



CMC 3.5

Presented for You

Completed With Source and Tool

- *Inline Assembly Power Basic
- *Kalkulasi Ceksum Header PE
- *Merubah Entry-Point PE
- *Membuat std-Dll
- *RTP dengan API
- *Belajar Assembly
- *Hack Password Archive
- *DLL

CN Magazine
codenesia
Build Indonesia With Code

#2

Antivirus | Virus | Hacking | Crack | Etc

CN-Zine Vol. #2

26 April 2010

19:06:20 WIB

Segala isi materi dan tutorial di-dalam majalah elektronik ini adalah hak cipta dan tanggung jawab masing-masing penulis. Anda diizinkan untuk mempublikasi ulang tanpa se-izin dari masing-masing penulis dengan tanpa merubah nama dan atribut penulis.

Ide dan Desain Cover :

Sonny Lazuardi

Layouter :

Rizka Anhar Pramudita

Editor :

A.M Hirin



Copyright @ 2010 - Codenesia

www.codenesia.com

PEMBUKA

Setelah melakukan uji coba peluncuran Ezine edisi pertama dari Codenesia yang dinamakan CNZine yang ternyata mendapat respon positif dari teman-teman baik dari dalam maupun luar komunitas. Untuk itu kami selaku team redaksi majalah elektronik Codenesia telah berhasil menyelesaikan edisi ke-2 dari CNZine dengan perubahan yang signifikan dari segi isi materi dan tutorial. Pada CNZine 2 tutorial dan artikel yang ada tidak seperti pada edisi sebelumnya yang hanya merupakan kumpulan artikel yang pernah diposting di web www.codenesia.com, namun kami secara khusus mengumpulkan tutorial yang sebagian besar belum pernah diposting pada web codenesia maupun web lainnya. Bahkan beberapa artikel merupakan artikel khusus yang sengaja dipesan oleh teman-teman melalui redaksi CNZine karena topik tersebut sedang hangat.

Kami selaku team redaksi CNZine berharap majalah elektronik codenesia edisi 2 dapat lebih bermanfaat untuk meningkatkan ilmu pengetahuan ddalam bidang Teknologi Informatika baik bagi komunitas codenesia mupun komunitas IT lainnya di Indonesia.

Dalam Ezine ini juga dilengkapi file pelengkap berupa tool yang dipakai untuk praktik materi serta source code yang ada dalam materi CNZine ini sehingga kami juga berharap CNZine ini juga dapat dinikmati oleh pemula sekalipun dengan tanpa kesulitan yang berarti.

Salam Maniz

Team Redaksi CNZine



Zine



DAFTAR ISI

Pembuka	3
Daftar Isi	4

[Tutorial]

Inline Asm dengan Power Basic (Mey Shimaro)	5
Kalkulasi Ceksum dengan Header PE (Black Cyber)	8
Merubah Entry-Point Program (A.M. Hirin)	15
Membuat Dll dengan Power Basic (HrXxX)	22
RTP Antivirus dengan API (Agus)	27
Belajar Assembly dengan MASM (Bidan Malware)	31
Hack Password Archive (Anharku)	36
Membuat Personas Firefox Sendiri (Anharku)	41
Trick Menghilangkan MyRecent Document (Anharku)	44
Command Prompt X (Gxry)	48
Folder Quarantine (Gxry)	51
Caesar Chiper C++ Mode (Gxry)	56

[Tutorial]

Produk	58
Redaksi	59
Link Download External	60

Inline Asm dengan Power Basic

by : Mey Shimaro

Pernahkah anda mengkombinasikan dua bahasa pemrograman dalam aplikasi anda? Mungkin sebagian anda sudah pernah melakukannya, terlebih-lebih banyak bahasa pemrograman yang secara default mendukung Inline Asm, namun ada juga bahasa pemrograman yang secara default tidak mendukung Inline Asm (menyisipkan bahasa assembly dalam script program), misalnya Visual Basic atau bahasa yang paling digemari oleh kebanyakan kalangan programmer saat ini, yang mana kita harus memakai alat tambahan pada visual basic agar dapat mendukung Inline Asm. Karena inline assembly tidak terintegrasi secara internal maka compiler antara bahasa internal program dan assembly akan terpisah, sehingga cukup merepotkan pemakainya jika ingin menyisipkan bahasa assembly pada program yang dibuatnya.

Namun anda tidak perlu khawatir karena tool sejenis Visual basic yang juga berbasis bahasa pemrograman Basic yaitu Power Basic telah mendukung Inline Asm secara default. Bahasa pada Power Basic hampir sama dengan bahasa pemrogramannya yang ada pada Visual Basic, untuk itu anda tidak perlu khawatir takut belajar Power Basic ini. Bagi anda yang penasaran bagaimana cara mengkombinasikan bahasa pemrograman (Basic) dengan Assembly, yuk mari kita belajar bersama. Bukalah Editor Power Basic (link download ada pada bagian akhir majalah ini) anda dan tuliskanlah code seperti berikut :

```
#COMPILE EXE "Asm.exe"
#DIM ALL

FUNCTION PBMAIN () AS LONG
    MSGBOX STR$(AsmLen("M31 Shimaro")) ' dengan fungsi Len buatan kita
    MSGBOX STR$(LEN("M31 Shimaro"))   ' fungsi Len default
END FUNCTION

FUNCTION AsmLen(BYVAL szToLen AS STRING) AS LONG
    !mov esi, szToLen
    !mov eax, [esi-4]
    !mov Function, eax
END FUNCTION
```

Lihatlah code di atas dengan seksama, fokuskan perhatian anda pada fungsi / scrip berikut :

```
FUNCTION AsmLen(BYVAL szToLen AS STRING) AS LONG
    !mov esi, szToLen
    !mov eax, [esi-4]
    !mov Function, eax
END FUNCTION
```

Script di atas mengandung bahasa Assembly, dimana sebelum menuliskan intruksinya kita memakai symbol ! (tanda seru) yang merupakan symbol sepesial pada Power Basic untuk mengintruksikan bahwa baris itu dapat ditulis oleh bahasa Asembly sesuai compiler Power Basic. Pada kasus di atas kita menulis code berupa fungsi bernama **AsmLen** dengan satu parameter bernama **szToLen** dengan nilai balik fungsi bertipe **Long**. Lalu perhatikan Scrip Assembly (isi fungsi) perbaris,

```
!mov esi, szToLen
```

Maksud baris tersebut adalah kita memindah (copy) isi variable bernama **szToLen** yang merupakan parameter dari fungsi ke Register ESI (register esi (source index) digunakan untuk penanganan operasi sumber string)

```
!mov eax, [esi-4]
```

Maksud baris di atas adalah memindah (copy) ukuran string yang sudah dipindahkan ke register esi ke register yang biasa menangani kegiatan matematis (EAX), perlu diketahui bahwa ukuran string yang dipindah ke register esi berada pada 4 byte sebelum posisi register source Index (esi) sehingga kita memindahkan (copy) 4 byte dibelakang **ESI** ke register **EAX**, sehingga EAX berisi ukuran dari string yang sudah kita pindahkan ke **ESI**.

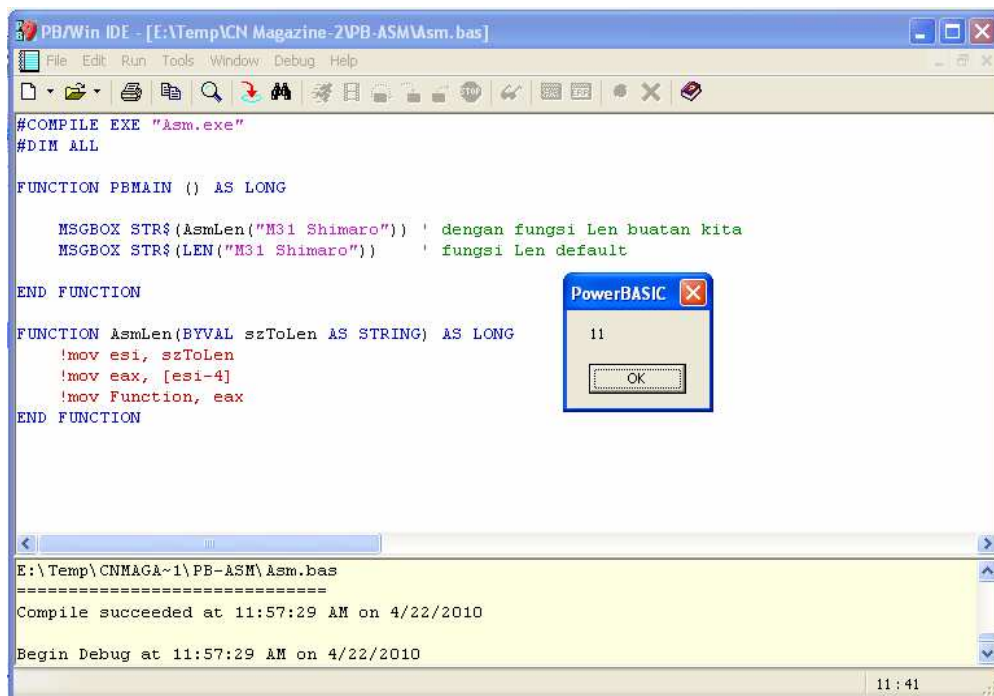
```
!mov Function, eax
```

Maksud baris di atas adalah memindah (copy) nilai yang ada pda register eax (ukuran string) ke Fungsi bernama **AsmLen**. Sehingga nilai balik fungsi adalah ukuran dari string yang dimasukan pada parameternya.

Nah untuk mengujinya kita harus membnadngkan hasil dari Fungsi **LEN** buatan kita ini (AsmLen) dengan fungsi Len sebenarnya yaitu dengan 2 baris script berikut:


```
MSGBOX STR$(AsmLen("M31 Shimaro")) 'dengan fungsi Len buatan kita
MSGBOX STR$(LEN("M31 Shimaro"))    ' fungsi Len default
```

Script baris pertama kita memanggil fungsi **AsmLen** dengan nilai yang diuji pada parameternya adalah “M31 Shimaro” yang terdapat 11 karakter, dan Script baris kedua kita memanggil fungsi **LEN** yang merupakan fungsi default dari PowerBasic untuk menghitung karakter yang diujikan pada parameternya. Untuk melihat hasilnya maka tekanlah **F5** untuk menjalankan program yang baru kita buat ini pada Power Basic. Maka hasilnya akan menampilkan kotak pesan dengan isi pesan “11” sebanyak 2x.



Nah bagi anda yang belum pernah sama sekali belajar Asm, ternyata mudah juga kan mengkombinasikan dua bahasa dengan Assembly. ☺ Dengan menguasai Inline Asm maka banyak hal luar biasa yang dapat kita lakukan dengan aplikasi yang kita buat, karena Assembly sendiri bahasa yang paling dasar yang ada saat ini sehingga eksekusi di memory pun lebih cepat jika ditulis secara efektif.

Kalkulasi Ceksum dengan Header PE

by : Black Cyber

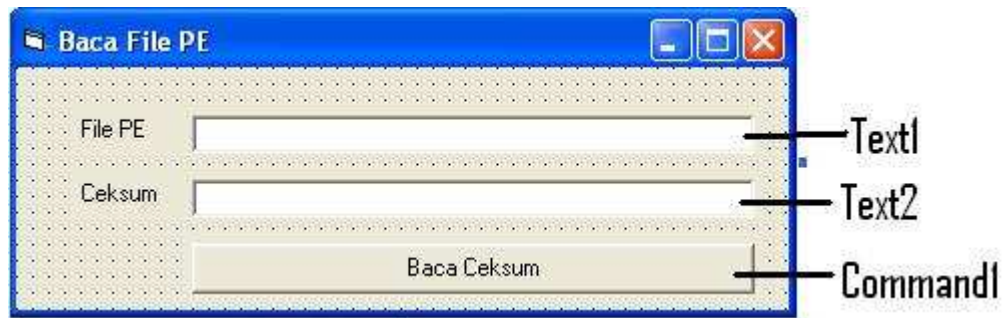
Tentunya sebagian besar dari anda sudah mengenal file PE atau biasa kita sebut dengan program. File PE sendiri mempunyai struktur data yang tersusun dan tertata rapi baik secara fisik maupun virtual (di memory) sehingga dapat kita baca data-data nya dengan baik. Bagi anda seorang coder Antivirus maka memahami struktur PE sangatlah penting, karena musuh dan teman yang akan anda hadapi sendiri adalah file-file program yang tentunya tergolong jenis file PE (Portable Executable). File PE ini sendiri memiliki keunikan data yang dapat membedakan (dijadikan penanda) dari file PE satu ke file PE lainnya. Kali ini saya akan membahas bagaimana cara membuat kalkulasi ceksum dengan memanfaatkan data dari Header PE yang cukup variatif sehingga kita dapat membuat penanda atau ceksum untuk virus dan hal lain sejenisnya. Sebenarnya hal ini kurang efektif bagi anda yang sudah ahli mengenai file PE karena bisa saja menimbulkan kesamaan kalkulasi ceksum dari file PE yang berbeda namun kemungkinannya sangat kecil. Disini saya memanfaatkan data-data yang kiranya setiap PE akan memiliki perbedaan yang nyata. Adapun data yang akan kita kalkulasi adalah sebagai berikut :

- **TimeStamp** : Data tanggal kompilasi PE
- **Characteristics pada Image File Header** : Pembeda antara PE exe, dll, dan sys
- **AddressOfEntryPoint** : Relative Virtual Address dari file PE

Ketiga data yang cukup variatif di atas lah yang akan kita satukan untuk menjadi penanda file PE sehingga menghasilkan nilai ceksum yang cukup variatif dan cepat. Dan perlu kita ketahui bahwa ceksum ini hanya bekerja pada file berjenis PE saja yaitu exe (termasuk scr. pif. com), dll, sys. Untuk itu mari kita buat programnya dengan visual basic.

