

MANIPULASI STRING PADA SCRIPT



BASH shell scripting tutorial
for complete beginner's

Apa Itu Bash Scripting?

A. Introduction

Latar Belakang

Bash adalah shell yang paling populer di dunia

Mengapa Belajar Shell Script?

- Automasi
Shell script memudahkan eksekusi perintah, membuatnya menjadi otomatis.
- Unix
Hampir semua sistem operasi open source, seperti UNIX, menggunakan shell script
- CGI
Banyak digunakan di programming CGI yaitu script untuk server untuk memproses data dan membangun halaman web atau aplikasi

Prasyarat?

- Memahami perintah dasar linux
- familiar dengan looping, variabel, dan pemrograman dasar lainnya.

B. Pengenalan Bash Shell Scripting

Apa itu shell?

shell adalah program (penterjemah perintah) yang menjembatani user dengan sistem operasi dalam hal ini kernel (inti sistem operasi), umumnya shell menyediakan prompt sebagai user interface, tempat dimana user mengetikkan perintah-perintah yang diinginkan baik berupa perintah internal shell (internal command), ataupun perintah eksekusi suatu file program (eksternal command), selain itu shell memungkinkan user menyusun sekumpulan perintah pada sebuah atau beberapa file untuk dieksekusi sebagai program.

Apa itu script?

Script dapat dibayangkan sebagai berkas program yang hanya mengandung, mencakup daftar perintah, dan dapat dieksekusi

Apa itu shell script?

Sedangkan shell script dapat dianggap sebagai file executable yang mengandung daftar perintah. Ketika kita menjalankan sebuah shell script, setiap perintah dieksekusi baris demi baris.

Apa itu bash shell script?

Maka, Bash shell scripting adalah salah satu jenis shell script yang ditulis dalam BASH. Ekstensi berkas ini adalah .sh. Bash shell merupakan shell yang cukup banyak digunakan pemakai linux karena kemudahan serta banyaknya fasilitas perintah yang disediakan.

C. Manipulasi String Pada Bash

Bash mendukung berbagai manipulasi string dan operasi aritmatika.

Catatan:

- untuk mencetak hasil, pakai perintah echo. Ikuti perintah dengan **echo**, atau simpan pada variabel tertentu perintah tersebut dan echo variabel tersebut.
- **\$** menandakan variabel
- semua perintah yang memakai **expr**, posisi mulai dari 1 bukan 0 dan perintah selain expr, indeks pertama mulai dari 0, bukan 1

1. Panjang String (sepadan dengan perintah **strlen()** pada C)

```
`expr length $variabel_string`
```

` adalah simbol di atas ~ pada keyboard

contoh:

```
stringZ=abcABC123ABCabc
echo `expr length $stringZ`
```

hasil dari contoh di atas adalah **15** dari panjang abcABC123ABCabc

2. Mencari panjang string yang sesuai dengan syarat tertentu

```
`expr "$string" : '$substring`
```

atau

```
`expr match "$string" '$substring`
```

contoh :

```
stringZ=abcABC123ABCabc
echo `expr match "$stringZ" 'abc[A-Z]*.2`
```

```
echo `expr "$stringZ" : 'abc[A-Z]*.2'`
```

contoh di atas adalah mencari panjang karakter yang mengandung huruf kecil, huruf besar, *, . , 2 yang berurutan

hasil dari contoh di atas adalah 8 dari panjang abcABC12

3. Mencari posisi dari substring tertentu mulai dari awal string (sepadan dengan perintah `strchr()` pada C)

```
`expr index "$variabel_string" syarat`
```

contoh:

```
stringZ=abcABC123ABCabc
```

```
echo `expr index "$stringZ" C12`
```

hasil dari contoh di atas adalah 6 dari posisi C pertama yang muncul

```
echo `expr index "$stringZ" 1c`
```

hasil dari contoh di atas adalah 3 dari posisi c pertama yang muncul sebelum angka 1

4. Ekstraksi Substring dari String tertentu

```
${string:posisi}
```

perintah di atas untuk mengambil karakter mulai dari posisi tertentu dari sebuah string

```
${string:posisi:panjang}
```

perintah di atas untuk mengambil karakter mulai dari 'posisi' sebanyak 'panjang' dari 'string' tertentu.

