

Simulasi Berbagai Fungsi Dasar StarOffice dengan Macro



StarOffice adalah paket Office luar biasa. Kemampuannya untuk dikembangkan lebih lanjut dengan bantuan macro membuat paket Office ini begitu fleksibel. Dengan macro, kita bahkan bisa menyimulasikan berbagai fungsi dari StarOffice.

Bagi kita yang sering menggunakan StarOffice, tentu saja kita selalu menggunakan berbagai fungsi-fungsi dasar seperti membuat dokumen baru, membuka dokumen baik secara *read/write* ataupun *read only*, menyimpan dokumen, mengubah *properties* dokumen, mengetahui statistik dokumen, mencari teks tertentu dalam dokumen, ataupun mencetak dokumen.

Semua fungsi tersebut adalah fungsi dasar StarOffice. Tak terlepas dari komponen apapun yang Anda gunakan, baik Writer, Calc ataupun Impress, membuka dan menyimpan dokumen adalah tugas dasar yang tak terelakkan.

Dengan macro, kita dapat mengembangkan apa saja di StarOffice. Membuat file manager sendiri pun bisa asalkan rajin karena tersedia banyak dukungan fungsi file sistem. Tidak mudah memang mempelajari macro di StarOffice. Sering berlatih adalah salah satu cara untuk mempelajarinya.

Membuat macro pada dasarnya adalah hal yang menarik. Banyak hal tersimpan di paket office berukuran ratusan megabyte ini. Di artikel kali ini, kita akan membuat macro untuk menyimulasikan berbagai fungsi dasar StarOffice yang sering kita akses.

Kita akan melihat secara detail apa saja yang perlu kita lakukan untuk membuat dokumen baru. Tak lupa, hal-hal tersembunyi dalam proses membuka dokumen tersimpan juga akan kita kuak habis-habisan. Termasuk di antaranya bagaimana membuka dokumen dalam modus *read-only*.

Setiap contoh macro yang ada sebisa mungkin akan kita sajikan dalam bentuk *form*, sehingga akan terlihat lebih bagus dan bagi *user* akan lebih mudah untuk digunakan.

Kita akan melihat, bahwa kemampuan StarOffice tidak kalah dibandingkan dengan paket office di Windows, seperti Microsoft Office.

Persiapan

Sebelum memulai, ada baiknya kita melakukan persiapan terlebih dahulu. Dalam beberapa artikel sebelumnya, ketika kita membahas macro StarOffice, kita mengetahui bahwa macro dapat disimpan di dua tempat: ikut StarOffice ataupun ikut dokumen. Artikel ini akan menggunakan cara kedua. Dengan demikian, kita perlu membuat sebuah dokumen teks baru. Berikan nama yang bersesuaian, sebagai contoh *macro.sxw*. Kemudian, buatlah modul baru dengan cara berikut ini:

- Akses *Tools|Macros|Macro*.
- Pilih *dokumen|Standard*
- Klik tombol *New*, berikan nama *Module1*.
- Tutup layar edit macro, kita belum membutuhkannya untuk saat ini.

Berikutnya, kita perlu memahami terlebih dahulu beberapa hal berikut ini:

- Dulu, di StarOffice 5, kita mengenal StarDesktop sebagai suatu objek yang *visible*. StarDesktop berfungsi sebagai kontainer berbagai komponennya. Kini, di StarOffice 7, banyak perubahan yang

terjadi. Salah satunya adalah pergantian fungsi StarDesktop. Kini, StarDesktop bukanlah komponen *visible*. Saat ini, StarDesktop lebih mengacu kepada konsep. StarDesktop adalah objek yang menaungi StarOffice itu sendiri. Berbagai fungsi umum StarOffice akan dilakukan oleh StarDesktop. Sebagai contoh, membuka dokumen, menyimpan dokumen, mencetak dokumen, dan lain sebagainya.

- Dalam artikel ini, kita akan banyak bermain dengan objek StarDesktop. Sebagian kecil contoh akan membahas bagaimana bekerja dengan dokumen teks.
- Cobalah merekam macro dan kemudian lihat *source code*-nya. Akan terlihat sangat kompleks. Jangan pedulikan untuk saat ini. Kita akan mencoba lihat macro dari sisi yang mudah.

