

# Apa Sih Sebenarnya...

## TCL?

### Bagian 1 dari 2 Artikel

## Charlie Stross yang merupakan Perl Guru, menanggalkan bahasa pilihannya untuk sedikit bermain dengan bahasa rivalnya.

### Akankah akronim tiga huruf akan berakhir? Apakah Tcl?

Tcl merupakan akronim dari *tool command language*, dan pengucapannya adalah “tikel”. Tcl merupakan bahasa pemrograman *interpreter* level sangat tinggi, seperti halnya Perl, Python, dan Ruby: bahasa yang menitikberatkan pada produktivitas *programmer*, bukannya memaksa pengembang mempermasalahakan *tetek bengek* mengenai arsitektur mesin. Mungkin yang paling pas membayangkan Tcl lebih mirip sebagai Visual Basic yang lahir di dunia Unix. Oh, dan satu hal lagi: Tcl itu lintas *platform*—dengan port-port untuk Windows dan MacOS.

Tapi Tcl bukan hanya itu saja. Sementara Perl dan Python utamanya didesain sebagai bahasa yang *standalone*, Tcl sendiri didesain sebagai sebuah bahasa yang dapat di-*embed*. Idenya adalah jika Anda seorang programmer yang membuat sebuah aplikasi, Anda dapat membuat link antara pustaka Tcl dengan aplikasi tersebut dan (setelah beberapa persiapan) membuat semua program Anda *scriptable* (menggunakan Tcl sebagai bahasa *macro*-nya). Demikianlah, Tcl memasuki sejumlah aplikasi, dari editor teks AlphaTk hingga server aplikasi web milik Interwoven. Kemudian, Tk.

### Kok? Saya kira kita ngebahas Tcl. Apakah semua itu sebenarnya mengenai Tk?

Tk merupakan sebuah *toolkit* grafis. Aslinya dibuat untuk Tcl, Tk telah dibelah menjadi bahasa pemrograman lainnya (seperti Perl dan Python) tetapi umumnya digunakan dengan Tcl, kombinasi keduanya dinamakan sebagai Tcl/Tk. Seperti halnya Tcl, Tk merupakan sebuah pustaka; Tk kaya akan

kumpulan perintah untuk menggambarkan *widget-widget* interface pengguna, dan Tk dapat di-*embed* dalam program-program lain dan telah diport pada MacOS dan Windows (demikian juga dengan X11).

Masih terdapat pustaka lainnya. Tcl membatasi sebuah bahasa inti: biasanya merupakan pustaka yang terpisah, TclX, yang menyediakan perintah-perintah ekstensi untuk berhadapan dengan *resource-resource* pada sistem Unix. Bersama, antara Tcl dan Tk terhubung bersama dalam sebuah shell interaktif yang simpel dinamakan Wish (*the windowing shell*) yang memberikan kemudahan kepada Anda untuk mengetikkan perintah-perintah Tcl/Tk agar secara interaktif membuat prototipe aplikasi grafis. Terdapat shell interaktif yang berbeda yang dinamakan wishX yang bergabung dengan ekstensi-ekstensi TclX, dan lain-lain.

### Yang pasti hal-hal tersebut tidak muncul dari alam gaib, terus langsung jadi dan terdokumentasi?

Tcl awalnya dibuat oleh Professor John Outsterhout dari Universitas California di Berkley pada tahun 1988. Tk muncul kemudian pada 1991, untuk mengisi kebutuhan yang diperlukan. Tujuan awal dari Outsterhout membuat Tcl ini untuk membuat sebuah bahasa *scripting* yang dapat di-*embed*; sementara kegunaan dari baris perintah Unix dapat dengan mudah dihubungkan bersama, pada saat itu belum terdapat aplikasi GUI yang serupa. Ide beliau bahwa aplikasi-aplikasi grafis dengan terembed *interpreter* Tcl akan dapat saling berkomunikasi dengan mengirimkan *event-event*—tidak seperti AppleScript, atau Windows OLE.

Contohnya, sebuah *interpreter* Tcl yang terembed dalam sebuah editor teks akan dapat membawa sebuah *debugger* dengan mengirim perintah-perintah kepada *debugger* tersebut, sementara *debugger* sendiri harus mampu mengirim kembali instruksi-instruksi ke editor untuk melakukan hal-hal seperti *update* posisi *cursor* dalam kode yang sedang di-*debug*.

Ide yang sama akhir-akhir ini sedang digunakan oleh Richard Stallman dan Free Software Foundation, tetapi FSF memilih bahasa ekstensi, GUILE, merupakan sebuah versi dari Scheme (sebuah dialek dari LISP) sementara sintaks-sintaks Tcl lebih familiar bagi para pengguna shell Unix.

