



Visual Basic Overview

Visual Basic adalah sebuah bahasa pemrograman dan lingkungan pengembangan (development environment) yang berjalan diatas Sistem operasi Windows. Bahasa pemrograman ini kaya akan feature dan kelebihan-kelebihan yang dapat membantu kita membuat berbagai macam tipe/jenis aplikasi/program yang berbeda. Kita dapat membuat Program yang berinteraksi dengan database, intranet, internet dan bahkan dengan perangkat keras/hardware sekalipun.

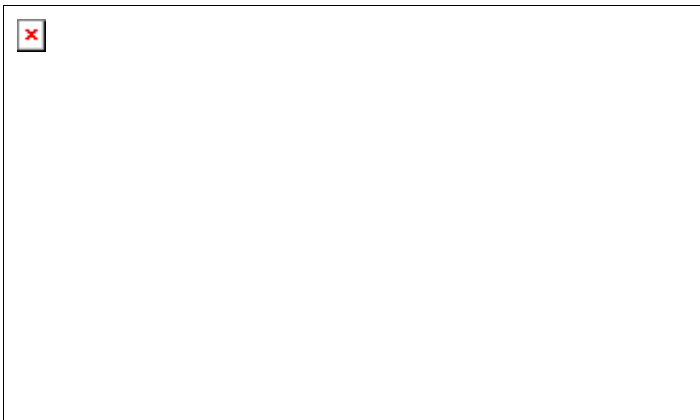
Apa itu Visual Basic

Visual Basic adalah sebuah bahasa pemrograman yang dapat menciptakan aplikasi yang berbasis Windows (windows based applications). Bahasa Visual Basic sebenarnya berdasarkan bahasa BASIC (Beginner's All-Purpose Symbolic Instruction Code) yang lebih tua. Apa yang membuatnya "Visual" adalah kita dapat "melukis" interface/tampilan dari program yang akan kita hasilkan dibandingkan dengan menulis kode per kode/baris per baris dalam bahasa BASIC yang lama.

Sejak masih berdasarkan bahasa BASIC yang lebih tua, Visual Basic telah berkembang dengan sangat cepat. Ratusan Fungsi dan tekhnologi terbaru programming telah ditambahkan kedalamnya membuat Visual Basic menjadi standar industri baru dalam membuat aplikasi yang multi guna diatas sistem operasi Windows.

Aplikasi Visual Basic

Untuk membuat sebuah aplikasi/program dengan Visual Basic kita harus mengambil "sedikit" komponen-komponen yang terdapat didalamnya dan "melekatkannya" untuk menjadi sebuah aplikasi/program. Komponen yang umumnya kita pakai adalah Form, Control, Class dan Procedure.



Form adalah sebuah window yang mana kita akan "melukis" interface/tampilan program kita kedalamnya. Control adalah "bangunan" (building blocks) dari interface yang kita buat tersebut. Control juga disebut sebagai ActiveX Control, adalah antarmuka seperti Label, TextBox, CommandButton dan masih banyak lagi yang lain. Control inilah nantinya yang akan menerima input dari user, menampilkan Informasi dan merespon aksi dari user tersebut.

Hirarki Aplikasi Visual Basic

Ketika menggunakan Visual Basic maka kita akan dengan cepat mempelajari bahwa ada urutan-urutan tertentu (hirarki) dari objek-objek (komponen) yang kita gunakan. Sebuah aplikasi dibentuk berdasarkan Form, Module dan Class. Sebuah Form dibuat atas Property, Procedure, Event dan Control. Control juga dibentuk atas Property dan Event. Dibawah ini adalah sedikit gambaran dari komponen-komponen dasar yang membentuk sebuah aplikasi Visual Basic.

Application

Application Properties

Forms

- Properties
- Property Procedures
- Event Handlers
- Other Procedures

Controls

- Properties
- Event Handlers

Modules

- Procedures
- Functions

Classes

- Properties
- Property Procedures
- Public Methods
- Private Methods

3 Langkah untuk membuat Aplikasi dengan Visual Basic

Tidak peduli tipe aplikasi apa yang akan dibuat, kita hanya memerlukan tiga langkah untuk itu :-). Jika kita selalu mengingat langkah-langkah ini, maka pengembangan aplikasi kita akan menjadi lebih lancar.

1. Create the Interface (membuat antarmuka/tampilan).
2. Set Properties for Controls (men-set property dari control-control yang terdapat didalamnya).
3. Write code (menulis kode).

Kita harus selalu membuat tampilan/interface program yang akan kita hasilkan sebelum menuliskan kode-nya. Jika kita "melukis" dulu tampilannya kemudian men-set control dan property-nya, maka kita akan dapat menghemat waktu pengembangan aplikasi dengan signifikan.

Sejarah Visual Basic

Visual Basic dikembangkan dari bahasa QuickBasic yang lebih tua yang berjalan diatas sistem operasi DOS. Versi awal Visual Basic pertama kali diciptakan oleh Alan Cooper dan kemudian dijual ke Microsoft. Kemudian Microsoft mengambil-alih pengembangan produk yang belum sepenuhnya jadi tersebut, dan memberikannya nama sandi (code-named) "Thunder". Dan kemudian menciptakan Bahasa Pemrograman yang dengan cepat menjadi Bahasa pemrograman utama didalam lingkungan Windows.

Perkembangan Visual Basic

- Versi 1.0 dirilis pada bulan Mei 1991.
- Professional Toolkit, sekumpulan custom control dimasukkan kedalam Visual Basic, pada bulan January 1992.
- Versi yang lebih "canggih" (2.0) dikeluarkan pada bulan Oktober 1992
- Versi 3.0 dirilis pada bulan Mei 1993. Dengan dikeluarkannya versi terbaru ini, maka Microsoft mulai saat itu telah benar-benar serius untuk membuat suatu "development tool" yang dapat diandalkan.
- Versi 4.0, dikeluarkan pada bulan September 1995, adalah ketika Visual Basic mulai menambahkan konsep bahasa pemrograman yang ber-orientasi objek (Object Oriented Programming), hal ini menjadikan Visual Basic semakin populer dan digunakan sebagai bahasa utama dalam mengembangkan aplikasi yang ber-basis Windows.
- Versi 5.0 dirilis pada bulan Maret 1997 dan menambahkan kemampuan untuk membuat ActiveX Control, sehingga menjadikannya pesaing serius dari Visual C++
- Versi 6.0 dirilis pada bulan Agustus 1998. Microsoft menambahkan banyak fitur-fitur lain yang lebih powerful dan berguna dalam mengembangkan aplikasi, contohnya: Banyaknya Wizard yang akan membantu kita mengembangkan aplikasi lebih cepat.

