistensi data dapat terjamin.

Dengan menggunakan teknik replikasi ini, data dapat didistribusikan ke lokasi yang berbeda melalui koneksi jaringan lokal maupun internet. Replikasi juga memungkinkan untuk mendukung kinerja aplikasi, penyebaran data fisik sesuai dengan penggunaannya, seperti pemrosesan transaksi online dan DSS (Decision Support System) atau pemrosesan database terdistribusi melalui beberapa server.

Keuntungan replikasi tergantung dari jenis replikasi tetapi pada umumnya replikasi mendukung ketersediaan data setiap waktu dan dimanapun diperlukan.

Adapun keuntungan lainnya adalah :

1. Memungkinkan beberapa lokasi menyimpan data yang sama. Hal ini sangat berguna pada saat lokasi-lokasi tersebut membutuhkan data yang sama atau memerlukan server yang terpisah dalam pembuatan aplikasi laporan.
2. Aplikasi transaksi online terpisah dari aplikasi pembacaan seperti proses analisis database secara online, data smart atau data warehouse.
3. Memungkinkan otonomi yang besar. Pengguna dapat bekerja dengan meng-copy data pada saat tidak terkoneksi kemudian melakukan perubahan untuk dibuat database baru pada saat terkoneksi
4. Data dapat ditampilkan seperti layaknya melihat data tersebut dengan menggunakan aplikasi berbasis Web
5. Meningkatkan kinerja pembacaan
6. Membawa data mendekati lokasi individu atau kelompok pengguna. Hal ini akan membantu mengurangi masalah karena modifikasi data dan pemrosesan query yang dilakukan oleh banyak pengguna karena data dapat didistribusikan melalui jaringan dan data dapat dibagi berdasarkan kebutuhan masing-masing unit atau pengguna.
7. Penggunaan replikasi sebagai bagian dari strategi standby server.

Replikasi dapat digunakan apabila sebuah organisasi atau perusahaan didukung oleh hardware dan aplikasi software dalam sebuah sistem yang terdistribusi. Aplikasi yang berbeda mempunyai kebutuhan yang berbeda untuk otonomi dan konsistensi data. Replikasi diperlukan dalam sistem terdistribusi apabila berikut ini:

1. Mengcopy dan mendistribusikan data dari satu atau lebih lokasi
2. Mendistribusikan hasil copy data berdasarkan jadwal
3. Mendistribusikan perubahan data ke server lain
4. Memungkinkan beberapa pengguna di beberapa lokasi untuk melakukan perubahan dan kemudian menggabungkan data yang telah dimodifikasi
5. Membangun aplikasi data yang menggunakan perlengkapan online maupun offline
6. Membangun aplikasi Web sehingga pengguna dapat melihat volume data yang besar.

Tiga tujuan utama dilakukannya proses replikasi:

A. *Performance Enhancements*

Kunci dari performance enhancements adalah dengan cara melakukan caching data pada client dan server. Cache browsers dan proxy servers mengopi sumber-sumber web untuk mempersingkat waktu mengambil sumber dari server yang asli. Selain itu data di replikasi secara transparan antara server-server yang berada di domain yang sama. Beban kerja (workload) di-share/di-distribusikan di antara server dengan cara menyatukan seluruh IP address server ke nama DNS situs yang bersangkutan, kemudian data-data yang dibutuhkan dikembalikan ke server-server tersebut dengan metode round-robin.

B. *Increased Availability*

Pengguna membutuhkan layanan untuk tersedia pada hampir setiap saat. Sehingga layanan yang dibutuhkan harus dapat diakses dengan persentasi keberhasilan mendekati 100%. Faktor-faktor yang mempengaruhi increased availability adalah sebagai berikut:

- kegagalan server
- partisi jaringan dan operasi yang terputus (network partitions and disconnected operation): diskoneksi komunikasi seperti ini merupakan kekurangan dari aspek mobilitas pengguna.

