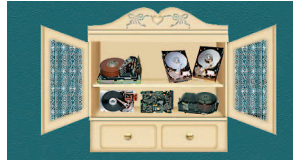


Software RAID di Linux

Data adalah bagian yang sangat penting dalam komputasi. Kehilangan data harus senantiasa dicegah. Berbagai cara bisa ditempuh. Menggunakan RAID adalah salah satunya.



Bayangkanlah skenario berikut ini. Perusahaan Anda menyimpan hampir segalanya secara tersentralisasi. Hampir semua data di simpan di dalam satu *server*. Dan di dalam satu harddisk. Backup dilakukan per minggu dan kemudian datanya disimpan ke dalam CDROM secara otomatis. Pada suatu hari, tanpa diduga, harddisk tersebut tiba-tiba berdemonstrasi mogok kerja seperti fenomena yang belakangan ini mudah ditemui di belantara ketenagakerjaan Indonesia. Apa yang pertama-tama terbayang sebagai solusi? Beli harddisk baru. Kembalikan data hasil *back-up* terakhir. Dan kalau cukup berminat, harddisk lama diserahkan ke tukang servis. Cukup banyak waktu yang terbuang. Dan cukup banyak darah yang mengalir ke otak karena *stress*.

Apabila Anda rajin melakukan back-up, hal tersebut tentunya sudah menjadi modal untuk mencegah kehilangan data. Namun, dengan investasi sekitar Rp1 juta atau kurang, dan membeli satu atau dua harddisk baru, kita dapat membuat RAID secara *software*. Walaupun dalam praktik ada harddisk yang harus dikorbankan dan hanya menjadi *mirror* bagi harddisk pertama, data yang kita miliki akan relatif lebih aman lagi karena apabila salah satu harddisk rusak, harddisk tersebut tinggal dicabut dan digantikan dengan harddisk baru. Proses *mirroring* kemudian dilakukan, dan kini Anda memiliki dua harddisk yang saling *mirror*. Sekali lagi: satu rusak, ganti, *mirror*, dan Anda pun tenang kembali. *Mirror* hanyalah salah satu dari cara kerja RAID. Masih terdapat beberapa cara lainnya. Artikel ini difokuskan pada RAID level 1 yang bekerja secara *mirroring*.

Untuk RAID Level 1 atau RAID1 untuk lebih singkat, kita perlu menyediakan dua atau lebih harddisk yang kemudian akan saling menyamakan isinya dalam *block* level. Artinya, apabila Anda memiliki dua

buah harddisk masing-masing berukuran 20 GB, maka kedua harddisk tersebut akan melakukan *mirroring* sehingga total kapasitas Anda hanyalah 20 GB. Apabila Anda menggunakan tiga harddisk, maka total kapasitas Anda tetap 20 GB. Sebuah pengorbanan memang. Satu lagi kabar buruk. Semua elemen RAID1 harus memiliki ukuran yang sama. Apabila ada ukuran yang tidak sama, maka kapasitas total adalah kapasitas yang paling kecil. Pemborosan terjadi di sana sini.

Anda masih dapat mempergunakan *spare disk*, yang berarti penggunaan harddisk akan menjadi lebih boros lagi. Namun, keamanan data akan lebih terjaga karena proses rekonstruksi data/*mirroring* akan segera dilakukan apabila salah satu dari *mirror* mengalami kerusakan.

Bagaimana dengan kecepatan penulisan pada RAID1? Kecepatannya akan sedikit di bawah kecepatan penulisan pada harddisk tunggal karena data yang ditulis harus di-*copy*-kan ke setiap elemen RAID. Begitupun dengan kecepatan baca. Untuk kernel 2.4, kecepatan baca/tulis telah dikembangkan sangat lanjut sehingga faktor kecepatan tidak perlu dikhawatirkan lebih lanjut.

Barangkali pertanyaan yang lebih mendasar. Kenapa harus menggunakan RAID? Jawabannya bisa sangat bervariasi. Salah satunya adalah karena keamanan data lebih terjamin. Backup saja tidak cukup karena umumnya ada selisih waktu yang cukup berarti. Dan selain itu, Anda bisa mempergunakan gabungan dari beberapa harddisk yang mungkin tidak terpakai. RAID tidak selalu harus *mirroring* (RAID1) yang boros.