Perhatikan bahwa posisi mulai dari 0

```
${string:-panjang}
```

```
${string:(-panjang)}
```

perintah di atas untuk mengambil karakter mulai dari kiri, sebanyak 'panjang' dari 'string' tertentu.

```
`expr substr $string $posisi $panjang`
```

perintah di atas untuk mengekstrak sebanyak 'panjang' dari 'string' mulai dari 'posisi'

```
`expr match "$string" "\($substring\)`
```

perintah di atas untuk mengekstrak \$substring dari awal 'string', dimana 'substring' adalah ekspresi regular

```
`expr "$string" : "\($substring\)`
```

perintah di atas untuk mengekstrak \$substring dari awal 'string', dimana 'substring' adalah ekspresi regular

```
`expr match "$string" '.*\($substring\)`
```

perintah di atas untuk mengambil karakter dari akhir string, dimana 'substring' adalah ekspresi regular

```
`expr "$string" : '.*\($substring\)`
```

perintah di atas untuk mengambil karakter dari akhir string, dimana 'substring' adalah ekspresi regular

*)ekspresi regular:

Untuk sementara, asumsikan semua string adalah ekspresi regular. Mengenai ekspresi regular, akan dibahas dalam mata kuliah lain.

contoh:

```
stringZ=abcABC123ABCabc
```

```
echo ${stringZ:0}
```

hasil dari contoh di atas adalah **abcABC123ABCabc** dari posisi **0** hingga akhir string

```
echo ${stringZ:1}
```

hasil dari contoh di atas adalah **bcABC123ABCabc** dari posisi **1** hingga akhir string

```
echo ${stringZ:7}
```

hasil dari contoh di atas adalah **23ABCabc** dari posisi **7** hingga akhir string

```
echo ${stringZ:7:3}
```

hasil dari contoh di atas adalah **23A** dari posisi **7**, ambil 3 karakter

```
echo ${stringZ:-4}
```

hasil dari contoh di atas adalah **Cabc** yang berasal dari 4 karakter diambil dari kanan atau akhir string.

```
echo ${stringZ:(-4)}
```

hasil dari contoh di atas adalah **Cabc** yang berasal dari 4 karakter di amabil dari kanan atau akhir string.

```
echo `expr substr $stringZ 1 2`
```

hasil dari contoh di atas adalah **ab** dari posisi 1, ambil 2 karakter

```
echo `expr match "$stringZ" '\([b-c]*[A-Z]..[0-9]\)`
```

```
echo `expr "$stringZ" '\([b-c]*[A-Z]..[0-9]\)`
```

```
echo `expr "$stringZ" '\(.....\)'
```

hasil dari contoh di atas adalah **abcABC1**, dengan detail sebagai berikut:

- `.` menandakan tepat 1 karakter
- `*` menandakan konkatenasi (seperti + dalam java)
- `[]` menandakan himpunan karakter yang ingin diambil. Umpamakan `[A-C]` maka huruf yang termasuk pada himpunan itu adalah A atau B atau C

```
echo `expr match "$stringZ" '.*\([A-C][A-C][A-C][a-c]*\)'
```

```
echo `expr "$stringZ" : '.*\(...)\)`
```

hasil dari contoh di atas adalah **ABCabc** karena:

- Pada contoh ke-1, menandakan mengambil 3 karakter dari `[A-C]` dari string, kebetulan hasil yang terbentuk adalah ABC
- pada contoh ke-2 menandakan mengambil 6 karakter akhir string.

5. Menghapus substring tertentu pada string

```
${string#substring}
```

perintah di atas untuk menghapus string terpendek yang diapit mulai dari karakter pertama string.

```
${string##substring}
```

perintah di atas untuk menghapus string terpanjang yang diapit mulai dari karakter pertama string.

```
${string%substring}
```

perintah di atas untuk menghapus string terpendek yang diapit mulai dari karakter terakhir string.

```
${string%%$substring}
```

perintah di atas untuk menghapus string terpanjang yang diapit mulai dari karakter terakhir string.