Pembuatan Program :

1. Desainlah form dengan penampakan sebagai berikut:



2. Tambahkan sebuah module bernama ModPE dan tuliskan codenya sebagai berikut :

```
Public Const IMAGE_NUMBEROF_DIRECTORY_ENTRIES = 16
Public Type IMAGE_DOS_HEADER
    e_magic           As Integer
    e_cblp            As Integer
    e_cp              As Integer
    e_crlc            As Integer
    e_cparhdr         As Integer
    e_minalloc        As Integer
    e_maxalloc        As Integer
    e_ss              As Integer
    e_sp              As Integer
    e_csum            As Integer
    e_ip              As Integer
    e_cs              As Integer
    e_lfarlc          As Integer
    e_ovno            As Integer
    e_res(0 To 3)     As Integer
    e_oemid           As Integer
    e_oeminfo         As Integer
    e_res2(0 To 9)    As Integer
    e_lfanew          As Long
End Type
```

```
Public Type IMAGE_NT_HEADERS
    SignatureLow           As Integer
    SignatureHigh          As Integer
End Type
```

```
Public Type IMAGE_FILE_HEADER
    Machine                As Integer
    NumberOfSections        As Integer
    TimeDateStamp           As Long
    PointerToSymbolTable    As Long
    NumberOfSymbols         As Long
    SizeOfOptionalHeader    As Integer
    Characteristics        As Integer
End Type
```

```
Public Type LARGE_INTEGER
    LowPart                As Long
    HighPart               As Long
End Type
```

```
Public Type IMAGE_DATA_DIRECTORY_32
    VirtualAddress         As Long
    nSize                  As Long
End Type
```

```
Public Type IMAGE_DATA_DIRECTORY_64
    VirtualAddress         As Long
    nSize                  As Long
End Type
```

Public Type IMAGE_OPTIONAL_HEADER_32

Magic	As Integer
MajorLinkerVersion	As Byte
MinorLinkerVersion	As Byte
SizeOfCode	As Long
SizeOfInitializedData	As Long
SizeOfUninitializedData	As Long
AddressOfEntryPoint	As Long
BaseOfCode	As Long
BaseOfData	As Long
ImageBase	As Long
SectionAlignment	As Long
FileAlignment	As Long
MajorOperatingSystemVersion	As Integer
MinorOperatingSystemVersion	As Integer
MajorImageVersion	As Integer
MinorImageVersion	As Integer
MajorSubsystemVersion	As Integer
MinorSubsystemVersion	As Integer
Win32VersionValue	As Long
SizeOfImage	As Long
SizeOfHeaders	As Long
Checksum	As Long
Subsystem	As Integer
DllCharacteristics	As Integer
SizeOfStackReserve	As Long
SizeOfStackCommit	As Long
SizeOfHeapReserve	As Long
SizeOfHeapCommit	As Long
LoaderFlags	As Long
NumberOfRvaAndSizes	As Long

DataDirectory(0 To IMAGE_NUMBEROF_DIRECTORY_ENTRIES - 1) As _

IMAGE_DATA_DIRECTORY_32

End Type

```
Public Type IMAGE_SECTION_HEADER
    SectionName(0 To 7)      As Byte
    VirtualSize              As Long
    VirtualAddress           As Long
    SizeOfRawData            As Long
    PointerToRawData         As Long
    PointerToRelocations     As Long
    PointerToLinenumbers     As Long
    NumberOfRelocations     As Integer
    NumberOfLinenumbers     As Integer
    Characteristics          As Long
End Type
```

3. Lalu ketikkan code pada jendela code Form Sebagai berikut:

```
Private Declare Sub RtlMoveMemory Lib "ntdll.dll" (ByVal pDestBuffer As Long,
ByVal pSourceBuffer As Long, ByVal nBufferLengthToMove As Long)
Private Declare Sub RtlFillMemory Lib "ntdll.dll" (ByVal pDestBuffer As Long,
ByVal nDestLengthToFill As Long, ByVal nByteNumber As Long)
Private Declare Sub RtlZeroMemory Lib "ntdll.dll" (ByVal pDestBuffer As Long,
ByVal nDestLengthToFillWithZeroBytes As Long)

Private Type NT_HEADER
    INTSIGN    As IMAGE_NT_HEADERS
    IFILEH     As IMAGE_FILE_HEADER
    IOPTH32    As IMAGE_OPTIONAL_HEADER_32
End Type

Private Function ReadFile(ByRef szFileTarget As String, ByVal nStart As Long,
ByRef OutData() As Byte, ByVal PanjangData As Long)
    ReDim OutData(PanjangData) As Byte
    Open szFileTarget For Binary As 1
        Get #1, nStart, OutData
    Close #1
End Function
```

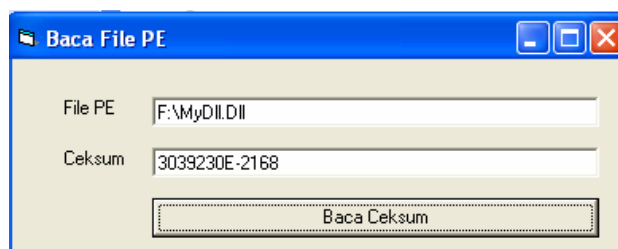
```

Private Sub Command1_Click()
    Dim IDOSH      As IMAGE_DOS_HEADER
    Dim NTHPE      As NT_HEADER
    Dim DataOut()  As Byte
    Dim NewHeader  As Long
    Dim nSection   As Long
    Dim iCount     As Long
    Dim MyCeksum   As String
    ' Image DOS Header
    Call ReadFile(Text1.Text, 1, DataOut, 64)
    Call RtlMoveMemory(VarPtr(IDOSH), VarPtr(DataOut(0)), Len(IDOSH))
    '----- Baca Struktur Image Dos Header (diambil 2 saja)
    NewHeader = IDOSH.e_lfanew ' posisi header baru (PE File Header)
    Call ReadFile(Text1.Text, NewHeader + 1, DataOut, Len(NTHPE))
    Call RtlMoveMemory(VarPtr(NTHPE), VarPtr(DataOut(0)), Len(NTHPE))
    MyCeksum = Hex$(NTHPE.IFILEH.TimeDateStamp)
    MyCeksum = MyCeksum & Hex$(NTHPE.IFILEH.Characteristics)
    MyCeksum = MyCeksum & "-" & Hex$(NTHPE.IOPTH32.AddressOfEntryPoint)
    Text2.Text = MyCeksum
End Sub

```

4. Selesai

Sekarang jalankan programnya dan uji untuk melakukan kalkulasi ceksum dari suatu file PE (exe, dll, sys). Masukkan alamat file PE yang akan di ceksum pada textbox bernama Text1. Pada kasus ini adalah MyDll.dll yang terletak pada driver F:\.



Nilai ceksum dari File PE akan muncul di textbox bernama text2, pada kasus ini nilai ceksum dari file PE bernama MyDll.dll adalah **3039230E-2168**, anda bisa mengujinya sendiri dengan file PE lainnya.

Kesimpulan:

Artikel ini hanyalah contoh sederhana cara membuat ceksum yang lebih systematis dengan memanfaatkan bagian-bagian dari Header PE yang kemungkinan besar akan bernilai variatif sehingga dapat kita pakai sebagai pembeda antara file PE satu dengan lainnya secara sederhana yang mungkin bisa anda pakai untuk keperluan lainya seperti ceksum untuk Antivirus. Namun untuk keperluan yang lebih urgent untuk Antivirus anda perlu mempertimbangkan dan menambah berbagai data sehingga ceksum yang anda pakai benar-benar variatif agar tidak terjadi kesamaan nilai antara dua file PE yang berbeda, mungkin anda bisa memanfaatkan data section dari file PE untuk variasi dan akurasi ceksum yang lebih baik. Selamat berkreasi ☺

Merubah Entry-Point Program

by : AM. Hirin

Mungkin sebagian dari anda telah memahami apa itu Entry-Point Program (File PE), namun ada baiknya saya bahas sedikit penjelasan mengenai Entry-Point. Entry-Point sangat erat kaitanya dengan suatu program, karena Entry-Point adalah bagian dari code yang terletak pada suatu section yang pertama kali di-eksekusi oleh Loader ketika file PE (program) dijalankan. Sebagian besar program yang terinfeksi oleh virus bagian pada Entry-Point akan berubah sesuai entry point infector sehingga pada beberapa kasus kita dapat membedakan virus penginfeksi program dengan melihat entry-point file yang terinfeksi, namun ada juga virus yang cara menginfeksinya sama sekali tidak merubah Entry-Point program target (contoh Win32/Polip). Disini saya tidak akan membahas jauh masalah entry point dan penerapannya untuk membedakan virus penginfeksi suatu file PE, saya hanya akan membahas bagaimana cara merubah posisi entry-point suatu program sehingga informasi suatu program entry-point berubah, teknik yang akan saya pakai disini adalah mencari code cave (byte kosong) yang ada pada suatu section dimana entry-point berada lalu menuliskan intruksi tertentu untuk mengeksekusi entry-point lama,. Sebelum mempraktikan caranya maka yang perlu anda siapkan adalah sebagai berikut:

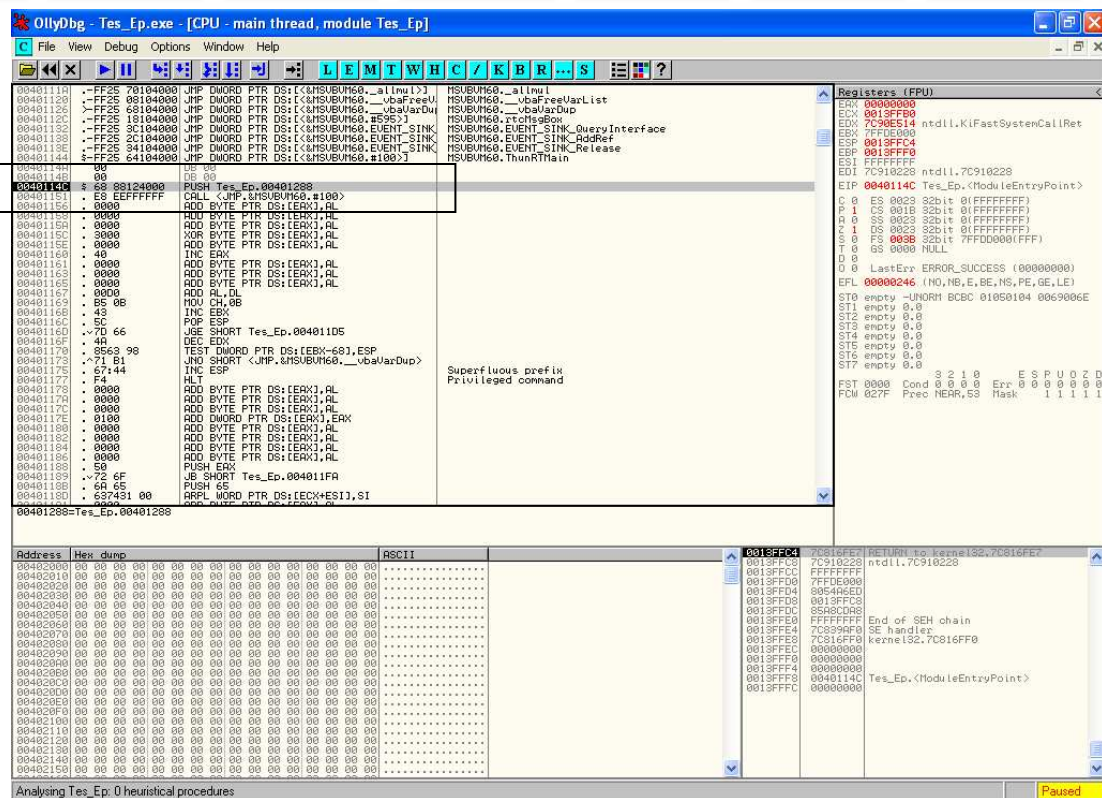
- **Program target**

Saya memakai program bernama Tes_ep.exe yang bisa anda ambil pada file **Source_dan_Tool_CNZINE2.zip**

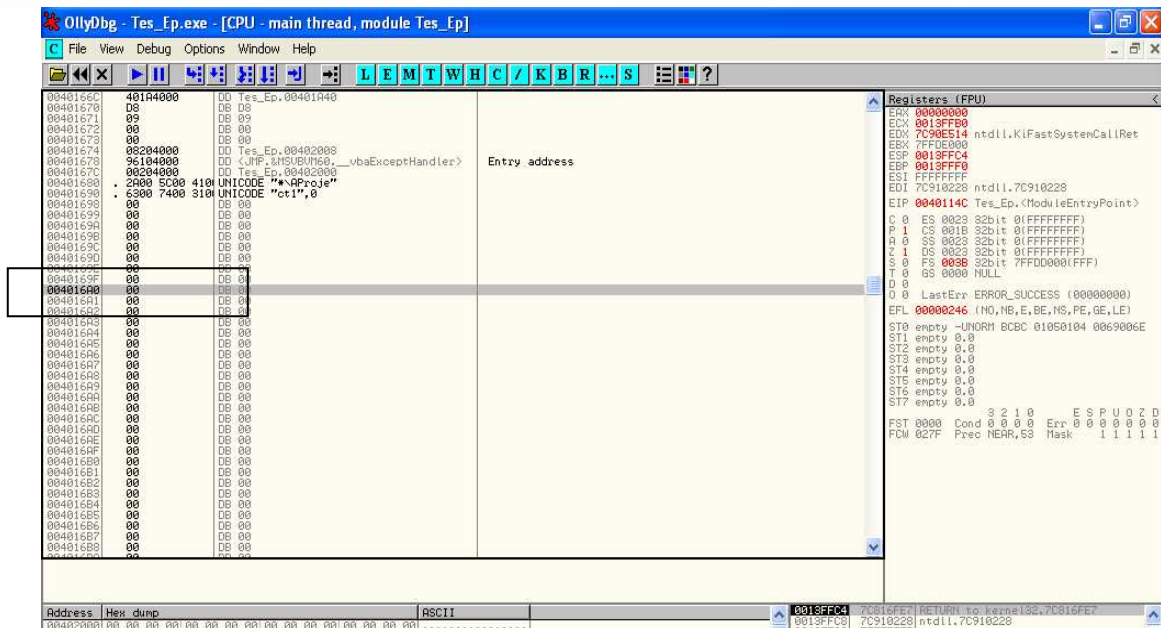
- **Tool Bantu**

Saya memakai OllyDbg v.110 dan LordPE Deluxe yang juga bisa anda ambil pada file **Source_dan_Tool_CNZINE2.zip**

Setelah anda siapkan semuanya, mari kita praktikan caranya. Jalankanlah program OllyDbg lalu bukalah file program target (Tes_ep.exe) dengan program tersebut sehingga tampilanya sebagai berikut:



Perhatikanlah baris yang ter-highlight, disitulah letak entry-point program bernama Tes_ep.exe berada dimana alamat virtualnya (VA) adalah “0040114C” dengan instruksi **PUSH 00401288**. Yang perlu kita ingat adalah nilai “0040114C” yang merupakan nilai entry-point program asli (OEP atau Original Entry Point) nah sekarang kita akan mencari dimana kita akan memindahkan entry-point, jika pada virus ini dikenal dengan istilah VEP (Virus Entry Point). Marilah kita mencari code cave atau byte kosong pada section, kita cari code cave terpanjang yang sudah tidak mengandung instruksi lagi dibawahnya. Untuk itu geserlah (scroll bar) kebawah hingga kita menemukan daerah kosong dari instruksi. Disini saya memilih alamat “004016A0” karena dibawahnya sudah tidak ada lagi intruksi, atau hal ini sering disebut section-alignment sehingga memungkinkan suatu PE berisi byte kosong pada bagian terakhir section karena ukurannya mengikuti pembulatan nilai section-alignment (pada memory) dan file-alignment (pada file fisik). Berikut adalah gambar dimana kita telah berada pada posisi beralamat “004016A0”



