

# Pengamanan *Data Access Pages*

( Suatu contoh pengamanan informasi pada tingkat aplikasi )

Oleh : Budiyanto (23202119)

---

---

## Daftar Isi :

<i>Pendahuluan</i>	1
<i>Apa itu Data Access Page</i>	3
<i>Mengapa Data Access Page perlu diamankan</i>	4
<i>Keamanan Access 2000 dan Format Database</i>	5
<i>Menginstall MSDE dan Bekerja dengan MSDE Database</i>	6
<i>Pengamanan File Data Access Page</i>	6
<i>Kesimpulan</i>	20
<i>Referensi</i>	20

## Pendahuluan

Kamanan data pada jaman teknologi informasi ini dihadapkan pada masalah yang sangat rumit sehingga memerlukan perhatian yang serius bagi pihak yang ingin menerapkannya dalam kehidupan organisasi. Resiko serangan secara umum dapat dikategorikan menjadi tiga [1] yaitu :

1. Sumber daya fisik ( *Physical Resources* )

Adalah asset yang berbentuk fisik, termasuk *server, workstation, hub, switch* dan sebagainya.

2. Sumber daya intelektual ( *Intellectual Resources* )

Agak lebih sulit diidentifikasi dari pada sumber daya fisik, sebab umumnya hanya berbentuk format elektronik. termasuk di dalamnya yaitu *software, database, informasi keuangan* dan sebagainya.

3. Sumber daya waktu ( *Time Resources* )

Waktu adalah sumber daya organisasi yang penting , waktu bagaimanapun merupakan asset yang bernilai bagi organisasi.

Sementara itu siapa yang potensial melakukan serangan adalah [1] :

1. Karyawan,
2. Karyawan berkala ( tidak tetap),
3. Pesaing,
4. Seseorang dengan pandangan atau tujuan yang berbeda secara radikal dengan organisasi,
5. Seseorang yang mempunyai dendam dengan organisasi atau salah satu karyawan,
6. Seseorang yang ingin memperoleh ketenaran,

Grafinkel [2] mengemukakan bahwa keamanan komputer meliputi empat aspek yaitu *privacy, integrity, authentication* dan *availability* . Dalam kaitannya dengan *electronic commerce* masih dikaitkan dengan dua aspek lagi yaitu *access control* dan *non-repudiation*.

Menurut W. Stalling [3] ada beberapa kemungkinan serangan berkaitan dengan system informasi, yaitu :

1. *Interuption*, perangkat system menjadi rusak atau tidak tersedia,
2. *Interception* , pihak yang tidak berwenang mengakses asset atau informasi,
3. *Modification*, pihak yang tidak berwenang mengakses dan bahkan mengambil atau mengubah aset atau informasi,
4. *Fabrication*, pihak yang tidak berwenang menyisipkan objek palsu ke dalam system.

Dalam laporan ini lebih khusus akan dibahas masalah keamanan yang menurut Chris Brenton berhubungan dengan sumber daya intelektual. Menjelaskan bagaimana menggunakan *Microsoft Access 2000* untuk membuat *data access pages* bagi *database* yang aman. Laporan ini menjelaskan juga perbedaan antara *data access pages* dan objek access tradisional yang lain dan membahas *step-by-step* contoh yang memperlihatkan bagaimana membuat *data access pages* untuk *Microsoft Access (.mdb)* dan *Microsoft Data Engine (MSDE)* yang aman .

Setelah pembahasan ini, diharapkan anda dapat :

- Mengidentifikasi beberapa perbedaan antara keamanan *Microsoft Access (Microsoft Jet database engine)* dan *Microsoft SQL Server*.
- Membuat *data access page* untuk *database access* yang diamankan dengan *password*.
- Membuat *data access page* untuk *database access* dengan keamanan *user-level*.
- Membuat *data access page* untuk *SQL Server database* dengan menggunakan validasi *logon* untuk autentikasi .
- Membuat *data access page* untuk *MSDE database* dengan dan tanpa informasi koneksi yang tersimpan dalam *page*.

### **Apa itu Data Access Page?**

*Data access pages* adalah hal yang baru pada Access 2000. *Data access page* adalah tipe khusus *Web page* yang memungkinkan user melihat dan bekerja dengan data yang tersimpan di dalam *Access database (.mdb)*, *SQL Server database*, atau *MSDE database* dari *Microsoft Internet Explorer 5*. *Data access pages* menggunakan kode HTML , *HTML intrinsic controls*, dan *group* dari *ActiveX® controls* yang disebut *Microsoft Office Web Components* untuk menampilkan data dari *Web page*. Berbeda dengan *database object* yang lain, seperti *forms and reports*, *data access page* adalah bagian file (.htm) yang disimpan di luar suatu *Access database (.mdb)* atau *Access project (.adp)* di dalam *file sistem* atau pada *Web server*.

Dalam beberapa hal, *data access pages* menyediakan fitur yang sama guna mengakses *forms* dan *reports* dari suatu *Web page*. Seperti *Access forms*, anda dapat membuat *data access page* untuk *view*, *edit*, *add*, atau *delete records* di dalam suatu *record source* utama . Anda dapat juga memasukkan *Microsoft Office PivotTable®*, *Microsoft Office Chart*, dan *Microsoft Office Spreadsheet ActiveX controls* untuk menampilkan data seperti daftar *PivotTable*, *chart*, atau *spreadsheet*.

Sebagaimana dengan *report* pada *Access*, anda dapat memanfaatkan *sort*, *filter*, dan *group record* yang muncul menurut kriteria yang anda tentukan. *Data access pages* dapat juga juga menampilkan data dalam suatu format hirarki interaktif dengan memanfaatkan indikator tambahan . Dalam hal ini user diijinkan untuk mengubah suatu *view* yang

menampilkan ringkasan informasi umum, seperti daftar dari semua *range (region)* dan kombinasi *sales total*, untuk memperlihatkan (*list*) secara detail tentang *individual sales* di tiap *region*.

