

Kompilasi kernel 2.4.x di Linux

Asfihani (asfik@cs.perbanas.edu)

08 Juni 2003

Dokumen ini dibuat sebagai materi workshop internal pada Kelompok Studi Linux di STIMIK Perbanas Jakarta. Tujuannya adalah sebagai bantuan maupun referensi kepada anda yang akan melakukan update/upgrade kernel 2.4.x di linux secara step by step dengan menggunakan sistem kernel modular dan built-in (monolitik). Kompilasi kernel menggunakan kernel 2.4.20 dilakukan pada distro Redhat 8.0 (2.4.18-14), AMD Duron 1 Ghz, 256 Mb RAM, 40 GB IDE harddisk. Anda bebas mendistribusikan, menyalin, mencetak, menggandakan, maupun memodifikasi dokumen ini (dengan menyertakan nama penulis asli). Saran, kritik, koreksi, maupun "flame" silakan dikirim ke alamat diatas atau IRC EFnet channel [#indolinux](#) . Terima Kasih.

1 Pendahuluan

1.1 Tentang kernel

Seperti yang telah kita ketahui bersama bahwa Linus Torvalds adalah penulis dan pengembang sebuah sistem operasi berbasis Unix yang disebut Linux. Pada suatu sistem Linux, kernel merupakan penghubung (jembatan/mediator) antara suatu program (aplikasi) dengan hardware seperti misalnya dalam hal manajemen memori untuk proses-proses yang sedang berjalan, input-output terhadap dan dari suatu device dan masih banyak lagi fungsi tambahan yang lainnya. Intinya adalah kernel merupakan suatu penghubung (antara software dan hardware). Walaupun fungsi kernel yang lain masih banyak, tapi saya tidak membahasnya disini, paling tidak anda sudah tahu fungsi dasar dari sebuah kernel :)

1.2 Mengapa harus update/upgrade kernel?

Ada 1001 alasan anda harus (baca: ingin) mengupdate/menguprade kernel, antara lain (don't take it personally) :

1. Anda menginginkan optimasi terhadap sistem anda
2. Anda tidak puas dengan kinerja kernel bawaan dari vendor distro anda
3. Karena aplikasi yang anda gunakan membutuhkan kernel yang lebih baru
4. Karena hardware yang baru saja anda beli tidak disupport oleh kernel yang anda pakai sekarang
5. Karena kernel yang lama ada *bug*
6. Karena kernel yang baru (menurut anda) menawarkan *fitur* yang lebih bagus (misalnya manajemen memory, manajemen proses, dukungan driver dan lain lain)
7. Anda mendapatkan PR kompilasi kernel dari suatu mata kuliah
8. Anda ingin pacar anda tahu bahwa anda bisa melakukan kompilasi kernel
9. Anda habis putus dengan pacar anda dan pada malam minggu tidak anda kegiatan yang lain

10. Anda iseng, senang-senang, untuk mengeteset kecepatan CPU
11.[tuliskan alasan anda disini]

Meskipun demikian, ada pepatah di internet yang bunyinya "If it ain't broken, don't fix it" ...and you have been warned :)

1.3 Modular vs builtin (monolitik)

Sebelum mengkompilasi kernel, ada baiknya anda menentukan terlebih dahulu, kira-kira format kernel yang bagaimana yang sesuai dengan kebutuhan anda. Sebagai contoh jika anda sering gonta-ganti hardware, sistem kernel yang *modular* akan lebih cocok daripada sistem kernel yang *builtin (monolitik)*. Sementara ini keuntungan dan kerugian yang saya tahu adalah seperti dibawah, jika anda lebih tahu, saya akan sangat senang untuk menambahkannya disini :)

1.3.1 Kernel modular

Salah satu keuntungan kernel yang bersifat modular, gonta-ganti hardware menjadi lebih mudah, karena anda tinggal menge-*probe* suatu modul, atau jika belum ada anda tinggal mem-*build* satu modul saja. Kerugiannya adalah relatif rentan terhadap masalah security, karena biasanya *script kiddies* memasukkan suatu modul ke dalam kernel (dengan harapan proses yang dimilikinya tidak diketahui oleh admin sistem yang bersangkutan)

1.3.2 Kernel builtin (monolitik)

Dari segi security, sebuah kernel builtin (monolitik) akan relatif aman. Namun dari segi kemudahan, jika anda menambah atau mengganti suatu hardware, maka otomatis anda harus mengkompilasi ulang kernel anda

