

# Python untuk Internet: Seberapa Pantas?

Python sebagai bahasa pemrograman *general purpose* kian berkembang pesat. Lantas, apakah kita juga dapat mengandalkan Python untuk pemrograman Internet?



Ada sebuah pendapat yang cukup radikal di Internet: Di tahun 2020, pemrograman akan dilakukan menggunakan Python dan C. Radikal bukan? Tentunya pendapat tersebut pun ditanggapi secara berbeda-beda oleh berbagai pihak. Terlepas dari opini tersebut, harus diakui, sebagai bahasa pemrograman yang relatif baru, perkembangan Python pantas diberikan acungan jempol.

Perkembangan Python telah meluas ke berbagai sektor pemrograman. Sebut saja mulai dari pemrograman yang cukup mendasar dan berkuat dengan algoritma, pemrograman jaringan, pemrograman berbasis GUI, pemrograman berbasis web, dan berbagai sektor lainnya. Ini adalah fenomena yang sangat menarik. Tentu saja tidak semua sektor pemrograman tersebut dapat mengandalkan Python. Saat ini, kita akan membahas seberapa andal Python digunakan untuk pemrograman di jagad maya tersebut

dan seberapa banyak tool yang telah siap pakai untuk melakukan pemrograman Internet.

## Seberapa cepat?

Salah satu hal yang harus disadari adalah masih mahalanya *bandwidth* di negara kita. Ini adalah kondisi yang susah untuk dikompromikan dengan pemrograman, kecuali dengan mengirimkan data yang seringkas mungkin. Berbeda dengan para administrator jaringan, yang mungkin akan segera mencetuskan penggunaan *mod\_gzip* pada Apache *web server*.

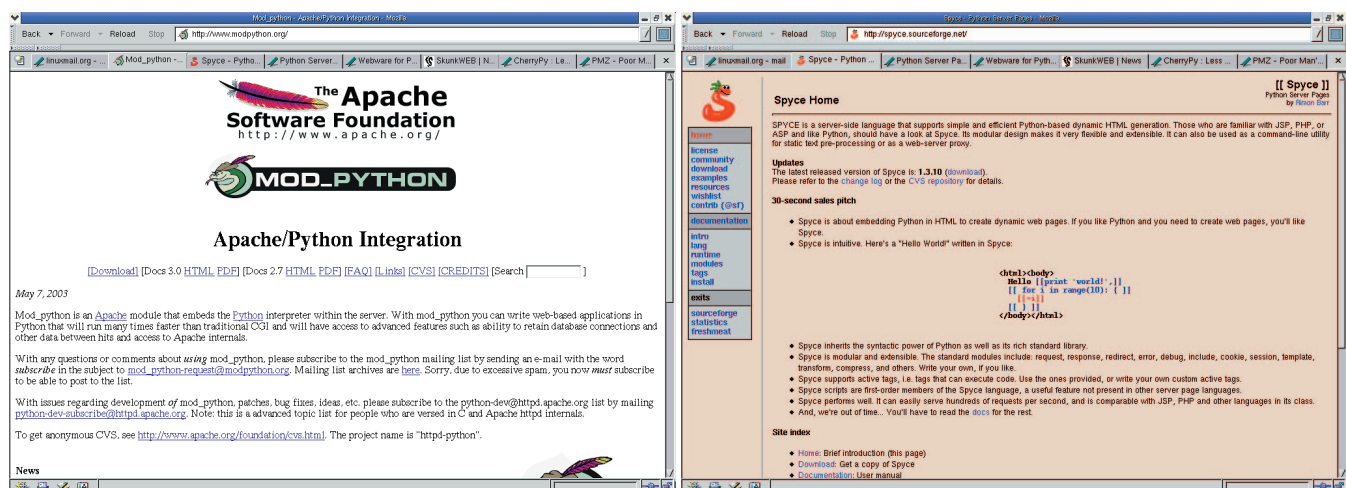
Bagi sang programmer, salah satu hal yang dapat dikerjakan untuk membantu adalah dengan membuat program yang dapat berjalan dengan cepat. Paling tidak, apabila bandwidth kecil, kita tidak menambah lagi penderitaan *user* dengan membuat program yang tidak efisien.

