

Security Pada SQL Server

Agus Pamujiono

ag_kanedrew@yahoo.com

Lisensi Dokumen:

Copyright © 2004 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pada pembuatan database client-server tidak boleh diabaikan masalah security. Security database di SQL Server cukup handal untuk database menengah ke atas. Security system ini berhubungan erat dengan otoritas sebuah client dalam berkomunikasi dengan database.

Sebuah user di SQL server dapat kita buat langsung dari SQL Server Enterprise Manager atau mapping dari user di Windows NT/2000. User default Administrator di SQL Server adalah “sa” dengan password <blank>. User ID tidak bisa kita ganti, akan tetapi kita bisa membuat User ID yang mempunyai Otoritas yang sama dengan “sa” sebagai Administrator.

Password untuk user “sa” dapat kita ganti sesuka kita meskipun database sudah terbantu. Hal ini tidak akan memberi efek apa-apa pada database. Akan tetapi bisa menyebabkan error koneksi pada aplikasi yang sudah kita bangun sebelumnya. Error koneksi pada aplikasi dapat kita atasi dengan cara membuka source programnya dan mengganti password. Error koneksi tidak akan terjadi di aplikasi jika aplikasi kita bangun dengan koneksi langsung ke database SQL Servernya.

TEKNIK SECURITY

Ada banyak teknik security pada aplikasi. Security pada aplikasi dapat kita ciptakan sendiri dengan cara membuat sebuah tabel user beserta otoritasnya untuk sebuah aplikasi tertentu. Akan tetapi hal ini kurang aman bilamana tabel yang kita buat sampai diketahui oleh orang yang tidak berkepentingan.

Teknik kedua adalah membuat mapping untuk user sebuah Windows Server untuk akses database. Meskipun hal ini mudah untuk dilakukan akan tetapi kita akan mengalami kendala yang sama pada teknik kedua. Lebih dari itu, akan sangat rawan apabila user kita ceroboh saat meninggalkan komputernya dalam keadaan On dan tanpa Re-Login.

Teknik lain adalah dengan cara mengintegrasikan User di database dengan aplikasi. Hal ini sangat terjamin keamanannya, karena encrypt password dan Otoritasnya sudah di tangani oleh SQL Server itu sendiri. Kerahasiaan password murni ada di tangan user dan administrator database. Tinggal pandai-pandainya kita dalam membangun aplikasi untuk menangani error yang di timbulkan oleh SQL Server. Error hande yang perlu kita tangani diantaranya adalah :

1. Login Error.
2. Autoritas atas permission komponen database (table, view, stored procedure, trigger dan function (Untuk SQL Server 2000))

Untuk lebih mengintegritaskan interface user permission dengan aplikasi kita, sebenarnya dapat kita tangani dengan memanfaatkan teknik DMO atau NameSpace pada SQL Server (tidak dibahas pada tulisan ini).

Selain teknik pembuatan user, yang perlu kita perhatikan dalam hal security adalah teknik koneksi. Teknik koneksi juga banyak macamnya.

Teknik koneksi pertama adalah dengan membuat koneksi untuk sebuah user yang login sampai dia log off, dan membiarkan user login dengan user id dan password yang sama di komputer lain. Teknik ini sangat rawan sekali.