Replikasi berperan pada aspek menjaga ketersediaan data meskipun ada kegagalan server secara otomatis. Jika data direplikasikan di server yang bebas dari kegagalan maka client dapat mengakses data yang dibutuhkan di server yang melakukan replikasi tersebut (alternate server) bukan di default server. Jika setiap server n memiliki probabilitas p untuk gagal, sehingga ketersediaannya dapat dirumuskan sebagai berikut:

$$1 - \text{probabilitas gagal} = 1 - p^n$$

C. *Fault Tolerance*

Data yang tersedia dengan baik belum tentu merupakan data yang benar. Data tersebut belum tentu merupakan data yang ter-update. Sebuah layanan fault tolerance selalu menyediakan data yang paling baru dan benar bagi client yang membutuhkannya.

II. MERENCANAKAN REPLIKASI

Perencanaan yang baik sebelum replikasi dapat memaksimalkan konsistensi data, meminimalkan kebutuhan jaringan dan menghindari beberapa masalah. Beberapa hal yang menjadi pertimbangan dalam perencanaan replikasi :

1. Kebutuhan data yang akan diubah dan siapa yang mengubah
2. Pendistribusian data memerlukan konsistensi, otonomi dan kesinambungan
3. Kelengkapan replikasi yang meliputi kebutuhan user, infra struktur teknik, jaringan dan keamanan serta karakteristik data
4. Jenis replikasi dan pilihannya
5. Topologi replikasi dan bagaimana mewujudkannya agar sesuai dengan jenis replikasi

III. JENIS - JENIS REPLIKASI

A. *Snapshot replication*

Mendistribusikan data yang dapat dilihat pada saat tertentu tanpa melakukan update. Biasanya digunakan pada saat memerlukan tampilan data seperti : daftar harga, katalog, data yang digunakan untuk pengambilan keputusan. Data-data ini sifatnya hanya 'read only'.

Replikasi ini membantu pada saat :

- data sebagian besar statis dan tidak sering berubah

- dapat menerima copy data yang telah melewati batas waktu yang ditentukan
- datanya sedikit

B. Transactional replication

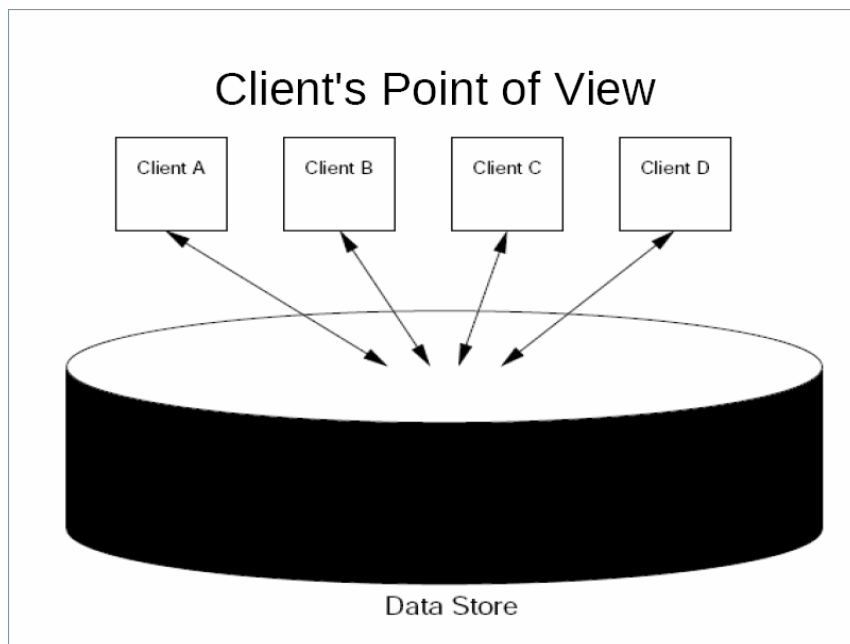
Memelihara kekonsistenan transaksi yang terjadi

