

REPLIKASI DATA SQL SERVER CLUSTER MENGGUNAKAN MIRRORING SQL SERVER VIRTUAL UNTUK DISASTER RECOVERY SITE

Nanang Sadikin¹⁾, Marlina Sari²⁾, Marlina Dian Safitri³⁾

¹⁾³⁾, Teknik Informatika Universitas Islam Attahiriyah Jakarta
Jalan melayu Kecil III No 15 Bukit Duri

²⁾, Computer Engineering Politeknik Negeri Medan
Jalan. Universitas Sumatra Utara

Email : nanang_sadikin@yahoo.com¹⁾, :marlina.sari@hotmail.com²⁾, marlinadiansafitri@gmail.com³⁾,

Abstrak

Database merupakan fungsi server untuk infrastruktur teknologi informasi yang penting. Saat server database server terhenti, server aplikasi tidak bisa mengakses database layanan terhadap pengguna aplikasi terhenti. Kita bisa menggunakan beberapa server pada SQL Server cluster di satu data center untuk mencegah kegagalan perangkat keras. Namun, hal ini tidak bisa membantu jika data centernya sendiri mengalami bencana.

Pada paper ini akan dijelaskan solusi untuk mengatasi data center utama yang mengalami bencana dimana SQL Server berada. Di sini kita akan menghubungkan SQL Server Cluster yang merupakan server fisik yang berada di data center utama dengan SQL Server virtual yang berada di data center cadangan atau DRC, menggunakan fitur mirroring yang terdapat pada SQL Server.

Kata kunci: Disaster Recovery, SQL Cluster, SQL Mirroring, Virtualization

1. Pendahuluan

Database merupakan komponen penting dan kritical di dalam infrastruktur Information Technology. Banyak aplikasi tergantung dengan database untuk menyediakan layanan ke pemakai, misalnya aplikasi Enterprise Resource Planning (ERP). Beberapa jenis fungsi aplikasi ERP misalnya Customer Relationship Management (CRM), Business Intelligence and Data Warehouse (BIDW), Supply Chain Management (SCM), Financial Controlling (FICO), dan Human Resource Management (HRM) harus menggunakan database. Karena database merupakan komponen yang sangat penting, berbagai cara akan dilakukan untuk melindungi database. Cara yang pertama adalah dengan menggunakan backup. Database akan dibackup secara teratur menggunakan perangkat lunak backup yang menggunakan backup agent untuk database. Database akan dibackup ke media yang berupa disk atau tape. Database akan dibackup menggunakan jenis backup full backup atau incremental/differential backup. Full backup umumnya berjalan dalam jangka waktu mingguan atau bulanan dan

jenis backup incremental/differential backup berjalan harian.

Keuntungan disk backup adalah memiliki kapasitas yang besar dan waktu backup yang cepat, namun backup ke tape menyediakan fleksibilitas. Tape backup bisa mengirim ke off-site untuk menyediakan salinan data yang aman. Tape bisa dikirim ke safe deposit di bank atau ke perusahaan arsip dan layanan penyimpanan data atau ke *third party data center*. Jika database di data center hilang, kita bisa memulihkan database dari backup. Restore bisa dijalankan ke server semula atau server/lokasi yang lain.

Mekanisme perlindungan lainnya untuk mengamankan database adalah menggunakan *high availability*. SQL Server Failover Cluster digunakan untuk menyediakan *high availability* untuk SQL Server database. SQL Server Failover Cluster terdiri dari dua atau lebih SQL Server Node dan sebuah Storage Area Network (SAN) yang menyediakan *shared disk* antara nodes SQL Server. Jika sebuah node down maka SQL Server node lainnya mengambil alih SQL Server cluster resource, misalnya *services* dan *disk resources*. Namun tetap saja dalam hal ini, Storage Area Network (SAN) merupakan *single point of failure*. Jika Storage Area Network rusak, seluruh cluster akan runtuh dan database tidak bisa diakses. Selain itu, cluster merupakan sistem yang rumit, yang tidak bisa dibangun hanya dalam satu malam. Kita bisa melindungi SQL Server cluster menggunakan backup, tapi tetap saja dibutuhkan waktu untuk memulihkan database.