### **Mengapa perlu mengamankan *Database* atau *Data Access Page*?**

Ada beberapa alasan mengapa anda perlu mengamankan *database* atau *data access page* yaitu :

- Untuk Memproteksi informasi penting yang tersimpan dalam *database*. Contoh yang bagus misalnya informasi gaji yang tersimpan dalam tabel pegawai. *Management* dan *accounting* sering menggunakan informasi ini , meskipun umumnya hal ini disampaikan kepada umum(*published*) ke seluruh perusahaan.
- Untuk memproteksi *data access page (.htm )* dan objek *database access* dari usaha perubahan. Perubahan yang tidak berhak pada *database* atau *data access page (.htm)* dapat menyebabkan kerusakan atau tidak berfungsi.
- Untuk memproteksi kekayaan intelektual dari desain *database* dan kode dalam aplikasi.

Ketika anda men-set keamanan untuk *data access pages* dan *network databases* dimana terletak sumber data anda , langkah pertama adalah berpikir tentang apakah tipe keamanan yang anda perlukan. Anda perlu mempertimbangkan jawaban dari pertanyaan di bawah ini:

- Kelompok mana yang perlu mengakses *database*?
- Informasi penting yang harus dibatasi dari kelompok tertentu?
- Apakah saya aman jika *user* mengubah desain *data access pages* saya?
- Apakah saya perlu ingin memprotek hak intelektual atas *data access page* saya?
- Bagaimanakah *set up network* dan *file system* menangani kerahasiaan file?

Jawaban atas dua pertanyaan pertama membantu anda mengorganisir *user groups* dan men-set ijin akses *database*, sementara pertanyaan lainnya menentukan bagaimana mengkonfigurasi *file system* dan keamanan *Web server* untuk file *data access page (.htm )*.

## Keamanan Access 2000 dan format database

Access 2000 bekerja dengan tiga tipe *database*:

- *Database Access* traditional,
- *MSDE databases*,
- *database* yang tersimpan pada mesin yang menjalankan *SQL Server 6.5* atau *Access databases* yang lebih baru, tersimpan dalam file *.mdb* yang dapat dibagi (*shared* ) dengan *user* lain dengan meletakkannya dalam folder yang terbagi (*shared network folder*).

*MSDE* adalah suatu teknologi baru yang dapat diinstall pada komputer yang menjalankan *Microsoft Windows® 95, Microsoft Windows 98, or Microsoft Windows 2000 Professional* yang menyediakan kemampuan penyimpanan lokal atau terbagi(*shared*).

Ketika anda mengakses *SQL Server* dan *MSDE databases*, informasi koneksi dan objek access khusus (seperti *forms, reports, modules, and links to data access pages*) disimpan dalam suatu file *Access project (.adp)* yang berperan sebagai *interface* ke database.

Karena *Microsoft Jet, SQL Server, dan MSDE database engines* mendukung fitur keamanan yang berbeda, anda akan menjamin bahwa fitur keamanan terbuka dalam mengakses *user interface* adalah berbeda tergantung pada apakah anda sedang bekerja dalam *database access (.mdb)* atau *project file Access (.adp)*.

Sekumpulan fitur keamanan *level user* disediakan oleh *Access* dan *Microsoft Jet database engine* untuk *database access (.mdb)* adalah yang terbaik dalam pasar manajemen database. Ketika memutuskan apakah menggunakan *database access* atau *Access project* yang dihubungkan ke *MSDE* atau *SQL Server database*, anda harus memutuskan persyaratan *multiuser*, ukuran *database*, dan konfigurasi *network*.

Keamanan *level user (User-level security)* untuk *database access* menuntut anda untuk menentukan file informasi kelompok kerja yang terpisah (*separate workgroup information file* ) untuk mengisi *account user* atau *group* keamanan yang asli, sedangkan keamanan *MSDE* dan *SQL Server* dapat diintegrasikan untuk menggunakan *account security Windows 2000*. Jika aplikasi anda menuntut fitur yang tangguh (*robust* ) yang secara kuat diintegrasikan dengan *Server network Windows 2000*, anda harus mempertimbangkan penggunaan *MSDE* atau *SQL Server*.

### **Menginstall dan bekerja dengan *database MSDE***

Untuk menginstall *MSDE*, jalankan *Setupsql.exe*, yang ada di *sub folder* *\SQL\x86\Setup* di *Disc-1* pada *Microsoft Access 2000*, *Microsoft Office 2000 Professional*, *Microsoft Office 2000 Premium*, dan *Microsoft Office 2000 Developer versions*.

Setelah meginstall *MSDE*, anda dapat membuat *database MSDE* baru di Access dengan memilih menu *New* pada menu *File*, dan kemudian *double-click Project (New Database)*. Anda dapat menghubungkan ke *MSDE* atau *SQL Server database* dengan *double-click Project (Existing Database)*, atau dengan menggunakan *Microsoft Access Upsizing Wizard* untuk meng-convert *database access (.mdb)* yang telah ada ke *MSDE* atau *database SQL Server*.

Lebih jelasnya ikuti langkah berikut:

Buka *.mdb* file.

- Arahkan ke *Database Utilities*
- *Tools*,
- *Upsizing Wizard*
- Ikuti petunjuk *wizard* selanjutnya .

