

Dd, Utiliti Kecil yang Berguna

Dalam kegiatan menggunakan komputer sehari-hari, banyak tugas yang sebenarnya bisa dilakukan dengan bantuan utiliti dd, yang merupakan bagian dari GNU Coreutils. Program kecil ini memiliki banyak peranan, terutama ketika kita bekerja lebih mendalam dengan filesystem.

Ukurannya memang kecil. Hanya sekitar 40-an KB. Namun, tanpa dd, mungkin banyak kegiatan yang tidak bisa dilakukan. Banyak sekali *script* yang mengandalkan program ini. Perannya banyak sekali dalam mendukung hadirnya sistem operasi lengkap. Memanfaatkan dd, kita bahkan dapat menyelesaikan banyak tugas yang hanya bisa dilakukan dengan *software* terpisah di sistem operasi lain seperti Windows, misalnya.

Dd adalah program yang berguna untuk meng-copy sebuah file ke file lain, namun dengan fleksibilitas untuk mengubah ukuran blok dan mengizinkan adanya konversi ketika melakukan peng-copy-an. Semua ini kemudian ditambah lagi dengan betapa fleksibelnya Linux dalam banyak hal, yaitu hampir semua adalah file di Linux. Dengan kombinasi tersebut, dd dapat melakukan banyak sekali tugas yang berhubungan dengan filesystem.

Dd merupakan bagian dari GNU Coreutils, sebuah proyek GNU yang berisikan *tool-tool* kecil, namun yang sangat dibutuhkan oleh sebuah sistem operasi. Beberapa contoh tool yang datang bersama paket coreutils, antara lain:

- cat
- chmod
- chown
- cp
- date
- mkdir
- mv
- ls
- ln
- dan masih banyak lagi.

Paket GNU Coreutils juga ditulis oleh

hacker-hacker kelas dunia, seperti Richard Stallman, David MacKenzie, Peter Anvin, Ulrich Drepper, dan lain sebagainya.

Apa yang bisa dilakukan dd?

Pada dasarnya, dd bisa membantu kita apabila kita ingin melakukan tugas yang berhubungan dengan proses copy meng-copy file. Sekali lagi, pengertian file sangatlah luas di Linux. Meng-copy file di dd bisa berarti meng-copy-kan sebuah file dari suatu lokasi ke lokasi lain. Namun, bisa pula berarti membuat *image* CDROM. Ataupun yang lainnya. Berikut ini, kita akan melihat apa saja yang bisa dilakukan dengan dd:

- Meng-copy satu file ke file lain. Ini berarti seperti apa yang dilakukan oleh program cp. Tentu saja, apabila Anda ingin melakukan peng-copy-an file yang bisa dilakukan oleh program cp, sebaiknya program cp lah yang digunakan. Program cp memiliki fasilitas yang lebih apabila tugas yang dilakukan adalah mengopi file apa adanya.
- Meng-copy file dan melakukan konversi. Semisal Anda memiliki sebuah file teks dengan nama file a, yang berisikan teks-teks yang ditulis dengan huruf kecil. Anda ingin agar pada proses peng-copy-an ke file b, misalnya, semua huruf kecil di dalam file a tersebut langsung dikonversi ke huruf besar. Beberapa cara bisa dilakukan. Misal, Anda bisa menggunakan program cat untuk menampilkan isi file, kemudian disaring oleh program tr untuk melakukan konversi, dan hasilnya diarahkan ke file b. Atau, tentu saja, Anda bisa mempergunakan program dd yang kita bahas kali ini. Selama proses peng-copy-an, program dd bisa melaku-

kan beberapa aksi konversi yang cukup dibutuhkan.

