

SQL SERVER 2000 SEBAGAI SOFTWARE DATABASE DALAM SMS BANKING PADA BANK SUMSEL

Maria Agustin

Staf Pengajar Jurusan Teknik Komputer Politeknik Negeri Sriwijaya

ABSTRAK

Kebutuhan untuk meningkatkan pelayanan dengan memberikan kemudahan pada nasabah atau dengan kata lain lebih memanjakan nasabah dalam melakukan transaksi telah mendorong Bank Pembangunan Daerah Sumatera Selatan untuk memanfaatkan teknologi informasi. Pemanfaatan teknologi informasi diwujudkan oleh Bank SUMSEL dengan adanya layanan SMS Banking. SMS Banking pada Bank SUMSEL bertujuan untuk mempermudah para nasabah untuk melakukan transaksi seperti transfer, cek rekening, informasi saldo dll. Software yang dipakai adalah Ms SQL Server 2000.

Kata Kunci: SQL Server, Database, SMS Banking

Pendahuluan

SMS (*Short Message Service*) Banking atau yang lebih dikenal dengan "Telepati". Telepati merupakan salah satu inovasi terbaru yang ditawarkan Bank SUMSEL dalam melayani para nasabahnya. Dengan Telepati, nasabah diharapkan dapat lebih mudah dalam melakukan transaksi perbankan sehingga menjadi lebih cepat dan praktis dimana pada jaman sekarang tidaklah aneh lagi apabila suatu kemajuan teknologi sangat dibutuhkan dalam kehidupan sehari-hari. Cukup hanya dengan mengandalkan sebuah *handphone* (telepon genggam), para nasabahnya dapat melakukan transaksi perbankan kapan dan dimana saja ia berada.

Di dalam pelayanan telepati atau SMS Banking ini terjadi suatu proses pengolahan data yang menggunakan aplikasi software Microsoft SQL Server 2000. MS SQL Server adalah salah satu produk *Relational Data Base Management System* (RDBMS) yang populer saat ini. Fungsi utamanya adalah sebagai *database*.

Server yang mengatur semua proses penyimpanan data dan transaksi suatu aplikasi. Melalui fungsi – fungsi ini Bank SUMSEL dapat melakukan pengolahan data dari nasabah yang melakukan kegiatan layanan melalui SMS Banking. Layanan- layanan ini dapat berupa ganti PIN, konfirmasi transfer, konfirmasi masalah, rekening asal, rekening tujuan, nilai, info layanan, informasi saldo, transfer, dan notifikasi.

Microsoft SQL Server 2000 memiliki fitur – fitur yang lengkap untuk membangun aplikasi mulai dari skala kecil sampai dengan tingkat *enterprise*. Oleh karena itu SMS Banking pada Bank SUMSEL menggunakan aplikasi software ini sebagai *database server* yang mengatur semua proses pengolahan data.

Landasan Teori

SQL Server 2000 memakai sebuah tipe *database* yang dinamakan *database* relasional. *Database* relasional adalah *database* yang mengorganisasikan data dalam bentuk tabel. Tabel dibentuk dengan mengelompokkan data yang mempunyai subjek yang sama. Tabel berisi baris-baris dan kolom-kolom informasi. Tabel-tabel dapat saling berhubungan jika diinginkan.

Database adalah sekumpulan data yang berhubungan. Pada waktu lalu, *database* merupakan sebuah *file*, misalnya mahasiswa.dbf, terdapat kolom-kolom yang berhubungan, yaitu NPM, Nama, Alamat, Tgl lahir, dan seterusnya. Seorang mahasiswa mempunyai sebuah baris data. Index dipakai untuk mempercepat pengaksesan data dan merupakan *file* terpisah.

Dalam SQL Server, *database* bukanlah sebuah *file* tetapi merupakan sebuah konsep logis yang berisi sekumpulan objek-objek yang berhubungan. Misalnya sebuah *database* berisi data, struktur *database*, *index*, *sekuen*, *view*, dan *stored procedure*.

Objek-objek dalam sebuah *database* :