### **Mengamankan file *Data Access Page***

Karena file *data access page* disimpan sebagai file individual *.htm* yang terlepas dari file aplikasi *database*, anda tidak dapat menentukan wewenang akses *database* (seperti *read* dan *write*). Bagaimana anda mengontrol pembacaan dan penulisan *access* ke file *data access page* tergantung pada lokasinya. Jika file *data access page* diakses dari *folder network* yang terbagi (*shared*), anda harus menggunakan fitur keamanan dari *file system*. Sebagai contoh, di bawah semua versi Windows dapat mengontrol akses ke *shared folders*; sistem operasi network yang lain, seperti *Novell NetWare*, juga menyediakan fitur untuk mengontrol akses ke file dan *folder*.

Jika file *data access page* ditempatkan pada *Web server*, anda harus menggunakan fitur keamanan *Web server*. Sebagai contoh, *Microsoft Internet Information Server (IIS) 4.0*

atau yang terbaru dan *Web-management tools* dari *Microsoft FrontPage* menyediakan dukungan untuk pengontrolan akses ke file.

### **Keamanan *Microsoft Jet and Data Access Page***

*Access* dan *Microsoft Jet database engine* menyediakan keamanan *share-level* dan keamanan *user-level* untuk mengakses *database (.mdb files)*. Dengan keamanan *share-level*, anda menentukan *password* ke *database*, dan siapapun yang mengetahui *password* dapat membuka *database*. Keamanan *User-level* memerlukan lebih banyak pekerjaan tetapi menawarkan lebih banyak fasilitas. Anda dapat meng- *create user and group* dan kemudian memberi ijin kepadanya ke objek *database* (seperti *table and query*).

### **Keamanan *Share-level* dan *password database***

Penambahan *password database* adalah cara mudah untuk mencegah pengaksesan *database* yang tidak benar (*unauthorized*). Pendekatan ini digunakan ketika anda memerlukan pengontrolan siapa saja yang diperkenankan membuka *database*, tetapi tidak mengontrol apa yang mereka kerjakan setelah seseorang dapat megakses *database* menggunakan *password* yang benar. Metode ini hanya dapat digunakan dengan *file .mdb*

### **Men- set *password database***

- Tutup *database*, jika *database* sedang di-*sharing* pada *network*, yakinkan bahwa tidak ada user lain sedang membukanya .
- Buat *backup database* anda.
- Pada menu *File*, click *Open*.
- Arahkan ke *folder* yang berisi *database* dan pilih pada daftar file, click tanda panah selanjutnya pilih tombol *Open*, dan kemudian click *Open Exclusive*.
- Pada menu *Tools*, arahkan ke *Security*, dan kemudian click *Set Database Password*.
- Pada kotak *Password*, ketik *password(case-sensitive)*.
- Pada kotak *Verify*, ketik ulang *password* untuk konfirmasi, dan kemudian click *OK*.
- *Password* sekarang sudah terpasang.

Setiap saat *user* mencoba membuka *database*, kotak dialog menampilkan pesan yang meminta *password*. Jika anda membuat *data access page* yang berhubungan dengan *database* ini , ketika *user* membuka *data access page*, *Internet Explorer* menampilkan kotak dialog yang meminta *password* sebelum penampilan data pada *data access page*.

### **Keamanan *User-level***

Dengan keamanan *user-level*, anda men-*create user* dan *group*, kemudian memberi ijin mengakses objek *database* (seperti *table* dan *query*). *Implicit permissions* diberikan kepada *group user*. *Explicit permissions* diberikan secara langsung kepada *user individu*. Sebagaimana telah disebutkan sebelumnya, *data access pages* disimpan di luar *database* dan tidak diberi ijin mengerjakan sesuatu (seperti *read* dan *write*) di dalam *database*, tetapi harus dikerjakan melalui *Web server* atau *file system*.

Access menyimpan informasi keamanan seperti *permission* pada *database*-nya dan menyimpan informasi *account user* and *group* dalam file informasi *workgroup* (*System.mdw*). Secara *default*, informasi *account user* and *group* disimpan di dalam file informasi *workgroup* yang diberi nama *System.mdw* yang dapat dijumpai pada *folder C:\Program Files\Microsoft Office\Office*. *Individual object permissions* disimpan dalam file *database* khusus. Cara terbaik untuk men-set keamanan *user-level* untuk *database* adalah menggunakan *User-Level security Wizard*.

*Wizard* ini membantu anda untuk men- set *account group* dan *user* dan membuat file informasi *workgroup* yang baru.

### **Cara mengamankan *database* dengan *User-Level Security Wizard***

- Buka *database* yang akan diamankan.
- Pada menu *Tools*, pilih *Security*, dan kemudian click *User-Level Security Wizard*.
- Ikuti petunjuk yang disediakan *wizard*.

Ketika anda membuat *data access pages* yang berhubungan ke *database access* dengan keamanan *user-level*, anda harus menentukan keamanan file informasi *workgroup* sebelum membuat dan mengedit *data access page*. Untuk melakukan hal ini, buka *database* dengan menggunakan *icon* yang dibuat pada *desktop* anda dengan *User-Level*

*Security Wizard*. Cara lain untuk join ke *workgroup* adalah menggunakan pilihan perintah `/wrkgrp` startup *command-line* diikuti dengan *path* penuh ke database *workgroup*.

Contoh:

```
"c:\Program Files\Microsoft Office\Office\msaccess.exe"
```

```
c:\apps\myapp.mdb /wrkgrp c:\apps\wrkgrp.mdw
```

Catatan : jika *path* berisi spasi, anda harus melengkapinya dengan *double quotation* (“ ”).