C. Merge replication

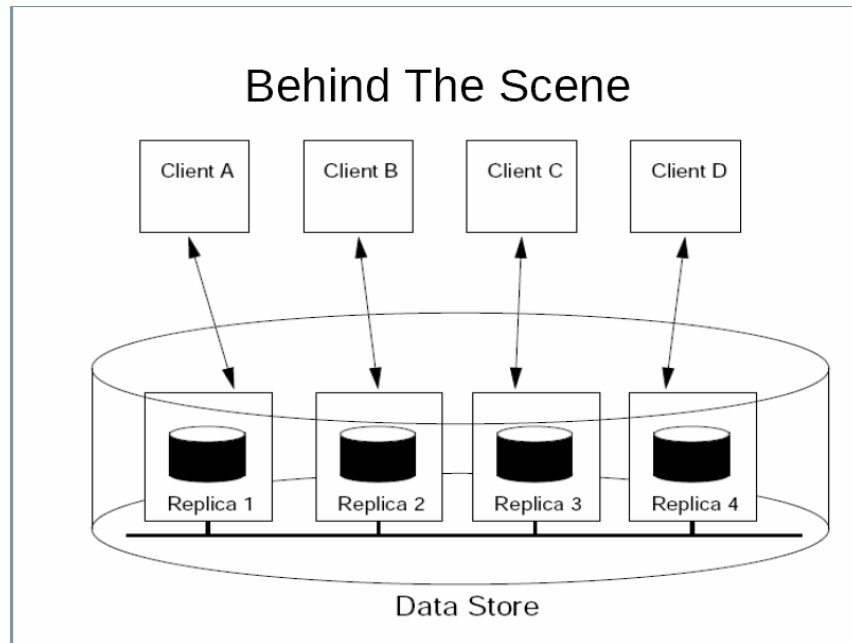
Merge replication memungkinkan pengguna bekerja dan merubah data sesuai dengan wewenangnya. Pada saat server tidak dikoneksikan ke seluruh lokasi dalam topologi, replikasi merubah ke nilai data yang sama.

IV. TRANSAKSI DENGAN DATA YANG TEREPLIKASI

Dari sudut pandang client, sebuah transaksi pada objek-objek yang ter-replikasi harus tampak sama dengan objek yang tak-ter-replikasi. Pada sebuah sistem yang tak ter-replikasi, transaksi tampak dieksekusi satu per satu pada urutan tertentu. Hal ini dapat terjadi dengan memastikan transaksi client terorganisasi secara serial.



Gambar 1. Client's Point of View



Gambar 2. Behind The Scene

Terdapat tiga skema replikasi yang akan berfungsi dengan baik ketika kumpulan replica managers dibagi menjadi beberapa subgrup oleh sebuah partisi jaringan:

a.) Kopi tersedia dengan validasi

Replikasi kopi yang tersedia di implementasikan di setiap partisi dan ketika sebuah partisi sedang diperbaiki, maka sebuah prosedur validasi diaplikasikan dan ketidak-konsistenan diperbaiki.

b.) Quorum Consensus

Sebuah subgroup harus memiliki sebuah quorum (jumlah member yang cukup) agar dapat melangsungkan layanan dengan hadirnya sebuah partisi. Ketika sebuah partisi sedang diperbaiki dan ketika sebuah replikasi manager sedang restart setelah mengalami kegagalan, replica managers memperbarui objek-objek mereka dengan prosedur recovery.

