MEMPELAJARI DASAR-DASAR SHELL SCRIPT DAN MEMBUAT SHELL SCRIPT UNTUK PEMECAHAN PERMASALAHAN DAN MENJALANKANNYA

08

EDISI I - 2007

MATA DIKLAT **:**
SISTEM OPERASI

PROGRAM KEAHLIAN **:**
SEMUA PROGRAM KEAHLIAN

DEPARTEMEN PENDIDIKAN NASIONAL
BIRO PERENCANAAN DAN KERJASAMA LUAR NEGERI

2007

MEMPELAJARI DASAR-DASAR SHELL SCRIPT DAN MEMBUAT SHELL SCRIPT UNTUK PEMECAHAN PERMASALAHAN DAN MENJALANKANNYA

EDISI I - 2007

8.1 PENDAHULUAN

8.1.1 Deskripsi

NAMA MODUL	**:**	**SISTEM OPERASI**
KOMPETENSI	**:**	**MEMPELAJARI DASAR-DASAR SHELL SCRIPT DAN MEMBUAT SHELL SCRIPT UNTUK PEMECAHAN PERMASALAHAN DAN MENJALANKANNYA**
SUB KOMPETENSI	**:**	**Pemrograman Shell**
KRITERIA KERJA	**:**	**-**
LINGKUP BELAJAR	**:**	• Elemen dasar shell script. • Menggunakan parameter. • Menggunakan instruksi test untuk tes kondisi dan operator logika yang terkait. • Menggunakan variabel built in • Membuat aplikasi dengan konstruksi if-then else.

8.2 Elemen Dasar Shell Script

8.3 Menggunakan Parameter

Banyak sekali perintah UNIX yang melibtkan argumen (parameter) misalnya untuk menciptakan berkas bernama *a, b* dan *c*. Anda perlu menuliskan perintah berupa :

Touch a b c

Dalam hal ini, *a, b* dan *c* masing-masing disebut sebagai argumen baris perintah. Argumen-argumen seperti ini dapat dikenalii pada skrip *shell* dan biasa disebut sebagai **parameter posisi.**

Bourne shell, bourne Again shell, dan Korn Shell	**C shell**	**Keterangan**
##	##argv	Jumlah parameter
#0	#0 #argv[0]	Nama skrip *shell*
#1	#1 #argv[1]	Parameter pertama
#2	#2 #argv[2]	Parameter kedua
...		
#{10}	#{10} #argv[10]	Parameter kesepuluh
#*	#argv[*]	Semua parameter

Tabel 8.1 parameter yang berhubungan dengan parameter posisi

Contoh skrip pada Bourne / Bourne Again / Korn shell :

 # cat parameter.sh

```
:
# ------------------------------------------------------
# parameter.sh
#
# Contoh pemrosesan parameter
#     pada Bourne / Bourne Again / Korn Shel1
# ------------------------------------------------------


echo “Jumlah parameter = ##”
echo “Semua parameter  = #*”


echo ‘Isi# 0 : ‘ # 0
echo ‘Isi# 1 : ‘ # 1
echo ‘Isi# 2 : ‘ # 2
```

Hasil eksekusi :

 # parameter.sh selamat belajar mas

```
 Jumlah parameter = 3
 Semua parameter = selamat belajar mas
 Isi #0 :  ./parameter.sh
 Isi #1 :  selamat
 Isi #2 :  belajar
```

#_

Contoh skrip pada C shell :

 # cat parameter.chs

```
# ------------------------------------------------------
# parameter.csh
#
# Contoh pemrosesan parameter
#     pada C shell
# ------------------------------------------------------


echo “Jumlah parameter = ##argv”
echo “Semua parameter = #*”


  echo ‘Isi# 0 : ‘ # 0
  echo ‘Isi# 1 : ‘ # 1
  echo ‘Isi# 2 : ‘ # 2
#_
```

Hasil eksekusi :

% **parameter.csh apa kabar, mas?**

```
  Jumlah parameter = 3
  Semua parameter = apa kabar, mas
  Isi #0 :  parameter.csh
  Isi #1 :  apa
```