Ketika anda mendapat ijin pada suatu halaman yang berhubungan ke *database access* yang menggunakan *user-level security enabled*, *path* ke file informasi *workgroup* standar terbaru selalu disimpan dalam *property ConnectionString* dari kontrol *Data Source* pada *data access page*. Jika anda tidak berhak masuk ke file informasi *workgroup* sebelum pembuatan *data access page*, *data access page* tidak akan mereferensi ke informasi *user* dan *group* yang benar.

Jika anda telah membuat *data access pages* sebelum menjalankan *User-Level Security Wizard*, anda dapat menggunakan parameter *Jet OLEDB:System database* pada *property ConnectionString* dari kontrol *Data Source (ID attribute = MSODSC)* pada *data access page* dengan menggunakan *Microsoft Script Editor* ( pada menu Tools, arahkan ke Macro dan click *Microsoft Script Editor*) atau dengan menggunakan *text editor* seperti *Notepad*.

Ketika *user* membuka *data access page*, jika *Internet Explorer* tidak dapat membuka file informasi *workgroup* yang ditentukan pemilik *data access page*, *Internet Explorer* akan menggunakan file informasi *workgroup* standar terbaru. Hal ini tidak menjadi problem untuk *database* yang tidak riskan, tetapi *data access page* yang mengakses *database* rahasia harus dapat membuka file informasi *workgroup* yang benar atau kontrol *Data Source* pada *data access page* tidak akan dapat menghubungkan *database* dan menampilkan data, setelah *data access page* dipanggil di *Internet Explorer*.

Cara termudah untuk menjamin bahwa *data access page* menuju file informasi *workgroup* yang benar adalah meletakkan file dalam lokasi file terbagi (*shared*), kemudian sebagai *user* yang pertama join dengan file informasi *workgroup* untuk membuka *database* dan membuat *data access page*. Suatu ketika Access menunjuk ke file informasi *workgroup* terbagi, ada juga *data access pages* yang diberi ijin dalam access akan menunjuk ke file terbagi (*shared*) tersebut.

### **Pembatasan wewenang pada *data access pages* dan *Microsoft Jet***

Anda tidak dapat menggunakan metode *CurrentUser* untuk mengembalikan nama *account user* dari *script* yang menjalankan *data access page*. Metode ini adalah metode khusus yang tidak didukung oleh *ActiveX Data Objects (ADO)* atau *Data Access Objects (DAO)* dan tidak jalan pada *data access page*.

### **Keamanan *Data Access Pages* dan *MSDE***

MSDE menyediakan dua mode untuk keamanan logon. Windows 2000 mengintegrasikan *security* dan autentikasi *SQL Server*. Mode terintegrasi menggunakan *account Windows 2000* ( yang diautentikasi ketika *user log on* ke *network*) untuk men-validasi *user logging on* ke *server database*. Mode *SQL Server authentication* menggunakan *account* dan *password* awal yang didefinisikan pada *server MSDE* untuk autentikasi semua koneksi ke *database*. *Internet Information Server (IIS)* menjamin integrasi keamanan sebagai koneksi terpercaya and autentikasi *SQL Server* dianggap koneksi yang tidak terpercaya. Kedua tipe autentikasi ini dapat digunakan ketika anda men-set MSDE dan ketika anda mendefinisikan koneksi untuk *data access page*.

*SQL Server 7.0* dan *MSDE* menyediakan fungsi baru (*SUSER\_SNAME*) yang mengembalikan nama *user Windows 2000* dari *user* baru tanpa memerlukan informasi *user* yang memasukkan *logon* dan *password* baru. Fitur ini mengijinkan anda untuk membuat tampilan (*views* ) yang secara *intelligen* menggunakan informasi *user's logon* baru.

**Catatan :** *MSDE* pada Windows 95 dan 98 tidak mendukung *Windows 2000 integrated security* tetapi hanya mendukung *SQL Server authentication*. Contoh –contoh keamanan terintegrasi berikut tidak akan bekerja tanpa *database server* yang dibuat dengan teknologi Windows 2000.

### ***Set up MSDE dan account user***

Untuk menginstall *MSDE*, hanya dengan meng-klik menu “*Installing MSDE and Working with MSED Databases*”. Sekalipun yang anda pakai selain *database MSDE* atau *SQL Server 6.5* atau *7.0* yang dapat dijalankan pada *network*, anda harus menginstall *MSDE* secara local untuk menjalankan fitur *Access project security*.

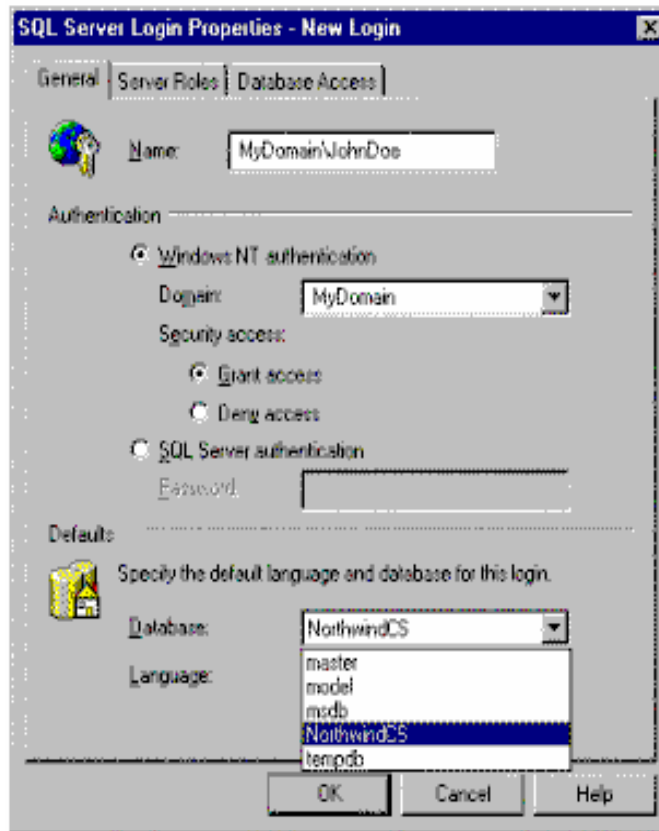
### ***Menginstall file project SQL Server Northwind.***