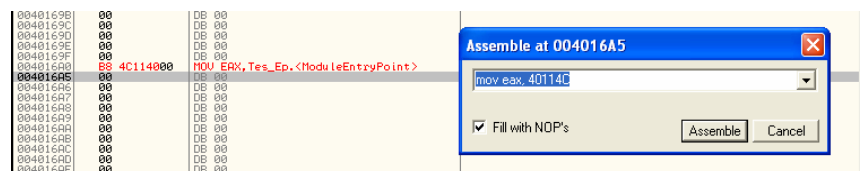
Nah alamat **“004016A0”** ini yang akan kita jadikan sebagai entry-point baru program, tentunya suatu entry-point harus berisi intruksi-intruksi mesin atau rutin, untuk itu kita harus menuliskan intruksi agar nantinya program tidak rusak dan sekan-akan masih utuh seperti semula tanpa perubahan (seperti yang dilakukan virus). Untuk menuliskan intruks pada alamat terpilih **“004016A0”** maka tekanlah tombol **“Space”** pada keyboard dan akan muncul suatu dialog untuk kita menuliskan instruksi baru kita. Tuliskanlah instruksi berikut :

MOV EAX, 40114C

Yang artinya kita memasukan nilai **40114C** ke register EAX

Ingat lah bahwa **40114C** adalah nilai Virtual Address (VA) entry-point yang asli program

Lalu tekanlah tombol bertuliskan **“Assemble”** untuk menyimpang instruksi lihatlah gambar berikut:

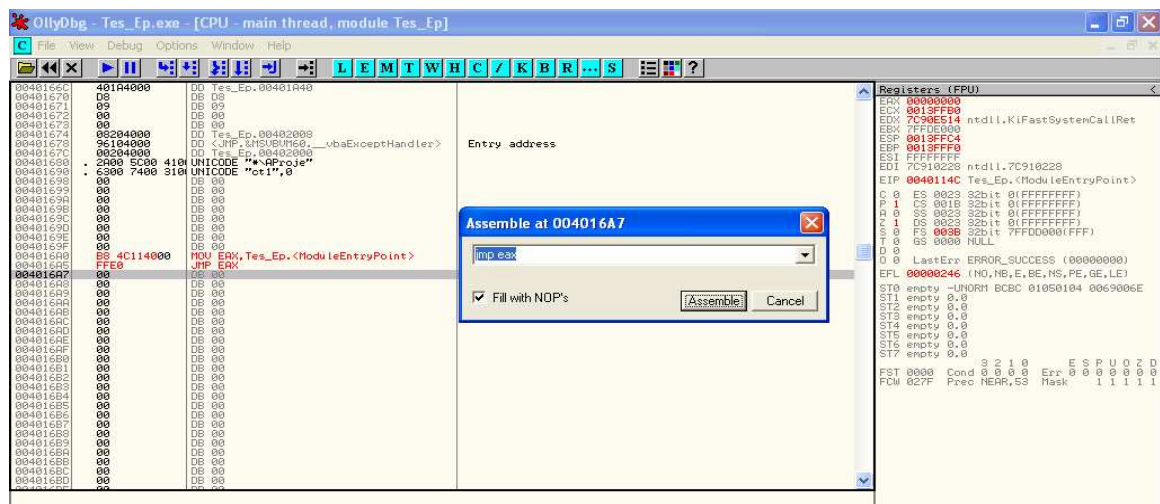


Setelah kita tekan “Assemble” maka alamat yang tadi kita pilih akan berubah menjadi merah dengan instruksi sesuai yang kita berikan. Lalu pindahlah kebaris selanjutnya, yaitu baris setelah warna merah dan tekanlah tombol “Space” untuk menuliskan instruksi lagi:

Instruksi sekarang yang akan kita tuliskan adalah :

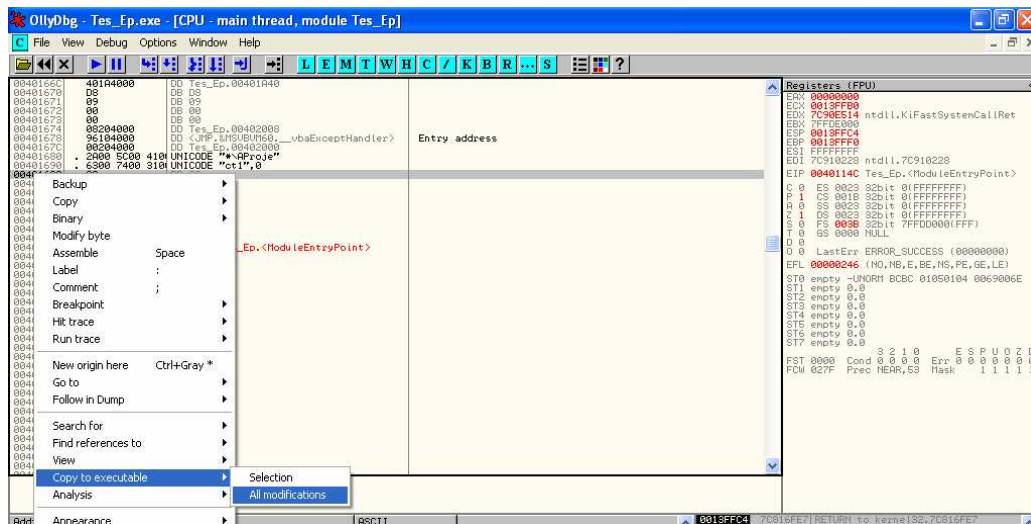
JMP EAX

Arti instruksi di atas adalah untuk menuju alamat yang sudah tersimpan pada register EAX, dimana sebelumnya register EAX menyimpan nilai **40114C** yang merupakan Virtual Address dari entry-point asli program yang akan kita rubah. Berikut adalah gambarnya

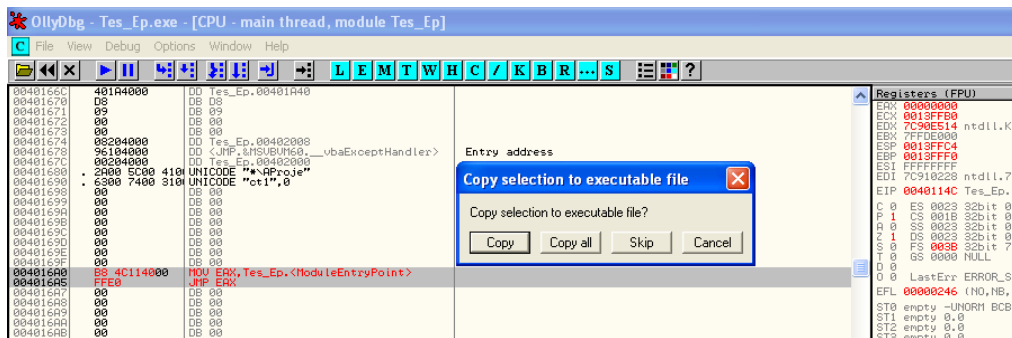


Nah setelah itu waktunya kita menyimpan hasil pekerjaan kita, sebagai file baru yaitu dengan cara sebagai berikut:

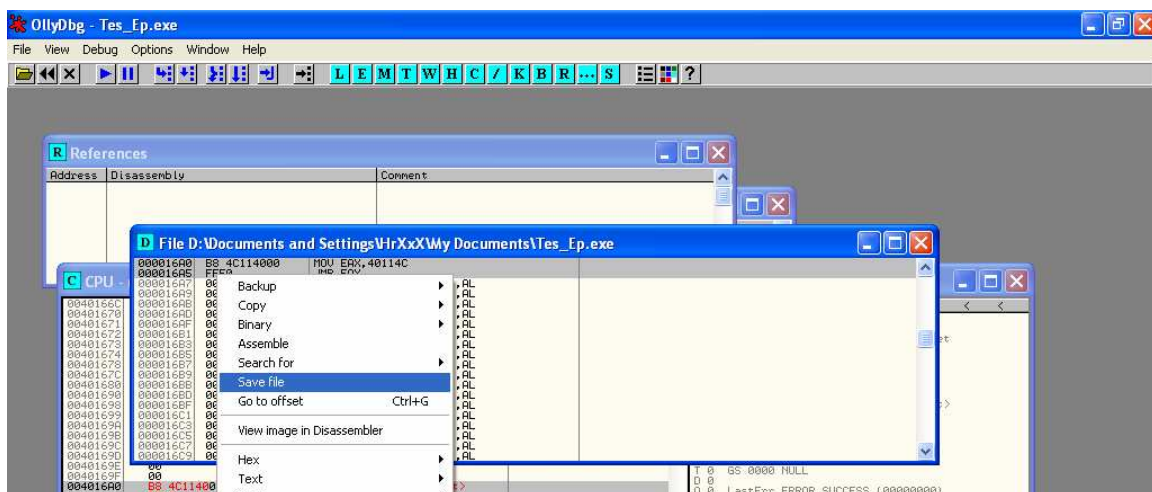
- Klik kanan jendela (column) Main Thread atau jendela yang anda rubah tadi, maka akan muncul popup menu dan pilihlah menu “Copy to executable” pilih sub menu “All modification” lihat gambar berikut :



Lalu pilih “Copy all” pada kotak pesan yang muncul seperti berikut :

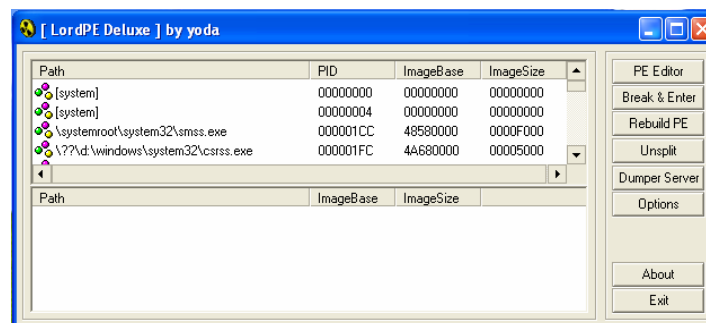


Maka setelah itu akan muncul jendela baru (bericon huruf **D**) seperti berikut, lalu klik kanan daerah dalam kolom jendela baru lalu pilihlah menu “Save File” :

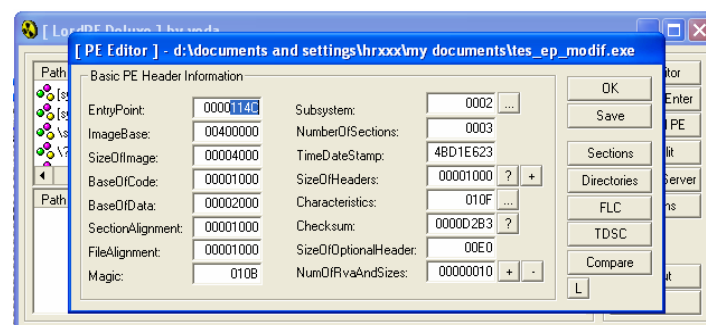


Simpanlah file tersebut sesuai nama anda, misalnya “**Tes_Ep_modif.exe**” dan selesailah sementara tugas kita. Tutuplah aplikasi OllyDbg untuk sementara waktu. Apakah tugas kita sudah berakhir? Tentu saja belum, walaupun kita sudah menentukan entry-point baru untuk program dan menuliskan instruksinya, namun secara default entry-point PE belum berubah karena informasi pada Header PE (Optional Header) yaitu *Address of Entry Point* belum kita rubah, untuk itu kita butuh aplikasi Bantu kedua untuk melakukan perubahan tersebut agar Entry-Point program menunjuk pada entry-point baru yang telah kita tetapkan tadi.

Bukalah aplikasi LordPE Deluxe sebagai berikut :



Bukalah aplikasi yang baru kita utak-atik dengan OllyDbg yaitu “**Tes_Ep_modif.exe**” dengan menekan tombol “**PE Editor**” lalu pilihlah file bernama “**Tes_Ep_modif.exe**” tersebut sehingga akan keluar penampakan seperti berikut:



Perhatikanlah textbox yang didepanya bertuliskan “Entry-Point” yaitu nilainya **0000114C**. Lalu apa kaitanya dengan nilai Virtual Address **040114C** dari entry-point yang kita dapatkan dengan OllyDbg. Kaitanya adalah **0000114C** adalah nilai RVA (Relative Virtual Address) sedangkan **040114C** adalah nilai VA (Virtual Address), dimana kaitan antara VA dan RVA adalah sebagai berikut :

$$VA = \text{ImageBase} + \text{RVA}$$

$$0040114C = 0400000 + 0000114C$$

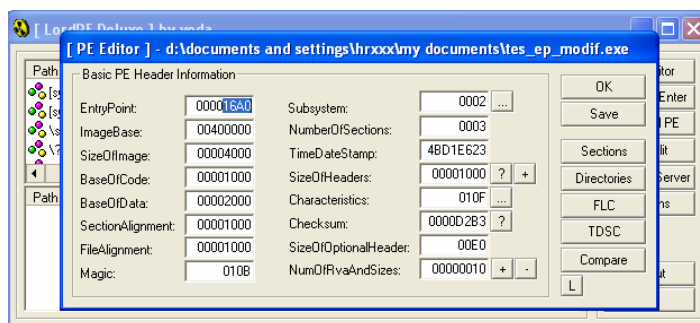
Penjelasan :

VA : Alamat di memori

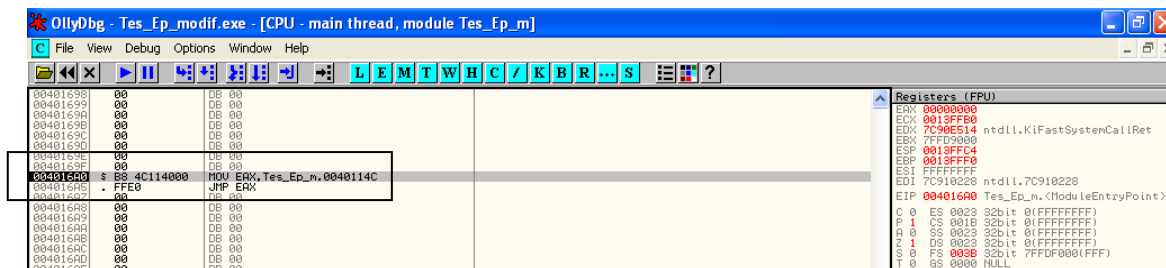
RVA : Alamat relatif di memori yang tergantung pada ImageBase dan VA

ImageBase : Alamat di memori terendah dimana PE terkait diload.

Rubahlah nilai **114C** (RVA entry-point lama) dengan nilai **16A0** (RVA entry-point baru), lihat pada gambar berikut :



Tekanlah tombol “Save” dan “OK” untuk kembali ke program LordPE dan tutuplah aplikasinya. Sekarang kembali ke OllyDbg bukalah file program yang barukita modifikasi “**Tes_Ep_modif.exe**” maka entry-point program sudah berubah seperti berikut :



Entry-Point program sekarang sudah tidak lagi berada pada Virtual Address **0040114C** namun sudah berganti pada Virtual Address **004016A0**. Sukseslah pekerjaan kita, sebagai tambahan informasi bahwa sebagian besar virus akan bertindak demikian yaitu merubah entry-point asli ke entry-point yang ia tetapkan dalam melaksanakan rutin infeksi. Hati-hari yah kalo memang belum memahai instruksi assembly disdarnkan mengikuti artikel ini apa adanya, agar praktik berhasil. Selamat mencoba ☺

Membuat Dll dengan Power Basic

by : HrXxX

Terkadang untuk keperluan yang lebih luas kita dituntut untuk tidak hanya membuat sebuah program executable (exe) saja, namun kita juga perlu menguasai pembuatan file PE lainnya seperti library (dll) dan driver (sys). Kali ini saya akan membahas bagaimana cara membuat file library atau std-dll yang mana dapat menyimpang fungsi yang dapat kita panggil secara external melalui program lain. Hal ini tentunya sangat menarik, karena untuk keperluan tertentu kita dapat membuat fungsi secara global yang dapat dimanfaatkan oleh programmer lain yang didistribusikan melalui file DLL.

