Memahami GRUB Lebih Lanjut

GRUB saat ini digunakan sebagai *boot loader default* pada berbagai distribusi Linux populer. Selain berfungsi sebagai bootloader, GRUB memiliki beberapa pengaturan lain yang bermanfaat untuk menjadikan proses boot kita lebih menarik, aman, dan fleksibel. Kita akan membahas beberapa di antaranya.

Pertama-tama, kita akan membahas file konfigurasi GRUB secara sekilas, untuk melihat opsi penting yang mungkin berguna. Setelah itu, kita akan melihat apa yang harus kita lakukan apabila file konfigurasi GRUB hilang ataupun mengalami masalah. Berikutnya, kita akan melihat bagaimana mengedit *command line* kernel pada saat menu GRUB ditampilkan. Dan, yang terakhir, kita akan melihat *tool grubonce*, yang sangat berguna.

Tulisan ini dibangun di atas sistem openSUSE 10.2, namun seharusnya dapat diterapkan tanpa masalah pada sistem lainnya.

Menu.lst

Secara umum, file konfigurasi menu GRUB—secara default tersimpan di /boot/grub/menu.lst—relatif sederhana dan dapat dimengerti dengan mudah. File konfigurasi itu bisa dibagi dalam dua seksi (tidak kaku, diberikan hanya untuk memudahkan pembahasan):

- Umum.
- Per menu entry.

Berikut ini contoh konfigurasi di komputer penulis:

```
default 0
timeout 3
color light-blue/black blue/light-gray

title   openSUSE 10.2 (hda1)
        root    (hd0,0)
        kernel  /boot/vmlinuz-2.6.18.2-34-default root=/dev/hda1 vga=0x317    maxcpus=0 resume=/dev/hda5 splash=verbose showopts
        initrd  /boot/initrd-2.6.18.2-34-default

title   openSUSE 10.2 (hda7)
        kernel  (hd0,6)/boot/vmlinuz-2.6.18.2-34-default root=/dev/hda7 vga=0x317 maxcpus=0 resume=/dev/hda5 splash=silent showopts
        initrd  (hd0,6)/boot/initrd-2.6.18.2-34-default

title   Debian GNU/Linux 3.1 (hda6)
        root    (hd0,5)
        kernel  /boot/vmlinuz-2.6.17.9-nop-20060825-1 root=/dev/hda6 ro vga=791 suspend2=/dev/hda5
        initrd  /boot/initrd.img-2.6.17.9-nop-20060825-1

title   Windows
        rootnoverify (hd0,0)
        chainloader (hd0,1)+1
```

Bisa kita lihat, tiga baris pertama merupakan konfigurasi seksi umum, yang akan mempengaruhi menu GRUB secara umum. Berikut ini adalah penjelasannya:

- Default menyatakan entri default yang akan di boot. Entri dimulai dari 0.
- *Timeout* menyatakan waktu tunggu sebelum melakukan booting ke nilai default, kecuali diinterupsi oleh pengguna.
- Baris ketiga menyatakan warna yang akan digunakan. Berikut ini *prototype* perintah:

```
color normal [highlight]
```

Untuk normal dan [highlight] sendiri, warna diberikan dalam pasangan FOREGROUND/BACKGROUND.

Warna yang tersedia untuk BACKGROUND adalah *black, blue, green, cyan, red, magenta , brown, light-gray*.

Semantara, untuk warna FOREGROUND, warna BACKGROUND bisa dipergunakan, ditambah dengan *dark-gray, light-blue, light-green, light-cyan, light-red, light-magenta, yellow, white*.

Sebagai catatan, *color* sendiri merupakan perintah general, yang mana artinya bisa diberikan pada seksi umum ataupun per menu entry. Sekali lagi, pengelompokan yang dilakukan ditulisan ini semata untuk memudahkan pembahasan.

password

Selain tiga perintah tersebut, ada baiknya kita menambahkan perintah *password*, yang akan mencegah pengguna untuk mengedit entry menu apabila password yang dimasukkan salah. Password bisa diberikan dalam format plain ataupun md5.

Untuk memberikan password dalam format plain, berikanlah perintah password diikuti oleh password yang diinginkan. Contoh:

```
password rahasia
```

Untuk memberikan password dalam format md5, berikanlah perintah password, diikuti oleh opsi –md5 dan password dalam format md5, yang bisa dihasilkan program grub-md5-crypt.

Contoh:

```
# grub-md5-crypt
Password:
Retype password:
$1$/MvDo1$ZO4E.crkqNAEeHntiQIHJ.
```

Contoh entry grub:

```
password --md5 $1$/MvDo1$ZO4E.
crkqNAEeHntiQIHJ.
```

Sebagai catatan, agar lebih aman, berikanlah hak baca tulis terhadap file menu.lst hanya untuk root, dengan perintah:

```
# chown root:root /boot/grub/menu.
lst
# chmod 600 /boot/grub/menu.lst
```

Sebagai catatan, password juga bisa diberikan pada seksi per menu entry, yang mengakibatkan user harus memasukkan password agar dapat melakukan *booting* menu *entry* yang diinginkan.

Menu.lst hilang!

Ampun! Kenapa saya menghapus menu.lst yang begitu penting itu? Jadinya, setiap kali melakukan booting, yang dijumpai bukan lagi menu grub yang indah, namun prompt grub.

Penulis beberapa kali menjumpai kasus tersebut. Apabila kita bisa yakin bahwa permasalahan terletak pada hilangnya file menu.lst (bukan yang lebih parah, seperti permasalahan pada boot sector, internal grub atau lainnya), kita masih bisa setidaknya melakukan tiga hal berikut:

- Boot dengan CD rescue dan meng-*copy*-kan kembali menu.lst dari backup. Tidak punya *back-up*? Ambil dari *template*. Sistem Anda tidak menyediakan template? Copy dari teman. Tidak punya? Ketik lagi sendiri.
- Langsung memberikan perintah pada prompt, seperti yang ditemukan pada menu entry.
- Load file konfigurasi back-up langsung dari prompt grub, tentunya, selama Anda memiliki back-up file tersebut.

Melihat kemungkinan hilangnya file menu.lst tersebut, tentunya kita perlu menyiapkan back-up.

Untuk langsung memberikan perintah pada prompt, kita harus menghafal perintah-perintah pada menu entry yang ingin di boot. Setelah itu, baris demi baris perintah tersebut harus diberikan pada prompt grub. Terakhir, setelah semua baris perintah diketikkan, berikanlah perintah 'boot'.

Namun, menghafal command sepanjang itu tentunya bukanlah pekerjaan kita yang paling penting. Alangkah baiknya kalau kita bisa melakukan load file konfigurasi backup dengan memberikan perintah berikut pada prompt grub:

```
configfile <FILE>
```

Contoh:

```
configfile (hd0,0)/root/BACKUP/
SYSTEM/GRUB/menu.lst
```

Device yang dipergunakan, seperti (hd0,0) tentunya harus merujuk kepada aturan GRUB. Singkatnya, umumnya, hda (atau, kalau SCSI: sda) akan dipetakan sebagai hd0 dan partisi akan dimulai dari 0. Bacalah dokumentasi grub untuk informasi selengkapnya.

Mengedit langsung menu entry

Kita bisa langsung mengedit menu entry di grub (tentunya, setelah mengisikan password yang benar, apabila password digunakan). Hal ini akan sangat berguna terutama ketika Anda ingin memberikan parameter-parameter spesial dan atau temporer ke kernel atau sistem yang Anda gunakan.

Untuk mengedit perintah menu entry, lakukanlah langkah-langkah berikut ini:

- Pilihlah entry yang ingin diedit.
- Tekan tombol e untuk mengedit.
- Baris-baris perintah untuk menu entry terpilih akan ditampilkan. gunakan tombol panah untuk navigasi.
  - Tekan tombol o untuk menambah baris baru setelah baris aktif.
  - Tekan tombol O untuk menambah baris baru sebelum baris aktif.
  - Tekan tombol d untuk menghapus baris aktif.
  - Tekan tombol b untuk boot.
  - Tekan tombol ESC untuk membatalkan perubahan dan kembali ke dialog sebelumnya.

Di samping nilai gunanya, kemampuan untuk mengedit menu GRUB, terutama pada command line kernel, bisa menjadi sangat berbahaya. Beberapa sistem menjadi sangat mudah bobol hanya dengan beberapa perintah yang bertujuan untuk mengalihkan init ataupun pengaturan runlevel tertentu. Beberapa sistem telah memiliki proteksi tertentu. Namun, secara umum tetap berbahaya.

Oleh karena itu, untuk alasan keamanan, selalu gunakan password untuk boot loader (lebih jauh: dan bios untuk password dan boot device, dan casing komputer, dan ruang komputer dan lokasi penyimpanan komputer).

Grubonce

Beberapa desktop populer menawarkan boot ke sistem operasi lain, yang terdaftar pada menu.lst, ketika perintah logout/reboot diberikan. Ketika kita memilih entry lain misalnya, komputer akan di-*reboot* dan lalu, entry yang kita pilih sebelumnya akan diboot. Menarik sekali, karena kita tidak perlu secara manual memilih pada saat boot. Terutama, kalau entry yang dipilih bukanlah entry default.

Sebenarnya, untuk kebutuhan tersebut, kita bisa mempergunakan tool yang telah disediakan, yaitu grubonce. Sesuai namanya, perintah ini akan memungkinkan kita untuk boot pada menu entry yang kita tentukan pada boot selanjutnya. Setelah boot selanjutnya dilakukan, pengaturan kembali ke normal.

Apabila dijalankan (sebagai root) tanpa parameter apapun , program grubonce akan menampilkan entry yang tersedia. Contoh:

```
# grubonce
0: openSUSE 10.2 (hda1)
1: openSUSE 10.2 (hda7)
2: Debian GNU/Linux 3.1 (hda6)
3: Windows
```

Selanjutnya, untuk boot ke entry tertentu, berikanlah parameter (berupa nomor urut) ketika menjalankan grubonce. Contoh:

Contoh:

```
# grubonce 3
Using entry #3: Windows
```

Sebagai catatan: kita bisa tetap bekerja. Komputer tidak akan di-reboot ataupun d-*shutdown* sampai kita melakukannya sendiri. Boot berikutnya, menu entry yang kita pilih akan di-boot.

Sampai di sini dulu pembahasan kita. GRUB sendiri adalah sistem yang kompleks. Rujuklah kepada dokumentasi GRUB untuk informasi selengkapnya. Selamat mencoba!

Noprianto [noprianto@infolinux.co.id

Manajemen Kapasitas Harddisk Menggunakan LVM

Bagian 1 dari 2 artikel

Logical Volume Manager atau yang biasa disingkat dengan LVM merupakan metode alokasi kapasitas di suatu media penyimpanan, yang lebih fleksibel daripada skema partisi konvensional. Dengan menggunakan LVM, kita dapat dengan mudah mengubah kapasitas harddisk tanpa perlu khawatir kehilangan data.

Kesulitan utama pada saat instalasi Linux yang dirasakan oleh kebanyakan pengguna Linux adalah pada saat melakukan partisi harddisk. Diperlukan estimasi yang tepat dalam menentukan berapa banyak *space* yang dibutuhkan untuk file system, dan user file yang dapat membuat proses instalasi menjadi lebih kompleks.

Setelah user menentukan berapa banyak kapasitas yang dibutuhkan untuk partisi /home, /usr, dan /, pada suatu saat salah satu partisi di harddisk tersebut akan penuh dan beberapa partisi masih memiliki kapasitas yang cukup.

Salah satu cara agar suatu partisi yang penuh dapat menggunakan kapasitas yang masih kosong di partisi lainnya adalah dengan menggunakan LVM. LVM adalah metode alokasi kapasitas media penyimpanan, yang dapat kita gunakan untuk menambah, menghapus, dan mengubah kapasitas partisi pada saat kita butuhkan. Dengan menggunakan LVM, maka kapasitas harddisk akan di alokasikan ke *single volume group* dan *logical volume* yang dibuat untuk menyatukan partisi /home, /usr, dan /, dalam contoh kasus di atas.

Sebagai contoh, jika partisi /home telah penuh, maka dimungkinkan untuk menggunakan kapasitas di partisi /usr yang masih tersedia, dengan jalan mengurangi partisi /usr sesuai kapasitas yang dibutuhkan, dan dialokasikan kapasitas tersebut ke partisi /home. Cara lainnya adalah dengan mengalokasikan kapasitas minimal untuk tiap logical volume, dan membiarkan beberapa kapasitas disk *unallocated*. Dan disaat partisi tersebut penuh, maka partisi tersebut akan dapat bertambah pada saat dibutuhkan.

Untuk lebih memahami cara kerja LVM, pada tutorial kali ini *InfoLINUX* akan membahas cara kerja LVM berikut dengan cara penggunaannya.

LVM layout

Terdapat beberapa istilah yang perlu Anda ketahui dalam penggunaan LVM. Beberapa terminologi yang perlu diketahui di antaranya:

- Physical volumes
  Istilah ini mengacu pada physical disks, atau disk partitions, seperti /dev/hda, /dev/hdb1, dan sebagainya. Dengan LVM, kita dapat mengombinasikan beberapa *physical volume* ke dalam *volume groups*.
- Volume groups
  Istilah ini mengacu ke *real physical volumes* dan media penyimpanan yang digunakan untuk membuat logical volumes. Kita dapat mengasumsikan volume groups sebagai sebuah partisi virtual yang mengacu ke sebuah physical volume.
- Logical volumes
  Istilah ini mengacu pada volume yang dapat Anda akhiri kapan saja pada saat proses *mount* yang sedang terjadi pada sistem Anda. Logical volume ini dapat ditambah, dihapus, dan diubah ukurannya pada saat berjalan.

Skema LVM yang akan dibuat.

Untuk lebih memahami istilah di atas, perhatikan Gambar-1. Pada Gambar-1 dimisalkan Anda memiliki beberapa physical volume (sebagai contoh /dev/sdb1-/dev/sde1), dan dalam physical volume ini, kita dapat menciptakan satu atau lebih volume groups (sebagai contoh fileserver), dan dalam setiap volume group Anda dapat menciptakan satu atau lebih logical volumes.

Jika kita menggunakan beberapa physical volume, setiap logical volume dapat lebih besar ukurannya dibandingkan salah satu partisi yang terdapat pada physical volume (tentu saja jumlah kapasitas total dari logical volumes tidak boleh melebihi dari jumlah kapasitas total dari physical volumes). Sebagai langkah yang baik, jangan alokasikan semua kapasitas harddisk ke logical volumes, tetapi sisakan beberapa space yang tidak digunakan. Dengan cara ini, kita dapat menambah kapasitas ke satu atau beberapa logical volumes di kemudian hari.

Sebagai contoh di sini, kita akan membuat sebuah volume group yang bernama fileserver. Di volume group ini, kita akan membuat logical volumes /dev/fileserver/share, /dev/fileserver/backup, /dev/fileserver/backup, dan /dev/fileserver/media (yang mana hanya akan kita gunakan setengah dari kapasitas yang ditawarkan oleh physical volume kita saat ini. Dengan cara ini, nantinya kita dapat berpindah ke RAID1).

Persiapan

Dalam DVD *InfoLINUX* edisi ini, kami telah menyertakan file Debian_Etch_LVM.zip, yang berisikan Debian Etch VMware image yang akan digunakan dalam tutorial ini. Image file ini dapat Anda buka menggunakan VMWare Player yang dapat di-*download* dari url *http://www.vmware.com/products/player/*. Untuk cara menggunakan VMWare Player, Anda dapat mempelajarinya dari url berikut, *http://www.howtoforge.com/import_vmware_images*.

Sistem ini memiliki enam buah SCSI harddisk, yaitu /dev/sda-/dev/sdf. Partisi /dev/sda akan digunakan untuk sistem Debian Etch, dan partisi /dev/sdb-/dev/sdf untuk LVM dan RAID. Partisi /dev/sdb-/dev/sdf memiliki kapasitas sekitar 80 GB.

Sebagai contoh awal, kita akan menggunakan sekitar 25GB kapasitas yang tersedia dari masing-masing harddisk. Dalam contoh ini kita juga akan mendemonstrasikan bagaimana cara memindahkan kapasitas dari harddisk yang berukuran kecil ke harddisk yang berukuran lebih besar (sebagai contoh, memindahkan harddisk yang memiliki kapasitas 25 GB ke harddisk yang memiliki kapasitas 80 GB).

Konfigurasi LVM

Kami anggap, Anda sudah dapat menginstalasi VMWare Player dengan baik. Jalankan VMWare Player, lalu klik menu *File -> Open*. Pilih file Other Linux 2.6.x kernel.vmx, yang terdapat pada direktori Debian Etch VMware image, yang telah Anda ekstrak.