Contoh di bawah menggunakan project file (.adp) versi contoh dari Northwind, yang tidak diinstall secara *default*. Untuk menginstall file *project Northwind* ikuti langkah berikut:

- Buka *Microsoft Access*.
- Pada kotak dialog Access startup, pada daftar *Open an existing file*, double-click *File Northwind SQL Project*.
- Ketika ditanya apakah akan menginstall file ini, click *Yes* untuk menginstall dan membuka Northwind SQL Project file.
- *File Project Northwind SQL* terinstall pada folder *C:\Program Files\Microsoft Office\Office\Samples* sebagai file *NorthwindCS.adp* . (Anda dapat juga menginstall file database contoh dengan double-clicking pada *icon Add/Remove Programs* di *Windows Control Panel*)

Saat pertama kali anda membuka *file project Northwind SQL*,

- arahkan ke menu untuk menginstall sample database, click **Yes** untuk membuat tabel dan objek database lain pada server lokal MSDE.
- Masukkan account logon ke database, secara default, koneksi dalam *data access page* menggunakan *account system administrator* untuk *log on* ke database.
- Arahkan ke *Security* pada menu *Tools*, dan click *Database Security*. maka kotak dialog *Server Security* terbuka.
- Pada *Server Logins*, click *Add*.
- Pada menu *General*, pada kotak *Name*, ketik nama *domain/ username* (contoh, *MyDomain\jono*). Kemudian pilih bahasa *default* dan *database* untuk *account* baru.



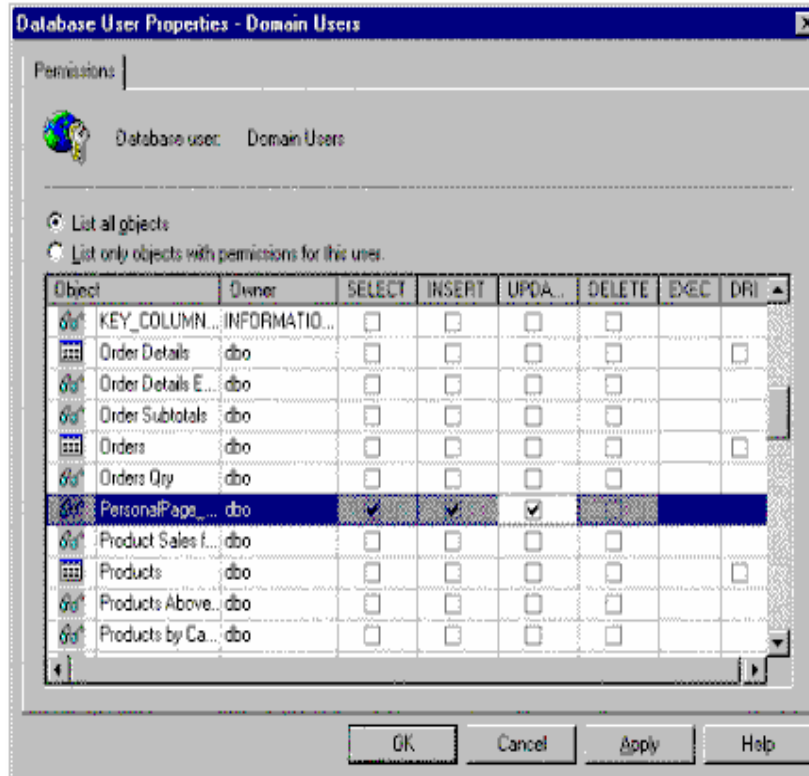
**Catatan** : Anda harus telah menginstall *MSDE* secara local untuk mengatur keamanan dari perbedaan database *MSDE* dan *SQL Server 7.0*.

Pada langkah ini adalah saat untuk menentukan *user* dan *group* yang akan mengakses *database*. Fitur utama tentang *MSDE* adalah cara yang memungkinkan anda mendapatkan kemudahan/keuntungan dari konfigurasi *network* *Windows 2000*. Contoh, anda dapat menambahkan *user group Windows 2000* sebagai *user* dan kemudian menentukan hak sebagai *user group* lebih dari *individual user*. Fitur ini membuat *seting security* lebih cepat dan biaya pendukung aplikasi yang lebih rendah.

Jika hendak diaplikasikan, berikan akses ke *user group Windows 2000*.

- Pada tombol *General*, pada kotak *Name*, ketik nama *group* yang akan anda tambahkan, menggunakan format *domainname\groupname*.
- Click *OK* untuk menambahkan *account logon* baru ke *database SQL Server*.
- Pada kotak *dialog SQL Server Security*, click tombol *Database Users*.

- Pilih *account* atau *group* yang akan anda beri hak/ijin, dan kemudian **Edit**.
- Pada kotak dialog *Database User Properties*, *click Permissions*.
- Tandai ijin untuk *account user* atau *group* yang sesuai.



**Catatan :** Yakinkan anda telah memberi ijin yang benar/sesuai. Secara *default*, *MSDE* menolak semua ijin(permission) yang tidak diberikan secara benar.