Bencana bisa saja terjadi tanpa bisa diprediksi sebelumnya dalam bentuk bencana alam seperti gempa bumi, banjir, gunung meletus atau bahaya kebakaran. Bencana ini akan menjadi ancaman untuk data center. Akan merusak data center dan keseluruhannya sebagai infrastruktur, termasuk database. Untuk mencegah hal ini terjadi, kita memerlukan Disaster Recovery site sebagai replika Data center site untuk melindungi secara keseluruhan aplikasi dan database.

SQL Server Mirroring

Database mirroring merupakan solusi untuk meningkatkan ketersediaan *SQL Server database*. Mirroring diterapkan *per-database basis* dan bekerja hanya dengan databases yang menggunakan *full*

recovery model. Keuntungan Database Mirroring adalah sebagai berikut:

1. Meningkatkan ketersediaan database

Pada saat terjadinya disaster, di *high-safety mode with automatic failover*, failover secara cepat membawa *standby copy database* menjadi online (tanpa kehilangan data). Pada operating modes lainnya, database administrator memiliki alternatif untuk memaksa service (dengan kemungkinan kehilangan data) ke *standby copy* dari database.

2. Meningkatkan perlindungan data

Database mirroring menyediakan redundansi data *complete* atau *almost complete*, tergantung apakah *operating mode* menggunakan *high-safety* atau *high-performance*. Database mirroring partner yang berjalan di SQL Server 2008 Enterprise atau versi yang lebih tinggi secara otomatis mencoba untuk memperbaiki beberapa jenis error yang mencegah membaca data page. Partner yang tidak bisa membaca page meminta data page yang baru dari partner lainnya. Jika permintaan ini berhasil, page yang tidak bisa dibaca ditimpa oleh salinannya, yang umumnya menyelesaikan error tersebut.

3. Meningkatkan ketersediaan production database selama upgrades

Untuk meminimalkan downtime untuk *mirrored database*, kita bisa secara berurutan meng *upgrade instances* SQL Server yang menghosting *failover partners*. Ini akan incur the downtime hanya *single failover*. Bentuk upgrade seperti ini dikenal sebagai *rolling upgrade*.

Database mirroring menyimpan dua copy *single database* yang harus ada pada server instances SQL Server Database Engine yang berbeda. umumnya, server instances ini berada pada komputer di lokasi berbeda. *Starting database mirroring* pada sebuah database, menginisiasi sebuah hubungan, yang dikenal sebagai *database mirroring session*, antara kedua *server instances* ini.

Satu *server instance* melayani database ke client (*principal server*). Instance lainnya bertindak sebagai *hot* atau *warm standby server* (*mirror server*), tergantung pada konfigurasi dan state dari *mirroring session*. Saat database mirroring session dilakukan sinkronisasi, database mirroring menyediakan *hot standby server* yang mendukung failover yang cepat tanpa kehilangan data dari transaksi yang telah ditulis. Saat session tidak disinkronisasi, mirror server umumnya tersedia sebagai *warm standby server* (dengan kemungkinan data hilang).

Principal dan *mirror server* berkomunikasi dan bekerjasama sebagai partner di dalam *database mirroring session*. Kedua partners menjalankan peran yang saling melengkapi di dalam session: peran *principal* dan peran *mirror*. Dalam waktu tertentu, satu partner menjalankan peran *principal*, dan partner lainnya menjalankan peran *mirror*. Masing-masing partner

digambarkan sebagai pemilik perannya saat ini. Partner yang memiliki peran *principal* dikenal sebagai *principal server*, dan database yang dimilikinya saat ini merupakan *principal database*. Partner yang memiliki peran *mirror* dikenal sebagai *mirror server*, dan database yang dimilikinya merupakan *mirror database*. Saat database mirroring dijalankan di dalam sebuah lingkungan produksi, *principal database* merupakan database produksi.