Membuat program yang dapat berjalan cepat tentunya melibatkan berbagai faktor, di antaranya penerapan

algoritma yang benar dan penggunaan bahasa pemrograman yang dapat diandalkan. Kita tidak akan membahas tentang penerapan algoritma karena tentunya akan berkembang menjadi sangat luas.

Apakah Python dapat diandalkan untuk membuat aplikasi yang dapat berjalan dengan cepat? Berikut ini kita akan melakukan pengujian sederhana untuk membandingkan seberapa cepat PHP dan Python (yang diwakili oleh Zope) mampu menghitung jumlah penambahan dari 0 sampai 1 juta - 1. Versi PHP yang digunakan adalah 4.1.2 sementara versi Zope yang digunakan adalah 2.6.1. Untuk web server, PHP menggunakan Apache 1.3.26, sementara Zope menggunakan web server internal. Teks yang dikeluarkan ke *web browser* adalah teks biasa (plain text) dan bukannya HTML. Tool pemercepat untuk Python seperti *psyco* tidak digunakan.

Rasa-rasanya memang tidak adil untuk membandingkan PHP dan

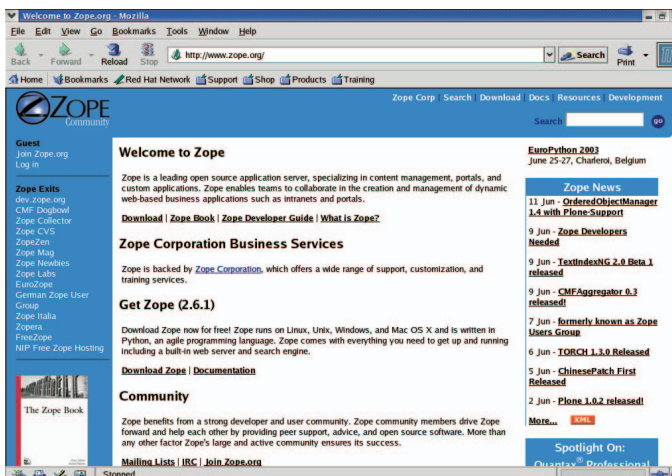


Website mod\_python

Situs spyspy



▲ Situs PSP



▲ Situs Zope

Python dalam kondisi Python diwakili oleh Zope. Di dunia pemrograman web, harus diakui terdapat sangat banyak *tool* pemrograman web yang berbasiskan pada Python, dan Zope adalah satu dari yang paling lambat kalau tidak mau dikatakan sebagai yang terlambat. Akan tetapi, Zope dipertimbangkan cukup lengkap, skalabel dan diperkirakan akan menjadi *framework* pemrograman web berbasiskan Python yang terbesar.

Ketidakadilan lain adalah karena pengujian dilakukan pada perhitungan dimana Python terkenal sebagai jagonya. Akan tetapi, karena perhitungan tersebut adalah perhitungan dasar, maka diasumsikan PHP mampu melakukannya dengan baik. Penulis memohon maaf apabila tes yang dilakukan terkesan kurang mencerminkan keadilan.

Pertama-tama, pengujian dilakukan terlebih dahulu pada PHP sebanyak lima kali percobaan. Setelah itu, barulah percobaan dilakukan pada Zope, juga sebanyak lima kali. Percobaan dilakukan dalam kondisi penggunaan CPU oleh sistem berada di bawah 1%.

Berikut ini adalah hasil percobaan dalam satuan detik:

	PHP	Zope
1	3.00	2.87
2	2.98	2.83
3	2.96	2.83
4	3.01	2.82
5	2.97	2.83

Dari hasil percobaan, kita melihat bahwa Python cukup dapat untuk diandalkan. Akan tetapi, sekali lagi, ini adalah pengujian dalam melakukan perhitungan matematis. Hal-hal lain seperti kecepatan konektivitas database belum dibandingkan sama sekali.