Namun demikian, skema kernel yang bagaimana menurut anda lebih sesuai, anda sendiri yang menentukan :), link terkait tentang design dan implementasi kernel :

- <http://www.fw.cz/mitr/microkernel.html>
- [http://www.wikipedia.org/wiki/Kernel_\(computers\)](http://www.wikipedia.org/wiki/Kernel_(computers))

2 Konfigurasi dan kompilasi kernel

2.1 Persiapan

Karena kompilasi kernel merupakan proses baku yang mungkin akan sering anda lakukan, sebelum melakukan instalasi kernel ada baiknya anda membackup sistem anda, jika nanti pada suatu waktu sistem anda mengalami *crash* atau hal-hal yang lain, tapi ini hanya anjuran bukan ketetapan.

Langkah yang paling awal adalah menginventarisasi hardware, peripheral, dan aplikasi yang berjalan di sistem anda sekarang misalnya, dalam hal ini yang perlu diperhatikan adalah jenis prosesor, PCI Card (ethernet/soundcard), media penyimpanan (IDE/EIDE/SCSI), USB, PCMCIA Card, *file-system* yang digunakan dan hal-hal lain yang dirasa perlu untuk diperhatikan.

Untuk mengetahui jenis prosesor anda (tentunya selain dengan membuka *casing* atau dari nota kwitansi dari vendor anda) bisa dilakukan dengan melihat file `/proc/cpuinfo`, misalnya disini dicontohkan adalah Prosesor AMD Duron 1 GHz :

```
[root@hebron root]# cat /proc/cpuinfo
processor      : 0
vendor_id     : AuthenticAMD
cpu family    : 6
model         : 7
model name    : AMD Duron(tm) processor
stepping      : 1
cpu MHz       : 1101.922
cache size    : 64 KB
fdiv_bug     : no
hlt_bug      : no
f00f_bug     : no
coma_bug     : no
fpu          : yes
fpu_exception : yes
cpuid level   : 1
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 mmx fxsr sse
bogomips     : 2182.57
```

Untuk melihat jenis ethernet card anda, bisa dilakukan dari perintah `dmesg`, misalnya dalam hal ini saya menggunakan ethernet card RTL8139 :

```
[root@hebron root]# dmesg | grep eth
eth0: RealTek RTL8139 Fast Ethernet at 0xd00d1000, 00:e0:4c:7c:2c:19, IRQ 11
eth0: Identified 8139 chip type 'RTL-8139B'
eth0: Setting 100mbps full-duplex based on auto-negotiated partner ability 41e1.
```

Untuk melihat tipe *file-system* yang digunakan bisa dengan menggunakan perintah `mount` misalnya dalam hal ini saya menggunakan tipe *file-system* journaling EXT3 pada semua partisi (/ , /home , /usr dan /var):

```
[root@hebron root]# mount
/dev/hda1 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda6 on /home type ext3 (rw)
none on /dev/shm type tmpfs (rw)
/dev/hda2 on /usr type ext3 (rw)
/dev/hda3 on /var type ext3 (rw)
```

Misalnya anda menggunakan harddisk SCSI, untuk melihat module yang digunakan oleh SCSI card anda adalah dengan perintah *dmesg*, sebagai contoh pada salah satu server saya (*nablus*), SCSI Card yang digunakan adalah Symbios 53C8XX dengan dua buah disk SCSI IBM :

```
[root@nablus root]# dmesg | grep scsi
scsi0 : sym53c8xx-1.7.1-20000726
scsi : 1 host.
Detected scsi disk sda at scsi0, channel 0, id 6, lun 0
Detected scsi disk sdb at scsi0, channel 0, id 10, lun 0
scsi : detected 2 SCSI disks total.
```

Setelah anda selesai anda selesai menginventarisir semua peripheral anda, langkah selanjutnya adalah download source kernel. Mendapatkan source kernel terbaru tidaklah sulit, silakan periksa <http://www.kernel.org> untuk versi kernel yang paling baru dan tentunya sesuai dengan kebutuhan anda, ketika tulisan ini dibuat kernel (stable) yang paling baru adalah versi 2.4.20. Jika anda terhubung ke internet, anda bisa menggunakan perintah *wget* untuk mendownload source kernel yang paling baru, namun sebelumnya jika anda belum berada pada direktori **/usr/src** silakan anda *cd* (*change directory*) terlebih dahulu ke direktori tersebut dengan menggunakan account **root** :

```
[root@hebron root]# cd /usr/src
[root@hebron src]# wget http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.20.tar.bz2
```