Feature yang terdapat didalam Visual Basic

Visual Basic mempunyai banyak fitur yang sangat berguna dalam mengembangkan aplikasi yang berbasis Windows. Bahkan "terlalu banyak" sehingga akan kesulitan bagi kita untuk men-daftar semua fitur-fitur tersebut. Berikut ini adalah beberapa fitur yang paling umum dan banyak digunakan:

- Sekumpulan coding (Superset) dari bahasa pemrograman VBA 5.0 (Visual Basic for Application) telah disertakan didalam semua produk Microsoft Office.
- Tampilan lingkungan pengembangan (design environment) yang GUI (Graphical User Interface) sehingga memudahkan kita men-develop aplikasi yang user-friendly.
- Kemampuan untuk mengembangkan/men-design dan mencaoba program/aplikasi yang sedang kita buat menggunakan kemampuan "interpretive Run Function"
- Menciptakan/compile file .exe secara P-Code atau native code (akan diterangkan lebih lanjut)
- Object based development dengan menggunakan Class modules.
- Rapid Application Development
- Dapat membuat COM (Component Object Model) seperti ActiveX Control, DLL dan file .EXE
- Kemungkinan untuk mengembangkan aplikasi yang internet-ready
- Membuat ActiveX dokumen
- Membuat aplikasi DHTML
- Membuat aplikasi IIS (Internet Information Server)
- Kontrol Web Browser
- Mendukung FTP (File Transfer Protokol)
- Kontrol Winsock untuk mengembangkan aplikasi yang berbasis IP.
- Mempunyai fasilitas Help dan Books online yang terintegrasi dengan rapi
- Fasilitas Debugging yang bagus
- Mempunyai banyak wizard yang sangat membantu dalam mengerjakan tugas yang berulang-ulang
- Visual Basic juga dapat ditingkatkan kemampuannya dengan mudah dengan menggunakan pemanggilan fungsi API, ratusan control/OCX/DLL pihak ketiga dan kemampuan lainnya yang dapat mengintegrasikan aplikasi berbasis Windows melalui COM dan DCOM
- Mempunyai akses ke database dengan berbagai macam metode dan berhubungan dengan hampir semua database engine.

Keuntungan Menggunakan Visual Basic

Visual Basic mempunyai banyak kelebihan dibandingkan Software/bahasa pemrograman yang lain. Dibawah ini adalah beberapa keuntungan tersebut:

- Kurva pembelajaran dan pengembangan yang lebih singkat dibandingkan bahasa pemrograman yang lain seperti C/C++, Delphi atau bahkan PowerBuilder sekalipun.
- Menghilangkan kompleksitas pemanggilan fungsi Windows API, karena banyak fungsi-fungsi tersebut sudah di-"embeded" ke dalam syntax Visual Basic.
- Cocok digunakan untuk mengembangkan aplikasi/program yang bersifat "Rapid Application Development".
- Juga sangat cocok digunakan untuk membuat program/aplikasi Bisnis.
- Digunakan oleh hampir semua keluarga Microsoft Office sebagai bahasa Macro-nya, segera akan diikuti oleh yang lain.
- Dapat membuat ActiveX Control
- Dapat menggunakan OCX/Komponen yang disediakan oleh pihak ketiga ("third party") sebagai "tool" pengembangan.
- Menyediakan wizard yang sangat berguna untuk mempersingkat/mempermudah pengembangan aplikasi
- Mendekati Object Oriented Programming
- Dapat di-integrasikan dengan Internet, baik itu pada sisi Client maupun pada sisi Server
- Dapat membuat ActiveX Automation Server
- Integrasi dengan Microsoft Transaction Server

- Dapat menjalankan server tersebut dari mesin yang sama atau bahkan dari mesin/komputer yang lain.

Kekurangan/Kelemahan Visual Basic

Walaupun Visual Basic mempunyai banyak kelebihan, tetapi tetap saja mempunyai kekurangan/kelemahan. Kelemahan-kelemahan ini sebenarnya disebabkan oleh keterbatasan Visual Basic dalam "mengambil" fungsi-fungsi yang bersifat low-level yang berhubungan dengan Hardware maupun Operating System (Windows) itu sendiri, antara lain:

- File Distribusi runtime-nya lebih besar dari kepunyaan C/C++.
- Tidak mempunyai fungsi-fungsi untuk mengambil feature-feature dari OS sebanyak C/C++.

Code Generation Options

Pada dasarnya ada dua jenis bahasa pemrograman: *interpreter* dan *compiler*. Visual Basic memperbolehkan kita menggunakan keuntungan-keuntungan metode tersebut didalam satu lingkungan pengembangan!. Ini adalah salah satu feature yang terbaik yang terdapat didalam Visual Basic. Interpreter adalah sebuah bahasa yang akan menjalankan aplikasi setelah semua baris-baris code yang kita tulis diterjemahkan kedalam bahasa mesin seluruhnya. Sedangkan Compiler adalah bahasa yang menterjemahkan baris-baris code tersebut sekaligus "sekali jalan". Keuntungan Interpreter adalah waktu pengembangan yang singkat, sedangkan Compiler unggul dalam menjalankan program yang sudah jadi (.exe)

Dengan Visual Basic kita mendapatkan keduanya, karena Visual Basic mempunyai kemampuan untuk menjalankan Interpreter pada saat desain aplikasi dan meng-compile source-codenya untuk menciptakan file runtime (.exe) yang siap pakai. Visual Basic menggunakan versi modifikasi dari bahasa Interpreter yang disebut dengan **P-Code**. Sekarang marilah kita lihat perbedaan antara metode P-Code dengan Native Compiled.

P-Code

Bahasa Interpreter harus menterjemahkan setiap baris dari source-code kedalam bahasa mesin (satu dan nol) setiap kali baris code tersebut dieksekusi. Jadi jika ada sebuah baris code yang dieksekusi 100 kali maka Interpreter harus menterjemahkannya 100 kali pula. Untuk menulis bahasa interpreter, kita hanya perlu menuliskan baris source code-nya dan menjalankannya. Contoh dari bahasa ini adalah Bahasa pemrograman tradisional seperti GW-Basic dan batch-language.

Bahasa Compiler akan menterjemahkan semua baris source code sekaligus kedalam bahasa mesin. Untuk menulis program yang bersifat compiler kita hanya tinggal menuliskan source-codenya, meng-compile dan menghubungkannya lalu kemudian menjalankannya. Contoh dari bahasa compiler ini adalah COBOL, Fortran dan C.