- Meng-copy file dengan jumlah yang bisa ditentukan. Program dd memungkinkan kita untuk meng-copy sejumlah byte tertentu dalam pengopian. Ini akan sangat berguna apabila Anda hanya ingin menimpa beberapa blok dari suatu *device*. Seperti kita ketahui, Linux sangatlah fleksibel dalam menangani file. Hampir semua *device* yang *stream-oriented* akan direpresentasikan oleh file di dalam filesystem, yang disimpan di direktori / dev. Pada dasarnya, kita mengenal dua macam *stream-oriented device*: *char device* dan *block device*. Char device adalah *device* yang transfer datanya dilakukan per karakter. Sementara, block device adalah *device* yang transfer datanya dilakukan per blok. Dengan penggunaan dd, transfer data pun bisa dilakukan per karakter, walaupun untuk block device. Dan, tentu saja, kita tidak bisa mengartikan bahwa block device pun merupakan char device. Semua ini bisa kita lakukan karena fleksibilitas Linux. Untungnya, hal ini sepenuhnya transparan bagi user. Struktur data internal kernel yang akan menangani semua kerepotan ini.
- Meng-copy file dari blok tertentu. Dengan dikombinasikan dengan peng-copy-an sejumlah blok tertentu, kita bisa 'mengganti' blok-blok dalam range tertentu pada suatu *device*.
- Meng-copy isi disket ke disket lain. Beberapa dari Anda mungkin akan bertanya. Kenapa harus mempergunakan dd? Bukanlah kita bisa membuat sebuah folder sementara yang berisi semua file di dalam disket dan setelah itu meng-copy-

kan balik semua file di direktori sementara tersebut ke disket lain. Cara tersebut memang bisa digunakan. Tapi, waktu yang dibutuhkan akan sangat lama apabila disket source berisikan ratusan file kecil. Baik proses peng-copy-an dari source ataupun peng-copy-an ke disket tujuan akan memakan waktu yang sangat lama. Dengan dd, kita bisa membuat image dari suatu disket (yang akan menghasilkan sebuah file image), dan setelah itu, dengan menggunakan dd kembali, kita meng-copy-kan image tersebut ke disket lain. Cara ini digunakan pula oleh program WinImage di Windows.

- Membuat image CDROM. Anda mungkin sering mengalami hal seperti ini. Teman Anda baru saja membeli CD Distro terbaru. Dan, Anda ingin mengopinya. Namun, setelah CD-CDnya Anda terima, Anda mulai kebingungan karena tidak tahu bagaimana meng-copy CD-CD. Sistem Anda hanya terinstal Linux berbasis teks. Anda tidak memiliki Windows. Daripada repot-repot mencari program yang bisa digunakan, atau meminjam Windows teman lain, gunakan saja program dd. Dengan satu perintah yang sangat sederhana, yang memiliki konsep yang sama dengan meng-copy satu file ke file lain, kita dapat meng-copy isi CDROM secara komplit. Apabila CDROM bootable, maka hasil copy-nya pun akan bootable. Beda dengan ketika kita meng-copy-kan semua isi file ke direktori lain dan meng-copy-kan balik semua file tersebut ke CDROM baru. CDROM baru tidak akan bootable.
- Kloning harddisk. Sama seperti apa yang dilakukan oleh Norton Ghost dan kawan-kawannya. Proses kloning mungkin akan berjalan lebih lambat dan *user interface* dari dd mungkin kurang bersahabat, namun dd sepenuhnya bisa digunakan sebagai *cloning tool*. Prinsipnya sama persis dengan peng-copy-an file. Namun, file yang di-copy adalah /dev/hdb ke /dev/hdc misalnya. File /dev/hdb adalah harddisk IDE kedua dan file /dev/hdc adalah harddisk IDE ketiga. Tentu saja, dengan konsep yang sama, Anda juga bisa melakukan kloning partisi. Sayangnya, apabila Norton Ghost bisa melakukan kloning dari jaringan, dd tidaklah dapat

digunakan langsung. Namun, dengan modal yang diberikan oleh dd, kloning lewat jaringan bisa dilakukan.

- Men-zero-kan device (misal: disk). Ini adalah tindakan yang kejam. Tapi, dalam kondisi tertentu dibutuhkan. Di Linux, kita mengenal sebuah device /dev/zero, yang akan men-zero-kan sebuah device apabila digunakan sebagai source dalam peng-copy-an. Contoh yang umum adalah ketika Anda ingin membuat ramdisk dan ingin memastikan semuanya bersih. Program dd bisa digunakan untuk kebutuhan tersebut.
- Berbuat kejahatan. Bisa-bisa aja. Ingat bahwa kita memiliki /dev/zero di harddisk. Apa yang akan terjadi apabila kita menggunakan /dev/zero sebagai source dalam mengopi menggunakan dd, dan tujuan kita adalah /dev/hda misalnya? Ini berarti kita akan mengosongkan device. Tidak perlu berlama-lama. Cukup rusak saja MBR dengan mengirimkan 512 byte /dev/zero ke /dev/hda. Ini sudah cukup merepotkan. Dan, dd memungkinkan kita untuk berbuat sangat banyak kejahatan, yang parahnya, bisa dilakukan dengan sangat cepat. Contoh lain yang cukup menyebalkan adalah dengan memotong atau menyisipkan kode tertentu ke dalam aplikasi yang telah diketahui strukturnya.