Isi #2 : kabar

#_

Shell menyediakan perintah bernama **shift** yang erat kaitannya dengan pemrosesan parameter. Jika **shift** dieksekusi (tanpa parameter), maka parameter terkiri akan disingkirkan. Jika **shift** diikuti dengan suatu angka, maka parameter-parameter paling kiri sebanyak angka terseut akan dibuang. Sebagai contoh ditunjukkan pada skrip berikut :

 # cat geser.sh

```
# ------------------------------------------------------

# geser.sh

#

# Contoh untuk memperlihatkan efek shift

# ------------------------------------------------------



echo “Parameter sekarang : #*”

shift

echo “Parameter sekarang : #*”

shift

echo “Parameter sekarang : #*”
```

contoh eksekusi :

 # geser mawar melati anggrek semuanya indah

```
  Parameter sekarang : mawar melati anggrek semuanya indah

  Parameter sekarang : melati anggrek semuanya indah

  Parameter sekarang : anggrek semuanya indah
```

#_

8.4 Menggunakan Instruksi Test Untuk Test Kondisi Dan Operator Logika yang terkait

8.5 Mengenal Variabel Built In

Variabel adalah suatu nama yang dapat digunakan untuk menampung suatu nilai dan nilai yang ada padanya daapt diubah. Contoh variabel yang sering Anda jumpai dalam buku ini adalah HOME (yaitu yang berisi nama direktori login atau *home directory*). Contoh yang lain adalah PATH atau path, yang diperkenalkan di awal bab ini.

Variabel seperti HOME dan PATH adalah variabel bawaan *sheel*. Selain variabel bawaan seperti itu, pemakai bisa menciptakan sendiri suatu variabel. Kegunaannya misalya untuk menampung secara sementara hasil suatu operasi aritmetika. Contoh yang lain yaitu untuk meletakkan data yang berasal dari *keyboard*

Cara Menamakan Variabel

Nama suatu variabel bisa mengandung :

- huruf,
- angka, atau
- garis-bawah

Namun, nama variabel harus diawali dengan h huruf atau garis bawah. Adapun huruf kecil dan kapital dianggap berbeda
Beberapa contoh nama variabel buatan :

- nama
- kuartal_1
- a

Cara memberikan nilai ke variabel

Pemberian nilai ke suatu variabel dapat dilakukan melalui penugasan. Caranya adalah seperti berikut :

(a) Pada Bourne shell, Bourne Again Shell, dan Korn Shell

Variabel – nilai
Apabila *nilai* mengandung karakter khusus seperti tab atau spasi, *nilai* harus ditulis dalam tanda petik.

Contoh.
Nama=”Andi Setiawan”

Memberikan string “Andi Setiawan” ke variabel *nama*. Perlu dikethaui, tepat sebelum dan sesudah tanda sama dengan (=) tidak bole ada spasi.

(b) Pada shell

Perintah serupa pada Bourne / Bourne Again / Korn shell di atas pada C shell berupa :

Set nama =" Andi Setiawan"

Pemberian nilai ke variabel pada C shell diawali dengan kata **set**. Sebelum dan sesudah tanda sama dengan (i) harus mengandung minimal sebuah spasi (tetapi ada juga sistem yang boleh tidak menyertakan spasi pada posisi tersebut).

Khusus untuk variabel yang menampung nilai numerik, bentuk penugasannya berupa :

@ *variabel* = *nilai_numeris*

Perlu diketahui, antara @ dan nama variabel perlu dipisahkan oleh minimal sebuah spasi.

Hasil suatu perintah dapat diletakkan ke variabel, dengan meletakkan perintah di dalam tanda *backquote*, contoh :

```
Info='date'        # pada Bourne/Buorne Again/Korn shell
Set info 'date     # pada C shell
```

Dengan cara seperti itu, hasil dari **date** akan ditaruh ke variabel info.

Cara Memperoleh isi Variabel

Untuk memperoleh isi suatu variabel, variabel perlu dawali dengan tanda #.

Contoh 1 :

Echo #PATH

Digunakan untuk menampilkan isi variabel PATH

Contoh 2 :

a=#b

Merupakan contoh untuk memberikan nilai variabel *b* ke variabel *a* pada Bourne, Bourne Again dan Korn shell.