#### Source code PHP:

```
<?
function getmicrotime()
{
    list($usec, $sec) = explode("
",microtime());
    return ((float)$usec + (float)$sec);
}

Header("content-type: text/plain");

$start = getmicrotime();

$sum = 0;
for ($i=0; $i<1000000;$i++) $sum
+= $i;

$finish = getmicrotime();
$delta = $finish - $start;

echo("sum: $sum, start: $start, finish:
$finish, delta: $delta");
?>
```

**Catatan:** Penggunaan fungsi dapat mengacaukan keabsahan pengujian. Akan tetapi karena siklus pada `getmicrotime()` sendiri cukup pendek dan hanya dipanggil sebanyak dua kali,

maka masalah penggunaan fungsi dianggap tidak signifikan.

#### Source code Python Script di Zope:

```
from DateTime import DateTime

start = DateTime().timeTime()

sum = 0l
for i in range(1000000): sum += i

finish = DateTime().timeTime()
delta = finish - start

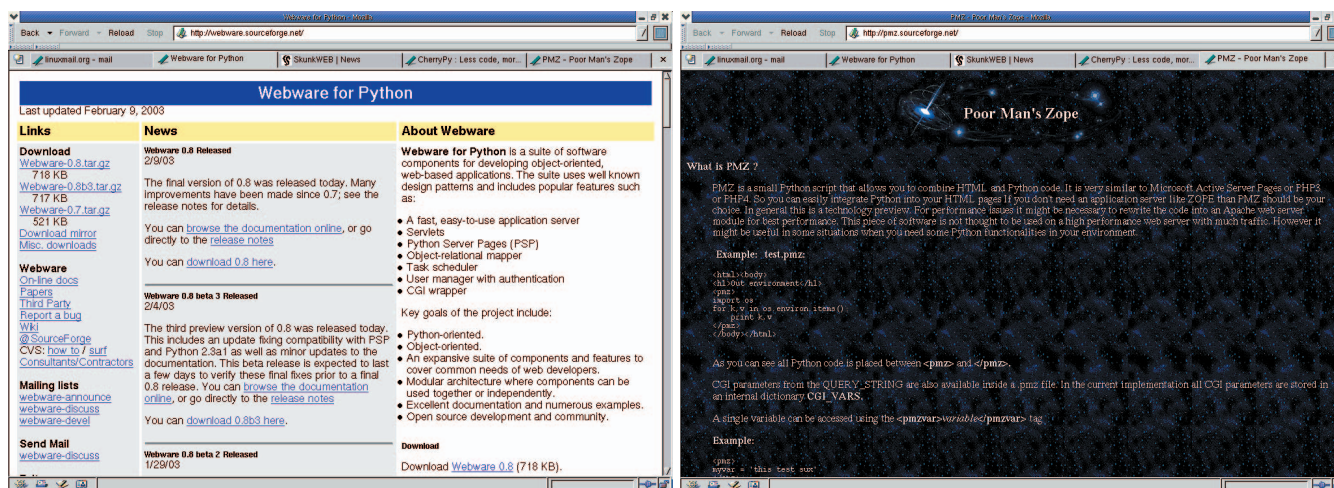
print "sum: %li, start: %f, finish: %f, delta:
%f" %(sum, start, finish, delta)

return printed
```

### Seberapa skalabel?

Ketika kali pertama berkenalan dengan pemrograman, umumnya kita akan membuat program yang sangat sederhana. Begitupun pemrograman web. Mungkin beberapa di antara kita akan berkuat dengan HTML dan tampilan. Beberapa yang lain mungkin akan mendalami *scripting*. Beberapa di antaranya mungkin akan berurusan dengan database dan lain sebagainya. Yang pasti, umumnya apa yang dilakukan adalah pekerjaan yang tidak kompleks.