- a. Tabel. Objek yang berisi tipe-tipe data dan data mentah.
- b. Kolom. Sebuah *table* berisi kolom-kolom untuk menampung data. Kolom mempunyai sebuah tipe dan nama yang unik.
- c. Tipe Data. Sebuah kolom mempunyai sebuah tipe data. Tipe-tipe yang dapat dipilih adalah karakter, numeric, tanggal, boolean, dan lain-lain.
- d. Stored Procedure. Merupakan perintah-perintah SQL yang membentuk makro. Dengan menjalankan *stored procedure* berarti anda menjalankan perintah-perintah SQL di dalam sebuah *procedure*.
- e. *Trigger* adalah *stored procedure* yang diaktifkan pada data yang ditambahkan, diubah, atau dihapus dari *database*. *Trigger* dipakai untuk menjamin aturan integrasi di dalam *database*. Misalnya, sebuah *trigger* dapat menjamin bahwa setiap Mahasiswa mempunyai NPM yang sesuai dengan peraturan.
- f. *Rule*. *Rule* diberlakukan pada kolom sehingga data yang dimasukkan harus sesuai dengan aturan.
- g. Kunci utama (*Primary key*). Kunci utama menjamin setiap baris data unik, dapat dibedakan dari data lain.
- h. Kunci tamu (*foreign key*). Kunci tamu adalah kolom-kolom yang mengacu kunci utama atau konstrain unik dalam tabel lain kunci utama dan kunci tamu dipakai untuk menghubungkan sebuah tabel dengan tabel lain.
- i. Konstrain. Konstrain adalah mekanisme integritas data yang berbasis *server* dan diimplementasikan oleh sistem.
- j. *Default*. *Default* dinyatakan pada *field* (kolom) sehingga jika kolom tersebut tidak diisi data, maka diisi dengan nilai *default*.
- k. *View*. *View* adalah *query* yang memakai beberapa tabel, dan disimpan dalam *database*. *View* dapat memilih beberapa kolom dari sebuah tabel atau menghubungkan beberapa tabel. *View* dapat dipakai untuk menjaga keamanan data.
- l. *Index*. *Index* membantu mengorganisasi sehingga *query* menjadi lebih cepat.

Struktur SQL Server 2000
SQL Server 2000 dikomersialkan pada tahun 2000 dan mempunyai
modern. SQL Server 2000 adalah mesin *database client/server* yang
dengan *database computer* tunggal tradisional yang memakai sistem
memakai *file* secara bersama-sama (misalnya dBASE, Microsoft Jet,
Microsoft Visual, Microsoft Visual FoxPro). *Database system* memakai *file*
bersama-sama bergantung pada sebuah proses *client* tunggal per user
memanipulasi data *file* yang dipakai bersama pada *server*
Dalam lingkungan *multiuser* akan muncul berbagai masalah, yaitu
kontrol konkurensi yang memakai mekanisme *locking* pada lapisan
work. Fasilitas sekuritas untuk menulis data pada jaringan sehingga user
ingin berbuat curang dapat memakai alat bantu lain untuk
manipulasi data. Oleh sebab itu pengontrolan data menjadi sulit.

Sistem *database client/server* seperti SQL Server 2000 memakai
proses *server* untuk memanipulasi data, dan mengharuskan proses
berhubungan dengan proses *server* menggunakan mekanisme IPC
(*inter-process communication*) aplikasi *server* yang memproses perintah-
perintah SQL. Proses *server* juga menangani konkurensi dengan memakai
mekanisme *locking* yang lebih canggih dari sistem *file* jaringan yang dipakai
bersama-sama. *Server* juga menangani sekuritas. Setelah proses *server*
melakukan perintah-perintah, hasilnya akan dikirim kembali ke proses
melalui mekanisme IPC. Dengan cara ini sistem *client/server*
memberikan pengaksesan yang lebih baik pada data yang dipakai bersama-
nya oleh banyak user.

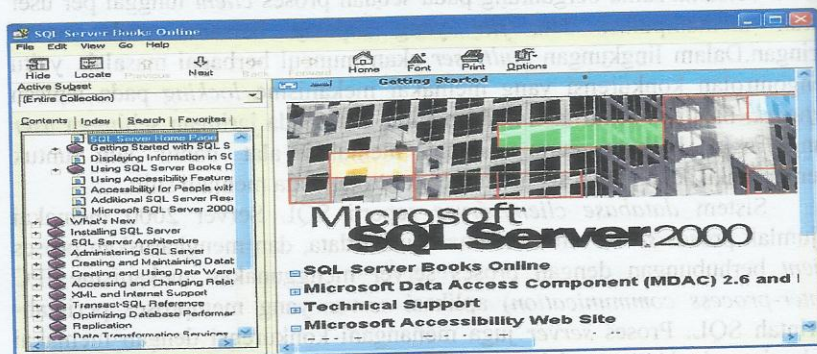
SQL Server 2000 memberikan bahasa dan antarmuka yang baik untuk
programan dan komunikasi pada *server*. Transact-SQL adalah bahasa
programan *server* yang merupakan superset dari ANSI-SQL. ANSI-SQL
mendefinisikan empat perintah dasar yaitu, *SELECT*, *INSERT*, *UPDATE*,
DELETE dan sejumlah perintah untuk mendefinisikan struktur *database*.
Transact-SQL menambahkan beberapa hal pada ANSI-SQL. Penambahan
tersebut adalah konstruksi pemograman yang memungkinkan pemakaian
prosedur untuk mengubah data dan *trigger* yang akan dijalankan,
ketika terjadi event tertentu.

