

# Task Scheduler di Linux

Menggunakan Linux berarti kita bebas meminta Linux untuk melakukan tugas tertentu sesuai jadwal yang kita miliki.

Linux adalah sistem yang dapat diandalkan untuk melayani terus menerus selama berbulan-bulan. Sistem Linux jarang harus di-restart setiap beberapa hari. Umumnya, berdasarkan pengalaman, restart hanya dibutuhkan setiap beberapa bulan sekali. Luar biasa, bukan?

Dalam keseharian kita sebagai pengguna, administrator ataupun *developer*, terkadang ada tugas tertentu yang ingin kita kerjakan setiap waktu tertentu. Mulai dari mem-burn CD setiap selesai jam kantor, melakukan *back-up* setiap tengah malam, melakukan *update* setiap subuh, dan sebagainya. Tentunya, tidak mungkin bagi kita untuk melakukannya secara manual.

Sekilas tugas yang akan kita lakukan terdengar cukup kompleks. Untungnya, di Linux, kita sudah mendapatkan fasilitas tersebut tanpa harus mengeluarkan uang sepeserpun. Kita tidak membutuhkan daemon-daemon canggih yang perlu dibeli dari pihak ketiga dan sebagainya. Yang kita butuhkan hanyalah sebuah aplikasi dengan nama yang cukup sederhana: *cron*.

Sejarah *cron* sudah dimulai belasan sampai puluhan tahun lalu. *Cron* tidak hanya digunakan di Linux, namun juga digunakan diberbagai varian UNIX. Penerapan *cron* sendiri juga tercatat cukup banyak. Walau, umumnya penerapan *cron* yang sering ditemukan di berbagai distro Linux adalah Vixie Cron, *cron* yang diterapkan oleh Paul Vixie ([paul@vix.com](mailto:paul@vix.com)).

Dalam instalasi Linux *default*, umumnya *cron* sudah terinstal dan dijalankan secara otomatis pada saat *booting*. Untuk mengetahuinya, cobalah periksa proses *booting* Anda. Carilah sesuatu yang mirip-mirip dengan Starting CRON Daemon, Starting crond, atau Starting Task Scheduler. Apabila Anda menemukan tulisan tersebut, maka dipastikan *cron* telah terinstal di sistem Anda.

Apabila Anda tidak menemukan tulisan yang mirip-mirip dengan salah satu yang disebutkan sebelumnya, maka periksalah apakah *cron* telah terinstal. Cobalah jalankan perintah berikut ini:

```
crontab -l
```

Apabila tidak terdapat pesan kesalahan karena file tidak ditemukan, maka *cron* telah terinstal. Anda dapat menjalankannya dengan perintah berikut ini:

```
/etc/init.d/cron start
```

Apabila perintah *crontab -l* tersebut menampilkan kesalahan karena file tidak ditemukan, maka sepertinya *cron* tidak terinstall di sistem. Installah terlebih dahulu. Tidak perlu membuka Internet atau mencari ke teman-teman, karena boleh dikatakan hampir 100% distro Linux populer pasti memaketkan *cron* bersamanya.

Apabila *cron* daemon telah dijalankan, maka ada sebuah daemon yang akan memeriksa daftar tugas Anda. Dengan demikian, apa yang perlu Anda lakukan hanyalah membuat daftar tugas dan memerintahkan setiap kapan saja tugas-tugas tersebut perlu dijalankan.

Konfigurasi *cron* terdapat di */etc/crontab*. Pada kenyataannya, *cron* memiliki konfigurasi system wide seperti */etc/crontab* tersebut, namun, *cron* juga mengerti bahwa setiap user berhak untuk memiliki daftar tugas sendiri. Oleh karena itu, kita tidak akan mengutak-atik sedikitpun */etc/crontab*. Kita hanya akan berurusan dengan tugas-tugas kita sendiri.

Daftar tugas disebut sebagai *cron table*, atau *crontab* untuk singkatnya. Sebelum kita melihat bagaimana membuat *cron table* tersebut, kita perlu memahami bahwa aplikasi yang kita akan gunakan hanyalah aplikasi *crontab*, yang akan menerima sejumlah parameter untuk bekerja dengan *cron table* kita.

*Cron table* pada dasarnya adalah sebuah file yang umumnya diletakkan di /



*var/spool/cron*. Namun, tidak disarankan untuk mengedit file di direktori tersebut langsung dengan editor. Selalu gunakan *crontab*.

Untuk melihat *cron table* kita, berikan perintah berikut ini:

```
crontab -l
```

Untuk menghapus *cron table* kita, berikan perintah berikut ini:

```
crontab -r
```

Untuk mengedit *cron table* kita, berikan perintah berikut ini:

```
crontab -e
```

Secara default, aplikasi *crontab* akan mengedit *cron table* untuk user aktif.