Jika sudah selesai, ekstrak source kernel tersebut (akan menghasilkan direktori **linux-2.4.20**) :

```
[root@hebron src]# tar -jxvf linux-2.4.20.tar.bz2
```

Jika sebelumnya anda sudah pernah melakukan kompilasi kernel, atau pada direktori **/usr/src** terdapat direktori *linux*, maka anda pindahkan direktori tersebut ke nama yang lain, atau jika *linux* tersebut berupa file *symlink* (simbolik link) maka anda bisa menghapusnya :

```
[root@hebron src]# rm -rf linux
```

Buat *symlink* direktori **linux-2.4.20** ke **linux** :

```
[root@hebron src]# ln -s linux-2.4.20 linux
```

Pindah ke direktori **linux** :

```
[root@hebron src]# cd linux
```

Langkah selanjutnya adalah mengkonfigurasi dan melakukan kompilasi

2.2 Konfigurasi dan kompilasi

Setelah selesai melakukan peng-ekstrak-an source kernel linux, sekarang saatnya anda mengkonfigurasinya. Setelah anda berada pada direktori `/usr/src/linux` maka anda lakukan perintah `make menuconfig` (jika anda penggemar tampilan ala program di DOS dengan warna dominan biru dari software `ncurses`) atau `make xconfig` (jika anda penggemar tampilan GUI) atau `make config` (*the hard way*, tidak disarankan :)), saya pilih `make menuconfig` :

```
[root@hebron src]# make menuconfig
```

Maka akan muncul tampilan dari menu-menu yang bisa dipilih untuk digunakan atau dihapus dari suatu konfigurasi kernel, saya tidak akan bahas semuanya karena memang saya belum pernah menggunakan semuanya :) . Untuk menambahkan suatu item untuk sistem kernel yang **modular** pada menu ini anda bisa menekan tombol **M** di keyboard anda, untuk sistem kernel yang **builtin** tekan tombol **Y** (atau dalam beberapa hal anda bisa juga menggunakan **space bar**), untuk membatalkan pilihan anda terhadap suatu item tekan **N**. Simbol `—>` merupakan pilihan yang mempunyai beberapa subpilihan, simbol `[*]` menjelaskan bahwa item tersebut merupakan *builtin*, simbol `[M]` merupakan *modular*, simbol `[]` merupakan item yang tidak dipilih atau modul dengan tipe builtin, dan simbol `< >` merupakan simbol item yang bisa dan dapat digunakan sebagai modul maupun builtin. Untuk memilih dari item pilih `<Select>` atau tekan **Enter**, untuk kembali ke menu awal tekan `<Exit>` atau **Esc**

Pada pilihan pertama untuk digunakan oleh kernel untuk melakukan proses *debugging* pada module yang masih tahap development, supaya aman masukkan kedalam kernel :

```
Code maturity level options  --->
[*] Prompt for development and/or incomplete code/drivers
```

Pilihan selanjutnya jika anda berencana menggunakan sistem kernel builtin maka hilangkan dari konfigurasi dengan menekan tombol **N**, namun jika anda menggunakan sistem modular anda bisa melaluinya karena secara default sudah dimasukkan kedalam konfigurasi :

```
Loadable module support  --->
[ ] Enable loadable module support
```

Pilihan selanjutnya adalah tipe prosesor yang anda gunakan, pilih yang sesuai dengan prosesor anda, karena jika salah, kernel yang anda hasilkan kemungkinan besar tidak akan bisa diboot :

```
Processor type and features  --->
```

Pilih pada pilihan paling atas untuk menyesuaikan, biasanya secara default adalah :

```
(Pentium-III/Celeron(Coppermine)) Processor family
```

Sebagai contoh sesuai dengan prosesor yang saya pakai adalah AMD Duron 1 Ghz, pilih menjadi :

```
(Athlon/Duron/K7) Processor family
```