Model *client/server* dipakai untuk menggambarkan arsitektur *two-tier*
aplikasi *enterprise* yang mempunyai *client* "berat" karena
implementasikan *interface user* dan proses bisnis yang rumit dan
terhubung ke sebuah *database backend* yang canggih seperti SQL Server

Memakai Alat Bantu dan Utilitas SQL SERVER 2000.

a. Books Online

Books Online adalah sumber *interface* utama. Buku ini merupakan kumpulan halaman HTML. Anda dapat mencari informasi dan mencetak jika diperlukan. Untuk mencetak sebuah topik, klik kanan pada sebuah topik lalu pilih *Print*.



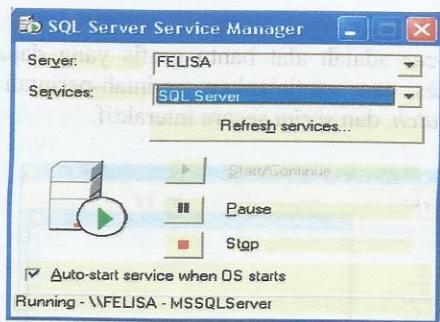
Gambar 1. Tampilan awal Sql Server 2000

Bagian kiri *Books Online* dibagi menjadi empat tab :

- *Contents* menampilkan setiap buku yang ada. Jika Anda mengetahui keberadaan sebuah topik, buka simpulnya dengan mengetik tanda +.
- *Index* berisi kata kunci dalam buku yang terurut secara *alfabetis*. Ketikkan kata yang akan dicari, *index* akan berpindah ke kata yang terdekat. Untuk menampilkan sebuah topik, klik ganda kanan topik tersebut.
- *Search* untuk mencari sebuah konsep topik, atau option. Setelah mengetikkan sebuah kata atau klausa, klik *List Topic*, ditampilkan topik-topik yang berhubungan. Untuk menampilkan sebuah topik, klik ganda topik tersebut.
- *Favorites* untuk menyimpan daftar topik favorit di dalam buku selanjutnya yang dapat dengan mudah anda dapatkan di sini. Pada saat menampilkan sebuah topik, klik tombol *Add* untuk menambahkannya ke dalam *Favorites*.

b. Service Manager

Utilitas *Service Manager* dipakai untuk menjalankan (*start*) atau memberhentikan (*stop* dan *pause*) komponen-komponen server. Komponen-komponen tersebut dijalankan sebagai servis pada Microsoft Windows 95 dan Windows 98 sebagai program exe yang terpisah.



Gambar 2. SQL Service Manager

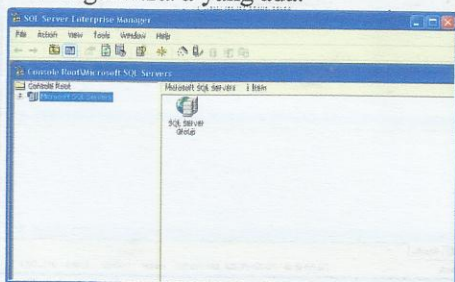
Field Server berisi nama *server* yang akan dimonitor. Kotak Service menampilkan servis-servis yang ada serta tampilan grafis dari status servis. Jika sebuah servis sedang aktif, maka tampilan akan berwarna hijau.

Service Manager mempunyai beberapa fungsionalitas yang tersembunyi. Mengklik *icon* di sebelah kiri atas, akan menampilkan menu aplikasi dengan dua menu tambahan yaitu *Connect* dan *Options*.

c. Enterprise Manager

Enterprise Manager adalah alat bantu *administrative*. Hal-hal yang dapat anda kerjakan melalui *Enterprise Manager* :

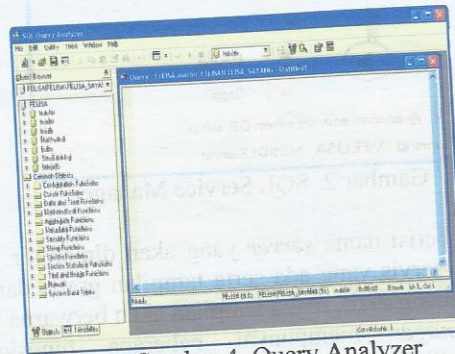
- Mendefinisikan kelompok-kelompok *server* yang menjalankan SQL Server.
- Mendaftarkan sebuah *server* ke dalam sebuah grup
- Membuat *database*, objek, *login*, *user*, dan hak-hak pada setiap *server*.
- Memanggil *Query Analyzer* untuk membuat perintah SQL dan menjalankannya.
- Membuat berbagai *wizard* yang ada.



Gambar 3. Enterprise Manager

d. Query Analyzer

Query Analyzer adalah alat bantu grafis yang dapat dipakai mendisain, mengetes dan menjalankan perintah-perintah *Transact SQL*, *stored procedure*, *batch*, dan *script* secara interaktif.