Setelah konsisten berlatih, kita pun menjadi semakin maju. Apa yang kita kerjakan tidak lagi mengurus kode HTML dan koneksi database sederhana. Sekarang, kita mungkin harus peduli



▲ Situs Webware

▲ Situs PMZ

dengan aplikasi kolaboratif yang berhubungan dengan banyak *user* dan banyak melibatkan *workflow*. Jumlah anggota tim yang semakin besar juga harus memaksa kita dapat bekerja lebih modular. Developer tidak lagi mengurus presentasi. Pembagian tugas pun menjadi lebih jelas. Dalam kondisi tersebut, sanggupkah bahasa pemrograman web yang kita gunakan mampu mendukung? Mampukah Python mendukung kita dalam situasi tersebut?

Jawabannya akan sangat tergantung pada tool yang kita gunakan dalam pemrograman web. Tool pengembangan web berbasis Python sangatlah banyak. Tidak semua tool dapat diandalkan. Beberapa bahkan sangat parah. Beberapa bahkan tidak dikembangkan lagi.

Salah satu tool yang mampu untuk sangat skalabel adalah Zope. Dengan menggunakan Zope, Anda dapat melakukan pemrograman web mulai dari situs kecil dengan sedikit pengguna sampai situs raksasa dengan ribuan pengguna dan database ratusan giga atau lebih. Dan semua hal tersebut tidak memerlukan penanganan khusus. Terapkan saja pemrograman Zope yang baik dari awal, dan Anda pun siap untuk terbang ke sistem raksasa.

Bagaimana dengan *mod\_python*, *poor man's zope*, *spyce*, atau tool sejenis lainnya? Berbeda dengan Zope yang merupakan framework super lengkap, mereka lebih berkutat dengan

fungsi scripting. Oleh karena itu, skalabel atau tidak, sangat tergantung kepada programmer-nya.

Dengan demikian, terhubung tool berbasis Python sangatlah banyak, maka seberapa skalabel Python untuk pemrograman web sangatlah bergantung kepada tool tersebut. Python sendiri sangatlah skalabel mengingat dukungan modul dan pustakanya sangatlah banyak. Ketersediaan berbagai modul dan pustaka membuat kita fleksibel untuk mengembangkan aplikasi yang luar biasa.

Dengan kreativitas, pemahaman Python yang baik dan pemahaman akan bahasa C/C++ misalnya, kita bahkan dapat menulis tool baru. Zope sendiri dibangun dengan Python, sementara urusan yang memerlukan kecepatan tinggi diserahkan kepada C.

## Seberapa banyak pilihan?

Berbeda dengan PHP yang mengkhususkan dirinya pada pemrograman web, Python adalah bahasa pemrograman serba bisa. Dengan demikian, apa yang bisa langsung dilakukan oleh Python untuk pemrograman web adalah penggunaan Python dalam menulis CGI. Dan hal tersebut adalah mimpi buruk. Kenapa?

Pertama-tama, Python sendiri sangatlah lambat. Dengan menggunakan Python untuk menulis aplikasi CGI, kita hanya menggunakan modul-modul standar Python dan fasilitas

standar. Kondisi demikian tidak memungkinkan aplikasi web yang efektif.

Alasan kedua adalah relatif sulitnya pemrograman CGI. Python sendiri sangat mudah untuk digunakan. Akan tetapi, karena harus menuruti aturan pemrograman CGI biasa, walaupun telah dibantu oleh *module* standar cgi, pemrogramannya tetaplah relatif sulit. Akan jauh lebih mudah apabila langsung menggunakan PHP, ASP ataupun scripting lainnya yang memang telah banyak memudahkan pemrograman web.

Dua alasan tersebut telah cukup membuat kita menolak pemrograman aplikasi CGI menggunakan Python. Walaupun mungkin lebih praktis karena kita akan mengetahui persis apa yang kita lakukan, pengorbanan yang harus kita berikan tidaklah sedikit.

