

SQL Server AlwaysOn Availability Groups

Perencanaan, Konfigurasi, dan Optimasi

Choirul Amri

SQL Server AlwaysOn Availability Groups

Perencanaan, Konfigurasi, dan Optimasi

Choirul Amri

Buku ini dijual di <http://leanpub.com/sqlagindo>

Versi ini diterbitkan pada 2019-05-20



Ini adalah sebuah buku [Leanpub](#). Leanpub memberdayakan penulis dan penerbit dengan proses Lean Publishing. [Lean Publishing](#) adalah model penerbitan ebook dalam-proses menggunakan piranti ringan dan sejumlah iterasi untuk memperoleh masukan dari pembaca, menerapkan pivot hingga Anda dapat mewujudkan komposisi buku yang pas dan menarik.

© 2014 - 2019 Choirul Amri

Tweet Buku Ini!

Bantulah Choirul Amri dengan mewartakan buku ini via [Twitter](#)!

Tweet yang disarankan untuk buku ini adalah:

[Read my upcoming book on SQL Server Availability Group, written in Bahasa Indonesia](#)

Tagar yang disarankan untuk buku ini adalah [#alwaysonID](#).

Temukan kata orang tentang buku ini dengan mengklik tautan ini untuk menampilkan tagar ini di Twitter:

[#alwaysonID](#)

Contents

| | |
|---|-----------|
| README | 1 |
| Bab 1. Pengenalan AlwaysOn Availability Groups | 2 |
| High Availability dan Disaster Recovery | 2 |
| Service Level Agreement | 4 |
| Recovery Point Objective dan Recovery Time Objective | 6 |
| Fitur SQL Server untuk mendukung HA dan DR | 6 |
| Bab 2. Arsitektur AlwaysOn Availability Groups | 7 |
| Fitur Utama Availability Groups | 8 |
| Windows Server Failover Cluster (WSFC) | 9 |
| Availability Replica | 10 |
| Active Secondary | 12 |
| Availability Mode | 13 |
| Metode Failover | 14 |
| Availability Listener | 15 |
| Bab 3. Konfigurasi Windows Cluster | 17 |
| Persiapan Windows Server 2012 R2 | 18 |
| Konfigurasi Kartu Jaringan | 21 |
| Menyiapkan Akun Layanan | 27 |
| Menjalankan Cluster Validation | 32 |
| Konfigurasi Windows Cluster | 35 |
| Konfigurasi Quorum | 37 |
| Bab 4. Konfigurasi AlwaysOn Availability Groups | 41 |
| Instalasi SQL Server | 41 |
| Konfigurasi AlwaysOn Availability Groups | 45 |
| Pengujian Failover | 55 |
| Bab 5. Konektivitas Aplikasi dan Read Only Routing | 57 |
| Bab 6. Failover dan Perpindahan Role | 58 |
| Relasi antara Failover dan Availability Mode | 58 |
| Automatic Failover (tanpa data loss) | 60 |

CONTENTS

| | |
|---|-----------|
| Planned Manual Failover (tanpa data loss) | 67 |
| Forced Manual Failover (dengan data loss) | 77 |
| Bab 7. Konfigurasi Quorum Tingkat Lanjut | 90 |
| Bab 8. Optimasi dan Pemantauan Kinerja | 91 |
| Bab 9. Integrasi dengan Microsoft Azure | 92 |
| Bab 10. Perawatan dan Operasional | 93 |

README

Untuk siapa buku ini ditulis

Buku ini ditulis untuk profil pembaca sebagai berikut:

- Database Administrator dan Database Developer yang sudah familiar dengan SQL Server, tetapi masih awam dengan teknik High Availability khususnya AlwaysOn Availability Groups.
- Administrator Windows Server yang sudah familiar dengan teknik clustering dan Active Directory, serta ingin mempelajari konfigurasi dan administrasi SQL Server.
- Para administrator database non SQL Server yang ingin mempelajari penggunaan SQL Server untuk High Availability dan Disaster Recovery
- Mereka yang sudah berpengalaman dengan SQL Server cluster dan mirroring, dan ingin mempelajari fitur baru Availability Groups.

Prasyarat pengetahuan

Untuk dapat memperoleh manfaat buku ini, pembaca sebaiknya memiliki pengetahuan berikut:

- Administrasi umum Windows Server dan jaringan, terutama yang berhubungan dengan Active Directory.
- Instalasi dan administrasi SQL Server, terutama versi SQL Server 2008 atau yang lebih baru

Versi software

Buku ini menggunakan Windows Server 2012 R2 dan SQL Server 2014 sebagai media panduan praktikum. Namun demikian semua konsep yang dijelaskan juga dapat diterapkan untuk Windows Server 2008 dan SQL Server 2012. Apabila terdapat perbedaan mendasar yang hanya terdapat di versi tertentu, maka catatan pengingat akan disertakan sebagai bahan rujukan.

Bab 1. Pengenalan AlwaysOn Availability Groups

Database yang bersifat *mission critical* sangat penting peranannya dalam mendukung roda bisnis perusahaan. Masalah yang terjadi pada system database menyebabkan kerugian finansial dan menurunkan citra perusahaan. Sebuah database yang menyimpan data pelanggan dan reservasi tiket online harus tersedia 24 jam untuk melayani aplikasi yang diakses pelanggan. Jika database tersebut mati atau mengalami gangguan, fungsi penjualan dan operasional menjadi terhambat, kepercayaan pelanggan menurun, disertai dengan kerugian finansial karena tidak dapat melakukan penjualan.

Untuk itu diperlukan teknologi yang dapat menjamin ketersediaan database sehingga agar selalu siap 24 jam sehari sepanjang tahun. Jika database mengalami masalah, fungsinya harus segera dipulihkan dengan segera sehingga tidak mengganggu operasional bisnis. Teknik ini biasa dikenal sebagai *High Availability*, yang menggunakan berbagai teknologi pendukung untuk memulihkan ketersediaan layanan dengan metode mengalihkan database dari satu server ke server lainnya jika ternyata server utama mengalami gangguan. Proses tersebut harus terjadi secara cepat, dengan keterlibatan minimum oleh administrator maupun tim pengembang.

High Availability dan Disaster Recovery

Selain *High Availability (HA)*, dikenal pula istilah *Disaster Recovery* atau pemulihan bencana yang juga biasa disingkat dengan DR. Pemulihan bencana merupakan upaya memulihkan layanan yang bermasalah karena terjadinya bencana alam atau *force majeure* seperti banjir, kebakaran, atau gempa bumi sehingga menyebabkan infrastruktur pendukung server utama tidak dapat diakses. Dalam kondisi tersebut maka diperlukan teknologi dan prosedur pendukung untuk memulihkan layanan dengan mengalihkan infrastruktur ke lokasi yang aman. Prosedur tersebut dapat berlangsung dalam hitungan jam atau hari, tergantung pada besar kecilnya kerusakan dan tingkat kepentingan sistem yang akan dipulihkan.

Perbedaan utama antara HA dan DR adalah pada metode failover [CA1] dan lama waktu pemulihan. Konfigurasi HA umumnya mensyaratkan bahwa failover harus terjadi secepat mungkin dalam hitungan detik atau menit baik secara manual maupun otomatis. Sedangkan DR terjadi secara manual dan memerlukan waktu lebih lama karena juga berkaitan dengan penyelamatan jiwa manusia serta infrastruktur pendukung.

Sistem bersifat *mission critical* biasanya mensyaratkan adanya HA dan DR untuk mendukung ketersediaan secara maksimum. Diagram dibawah memberikan gambaran umum sistem HA dan DR untuk database SQL Server 2014 yang dikonfigurasi menggunakan AlwaysOn Availability Groups.

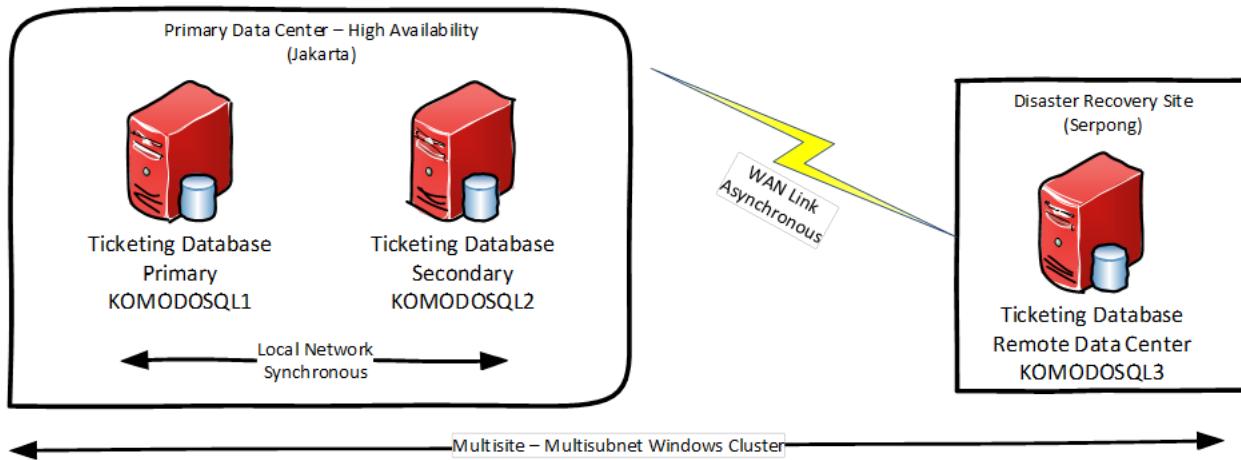


Diagram tersebut memberikan gambaran umum fasilitas HA dan DR untuk database Ticketing yang dapat dijelaskan sebagai berikut:

- Dua server database KOMODOSQL1 dan KOMODOSQL2 terletak di dalam satu data center di Jakarta, dengan KOMODOSQL1 sebagai server utama (Primary) dan KOMODOSQL2 sebagai secondary. Sinkronisasi antara kedua server tersebut dilakukan secara synchronous menggunakan jaringan kecepatan tinggi untuk menghindari kehilangan data.
- Server KOMODOSQL3 terletak di Serpong dan merupakan lokasi backup yang berfungsi sebagai fasilitas DR. Sinkronisasi dari Jakarta ke Serpong dilakukan secara asynchronous karena keterbatasan kecepatan jaringan.
- Jika terjadi masalah dengan database di KOMODOSQL1, availability groups akan melakukan *automatic failover* ke KOMODOSQL2 dalam waktu singkat sehingga tidak mengganggu kelangsungan aplikasi klien. Proses failover secara cepat dan otomatis tersebut merupakan bagian dari strategi *high availability*.
- Jika primary data center di Jakarta kejadian bencana, maka dilakukan prosedur pemulihan ke lokasi DR yang berlokasi di Serpong.

Tabel berikut menjelaskan perbedaan utama antara HA dan DR, serta prasyarat dan kondisi dimana masing-masing teknologi diterapkan.

| Aspek Pembeda | High Availability | Disaster Recovery |
|---|---|---|
| Penyebab atau alasan melakukan failover | Kendala teknis di server utama atau melakukan maintenance rutin | Keadaan bencana tak terduga seperti kebakaran, banjir, gempa bumi |
| Lokasi server | Dalam satu data center atau berdekatan dengan jaringan kecepatan tinggi | Data center terpisah di lokasi aman bencana |

| Aspek Pembeda | High Availability | Disaster Recovery |
|-------------------------------------|--|---|
| Proses Failover | Cepat dengan metode otomatis atau semi manual | Memerlukan prosedur formal tahap demi tahap, karena menyangkut perpindahan lokasi dan keselamatan jiwa. Proses failover dilakukan secara manual |
| Keterlibatan Tim | Team IT, khususnya administrator database dan jaringan | Semua tim baik IT maupun bisnis, karena menyangkut prosedur pemindahan lokasi dan penurunan kualitas layanan |
| Unjuk kerja sistem setelah failover | Unjuk kerja database harus sama dengan sebelum dipindahkan, sehingga diperlukan server yang setara | Unjuk kerja umumnya menurun karena kelengkapan server dan infrastruktur di lokasi DR lebih rendah dari server utama |

Secara umum, proses HA lebih menitikberatkan pada dukungan teknologi untuk mendukung ketersediaan sistem secara maksimum. Sedangkan DR menitikberatkan pada kesiapan proses dan *standard operating procedure* (SOP) yang melibatkan semua komponen dalam perusahaan. Proses HA juga memerlukan dukungan sumberdaya manusia untuk memastikan kelayakan teknis, namun lingkupnya lebih kecil dibandingkan DR.

Service Level Agreement

Salah satu poin penting yang sering penulis tanyakan kepada customer yang akan mengimplementasikan HA atau DR adalah *service level agreement* (SLA). Banyak organisasi ternyata belum memiliki definisi jelas mengenai SLA yang harus dipenuhi oleh sistem mereka. Sebaiknya setiap Manager IT memiliki kejelasan mengenai SLA dari tiap sistem yang dijalankan, karena pemilihan teknologi dan investasi dalam HA maupun DR sangat tergantung pada SLA yang harus dicapai.

SLA merupakan komitmen yang diberikan oleh penyedia layanan (departemen IT) untuk menjamin ketersediaan sistem dalam rentang waktu tertentu, biasanya dihitung per tahun. Sebagai contoh, tingkat SLA 99% berarti mengijinkan downtime [CA2] sebanyak 1 persen dari jumlah hari satu tahun. Jika satu tahun berjumlah 8760 jam, maka tingkat SLA 99% mengijinkan layanan tidak tersedia selama 87,6 jam per tahun atau sama dengan 3,65 hari. Berikut adalah tabel berbagai tingkat SLA dan berapa lama waktu downtime yang diijinkan per tahun.

| Tingkat SLA | Downtime/tahun |
|-------------|----------------|
| 99% | 87,6 jam |
| 99,9% | 8,76 jam |
| 99,99% | 52,56 menit |

Prasyarat SLA 99% masih cukup layak untuk dicapai, sedangkan 99.99% sangat sulit untuk dipenuhi karena hanya mengijinkan kejadian downtime selama 52,56 menit per tahun. Kondisi tersebut memerlukan perpaduan teknologi, SDM [CA3] pendukung, kondisi infrastruktur dan lingkungan yang kondusif.

Tingkat SLA 99% masih dapat dicapai dengan menggunakan teknologi log shipping atau database mirroring, sedangkan untuk mencapai 99,9% harus menggunakan SQL Server Failover Cluster atau Availability Groups.

Tingkat kesiapan Tim di dalam organisasi juga sangat menentukan dalam pemilihan teknologi. Penggunaan log shipping memerlukan banyak keterlibatan administrator dan tim pengembang. Teknologi SQL Server cluster dan availability groups tidak terlalu memerlukan keterlibatan tim pengembang aplikasi karena memiliki fasilitas listener[CA4] . Kondisi tersebut menyebabkan banyak dua teknologi terakhir lebih banyak dipilih karena memberikan tingkat jaminan lebih tinggi walaupun SLA yang ingin dicapai masih dibawah 99%.

Planned dan Unplanned downtime

Salah satu aspek yang perlu diperhitungkan dalam menetapkan SLA adalah bagaimana memasukkan kejadian planned downtime dan unplanned downtime.

- *Planned downtime*; merupakan downtime terencana karena adanya kegiatan perawatan dan upgrade sistem seperti penambahan memory, penggantian hardware, instalasi service pack, patches, dan hotfix. Proses tersebut harus diberitahukan ke semua pihak yang berkepentingan, umumnya 2 minggu sebelum pelaksanaan pekerjaan.
- *Unplanned downtime*; adalah kejadian downtime tidak terencana karena masalah teknis maupun non teknis. Kondisi yang sering terjadi adalah kerusakan perangkat keras, file corruption pada database, permasalahan jaringan, maupun tidak tersedianya aliran listrik.

Masalah yang sering diperdebatkan adalah apakah planned downtime harus disertakan dalam perhitungan SLA. Jika planned downtime dimasukkan dalam SLA, maka tingkat ketersediaan 99% sudah termasuk kegiatan terencana seperti perawatan sistem dan upgrade baik perangkat keras maupun lunak. Menyediakan 87,6 jam per tahun untuk kejadian terencana maupun tak terduga merupakan kondisi yang cukup menantang untuk dicapai.

Sebagai contoh, jadwal berikut biasa dilakukan tim IT untuk melakukan perawatan sistem:

1. Installasi patches dan hotfix sebanyak 3x per tahun dengan durasi 6 jam/kejadian = total 18 jam
2. Upgrade memory server 1x per tahun dengan durasi 6 jam = total 24 jam
3. Testing failover rutin 2x per tahun dengan durasi 3 jam/test = total 6 jam
4. Testing DR rutin 1x per tahun dengan durasi 6 jam/test = total 6 jam

Total waktu yang dibutuhkan untuk kegiatan terencana adalah 54 jam per tahun. Dengan demikian hanya tersisa 33,6 jam per tahun untuk unplanned downtime. Kondisi tersebut masih dapat dipenuhi dengan perencanaan dan pemilihan teknologi yang tepat, misalnya:

- Proses upgrade patches dan service pack tidak memerlukan downtime, dengan menggunakan teknologi SQL Server Availability Groups atau Failover Cluster Instance. Proses *patching* dilakukan secara bergiliran pada tiap node.
- Testing failover juga tidak memerlukan downtime, karena availability groups menyediakan fitur planned manual failover tanpa harus kehilangan data.

Keputusan untuk memasukkan atau mengecualikan planned downtime dalam SLA sebenarnya dikembalikan pada kebijakan organisasi masing-masing. Penulis lebih memilih untuk memasukkan dalam SLA, sehingga memotivasi tim untuk melakukan perencanaan yang tepat dalam kegiatan planned downtime. Tidak memasukkan aspek tersebut dalam SLA dapat berdampak negative karena tidak ada batasan pasti berapa kali dan total durasi yang dibutuhkan untuk proses tersebut.

Jika 99% dirasa masih cukup sulit dicapai, disarankan untuk menurunkan tingkat SLA menjadi 98% sehingga mengijinkan downtime total sebanyak 175,2 jam atau sama dengan 7,3 hari per tahun.

Penetapan SLA harus disepakati semua pihak, baik tim IT maupun pengguna dari sisi bisnis. Penulis banyak menemui business user yang menginginkan SLA 99%, padahal sistem tersebut tidak digunakan di saat akhir pekan (Sabtu dan Minggu) dan tidak terdapat kegiatan operasional setelah jam 8 malam. Kondisi tersebut sebenarnya mengijinkan penetapan SLA yang lebih rendah, selama masih dapat memenuhi jaminan bahwa sistem selalu tersedia pada jam kerja.

Patut selalu diingat bahwa penambahan angka dalam SLA selalu berdampak pada teknologi dan investasi yang harus disediakan.

Recovery Point Objective dan Recovery Time Objective

TO DO, in progress

Fitur SQL Server untuk mendukung HA dan DR

TO DO, in progress

[CA1]Pemindahan layanan dari satu server ke server lain

[CA2]Layanan tidak tersedia

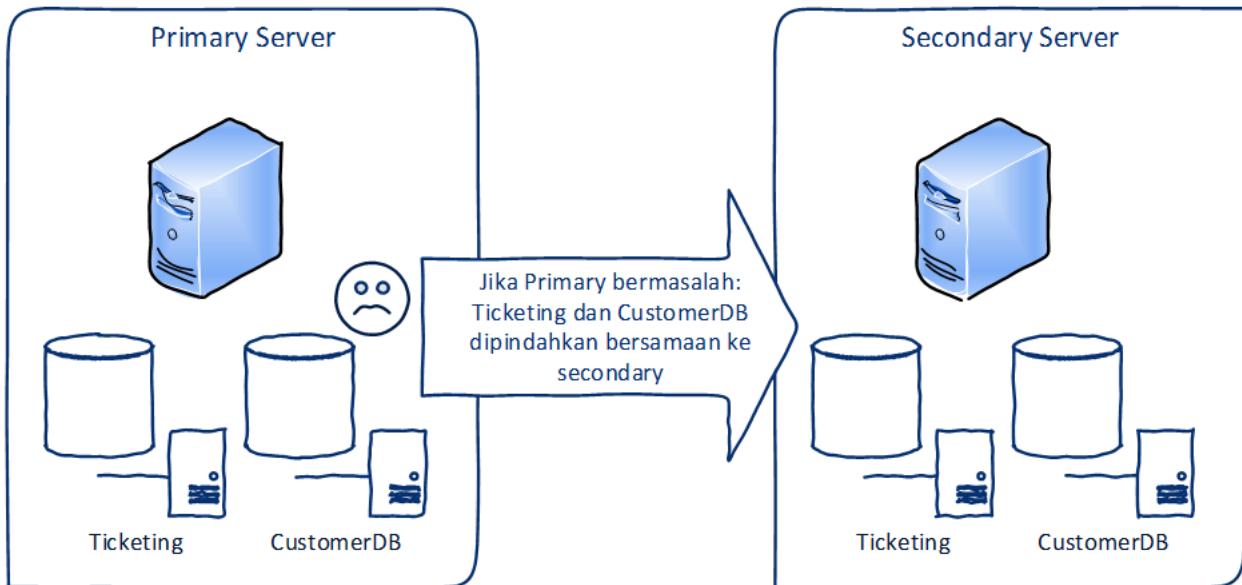
[CA3]Sumber daya manusia

[CA4]Akan dijelaskan di bab berikutnya

Bab 2. Arsitektur AlwaysOn Availability Groups

Fitur AlwaysOn availability groupss mulai dikenalkan sejak SQL Server 2012 sebagai teknologi untuk mendukung HA maupun DR. Teknologi tersebut menyatukan sekumpulan database (2 atau lebih) yang dikonfigurasi dalam satu kelompok *availability* sehingga dapat melakukan failover secara bersamaan jika database utama atau primary mengalami masalah.

Sebagai contoh, sebuah perusahaan memiliki aplikasi Online Booking yang selalu membutuhkan akses ke dua database secara bersamaan: Ticketing dan CustomerDB. Kedua database tersebut saling terkait, sehingga jika salah satu mengalami masalah maka aplikasi tidak dapat berjalan. Dengan demikian keduanya harus berada dalam satu availability groups untuk menjamin ketersediaan dan selalu melakukan *failover* secara bersamaan. Diagram berikut menjelaskan konsep tersebut yang merupakan ide dasar teknologi availability groups:



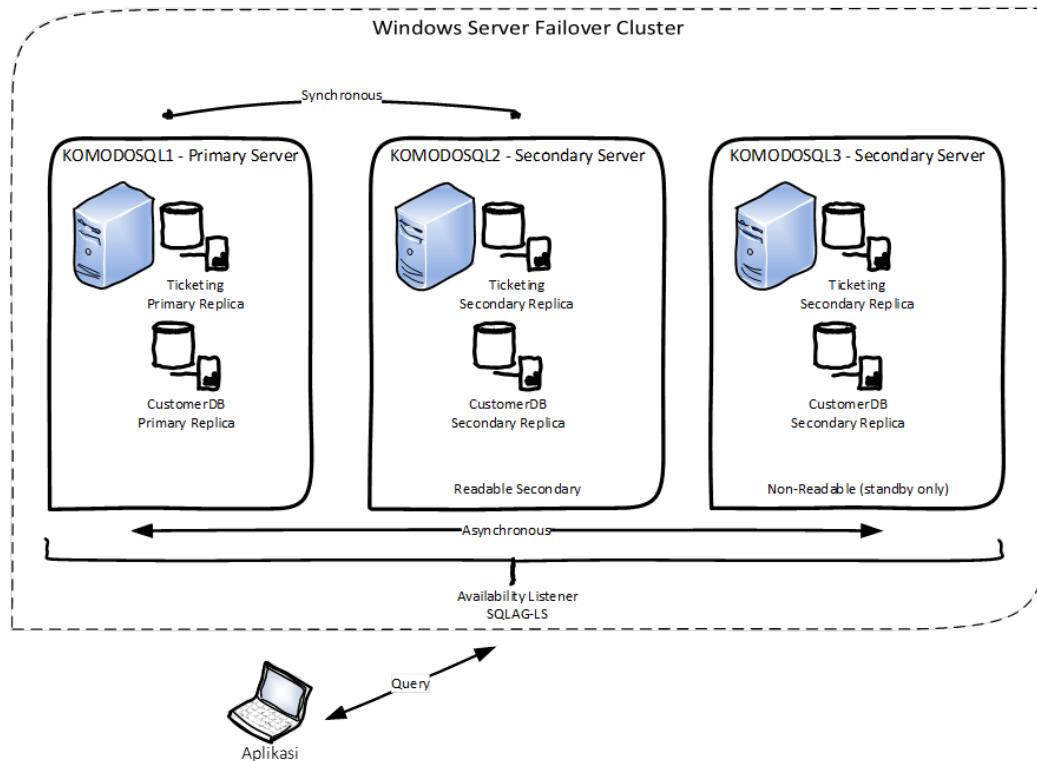
Dalam contoh diatas terdapat 2 database Ticketing dan CustomerDB yang berada dalam 1 availability groups dan ditempatkan dalam satu server database yang sama. Apabila salah satu database mengalami masalah sehingga tidak dapat diakses, maka kedua database tersebut dipindahkan (failover) ke secondary server secara bersama untuk menjamin bahwa keduanya selalu berada dalam satu kelompok.

Fitur Utama Availability Groups

Teknologi availability groups memiliki berbagai fitur untuk mendukung HA dan DR sehingga mampu menjaga ketersediaan database dengan failover secara otomatis maupun manual. Fitur-fitur penting availability groups dapat dijabarkan dalam beberapa point berikut:

- Setiap database yang menjadi anggota availability groups memiliki salinan database asli yang disebut **availability replica**. Replica tersebut merupakan salinan yang identik dengan sumber sehingga dapat digunakan sebagai sarana pemulihan apabila database asli mengalami masalah. Dalam satu konfigurasi availability groups terdapat dua jenis replica yaitu primary replica yang bersifat read/write dan secondary replica yang selalu bersifat readonly.
- **Primary replica** terletak di dalam SQL Server instance yang berada di primary server, sedangkan **secondary replica** berada di secondary server.
- Sinkronisasi antara primary dan secondary dapat berlangsung secara synchronous maupun asynchronous. Metode **synchronous** memungkinkan transfer data tanpa *data loss*, sedangkan **asynchronous** memiliki resiko kehilangan data.
- Semua mesin yang menjadi anggota availability groups harus dikonfigurasi dalam satu **Windows Server Failover Cluster** atau **WSFC**. Availability groups memerlukan fitur clustering walaupun SQL Server tidak dikonfigurasi sebagai Failover Cluster Instance
- Availability groups memiliki **availability listener** yang merupakan Virtual Network Name (VNN) dan didukung oleh infrastruktur WSFC. Fungsi listener adalah melayani permintaan koneksi dari klien kemudian mengarahkan ke database yang sesuai, baik primary maupun secondary. Dengan demikian klien tidak perlu mengetahui node mana yang sedang aktif dan berperan sebagai primary.
- Secondary replica dapat diaktifkan sebagai **readable secondary** untuk mengurangi beban kerja primary yang terkait dengan operasi readonly, misalnya untuk kegiatan reporting.
- Administrator juga dapat melakukan operasi **backup di secondary replica** sehingga tidak mengganggu kinerja primary jika proses backup berlangsung lama.

Diagram berikut memberikan gambaran umum fitur “fitur availability groupss sebagaimana dijelaskan dalam point diatas:



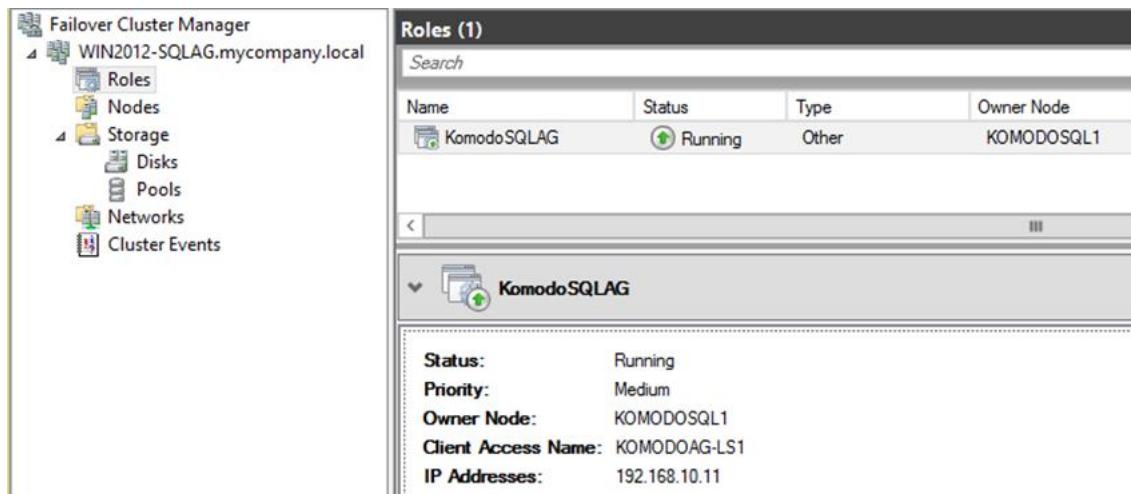
Availability groups dalam diagram tersebut terdiri dari 3 server dengan sinkronisasi data sebagai berikut:

- Metode pengiriman data synchronous antara KOMODOSQL1 dan KOMODOSQL2
- Metode pengiriman data asynchronous antara KOMODOSQL1 dan KOMODOSQL3

Konfigurasi tersebut menggunakan listener bernama SLAG-LS sehingga aplikasi klien cukup menyebutkan listener tersebut sebagai target server dan tidak perlu menyebutkan nama SQL Server instance.

Windows Server Failover Cluster (WSFC)

WSFC merupakan persyaratan utama dalam konfigurasi availability groups. Setiap availability groups yang dibuat harus diletakkan di dalam resource group di dalam cluster. Resource group tersebut dimonitor oleh WSFC cluster sehingga selalu dapat dipantau kondisi performa dan kelayakannya. Gambar dibawah memperlihatkan availability groups bernama KomodoSQLAG yang berada di dalam cluster sebagai salah satu Roles didalam WSFC.



Konfigurasi di dalam gambar diatas dapat dijelaskan sebagai berikut:

- Nama availability groups adalah KomodoSQLAG dan saat ini sedang berjalan di mesin KOMODOSQL1 sebagai primay server
- Nama Availability Listener adalah KOMODOAG-LS1 yang juga disebut sebagai Client Access Name dalam terminology WSFC
- Listener tersebut memiliki alamat IP sendiri yaitu 192.168.10.11. Alamat IP tersebut dikhususkan untuk listener dan berbeda dengan yang dimiliki oleh tiap server di dalam cluster

Semua mesin yang menjadi anggota availability groups harus dikonfigurasi sebagai node didalam WSFC sebelum SQL Server diinstall. SQL Server diinstall secara *stand alone* di tiap node yang selanjutnya dikonfigurasi sebagai availability groups. Penjelasan mengenai prosedur instalasi WSFC akan dibahas secara detail di Bab 3.

Peranan WSFC sebagai penyangga infrastruktur availability groups adalah menjalankan fungsi-fungsi berikut:[CA2]

- Sinkronisasi metadata antar node perlu dilakukan untuk mengetahui kondisi kesehatan dan performa primary maupun secondary.
- Menjalankan mekanisme voting untuk menentukan apakah layanan cluster dan availability group tetap tersedia jika terdapat node yang bermasalah. Layanan tetap tersedia apabila quorum dapat dicapai.