Gambar 4. Query Analyzer

Query Analyzer memberikan sejumlah jendela, kotak dialog, dan wizard yang dapat membantu Anda untuk mengelola database serta yang tersimpan di dalamnya. Bagian-bagian berikut ini akan membahas fasilitas-fasilitas yang terdapat pada *Query Analyzer*.

e. Query Window

Query window dibagi menjadi dua bagian, yaitu jendela *Editor* dan jendela *Result*. Pada saat pertama kali Anda memanggil *Query Analyzer*, hanya jendela *Editor* yang ditampilkan. Jendela *Result* akan muncul secara otomatis pada saat anda menjalankan sebuah perintah *Transact SQL*. Anda juga dapat menampilkan jendela *Result* secara manual dengan mengklik tombol *Show Result Page* pada *toolbar* atau tekan tombol *Ctrl+R*.



Gambar 5. Jendela Editor dan Jendela Results

Anda dapat mengubah tampilan kedua jendela tersebut dengan mengakses kotak dialog *Options* dari menu *Tools|Options*.

Jendela *Editor* (bagian atas) adalah jendela untuk memasukan perintah SQL dan menjalankannya. Anda dapat memasukan perintah dengan salah satu dari cara-cara berikut:

- Mengetikan perintah SQL langsung ke jendela *Editor*
- Membuka sebuah *script* SQL yang telah disimpan. Isi *script* tersebut akan ditampilkan di jendela *Editor* dan Anda dapat mengubahnya.
- Membuka sebuah *file template* dan mengubahnya.
- Memakai fasilitas *scripting* dari *Object Browser* untuk mengkopi perintah-perintah SQL dari objek *database* yang dipilih.

Jendela *Editor* memberikan alat bantu untuk penyuntingan perintah, yaitu *Undo*, *Cut*, *Copy*, *Paste*, *Select All*. Anda juga dapat mencari dan mengganti teks, memindahkan teks, menyisipkan dan membuang tanda komentar.

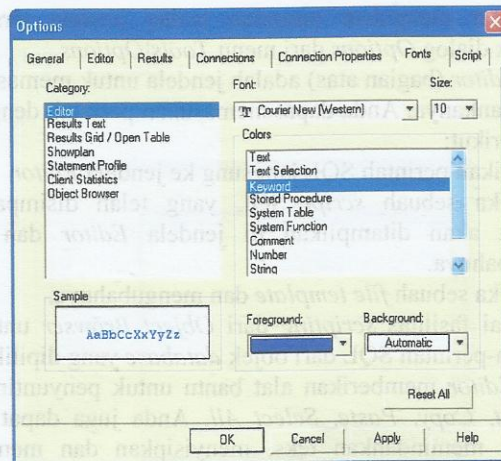
Perintah yang dimasukan ke dalam jendela *Editor* diberi warna sesuai dengan kategori. Tabel berikut memberikan warna-warna *default* untuk beberapa kategori.

Table 1. Warna Default

Warna	Kategori
Merah	String Karakter
Merah Tua	Stored Procedur
Hijau	Tabel Sistem
Hijau Tua	Komentar
Magenta	Function system
Biru	Kata kunci
Abu-Abu	Operator

Anda dapat mengubah warna-warna tersebut dari tab *Fonts* dari kotak dialog *Options*.

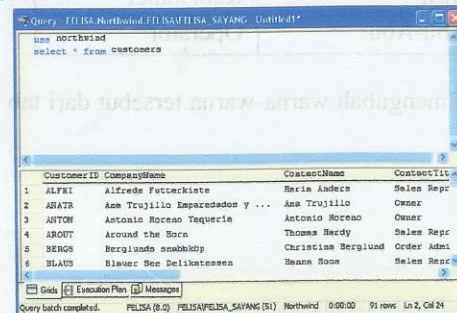




Gambar 6. Tab Fonts dari Kotak Dialog Options

Pemakaian warna akan membantu Anda dalam mencari kesalahan. Misalnya Anda mengetikkan sebuah kata kunci dan tidak ditampilkan dengan warna biru, berarti kata kunci tersebut salah. Misalnya, jika terlalu banyak perintah yang ditampilkan dengan warna merah yang menunjukkan string karakter, mungkin anda lupa memberikan tanda petik penutup sehingga seluruh teks dianggap sebagai *string*.

Di bagian bawah jendela *results* terdapat tiga buah tab, yaitu *Grids*, *Execution Plan*, dan *Messages*. Tab *Grids* menampilkan data dalam bentuk *grid*. Tab *Execution Plan* akan menampilkan rencana eksekusi dalam bentuk grafik. Jika anda memakai perintah *SET SHOWPLAN_ALL* atau *SET SHOWPLAN_TXT*, rencana eksekusi ditampilkan dalam bentuk tabel. Tab *Grids* selalu bersama-sama dengan tab *Message* yang menampilkan pesan untuk *query* tertentu.