Klik icon *Power On* untuk mulai menjalankan Debian Etch VMware image. Setelah menampilkan login prompt, login dengan username 'root' dan password 'howtoforge' (jangan sertakan tanda kutip).

Setelah masuk ke prompt, jalankan perintah berikut untuk merubah keyboard keymap ke US English.

```
# dpkg-reconfigure console-data
```

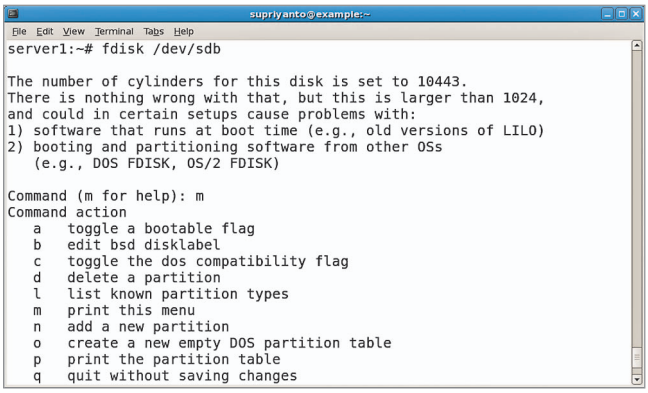
Pada halaman Configuring console-data, pilih *Select keymap from arch list*. Setelah itu, pilih qwerty pada pilihan Keyboard layout family. Pilih US american pada pilihan Keyboard layout. Pada keyboard variant, pilih *Standard*. Terakhir pada pilihan Keymap, pilih Standard.

Setelah keymap keyboard Anda sudah terkonfigurasi dengan baik, sekarang kita akan mulai masuk pembahasan utama. Pertama, jalankan perintah fdisk untuk melihat informasi yang terdapat di harddisk Debian Etch VMWare image.

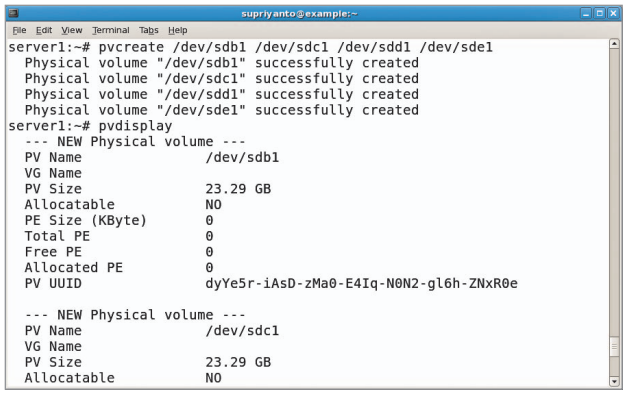
```
server1:~# fdisk -l
```

Dari hasil output perintah di atas, dapat terlihat kalau belum terdapat partisi sama sekali pada harddisk /dev/sdb-/dev/sdf. Untuk itu, kita akan membuat partisi /dev/sdb1, /dev/sdc1, /dev/sdd1, /dev/sde1, dan membiarkan /dev/sdf. Sebagai latihan, kita akan membuat kapasitas sebesar 25 GB yang akan diberikan ke partisi /dev/sdb1-/dev/sde1. Jalankan perintah berikut, untuk mulai membuat partisi harddisk:

```
server1:~# fdisk /dev/sdb
.....
.....
Command (m for help): n

(Note : ketikkan n untuk membuat
sebuah partisi baru)
.....
.....
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1

(Note : ketikkan p, kemudian pilih
1, untuk membuat sebuah partisi di
primary partition di partisi ke-1)
.....
.....
First cylinder (1-10443, default 1):
Using default value 1
Last cylinder or +size or +sizeM or
+sizeK (1-10443, default 10443):
+25000M

(Note : Tekan 1 atau isikan 1 untuk
first cylinder, kemudian isikan
+25000M untuk membuat partisi
sebesar 25GB)
.....
.....
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 8e
Changed system type of partition 1
to 8e (Linux LVM)
```



Proses pembuatan partisi sebesar 25 GB ke empat buah harddisk.



Pembuatan partisi LVM dan menampilkan hasilnya.

```
(Note : Isikan t untuk memberikan
sistem di partisi yang baru saja
kita buat. Lalu isikan 8e di
option Hex code untuk memberikan
Linux LVM sebagai file systemnya).
.....
.....
Command (m for help): w
The partition table has been
altered!
```

```
(Note : Isikan w untuk menulis
semua perubahan option yang telah
dilakukan di fdisk).
.....
.....
Calling ioctl() to re-read
partition table.
Syncing disks.
```

Setelah membuat partisi /dev/sdb1 sebesar 25 GB di /dev/sdb, lakukan hal yang sama untuk membuat partisi sebesar 25GB di /dev/sdc-/dev/sde.

```
server1:~# fdisk /dev/sdc
server1:~# fdisk /dev/sdd
server1:~# fdisk /dev/sde
```

Kemudian jalankan perintah berikut untuk melihat apakah partisi yang baru saja dibuat sudah ada.

```
server1:~# fdisk -l
.....
.....
Disk /dev/sde: 85.8 GB,
85899345920 bytes
255 heads, 63 sectors/track, 10443
cylinders
Units = cylinders of 16065 * 512 =
8225280 bytes
```

```
   Device Boot      Start
End      Blocks   Id  System
/dev/sde1               1
3040    24418768+  8e  Linux LVM
.....
.....
.....
Disk /dev/sdf doesn't contain a va
lid partition table
```

Dari hasil output perintah di atas, dapat terlihat kalau kita sudah berhasil membuat partisi sebesar 25 GB di masing-masing partisi /dev/sdb1-/dev/sde1.

Berikutnya, kita akan membuat sebuah partisi baru untuk LVM.

```
server1:~# pvcreate /dev/sdb1
/dev/sdc1 /dev/sdd1 /dev/sde1
  Physical volume "/dev/sdb1"
successfully created
  Physical volume "/dev/sdc1"
successfully created
  Physical volume "/dev/sdd1"
successfully created
  Physical volume "/dev/sde1"
successfully created
```

Sebagai latihan untuk menghapus partisi LVM yang baru saja Anda buat, jalankan perintah berikut:

```
server1:~# pvremove /dev/sdb1
/dev/sdc1 /dev/sdd1 /dev/sde1
  Labels on physical volume "/dev/
sdb1" successfully wiped
  Labels on physical volume "/dev/
sdc1" successfully wiped
  Labels on physical volume "/dev/
sdd1" successfully wiped
  Labels on physical volume "/dev/
sde1" successfully wiped
```

Buat kembali, dengan menggunakan perintah pvcreate.

```
server1:~# pvcreate /dev/sdb1
/dev/sdc1 /dev/sdd1 /dev/sde1
```

Untuk melihat status saat ini yang terjadi pada physical volume Anda, jalankan perintah pvdisplay.

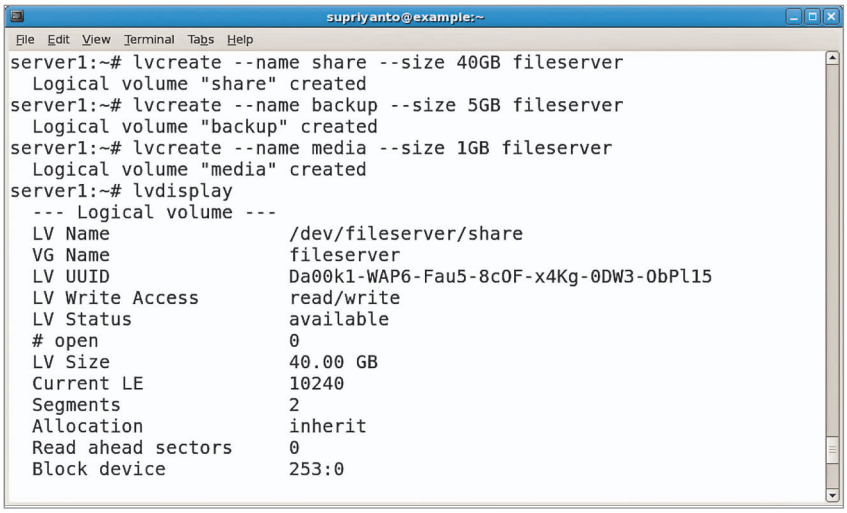
```
server1:~# pvdisplay
.....
.....
  --- NEW Physical volume ---
  PV Name                /dev/sde1
  VG Name
  PV Size                23.29 GB
  Allocatable            NO
  PE Size (KByte)        0
  Total PE               0
  Free PE                0
  Allocated PE           0
  PV UUID                OYJDSN-
VjSl-u0A6-AFBK-iOEf-K7WA-Npc5S0
```

Langkah selanjutnya, kita akan membuat sebuah volume group bernama fileserver dan menambahkan partisi /dev/sdb1-/dev/sde1 ke dalam volume group fileserver.

```
server1:~# vgcreate fileserver
/dev/sdb1 /dev/sdc1 /dev/sdd1
/dev/sde1
  Volume group "fileserver"
successfully created
```

Untuk melihat status volume group yang baru dibuat, jalankan perintah berikut:

```
server1:~# vgdisplay
  --- Volume group ---
  VG Name               fileserver
  System ID
  Format                lvm2
.....
```



Pembuatan logical volume di volume group file server.

```
  Free  PE / Size       23844 /
93.14 GB
  VG UUID               6Ne4jD-
9DGL-1Kku-Wxw1-Yh39-8T4Y-VsTfdL
```

Perintah lainnya, yang dapat digunakan untuk men-*scan* volume group yang terdapat pada harddisk dan *rebuild caches* adalah vgscan.

```
server1:~# vgscan
  Reading all physical volumes.
This may take a while...
  Found volume group "fileserver"
using metadata type lvm2
```

Untuk latihan, kita juga dapat mengubah name volumegroup fileserver menjadi nama lainnya (misal data), dengan menggunakan vgrename.

```
server1:~# vgrename fileserver data
  Volume group "fileserver"
successfully renamed to "data"

server1:~# vgdisplay
  --- Volume group ---
  VG Name               data
  System ID
  Format                lvm2
.................................
  Free  PE / Size       23844 /
93.14 GB
  VG UUID               6Ne4jD-
9DGL-1Kku-Wxw1-Yh39-8T4Y-VsTfdL
```

Sebagai latihan, kita juga dapat menghapus volume group data dengan menggunakan perintah vgremove

```
server1:~# vgremove data
  Volume group "data" successfully
removed
```

Untuk memastikan kalau volume group data sudah tidak terdapat lagi pada harddisk, Anda dapat menggunakan perintah vgdisplay dan vgscan. Setelah itu, buat kembali volume group fileserver yang telah dijelaskan sebelumnya, dengan menggunakan vgcreate.

```
server1:~# vgcreate fileserver
/dev/sdb1 /dev/sdc1 /dev/sdd1
/dev/sde1
  Volume group "fileserver"
successfully created
```

Selanjutnya kita akan mempelajari cara membuat logical volume. Sebagai latihan, kita akan membuat logical volumes bernama share (40 GB), backup (5 GB), dan media (1 GB) dalam volume group fileserver. Ketiga logical volume ini hanya menggunakan kapasitas yang tidak mencapai 50% dari kapasitas yang tersedia (yang nantinya dapat digunakan untuk membuat RAID1). Untuk membuat logical volumes yang telah disebutkan, jalankan perintah berikut:

```
server1:~# lvcreate --name share
--size 40G fileserver
  Logical volume "share" created
server1:~# lvcreate --name backup
--size 5G fileserver
  Logical volume "backup" created
server1:~# lvcreate --name media
--size 1G fileserver
  Logical volume "media" created
```

Untuk mengetahui informasi logical volumes yang baru saja dibuat, Anda dapat menggunakan perintah lvdisplay.

```
server1:~# lvdisplay
  --- Logical volume ---
  LV Name                /dev/
fileserver/share
  VG Name                fileserver
.................................
.................................
  Read ahead sectors     0
  Block device           253:2
```

Perintah lainnya, yang dapat digunakan untuk menscan logical volume yang terdapat pada harddisk adalah lvscan.

```
server1:~# lvscan
  ACTIVE            '/dev/
fileserver/share' [40.00 GB]
inherit
  ACTIVE            '/dev/
fileserver/backup' [5.00 GB]
inherit
  ACTIVE            '/dev/
fileserver/media' [1.00 GB]
inherit
```

Untuk latihan, kita akan mengubah logical volume yang bernama media menjadi films dengan menggunakan perintah lvrename.

```
server1:~# lvrename fileserver
media films
  Renamed "media" to "films" in
volume group "fileserver"

server1:~# lvdisplay
.................................
  --- Logical volume ---
  LV Name                /dev/
fileserver/films
```

Lalu, hapus logical volume films.

```
server1:~# lvremove /dev/
fileserver/films
Do you really want to remove
active logical volume "films"?
[y/n]: y
  Logical volume "films"
successfully removed
```

Buat kembali logical volume media.

```
server1:~# lvcreate --name media
--size 1G fileserver
  Logical volume "media" created
```

Sekarang kita akan menambah kapasitas logical volume media dari 1 GB menjadi 1.5

GB dengan menggunakan perintah lvextend.

```
server1:~# lvextend -L1.5GB /dev/
fileserver/media
  Extending logical volume media
to 1.50 GB
  Logical volume media
successfully resized
```

Kurangi kembali ukurannya menjadi berukuran 1GB dengan menggunakan perintah lvreduce.

```
server1:~# lvreduce -L1G /dev/
fileserver/media
  WARNING: Reducing active logical
volume to 1.00 GB
  THIS MAY DESTROY YOUR DATA
(filesystem etc.)
Do you really want to reduce
media? [y/n]: y
  Reducing logical volume media to
1.00 GB
  Logical volume media
successfully resized
```

Sampai saat ini kita telah memiliki tiga buah logical volume. Tetapi, ketiga buah logical volume tersebut belum memiliki filesystem sehingga kita tidak dapat melakukan save file apapun pada logical volume tersebut. Untuk itu, kita akan memberikan ext3 filesystem pada logical volume di share, xfs filesystem di backup, dan reiserfs filesystem di media.

```
server1:~# mkfs.ext3 /dev/
fileserver/share
mke2fs 1.40-WIP (14-Nov-2006)
Filesystem label=
OS type: Linux
.....
.....
Use tune2fs -c or -i to override.

server1:~# mkfs.xfs /dev/
fileserver/backup
meta-data=/dev/fileserver/
backup isize=256    agcount=8,
agsize=163840 blks
.....
.....
realtime =none
extsz=65536  blocks=0, rtextents=0

server1:~# mkfs.reiserfs /dev/
fileserver/media
mkfs.reiserfs 3.6.19 (2003 www.
namesys.com)
.....
.....
ReiserFS is successfully created on
/dev/fileserver/media.
```

Sekarang kita siap untuk melakukan mount ke logical volumes. Contoh, kita akan melakukan mount logical volume share ke /var/share, backup ke /var/backup, dan media ke /var/media. Untuk itu, kita harus membuat semua direktori itu terlebih dahulu.

```
server1:~# mkdir -p /var/media
/var/backup /var/share
```

Sekarang, kita dapat memproses mount ke logical volume yang baru saja dibuat.

```
server1:~# mount /dev/fileserver/
share /var/share
server1:~# mount /dev/fileserver/
backup /var/backup
server1:~# mount /dev/fileserver/
media /var/media
```

Jalankan juga perintah di bawah ini untuk melihat sisa kapasitas logical volumes yang baru saja kita mount.

```
$ df -h
Filesystem            Size  Used
Avail Use% Mounted on
.....
.....
/dev/mapper/fileserver-share
                       40G  176M
38G   1% /var/share
/dev/mapper/fileserver-backup
                      5.0G  144K
5.0G   1% /var/backup
/dev/mapper/fileserver-media
                      1.0G   33M
992M   4% /var/media
```

Sampai saat ini, kita sudah mengonfigurasi sistem LVM pertama kita. Sekarang, Anda dapat membaca dan menulis dari partisi /var/share, /var/backup, dan /var/media.