Di lain sisi, Zope sendiri sebagai suatu framework yang besar juga bukan selalu menjadi pilihan yang tepat. Boleh jadi, Zope datang dengan Zope Object Database yang luar biasa. Boleh jadi Zope datang dengan serangkaian produk luar biasa yang banyak mempermudah hidup kita. Boleh jadi Zope sangat skalabel dengan Zope Enterprise Object-nya. Akan tetapi, Zope sangatlah kompleks. Belum lagi ditambah sedikitnya perusahaan *web hosting* yang mendukung Zope, terutama di Indonesia. Dan selain itu, Zope relatif susah untuk dipelajari. Ada



banyolan yang mengatakan bahwa dengan zope, *difficult thing easy, easy thing difficult*, hal yang susah menjadi mudah dan terkadang hal yang mudah menjadi susah.

Lantas, apa saja alternatif tool yang bisa kita gunakan apabila kita menghindari kedua hal tersebut? Berikut ini adalah beberapa tool alternatif, yang bahkan tidak kalah populer. Beberapa di antaranya telah digunakan secara meluas di *www* dan dapat disejajarkan dengan PHP ataupun scripting web lainnya.

### 1. mod\_python

*mod\_python* adalah modul untuk *web server* Apache. Dengan menggunakan *mod\_python*, aplikasi web yang Anda tulis dapat berjalan lebih cepat beberapa kali dibandingkan aplikasi CGI biasa (tradisional). Selain itu, konektivitas database pun bukan menjadi masalah yang merepotkan. Dan apabila Anda perlu mengakses internal Apache, tool yang satu ini sangat dapat diandalkan.

Bicara kecepatan dalam merespon *request*, tool yang satu ini, seperti yang telah disebutkan, jauh melebihi aplikasi CGI. Berikut ini adalah perbandingan antara *mod\_python* dan aplikasi CGI menurut dokumentasi resminya di *www.mod\_python.org*:

Standard CGI:	23 requests/s
Mod_python cgihandler:	385 requests/s
Mod_python publisher:	476 requests/s
Mod_python handler:	1203 requests/s

Pengujian dilakukan menggunakan program benchmarking ab.

### 2. Spyce

Tool yang satu ini sangatlah pantas disandingkan dengan PHP, ASP, JSP dan lain sebagainya. Tujuan dari tool yang satu ini adalah *server-side* scripting berbasis Python yang mudah dan efisien. Sama seperti PHP, tool yang satu ini dapat diselipkan di dalam kode html. Semua program *spyce* berakhiran *.spy*.

Bagaimana dengan kecepatannya? Berikut ini adalah data hasil pengujian



▲ Situs Skunkweb

yang dikutip dari dokumentasi resminya di *spyce.sourceforge.net*:

Konfigurasi	Hello world (per detik)
Spyce-modpython	250
Spyce-proxy	180
Spyce-FCGI	100
Spyce-CGI	5
JSP	100
PHP	450
Python-FCGI	140
C-CGI	180
Python-CGI	25
Static HTML	1500

Konfigurasi dengan menggabungkan *spyce* dan *mod\_python* dapat meningkatkan kecepatan sampai hampir 10 kali dibandingkan aplikasi CGI biasa.

Pengguna *spyce* sendiri perlu untuk memahami tujuan dari desain *spyce*, karena beberapa diantaranya cukup mengejutkan. Salah satunya adalah *spyce* didesain untuk miskin fitur. Ya, dengan tidak memberikan berbagai fitur yang terkadang menjadikan suatu tool rakus sumber daya sistem, *spyce* diharapkan dapat membantu Anda dalam menulis aplikasi yang efisien.

Berikut ini adalah contoh kode *spyce*:

```
<html><body>
Hello [[print 'world!']]
```

```
[[ for i in range(10): { ]]
```

```
[[=i]]
```