Gambar 7. tiga buah tab pada Jendela Results

1. Object Browser

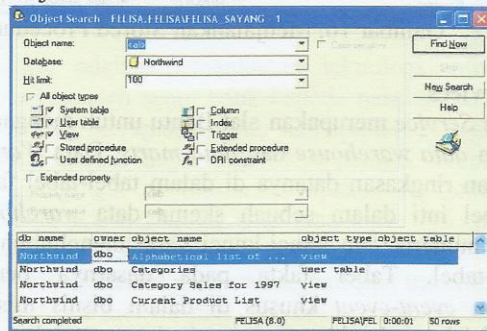
Object Browser adalah alat bantu dalam bentuk pohon untuk melacak objek-objek dalam sebuah *database*. Selain fungsi *navigasi*, *Object Browser* memungkinkan pembuatan *script*, mengeksekusi *stored procedure*, dan mengakses tabel atau *view*.

Object Browser mempunyai dua tab :

- Tab *Object* yang menampilkan semua objek dalam sebuah *database* dan menampilkan objek-objek umum seperti *function-function* dan tipe data yang sudah diberikan oleh SQL Server.
- Tab *Templates* dipakai untuk mengakses folder *Templates*.

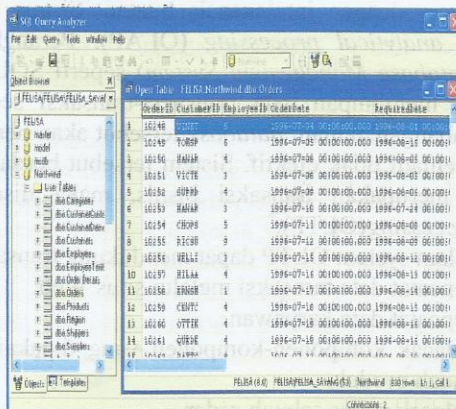
Beberapa hal yang dapat dijalankan pada *Object Browser*:

- 1) Mencari Objek dalam *database* :



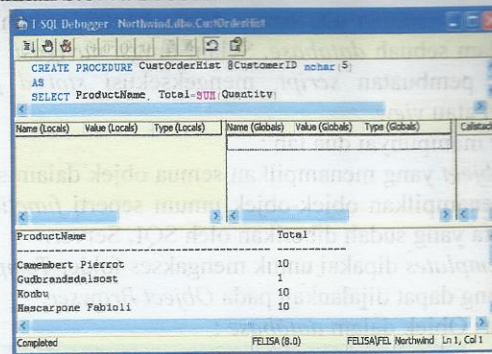
Gambar 8. Kotak Dialog Object Search

- 2) Menampilkan Isi Tabel



Gambar 9. Menampilkan Isi Tabel

3) Menjalankan *Stored Procedure*



Gambar 10. Menjalankan *Stored Procedure*

g. Analysis Services

Analysis Service merupakan alat Bantu untuk menganalisis data yang disimpan dalam *data warehouse* dan *data marts*. *Data Warehouse* dan *data mart* menyimpan ringkasan datanya di dalam tabel-tabel fakta. Tabel fakta merupakan tabel inti dalam sebuah skema *data warehouse* yang berisi ukuran-ukuran numerik dan kunci-kunci yang menghubungkan fakta-fakta dengan tabel-tabel. Tabel fakta pada dasarnya berisi data yang menggambarkan *event-event* khusus di dalam bisnis misalnya transaksi-transaksi di dalam bank atau penjualan produk-produk. Aplikasi berhubungan dengan *Analysis Service* melalui ADO dan OLE DB.

Data Warehouse dan *data mart* dipakai untuk sistem pakar yang memproses *query* dengan intelegensia buatan. Sistem tersebut dinamakan sistem *online analytical processing* (OLAP). Perangkat lunak RDBMS (*Relational Database Management System*) seperti SQL Server 2000, pada mulanya akan menyimpan data transaksi-transaksi secara terpusat. Oleh karena perkembangan zaman, *database* tersebut akan berukuran sangat besar pemrosesan menjadi tidak efektif. Sistem tersebut berdasarkan pada *record-record* yang merupakan transaksi bisnis, maka disebut *system online transaction processing* (OLTP).

Data dalam system OLTP dapat mendukung transaksi-transaksi :

- Mencatat data transaksi melalui situs Web.
- Mencatat data karyawan.
- Melacak komponen-komponen yang dipakai untuk membangun sebuah produk.
- Melacak status sebuah order.

Transaksi ini dilakukan tidak terduga, namun atau ributnya untuk dipakai untuk dalam bentuk yang manajer memerlukan untuk faktor-faktor. Sistem OLAP managerial. Sistem jumlah besar.