Contoh sederhana

Berikut ini, kita akan melihat contoh sederhana penggunaan dd. Kita akan memulai dengan bagaimana meng-copy-kan satu file ke file lainnya.

```
$ dd if=a of=b
20+1 records in
20+1 records out
10250 bytes (10 kB) copied,
0.020628 seconds, 497 kB/s
```

Penjelasan:

- File sumber adalah file a, dan file tujuan adalah file b
- argumen if= menunjukkan input file
- argumen of= menunjukkan output file
- Kita harus memastikan jumlah record yang tertulis adalah sama dengan jumlah record yang terbaca.

Untuk membuktikan bahwa kedua file

tersebut berhasil di-copy dengan baik, gunakanlah perintah md5sum untuk melihat md5sum masing-masing:

```
$ md5sum a b
5a3a8f46abaf3f4d2d6ddb16b7fbf9
6 a
5a3a8f46abaf3f4d2d6ddb16b7fbf9
6 b
```

Pada dasarnya, dd cukup digunakan dengan pemberian argumen if dan of tersebut.

Contoh-contoh lainnya

Contoh sebelumnya bisa juga diterapkan untuk melakukan proses kloning harddisk. Hanya, bedanya, argumen if dan of harus disesuaikan dengan device harddisk. Sebagai contoh, apabila Anda ingin melakukan kloning dari harddisk kedua ke harddisk ketiga, maka berikut ini adalah perintahnya:

```
# dd if=/dev/hdb of=/dev/hdc
```

Apabila pembuatan image diinginkan, maka gantilah *argumen of* menjadi nama file yang diinginkan untuk image. Image tersebut selanjutnya bisa digunakan kembali sebagai input file apabila kloning ingin dilakukan dari image.

Pembuatan image juga dapat diterapkan juga untuk peng-copy-an dari floppy ke floppy. Tentu, kita perlu membuat image terlebih dahulu apabila hanya satu floppy drive yang tersedia. Pembuatan image floppy dapat dilakukan dengan perintah berikut:

```
$ dd if=/dev/fd0 of=bootdisk
2880+0 records in
2880+0 records out
1474560 bytes (1.5 MB) copied,
42.973 seconds, 34.3 kB/s
```

Untuk melihat tipe file bootdisk yang dihasilkan, gunakan perintah file berikut ini:

```
$ file bootdisk
bootdisk: x86 boot sector,
SYSLINUX bootloader (2.06), code
offset 0x3c, OEM-ID "SYSLINUX",
sectors/cluster 4, FAT 1, root
entries 112, sectors 20160
(volumes <=32 MB), sectors/FAT
20, serial number 0x94d438a0,
label: "BOOTDISK1 ", FAT (16
bit)
```

Dari output perintah file tersebut, bisa kita lihat bahwa file bootdisk yang dihasil-

kan merupakan bootdisk dengan bootload-er SYSLINUX. Label dan informasi lain juga ditampilkan dengan sangat detil. File bootdisk tersebut merupakan image dari floppy.

Dengan script sederhana, kita bisa membuat semacam program diskcopy di Linux. Langkah pertama, mintalah user untuk memasukkan floppy yang ditandai dengan penekanan enter. Setelah dd selesai, mintalah user untuk mengganti dengan floppy lain yang ditandai dengan penekanan enter pula. Setelah itu, biarkanlah proses peng-copy-an ke disket tujuan berlangsung.

Selain pada harddisk dan floppy, dd juga dapat digunakan untuk membuat image CDROM. Berikut ini adalah contoh perintahnya (diasumsikan CDROM device adalah /dev/hdc):

```
$ dd if=/dev/hdc of=cd01.iso
```

File image yang dihasilkan akan berupa ISO Image. Berikut ini adalah buktinya:

```
$ file cd01.iso
cd01.iso: ISO 9660 CD-ROM
filesystem data 'Debian 3.1 r0a
i386 Bin-1' (bootable)
```

Kembali ke dasar penggunaan dd. Seperti diketahui bahwa dd bisa melakukan konversi sementara melakukan peng-copy-an. Kita akan melihat bagaimana melakukan pengopian file teks sambil mengonversi semua karakter menjadi huruf kapital.