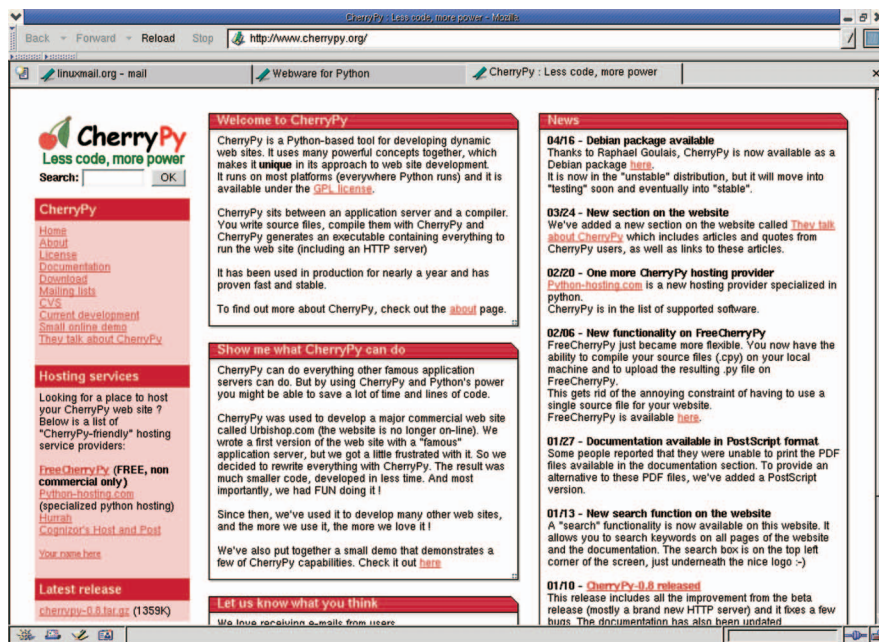
```
[[ } ]]
```

```
</body></html>
```

### 3. PSP (Python Server Pages) dan Webware

Dengan kemiripan namanya dengan beberapa *server-side scripting* populer, PSP datang dan menawarkan kemudahan dalam membuat aplikasi web. Beberapa sintaks dalam PSP pun sangat mirip dengan ASP. PSP sendiri adalah *server-side scripting* yang didesain untuk mirip dengan ASP dan JSP. Satu hal yang sangat mengejutkan adalah PSP dibangun dengan 100% Java! PSP sendiri menggunakan Jpython sebagai bahasa scripting-nya. Hal ini menjadikannya jauh lebih mudah dibandingkan JSP. Semua program PSP berakhiran *.psp*.

Keuntungan yang ditawarkan oleh scripting yang satu ini adalah kemudahan dan dukungan berbagai modul yang kompatibel dengan Jpython. Dan karena Jpython sendiri ditulis menggunakan Java, maka dengan PSP, Anda dapat mengakses paket-paket Java! Luar biasa! PSP bisa di-download di *http://www.ciobriefings.com/psp/*.



Website CherryPy

Berikut ini adalah contoh kode PSP:

```
<HTML>
<HEAD>
<TITLE>Hello World </TITLE>
</HEAD>
<BODY>
<H1>Hello World </H1>
<BR>
$[
Response.write( "Hello, %s, from Python
Server Pages." % (Request.server
["REMOTE_ADDR"]) )
]$
<BR>
</BODY>
</HTML>
```

Jika Anda menginginkan suatu *suite* untuk membangun aplikasi yang menggunakan PSP, maka *Webware for Python* adalah salah satu yang sangat terkenal. Webware sendiri adalah kumpulan komponen *software* untuk membangun aplikasi web yang berorientasi objek. Berikut ini adalah fitur dari webware:

- Server aplikasi yang cepat dan mudah digunakan.
- Servlets.
- Penggunaan Python Server Pages.
- Penjadwalan tugas.
- User manager.
- Wrapper CGI.

Tujuan desain dari Webware di antaranya adalah aplikasi yang berorientasi Python (dan objek), lengkap dengan komponen-komponen yang dibutuhkan oleh para *developer* web dan dokumentasi yang lengkap. Webware bisa di-download di <http://webware.sourceforge.net/>.

#### 4. Poor Man's Zope

Zope termasuk salah satu server aplikasi yang hangat dibicarakan di Internet. Salah satu keengganan untuk menggunakan Zope adalah borosnya penggunaan sumber daya sistem. Hal tersebut sebenarnya cukup wajar apabila dibandingkan dengan fitur yang datang bersamanya.