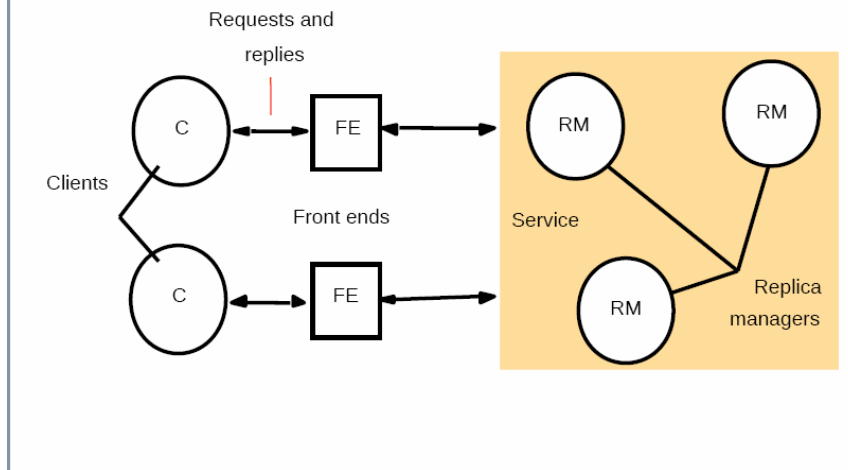
c.) Virtual Partition

Kombinasi antara Quorum Consensus dan kopi yang tersedia.

V. SYSTEM MODEL DAN GROUP COMMUNICATION

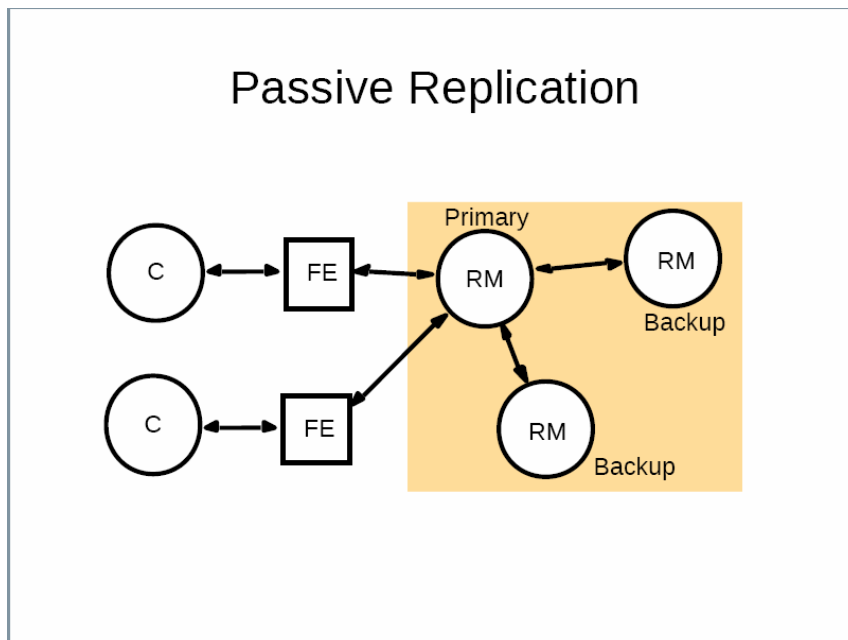
Data yang terdapat pada sistem yang biasa kita gunakan terdiri dari kumpulan unsur yang dapat dinamakan sebagai objek. Objek tersebut dapat berupa sebuah file dan lain-lain. Setiap objek tersebut diimplementasikan oleh sebuah kopi yang dinamakan sebagai replika. Partisi jaringan membuat pendeteksian kegagalan semakin sulit, padahal pendeteksian kegagalan sangat diperlukan untuk menghasilkan proses multicast yang handal dan teratur secara sistematis.