Tertarik? Kalau ya, ternyata RAID cukup manja. Anda harus mempersiapkan segala sesuatu yang diperlukan. Kita akan bekerja dengan RAID1.

Yang pertama-tama, siapkan satu harddisk untuk percobaan. Atau, dalam

bentuk yang lebih jelek lagi, siapkan dua partisi kosong berukuran sama apabila Anda tidak memiliki harddisk ekstra. Penulis menggunakan satu harddisk pinjaman yang penulis bagi menjadi dua partisi berukuran sama besar. Harap diperhatikan bahwa seharusnya penggunaan RAID tidak dilakukan pada harddisk yang sama karena fungsinya akan menjadi tidak jelas. Performa pun akan turun dengan drastis. Kita mempergunakan harddisk yang sama hanya sebagai contoh.

Dua elemen RAID1 yang dibuat harus memiliki tipe partisi Linux RAID Autodetect, yang memiliki kode fd.

Yang kedua, dan yang paling merepotkan, adalah mengaktifkan dukungan RAID pada kernel. Saat ini, di kernel 2.4, penggunaan *software RAID* telah mengalami banyak peningkatan performa dan kemudian. Pastikan Anda memiliki dukungan RAID diaktifkan di dalam kernel Anda. Apabila belum, maka kompilasi kernel harus dilakukan. Aktifkan dukungan RAID pada bagian *Multi-device support* | *Multiple devices driver support* (RAID and LVM). Saat ini, kernel mendukung mode *Linier*, *stripping* (RAID0), *mirroring* (RAID1), *RAID4/5*, *Multipath*. Sebaiknya, masukkan fasilitas ini sebagai fasilitas *built-in*.

Yang terakhir, siapkan program *mdadm*. Beberapa saat yang lalu, penulis sibuk menggunakan kumpulan program *raidtools*. Akan tetapi, ternyata *mdadm* lebih mudah untuk digunakan. Apabila Anda menggunakan Debian, berikan perintah berikut ini untuk melakukan instalasi *mdadm*:

```
apt-get install mdadm
```

Untuk distro lain, carilah terlebih dahulu paket untuk distro Anda. Apabila tidak ditemukan, downloadlah *mdadm* di <http://www.cse.unsw.edu.au/~neilb/source/mdctl/>. *Mdadm* dahulu bernama *mdctl*.

Apabila ketiga persyaratan tersebut telah dipenuhi, maka kita siap untuk bekerja dengan RAID.

Penulis menggunakan contoh yang jelek dengan penggunaan partisi-partisi dalam satu harddisk. Device */dev/hdb1* dan */dev/hdb2* akan penulis korbankan sebagai

elemen-elemen dalam RAID1. Device RAID sendiri tidak akan menggunakan lagi nama *device* setiap elemennya, melainkan menggunakan nama *device* /dev/mdx, di mana x dimulai dari 0. Dengan demikian, mulai saat ini, kita tidak lagi bicara /dev/hdb1 ataupun /dev/hdb2. Kita akan bicara /dev/md0.

Untuk menggunakan RAID, kita harus membuatnya terlebih dahulu. Berikan perintah berikut ini untuk membuat RAID1 bernama /dev/md0 yang berisikan 2 elemen, /dev/hdb1 dan /dev/hdb2:

```
# mdadm --create /dev/md0 --level=1 --
raid-devices=2 /dev/hdb1 /dev/hdb2
```

Apabila harddisk Anda tiba-tiba bekerja cukup keras, maka proses konstruksi/mirroring sedang dilakukan. Lihat prosesnya dengan membaca file /proc/mdstat.

```
# cat /proc/mdstat
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 hdb2[1] hdb1[0]
      511936 blocks [2/2] [UU]
      [= = >.....] resync =
      15.6% (80384/511936) finish=5.2min
      speed=1356K/sec
      unused devices: <none>
```

Setelah file sistem dibuat di /dev/md0, maka untuk selanjutnya, device tersebut telah siap digunakan. Anda dapat melakukan proses mount seperti halnya proses mount pada device biasa. Berikut ini adalah contoh keluaran program df -h untuk /dev/md0 yang telah di-mount ke /mnt.

```
# df -h
Filesystem      Size  Used Avail Use%
Mounted on
/dev/hda1       3.5G  3.2G  264M
93% /
/dev/md0        1.2G  47M  1.2G
4% /mnt
```

Tertarik melihat informasi seputar /dev/md0 kita? Cobalah berikan perintah berikut ini:

```
# mdadm --query --detail /dev/md0
```

Informasi akan ditampilkan dengan sangat mendetail. Dari informasi yang ditampilkan, telitilah memperhatikan State, Active Devices, Working Devices, dan

Failed Devices. Terkadang, salah satu elemen dalam RAID Anda mungkin mengalami masalah.