Hasil Dan Pembahasan

1. SMS Banking
SMS Banking layanan yang diberikan transaksi. SMS Banking SQL Server 2000, seluruh provider Simpati, Kartu Prepaid Banking yaitu transaksi menunggu karena Banking dapat dilakukan dapat menerimanya. Transaksi yang setiap saat, praktis dan Notifikasi SMS instansi pemerintah Banking ini dipromosikan.

2. Proses Pembuatan

Dalam pembuatan Server 2000 pada Samsel) dapat menggunakan Server 2000 dan Diagram, Select, Insert. Berikut adalah menggunakan perintah

Transaksi individual dapat diakses dengan cepat, karena data yang diakses tidak terlalu banyak. Sistem OLTP dirancang untuk menangani ratusan atau ribuan transaksi yang dijalankan pada saat bersamaan. OLTP cocok dipakai untuk tingkat operasional. Data OLTP tidak diorganisasikan dalam bentuk yang mudah untuk tingkat manajerial. Biasanya, seorang manajer memerlukan informasi ringkasan untuk analisis yang didasarkan untuk factor-faktor kritis.

Sistem OLAP menjawab masalah pada system OLTP untuk tingkat manajerial. Sistem OLAP akan memproses *query* yang mendata dalam jumlah besar.

Hasil Dan Pembahasan

1. SMS Banking

SMS Banking adalah pemanfaatan teknologi informasi sebagai layanan yang diberikan oleh suatu bank kepada nasabah untuk kemudahan transaksi. SMS Banking merupakan salah satu aplikasi penggunaan database SQL Server 2000. Layanan dapat digunakan pemilik telepon seluler dari seluruh provider yang ada di Indonesia, apakah itu produk telkomsel, Simpati, Kartu Hallo, Kartu Matrix, IM3 atau kartu AS. Kelebihan SMS Banking yaitu transaksi bisa dilakukan on line dan real time tanpa harus menunggu karena tidak terdapat jeda waktu. Jadi saat transaksi selesai, SMS Banking dapat dilakukan saat itu juga dan orang yang melakukan transaksi dapat menerimanya.

Transaksi yang dilakukan pun begitu fleksibel namun tetap aman setiap saat, praktis dan efisien selama 24 jam. Inquiry (lihat / mencek) Saldo dan Notifikasi SMS Banking bisa dinikmati oleh nasabah perorangan dan instansi pemerintah daerah / kabupaten / kota. Untuk keamanan SMS Banking ini diproteksi dengan menggunakan PIN.

2. Proses Pembuatan Database

Dalam pendesainan database dengan menggunakan Microsoft SQL Server 2000 pada Layanan SMS Banking di BPD Sumatera Selatan (Bank Sumsel) dapat memakai beberapa perintah yang disediakan di dalam SQL Server 2000 diantaranya New Database, New Table, Create Database Diagram, Select, Inner Join dan lain sebagainya.

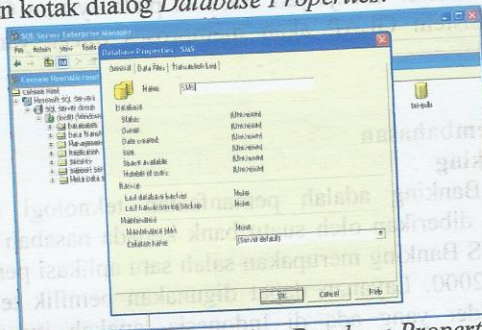
Berikut adalah langkah-langkah pendesainan database dengan menggunakan perintah-perintah di atas.

2.1 Membuat Database Memakai *Enterprise Manager*

2.1 Membuat Database Memakai Enterprise Manager

Database adalah sekumpulan data dalam tabel yang saling berhubungan antara satu tabel dengan tabel lainnya. Pada tahap awal kita memulai dengan membuat database. Berikut langkah-langkah membuat database dari Enterprise Manager :

1. Buka tab **Tree** sampai menemukan server.
2. Klik kanan pada simpul *databases*, lalu klik **New Database**.
Ditampilkan kotak dialog *Database Properties*.



Gambar 11. Kotak dialog *Database Properties*

3. Ketikkan nama *database* pada **Field Name** lalu klik **OK**.

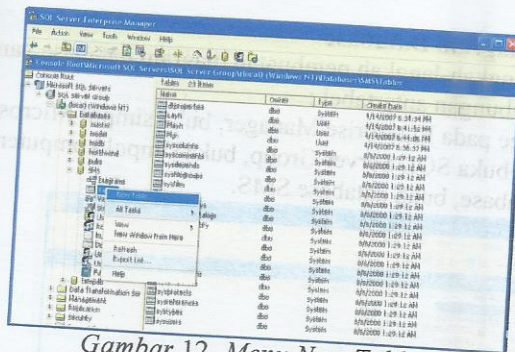
2.2 Membuat Tabel

2.2 Membuat Tabel

Tabel adalah kumpulan data mengenai sebuah objek, misalnya langganan, barang, mahasiswa, penjualan. Sebuah tabel harus disimpan dalam sebuah database. Sebuah tabel berisi sejumlah kolom. Setiap kolom dinyatakan dengan sebuah atribut. Jika Anda membuat tabel minimal Anda harus mendefinisikan nama tabel, nama-nama kolom, tipe data setiap kolom, apakah kolom boleh berisi nilai NULL dan konstrain-konstrain lainnya.