Disini kita akan mempraktikkan pembuatan DLL dan pemanggilan fungsi yang terkandung didalamnya (yang di export) dengan bahasa Basic, sebelumnya perlu saya jelaskan bahwa untuk pembuatan DLL nya sendiri saya memakai Power Basic dan untuk program pemanggil fungsi DLL saya akan menggunakan Visual Basic. Mungkin sebagian dari Anda belum kenal dengan Power Basic, ada baiknya untuk sedikit saya beri penjelasan bahwa Power basic hamper sama dengan Visual Basic tentunya punya kelebihan dan kekurangan masing-masing. Hanya saja karena Power Basic mendukung berbagai fitur yang tidak terdapat pada Visual Basic seperti membuat std-dll maka saya memilih Power Basic untuk pembuatan DLL nya.

Untuk mendapatkan Power Basic secara legal memang tidak mudah, karena sifatnya berbayar, namun ada juga yang gratis di Internet atau anda bisa langsung masuk ke forum Codenesia untuk mendapatkan link Power Basic versi 9.0x yang gratis untuk anda sedot. ☺ Link sudah ada di CNZine ini.

Membuat File DLL

Okay langsung saja siapkan Power Basic Anda dan tuliskan code seperti berikut pada code editor PowerBasic :

```

#COMPILE DLL "MyDll.Dll"
#DIM ALL
#include "Win32API.inc"

GLOBAL ghInstance AS DWORD

'Fungsi EntryPoint (Wajib)
FUNCTION LIBMAIN (BYVAL hInstance AS LONG, _
                  BYVAL fwdReason AS LONG, _
                  BYVAL lpvReserved AS LONG) AS LONG

    SELECT CASE fwdReason

    CASE %DLL_PROCESS_ATTACH

        ghInstance = hInstance
        FUNCTION = 1    'success!

    CASE %DLL_PROCESS_DETACH

        FUNCTION = 1    'success!

    CASE %DLL_THREAD_ATTACH

        FUNCTION = 1    'success!

    CASE %DLL_THREAD_DETACH

        FUNCTION = 1    'success!

    END SELECT
END FUNCTION

' Nah ini fungsi yang bisa dipanggil (export)
FUNCTION Jumlah ALIAS "Jumlah" (BYVAL Bil1 AS LONG, BYVAL Bil2 AS LONG) EXPORT _
AS LONG
    FUNCTION = Bil1 + Bil2
END FUNCTION

```

Penjelasan Code Penting :

#COMPILE DLL "MyDll.Dll"

- Kita mengcompile sebuah DLL dengan nama **MyDll.Dll** akan tercompile disamping file sumber setelah kita memilih menu compilasi pada Power Basic.

```

FUNCTION LIBMAIN (BYVAL hInstance AS LONG, _
                  BYVAL fwdReason AS LONG, _
                  BYVAL lpvReserved AS LONG) AS LONG
..... potongan code
END FUNCTION

```

- Adalah fungsi standar Entry Point DLL pada Power Basic, harus ditulis demikian.

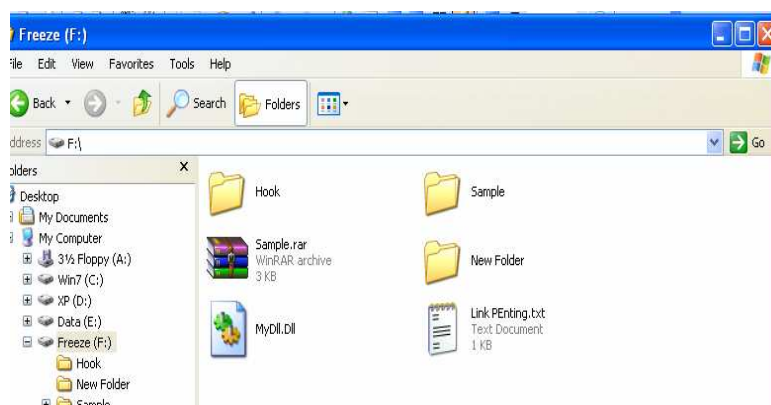
```

FUNCTION Jumlah ALIAS "Jumlah" (BYVAL Bil1 AS LONG, BYVAL Bil2 AS LONG) EXPORT _
AS LONG
    FUNCTION = Bil1 + Bil2
END FUNCTION

```

- Fungsi yang kita buat untuk bisa dipanggil oleh program yang akan memanggilnya, fungsi akan dikenal dengan nama “Jumlah” secara external dan internal dengan tipe pengembalian fungsi adalah tipe data **Long** (AS LONG). KeyWord **Export** (wajib) menandakan bahwa fungsi ini nampak dari luar dan dapat dipanggil secara external (dari program lain), serta masuk pada daftar fungsi yang di export oleh dll tersebut setelah kita kompilasi. Fungsi yang kita tulis ini hanya untuk menjumlahkan bilangan yang ada dalam parameternya.

Compile dengan memilih menu Run-Compile atau dengan menekan Ctrl+M agar menjadi DLL yang baru kita tulis codenya menjadi DLL yang siap pakai. Ambilah DLL yang baru kita compile lalu letakan DLL tersebut di tempat yang kita inginkan, misal di root drive “F:\”. Berikut adalah DLL yang berhasil kita kompilasi dan kita taruh di drive F:\, atau drive sesuka anda.



Membuat Program Pemanggil

Setelah membuat DLL tentunya kita harus membuat program untuk memanggil fungsi dari DLL tersebut (disini kita baru kita memakai Visual Basic), pada kasus ini fungsi yang akan kita panggil adalah fungsi dengan nama “Jumlah” dengan 2 parameter didalamnya dan fungsi balik bertipe data **LONG**. Nah untuk itu silahkan buka Visual Basic Anda lalu ketikan code berikut:

```
Private Declare Function Jumlah Lib "F:\MyDll.dll" _
    (ByVal Bil1 As Long, ByVal Bil2 As Long) As Long
Private Sub Form_Load()
    MsgBox Jumlah(2, 3)
End Sub
```

Penjelasan Code :

```
Private Declare Function Jumlah Lib "F:\MyDll.dll" _
    (ByVal Bil1 As Long, ByVal Bil2 As Long) As Long
```

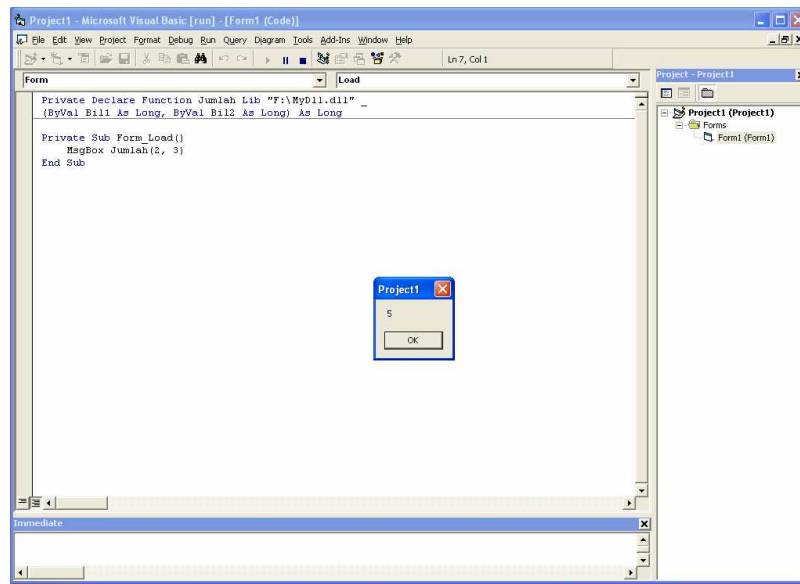
- Ini adalah deklarasi fungsi untuk memanggil fungsi bernama Jumlah yang di export oleh DLL bernama **MyDll.dll** yang kebetulan kita letakan pada drive F:\. Jika sudah ada (diletakan) pada folder *%system32%* atau *%windows%* maka cukup kita tuliskan namanya saja tanpa path-nya.

```
Private Sub Form_Load()
    MsgBox Jumlah(2, 3)
End Sub
```

- Untuk memastikan code berjalan sebagai mana mestinya, maka jika program dijalankan pesan yang akan ditampilkan adalah 5 (penjumlahan 2 dan 3).

Praktikan !

Jalankan program VB yang baru kita buat, maka hasilnya adalah sebagai berikut :



Ok, fungsi yang kita panggil dari DLL bekerja sebagai mana mestinya.. Dimana fungsi balik menghasilkan angka 5 pada kotak pesan. Angka 5 diperoleh dari hasil pengolahan bilangan yang kita masukan dalam parameter fungsi bernama “Jumlah” yang kita panggil dari **MyDll.dll**.

Kesimpulan :

Bagi programmer menulis program executable memang suatu kewajiban, tapi ingat bahwa tipe file Binary Executable pada Windows atau yang sering disebut PE punya berbagai jenis yang secara garis besar adalah Exe, Dll, dan Sys. Maka untuk itu kita di tuntut untuk menguasai pembuatan dan eksploitasi masing-masing jenis binary executable. Selamat berkreasi dengan DLL anda.

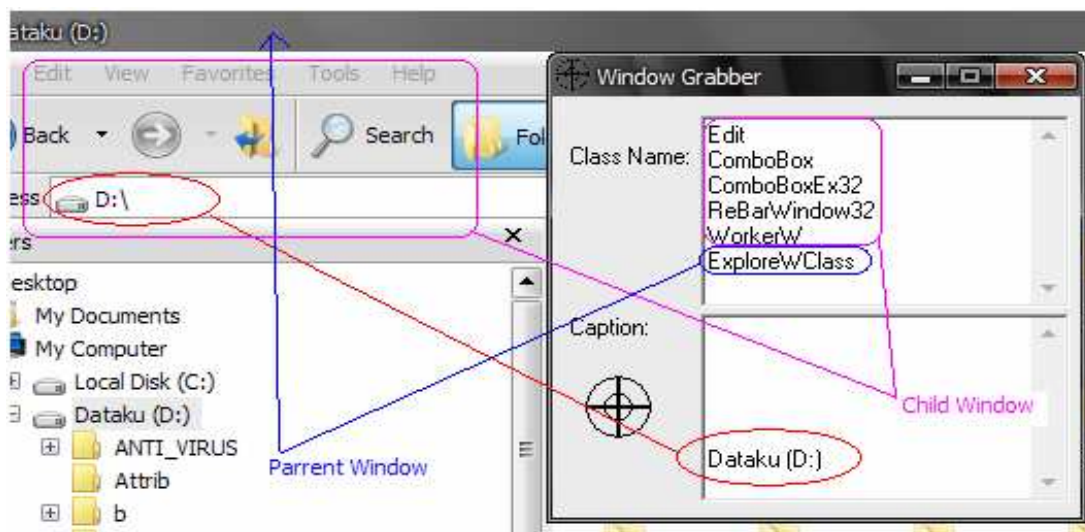
Membuat RTP Menggunakan API

by : Agus

Telah kita ketahui bahwa sudah semakin banyak Antivirus-Antivirus dalam negeri yang sudah menggunakan teknik RTP (Real Time Protector) pada Antivirus nya masing-masing, mulai dengan teknik yang sederhana sampai dengan teknik yang rumit. Di sini kita akan belajar bersama untuk membuat RTP Antivirus sederhana bermodalkan fungsi API saja. Langsung saja tanpa basa-basi lagi kita pelajari bagaimana teknik RTP yang akan kita buat bekerja, alur logikanya sederhana.

Kita mencari Handle Parent Window Explorer dari Class Name "ExploreWClass". Kemudian mencari handle Child Window untuk mengetahui lokasi Path yang terbuka di Explorer

Supaya lebih jelas kita lihat gambar berikut.



Nah dari gambaran di atas dijelaskan apa itu Parent Window dan Child Window, disana terlihat banyak nama - nama Class Name Child Window yang terdapat pada Parent Window . Biar lebih jelas kita tulis ulang nama-namanya.

"ExploreWClass" = Class Name Parent Window Explorer

"WorkerW" = Class Name Child Window Pertama

"RebarWindow32" = Class Name Child Window Kedua

"ComboBoxEx32" = Class Name Child Window Ketiga
"ComboBox" = Class Name Child Window Keempat
"Edit" = Class Name Child Window Kelima

Nah dari penjelasan diatas dapat disimpulkan bahwa kita bisa mendapatkan Path (Alamat) Explorer dari Caption Class Name "Edit" dengan memanfaatkan SendMessage. Oke saya rasa cukup pembahasan tentang Parent Window dan Child Window. Mari kita mulai penerapan Codingnya di VB Let's Go ...

