

Praktikum Sistem Operasi
(Pertemuan 03)

Ekspansi Shell

Husni
husni@mail.ugm.ac.id

Garis Besar

- Perintah dan Argumen
- Operator Kendali
- Variabel

Perintah dan Argumen

- Perintah echo
- Argumen-argumen
- Perintah-perintah
- Alias

Perintah echo

- Menampilkan input yang diterimanya
- Contoh

echo Pra-S2

Pra-S2

echo Master of Computer Science UGM Yogyakarta

Master of Computer Science UGM Yogyakarta

Tahapan Eksekusi Perintah

- **Scanning** → Shell Linux melakukan scan terhadap perintah yang dimasukkan
(semua input, baik perintah atau bukan, dianggap sebagai argumen)
- **Modification** → Shell melakukan penyesuaian atau perubahan terhadap argumen
Proses ini dinamakan Shell Expansion
- **Execution** → Perintah dijalankan

Hilangnya *White Space*

- Bagian-bagian yang dipisahkan oleh satu atau lebih *white space* (atau tab) dianggap sebagai argumen
- Setiap *white space* dihapus, tidak termasuk argumen
- Contoh: perintah **echo** mencetak setiap argumen yang diterima dan dipisahkan dengan satu spasi

echo Praktikum Sistem Operasi

Praktikum Sistem Operasi

echo Praktikum Sistem Operasi

Praktikum Sistem Operasi

echo Praktikum Sistem Operasi

Praktikum Sistem Operasi

Tanda Petik Tunggal & Ganda

- Apa yang ada diantara dua tanda petik dianggap sebagai satu argumen. Tidak ada white space dalam tanda petik yang dihilangkan

- Contoh

```
echo 'Praktikum  Sistem Operasi'
```

```
Praktikum  Sistem Operasi
```

```
echo "Praktikum  Sistem Operasi"
```

```
Praktikum  Sistem Operasi
```

- Apa bedanya? :-)

Echo dan Tanda Petik

- Perintah `echo -e` dapat mengenali karakter khusus yang ada di antara 2 tanda petik
- Karakter khusus tersebut, misalnya `\n` (ganti baris) dan `\t` (tab, biasanya 8 spasi)
- Contoh
- `echo -e 'Sebentar lagi akan ada \nbaris baru'`
Sebentar lagi akan ada
baris baru
- `echo -e "Sebentar lagi akan ada \nbaris baru"`
Sebentar lagi akan ada
baris baru
- `echo -e "Sebentar lagi ada \ttabulasi di sini"`
Sebentar lagi ada tabulasi di sini
- `echo -e 'Sebentar lagi ada \ttabulasi di sini'`
Sebentar lagi ada tabulasi di sini

Perintah

- Ada 2 jenis perintah: built-in & Eksternal
- built-in: bawaan sistem operasi dan merupakan bagian dari shell.
- Eksternal: program yang mempunyai binernya sendiri, biasanya diletakkan dalam /bin atau /sbin
- Perintah type dapat digunakan untuk memeriksa jenis dari file (built in & eksternal)

Contoh

type cd

cd is a shell builtin

type more

more is /bin/more

type ls

ls is aliased to `ls --color=auto`

Menjalankan Program Eksternal

- Beberapa program mempunyai versi built-in dan eksternalnya. Built-in mempunyai prioritas lebih tinggi
- Eksekusi program eksternal harus menyebutkan lokasi file program tersimpan
- Contoh

```
type -a echo
```

```
echo is a shell builtin
```

```
echo is /bin/echo
```

```
/bin/echo "Praktikum SO      telah dimulai"
```

```
Praktikum SO      telah dimulai
```

Perintah which

- Digunakan untuk mencari program (eksternal) dalam variabel lingkungan \$PATH
- Contoh: (cd dan type perintah built-in)

which cp mv cd ls cat echo man touch more type

/bin/cp

/bin/mv

/bin/ls

/bin/cat

/bin/echo

/usr/bin/man

/usr/bin/touch

/bin/more

Alias = Nama Lain

- Dibuat menggunakan perintah **alias**
- Contoh

```
cat > hitung.txt
```

```
satu
```

```
dua
```

```
tiga
```

```
empat
```

```
alias mundur=tac
```

```
mundur hitung.txt
```

```
empat
```

```
tiga
```

```
dua
```

```
satu
```

Menyingkatkan Perintah

- Perintah yang panjang, biasanya banyak parameter dapat disingkat dengan alias
- Contoh

```
alias ll='ls -lh --color=auto'
```

```
ll
```

```
alias c=clear
```

```
alias x=exit
```