Common Replication Architecture

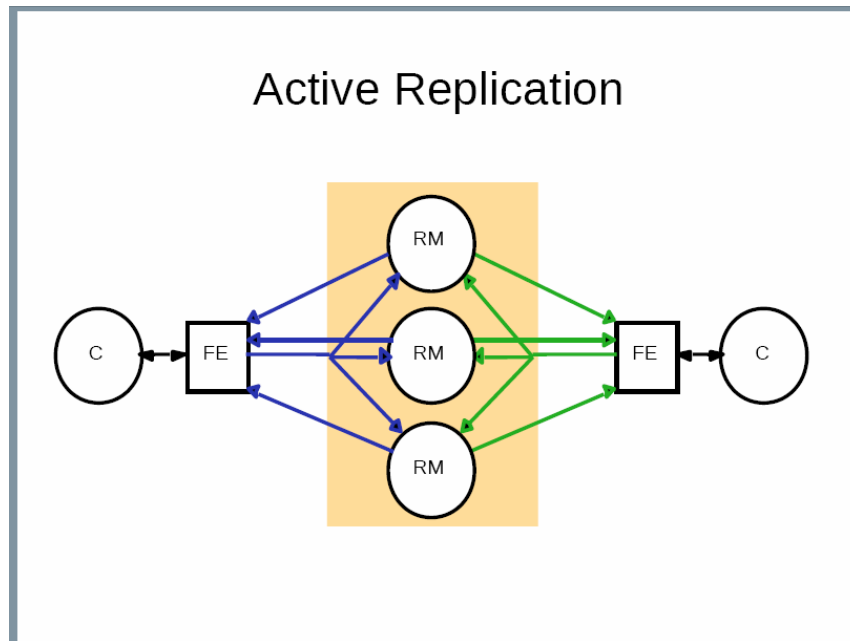


Gambar 3. Common Replication Architecture

Passive Replication



Gambar 4. Passive Replication



Gambar 5. Active Replication

A. Coordination

Replica managers berkoordinasi dalam persiapan untuk mengeksekusi request secara konsisten. Selain itu replica managers juga menentukan urutan dari request-request tersebut. Jenis-jenis pengurutan adalah sebagai berikut:

1) FIFO Ordering

First in first out. Jadi data yang dimasukkan pertama kali akan keluar sebagai output yang pertama.

2) Causal Ordering

Pengurutan diurutkan berdasarkan alasan mengapa request tersebut perlu di proses terlebih dahulu.

3) Total Ordering

Hampir seluruh aplikasi menggunakan jenis ordering FIFO.

B. Execution

Replica managers mengeksekusi request-request yang ada dengan metode tertentu, dimana efeknya dapat dikembalikan seperti semula.

C. Agreement

Replica managers menemukan kata sepakat akan efek yang akan dihasilkan oleh sebuah request. Sebagai contoh, di dalam sebuah sistem transaksi (transactional system) replica managers akan melakukan perundingan untuk menentukan apakah transaksi dapat dilanjutkan atau dibatalkan.

D. Response

Satu atau lebih replica managers merespon ke front-end. Pada beberapa sistem, satu replica manager mengirimkan responnya. Sementara pada sistem lainnya front end menerima respon dari sekumpulan replica managers dan front end memilih sebuah respon tunggal untuk dikirimkan ke client.

V. KOMUNIKASI GRUP

Implementasi penuh dari komunikasi grup terdiri dari sebuah group membership service untuk me-manage membership dinamis dari grup-grup yang ada, sebagai tambahan komunikasi multi-cast.

Sebuah group membership service memiliki empat tugas utama

a.) Memberikan antarmuka terhadap group membership changes

Layanan membership memberikan kemampuan operasi untuk membuat dan menghancurkan process groups dan untuk menambahkan serta menarik sebuah proses dari atau ke dalam sebuah group. Pada banyak sistem, sebuah proses tunggal dapat dimiliki oleh beberapa grup pada saat yang bersamaan. Sebagai contoh adalah multicast IP.

b.) Mengimplementasikan sebuah pendeteksian kegagalan

Layanan ini mengimplementasikan sebuah pendeteksi kegagalan. Layanan ini memonitor group members sehingga bisa mengetahui apabila sistem tersebut crash ataupun tidak bisa dicapai karena kegagalan komunikasi. Pendeteksi ini mengkategorikan proses sebagai suspected atau unsuspected. Layanan ini menggunakan pendeteksi kegagalan untuk membuat sebuah keputusan mengenai sebuah membership grup. Layanan akan mentidak ikutsertakan sebuah proses dari membership apabila suspected (diperkirakan) gagal ataupun sulit untuk dicapai.

c.) Menotifikasi member dari perubahan group membership

Layanan ini akan memberitahu anggota group ketika terdapat sebuah proses yang ditambahkan atau tidak diikuti sertakan .

d.) Melakukan perluasan alamat grup

Ketika sebuah proses memulticast sebuah pesan, proses tersebut tidak akan memberikan daftar proses yang terjadi dalam group melainkan akan memberikan identifier group. Sistem management membership akan memperluas identifier ke dalam group membership yang bersangkutan untuk proses pengiriman. Layanan ini dapat mengkoordinasikan pengiriman delivery dengan perubahan membership dengan cara mengatur perluasan alamat.