Dalam event, membuat aplikasi-aplikasi grafis besar dengan bahasa *embed* merupakan kerja keras: para pengguna Tcl biasanya memfokuskan diri pada penambahan pustaka ekstra dengan perintah-perintah yang melebarkan bahasa inti—maka Outsterhout merubah fokus dan menginvestigasi masalah programming GUI. Toolkit Tk memberikan tampilan dengan level sangat tinggi pada aplikasi GUI dengan menyediakan *widget-widget* seperti beberapa menu, tombol, dan *container* yang dapat diprogram secara prosedur—seperti Tcl, Tk dulunya didesain untuk di-*embed* dalam aplikasi yang dibuat dalam bahasa C atau C++, dan struktur dari interface pengguna dibangun menggunakan Tk akan secara tipikal diprogram dalam Tcl, dengan panggilan-panggilan ke rutin C yang memiliki level lebih rendah untuk menaikkan level.

Pada tahun 1994, Outsterhout bergabung dengan grup peneliti Sun untuk mencari cara melebarkan kegunaan Tcl. Hal ini ternyata merubah menjadi *start-up*

yang terpisah, mengubah nama menjadi Scriptics, diambil alih oleh sistem-sistem Ajuba, dibeli oleh Interwoven (sebuah vendor *framework* aplikasi web) dan kemudian terapung-apung ke dalam lautan *open source*. Kelihatannya terdapat sedikit potensi komersial untuk mendukung sebuah bahasa pemrograman open source seperti Tcl—keberhasilan O-Reilly yang patut dicatat dengan mengenyampingkan Perl—tetapi Tcl mencapai basis pengguna yang luas dan proyeknya sendiri sekarang terdapat pada Sourceforge.

### **Kenapa saya harus melakukan program dengan Tcl/Tk, apakah Tcl/Tk memiliki kelebihan-kelebihan dibanding bahasa-bahasa lainnya bersaing untuk memperoleh ruang dalam kepala saya?**

Tcl/Tk secara komparatif mudah untuk dipelajari. Terutama, Tcl/Tk merupakan sebuah tool yang *powerful* jika Anda ingin membuat aplikasi grafis yang berjalan di atas Windows, MacOS, dan Linux dengan sedikit perubahan. Dan juga, Tcl/Tk merupakan satu cara yang termudah untuk menghubungkan fungsi-fungsi GUI dengan aplikasi yang dibuat dalam bahasa C tanpa harus melakukan pembuatan ulang, dan menyediakan kapabilitas *scripting* pada sebuah aplikasi yang ada. Anda mungkin ingin menggunakan Tcl/Tk pada Linux seperti halnya seorang pengembang Windows akan menggunakan Visual Basic. Dengan tanda yang sama, Tcl/Tk tidak cocok pada beberapa tugas—terlebih pada pengembangan aplikasi-aplikasi berat, atau membuat aplikasi dimana kecepatan memegang peran, atau mendukung kumpulan widget yang berbeda (seperti Qt) adalah tidak dipenuhi.

### **Di mana saya bisa memperoleh Tcl/Tk (untuk Linux)?**

Jika Anda menggunakan distro besar seperti SUSE atau RedHat, Anda kemungkinan sudah memilikinya. Tcl/Tk merupakan produk open source, dan disertakan sebagai standar bagi semua distro besar—yang mana, Tcl berguna sesuai yang diperlukan oleh berbagai macam aplikasi grafis yang umum.

Jika Anda ingin mengompilasi salinan Tcl dan Tk sendiri, terdapat sumber-sumber yang tercatat. Pertama, <http://sourceforge.net/projects/tcl/> dan [\[projects/tktoolkit/\]\(http://sourceforge.net/projects/tclpro/\) merupakan sumber yang pasti untuk kode source yang kasar.](http://sourceforge.net/</a></p>
</div>
<div data-bbox=)

Kedua, perusahaan Scriptics milik Outsterhout mengembangkan sebuah kit yang dinamakan TclPro yang sekarang open source. TclPro sekarang bersarang di <http://sourceforge.net/projects/tclpro/>. (Catat bahwa Anda perlu untuk mengunjungi [http://tcl.activestate.com/ftp/tclpro/download/FREE\\_KEYS](http://tcl.activestate.com/ftp/tclpro/download/FREE_KEYS) demi memperoleh kunci lisensi untuk rilis sebelum 1.5 yang sebelumnya merupakan rilis yang *closed source*).