Pertama, Buat Modul baru dan ketik code dibawah ini:

```
Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal  
hwndInsertAfter As Long, ByVal X As Long, ByVal Y As Long, ByVal cx As Long,  
ByVal cy As Long, ByVal wFlags As Long) As Long  
Public Declare Function FindWindowEx Lib "user32" Alias "FindWindowExA" (ByVal  
hwnd1 As Long, ByVal hwnd2 As Long, ByVal lpsz1 As String, ByVal lpsz2 As  
String) As Long  
Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal  
hwnd As Long, ByVal wParam As Long, ByVal lParam As Any) As Long  
Public Const WM_GETTEXT = &HD  
Public Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal  
lpClassName As String, ByVal lpWindowName As String) As Long  
Public JumVir As Long  
  
Dim hand1, hand2, hand3, hand4, hand5, hand6 As Long  
Dim temp As String * 256  
Public Function Scan_Folder(SingPath As String)  
Dim FileNow As String  
Dim FSO As Object  
Dim sFile As Object  
Set FSO = Nothing  
On Error Resume Next  
Set FSO = CreateObject("Scripting.FileSystemObject")  
For Each sFile In FSO.GetFolder(SingPath).Files  
DoEvents  
FileNow = sFile  
If isFileX(FileNow) = True Then  
frscan.lblscan = FileNow  
JumVir = JumVir + 1  
frscan.lblJum.Caption = "Jumlah File yang discan : " & JumVir  
End If  
Next  
End Function  
  
Private Function isFileX(filanya As String) As Boolean  
On Error GoTo palse  
If FileLen(filanya) > 0 Then  
isFileX = True  
Else
```

```

        isFileX = False
End If
Exit Function:
palse:
isFileX = False
End Function

' ##### Hanya Untuk Menempatkan Form paling Atas #####
Public Sub KeepOnTop(F As Form, yakin As Boolean)
    If yakin Then
        SetWindowPos F.hwnd, -1, 0, 0, 0, 0, 2 Or 1
    Else
        SetWindowPos F.hwnd, -2, 0, 0, 0, 0, 2 Or 1
    End If
End Sub

' ##### Untuk Monitoring Explorer #####
Public Function Lihat() As String
'Dapatkan Handle pertama / Parent Window dari Class Name
    hand1 = FindWindow("ExploreWClass", vbNullString)
'Dapatkan handle kedua ( Child Window Pertama dari Class Name )
    hand2 = FindWindowEx(hand1, 0&, "WorkerW", vbNullString)
'Dapatkan handle ketiga ( Child Window KeDua dari Class Name )
    hand3 = FindWindowEx(hand2, 0&, "RebarWindow32", vbNullString)
'Dapatkan handle keempat ( Child Window KeTiga dari Class Name )
    hand4 = FindWindowEx(hand3, 0&, "ComboBoxEx32", vbNullString)
'Dapatkan handle kelima ( Child Window KeEmpat dari Class Name )
    hand5 = FindWindowEx(hand4, 0&, "ComboBox", vbNullString)
'Dapatkan handle keenam ( Child Window KeLima dari Class Name )
    hand6 = FindWindowEx(hand5, 0&, "Edit", vbNullString)
'Mendapatkan Text dari Handle kemudian masukkan ke variable Temp
    SendMessage hand6, WM_GETTEXT, 200, ByVal temp
    Lihat = temp
End Function

```

Kedua buat form baru dan ikuti aturan berikut ini

```

Nama Form    = "frscan"
TextBox      = "Text1"
Label 1      = "Lblscan"
Label 2      = "lblJum"
Timer        = "Timer1"

```

Kemudian tulis code dibawah ini pada form tersebut

```

Private Sub Form_Load()
    ' Taruh form paling Atas
    KeepOnTop frscan, True
End Sub

Private Sub Text1_Change()
    'Kosongkan dulu variabel
    JumVir = 0

```

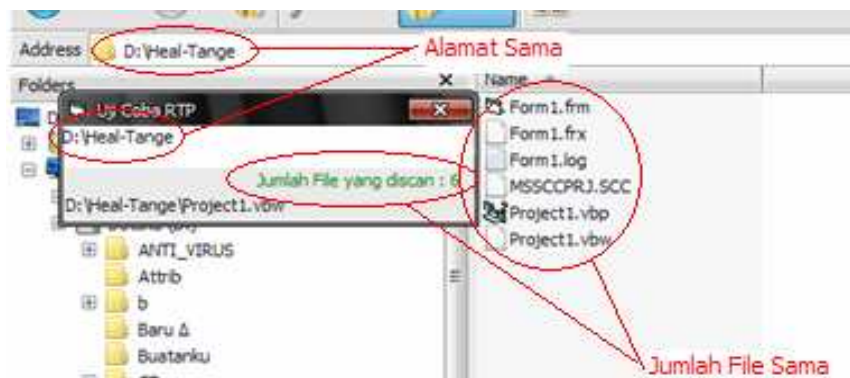
```

'##### Scan Folder #####
Call Scan_Folder(Text1.Text)
End Sub

Private Sub Timer1_Timer()
'##### Monitoring Explorer #####
Text1.Text = Lihat
End Sub

```

Selesai deh, saya tidak perlu menjelaskan lagi code tersebut karena sudah dijelaskan tadi kodenya. Oh iya dalam code tersebut belum ada metode – metode untuk mendapatkan virus karena ini bentuk kreasi anda, selamat berkreasi. Oke mari kita coba jalankan. Lihat ini hasilnya



Thank's to **Codenesia**

Salam kenal

Agus Minanur Rohman

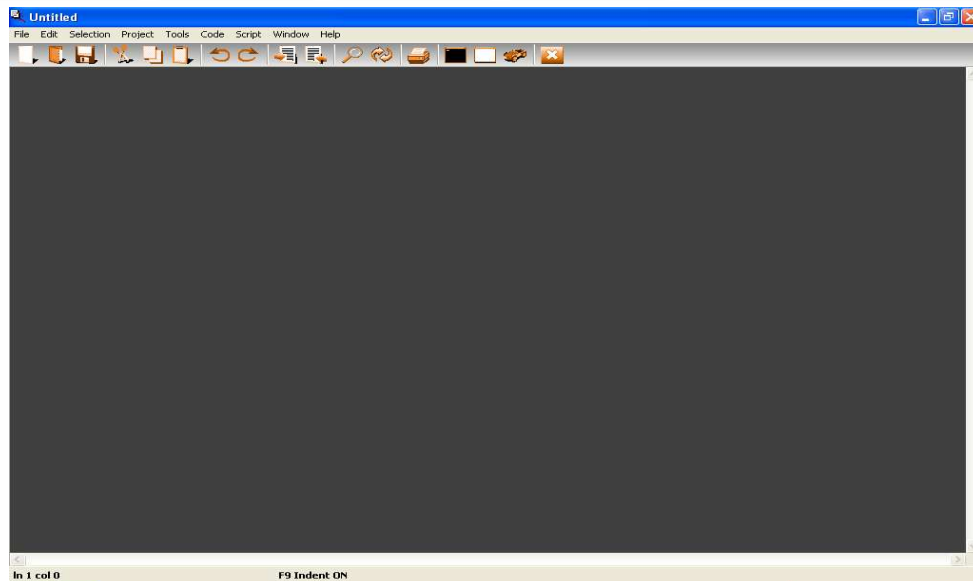
Belajar Assembly dengan MASM

by : **Bidan Malware**

Assembly bisa dikatakan bahasa pemrograman tingkat rendah yang menjadi momok bagi programmer pemula, mungkin saya dan anda juga orang yang termasuk takut ketika berurusan dengan pemrograman assembly. Namun segala sesuatu yang ada didunia ini pasti bisa dipelajari dengan mudah, asalkan mau dan berusaha untuk mencoba nya. Seiring dengan perkembangan zaman, bahasa Assembly sudah tidak sesulit dulu lagi karena saat ini muncul tool-tool development assembly yang dirancang khusus untuk mempermudah pengguna dalam melakukan koding assembly baik secara console dan graphical, karena sudah mengandung banyak macro bawaan sehingga menyerupai bahasa tingkat tinggi seperti bahasa Basic ataupun Pascal. Nah kali ini saya akan coba mengajak anda sedikit belajar pemrograman assembly dengan tool yang paling populer saat ini adalah MASM (Macro Assembly) yang bisa anda unduh di websitenya secara gratis di www.masm32.com.

Bagaimanapun juga bahasa assembly memang lebih susah dibanding bahasa lainnya, namun keuntungannya cukup banyak karena program yang dibuat oleh bahasa assembly akan relative lebih kecil selain itu akses program juga lebih cepat asalkan kita dapat mengefisiensi dan memanfaatkan kode assembly yang ada dengan baik.

Jika anda belum memiliki tool MASM maka unduhlah terlebih dahulu tool tersebut, installah pada OS anda lalu jalankan Code Editornya untuk memulai menuliskan programnya. Berikut adalah tampilan dari Code Editor program MASM.



Disini saya memilih MASM karena MASM adalah bahasa assembly yang cukup populer, dan banyak yang memakai MASM serta MASM mendukung compiler untuk semua jenis PE (exe, dll, sys). Namun kusus bagi anda yang ingin menulis driver dengan MASM sudah tersedia **KmdKit** bagi anda dengan ukuran yang minimalis disbanding anda memakai WDK dari bahasa C/C++.

Mari kita mulai menulis code program yang paling sederhana di MASM, program hanya menampilkan kotak pesan ketika dijalankan. Berikut adalah code programnya :

```
.486                                ; directive untuk instruksi processor
.model flat, stdcall                ; 32 bit memory model
option casemap :none                ; case sensitive

include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib
includelib \masm32\lib\user32.lib

.data
    szKata db "Hello Indonesia", 0
.code

start:
    invoke MessageBox, NULL, ADDR szKata, ADDR szKata, MB_OK
```

```
invoke ExitProcess, 0  
end start
```

Penjelasan Code

Simple bukan? Sekarang mari kita bahas code di atas perbagian.

.486

Directive (petunjuk) pada MASM agar mendukung instruksi prosesor 486 dan di atasnya.

.model flat, stdcall

Directive **.MODEL** berfungsi untuk memberikan intruksi MASM terhadap memory program yang di buat. **Flat** artinya model yang paling sesuai untuk program Windows. **STDCALL** artinya metode untuk passing parameter dalam code, stdcall mengintruksikan bahwa passing parameter dilakukan dari kanan ke kiri dengan **PUSH**.

```
include \masm32\include\windows.inc  
include \masm32\include\kernel32.inc  
include \masm32\include\user32.inc
```

File-file pendukung standar pada MASM yang akan diikuti sertakan, file standar tersebut sebagai pengganti fungsi-fungsi deklarasi fungsi pada DLL jika dalam pemrograman seperti Visual Basic.

```
includelib \masm32\lib\kernel32.lib  
includelib \masm32\lib\user32.lib
```

Pasangan dari masing-masing file *.inc

.data

Directive untuk memulai blok data yang sudah diinisialisasi, semua variable yang sudah diinisialisasi nilainya akan masuk disini.

szKata db "Hello Indonesia", 0

Mendefinisikan suatu variable bernama **szKata** bertipe string byte (db) dengan nilai "Hello Indonesia". Tipe stringnya adalah ANSI sehingga kita berikan nilai 0 setelah tanda koma (,).

.code

Directive untuk memulai blok kode program

start :

Label wajib untuk memulai penulisan kode.

invoke MessageBox, NULL, ADDR szKata, ADDR szKata, MB_OK

Invoke hanya ada pada MASM, fungsinya seperti memanggil fungsi API yang telah di-deklarasikan jika pada bahasa pemrograman tingkat tinggi. Arti kode di atas adalah kita memanggil fungsi API bernama MessageBox yang sudah dideklarasikan pada file **user32.inc** yang kita ikut sertakan pada kode program ini. Perhatikan kata tercadang **ADDR**, pada MASM ini berfungsi untuk mendapatkan alamat dari variable (szKata) yang menyimpan string *"Hello Indonesia"*. Sehingga dengan demikian program akan menampilkan kotak pesan berisi kata Hello Indonesia dengan Title sama, dan Kotak pesan memiliki satu tombol saja (OK) karena tipe tombol yang kita masukan pada parameternya adalah **MB_OK**.

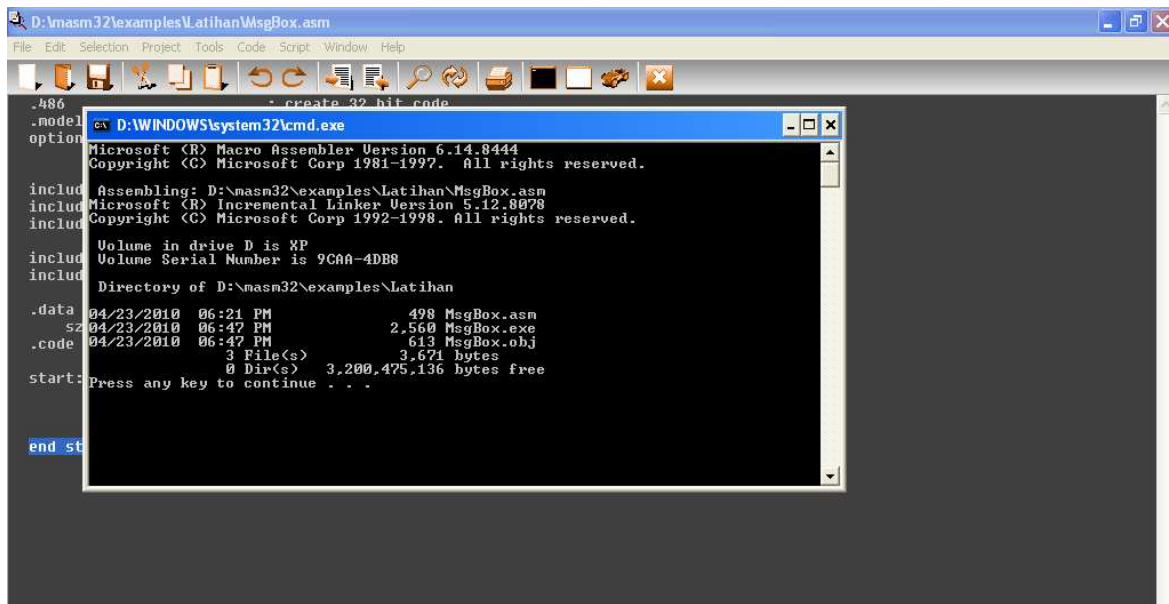
invoke ExitProcess, 0

Memanggil fungsi API bernama ExitProcess pada kernel32.dll, dengan parameter 0. Ini berfungsi untuk keluar dari program.

end start

Suami/istri/selingkuhan/pasangan dari label **start**:

Untuk menjalankan program yang baru kita tulis, simpanlah terlebih dahulu code di atas dengan tombol save pada code editor lalu kita pilih menu "Project-Build All" maka akan muncul tampilan seperti berikut :



Setelah berhasil mengkompilasi, maka untuk menjalankan programnya melalui shortcut code editor anda dapat memilih menu “**Project – Run Program**” maka akan tampil kotak pesan sesuai yang kita tuliskan pada code:



Selesai sudah perjalanan awal kita bermain dengan Macro Assembly (MASM), tentu saja ini hanya contoh sederhana karena pada program yang kompleks kita akan dihadapkan dengan permainan register, mnemonics dan stack untuk itu belajar bahasa assembly butuh ketekunan khusus. Namun jangan khawatir karena sekarang juga sudah banyak tool untuk lebih mempermudah dalam penulisan code Assembly dengan MASM sehingga MASM seakan-akan semudah bahasa tingkat tinggi. Tool Bantu tersebut diantaranya adalah *RadAsm* dan *Easy Code GoAsm*, gak percaya download aja programnya karena link-nya ada di akhir majalah ini. ☺, atau cari saja dengan google. Selamat belajar... !