Kita telah membahas pembuatan dialog pada pembahasan macro StarOffice sebelumnya. Kali inipun, kita akan menggunakan macro. Hanya, perlu diperhatikan bahwa kita tetap membutuhkan Module untuk menyimpan berbagai subrutin kita. Dialog hanya berfungsi sebagai *user interface*, dan tetap akan memanggil berbagai subrutin yang disimpan di dalam Module tersebut.

Membuat dokumen baru

Untuk membuat dokumen baru, kita perlu meminta bantuan StarDesktop. Berikut ini adalah contoh kode dan penjelasannya:

```
Sub CreateNew
Dim doc as object
```

```

dim URL as string
dim Dummy()
URL = "private:factory/swriter"
doc = StarDesktop.loadComponent
FromURL(URL, "_blank", 0 ,
Dummy())
end sub

```

Subrutin tersebut, apabila dijalankan, akan membuka sebuah dokumen teks baru. Perhatikanlah isi dari variabel URL. Pada contoh ini, kita mengisi nilainya dengan `private:factory/swriter`.

Untuk membuka *spreadsheet* baru atau presentasi baru, gantilah `swriter` dengan `scal` dan `simpres`. Adanya kata kunci *private* kurang lebih melambangkan bahwa apa yang kita panggil merupakan milik internal StarOffice. StarOffice sangat kompleks dan dapat mengizinkan komponen lain mengontrol dokumen yang dibuka untuk diakses dari komponen luar.

Sementara, `Dummy()` adalah sebuah *array* kosong yang dilewatkan pada method `loadComponentFromURL` milik `StarDesktop`. Method tersebut mewajibkan adanya properti file dalam bentuk *array*. Karena kita belum membutuhkannya saat ini, kita menggunakan *array* kosong. Ini sepenuhnya valid di StarOffice.

`Doc` adalah variabel yang mengacu kepada objek dokumen baru yang dibuat.

Perhatikanlah method `loadComponentFromURL` tersebut. Parameter pertama adalah URL. Parameter kedua, yang saat ini bernilai `_blank`, akan membuka dokumen pada *window* baru, parameter ketiga adalah *search flag*, kita tidak akan membutuhkannya untuk saat ini. Parameter terakhir adalah properti pembukaan dokumen. Kita akan melihat contohnya nanti.

Setelah mengetahui cara-cara dasarnya, kita akan membungkusnya dengan user interface yang baik. Kita akan membuat dialog dengan beberapa tombol, yang masing-masingnya apabila diklik akan membuka dokumen baru.

Pilihlah *Tools|Macros|Macro*. Klik tombol *Organize* dan pilihlah *New Dialog* pada *library Standard*. Berikan nama `Dialog1`. Kliklah tombol *Edit*. Susunlah dialog seperti gambar contoh.

Setelah itu, buatlah tiga subrutin tambahan berikut di dalam `Module1`.

```

Sub CreateNewText
Dim doc as object
dim URL as string
dim Dummy()
URL = "private:factory/swriter"
doc = StarDesktop.loadComponent
FromURL(URL, "_blank", 0 ,
Dummy())
end sub

```

```

Sub CreateNewSpreadSheet
Dim doc as object
dim URL as string
dim Dummy()
URL = "private:factory/scalc"
doc = StarDesktop.loadComponent
FromURL(URL, "_blank", 0 ,
Dummy())
end sub

Sub CreateNewPresentation
Dim doc as object
dim URL as string
dim Dummy()
URL = "private:factory/simpres"
doc = StarDesktop.loadComponent
FromURL(URL, "_blank", 0 ,
Dummy())
end sub

```

Setelah itu, assignlah event *Mouse Button Released* masing-masing tombol dengan subrutin yang bersesuaian. Kenapa event *Mouse Button Released*? Sebenarnya, kita bisa saja memproses event *Mouse Button Pressed*. Namun, dengan asumsi bahwa aksi klik adalah menekan dan kemudian melepaskan tombol mouse, dengan kemungkinan user membatalkan aksi klik setelah menekan tombol mouse, memproses event ketika tombol mouse dilepaskan akan terasa lebih bijak.