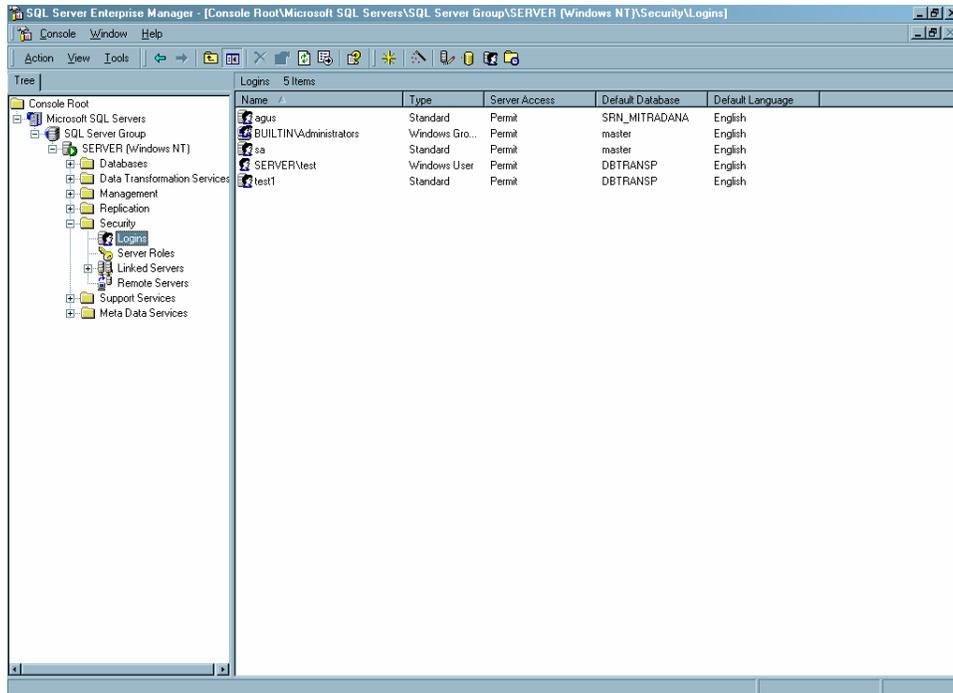
Teknik kedua adalah dengan membuat koneksi untuk sebuah user yang login sampai dia log off dan tidak mengijikan user login di komputer lain dengan user id dan password yang sama. Teknik ini lebih bagus dari teknik pertama.

Teknik ketiga adalah dengan membuat koneksi untuk sebuah user yang login sampai batas waktu tertentu jika ia tidak beraktifitas di aplikasi tersebut dan atau sampai ia log off, juga tidak mengijikan login di komputer lain dengan user id dan password yang sama dengan cara memutus salah satu koneksinya. Untuk menentukan koneksi mana yang diputus, user ditawarkan pilihan, ia akan memakai koneksi yang baru dengan memutus yang lama atau dia membatalkan koneksi yang baru dengan membiarkan aktif koneksi yang lamanya. Teknik ini sangat aman, meski agak sulit untuk diterapkan.

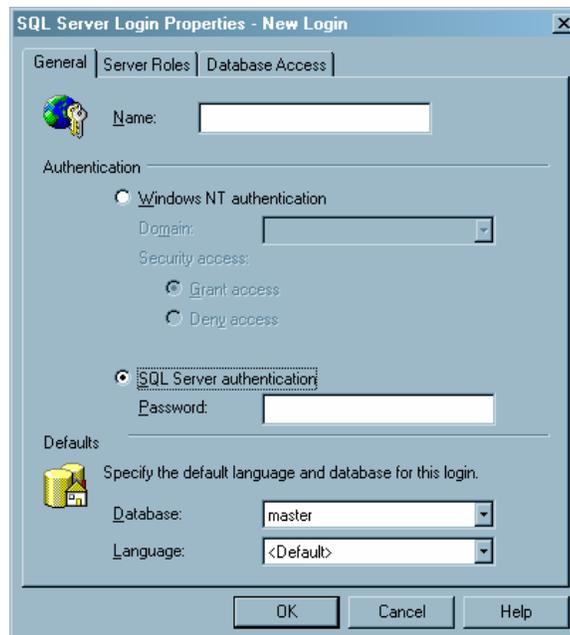
CARA MEMBUAT USER DI SQL SERVER

Kita dapat mendefinisikan user-user lain yang akan mengakses database tersebut di SQL Server Enterprise Manager beserta otoritasnya, seperti Select, Insert, Update, Delete dan Execute. Cara membuat user baru :