Database mirroring melakukan redoing setiap operasi insert, update, dan delete yang terjadi pada *principal database* ke dalam *mirror database* secepat mungkin. Redoing dilakukan dengan cara mengirim stream dari active transaction log records ke *mirror server*, yang menulis *log record* ke *mirror database*, secara urut, dalam waktu sesingkat mungkin. Tidak seperti replication, yang bekerja pada level logical, database mirroring bekerja pada *log record* level fisik. Mulai SQL Server 2008, *principal server* memampatkan arus transaction log record sebelum mengirimnya ke *mirror server*. Pemampatan log ini berlangsung di seluruh *mirroring sessions*.

Operating Modes

Database mirroring session berjalan dalam operasi *synchronous* atau *asynchronous*. Dalam operasi *asynchronous*, transactions commit tanpa menunggu *mirror server* untuk menulis log ke disk, yang mana memaksimalkan *performance*. Dalam *synchronous operation*, sebuah transaction di commit di kedua partners, tapi dengan biaya meningkatnya *transaction latency*.

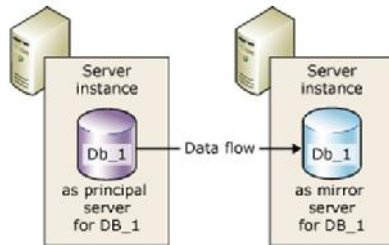
Ada dua jenis mirroring operating modes. Satu diantaranya, *high-safety mode* mendukung *synchronous operation*. Di bawah *high-safety mode*, saat session dimulai, *mirror server* mensinkronisasi *mirror database* bersama dengan *principal database* dalam waktu yang sesingkat-singkatnya. Begitu databases tersinkronisasi, transaction ditulis di kedua partner, dengan resiko meningkatnya *transaction latency*.

Operating mode kedua, *high-performance mode*, berjalan secara *asynchronous*. Mirror server mencoba untuk keep up dengan log records yang dikirimkan oleh *principal server*. *Mirror database* mungkin tertinggal di belakang *principal database*. Bagaimanapun, secara umum, selisih antar databases itu kecil. Akan tetapi, selisihnya akan menjadi signifikan jika *principal server* punya beban kerja yang berat atau sistem di mirror server memiliki beban lebih.

Pada *high-performance mode*, begitu *principal server* mengirim *log record* ke *mirror server*, *principal server* mengirim konfirmasi ke client. Dia tidak akan menunggu sebuah *acknowledgement* dari mirror server. Ini berarti bahwa *transactions commit* tanpa menunggu *mirror server* untuk menulis *log* ke disk. Operasi *asynchronous* memungkinkan *principal server* untuk berjalan dengan

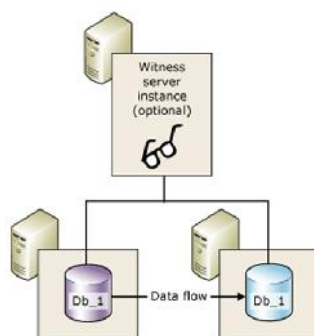
minimum transaction latency, dengan potensi resiko kehilangan beberapa data.

Semua sesi *database mirroring* mendukung hanya satu principal server dan satu mirror server. Konfigurasi ini ditunjukkan pada gambar 1 di bawah ini:



Gambar 1. Database Mirroring with Principal and Mirror Server

High-safety mode with automatic failover memerlukan *server instance* yang ketiga, yang dikenal sebagai *witness*. Tidak seperti dua partner yang lain, *witness* tidak melayani database. *Witness* mendukung *automatic failover* dengan memeriksa apakah *principal server* jalan dan berfungsi. Mirror server melakukan *automatic failover* hanya jika mirror dan witness tetap terhubung satu sama lain setelah keduanya terputus dari *principal server*.



Gambar 2. Witness Server pada Database Mirroring

Transaction Safety dan Operating Modes

Operating mode itu *asynchronous* atau *synchronous* tergantung setting *transaction safety*. Jika SQL Server Management Studio digunakan untuk mengkonfigurasi *database mirroring*, *transaction safety settings* dikonfigurasi secara otomatis saat memilih *operation mode*.