VI. Model Konsistensi

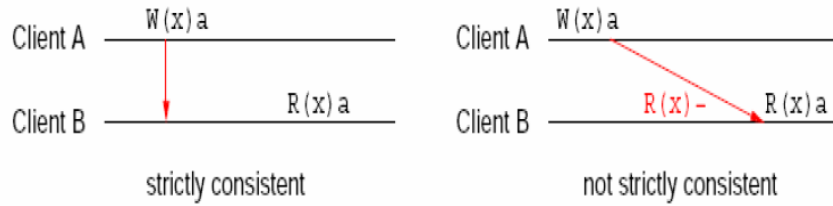
A. Data Centric

Kontak antara proses dan penyimpanan data. Jika proses mematuhi aturan tertentu penyimpanan data dapat terjadi. Setiap model akan berupaya untuk mengembalikan hasil dari proses write terakhir terakhir dari sebuah proses read.

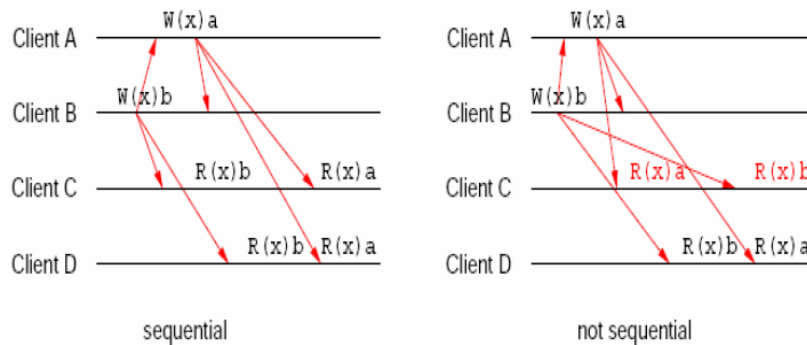
Terdapat 2 jenis ordering :

- Strong ordering
- Weak ordering

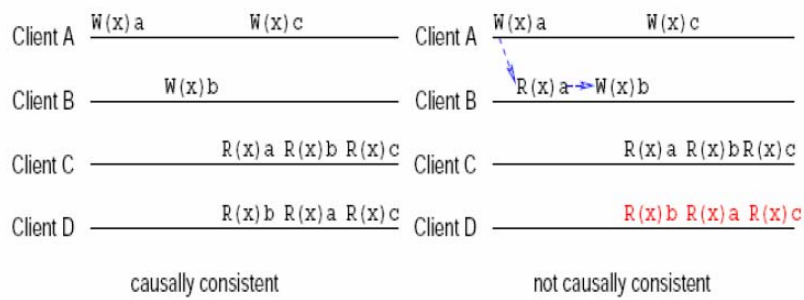
Berikut adalah skema dari macam-macam jenis ordering Data Centric :