Availability Replica

Setiap database yang menjadi anggota availability groups merupakan availability replica. Sebagaimana dijelaskan sebelumnya, replica merupakan salinan database primary sehingga memiliki struktur dan isi sama dengan database asli. Dengan demikian terdapat 2 jenis replica dalam availability groupss:

- **Primary Replica**, yang merupakan database asli dan selalu bersifat read/write. Pada saat aplikasi melakukan koneksi read/write melalui listener, trafik akan selalu diarahkan ke primary replica. Dalam satu availability groups hanya diperbolehkan terdapat satu primary replica.
- **Secondary replica**, adalah database yang merupakan salinan dari primary replica dan selalu bersifat readonly. Availability groups mensyaratkan minimum harus terdapat satu secondary replica yang berfungsi sebagai target failover jika primary tidak dapat diakses. Secondary replica dapat dikonfigurasi menjadi readable secondary sehingga mampu menerima readonly trafik dari klien.
- SQL Server 2014 dapat memiliki 8 secondary replica
- SQL Server 2012 dibatasi sampai dengan 4 secondary replica

Jumlah secondary replica dalam availability groups ditentukan oleh jenis kebutuhan dan beban kerja yang akan ditangani oleh database. Berikut beberapa contoh yang sering ditemui di lapangan:

Konfigurasi kecil untuk HA

Topologi ini sering ditemui untuk aplikasi yang tidak memerlukan pendukung DR yang berfungsi sebagai pemulihan bencana. Konfigurasi ditekankan pada kebutuhan HA dalam satu lokasi jaringan.

- Terdiri dari 2 replica yaitu satu primary dan satu secondary di masing-masing server.
- Konfigurasi tersebut terdiri dari dua server dalam satu data center.
- Sinkronisasi antara primary dan secondary dilakukan secara synchronous

--

Konfigurasi sedang untuk HA dan readonly workload

Terdapat 3 replica dengan tiga server terdiri dari 1 primary dan 2 server untuk secondary.

- Satu primary replica untuk read/write yang merupakan database utama
- Satu secondary replica yang dikonfigurasi sebagai readable secondary untuk mengurangi beban primary terkait aplikasi reporting
- Satu secondary replica yang bersifat readonly, tetapi tidak dikonfigurasi sebagai readable secondary. Dengan demikian database tersebut tidak dapat diakses dan dialokasikan sebagai server standby
- Semua server tersebut terletak dalam satu data center

--

Konfigurasi Lanjut untuk HA dan DR

Konfigurasi ini mirip dengan konfigurasi sedang sebagaimana dijelaskan diatas, dengan tambahan secondary replica yang terletak di data center berbeda dan merupakan lokasi aman ketika terjadi bencana. Dengan demikian dibutuhkan 4 server sebagai berikut:

- Tiga server terletak di data center utama, dan 1 server di lokasi pemulihan bencana.

- Satu primary dan 2 secondary replica terletak di data center utama, sedangkan 1 secondary replica di lokasi aman.
- Sinkronisasi antara lokasi utama dengan lokasi DR dilakukan secara asynchronous karena umumnya kecepatan jaringan tidak sebagus di jaringan lokal.

Active Secondary

Active secondary merupakan fitur unggulan yang tidak terdapat di teknologi sebelumnya seperti database mirroring atau log shipping. Dengan kemampuan tersebut, secondary replica dapat dimanfaatkan untuk mengurangi beban kerja primary sehingga investasi server untuk secondary dapat lebih dimaksimalkan dan tidak sekedar menyediakan server standby.

Fungsi utama secondary replica adalah sebagai database pendukung yang selalu siap untuk dinaikkan fungsinya menjadi primary replica jika ternyata database utama tersebut bermasalah. Dengan demikian secondary bersifat standby dan menerima kiriman sinkronisasi data dari primary. Fungsi tersebut dapat ditambahkan sehingga secondary bersifat aktif yaitu dengan melakukan hal-hal berikut:

- **Readable secondary**; yaitu kemampuan untuk menerima akses data readonly yang umumnya digunakan untuk aplikasi reporting dan analytic.
- **Backup database**; yaitu kemampuan untuk melakukan backup di database secondary sehingga proses backup tidak mengganggu database primary.[CA3]

Kedua fitur tersebut dapat digunakan untuk mengurangi beban kerja primary sehingga unjuk kerja database lebih merata. Server secondary juga dapat dimanfaatkan untuk kegiatan readonly dan backup sehingga tidak bersifat *idle*.

Readable Secondary

Secondary replica selalu bersifat readonly tetapi tidak berarti selalu dapat melayani koneksi dari klien. Kemampuan tersebut tergantung pada konfigurasi yang diterapkan, apakah fungsi readable secondary diaktifkan atau tidak. Fungsi tersebut dapat diaktifkan dengan mililih Yes atau read-intent only pada konfigurasi replica sebagaimana gambar berikut:

| Availability Replicas | | | | | | | |
|-----------------------|---------------|-------------------|---------------|-----------------------------|-------------------------------|--|--|
| Server Instance | Role | Availability Mode | Failover Mode | Connections in Primary Role | Readable Secondary | | |
| KOMODOSQL1\SQL2... | Primary | Synchron... ▼ | Autom... ▼ | Allow read/write c... ▼ | Yes ▼ | | |
| KOMODOSQL2\SQL2... | Secon... ▼ | Synchron... ▼ | Autom... ▼ | Allow all connectio... ▼ | Read-intent only ▼ | | |
| | | | | | No Read-intent only Yes | | |

Fungsi readable secondary adalah untuk mengurangi beban primary dengan mengarahkan trafik readonly ke secondary. Dengan demikian primary dapat digunakan untuk operasi read/write yang merupakan fungsi utama dalam transaksi. Penerapan umum dari fitur ini adalah menggunakan secondary replica sebagai sumber data untuk aplikasi reporting dan dashboard, sehingga primary dapat difokuskan untuk melayani transaksi data yang bersifat read-write.

Penjelasan lebih lanjut mengenai hal ini dapat dibaca di Bab 5.

Operasi Backup di Secondary Replica

Secondary replica juga dapat dimanfaatkan untuk melakukan operasi backup, dan hasil backup selalu kompatibel dengan database primary. Backup tersebut dapat dilakukan dari secondary manapun, tetapi sebaiknya selalu dilakukan dari replica yang menggunakan mode synchronous sehingga integritas data lebih terjamin.

- Jenis backup yang dapat dilakukan adalah Full dan Log backup
- Differential backup tidak dapat dilakukan dari secondary

Operasi backup untuk database berukuran besar dapat berlangsung sangat lama dan mengganggu aktivitas harian. Penggunaan secondary replica untuk backup dapat mengurangi hal tersebut, karena dilakukan di secondary yang memang tidak diakses secara intensif oleh aplikasi. Penjelasan lebih lanjut mengenai tata cara melakukan backup dari secondary replica akan dijelaskan di Bab 9.

Availability Mode

Konfigurasi ini menentukan metode sinkronisasi data yang terjadi antara primary dan secondary pada tiap replica. Pilihan availability mode menentukan apakah sinkronisasi dilakukan secara synchronous atau asynchronous yang dijelaskan dalam point-point berikut.

Mode Asynchronous-commit

Metode asynchronous berarti primary replica melakukan *committed transaction* tanpa menunggu infomasi kondisi transaksi di secondary. Dengan demikian transaksi di primary replica akan mencapai posisi *committed* walaupun secondary replica belum menuliskan transaksi tersebut di file log. Kondisi ini memungkinkan proses sinkronisasi menjadi lebih cepat dengan resiko terjadi *data loss* karena adanya perbedaan data antara primary dan secondary.

Asynchronous-commit sangat sesuai untuk diterapkan dalam scenario disaster recovery yang umumnya perlu melakukan sinkronisasi data antara lokasi yang berjauhan dengan bandwith jaringan terbatas. Resiko kehilangan data juga dapat ditolerir selama masih memenuhi standar RPO dan RTO yang telah ditetapkan.

Mode Synchronous-commit

Berbeda dengan asynchronous-commit mode, pada metode ini primary replica harus menunggu pemberitahuan dari secondary replica bahwa transaksi terkait telah selesai dituliskan di file log sebelum akhirnya primary melanjutkan transaksi ke proses committed. Dengan demikian integritas data antara primary dan secondary lebih terjaga daripada proses asynchronous. Perlindungan ini menyebabkan proses transaksi menjadi lebih lama dan memerlukan kondisi jaringan yang prima untuk menghindari terjadinya *high latency*. Jumlah maksimum yang dapat dikonfigurasi sebagai synchronous-commit adalah 3 replica, termasuk primary replica.

Proses synchronous-commit umumnya digunakan dalam konfigurasi high availability yang memungkinkan proteksi data dengan jaringan yang cukup terjamin dalam satu data center. Kebutuhan high availability biasanya juga mensyaratkan automatic failover yang hanya dapat dipenuhi dalam kondisi synchronous-commit.

Metode Failover

Failover merupakan proses pemindahan database primary dari satu node ke node lainnya karena database yang saat ini bertindak sebagai primary mengalami masalah sehingga tidak dapat diakses. Failover terjadi dari primary replica ke secondary replica, sehingga database yang saat ini menjadi secondary dinaikkan statusnya sebagai primary replica.

Proses failover merupakan fitur yang sangat penting dalam availability groups karena menyediakan kemampuan pemulihan layanan database sehingga aplikasi klien tetap dapat berjalan seperti biasa. Terdapat dua kategori failover yaitu automatic dan manual.

Automatic failover (*tanpa data loss*)

Proses pemindahan pada automatic failover terjadi secara otomatis jika primary replica mengalami masalah dan tidak dapat diakses. Untuk dapat mencapai kondisi tersebut baik primary maupun secondary harus berada dalam konfigurasi Synchronous-commit. Secondary replica yang menjadi target harus berada dalam kondisi synchronized, yang berarti layak dipromosikan sebagai primary replica.

Planned manual failover (*tanpa data loss*)

Proses pemindahan dilakukan menggunakan SQL Server Management Studio atau T-SQL sehingga posisi primary berpindah dari satu node ke node yang lainnya. Persyaratan metode ini sama dengan automatic failover yaitu replica harus berada dalam mode synchronous-commit sehingga tidak terjadi *data loss*. Metode ini umumnya dilakukan untuk kegiatan testing, maintenance atau upgrade system yang memerlukan pemindahan layanan secara bergiliran.

Forced manual failover (dengan *data loss*)**

Metode ini meruapakan failover manual dengan tambahan forced operation. Proses forced failover perlu dilakukan karena database berada dalam konfigurasi asynchronous-commit sehingga terdapat kemungkinan kehilangan data jika terjadi failover. Perlu diingat kembali bahwa replica yang berada dalam mode asybnchronous-commit tidak dapat melakukan automatic failover sehingga prosesnya harus dilakukan secara manual.

Operasi ini umumnya dilakukan dalam kegiatan disaster recovery, dimana kondisi database berada dalam posisi asynchronous dan kehilangan data masih dapat diterima.

Gambar berikut menjelaskan konfigurasi failover untuk tiap replica berikut dengan metode sinkronisasi data yang terkait.

| Availability Replicas | | | |
|-----------------------|-----------|--------------------|---------------|
| Server Instance | Role | Availability Mode | Failover Mode |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | Automatic |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | Automatic |

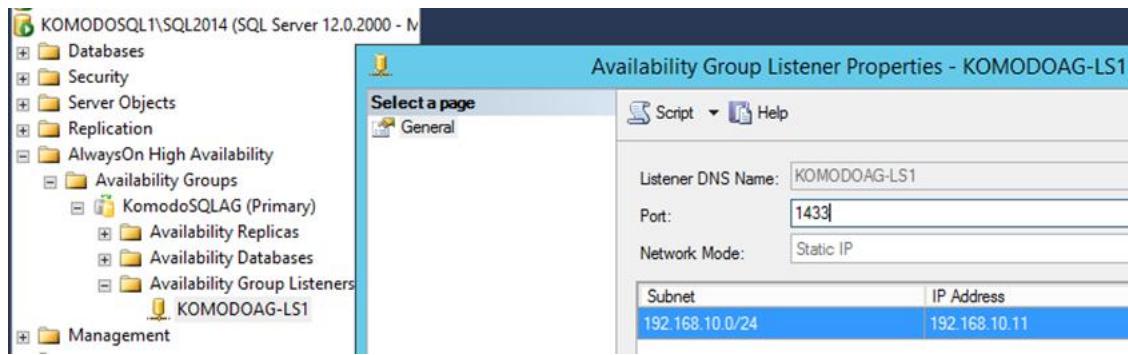
Automatic
Manual

Availability Listener

Avalability group listener merupakan virtual network name (VNN) dimana klien dapat melakukan koneksi ke database tanpa perlu mengetahui nama SQL Server instance. Klien dapat mengakses primary maupun secondary replica tergantung pada konfigurasi connection string dan readable secondary si sisi server.

Manfaat utama listener adalah aplikasi klien tidak perlu melakukan perubahan connection string ketika terjadi failover di sisi server. Koneksi langsung diarahkan ke database yang dituju sesuai dengan routing yang telah ditetapkan. Dengan demikian aplikasi tetap dapat berjalan tanpa gangguan berarti.

Gambar berikut menunjukkan lokasi dan konfigurasi listener beranama KOMODOAD- LS1 di dalam availability group.



Pembahasan lebih lanjut mengenai listener dan konektivitas aplikasi dapat dibaca pada Bab 5.

[CA2]TODO: perlu dijelaskan lebih detail

[CA3]TO-DO: Perlu ditambahkan overview DBCC di secondary

Bab 3. Konfigurasi Windows Cluster

Availability groups membutuhkan layanan Microsoft Failover Clustering dan TCP/IP sehingga WSFC harus disiapkan terlebih dahulu sebelum SQL Server diinstall di tiap node. Bagian ini menjelaskan langkah-langkah konfigurasi Windows Cluster untuk mendukung availability groups. Karena tujuan konfigurasi hanya untuk mendukung fitur tersebut, maka buku ini tidak menjelaskan aspek-aspek tingkat lanjut dalam WSFC yang tidak terkait dengan availability groups.

Konfigurasi WSFC yang dijelaskan dalam buku ini memberikan penekanan pada hal-hal berikut:

- Konfigurasi cluster tidak menggunakan *shared storage* karena availability groups tidak mewajibkan penggunaan *shared storage*.
- Quorum yang digunakan adalah *node majority* dengan jumlah node 3 server. Pembahasan lebih lanjut mengenai quorum dapat dibaca di Bab 7.
- Proses *Cluster Validation* tidak memasukkan komponen validasi storage, karena di dalam availability groups masing-masing tiap node memiliki disk sendiri yang tidak menjadi bagian dari *cluster services*.

Cluster yang akan dikonfigurasi berjumlah tiga node dengan peran sebagai berikut:

- Mesin 1 bernama **KomodoSQL1** berperan sebagai primary server yang menjalankan primary replica
- Mesin 2 bernama **KomodoSQL2** berperan sebagai secondary untuk menjalankan secondary replica
- Mesin 3 bernama **KomodoSQL3** berperan sebagai secondary yang juga menjalankan secondary replica

Secondary replica yang dikonfigurasi berjumlah 2 server masing-masing dengan metode synchronous dan asynchronous. Buku ini tidak menjelaskan proses intalasi Windows Server 2012 R2 karena umumnya para administrator dan DBA telah terbiasa dengan proses tersebut.

Sebelum melanjutkan ke langkah-langkah berikutnya di Bab ini, berikut ini adalah beberapa hal penting yang perlu dilakukan di setiap server yang menjadi anggota cluster:

1) Ketiga mesin KomodoSQL1, KomodoSQL2, dan KomodoSQL3 tersebut harus bergabung ke Active Directory. Buku ini tidak menjelaskan langkah-langkah tersebut secara detail. Jika Anda belum terbiasa dengan hal tersebut, silakan membaca petunjuk di tautan berikut [http://technet.microsoft.com/en-us/library/cc770919\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc770919(v=ws.10).aspx)

.

- 2) Sebaiknya tersedia 2 kartu jaringan di masing-masing mesin, satu ditujukan untuk jaringan internal antar node (*heart beat*), dan yang lainnya untuk komunikasi eksternal dengan domain controller dan mesin klien. Fitur clustering memang sudah tidak mewajibkan hal tersebut, namun memiliki 2 kartu jaringan dalam satu mesin sangat disarankan untuk menambah performa dan ketersediaan sistem secara keseluruhan.
- 3) Akun untuk administrator cluster dan _service account _SQL Server telah dibuat di Active Directory dan diberikan hak yang sesuai sebagaimana dijelaskan bagian berikutnya.

Persiapan Windows Server 2012 R2

Bagian memberikan ini panduan untuk mempersiapkan Windows Server 2012 R2 sehingga dapat digunakan untuk menjalankan WSFC dan availability groups. Setelah Windows selesai diinstall, beberapa persiapan berikut perlu dilakukan di tiap mesin yang digunakan di dalam cluster:

- Instalasi .NET Framework 3.5
- Konfigurasi Role dan Features Windows Server 2012
- Menyiapkan account untuk layanan SQL Server dan administrator

Anda perlu login ke tiap server sebagai *local administrator* untuk menjalankan semua prosedur tersebut.

Menyiapkan Role dan Features

Windows Server 2012 memerlukan beberapa Roles dan Features untuk mendukung fungsi clustering dan SQL Server. Berikut adalah Roles dan Features yang perlu disertakan di setiap mesin yang menjadi anggota cluster:

Roles:

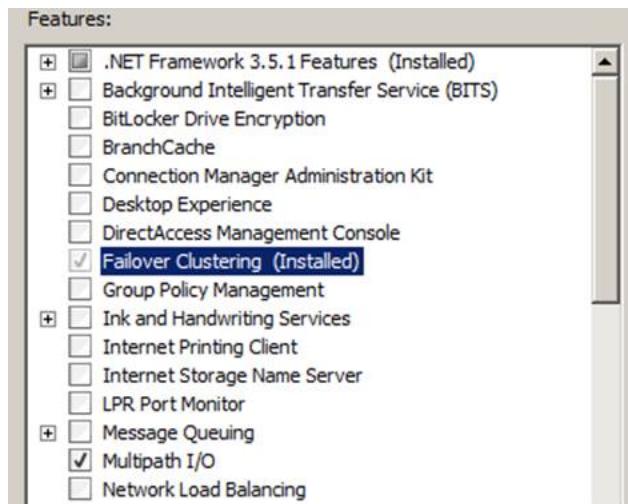
- Application Server

Features:

- .Net Framework 3.5
- Failover Clustering

Berikut adalah prosedur untuk mempersiapkan Roles dan Features Windows Server 2012

- 1) Login ke KomodoSQL1 sebagai *local administrator* kemudian buka Server Manager > Features > lalu pilih Add Features. Tambahkan semua fitur yang diperlukan untuk mendukung failover cluster. Fitur yang paling penting adalah **Failover Clustering**.



- 2) Klik Next dan mengkonfirmasi penyelesaian instalasi fitur
- 3) Buka Server Manager, pilih Roles > lalu pilih Add Roles. Tambahkan semua Roles yang diperlukan sebagai berikut:



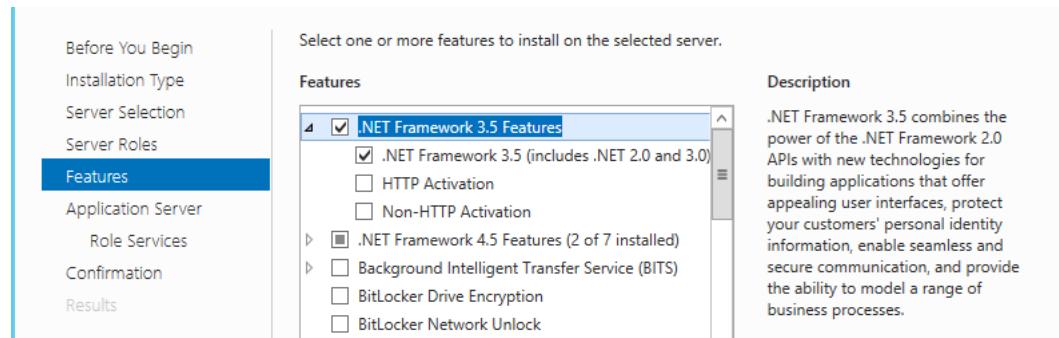
- 4) Klik Next dan mengkonfirmasi instalasi fitur.
- 5) Ulangi semua langkah tersebut untuk KomodoSQL2 dan KomodoSQL3.

Instalasi .NET Framework 3.5

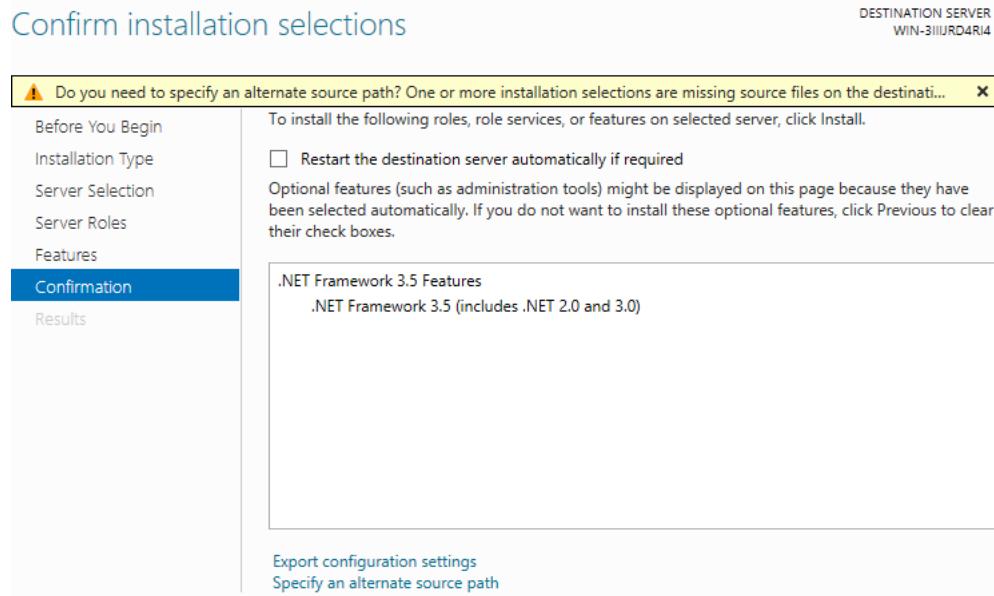
SQL Server membutuhkan .Net Framework 3.5 namun fitur ini tidak tersedia di Windows Server 2012 R2 secara default. Instalasi SQL server pada Windows Server 2012 akan gagal dengan pesan berikut:



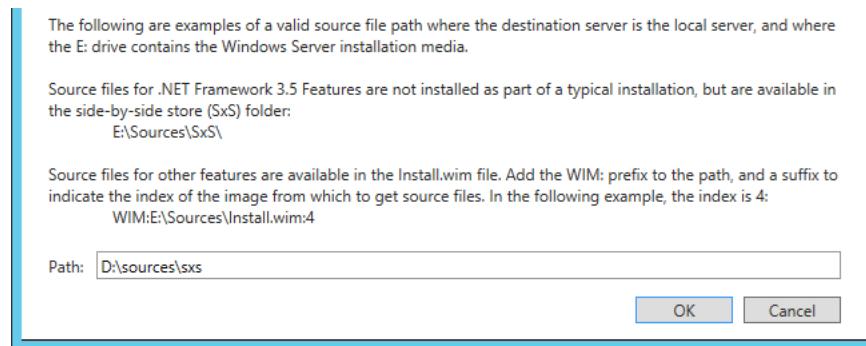
Untuk menginstall komponen tersebut diperlukan CD sumber Windows Server 2012 R2. Buka Server Manager dan jalankan Add Roles and Features Wizard sehingga sampai ke layar berikut:



Pilih .NET Framework 3.5 Features dan klik Next, selanjutnya pada layar konfirmasi akan menampilkan pilihan lokasi *source path*. Klik **Specify an alternative source path** di bagian bawah untuk menunjukkan lokasi sumber.



Pada layar berikutnya, isikan lokasi sumber sesuai dengan lokasi CD. Pada contoh dibawah, CD terletak di drive D sehingga lokasi sumber adalah D:\sources\sxs.



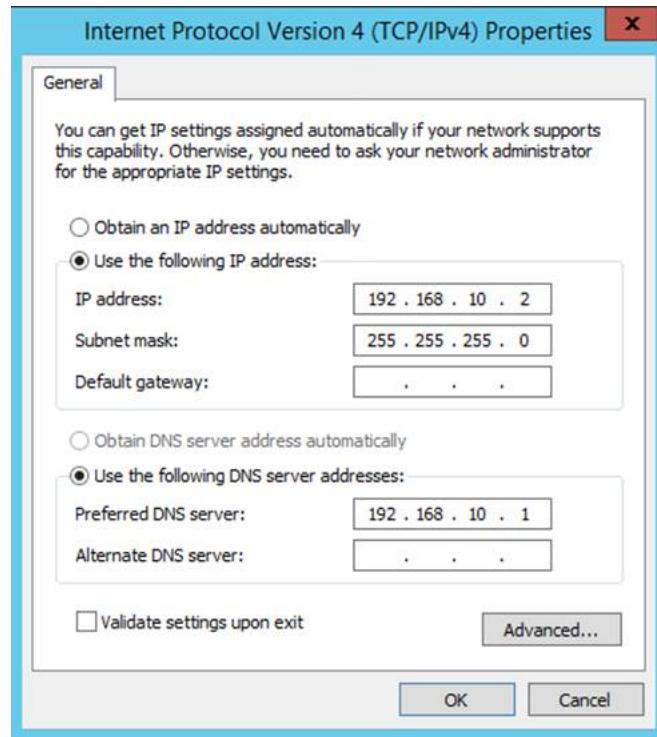
Klik OK dan lanjutkan proses instalasi dan menginstal .Net Framework 3.5. Ulangi semua langkah tersebut untuk KomodoSQL2 dan KomodoSQL3.

Konfigurasi Kartu Jaringan

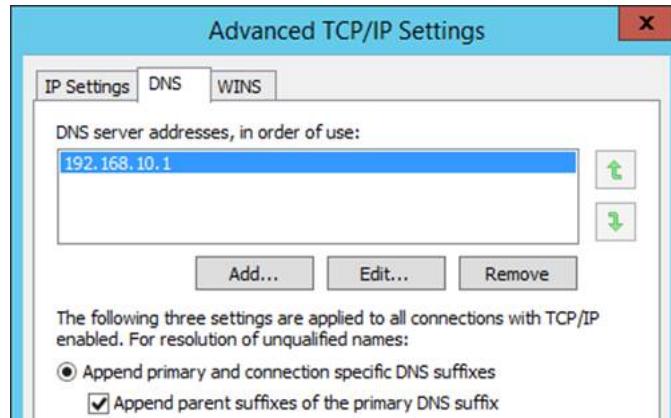
Konfigurasi Public Network

Public network digunakan untuk komunikasi eksternal antara node bersangkutan dengan domain controller, klien, dan aplikasi lain.

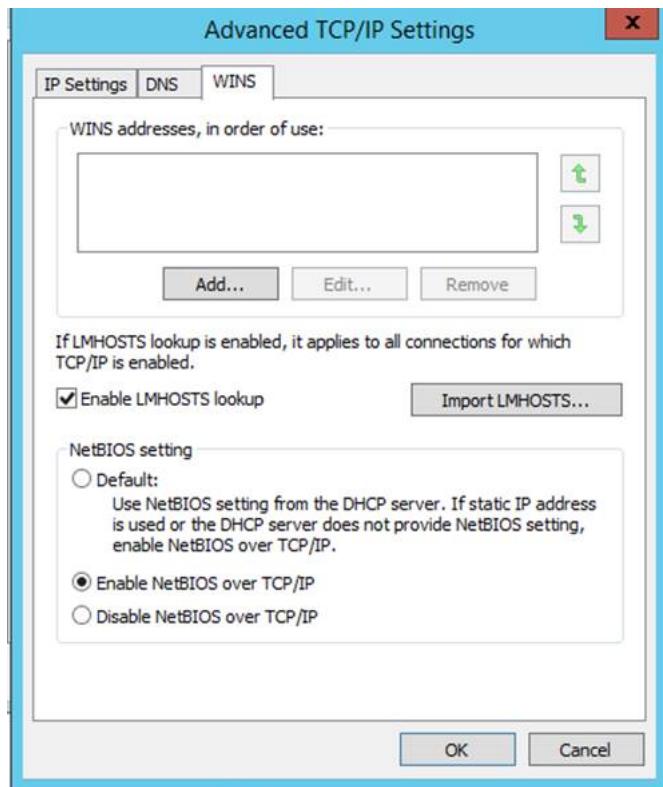
- 1) Login ke KomodoSQL1 yang akan berfungsi sebagai primary node sebagai *local administrator*.
- 2) Buka konfigurasi kartu jaringan yang akan digunakan untuk jaringan publik. IP address yang diisikan harus dapat terkoneksi ke domain controller dan dapat diakses oleh aplikasi klien. Gunakan alamat IP berikut:



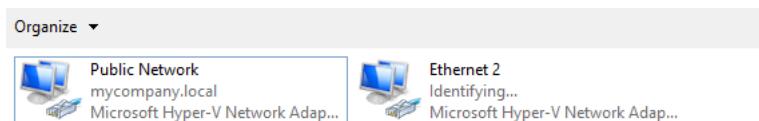
3) Klik tombol Advanced untuk mengecek bahwa konfigurasi DNS telah diaktifkan sebagai berikut:



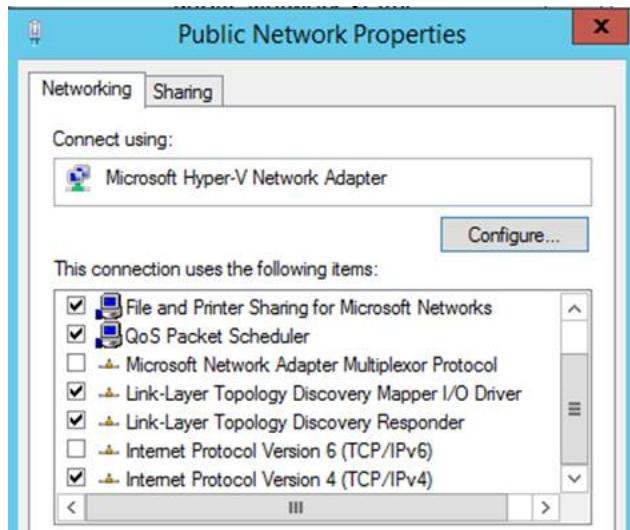
4) Public network memerlukan NETBIOS untuk mendukung komunikasi jaringan. Buka tab WINS dan pastikan bahwa NetBIOS diaktifkan:



- 5) Klik OK untuk menyimpan konfigurasi alamat IP tersebut.
- 6) Langkah berikutnya adalah mengubah nama kartu jaringan sehingga lebih mudah dikenali. Buka Control Panel > Network and Sharing Center > View Adapter Settings. Klik kanan kartu jaringan yang baru saja dikonfigurasi sebagai public network, Pilih **Rename** dan ubah namanya ke **Public Network**.



- 7) Lepaskan *binding* yang tidak diperlukan dari kartu jaringan tersebut. Klik kanan Public Network > pilih **Properties**. Protokol IPv6 dapat dilepaskan dari binding karena tidak digunakan dalam konfigurasi tersebut.