Bagaimana dengan server yang memiliki *resource* terbatas? Bagaimana dengan para programmer web yang tidak memerlukan kesuruhan fitur dari Zope? Poor Man's Zope adalah jawabannya. Fungsi PMZ dapat disejajarkan dengan PHP ataupun bahasa scripting lainnya. Walau, untuk saat ini, sangat susah untuk menggunakan PMZ di lingkungan kerja produktif. Fitur yang ditawarkan masih sangat sedikit. PMZ dapat di-download di <http://pmz.sourceforge.net/>.

Terlepas dari mininya PMZ untuk saat ini, dukungan XML tetap menjadi

prioritas. Aplikasi yang ditulis dengan PMZ adalah aplikasi yang siap untuk bekerja dengan XML. Semua program PMZ berakhiran *.pmz*.

Berikut ini adalah contoh kodenya:

```
<html> <body>
<h1> Out environment </h1>
<pmz>
import os
for k,v in os.environ.items():
    print k,v
</pmz>
</body> </html>
```

#### 5. Skunkweb

Aplikasi yang satu ini juga merupakan server aplikasi, seperti halnya Zope. Tujuan dari Skunkweb sendiri adalah untuk menyediakan lingkungan pemrograman yang mudah dan cepat untuk membangun aplikasi web. Dalam perbandingan sederhana, Skunkweb dapat disejajarkan dengan berbagai scripting lainnya.

Dalam beberapa hal, tujuan dari Skunkweb sangatlah mirip dengan Webware dan Zope. Berbeda dengan Webware dan Zope, Skunkweb mengklaim dirinya lebih skalabel dengan mampu menggunakan keseluruhan sumber daya sistem lebih baik. Skunkweb menggunakan model *forking*, dan berbeda halnya dengan Zope dan Webware yang menggunakan *threading*.

Fitur utama yang datang bersama Skunkweb adalah penggunaan Python. Dapat dikembangkan, menggunakan sistem komponen, *Message catalog*, kecepatan, dan dokumentasi.

Tool yang satu ini banyak mengklaim dirinya lebih baik dari tool lain, mulai yang besar seperti Webware dan Zope, sampai yang berdiri sendiri seperti PHP. Penasaran mencoba? Kunjungi situsnya di [skunkweb.sourceforge.net](http://skunkweb.sourceforge.net).

#### 6. CherryPy

Tool yang satu ini termasuk tool baru (kali pertama diluncurkan pada musim panas 2002). CherryPy sendiri pada dasarnya adalah pre processor Python yang mengizinkan pemakainya untuk

menggabungkan kode Python dan HTML.

Walau termasuk baru, tool yang satu ini juga datang bersama dengan *templating language* sendiri: *Cherry HTML Template Language* (CHTL) dan *Cherry Generic Template Language* (CGTL). Kedua templating language ini dapat digabungkan bersama dengan HTML ataupun XML. CherryPy dapat di-download di <http://www.cherrypy.org/>. Selain CherryPy, yang datang dengan templating language adalah Zope (*Zope Page Template*). CherryPy sendiri berada di antara server aplikasi dan kompiler. Pengguna Cherry menulis *source* program, kemudian dikompilasi dengan CherryPy dan sebuah program yang siap jalan pun dihasilkan.

Berikut ini adalah contoh kode CHTL:

```
<p>Hello <b py-eval="self.getUsername()>Example User</b>
```

Di luar semua yang telah dibahas, terdapat masih sangat banyak tool

pengembangan web berbasis Python. Masing-masing berlomba-lomba untuk menawarkan kemudahan dan fitur-fitur lainnya. Bagi kita sebagai pemakai akhir, tentunya ada beberapa hal yang harus diperhatikan.