Mengatur Default Program

- Suatu program dapat dibuat berjalan dengan parameter default
- Misal: perintah rm dibuat agar defaultnya rm -i

```
touch file.txt
```

```
alias rm='rm -i'
```

```
rm file.txt
```

```
rm: remove regular empty file `file.txt'? y
```

```
ll file.txt
```

```
ls: cannot access file.txt: No such file or directory
```

-

Melihat & Menghapus Alias

- Perintah alias tanpa argumen menampilkan semua alias yang ada (aktif)
- Perintah alias diikuti argumen hanya menampilkan alias yang bernama argumen

alias ls la rm

alias ls='ls --color=auto'

alias la='ls -A'

alias rm='rm -i'

- Perintah **unalias** digunakan untuk menghapus alias

Latihan (1)

- Berapa jumlah argumen pada baris ini (selain perintah)?

```
touch '/etc/apt/source.list' 'linux ubuntu 12.04' "repo.ugm.ac.id"
```

- Apakah tac merupakan perintah shell builtin?
- Apakah ada alias untuk perintah rm
- Buat dan hapus file menggunakan perintah rm dengan opsi -i
- Buat alias bernama rm untuk rm -i. Uji dengan suatu file. Berhasil?
- Tampilkan semua alias yang ada
- Buat alias bernama kota untuk menampilkan kota asal anda. Uji. Berhasil?
- Gunakan perintah set -x untuk menampilkan ekspansi shell untuk setiap perintah
- Uji fungsi set -x dengan menjalankan alias kota dan rm
- Eksekusi set +x untuk menghentikan tampilan ekspansi shell

Latihan (1)

- Hapus alias kota
- Dimana lokasi perintah cat dan passwd?
- Jelaskan perbedaan perintah berikut:
echo
/bin/echo
- Jelaskan perbedaan perintah berikut:
echo Hello
echo -n Hello
- Tampilkan A B C dengan dua spasi antara B dan C
- Bagaimana menampilkan output berikut tanpa menggunakan spasi?
4+4 = 8
10+14 = 24
- Gunakan echo untuk menampilkan:
??\
- Gunakan satu echo untuk menampilkan tiga kata pada 3 baris!

Operator Kendali

- ; → semicolon
- & → ampersand
- \$? → dolar tanda tanya
- && → ampersand ganda
- || → bar vertikal ganda
- Kombinasi && dan ||
- # → tanda pound
- \ → karakter khusus escape

; Titik-koma

- Lebih satu perintah pada satu baris dapat dipisahkan dengan titik-koma (;)
- Eksekusi terhadap perintah-perintah dilakukan secara urut
- Contoh:

```
echo Praktikum;echo Sistem Operasi; file hitung.txt
```

```
Praktikum
```

```
Sistem Operasi
```

```
hitung.txt: ASCII text
```

& Ampersand

- Digunakan untuk segera mendapatkan shell meskipun eksekusi belum selesai. Perintah akan dieksekusi di Background
- Contoh: menyalin banyak (lama waktunya) file di background dan shell digunakan untuk pekerjaan lain yang singkat-singkat.

\$? Dolar Tanda Tanya

- Parameter dari Shell
- Memegang status exit dari perintah yang dieksekusi sebelumnya
- Nilai 0 menunjukkan bahwa eksekusi berhasil dengan baik
- Contoh

```
touch satu.txt; rm satu.txt
```

```
rm: remove regular empty file `satu.txt'? y
```

```
echo $?
```

```
0
```

```
rm satu.txt
```

```
rm: cannot remove `satu.txt': No such file or directory
```

```
echo $?
```

```
1
```

&& Ampersand Ganda

- && dianggap sebagai logical AND. Digunakan diantara dua perintah.
- Perintah kedua dieksekusi hanya jika eksekusi perintah pertama sukses (status exit 0)
- Contoh

echo satu && echo dua

echor satu && echo dua ← error

cd Downloads/ && ls

cd Downloads/ && ls ← error

|| Bar Vertikal Ganda

- || dianggap sebagai logical OR.
- Perintah kedua dieksekusi hanya jika perintah pertama gagal (status exit tidak 0)
- Contoh

```
echo satu || echo dua; echo tiga
```

```
echor satu || echo dua; echo tiga ← error
```

```
cd Downloads/ || ls
```

```
cd Downloads/ || ls ← error
```

Kombinasi && dan ||

- Digunakan untuk membentuk struktur if-then-else
- Contoh: Jika eksekusi pertama (rm) berhasil maka jalankan perintah kedua (echo), jika gagal jalankan perintah ketiga (echo)

`touch filesatu.txt`

`rm filesatu.txt && echo Delete Berhasil || echo Delete Gagal`

rm: remove regular empty file `filesatu.txt'? y

Delete Berhasil

`rm filesatu.txt && echo Delete Berhasil || echo Delete Gagal`

rm: cannot remove `filesatu.txt': No such file or directory

Delete Gagal

Tanda Pound

- Apapun yang ditulis setelah tanda # diabaikan oleh Shell, dianggap sebagai komentar
- Contoh