8) Langkah berikutnya adalah mengubah mode kecepatan jaringan dari Auto ke Full 1000 Mbps duplex. Tujuannya untuk memaksimalkan kecepatan kartu jaringan. Jika kartu Anda memiliki kecepatan lebih tinggi, maka pilih kecepatan yang sesuai. Dalam contoh ini digunakan kartu Gigabit LAN yang memiliki kecepatan sampai dengan 1000 Mbps.

9) Klik kanan kartu jaringan Public Network > pilih **Properties**, kemudian klik **Configure**. Pilih tab **Advanced**. Ubah kecepatan link dari **Auto 1000 Mbps Full Duplex**.

Konfigurasi tersebut mengasumsikan Anda menggunakan Gigabit Switch yang memiliki kecepatan 1000 Mbps. Performa jaringan sangat berpengaruh pada kinerja database, sehingga tidak disarankan menggunakan switch dengan kecepatan dibawah 1000 Mbps.

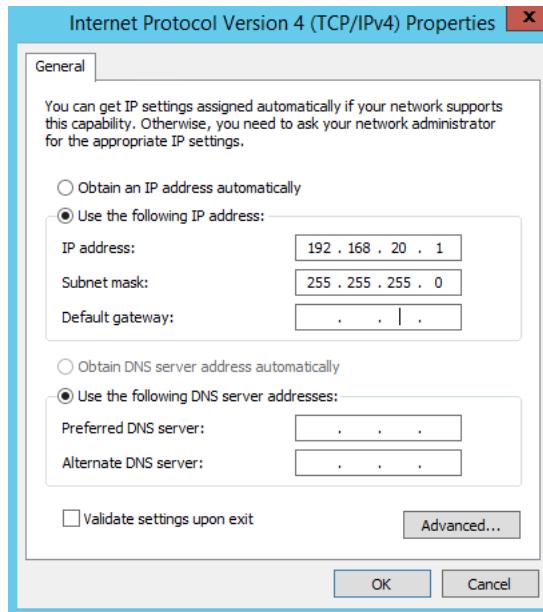
10) Ulangi langkah 1 sampai 8 untuk mengkonfigurasi jaringan publik pada mesin **KomodoSQL2**** dan **KomodoSQL3**** yang digunakan sebagai secondary node. Alamat IP yang digunakan adalah sebagai berikut:

|| KomodoSQL2 | KomodoSQL3 | | | **Alamat IP** | 192.168.10.3 | 192.168.10.4 | | **Subnet mask** | 255.255.255.0 | 255.255.255.0 | |**Default Gateway**||| |**DNS** | 192.168.10.1|192.168.10.1|

Konfigurasi Private Network

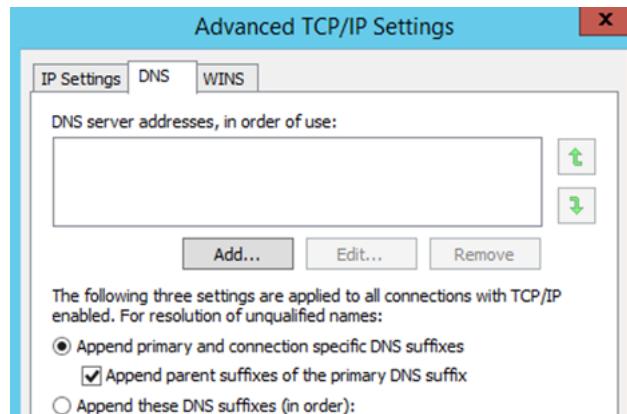
Private network merupakan jaringan internal yang digunakan sebagai untuk komunikasi antar node. Jalur ini berfungsi sebagai *heartbeat* yang mengirimkan informasi internal antar node dalam cluster.

- 1) Login ke server **KomodoSQL1** menggunakan akun dengan hak administrator lokal.
- 2) Buka konfigurasi kartu jaringan yang digunakan untuk jaringan internal kemudian gunakan alamat IP berikut:

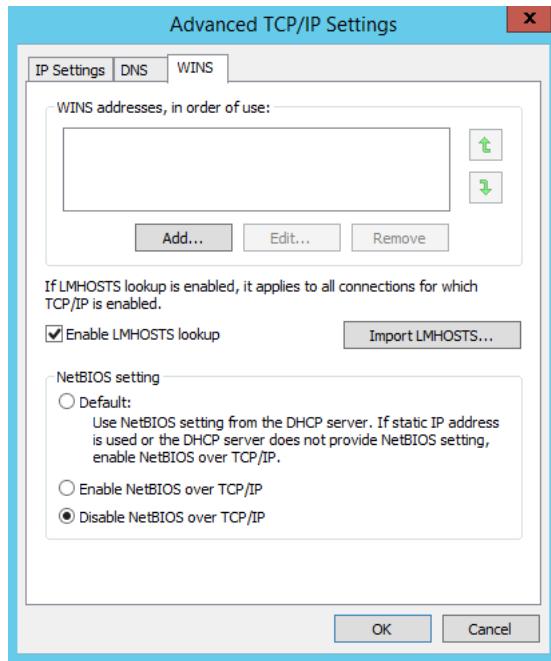


Kosongkan default gateway dan DNS karena komunikasi antara node dilakukan secara *private* sehingga kedua isian tersebut tidak diperlukan.

3) Selanjutnya klik tombol Advanced dan pastikan bahwa DNS dalam kondisi kosong sebagai berikut:

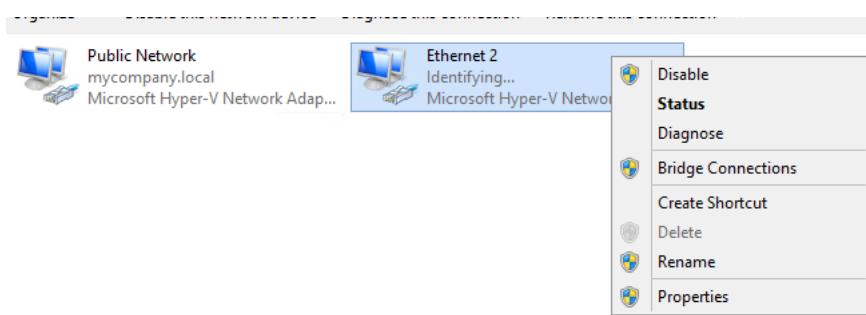


4) Buka tab WINS dan pastikan bahwa NetBIOS tidak diaktifkan sebagaimana berikut:

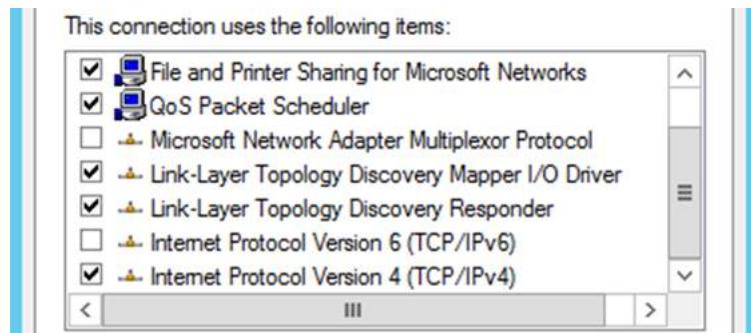


5) Klik OK untuk menyimpan konfigurasi alamat IP.

6) Ubah nama kartu jaringan sehingga lebih mudah dikenali. Buka Control Panel > Network and Sharing Center > View Adapter Settings. Klik kanan kartu tersebut dan pilih **Rename** untuk mengubah nama ke **Private Network**.



7) Sebagaimana konfigurasi untuk Public Network, Anda perlu menon-aktifkan protokol yang tidak dipakai oleh kartu jaringan tersebut sebagai berikut:



8) Langkah terakhir adalah menyesuaikan kecepatan jaringan dari Auto ke Full 1000 Mbps duplex. Klik kanan Private Network > pilih **Properties** > **Configure**. Pilih tab **Advanced** untuk mengubah kecepatan dari **Auto** **ke 1000 Mbps Full Duplex.**

9) Ulangi langkah 1 sampai 8 untuk mengkonfigurasi Private Network pada mesin **KomodoSQL2** dan **KomodoSQL3**. Alamat IP masing-masing adalah sebagai berikut:

- KomodoSQL2 adalah 192.168.20.2
- KomodoSQL3 adalah 192.168.20.3

Menyiapkan Akun Layanan

Beberapa akun layanan (*service account*) perlu dibuat di dalam Active Directory untuk keperluan Windows Cluster dan availability groups. Setiap akun memiliki hak terpisah untuk menjaga keamanan. Daftar pengguna berikut adalah *domain account* yang harus disiapkan di Active Directory:

1) **SQLCIsAdmin**; merupakan administrator Windows dan digunakan selama melakukan instalasi Windows cluster. Akun tersebut memiliki hak sebagai berikut:

- Administrator lokal di tiap mesin yang menjadi anggota cluster
- Hak untuk “Create computer objects” di dalam Active Directory

2) **SQLService**; adalah akun yang digunakan sebagai *service account* untuk layanan SQL Server. Dengan demikian layanan Database Engine berjalan atas nama akun tersebut. Hak yang diberikan di tiap server adalah sebagai berikut:

- Adjust memory quotas for a process
- Log on as a service
- Perform volume maintenance tasks

3) **SQLAgent**; adalah akun yang digunakan sebagai *service account* untuk SQL Agent. Akun ini memiliki hak yang sama dengan SQLService sebagaimana disebutkan diatas.

Untuk menjaga keamanan server dan layanan database, perlu diingat hal-hal berikut:

- Semua akun yang digunakan untuk instalasi dan *service account* tidak boleh menjadi anggota Domain Administrators.
- Akun SQLService sebaiknya tidak menjadi anggota administrator local

Kenyataan di banyak customer memberikan bukti mengkhawatirkan mengenai konfigurasi akun untuk cluster dan SQL Server. Akun-akun tersebut diberikan hak yang tidak perlu, misalnya sebagai Domain Administrators dan *local Administrators* sehingga membahayakan keamanan server. Staff IT melakukan hal tersebut karena alasan kepraktisan sehingga tidak perlu melakukan pemberian hak kepada setiap akun.

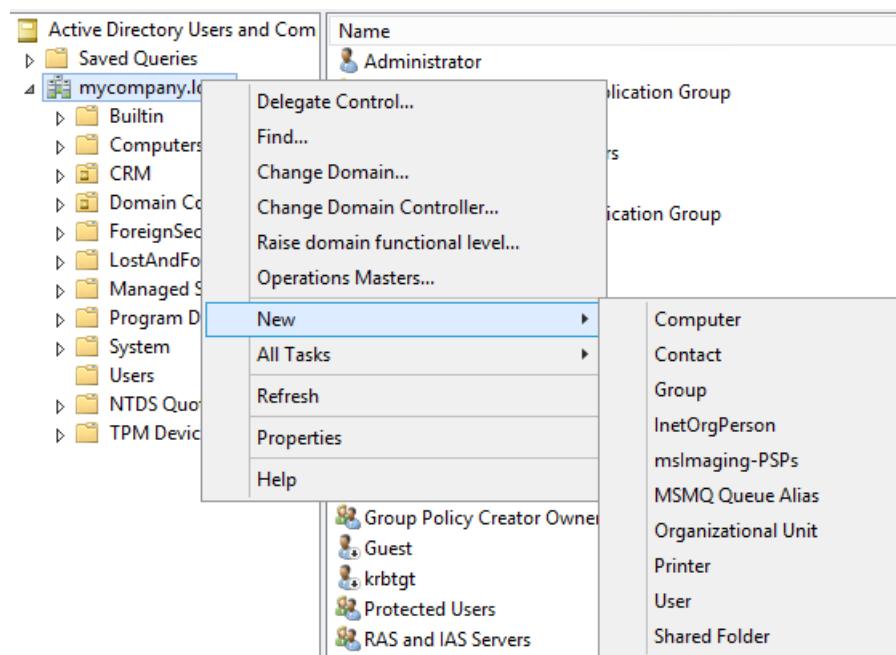
Semua latihan di dalam buku ini hanya menggunakan satu akun **SQLService** sebagai *service account* untuk Database Engine dan SQL Agent. Jika Anda melakukan konfigurasi server untuk lingkungan *production*, sebaiknya akun untuk kedua layanan tersebut dipisahkan. Demikian juga jika terdapat layanan lain seperti Reporting Services dan Analysis Services.

Bagian berikutnya menjelaskan proses pembuatan akun tersebut dan pemberian hak sesuai dengan best practice kemanan server.

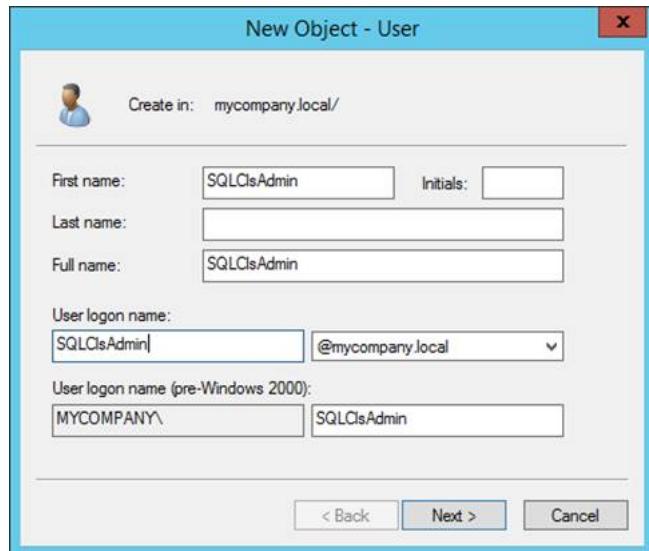
Membuat Akun di Active Directory

Lakukan langkah-langkah berikut untuk menyiapkan akun layanan dan memberikan hak yang dibutuhkan

1) Login ke Domain Controller untuk sebagai domain administrator untuk membuat akun SQL-ClAdmin dan SQLService. Buka Active Directory Users and Computer > Klik kanan > New > User

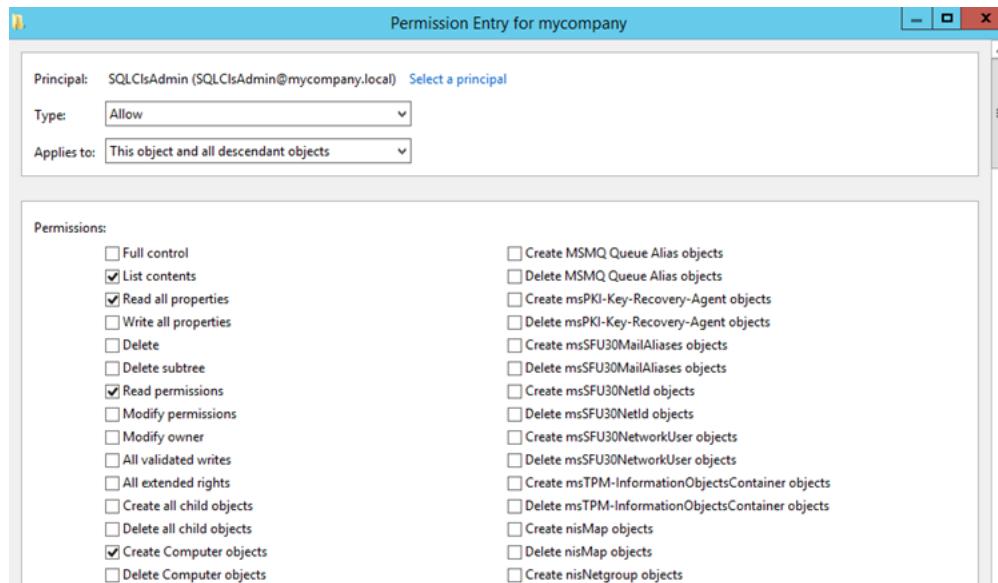


2) Isikan data berikut untuk membuat akun SQLCIsAdmin:



Ulangi langkah-langkah tersebut untuk membuat akun **SQLService**.

- 3) Akun SQLCIsAdmin perlu diberikan hak **Create computer objects**. Buka Active Directory Users and Computers > View > Advanced Features.
- 4) Klik kanan container Computers > Properties > Security > klik Advanced.
- 5) Tambahkan akun SQLCIsAdmin kemudian berikan hak **Create computer objects** dan **Read all properties** sebagaimana gambar dibawah.



- 6) Klik OK dan tutup layar untuk menyimpan konfigurasi tersebut.

Jika Anda belum belakukan konfigurasi tersebut dengan benar, maka proses pembuatan cluster akan

menampilkan pesan kesalahan sebagai berikut:

```
Check whether the computer object KomodoSQL2 for node KomodoSQL2.mycompany.local exists in the domain. Domain controller \\MAHAMERU.mycompany.local.
Creating computer object in organizational unit CN=Computers,DC=mycompany,DC=local where node KomodoSQL2.mycompany.local exists.
Create computer object WIN2012-SQLAG on domain controller \\MAHAMERU.mycompany.local in organizational unit CN=Computers,DC=mycompany,DC=local. A constraint violation occurred.
Unable to successfully cleanup.
```

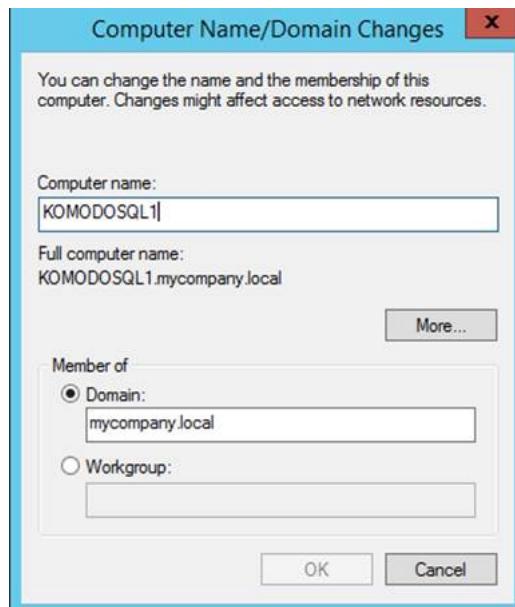
Menyiapkan Akun SQL Server dan Menggabungkan ke Domain

Tahap berikutnya harus dilakukan di tiap mesin yang akan menjadi node di dalam cluster dengan tujuan berikut:

- Menggabungkan tiap mesin ke Active Directory
- Menambahkan SQLCIsAdmin sebagai *local administrator*
- Memberikan hak yang dibutuhkan oleh SQLService untuk menjalankan layanan SQL Server

1) Tahap pertama adalah menggabungkan mesin **KOMODOSQL1** ke domain. Login sebagai *local administrator* kemudian buka System > klik Change Settings > Change.

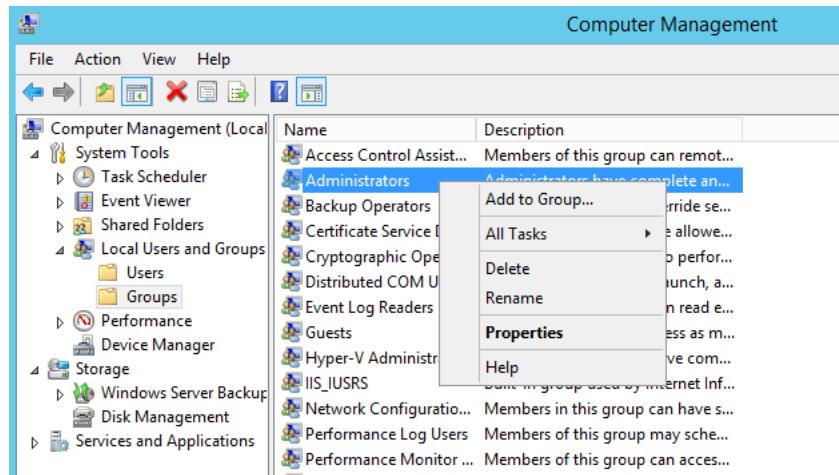
2) Masukkan data berikut sesuai dengan domain yang digunakan:



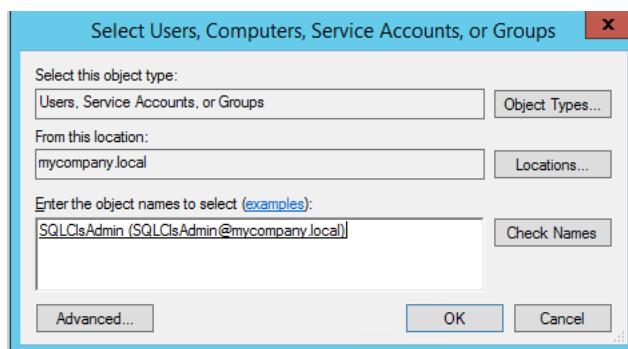
3) Klik OK dan masukkan password *domain administrator* jika diminta.

Tahap berikutnya dilakukan untuk menambahkan SQLCIsAdmin sebagai *local administrator*. Login ke **KomodoSQL1** sebagai *local administrator* _ dan buka _ Computer Management.

4) Buka Local Users and Groups > klik kanan group Administrators > Add to Group.



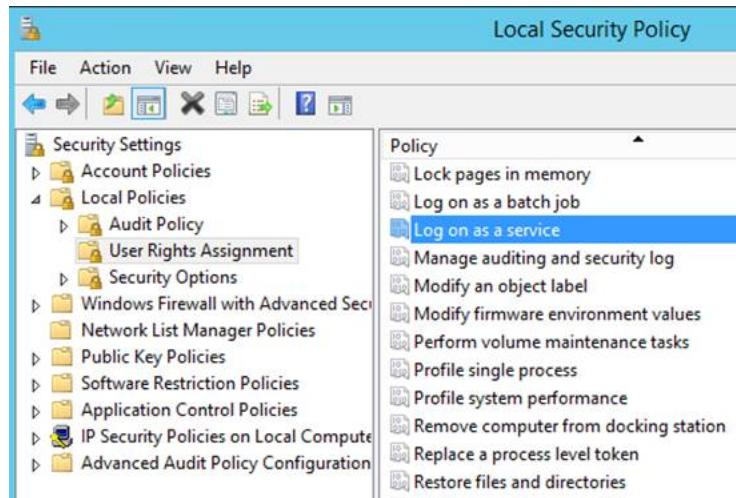
- 5) Klik tombol Locations dan pilih nama domain Active Directory yang Anda gunakan. Pada contoh dibawah domain yang digunakan adalah mycompany.local.
- 6) Selanjutnya isikan nama akun yang akan dijadikan *local administrator* yaitu **mycompanySQL-ClsAdmin**.



- 7) Klik OK untuk menyimpan penambahan tersebut.

Tahap terakhir adalah memberikan hak yang dibutuhkan akun SQLService untuk menjalankan layanan SQL Server. Login ke **KomodoSQL1** dengan menggunakan akun **mycompanySQLClsAdmin**.

- 8) Buka Local Security Policy > Local Policies > klik kanan User Rights Assignment > klik kanan **Log on as a service**.



- 9) Klik Add User or Group dan tambahkan akun **mycompanySQLService**.
- 10) Ulangi langkah-langkah tersebut untuk memberikan hak akses ke policy berikut:

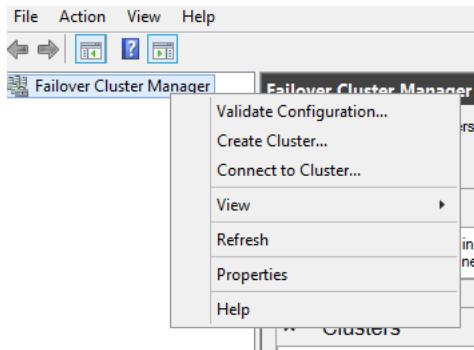
- Adjust memory quotas for a process
- Perform volume maintenance tasks

Ulangi semua langkah diatas untuk tiap server yang digunakan sebagai node di dalam cluster yaitu **KomodoSQL2** dan **KomodoSQL3**.

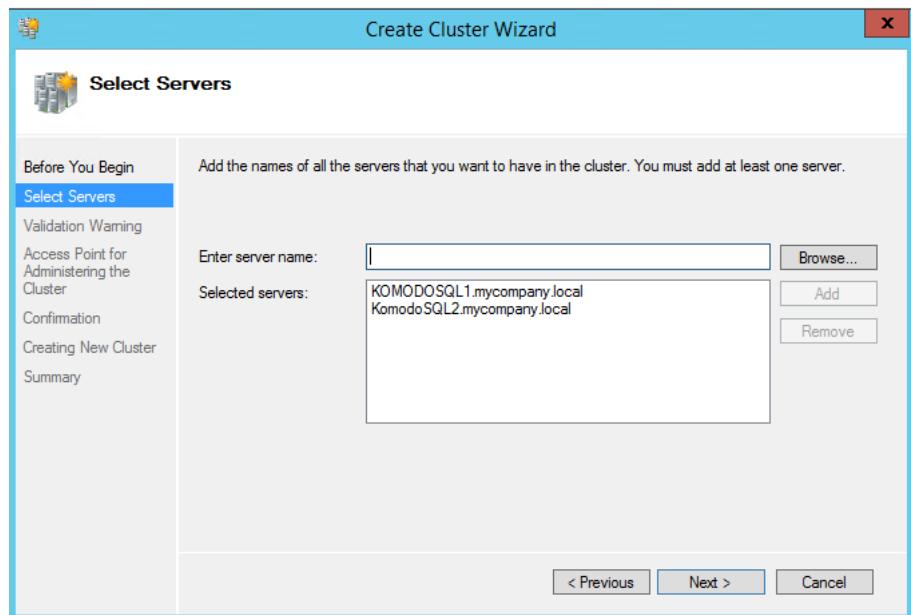
Menjalankan Cluster Validation

Cluster Validation berguna untuk memeriksa berbagai konfigurasi terkait Windows Cluster seperti alamat IP, konsistensi versi Windows dan Service Pack antar node, serta kompatibilitas hardware. Salah satu komponen terpenting adalah validasi storage, namun tidak akan dijalankan karena availability groups tidak memerlukan *shared storage*. Validasi storage harus dijalankan apabila SQL Server dikonfigurasi sebagai Failover Cluster Instance.

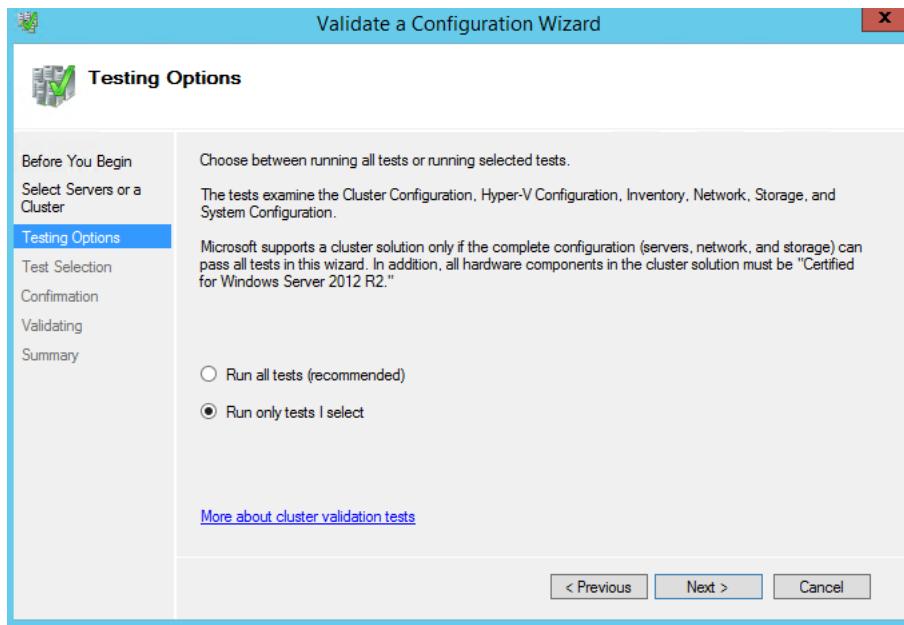
- 1) Login ke salah satu node (**KomodoSQL1**) dengan account administrator: **mycompanySQLClsAdmin**.
- 2) Buka Failover Cluster Manager kemudian pilih Validate a Configuration



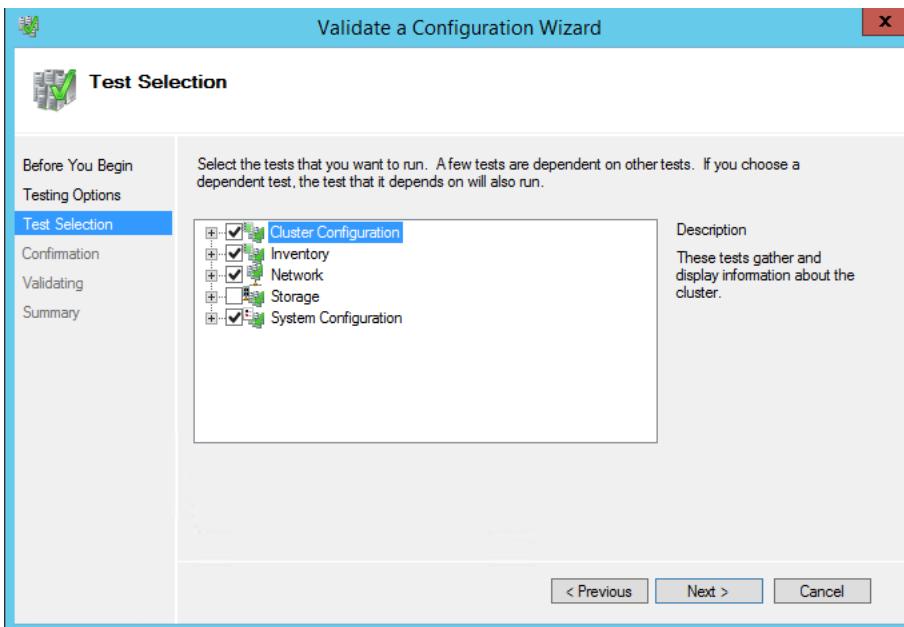
3) Pada jendela **Select Servers or a Cluster**, tambahkan dua node yang akan digunakan untuk cluster: **KomodoSQL1**** KomodoSQL2, KomodoSQL3.**



4) Selanjutnya pilih “Run only tests I select” dan klik Next untuk melanjutkan.

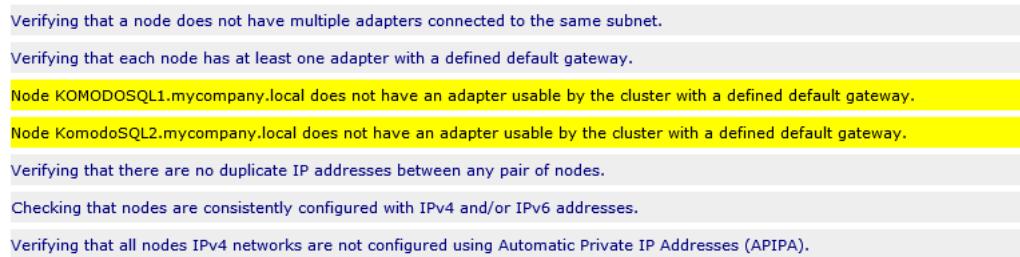


5) Hilangkan check pada pilihan “Storage” sehingga konfigurasi tersebut tidak dimasukkan dalam validasi.