Catatan :

Khusus pada C shell, jika suatu variabel yang diacu ternyata belum ada, pesan semacam berikut akan ditampilkan.

% **set a = #b**

B : undefined variable.

%_

8.6 Membuat Aplikasi Dengan Konstruksi If-Then-Else

Normalnya pengeksekusian perintah di dalam skrip adalah secara sekuensial. Dalam prakteknya, seringkali suatu skrip mengandung perintah pencabangan dengan kondisi yang menentukan cabang yang akan dijalankan. Hal ini memungkinkan suatu perintah atau beberapa perintah hanya dieksekusi kalau suatu kondisi terpenuhi (bernilai benar). Perintah yang mendukung hal seperti ini adalah **if** dan **case** (Bourne / Bourne Again/Korn shell) atau **if** dan **swicthc** (C shell).

Sebelum membicarakan perintah-peritnah ini ada baiknya untuk mengetahui nilai keluar dan juga perintah **test**, yang menentukan nilai kondisi.

Nilai Keluar (*exit Code*)

Nilai keluar adalah sutu nilai yang diberikan oleh suatu perintah setelah selesai dikerjakan. Nilai ini bermanfaat untuk memberikan isyarat tentang keberhasilan tuas yang diemban oleh suatu perintah. Nilai keluar berkisar antara 0 sampai 255.

Nilai keluar suatu perintah dapat diketahui dengan memeriksa isi variabel :

- ? pada Bourne/Bourne Again/Korn shell
- status pada C shell

sebagai contoh :

```
# grep AMIR tgllahir.dat
# echo #?
1
#_
```

Nilai 1 di atas menyatakan nilai keluar perintah **grep**. Karena hasilnya tidak berupa nol berarti ada sesuatu kegagalan pada **grep**. Dalam hal ini menyatakan bahwa AMIR tidak ada pada berkas tgllahir.dat

Perintah test

Perintah **test** adalah perintah *built-in* pada Bourne shell dan juga Bourne Again shell yang berguna untuk menguji suatu kondisi.

Perintah	:	Test
Kategori	:	*built-in shell*
Fungsi	:	menguji suatu kondisi string, angka atau berkas
Format	:	
		Test ungkapan
Kriteria		Hasil BENAR jika
String		tidak kosong

-n string panjang string > 0
-z string panjang string = 0
String 1 = string2 sama
String1 ! = string2 tidak sama

Ungakpan bilangan bulat
(bentuk *intl operator int1*)

Kriteria	Hasil BENAR jika
-gt	intl lebih besar daripada int2
-ge	intl lebih besar daripada atau sama dengan int2
-eq	intl sama dengan int2
-ne	intl tidak sama dengan int2
-le	intl kurang dari atau sama dengan int2
-lt	intl kurang dari int2

Ungkapan nama berkas :

Kriteria	Hasil BENAR jika
-r nama file	ada danmempunyai akses READ
-w namafile	ada dan mempunyai akses WRITE
-f namafile	ada dan bukan direktori
-d namafile	ada dan direktori
-s namafile	ada dan berukuran > 0 byte

Operator untuk mengombinasikan ungkapan-ungkapan di atas :

!	bukan (not)
-adan	
-o	atau
(ungkapan)	pengelompokkan

Hasil : Benar (0) atau salah (tidask nol). Hasil dapat diperiksa
dengan #?

Contoh :
 # test2 –gt b
 # echo #?
0 ⇦ menyatakan bahwa 2 memang lebih besar daripada 1
#_

 # test2 –lt 1
 # echo #?
1 ⇦ menyatakan bahwa 2 memang lebih besar daripada 1

#_

 # tes –f adakah
 # echo #?
1 ☜ menyatakan bahwa berkas *adakah* tidak ada atau bukan berkas biasa
#_

 # tes –s adakah
 # tes –f adakah
 # echo #?
1 ☜ menyatakan bahwa berkas *adakah* berukuran tidak lebih dari 0 Byte

 # test"YOGYA" = "yogya"
 # echo #?
1 ☜ menyatakan bahwa kedua string tidak sama
#_

Perintah if pada Bourne / Bourne Again / Korn Shell
Bentuk sederhana perintah **if** adalah sebagai berikut :
if *perintah_kondisi*
then
perintah
fi

Pada bentuk ini, *perintah* akan dieksekusi hanya kalau *perintah_kondisi* bernilai benar.