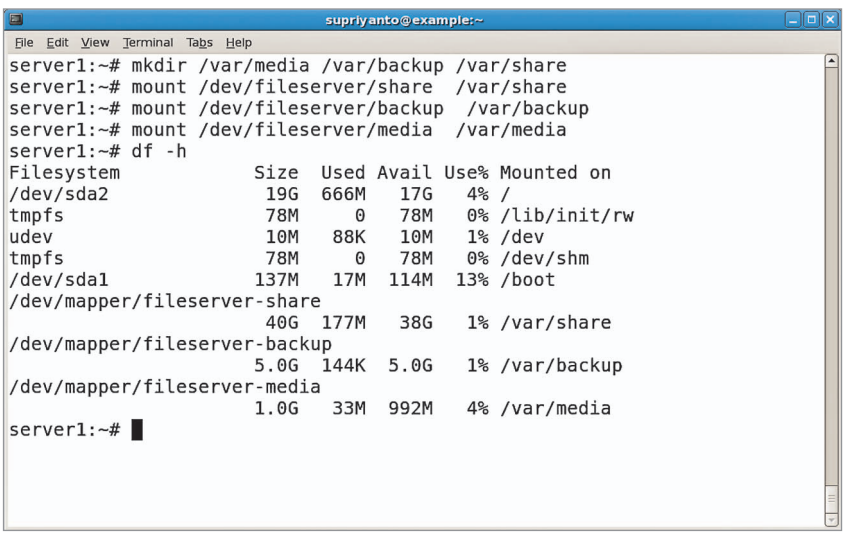
Agar proses mount partisi /var/share, /var/backup, dan /var/media, dapat berjalan secara otomatis, Anda dapat menambahkan baris berikut ke dalam file /etc/fstab.

```
server1:~# vim /etc/fstab
# /etc/fstab: static file system
information.
.....
.....
/dev/fileserver/share   /var/share
ext3       rw,noatime    0 0
/dev/fileserver/backup  /var/
backup     xfs        rw,noatime
0 0
/dev/fileserver/media   /var/media
reiserfs   rw,noatime    0 0
```

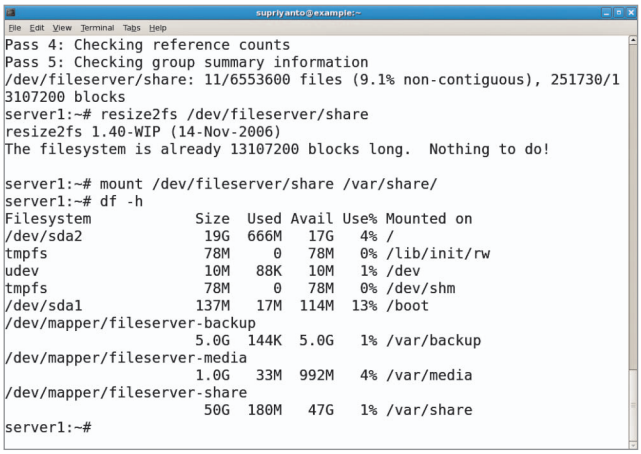
Setelah komputer *restart*, secara otomatis ketiga partisi tersebut akan langsung ter-mount pada direktori yang telah disebutkan.

Mengubah ukuran logical volumes dan filesystem

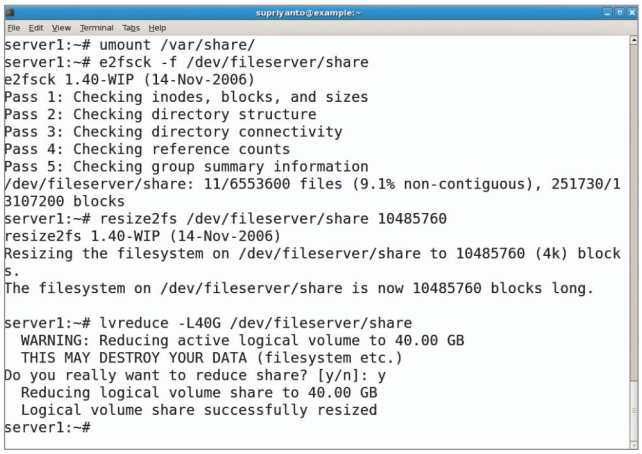
Setelah kita mengetahui konsep dasar LVM. berikutnya kita akan belajar cara mengubah ukuran logical volume share yang meng-



Mounting tiga volume LVM ke direktori yang ada di /var.



Merubah ukuran logical volume share dan melihat kapasitasnya.



Memperkecil ukuran logical volume share, dan melihat kapasitasnya saat di mount.

gunakan ext3 filesystem. Sebelum melakukan hal ini, lakukan proses umount partisi tersebut terlebih dahulu.

```
server1:~# umount /var/share
```

Berikutnya, kita akan menambah kapasitas logical volume share dari 40 GB menjadi 50 GB. Untuk mengubah kapasitas logical volume, Anda dapat menggunakan perintah lvextend.

```
server1:~# lvextend -L50G /dev/
fileserver/share
  Extending logical volume share to
50.00 GB
  Logical volume share successfully
resized
```

Saat ini, kita hanya membesarkan logical volume share, bukan ext3 filesystem yang terdapat di share. Untuk itu, kita akan mengubah juga besar ext3 filesystem yang terdapat pada logical volume share.

```
server1:~# e2fsck -f /dev/
fileserver/share
e2fsck 1.40-WIP (14-Nov-2006)
.....................................
/dev/fileserver/share: 11/5242880
files (9.1% non-contiguous),
209588/10485760 blocks
```

Catat total blocks yang terdapat pada partisi tersebut (10485760), yang kita butuhkan untuk memperkecil kembali ukuran logical volume tersebut nanti. Sekarang, lakukan perubahan ukuran ext3 filesystem logical volume share.

```
server1:~# resize2fs /dev/
fileserver/share
resize2fs 1.40-WIP (14-Nov-2006)
Resizing the filesystem on /dev/
fileserver/share to 13107200 (4k)
blocks.
The filesystem on /dev/fileserver/
share is now 13107200 blocks long.
```

Mount kembali partisi tersebut.

```
server1:~# mount /dev/fileserver/
share /var/share
```

Pada saat kita melihat kapasitas yang terdapat pada logical volume share yang di-mount, dapat terlihat kalau kapasitas logical volume share sudah berubah menjadi 50 GB, dari sebelumnya yang hanya 40 GB.

Selain memperbesar kapasitas logical volume, kita juga dapat memperkecil kapasitas logical volume. Sebagai contoh, kita akan memperkecil kapasitas logical volume share menjadi 40 GB kembali.

Pertama, umount kembali logical volume share yang telah di-mount ke partisi /var/share.

```
server1:~# umount /var/share
```

Lakukan check partisi logical volume share.

```
server1:~# e2fsck -f /dev/
fileserver/share
```

Sekarang kita akan mengubah ukuran ext3 filesystem logical volume share menjadi 40 GB. Dari operasi sebelumnya, kita sudah mendapat catatan kalau 40 GB sama dengan 10485760 blocks. Jalankan perintah berikut untuk mengubah ukuran.

```
server1:~# resize2fs /dev/
fileserver/share 10485760
resize2fs 1.40-WIP (14-Nov-2006)
Resizing the filesystem on /dev/
fileserver/share to 10485760 (4k)
blocks.
The filesystem on /dev/fileserver/
share is now 10485760 blocks long.
```

Setelah memperkecil kapasitas filesystem, kita juga perlu memperkecil ukuran logical volume-nya juga. Jalankan perintah berikut untuk melakukannya.

```
server1:~# lvreduce -L40G /dev/
fileserver/share
  WARNING: Reducing active logical
volume to 40.00 GB
  THIS MAY DESTROY YOUR DATA
(filesystem etc.)
Do you really want to reduce share?
[y/n]: <-- y
  Reducing logical volume share to
40.00 GB
  Logical volume share successfully
resized
```

Setelah melakukan langkah di atas, mount kembali partisi logical volume share yang sudah diperkecil kembali ukurannya.

```
server1:~# mount /dev/fileserver/
share /var/share
```

Selanjutnya, lihat status kapasitas partisi yang baru saja dirubah.

```
server1:~# df -h
```

Sampai disini dahulu tutorial LVM untuk edisi ini. Pada bagian berikutnya, kita akan mempelajari metode penggunaan LVM yang lain, dalam memanajemen kapasitas harddisk. Sampai jumpa !.

Supriyanto [supriyanto@infolinux.co.id]

Plot dan Analisis Data dengan SciGraphica

Plot dan analisis data biasanya digunakan untuk membantu dalam interpretasi data. Microcal Origin adalah perangkat lunak untuk plot dan analisis data di lingkungan Windows yang luar biasa. Sayangnya, Microcal Origin hanya dibuat untuk Windows dan tidak gratis. Bagaimana dengan aplikasi serupa di Linux?

Ada beberapa perangkat lunak yang sama fungsinya dengan Microcal Origin, salah satunya adalah SciGraphica.

1. Instalasi

Sampai saat ini SciGraphica sudah merilis versi 2.1.0. Namun, Anda dapat men-*download* SciGraphica versi 2.0.0 yang sudah stabil di situs *sourceforge.net*. Tiga paket yang harus Anda download untuk instalasi Scigraphica versi 2.0.0 yaitu gtk+extra-2.0.0.tar.gz (873,3 KB), libscigraphica-2.0.0.tar.gz (758.0 KB) dan scigraphica-2.0.0.tar.gz (526.2 KB). Lakukan instalasi untuk ketiga paket tersebut secara berurutan dengan perintah berikut ini:

```
# tar -xzvf namapaket-2.0.0.tar.gz
# ./configure
# make
# make check
# make install (user root)
```

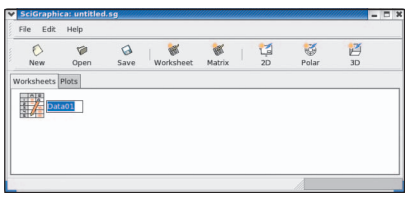
Karena SciGraphica menggunakan modul yang ditulis dengan intepreter Python untuk penyelesaian analisis numerik maka kegagalan instalasi biasanya karena tidak adanya salah satu paket pendukung Python, yaitu Numpy atau Numarray. Anda juga dapat men-download-nya di sourceforge.net (numarray-1.5.2.tar.gz (1.1 MB), numpy-1.0.1.tar.gz (1.2 MB)), Jika Anda tidak ingin repot Anda juga dapat men-download binary dari 5 paket tersebut. Jadi sebaiknya, sebelum melakukan instalasi SciGraphica selain Anda menggunakan GNOME (Gtk+) juga Anda instalasikan Python serta paket pendukungnya.

2. Mari Memulai

Penulis melakukan instalasi SciGraphica di RedHat 9, Untuk memulainya di terminal berikan perintah berikut:

```
$ /usr/local/bin/scigraphica &
```

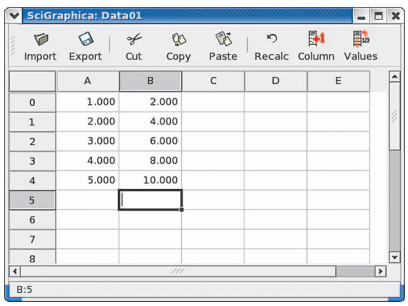
Kali pertama Anda akan disuguhi GUI dari SciGraphica (Gambar 1), Anda dapat membuka *Worksheets* tempat memasukkan data dengan melakukan klik dua kali pada icon Worksheets. Penulis mencoba dengan memasukkan data yang sederhana seperti pada Gambar 2. Simpan dan beri nama pekerjaan Anda (*File* > *Save as*).



Gambar 1. Graphical User Interface dari SciGraphica.

Setelah itu, lakukan plot untuk data Anda dengan cara memilih icon 2D kemudian klik dua kali icon plot hingga tampil layar tempat plot grafik. Hal pertama yang harus Anda lakukan sebelum memplot data adalah mengatur skala dari grafik plot dengan cara klik dua kali di daerah grafik plot hingga tampil Layer Control seperti pada Gambar 3.

Klik Axes kemudian atur skala untuk sumbu X dan sumbu Y disesuaikan dengan



Gambar 2. Memasukkan data pada Worksheets.

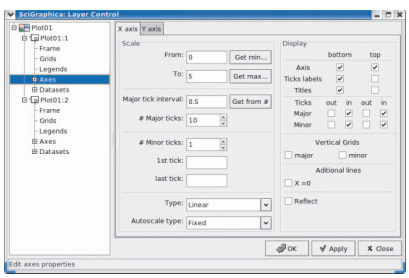
data yang telah Anda masukkan pada Worksheets penulis mengatur skala sumbu X dari 0 sampai 5 dan skala sumbu Y dari 0 sampai 10, Seperti Gambar 3. Setelah itu, plot data Anda dengan cara:

Tekan tombol Add dataset untuk masukkan data yang telah Anda buat.

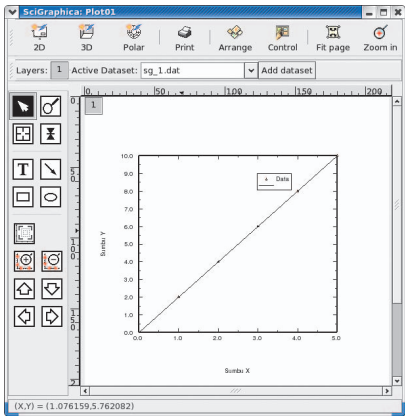
Kemudian pilih cara data diplot. Penulis memilih melakukan plot dengan X-Y Scatter.

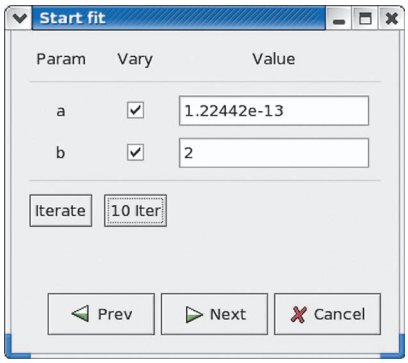
Setelah data diplot kemudian penulis mencoba kemampuan Fit Curve dari SciGraphica dengan memilih icon Fit Curve pada GUI plot.

Karena data yang dibuat adalah data linear, maka penulis melakukan *fitting* kurva



Gambar 3. Mengatur skala plot untuk grafik.





Gambar 4 a dan b. Plot X-Y Scatter dan Hasil Fit Curve pada Scigraphica dengan data linear.

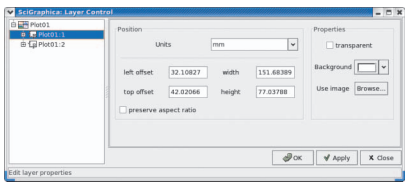
linear dan hasilnya fitting dan plot ditunjukkan seperti Gambar 4 a dan b.

Anda dapat menyimpan hasil plot grafik tersebut dalam format *.sgp atau Anda juga dapat meng-export hasil plot grafik tersebut dalam format lain. Mudah, bukan?

Plot linear adalah contoh yang mudah sehingga kelebihan Scigraphica tidak dapat diunggulkan dengan software kelas office seperti Microsoft Excel atau OpenOffice Calc. Untuk urusan fitting kurva plot Anda diberikan banyak fungsi yang kompleks untuk melakukan interpolasi semacam fungsi kuadratik, logaritmik, eksponensial, atau bahkan fungsi yang lebih kompleks seperti Gauss, Boltzmann semua fungsi fitting kurva tersebut dapat diselesaikan dengan Scigraphica.

3. Bekerja dengan Layer

Sering kali dalam melakukan plot grafik, Agar mudah membandingkan dua atau lebih data. Grafik ditempatkan pada satu layer baik disatukan atau dipisahkan artinya terdapat dua grafik atau lebih pada satu layer. Hal ini dapat juga dilakukan oleh SciGraphica dengan cara:



Gambar 5 Mengatur posisi plot grafik dari masing-masing layer.

Klik kanan plot yang telah Anda buat kemudian tambahkan layer baru (*Layer* > *New* > *2D*).

Layer baru ini dapat Anda atur posisinya yaitu dengan cara membuka *Layer Control* (klik dua kali plot grafik).

Atur posisi dari plot tersebut dengan memilih nama dari plot grafik Anda pada Tab Position seperti pada Gambar 5.

Penulis melakukan plot dua grafik pada satu layer dengan terlebih dahulu menambah Worksheets untuk data kedua atau Anda dapat juga menambahkan data pada Worksheets yang pertama. Langkahnya sama seperti dituliskan pada langkah pertama dan hasilnya seperti pada Gambar 6.

Layer sangat berguna terutama jika Anda harus membandingkan dua plot grafik yang menggunakan sumbu X-Y dengan range yang sama. Jika data yang Anda buat diperoleh dari kondisi yang berbeda namun masih menggunakan parameter yang sama, maka Anda dapat menyimpulkan pengaruh yang terjadi akibat pemberian kondisi yang berbeda tersebut hanya dengan menganalisis plot dari data yang berbeda tersebut. Untuk itu Anda perlu menggabungkan grafik yang Anda miliki. Layer di Scigraphica dapat dibuat lebih dari dua tergantung kebutuhan Anda.