Contoh:

```
stringZ=abcABC123ABCabc
echo ${stringZ#a*C}
```

menghapus dari a sampai C pertama yang ditemui. Hasilnya adalah **123ABCabc**

```
echo ${stringZ##a*C}
```

menghapus dari a sampai C terakhir yang ditemui. Hasilnya adalah **abc**

```
echo ${stringZ%b*c}
```

menghapus dari b sampai C pertama yang ditemui. Penghapusan dari kanan atau akhir string. Hasilnya adalah **abcABC123ABCa**

```
echo ${stringZ%%b*C}
```

menghapus dari b sampai C terakhir yang ditemui. Penghapusan dari kanan atau akhir string. Hasilnya adalah **a**

6. Penggantian Substring

```
${string/substring/replacement}
```

perintah di atas untuk mengganti string pertama yang ditemukan yang sesuai dengan 'substring' mulai dari karakter pertama string dengan 'replacement'.

```
${string//substring/replacement}
```

perintah di atas untuk mengganti seluruh string yang sama dengan 'substring' dengan 'replacement'.

```
${string/#substring/replacement}
```

perintah di atas untuk mengganti 'substring' pertama yang ditemukan pada 'string' dengan 'replacement'

```
${string/%substring/replacement}
```

perintah di atas untuk mengganti 'substring' terakhir yang ditemukan pada 'string' dengan 'replacement'

contoh:

```
stringZ=abcABC123ABCabc
echo ${stringZ/abc/xyz}
```

perintah di atas mengganti substring 'abc' yang pertama ditemukan dengan 'xyz'. Hasilnya adalah **xyzABC123ABCabc**

```
echo ${stringZ//abc/xyz}
```

perintah di atas mengganti seluruh substring 'abc' pada string dengan 'xyz'. Hasil

```
xyzABC123ABCxyz
```

```
echo ${stringZ/#abc/XYZ}
```

perintah di atas mengganti substring terdepan bila berupa 'abc' dengan 'XYZ'. Hasil

```
XYZABC123ABCabc
```

```
echo ${stringZ/%abc/XYZ}
```

perintah di atas mengganti substring paling belakang bila berupa 'abc' dengan 'XYZ'.

```
abcABC123ABCXYZ
```

7. Filter seluruh isi teks

Gunakan **grep** untuk mencari string atau substring tertentu dari isi suatu berkas secara keseluruhan.

Contoh penggunaan:

```
`grep -o cc $1 | wc -w`
```

perintah di atas untuk menghitung banyaknya 'cc' dalam berkas tersebut. Perhatikan bahwa **wc** adalah word count.

D. Manipulasi Operasi Matematika Pada Bash

Bash shell scripting juga dapat melakukan operasi matematika. Ada 3 cara, perhatikan contoh berikut:

```
a=10
b=5
#memakai let
let jumlah=$a+$b
let kurang=$a-$b
let kali=$a*$b

#memakai expr
bagi=`expr $a / $b`

#memakai perintah substitusi $((ekspresi))
modul=$(( $a%$b)) #sisa pembagian

echo "$a+$b=$jumlah" #10+5=15
echo "$a-$b=$kurang" #10-5=5
echo "$a*$b=$kali" #10*5=50
echo "$a/$b=$bagi" #10/5=2
echo "$a%$b=$modul" #10%5=0

#dengan substitusi, dapat langsung dilakukan perhitungan

echo "$[a+b]" #15
echo "$[a-b]" #5
echo "$[a*b]" #50
echo "$[a/b]" #2
echo "$[a%5]" #0
```

menandakan comment

Contoh keluaran ditandari warna hijau

Recommended Link :

Cooper, Mendel(2012). *Advance bash-scripting guide: chapter 10. manipulating variables. July 22, 2012.*
<http://tldp.org/LDP/abs/html/string-manipulation.html>

Fajar, Mochammad (n.d.). *Pengantar pemrograman bash shell di linux. July 22, 2012.*
<http://pemula.linux.or.id/programming/bash-shell.html>



Mills, Ashley J.S.(2005). *Unix shell scripting tutorial. July 22, 2012. The University of Birmingham.*
<https://supportweb.cs.bham.ac.uk/documentation/tutorials/docsystem/build/tutorials/unixscripting/unixscripting.html>

Revisi Terakhir :

http://bebas.vlsm.org/v24/sosp12/root/tutorial-lab/manipulasi_string-r04.pdf

Disusun Oleh : Annisa Prida Rachmadianty dari tim SOSP12

Kontributor:

Pembina :

1. Rahmat M Samik-Ibrahim
2. Ramot Stephanus
3. Rizki Mandala Putra

Tim SOSP12 (<http://sosp.vlsm.org/>):

1. Aditya Try Anggoro
2. Annisa Prida Rachmadianty
3. Ardanella Nelwan
4. Farah Nuraini
5. Nisrina Luthfiyati
6. Siska Eka Putriani
7. Zulfikar Ian Ifetayo



Copyright 2012 by SOSP12. all rights reserved. This material may be freely copied and distributed subject to inclusion of this copyright notice and our World Wide Web URL <http://sosp.vlsm.org>