

Otomasi Shell Script Lanjut

Zaid Romegar Mair

romegardm@gmail.com

<http://mairzaid.com> atau <http://mairzaid.wordpress.com>

Lisensi Dokumen:

Copyright © 2003-2018 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Dalam *script* tidak hanya mengeksekusi perintah sekuensial saja, yang kita perlukan adalah memeriksa kondisi tertentu atau melanjutkan ke logika tertentu sesuai dengan apa yang diinginkan oleh *script* tersebut yang harus dijalankan. Inilah yang kita lakukan dalam otomasi. Otomasi mengacu pada *performing* tugas yang dilakukan, urutannya akan berubah sesuai dengan perubahan dalam lingkungan pemrograman. Contoh sederhana, dengan memeriksa apakah direktori ada, jika ada maka ubahlah kedirektori itu, jika tidak buatlah sebuah direktori baru dan kemudian lanjutkan. semua kegiatan ini dibuat dalam keputusan *shell scripts*.

Otomasi script yang ada pada tulisan ini adalah : Konstruksi script dengan `looping`. Mengimplementasikan `for loop`, `while` dan `until`.

Perintah Looping dengan For

Untuk operasi *iteratif*, *shell bash* menggunakan tiga jenis *looping* yaitu : `for`, `while`, dan `until`. Dengan menggunakan perintah *looping*. kita bisa menjalankan sekumpulan perintah dengan jumlah terbatas untuk setiap *item* dalam daftar. Tujuan dari `looping for` adalah untuk memproses daftar elemen. Berikut sintaksnya :

```
for variable in element1 element2 element3  
  
do  
  
commands  
  
done
```

Berikut ini adalah contoh *script* sederhana dengan menggunakan looping `for`, yang menyajikan informasi tentang jeda waktu dalam menampilkan *output* dari *script* perintah yang diinstruksikan. Jeda waktu yang diberikan adalah 2 detik, dengan menunjukkan *output* berupa tanggal, kalender dibulan ini, *user* yang aktif dan membersihkan layar. *User* akan kembali pada posisi semula.

```
for jeda in `date cal who clear`
do
    sleep 2
    $jeda
done
```

Bentuk lain dari *looping* ini adalah dengan sintaks berikut:

```
for ((; ;))
do
    command
done
```

Buat *file script* dengan `for_01.sh` dan eksekusi.

```
#!/bin/bash
for angka in {1..5}
do
    echo $angka
done
```

Keluar dari Iterasi Loop dengan Continue

Dengan bantuan perintah `continue`, memungkinkan untuk keluar dari iterasi `loop` dan lanjut ke iterasi `loop` berikutnya. Menggunakan perintah `while` atau perintah `until` untuk iterasi `loop`. Berikut adalah skrip `forIL_01.sh` untuk `loop` dengan perintah `continue` untuk melewati bagian tertentu dari perintah `loop`:

```
#!/bin/bash
n=$1
```

```
if [ $# -eq 0 ]
then
    echo "shell script akan mencetak sebagai penjumlahan"
    echo "menggunakan :$0 nomor"
    exit 1
fi
for i in {1..10}
do
    echo "$n + $i = $(( $i + $n))"
done
$ ./forIL_01.sh
$ ./forIL_01.sh 10
```

Ada variabel tertentu yang memiliki arti khusus bagi *shell*, yang telah ditentukan oleh *shell* itu sendiri. Beberapa variabel tertentu yang termasuk dalam parameter khusus diantaranya: \$#, \$*, @\$.

- ✓ Variabel dengan simbol \$# digunakan untuk menyimpan sejumlah parameter yang diberikan pada baris perintah.
- ✓ Variabel dengan simbol \$* Digunakan untuk menyimpan nilai semua parameter yang diketikkan pada baris perintah.
- ✓ Variabel dengan simbol @\$ mempunyai fungsi sama dengan \$*, kecuali jika dalam kutip ganda ("\$@"), maka akan menempatkan setiap argumen posisi dalam kutip ganda.

Keluar dari Looping dengan Break

Dalam skrip forLB_01.sh, akan diperiksa konten direktori. Jika direktori ditemukan, maka Kita keluar dari loop dan mencetak pesan bahwa direktori pertama adalah contoh3, karena perintah untuk membuat direktori adalah pada contoh3, jika contoh tiga telah ada pada sistem komputer maka akan mencetak perintah berikutnya, dan akan terus berulang hingga tidak ada direktori yang sama namanya maka akan dicetak:

```
$ nano forLB_01.sh
#!/bin/bash
rm -rf contoh*
echo > contoh1
echo > contoh2
mkdir contoh3
echo > contoh4
for file in contoh*
do
    if [ -d "$file" ]; then
        break
    fi
done
echo direktori pertama adalah $file
rm -rf contoh*
exit 0
```

Looping dengan Do While

Perintah `while` memiliki kesamaan dengan perintah `for` dalam operasi loop. Jika berhasil atau 0 maka perintah di dalam `do` dan `done` dieksekusi. Tujuan dari sebuah loop adalah untuk menguji suatu kondisi atau ekspresi tertentu dalam menjalankan perintah yang diberikan ketika kondisinya benar (`while` loop) atau sampai kondisi menjadi *True* (`until` loop). Berikut ini merupakan sintaks untuk `while` loop atau `until` loop.

<pre>while kondisi do perintah done</pre>	<pre>until kondisi do perintah done</pre>
---	---

Contoh : Tulis *script* berikut dengan nama *file* `while_01.sh`

```
#!/bin/bash
declare -i x
var=0
while [ $var -le 5 ]
do
    echo $var
    var=$((var+1))
done
```

Menggunakan Until

Perintah `until` mirip dengan perintah `while`. Pernyataan yang diberikan dalam `loop` dijalankan selama kondisinya merupakan *true*. Begitu kondisi menjadi *false*, maka `loop` tersebut keluar. Sintaksnya yaitu :

```
until perintah
do
    perintah
done
```

Contoh `until_02.sh`

```
#!/bin/bash
input=""
until [ "$input" = keluar ]
do
    echo ""
    echo 'masukkan kata anda'
    echo '(ketik keluar untuk exit) '
    read input
```

```
echo "yang anda ketikkan adalah : $input"
```

```
done
```

Referensi

1. Diktat Kuliah dan Modul Zaid Romegar Mair, ST., M.Cs
2. Learning Linux Shell Scripting, Ganesh Sanjiv Naik
3. Pengenalan Unix dan Linux, Abdul Kadir.

Biografi Penulis



Zaid Romegar Mair. Menyelesaikan S1 Teknik Informatika di Universitas Ahmad Dahlan Yogyakarta tahun 2008 dan mendapatkan gelar Master of Computer Science (M.Cs) dari Universitas Gadjah Mada Yogyakarta pada tahun 2013 sebagai program Pascasarjana. Sekarang menjadi dosen tetap di Politeknik Sekayu Musi Banyuasin Sumatera Selatan Indonesia sejak tahun 2014 sampai dengan saat ini.