4. Plot 3D dan visualisasi kontur

Plot 2D membantu Anda dalam intepretasi data yang hanya memiliki dua parameter. Bagaimana untuk data yang memiliki tiga parameter, katakanlah terdiri dari sumbu X, Y dan Z? Artinya, Anda memiliki minimal tiga informasi yang dapat digunakan untuk interpretasi data. Data dengan tiga parameter dapat divisualisasikan dalam bentuk 3D ataupun kontur. Scigraphica memberikan kemudahan untuk melakukan plot 3D atau visualisasi kontur. Biasanya lebih mudah melakukan interpretasi data dengan plot 3D atau visualisasi kontur jika memberikan gradasi warna, dengan demikian warna-warna tersebut dapat dijadikan indikasi bagi Anda untuk menganalisis data.

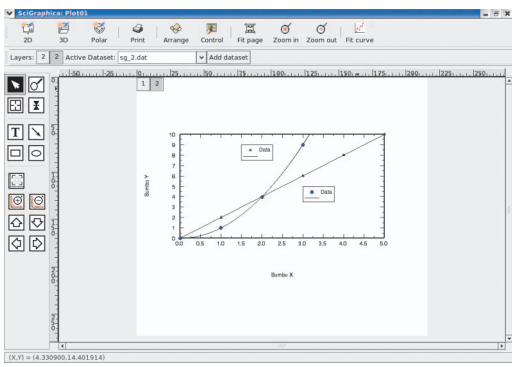
Pengaturan warnapun dapat dilakukan dengan mudah menggunakan Scigraphica, Anda hanya perlu memberikan indikasi warna pada nilai yang Anda miliki. Misalnya nilai maksimum Anda berikan warna merah dan nilai minimum diberikan indikasi warna hijau. Gradasi yang dihasilkan secara otomatis akan diselesaikan oleh Scigraphica. Berikut ini contoh plot 3D dan visualisasi kontur dengan data yang dimiliki penulis.

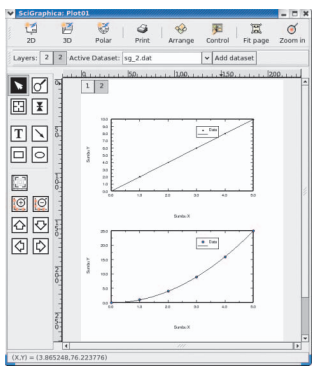
Bagaimana dengan plot 3D atau visualisasi dalam bentuk kontur. Kebetulan penulis memiliki contoh data 3D yang akan diplot kemudian dibuat juga konturnya. Untuk melakukan plot 3D, caranya:

Pilih icon 3D lalu klik dua kali icon plot setelah itu atur skala plot sesuai dengan data yang ada.

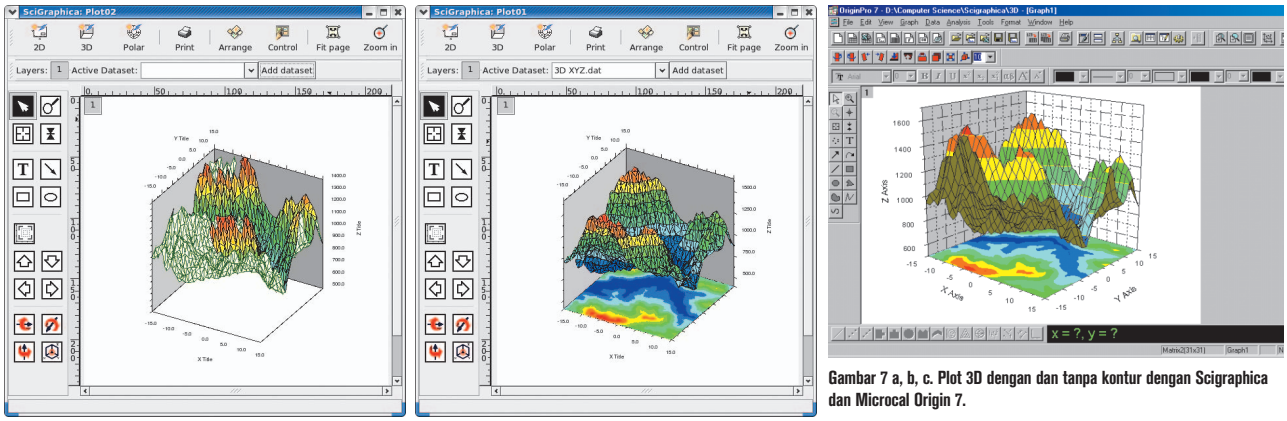
Kemudian pada Add dataset tambahkan datanya. Anda akan diberikan pilihan untuk memplot data. Plot 3D Surface untuk plot 3D sedangkan untuk plot 3D dan juga konturnya Anda dapat memilih 3D Contour. Hasilnya ditunjukkan pada Gambar 7. Sebagai perbandingan plot 3D kontur dari Microcal Origin juga diperlihatkan.

Gradasi warna dapat diatur dengan klik dua kali hasil plot kemudian pilih (*Dataset* >





Gambar 6. Plot dua grafik dengan penambahan layer.



Gambar 7 a, b, c. Plot 3D dengan dan tanpa kontur dengan Scigraphica dan Microcal Origin 7.

Properties) pada Tab Levels + Colors seperti ditunjukkan Gambar 8.

Anda dapat mengatur skala warna, sedangkan penambahan kontur dapat dilakukan dengan memilih Tab Contour dan kostumisasi kontur pun dapat dilakukan.

Keunggulan Scigraphica dalam hal melakukan plot 3D dan visualisasi kontur adalah data yang digunakan untuk melakukan plot disiapkan dalam format *.dat dan disusun sesuai dengan parameter yang digunakan. Misalnya kolom pertama untuk sumbu X, kolom kedua untuk sumbu Y, dan seterusnya. Hal ini biasa dilakukan jika Anda mengambil data yang diubah ke format ASCII sehingga Anda dapat langsung menggunakannya untuk melakukan plot. Tidak demikian halnya jika Anda menggunakan Microcal Origin, data yang telah diformat menjadi ASCII harus terlebih dahulu diubah menjadi susunan matrix agar dapat melakukan plot 3D atau visualisasi kontur. Ini selain merepotkan juga akan menambah penuh ruang harddisk Anda untuk menyimpan data matrix tersebut.

Jika ditinjau dari hasil plot Microcal Origin tidak lebih baik dari Scigraphica. Hal ini juga yang dibanggakan oleh pembuat Scigraphica seperti pada gambar 7 dan 8. Plot yang dihasilkan Microcal Origin adalah hasil konversi ke matrix sehingga agak berbeda dengan plot dari Scigraphica.

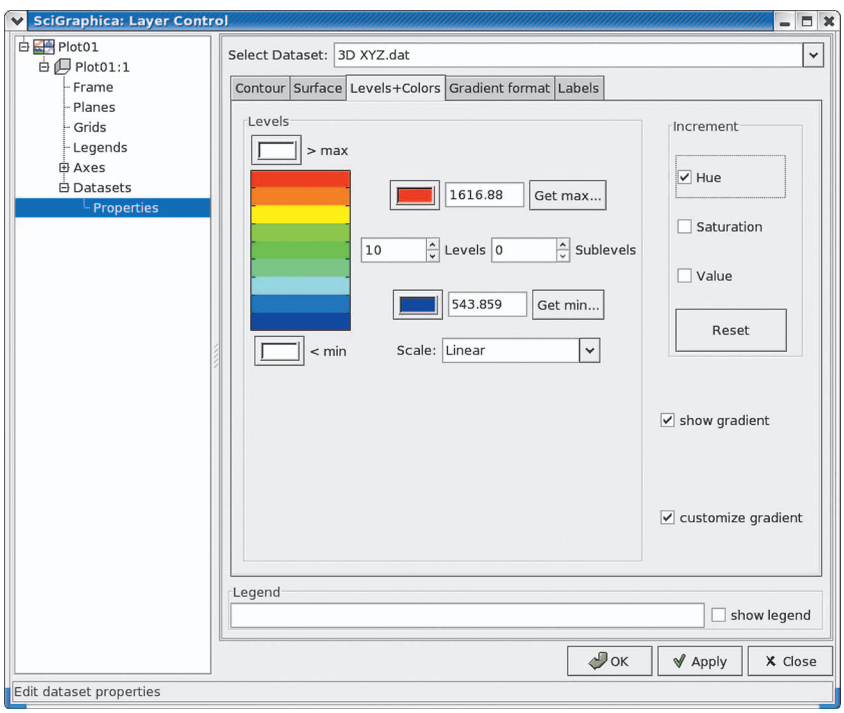
5. Microcal Origin dan Scigraphica

Apa yang dikatakan oleh pembuat Scigraphica adalah kloning dari Microcal Origin memang benar adanya. Menurut penulis yang pernah menggunakan Microcal Origin, Scigraphica hampir memiliki kemampuan yang sama dengan Microcal Origin. Misalnya untuk urusan fitting kurva fungsi-fungsi yang disediakan cukup banyak seperti gauss, boltzmann, dan lain-lain. Kemudian kemampuan layer dari Scigraphica juga luar biasa bahkan menurut penulis lebih mudah dibandingkan Microcal Origin.

Demikian pula dalam hal plot 3D kontur, Scigraphica dapat langsung melakukan plot dari data *.dat tidak seperti Microcal Origin yang mengharuskan data *.dat diubah terlebih dahulu dalam bentuk matrix. Satu hal lain yang menarik bagi penulis adalah plot grafik yang dihasilkan oleh Scigraphica lebih baik jika dibandingkan dengan plot grafik Microcal Origin, dilihat dari kualitas cetaknya.

Bagaimana dengan fungsi-fungsi yang tidak umum digunakan untuk melakukan fitting kurva? Jika di Microcal Origin fungsi tersebut dapat ditambahkan dengan menggunakan compiler C yang disertakan sebagai paket Microcal Origin, maka Scigraphica memiliki intepreter Python dengan *site-package*-nya yang luar biasa dan dapat Anda gunakan untuk dijadikan modul sebagai tambahan fungsi yang Anda perlukan. Dengan masih terus dikembangkannya SciGraphica yang memberikan lingkungan plot yang WYSIWYG bukan tidak mungkin keberadaan Microcal Origin dapat digantikan, khususnya dalam hal perangkat lunak yang dapat berjalan di banyak platform, dan tanpa biaya lisensi alias gratis.

Said Sesiaria [sesiaria@gmail.com]



Gambar 8. Pengaturan skala warna untuk plot 3D.

Menaklukkan Modem Internal

Salah satu keuntungan membeli notebook kosongan (tanpa OS) adalah selain harganya jauh lebih murah (tapi tidak murahan), juga bebas menentukan pilihan sistem operasi yang akan digunakan. Berikut ini contoh pengalaman penulis dalam mengonfigurasi modem internal di sebuah notebook.

Dalam melakukan uji coba, penulis menggunakan notebook Acer 5583NWXMi. Notebook ini menjadi pilihan karena spesifikasinya cukup bagus, yaitu Processor Intel Core 2 Duo T5500 (1,66 GHz, 667 MHz FSB, EMT64, WiFi, i945GM, dan 512 DDR2), dengan harga cukup murah. Pilihan penulis jatuh pada distro Linux openSUSE 10.2, yang dalam satu paket terdiri dari 5 CD dan 1 CD tambahan. openSUSE 10.2 dirilis akhir Desember 2006 dan DVD-nya disertakan dalam majalah *InfoLINUX* edisi 02/2007. Spesifikasi openSUSE 10.2, antara lain: Kernel 2.6.18.2-34, KDE 3.5, GNOME 2.16, OpenOffice 2.0.4, termasuk driver atau modul kernel dari Smart Link Soft Modem.

PERMASALAHAN

Proses instalasi openSUSE 10.2 berjalan mulus. Dan kesan pertama penulis, ini baru Linux-ku! Giliran *setting* modem, ternyata modem-nya tidak langsung dapat digunakan. Bagaimana ini? Upaya pertama penulis adalah melakukan pengecekan ulang paket-paket yang sudah terinstalasi, barangkali ada yang tertinggal, atau masalah dependensi. Hasilnya tetap, modem tidak dikenali. Kemudian dengan perintah wvdialconf, tidak menemukan modem. Berikutnya dengan kppp, hasilnya sama saja.

Penulis kemudian melakukan instalasi ulang modul smartlink beserta kompilasi kernelnya. Hasilnya juga sama saja, modem belum dapat digunakan. Jadi, bagaimana *dong*?

Penulis teringat "Praktik Instan" di *InfoLINUX* edisi 08/2006 ("Trik Menjinakkan Modem Internal"). *Yah... this is the hard way to do the right thing*. Awalnya penulis paling malas dengan yang namanya *command line*. Justru untuk menghindarinya, maka penulis memilih openSUSE 10.2. Dengan adanya masalah modem ini, maka terpaksa penulis mencoba perintah baris di terminal.

Setelah berjam-jam mengikuti proses praktik instan tersebut, hasilnya seperti berikut ini:

```
linux-ku:/ # cd /sbin
linux-ku:/sbin # modprobe slamr
FATAL: Module slamr not found.
linux-ku:/sbin #
```

Artinya? Ya, gagal lagi! Bagaimana agar menemukan modemnya, bukankah prosedur sebelumya tidak terjadi kesalahan?

Akhirnya, seperti pepatah mengatakan kepalang basah, mandi saja sekalian! Jadi, dengan berbekal semangat dan kebulatan tekad, penulis mencoba dengan cara sendiri. Hitung-hitung sambil belajar Linux dan menambah pengalaman.

PROSEDUR

Langkah pertama, instalasi ulang modul smartlink modem.

Kedua, melakukan investigasi pustaka yang berhubungan dengan modem internal ini. Cara yang paling efektif menggali informasi adalah dari paket modem itu sendiri, di samping mengumpulkan berbagai hal yang berhubungan dengan permasalahan modem internal. Informasi dapat dicari pada situs-situs forum diskusi dan tanya-jawab di internet.

Langkah ketiga, berbekal informasi yag diperoleh, persiapkan jurus-jurus jitu "mengoprek" program eksekusi modem.

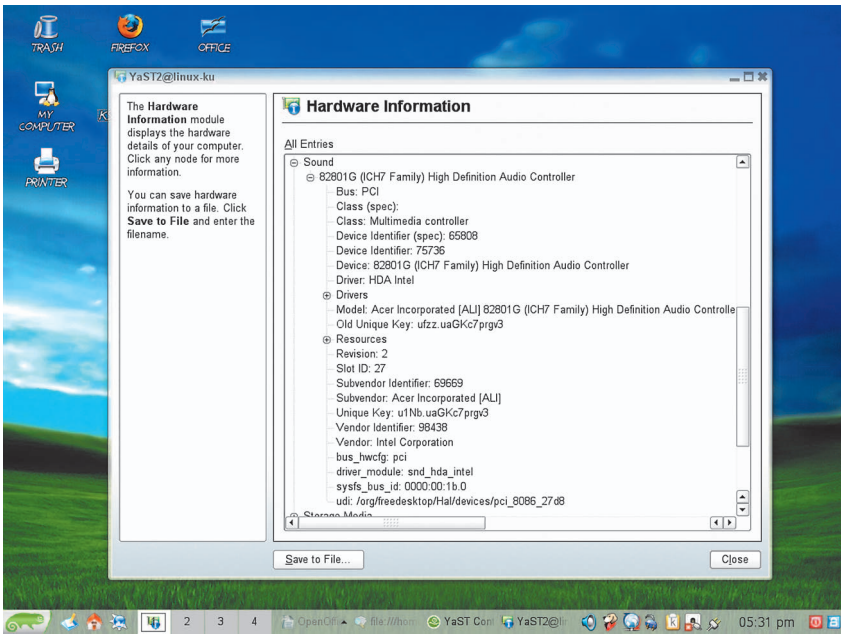
PENYELESAIAN

Pertama, kenali *hardware* (modem) yang kita miliki, bagaimana sistem kerja pada umumnya, dan pahami dukungan modem yang dimiliki oleh *software*.

Dari user's guide Aspire 5580/5570/3680 Series yang menyertai Notebook Acer Aspire 5583NWXMi, spesifikasi modem: 56K ITU V.92 with PTT approval; wake-on-ring ready. Maksudnya (secara umum) spesifikasi tersebut adalah standard modem *dial-up* (56 K), versi (ITU V.92), yang aktivitasnya (Tx/Rx) berdasarkan program (PTT).