### **Membuat data access page personalized**

Failitas ini adalah untuk menggunakan fitur yang mempesona (cool) dari *MSDE* dan *data access pages*, yaitu cara untuk membuat suatu halaman yang secara *instant* memanggil informasi personal tentang user. Untuk melakukan hal ini, anda akan menggunakan fungsi *SUSER\_SNAME* guna menentukan siapakah *particular user* dan mengaitkan nama tersebut untuk membuat tabel yang menghubungkan *user* ke informasi personalnya. Fungsi *SUSER\_SNAME* adalah fungsi *system SQL Server 7.0* dan

MSDE yang mengembalikan nama domain dan nama *user* dari seseorang yang terbaru menampilkan *data access page*.

- Pertama, modifikasi tabel *Employee* untuk menyertakan informasi nama *user*.
- Pada *window Database*, click *Tables* di bawah *Objects*.
- Click *table Employees*, dan kemudian *click Design*.
- Sisipkan baris baru, berilah nama pada kolom *Username*, dan rubah tipe datanya menjadi *varchar*.

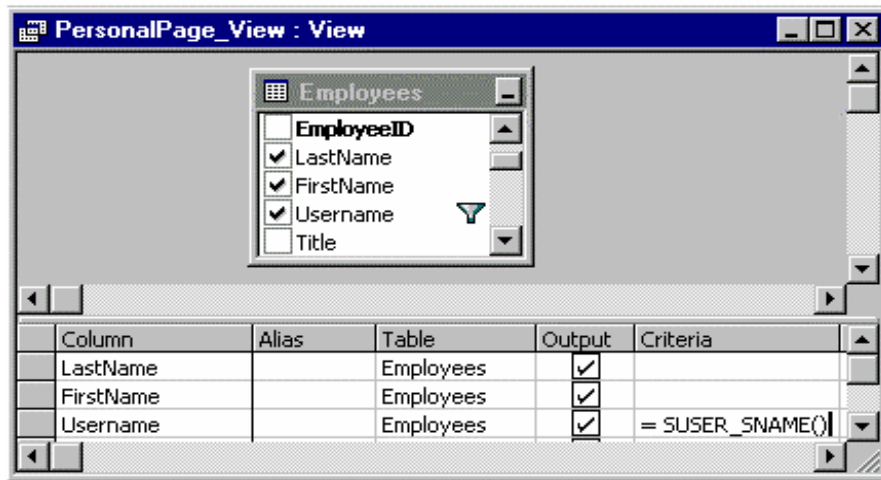
Column Name	Datatype	Length	Precision	Scale	Allow Null:
EmployeeID	int	4	10	0	<input type="checkbox"/>
LastName	varchar	20	0	0	<input type="checkbox"/>
FirstName	varchar	10	0	0	<input type="checkbox"/>
Username	varchar	50	0	0	<input checked="" type="checkbox"/>
Title	varchar	30	0	0	<input checked="" type="checkbox"/>
TitleOfCourtesy	varchar	25	0	0	<input checked="" type="checkbox"/>
BirthDate	datetime	8	0	0	<input checked="" type="checkbox"/>

- Simpan perubahan pada table *Employees*, dan kemudian pindahkan *datasheet view* dan tambahkan nama *account user* yang akan anda set up ( contoh, MyDomain\jono).

LastName	FirstName	UserName
Doe	Jono	MyDomain\jonodoe

Buatlah tampilan yang secara otomatis akan menampilkan informasi pribadi dengan memanfaatkan fungsi *SUSER\_SNAME*.

- Pada *window Database*, click *Views* di bawah *Objects* dan kemudian *click New*.
- *Click Show Table*, dan kemudian *drag table Employees* dari daftar *Show Table* ke *View designer*.
- Pada daftar *field* dari *table Employees*, pilih *field* yang memungkinkan diakses pada view, pada contoh ini, pilih *LastName*, *FirstName*, dan *Username*.
- Pada colom *Criteria* pada item *UserName*, ketik = *SUSER\_SNAME ( )*
- Simpan.



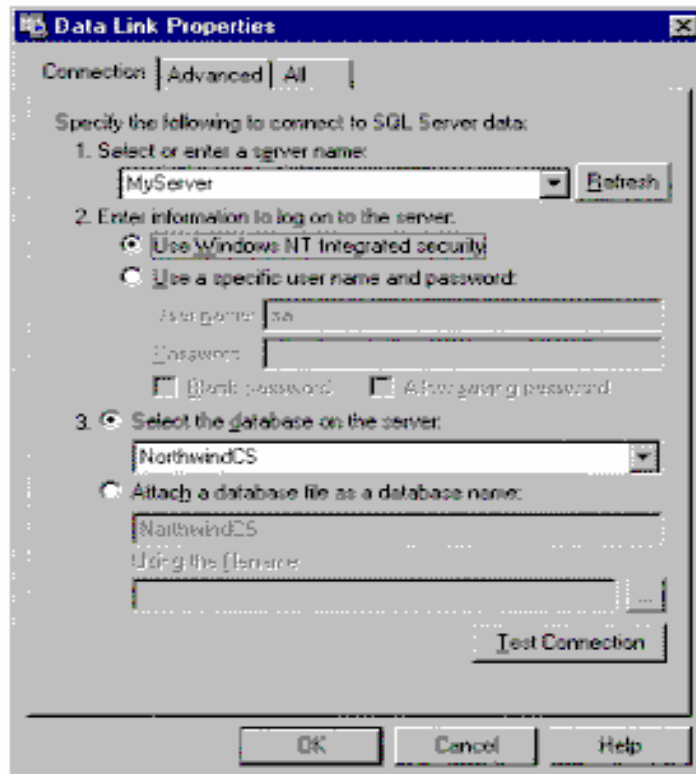
Setelah anda membua view ini , tetapi belum membuat dan menggunakan *data access page*, pilih *logon* dan *user groups* kepada yang anda ijinakan unuk melakukan operasi *SELECT, INSERT, UPDATE, atau DELETE*:

- Arahkan ke *security* pada menu *Tools*, dan kemudian *click Database Secutiry*.
- Click tombol *Database Users* .
- Pilih *account user* atau *group* yang anda ijinakan untuk mengeksekusi *view*, dan *click Edit*.
- Pada kotak dialog *Database User Properties*, *click Permissions*.
- Masukkan *account user* atau *group* untuk *object* yang sesuai pada *PersonalPage\_View*.