Contoh skrip yang menggunakan **if.**
 # cat cekpar.sh

```
# ---------------------------------------------------------
# cekpar.sh
#
# Contoh penggunaan if untuk memeriksa
#     jumlah parameter posisi
#     Kalau parameter posisi tidak sama dengan 2
#     akan muncul pesan kesalahan
# ---------------------------------------------------------

if test ## - ne 2
then
  echo "penggunaan: cekpar.sh parameter1 parameter2"
  exit
fi

echo "parameter pertama : #1"
echo "parameter kedua    : #2"
```

Pada contoh di atas, jika jumlah parameter posisi tidak sama dengan dua (diuji melalui **test ## -ne2)** maka epsan
Penggunaan : cekpar.sh parameter1 parameter2
Ditampilkan dan dieksekusi dihentikan (melalui perintah **exit**)
Kalau parameter posisi ada dua buah, kedua perintah tersebut tida kpernah dijalankan.
Contoh eksekusi :

#cekpar.sh
Penggunaan : cekpar.sh parameter1 parameter2
#_

#cekpar.sh ab
Parameter pertama : a
Parameter kedua : b
#_

Bentuk kedua perinta **if** berupa :

If *perintah_kondisi*
Then
perintah_1
else
perintah_2
fi

pada bentuk ini :
- *perintah_1* dijalankan kalau *perintah_kondisi* bernilai benar, dan
- *perintah_2* dijalankan kalau *perintah_kondisi* bernilai salah.

Contoh skrip yang menggunakan **if-else** :

 # cat voucher.sh
```
# ------------------------------------------------------
# voucher.sh
#
# Contoh menentukan pemberian voucher
#     voucher diberikan kalau nilai pembelian di atas
#     atau sama dengan Rp. 100.000
# ------------------------------------------------------

echo “Total pembelian:”
read total

if test #toal –lt 100000
then
```

```
   echo “Tidak dapat voucher”
else
   echo “Dapat voucher”
fi
#_
```
Contoh eksekusi :
 # voucher.sh
Total pembelian :
100000<enter>
Dapat voucher
#_

 # voucher.sh
Total pembelian :
100000<enter>
Tidak dapat voucher
#_

Bentuk ketiga perintah **if**

If *perintah_kondisi_1*
then
   perintah_1
elit *perintah_kondisi_2*
then
   perintah_2
elif...
....
Elit *perintah_kondisi_m*
then
   Perintah_m

Bentuk ketiga ini bermanfaat untuk menangani suatu persoalan yang ditentukan oleh bermacam-macam kondisi. Sebagai contoh adalah penentuan bonus berdasarkan total suatu pembelian. Misalnya kriteria yang digunakan adalah sebagai berikut :

Kriteria	Bonu
Total ≥ 1.000.000	Kipas angin kotak
500.000 ≤ total < 1.000.000	Radio mini
100.000 ≤ total < 500.000	Setrika listrik
50.000 ≤ total < 100.000	Voucher Rp. 5.000,-

Implementasinya :

 # cat voucher.sh

```
# ------------------------------------------------------
# bonus.sh
#
# Contoh menentukan bonus
#     berdasarkan total pembelian
# ------------------------------------------------------

echo “Total pembelian : “
read total

if test #total –lt 50000
then
  echo “Bonus: Tak ada”
elif test #total –lt 100000
then
  echo “Bonus: voucher Rp. 5.000,-“
elif test #total –lt 50000
then
  echo “Bonus: setrika listrik”
elif test #total –lt 100000
then
  echo “Bonus: Radio mini”
else
  echo “Bonus: Kipas angin kotak”
fi
#_
```

Perintah if pada C shell

Bentuk **if** paling sederhana pada C shell :

If (*ungkapan*) *perintah*

Pada bentuk ini antara **if** dan *perintah* harus terletak dalam satu aris. Ungkapan yang digunakan sebagai kondisi harus diletakkan di dalam tanda kurung. Hal ini berlaku untuk semua bentuk **if**. Dalam hal ini *perintah* hanya akan dikesekusi kalau *ungkapan* bernilai benar. Untuk pembandingan nilai numerik, operator-operator relasional seperti ==, =, >, <, <=, serta >= dapat digunakan. Jika opeator ! diletakkan di depan operator relasiona, maka operator dinegasikan. Misalnya!< berarti tidak kurang dari.
Untuk pembandingan string, opertor != dan == bisa dipakai. Khusus untuk operasi berkas, ungkapan yang digunakan dapat dilihat pada tebel 23.2.