Untuk mengetahui di mana lokasi dan identitas lain dari modem yang terpasang pada notebook Acer ini tidak perlu membongkar casing, cukup dengan menggunakan perintah "aplay -l" sebagai root, seperti berikut ini:

```
  root:/# aplay -l
**** List of PLAYBACK Hardware
Devices ****
card 0: Intel [HDA Intel], device
0: ALC883 Analog [ALC883 Analog]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: Intel [HDA Intel], device
```



Gambar 1. YaST menampilkan informasi hardware.

```
2: ALC883 Digital [ALC883 Digital]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: Intel [HDA Intel], device
6: Si3054 Modem [Si3054 Modem]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
```

Perintah "aplay -l" menginformasikan lokasi modem, yang berada pada card:0 dan device:6. Pada sistem driver ALSA dikenal sebagai hw:0 (bisa juga ditulis hw:0,6). *List of Playback* tersebut juga menginformasikan bahwa driver modem menggunakan driver soundcard HDA Intel dengan identitas device Si3054 Modem.

Selanjutnya perlu diketahui identitas modul sound card yang dikenali oleh kernel, sehingga pada saat *booting* dapat di-porting. Dengan membuka Yast hardware information, diketahui bahwa notebook ini menggunakan driver module: snd_hda_intel. Identitas modul driver sound card tersebut diperlukan nanti pada saat akan mengedit kernel pada konfigurasi sistem modem.

Kembali pada smartlink modem. Untuk mendapatkan informasi tentang modem ini, ada cara yang paling efektif, yaitu informasi dari software modem itu sendiri. Caranya dengan membuka file software yang sudah terinstalasi pada control center (Gambar 1). Informasi dari softmodem/README.ALSA menjelaskan bahwa softmodem ALSA mendukung driver modem "snd_hda_intel".

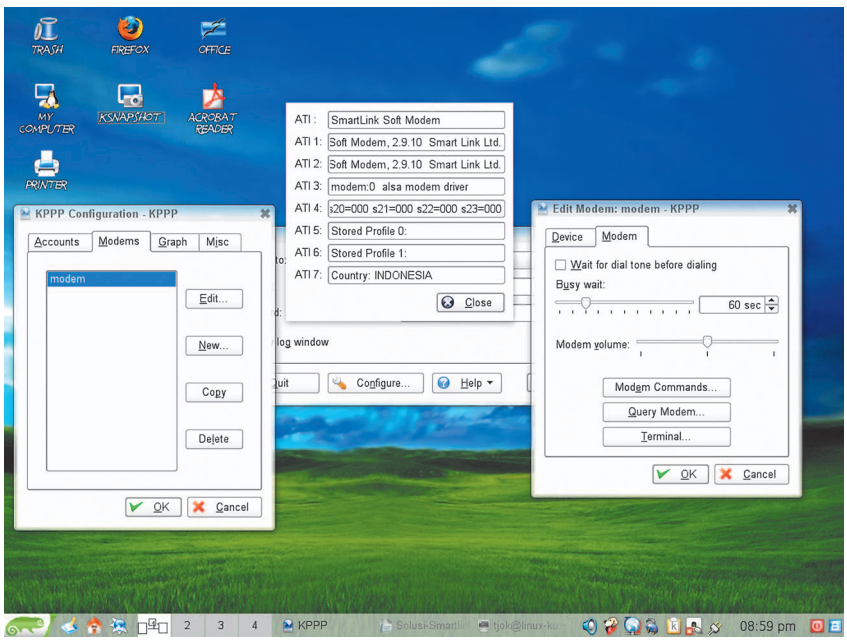
Langkah selanjutnya adalah mengikuti prosedur konfigurasi yang ditentukan. Namun tidak sepenuhnya sesuai dengan yang tertulis, karena ada hal-hal yang perlu disesuaikan lagi. Oleh sebab itu perlu dilakukan dengan cara yang lebih pasti, yakni dengan mengedit file slmodemd. Ada beberapa pilihan editor, seperti Vim (vi), pico, atau KEdit. Penulis menggunakan Vim. Berikut ini langkah-langkah editing slmodemd:

EDITING FILE SLMODEMD

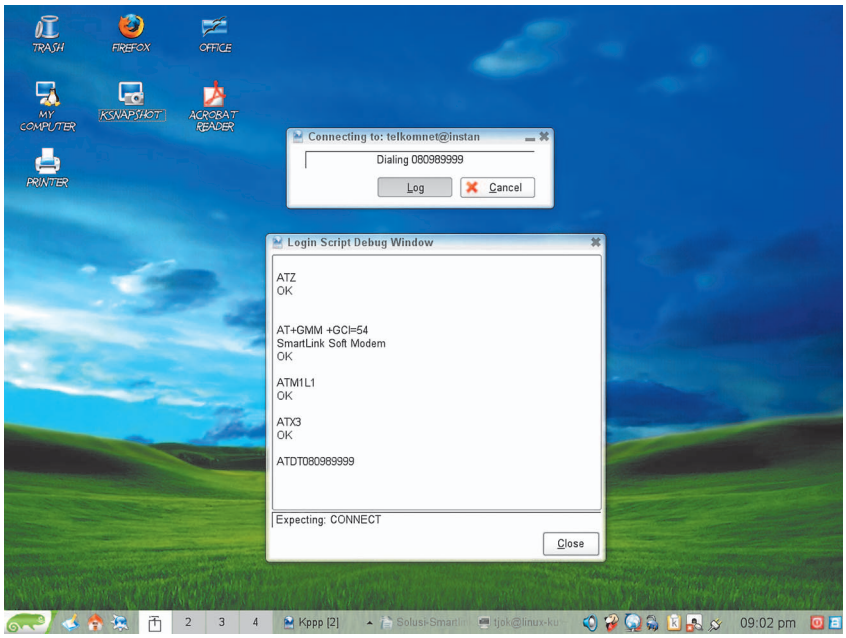
1. File /etc/sysconfig/slmodemd

Buka file slmodemd pada root /etc/sysconfig/ dengan perintah vi slmodemd, ganti SLMODEMD_COUNTRY dengan "INDONESIA", ganti SLMODEMD_DEVICE dengan "modem:0", dan SLMODEMD_USE_ALSA dengan "yes", seperti gambar berikut:

```
.............................
.............................
#
# Country, for which the modem
shall be configured for.
#
# A list of all supported
country names can be retrieved
# by calling "slmodemd --
countrylist" from the shell
prompt.
#
SLMODEMD_COUNTRY="INDONESIA"

## Type:   string
## Default:  "modem:1"
## ServiceRestart: slmodemd
#
# Hardware(not modem) device for
the slmodemd.
.............................
.............................

#
SLMODEMD_DEVICE="modem:0"

## Type:   yesno
```



Gambar 2. Tes atau query modem melalui program kppp.



Gambar 3. Kppp sedang melakukan dial-up.

```
## Default:  "no"
## ServiceRestart: slmodemd
#
# If set to yes the Advanced
Linux Sound Architecture
subsystem is used
# to make your modem working.
This variable must be set to yes
if you
# are using an ATI IXP modem.
#
SLMODEM_USE_ALSA="yes"

.............................
.............................
```

Kemudian tutup dan simpan file tersebut, langkah selanjutnya membuka file /etc/sysconfig/kernel.

2. File /etc/sysconfig/kernel
   Masih di root /etc/sysconfig lakukan editing terhadap kernel, pada baris MODULES_LOADED_ON_BOOT tambahkan "snd_hda_intel", seperti script berikut:

```
.............................
#
MODULES_LOADED_ON_BOOT="snd_hda_
intel"

## Type: string
## Default:     ""
#
.............................
```

   Simpan dan keluar dari editor. Sekarang lakukan perintah berikut "modprobe snd_hda_intel", diikuti Enter. Kemudian ketikkan perintah "insserv slmodemd". Berikutnya masuk ke direktori /etc/init.d/ untuk mengedit file slmodemd.

3. File /etc/init.d/slmodemd
   Bukalah file slmodemd, kemudian gantilah semua kata "slamr0" dengan "slmodemd". Simpan dan keluar dari editor, kemudian ketikkan perintah "stat slmodemd", maka akan tampil informasi perubahan slmodemd.

```
root:/etc/init.d # stat slmodemd
File: `slmodemd'
Size: 2912     Blocks: 8     IO
Block: 4096    regular file
.............................
.............................
Change: 2007-01-29
23:43:48.000000000 +0700
```

Untuk memastikan apakah sudah berhasil melakukan editing, ketikkan perintah berikut "slmodemd -a hw:0" maka akan tampil seperti berikut:

```
root:/etc/init.d # slmodemd -a
modem:0
SmartLink Soft Modem: version
2.9.10 Nov 29 2006 13:53:16
symbolic link `/dev/ttySL0' ->
`/dev/pts/3' created.
modem `modem:0' created. TTY is
`/dev/pts/3'
Use `/dev/ttySL0' as modem device,
Ctrl+C for termination.
```

Agar modem dikenali sebagai /dev/modem, maka lakukan perintah ln -sf untuk membuat *soft link* secara paksa:

```
# ln -sf /dev/ttySL0 /dev/modem
# ln -sf /dev/ttySL0 /lib/udev/
devices/modem.
```

Setelah di-reboot, cobalah *setting* dengan perintah kppp& pada root. Penulis menggunakan ISP telkomnet@instan. Untuk mengetahui respon dari modem, coba klik tombol Query Modem, kalau berhasil maka akan tampil seperti Gambar 2. Sedangkan, catatan koneksinya (Log) seperti Gambar 3. Setup internet juga dapat dilakukan melalui Yast.

KESIMPULAN

Dari proses penyelesaian masalah tersebut, ternyata persoalannya terletak pada desain program smart link soft modem itu sendiri. Pada setiap ekstraksi paket modul, smart link soft modem akan memerintahkan pembentukan modules dengan nama slamr dengan virtual device: /dev/slamr0. Padahal notebook Acer 5583NWXMi menggunakan chipset suara Intel (snd_hda_intel), yang termasuk dalam dukungan ALSA. Sehingga, ketika menggunakan driver modem ALSA, tidak perlu me-*load* modules slamr. Karena dalam hal ini driver modem ALSA hanya mengenal slmodemd dengan virtual device: /dev/slmodemd.

Sayangnya, hal tersebut tidak terjadi secara otomatis, jadi harus dilakukan secara manual. Oleh sebab itu, tanpa editing file /etc/init.d/slmodemd, maka setting modem akan gagal. Karena meskipun sudah dilakukan perubahan data pada /etc/sysconfig/slmodemd dan /etc/sysconfig/kernel, ternyata status data tidak mengalami perubahan otomatis pada file /etc/init.d/slmodemd.

Adanya perubahan identifikasi pada sistem modules kernel, menyebabkan setting modem tidak langsung dapat dikenali, sehingga perlu *load module* secara manual atau lakukan reboot ulang. Tujuan reboot juga untuk menguji apakah modul kernel yang baru dapat dikenali saat booting.

Tjokorda Bagus Putra M. [tjok_bpm@yahoo.co.id]

Mengedit File Binary

Di sistem *proprietary*, di mana *source code* suatu program merupakan rahasia besar *developer* dan file binary adalah yang digunakan oleh user, tidak jarang user mengedit file binary tersebut untuk melakukan berbagai perubahan sederhana, seperti mengganti tulisan *start* pada *taskbar* Windows dengan tulisan mulai. Bukan tindakan yang legal, namun cukup menarik.

Di Linux, di mana sebagian besar program dilisensikan di bawah lisensi *free software/open source*, mengubah tulisan tertentu dari *interface* program bukanlah masalah besar. Tentu saja, kita tinggal mengambil source code program tersebut, melakukan modifikasi bagian yang diinginkan, melakukan kompilasi, dan kita akan mendapatkan apa yang kita inginkan.

Walaupun, melakukan pengeditan file binary itu sendiri, terlepas dari ketersediaan source code, tetap merupakan hal yang menarik. Dan, terkadang, barangkali, kita ingin melakukan sedikit perubahan tulisan, namun tidak memiliki source code program tersebut atau tidak memiliki semua paket yang dibutuhkan untuk kompilasi, atau merasa bahwa kita hanya ingin mengubah sedikit saja bagian program dan malas melakukan kompilasi. Apapun alasan Anda.

Di tulisan ini, kita akan membahas cara mengubah tulisan tertentu pada user interface beberapa contoh program. Pengubahan file binary bisa saja menjadi sangat kompleks dengan hasil akhir yang sangat signifikan. Apa kita bahas di sini hanyalah merupakan contoh sederhana. Dan sebagai catatan, apa yang dilakukan di sini hanyalah merupakan contoh dan bukannya merupakan ajakan untuk mengedit berbagai binary (termasuk yang melanggar lisensi) untuk kepentingan yang ilegal.

Tulisan ini dan semua contoh yang ada dibuat di sistem Zenwalk Linux 4.2, namun seharusnya dapat diterapkan pada sistem lain tanpa masalah sama sekali.

Ghex, senjata ampuh kita

Pertama-tama, sebelum melakukan pengeditan, kita membutuhkan editor file binary yang nyaman digunakan. Sebagai contoh, penulis menggunakan Ghex versi 2.8.2.

Carilah Ghex di dalam instalasi atau media distribusi sistem Anda. Ghex dipaketkan di dalam cukup banyak distribusi Linux. Apabila tidak ditemukan, carilah paket untuk distribusi Anda di Internet atau komunitas lokal Anda. Apabila tidak ditemukan juga, *download*-lah source code Ghex di *http://pluton.ijs.si/~jaka/soft.html#GHEX* dan lakukanlah kompilasi sendiri.

Setelah Ghex terinstal, jalankanlah Ghex dengan perintah:

```
$ ghex2
```

Apabila Ghex dapat dijalankan, maka kita pun siap untuk memulai.

Contoh pertama: ls

Sebagai contoh pertama, kita akan mencoba mengubah teks tertentu pada program ls. Kopikanlah program ls yang ada di /bin ke direktori percobaan Anda, misal di /tmp/HEX/.

Pastikan program ls yang di-*copy*-kan tersebut bisa dijalankan:

```
$ ./ls --version
ls (GNU coreutils) 5.97
Copyright (C) 2006 Free Software
Foundation, Inc.
This is free software.  You may
redistribute copies of it under
the terms of
the GNU General Public License
<http://www.gnu.org/licenses/gpl.
html>.
There is NO WARRANTY, to the
extent permitted by law.

Written by Richard Stallman and
David MacKenzie.
```

Sebagai contoh pertama, kita akan mengganti tulisan versi 5.97 tersebut menjadi versi 9.99.

Lakukanlah langkah-langkah berikut ini:

- Jalankan Ghex.
- Bukalah program ls yang kita copy tersebut dengan mengakses menu *File* -> *Open*.
- Di bagian panel kanan, scrol-lah ke bagian agak bawah di mana kita bisa menemukan tulisan versi 5.97 yang ingin kita ganti. Di sistem penulis, tulisan ini ditemukan pada offset D22D. Cara yang lebih mudah apabila kita mengetahui offsetnya adalah dengan mengakses menu *Edit* -> *Goto Byte*... dan memasukkan nilai 0xD22D.

- Gantilah tulisan 5.97 dengan 9.99.
- Simpanlah dengan mengakses menu *File -> Save*.
- Keluarlah dari program dengan mengakses *File -> Exit*.

Cobalah kembali menjalankan program ls kita tersebut:

```
$ ./ls --version
ls (GNU coreutils) 9.99
Copyright (C) 2006 Free Software
Foundation, Inc.
This is free software.  You may
redistribute copies of it under
the terms of
the GNU General Public License
<http://www.gnu.org/licenses/gpl.
html>.
There is NO WARRANTY, to the
extent permitted by law.

Written by Richard Stallman and
David MacKenzie.
```

Kini, versi ls kita adalah 9.99.

Ukuran program ls yang sesungguhnya dan program ls yang telah dimodifikasi tetap sama. Namun sesungguhnya, md5sum kedua program tersebut berbeda karena memang ada informasi yang berbeda:

```
$ md5sum /bin/ls ls
c205a97140d3e0c85c0e76bbe7749076
/bin/ls
3c5633fd029bcaf8811b94e2945b4497  ls
```

MC: command menjadi perintah

Pada contoh kedua ini, kita akan mengubah tulisan Command pada bar bagian atas mc menjadi tulisan Perintah.

Kopikanlah program mc yang ada di /usr/bin ke direktori percobaan Anda, misal di /tmp/HEX/.