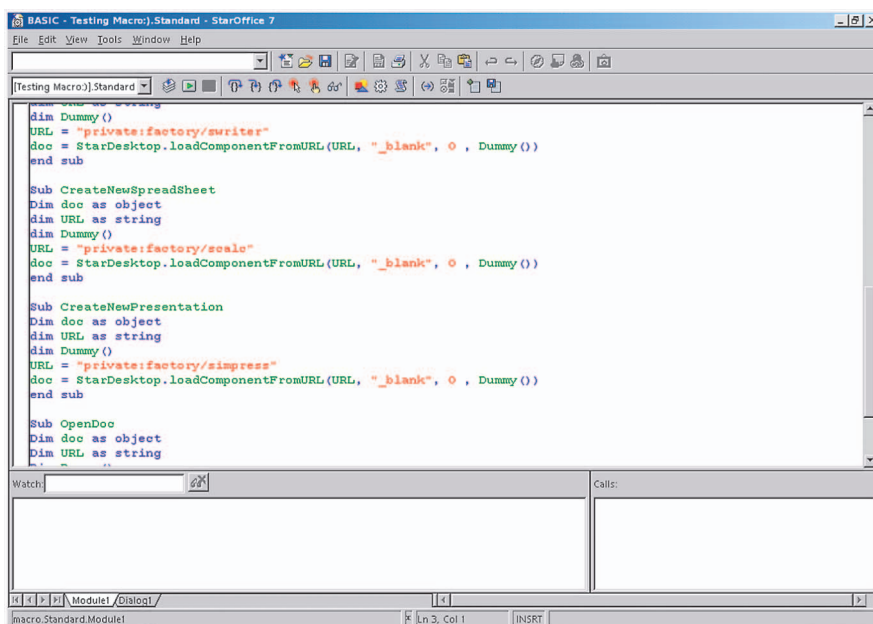
Anda bisa saja membuat modul-modul tersebut menjadi lebih sederhana dengan melakukan passing parameter. Sediakan satu handler untuk ketiga button. Handler tersebut akan menerima parameter berupa nama button tersebut. Parameter tersebut kemudian akan menentukan komponen apa yang akan dibuka. Cara ini lebih sederhana, namun, penerapannya akan lebih susah.

Buat subrutin *SimulateNew* di `Module1` yang berisikan kode-kode berikut ini:

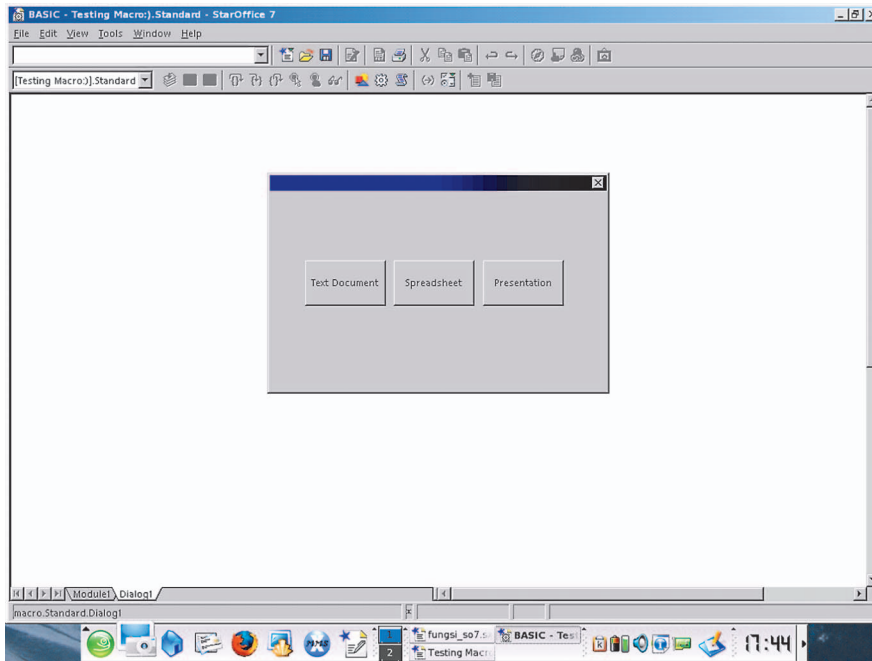
```

Sub SimulateNew
DimDlg As Object
DialogLibraries.LoadLibrary
("Standard")
Dlg = CreateUnoDialog
(DialogLibraries.Standard.
Dialog1)
Dlg.Execute()
Dlg.dispose()
End Sub

```



Layar edit Module.