6) Klik Next untuk menjalankan validasi. Setelah validasi selesai, buka laporan dengan mengklik **View Report**. [CA1]

7) Periksa kesalahan atau masalah dan menyelesaiannya sesuai rekomendasi. Jika Anda mengikuti semua langkah-langkah diatas, maka pesan peringatan berikut akan tampil di laporan validasi:



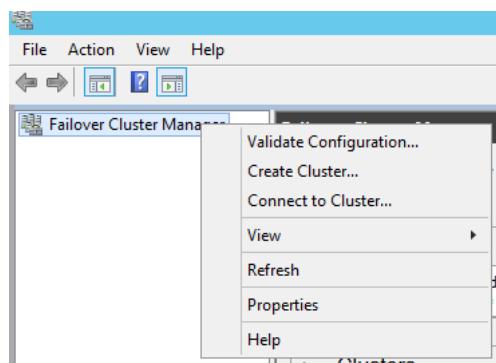
Proses konfigurasi belum dapat dilanjutkan jika terdapat pesan kesalahan yang bersifat *blocker* yang ditandai dengan warna merah. Konfigurasi yang bermasalah harus diperbaiki terlebih dahulu sebelum melanjutkan ke proses berikutnya.

Peringatan berwarna kuning dapat saja diabaikan jika memang dianggap tidak berpengaruh secara signifikan atau memang sengaja dilakukan dengan alasan tertentu. Pada contoh diatas, terdapat peringatan karena isian default gateway dibiarkan kosong. Konfigurasi private network memang tidak menyertakan default gateway karena hanya digunakan untuk komunikasi antar node (*heart beat*).

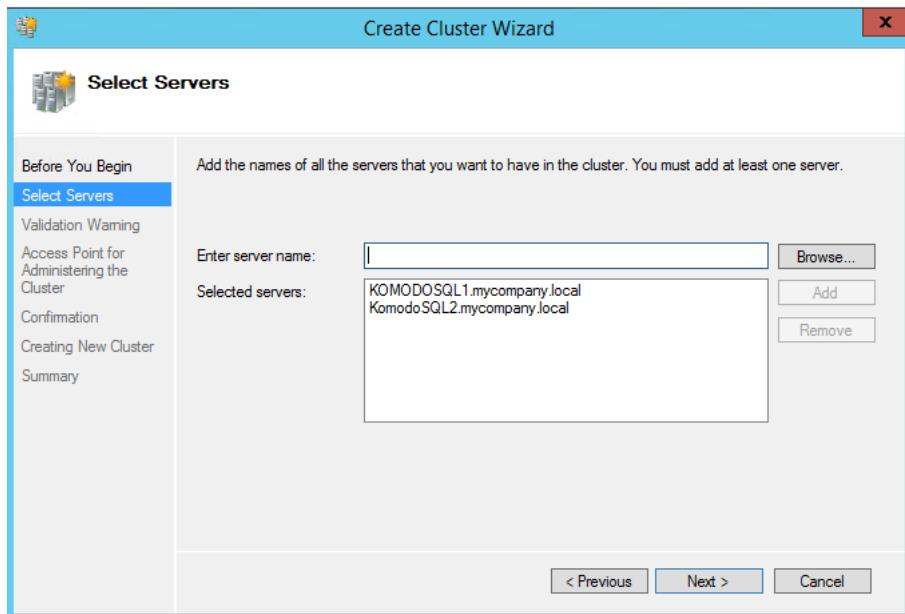
Konfigurasi Windows Cluster

Tahap berikutnya adalah membuat cluster atau WSFC baru setelah semua persiapan selesai dilakukan. Ikuti langkah-langkah dibawah untuk membuat cluster dengan nama **WIN2012-SQLAG**.

- 1) Log in ke node KomodoSQL1 dengan menggunakan akun **mycompanySQLC1sAdmin**.
- 2) Buka Failover Cluster Manager dan pilih **Create Cluster**.

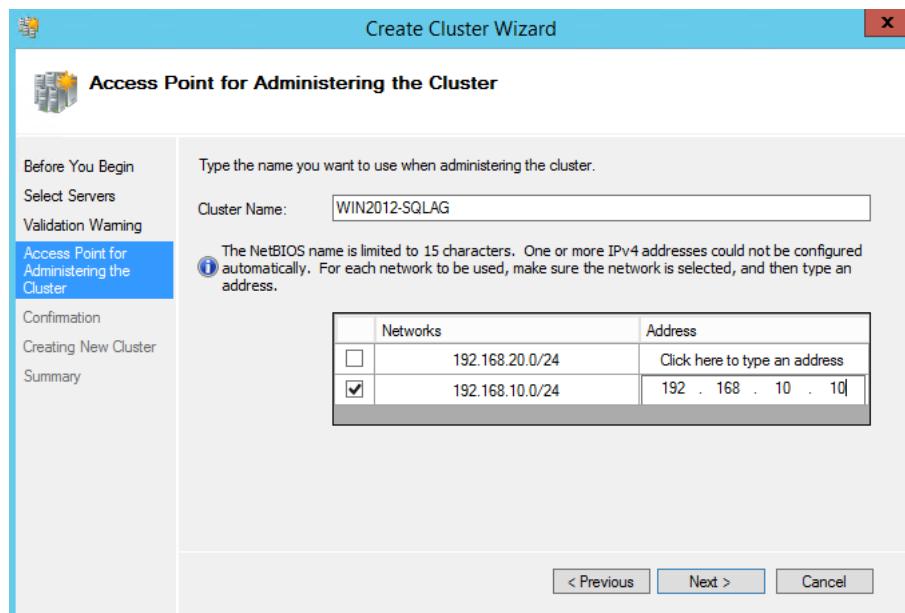


- 3) Tambahkan semua node yang menjadi anggota: **KomodoSQL1, KomodoSQL2** dan **KomodoSQL3**.



4) Cluster Wizard akan menyelesaikan proses verifikasi. Setelah selesai, menu **Access Point for Administering the Cluster** ditampilkan. Masukkan entri berikut:

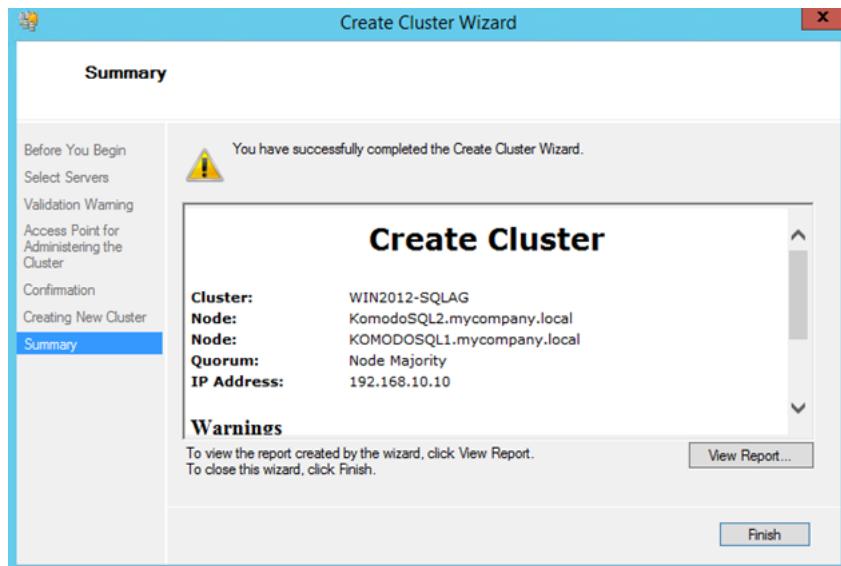
- Nama Cluster: WIN2012-SQLAG
- IP Address: 192.168.10.10



WIN2012-SQLAG tersebut adalah identitas Windows cluster di dalam jaringan, yang dapat digunakan untuk mengakses Failover Cluster Manager dari komputer lain. Nama tersebut bukanlah

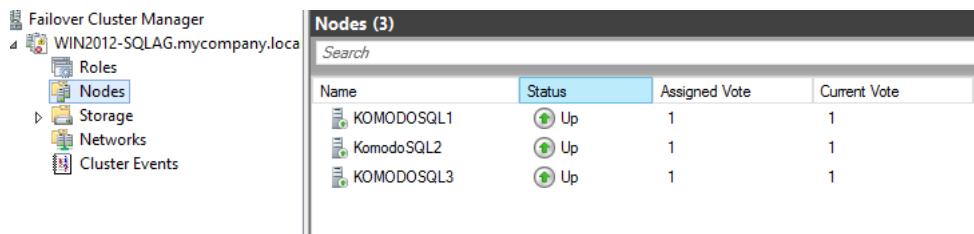
identitas untuk SQL Server Cluster atau availability groups. Secara teknis, nama ini disebut **Cluster Computer Name**.

5) Klik Next untuk memulai proses konfigurasi. Ketika konfigurasi cluster selesai dan berhasil, ringkasan konfigurasi ditampilkan beserta laporan keberhasilan maupun masalah yang terjadi.



cluster-success

Untuk melihat cluster yang telah dibuat, buka Failover Cluster Manager > WIN2012-SQLAG > Nodes. Cluster tersebut terdiri dari 3 node sebagaimana gambar dibawah.

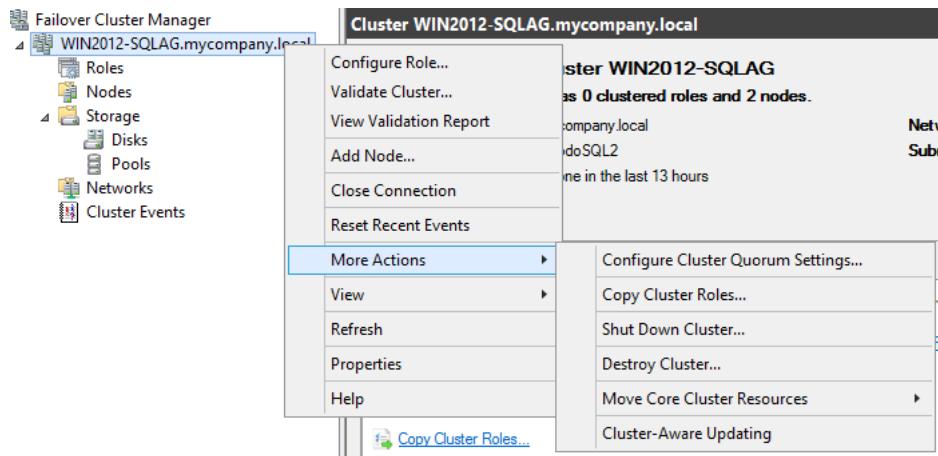


created-cluster

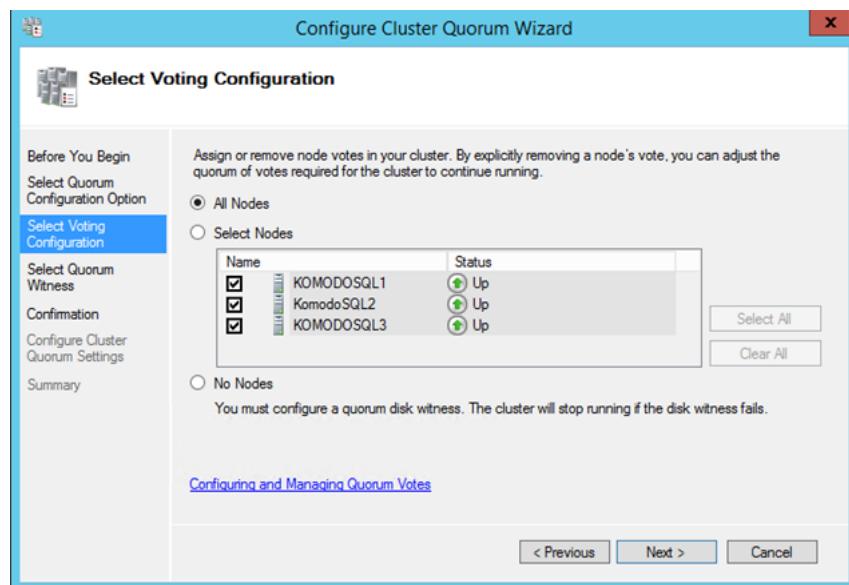
Konfigurasi Quorum

Quorum dikonfigurasi secara otomatis selama wizard dijalankan, tetapi Administrator perlu melakukan verifikasi untuk memastikan bahwa quorum yang dipilih sesuai dengan konfigurasi cluster. Cluster yang telah dibuat berjumlah 3 node sehingga quorum yang digunakan sebaiknya adalah *node majority*.

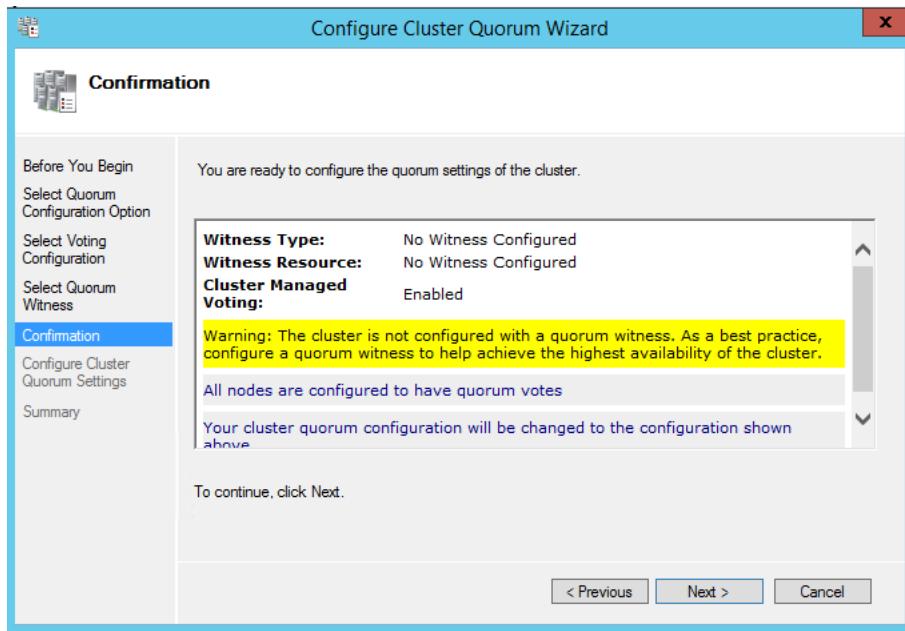
1) Buka Failover Cluster Manager dan klik kanan WIN2012-SQLAG > More Actions, lalu pilih **Configure Cluster Quorum Settings**.



- 2) Pada dialog **Select Quorum Configuration Options**, pilih metode konfigurasi quorum yaitu **Advanced quorum configuration**. Klik Next untuk melanjutkan.
- 3) Selanjutnya pilih **All nodes** untuk memasukkan semua mesin sebagai anggota quorum dalam node majority. Artinya setiap mesin memiliki hak suara yang sama dalam voting.

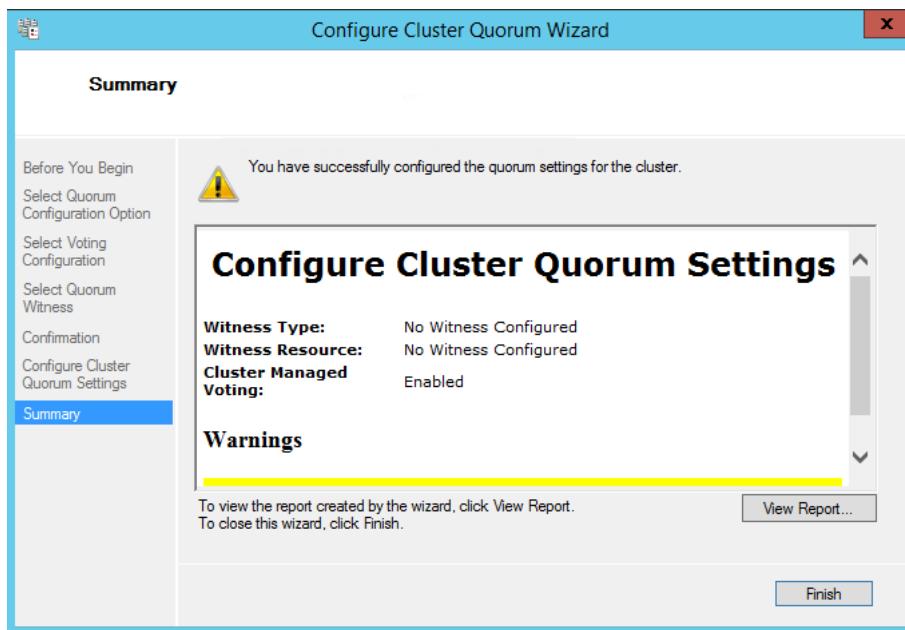


- 4) Konfigurasi node majority tidak mewajibkan adanya *quorum witness*, sehingga pada jendela selanjutnya pilih **Do not configure a quorum witness**.
- 5) Selanjutnya jendela Confirmation menampilkan konfigurasi yang telah dilakukan. Pesan peringatan berikut akan ditampilkan karena tidak terdapat witness di dalam quorum.



Abaikan pesan tersebut dan klik Next.

6) Laporan berikut ditampilkan setelah semua proses selesai dan quorum berhasil dibuat.



7) Selanjutnya Anda juga dapat melakukan verifikasi jenis quorum dengan menggunakan perintah PowerShell berikut.

```
Get-ClusterQuorum | FL
```

8) Hasilnya adalah sebagai berikut:

```
PS C:\Windows\system32> get-clusterquorum|FL
Cluster      : WIN2012-SQLAG
QuorumResource :
QuorumType   : NodeMajority

PS C:\Windows\system32> _
```

[CA1]TO-DO: perlu diberikan list problem yang sering terjadi

Bab 4. Konfigurasi AlwaysOn Availability Groups

Konfigurasi availability groups hanya dapat dilakukan jika WSFC telah dikonfigurasi dengan benar. Bagian ini menjelaskan langkah-langkah pembuatan availability groups, termasuk iinstalasi SQL Server dan berbagai persiapan yang diperlukan.

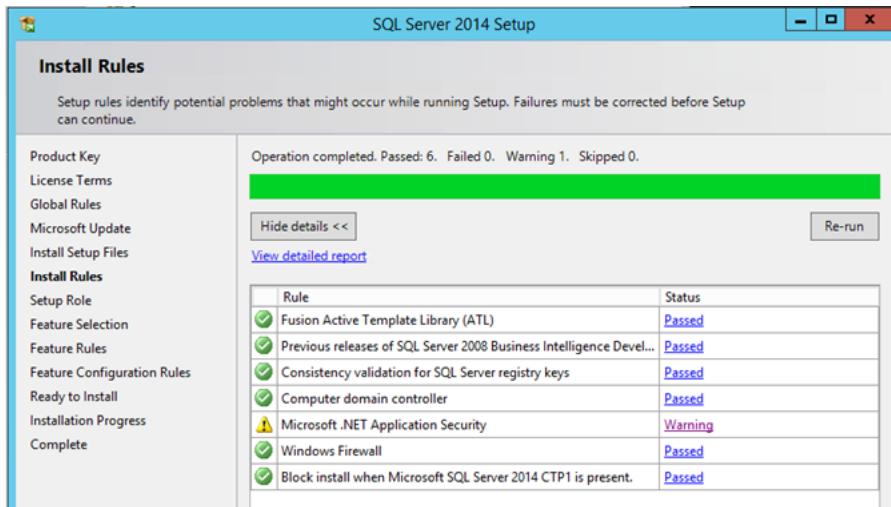
Instalasi SQL Server

Instalasi SQL Server untuk availability groups dilakukan secara *stand alone* di setiap node. Hal ini berbeda dengan FCI, dimana instalasi di tiap mesin harus menggabungkan instance SQL Server ke failover cluster. Availability groups pada hakekatnya adalah perlindungan dan sinkronisasi di tingkat database, sedangkan pada FCI perlindungan dilakukan di lingkup SQL Server instance.

Langkah-langkah instalasi dan konfigurasi yang dijelaskan dibawah berlaku untuk setiap node dalam availability groups. Anda akan membuat konfigurasi yang terdiri dari 3 mesin berikut:

- KomodoSQL1
- KomodoSQL2
- KomodoSQL3

- 1) Login ke KOMODOSQL1 sebagai **mycompanySQLCIsAdmin** yang merupakan administrator lokal. Jalankan **Setup.exe** dari SQL Server 2014 Enterprise Edition DVD.
- 2) Setelah Jendela SQL Server Installation Center tampil, ****pilih **Installation** di sebelah kiri dan **New** SQL Server stand-alone installation or add features...**** di sebelah kanan.
- 3) Proses setup akan memeriksa kesiapan komponen Windows dan prasyarat pendukungnya sebagaimana gambar berikut.



Jika terdapat peringatan atau kesalahan, periksa status tersebut dan selesaikan berdasarkan saran yang ditampilkan. Klik Re-run untuk melakukan validasi ulang

4) Pesan peringatan terkait Windows Firewall seperti dibawah ini akan tampil jika konfigurasi port belum dilakukan.

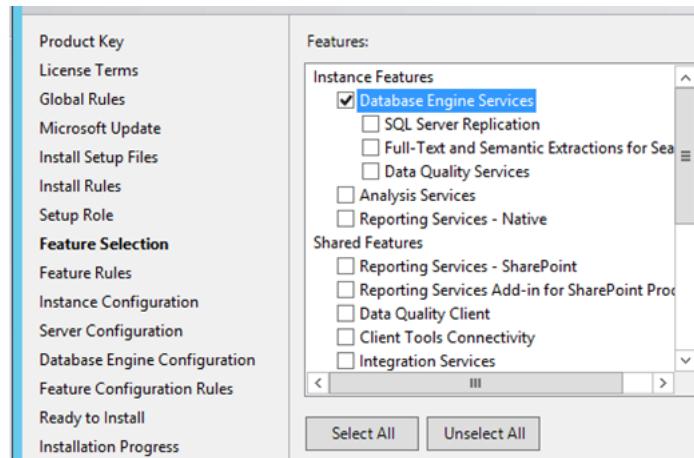


Bagian firewall berubah berwarna kuning karena port yang dibutuhkan masih tertutup dan belum dikonfigurasi sesuai kebutuhan SQL Server. Port – port tersebut perlu dibuka, yang dapat dilakukan setelah proses instalasi selesai[CA1] . Status peringatan tersebut tidak menjadi penghambat untuk melanjutkan proses instalasi.

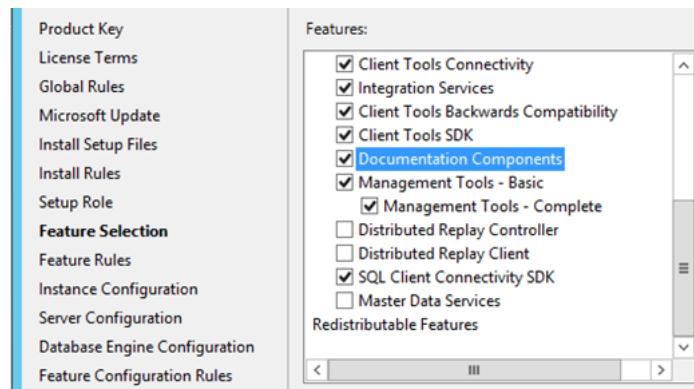
5) Klik Next untuk melanjutkan memilih fitur yang akan diinstall terkait kebutuhan availability groups sebagai berikut:

- Database Engine Services dan semua fitur terkait dibawahnya Replication and Full-text search
- Semua komponen klien seperti SSMS dan berbagai library terkait

Analysis Services and Reporting Services tidak termasuk dalam fitur yang kompatibel dengan availability groups, sehingga tidak perlu diinstall.

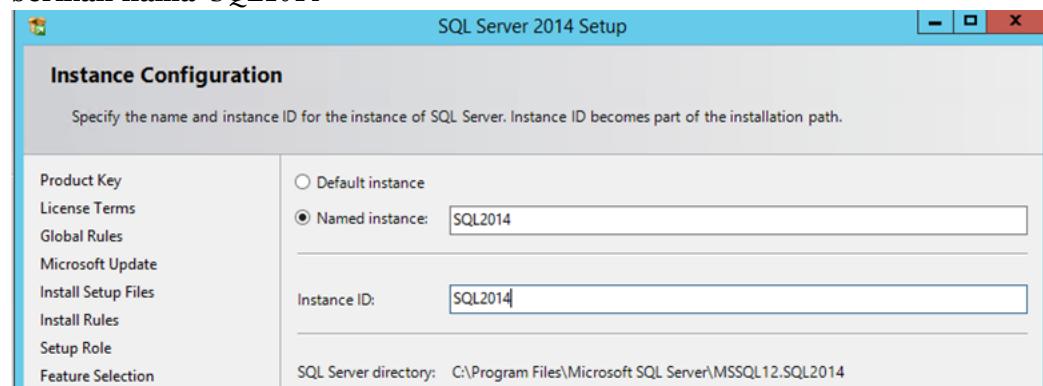


Komponen klien berikut perlu diinstall untuk memberikan dukungan akses data dan SSMS.



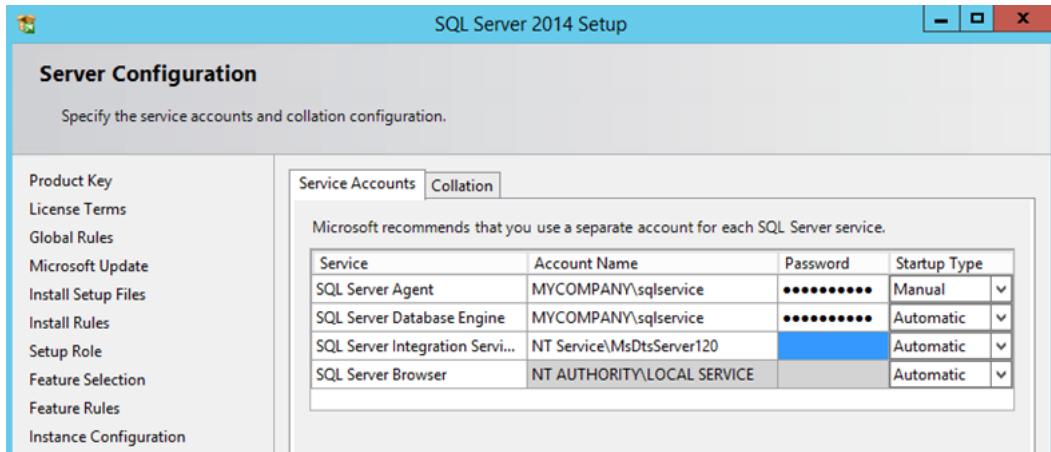
Analisis Services, Reporting Services, dan Integration dapat diinstall di tiap node tetapi layanan tersebut tidak dapat menjadi bagian dari availability groups. Komponen yang diinstall di tiap node harus sama untuk menghindari masalah yang timbul pada saat failover.

6) Klik Next untuk menyelesaikan Installation Rules. Lanjutkan proses ke konfigurasi instance. Terdapat 2 pilihan instance: **Standar** atau **Named Instance**. Dalam hal ini pilih **Named Instance** ** dan berikan nama: **SQL2014****

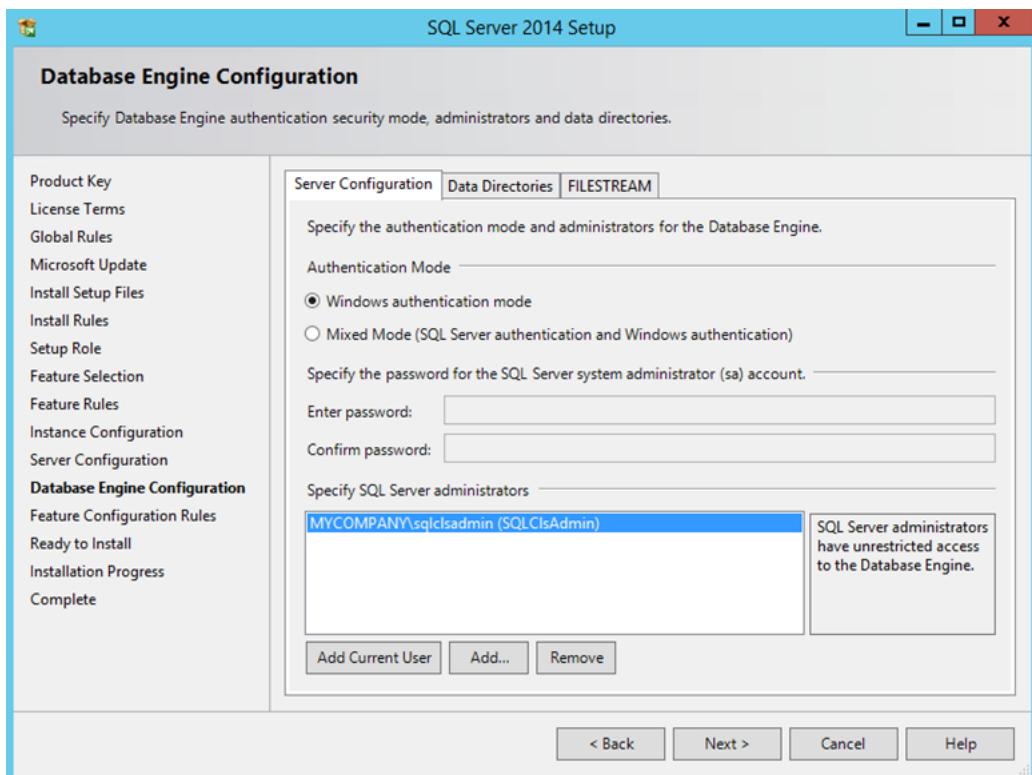


7) Periksa Disk Space Requirement dan klik Next untuk melanjutkan. Untuk *service account*,

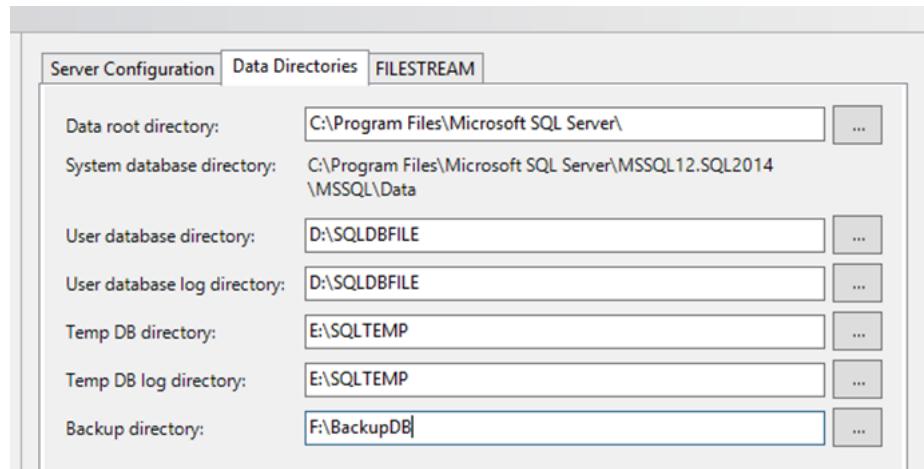
gunakan akun yang telah dipersiapkan sebelumnya yaitu **mycompanySQLService**.



8) Untuk konfigurasi otentifikasi, pilih **Windows Authentication** sebagai default dan klik **Add current user** untuk menambahkan akun sebagai administrator SQL Server.

