

## Bagian Pertama

# Bochs x86 Emulator: Hadiah Komunitas Open Source untuk Programmer OS dan Device Komputer

**Hendro Subagyo**

hendro-s@casl.cs.uec.ac.jp

### *Lisensi Dokumen:*

*Copyright © 2003 IlmuKomputer.Com*

*Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.*

## Berkenalan dengan Bochs

Bagi Anda para pemula programmer OS atau yang ingin belajar lebih dalam OS atau device driver programming, maka Bochs akan menjawab peningnya kepala Anda atau kekhawatiran Anda akan bahaya rusaknya komputer Anda karena bug di software yang Anda buat. Bagi Anda yang sedang belajar assembler, Bochs pilihan yang tepat untuk mencoba program Anda. Bagi Anda yang sedang belajar OS, Anda dapat mempelajari step by step setiap proses booting sebuah OS.

Bochs (dibaca box) adalah sebuah software yang mengemulasi PC (personal computer) keluarga prosesor x86 (IA-32) secara virtual. Ia mengemulasi prosesor, BIOS, VGA board, IDE HDD, CDROM, PS/2 keyboard, PS/2 Mouse dll.

Versi terbaru Bochs adalah 2.0.2 yang dapat Anda download dari <http://bochs.sourceforge.net/>.

Bochs versi ini dapat dicompile untuk mengemulasi prosesor 386, 486 atau Pentium. Bochs dapat menjadi emulator untuk menjalankan sebagian besar OS termasuk Linux, FreeBSD, OpenBSD, BeOS, DOS, MAC OS, Hurd, Windows 95, Windows NT. Proyek terbaru Bochs saa ini adalah agar ia dapat pula mengemulasi OS Mona dan EOTA. Mona adalah sebuah OS baru yang dibangun dengan menggunakan Bochs sebagai testbed sejak dari awalnya Sedangkan EOTA adalah OS keturunan B-Free OS (sebuah OS yang compatible dengan BTRON). Bila Anda menggunakan mesin non-x86, maka Bochs dapat digunakan untuk menjalankan x86 software di atas mesin Anda. Saat ini Bochs dapat berjalan di atas Solaris (Sparc), Linux (PowerPC/Alpha), MacOS (PowerPC), IRIX (MIPS), BeOS (PowerPC), Digital UNIX (Alpha) dan AIX (PowerPC).

## Emulator vs Virtual Machine

Dengan Bochs, Anda dapat menjalankan sebuah OS lain (OS tamu) diatas host OS, yaitu OS sebenarnya yang jalan di atas komputer kita. Misalnya Anda dapat menjalankan Linux (OS tamu) di atas Windows (host OS), atau sebaliknya.

Barangkali Anda pernah mendengar VMWare atau Virtual PC, yang memberikan efek yang sama. Tetapi *virtual machine* dan *emulator* merupakan barang yang berbeda.

*Emulator* adalah software yang menjalankan secara seluruh perintah-perintah sebuah prosesor secara software. Jadi *emulator* mengemulasi seluruh perintah-perintah prosesor. Sedangkan *Virtual Machine* mengeksekusi perintah umum prosesor apa adanya, dan mengemulasi perintah-perintah khusus serta perintah-perintah yang berkaitan dengan input-output, untuk membuat sebuah komputer virtual.

Dari kedua prinsip yang berbeda tersebut, dapat kita ketahui bahwa dilihat dari kecepatannya maka tentu saja *virtual machine* lebih cepat dari *emulator*. Keunggulan *emulator* adalah ia memiliki portabiliti tinggi, sehingga ketergantungan kepada jenis CPU atau komputer sangat rendah. Sebagai contoh, Bochs yang berjalan di atas mesin dengan CPU x86, dapat mengemulasi mesin AMD x86-64. Sedangkan Bochs yang berjalan di atas mesin UP (Uni-Processor), dapat mengemulasi mesin SMP (Shared-Memory-Processors). Akhir-akhir ini bahkan kita dapat menemukan sebuah mesin Linux Zaurus dengan prosesor StrongARM, dapat menjalankan Windows 98. Jadi dengan adanya *emulator* semacam Bochs, di masa depan, software-software kita yang ada sekarang akan dapat terus hidup.