Untuk demonstrasi ini , berilah ijin untuk melakukan *SELECT, INSERT, and UPDATE* pada *account logon Windows 2000*.

Kini anda telah memiliki hak untuk mengeksekusi *PersonalPage\_View*, anda perlu menguji *view* dengan *logging on* menggunakan *Windows 2000 Integrated Security* dan *SQL Server Security*.

- *Click Connection* pada menu *File*
- Pilih *Use Windows NT Integrated Security*.
- Anda tidak harus memasukkan *user name* dan *password! MSDE, Windows 2000 and Access* bekerja sama untuk mendeteksi siapa anda dan apakah anda punya ijin akses ke *database*.

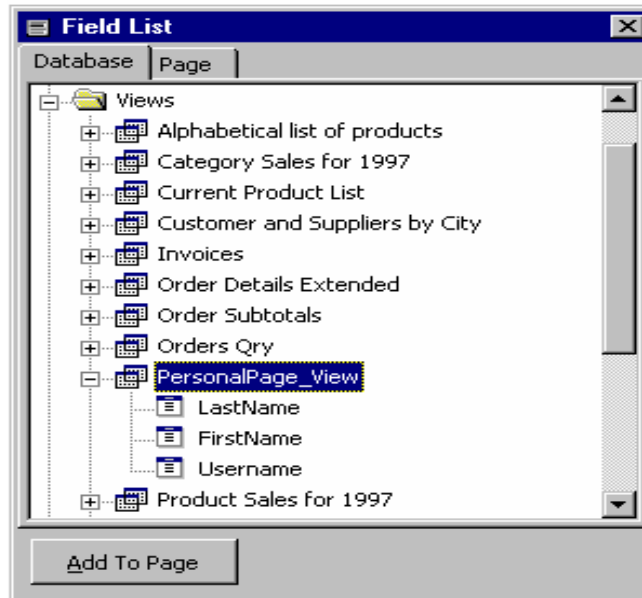


- Setelah *logging on* menggunakan *Windows NT Integrated security*, anda perlu menguji tampilan(view).
- Pada *window Database*, *click Views* di bawah *Objects*.
- *Double-click PersonalPage\_View*.

*View* akan mengembalikan semua *record* dimana *current user's domain* dan *username* sesuai dengan nilai yang ada pada kolom *UserName*.

Sekarang, buatlah *data access page* menggunakan *recordset* yang dikembalikan oleh *view (query)* baru.

- Pada *window Database*, *click Pages* di bawah *Objects*.
- *Double-click Create data access page* pada *Design view*.
- Pada menu *View*, *click Field List*.
- Lebarakan(Expand) *folder Views* dan kemudian lebarakan *Personal Page\_View*.
- Drag tiga field pada *PersonalPage\_View* dari daftar *field* ke halaman *design*.
- *Save data access page*.



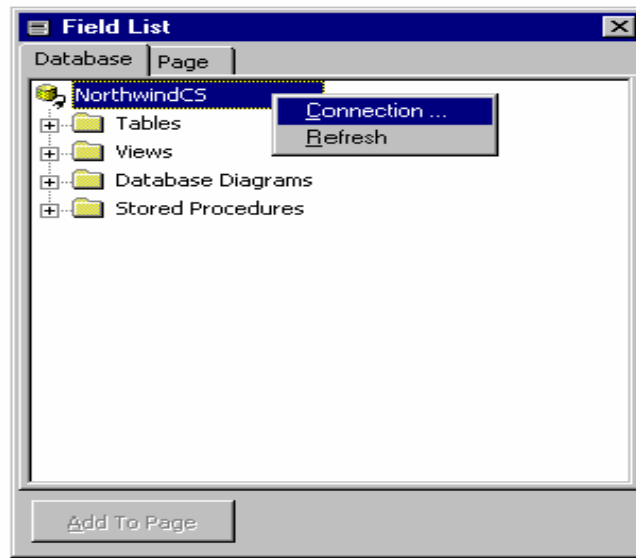
*Preview data access page.*

Catatan bahwa hanya *record* yang sesuai dengan nilai pada kolom *Username* dengan nama *logon user windows 2000* yang dikembalikan. Jika anda *shut down Access* dan menjalankan suatu halaman/page secara langsung dari *Internet Explorer*, halaman/page juga hanya akan menampilkan informasi pribadi (personalized information).

### **Nama dan password MSDE**

Cara lain dimana anda dapat memanfaatkan data access pages dengan MSDE adalah dengan melibatkan user untuk memasukkan *logon user name* dan *password MSDE (SQL Server)*. Untuk melakukan hal ini sangat sederhana.

- Pada menu *Tools*, arahkan ke *Security*, dan kemudian *click Database Security*.
- Pada kotak dialog *SQL Server Security*, buatlah *account user group* atau *individual* yang menggunakan autentikasi *SQL Server*, dan kemudian menetapkan *account object* khusus.
- Buat *data access page* baru dan tambahkan *field* yang sesuai pada *design*.
- Pada menu *View*, *click Field List*.
- Pada daftar *field*, klik kanan nama *database/server* dan kemudian *click Connection* pada menu *shortcut*.