Layar desain dialog.

Untuk menjalankan Dialog1 tersebut, jalankanlah subrutin *SimulateNew*. Bisa dengan shortcut ataupun penambahan menu item (baca edisi sebelumnya untuk melakukan hal tersebut). Harap diperhatikan bahwa apa saja yang dijalankan dari dialog ini tidak akan bisa ditutup ketika dialog belum ditutup. *Method dispose* pada dialog akan menutup dialog sekaligus mengembalikan memori yang telah dipakai.

Sampai di sini, kita telah mengetahui bagaimana cara StarOffice membuat dokumen baru. Kita akan melanjutkan dengan menyimulasikan membuka dokumen tersimpan.

Membuka dokumen tersimpan

Untuk membuka dokumen tersimpan, sama seperti membuat dokumen baru, kita juga tetap meminta bantuan StarDesktop. Berikut ini adalah contoh kode dan penjelasannya:

```
Sub OpenDoc
Dim doc as object
Dim URL as string
Dim Dummy()
URL = "/tmp/a.sxw"
URL = ConvertToURL(URL)
doc = StarDesktop.loadComponent
FromURL(URL, "_blank", 0,
Dummy())
end sub
```

Perbedaan dengan membuat dokumen baru hanyalah terletak pada URL-nya. Apabila pada pembuatan dokumen baru kita menyebutkan komponen StarOffice, maka kali ini kita menyebutkan nama file-nya secara langsung.

Satu hal yang perlu diperhatikan adalah perihal nama file. StarOffice adalah paket office *multiplatform*. Dengan demikian, nama file adalah hal yang penting. Linux tidak mengenal istilah drive C:. Demikian juga pemisah direktori di Windows berbeda dengan pemisah direktori di Linux. Oleh karena itu, setiap nama file akan dikonversi ke bentuk URL yang standar. Untuk perubahan ini, kita mengenal dua fungsi: *ConvertToURL* dan *ConvertFromURL* dengan fungsi sesuai dengan namanya.

Berikut ini kita akan membuat *user interface* yang lebih baik. Kali ini kita tidak akan membuat dialog sendiri, melainkan akan menggunakan cara yang lebih mudah: menggunakan *inputbox*. Buatlah subrutin ini di dalam Module1.

```
Sub SimulateOpen
Dim doc as object
Dim URL as string
Dim Dummy()
URL = inputbox("Nama
dokumen","Masukkan nama
dokumen","/tmp/a.sxw")
```

```
URL = ConvertToURL(URL)
doc = StarDesktop.loadComponent
FromURL(URL, "_blank", 0,
Dummy())
end sub
```

Panggilah *SimulateOpen* untuk membuka dokumen baru. Untuk saat ini, pengguna perlu menyetikkan *path* lengkap ke dokumen. Ekstensi dokumen bebas dan StarOffice akan menentukan dengan komponen apa dokumen tersebut dibuka.

Inputbox adalah fungsi yang berguna untuk menampilkan layar input kepada user. Bagi Anda yang pernah menggunakan Delphi/Kylix, fungsi ini serupa baik nama dan fungsinya.

Sebagai tambahan dari membuka dokumen dan memahami cara membuka dokumen dengan pengubahan properti, kita akan membuka dokumen secara *read only*. Ketikkan subrutin berikut ini di module1:

```
Sub OpenDoc2
Dim doc as object
Dim URL as string
Dim FileProperties(0) as New
com.sun.star.beans.
PropertyValue
URL = "/tmp/a.sxw"
URL = ConvertToURL(URL)
FileProperties(0).Name =
"ReadOnly"
FileProperties(0).Value = True
doc = StarDesktop.loadComponent
FromURL(URL, "_blank", 0,
FileProperties())
end sub
```

Selama ini, parameter terakhir *loadComponentFromURL* selalu kita isi dengan array kosong *Dummy*. Kali ini, kita akan mengubah satu properti yaitu properti *ReadOnly*. Sebelum itu, kita mendeklarasikan *FileProperties(0)* sebagai instans dari *com.sun.star.beans.PropertyValue*.