Pastikan program mc yang di-copy-kan tersebut bisa dijalankan:

```
$ ./mc --version
GNU Midnight Commander 4.6.1
Virtual File System: tarfs, extfs,
cpiofs, ftpfs, fish, undelfs
With builtin Editor
Using system-installed S-Lang
library with terminfo database
With subshell support as default
With support for background
operations
With mouse support on xterm and
Linux console
With internationalization support
With multiple codepages support
```

Lakukanlah langkah-langkah berikut ini:

- Jalankan Ghex.
- Bukalah program mc yang kita copy tersebut dengan mengakses menu *File -> Open*.
- Di bagian panel kanan, *scroll*-lah ke bagian agak bawah di mana kita bisa menemukan tulisan Command yang ingin kita ganti. Tulisan Command berada di sekitar &File . &Command . &Options .[Options Menu]. .Di sistem penulis, tulisan Command ini ditemukan pada offset 6E916.
- Gantilah tulisan Command menjadi Perintah. Harap diperhatikan bahwa Command hanya terdiri dari 7 karakter sementara Perintah terdiri dari 8 karakter. Karakter terakhir kita ketikkan begitu saja sebagai kelanjutan dari karakter sebelumnya.
- Simpanlah dengan mengakses menu *File -> Save*.
- Keluarlah dari program dengan mengakses *File -> Exit*.

Cobalah kembali menjalankan program mc kita tersebut.

Mengedit dan menembus batas

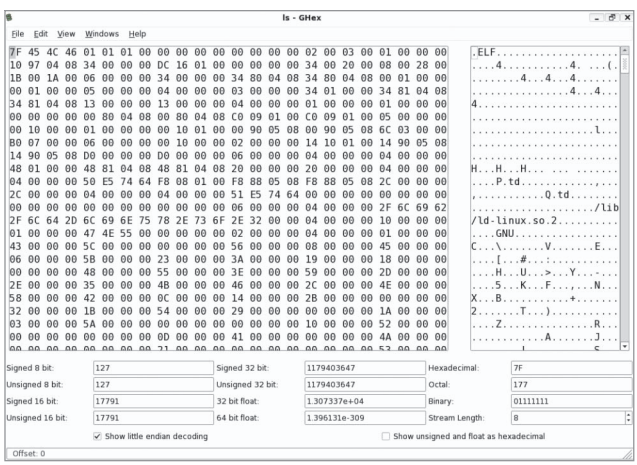
Pada contoh sebelumnya (mc), kita melihat bahwa kita mengganti tulisan yang terdiri dari 7 karakter dengan tulisan yang terdiri dari 8 karakter. Ketika kita mengedit file binary, maka alangkah baiknya kita tidak melakukan tindakan yang berbahaya seperti ini, kecuali kita tahu persis apa yang sedang kita lakukan, karena berbeda program mungkin saja berbeda tingkah laku ketika tindakan seperti ini dilakukan.

Khusus untuk program mc tadi, setelah penulis mencoba-coba, kita sebenarnya masih bisa mengganti sampai 10 karakter (aslinya: 7 karakter) dengan resiko minimal. Sebagai contoh, kita bisa mengganti tulisan Command menjadi Perintahku. Namun, dalam melakukan penggantian, offset penggantian tidaklah sama dengan offset tulisan Command. Kita mulai mengganti dari 2 offset sebelumnya.

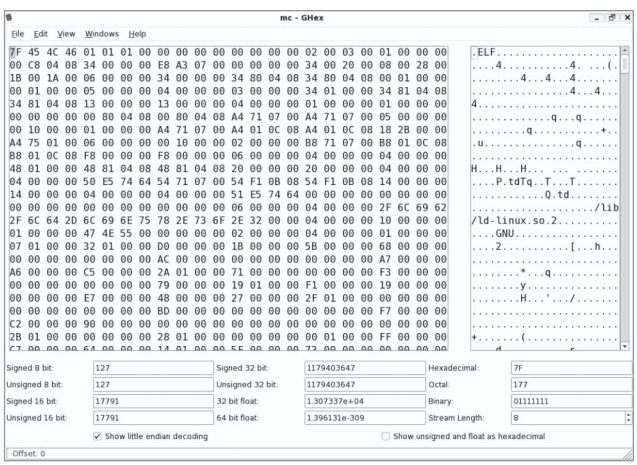
Apakah yang terjadi kalau kita mengganti sampai lebih dari 10 karakter? Di mc penulis, menu akan tampil kacau, dimana tulisan yang kita buat akan digabung dengan Options, walaupun Options akan ditampilkan lagi di sebelah kanannya. Cobalah bereksperimen.

Susahnya menemukan!

Dalam pekerjaan sederhana mengedit file binary seperti mengganti teks, tentunya kita perlu mengetahui di *offset* mana teks



Ghex membuka program ls.



Ghex membuka program mc.

yang ingin kita ganti. Kalau teks tersebut cukup unik, seperti versi program, maka hal ini tidaklah menjadi masalah. Namun, kalau kita ingin mengganti teks yang cukup umum ditemukan di *user interface program*, maka kita perlu sedikit usaha untuk mencari.

Membuat pengeditan lebih susah

Bagi *developer* yang ingin menjadikan binary programnya lebih susah diedit, maka alat bantu *packer* seperti UPX cukup dapat diandalkan. Sedianya, UPX akan mengompres file executable menjadi lebih kecil. Tapi, sebagai konsekuensi dari proses kompresi, string yang terdapat di dalam suatu executable menjadi jauh lebih susah untuk ditemukan, dan karenanya menjadi lebih susah untuk diedit.

UPX bisa di-*download* di *http://upx.sourceforge.net* dan telah disertakan ke dalam berbagai distribusi popular. Carilah ke dalam media distribusi Anda sebelum men-download dan melakukan kompilasi sendiri.

Sebagai contoh, penulis akan melakukan kompresi pada file executable mc, yang sudah dimodifikasi sebelumnya.

```
$ upx mc
                       Ultimate
Packer for eXecutables
    Copyright (C) 1996,1997,1998,
1999,2000,2001,2002,2003,2004,200
5,2006
UPX 2.02        Markus Oberhumer,
Laszlo Molnar & John Reiser   Aug
13th 2006

        File size         Ratio
Format      Name
   --------------------   ------
-----------   -----------
    501792 ->    217796   43.40%
linux/elf386   mc


Packed 1 file.
```

Ketika file tersebut dibuka kembali dengan ghex, kita tidak bisa menemukan lagi hasil modifikasi kita ataupun teks panjang lainnya, yang sebelum dikompres bisa kita temukan. Sudah kecil, susah diedit pula. Seru bukan?

Hexdump: tool berguna lainnya

Tool editor seperti ghex memang berguna untuk mengedit file binary. Namun, kalau kita hanya ingin melihat dump heksadesimal dan ASCII dari file binary kita, program hexdump dapat sangat diandalkan. Program hexdump merupakan anggota paket util-linux dan harusnya dapat ditemukan pada hampir semua distribusi Linux.

Berikut ini contoh dump heksadesimal file executable mc kita:

```
$ hexdump mc|head -n 10
0000000 457f 464c 0101 0001 694c
756e 0078 0000
0000010 0002 0003 0001 0000 5b2c
00c3 0034 0000
0000020 0000 0000 0000 0000 0034
0020 0002 0000
0000030 0000 0000 0001 0000 0000
0000 1000 00c0
0000040 1000 00c0 52a0 0003 52a0
0003 0005 0000
0000050 1000 0000 0001 0000 0748
0000 7748 080d
0000060 7748 080d 0000 0000 0000
0000 0000 0000
0000070 1000 0000 7f28 63d2 5055
2158 0772 0c0d
0000080 3f77 7b97 a820 0007 a820
0007 0134 0000
0000090 00a7 0000 0002 0000 3f63
f964 457f 464c
```

Dan, berikut ini adalah bentuk canonical heksadesimal dan ASCII:

```
$ hexdump -C mc|head -n 10
00000000  7f 45 4c 46 01 01 01
00  00 00 00 00 00 00 00
|.ELF............|
00000010  02 00 03 00 01 00 00
00  00 c8 04 08 34 00 00 00
|.........È..4...|
00000020  e8 a3 07 00 00 00 00
00  34 00 20 00 08 00 28 00
|è£......4. ...(.|
00000030  1b 00 1a 00 06 00 00
00  34 00 00 00 34 80 04 08
|........4...4...|
00000040  34 80 04 08 00 01 00
00  00 01 00 00 05 00 00 00
|4...............|
00000050  04 00 00 00 03 00 00
00  34 01 00 00 34 81 04 08
|........4...4...|
00000060  34 81 04 08 13 00 00
00  13 00 00 00 04 00 00 00
|4...............|
00000070  01 00 00 00 01 00 00
00  00 00 00 00 00 80 04 08
|................|
00000080  00 80 04 08 a4 71 07
00  a4 71 07 00 05 00 00 00
|....¤q..¤q......|
00000090  00 10 00 00 01 00 00
00  a4 71 07 00 a4 01 0c 08
|........¤q..¤...|
```

Kita bisa pula mencari teks dengan bantuan program grep. Sebagai contoh, kita akan mencari tulisan midnight:

```
$ hexdump -C mc| grep -i
'midnight'
0006c040  20 74 68 65 20 4d 69
64  6e 69 67 68 74 20 43 6f  | the
Midnight Co|
0006d360  73 74 00 4d 69 64 6e
69  67 68 74 20 43 6f 6d 6d  |st.
Midnight Comm|
0006e8d0  31 30 00 20 54 68 65 20
4d 69 64 6e 69 67 68 74  |10. The
Midnight|
0006e970  55 20 4d 69 64 6e 69
67  68 74 20 43 6f 6d 6d 61  |U
Midnight Comma|
0006ec10  74 68 65 20 4d 69 64
6e  69 67 68 74 20 43 6f 6d  |the
Midnight Com|
0006f1a0  20 54 68 65 20 4d 69
64  6e 69 67 68 74 20 43 6f  | The
Midnight Co|
00071030  2e 0a 00 00 47 4e 55 20
4d 69 64 6e 69 67 68 74  |....GNU
Midnight|
000715c0  69 72 73 00 4d 69 64
6e  69 67 68 74 20 43 6f 6d  |irs.
Midnight Com|
000718e0  4d 69 64 6e 69 67 68
74  2d 43 6f 6d 6d 61 6e 64
|Midnight-Command|
00075270  4d 69 64 6e 69 67 68 74
20 43 6f 6d 6d 61 6e 64  |Midnight
Command|
```

Sediakan selalu back-up

Apa yang kita lakukan di sini merupakan tindakan yang cukup berbahaya. Apabila kita salah mengubah, maka program bisa tidak dapat bekerja. Sediakanlah selalu *back-up* sebelum melakukan modifikasi.

Sampai di sini dulu pembahasan kita. Pengeditan file binary bisa sangat berguna selama kita melakukannya dengan benar. Selamat mencoba!

Noprianto [noprianto@infolinux.co.id

IKLAN

Membangun Paket Slackware Sederhana dengan Cepat dan Mudah

Slackware merupakan salah satu distribusi Linux tertua yang masih di-*maintain* dengan baik. Slackware juga merupakan distribusi Linux yang stabil dan dirancang dengan sangat hati-hati. Bagi Anda yang senang menggunakan Slackware ataupun ingin memaketkan aplikasi Anda untuk pengguna Slackware, kita akan membahas cara cepat dan mudah membangun paket.

Untuk membangun paket Slackware, kita bisa menggunakan berbagai cara yang kita sukai. Hal ini disebabkan karena paket Slackware sendiri, dibandingkan paket RPM ataupun DEB, jauh lebih sederhana dan mudah dibangun. Paket Slackware sendiri merupakan file tar.gz biasa (dengan ekstensi .tgz) yang dilengkapi dengan informasi dan *script* opsional.

Kita bisa membangun paket Slackware dengan cara manual memanfaatkan *tool-tool* standar, tar dan gzip. Kita juga bisa menggunakan berbagai tool yang datang dengan distribusi Slackware dan memang dikhususkan untuk pembuatan paket. Apabila diperlukan, kita juga bisa memanfaatkan/mencontoh berbagai *slackbuild script* yang tersedia. Dan, kalaupun ingin lebih canggih, kita bisa menggunakan bantuan tool pihak ketiga seperti:

- checkinstall.
- installwatch.
- protopkg.

Untuk mendapatkan paket Slackware, kita pun bisa mengonversi paket RPM menjadi paket Slackware dengan bantuan tool:

- rpm2tgz.
- alien.

Di dalam tulisan ini, kita tidak akan menggunakan konversi paket ataupun tool-tool pihak ketiga seperti disebutkan. Kita juga tentu saja tidak akan membangunnya secara manual (tar, gzip manual). Untuk kesempatan kali ini, kita akan membangun menggunakan tool standar makepkg.

Semua contoh tulisan ini dibangun di atas sistem Zenwalk Linux 4.2, namun seharusnya dapat diterapkan pada sistem Slackware dan berbagai turunannya, tanpa atau dengan sedikit penyesuaian.

Sebagai catatan, cara yang digunakan di dalam pembahasan ini merupakan cara yang sederhana dan terdapat kemungkinan tidak berlaku untuk semua program. Untuk metode yang lebih baik, Anda bisa mempergunakan bantuan program checkinstall (akan dibahas di tulisan lain).

Langkah 1: Siapkan program

Sebagai langkah pertama, siapkanlah program yang ingin dipaketkan. Penulis akan mengasumsikan Anda melakukan kompilasi sendiri dari *source code*. Contoh paket yang akan penulis bangun adalah moc, sebuah audio player *powerful* yang berjalan di *console*.

Lakukanlah kompilasi seperti biasa, tapi dengan sebelumnya mengarahkan prefix ke suatu direktori khusus. Sebagai contoh, siapkan sebuah direktori khusus /tmp/program.

Semua aplikasi dikompilasi dengan prefix /tmp/program/<NAMA_APP>. Apabila kita membangun aplikasi moc, maka prefix untuk moc akan diset ke /tmp/program/moc. Cara kompilasi tidak dibahas di dalam tulisan ini. Rujuklah ke dokumentasi kompilasi source code program.

Setelah dikompilasi dan diinstal, semua file-file yang dibutuhkan akan berada di dalam /tmp/program/moc. Contoh isi direktori moc penulis:

```
$ find
.
./bin
./bin/mocp
./lib
./lib/moc
./lib/moc/decoder_plugins
./lib/moc/decoder_plugins/
libmusepack_decoder.la
./lib/moc/decoder_plugins/
libmusepack_decoder.so
./lib/moc/decoder_plugins/libvorbis_
decoder.la
./lib/moc/decoder_plugins/libvorbis_
decoder.so
./lib/moc/decoder_plugins/libflac_
decoder.la
./lib/moc/decoder_plugins/libflac_
decoder.so
./lib/moc/decoder_plugins/libmp3_
decoder.la
./lib/moc/decoder_plugins/libmp3_
decoder.so
./share
./share/doc
./share/doc/moc
./share/doc/moc/keymap.example
./share/doc/moc/README
./share/doc/moc/THANKS
```

```
./share/doc/moc/config.example
./share/man
./share/man/man1
./share/man/man1/mocp.1
./share/moc
./share/moc/themes
./share/moc/themes/example_theme
./share/moc/themes/yellow_red_theme
./share/moc/themes/black_theme
./share/moc/themes/red_theme
./share/moc/themes/green_theme
./share/moc/themes/moca_theme
./share/moc/themes/nightly_theme
./share/moc/themes/transparent-
background
./share/moc/themes/darkdot_theme
```

Langkah 2: Buat struktur direktori isi paket

Anda telah memiliki sebuah tree direktori aplikasi di /tmp/program/<NAMA_APP>, sebagai contoh: /tmp/program/moc.

Di tahap ini, kita akan membangun struktur direktori baru untuk persiapan pembuatan paket. Buatlah direktori khusus /tmp/package/<NAMA_APP>. Contoh, penulis membuat direktori /tmp/package/moc.

Di dalam /tmp/package/moc, penulis selanjutnya mempersiapkan *prefix* sesungguhnya untuk sistem target. Apabila moc ingin diinstal di /usr, maka kita akan membuat direktori usr di dalam /tmp/package/moc.

Selanjutnya, *copy*-kanlah semua file program ke dalam /tmp/package/<NAMA_APP>/<PREFIX_TARGET>. Sebagai contoh, kopikanlah semua isi /tmp/program/moc/ ke /tmp/package/moc/usr/:

```
$ cd /tmp/package/moc/usr/
$ cp -a /tmp/program/moc/* .
```

Langkah 3: Buat deskripsi paket

Masih di /tmp/package/<NAMA_APP>, buatlah direktori install.

```
$ mkdir install
```

Kemudian, buatlah file install/slack-desc dengan pattern isi sebagai berikut:

```
<NAMA_APP>: <DESKRIPSI SINGKAT>
<NAMA_APP>: <BARIS KOSONG>
<NAMA_APP>: <DESKRIPSI LEBIH LANJUT>
<NAMA_APP>: <DESKRIPSI LEBIH LANJUT>
<NAMA_APP>: <DESKRIPSI LEBIH LANJUT>
<NAMA_APP>: <DESKRIPSI LEBIH LANJUT>
<NAMA_APP>: <DESKRIPSI LEBIH LANJUT>
<NAMA_APP>: <DESKRIPSI LEBIH LANJUT>
<NAMA_APP>: <BARIS KOSONG>
```

Contoh isi install/slack-desc untuk paket moc:

```
moc: moc version 2.4.1, console
audio player for LINUX/UNIX
moc:
moc: moc features:
moc: - simple mixer
moc: - color themes
moc: - searching support
moc: - configurable title creation
from tags
moc: - optional character set
conversion for file tags
moc: - OSS, ALSA and JACK output
moc: - user defined key
moc:
```

Langkah 4: Bangun paket

Kita sampai di langkah terakhir. Aktiflah di direktori /tmp/package/<NAMA_APP>, sebagai contoh /tmp/package/moc.