Namun yang terpenting, setidaknya bagi saya, karena Bochs adalah open source software, yang berarti pula ia adalah free software. Jadi Bochs murah. Sedangkan *virtual machine* yang ada sekarang kebanyakan bukan freeware dan mahal. ;-)

## Bochs Berguna untuk Siapa?

Bochs dapat digunakan untuk berbagai keperluan. Sebagian orang menggunakannya untuk menjalankan aplikasi di OS kedua, tanpa memerlukan dua komputer atau dual-booting. Misalnya, sebagian besar pemakai memakai bochs untuk menjalankan software-software Windows di atas workstation no-x86 atau di atas sebuah x86 UNIX.

Bochs menyediakan fasilitas untuk melihat setiap baris dari instruksi hardware dan setiap baris dari kode simulator dapat diakses secara mudah, sehingga Bochs sering digunakan untuk mendebug (debugging) OS baru. Jika Anda menulis sebuah kode booting untuk x86 OS anda dan ia tidak bekerja, maka dengan memboot OS Anda itu diatas Bochs, Anda dapat melihat apa yang sedang terjadi.

Bochs juga telah banyak digunakan untuk kelas-kelas OS, dimana siswa dapat menggunakan dan memodifikasinya sehingga mereka belajar bagaimana cara kerja hardware PC. Siswa dapat bereksperimen dengan membuat satu device baru sehingga mereka belajar cara kerja IO port, interrupt dan device driver. Di industri, Bochs digunakan untuk mensupport *legacy application* di atas hardware modern, dan sebagai model referensi untuk mentest hardware x86-compatible baru.

Anda dapat pula menjalankan Bochs sekedar untuk menjalankan program DOS lama Anda, atau game DOS Anda. Atau bahkan untuk keperluan bagaimana cara membuat program di Linux tanpa keluar dari Windows Desktop. Kemungkinan lain? Tergantung daya kreasi Anda ;-))

Bila interest Anda adalah kecepatan dan performance, sedangkan Anda tidak ingin melakukan tweaking pada beberapa file, maka Bochs bukan untuk Anda. Anda dapat menggunakan Plex86 (Bochs berasal dari Plex86), atau VMWare, Wine dsb.

Tetapi bila interest adalah untuk mendebug software dan hardware, maka Bochs untuk Anda. Atau bila Anda ingin menjalankan software yang hanya jalan di komputer dengan hardware lama, maka Bochs untuk Anda pula. IT adalah bidang yang tumbuh tercepat dibandingkan bidang lain. Kita dapat dengan mudah melupakan software-software yang datang dan pergi di masa transisi. Tetapi sejarah sangat penting bagi seluruh bidang kehidupan, tidak terkecuali IT. Untuk membangun masa depan, kita perlu

memahami masa lalu. Programmer komputer tidak sama dengan arsitek yang dapat dengan mudah pergi ke Mesir untuk melihat piramida, atau mengunjungi kota-kota tua. Dengan Bochs Anda dapat kembali ke masa lalu IT.

Untuk siapa:

Hacker/programmer OS

- Anda tidak perlu khawatir akan merusak hardware ataupun system benaran
- Tidak perlu remote-debugging
- Konfigurasi mudah
- Portable (jalan di Windows, Linux, dll)

- Siswa yang belajar OS atau assembler

Bochs memang tepat untuk development sebuah OS, atau men-debug OS, juga tepat untuk alat bantu belajar mengajar OS.