Pada dua pilihan selanjutnya anda bisa mengabaikannya, kecuali jika menggunakan PCMCIA, CardBus, Compatible Bridge, APM Bios dan Embedded Devices, pilihan tersebut adalah :

```
General setup --->
Memory Technology Devices (MTD) --->
```

Jika anda tidak menggunakan parallel port yang biasanya digunakan untuk koneksi *PLIP* (*Parallel Line Internet Protocol Link*) maka pilihan selanjutnya bisa anda abaikan yaitu :

```
Parallel port support --->
```

Pilihan selanjutnya bisa anda abaikan karena secara default sudah dimasukkan kedalam konfigurasi kernel :

```
Plug and Play configuration --->
```

Jika anda sering menggunakan sistem anda untuk mengeksplorasi suatu file data misalnya file ISO, maka pada pilihan berikut masukkan *Loopback device support* kedalam konfigurasi anda, sehingga nantinya anda bisa mempunyai *device /dev/loop0, /dev/loop1* dst, jika anda tidak menggunakannya, abaikan :

```
Block devices --->
< > Loopback device support
```

Jika anda tidak menggunakan sistem *RAID* maupun *LVM* (*Logical Volume Management*), maka pilihan selanjutnya bisa diabaikan :

```
Multi-device support (RAID and LVM) --->
```

Pada pilihan selanjutnya adalah tentang *networking* termasuk didalamnya fungsi firewall, konfigurasi IP secara otomatis pada level kernel, QoS (*Quality of Service*) dan lain lain, anda bisa membuat item-item didalam pilihan tersebut menjadi modul, tetapi dalam beberapa hal akan lebih baik jika anda builtin semua didalam kernel :

```
Networking options --->
[*] Network packet filtering (replaces ipchains)
[*] Network packet filtering debugging (NEW)
[*] Socket Filtering utk router (berserta dibawahnya)
[*] IP: advanced route
[*] IP: policy routing (NEW)
[*] IP: use netfilter MARK value as routing key (NEW)
[*] IP: fast network address translation (NEW)
[*] IP: equal cost multipath (NEW)
[*] IP: use TOS value as routing key (NEW)
[*] IP: verbose route monitoring (NEW)
[*] IP: large routing tables (NEW) untuk support (DHCP/BOOTP/RARP)
```

Jika anda menggunakan DHCP, BOOTP dan RARP maka anda harus memilih item dibawah ini :

```
[*] IP: kernel level autoconfigurationr
[*]   IP: DHCP support (NEW)
[*]   IP: BOOTP support (NEW)
[*]   IP: RARP support (NEW)
```

TCP syscookie digunakan untuk mencegah serangan *SYN Flooding* yang termasuk dalam serangan *DoS* (*Denial of Service*), untuk mengaktifkannya anda bisa menambahkan "echo 1 >/proc/sys/net/ipv4/tcp_syncookies" pada script start-up sistem anda :

```
[*] IP: TCP syncookie support (disabled per default)
```

Untuk mengaktifkan fungsi firewall, NAT yang termasuk dalam *iptables* (*netfilter*), masukkan semua pilihan pada item :

```
IP: Netfilter Configuration --->
```

Jika anda berencana akan membatasi trafik yang keluar maupun masuk ke sistem anda (dengan menggunakan CBQ/HTB/SFQ/Diffserv dll), silakan masuk pada pilihan berikut ini, jika tidak menggunakannya anda bisa mengabaikan :

```
QoS and/or fair queueing --->
```

Dua pilihan pada menu utama selanjutnya bisa anda abaikan jika memang anda tidak menggunakannya :

```
Telephony Support --->
ATA/IDE/MFM/RLL support --->
```

Jika anda tidak menggunakan SCSI Card, pilihan selanjutnya bisa diabaikan, namun jika anda menggunakan SCSI Card sesuaikan dengan jenis yang anda miliki, masukkan secara builtin di konfigurasi kernel :

```
SCSI support --->
```

Tiga buah pilihan selanjutnya bisa anda abaikan jika anda tidak menggunakannya :

```
Fusion MPT device support --->
IEEE 1394 (FireWire) support (EXPERIMENTAL) --->
I2O device support --->
```