P-Code adalah gabungan dari bahasa Interpreter dan compiler. Program compiler akan meng-compile source-code tetapi tidak menghasilkan bahasa mesin, tetapi ia akan menghasilkan sekumpulan symbo-symbol yang mewakili source-code. Simbol-simbol ini membentuk sebuah baris dari code yang mempunyai 5 kata kedalam hanya 1 atau 2 character. Ketika kita menjalankan program tersebut maka , interpreter P-Code akan menterjemahkan setiap baris tersebut kedalam bahasa mesin. Ketika ia (P-Code) dieksekusi akan lebih cepat dari pada code interpreter karena membaca character yang lebih sedikit, tetapi tetap saja lebih lambat dari code yang di compile dengan bahasa Compiler. Visual Basic mempunyai pilihan untuk menghasilkan P-Code atau native code.

Native Code

Visual Basic juga mempunyai kemampuan untuk menghasilkan/generate file native code .exe. Hal ini dapat meningkatkan performance dari aplikasi yang menggunakan perhitungan CPU yang intensive.

Jika kita mempunyai bagian dari program yang kita buat yang menggunakan perhitungan CPU secara intensive maka kita dapat menghasilkan file .exe-nya kedalam native code. Atau kita juga dapat menghasilkan native code untuk membuat aplikasi DLL bahasa C yang kita panggil didalam Visual Basic.

Metode mana yang akan kita pilih ?

Ketika menjalankan program yang berbasis Windows, kebanyakan sumber-daya CPU digunakan untuk memanggil fungsi-fungsi Windows yang low-level ataupun DLL lain dan tidak oleh source-code yang terdapat didalam program. Kebanyak dari sumber-daya tersebut juga dipakai untuk menunggu user menjalankan perintah selanjutnya; seperti meng-click tombol next atau bahkan mengetik character berikutnya didalam TextBox. Oleh karena itu metode P-Code Visual Basic cukup cepat digunakan untuk membuat aplikasi bisnis sekalipun.

Masa Depan Visual Basic

Sejak diperkenalkannya Visual Basic for Application (VBA), Microsoft telah menempatkan Visual Basic sebagai bahasa pemograman utama didalam lingkungan Sistem operasi Windows. Ia dijadikan alat untuk "merekatkan" component-component tersebut kedalam sebuah Solusi perangkat lunak (Software Solution). Semua produk Microsoft Office dewasa ini telah menyertakan Visual Basic sebagai bahasa macro-nya. Dan akan banyak lagi pembuat perangkat lunak pihak ketiga yang akan melisensi bahasa VBA ini kedalam produk mereka.

Competition

Kompetitor/saingan utama dari Visual Basic adalah PowerBuilder ciptaan PowerSoft dan tentu saja produk Microsoft lainnya seperti Visual C++, Access, Visual Foxpro dan Borland Delphi. Tetapi walaupun bahasa-bahasa tersebut cukup bagus tetapi tetap saja tidak sesukses dan "senikmat" menggunakan Visual Basic. :-)

Versi-versi Visual Basic

Ada 3 jenis versi berbeda dari Visual Basic 6.0. Dibawah ini adalah penjelasan singkat tentang ke-tiga versi tersebut dan untuk siapa saja versi-versi tersebut diperuntukkan.

Visual Basic Learning Edition

Versi ini diperuntukkan bagi mereka yang hanya sekedar ingin mencoba dan mempelajari bahasa Visual Basic, atau seorang pengajar yang ingin memberikan pelajaran Visual Basic bagi murid-muridnya. Edisi ini datang dalam sebuah CD-ROM yang dengan interaktif menunjukkan kita bagaimana menggunakan VB. Versi ini sekaligus mengemas sekumpulan control-control standar dan memperbolehkan kita membuat program .EXE dan COM DLL.

Visual Basic Proffesional Edition

Versi ini ditujukan bagi mereka-mereka yang akan membuat aplikasi/program yang sederhana. Terdiri dari banyak control-control standar dan sekaligus mengemas model object ADO yang lebih lengkap dibandingkan dengan versi Learning Edition yang hanya menyertakan ADO Data Control. Versi ini mampu menghasilkan native code .EXE, ActiveX Control, ActiveX Document dan ActiveX EXE atau DLL. Juga terdapat wizard-wizard yang sangat membantu dalam pengembangan aplikasi.

Visual Basic Enterprise Edition

Versi ini mencakup semua fasilitas dan feature yang terdapat didalam VB Profesional Edition dan juga tools-tools lain yang dapat membantu kita untuk menghasilkan suatu aplikasi enterprise yang lebih kompleks. Edisi ini ditujukan untuk mereka yang akan membangun sebah aplikasi yang lebih besar dan luas didalam sebuah perusahaan atau korporasi. Versi ini datang dengan banyak sekali control-control standar dan juga:

- SQL Server 6.5 Developer Edition
- Microsoft Transaction Server
- Visual SourceSafe
- Visual Database Tools
- Integrated T-SQL Server Debugger
- dll.

Tipe Aplikasi yang dihasilkan oleh Visual Basic

Ada berbagai macam jenis/tipe aplikasi yang dapat dihasilkan oleh Visual Basic. Aplikasi-aplikasi yang berbeda ini digunakan untuk memecahkan sebuah persoalan yang spesifik. Marilah kita lihat apa saja tipe aplikasi tersebut.

Standard EXE

Kita dapat memilih tipe aplikasi ini jika kita akan membuat sebuah program aplikasi biasa untuk keperluan bisnis yang umum tanpa harus berhubungan dengan aplikasi lain.

ActiveX EXE

Tipe aplikasi ini biasanya dibuat untuk digunakan oleh aplikasi lain sebagai bagian dari suatu service. Service ini dapat berjalan didalam komputer yang sama ataupun pada komputer lain melalui "remote automation" atau DCOM. Biasanya service ini tidak menyediakan interface visual.

ActiveX DLL

Tipe aplikasi ini biasanya dibuat untuk digunakan oleh aplikasi lain sebagai bagian dari suatu service. DLL ini harus terdapat pada komputer yang sama dengan aplikasi yang menjalankannya. Aplikasi ini tidak menyediakan Interface Visual. Tetapi ia dapat juga dijalankan pada komputer yang lain yang menjalankan Microsoft Transaction Server.