Login-lah sebagai root dan berikanlah pattern perintah berikut ini:

```
# makepkg <NAMA_APP>-<VERSI>-<ARCH>-
<REV>.tgz
```

Sebagai contoh:

```
# makepkg moc-2.4.1-i586-nop1.tgz

Slackware package maker, version
2.1.

Searching for symbolic links:

No symbolic links were found, so we
won't make an installation script.
You can make your own later in ./
install/doinst.sh and rebuild the
package if you like.

This next step is optional - you
can set the directories in your
package
to some sane permissions. If any of
the directories in your package have
special permissions, then DO NOT
reset them here!

Would you like to reset all
directory permissions to 755 (drwxr-
xr-x) and
directory ownerships to root.root
([y]es, [n]o)?
```

Sampai di pertanyaan ini, jawablah yes untuk me-*reset* semua hak akses direktori ke 755, dengan kepemilikan pada user root.root. Ini merupakan kondisi standar. Apabila Anda memang mengatur hak akses dan kepemilikan secara khusus, jangan jawab yes di sini.

```
Creating tar file moc-2.4.1-i586-
nop1.tar...

./
usr/
usr/bin/
usr/bin/mocp
usr/lib/
usr/lib/moc/
...
...
...
install/slack-desc
tar-1.13: moc-2.4.1-i586-nop1.tar
is the archive; not dumped

Gzipping moc-2.4.1-i586-nop1.tar...

Renaming moc-2.4.1-i586-nop1.tar.gz
to moc-2.4.1-i586-nop1.tgz...

Package creation complete.
```

Selesai sudah. Kini, kita memiliki file paket yang diinginkan di direktori aktif, sesuai nama yang kita berikan. Kita bisa menginstal paket tersebut dengan perintah installpkg.

Seperti telah disebutkan sebelumnya, cara seperti ini tidak selalu berlaku untuk semua program. Terutama apabila program hasil kompilasi memaksa untuk membuka file tertentu sesuai prefix instalasi. Atau, program/pustaka yang datang dengan X-config, di mana X merupakan nama program/pustaka, yang ketika dijalankan akan memberikan informasi termasuk prefix. Namun, cara seperti ini sangat membantu apabila kita ingin membangun paket Slackware dengan cepat dan mudah.

Di kesempatan lainnya, kita akan membahas cara membangun paket Slackware yang lebih canggih dan benar, dengan bantuan checkinstall. Selamat mencoba! Sampai jumpa di kesempatan berikutnya.

Noprianto [noprianto@infolinux.co.id

Menginstal USB Wireless LAN Adapter Chipset zd1211

Dalam kesempatan yang lalu, penulis pernah membahas penggunaan ndiswrapper untuk instalasi USB wireless LAN adapter chipset zd1211, menggunakan driver Windows. Sebenarnya, *chipset* ini telah didukung oleh Linux. Tulisan ini akan membahas instalasi dua driver yang tersedia di pasaran agar dapat berjalan di Linux.

Kalau dipikir-pikir, penulis cukup beruntung memiliki adapter wlan 802.11/g tanpa merk jelas ini. Kebetulan bisa didapat dengan harga relatif murah, bisa digunakan di Windows dan di Linux pula. Bahkan, di Linux, kita bisa mempergunakan tiga driver:

- Driver Windows (bawaan produsen) menggunakan ndiswrapper (telah dibahas sebelumnya), atau
- Driver bawaan produsen/Vendor Driver (terdapat di CDROM driver ataupun versi update), atau
- Driver pengembangan lebih lanjut oleh komunitas/Vendor Based Driver.

Semua dari driver tersebut bisa bekerja dengan baik. Instalasi Driver Windows menggunakan ndiswrapper telah dibahas sebelumnya. Kini, kita akan membahas instalasi dua driver lainnya.

Semua pembahasan di dalam tulisan ini dibangun di atas sistem Debian GNU Linux 3.1 kernel *default* (dan juga bekerja dengan baik di Zenwalk Linux 4.2), dengan kernel source terinstal dan terkonfigur dan paket lain untuk kompilasi juga telah terinstal. Pengguna distro lain harusnya dapat mengikuti tanpa masalah berarti.

Driver dari komunitas/vendorbased

Driver untuk komunitas bisa di-*download* di *http://zd1211.ath.cx*. Lakukanlah langkah-langkah berikut setelah mendownload driver.

- Lihatlah ID device. Tancapkanlah *device* dan berikan perintah berikut:

```
$ lsusb
Bus 004 Device 002: ID 1582:6003
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
```

Catatlah ID device (dalam hal ini: 1582:6003) dan cocokkan ke *http://zd1211.ath.cx*. Dalam melakukan pencocokan, tidak masalah merk di website tersebut berbeda dengan yang Anda gunakan. Yang penting, patokannya adalah ID device. Pastikan Anda mengetahui apakah Anda harus mengisikan 1 atau 0 untuk variabel ZD1211REV_B (di Makefile; dibahas nanti). Device yang penulis gunakan, sesuai URL tersebut, menggunakan zd1211b sehingga penulis harus mengisikan 1 pada saat kompilasi (dibahas nanti).

- Ekstra arsip driver dan masuk ke direktori hasil ekstrak:

```
$ tar zxvf zd1211-driver-r83.tgz

$ cd zd1211-driver-r83
```

- Pengeditan variabel ZD1211REV_B di Makefile. Bukalah Makefile dengan editor favorit Anda. Isikanlah dengan angka 0 ataupun 1 sesuai dengan informasi yang telah Anda miliki.
- Lakukan kompilasi dengan memberikan perintah (*warning* tidak ditampilkan pada output perintah berikut):

```
$ make
/lib/modules/2.6.8-2-386/build
/root/ZD1211BU/zd1211-driver-r83
-I/root/ZD1211BU/zd1211-
driver-r83/src/include -
fomit-frame-pointer -O2 -Wall
-Wstrict-prototypes -pipe
-DZDCONF_WE_STAT_SUPPORT=1 -
DHOST_IF_USB -DAMAC -DGCCK -DOFDM
-DHOSTAPD_SUPPORT -DUSE_EP4_SET_
REG -DDOWNLOADFIRMWARE -DfTX_
GAIN_OFDM=0 -DfNEW_CODE_MAP=1
-DfWRITE_WORD_REG=1 -DfREAD_MUL_
REG=1 -DENHANCE_RX=1 -DZD1211B
...
...
  CC [M]  /root/ZD1211BU/zd1211-
driver-r83/src/zd1211.o
  LD [M]  /root/ZD1211BU/zd1211-
driver-r83/zd1211b.o
  Building modules, stage 2.
  MODPOST
  CC      /root/ZD1211BU/zd1211-
driver-r83/zd1211b.mod.o
  LD [M]  /root/ZD1211BU/zd1211-
driver-r83/zd1211b.ko
make[1]: Leaving directory `/usr/
src/kernel-headers-2.6.8-2-386'
```

Setelah kompilasi selesai, sebuah module dengan nama zd1211b.ko (atau zd1211.ko) akan terbentuk.

- Lakukanlah instalasi dengan memberikan perintah:

```
# make install
```

```
/lib/modules/2.6.8-2-386/build
/root/ZD1211BU/zd1211-driver-r83
-I/root/ZD1211BU/zd1211-
driver-r83/src/include -
fomit-frame-pointer -O2 -Wall
-Wstrict-prototypes -pipe
-DZDCONF_WE_STAT_SUPPORT=1 -
DHOST_IF_USB -DAMAC -DGCCK -DOFDM
-DHOSTAPD_SUPPORT -DUSE_EP4_SET_
REG -DDOWNLOADFIRMWARE -DfTX_
GAIN_OFDM=0 -DfNEW_CODE_MAP=1
-DfWRITE_WORD_REG=1 -DfREAD_MUL_
REG=1 -DENHANCE_RX=1 -DZD1211B
...
...
make[1]: Leaving directory `/usr/
src/kernel-headers-2.6.8-2-386'
mkdir -p /lib/modules/2.6.8-2-
386/net
cp zd1211b.ko /lib/modules/2.6.8-
2-386/net
depmod -a 2.6.8-2-386
gcc -o apdbg apdbg.c
chmod +x apdbg
cp ./apdbg /sbin/apdbg
```

- Menjalankan modul kernel
  Berikanlah perintah berikut untuk menjalankan modul kernel yang baru saja kita install. Apabila semuanya berjalan lancar, maka kita akan mendapatkan sebuah *interface* dengan nama wlan0 secara default.

```
# modprobe zd1211b
ZD1211B - http://zd1211.ath.cx/
- r83
Based on www.zydas.com.tw driver
version 2.5.0.0
usb 4-5: reset high speed USB
device using address 2
Release Ver = 4810
EEPORM Ver = 4810
PA type: 0
Airoha AL2230S_RF
AllowedChannel = 000107ff
Region: 48
usbcore: registered new driver
zd1211b registered new driver
zd1211b
```

  Gunakanlah tool favorit Anda untuk mengatur konektivitas.

Driver dari Vendor

Driver dari vendor disertakan di dalam CD driver pada direktori Linux. Namun, karena versinya terlalu kuno (2.2.0.0), penulis men-download versi yang lebih baru (2.16.0.0) dari *http://zd1211.ath.cx* atau *www.deine-taler.de/zd1211/*. Download-lah juga firmware (penulis menggunakan versi 1.3) dari *http://sourceforge.net/project/showfiles.php?group_id=129083*.

Lakukanlah langkah-langkah berikut untuk melakukan instalasi:

- Ekstrak arsip driver dan masuk ke direktori hasil ekstrak:

```
$ tar zxvf ZD1211LnxDrv_2_16_0_
0.tar.gz

$ cd ZD1211LnxDrv_2_16_0_0
```

- Lakukanlah kompilasi dengan memberikan perintah berikut. Sebagai catatan, perintah berikut juga akan langsung melakukan instalasi sehingga harus dijalankan oleh root.

```
# make (warning tidak
ditampilkan pada output perintah
berikut):

make both
make[1]: Entering directory
`/root/ZD1211BU/FROM_VENDOR/
ZD1211LnxDrv_2_16_0_0'
make clean
make[2]: Entering directory
`/root/ZD1211BU/FROM_VENDOR/
ZD1211LnxDrv_2_16_0_0'
rm -rf .tmp_versions .*.cmd
*.ko *.mod.c *.mod.o *.o src/*.
o  src/.*.o.cmd menudbg apdbg
winevl_iface
make[2]: Leaving directory
`/root/ZD1211BU/FROM_VENDOR/
ZD1211LnxDrv_2_16_0_0'
make ZD1211REV_B=0
make[2]: Entering directory
`/root/ZD1211BU/FROM_VENDOR/
ZD1211LnxDrv_2_16_0_0'
/lib/modules/2.6.8-2-386/build
...
...
make -C /lib/modules/2.6.8-
2-386/build SUBDIRS=/root/
ZD1211BU/FROM_VENDOR/
ZD1211LnxDrv_2_16_0_0 modules
make[3]: Entering directory
`/usr/src/kernel-headers-2.6.8-
2-386'
  Building modules, stage 2.
  MODPOST
make[3]: Leaving directory
`/usr/src/kernel-headers-2.6.8-
2-386'
mkdir -p /lib/modules/2.6.8-2-
386/net
cp zd1211b.ko /lib/
modules/2.6.8-2-386/net
depmod -a
make[2]: Leaving directory
`/root/ZD1211BU/FROM_VENDOR/
ZD1211LnxDrv_2_16_0_0'
make[1]: Leaving directory
`/root/ZD1211BU/FROM_VENDOR/
ZD1211LnxDrv_2_16_0_0'
```

- Ekstraklah firmware zd1211-firmware 1.3.tar.bz2 ke /lib/firmware/zd1211

```
# mkdir /lib/firmware

# tar jxvf zd1211-
firmware1.3.tar.bz2
zd1211-firmware/
zd1211-firmware/zd1211b_uphm
zd1211-firmware/zd1211_uphr
zd1211-firmware/README
zd1211-firmware/zd1211b_uph
zd1211-firmware/zd1211b_ub
zd1211-firmware/zd1211b_uphr
zd1211-firmware/zd1211_ub
zd1211-firmware/zd1211_ur
zd1211-firmware/zd1211_uph
zd1211-firmware/zd1211b_ur
zd1211-firmware/zd1211_uphm

# mv zd1211-firmware zd1211

# mv zd1211 /lib/firmware/
```

- Menjalankan modul kernel
  Berikanlah perintah berikut untuk menjalankan modul kernel yang baru saja kita instal. Apabila semuanya berjalan lancar, maka kita akan mendapatkan sebuah interface dengan nama eth1 (atau eth0) secara default.

```
# modprobe zd1211b
```

  Gunakanlah tool favorit Anda untuk mengatur konektivitas.

USB wireless lan adapter kita pun bisa digunakan. Sampai di sini dulu pembahasan kita. Selamat mencoba!

Noprianto [noprianto@infolinux.co.id

Mengamati Sistem dengan dmesg

Untuk mengamati *log* sistem, dmesg bisa sangat membantu. Tulisan ini akan membahas berbagai aspek pengamatan sistem dengan dmesg. Dengan memahami apa yang terjadi, kita bisa mengambil langkah-langkah yang tepat untuk menyelesaikan berbagai masalah.

Pada dasarnya, sistem Linux merupakan sistem yang sangat nyaman untuk digunakan, dalam artian user bisa memahami banyak hal yang terjadi, dengan bantuan log sistem. Apa yang sedang terjadi tidak akan disembunyikan. Begitu sistem mengenali dan menggunakan filesistem swap, kita juga tahu. Begitu sistem mulai mengenali harddisk beserta partisinya, kita pun juga tahu. Bahkan, ketika *hardware* baru ditancapkan, kita bisa mengamati apa yang sedang terjadi. Banyak masalah terkadang bisa diselesaikan selama kita teliti mengamati log, memahami apa yang sedang terjadi dan pada akhirnya, bisa mengambil tindakan yang tepat. Jadi, kurang apa lagi?

Untuk mengamati log, kita bisa mengamati secara langsung, ataupun dengan menggunakan berbagai tool yang ada. Salah satu tool yang ada dan sangat berguna adalah dmesg. Kita akan membahas beberapa aspek pengamatan sistem dengan tool ini. Semua contoh di tulisan ini dibuat di atas sistem Zenwalk Linux 4.2, namun seharusnya bisa diterapkan pada sistem lainnya tanpa masalah sama sekali.

Dmesg

Program dmesg merupakan anggota dari paket util-linux dan seharusnya bisa ditemukan dalam hampir semua distribusi Linux standar. Untuk mengamati log sistem, dmesg bisa dijalankan oleh user biasa.

Berikut ini contoh 10 baris terakhir *output* dmesg di sistem yang penulis gunakan.

```
$ dmesg |tail -n10
ReiserFS: hda3: found reiserfs
format "3.6" with standard journal
ReiserFS: hda3: using ordered data
mode
ReiserFS: hda3: journal params:
device hda3, size 8192, journal
first block 18, max trans len
1024, max batch 900, max commit
age 30, max trans age 30
ReiserFS: hda3: checking
transaction log (hda3)
ReiserFS: hda3: Using r5 hash to
sort names
eth0: link down
[drm] Initialized drm 1.0.1
20051102
ACPI: PCI Interrupt
0000:00:02.0[A] -> Link [LNKA] ->
GSI 11 (level, low) -> IRQ 11
[drm] Initialized i915 1.5.0
20060119 on minor 0
[drm] Initialized i915 1.5.0
20060119 on minor 1
```

Setiap ada tindakan yang berhubungan dengan sistem, log sistem akan bertambah sehingga baris-baris terakhir akan senantiasa berubah.

Apa yang kita lihat pada output tersebut setidaknya telah dituliskan dalam bahasa yang relatif bisa dipahami. Paling tidak, cukup informatif. Sebagai contoh, pada baris:

```
eth0: link down
```

Kita bisa melihat informasi bahwa *link* pada eth0 berada dalam status *down*.