Isi file sumber, nama file adalah a:

```
$ cat a
halo
apa kabar
semuanya?

Baik
baik
saja
bukan?
```

Berikut ini adalah perintah untuk melakukan peng-copy-an dan konversi:

```
$ dd if=a of=b conv=ucase
0+1 records in
0+1 records out
50 bytes (50 B) copied, 0.000221
seconds, 226 kB/s
```

Dan, berikut ini adalah hasil file peng-copy-an, file b:

```
$ cat b
HALO
APA KABAR
SEMUYANYA?

BAIK
BAIK
SAJA
BUKAN?
```

Proses konversi yang dapat dilakukan tidak hanya mengubah ke huruf besar atau kecil, namun juga meliputi konversi dari ASCII ke EBCDIC dan sebaliknya, mengkonversi ASCII ke EBCDIC alternatif, mengganti *newline* dengan *space* dan sebaliknya, dan berbagai hal lainnya.

Contoh berikut ini akan mengganti *newline* dengan spasi. File source yang digunakan masih file a:

```
$ dd if=a of=c conv=block cbs=10
0+1 records in
0+1 records out
2 truncated records
64 bytes (64 B) copied, 0.000223
seconds, 287 kB/s
```

Berikut ini adalah file c yang dihasilkan:

```
$ cat c
halo    apa kabasemuanya
Baik    baik    saja    bukan?
```

Proses konversi tidak hanya bisa satu macam, namun juga bisa lebih selama tidak bentrok. Kita akan melengkapi penggantian *newline* dengan spasi, sekaligus mengonversi ke huruf kapital:

```
$ dd if=a of=c conv=block,ucase
cbs=10
0+1 records in
0+1 records out
2 truncated records
64 bytes (64 B) copied, 0.000219
seconds, 292 kB/s
```

```
$ cat c
HALO    APA KABASEMUANYA
BAIK    BAIK    SAJA    BUKAN?
```

Argumen *cbs* yang digunakan menunjukkan jumlah byte yang akan dikonversi pada suatu waktu. Contoh ini sebenarnya hanya menunjukkan kemampuan dasar dd dalam melakukan konversi.

Contoh berikutnya akan membahas bagaimana meng-copy sejumlah blok tertentu saja. Misal, kita memiliki file bootdisk dengan ukuran 1.5 MB. Kita akan meng-copy 1 blok pertama ke dalam file bootdisk2:

```
$ dd if=bootdisk of=bootdisk1
count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied,
0.000201 seconds, 2.5 MB/s
```

Dalam beberapa sistem, default ukuran 1 blok adalah 512 byte. Dengan demikian, pengopian satu blok akan meng-copy sebanyak 512 byte. Pertanyaannya adalah, bagaimana kalau kita ingin meng-copy hanya 256 byte? Apakah kita bisa memberikan *count=0.5*? Jawabannya adalah tidak.

Apabila kita ingin meng-copy sebanyak 256 byte sementara 1 blok berukuran 512 byte dan kita tidak bisa memberikan nilai 0.5 untuk argumen *count*, maka kita harus mengubah argumen *bs* (block size) seperti contoh berikut ini:

```
$ dd if=bootdisk of=bootdisk1
count=256 bs=1
256+0 records in
256+0 records out
256 bytes (256 B) copied,
0.001403 seconds, 182 kB/s
```

Dengan *bs* 1 byte dan *count* 256, maka kita akan meng-copy-kan 1x256 byte dari *if* ke *of*. Tentu saja, Anda bisa memberikan perkalian yang diinginkan, misal: 4x64 atau 2x128, atau 8x32 seperti contoh berikut:

```
$ dd if=bootdisk of=bootdisk1
count=32 bs=8
32+0 records in
32+0 records out
256 bytes (256 B) copied,
0.000399 seconds, 642 kB/s
```

Argumen lain yang juga sangat menarik selain *bs* dan *count* adalah *seek* dan *skip*. Keduanya dapat digunakan untuk meng-copy dari blok tertentu ataupun melompati beberapa blok. Tentu saja, kombinasi antara *bs*, *count* dan *seek* harus digunakan dengan sebaik-baiknya. Terutama *seek* atau *skip*. Dalam kondisi normal, keduanya tidak diperlukan.