Jika Transact-SQL digunakan untuk mengkonfigurasi *database mirroring*, harus dipahami cara untuk menset *transaction safety*. *Transaction safety* diatur oleh *SAFETY property* dari perintah ALTER DATABASE. Pada database yang di mirror, pilihan SAFETY bisa FULL atau OFF.

a. Jika opsi SAFETY di set ke FULL, operasi database mirroring adalah *synchronous*, setelah fase *initial synchronizing*. Jika *witness* di set dalam *high-*

safety mode, session tersebut mendukung *automatic failover*.

b. jika opsi SAFETY di set ke OFF, operasi database mirroring adalah *asynchronous*. *Session* berjalan dalam *high-performance mode*, dan opsi WITNESS harus dijadikan OFF.

Role Switching

Di dalam konteks *database mirroring session*, principal and mirror roles umumnya saling bergantian di dalam sebuah proses yang dikenal sebagai *role switching*. *Role switching* melibatkan perpindahan principal role ke mirror server. Pada *role switching*, *mirror server* akan bertindak sebagai *failover partner* untuk *principal server*. Saat *role switch* terjadi, mirror server mengambil alih peran *principal* dan membawa *copy database* nya *online* sebagai *principal database* yang baru. *Principal server* sebelumnya, jika ada, bertindak sebagai *mirror role*, dan database nya menjadi mirror database yang baru. Secara umum, peran tersebut bisa bertukar satu sama lain.

Ada tiga bentuk role switching.

1. Automatic failover

Role switching ini membutuhkan *high-safety mode* dan kehadiran mirror server dan witness. Database harus sudah tersinkronisasi, dan *witness* harus terhubung ke *mirror server*.

Peran witness adalah untuk memeriksa apakah partner server yang ditunjuk online dan berfungsi. Jika mirror server kehilangan koneksinya ke principal server tapi witness masih terhubung principal server, mirror server tidak akan melakukan *failover*.

2. Manual failover

Jenis ini membutuhkan *high-safety mode*. Partners harus terhubung satu sama lain, dan database harus sudah tersinkronisasi.

3. Forced service (dengan kemungkinan data hilang)

Di bawah *high-performance mode* dan *high-safety mode without automatic failover*, memaksa service memungkinkan jika *principal server* telah gagal dan *mirror server* tersedia.

2. Pembahasan

Gambar 3 di bawah ini mengilustrasikan Data Center dan Disaster Recovery Sites.



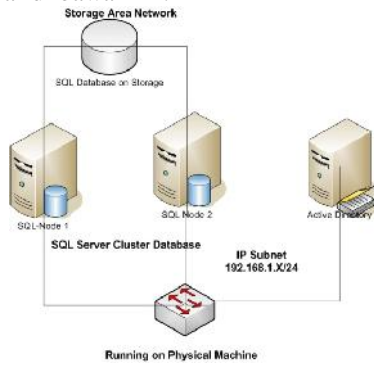
Gambar 3. Data Center dan Disaster Recovery Site

Data Center untuk produksi terletak di kota pusat dan dihubungkan oleh Wide Area Network ke Disaster Recovery Site yang jauhnya sepanjang 60 km. Kebanyakan servers infrastruktur yang berada

datacenter utama menggunakan satu server fisik per satu aplikasi, termasuk SQL Server untuk server database server. Data center DR dibangun menggunakan virtualisasi untuk semua server-server infrastruktur tersebut.

Konfigurasi SQL Server Cluster Fisik

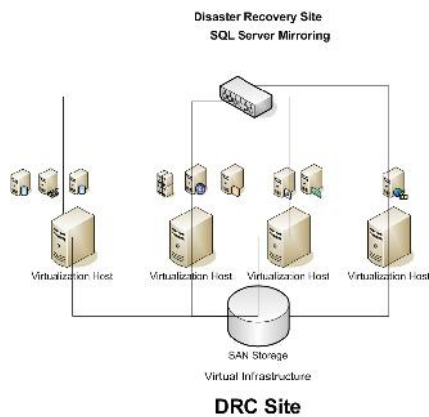
Konfigurasi SQL Server Clustering fisik ditunjukkan pada gambar di bawah ini:



Gambar 4. SQL Server Failover Cluster

Pada konfigurasi ini terdapat dua SQL Server node yang berjalan di atas mesin fisik. SQL Server cluster node berjalan menggunakan versi sistem operasi dan SQL Server yang sama. SQL Server Cluster node terhubung ke Storage Area Network (SAN) yang berisi database files untuk SQL Server. SQL Server cluster node juga terhubung ke jaringan dengan subnet 192.168.1.x. Active Directory Server diperlukan oleh SQL Server karena SQL Services harus berjalan di bawah Domain.