Ketiga, ActiveState, vendor dari Perl, Python, dan tool-tool pengembangan lainnya untuk UNIX hingga Windows, secara aktif mendukung Tcl; port-port mereka untuk Tcl 8.4 untuk Windows, MacOS, dan Linux/UNIX, tersedia dalam versi kode source dan binary di <http://tcl.activestate.com/software/tcltk/downloadnow84.html> atau melalui ftp di [ftp://tcl.activestate.com/pub/tcl/tcl8\\_4/](ftp://tcl.activestate.com/pub/tcl/tcl8_4/).

### **Bagaimana saya menginstalasi dan menggunakan Tcl?**

Versi kode source hadir dalam dua bentuk kompresi tar. Ekstrak keduanya dan Anda akan menemukan masing-masing berisi sebuah direktori dengan subdirektori “unix”. Buka sebuah jendela shell, jadilah root, masuk ke subdirektori Tcl UNIX dengan perintah `cd`, dan ketikkan `./configure && make install`—perintah ini secara otomatis akan mengkonfigurasi lingkungan bangunan pada mesin Anda, kompilasi Tcl dan lakukan instalasi. Ulangi langkah-langkah tersebut pada direktori Tk, dan Anda akan memiliki sistem Tcl/Tk yang *fresh* untuk digunakan.

Catat bahwa skrip `configure` memiliki banyak opsi yang dapat dikonfigurasi—ketik `./configure—help` untuk menampilkan daftar-daftar opsi, dan lihat file README untuk penjelasan mengenai daftar tersebut.

Ketika Anda menginstalasi Tcl, untuk menambahkan ke sejumlah pustaka yang dapat dihubungkan dengan program-program C Anda, dan file-file header yang menyediakan sebuah interface untuk pustaka-pustaka tersebut, Anda akan mendapatkan sebuah interpreter Tcl—dinamakan `tcl`, dan biasanya terinstalasi dalam `/usr/bin` atau `/usr/local/bin`. Ketik `tcl` pada sebuah prompt shell untuk menjalankan interpreter `tcl`. (Perintah untuk keluar dari interpreter tersebut adalah `exit`.) Catat: jangan tertukar

antara `tcl` dan `tclsh`! Yang pertama merupakan sebuah interpreter untuk bahasa `tcl` yang dapat dipanggil melalui baris perintah. Yang terakhir merupakan sebuah shell UNIX (seperti halnya shell `bash` atau `Zorn`) yang menggunakan Tcl sebagai bahasanya bukannya sintaks-sintaks shell normal.

Tcl memiliki kumpulan halaman man yang ekstensif (manual-manual UNIX) yang dapat diakses melalui perintah `help` (ketik “`help`” dalam interpreter Tcl) atau melalui perintah `man` dalam shell. Tetapi untuk menggunakan Tcl, akan lebih mudah jika sedikit mengetahui tentang bahasa ini...

### **Saya sangat tertarik dan ingin mengetahui lebih lanjut. Adakah buku-buku mengenai Tcl/Tk yang bagus?**

Buku aturan mengenai Tcl adalah Tcl and the Tk Toolkit oleh John K. Outsterhout (penerbit Addison-Wesley, ISBN 0-201-63337-X). Ini merupakan buku asli yang menjelaskan bahasa Tcl/Tk; buku ini berisi sebuah tutorial dan panduan Tcl, sebuah tutorial dan panduan Tk, dan kemudian sebuah bagian yang membahas hubungan Tcl dan Tk terhadap aplikasi C.

Sebagai tambahan, terdapat buku *Practical Programming in Tcl and Tk* oleh Brent Welch—salah satu lulusan Outsterhout dan penulis dari *e-mail client* Exmh (penerbit Prentice Hall, ISBN 0-13-182007-9). Buku ini merupakan tutorial dan pengenalan terhadap pengembangan program dalam bahasa Tcl/Tk yang sangat komprehensif.

Pustaka Expect dari Don Libes merupakan salah satu *add-on* yang paling *powerful* dan berguna untuk Tcl; Expect (dari <http://sourceforge.net/projects/expect>) merupakan sebuah *toolkit* yang berguna untuk mengotomasi sesi-sesi perintah interaktif (contohnya `telnet`, `ftp`, atau perintah apapun yang menantikan input interaktif dari terminal). Expect menggunakan Tcl sebagai interface skripnya; buku Don Libes, *Exploring Expect* (penerbit O'Reilly & Associates, Inc: ISBN 1-56592-090-2) mengandung sebuah pengenalan ekstensif mengenai Tcl (tanpa Tk) dan merupakan teks definitif pada pemrograman Expect. (Dengan sebuah ironi yang menarik, Expect mencapai apa yang merupakan tujuan yang awalnya diimpikan Outsterhout ketika menjelajahi Tcl pada tahun 1988). (LXF20)