Untuk menggunakan autentikasi *SQL Server*, tandai kotak check *Use a specific user name and password* pada tombol *Connection*.

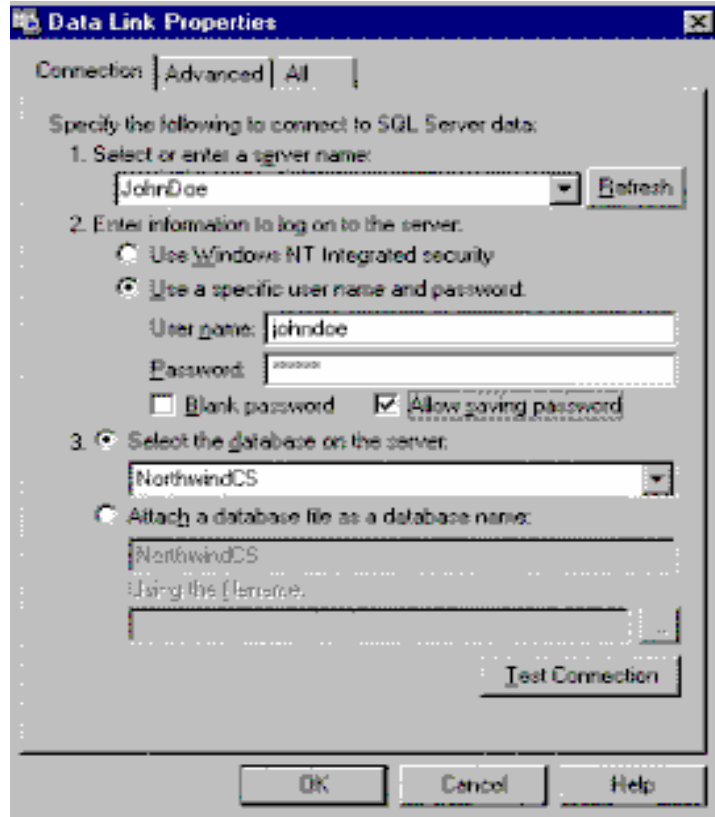
Untuk mencegah penyimpanan berulang informasi password pada *data access page's connection string*, hilangkan tanda check pada *Allow saving password*. Ketika *user* membuka *data access page*, mereka harus menyediakan nama *user* dan *password* yang valid untuk menampilkan data.

### **Menutupi (encapsulated) informasi logon menggunakan MSDE**

Fitur yang digambarkan dalam prosedur ini memungkinkan anda untuk membuat *data access pages* yang menampilkan informasi personal atau mengarahkan nama *user* dan *password* ketika dibuka. Ada beberapa situasi yang menginginkan menyampaikan informasi yang sama kepada setiap orang. Pada kasus ini, anda ingin menyediakan kepada setiap orang dengan nama *user* dan *password* yang sama kepada setiap orang. *Data access pages* memungkinkan anda untuk menyimpan nama *user* dan *password* pada dokumen HTML. Jika anda memilih untuk mengerjakan hal ini, ingat bahwa setiap orang dapat membuka dokumen HTML dan mendapatkan nama *user* dan *password*. Informasi *account* yang anda simpan di dalam halaman ini dapat digunakan bersamaan dengan mengakses *database*. Oleh karena itu, yakinkan bahwa anda membatasi ijin kepada nama *user* secara benar.

Perhatikan bagaimana kita membuat *data access page* dengan *embedded logon information*.

- Pada menu *Tools*, arahkan ke *Security*, dan kemudian click *Database Security*.
- Pada kotak dialog *SQL Server Security*, buatlah *account group* atau *user* yang menggunakan autentikasi *SQL Server*, dan kemudian tentukan *account object permissions* khusus .
- Buatlah *data access page* baru dan tambahkan *field* untuk tempat *design*.
- Pada menu *View*, click *Field List*.
- Pada daftar *field*, click kanan nama *database/server* dan kemudian pilih menu *Connection*.
- Untuk menggunakan autentikasi *SQL Server*, pilih *Use a specific user name and password* pada tombol *Connection* dan ketik nama *user* dan *password* dari account yang akan anda simpan dalam halaman tersebut.
- Berilah tanda check pada *Allow saving password* di bawah kotak *Password*.
- Simpan *data access page* ke *file system* atau *Web server*.



## **Kesimpulan**

Database merupakan asset yang sangat perlu diamankan diantaranya menggunakan password dan menentukan akses kontrol. Hal itu tentu saja sangat tidak cukup mengamankan data, komponen lain tentu saja harus saling melengkapi mulai dari kebijakan manajemen untuk menentukan anggaran bagi keamanan data, kesadaran akan pentingnya keamanan data, sumber daya manusia termasuk administrator yang mumpuni serta terus meng-*update* pengetahuannya. Ancaman terhadap keamanan data dapat datang dari mana saja, levelnya beragam misalnya, level fisik, sistem operasinya maupun dari level aplikasi. Tulisan ini hanya membahas sedikit masalah keamanan pada level program aplikasi, bagaimana cara pengamanannya ?

## **Referensi :**

1. Budi Rahardjo, "*Keamanan Sistem Informasi Berbasis Internet*"  
<http://budi.insan.co.id/courses/el695>
2. Chris Brenton, "*Mastering Network Security*", Sybex, Network Press, 1999.
3. Clint Covington, "*Creating Secure Data Access Pages*" Microsoft Corporation, 1999.  
<http://www.msdn.microsoft.com/>
4. Simson Garfinkel, "*PGP: Pretty Good Privacy*," O'Reilly & Associates Inc., 1995.
5. William Stallings, "*Network and Internetwork Security*," Prentice Hall, 1995.