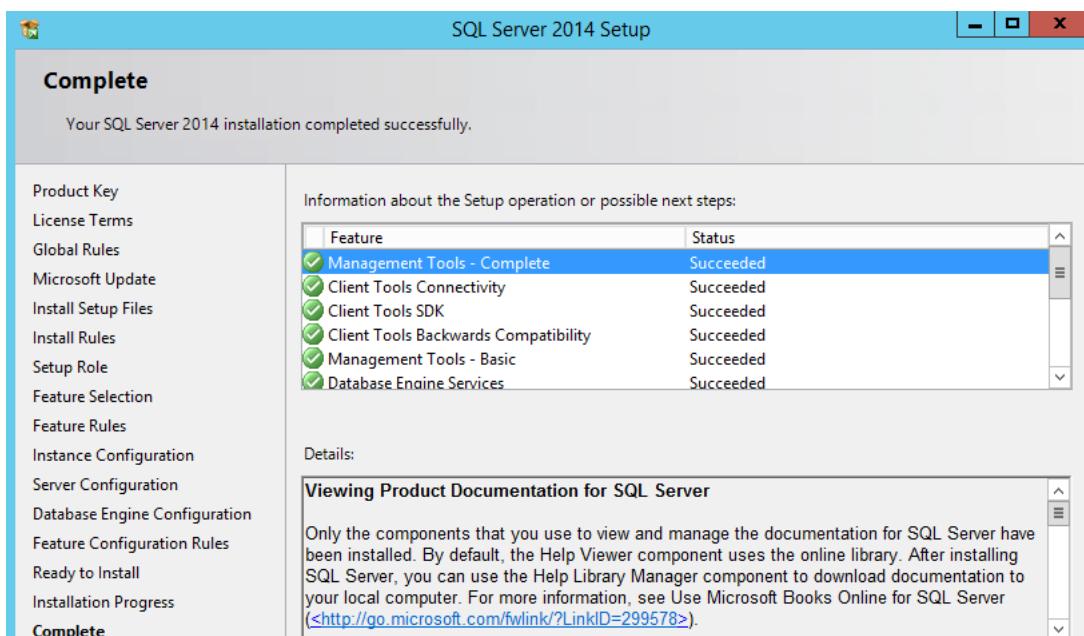


9) Klik tab **Data Directories** untuk menentukan lokasi database. Untuk memudahkan maintenance dan meningkatkan unjuk kerja, selalu dianjurkan penempatan database di lokasi terpisah sesuai dengan peruntukannya. Berikut ini adalah contoh penempatan file-file database yang digunakan:



Lokasi folder dan drive tersebut sebaiknya sama di semua node, sehingga memudahkan administrasi dan menghindari masalah tak terduga saat failover.

10) Setelah semuanya selesai, klik **OK** dan lanjutkan ke proses terakhir untuk melanjutkan instalasi.



Ulangi semua langkah-langkah instalasi diatas di server KOMODOSQL2 dan KOMODOSQL3.

Konfigurasi AlwaysOn Availability Groups

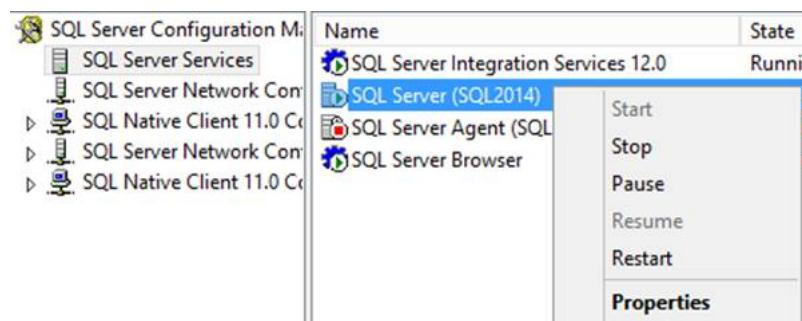
Setelah instalasi selesai, fungsi availability groups perlu diaktifkan pada setiap instance SQL Server di tiap node. Konfigurasi dilakukan menggunakan **SQL Server Configuration Manager**. Dalam contoh ini, konfigurasi harus dilakukan di *instance* berikut:

|Server| SQL Server Instance | | | KomodoSQL1 |KomodoSQL1SQL2014 | KomodoSQL2 | KomodoSQL2SQL2014
 |KomodoSQL3|KomodoSQL3SQL2014|

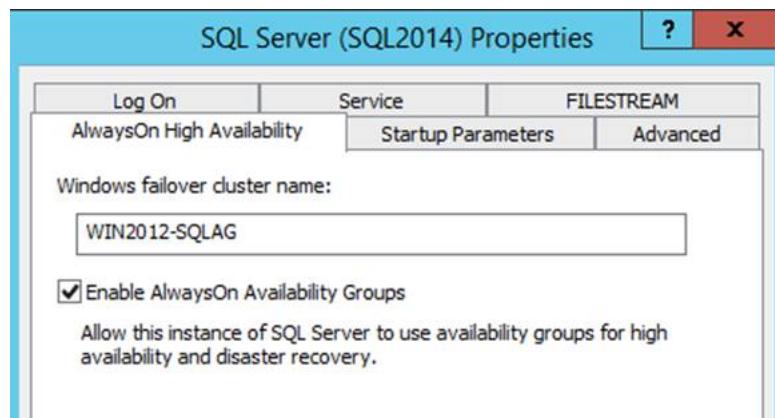
Mengaktifkan Availability Groups di tiap Instance

Lakukan langkah berikut untuk mengaktifkan fitur availability groups di tiap SQL Server *instance*:

1. Buka SQL Server Configuration Manager dan buka **Properties** SQL Server *instance* yang akan diaktifkan.



1. Buka tab AlwaysOn High Availability dan aktifkan pilihan **Enable AlwaysOn Availability Groups** seperti yang terlihat pada gambar di berikut.



Klik **OK** dan restart instance SQL Server tersebut untuk mengaktifkan perubahan yang telah dilakukan.

Catatan:

Jika Anda tidak menemukan pilihan **Enable AlwaysOn Availability Groups**, pastikan Windows cluster telah berjalan dengan benar. Pilihan tersebut tidak akan tersedia jika layanan Failover Cluster tidak aktif.

Konfigurasi Hak Akses untuk Service Account

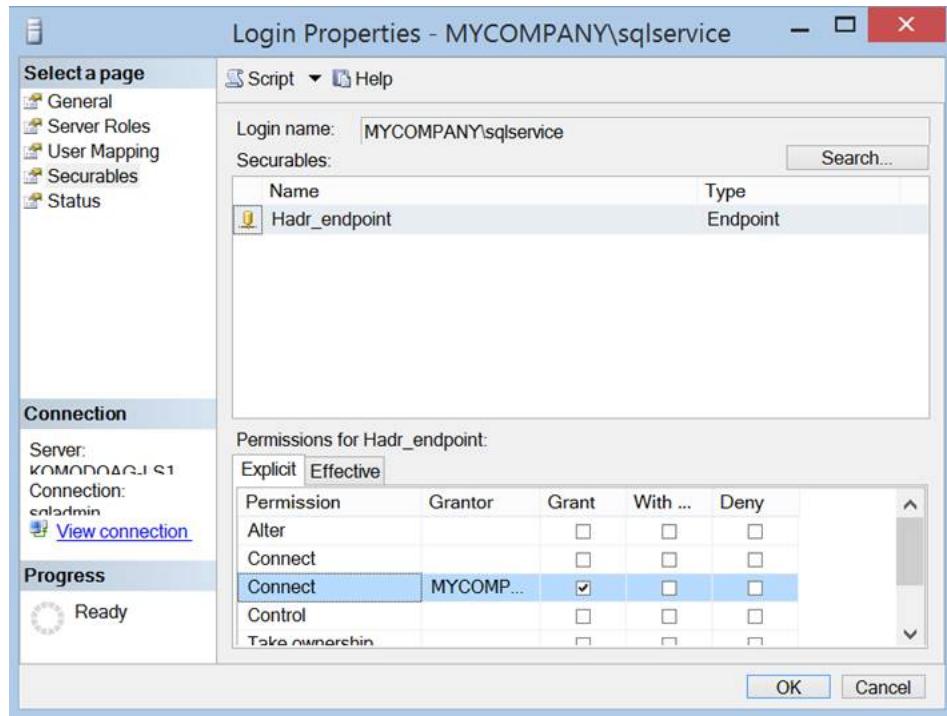
Akun yang digunakan sebagai *service account* SQL Server perlu diberikan hak akses yang sesuai sehingga dapat digunakan untuk berkomunikasi antar instance. Komunikasi tersebut menggunakan endpoint database mirroring, sehingga *service account* perlu diberikan hak **Connect** ke endpoint.

Terdapat beberapa alternatif mengenai service account yang digunakan untuk SQL Server tersebut, yaitu:

- **Menggunakan domain account yang sama** untuk semua SQL Server instance di semua node. Cara ini paling mudah dilakukan dan merupakan metode yang digunakan di dalam buku ini. Konfigurasi hak akses ke endpoint terjadi secara otomatis ketika availability groups dibuat, sehingga tidak diperlukan konfigurasi tambahan yang berhubungan dengan hak tersebut.
- **Menggunakan domain account yang berbeda** untuk setiap SQL Server instance di tiap node. Semua service account perlu ditambahkan ke database **master** tiap SQL Server instance dan diberikan hak Connect ke endpoint.
- **Menggunakan akun Network Service** di tiap instance SQL Server. Metode ini mengharuskan setiap computer account (DomainComputerName\$) ditambahkan ke database **master** tiap node, dan diberikan hak akses Connect ke endpoint.
- **Menggunakan akun local dan tidak tergabung ke domain.** Metode ini membutuhkan konfigurasi sertifikat keamanan di tiap instance sehingga dapat saling berkomunikasi menggunakan endpoint.

Penggunaan domain account sebagai service account adalah praktik yang direkomendasikan sehingga pendekatan tersebut sebaiknya selalu digunakan dalam konfigurasi SQL Server.

Instalasi SQL Server yang telah dijelaskan di Bagian 4.1 menggunakan akun **mycompanySQLService** sebagai *service account* di semua node. Dengan demikian hak akses **Connect** ke endpoint akan ditambahkan secara otomatis ketika availability groups dibuat. Hak akses untuk akun tersebut dapat dilihat sebagaimana gambar berikut:



Menyiapkan Database Untuk Availability Groups

Sebelum membuat availability groups perlu dilakukan beberapa persiapan terkait database dan lokasi backup. Database di dalam availability groups harus memenuhi beberapa syarat berikut:

- Database harus menggunakan recovery model **Full** karena detail catatan transaction log dibutuhkan dalam proses sinkronisasi.
- Semua database harus menggunakan SQL Collation yang sama.
- **Full backup** telah dilakukan sehingga file backup tersebut dapat digunakan sebagai media restore ke secondary. Usahakan untuk selalu menggunakan full backup terbaru sehingga durasi sinkronisasi awal antar replica dapat ditekan seminimal mungkin.

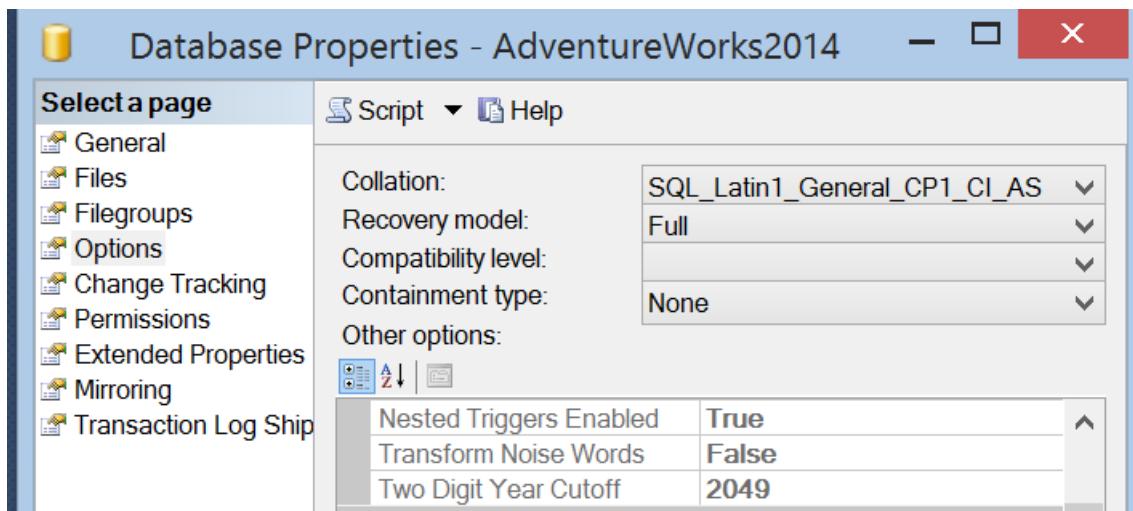
Selain persiapan di sisi database tersebut, perlu juga dibuat *file share* di primary server yang merupakan lokasi backup. Proses restore ke secondary dilakukan dengan mengakses file backup yang terdapat di *file share* tersebut.

Alternatif lain yang lebih cepat adalah melakukan restore ke secondary secara terpisah sebelum proses pembuatan availability groups. Cara ini lebih cepat jika ukuran database sangat besar, sehingga mengurangi proses penyalinan database melewati jaringan.

Bagian berikutnya menjelaskan persiapan database untuk availability groups dengan menggunakan SQL Server Management Studio.

Menyiapkan Database

- 1) Login ke KOMODOSQL1 sebagai mycompanySQLCIsAdmin kemudian buka Microsoft SQL Server Management Studio.
- 2) Klik kanan database AdventureWorks2014 > Properties, dan pastikan recovery model yang digunakan adalah Full. Jika belum, ubah ke Full dan klik OK untuk menyimpan konfigurasi.



Catatan Collation yang digunakan sebagai referensi untuk database di secondary. Semua replica database terkait harus menggunakan Collation yang sama.

- 3) Selanjutnya perlu dibuat salinan Full backup untuk direstore secondary. Gunakan script berikut untuk melakukan Full backup dan meletakkan file di c:BackupDB.

```

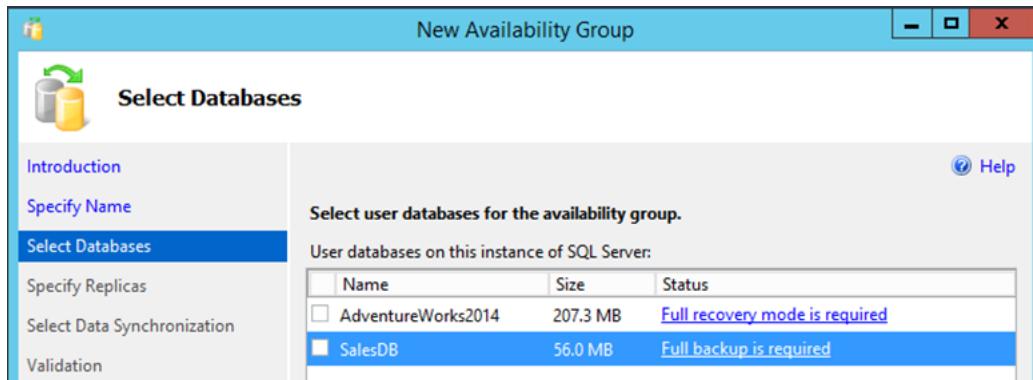
1 BACKUP DATABASE [AdventureWorks2014]
2 TO DISK = N'C:\BackupDB\AdvWorks2014Fullbackup.bak'
3 WITH NOFORMAT, NOINIT,
4 NAME = N'AdventureWorks2014-Full Database Backup', SKIP, NOREWIND, NOUNLOAD,
5 STATS = 10
6 GO

```

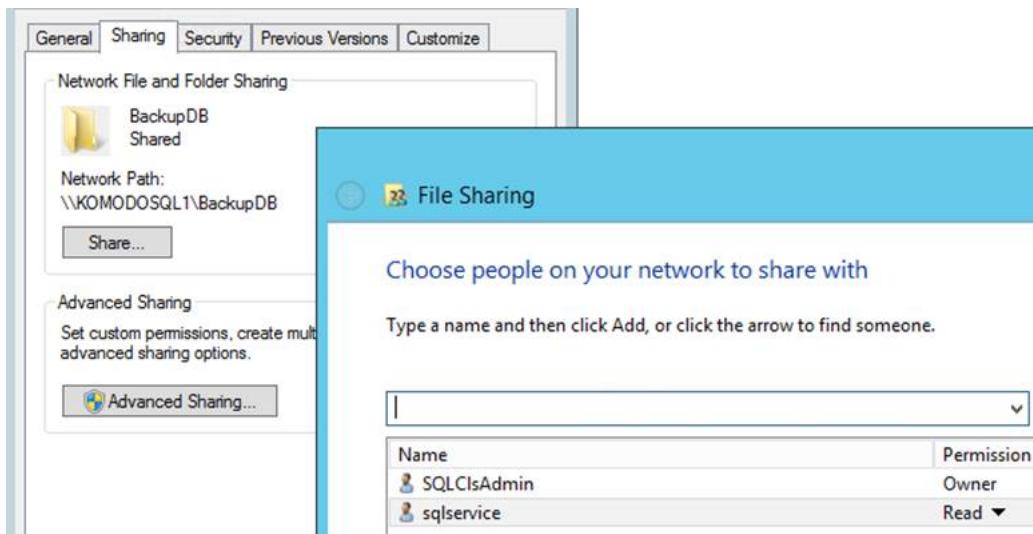
Anda juga dapat menggunakan SSMS dengan memilih database AdventureWorks2014 > klik kanan > pilih Tasks > Back Up.

- 4) Lakukan langkah-langkah tersebut untuk semua database yang akan dimasukkan ke dalam availability groups. Dalam contoh di buku ini terdapat 2 database yaitu AdventureWorks2014 dan SalesDB.

Semua persiapan tersebut sebaiknya dilakukan sebelum pembuatan availability groups dimulai. Jika terdapat database yang belum memenuhi prasyarat tersebut, maka pesan peringatan berikut akan tampil ketika proses konfigurasi:



Selanjutnya diperlukan folder BackupDB di lokasi c:BackupDB yang harus disetup sebagai *share folder* sehingga dapat diakses oleh *service account* selama proses restore. Akun **mycompanySQLService** harus diberikan hak akses untuk membaca folder tersebut sebagaimana gambar berikut.



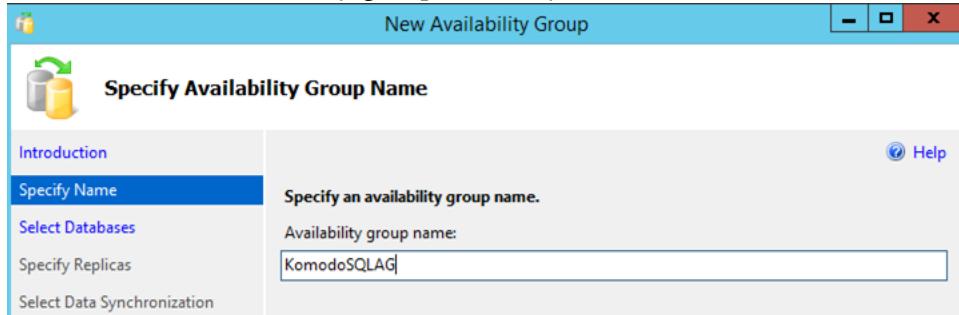
Rekomendasi lokasi backup di server production sebaiknya tidak diletakkan di drive C, dan harus dipisahkan dari lokasi database.

Membuat Availability Groups

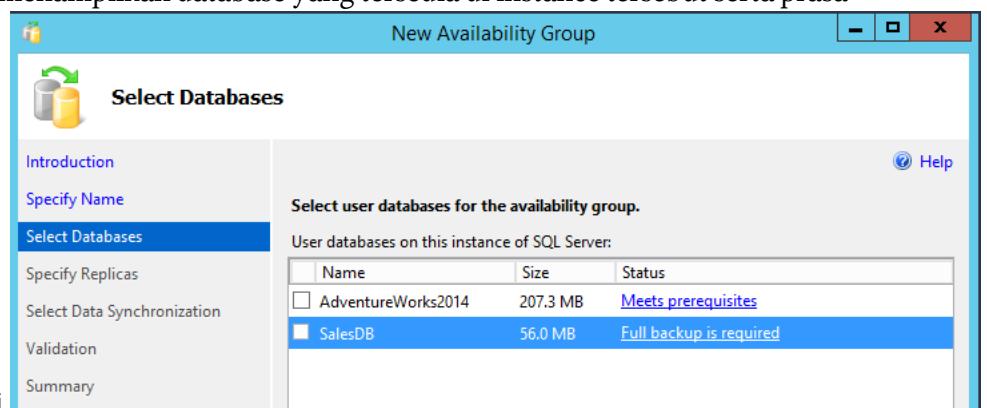
Setelah semua persiapan selesai, langkah berikutnya adalah membuat availability groups baru yang akan memasukkan AdventureWorks2014 dan SalesDB ke dalam satu kelompok.

1. Buka Microsoft SQL Server Management Studio > Availability > ** ****AlwaysOn Availability Groups. Klik kanan Availability Groups** dan pilih New** availability groups Wizard.**

2. Tentukan nama availability groups, misalnya KomodoSQLAG dan klik Next.



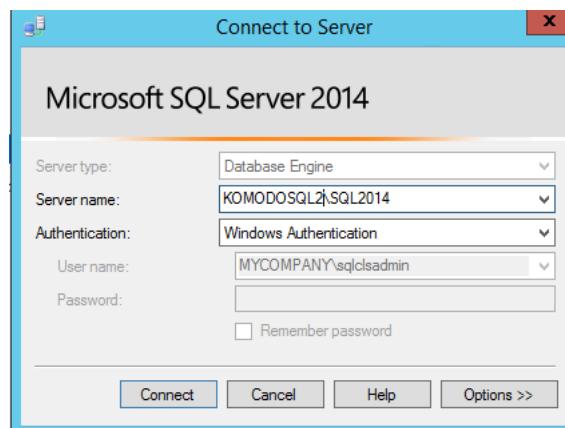
3. Dalam langkah berikutnya Anda harus memilih database dijadikan anggota availability groups. Layar berikutnya menampilkan database yang tersedia di instance tersebut serta prasyarat yang harus dipenuhi.



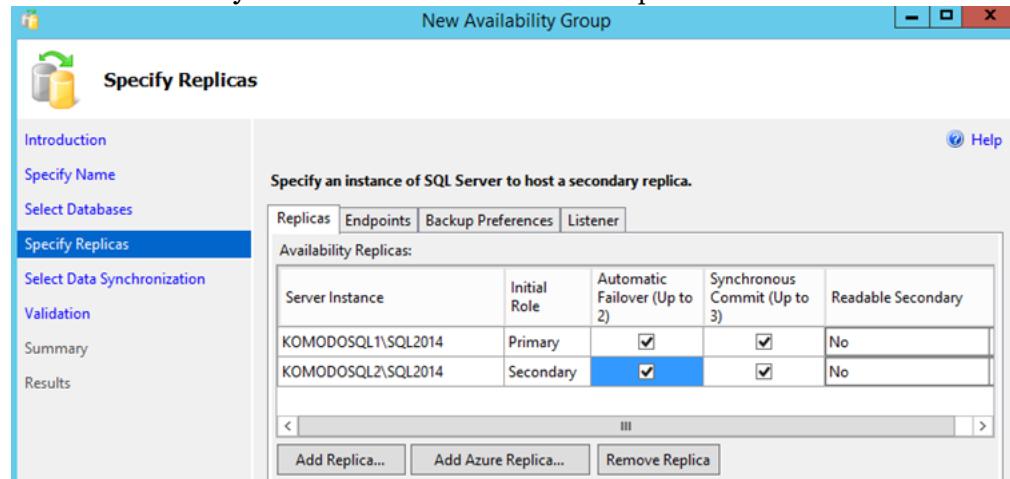
yarat yang harus dipenuhi.

Dalam gambar diatas SalesDB tidak memenuhi syarat karena Full backup belum dilakukan. Anda harus melakukan hal tersebut sebelum melanjutkan ke proses berikutnya.

4. Jendela berikutnya adalah menambahkan replica ke dalam anggota availability groups. Anda harus terkoneksi ke tiap SQL Server instance untuk menjadikan server tersebut sebagai replica. Klik Add Replica ... untuk menambahkan node kedua dan gunakan Windows Authentication untuk menghubungkan ke KOMODOSQL2SQL2014.



5. Aktifkan pilihan **Automatic Failover** serta **Synchronous commit...** untuk replica tersebut se-



bagaimana jendela berikut.

Dengan demikian kedua node tersebut menjadi pasangan Automatic failover serta menggunakan mode Synchronous commit, tanpa *data loss*. Untuk sementara Anda belum perlu mengaktifkan pilihan Readable Secondary. Bab 5 menjelaskan lebih lanjut mengenai pilihan tersebut dan bagaimana cara mengaksesnya dari aplikasi.

6. Tambahkan node ketiga yaitu KOMODOSQL3\SQL2014. Klik **Add replica** dan jadikan node tersebut tanpa Automatic failover dengan mode Asynchronous commit. Hasil akhir konfigurasi

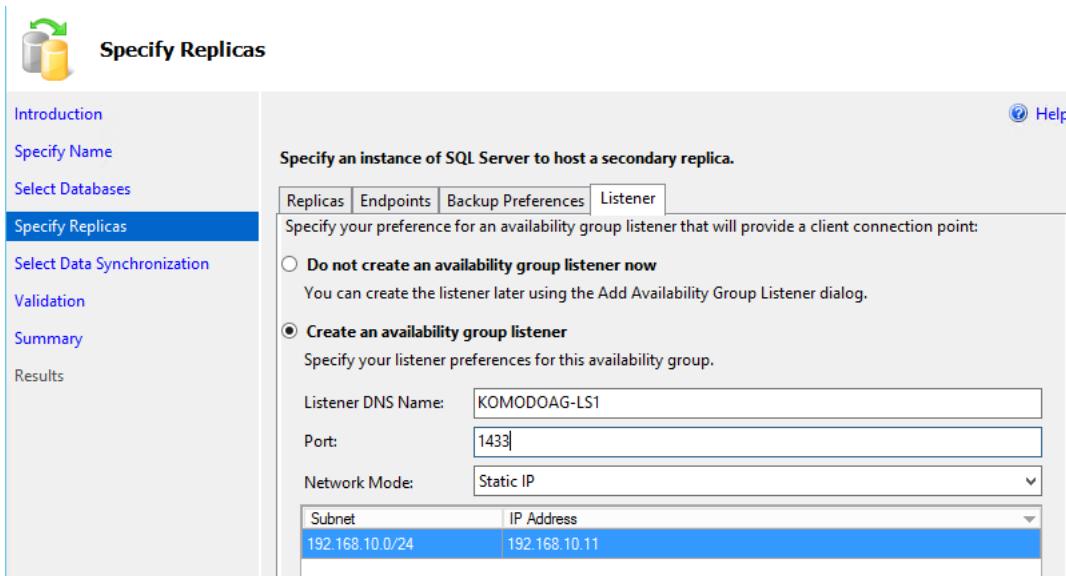
| Availability Replicas | | | | | | |
|-----------------------|-----------|---------------------|---------------|-----------------------------|--------------------|--|
| Server Instance | Role | Availability Mode | Failover Mode | Connections in Primary Role | Readable Secondary | |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | Automatic | Allow all conn... | No | |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | Automatic | Allow all conn... | No | |
| KOMODOSQL3\SQL2014 | Secondary | Asynchronous commit | Manual | Allow all conn... | No | |

tersebut adalah sebagai berikut.

[CA2]

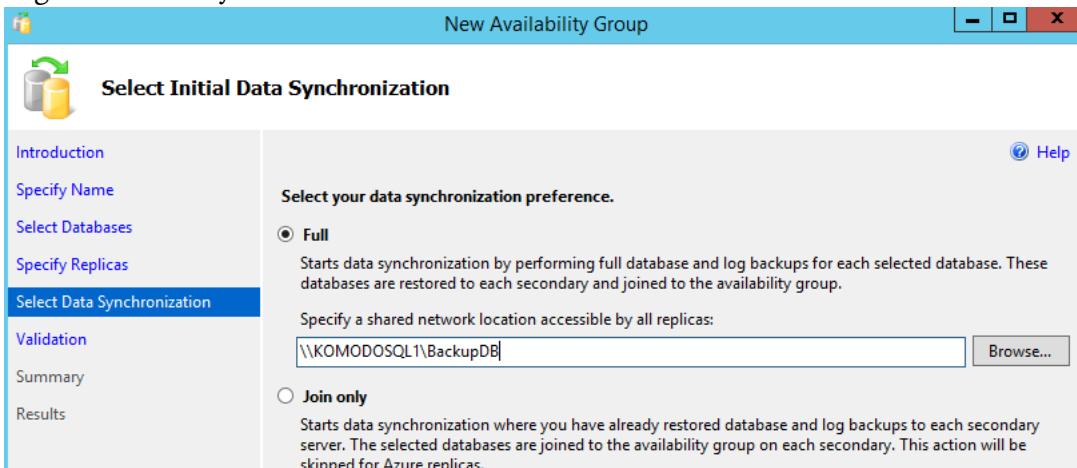
7. Selanjutnya perlu dibuat listener sehingga aplikasi klien dapat menggunakannya untuk mengakses database. Klik tab **Listener** dan pilih **Create an availability groups listener**. Masukkan data berikut:

- Listener DNS Name: KOMODOAG-LS1
- Port: 1433
- Network Mode: Static IP
- Subnet: 192.168.10.0/24
- IP Address: 192.168.10.11



Klik **Next** untuk melanjutkan ke pilihan sinkronisasi awal database.

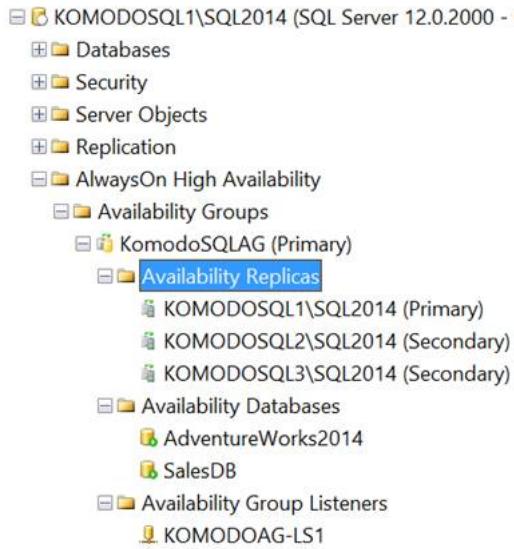
1. Cara termudah adalah memilih opsi **Full** di mana sinkronisasi awal dilakukan melalui jaringan dengan mengakses folder **BackupDB** yang tersedia di primary. Isikan lokasi folder **\KOMODOSQL1\BackupDB**¹ dan pastikan *folder share* tersebut telah dibuat sebagaimana dijelaskan di bagian sebelumnya.



Pilihan ini tersebut jika ukuran database tidak terlalu besar, sehingga proses backup dan restore dilakukan saat konfigurasi dibuat.

2. Klik **Next** untuk menjalankan validasi dan melanjutkan proses pembuatan availability groups. Jika terdapat beberapa kesalahan, Anda harus membetulkan hal tersebut sesuai dengan pesan peringatan yang diberikan.
3. Setelah semua proses selesai, availability groups dapat dilihat melalui SSMS sebagai berikut.

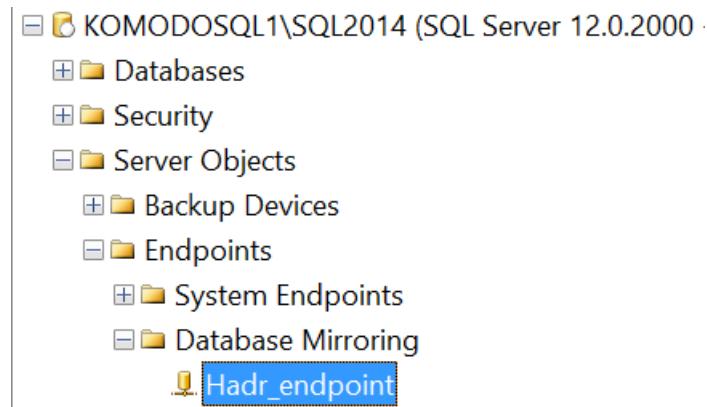
¹file:///\\KOMODOSQL1\\BackupDB



4. Buka Failover Cluster Manager dan periksa cluster Roles. Terlihat bahwa availability groups yang baru saja dibuat sudah masuk ke dalam cluster role di dalam WSFC.