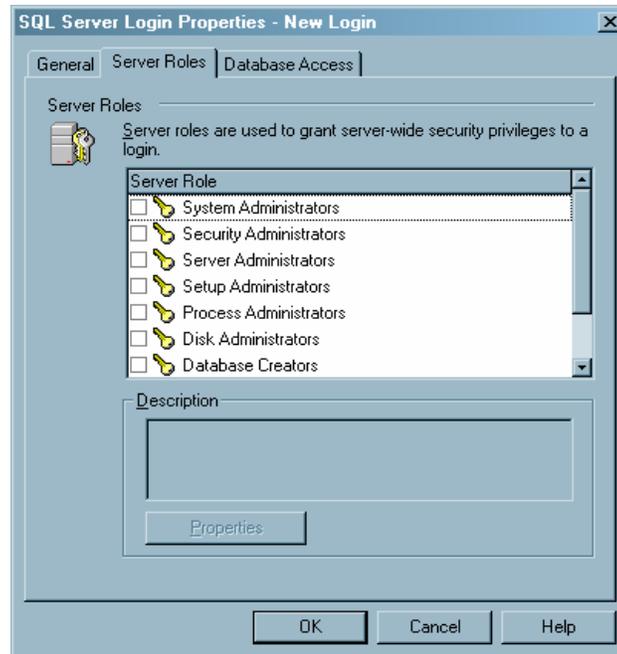
1. Buka SQL Server Enterprise Manager
2. Masuk server yang terkoneksi.
3. Pilih Security – Logins.



4. Jika memilih Windows NT Authentication, Isikan Domain tempat user login dan pilih type Access (Grant atau Deny). Jika memilih SQL Server Authentication Isikan User ID pada Name, dan password.
Pilih database, dan pilih bahasa yang digunakan atau pilih <Default>.

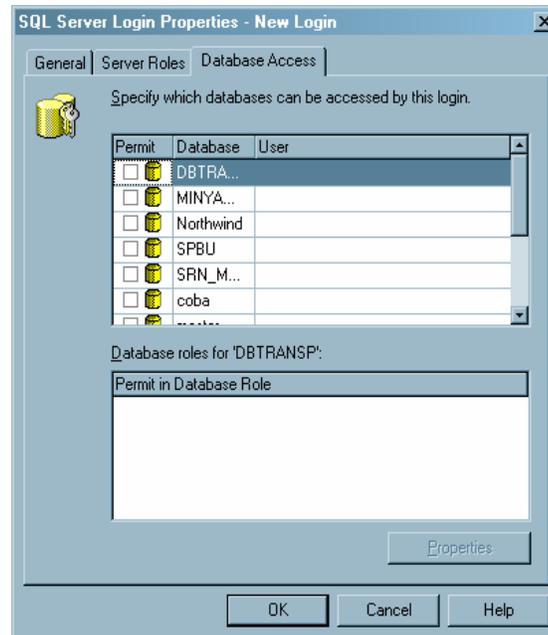


5. Kemudian klik tab Server Roles untuk memilih group user dengan segala permissionnya.



Roles dapat anda buat untuk kelompok user, supervisor dll, selain yang ada di default dari SQL Server tersebut. Untuk membuat Roles baru, lihat pada bagian Roles.

6. Lalu klik tab Database Access untuk memilih database yang bisa diakses oleh user tersebut.



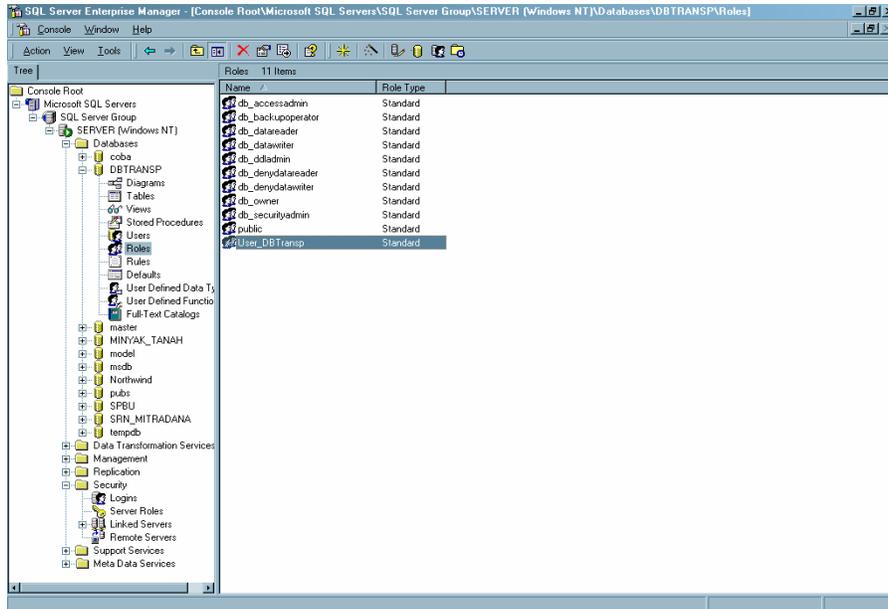
Setelah user baru terbentuk, anda bisa mengeditnya dengan cara klik kanan user tersebut, lalu pilih Properties. Anda bisa mencoba login dengan user tersebut pada Query Analyzer.