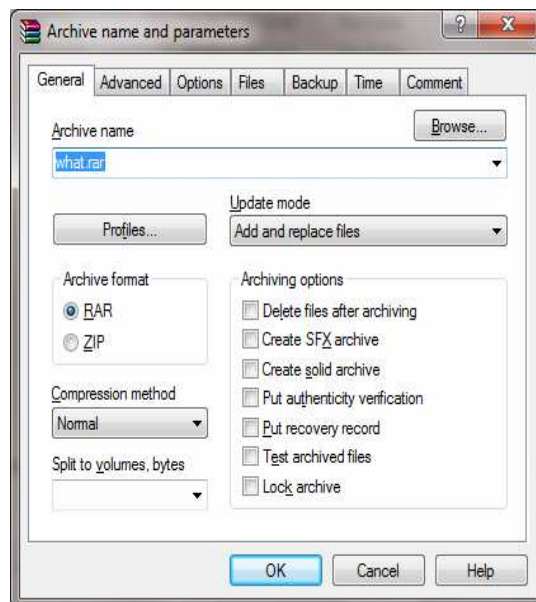
Hack Password Archive

by : Anharku

Pasti kalian semua sudah tidak asing lagi dengan Arciver yang satu ini? RAR adalah salah satu Archiver yang paling populer digunakan selain WinZip. Pernahkah anda mendownload sesuatu dari internet namun anda tidak dapat menggunakan hasil download tersebut karena file tersebut dilindungi dengan file archiver RAR yang terpassword? Mungkin anda yang ga mau nyerah mencoba-coba memasukkan password dengan segala kemungkinan password ingat teknik *social engginering*? Menebak password dengan segala sesuatu yang mendekati/ menyangkut sang pemilik file misalnya tanggal lahir, nama belakang, nama pacar,dll namun dari semua itu **GATOT** alias **GA**gal **TOT**al. Sebentar jangan menyerah dulu masih ada satu cara gunakan software untuk mendapatkan kembali password Archive RaR yaitu **RAR Password Recovery** yang dapat anda download di:

http://www.intelore.com/rar_password_recovery.php

Sebagai bahan percobaan kaya' di laboratorium aja uji coba hehehe☺ mari kita membuat file yang kita lindungi dengan Archive RAR yang dilengkapi password. Instal WinRAR dikomputermu. Klik kanan sebuah file dan **Add to Archive**.



Berpindah ke Tab **Advanced**. Lalu tambahkan password pada kolom enter password, misalnya 123 pada kolom reenter password for verification tulis kembali password yang tadi telah di masukkan (123).



Untuk melihat password tersebut beri checklist pada *Show password*. Lalu tekan **OK**

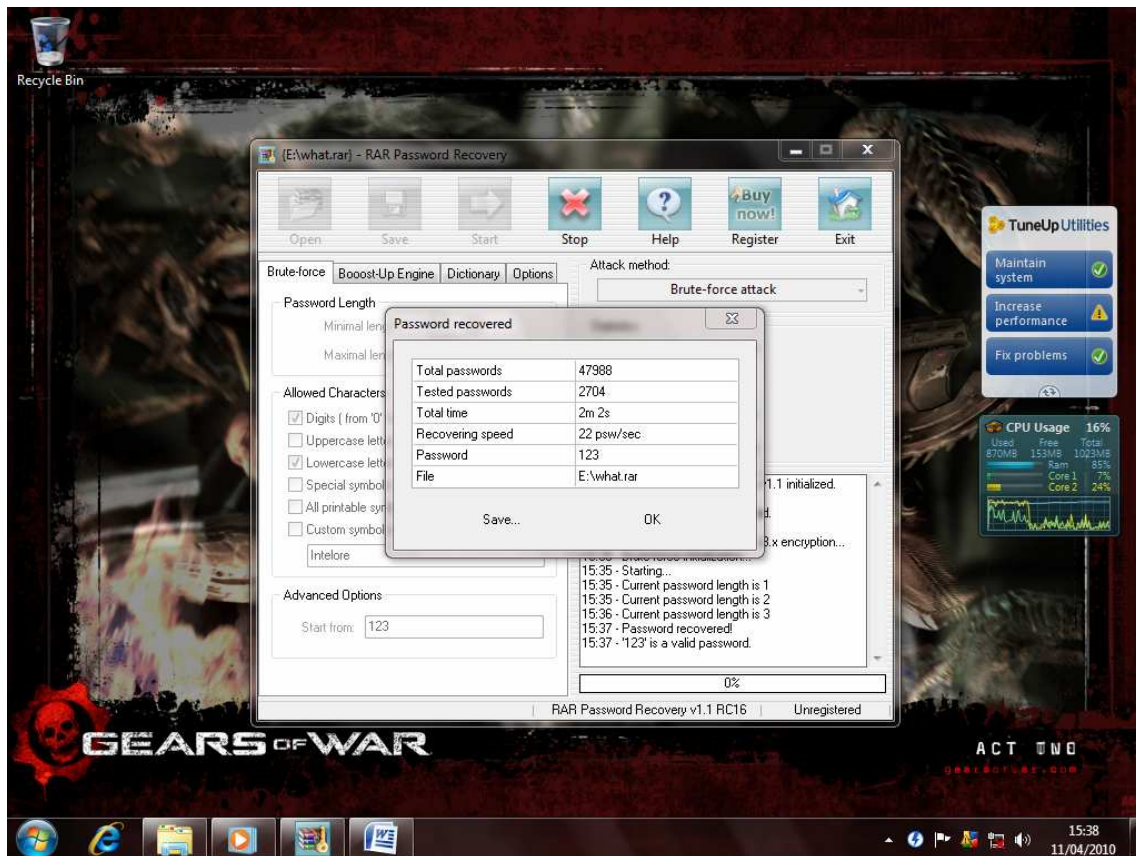


Instal **RAR_Password_Recovery.exe** lalu pindahkan/replace **urpwdr11rc16.exe (crack)** dengan file **urpwdr11rc16.exe** yang ada di Program Files\Intelore\RAR-PR. Setelah itu tekan Open lalu cari file yang akan kita cari passwordnya misal (what.rar). Pilih metode pencarian password yang akan di gunakan , Ada beberapa metode pencarian password yaitu:

Attack method:

- Brute-force-attack
- Booots-Up-attack
- Dictionary attack

Kita pilih disini metode yang kita gunakan dalam pencarian password yaitu Brute-force-attack metode Brute-force adalah metode melakukan pencarian password dengan membandingkannya pada kombinasi huruf besar/kecil dan angka serta symbol pada keyboard yang membentuk password dari karakter-karakter. Biasanya metode Brute Force membutuhkan waktu lama karena lama dalam mencari password menggunakan semua kombinasi yang mungkin.



Lihat proses yang dilakukan oleh software RAR Password Recovery terhadap file what.rar

```
15:34 - RAR Password Recovery v1.1 initialized.
15:34 - Unregistered version!
15:35 - 'what.rar' succesfully loaded.
15:35 - Opening 'E:\what.rar'
15:35 - Detected RAR/WinRAR v3.x encryption...
15:35 - Brute-force initialization...
15:35 - Starting...
15:35 - Current password length is 1
15:35 - Current password length is 2
```

15:36 - Current password length is 3
15:37 - Password recovered!
15:37 - '123' is a valid password.

Tunggu hingga pencarian password (Checked passwords) selesai biasanya lama pencarian password dapat kita lihat pada bagian **Remaining Time:n/a** atau jika pencarian password sudah selesai kita lihat **Total time** pada bagian laporannya.



Pada laporan tersebut terlihat berapa jumlah total password, tested passwords, total time, recovering speed. Lalu **terlihat password** dari file **E:\what.rar** tersebut yaitu **123**.

Hm....Sukses password telah ditemukan ☺ sekarang aku buka dalemannya, eh salah bukan daleman kayak pakaian aja ada daleman hehehe...aku buka isinya maksudnya ☺ wew ternyata hanya sebuah file 3gp yang ga begitu hot ☺

Anda pun dapat mencari password **WinZIP** yang hilang atau tidak anda ketahui dengan software **Advanced Archive Password Recovery** coba download di: www.elcomsoft.com/archpr.html

Catatan:

Gunakan password yang baik untuk melindungi file misalnya dengan menggunakan kombinasi huruf, angka dan symbol, dengan melakukan kombinasi dan memperbanyak jumlah karakter pada password, maka password akan lebih kuat dan susah untuk diJebol. Untuk lebih meningkatkan proteksi/perlindungan atas file, anda dapat menambahkan enkripsi(misalnya dengan software

encryptor) pada file anda. Tp ingat ***Nothing is Secure*** ,tidak ada sesuatupun yang aman di dunia ini. Seperti kata tidak ada yang sempurna di dunia ini, Kesempurnaan hanya milik Nya (Allah S.W.T)

Penting:

Jangan pernah berkata hacking menggunakan tools hanyalah hacking yang dilakukan oleh **script-kiddie** karena belum tentu mereka menggunakan tool tersebut tidak mengetahui atau memahami cara kerja tools. Jangan-jangan mereka tidak hanya memahami cara kerja tools bahkan mampu melakukan kustomisasi atau kombinasi berbagai tools dan teknik untuk melakukan eksploitasi. (selesai)



Membuat Personas Firefox Sendiri(Personas Codenesia)

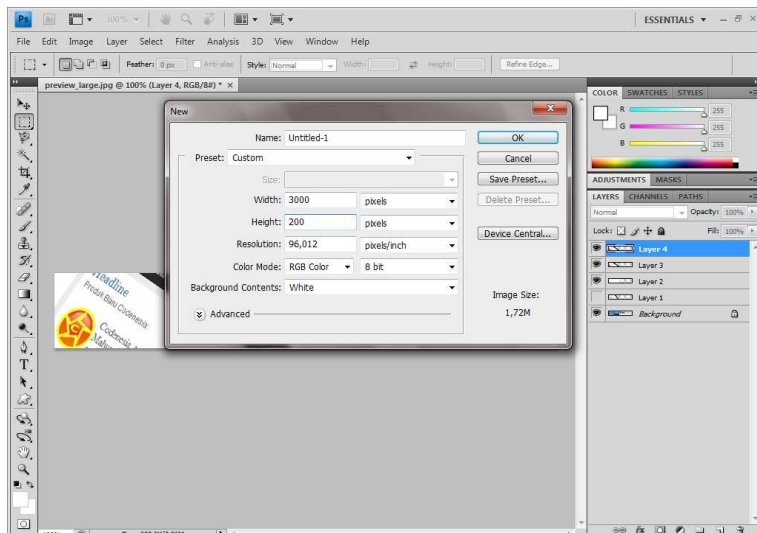
by : Anharku

Jika anda bosan dengan tampilan mozilla firefox yang itu-itu saja, polos-polos saja, artikel ini dapat membantu anda untuk membuat personas Firefox sendiri, membuat tampilan firefox lebih menarik, caranya adalah sebagai berikut:

1. Siapkan **Header**, buat dengan aplikasi pengolah gambar seperti **PhotoShop**.

Dengan Syarat:

- ✓ Dimensi 3000px wide x 200px high
- ✓ File PNG atau JPG
- ✓ Ukuran file maximal 300kb



2. Siapkan **Footer**.

Dengan Syarat:

- ✓ Dimensi 3000px wide x 100px high
- ✓ File PNG atau JPG
- ✓ Ukuran filenya maximal 300kb



3. Upload

Sebelum anda upload, Anda harus registrasi terlebih dahulu

Untuk membuat personas, buka website <http://www.getpersonas.com>

Cari dan tekan tombol **Get Personas Plus**, maka browser akan menuju ke lembar pendaftaran.

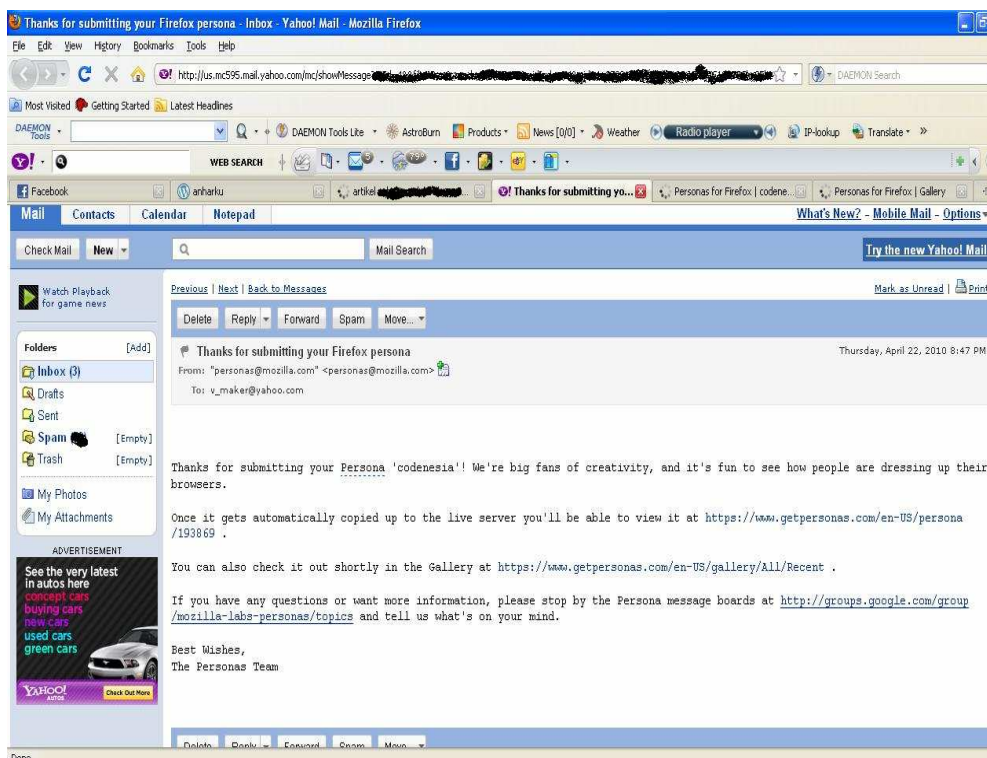
Isi dan lengkapi data-data anda, jangan lupa masukkan kata-kata yang ada pada CAPCHA yang terpampang misalnya **tweaking study**, lalu submit /kirim lembar pendaftaran tersebut.

Masukkan semua data mengenai personas yang kamu buat, nama personas, description/ deskripsi dari personas, category, lalu masukkan personas yang kamu buat tadi melalui tombol **Browse**.

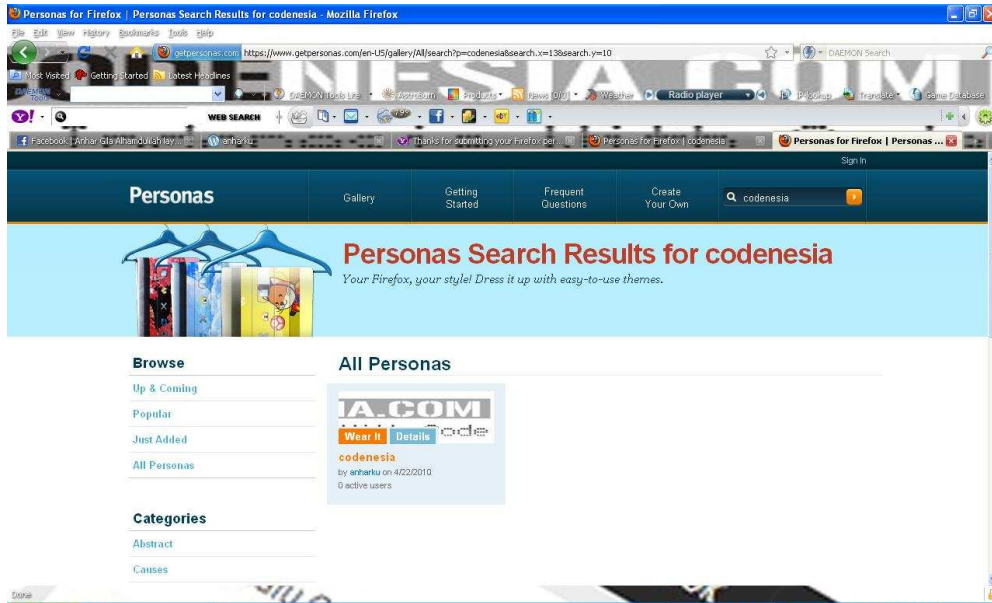


Setelah proses upload personas selesai maka akan keluar tampilan **Success!**

Buka e-mail yang kamu gunakan untuk mendaftar...lalu buka e-mail dari personas@mozilla.com yang berisi *"Thanks for submitting your Persona 'codenesia'! We're big fans of creativity, and it's fun to see how people are dressing up their browsers. Once it gets automatically copied up to the live server you'll be able to view it at <https://www.getpersonas.com/en-US/persona/193869> ."*



Biasa lah ucapan terima kasih telah mengupload persona codenesia, dan kita dapat melihat hasil upload persona kita pada link e-mail balasan tersebut. Untuk menggunakan personas, kita tinggal menekan tombol **Wear it**, Dengan begitu kita telah meng **add-on** themes codenesia ke dalam browser firefox kita.