ActiveX Control

Tipe aplikasi ini biasanya digunakan untuk membuat suatu component control atau yang biasa disebut OCX. OCX ini nantinya dapat digunakan oleh aplikasi-aplikasi lain dalam lingkungan pengembangan.

ActiveX Document DLL

Tipe aplikasi ini biasanya dikembangkan untuk digunakan didalam context aplikasi lain. Sangat mirip dengan ActiveX EXE dan DLL tetapi dapat menyediakan interface visual bagi aplikasi yang memanggilnya. DLL ini dijalankan pada "program space" yang sama dengan aplikasi yang memanggilnya. Ada sebuah document (VBD) yang menyertainya, document ini harus dijalankan didalam sebuah container seperti Microsoft Binder ataupun Microsoft Internet Explorer versi 4 keatas.

ActiveX Document EXE

Tipe aplikasi ini biasanya dikembangkan untuk digunakan didalam context aplikasi lain. Sangat mirip dengan ActiveX EXE dan DLL tetapi dapat menyediakan interface visual bagi aplikasi yang memanggilnya. EXE ini dijalankan pada "program space" yang berbeda dari aplikasi yang memanggilnya. Ada sebuah document (VBD) yang menyertainya, document ini harus dijalankan didalam sebuah container seperti Microsoft Binder ataupun Microsoft Internet Explorer versi 4 keatas.

IIS Application

Sebuah aplikasi yang kita buat untuk dijalankan dibawah Internet Information Server. Aplikasi ini akan dipanggil dari sebuah file ASP untuk mengerjakan suatu proses khusus yang biasanya akan sangat sulit untuk dilakukan oleh sebuah file ASP. Tipe aplikasi ini bersifat browser independent. Artinya kita tidak membutuhkan sebuah browser tertentu karena code-code-nya akan dijalankan oleh server.

DHTML Application

Kita dapat membuat dan mencoba code snippet dari Dynamic HTML yang nantinya akan dijalankan pada Internet Explorer versi 4.0 keatas di komputer client. Dengan menggunakan Visual Basic maka kita akan dapat mencoba aplikasi DHTML lebih mudah jika dibandingkan dengan menggunakan editor lain seperti NotePad ataupun FrontPage.

Visual Basic Application Components

Ketika pertama kali akan membuat sebuah aplikasi dengan Visual Basic, pertama-tama kita harus mengenal dulu apa saja tipe file (disk file) yang akan disimpan oleh Visual Basic. Ada banyak macam type/jenis file yang akan dihasilkan oleh VB, tergantung jenis aplikasi apa yang akan kita buat.

Projects

Sebuah Project Visual Basic dapat terdiri dari sekumpulan form, control, reference, setting aplikasi dan informasi lainnya. Kesemua component ini mempunyai tipe file sendiri-sendiri tetapi terkumpul didalam sebuah project yang sama. Sebuah file Project sebenarnya adalah merupakan file ASCII biasa yang berekstension .VBP (Visual Basic Project). Didalam file tersebut kita akan melihat daftar dari file-file lain yang membentuk project tersebut. Jika kita menggabungkan project-project tersebut kedalam satu lingkungan pengembangan (design environment) maka VB akan membuat sebuah project group yang berekstensi .VBG

Forms

Sebuah form adalah sebuah window yang akan ditampilkan pada desktop Windows (Operating System) ataupun didalam sebuah Form MDI. Form umumnya merupakan tempat untuk meletakkan control-control dari aplikasi yang akan kita buat, juga terdapat code-code program yang berhubungan dengan aplikasi, form disimpan dengan ekstensi .FRM. Jika didalam form tersebut terdapat control yang bertipe graphic maka graphic tersebut akan disimpan didalam file yang berbeda (.FRX) tetapi dengan nama yang sama.

Controls

Control adalah sebuah alat/tool yang akan menampilkan informasi ataupun mendapatkan informasi dari user. Contoh dari Control adalah button, label, graphic, textbox dan masih banyak lagi. Control-control ini biasanya diletakkan diatas sebuah form sebagai user interface. Control terdapat pada jendela Toolbox didalam lingkungan pengembangan Visual Basic. Terdiri atas dua jenis: Standar dan Custom. Standar adalah control default yang sudah disediakan oleh VB, sedangkan Custom Control biasanya kita dapatkan dari pihak ketiga (third-party). Control berekstensi .OCX

Code Modules

Module adalah tempat dimana kita akan menulis kode sumber didalamnya yang bersifat global (dapat dikenal oleh semua form didalam sebuah project VB). Kode-kode ini disimpan dalam file ASCII yang berekstensi .BAS

Class Modules

Class Module memperbolehkan kita membuat object sendiri. Class module ini adalah prototype dari object yang akan kita buat tersebut (property dan method-nya). Class module disimpan dalam sebuah file yang berekstensi .CLS

User Controls

Kita dapat membuat custom control sendiri dengan Visual Basic disebut User Control. Custom Control ini disimpan didalam sebuah file .CTL yang juga akan membuat file .CTX jika didalamnya terdapat graphics.

Property Pages

Property pages digunakan didalam context sebuah User Control. Page-page ini dibuat untuk merubah ataupun melihat property yang terdapat pada sebuah user control dengan cara yang mudah dimengerti. Disimpan didalam sebuah file yang berekstensi .PAG

User Document (ActiveX Document)

Dengan menggunakan Visual Basic kita dapat membuat sebuah form ataupun aplikasi dan memberinya ekstensi .VBD. Hal ini tidak hanya akan menghasilkan file .VBD itu sendiri tetapi file ActiveX atau .EXE yang diperlukan untuk menampilkannya. Active Document ini dapat disertakan kedalam container lain seperti Microsoft Binder ataupun Internet Explore.

IIS Application

IIS Application dibuat berdasarkan sekumpulan WebClass dan WebItem. Sebuah WebClass adalah DLL yang berjalan dibawah IIS 4.0 dan mempunyai sekumpulan WebItems yang dapat berbentuk template HTML yang akan dikirimkan ke browser. File-file yang dihasilkan berekstensi .DSR dan .DSX

DHTML Application

Aplikasi DHTML mempunyai kode-kode yang disimpan didalam Code Modules dan file-file .DSR atau .DSX. Kita juga akan mempunyai file .HTM yang berasosiasi dengan project ini. Setelah kita meng-compile project ini dan akan mendistribusikannya, maka kita akan mengirimkan file HTML dan file DLL ke end user.