`mkdir dirsatu #membuat direktori dirsatu`

`cd dirsatu #masuk ke direktori dirsatu`

\ Karakter Escape

- Digunakan untuk memasukkan karakter khusus yang tidak diterjemahkan oleh Shell, misalnya tanda ; & \$ #

- Contoh

echo titik \; koma

echo \& \\$ \# @

echo escaping \\? * \'"\'

- Apa perbedaan dua perintah ini?

echo *

echo *

\ pada akhir baris

- Jika diletakkan di akhir baris, Shell akan memberikan baris baru tetapi \ tidak diproses saat eksekusi

- Contoh

echo This command line \

is split in three \

parts

This command line is split in three parts

Latihan (2)

- Saat menuliskan passwd, file mana yang dieksekusi?
- Jenis file apakah itu?
- Jalankan perintah pwd dua kali
- Jalankan ls setelah cd /etc, tetapi hanya jika cd /etc tidak error
- Jalankan cd /etc setelah cd etc, hanya jika cd etc gagal
- Tampilkan pesan “Berhasil” ketika touch file123 berhasil. Jika tidak tampilkan “Gagal”. Coba ini dalam home directory dan /bin
- Jalankan perintah sleep 6, apa yang dilakukan perintah ini?
- Jalankan sleep 200 pada background
- Tulis perintah untuk menampilkan “Berhasil” jika penghapusan file123 berhasil. Jika tidak tampilkan “Gagal”
- Gunakan echo untuk menampilkan "Hello World with denganstrange' characters \ * [] ~ \ \ ."
(termasuk semua tanda petik)

Variabel

- Mengakses Variabel
- Variabel \$PS1 dan \$PATH
- Membuat Variabel
- Tanda Petik Ganda dan Tunggal
- Perintah set dan unset
-

Mengakses Variabel

- Nama variabel didahului oleh tanda \$
- Bersifat *case-sensitive*
- Contoh (variabel lingkungan Linux)

`echo $HOSTNAME`

`echo Komputer $HOSTNAME digunakan oleh $USER dengan home directory $HOME`

Variabel \$PS1

- Digunakan untuk menentukan tampilan prompt Shell
- Karakter escape \u mewakili username, \w mewakili working directory
- Contoh

PS1=prompt

PS1='Prompt >'

PS1='\u@\h\$ '

- **Homework** (harus login sebagai root)

Bagaimana agar username untuk pengguna biasa berwarna **HIJAU**, sedangkan untuk root berwarna **MERAH**?

Variabel \$PATH

- Menunjukkan lokasi dimana file-file program yang dapat dieksekusi berada

- Contoh

```
echo $PATH
```

- Menambah PATH

```
PATH=$PATH:..
```

← . mewakili directory aktif

Membuat Variabel

- Membuat variabel tidak melibatkan \$
- Mengakses variabel harus didahului \$
- Contoh

```
var1="Test 1 2 3"
```

```
echo $var1
```

```
Test 1 2 3
```

Tanda Petik

- Perhatikan perbedaan yang diberikan oleh petik tunggal (') dan ganda (")!

```
saldo=1250000
```

```
echo $saldo
```

```
echo "$saldo"
```

```
echo '$saldo'
```

```
echo "Saldo bulan ini: Rp. $saldo"
```

```
echo 'Saldo bulan ini: Rp. $saldo'
```

Perintah set dan unset

- Perintah set menampilkan daftar variabel aktif
- Perintah unset digunakan untuk meniadakan suatu variabel
- Contoh

```
set | more
```

```
var1=123
```

```
echo $var1
```

```
unset var1
```

```
echo $var1
```

Homework

- Apa beda dua baris berikut?
`echo `var1=5;echo $var1``
`echo 'var1=5;echo $var1'`
- Apa hasil eksekusi baris berikut?
`echo `cd /etc; ls -d * | grep pass``
- Apa manfaat dari perintah `set -u`?
`set -u; echo $Myvar`
`set +u; echo $Myvar`

Homework

- Apakah baris berikut mencetak **Halo Superman dan Supergirl**? Jika TIDAK, betulkan!

prefix=Super; echo Halo \$prefixman dan \$prefixgirl

- Apa perbedaan antara perintah env dan set?
- Apa manfaat perintah export?
- Tampilkan username yang anda gunakan!
- Salin username tersebut ke variabel \$pengguna
- Buat dua variabel bernilai 'Dumb' dan 'do'. Gunakan echo untuk mencetak teks 'Dumbledore'
- Tambahkan waktu (time) ke Prompt PS1