Konfigurasi Data Center untuk Disaster Recovery ditampilkan pada gambar di bawah ini.



Gambar 5. Disaster Recovery Site

Pada konfigurasi ini Disaster Recovery Site terletak 60 km dari Data Center yang dihubungkan menggunakan Wide Area Network (WAN). Disaster Recovery Site dibangun menggunakan infrastruktur virtualisasi. Storage Area Network (SAN) berisi file-file untuk virtual machine yang berjalan di atas virtualization host. Storage Area Network (SAN) juga berisi data data yang dibuat oleh aplikasi. Virtualization host merupakan mesin fisik yang menjalankan hypervisor. Virtualization host terhubung ke Storage Area Network menggunakan

fibre channel adapter dan kabel fiber channel. Virtualization host juga terhubung ke jaringan atau switch menggunakan network adapter card. Di atas virtualization host, terdapat beberapa virtual machine yang menjalankan beberapa aplikasi, termasuk SQL Server. Virtualization host terhubung menggunakan High Availability Cluster, jika ada satu node virtualization host yang down kemudian virtual machine running yang berjalan di atasnya akan dipindahkan ke virtualization host yang lain.

Tabel 1 di bawah ini merangkum virtualization host yang ada:

Tabel 1. Virtualization host configuration

No	Host	CPU	RAM	Disk	Network
1	VH1	2 Socket	512 GB	300 GB	4
2	VH2	2 Socket	512 GB	300 GB	4
3	VH3	2 Socket	1024 GB	300 GB	4
4	VH4	2 Socket	1024 GB	300 GB	4

Masing-masing virtualization host menjalankan aplikasi-aplikasi yang berbeda.

Tabel 2 di bawah ini merangkum aplikasi yang berjalan di atas virtualization host:

Tabel 2. Virtual Machine Configuration

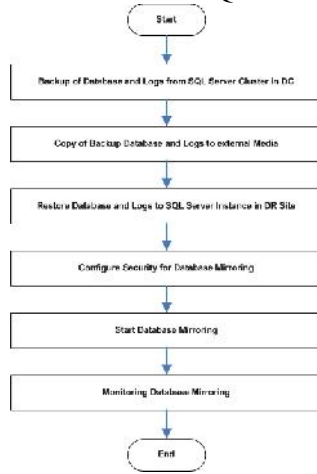
No	Host	Virtual Machine	vCPU	vRAM	vDisk	vNIC
1	VH1	VM1 (Active Directory)	2	2 GB	100 GB	1
		VM2 (SQL Server)	2	16 GB	300 GB + 2400 GB	1
		VM3 (SQL Server)	2	8 GB	300 GB + 400 GB	1
2	VH2	VM4 (Application Server)	2	8 GB	300 GB	1
		VM5 (E-mail Server)	2	16 GB	300 GB	1
		VM6 (Proxy Server)	1	4 GB	300 GB	1
3	VH3	VM7 (Web Server)	1	8 GB	300 GB	1
		VM8 (FTP Server)	1	4 GB	300 GB	1
4	VH4	VM 9(ERP Server)	4	32 GB	300 GB	1

Pada tabel 2 di atas, konfigurasi virtual machine beragam tergantung peran dan fungsi virtual machine tersebut. Sumber daya yang dialokasikan untuk virtual

machine (CPU, RAM, Disk, NIC) harus dihitung sebelum diberikan kepada virtual machine.

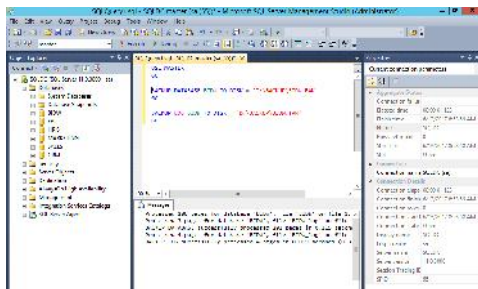
1. Database Mirroring Setup

Gambar di bawah ini mengilustrasikan langkah-langkah untuk setting database mirroring antara SQL Server Cluster di Data Center Site dan SQL Server di DR Site .