CARA MEMBUAT ROLES DI SQL SERVER

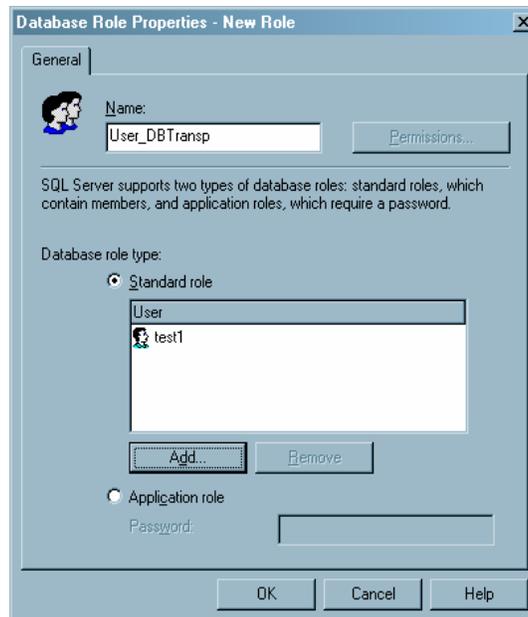
Roles adalah sebuah group user, yang artinya mengelompokkan beberapa user SQL Server yang mempunyai hak akses sama. Hal ini memberi kemudahan saat beberapa user dalam kelompok tersebut mengalami perubahan hak akses, sehingga kita tinggal merubah hak akses roles-nya maka user-user yang ada dalam roles tersebut akan ikut berubah.

STANDARD ROLES

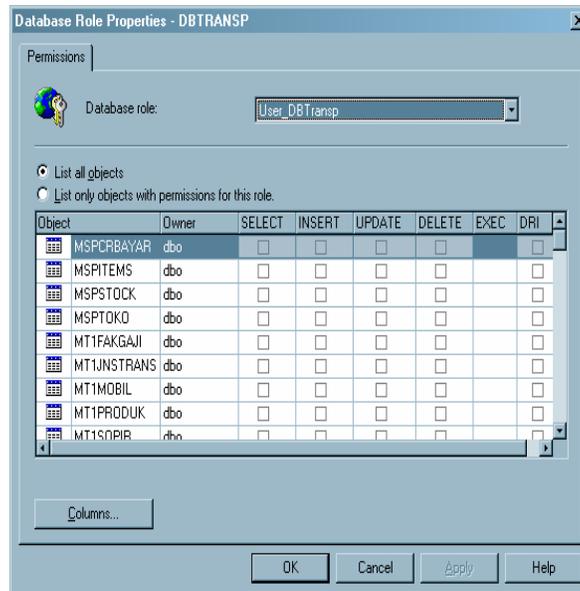
Untuk membuat sebuah roles (user group), masuk ke database tertentu lalu pilih roles. Kemudian klik kanan pada window sebelah kanan, pilih New Database Role...



Isikan nama Role yang akan anda buat, lalu pilih anggotanya dengan menekan tombol Add... dan pilih user-nya.



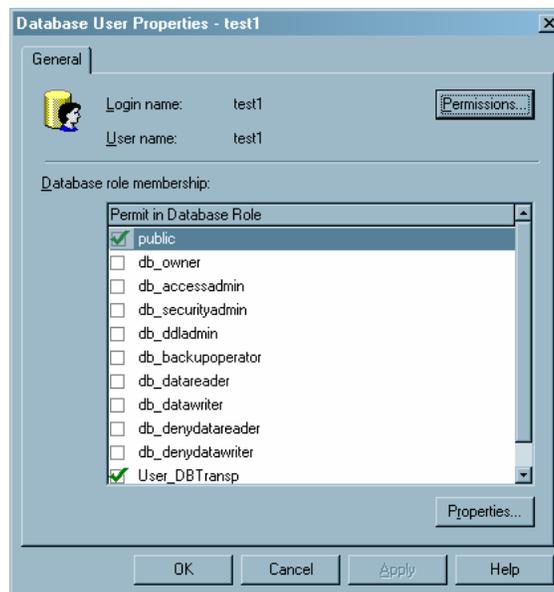
Setelah sebuah Role terbentuk barulah anda bisa menentukan akses permission pada role tersebut dengan meng-klik kanan role yang anda buat, lalu klik tombol Permissions.