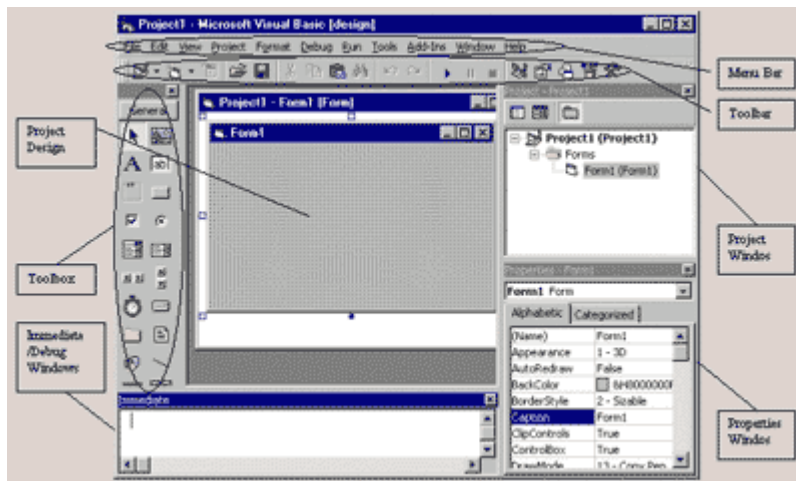
Resource File

Sebuah file resource terdiri dari kumpulan string, bitmap, cursor dan juga icon yang nantinya dapat kita load kedalam aplikasi yang kita kembangkan. Salah satu contoh penggunaan file ini adalah ketika kita akan menciptakan aplikasi lintas bahasa. Dengan menggunakan file Resource maka kita hanya perlu memanipulasi string yang terdapat didalam file resource tersebut tanpa harus mengganti setiap baris kode program. File ini berekstensi .RC dan setelah dikompilasi maka kita akan mendapatkan sebuah file .RES yang siap digunakan.

Visual Basic Environment (Lingkungan Pengembangan Visual Basic)

Lingkungan pengembangan Visual Basic adalah sebuah aplikasi MDI (Multi Document Interface). Interface/antar muka VB sama dengan lingkungan pengembangan Microsoft C++ dan Visual Interdev (produk-produk Developer Studio), tetapi tetap memiliki perbedaan tersendiri. Kita harus menguasai setiap elemen yang terdapat didalamnya untuk menjadi seorang programmer/developer yang handal.

Gambar 1: Lingkungan Pengembangan Visual Basic



Menu Bar

Menu bar berfungsi untuk menjalankan suatu perintah atau fungsi pengembangan project VB. Menu-menu ini dikelompokkan kedalam kelompok-kelompok tertentu yang akan membantu kita melakukan design aplikasi. Berikut penjelasan mengenai pengelompokkan-pengelompokkan menu-menu tersebut:

- File : terdiri dari perintah-perintah untuk membuka, menutup, menyimpan, men-cetak, meng-compile Project yang sedang kita kerjakan
- Edit : kumpulan perintah yang membantu kita memanipulasi penulisan code (listing program)
- View : terdiri dari perintah untuk melihat/berpindah antar window didalam lingkungan pengembangan
- Project : sekumpulan perintah untuk menambah component Project (Form, Module, dll) dan tempat dimana kita akan men-set-up properties Project
- Format : perintah-perintah untuk memanipulasi Control yang terdapat pada sebuah Form
- Debug : Sekumpulan perintah yang berguna untuk melacak bugs ataupun melacak/trace listing program yang sedang kita kerjakan line-per-line
- Run : perintah untuk menjalankan, pause dan menghentikan jalannya Program/aplikasi
- Tools : terdiri dari wizard-wizard yang sangat membantu pengembangan aplikasi
- Add-Ins : sekumpulan component/wizard yang bisa digunakan dalam pembuatan aplikasi
- Window : kumpulan window-window yang terdapat didalam Project
- Help : berisi file-file bantuan/help dan tentang Visual Basic itu sendiri

Toolbar

Berisi icon-icon yang mewakili perintah-perintah yang terdapat didalam Menu Bar (shortcut). Toolbar ini secara default berisi perintah-perintah umum untuk memanipulasi Project. Toolbar bisa di-set sesuai dengan keinginan kita sendiri.

Toolbox

Berisi control-control yang dapat kita pakai/letakkan didalam sebuah form. Secara default berisi control-control umum yang sudah disiapkan oleh VB. Kita dapat menambahkan sendiri control-control yang kita inginkan. (Click kanan pada area Toolbox dan Pilih: Components...)

Form

Sebuah form adalah komponen utama dari aplikasi yang akan kita hasilkan. Kita dapat menambahkan sebanyak mungkin form kedalam aplikasi kita sesuai dengan kebutuhan.

Project Window

Project Window berisi semua component yang terdapat didalam Project yang sedang kita kembangkan (Form, Module, Class, dll).

Properties Window

Properties Window akan memperlihatkan semua properti yang tersedia pada sebuah object/control/component yang sedang aktif (focus). Kita dapat melakukan manipulasi properties terhadap objek-objek tersebut. Secara default properti-properti itu diurutkan berdasarkan abjad.

Immediate or Debug Window

Immediate window (atau lebih dikenal dengan Debug Window) adalah tempat dimana kita dapat melihat jalannya program secara step-by-step jika Break Mode dijalankan. Didalam window ini kita dapat melakukan perhitungan aritmatika, melihat isi variabel dan bahkan memanggil sebuah procedure yang terdapat didalam listing program.

Online Help

Jika kita menekan tombol F1 dimana saja didalam lingkungan pengembangan Visual Basic maka secara otomatis Visual Basic akan menampilkan artikel bantuan yang kita butuhkan. Coba letakkan pointer pada sebuah baris code (katakanlah pada syntax DateAdd), tekan F1 -> maka secara otomatis VB akan menampilkan help tentang Syntax DateAdd tersebut, tetapi kita harus menginstall MSDN (Microsoft Developer Network) sebagai file Help-nya. Biasanya MSDN dikeluarkan secara periodik oleh Microsoft dalam bentuk CD yang dapat kita beli dipasaran.

Panduan Gaya Pemrograman

Bahasa Basic mudah atau sulit dibaca? Jawabannya tergantung pada siapa Kita berbicara, apa latar belakang Kita dan seberapa bersih kebiasaan pemrograman. Di masa lampau, nomor baris dan pernyataan GOTO memudahkan pembuatan kode yang acak-acakan. Untungnya, versi Basic saat ini jauh lebih terstruktur dan berorientasi objek. Meskipun bila mau, Kita dapat mempertahankan agar kode tetap mudah dibaca.