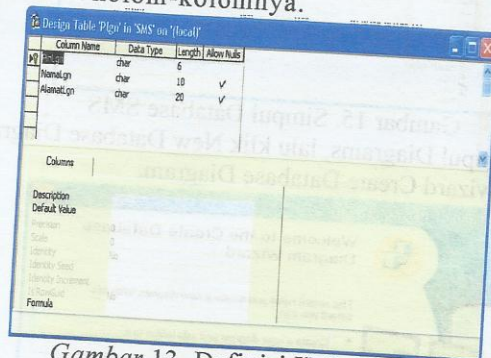
Berikut langkah-langkah pembuatan tabel dengan menggunakan console tree yang ada pada Enterprise Manager :

1. Buka folder **Databases** dan cari database yang akan ditambah tabel baru.
2. Klik kanan pada folder **Tables**, pilih menu **New Tables**. Cara lain adalah dengan memilih menu **Action|New Tables**.



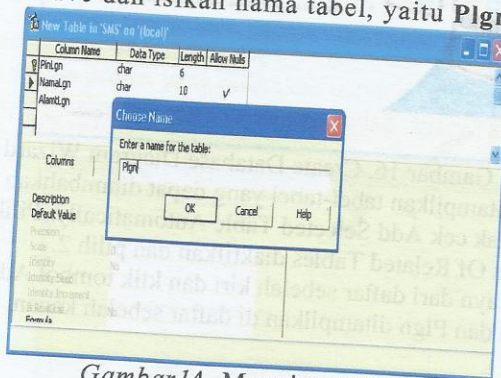
Gambar 12. Menu New Tables

3. Isikan definisi kolom-kolomnya.



Gambar 13. Definisi Kolom-Kolom

4. Klik icon Save dan isikan nama tabel, yaitu **Pign** (Pelanggan).

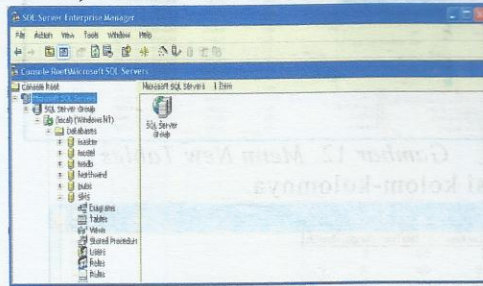


Gambar 14. Menyimpan Tabel

2.3 Membuat Diagram Database

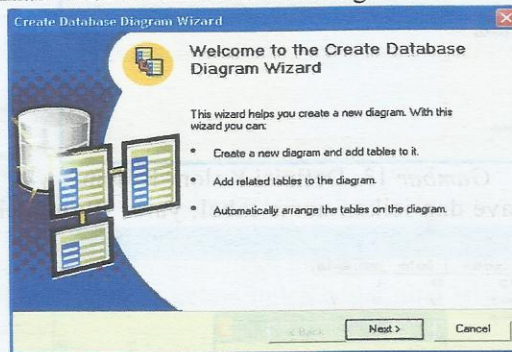
Berikut langkah-langkah pembuatan diagram database yang menunjukkan hubungan antartabel :

1. Dari tab Tree pada Enterprise Manager, buka simpul Microsoft SQL Server, lalu buka SQL Server Group, buka simpul komputer kita, bukalah simpul Database, buka database SMS.



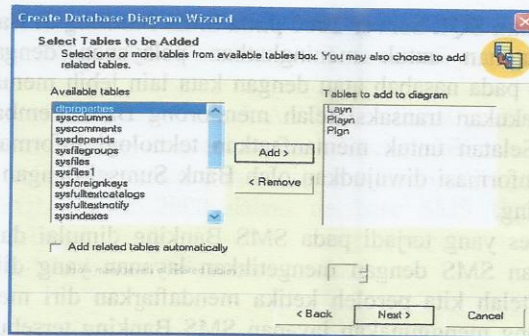
Gambar 15. Simpul Database SMS

2. Klik kanan simpul Diagrams, lalu klik New Database Diagram. Ditampilkan wizard Create Database Diagram.



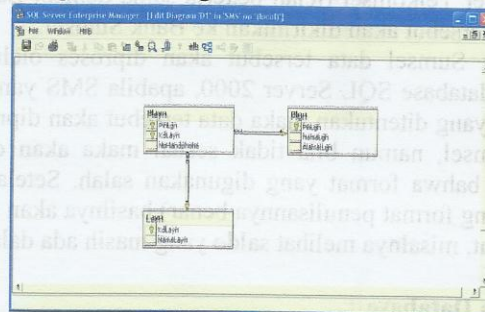
Gambar 16. Create Database Diagram Wizard

3. Klik Next, ditampilkan tabel-tabel yang dapat ditambahkan ke diagram.
4. Aktifkan kotak cek Add Selected Table Automatically. Pilihan How Many Levels Of Related Tables diaktifkan dan pilih 2.
5. Pilih tabel Layn dari daftar sebelah kiri dan klik tombol Add. Tabel Layn, Playn dan Plgn ditampilkan di daftar sebelah kanan.