Perhatikanlah kedua baris berikut:

```
FileProperties(0).Name =
"ReadOnly"
FileProperties(0).Value = True
```

Pada bahasa lain, akan lebih mudah dipahami apabila dituliskan sebagai berikut:

```
FileProperties(0).ReadOnly =
True
```

Akan tetapi, kita perlu memahami aturan StarOffice yang menuliskan pengubahan properti dengan dua baris tersebut.

Menyimpan dokumen

Sama seperti membuat dokumen baru ataupun membuka dokumen, menyimpan dokumen juga dilakukan dengan bantuan StarDesktop. Berikut ini adalah contoh penyimpanan dokumen. Ketikkan kode ini di dalam bagian subrutin Anda:

```
Dim URL As String
Dim Dummy()
URL="/tmp/a.sxw"
URL = ConvertToURL(URL)
Doc.storeAsURL(URL, Dummy())
```

Mencetak dokumen

Mencetak dokumen dengan bantuan macro adalah hal yang mudah, menarik, dan bagi sysadmin, adalah hal yang perlu diperhatikan. Kita akan membahas soal sysadmin terlebih dahulu. Asumsikan ada salah satu rekan kerja Anda yang nakal. Beliau kemudian membuat virus macro StarOffice yang sangat sederhana: mencetak puluhan kali ke berbagai printer di dalam jaringan. Bisa Anda bayangkan kalau tiba-tiba banyak printer yang sibuk mencetak berkali-kali?

Salah satu cara untuk mengontrol adalah dengan menerapkan *policy secure path list* yang tidak bisa diubah-ubah oleh user dengan mudah. Gunakan *StarOffice Configuration Manager* untuk melakukan pengubahan secara mudah. Harap diperhatikan bahwa SCM tidak dimiliki oleh OpenOffice.org.

Kembali ke pencetakan. Pencetakan dalam bentuk sederhana dapat dilihat pada contoh berikut. Ketikkan bagian kode ini di dalam bagian subrutin Anda:

```
Dim Dummy()
Doc.print(Dummy())
```

Untuk pencetakan yang lebih kompleks, perhatikanlah contoh berikut ini. Contoh berikut akan mencetak halaman tertentu saja:

```
Dim Doc As Object
Dim PrintProperties(0) As New
com.sun.star.beans.PropertyValue
PrintProperties(0).Name="Pages"
PrintProperties(0).Value="1-3;
7; 9"
Doc.print(PrintProperties())
```

Contoh berikut akan mencetak lebih kompleks lagi. Lengkap dengan pemilihan printer.

```
Dim Doc As Object
Dim PrinterProperties(1) As New
com.sun.star.beans.PropertyValue
Dim PaperSize As New
com.sun.star.awt.Size
PaperSize.Width = 20000' 20 cm
PaperSize.Height = 20000' 20 cm
PrinterProperties
(0).Name="Name"
PrinterProperties (0).Value="HP
Laserjet"
PrinterProperties
(1).Name="PaperSize"
PrinterProperties
(1).Value=PaperSize
Doc.Printer =
PrinterProperties()
Doc.print(PrintProperties())
```

Kita tidak membuat user interface dengan dialog kali ini.

Mengubah properties dokumen

Bukalah *File|Properties*. Maka Anda akan dapat mengubah properti dokumen dengan mudah. Dengan mengabaikan dialog tersebut, kita akan melihat bagaimana mengubah properti dokumen dengan bantuan macro.

Berikut ini adalah contoh subrutinnya:

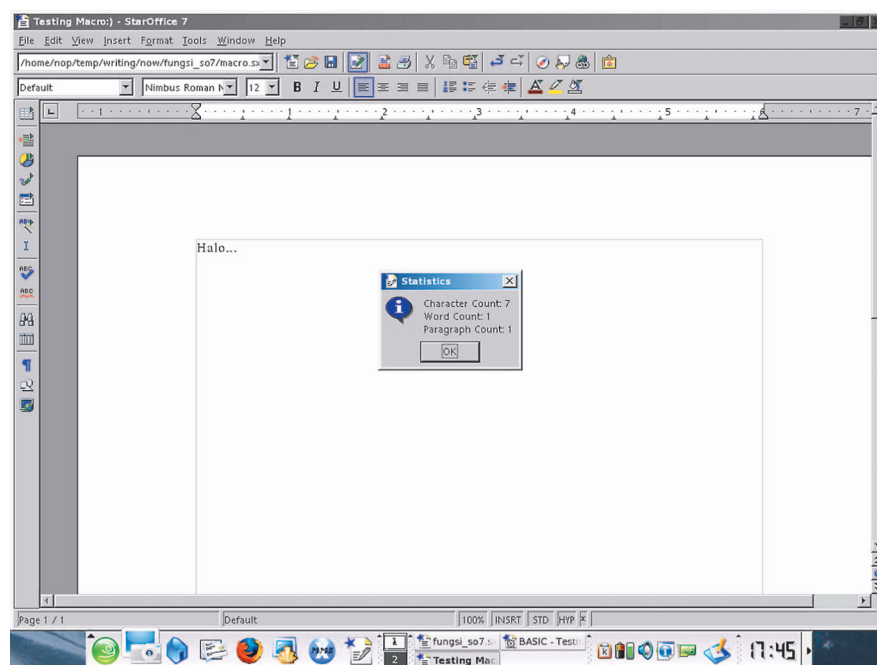
```
Sub ChangeDocProperty
doc = StarDesktop.
CurrentComponent
doc.DocumentInfo.Title =
"Testing Macro:")
End Sub
```

Satu hal yang menarik dari kode tersebut adalah kode *StarDesktop.CurrentComponent*. Kita akan menggunakan cara seperti ini untuk mendapatkan dokumen yang sedang aktif dalam *window*. Berbeda dengan macro pada Microsoft Office yang membedakan dokumen dan *spreadsheet* aktif, StarOffice menerapkan cara yang sama untuk mengambil dokumen aktif. Sangat luar biasa!

Melihat statistik dokumen

Bagi penulis, salah satu fasilitas yang paling sering diakses adalah melihat statistik dokumen. Penulis harus mengetahui berapa karakter yang ditulis, apakah sudah mencukupi persyaratan satu artikel. Untuk mendapatkan fungsi tersebut, kita dapat mengakses menu *File|Properties|Statistics*.

Berikut ini kita akan menampilkan informasi tersebut dengan lebih cepat dengan fungsi statistik buatan sendiri. Ketikkan subrutin berikut ini di dalam Module1.



Statistik dokumen buatan sendiri.


```

Sub MyStat
Dim StatString as String
doc = StarDesktop.
CurrentComponent
StatString = StatString +
"Character Count: " +
doc.CharacterCount + chr(13)
StatString = StatString + "Word
Count: " + doc.WordCount +
chr(13)
StatString = StatString +
"Paragraph Count: " +
doc.ParagraphCount + chr(13)
msgbox StatString,64,
"Statistics"
end sub

```

Berikan *shortcut* agar mudah dijalankan. Secara mendasar, apa yang perlu kita dapatkan adalah properti *CharacterCount*, *WordCount*, dan *ParagraphCount* milik dokumen aktif. Selanjutnya, kita menggabungkan semua informasi dalam satu string, yang kemudian dicetak dengan fungsi *msgbox*.

Parameter 64 menunjukkan dialog berupa *information*. Fungsi ini lebih cepat diakses daripada harus membuka *File|Properties|Statistics*.

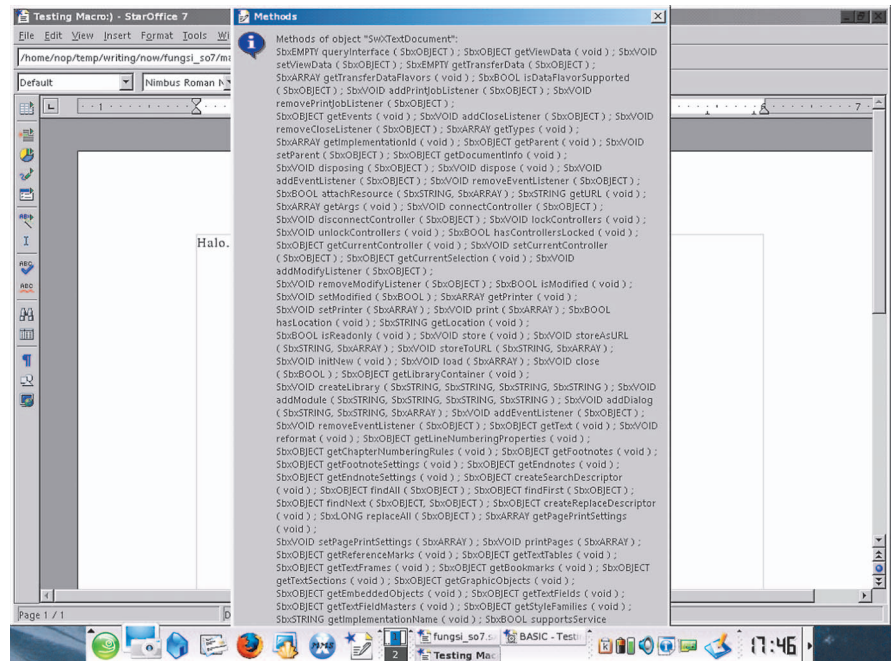
Mengatasi kelemahan Editor Macro StarOffice

