

BAB III

LANDASAN TEORI

3.1 Microsoft Visual Basic .NET 2003

Visual Basic .NET 2003 merupakan pengembangan dari bahasa pemrograman sebelumnya yaitu *Visual Basic 6.0*. Beberapa keunggulan *Visual Basic .Net 2003* dibandingkan dengan *VB 6.0* versi sebelumnya adalah:

1. Menyederhanakan *Deployment*

Visual Basic 2003 mengatasi masalah seputar *deployment* dari aplikasi berbasis *Windows* yaitu ‘DLL Hell’, sehingga dapat mempermudah *deployment* aplikasi yang berbasis *Windows*.

2. Menyederhanakan Pengembangan Perangkat Lunak

Visual Basic 2003 memiliki fasilitas penanganan *bug* yang bekerja secara efisien dan bersifat *real-time background compiler* dan daftar *task*, untuk penanganan kesalahan/*bug* program, sehingga pengembang dapat secara langsung memperbaiki kesalahan kode program yang terjadi secara tepat dan cepat.

3. Mendukung OOP

Dalam *Visual Basic 2003* dapat membuat kode *class* yang menggunakan secara penuh konstruksi berbasis objek. *Class-class* tersebut dapat digunakan kembali. *Visual basic 2003* memiliki fitur bahasa pemrograman berorientasi objek

termasuk implementasinya secara penuh: *inheritance*/pewarisan, *encapsulation*/pembungkusan, dan *polymorphism*/ banyak bentuk.

4. Mempermudah Pengembangan Aplikasi Berbasis Web

Untuk mengembangkan aplikasi *Web* disediakan *designer form Web* yang baru, sehingga pengembang dapat dengan mudah membuat aplikasi berbasis *Web* yang berjalan di *browser* dan *platform* manapun. *Designer HTML* yang baru menggunakan *intelliSense Statement* untuk *tag HTML* dan *User Interface (UI)* yang terpisah dengan kode-kode yang efisien.

5. Mempermudah Migrasi dari VB.6 ke Visual Basic 2003

Microsoft Visual Basic 2003 mempunyai kemudahan dan kemiripan kode untuk berbagai macam bahasa pemrograman yang diusung. Hal ini akan mempermudah bagi pengembang yang mempunyai latar belakang bahasa pemrograman yang berbeda untuk menguasai *Visual Basic 2003*. *Visual Basic 2003* memungkinkan pengembang dapat melakukan migrasi lebih dari 95% kode *VB 6.0* menjadi kode *Visual Basic 2003*

3.2 SQL Server

Database bukanlah sebuah file tetapi merupakan sebuah konsep logis yang berisi sekumpulan objek-objek yang berhubungan. Sebuah database berisi *data*, *struktur database*, *index*, *sekuritas view*, dan *stored procedure*.

Database *SQL Server* adalah database yang memiliki sistem *client/server* serta memakai sejumlah proses *server* untuk memanipulasi data, dan

mengharuskan proses *client* berhubungan dengan proses server menggunakan mekanisme *IPC (inter-process communication)* lokal atau remote, misalnya *socket TCP/IP*. Proses server adalah proses aplikasi server yang memproses perintah SQL. Proses server juga menangani konkurensi dengan memakai mekanisme *locking* yang lebih canggih dari pada jika sistem file jaringan dipakai secara bersama-sama. Server juga menangani sekuritas. Setelah proses server menjalankan perintah-perintah SQL, hasilnya akan dikirim kembali ke proses *client* melalui mekanisme *IPC (client/server)*.

SQL Server memberikan bahasa dan antarmuka yang baik untuk pemrograman dan komunikasi pada server. *Transact-SQL* adalah bahasa pemrograman server yang merupakan *superset* dari *ANSI-SQL*. *ANSI-SQL* mendefinisikan empat perintah dasar untuk memanipulasi data yaitu *SELECT*, *INSERT*, *UPDATE*, dan *DELETE*, serta sejumlah perintah untuk mendefinisikan struktur database. *Transact-SQL* menambahkan beberapa hal pada *ANSI-SQL* penambahan tersebut adalah konstruksi pemrograman yang memungkinkan pemakaian *stored procedure* untuk mengubah data dan *trigger* yang akan dijalankan, karena terjadi event tertentu.

Objek-objek dalam *SQL Server* adalah:

1. Tabel

Objek yang berisi tipe-tipe data dan data mentah.

2. Kolom

Sebuah tabel berisi kolom-kolom untuk menampung data. Setiap Kolom memiliki nama yang unik dan tipe tertentu.

3. Tipe Data

Sebuah kolom mempunyai sebuah tipe data. Tipe-tipe yang dapat dipilih adalah karakter, numerik, tanggal, dan Boolean.

4. Stored Procedure

Merupakan perintah *SQL* yang membentuk makro. Dengan menjalankan *stored procedure* berarti pengembang menjalankan perintah-perintah *SQL* di dalam sebuah *procedure*.

5. Trigger

Merupakan *stored procedure* yang diaktifkan pada saat ditambahkan, diubah, atau dihapus dari database. *Trigger* dipakai untuk menjamin integritas di dalam database.

6. Rule

Rule diberlakukan pada kolom sehingga data yang dimasukkan harus sesuai dengan aturan.

7. Primary Key

Berfungsi menjamin data setiap baris yang bersifat unik, dan dapat dibedakan dari data lain.

8. Foreign Key

Foreign Key adalah kolom-kolom yang mengacu pada *Primary Key* atau konstrain unik pada tabel lain. *Primary Key* dan *Foreign Key* dipakai untuk menghubungkan sebuah tabel dengan tabel yang lain.

9. Konstrain

Konstrain adalah mekanisme integritas data yang berbasis server dan diimplementasikan oleh sistem.

10. Default

Default dinyatakan pada field (kolom) sehingga jika kolom tersebut tidak diisi data, maka diisi dengan nilai *default*.

11. View

View adalah *query* yang memakai beberapa tabel, dan disimpan di dalam database. *View* dapat memilih beberapa kolom dari sebuah tabel atau menghubungkan beberapa tabel. *View* juga dapat dipakai untuk menjaga keamanan data.

12. Index

Index berfungsi membantu mengorganisasi data sehingga proses *query* menjadi lebih cepat.

Daya tampung database *SQL Server* tidak terbatas hanya pada ukuran 2GB, bahkan 1 tabel saja dari database *SQL Server* mampu menampung data sebanyak 2GB. Dengan kata lain database *SQL Server* dapat menampung data hingga 1 *Terabyte*.

SQL Server memakai sebuah tipe database yang dinamakan database relation. Database relation adalah database yang mengorganisasikan data dalam bentuk tabel. Tabel dibentuk dengan mengelompokkan data yang mempunyai

subjek yang sama. Tabel berisi baris dan kolom informasi. Tabel - tabel dapat saling berhubungan jika diinginkan.