Sebenarnya, dengan penanganan yang hati-hati, source-code Visual Basic dapat lebih mudah dibaca dan diikuti daripada bahasa manapun. Segudang anjuran mengenai gaya pengkodean standar telah muncul beberapa tahun terakhir ini, sebagian diantaranya ada yang bagus namun ada pula yang tidak begitu bermanfaat. Tetapi yang paling penting adalah menentukan standar kita sendiri dan tentu saja standarisasi pengkodean bersama jika sebuah aplikasi akan dikerjakan bersama-sama dengan programmer yang lain. Tetapi tetap saja, ada baiknya standar tersebut juga dapat diikuti dan mudah dibaca oleh programmer secara umum, dengan begitu reputasi kita sebagai programmer yang handal akan semakin meningkat.

Setting Awal Aplikasi

Sebelum memulai membuat sebuah aplikasi/program dengan VB, ada baiknya kita mendokumentasi terlebih dahulu apa yang akan kita kerjakan dan "planning" selanjutnya. Selalu setting Project Properties program yang akan kita buat. Tentukan Project Type, Project Name, Project Description, Versi Program, Comments, Copyrights, dan lain sebagainya. Memang hal ini bagi sebagian programmer dirasa terlalu "nyeleneh" :-), tetapi akan sangat terasa sekali manfaatnya jika suatu saat nanti kita akan membuat sebuah aplikasi yang besar dan kompleks.

Control Prefiks

Salah satu teknik termudah yang dapat diterapkan adalah dengan memberi nama masing-masing control/objek dengan prefiks standar yang terdiri dari tiga huruf, yang mengidentifikasikan rujukan control/objek di source-code dan form. Hal ini benar-benar akan meningkatkan kemudahan membaca listing program. Sebagai contoh, andaikan suatu event-driven subprogram dengan nama Print_Click. Merujuk ke tombol perintah ? gambar ? atau pilihan ? mungkin kita akan sangat kebingungan, Print ini merujuk ke objek atau control apa ?. Tentunya cmdPrint_Click akan mudah untuk dikenali sebagai milik suatu Command Button (tombol perintah), picPrint_Click milik picture control dan sebagainya. Daftar prefiks standar yang telah diterima secara luas dan telah dipublikasikan di berbagai tempat banyak sekali. Kita coba akan melihat standarisasi yang telah dikenal tersebut sekali lagi, berikut ini adalah beberapa contoh diantaranya:

Prefiks	Control / Objek	Prefiks	Control / Objek
cbo	Combo box	lsv	List view
chk	Check box	med	MaskedEdit Multimedia

cmd	Command Button	mci	MCI
dat	data	mnu	Menu
db	ODBC database	ole	OLE Container
dbg	Data-bound grid	opt	Option Button
dir	Directory list box	otl	Outline
dlg	Common dialog box	pic	Picture box
drv	Drive list box	pnl	Panel
fra	Frame	prb	ProgressBar
frm	Form	rtb	Rich textbox
grd	Grid	shp	Shape
img	Image	stb	Status Bar
iml	ImageList	tbs	Tabstrip
lbl	Label	tmr	Timer
lin	Line	tlb	Toolbar
lst	List box	txt	Text box

Nama Variabel

Beberapa orang, khususnya yang datang dari dunia pemrograman C, menganjurkan penamaan seluruh variabel dengan prefiks Notasi Hungaria, serupa dengan control prefiks yang tertera sebelumnya. Tetapi teknik ini agak meragukan. Disatu sisi, lebih baik bila kita mengetahui tipe variabel mana yang ditangani saat membaca di kode, namun di sisi lain, kadang menyebabkan sintaks menjadi berantakan dan agak sulit dibaca.

Sejak awal, programmer VB telah memiliki cara penamaan variabel dengan sufiks untuk mengidentifikasi tipe variabel. Sebagai contoh, X% adalah Variabel X bertipe integer, X! berisikan bilangan floating-point dan X\$ adalah untuk string. Sebenarnya ini hanyalah masalah gagasan apakah nama ini lebih baik daripada prefiks Hungaria. Satu teknik lagi adalah dengan menggunakan Prefiks standar yang telah kita pergunakan seperti menamai control atau objek, contoh str untuk String, int untuk Integer, dan seterusnya. Ada satu lagi yang harus kita perhatikan, yaitu menambahkan satu character lagi sebelum prefiks tersebut untuk menandai tempat atau "scope" variabel tersebut, contoh **mStrNama** adalah variabel Nama yang bertipe string dan mempunyai scope didalam sebuah Module (bisa itu Form ataupun Class Module). Sedangkan gStrNama bersifat Global atau Public. Dan untuk variabel local tidak usah ditambahi prefiks lagi.

Berikut ini adalah tabel beberapa cara penamaan variabel, kita dapat memilihnya sesuai dengan selera pemrograman masing-masing, tetapi yang harus diingat adalah "konsistensi"...

Tabel. 1: Sufiks Penamaan Variabel

Sufiks	Tipe Variabel
%	Integer 2 byte
&	Integer 4 byte
@	Currency
!	single
#	double
\$	String
Tidak ada	Boolean
Tidak ada	Byte
Tidak ada	Date
Tidak ada	Variant

Tabel. 2: Prefiks Notasi Hungaria

Sufiks	Tipe Variabel
--------	---------------

a	Array
b	Boolean
c	Currency
d	Double
f	Float/Single
h	Handle
i	Index
l	Long
n	Integer
s	String
vnt	Variant
w	Word

Tabel. 3: Prefiks Standar Penamaan Variabel

Sufiks	Tipe Variabel
int	Integer
wrđ	Word
cur	Currency
sgl	single
dbl	double
str	String
bool	Boolean
byt	Byte
dt	Date
vnt	Variant

Deklerasi Variabel

Sebuah teknik yang amat penting untuk mencatat nama variabel adalah dengan menggunakan pernyataan **Option Explicit**. Dengan meletakkan pernyataan ini pada awal kode module atau form, maka setiap akan menggunakan variabel kita harus mendeklarasikannya terlebih dahulu. Dengan cara ini maka nama variabel akan mudah untuk dibaca dan yang lebih penting lagi adalah untuk menghindari kesalahan program yang akan terjadi akibat variabel yang kita gunakan tidak dikenali oleh Compiler.

Satu lagi hal yang harus diperhatikan, khususnya bagi yang sudah menggunakan bahasa C sebelumnya, bahwa mendeklarkan beberapa variabel dalam satu baris/pernyataan tidak akan menghasilkan tipe variabel yang sama...