Bagi Anda yang pernah bekerja dengan StarOffice dan Microsoft Office, coba bandingkan kemampuan IDE kedua paket office tersebut. Harus diakui bahwa editor milik Microsoft Office jauh lebih enak digunakan.

Begitu kita mengetikkan suatu objek diikuti tanda titik, maka properti dan method yang bisa dipanggil akan ditampilkan. Kita tidak perlu mengetikkan secara manual. Bagi kita yang mudah lupa, hal ini akan terasa sangat membantu.

Sebaliknya, editor untuk StarOffice akan terasa sangat sederhana. Namun, tentu saja kelemahan kecil tersebut tidak sebanding dengan kemampuan paket Office itu sendiri. StarOffice memiliki desain lebih baik, jauh lebih stabil, jauh lebih bisa diandalkan, dan bagi penulis pribadi, jauh lebih menghasilkan.

Pada awalnya, penulis memiliki cukup banyak kesulitan dengan editor yang susah



Mendapatkan method suatu objek.

diajak kerja sama tersebut. Namun, penulis kemudian mendapatkan cara yang sedikit membantu. Yaitu, mencetak keluar properti dan methodnya secara manual dalam sebuah *messagebox*.

Perhatikan contoh berikut:

```

doc = StarDesktop.
CurrentComponent
msgbox doc.dbg_properties,64,
"Properties"
dan
doc = StarDesktop.
CurrentComponent
msgbox doc.dbg_methods,64,
"Methods"

```

Dengan mencetak properti *dbg_properties* dan *dbg_method* ke dalam *messagebox*, kita dapat melihat properti dan method apa saja yang dimiliki oleh suatu objek. Tidak interaktif dan menarik memang, tapi tetap informatif.

Harapannya, dengan demikian kita tidak lagi mengalami masalah dengan mengamati properti dan method suatu objek. Hanya, karena kita mengambil objek aktif, maka hati-hatilah dalam menjalankan macro tersebut. Pastikan Anda menjalankan macro ini di objek yang bersangkutan. Sebagai contoh, untuk mengetahui properti apa saja yang dimiliki oleh komponen *Writer*, jalan-

kanlah di *Writer* dan bukannya tanpa sengaja di layar edit macro.

Untuk mudahnya, macro tersebut disimpan ikut StarOffice dan diberikan *shortcut* umum yang dapat dijalankan dari seluruh komponen StarOffice.

Satu kelemahan cara ini adalah ketika suatu objek menjadi begitu kaya raya akan properti dan method. Dengan demikian, satu layar *messagebox* tidak akan mampu menampilkannya secara tuntas. Gabungan dengan kemampuan menulis pada *Writer* untuk mendapatkan properti dan methodnya ditampilkan dalam dokumen *Writer*.

Salah satu kelebihan StarOffice yang tidak dimiliki oleh OpenOffice.org adalah dokumentasi yang sangat baik. Apabila dengan mengingat properti dan method suatu objek masih tidak membantu, cobalah mengingat dokumentasinya.

Sampai di sini perkenalan kita dengan simulasi fungsi dasar StarOffice dengan macro. Selamat mencoba, dan sampai ketemu lagi dengan berbagai tip macro StarOffice. Kita buktikan bahwa menggunakan Linux bisa jauh lebih produktif dibandingkan dengan menggunakan Windows. Bebas virus dan lebih stabil. Pokoknya luar biasa, apalagi dengan kehadiran kernel 2.6.🐘

Noprianto (noprianto@infolinux.co.id)