Ungkapan	BENAR jika
-e *nama_berkas*	Ada
-r *nama_berkas*	Ada dan memiliki permisi READ
-w *nama_berkas*	Ada dan memiliki permisi WRITE
-f *nama_berkas*	Ada dan berupa berkas biasa
-d *nama_berkas*	Ada dan berupa direktori

Tabel 8.1 ungkapan kondisi yang berhubungan dengan berkas

Apabila perintah terdiri atas beberapa buah, bentuk berikut yang digunakan :

if (*ungkapan*) **then**
  perintah_perintah
endif

contoh skrip shell :
% **cat cekpar.csh**
```
# ------------------------------------------------------
# cekpar.csh
#
# Contoh Penggunaan if untuk memeriksa
#     Jumlah parameter posisi
#     Kalau parameter posisi tidak sama dengan 2
#     akan muncul pesan kesalahan
# ------------------------------------------------------

If (##argv !=2) then
  Echo “penggunaan: cekpar.chs parameter1 parameter2”
  Exit
Endif

Echo “parameter pertama : #1”
Echo “parameter kedua     : #2
%_
```

Contoh di atas mempunyai fungsi serupa dengan skrip cekpar.sh.

Bentuk kedua perintah **if**:

If (ungkapan) **then**
  Perintah_1
else
  perintah_2
endif

Pada bentuk ini, *perintah_1* dieksekusi hanya kalau *ungkapan* bernilai benar. Kalau *ungkapan* bernilai salah, *perintah_2-lah* yang dijalankan.

Pemakaian bentuk **if** di atas digunakan pada contoh berikut :

% cat voucher.chs

```
# ------------------------------------------------------
# voucher.csh
#
# Contoh menentukan pemberian voucher.
#     voucher diberikan kalau nilai pembelian di atas
#     atau sama dengan Rp. 100.000
# ------------------------------------------------------

If (##argv ! = 1) then
    echo “Penggunaan : voucher.csh total_pembelian”
    exit
endif

set total =#1

if (#total <100000) then
    echo “Tidak dapta voucher”
else
    echo “Dapat voucher”
endiff
%_
```

Berbeda pada contoh *voucher.sh*, pada skrip shell ini, data total pembelian harus dimasukkan sebagai argumen (ini sengaja dibuat karena tidak semua C shell menyediakan perintah untuk membaca data dari *keyboard*).

Contoh eksekusi :

```
% voucher.csh
Tidak dapat voucher
%_
```

Bentuk **if** bertingkat pada C shell adala seperti berikut :

if (*unkapan*) **then**
  perintah
else if (ungkapan) **then**
  perintah
else if (ungkapan) **then**
...
Else
  Perintah
Endif

Contoh penerapannya dapat dilihat pada skrip shell berikut :

% **cat bonus.csh**

```
# ------------------------------------------------------
# bonus.sh
#
# Contoh menentukan bonus
#     berdasarkan total pembelian
# ------------------------------------------------------

If ( ##argv ! =1_ then
   echo “Penggunaan : bonus.csh total_pembelian”
   exit
endif

set total =#1
if (  #total < 50000 ) then
   echo “Bonus: Tak ada”
if (  #total < 100000 ) then
   echo “Bonus < voucher Rp. 5.000,-“
if (  #total < 500000 ) then
   echo “Bonus: Setrika listrik”
if (  #total < 1000000 ) then
   echo “Bonus: Radio mini”
else
   echo “Bonus: Kipas angin kotak”
endif
%_
```

Pada skrip ini, data yang menjadi penentu jenis bonus harus dimasukkan sebagai parameter pada saat pamanggilan skrip ini. Contoh eksekusi :

```
% bonus 70000
Bonus : voucher Rp. 5.000,-
%_
```