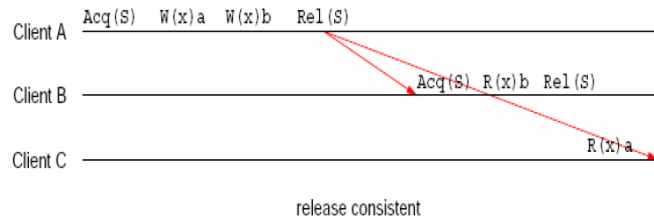
Gambar 6. Skema Strict Consistency



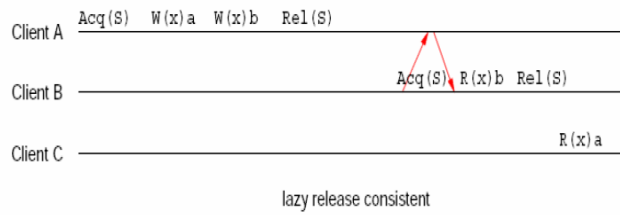
Gambar 7. Sequential Consistency



Gambar 8. Casual Consistency



Gambar 9. Release Consistency

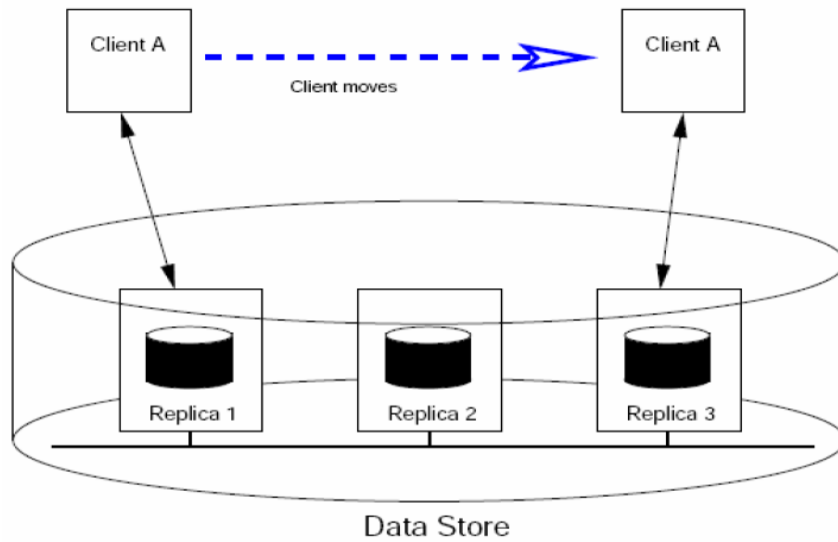


Gambar 10. Lazy Release Consistency

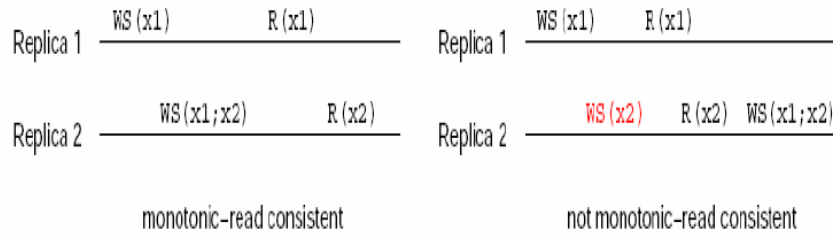
B. Client Centric

- Memberikan kepastian mengenai proses order sebuah klien tunggal.
- Klien tunggal dapat mengakses penyimpanan data.
- Klien dapat mengakses replika yang berbeda-beda.
- Klien tidak melakukan share data

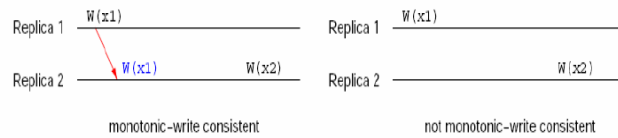
Berikut adalah skema dari macam-macam Client Centric :



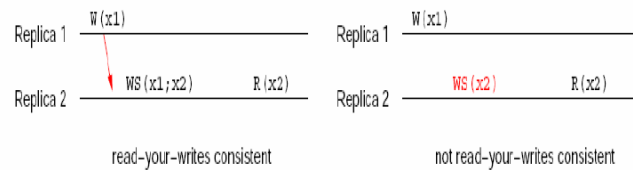
Gambar 11. Data Store



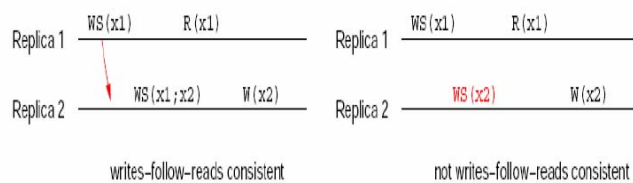
Gambar 12. Monotonic Reads



Gambar 13. Monotonic Write



Gambar 14. Read Your Write



Gambar 15. Write Follows Read

VII. PROSES REPLIKASI PADA IBM AS/400

Replikasi data terdiri dari 2 proses dasar, yaitu :