12. Proses pembuatan availability groups tersebut juga menghasilkan endpoint database mirroring yang digunakan sebagai jalur sinkronisasi data antar replica. Secara default, nama endpoint tersebut adalah **Hadr_endpoint** dan berjalan di port 5022 TCP sebagai berikut.



13. Untuk memastikan bahwa sinkronisasi data berjalan baik dan konfigurasi tersebut siap melakukan failover, buka dashboard availability groups. Buka SSMS > AlwaysOn High Availability > Availability Groups > KOMODOSQLAG > **Show Dashboard**.

 KomodoSQLAG: hosted by KOMODOSQL1\SQL2014 (Replica role: Primary)

Availability group state:  Healthy

Primary instance: KOMODOSQL1\SQL2014

Failover mode: Automatic

Cluster state: WIN2012-SQLAG (Normal Quorum)

Availability replica:

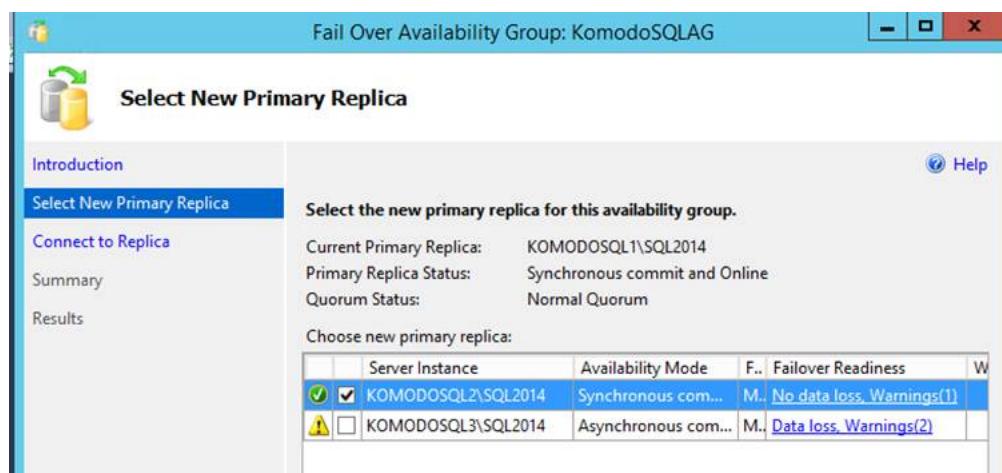
| Name | Role | Failover Mode | Synchronization S... | Issues |
|--|-----------|---------------|----------------------|--------|
|  KOMODOSQL1\SQL2014 | Primary | Automatic | Synchronized | |
|  KOMODOSQL2\SQL2014 | Secondary | Automatic | Synchronized | |
|  KOMODOSQL3\SQL2014 | Secondary | Manual | Synchronizing | |

Semua replica yang merupakan pasangan Automatic failover harus berada pada kondisi synchronized, sedangkan lainnya dapat berupa synchronizing.

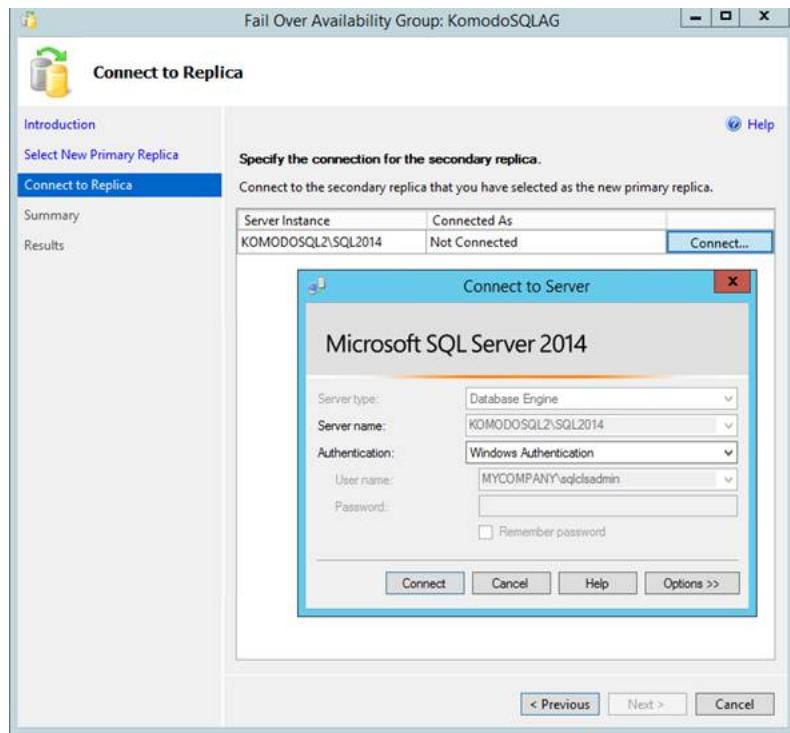
Pengujian Failover

Konfigurasi yang telah dibuat perlu diuji apakah memang dapat melakukan failover dengan baik. Pada bab ini hanya dilakukan proses failover sederhana dengan metode planned manual failover. Penjelasan lebih detail tentang berbagai macam metode failover dijelaskan di Bab 6.

- 1) Dari dalam jendela SSMS, buka menu AlwaysOn High Availability > Availability Groups > KomodoSQLAG > Failover
- 2) Menu failover wizard terbuka dan menampilkan daftar replica yang dapat dijadikan target failover. Pilih KOMODOSQL2SQL2014 dan klik Next untuk melanjutkan.



- 3) Klik Connect dan pastikan KOMODOSQL2SQL2014 tampil sebagai Server name. Klik Connect and Next untuk melanjutkan.



4) Bagian terakhir menampilkan summary dari konfigurasi yang telah dilakukan. Klik Finish untuk mengesekusi proses failover. Selanjutnya buka dashboard untuk memantau hasil akhir proses tersebut.

KomodoSQLAG: hosted by KOMODOSQL2\SQL2014 (Replica role: Primary)

| | |
|---------------------------|-------------------------------|
| Availability group state: | Healthy |
| Primary instance: | KOMODOSQL2\SQL2014 |
| Failover mode: | Manual |
| Cluster state: | WIN2012-SQLAG (Normal Quorum) |

Availability replica:

| | Name | Role | Failover Mode | Synchronization State | Availability Mode |
|--|-------------------|-----------|---------------|-----------------------|---------------------|
| | KOMODOSQL1\SQL... | Secondary | Automatic | Synchronized | Synchronous commit |
| | KOMODOSQL2\SQL... | Primary | Manual | Synchronized | Synchronous commit |
| | KOMODOSQL3\SQL... | Secondary | Manual | Synchronizing | Asynchronous com... |

Proses failover tersebut berhasil dan KOMODOSQL2\SQL2014 saat ini mengambil alih peran sebagai primary sedangkan KOMODOSQL1\SQL2014 berubah menjadi secondary.

Jalankan kembali Failover wizard untuk mengembalikan posisi primary ke KOMODOSQL2\SQL2014.

[CA1]TO-DO: perlu dijelaskan konfigurasi port setelah instalasi

Bab 5. Konektivitas Aplikasi dan Read Only Routing

Bab ini tidak tersedia dalam sampel PDF

Persyaratan Aplikasi

Konfigurasi Listener

Konektivitas Aplikasi ke Availability Group Listener

Koneksi ke Primary Replica Menggunakan Listener

Koneksi ke Readable Secondary Menggunakan Listener

Konfigurasi Read Only Routing

Bab 6. Failover dan Perpindahan Role

Proses failover merupakan kejadian paling kritikal dalam operasi availability groups di lingkungan *production*. Semua konfigurasi database, infrastruktur, dan aplikasi pada dasarnya merupakan upaya untuk mendukung terjadinya failover secara aman dan terkendali.

Pada saat failover tersebut terjadi, aplikasi klien diharapkan dapat terkoneksi kembali ke server secondary dalam waktu singkat. Proses tersebut dapat berlangsung secara manual maupun otomatis, tergantung pada konfigurasi SQL Server dan kompatibilitas aplikasi klien. Pada Bab 2 kita sudah membahas peranan metode asynchronous dan synchronous untuk mendukung proses failover. Pada bagian ini Kita akan mendalami lebih detail mengenai proses tersebut, syarat-syarat yang diperlukan, serta metode testing untuk melakukan verifikasi kemampuan failover.

Relasi antara Failover dan Availability Mode

SQL Server mendukung tiga jenis failover yang dapat dilakukan pada konfigurasi availability groups sebagai berikut:

- Automatic failover (tanpa *data loss*)
- Planned manual failover (tanpa *data loss*)
- Forced manual failover (dengan *data loss*)

Kemampuan untuk melakukan ketiga jenis failover tersebut tergantung pada konfigurasi availability mode, yaitu sinkronisasi data yang digunakan pada tiap replica. Kompatibilitas antara availability mode dengan jenis failover dapat disarikan dalam table berikut:

| | | | | | | | | | | | |
|-------------------|--------------------------------------|--------------------------------------|---|--------------------|----|----|----|---------------------|-------|-------|----|
| Availability mode | Automatic failover (tanpa data loss) | Automatic failover (tanpa data loss) | Forced Manual failover (dengan data loss) | Synchronous commit | Ya | Ya | Ya | Asynchronous commit | Tidak | Tidak | Ya |
|-------------------|--------------------------------------|--------------------------------------|---|--------------------|----|----|----|---------------------|-------|-------|----|

Berdasarkan tabel tersebut, maka dapat disimpulkan beberapa hal sebagai berikut:

- Jika failover yang diinginkan adalah tanpa data loss, maka semua replica yang terlibat harus menggunakan konfigurasi Synchronous commit.
- Replica dengan mode Synchronous commit juga dapat melakukan forced manual failover dengan resiko kehilangan data. Kondisi ini dapat terjadi jika proses sinkronisasi bermasalah sehingga secondary replica tidak berada dalam kondisi *healthy state*.

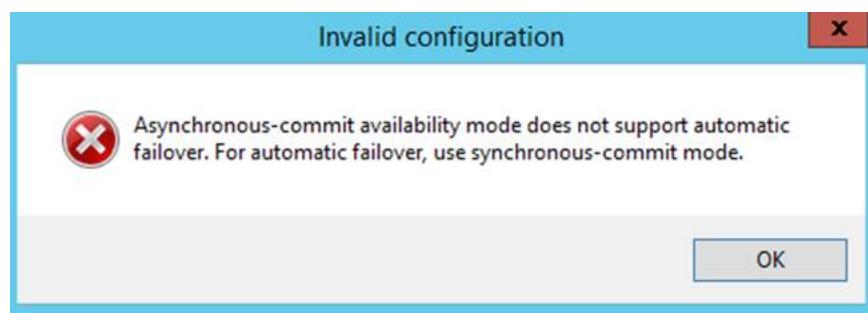
- Replica dengan mode Asynchronous commit hanya dapat menggunakan forced manual failover, kesamaan data antar replica tidak dapat dijamin. Replica yang terlibat tidak pernah mencapai kondisi synchronized, sehingga failover harus dilakukan dengan deklarasi ALLOW_DATA_LOSS.

Gambar berikut memberikan contoh konfigurasi yang terdiri dari 3 server dengan konfigurasi berikut:

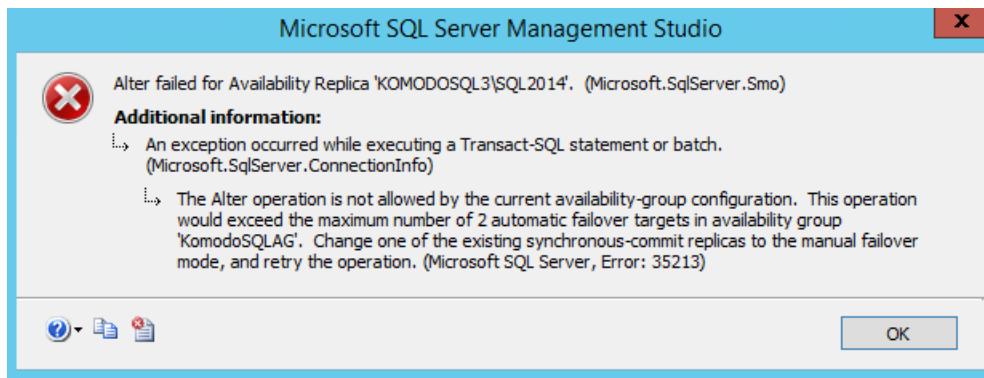
- Dua node yang menggunakan automatic failover dengan mode Synchronous commit.
- Satu node menggunakan manual failover dengan availability mode Asynchronous commit.

| Availability Replicas | | | | | |
|-----------------------|-----------|---------------------|---------------|-----------|-----------|
| Server Instance | Role | Availability Mode | Failover Mode | | |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | Automatic | Automatic | Automatic |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | Automatic | Automatic | Automatic |
| KOMODOSQL3\SQL2014 | Secondary | Asynchronous commit | Manual | Manual | Manual |

Jika Anda mencoba merubah metode failover di KOMODOSQL3SQL2014 menjadi Automatic, maka akan tampil pesan kesalahan sebagai berikut:



Jumlah maksimum pasangan automatic failover yang diijinkan dalam availability groups hanya 1 pasang, sehingga hanya dua replica dapat menggunakan konfigurasi Automatic. Jika KomodoSQL3SQL2014 diubah ke mode synchronous-commit dengan failover secara Automatic dengan maka pesan kesalahan sebagai berikut akan ditampilkan:



Dengan konfigurasi sesuai pada gambar diatas, maka kondisi failover yang dapat dilakukan antara node adalah sebagai berikut:

- SQL Server instance KOMODOSQL1SQL2014 dan KOMODOSQL2SQL2014 merupakan pasangan automatic failover. Artinya perpindahan replica database antara kedua node tersebut dapat berlangsung secara automatic maupun planned manual failover (tanpa *data loss*)
- Perpindahan dengan target KOMODOSQL3SQL2014 hanya dapat dilakukan secara manual, yaitu forced manual failover. Berarti jika database di KOMODOSQL1SQL2014 mengalami masalah dan harus dipindahkan ke KOMODOSQL3SQL2014, maka proses tersebut harus dilakukan secara forced manual failover dengan konsekuensi terjadi kehilangan data

Pada bagian selanjutnya akan menjelaskan tiga jenis failover dalam availability groups beserta persyaratan yang diperlukan serta metode testing yang digunakan. Pemahaman tentang cara kerja failover sangat penting dalam melakukan perencanaan untuk mengetahui checklist tindakan yang diperlukan jika terjadi masalah. Pemahaman ini sangat diperlukan untuk menghindari kepanikan dan menentukan tindakan yang tepat.

Untuk memudahkan penjelasan maka konfigurasi berikut digunakan sebagai skenario awal dan menjadi dasar untuk melangkah ke sub bab berikutnya.

|SQL Server instance|Role |Availability mode|Failover mode|Database || KOMODOSQL1SQL2014
 |Primary | Synchronous commit |Automatic |AdventureWorks2014 SalesDB | KOMODOSQL2SQL2014
 |Secondary| Synchronous commit |Automatic | AdventureWorks2014 SalesDB | KOMODOSQL3SQL2014
 | Secondary | Asynchronous commit | Manual |AdventureWorks2014 SalesDB

Automatic Failover (tanpa data loss)

Kejadian failover otomatis merupakan kondisi ideal yang diharapkan dalam availability groups. Ketika terjadi masalah teknis di primary, database berpindah dari primary ke secondary server secara otomatis tanpa campur tangan administrator. Aplikasi klien tetap dapat mengakses database seperti sediakala karena koneksi ke listener memungkinkan pengalihan otomatis dari satu node ke node lain tanpa perlu mengetahui SQL Server instance mana yang sedang aktif sebagai primary.

Tahap-tahap terjadinya automatic failover adalah sebagai berikut:

- 1) Database di primary server (KOMODOSQL1SQL2014) mengalami masalah, misalnya karena file database *corrupt* atau layanan SQL Server *offline*.
- 2) Proses ping antara primary dan secondary menjadi terhenti, dan KOMODOSQL2SQL2014 tidak menerima respon balik dari proses ping yang dikirim ke KOMODOSQL1SQL2014
- 3) Database di KOMODOSQL2SQL2014 yang merupakan secondary replica akan dipromosikan menjadi primary database. Dengan demikian primary server berpindah ke KOMODOSQL2. Proses otomatis dapat terjadi dengan syarat database secondary berada dalam kondisi *synchronized*.[CA1]
- 4) Setelah KOMODOSQL2SQL2014 menjadi primary replica maka beberapa hal berikut dapat terjadi:
 - Jika KOMODOSQL1SQL2014 masih dapat berjalan baik maka node tersebut berubah peran menjadi secondary replica dengan konfigurasi Synchronous commit dan automatic failover
 - Jika KOMODOSQL1SQL2014 tidak dapat dapat diakses (down) maka node tersebut dianggap *unhealthy* dan berada pada kondisi error. Dashboard availability group akan menampilkan pesan error di node tersebut. Hal ini tidak mempengaruhi kelayakan primary replica yang telah berjalan di KOMODOSQL2SQL2014
- 5) KOMODOSQL3SQL2014 tetap berperan sebagai secondary replica dengan kondisi asynchronous dan manual failover.
- 6) Aplikasi klien tetap terkoneksi ke database AdventureWorks2014 dan SalesDB tanpa merubah connection string selama koneksi tersebut menggunakan availability listener.

Persyaratan untuk Automatic Failover

Automatic failover memerlukan beberapa persyaratan berikut untuk mendukung proses perpindahan role secara otomatis:

- 1) Database yang berperan sebagai secondary replica harus berada pada kondisi *synchronized*.
- 2) Minimal terdapat satu pasang replica dengan konfigurasi yang kompatibel untuk mendukung automatic failover. Dalam contoh konfigurasi yang digunakan dalam bab ini, baik KOMODOSQL1SQL2014 dan KOMODOSQL2SQL2014 memiliki konfigurasi yang memenuhi syarat tersebut yaitu:
 - Availability mode menggunakan Synchronous commit
 - Metode failover adalah Automatic

| Availability Replicas | | | | | |
|-----------------------|-----------|--------------------|---|---------------|---|
| Server Instance | Role | Availability Mode | | Failover Mode | |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | ▼ | Automatic | ▼ |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | ▼ | Automatic | ▼ |

3) Konfigurasi WSFC berada dalam kondisi baik dan memenuhi quorum. Apabila quorum tidak tercapai maka automatic failover tidak terjadi, karena proses tersebut hanya dapat dilakukan jika kondisi quorum terpenuhi.

4) Primary replica tidak dapat diakses dan telah melewati batas yang ditentukan di dalam *flexible failover policy*. Secara default, kondisi tersebut tercapai apabila secondary tidak menerima balasan ping dalam rentang waktu 10 detik. Proses failover terjadi apabila batas waktu tersebut terlewati dan primary tetap tidak dapat dijangkau.

Flexible failover policy merupakan Policy Management Framework (PMF) yang dikonfigurasi secara yang tersedia secara default ketika konfigurasi availability groups selesai dilakukan. Policy tersebut dapat diubah maupun ditambah sesuai kebutuhan. Penjelasan mengenai topic tersebut dapat diikuti di Bab 8.

1

Latihan 6.1. Konfigurasi dan Pengujian Automatic Failover[CA2]

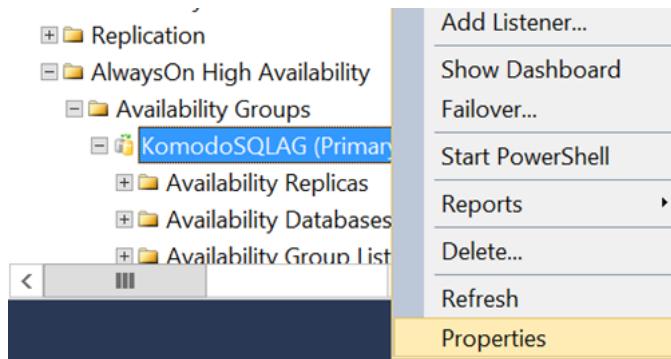
Di dalam latihan ini Anda akan melakukan 2 hal berikut:

- Memastikan konfigurasi replica untuk memastikan semua persyaratan automatic failover sudah terpenuhi.
- Simulasi automatic failover dengan cara mematikan instance SQL Server di KOMODOSQL1 yang merupakan primary server.
- Mematikan KOMODOSQL1SQL2014 akan memicu automatic failover dan primary replica berpindah ke KOMODOSQL2SQL2014.

Konfigurasi replica untuk mendukung automatic failover

Lakukan langkah-langkah berikut untuk menyiapkan automatic failover pada availability group.

- 1) Login ke KOMODOSQL1 dengan account **mycompany\sqlclsadmin**.
- 2) Buka SQL Server Management Studio dan tampilkan property availability group dengan membuka AlwaysOn High Availability > availability groups > KomodoSQLAG > **Properties**



3) Jadikan KOMODOSQL1\SQL2014 dan KOMODOSQL2\SQL2014 sebagai pasangan automatic failover dengan konfigurasi berikut:

| Availability Replicas | | | | | | |
|-----------------------|-----------|---------------------|---------------|-----------------------------|--------------------|--|
| Server Instance | Role | Availability Mode | Failover Mode | Connections in Primary Role | Readable Secondary | |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | Automatic | Allow all conn... | Yes | |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | Automatic | Allow all conn... | Read-intent only | |
| KOMODOSQL3\SQL2014 | Secondary | Asynchronous com... | Manual | Allow all conn... | Read-intent only | |

4) Konfigurasi **Connections in Primary Role** dan **Readable Secondary** tidak mempengaruhi kemampuan automatic failover sehingga dapat diabaikan untuk saat ini. Klik OK untuk menyimpan konfigurasi dan lanjutkan ke langkah berikutnya untuk memeriksa healthy state tiap replica.

5) Tampilkan dashboard availability group dengan membuka AlwaysOn High Availability > availability groups > KomodoSQLAG > Show Dashboard.

| Name | Role | Failover Mode | Synchronization State | Issues |
|--------------------|-----------|---------------|-----------------------|--------|
| KOMODOSQL1\SQL2014 | Primary | Automatic | Synchronized | |
| KOMODOSQL2\SQL2014 | Secondary | Automatic | Synchronized | |
| KOMODOSQL3\SQL2014 | Secondary | Manual | Synchronized | |

| Name | Replica | Synchronization State | Failover Readi... |
|--------------------|--------------------|-----------------------|-------------------|
| KOMODOSQL1\SQL2014 | | | |
| AdventureWorks2014 | KOMODOSQL1\SQL2014 | Synchronized | No Data Loss |
| KOMODOSQL2\SQL2014 | | | |
| AdventureWorks2014 | KOMODOSQL2\SQL2014 | Synchronized | No Data Loss |
| KOMODOSQL3\SQL2014 | | | |
| AdventureWorks2014 | KOMODOSQL3\SQL2014 | Synchronized | No Data Loss |

Berdasarkan laporan dashboard tersebut, dapat ditarik kesimpulan hal-hal berikut:

- Server secondary KOMODOSQL2SQL2014 yang menjadi target automatic failover berada dalam kondisi baik dan *Synchronized*.
- Database AdventureWorks2014 berada dalam kondisi baik dan *Synchronized*, baik di primary maupun secondary
- Failover Readiness menunjukkan kesiapan **No Data Loss** karena semuanya dikonfigurasi menggunakan mode Synchronous commit.

*Berdasarkan kondisi tersebut, konfigurasi availability group telah memenuhi kondisi untuk terjadinya automatic failover. Proses automatic failover tidak dapat dilakukan jika replica yang terlibat masih berada dalam kondisi *Synchronizing*. Kondisi tersebut biasa terjadi jika jumlah transaksi sangat besar atau terjadi masalah jaringan sehingga menghambat sinkronisasi antar replica.*

6) Jalankan perintah PowerShell berikut untuk memeriksa kesiapan quorum di cluster tersebut:

1 Get-ClusterQuorum | FL

Testing Automatic Failover

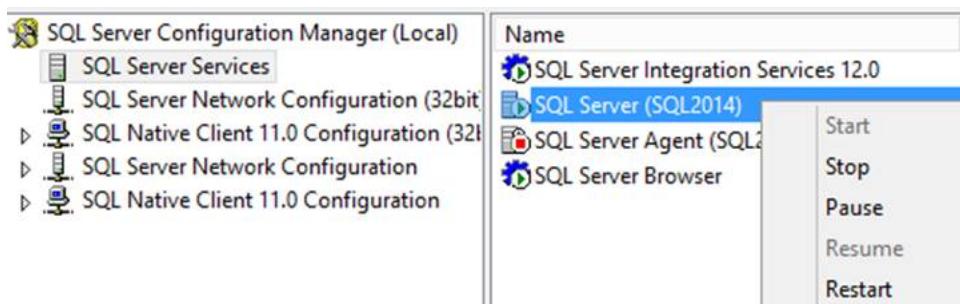
Untuk menguji apakah failover dapat dipicu secara otomatis, Anda akan melakukan hal-hal berikut:

- Mematikan service SQL Server di KOMODOSQL1SQL2014 untuk memicu perpindahan primary replica secara otomatis dari ke KOMODOSQL2SQL2014
- Menghidupkan kembali KOMODOSQL1SQL2014 dan mempelajari kondisi availability groups setelah server yang mati atau bermasalah diperbaiki dan berfungsi kembali

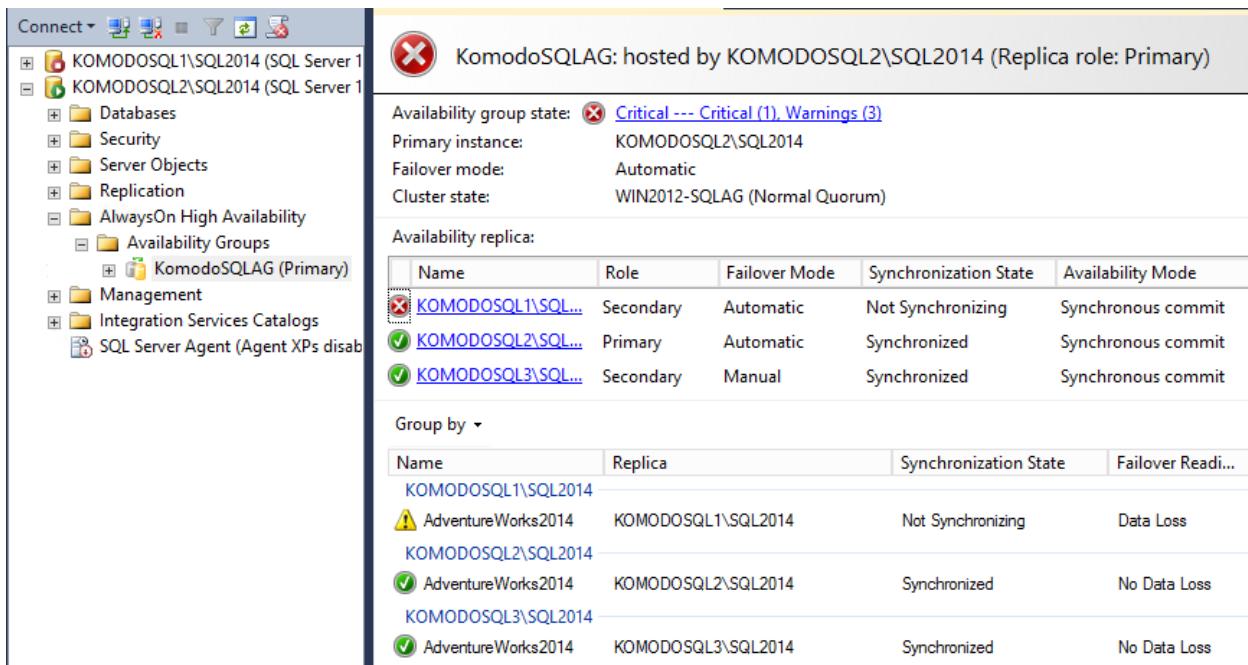
Lakukan langkah-langkah berikut untuk melakukan pengujian automatic failover:

1) Login ke KOMODOSQL1 dan buka SQL Server Configuration Manager

2) Matikan service SQL Server dengan mengklik kanan > Stop



3) Tunggu beberapa saat dan pantau perpindahan primary replica dari dashboard availability group. Terlihat bahwa KOMODOSQL2SQL2014 telah dipromosikan menjadi **primary** secara otomatis dan database Adventureworks2014 dapat diakses di node tersebut.



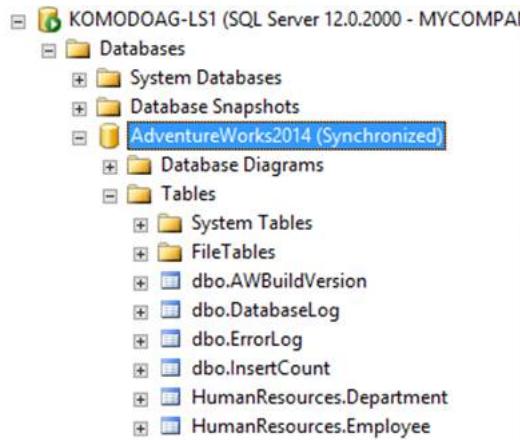
Catatan: Untuk menampilkan dashboard diatas, Anda harus terkoneksi ke KOMODOSQL2SQL2014 karena KOMODOSQL1SQL2014 saat ini sedang dimatikan.

Berdasarkan dashboard tersebut, dapat dibaca bahwa KOMODOSQL1SQL2014 berada dalam kondisi *Not Synchronizing* dan availability group berada dalam kondisi *Critical* karena salah satu node yang merupakan pasangan automatic failover sedang bermasalah.

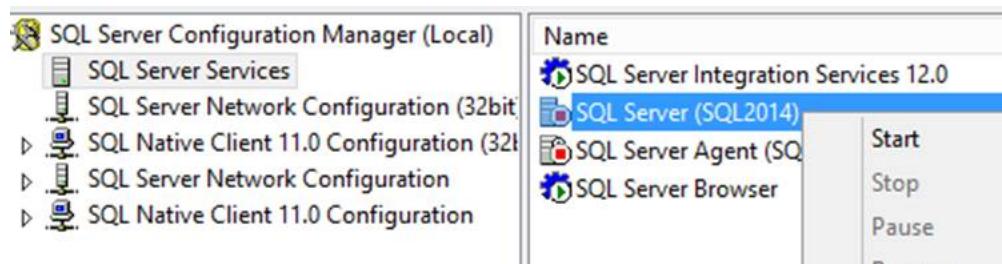
4) Untuk membuktikan bahwa Anda tetap dapat mengakses database meskipun sudah berpindah node, buka SQL Server Management Studio dan hubungkan ke listener KOMODOAG-LS1 sebagai berikut:



5) SSMS akan terhubung ke database AdventureWorks2014 tanpa perlu mengetahui bahwa primary replica telah berpindah dari KOMODOSQL1SQL2014 ke KOMODOSQL2SQL2014.



6) Langkah berikutnya adalah menghidupkan kembali instance KOMODOSQL1SQL2014 dan melihat dampaknya pada konfigurasi availability groups. Buka SQL Server Configuration Manager di KOMODOSQL1 dan hidupkan kembali service SQL Server.