Pada pilihan selanjutnya dari menu utama adalah pemilihan jenis ethernet card yang dipakai, sesuaikan dengan milik anda. Disini anda bisa memasukkannya sebagai module dengan menekan **M** (jika anda sering gonta-ganti ethernet) ataupun builtin di kernel. Anda bisa menambahkan modul ethernet sebanyak jenis ethernet yang anda miliki jika sisten kernel anda modular, namun jika builtin saya rasa itu bukan ide yang bagus. Sebagai contoh untuk ethernet card dengan chipset RTL8139 kita masukkan sebagai berikut :

```
Network device support --->
Ethernet (10 or 100Mbit) ---->
<*>   RealTek RTL-8139 C+ PCI Fast Ethernet Adapter support (EXPERIMENTAL)
<*>   RealTek RTL-8139 PCI Fast Ethernet Adapter support
[*]   Use PIO instead of MMIO
[*]   Support for uncommon RTL-8139 rev. K (automatic channel equalization)
[*]   Support for older RTL-8129/8130 boards
```

Tujuh buah pilihan dari menu utama selanjutnya bisa anda abaikan (lagi, jika anda tidak menggunakannya) :

```
Skip Amateur Radio support ---->
IrDA (infrared) support ---->
ISDN subsystem ---->
Old CD-ROM drivers (not SCSI, not IDE) ---->
Input core support ---->
Character devices ---->
Multimedia devices ---->
```

Pilihan selanjutnya ada tentang *file-system*, karena tipe file sistem yang saya gunakan dalam contoh ini adalah EXT3 maka saya pilih berikut ini, sesuaikan dengan kebutuhan dan konfigurasi anda :

```
File systems ---->
<*> Ext3 journalling file system support
[*] JBD (ext3) debugging support
```

Jika pada sistem anda terdapat partisi DOS, VFAT (Win95/98) dan NTFS dan anda ingin menggunakannya maka pilih item berikut ini :

```
<*> DOS FAT fs support
<*> MSDOS fs support
<*> UMSDOS: Unix-like file system on top of standard MSDOS fs
<*> VFAT (Windows-95) fs support
<*> NTFS file system support (read only)
```

Jika anda berencana menggunakan Network File Sistem (NFS) jangan lupa memasukkan pilihan berikut ini :

```
Network File Systems ---->
<*> NFS file system support
<*> NFS server support
```

Jika sistem anda sering menggunakan resource dari sistem windows dalam hal file sharing, maka pilihan berikut jangan dilupakan :

```
<*> SMB file system support (to mount Windows shares etc.)
```


Pilihan selanjutnya adalah sound card, sesuaikan dengan konfigurasi anda, jika komputer anda hanya bisa mengeluarkan 'beep' :) maka abaikan pilihan ini :

```
Sound --->
```

Pilihan selanjutnya adalah dukungan terhadap USB, jika anda mempunyai peripheral yang terhubung ke USB, maka silakan *explore* pilihan ini :

```
USB support --->
```

Pilihan selanjutnya tentang Bluetooth bisa diabaikan jika anda tidak menggunakannya :

```
Bluetooth support --->
```

Jika anda sedang membuat suatu driver dan ingin menghasilkan *debugging* yang lebih baik, maka pilihan selanjutnya bisa anda pilih yaitu :

```
Kernel hacking --->
```

Jika anda tidak menggunakan kompresi maupun dekompresi *zlib* pada kernel, maka pilihan berikut bisa diabaikan :

```
Library routines --->
```

Dua pilihan yang paling terakhir dibawah tanda minus tiga kali (—), merupakan kontrol terhadap file konfigurasi kernel, misalnya anda ingin menyimpan dengan nama yang lain (defaultnya adalah file **.config** pada direktori source kernel), atau anda ingin menggunakan file konfigurasi kernel yang lain, pilihan tersebut adalah :

```
Load an Alternate Configuration File
Save Configuration to an Alternate File
```

Jika tidak anda gunakan pilihan tersebut diatas, maka pilih **Exit** kemudian pilih **Yes**

Langkah selanjutnya adalah melakukan kompilasi kernel yang sudah kita konfigurasi seperti disebutkan diatas, sebelum proses kompilasi set semua *dependencies* dari modul-modul tersebut :

```
[root@hebron linux]# make dep
[root@hebron linux]# make clean
```

Jika anda menggunakan sistem kernel yang modular, maka lakukan juga :

```
[root@hebron linux]# make modules
[root@hebron linux]# make modules_install
```