Lanjut untuk penggunaan RAID lebih lanjut. Terkadang kita ingin menghentikan penggunaan RAID karena sesuatu alasan. Oleh karena itu, berikanlah perintah seperti berikut ini:

```
# mdadm --stop /dev/md0
```

Untuk menjalankan kembali, berikanlah perintah berikut ini:

```
# mdadm --assemble /dev/md0 /dev/hdb1
/dev/hdb2
```

Sampai di sini, kita telah melihat penggunaan dasar dari RAID1. Dalam setiap contoh sebelumnya, kita bisa melihat bahwa semuanya bekerja dengan baik. Tidak ada *error* yang terjadi. Namun, terkadang di dunia nyata kita harus berurusan dengan rusaknya salah satu elemen *array* sehingga elemen tersebut harus dihentikan, atau RAID harus dijalankan dengan 1 elemen saja. Kita akan menyimulasikan kerusakan pada /dev/hdb2 sehingga /dev/md0 kini hanya memiliki elemen /dev/hdb1.

```
# mdadm --assemble --run /dev/md0 /dev/
hdb1
mdadm: /dev/md0 has been started with 1
drive (out of 2).
```

```
# cat /proc/mdstat
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 hdb1[0]
      511936 blocks [2/1] [U_]
      unused devices: <none>
```

Tiba-tiba, elemen kedua telah diperbaiki atau digantikan dengan yang baru. Kita bisa saja memasukkan kembali elemen tersebut ke dalam array yang sedang aktif dengan perintah berikut ini:

```
# mdadm -a /dev/md0 /dev/hdb2
mdadm: hot added /dev/hdb2
```

```
# cat /proc/mdstat
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 hdb2[2] hdb1[0]
      511936 blocks [2/1] [U_]
      unused devices: <none>
```

```
[>.....] recovery = 2.8%
(14528/511936) finish=6.2min
speed=1320K/sec
unused devices: <none>
```


Program mdadm juga dilengkapi dengan fasilitas untuk melakukan *monitoring*. Pada saat monitoring dilakukan, mdadm akan melaporkan *event* yang terjadi dalam periode waktu tertentu. Setiap laporan dapat dikirimkan kepada e-mail yang telah ditentukan.

Mdadm adalah program yang sangat kompleks. Rujuklah kepada manualnya untuk cara penggunaan lebih lanjut.

Bermacam-macam level RAID

Selain level 1, kita mengenal bermacam-macam level RAID. Tidak perlu harus menggunakan semuanya. Nomor level yang lebih tinggi tidak menunjukkan peningkatan kinerja atau peningkatan performa secara langsung. Pilih satu yang cocok Anda gunakan. Dalam beberapa kasus, RAID1 dan RAID5 adalah dua jenis RAID yang paling sering digunakan.

Pertama-tama, kita mengenal mode linear. Pada mode ini, beberapa harddisk akan ditambahkan sehingga kapasitas membesar. Tidak ada redundansi pada mode ini mengakibatkan risiko kehilangan data cukup besar. Berikutnya adalah RAID0 yang juga tidak mengenal redundansi data. Beda antara RAID0 dengan linear adalah cara penulisan pada RAID0 yang dilakukan secara paralel dengan membagi dalam *stripe-stripe* dan menuliskannya ke setiap elemen. RAID1 telah kita bahas. Sementara RAID4 dan RAID5, terutama RAID5, direncanakan sebagai RAID yang paling cocok digunakan. Menggunakan sistem *parity* yang tersebar ke setiap elemen, RAID5 lebih sering digunakan daripada RAID4 yang menyimpan parity dalam satu elemen.

RAID adalah salah satu cara untuk meminimasi risiko hilangnya data. Walaupun sedikit boros, hati akan lebih tenang. Setiap ada kerusakan, proses rekonstruksi secara otomatis akan dilakukan. Sampai di sini perkenalan kita dengan RAID di Linux. Selamat mencoba!  **Noprianto** (noprianto@infolinux.co.id)