Gambar 17. Memilih Tabel

6. Klik Next dan akhiri wizard dengan klik Finish. Diagram ditampilkan di jendela baru. Diagram berisi tiga tabel tersebut.



Gambar 18. Database Diagram

2.4 Tampilan Query dari Database

Query adalah proses yang dilakukan untuk menjalankan perintah-perintah yang digunakan. Berikut adalah tampilan Query yang dihasilkan oleh database :

	NamaLain	NoKartu	NamaLain
1	Uti	00135004788	info
2	Rai	007000000	info
3	Harold	001000000	cek saldo
4	Novi	000000000	cek saldo
5	Rai Rai	001000000	cek saldo
6	Harold	000000000	info
7	Rai	001000000	info
8	Rai	000000000	cek saldo

Gambar 19. Hasil Query Database Layanan SMS Banking

3 Penggunaan SQL Server 2000 pada SMS Banking di Bank Sumsel

Kebutuhan untuk meningkatkan pelayanan dengan memberikan kemudahan pada nasabah atau dengan kata lain lebih memanjakan nasabah dalam melakukan transaksi telah mendorong Bank Pembangunan Daerah Sumatera Selatan untuk memanfaatkan teknologi informasi. Pemanfaatan teknologi informasi diwujudkan oleh Bank Sumsel dengan adanya layanan SMS Banking.

Proses yang terjadi pada SMS Banking dimulai dari nasabah yang mengirimkan SMS dengan mengetikkan layanan yang diinginkan beserta PIN yang telah kita peroleh ketika mendaftarkan diri menjadi salah satu nasabah yang menggunakan layanan SMS Banking tersebut. SMS tersebut dikirimkan ke operator provider dengan nomor 3311 (khusus pengguna Telkomsel) atau 0811713311 (seluruh GSM). SMS ini akan diterima oleh operator provider Telkomsel (telah bekerja sama dengan Bank Sumsel) dan kemudian data tersebut akan dikirimkan ke Bank Sumsel.

Di Bank Sumsel data tersebut akan diproses oleh server dengan menggunakan database SQL Server 2000, apabila SMS yang dikirim sesuai dengan format yang ditentukan maka data tersebut akan diproses lebih lanjut oleh Bank Sumsel, namun bila tidak sesuai maka akan dikirimkan SMS pemberitahuan bahwa format yang digunakan salah. Setelah proses selesai (untuk SMS yang format penulisannya benar) hasilnya akan diberitahukan ke nasabah tersebut, misalnya melihat saldo yang masih ada dalam rekening.

4 Sistem Kerja Database

Pada database SQL Server 2000 terdapat data-data pelanggan yang menggunakan layanan SMS Banking. Selain itu terdapat pula data layanan-layanan yang akan diberikan oleh SMS Banking ini, misalnya data layanan cek saldo rekening tabungan. Pada database ini digunakan PIN untuk mengenali nasabah. Dengan PIN ini akan diketahui siapa nama nasabah pemilik PIN tersebut dan semua data yang berhubungan dengan nasabah tersebut, PIN ini juga digunakan sebagai penghubung antar tabel dimana PIN ini berfungsi sebagai Kunci Utama. Selain itu database ini juga menggunakan Kunci Tamu yang berupa layanan-layanan yang ada sehingga database ini dapat mengetahui dan memproses layanan yang diinginkan nasabah dengan membaca layanan yang diminta. Hingga hasil dari permintaan nasabah dikirimkan kembali kepada para nasabah.

Kesimpulan

Adapun Kesimpulan yang dapat kami tarik dari hasil pembahasan diatas :

1. SMS (Short Message Service) BANKING "TELEPATI" merupakan salah satu inovasi yang ditawarkan Bank Sumsel dalam melayani para nasabahnya dalam melakukan transaksi perbankan melalui ponsel.
2. Microsoft SQL Server 2000 dalam database SMS Banking dapat digunakan sebagai database server yang mengatur semua proses penyimpanan data dan transaksi suatu aplikasi.

Daftar Pustaka

- Amri, M. Choirul. 2003. *Pengantar Administrasi Microsoft SQL Server 2000*. <http://ilmukomputer.com/>.
- Martina, Inge. 2003. *Microsoft SQL Server 2000*. Elex Media Komputindo: Jakarta.