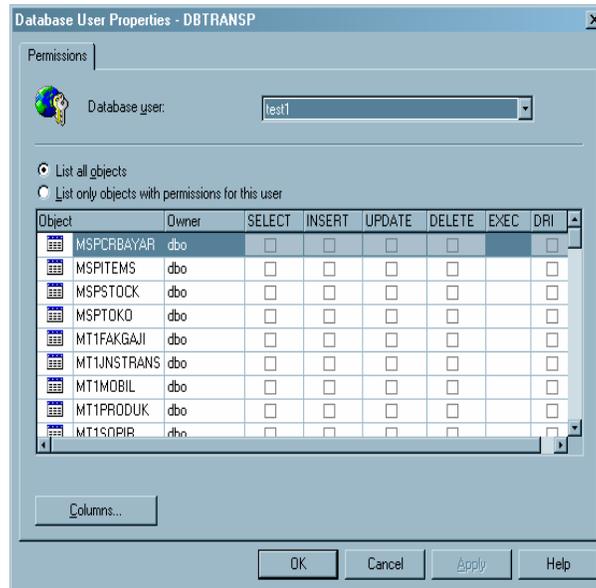
Dari sini anda dapat memilih tabel, view atau stored procedure yang dapat diakses oleh kelompok user tersebut dengan Select, Insert, Update, Delete, Execute atau DRI (declarative referential integrity).

CARA MEMBUAT PERMISSIONS

Untuk membuat permission pada sebuah user atau Roles tertentu anda bisa melakukannya dengan cara masuk ke database dimana user tersebut memiliki hak akses, lalu pilih users, lalu properties. Atau pilih Roles lalu klik properties. Kemudian klik tombol permissions.



Dari sini anda dapat menentukan tabel, view atau stored procedure yang dapat diakses oleh user tersebut dengan Select, Insert, Update, Delete, Execute atau DRI (declarative referential integrity).



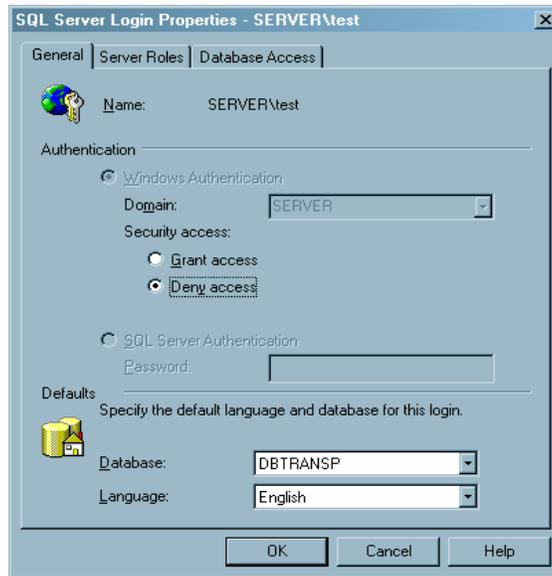
Anda dapat mengkombinasikan permission pada User maupun pada Role. Kedudukan kedua komponen database tersebut saling melengkapi. Akan tetapi apabila permission pada user dan role bertentangan, maka permission yang tidak mengijinkan akan lebih dominan. Misalnya seorang user U menjadi anggota role R dengan akses tabel X. Padahal permission pada user U tersebut tidak mengijinkan untuk mengakses tabel A. Maka pada implementasinya, user U tersebut tidak akan dapat mengakses tabel A.