## Aturan-aturan table cron

Berikut ini adalah format dari *crontab*:

```
<MINUTE> <HOUR>
<DAY_OF_MONTH> <MONTH>
<DAY_OF_WEEK> <CMD>
```

Nilai-nilai yang mungkin untuk setiap field tersebut:

```
<MINUTE> : 0-59
<HOUR> : 0-23
<DAY_OF_MONTH> : 1-31
<MONTH> : 1-12
<DAY_OF_WEEK> : 0-7 (0 dan 7 adalah minggu)
<CMD> : perintah yang akan dijalankan
```

Nilai field bisa juga dituliskan sebagai \*, yang berarti dari nilai minimal sampai nilai maksimal. Selain itu, nilai dapat juga dituliskan sebagai range, seperti 1-5. Yang menarik, kita juga dapat menentukan kelipatan seperti setiap 2 jam sekali. Untuk itu, kita dapat menuliskannya sebagai \*/2.

Berikut ini adalah contoh-contoh untuk memperjelas. Semua contoh ini diketikkan

di editor ketika Anda memberikan perintah crontab -e. Pastikan Anda menyimpan setiap kali keluar dari editor tersebut.

#### Contoh 1:

```
5 0 * * * $HOME/bin/daily.job >>
$HOME/tmp/daily.out 2>&1
```

Contoh ini akan menjelaskan eksekusi program daily.job yang disimpan di \$HOME/bin setiap jam 00:05 setiap harinya. Lima pada <MINUTE> artinya menit ke-5. Nol pada <HOUR> artinya jam ke-0. Sementara, \* yang pertama artinya adalah setiap hari dalam satu bulan. Tanda \* yang kedua artinya setiap bulan dalam satu tahun. Tanda \* terakhir artinya setiap hari dalam satu minggu. Cron table ini praktis akan menjalankan program daily.job setiap hari pada jam 00:05.

#### Contoh 2:

```
5 0 1 * * $HOME/bin/monthly.job >>
$HOME/tmp/monthly.out 2>&1
```

Contoh ini akan menjelaskan eksekusi program monthly.job yang disimpan di \$HOME/bin setiap jam 00:05 pada hari pertama setiap bulannya. Sebagai tambahan dari keterangan contoh sebelumnya, 1 pada <DAY\_OF\_MONTH> artinya setiap hari pertama dalam suatu bulan. Artinya, perintah monthly.job akan dijalankan pada pukul 00:05 setiap hari pertama setiap bulannya. Setiap bulan ditunjukkan dengan tanda \* pertama.

#### Contoh 3:

```
5 18 * * 1-5 $HOME/bin/workday.job >>
$HOME/tmp/workday.out 2>&1
```

Contoh ini akan menjelaskan eksekusi program workday.job yang disimpan di \$HOME/bin setiap jam 18:05 pada hari senin sampai jumat. Sebagai tambahan dari keterangan contoh sebelumnya, 1-5 merupakan *range* yang menunjukkan hari ke-1 sampai hari ke-5 dalam satu minggu. Lima pertama berarti menit ke lima. Sementara, 18 berarti jam 18, yang berarti pukul 18:05. Bintang pertama artinya setiap hari, sementara bintang kedua artinya setiap bulan.

#### Contoh 4:


```
5 0-23/2 * * * $HOME/bin/hourly2.job
>> $HOME/tmp/hourly2.out 2>&1
```

Contoh ini akan menjelaskan eksekusi program hourly2.job yang disimpan di \$HOME/bin setiap jam 00:05, 02:05, 04:05, 08:05 dan seterusnya, setiap hari. Kenapa bisa terjadi kelipatan dua? Karena kita menspesifikasikan notasi setiap jam (0-23), namun kita menambahkan notasi kelipatan dua dengan menambahkan /2. Artinya, dijalankan setiap dua jam.

Sebagai catatan, dalam contoh di atas, kita selalu menuliskan perintah yang diakhiri dengan >> dan 2>&1. Notasi tersebut artinya, keluaran dari setiap program yang kita jalankan akan diredireksi (dengan cara ditambahkan) ke file tertentu (yang diakhiri dengan .out). Apabila terjadi kesalahan, maka kesalahan juga akan dicatat dalam file .out tersebut. Apabila diperlukan, sebaiknya, kita menjaga agar keluaran dari program tidak ditampilkan ke standard output dan pesan kesalahan tidak ditambahkan di standard error.

Untuk program, tentunya kita dapat memanggil program-program standar sistem seperti ls, tar, gzip, dan sebagainya. Hanya, jangan lupa untuk memperhatikan PATH. Sebagai tips, apabila perintah yang akan Anda berikan bukan merupakan perintah tunggal, sebaiknya kumpulkan dalam sebuah *shell script*, dan panggil shell scrip tersebut sebagai perintah.

Banyak sekali kasus kesalahan yang terjadi hanya karena lupa untuk menyebutkan PATH untuk suatu program. Kalau menggunakan shell script, jangan lupa untuk memastikan bahwa shell script tersebut *executable*. Sebaiknya, sebelum diletakkan sebagai cron job, pastikan Anda telah mencoba shell script atau perintah-perintah yang diinginkan.

Penerapan yang cukup menarik barangkali adalah penerapan alarm yang memanggil XMMS dengan nama file tertentu, dan dijalankan setiap hari kerja pada jam bangun Anda. Bagi administrator sistem, cobalah untuk mencatat keadaan sistem secara periodik. Selamat mencoba! 

**Noprianto** ([noprianto@infolinux.co.id](mailto:noprianto@infolinux.co.id))

# IKLAN