

Otomasi Shell Script Lanjut

Zaid Romegar Mair

romegardm@gmail.com

<http://mairzaid.wordpress.com>

Lisensi Dokumen:

Copyright © 2003-2017 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.

Gambaran umum :

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*. Dalam bab ini, yang akan dipelajari adalah :

- **Bekerja dengan for loop,**
- **While loop,**
- **Controlling loops yang meliputi :**
 - ✓ ***Continue statement*, dan**
 - ✓ ***Break statement***

Setelah membaca beberapa materi pendahuluan, diharapkan mampu :

- ✚ Setelah mempelajari bab ini, diharapkan mampu:
- ✚ Mengetahui konstruksi looping.
- ✚ Mengimplementasikan for loop, while dan until.
- ✚ Membedakan konstruksi kontrol loop
- ✚ Memahami pengguna konstruksi yang bersesuaian dengan keputusan yang akan dibuat.
- ✚ Mengamati dan simpulan setiap evaluasi dan percobaan pada *shell script*.

Isi Pokok Bahasan

- ✓ Materi pendukung
- ✓ Percobaan
- ✓ Ringkasan
- ✓ Tugas

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
```

Percobaan

1. Ikuti langkah *script* berikut, simpanlah, periksa hak akses *file*, lakukan eksekusi *file*.

- a) Buatlah *script* dengan nama `for_1a.sh`. Lakukan modifikasi *script* dengan menyisipkan optional `-n` dan menghilangkan optional `-n` diantara variabel `I` dan `echo`. Jika petik yang mengapit variabel `I` di rapatkan. Apa yang terjadi?

```
#!/bin/bash
```

```
max=5
```

```
for ((i=1; i<=max; i++))
```

```
do
```

```
        echo "$i"
    done
```

b) Buatlah *script* dengan nama `for_1b.sh`

```
#!/bin/bash

for perintah in date pwd df
do
    echo
    echo "output dari $perintah perintah"
    $perintah
    echo
done
```

2. Iterasi *looping* dengan `continue`

```
$ nano forIL_02.sh

#!/bin/bash

rm -rf contoh*

echo > contoh1

echo > contoh2

mkdir contoh3

echo > contoh4

for file in contoh*
do
    if [ -d "$file" ]
    then
```



```
        echo "lewati direktori $file"
        continue
    fi
    echo file adalah $file
done
rm -rf contoh*
exit 0
$ ./forIL_02.sh
file adalah contoh1
file adalah contoh2
lewati direktori contoh3
file adalah contoh4
```

3. Menghitung luas persegi dengan nama *file* forLB_02.sh

```
#!/bin/bash
typeset -i sisi=0
while true
do
    echo -n "inputkan nomor (0 untuk exit):"
    read sisi
    if (( sisi == 0 ))
    then
        break
    else
        echo " inputkan sisi $sisi"
```

```
        echo " luas persegi adalah $(( sisi *
sisi ))"

        fi

done

echo

echo "Good, anda mengerjakannya dengan benar :)"
```

4. Buatlah *script* berikut dengan nama *file* while_02.sh

```
#!/bin/bash

input=""

while [ "$input" != keluar ]
do

        echo ""

        echo 'masukkan kata anda'

        echo '(ketik keluar untuk exit)' $

        read input

        echo "yang anda ketikkan adalah : $

done
```

Ringkasan

Pada pertemuan 12 dan 13 ini, kita belajar tentang pengambilan keputusan dalam skrip. Bekerja dengan Test, if-else, dan switching case. Pada bab ini juga dipelajari penggunaan select *looping* dengan menu. Untuk melakukan pengulangan tugas, seperti dalam daftar pemrosesan misalnya, bisa menggunakan for loop, while loop dan do while. Diakhir bab, juga dipelajari tentang control loop dengan menggunakan break statement dan continue statement. Pertemuan berikutnya akan membahas tentang fungsi.

Tugas

1. Buatlah perkalian 10 sebagaimana pada *file* `forIL_01.sh` ?
2. Hitunglah luas persegi panjang dengan *looping* `break`. Tulis *script* dan tampilkan *outputnya* ?
3. Buatlah *script* dengan melakukan inputan tiga suku kata nama anda yang diinputkan dengan perintah nama pertama anda, nama kedua anda dan nama ketiga anda. Kemudian gabunglah ke tiga suku kata tersebut dengan mencetak pernyataan nama lengkap anda adalah : nama pertama, kedua, ketiga. Ketikkan `exit` untuk keluar.
4. Perbaiki *script* berikut ini dan berikan *outputnya* berdasarkan pilihan (ada 3 menu pilih) yang dipilih.

```
#!/bin/bash

pilih=1

while
do

    echo "1. siapa aktif sekarang"

    echo "2. kalender bulan ini"

    echo "3.keluar"

    read pilih

    if [ $pilih -eq 3 ]

    then

        break

    clear

done

echo "selamat mengerjakan"
```

5. Buatlah *script* sederhana dengan menggunakan konstruksi `until`. Berikan penjelasan dari *script* yang anda tulis dan tampilkan output.
6. Amati dan berikan penjelasan anda dari pekerjaan yang anda lakukan pada percobaan satu sampai percobaan empat diatas !

Glosarium

1. Control : Kendali
2. Loop : Perulangan
3. Continue : Melanjutkan ke perintah berik
4. Break : Berhenti sejenak
5. Statement : Pernyataan
6. True : Operasi logika benar
7. False : Operasi logika salah

“ kalau orang lain bisa, maka pasti saya juga bisa melakukannya, dan jika orang lain tidak bisa, maka saya harus jadi orang pertama yang bisa.”

..: Selamat Mengerjakan :.

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.