Anda dapat mencari personas codenesia dengan mengetik di bagian search dengan kata “**codenesia**” Atau anda dapat membuka URL berikut ini <https://www.getpersonas.com/en-US/persona/193869>

Semoga tutorial ini dapat bermanfaat....dan jika kamu pencinta CODENESIA jangan lupa pakai personas firefox buatanku yagh atau bikin sendiri personas untuk codenesia tp yang jauh lebih bagus, lebih keren dan lebih kreatif dari yang aku buat hehehehe.. ☺

Thank's to: Codenesia.com

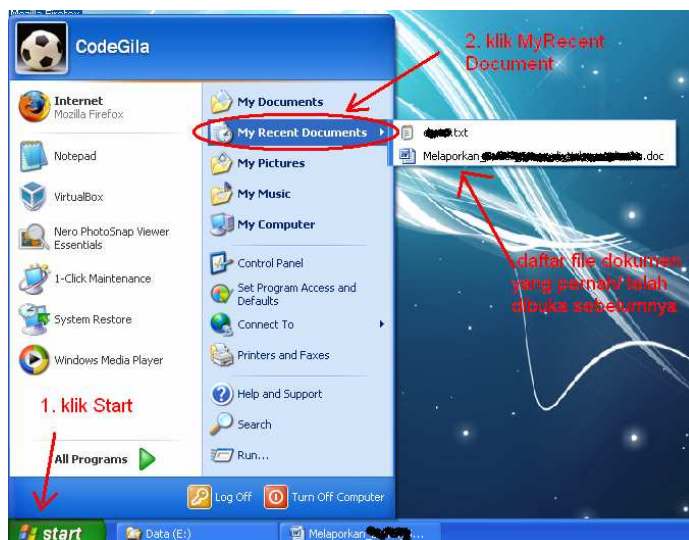
By: Anharku

Trick Menghilangkan MyRecent Document

by : Anharku

Jika anda adalah seorang yang tidak suka jika privasi anda diganggu, dokumen yang telah anda buka di OS Window's diketahui oleh orang yang meminjam komputer anda, maka anda dapat melakukan trik untuk menghilangkan *MyRecent Document*. Mengapa *MyRecent Document*? Ya karena fitur *MyRecent Document* ini berfungsi untuk memperlihatkan document yang baru atau telah dibuka. Trik menghilangkan *MyRecent Document* adalah sebagai berikut:

1. Lihat *MyRecent Document* dengan Klik **Start-MyRecent Document**, maka document yang telah kita buka sebelumnya akan diperlihatkan.



2. Untuk Menghilangkan MyRecent Document, Klik kanan **Start** dan pilih **Properties**.



3. Maka akan muncul jendela *Taskbar and Start Menu Properties*, tekan tombol **Coztumize**.



4. Keluar Jendela *Costumize Start Menu*, Hapus file yang terdaftar dalam *MyRecent Document* dengan menekan tombol **Clear List**, lalu hilangkan cheklist *List my lost recently opened document* untuk menghilangkan fitur MyRecent document, Tekan **OK**, lalu **OK** lagi...



5. Sekarang Cek apakah fitur *MyRecent Document* masih ada? Klik Start dan lihat fitur *MyRecent Document* sudah hilangkan?

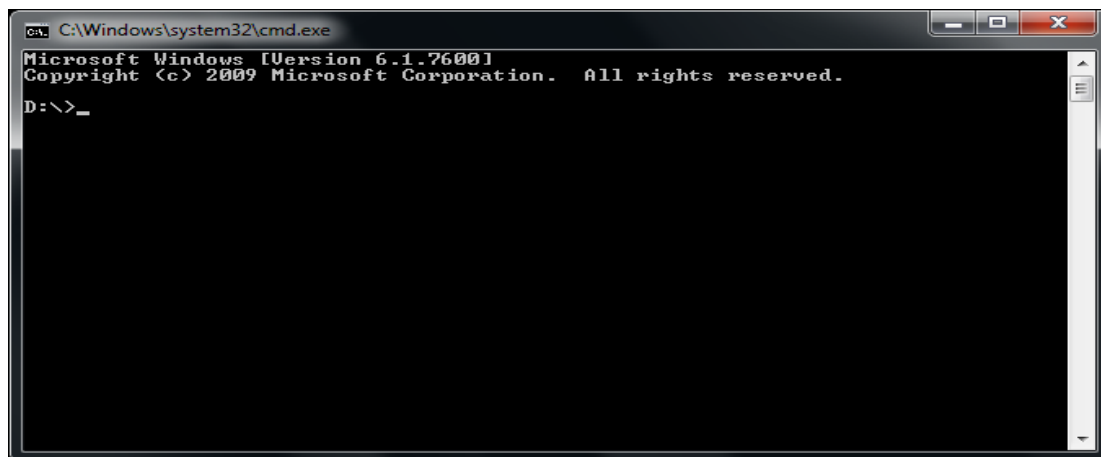


Dengan trik ini anda dapat sedikit menghirup nafas lega karna anda telah lebih berhati-hati menjaga privasi, sehingga orang yang menggunakan komputer anda tidak dapat melihat aktifitas atau dokumen yang anda buka, Sebaliknya saat anda meminjamkan komputer/ laptop ke teman anda aktifkan kembali fitur *MyRecent Document* dengan cara memberikan checklist ***List my lost recently opened document*** dengan begitu anda bisa mengawasi dokumen apa saja yang dibuka oleh teman anda. Semoga bermanfaat.. ☺

Command Prompt X

by : Gxry

Command prompt atau yang kita kenal cmd [cmd.exe] adalah suatu console yang benar2 hebat... hanya dari console tersebut kita dapat memanipulasi konfigurasi komputer, membuat folder, dan masih banyak kelebihan lainnya... dari hal tersebutlah muncul ide iseng yang katro...hehehe.. ide nya adalah membuat command prompt bajakan... maksudnya? Maksudnya begini, kita buat sebuah program [bisa dikatakan demikian] yang mempunyai fungsi yang sama dengan command prompt namun berbeda tampilannya. Ini adalah screen shot dari cmd yang asli :



Nah, bagaimana kalo kita rubah title, header, font color, back color, dan lainnya dari tu cmd gmana caranya?

oke, let's coding..

1. Buka notepad kalian
2. Kopas (Copy-Paste) ini script :

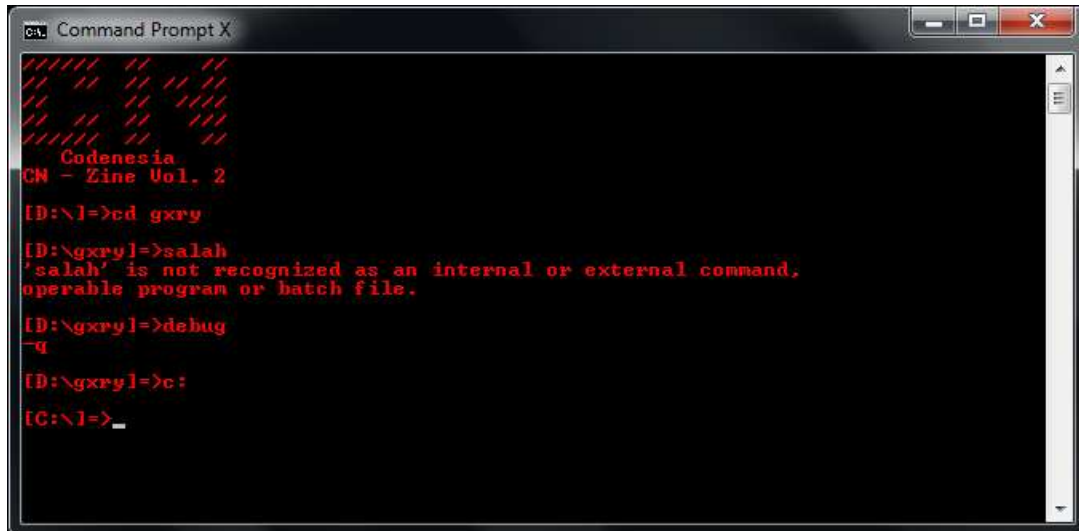
```
@echo off
color 0c
title Command Prompt X
echo // // // // //
echo // // // // //
echo // // // // //
echo // // // // //
echo // // // // //
echo // // // // //
echo Codenesia
```

```

echo CN - Zine Vol. 2
echo.
:ulang
set /p "cmdx=[%cd%]=>"
%cmdx%
echo.
goto ulang
::end of code::

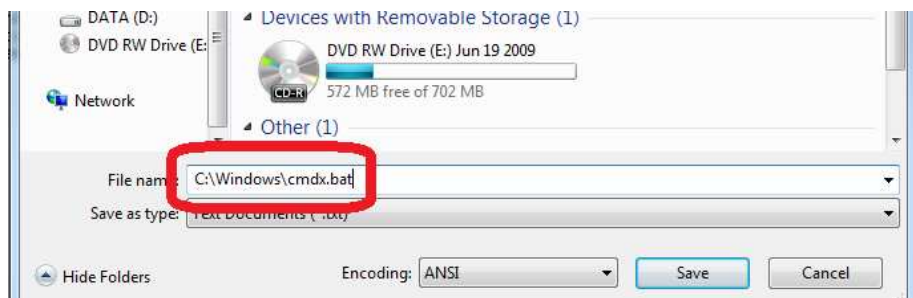
```

3. Save (simpan) dengan nama apa saja dengan ekstensi [*bat], misalnya : gary-clup-clup.bat.
4. Jalankan program-nya, atau file tadi sehingga akan muncul CMD baru seperti berikut.



Jika kalian ingin command prompt x ini juga bisa diakses di jendela "run", maka save/copy program kalian di path "C:\Windows" [jika C:\ adalah path system kalian] dan jangan memberi nama command prompt x ini "cmd.bat", sebab akan terjadi ambiguitas dan command prompt yang akan dieksekusi adalah command prompt yang asli.

Contoh kita menyimpan file tersebut dengan nama cmdx.bat pada folder %windows%.



Nah, Sekarang command prompt mu akan menjadi liar..., seperti cewek yang suka mangkal di perempatan taman lawang. ☺

Penjelasan scripting [menggunakan b.inggris, karena artinya lebih pas] :

- [-] @echo off = turns command echoing off.
- [-] echo = displays messages
- [-] echo. = new paragraph
- [-] title = sets the window title for the batch program "command prompt window".
- [-] color = sets the default console foreground and background color.

Corresponds to the background and the second the foreground. type "color /?" to know more the color

- [-] :ulang = label
- [-] goto = directs cmd.exe to a labeled line in a batch program
- [-] set /p = make a variable. in case "cmdx"
- [-] %cmdx% & %cd% = display the result of command

thx to.

- [-] My Jesus, My Everything
- [-] My Mom, Sister, and Brother
- [-] My Gemabel's Friends
- [-] Codenesia
- [-] All of you that read this patathic article [-] All of my friends, i couldn't mention your name one by one, guys...

Folder Quarantine

by : Gxry

Malware, yang ada di otak kita jika mendengar kata-kata itu selalu negatif dan berhubungan dengan worm, virus, atau apalah yang selalu merugikan.. Namun, sebagian orang mempunyai hobi yang cukup aneh, yaitu mengumpulkan Malware.. Utk apa? Ya, mungkin ada yang sekedar hobi, atau mungkin utk di analisa dan dibuatkan penangkalnya..

Ehm, apakah tidak bahaya?

Tentu saja tidak, asal kita mengerti, dan bisa menjaga keberadaannya, supaya Malware tsb tidak dieksekusi.. Maka dari itu, simpanlah Malware di tempat yang benar..jangan di taro sembarangan, kalo ke-eksekusi kan lumayan bikin horney..hahaaa...

Oke, Mari kita buat sebuah Folder Quarantine, Folder yang dapat memblokir file-file executable, seperti *.bat, *.vbs, *.exe, dsb. Jadi, kalo ada temen, sahabat, atau siapapun yang gak tau kalo itu adalah Malware, pada waktu dijalankan, malware tersebut tidak akan bisa dieksekusi.