7) Tunggu beberapa saat dan amati perubahan yang terjadi pada dashboard availability groups. Apabila semuanya berjalan baik, dashboard akan menampilkan data berikut:

| Name | Role | Failover Mode | Synchronization State | Availability Mode |
|------------------------------------|-----------|---------------|-----------------------|--------------------|
| KOMODOSQL1\SQL2014 | Secondary | Automatic | Synchronized | Synchronous commit |
| KOMODOSQL2\SQL2014 | Primary | Automatic | Synchronized | Synchronous commit |
| KOMODOSQL3\SQL2014 | Secondary | Manual | Synchronized | Synchronous commit |

| Name | Replica | Synchronization State | Failover Readi... | Issues |
|------------------------------------|--------------------|-----------------------|-------------------|--------|
| KOMODOSQL1\SQL2014 | | | | |
| AdventureWorks2014 | KOMODOSQL1\SQL2014 | Synchronized | No Data Loss | |
| KOMODOSQL2\SQL2014 | | | | |
| AdventureWorks2014 | KOMODOSQL2\SQL2014 | Synchronized | No Data Loss | |
| KOMODOSQL3\SQL2014 | | | | |
| AdventureWorks2014 | KOMODOSQL3\SQL2014 | Synchronized | No Data Loss | |

Setelah KOMODOSQL2SQL2014 dihidupkan kembali, server tersebut kembali bergabung ke availability groups sebagai **secondary** dan menjadi pasangan automatic failover untuk KOMODOSQL2SQL2014 yang saat ini telah berubah peran menjadi primary. Konfigurasi availability groups juga kembali berada ke kondisi *Healthy state*.

Primary server yang bermasalah dan dapat diperbaiki kembali akan bergabung ke availability groups sebagai secondary, dan tidak menjadi primary sebagaimana posisi semula sebelum mengalami masalah.

Latihan lanjutan

Anda dapat mengembalikan primary ke KOMODOSQL1SQL2014 dengan melakukan kebalikan dari langkah-langkah yang telah dijelaskan diatas. Lakukan latihan lanjutan sebagai berikut:

- Matikan service SQL Server di KOMODOSQL2SQL2014 dan tunggu sampai primary berpindah ke KOMODOSQL1SQL2014
- Hidupkan kembali SQL Server instance KOMODOSQL2SQL2014

Planned Manual Failover (tanpa data loss)

Proses planned manual failover merupakan kejadian failover yang terencana, dengan melakukan prosedur perpindahan melalui salah satu mekanisme berikut:

- Menggunakan failover wizard pada SQL Server Management Studio

- Menggunakan perintah T-SQL
- Menggunakan script Powershell

Berbeda dengan automatic failover yang berguna untuk mengantisipasi kejadian tidak terduga, planned manual failover merupakan kegiatan terencana yang umumnya menjadi prosedur rutin dalam operasi server di *production*. Berikut adalah beberapa kondisi yang membutuhkan dilakukannya planned manual failover:

- **Menghindari downtime saat proses maintenance server.** Proses maintenance dapat berupa hardware maupun software, seperti penggantian perangkat keras atau update hotfix pada sistem operasi. Node yang akan diberi tindakan dinon-aktifkan dengan cara manual failover dan memindahkan primary server ke node lain.
- **Pengujian rutin untuk high availability.** Konfigurasi availability group perlu diuji secara rutin untuk memastikan kemampuan failover dapat berjalan baik. Pengujian dilakukan diluar jam sibuk dengan jadwal yang telah ditetapkan.

Mengingat planned manual failover dilakukan dengan tujuan tanpa *data loss*, maka baik primary maupun secondary harus berada pada kondisi Synchronous commit. Sebagai contoh, konfigurasi berikut digunakan sebagai dasar untuk melakukan failover:

| Availability Replicas | | | | |
|-----------------------|-----------|---------------------|---------------|-----------|
| Server Instance | Role | Availability Mode | Failover Mode | |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | Automatic | Automatic |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | Manual | Manual |
| KOMODOSQL3\SQL2014 | Secondary | Asynchronous commit | Manual | Manual |

Primary replica yang saat ini berada di KOMODOSQL1SQL2014 akan dipindahkan ke KOMODOSQL2SQL2014 dengan metode manual failover tanpa *data loss*.

- Perlu diingat bahwa manual failover dapat dilakukan terhadap replica yang memnggunakan Failover Mode automatic maupun manual.
- KOMODOSQL2SQL2014 memenuhi syarat sebagai target manual failover tanpa *data loss* karena menggunakan mode Synchronous commit.

Cara kerja planned manual failover tanpa data loss adalah sebagai berikut:

- 1) Setelah administrator memulai proses failover, WSFC kemudian menonaktifkan primary replica untuk memastikan tidak ada transaksi dari pengguna yang mengakses primary replica.
- 2) Secondary replica di KOMODOSQL2SQL2014 harus menyelesaikan proses *rolling forward* terhadap semua log yang masih menunggu untuk diproses, sehingga posisi data di secondary sama dengan primary.

- 3) Secondary replica dinaikkan statusnya menjadi primary replica setelah semua log selesai diproses.
- 4) Database di KOMODOSQL2SQL2014 berubah menjadi primary replica sedangkan database di KOMODOSQL1SQL2014 turun statusnya menjadi secondary replica.

Persyaratan untuk planned manual failover (tanpa data loss)

Hal-hal berikut harus dipenuhi untuk memastikan proses manual failover berjalan dengan baik:

- Pasangan primary replica yang terlibat harus berada pada mode Synchronous commit dengan metode failover automatic atau manual.
- Minimal terdapat satu secondary replica yang akan menjadi target failover dengan mode Synchronous commit.
- Secondary replica yang menjadi target harus berada pada kondisi *Synchronized*.

Apabila secondary replica yang dijadikan target tidak berada pada kondisi *Synchronized* maka harus dilakukan forced manual failover. Tindakan tersebut menyebabkan resiko kehilangan data karena secondary tidak berada dalam kondisi *synchronized*.

Dashboard berikut menampilkan kondisi 3 replica yang siap untuk melakukan manual failover tanpa data loss.

KomodoSQLAG: hosted by KOMODOSQL1\SQL2014 (Replica role: Primary)

Availability group state: ✖ Critical --- Critical (1)

Primary instance: KOMODOSQL1\SQL2014

Failover mode: Automatic

Cluster state: WIN2012-SQLAG (Normal Quorum)

Availability replica:

| Name | Role | Failover Mode | Synchronization State | Availability Mo... | Issues |
|------------------------------------|-----------|---------------|-----------------------|--------------------|--------|
| KOMODOSQL1\SQL2014 | Primary | Automatic | Synchronized | Synchronous c... | |
| KOMODOSQL3\SQL2014 | Secondary | Manual | Synchronizing | Asynchronous ... | |
| KOMODOSQL2\SQL2014 | Secondary | Manual | Synchronized | Synchronous c... | |

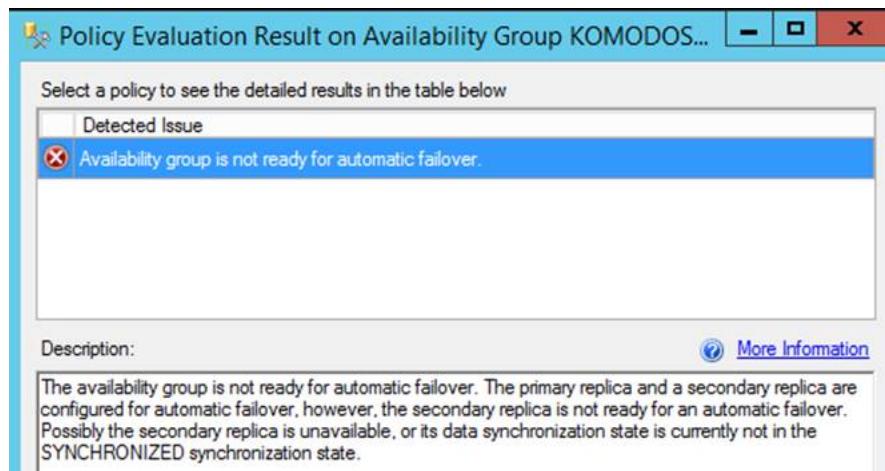
Group by ▾

| Name | Replica | Synchronization State | Failover Readi... | Issues |
|------------------------------------|--------------------|-----------------------|-------------------|--------|
| KOMODOSQL1\SQL2014 | | | | |
| AdventureWorks2014 | KOMODOSQL1\SQL2014 | Synchronized | No Data Loss | |
| SalesDB | KOMODOSQL1\SQL2014 | Synchronized | No Data Loss | |
| KOMODOSQL2\SQL2014 | | | | |
| AdventureWorks2014 | KOMODOSQL2\SQL2014 | Synchronized | No Data Loss | |
| SalesDB | KOMODOSQL2\SQL2014 | Synchronized | No Data Loss | |
| KOMODOSQL3\SQL2014 | | | | |
| AdventureWorks2014 | KOMODOSQL3\SQL2014 | Synchronizing | Data Loss | |
| SalesDB | KOMODOSQL3\SQL2014 | Synchronizing | Data Loss | |

Berdasarkan dashboard tersebut dapat disarikan beberapa hal berikut:

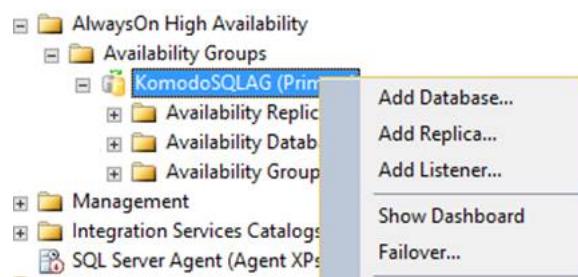
- KOMODOSQL2SQL2014 yang saat ini menjadi secondary merupakan target failover karena berada pada kondisi Synchronized
- KOMODOSQL3SQL2014 tidak dapat dijadikan target failover tanpa *data loss* karena berada pada kondisi Synchronizing. Kondisi tersebut terjadi karena mode availability yang digunakan adalah Asynchronous
- Tingkat kesehatan availability group berada pada kondisi critical karena tidak mungkin melakukan automatic failover. Saat ini tidak terdapat pasangan automatic failover karena hanya satu server yang menggunakan metode automatic.

Jika Anda mengklik link Critical maka pesan kesalahan berikut akan tampil:



Konfigurasi tersebut tidak mendukung automatic failover, tetapi tetap dapat melakukan manual failover tanpa *data loss*.

Cara termudah untuk melakukan failover adalah menggunakan failover wizard yang terdapat di dalam SQL Server Management Studio. Buka menu AlwaysOn High Availability > Availability Groups > KomodoSQLAG > Failover sebagaimana gambar berikut:



Latihan 6.2. Pengujian Planned Manual Failover (tanpa data loss)

Di dalam latihan ini Anda akan melakukan 2 hal berikut:

- Konfigurasi replica untuk memenuhi persyaratan planned manual failover tanpa data loss
- Simulasi planned manual failover

Lakukan langkah-langkah berikut untuk melakukan konfigurasi dan pengujian.

1) Login ke mesin yang saat ini menjadi primary, misalnya KOMODOSQL1 dengan menggunakan account **mycompany\sqlclsadmin**.

2) Buka property availability groups dan ubah konfigurasi replica sesuai dengan gambar berikut:

| Availability Replicas | | | | | | | |
|-----------------------|-----------|---------------------|-----------|---------------|-----------------------|-----------------------------|--------------------|
| Server Instance | Role | Availability Mode | | Failover Mode | | Connections in Primary Role | |
| | | Synchronous commit | Automatic | Manual | Allow all connections | Yes | Readable Secondary |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | Automatic | Manual | Allow all connections | Yes | Readable Secondary |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | Automatic | Manual | Allow all connections | Read-intent only | Readable Secondary |
| KOMODOSQL3\SQL2014 | Secondary | Asynchronous commit | Automatic | Manual | Allow all connections | Read-intent only | Readable Secondary |

Point terpenting pada konfigurasi diatas adalah harus terdapat minimum satu pasang replica (primary dan secondary) yang menggunakan mode Synchronous commit. Mode failover dapat berupa automatic maupun manual.

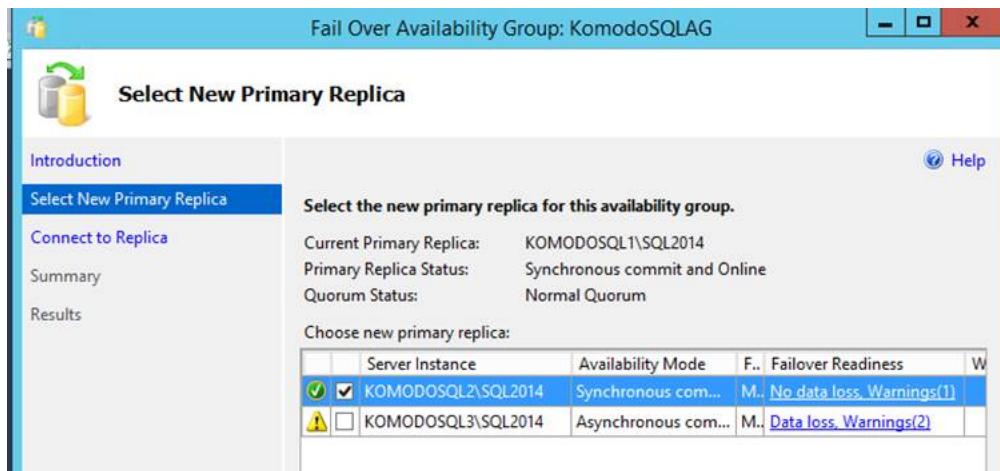
3) Buka dashboard availability group dan pastikan bahwa replica yang berada di KOMODOSQL1\SQL2014 maupun KOMODOSQL2\SQL2014 berada pada kondisi synchronized sebagaimana gambar berikut:

| Name | Replica | Synchronization State | Failover Readi... |
|--------------------|--------------------|-----------------------|-------------------|
| KOMODOSQL1\SQL2014 | | | |
| AdventureWorks2014 | KOMODOSQL1\SQL2014 | Synchronized | No Data Loss |
| SalesDB | KOMODOSQL1\SQL2014 | Synchronized | No Data Loss |
| KOMODOSQL2\SQL2014 | | | |
| AdventureWorks2014 | KOMODOSQL2\SQL2014 | Synchronized | No Data Loss |
| SalesDB | KOMODOSQL2\SQL2014 | Synchronized | No Data Loss |
| KOMODOSQL3\SQL2014 | | | |
| AdventureWorks2014 | KOMODOSQL3\SQL2014 | Synchronizing | Data Loss |
| SalesDB | KOMODOSQL3\SQL2014 | Synchronizing | Data Loss |

Gambar tersebut menampilkan terdapat 2 database dalam satu kelompok availability groups yaitu AdventureWorks2014 dan SalesDB. Keduanya akan dipindahkan ke secondary secara bersamaan karena berada dalam satu kelompok.

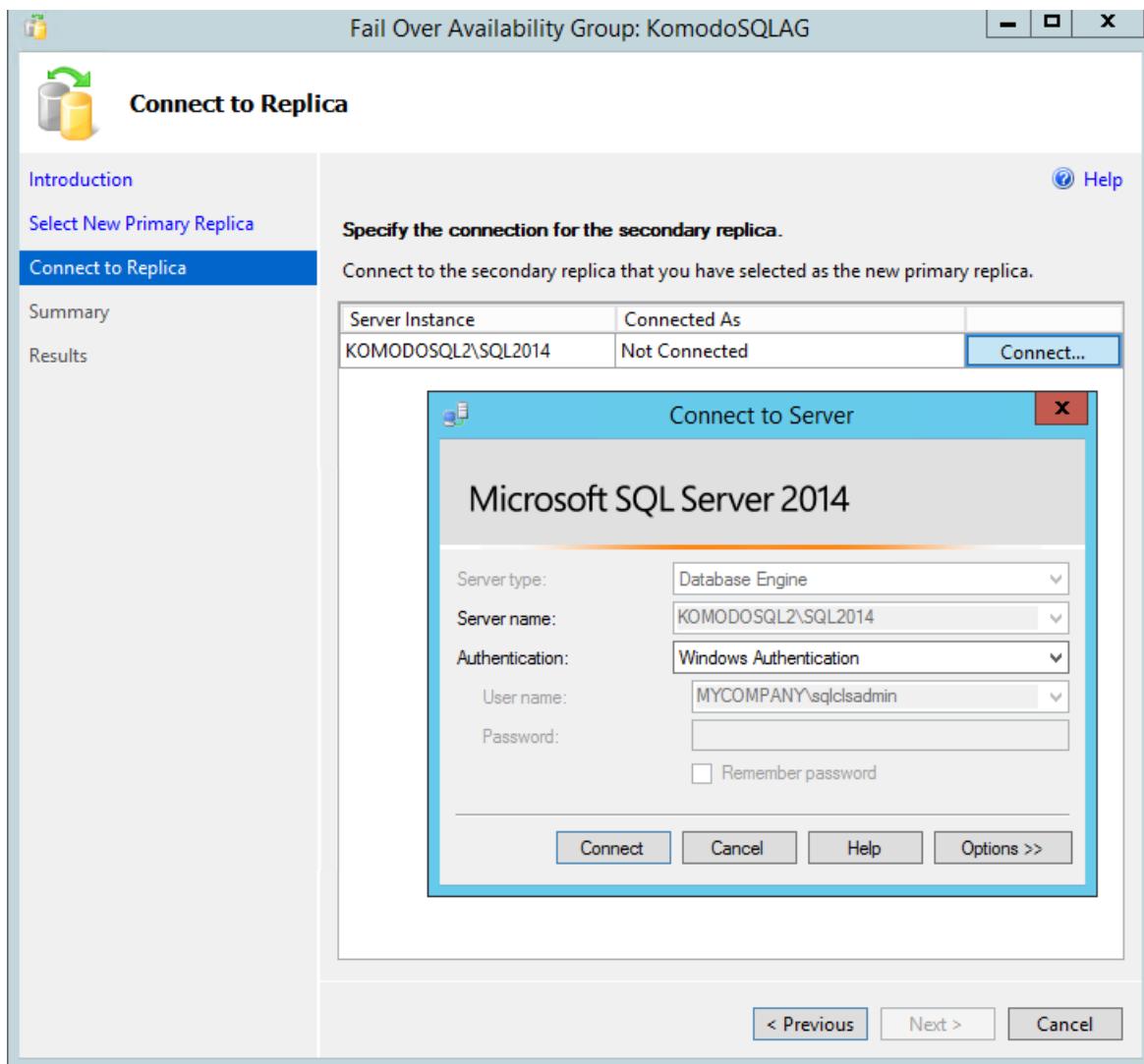
4) Untuk melakukan failover terencana menggunakan SQL Server Management Studio, buka menu **AlwaysOn High Availability > Availability Groups > KomodoSQLAG > Failover**

5) Menu failover wizard akan terbuka dan menampilkan daftar replica yang dapat dijadikan target failover sebagai berikut:



KOMODOSQL2\SQL2014 merupakan target ideal karena menggunakan mode Synchronous commit dan berada pada kondisi Synchronized. Pilih node tersebut dan klik Next untuk melanjutkan.

6) Klik Connect dan pastikan KOMODOSQL2\SQL2014 tampil sebagai Server name. Klik Connect dan Next untuk melanjutkan.



- 7) Bagian terakhir menampilkan summary dari konfigurasi yang telah dilakukan. Klik Script untuk membuat T-SQL yang juga dapat digunakan untuk melakukan failover.
- 8) Klik Finish untuk mengesekusi proses failover. Selanjutnya Anda dapat membuka dashboard untuk memantau hasil akhir proses tersebut.

Proses failover tersebut berhasil dan KOMODOSQL2SQL2014 saat ini mengambil alih peran sebagai primary sedangkan KOMODOSQL1SQL2014 berubah menjadi secondary.

Latihan 6.3. Memantau Proses Failover dengan Availability Groups Demonstrator

Availability Groups Demonstrator adalah utilitas yang dapat digunakan untuk menguji koneksi dari aplikasi klien ke availability groups. Tool tersebut dibuat oleh tim SQLSkills dan dapat diunduh gratis. Anda dapat menggunakannya untuk melihat pengaruh proses failover terhadap aplikasi klien.

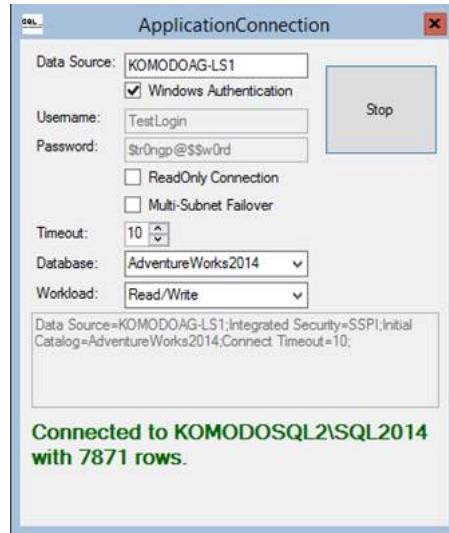
Tool tersebut dapat diunduh dari lokasi berikut:

<https://www.sqlskills.com/blogs/jonathan/updated-availability-group-demonstrator>

Untuk melakukan prosedur pengujian dibawah, koneksi harus diarahkan ke availability listener yaitu KOMODOAG-LS1.

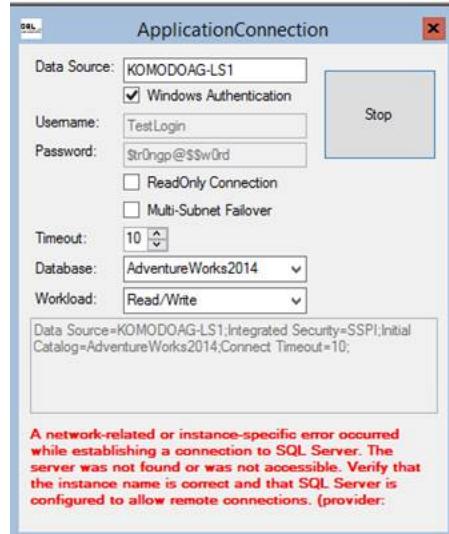
- 1) Ulangi proses diatas dengan melakukan failover dari KOMODOSQL2SQL2014 ke KOMODOSQL1SQL2014
- 2) Pada saat Anda melakukan failover tersebut, gunakan untuk Availability Groups Demonstrator mengamati apa yang terjadi terhadap aplikasi tersebut saat proses failover sedang terjadi

3) Berikut adalah kondisi sebelum failover dimana tester terkoneksi ke KOMODOSQL2SQL2014 yang merupakan primary server:

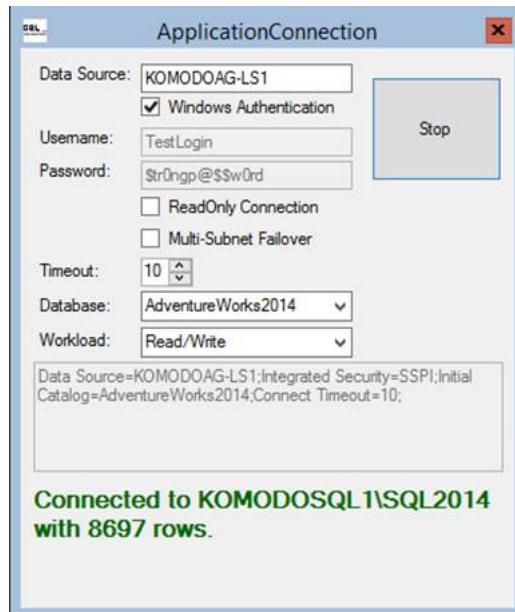


Tester tersebut terkoneksi ke KOMODOSQL2SQL2014 yang merupakan primary replica.

4) Pada saat failover sedang terjadi dan database tidak terjangkau, aplikasi akan menampilkan pesan berikut:



5) Setelah proses failover selesai, aplikasi berjalan normal kembali dan terkoneksi ke KOMODOSQL1SQL2014 yang merupakan primary server yang baru.



Jendela tersebut menampilkan informasi bahwa aplikasi saat ini terkoneksi ke KOMODOSQL1\SQL2014 yang merupakan primary replica.

Latihan 6.4. Planned Manual Failover dengan T-SQL

Metode yang lebih cepat untuk melakukan manual failover adalah menggunakan perintah T-SQL yang dieksekusi terhadap secondary server yang menjadi target failover. Anda dapat mengulangi latihan 6.2 dan mengganti prosedur failover wizard dengan T-SQL berikut:

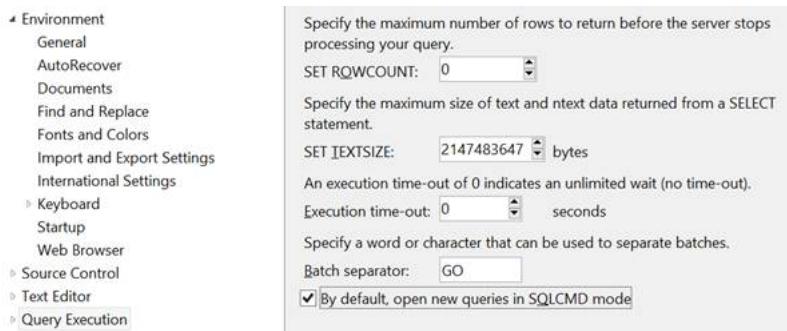
```

1  ---Script berikut harus dieksekusi menggunakan SQLCMD MODE
2
3  :Connect KOMODOSQL2\SQL2014
4
5  ALTER AVAILABILITY GROUP [KomodoSQLAG] FAILOVER;
```

Script diatas merubah peran KOMODOSQL2\SQL2014 dari secondary menjadi primary. Dengan demikian semua database yang terdapat di dalamnya dan masuk ke dalam kelompok availability groups KomodoSQLAG juga dinaikkan perannya menjadi primary replica.

Catatan:

Untuk menjalankan script dengan SQLCMD MODE, Anda harus merubah metode eksekusi script di SQL Server Management Studio. Buka Tools > Options > Query Execution dan pilih **SQLCMD mode**.



Forced Manual Failover (dengan data loss)

Metode ini dijalankan terhadap secondary replica yang tidak berada dalam kondisi Synchronized sehingga dapat terjadi kehilangan data. Forced manual failover sebaiknya dijalankan sebagai pilihan terakhir apabila metode planned manual failover tidak memungkinkan untuk dilakukan.

Ada beberapa kondisi darurat yang menyebabkan metode ini harus dilakukan, antara lain:

- Primary replica mengalami masalah dan tidak dapat dijangkau, namun automatic failover mengalami kegagalan karena semua secondary database tidak mencapai kondisi Synchronized. Keadaan tersebut mengharuskan administrator menaikkan status salah satu secondary replica menjadi primary dengan metode forced manual failover.
- Terjadi masalah kehilangan quorum pada WSFC sehingga harus dilakukan forced quorum untuk memperbaiki konfigurasi Windows Cluster. Masalah tersebut menyebabkan integrity data antar replica tidak dapat dijamin sehingga perlu dilakukan forced failover.
- Terjadi kondisi bencana yang menyebabkan data center utama tidak berfungsi. Pasangan automatic failover juga terdapat di dalam data center tersebut, sehingga perlu dilakukan prosedur disaster recovery untuk mengaktifkan secondary replica yang berada di lokasi aman. Replica di lokasi aman biasanya menggunakan Asynchronous commit karena jarak yang jauh, sehingga hanya dapat menggunakan forced manual failover.
- Terjadi masalah sinkronisasi antar replica yang menyebabkan semua replica dalam availability groups berada pada kondisi *resolving*. Pada kondisi tersebut tidak dapat diidentifikasi dengan jelas replica mana yang berperan sebagai primary maupun secondary.

Secara garis besar, forced manual failover merupakan teknik disaster recovery yang harus dilakukan dengan hati-hati karena adanya resiko kehilangan data.

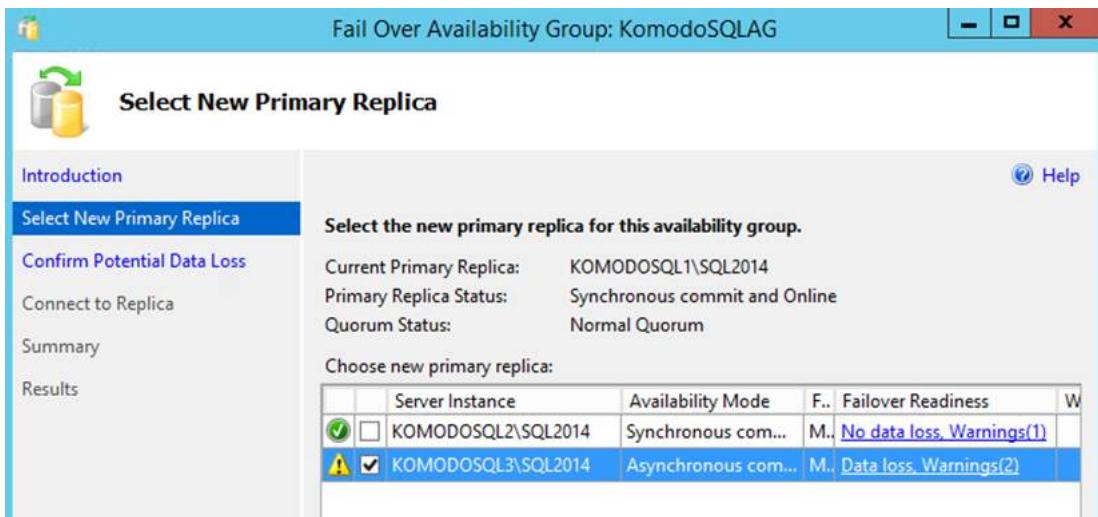
Cara kerja forced manual failover adalah sebagai berikut

- 1) Replica yang menjadi target menerima perintah untuk *forced failover* dan dipromosikan menjadi primary. Replica tersebut harus berada pada kondisi resolving atau berperan sebagai secondary.
- 2) Replica tersebut berpindah peran menjadi primary dan siap melayani akses data dari klien.

- 3) Apabila primary replica yang tengah bermasalah aktif kembali, maka akan menjadi secondary replica dengan status *Suspended*. Replica tersebut tidak berada pada kondisi sehat karena mata rantai sinkronisasi data telah terputus saat forced manual failover terjadi.
- 4) Semua replica lain kecuali primary replica akan berada pada kondisi *Suspended*. Hal tersebut terjadi karena tidak terjadi sinkronisasi data secara benar antar replica. Database tersebut harus diaktifkan kembali secara manual dengan prosedur *R_Resume Data Movement*.

Metode Forced Manual Failover dan Resume Data Movement

Proses forced manual failover dilakukan menggunakan failover wizard dengan memilih database yang berada dalam kondisi Asynchronous commit sebagai target sebagaimana gambar dibawah.



Berdasarkan gambar diatas terlihat bahwa KOMODOSQL3SQL2014 merupakan secondary replica dengan mode Asynchronous commit sehingga sesuai dijadikan target forced manual failover. Anda juga dapat menggunakan script T-SQL untuk melakukan forced manual failover. Script berikut dijalankan terhadap replica yang menjadi target (KOMODOSQL3SQL2014) untuk dinaikkan perannya menjadi primary:

```

1  \--- YOU MUST EXECUTE THE FOLLOWING SCRIPT IN SQLCMD MODE .
2  :Connect KOMODOSQL3\SQL2014
3  ALTER AVAILABILITY GROUP [KomodoSQLAG] FORCE_FAILOVER_ALLOW_DATA_LOSS;

```

Bagian terpenting dari script diatas adalah `FORCE_FAILOVER_ALLOW_DATA_LOSS` yang memicu terjadinya failover meskipun kondisi database tidak mencapa *Synchronized*.