Contoh:

```
Dim intA, intB, intC As Integer
```

Pernyataan diatas tidak akan menghasilkan semua variabel bertipe integer, tetapi hanya variabel intC saja yang bertipe Integer!!!

Kecuali contoh berikut yang akan menghasilkan tipe variabel yang kita harapkan:

```
Dim intA As Integer, intB As Integer, intC As Integer
Dim a%, b!, c#
```

Bekerja dengan Variabel "Date"

Variabel Tanggal dan Jam (Date) disimpan dalam bentuk IEEE 64-bit (8-byte) floating-point numbers yang mewakili tanggal antara 1 January 100 s.d 31 December 9999 dan Jam dari 0:00:00 to 23:59:59.

pemuatan/inisialisasi variabel date harus diapit oleh tanda (#), contohnya: #January 1, 1993# atau #1 Jan 93# atau #11/17/96 6:12:33 PM#.

Date Function

mengembalikan nilai Tanggal dan Jam (dalam bentuk variable **Variant**) dari sistem.

Contoh:

```
Dim MyDate
MyDate = Date ' MyDate contains the current system date.
```

Date Statement

Men-set tanggal sistem dengan nilai yang kita masukkan

Tanggal yang valid adalah dari January 1, 1980 s.d December 31, 2099. Untuk Microsoft Windows NT, dari January 1, 1980 s.d December 31, 2079.

Contoh:

```
Date = #04/01/2001#
```

DateSerial dan DateValue Function (TimeSerial dan TimeValue)

DateSerial = untuk mengkombinasikan baik tanggal maupun jam ke variabel date (Tahun, Bulan, Tanggal)

DateValue = untuk memuat variabel Date dari suatu representasi string.

Contoh:

```
D = DateSerial(2000, 12, 31)
atau bisa digunakan untuk kalkulasi:
D = DateSerial(2000 - 10, 12 - 10, 1 - 1) 'menghasilkan = 31 September 1990
T = TimeSerial(23, 59, 00)
```

```
D = DateValue("11/17/2000")
T = TimeValue("18:20:59")
```

```
DT = DateValue("11/17/2000") + TimeValue("18:20:59")
```

Formatting Date and Time

Dim D As Date

```
D = #04/29/2001 12:30:59 PM#
```

```
Print Format(D, "General Date") '04/29/2001 12:30:59 PM
Print Format(D, "Long Date") 'Sunday, April 29, 2001
Print Format(D, "Medium Date") '29-Apr-2001
Print Format(D, "Short Date") '04/29/2001
Print Format(D, "Long Time") '12:30:59 PM
Print Format(D, "Medium Time") '12:30 PM
Print Format(D, "Short Time") '12:30
```

Selain format bernama, kita dapat membuat format ciptaan sendiri:

```
D$ = Format(D, "dd/mm/yyyy hh:mm AM/PM") '29/04/2001 12:30 PM
M$ = Format(D, "mmm") 'April
'Lihat di MSDN untuk bentuk format yang lain
```

Mengekstraksi Rincian Variabel Date

Dim D As Date

```
D = #04/29/2001 12:30:59 PM#
```

```
Print Month(D) '4
Print Day(D) '29
Print Year(D) '2001
```

```
Print Hour(D)      '12
Print Minute(D)   '30
Print Second(D)   '59
Print WeekDay(D)  '1 = vbSunday
```

DateDiff Function

Mengembalikan nilai antara dua interval tanggal (Date)

Syntax:

DateDiff(*interval*, *date1*, *date2*[, *firstdayofweek*[, *firstweekofyear*]])

Settings:

interval :

yyyy	year
q	quarter
m	month
y	Day of year
d	Day
w	Weekday
ww	Week
h	Hour
n	Minute
s	Second

Contoh:

Dim D As Date

D = #4/29/2001 12:00:00 PM#

Print DateDiff("d", D, #4/30/2001#) '1 (hari)

Print DateDiff("m", D, #3/30/2001#) '-1 (bulan)

Print DateDiff("yyyy", D, #4/30/2010#) '9 (tahun)

DateAdd Function

Menambahkan variabel tanggal (Date)

Syntax:

DateAdd(*interval*, *number*, *date*)

Contoh:

Print DateAdd("d", 1, #4/29/2001#) '30 April 2001

Print DateAdd("m", 1, #4/29/2001#) '29 Mei 2001

DatePart Function

Mengambil nilai yang terdapat didalam variabel Tanggal (Date)

Syntax:

DatePart(*interval*, *date*[, *firstdayofweek*[, *firstweekofyear*]])

Contoh:

Print DatePart("q", #4/29/2001#) '2 (quarter ke-2)

Print DatePart("m", #4/29/2001#) '4 (bulan ke-4)

```
Print DatePart("yyyy", #4/29/2001#) '2001 (tahun ke-2001)
```

STRING MANIPULATION part #1

Pada bahasan kali ini kita akan mencoba untuk memahami syntax-syntax manipulasi terhadap variabel 'string' yang terdapat didalam Visual Basic. Berikut ini adalah beberapa contoh penggunaan fungsi-fungsi tersebut:

ASC

mengembalikan nilai character yang terdapat didalam kode ASCII.

```
Me.Print "ASC("A") = " & Asc("A") '---> = 65
Me.Print "ASC("B") = " & Asc("B") '---> = 66
Me.Print "ASC("a") = " & Asc("a") '---> = 97
Me.Print "ASC("1") = " & Asc("1") '---> = 49
Me.Print "ASC("P") = " & Asc("P") '---> = 222
```

CHR / CHR\$

Mengembalikan character yang sesuai dengan kode ASCII yang diminta. Perbedaan penggunaan CHR dan CHR\$ adalah nilai yang dikembalikan oleh fungsi ini adalah Variant dan String.

Tips: Selalu gunakan CHR\$ karena menggunakan tipe variabel 'string' yang lebih hemat memory dibandingkan 'variant'.

Contoh:

```
Me.Cls
Me.Print "CHR$(65) = " & Chr$(65) '---> = A
Me.Print "CHR$(66) = " & Chr$(66) '---> = B
Me.Print "CHR$(97) = " & Chr$(97) '---> = a
Me.Print "CHR$(44) = " & Chr$(49) '---> = 1
Me.Print "CHR$(200) = " & Chr$(200) '---> = E
```

Format / Format\$

Memformat suatu string sesuai keinginan.