- Menampung perubahan data dari tabel atau view dalam database sumber
- Meng-copy perubahan data dari tabel sumber ke satu atau beberapa tabel tujuan dalam database yang sama atau berbeda

Replikasi menentukan bagaimana perubahan data yang ditangkap dan bagaimana data dicopy ke tujuan. Replication administration merupakan proses dari pembuatan dan mengelola kedua proses di atas. Replication Administration meliputi :

- Pendefinisikan tabel, database atau view sebagai tabel sumber untuk merubah penangkapan dan replikasi
- Pembuat sebuah tabel replikasi tujuan
- Pendefinisian kondisi dimana data dari sumber yang telah ditentukan dicopy ke tujuan yang telah ditentukan
- Perubah penangkapan dan replikasi
- Pembuat sebuah tabel replikasi tujuan
- Pendefinisian kondisi dimana data dari sumber yang telah ditentukan dicopy ke tujuan yang telah ditentukan

Perencanaan dilakukan sebelum melakukan setting perlengkapan. Pada tahap ini ditentukan skenario replikasi yang telah diimplementasikan oleh orang lain dan merencanakan pada tingkat sistem dan aplikasi.

Perencanaan sistem akan menentukan kelayakan implementasi replikasi. Kelayakan ini ditentukan oleh kebutuhan sistem, seperti produk replikasi yang digunakan, kebutuhan software dan hardware, kebutuhan storage, kapasitas CPU, konfigurasi jaringan, otoritas keamanan dan user ID.

Perencanaan aplikasi akan mengevaluasi kebutuhan tingkat aplikasi terhadap kemampuan berbagai produk. Istilah kebutuhan tingkat aplikasi menunjukkan kebutuhan aplikasi tujuan, terutama kebutuhan jenis data dalam tabel tujuan (meliputi bagian dari setting, transformasi atau peningkatan terhadap data sumber), peredaran data hasil copy dan konsistensi data yang dicopy.

VIII. ALAT YANG DIGUNAKAN

Replikasi terdiri dari administrasi utama replikasi yaitu Control Center, Capture dan Apply Program. Control Center adalah alat administrasi database DB2 yang digunakan untuk administrasi replikasi. Control Center secara otomatis menge-nali fungsi seperti pembuatan tabel tujuan dan tabel kontrol pada saat menentukan informasi hasil copy tujuan. Capture Program merupakan alat replikasi yang merubah data dan membuat perubahan itu dapat dilakukan pada replikasi. Capture program berjalan pada database server tujuan.

Capture & Apply Program menangkap perubahan yang terjadi pada data dalam tabel sumber replikasi dengan cara membaca database log atau journal.

Administrasi replikasi mempunyai 2 tugas utama yaitu mendefinisikan sumber replikasi dan menentukan tujuan replikasi. Sumber replikasi berupa tabel atau view digunakan sebagai sumber untuk copy data ke satu atau beberapa tabel tujuan. Tujuan replikasi merupakan spesifikasi dari satu atau beberapa tabel tujuan beserta lokasinya, struktur dan waktu penjadwalan sebaik peningkatan SQL yang diperlukan.

Langkah praktis melakukan replikasi :

- Merencanakan skenario replikasi pada tingkat sistem dan aplikasi
- Mendefinisikan sumber replikasi
- Mendefinisikan tujuan replikasi
- Melakukan konfigurasi Capture dan Apply Program
- Memulai Capture Program
- Memulai Apply Program

REFERENCES

- [1] George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems Concept and Design," Addison Wesley, 3rd Edition, 2001.
- [2] Replicarion Consistency, <http://lecturer.ukdw.ac.id/>