Setelah proses failover selesai maka KOMODOSQL3SQL2014 berubah menjadi primary sedangkan semua node yang lain menjadi secondary dengan kondisi *Suspended*. Dashboard dibawah menampilkan contoh KOMODOSQL3SQL2014 yang saat ini menjadi primary sedangkan semua replica lain berstatus *Suspended*.

KomodoSQLAG: hosted by KOMODOSQL3\SQL2014 (Replica role: Primary)

Availability group state: ! [Warning --- Warnings \(2\)](#)

Primary instance: KOMODOSQL3\SQL2014

Failover mode: Manual

Cluster state: WIN2012-SQLAG (Normal Quorum)

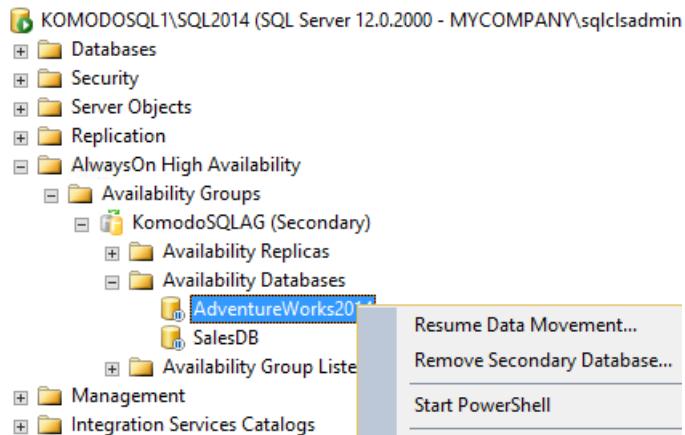
Availability replica:

| Name | Role | Failover Mode | Synchronization State | Availability Mo... |
|--|-----------|---------------|-----------------------|--------------------|
| ! KOMODOSQL1\SQL2014 | Secondary | Automatic | Not Synchronizing | Synchronous c... |
| ! KOMODOSQL2\SQL2014 | Secondary | Manual | Not Synchronizing | Synchronous c... |
| ✓ KOMODOSQL3\SQL2014 | Primary | Manual | Synchronized | Asynchronous ... |

Group by ▾

| Name | Replica | Synchronization State | Failover Readi... | Suspended |
|--|--------------------|-----------------------|-------------------|-----------|
| KOMODOSQL1\SQL2014 | | | | |
| ! AdventureWorks2014 | KOMODOSQL1\SQL2014 | Not Synchronizing | Data Loss | Suspended |
| ! SalesDB | KOMODOSQL1\SQL2014 | Not Synchronizing | Data Loss | Suspended |
| KOMODOSQL2\SQL2014 | | | | |
| ! AdventureWorks2014 | KOMODOSQL2\SQL2014 | Not Synchronizing | Data Loss | Suspended |
| ! SalesDB | KOMODOSQL2\SQL2014 | Not Synchronizing | Data Loss | Suspended |
| KOMODOSQL3\SQL2014 | | | | |
| ✓ AdventureWorks2014 | KOMODOSQL3\SQL2014 | Synchronized | No Data Loss | Resumed |
| ✓ SalesDB | KOMODOSQL3\SQL2014 | Synchronized | No Data Loss | Resumed |

Semua database berstatus *Suspended* harus diaktifkan kembali dengan cara **Resume Data Movement** sebagai berikut.



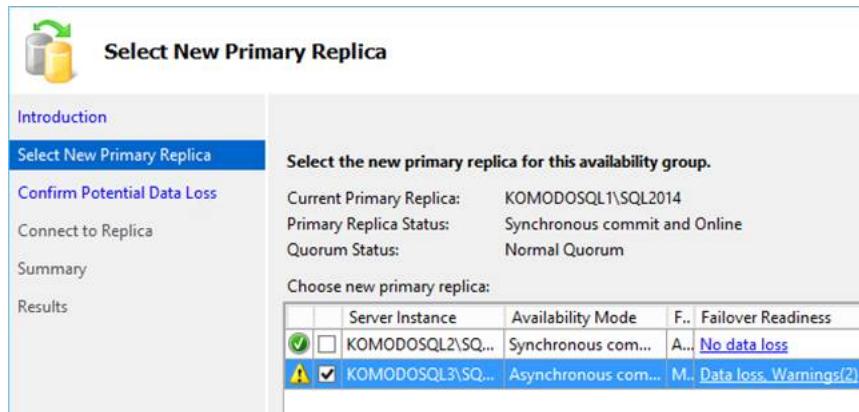
Latihan 6.5. Forced Manual Failover dengan data loss

Dalam latihan ini Anda akan melakukan prosedur forced manual failover dan resume data transfer sehingga semua replica dapat berjalan normal.

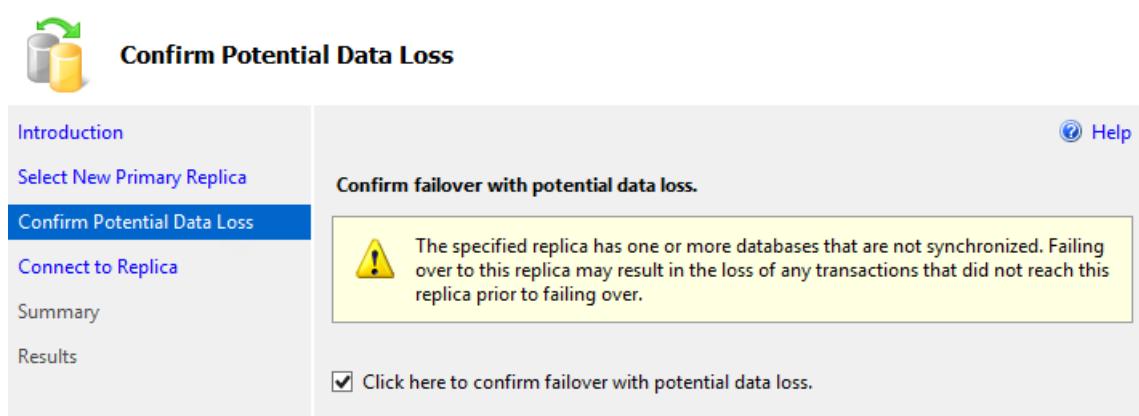
- 1) Login ke KOMODOSQL1SQL2014 dengan menggunakan account **mycompany\sqlclsadmin**.
- 2) Siapkan konfigurasi replica untuk simulasi forced manual failover dengan menggunakan sesuai gambar berikut.

| Availability Replicas | | | | |
|-----------------------|-----------|---------------------|---------------|-----------|
| Server Instance | Role | Availability Mode | Failover Mode | |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | Automatic | Automatic |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | Automatic | Automatic |
| KOMODOSQL3\SQL2014 | Secondary | Asynchronous commit | Manual | Manual |

3) Jalankan failover wizard dan pilih KOMODOSQL3SQL2014 sebagai target. Node tersebut berjalan dengan mode asynchronous sehingga memerlukan forced manual failover.



4) Terima konfirmasi bahwa akan terjadi kehilangan data, lalu klik next untuk melanjutkan.



5) Anda juga dapat menggunakan T-SQL berikut untuk melakukan forced failover dengan KOMODOSQL3SQL2014 sebagai target:

```

1  \--- YOU MUST EXECUTE THE FOLLOWING SCRIPT IN SQLCMD MODE.
2  :Connect KOMODOSQL3\SQL2014
3  ALTER AVAILABILITY GROUP [KomodoSQLAG] FORCE_FAILOVER_ALLOW_DATA_LOSS;

```

6) Setelah proses failover selesai maka dashboard menampilkan kondisi akhir semua replica. Dapat terlihat bahwa KOMODOSQL3SQL2014 telah berubah peran menjadi primary dan node lainnya

sebagai secondary. Semua database di secondary berada pada kondisi *Suspended* karena proses forced failover tidak menjaga sinkronisasi data antar replica.

KomodoSQLAG: hosted by KOMODOSQL3\SQL2014 (Replica role: Primary)

| | |
|---------------------------|--|
| Availability group state: | ! Warning --- Warnings (2) |
| Primary instance: | KOMODOSQL3\SQL2014 |
| Failover mode: | Manual |
| Cluster state: | WIN2012-SQLAG (Normal Quorum) |

Availability replica:

| Name | Role | Failover Mode | Synchronization State | Availability Mode |
|--|----------|---------------|-----------------------|---------------------|
| ! KOMODOSQL1\SQL2014 | Secon... | Automatic | Not Synchronizing | Synchronous commit |
| ! KOMODOSQL2\SQL2014 | Secon... | Automatic | Not Synchronizing | Synchronous commit |
| ✓ KOMODOSQL3\SQL2014 | Primary | Manual | Synchronized | Asynchronous commit |

Group by ▾

| Name | Replica | Synchronization State | Failover Readi... | Suspended |
|--|---------------------|-----------------------|-------------------|-----------|
| KOMODOSQL1\SQL2014 | | | | |
| ! AdventureWorks2014 | KOMODOSQL1\SQL20... | Not Synchronizing | Data Loss | Suspended |
| ! SalesDB | KOMODOSQL1\SQL20... | Not Synchronizing | Data Loss | Suspended |
| KOMODOSQL2\SQL2014 | | | | |
| ! AdventureWorks2014 | KOMODOSQL2\SQL20... | Not Synchronizing | Data Loss | Suspended |
| ! SalesDB | KOMODOSQL2\SQL20... | Not Synchronizing | Data Loss | Suspended |
| KOMODOSQL3\SQL2014 | | | | |
| ✓ AdventureWorks2014 | KOMODOSQL3\SQL20... | Synchronized | No Data Loss | Resumed |
| ✓ SalesDB | KOMODOSQL3\SQL20... | Synchronized | No Data Loss | Resumed |

7) Langkah berikutnya adalah mengaktifkan kembali secondary replica sehingga tidak berada pada kondisi *Suspended*. Dengan menggunakan SQL Server Management Studio, buka SQL Server instance KOMODOSQL1SQL2014 > AlwaysOn High Availability > Availability Groups > KomodoSQLAG > Availability Databases > AdventureWorks2014 > **Resume Data Movement**.

The screenshot shows the SQL Server Management Studio (SSMS) interface. The left pane displays a tree structure of the database instance 'KOMODOSQL1\SQL2014'. The 'AlwaysOn High Availability' node is expanded, showing the 'Availability Groups' node, which in turn shows the 'KomodoSQLAG (Secondary)' node. This node has an 'Availability Replicas' node and an 'Availability Databases' node. The 'AdventureWorks2014' database is selected under 'Availability Databases'. A context menu is open over the 'AdventureWorks2014' database, with the 'Resume Data Movement...' option highlighted in yellow. Other options in the menu include 'Remove Secondary Database...', 'Start PowerShell', and 'Reports'.

Anda harus mengulang langkah – langkah diatas untuk semua replica di semua secondary database. Setelah semua proses tersebut selesai maka data dashboard akan menampilkan data berikut:

| Availability replica: | | | | |
|-----------------------|-----------|---------------|-----------------------|---------------------|
| Name | Role | Failover Mode | Synchronization State | Availability Mode |
| ⚠ KOMODOSQL1\SQL2014 | Second... | Automatic | Synchronizing | Synchronous commit |
| ⚠ KOMODOSQL2\SQL2014 | Second... | Automatic | Synchronizing | Synchronous commit |
| ✓ KOMODOSQL3\SQL2014 | Primary | Manual | Synchronized | Asynchronous commit |

| Group by ▾ | | | | |
|----------------------|--------------------|-----------------------|-------------------|-----------|
| Name | Replica | Synchronization State | Failover Readi... | Suspended |
| KOMODOSQL1\SQL2014 | | | | |
| ⚠ AdventureWorks2014 | KOMODOSQL1\SQL2014 | Synchronizing | Data Loss | Resumed |
| ⚠ SalesDB | KOMODOSQL1\SQL2014 | Synchronizing | Data Loss | Resumed |
| KOMODOSQL2\SQL2014 | | | | |
| ⚠ AdventureWorks2014 | KOMODOSQL2\SQL2014 | Synchronizing | Data Loss | Resumed |
| ⚠ SalesDB | KOMODOSQL2\SQL2014 | Synchronizing | Data Loss | Resumed |
| KOMODOSQL3\SQL2014 | | | | |
| ✓ AdventureWorks2014 | KOMODOSQL3\SQL2014 | Synchronized | No Data Loss | Resumed |
| ✓ SalesDB | KOMODOSQL3\SQL2014 | Synchronized | No Data Loss | Resumed |

Replica di semua database berada pada kondisi *Resumed* dan tidak lagi *Suspended*.

- Status sinkronisasi di semua secondary berada pada kondisi *Synchronizing* karena KOMODOSQL3SQL2014 yang merupakan primary replica menggunakan mode *Asynchronous commit*.
- Dengan kondisi tersebut maka tetap terjadi resiko kehilangan data di KOMODOSQL1SQL2014 dan KOMODOSQL2SQL2014 walaupun kedua node tersebut menggunakan mode *Synchronous commit*.

8) Untuk menghindari terjadinya kehilangan data di secondary, ubah KOMODOSQL3SQL2014 ke mode *Synchronous commit*. Teknik ini dapat dilakukan selama kondisi jaringan antara primary dan secondary cukup baik untuk menerima beban *Synchronous commit*.

| Availability Replicas | | |
|-----------------------|-----------|---------------------|
| Server Instance | Role | Availability Mode |
| KOMODOSQL1\SQL2014 | Secondary | Synchronous commit |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit |
| KOMODOSQL3\SQL2014 | Primary | Asynchronous commit |
| | | Asynchronous commit |
| | | Synchronous commit |

9) Setelah proses sinkronisasi tercapai maka semua replica berada pada mode *Synchronous commit* dengan status *synchronized* sebagai berikut:

| Name | Replica | Synchronization State | Failover Readi... | Suspended |
|---------------------------|--------------------|-----------------------|-------------------|-----------|
| KOMODOSQL1\SQL2014 | | | | |
| ✓ AdventureWorks2014 | KOMODOSQL1\SQL2014 | Synchronized | No Data Loss | Resumed |
| ✓ SalesDB | KOMODOSQL1\SQL2014 | Synchronized | No Data Loss | Resumed |
| KOMODOSQL2\SQL2014 | | | | |
| ✓ AdventureWorks2014 | KOMODOSQL2\SQL2014 | Synchronized | No Data Loss | Resumed |
| ✓ SalesDB | KOMODOSQL2\SQL2014 | Synchronized | No Data Loss | Resumed |
| KOMODOSQL3\SQL2014 | | | | |
| ✓ AdventureWorks2014 | KOMODOSQL3\SQL2014 | Synchronized | No Data Loss | Resumed |
| ✓ SalesDB | KOMODOSQL3\SQL2014 | Synchronized | No Data Loss | Resumed |

10) Langkah berikutnya yang dapat dilakukan adalah mengembalikan KOMODOSQL1SQL2014 sebagai primary dengan metode planned manual failover. Tindakan ini dapat dilakukan apabila server tersebut memang berada pada kondisi baik dan dapat digunakan kembali sebagai primary replica.

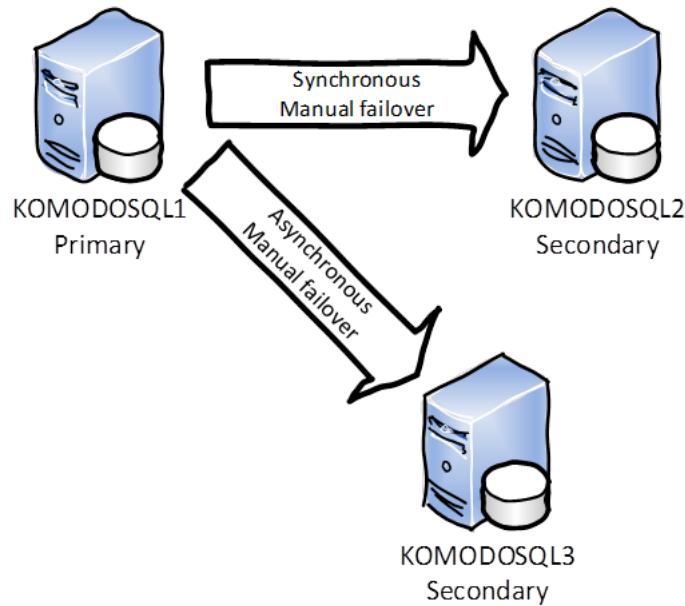
Forced Manual Failover untuk Mengatasi Kondisi Resolving

Status *Resolving* pada replica terjadi karena tidak terdapat primary yang menjadi acuan dalam sinkronisasi data. Replica dengan status tersebut tidak dapat menentukan peran masing masing, apakah sebagai primary maupun secondary.

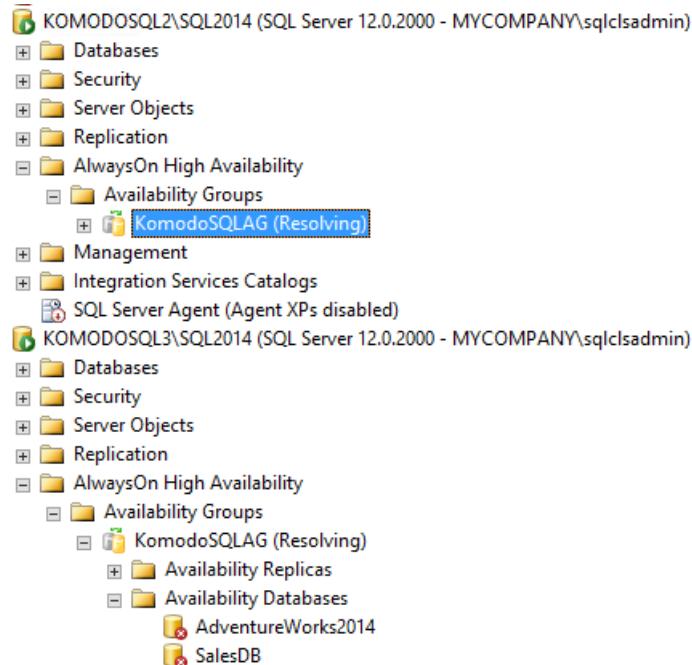
Kondisi tersebut umumnya terjadi karena primary bermasalah dan tidak ada secondary yang dapat menggantikannya sebagai primary. Akibatnya konfigurasi availability groups kehilangan acuan yang digunakan sebagai sumber sinkronisasi data. Kejadian ini dimungkinakan karena hal-hal berikut:

- Automatic failover gagal dilakukan karena ternyata secondary yang menjadi target juga mengalami masalah
- Tidak terdapat pasangan automatic failover dalam konfigurasi availability groups, sehingga ketika primary mengalami masalah maka tidak terdapat secondary yang dapat menggantikannya secara otomatis

Sebagai contoh, perhatikan diagram berikut yang menggambarkan konfigurasi tanpa pasangan automatic failover.



Dalam diagram tersebut tidak terdapat pasangan automatic failover. Jika primary mengalami masalah maka secondary replica kehilangan acuan untuk sumber sinkronisasi data. Akibatnya semua secondary replica berada pada kondisi *Resolving* dan tidak dapat diakses dari klien sebagaimana berikut:



Akses database menggunakan listener juga tidak dapat dilakukan karena tidak terdapat primary maupun readable secondary yang dapat diakses.



Sesuai dengan diagram diatas, untuk mengatasi masalah tersebut terdapat 2 pilihan sebagai berikut:

- Memperbaiki server primary sehingga dapat diaktifkan dan kembali menjadi primary replica
- Melakukan manual failover ke salah satu secondary sehingga terdapat primary replica baru di dalam availability groups. Pilihan failover dapat dilakukan terhadap KOMODOSQL2SQL2014 maupun KOMODOSQL3SQL2014.
 - Failover ke KOMODOSQL2SQL2014 dapat mencegah kehilangan data karena replica tersebut menggunakan mode Synchronous
 - Failover ke KOMODOSQL3SQL2014 mengandung resiko kehilangan data karena berjalan dengan mode Asynchronous sehingga harus dilakukan dengan forced manual failover

Jika server primary tidak dapat diperbaiki dalam waktu singkat maka manual failover harus dilakukan dengan resiko kehilangan data.

Kondisi *Resolving* dan kehilangan primary replica juga dapat terjadi walaupun pasangan automatic failover telah dikonfigurasi. Masalah tersebut terjadi jika secondary replica yang menjadi pasangan mengalami masalah, kemudian disusul terjadinya masalah di primary. Sebagai contoh, perhatikan tabel berikut:

| | | | |
|-------------------|--------------------|---------------------|-------------------|
| Server Instance | Availability mode | Failover mode | Role |
| KOMODOSQL1SQL2014 | Automatic | | |
| | Synchronous commit | Primary | KOMODOSQL2SQL2014 |
| | | Automatic | Automatic |
| | | Synchronous commit | |
| | Secondary | | KOMODOSQL3SQL2014 |
| | | Manual | Manual |
| | | Asynchronous commit | |
| | | | Secondary |

Berdasarkan konfigurasi pada tabel tersebut maka kondisi *Resolving* akan terjadi pada skenario berikut:

- 1) KOMODOSQL2SQL2014 mengalami masalah sehingga tidak dapat diakses. Kondisi ini tidak mempengaruhi primary karena secondary replica yang bermasalah tidak memicu proses failover.
- 2) Sebagai impact masalah tersebut, KOMODOSQL1SQL2014 kehilangan pasangan automatic failover. Proses sinkronisasi dengan KOMODOSQL3SQL2014 tetap berjalan normal dengan mode Asynchronous.
- 3) KOMODOSQL1SQL2014 mengalami masalah sehingga availability groups kehilangan primary replica. Proses failover otomatis ke KOMODOSQL3SQL2014 tidak terjadi karena node tersebut menggunakan mode Asynchronous.

- 4) KOMODOSQL3SQL2014 tidak memiliki acuan yang dapat dijadikan sebagai primary sehingga berstatus *Resolving* dan tidak dapat diakses.

Untuk mengatasi hal tersebut, dapat dilakukan forced manual failover terhadap KOMODOSQL3SQL2014 sehingga node tersebut berperan sebagai primary. Dengan demikian database tersedia untuk diakses sambil menunggu proses perbaikan di 2 server yang mengalami masalah.

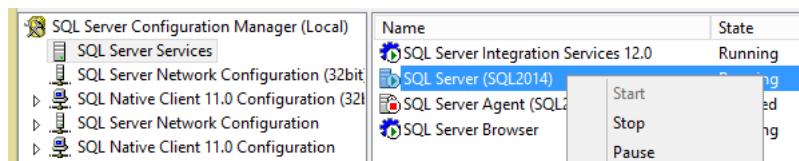
Latihan 6.6. Mengatasi Kondisi Resolving dengan Forced Manual Failover

Latihan ini menitikberatkan pada simulasi kondisi Resolving yang terjadi karena 2 mesin (primary dan secondary) bermasalah. Sebelum melakukan latihan ini, pastikan konfigurasi replica sesuai dengan kondisi berikut:

| Availability Replicas | | | | |
|-----------------------|-----------|---------------------|---------------|-----------|
| Server Instance | Role | Availability Mode | Failover Mode | |
| KOMODOSQL1\SQL2014 | Primary | Synchronous commit | Automatic | Automatic |
| KOMODOSQL2\SQL2014 | Secondary | Synchronous commit | Automatic | Automatic |
| KOMODOSQL3\SQL2014 | Secondary | Asynchronous commit | Manual | Manual |

1) Langkah pertama adalah mematikan layanan SQL Server di KOMODOSQL2SQL2014. Login ke KOMODOSQL2 menggunakan account **mycompany\sqlclsadmin**.

2) Buka SQL Server Configuration Manager dan matikan layanan KOMODOSQL2SQL2014 di node tersebut.



Tindakan tersebut mengakibatkan secondary replica menjadi non aktif. Replica tersebut merupakan pasangan automatic failover dengan KOMODOSQL1SQL2014.

3) Amati hasil proses tersebut dari dashboard availability group. Terlihat bahwa primary replica tetap berada di KOMODOSQL1 karena masalah yang terjadi dengan primary tidak memicu proses failover.

 KomodoSQLAG: hosted by KOMODOSQL1\SQL2014 (Replica role: Primary)

Availability group state:  [Critical --- Critical \(1\), Warnings \(3\)](#)

Primary instance: KOMODOSQL1\SQL2014

Failover mode: Automatic

Cluster state: WIN2012-SQLAG (Normal Quorum)

Availability replica:

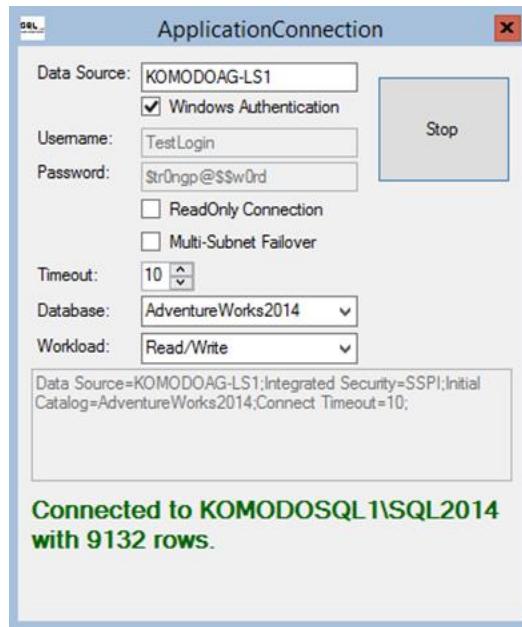
| Name | Role | Failover Mode | Synchronization State | Availability Mode |
|--|-----------|---------------|-----------------------|---------------------|
|  KOMODOSQL2\SQL2014 | Secondary | Automatic | Not Synchronizing | Synchronous commit |
|  KOMODOSQL1\SQL2014 | Primary | Automatic | Synchronized | Synchronous commit |
|  KOMODOAG\SQL2014 | Secondary | Manual | Synchronizing | Asynchronous commit |

- 4) Dengan kondisi yang terjadi di KOMODOSQL2 maka availability group telah kehilangan kemampuan automatic failover karena secondary replica untuk mendukung proses tersebut tidak aktif. Klik tautan Critical.... untuk membaca penjelasan lebih lanjut mengenai masalah yang terjadi.

Select a policy to see the detailed results in the table below

| Detected Issue | |
|---|---|
|  | Availability group is not ready for automatic failover. |
|  | Some availability replicas are disconnected. |
|  | Some availability replicas are not synchronizing data. |
|  | Some synchronous replicas are not synchronized. |

- 5) Dalam kondisi tersebut aplikasi klien tetap dapat mengakses database sebagaimana biasa karena primary replica masih tetap tersedia. Jalankan availability groups Demonstrator_ _dan hubungan ke listener KOMODOAG-LS1 untuk menguji hal tersebut.



- 6) Selanjutnya Anda harus melakukan simulasi dengan mematikan layanan database di KOMODOSQL1 yang merupakan primary replica. Login ke KOMODOSQL1 dan matikan SQL Server instance KOMODOSQL1SQL2014 dengan menggunakan SQL Server Configuration Manager.
- 7) Setelah availability groups kehilangan secondary dan primary replica, maka hanya tersisa KOMODOSQL3SQL2014 yang menggunakan mode Asynchronous commit. Automatic failover tidak dapat terjadi dan replica tersebut berada pada kondisi *Resolving*.

Dalam kondisi tersebut database tidak dapat dibaca dari klien baik menggunakan listener maupun akses langsung ke SQL Server instance.

- 8) Langkah berikutnya adalah menjadikan KOMODOSQL3SQL2014 sebagai primary dengan prosedur forced manual failover, dengan resiko kehilangan data.

Kondisi ini mensimulasikan tindakan disaster recovery dimana semua server primary dan secondary

di data center utama tidak dapat diakses dan hanya tersisa server secondary di lokasi terpidah.

9) Jalankan T-SQL tersebut untuk menjadikan database di KOMODOSQL3SQL2014 sebagai primary replica.

```

1  \--- YOU MUST EXECUTE THE FOLLOWING SCRIPT IN SQLCMD MODE.
2
3  :Connect KOMODOSQL3\SQL2014
4  ALTER AVAILABILITY GROUP [KomodoSQLAG] FORCE_FAILOVER_ALLOW_DATA_LOSS;

```

10) Setelah eksukusi perintah tersebut berhasil maka KOMODOSQL3SQL2014 telah berubah peran menjadi primary dan siap melayani aplikasi.

| Availability replica: | | | | | |
|-----------------------|-------------------|-----------|---------------|-----------------------|---------------------|
| | Name | Role | Failover Mode | Synchronization State | Availability Mode |
| ✗ | KOMODOSQL1\SQL... | Secondary | Automatic | Not Synchronizing | Synchronous commit |
| ✗ | KOMODOSQL2\SQL... | Secondary | Automatic | Not Synchronizing | Synchronous commit |
| ✓ | KOMODOSQL3\SQL... | Primary | Manual | Synchronized | Asynchronous commit |

11) Anda dapat menggunakan Availability Groups Demonstrator atau SSMS untuk memastikan bahwa database sudah dapat diakses dari klien. Perlu diingat bahwa konfigurasi availability group masih belum berada pada kondisi sempurna karena saat ini hanya terdapat satu node yang aktif.

Latihan Lanjutan

Kondisi availability groups masih belum sempurna karena 2 node belum berfungsi normal. Anda dapat melanjutkan latihan lanjutan dengan melakukan langkah- langkah berikut.

- 1) Aktifkan kembali SQL Server di KOMODOSQL1 dan KOMODOSQL2.
- 2) Database akan berada pada kondisi *Suspended*. Lakukan prosedur **Resume Data Movement** sehingga semua replica berada pada kondisi *Resumed*.
- 3) Kembalikan posisi primary ke KOMODOSQL1SQL2014 dengan melakukan manual failover menggunakan failover wizard atau T-SQL

[CA1] Stress that no data lost happens because of sync mode

[CA2] If problem, what to check >>> see different chapter on FAQ and common problem

Bab 7. Konfigurasi Quorum Tingkat Lanjut

Bab ini tidak tersedia dalam sampel PDF

Konfigurasi Quorum Tingkat Lanjut

Peranan Quorum dalam Cluster

Node majority

Node and disk majority

Node and file share majority

Dynamic Quorum di Windows Server 2012

Dynamic Witness di Windows 2012 R2

Bab 8. Optimasi dan Pemantauan Kinerja

Bab ini tidak tersedia dalam sampel PDF

Proses Sinkronisasi Data Antar Replica

Pemantauan menggunakan dashboard availability group

Pemantauan menggunakan SQL DMV

Otomasi pemantauan kinerja

Optimasi Infrastruktur

Penggunaan Statistik di Active Secondary

Bab 9. Integrasi dengan Microsoft Azure

Bab ini tidak tersedia dalam sampel PDF

Skenario Availability Groups dengan Microsoft Azure

Skenario Hybrid IT dalam Availability Group

Konfigurasi Site to Site VPN

Menambah Azure Replica ke Availability Group

Konfigurasi “Azure Only” Availability Group

Bab 10. Perawatan dan Operasional

Bab ini tidak tersedia dalam sampel PDF

Patching dan upgrade sistem operasi

Patching dan upgrade SQL Server

Pengujian rutin HA dan DR

Penggantian dan upgrade hardware

Perubahan Alamat IP

Perubahan akun dan password