Contoh terakhir yang akan kita bahas

dalam tulisan kali ini adalah pembuatan beberapa bagian dari ramdisk yang tersedia untuk digunakan seperti halnya disket ataupun block device lain.

Kita akan membuat ramdisk untuk digunakan, setelah di-mount pada /mnt/ram. Namun, Anda perlu login sebagai root untuk mengakses ramdisk.

Langkah pertama, kosongkan terlebih dahulu ramdisk yang ingin digunakan:

```
# dd if=/dev/zero of=/dev/ram0
bs=1MB count=64
64+0 records in
64+0 records out
64000000 bytes (64 MB) copied,
0.837714 seconds, 76.4 MB/s
```

Langkah kedua, kita kemudian dapat membuat filesystem pada /dev/ram0 dengan perintah mkfs:

```
# mkfs /dev/ram0
mke2fs 1.36 (05-Feb-2005)
Filesystem label=
OS type: Linux
```

```
Block size=1024 (log=0)
Fragment size=1024 (log=0)
16000 inodes, 64000 blocks
3200 blocks (5.00%) reserved for
the super user
First data block=1
8 block groups
8192 blocks per group, 8192
fragments per group
2000 inodes per group
Superblock backups stored on
blocks:
            8193, 24577, 40961,
57345
Writing inode tables: done
Writing superblocks and
filesystem accounting
information: done
This filesystem will be
automatically checked every 22
mounts or
```

```
180 days, whichever comes
first. Use tune2fs -c or -i to
override.
```

Langkah ketiga, kita perlu melakukan mounting agar /dev/ram0 bisa dipergunakan:

```
# mount /dev/ram0 /mnt/ram
```

Kita dapat melihat output pada df untuk melihat ukuran ramdisk yang kita gunakan:

```
# df -h /dev/ram0
Filesystem      Size  Used
Avail Use% Mounted on
/dev/ram0        61M   13K
58M    1% /mnt/ram
```

Luar biasa bukan program dd? Apa yang kita bahas hanyalah sebagian dari kemampuan dd. Dengan penggunaan yang tepat, dd bisa sangat banyak membantu selama kita bekerja dengan Linux. Selamat mencoba!

Noprianto (noprianto@infolinux.co.id)

Professional 100% Linux Training & Solution

Ingin Menguasai Linux Secara LENGKAP ?!

Paket A-Z Linux

- Linux Concept & Fundamental
- Linux System Administration
- Linux Internet + Intranet Server
- Linux Security

56 hours (14 day @ 4 hours)

Only : Rp.4.850.000,-

Special Offer Crash Programme !

PATIN (Paket Intensif)

- Linux Concept and Fundamental
- Linux System Administration
- Linux Internet + Intranet Server

42 hours (6 days@ 7 hour)

Only : Rp.3.750.000,-

PAKIS (Paket Ekonomis)

- Linux Concept and Fundamental
- Linux System Administration
- Linux Internet + Intranet Server.

44 hours (11 days@ 4 hour)

Only : Rp.3.650.000,-

Ketik: Info PATIN atau Info PAKIS kirim SMS ke 0856 7771030 SMS Server powered by eSMSis (www.eSMSis.com)

SMS Server & Gateway

eSMSis Ver. 1.5

- Web based, Internet Ready
- Broadcast, GroupCast, MemberCast
- Scheduled SMS, Product Informations
- Auto Response, Remote SMS, Alert, etc

visit: www.eSMSis.com

MySMSPass

Start Making Money from your Website!

- SMS Authentication System for Web Content
- Short Number by Telco Operators

Demo Website : www.InfoLINUX.web.id/sections



Linuxindo

PUSAT : Wisma Bisnis Indonesia Suite #415 - JAKARTA

BANDUNG: (022) 7234192 - CIREBON: (0231) 200418 - SOLO: (0271) 662318

PERINGATAN ! Linux bisa membuat Anda kecurdian, menambah PD dan belum ada obatnya. Tidak Setiap Paket Promosi tersedia di Cabang.

(021) 5362390
www.Linuxindo.com