## Spesifikasi

Spesifikasi	Support	Keterangan
Configure script	Ya	Bochs memakai GNU autoconf untuk mengkonfigurasi Makefile dan header. Dengan Autoconf Bochs dapat dikompilasi ke berbagai macam platform
Emulasi 386, 486, Pentium	Ya	Bochs dapat dikonfigurasi untuk mengemulasi hardware keluarga Intel. Beberapa karakter Pentium telah tersupport.
Emulasi Pentium Pro	Belum komplit	Sedikit karakter dari Pro Pentium telah tersupport, seperti on-chip APIC untuk simulasi SMP
Cmd Line Debugger	Ya	Tersedia powerful command line debugger, sehingga Anda dapat menstop eksekusi dan melihat kondisi register dan memory, menset breakpoint dll
Floating Point	Ya	Menggunakan floating point library dari Bill Metzenthen
Enhanced BIOS	Ya	Implementasi spesifikasi EITorito, EDD v3.0, PCI32 directory service
VGA	Ya	Emulasi VGA color graphic dalam sebuah window
VESA (display) support	Ya	Saat ini mampu mensupport sampai 1024x786x8pp.
Floppy disk	Ya	Mensupport 1.44M 3.5", 1.2M 5.25" dan 720K 3.5". Di atas Unix dan Windows XP/NT/2000, Bocsh dapat mengakses physical floppy drive
Multiple ATA channels	Ya	Mengemulasi sampai 4 buah ATA channel. Tiap satu ATA channel dapat mengemulasi 2 ATA/ATAPI device. Sehingga Anda dapat memiliki 8 hardisk, atau 7 hardisk + 1 CDROM, atau 4 hardisk + 4 CDROM, dst.
Hard disk	Ya	Mengemulasi ATA-2/IDE harddrives via image file. Maximum sampai 32GB.
CDROM	Ya	Mengemulasi ATAPI-4/IDE CDROM. CDROM dapat membaca ISO disk image. Di atas Windows (95/98/NT/2000), Linux, SunOS, FreeBSD, NetBSD, Amiga/MorphOS, dan BeOS, Bochs dapat membaca physical cdrom. Sejak versi 1.4, Bocsh dapat diboot dari bootable cd ata bootable iso image.
Keyboard	Ya	Mengemulasi PS/2.
Mouse	Ya	Mengemulasi PS/2 Mouse dengan 2 button
Sound Blaster	Ya	Mengemulasi Sound Blaster 16. Di atas Windows, Linux dan FreeBSD, Anda dapat mengirimkan output ke sound system host computer. Di atas MacOS, hal ini belum tersupport.
Network Card	Ya	Mengemulasi NE2000 compatible network card. Di atas Windows NT/2000. Linux, FreeBSD, dan NetBSD, Bochs dapat memforward packet ke dan dari OS sehingga OS tamu dapat berkomunikasi dengan physical network. Sayangnya, pada sebagian platform, OS tamu dapat bercakap dengan mesin-mesin lain di network, tetapi tidak dapat bercakap dengan host OS.
Parallel Port	Ya	Data yang terkirim ke parallel port oleh OS tamu dapat disimpan ke sebuah file atau dikirim langsung ke parallel port device (khusus Unix)
Serial Port	Ya	Serial port (singge 16450 UART emulation), minimal dengan linux sebagai host dan tamu.

PCI	Tidak komplit	Tersedia Host-to-PCI bridge dan Primary Memory Controller. Tapi PCI-to-IDE, PCI-to-USB, dsb, belum terimplementasi.
Plugins	Ya	Untuk Linux, MacOS X, Solaris dan Cygwin tersupport
16/32 bit addressing	Ya	16 atau 32 bit untuk operand size, stack size dan addressing
V8086/paging	Ya	Virtual-8086 mode dan paging
PIC	Ya	Programmable Interrupt Controller Master dan Slave
Fungsi-2 CMOS	Ya	
Dynamic Translation/ Virtualization	Tidak	Karena Bochs didesain agar portable.
Mensimulasi sebuah Multiprocessor	Ya	Bochs dapat dikonfigurasi sampai 15 processor.
Mengambil keuntungan dari SMP box Anda	No	Karena saat ini Bochs tidak menggunakan thread atau parallel processing, sehingga ia tidak dapat jalan lebih cepat diatas multiprocessor hardware
Copy and Paste	Ya	Anda dapat copy and paste text didalam Text-Mode screen.

## Referensi

- [1] Bochs, <http://bochs.sourceforge.net>
- [2] Proyek OS Mona, <http://sourceforge.jp/projects/mona>
- [3] Proyek EOTA, <http://www.rbt.his.fukui-u.ac.jp/~naniwa/comp/OS>