Pada pembahasan-pembahasan berikut, kita akan melihat beberapa contoh pengamatan sistem. Siapkan selalu terminal emulator yang menyenangkan apabila Anda berada di X.

Informasi-informasi seputar CPU

Di bagian ini, kita akan melihat beberapa informasi seputar CPU yang ditemukan. Sebagai catatan, informasi mengenai CPU itu sendiri bisa dibaca di /proc/cpuinfo.

Untuk mudahnya, kita akan menggunakan bantuan grep untuk menyaring secara *case-insensitive* hal-hal seputar CPU. Berikut ini adalah contohnya:

```
$ dmesg |  grep -i  CPU
Initializing CPU#0
CPU: After generic identify,
caps: bfebf9ff 00000000 00000000
00000000 00004400 00000000
00000000
CPU: After vendor identify,
caps: bfebf9ff 00000000 00000000
00000000 00004400 00000000
00000000
CPU: Trace cache: 12K uops, L1 D
cache: 8K
CPU: L2 cache: 128K
CPU: Hyper-Threading is disabled
CPU: After all inits, caps:
bfebf9ff 00000000 00000000
00000080 00004400 00000000
00000000
```

```
CPU0: Intel(R) Celeron(R) CPU
2.50GHz stepping 09
Brought up 1 CPUs
ACPI: CPU0 (power states: C1[C1]
C2[C2])
```

Catatan penting

Harus kita perhatikan bahwa penyaringan menggunakan program grep seperti ini menjadikan kita tidak memiliki informasi jeda atau informasi lain yang terdapat pada log yang sesungguhnya. Misalnya, dari hasil penyaringan yang terdiri dari 10 baris tersebut, kita tidak bisa memastikan bahwa baris kedua mengikuti baris pertama persis di bawahnya, tanpa ada informasi lain diantara mereka. Adanya informasi lain antara baris satu dan baris lainnya bisa menjadi sangat penting.

Kalau kita menyaring begitu saja seperti contoh ini, maka bisa-bisa informasi penting antara baris tersaring terlewatkan. Dan, untuk *problem solving*, hal ini bisa mengganggu. Terkadang, suatu baris tidak menuliskan *keyword* yang kita saring, namun sebenarnya memiliki peran penting.

Sebagai solusinya, kita sebaiknya memberikan nomor baris pada output dmesg sebelum kita melakukan penyaringan. Paling tidak, agar kita tahu bahwa mungkin terdapat baris-baris sebelumnya atau setelahnya yang mungkin relevan ketika kita mencoba untuk menyelesaikan suatu permasalahan. Untuk memberikan nomor baris, kita bisa menggunakan program nl, yang menjadi bagian dari paket coreutils (terdapat di hampir semua distribusi Linux).

Contoh pemberian nomor baris:

```
$ dmesg | nl | head -n4
     1  Linux version 2.6.18.6
(root@hal) (gcc version 3.4.6) #1
SMP PREEMPT Wed Dec 20 16:47:04
CET 2006
     2  BIOS-provided physical RAM
map:
     3  BIOS-e820:
0000000000000000 -
000000000009f800 (usable)
     4  BIOS-e820:
000000000009f800 -
00000000000a0000 (reserved)
```

Contoh penyaringan dengan keyword CPU seperti dibahas sebelumnya:

```
$ dmesg | nl |  grep -i  CPU
    41  Initializing CPU#0
    50  CPU: After generic
identify, caps: bfebf9ff 00000000
00000000 00000000 00004400
00000000 00000000
    51  CPU: After vendor
identify, caps: bfebf9ff 00000000
00000000 00000000 00004400
00000000 00000000
    52  CPU: Trace cache: 12K
uops, L1 D cache: 8K
    53  CPU: L2 cache: 128K
    54  CPU: Hyper-Threading is
disabled
    55  CPU: After all inits,
caps: bfebf9ff 00000000 00000000
00000080 00004400 00000000
00000000
    62  CPU0: Intel(R) Celeron(R)
CPU 2.50GHz stepping 09
    65  Brought up 1 CPUs
   336  ACPI: CPU0 (power states:
C1[C1] C2[C2])
```

Dari informasi tersebut, setidaknya kita bisa memahami informasi seputar CPU itu sendiri (seperti kecepatan, ukuran cache level 2 dan status hyperthreading), beserta sistem telah mengenali 1 CPU dan mengaktifkannya (Brought up 1 CPUs). Kita bisa melihat bahwa subsistem ACPI juga telah bekerja dengan CPU pertama (ACPI: CPU0 (power states: C1[C1] C2[C2])).

Storage device

Dengan menggunakan bantuan program dmesg, kita juga bisa melihat informasi seputar storage device seperti harddisk, CDROM, removeable storage device seperti USB flash disk dan lain sebagainya.

Harddisk (IDE)

Untuk mengamati apa yang terjadi dengan harddisk IDE, kita bisa menyaring berdasarkan keyword hdX, di mana X adalah a, b, c dan d. Selain itu, kita juga bisa menyaring berdasarkan keyword ideX di mana X adalah 0, 1, 2 dan 3.

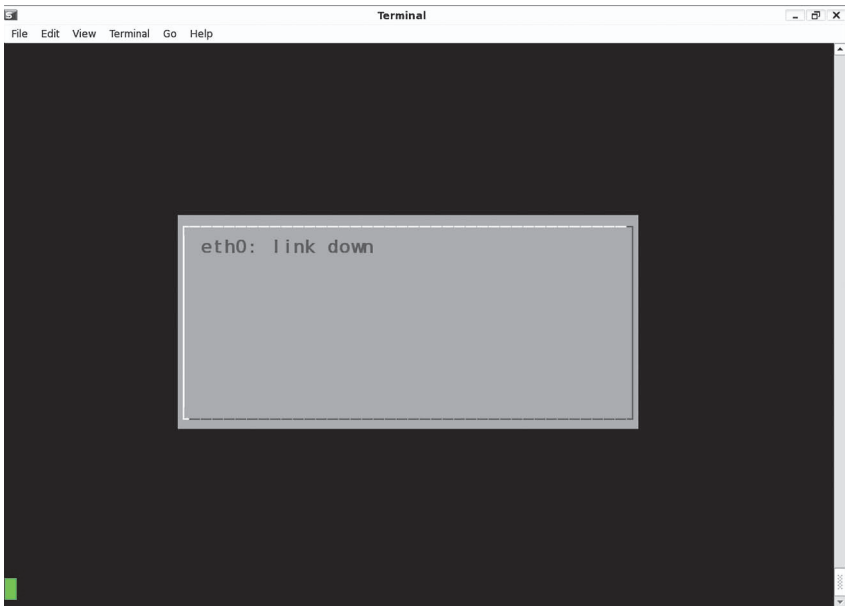
Contoh penyaringan dengan keyword ideX:

```
$ dmesg | nl |  grep -i  ide[0-3]
   170      ide0: BM-DMA at
0x1810-0x1817, BIOS settings: hda:
DMA, hdb:pio
   171      ide1: BM-DMA at
0x1818-0x181f, BIOS settings: hdc:
DMA, hdd:pio
   172  Probing IDE interface
ide0...
   174  ide0 at 0x1f0-0x1f7,0x3f6
on irq 14
   175  Probing IDE interface
ide1...
   177  ide1 at 0x170-0x177,0x376
on irq 15
```

Dari output tersebut, bisa kita lihat bahwa sistem mendeteksi adanya dua IDE, ide0 dan ide1, yang masing-masing berada pada IRQ 14 dan 15. Terdapat informasi lain yang lebih teknikal, yang akan sangat berguna bagi *developer* sistem.

Contoh penyaringan dengan keyword hda:

```
$ dmesg | nl |  grep -i  hda
    35  Kernel command line:
root=/dev/hda1 resume2=swap:/dev/
hda5 splash=silent vga=773
   170      ide0: BM-DMA at
0x1810-0x1817, BIOS settings: hda:
DMA, hdb:pio
   173  hda: SAMSUNG HM100JC, ATA
DISK drive
   178  hda: max request size:
512KiB
   179  hda: 195371568 sectors
(100030 MB) w/8192KiB Cache,
CHS=16383/255/63, UDMA(100)
   180  hda: cache flushes
supported
   181   hda: hda1 hda2 hda3 hda4
< hda5 hda6 hda7 >
   250  XFS mounting filesystem
hda1
   252  Ending clean XFS mount for
filesystem: hda1
   329  Adding 979924k swap on
/dev/hda5.  Priority:-1 extents:1
across:979924k
   341  ReiserFS: hda3: found
reiserfs format “3.6” with
standard journal
   342  ReiserFS: hda3: using
ordered data mode
   343  ReiserFS: hda3: journal
params: device hda3, size 8192,
journal first block 18, max trans
len 1024, max batch 900, max
commit age 30, max trans age 30
   344  ReiserFS: hda3: checking
```



Status link ethernet.

```
transaction log (hda3)
   345  ReiserFS: hda3: Using r5
hash to sort names
```

Beberapa informasi yang bisa didapatkan:

- Merk harddisk dan properti lainnya:

```
   173  hda: SAMSUNG HM100JC,
ATA DISK drive
   178 hda: max request size:
512KiB
   179 hda: 195371568 sectors
(100030 MB) w/8192KiB Cache,
CHS=16383/255/63, UDMA(100)
   180  hda: cache flushes
supported
```

- Tabel partisi:

```
   181   hda: hda1 hda2 hda3
hda4 < hda5 hda6 hda7 >
```

- Informasi mount berbagai partisi:

```
   250  XFS mounting filesystem
hda1
   252  Ending clean XFS mount
for filesystem: hda1
...
   341  ReiserFS: hda3: found
reiserfs format “3.6” with
standard journal
   342  ReiserFS: hda3: using
ordered data mode
..
```

- Penambahan swap:

```
   329  Adding 979924k swap on /
dev/hda5.  Priority:-1 extents:1
across:979924k
```

CD-ROM

Untuk mengamati informasi CDROM, kita bisa menyaring berdasarkan keyword cd-rom. Contoh:

```
$ dmesg | nl |  grep -i  cd-rom
   182  hdc: ATAPI 24X CD-ROM
drive, 128kB Cache, UDMA(33)
   183  Uniform CD-ROM driver
Revision: 3.20
```

Dari informasi tersebut, setidaknya kita bisa mengetahui cache cdrom (dalam hal ini 128kB), kecepatan baca (24X) dan merupakan device UDMA 33.

Removeable storage device (USB Flash Disk)

Tancapkanlah USB flash disk dan gunakan keyword sdX untuk penyaringan, di mana X adalah a,b,c, dan seterusnya.

Contoh penyaringan dengan keyword sda:

```
$ dmesg | nl | grep -i sda
   361  SCSI device sda: 3901952
512-byte hdwr sectors (1998 MB)
   362  sda: Write Protect is off
   363  sda: Mode Sense: 03 00 00
00
   364  sda: assuming drive cache:
write through
   365  SCSI device sda: 3901952
512-byte hdwr sectors (1998 MB)
   366  sda: Write Protect is off
   367  sda: Mode Sense: 03 00 00
00
   368  sda: assuming drive cache:
write through
   369   sda: sda1
   370  sd 0:0:0:0: Attached scsi
removable disk sda
```

Berdasarkan hasil penyaringan tersebut, kita bisa mendapatkan:

- Ukuran USB flash Disk dan properti lainnya.

```
   361  SCSI device sda: 3901952
512-byte hdwr sectors (1998 MB)
   362  sda: Write Protect is
off
   363  sda: Mode Sense: 03 00
00 00
   364  sda: assuming drive
cache: write through
```

Perangkat jaringan

Untuk menyaring informasi tentang perangkat jaringan, kita dapat menggunakan keyword nama *interface* seperti ethX atau wlanX (atau lainnya sesuai perangkat dan kernel; rujuklah ke manual device ataupun dokumentasi kernel), di mana X adalah 0, 1, 2, dan seterusnya.

Contoh keyword berupa eth0:

```
$ dmesg | nl | grep -i eth0
   324  eth0: RealTek RTL8139 at
0xef870000, 00:0a:e4:47:3e:11, IRQ
11
   325  eth0:  Identified 8139
chip type ‘RTL-8101’
   346  eth0: link down
```

Dari hasil penyaringan tersebut, bisa kita dapatkan:

- Merek ethernet card pertama dan informasi lain seperti MAC address:

```
   324  eth0: RealTek RTL8139 at
0xef870000, 00:0a:e4:47:3e:11,
IRQ 11
   325  eth0:  Identified 8139
chip type ‘RTL-8101’
```

- Status link (up/down):

```
   346  eth0: link down
```

Tips keyword penyaringan

Dalam melakukan penyaringan, kita akan membutuhkan keyword. Dan, kita yang harus menentukan keyword tersebut. Dengan demikian, kita perlu mengetahui beberapa

aturan yang digunakan oleh kernel. Sebagai contoh:

- Nama device harddisk IDE, yaitu hdX, di mana X adalah a,b,c,d, pada umumnya.
- Nama device harddisk SCSI, yaitu sdX, di mana X adalah a,b,c, dan seterusnya.
- Nama interface jaringan, seperti ethX, wlanX, di mana X adalah 0,1,2, dan seterusnya.
- Nama subsistem yang dikenal kernel seperti IDE, SCSI, USB, dan PCI.

Beberapa tip:

- Apabila kita ragu dengan nama device, misalnya kita tidak yakin apakah sdb atau sdc, maka awalnya, kita bisa memperluas pencarian dengan keyword sd dan bukannya sdb atau sdc saja. Bahkan, kita bisa mencari berdasarkan subsistemnya, misal USB.
- Apabila kita ingin memeriksa suatu perangkat dan lebih kurang kita mengetahui nama modul kernelnya (tidak harus spesifik device, bisa saja modul dasar), misal usb-storage, maka keyword tersebut bisa diberikan.
- Gunakan keyword case-insensitive.
- Penggunaan regular expression dengan bantuan program grep bisa pula dipergunakan.
- Untuk kondisi terakhir, kita bisa mempergunakan bantuan program tail.

Menyelesaikan berbagai masalah

Dengan mengamati log sistem, tak jarang kita bisa menyelesaikan berbagai masalah. Sebagai contoh:

- Apabila hardware yang Anda gunakan, yang telah dikenal oleh Linux (informasi ini Anda dapatkan dari milis/website terpercaya, sebagai contoh) tidak dapat bekerja (sementara Anda yakin modul kernel telah disertakan oleh distribusi/telah Anda kompilasi sendiri tanpa masalah), maka cobalah pelajari log sistem ketika hardware tersebut ditancapkan. Tak jarang wireless LAN adapter yang membutuhkan firmware tidak dapat bekerja karena firmwarenya tidak disertakan oleh distribusi, karena masalah legal. Umumnya, terdapat pesan (tidak semua hardware) tentang apa yang sedang dilakukan ketika hardware ditancapkan/diinisialisasi.
- Apabila Anda mengelola user di mana user Anda membutuhkan aplikasi semacam 'status box' yang menampilkan informasi perangkat tertentu, seperti ethernet 0 link up, ethernet 0 link down dan sebagainya, Anda bisa tetap memanfaatkan dmesg dan menyaring informasi yang berhubungan dengan perangkat yang Anda amati. Kita akan membangun contoh aplikasinya setelah ini.

Informasi status link ethernet

Kita akan membangun aplikasi kecil yang akan menampilkan status link ethernet: up (dan informasi tambahan lain) ataupun down.

Aplikasi akan kita bangun dengan shell script. User interface akan kita bangun dengan bantuan dialog:

Source code eth_link_stat.sh

```
#!/bin/sh

while [ 1 ]
do
        LINK_STAT="`dmesg | grep
-i 'eth[0-9]: link'| tail -n1`"
        dialog --infobox "$LINK_
STAT" 10 40
        sleep 2
done
```

Berikanlah hak akses executable pada script tersebut dengan perintah:

```
$ chmod +x eth_link_stat.sh
```

Untuk menjalankan aplikasi:

```
$ ./eth_link_stat.sh
```

Ketika aplikasi tersebut dijalankan, sebuah infobox berisi informasi status link ethernet akan ditampilkan.

Penjelasan source code:

- Aplikasi akan mengulang terus menerus
- Di dalam perulangan:
  - Kita akan mendapatkan status link eth, kemudian mengambil informasi terakhir.
  - Informasi kemudian ditampilkan.
  - Perulangan ditunda selama 2 detik.

Sebagai sistem yang terbuka, Linux memang sangat menarik. Selama kita rajin mempelajari dan mengutak-atik, maka Linux pun akan semakin ramah terhadap kita. Sampai di sini dulu pembahasan kita. Selamat mencoba dan mengembangkan!

Noprianto [noprianto@infolinux.co.id