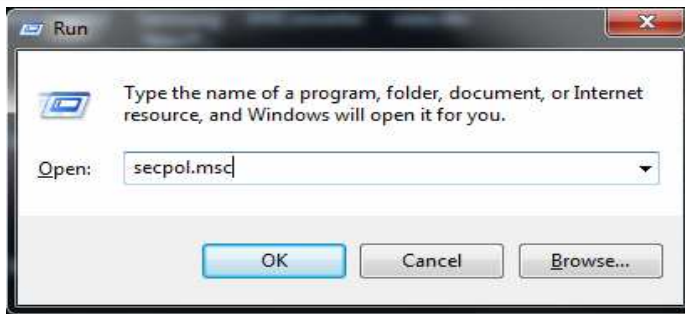
↳ **Pertama :**

Buatlah sebuah folder yang akan menjadi **Folder Quarantine** nantinya

Misal : *D:\Quarantine*

↳ **Kedua:**

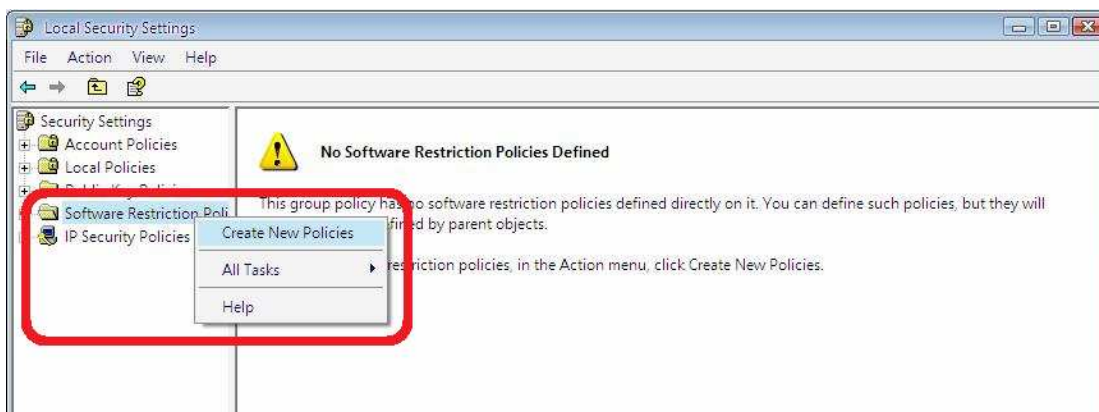
Kamu tekan tombol “**WinKey + R**” tanpa (curek), utk mengeluarkan jendela Run, kalo sudah, ketik “secpol.msc” dan enter, seperti yang terlihat pada gambar berikut:



Tekan “OK” sehingga akan keluar jendela “Local Security Policy”

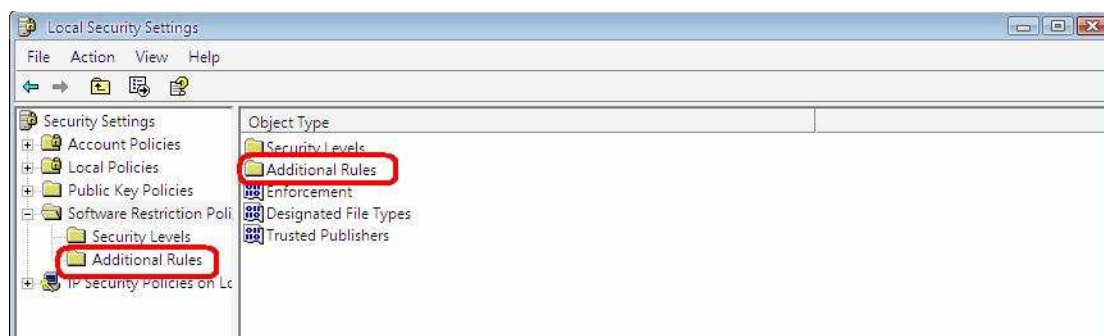
➤ **Ketiga :**

Kamu klik kanan di bagian "Software Restriction Policies", pilih "Create New Policies"



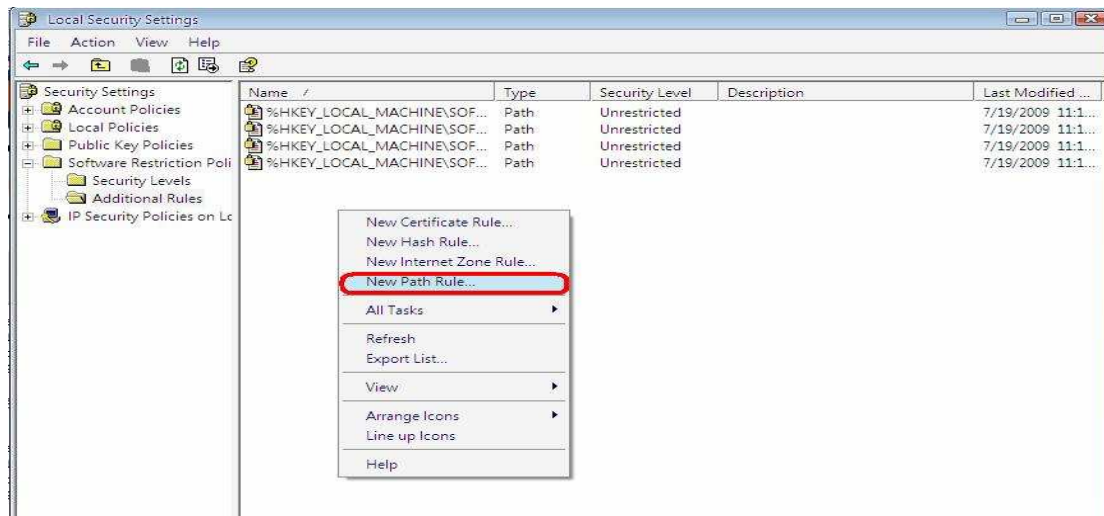
➤ **Keempat :**

Pilih "additional rules"

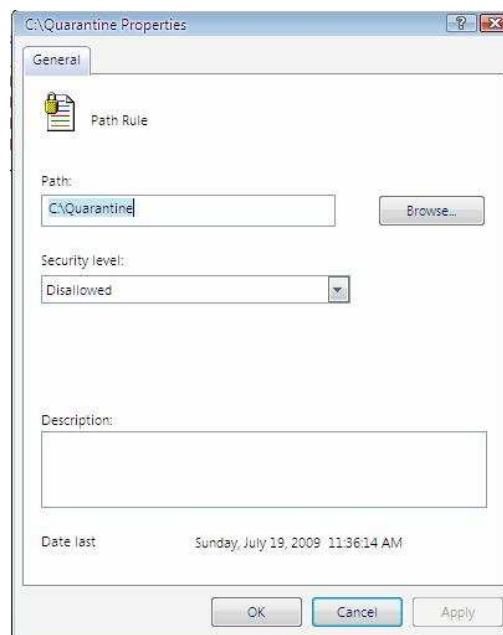


↩ Kelima :

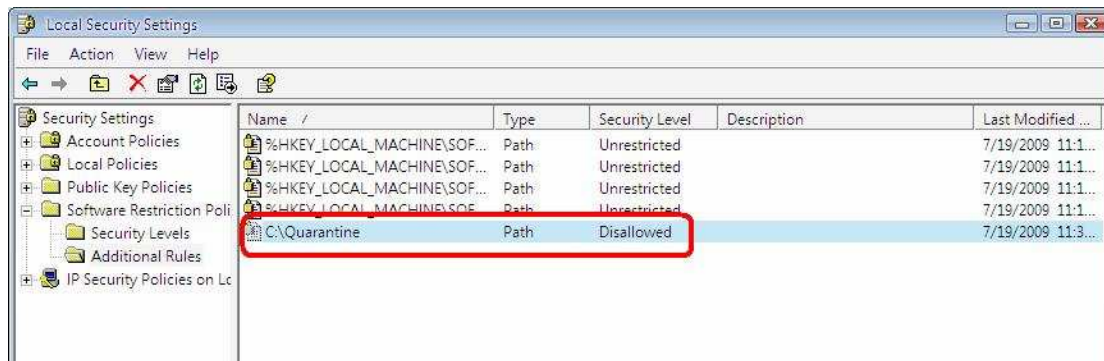
klik kanan di bagian kolom sebelah kanan, dan pilih "New Path Rule"



Jika sudah, akan keluar jendela "New Path Rule". Nah, sekarang kamu click "Browse" dan pilih folder yang tadi kamu buat atau folder yang menjadi folder quarantine-nya. Di bagian bawahnya, pilih "Disallowed", dan click "OK" atau "Apply"

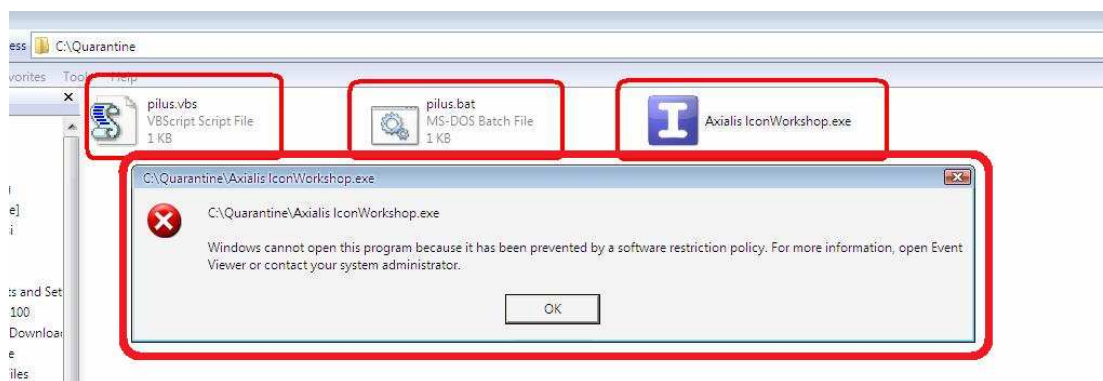


Nanti akan terlihat seperti ini..



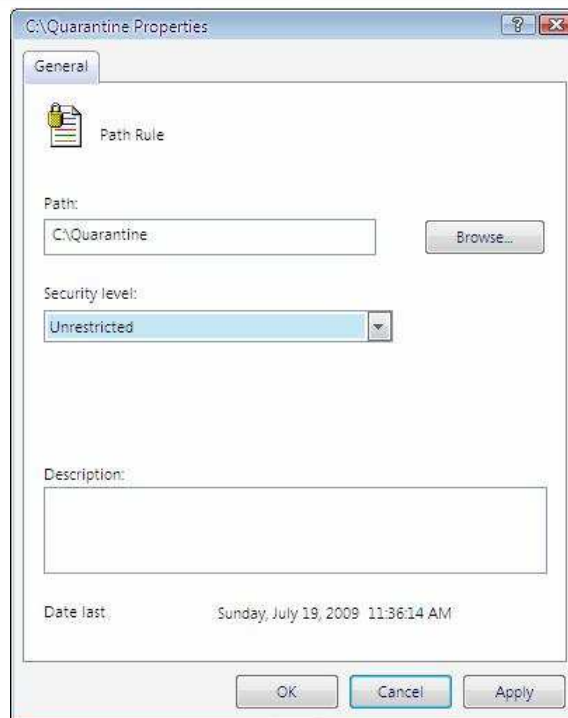
Nah, sekarang Folder Quarantine-nya sudah jadi...

Test dulu nih folder, Coba kamu taro di tu folder sebuah atau beberapa file executable, dan jalankan, pasti nggak bisa, dan akan seperti ini kan?...



Nah, Sekarang Malware kamu aman deh..

Jika, Folder tersebut ingin dibuat menjadi Folder yang biasa lagi, kamu hanya mengganti "Disallowed" menjadi "Unrestricted"



Oke, Sekarang gak usah takut lagi sama Malware, oke??. Tapi takutlah dengan saya ☺, lho..lho..

Tx to :

- My Jesus, My Everything
- My Family
- Gemabel, Codenesia..
- All of you guys, I couldn't mention your name one by one..

Caesar Chiper Coding C++

by : Gxry

Caesar Chiper, adalah trik Enkripsi yang mungkin bisa dibilang sangat lama sekali, soalnya, trik Enkripsi ini, digunakan pada waktu perang di masa Julius Caesar dulu. Teknik Enkripsi ini cukup sederhana, yaitu hanya dengan menggeser menambah/mengurangi jumlah bit saja...

Rumus :

Enkripsi :

$$En(x) = (x + n) \bmod 26$$

Dekripsi :

$$Dn(x) = (x - n) \bmod 26$$

Nah, itu rumus jadul, rumus utk sekarang, karena sudah ASCII, jadinya seperti ini :

Enkripsi :

$$En(x) = (x + n) \bmod 255$$

Dekripsi :

$$Dn(x) = (x - n) \bmod 255$$

Karena nilai ASCII [0-255], kan kalo huruf cuma sampe 26, "x" adalah nilai Ascii karakter yang akan di enkrip/dekrip, dengan "n" adalah jumlah geser bit, dan juga jangan lupa pake "mod 26", supaya tidak sampe ke huruf 27 [gak ada kan], makanya, dipake "mod 26", supaya jika kita geser 27 bit, akan kembali ke pertama...

oke, langsung kita coding aja dengan C++ seperti berikut :

```
#include <iostream.h>
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char kata[255]; //variabel kata hanya menampung 255 karakter saja
    int i = 0, y, x, n;

    cout<<"Original Code by : gxry"<<endl;

    cout<<"Masukkan Kata : ";gets(kata);
    cout<<"Masukkan bit geser : ";cin>>n;
```

```
while (kata[i] != 0) //looping sampai ketemu nilai kosong
{
    x = (int(kata[i]) + n) % 255; /*geser bit sebanyak n, dengan mod 255,
    karena ASCII hanya sampai 255*/
    cout<<char(x); /*menampilkan huruf dengan fungsi char dari nilai ASCII yang
    didapat*/
    i++; //pengubah nilai
}

cout<<endl;
system("pause");
}
```

Jadi begini, kalo rumus-rumus yang saya jabarkan diatas tadi, kan hanya utk [1] karakter saja ya, nah, kalo script yang saya kasih ini kan utk String [> 1 karakter]..

Logikanya berarti begini:

1. Ambil banyaknya jumlah karakter
2. Setiap karakter di enkrip satu persatu
3. Gabungkan karakternya

>> **Codenesia Malware Cleaner**



Kami juga mengembangkan produk Antivirus local yang kami beri nama CMC atau kependekan dari **Codenesia Malware Cleaner** yang tidak hanya dapat membasmi virus local namun juga sanggup membasmi beberapa virus asing secara tuntas, diman ketika majalah ini terbit versi terkahir dari CMC adalah PH 3.5 yang bisa anda unduh di website kami atau website CMC beralamat di www.cmc.codenesia.com secara gratis.

>> **RaX File Archiver**



RaX adalah Archiver program seperti Winrar, WinZip atau 7z, namun punya ekstensi .rax. RaX adalah produk pertama codenesia, yang bisa anda unduh di website codenesia atau di alamat hirin.4shared.com pada folder **Rax**.

Redaksi CN-Zine VOL #2

Email : info@codenesia.com

Layouter : Anharku

Editor : A.M Hirin

Cover : Sonny Lazuardi

CARA KIRIM ARTIKEL UNTUK CN-ZINE EDISI BERIKUTNYA

Isi materi artikel:

- ✓ Kategori Pemograman
- ✓ Kategori Hacking
- ✓ Kategori Cracking
- ✓ Kategori Antivirus
- ✓ Kategori Virus
- ✓ Kategori Etc (All of Komputer)

Kirimkan tulisan anda dengan format sebagai berikut :

- ✓ Filetype : .doc *)
- ✓ Page Setup : Paper size =A4
- ✓ Line spacing : 1,5 Lines
- ✓ Font : Times New Roman , size Judul Cambria = 16 (Heading1)
dan paragraph = 12

Catatan: isi materi diharapkan Original (tidak KOPI PASTE), tidak ada unsur penghinaan, tidak mengandung SARA', artikel yang masuk akan di seleksi terlebih dahulu oleh redaksi CN-Zine *).

Kirimkan tulisan anda ke Redaksi info@codenesia.com

*) hanya yang bertanda yang wajib

Link Download Tool External

Beberapa tool tidak kami sertakan dalam file **source_dan_tool** dikarenakan ukuran yang cukup besar dan suatu tindakan yang ilegal, untuk itu kami memberikan link bagi anda yang ingin mengunduh tool terkait yang belum ada di file **source_dan_tool** secara bebas.

Power Basic 9.01

<http://codenesia.com/smfforum/index.php/topic,261.0.html>

RadAsm


www.oby.ro/rad_asm/

GoASM

www.jorgon.freeseve.co.uk

QuickBat Compiler

http://www.4shared.com/file/238778845/37f76566/Quick_Batch_File_Compiler.html



**Berilah dukungan untuk Codenesia Magazine
VOL #3 agar menjadi lebih baik lagi.**

CODENESIA

Build Indonesia With Code