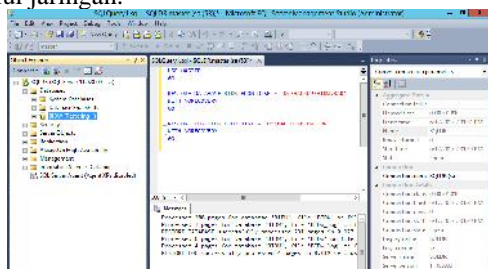
Gambar 6. SQL Server Mirroring Steps

Langkah pertama di dalam Database Mirroring adalah melakukan backup SQL Database dan Logs dari SQL Server Cluster instance pada Data Center Site. database dan log can bisa dibackup dengan menggunakan GUI SQL Server Management Studio atau menggunakan script di SQL query. Proses backup akan memakan waktu jika ukuran database besar (Tera Bytes). database mirroring bekerja satu persatu, sehingga backup akan diulangi untuk setiap database yang akan menjadi mirror.



Gambar 7. Backup Database

Setelah backup databases dan logs selesai, file-file tersebut kemudian di copy ke external media dan media tersebut di bawa ke Disaster Recovery site, karena akan memakan waktu jika dikopi ke Disaster Recovery site melalui jaringan.



Gambar 8. Restore Database

Langkah ketiga di dalam Database mirroring adalah melakukan restore database ke SQL Server instance di Disaster Recovery Site. Database dan log bisa di restore dengan menggunakan GUI SQL Server Management Studio atau menggunakan script in SQL query. Proses restore database umumnya lebih cepat daripada proses backup. Database mirroring bekerja satu demi satu, sehingga proses restore akan diulangi untuk setiap database yang akan menjadi mirror.

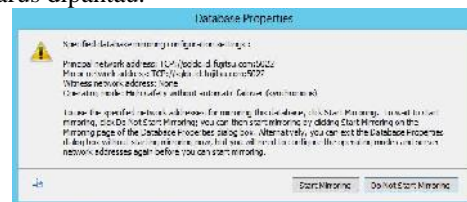


Gambar 9. Configure Database Mirroring Security Wizard

Langkah keempat di dalam melakukan database Mirroring adalah mengkonfigurasi security untuk Database Mirroring. Pada langkah ini, user account untuk SQL Server services dipilih untuk dihubungkan ke Mirroring End Point.

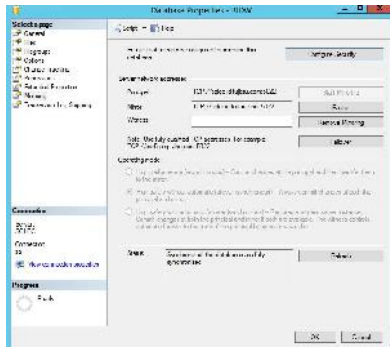
Langkah kelima di dalam database mirroring adalah Start Database Mirroring. Pada langkah ini, mirroring terbentuk antara database di Data Center Site dan Database di Disaster Recovery Site.

Langkah terakhir di dalam database Mirroring adalah melakukan monitoring database mirroring. Langkah ini termasuk memantau log SQL Server Database. Karena Database Mirroring membutuhkan recovery model dari simple ke Full, log SQL Server database akan berkembang dengan sangat cepat dan tidak terkendali. Hal ini membutuhkan penyimpanan yang sangat besar dan harus dipantau.



Gambar 10. Start database Mirroring

Setelah mengklik Start Mirroring, database akan melakukan sinkronisasi.

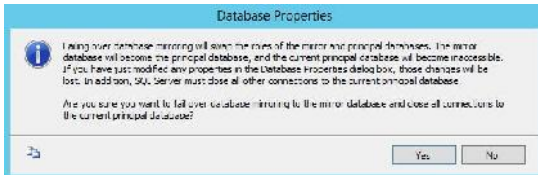


Gambar 11. Mirror Terbentuk

Seperti yang ditunjukkan pada gambar di atas, kita bisa melihat bahwa server sqldc akan menjadi principal database server, dan sqldr menjadi mirror server untuk database BIDW.