Akan tetapi jika ada membuat permission pada user U dapat mengakses tabel A,B,C dan membuat permission pada role R dapat mengakses tabel X,Y,X, dan user U tersebut menjadi anggota dari role R, maka pada implementasinya user tersebut dapat mengakses tabel A,B,C,X,Y,Z.

CARA MENON-AKTIVE-KAN USER

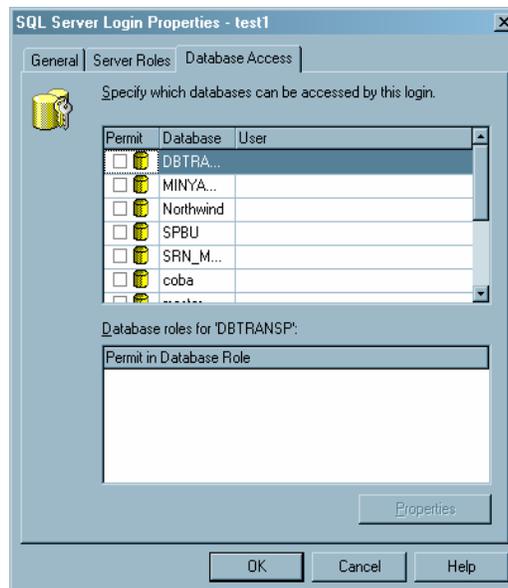
Cara menon-aktifkan user ada 2 cara, tergantung type dari user tersebut, apakah user tersebut dibuat dari user Windows NT atau dari SQL Server.

Apabila user tersebut dibuat dari user Windows NT, anda dapat melakukannya dengan cara berikut :
Masuklah ke sebuah server dari Enterprise Manager, lalu pilih Security, kemudian klik login. Pada window sebelah kanan pilih user yang akan di non-aktifkan dengan cara klik kanan, properties. Kemudian pilih option Deny access.



Dengan demikian user tersebut sudah tidak bisa lagi mengakses database yang semula anda definisikan untuknya. Akan tetapi jika setting tersebut anda kembalikan ke Grant access, maka segala setting, baik untuk database, role dan setting akses pada user tersebut akan kembali seperti semula.

Sedangkan untuk user yang dibuat dari SQL Server, anda tidak bisa membuat Deny access. Yang bisa anda lakukan bagi user yang dibuat di SQL Server untuk menon-aktifkan adalah dengan menghilangkan permission aksesnya untuk semua database. Dengan demikian, user tersebut masih ada akan tetapi tidak mempunyai akses ke database manapun. Saat itu anda lakukan untuk sebuah user, maka semua permission yang telah anda set untuk user tersebut akan hilang. Jika anda ingin mengaktifkan lagi, maka anda harus mengeset user tersebut pada database yang anda inginkan, termasuk juga keanggotaannya pada sebuah role dan permission akses yang ada pada user tersebut.



CARA MENGHAPUS USER

Untuk menghapus user account, anda dapat melakukannya dengan cara masuklah ke sebuah server dari Enterprise Manager, lalu pilih Security, kemudian klik login. Pada window sebelah kanan pilih user yang akan di hapus dengan cara klik kanan, lalu pilih Delete.

WINDOWS NT AUTHENTICATION Vs SQL SERVER AUTHENTICATION

Windows NT Authentication mempunyai manfaat lebih dibandingkan dengan menggunakan SQL Server Authentication.

Pertama, anda tidak perlu membuat user dan permission terpisah di SQL Server untuk mengakses database.

Kedua, keamanan Windows NT menyediakan fitur lebih dibanding SQL Server Authentication. Fitur ini meliputi password expired, panjang minimum password, dan maksimum login saat terjadi kesalahan password.

Ketiga, SQL Server membaca informasi tentang para pemakai dan kelompok manakala pemakai menghubungkan. Perubahan apapun yang dibuat untuk seorang user di account Windows NT akan direlasikan ke user SQL Server.

Akan tetapi penggabungan dua metode di atas akan lebih mudah untuk pengaturannya jika dibanding dengan hanya menggunakan Windows NT Authentication. Metode campuran ini cocok untuk database yang tidak "mission-critical" atau tidak berisi data rahasia. Anda juga bisa menggunakan hanya SQL Server Authentication untuk sebuah database dengan menyerahkan segala fasilitas dan keamanannya pada SQL Server.