Contoh:

```
Me.Print "Format$(50, "Currency") " & Format$(50, "Currency") '---> = $50.00
Me.Print "Format$(1250, "Currency") " & Format$(1250, "Currency") '---> =
$1,250.00
Me.Print
Me.Print "Format$(250, "###,##0") " & Format$(250, "###,##0") '---> = 250.00
Me.Print "Format$(1250.25, "###,##0.00") " & Format$(1250.25, "###,##.##") '--->
= 1,250.00
Me.Print
Me.Print "Format$(0, "Yes/No") " & Format$(0, "Yes/No") '---> = No
Me.Print "Format$(1, "Yes/No") " & Format$(1, "Yes/No") '---> = Yes
Me.Print "Format$(0, "True/False") " & Format$(0, "True/False") '---> = False
Me.Print "Format$(1, "True/False") " & Format$(1, "True/False") '---> = TRUE
Me.Print "Format$(0, "On/Off") " & Format$(0, "On/Off") '---> = OFF
Me.Print "Format$(1, "On/Off") " & Format$(1, "On/Off") '---> = ON
Me.Print
Me.Print "Format$("saya", ">") = " & Format("saya", ">") '---> = SAYA
Me.Print "Format$("KAMU", "<") = " & Format("KAMU", "<") '---> = kamu
Me.Print
Me.Print "Format$(1, "000") = " & Format(1, "000") '---> = 001
```

INSTR

Mencari tahu nilai/posisi suatu character/kumpulan character didalam suatu string.

```

Dim strName As String
strName = "Bill Gates"
Me.Print "strName = " & strName
Me.Print "pos --> " & "1234567890"
Me.Print
Me.Print "InStr(strName, "G") = " & InStr(strName, "G")           '--> 6
Me.Print "InStr(strName, "Gates") = " & InStr(strName, "Gates")   '--> 6
Me.Print "InStr(strName, "Clinton") = " & InStr(strName, "Clinton") '--> 0

```

INSTREV

Mencari tahu nilai/posisi suatu character/kumpulan character didalam suatu string tetapi dari yang paling ujung/kanan.

```
Dim strValue As String
```

```

strValue = "d:\train\database\employee.mdb"
Me.Print "strValue = " & strValue
Me.Print "pos --> " & "12345678901234567890"
Me.Print
Me.Print "InStrRev(strValue, "\"") = " & InStrRev(strValue, "\"")   '--> 18
Me.Print "Mid$(strValue, InstrRev(strValue, "\"")+1) = " & Mid$(strValue, InstrRev
(strValue, "\"") + 1)      '--> employee.mdb
Me.Print "InStrRev(strValue, ":",") = " & InStr(strValue, ":",")    '--> 3

```

JOIN

Menggabungkan isi dari suatu array.

```

Dim arrA(0 To 5) As String
arrA(0) = "B"
arrA(1) = "a"
arrA(2) = "s"
arrA(3) = "i"
arrA(4) = "c"
arrA(5) = "a"
Me.Cls
Me.Print "Join(arrA, "") = " & Join(arrA, "")           '--> Basica
Me.Print "Join(arrA, "-") = " & Join(arrA, "-")        '--> B-a-s-i-c-a
Me.Print "Join(arrA, vbTab) = " & Join(arrA, vbTab)     '--> B a s i c a

```

LCASE / LCASE\$

Menjadikan suatu string Lower Case (huruf kecil semua).

```
Dim strName As String
```

```

strName = "Bill Gates"
Me.Cls
Me.Print "strName = " & strName
Me.Print "LCase$(strName) = " & LCase$(strName)        '--> bill gates

```

UCASE / UCASE\$

Menjadikan suatu string Upper Case (huruf besar/kapital semua).

```
Dim strName As String
```

```

strName = "Bill Gates"
Me.Cls
Me.Print "strName = " & strName

```

```
Me.Print "UCase$(strName) = " & UCase$(strName) '--> BILL GATES
```

LEFT / LEFT\$

Mengambil character dari kiri suatu string.

```
Dim strName As String
```

```
strName = "Bill Gates"
Me.Cls
Me.Print "strName = " & strName
Me.Print "pos --> " & "1234567890"
Me.Print
Me.Print "Left$(strName, 4) = " & Left$(strName, 4) '--> Bill
Me.Print "Left$(strName, 7) = " & Left$(strName, 7) '--> Bill Ga
```

RIGHT / RIGHT\$

Mengambil character dari kanan suatu string.

```
Dim strName As String
```

```
strName = "Bill Gates"
Me.Cls
Me.Print "strName = " & strName
Me.Print "pos --> " & "1234567890"
Me.Print
Me.Print "Right$(strName, 7) = " & Right$(strName, 7) '--> l Gates
Me.Print "Right$(strName, 10) = " & Right$(strName, 10) '--> Bill Gates
```

MID / MID\$

Mengambil character dari posisi yang kita tentukan didalam suatu string.

```
Dim strName As String
```

```
strName = "Bill Gates"
Me.Cls
Me.Print "strName = " & strName
Me.Print "pos --> " & "1234567890"
Me.Print
Me.Print "Mid$(strName, 1, 4) = " & Mid$(strName, 1, 4) '--> Bill
Me.Print "Mid$(strName, 6, 2) = " & Mid$(strName, 6, 2) '--> Ga
Me.Print "Mid$(strName, 6) = " & Mid$(strName, 6) '--> Gates
```

LEN

Menghitung jumlah character suatu string.

```
Dim strName As String
```

```
strName = "Bill Gates"
Me.Cls
Me.Print "Len(strName) = " & Len(strName) '--> 10
```

LTRIM / LTRIM\$

Menghilangkan character spasi yang terdapat pada awal string.

```
Dim strName As String
```

```
strName = " Bill Gates"
```

```
Me.Cls
Me.Print "strName = " & strName
Me.Print
Me.Print "LTRIM$(strName) = " & LTrim$(strName)    '--> Bill Gates (tanpa spasi)
```

RTRIM / RTRIM\$

Menghilangkan character spasi yang terdapat pada akhir string.

```
Dim strName As String

strName = "Bill Gates  "
Me.Cls
Me.Print "strName = " & strName
Me.Print
Me.Print "RTRIM$(strName) = " & RTrim$(strName)    '--> Bill Gates (tanpa spasi)
```

REPLACE

Mengganti nilai/isi suatu string dengan character yang kita inginkan.

```
Dim strValue As String

strValue = "Microsoft Visual Basic 5.0"
Me.Cls
Me.Print "strValue = " & strValue
MsgBox "Replace(strValue, ""5.0"", ""6.0"") = " & Replace(strValue, "5.0", "6.0")
```