Kemudian untuk kompilasi kernel lakukan perintah berikut ini, pada langkah ini proses bisa memerlukan waktu yang lamanya tergantung dari jenis prosesor dan memori yang anda miliki, pada prosesor AMD Duron 1 Ghz Memori 256 MB diperlukan waktu sekitar 8 menit 23 detik untuk menghasilkan sebuah kernel baru :

```
[root@hebron linux]# make bzImage
```

Jika anda ingin menghitung waktu yang diperlukan, anda bisa lakukan dengan perintah sederhana berikut :

```
[root@hebron linux]# echo "Mulai: $(date)" > waktu; make bzImage; echo "Selesai: $(date)" >> waktu
```

Jika semuanya sukses dan tidak ada error, lihat file *waktu* pada direktori `/usr/src/linux` dengan perintah *cat*, misalnya :

```
[root@hebron linux]# cat waktu
Mulai: Sun Jun  8 11:41:07 WIT 2003
Selesai: Sun Jun  8 11:49:30 WIT 2003
```

Setelah selesai kompilasi kernel, langkah selanjutnya adalah mengkonfigurasi boot manager anda

2.3 Mengkonfigurasi boot loader

Sebelum kita mengkonfigurasi boot loader, salin file kernel yang baru dikompilasi dan file *System.map* ke direktori `/boot` :

```
[root@hebron linux]# cp /usr/src/linux-2.4.20/arch/i386/boot/bzImage /boot/vmlinuz-2.4.20
[root@hebron linux]# cp /usr/src/linux-2.4.20/System.map /boot/System.map-2.4.20
```

Kemudian tentukan dimana letak partisi **root** (/) dari sistem anda, anda bisa menggunakan perintah *mount*, misalnya :

```
[root@hebron linux]# mount | grep -w /
/dev/hda1 on / type ext3 (rw)
```

Atau dengan perintah *df* :

```
[root@hebron linux]# df /
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/hda1            1004024        94660    858360  10% /
```

Dari perintah diatas maka didapatkan informasi partisi **root** berada pada device `/dev/hda1`

2.3.1 Jika anda menggunakan *LILO*

Langkah selanjutnya adalah mengkonfigurasi file `/etc/lilo.conf` tambahkan baris berikut ini (sesuaikan pada **root=** dengan letak partisi **root** sistem anda) :

```
image=/boot/vmlinuz-2.4.20
label=linux-2.4.20
read-only
root=/dev/hda1
```

Untuk mengeset linux supaya setiap boot akan langsung menggunakan kernel yang baru, ubah pada file `/etc/lilo.conf` dari (namun jika anda masih *prefer* menggunakan kernel lama, silakan abaikan langkah berikut ini) :

```
default=linux
```

menjadi :

```
default=linux-2.4.20
```

Kemudian jalankan perintah `lilo`, sehingga konfigurasi yang baru akan dibaca (tanda `*` merupakan kernel default yang akan digunakan setelah sistem *direboot*) :

```
[root@hebron linux]# lilo
Added linux
Added linux-2.4.20 *
```

Untuk menambah *verbosity* (tampilan yang dihasilkan) anda bisa menggunakan argumen `-v` ketika menjalankan perintah `lilo` :

```
[root@hebron linux]# lilo -v
```

Jika tidak ada masalah yang berarti dalam pengupdatean lilo, maka anda bisa me-*reboot* sistem anda :

```
[root@hebron linux]# reboot
```

2.3.2 Jika anda menggunakan GRUB

Jika anda menggunakan grub sebagai boot loader, edit file `/etc/grub.conf`, tambahkan baris berikut ini (sesuaikan pada entry `root=` dengan letak partisi `root (/)` anda :

```
title Red Hat Linux (2.4.20)
root (hd0,0)
kernel /boot/vmlinuz-2.4.20 ro root=/dev/hda1
```

Anda bisa me-*reboot* sistem anda :

```
[root@hebron linux]# reboot
```

2.3.3 Setelah *reboot*

Untuk memeriksa apakah sistem anda sudah menggunakan kernel yang baru, gunakan perintah `uname`, misalnya :

```
[root@hebron root]# uname -a
Linux hebron.cs.perbanas.edu 2.4.20 #1 SMP Sun Jun 8 11:41:09 WIT 2003 i686 athlon i386 GNU/Linux
```