2. DR Drill

Disaster Recovery drill harus dijalankan untuk melakukan simulasi. Untuk Planned DR Drill, tes dijalankan dengan cara failover database dari DC ke DRC.

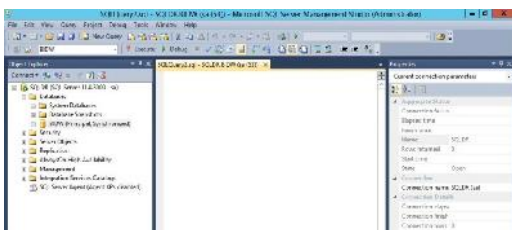


Gambar 12. Failover database

Failing over database akan menukar peran mirror dan principal database. mirror database akan menjadi principal database dan principal database sekarang akan menjadi tidak bisa diakses. Pada failover ini, SQL server harus menutup semua koneksi ke principal database yang sekarang.

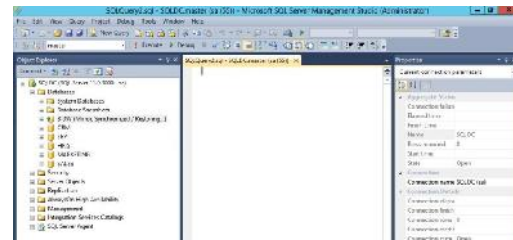
SQL Server akan menampilkan pemberitahuan jika database role berubah dari principal ke mirror atau sebaliknya sesuai yang ditampilkan pada gambar 12.

Setelah Failover, server sqldr akan menjadi principal role dan sqldc akan menjadi mirror role.



Gambar 13. SQLDR sebagai Principal Server

Setelah mengubah role, cluster SQLDR server menjadi Principal untuk BIDW database dan SQLDC server menjadi mirror server.



Gambar 14. SQLDC sebagai Mirror Server

Karena mirror server tidak bisa diakses dari aplikasi, maka administrator harus mengubah koneksi aplikasi di parameter koneksi pada aplikasi yang bersangkutan.

KESIMPULAN

Pada tulisan ini diperkenalkan bagaimana menggunakan fitur SQL mirroring yang terdapat pada SQL Server untuk membuat Disaster Recovery Site untuk SQL Server Database. DC and DR Site dipisahkan oleh lokasi geografis sejauh 60 km dan dihubungkan menggunakan Wide Area Network (WAN). Data Center menggunakan SQL Server Failover Cluster dan pada DR Site menggunakan standalone SQL Server. Database mirroring kemudian dibentuk antara DC site dan DR site. Jika Data Center site musnah karena bencana alam, kita bisa mengaktifkan mirror database di DR site untuk pemulihan.

Daftar Pustaka

- [1] Peter DeBetta, Greg Low, Mark Whitehorn et all, 2008, Introducing SQL Server 2008, Microsoft Press, Washington
- [2] Anonymous, SQL Server Documentation, link <https://docs.microsoft.com/en-us/sql/database-engine/database-mirroring/database-mirroring-sql-server>

Biodata Penulis

Nanang Sadikin, memperoleh gelar Sarjana Teknik (ST), Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Gunadarma, lulus tahun 2000. Memperoleh Gelar Magister Teknologi Informasi (MTI) Program Pasca Sarjana Fakultas Ilmu Komputer Universitas Indonesia, lulus tahun 2013. Saat ini menjadi Dosen Tetap di Jurusan Teknik Informatika Fakultas Teknik Universitas Islam Attahiriyah Jakarta

Marliana Sari, memperoleh gelar Sarjana Teknik (ST), Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Gunadarma, lulus tahun 2000. Memperoleh Gelar Magister Teknologi Informasi (MTI) Program Pasca Sarjana Magister Manajemen Sistem Informasi Bisnis, lulus tahun 2004. Saat ini menjadi dosen di Jurusan Computer Engineering Politeknik Negeri Medan.

Marlina Dian Safitri, saat ini merupakan mahasiswa semester 7 di Jurusan Teknik Informatika Fakultas Teknik Universitas Islam Attahiriyah Jakarta.