Fleksibilitas umumnya adalah hal yang perlu sekali untuk mendapat perhatian. Python sendiri cukup fleksibel, namun hal tersebut bukan sebuah jaminan untuk tool pengembangan web berbasis bahasa tersebut. Umumnya, yang pasti, tool-tool tersebut menawarkan penggunaan sintaks Python yang bersih dan jelas. Terlepas dari kemudahan yang didapat dari kesan pertama, perlu juga diperhatikan seberapa fleksibel tool tersebut.

Selain fleksibilitas, kecepatan adalah hal yang rasa-rasanya didambakan oleh setiap pihak. Akan tetapi, cepat tanpa fitur pun bukanlah hal yang terbaik. Umumnya, kecepatan harus dikorbankan dengan kurangnya fitur atau susahny pemrograman.

Datang bersama banyak fitur dan mudah dipakaipun bukanlah nilai mati.

Kita juga tetap perlu untuk memperhatikan seberapa matang tool tersebut. Persoalan di sini bukanlah hanya sekedar memiliki banyak *bug* atau tidak, akan tetapi lebih kepada desain tool. Desain yang jelek dapat membuat sebuah tool berubah di kemudian hari. Hal yang satu ini jelas menjengkelkan karena sebagai pengguna, kita harus mengubah lagi program yang telah dibuat sekian lama. Memilih tidak mengubah? Umumnya Anda tidak akan mendapatkan fitur terbaru.

Adapula faktor lain yang cukup penting. Beberapa tahun lalu, penulis memutuskan untuk menggunakan *mod\_snake* daripada *mod\_python*. Hasilnya? Beberapa saat yang lalu, proyek *mod\_snake* telah ditutup oleh pembuatnya. Film selesai, tidak ada cerita lagi. Untungnya, penulis telah beralih ke PHP dan Zope sebelum kecewa terlalu dalam. Bagaimana masa depan tool tersebut rasa-rasanya sangat perlu diperhatikan. Memang pada dasarnya, sebuah proyek dapat bertahan cukup lama, terutama apabila developer-nya terdiri dari berbagai kalangan, dan alangkah bagusny apabila ada dukungan dari perusahaan lain.

Dan terlepas dari semua pertimbangan tersebut, manakah yang paling sesuai dengan kebutuhan Anda? Membuat sebuah aplikasi sederhana menggunakan Zope bukanlah pilihan yang bijak. Zope terlalu berat dan rumit. Membuat aplikasi yang berat dengan *mod\_python* pun rasanya terlalu naif. Dan yang cukup penting, apakah Anda menyukai tool yang Anda gunakan? Dengan semakin menyukai, tentunya semangat dapat bertambah.

Masih banyak lagi hal yang perlu dibahas untuk menguji seberapa pantas python digunakan untuk pemrograman Internet. Dunia *open source* semakin bertambah luas. Anda pun bahkan bisa memperluas dunia tersebut. Tulisan ini tidak dimaksudkan untuk mengagungkan suatu bahasa tertentu dan mendiskreditkan bahasa lainnya. *Happy hacking!*  
Noprianto ([inoprianto@infolinux.co.id](mailto:inoprianto@infolinux.co.id))

Dapatkan...

# Discount 18%

Selama bulan promosi Mei-Agustus 2003,  
kami memberikan penawaran discount  
12% untuk pembayaran paket hosting 1 tahun  
18% untuk pembayaran paket hosting 2 tahun

**Hosting Windows 2000 dengan harga Linux** <<  
**Neo Lite - 100 MB Data Transfer 10 GB - @Rp 49.500,-** <<  
**Neo Medium - 500 MB Data Transfer 20 GB - @Rp 149.500,-** <<  
**Neo Lite - 800MB Data Transfer Unlimited - @Rp 349.500,-** <<  
**Co-location Hosting - Dedicated Hosting** <<

**Neohosting.biz** <<  
 Jl. Let. Jend. S. Parman  
 Kavling H-9 Kompleks BRI Slipi  
 Jakarta 11480 Indonesia  
 Phone (021) 536 76 111  
 Fax (021) 536 76 333

Toll free technical support  
**0800. 12. 99999**

**www.neohosting.biz**  
 group of : **neovnet**

Microsoft  
**.net**