Anda bisa memeriksa apakah kernel anda sudah sesuai dengan yang anda kehendaki, gunakan perintah *dmesg*, misalnya :

```
[root@hebron root]# dmesg
Linux version 2.4.20 (root@hebron.cs.perbanas.edu) (gcc version 3.2 20020903 (Red Hat Linux 8.0 3.2-7))
BIOS-provided physical RAM map:
BIOS-e820: 0000000000000000 - 00000000000a0000 (usable)
BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
BIOS-e820: 0000000000100000 - 0000000000f7f000 (usable)
BIOS-e820: 0000000000f7f000 - 0000000000f7f3000 (ACPI NVS)
BIOS-e820: 0000000000f7f3000 - 0000000000f800000 (ACPI data)
BIOS-e820: 00000000ffff0000 - 0000000100000000 (reserved)
247MB LOWMEM available.
On node 0 totalpages: 63472
zone(0): 4096 pages.
zone(1): 59376 pages.
zone(2): 0 pages.
Kernel command line: auto BOOT_IMAGE=linux-2.4.20 ro root=301 BOOT_FILE=/boot/vmlinuz-2.4.20
Found and enabled local APIC!
Initializing CPU#0
Detected 1101.968 MHz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 2195.45 BogoMIPS
[....]
```

Pastikan juga peripheral anda bekerja (PCI Card, PCMCIA, USB dan lain-lain). Jika sitem anda menggunakan skema kernel yang *builtin (monolitik)* semuanya, anda bisa mencoba dengan perintah *lsmod*, misalnya :

```
[root@hebron root]# lsmod
Module                Size Used by    Not tainted
lsmod: QM_MODULES: Function not implemented
```

Jika semuanya OK dan tidak ada masalah, selamat anda sudah sukses mengupdate/upgrade kernel anda.

3 Lain-Lain

3.1 (sedikit) Tips dan trik

3.1.1 Menggunakan konfigurasi kernel yang lama

Jika sebelumnya anda pernah mengupgrade kernel dan ingin menggunakan file konfigurasi pada kernel yang lama, sebelum *make manufconfig* anda cukup menyalin file *.config* dari direktori source kernel yang lama, misalnya upgrade dari kernel 2.4.19 ke 2.4.20 :

```
[root@hebron root]# cd /usr/src
[root@hebron src]# cp linux-2.4.19/.config linux
```

```
[root@hebron src]# cd linux
[root@hebron linux]# make menuconfig
```

Atau menggunakan argumen *oldconfig* :

```
[root@hebron linux]# make oldconfig
```

3.1.2 Mencatat hasil perintah *make*

Misalkan anda ingin mengarsip hasil dari perintah *make* (misalnya *make bzImage*) , jika anda menggunakan *bash* maka :

```
[root@hebron linux]# make bzImage 2>&1 | tee hasil-make-bzImage
```

Maka perintah tersebut akan menghasilkan file *hasil-make-bzImage* yang merupakan output perintah *make bzImage*

3.1.3 Penamaan kernel yang unik (**EXTRAVERSION**)

Untuk menambahkan penamaan yang unik pada kernel, anda bisa mengubah *EXTRAVERSION* pada file **Makefile** pada direktori source kernel, misalnya dari :

```
EXTRAVERSION =
```

ubah menjadi :

```
EXTRAVERSION = -intifada
```

3.1.4 Ekstra security

Jika anda menginginkan penambahan tingkat security pada kernel anda, sudah saatnya anda menggunakan *grsecurity patch* (<http://www.grsecurity.net>) untuk kernel seri 2.4.x atau menggunakan *openwall patch* (<http://www.openwall.com/linux>) untuk kernel seri 2.2.x dan 2.0.x. Patch tersebut akan mempersolid sistem anda dan mempersulit seorang *kiddies* untuk merusak sistem anda (*at least to reduce* :))

3.2 Format lain dari dokumen ini

- PDF (<http://people.cakraweb.com/~asfik/writings/kompilasi-kernel.pdf>)
- LaTeX (<http://people.cakraweb.com/~asfik/writings/kompilasi-kernel.tex>)
- ASCII (<http://people.cakraweb.com/~asfik/writings/kompilasi-kernel.txt>)

3.3 Referensi

- The Linux Kernel HOWTO (<http://www.tldp.org/HOWTO/Kernel-HOWTO>)
- File `/usr/src/linux/README`
- Direktori `/usr/src/linux/Documentation`