

Praktikum Sistem Operasi
(Pertemuan 04)

**History, Pembangunan
Nama File dan Redireksi &
Pipe**

Husni
husni@mail.ugm.ac.id

Garis Besar

- Sejarah Shell
- Pembangunan Nama File
- Redireksi dan Pipe

Sejarah Shell

- Mengulang Perintah
- Sejarah
- !n
- Ctrl r
- \$HISTSIZE, \$HISTFILE, \$HISTFILESIZE
- Ekspresi Regular

Pengulang Perintah Terakhir

- Perintah Terakhir, gunakan !! (baca: bang-bang)
- Contoh

```
tail -2 /etc/passwd
```

```
husni:x:1000:1000:husni,,,:/home/husni:/bin/bash
```

```
mysql:x:115:126:MySQL Server,,,:/nonexistent:/bin/false
```

```
!!
```

```
tail -2 /etc/passwd
```

```
husni:x:1000:1000:husni,,,:/home/husni:/bin/bash
```

```
mysql:x:115:126:MySQL Server,,,:/nonexistent:/bin/false
```

Mengulang Perintah Lainnya

- Gunakan satu bang diikuti satu atau lebih karakter yang mengawali perintah yang pernah dijalankan sebelumnya
- Contoh



```
echo ini baris pertama > test.txt
```

```
cat test.txt
```

```
ini baris pertama
```

```
echo ini baris kedua >> test.txt
```

```
!c
```

```
cat test.txt
```

```
ini baris pertama
```

```
ini baris kedua
```

```
echo ini baris ketiga >> test.txt
```

```
!c
```

```
cat test.txt
```

```
ini baris pertama
```

```
ini baris kedua
```

```
ini baris ketiga
```

Melihat Isi Sejarah (*History*)

- Perintah **history** digunakan untuk melihat semua perintah yang pernah dijalankan
- Perintah **history n** digunakan untuk melihat **n** perintah terakhir yang dijalankan

- Contoh  **history 10**

```
217 echo ini baris kedua >> test.txt
218 cat test.txt
219 touch test.txt
220 cat test.txt
221 echo ini baris ketiga >> test.txt
222 cat test.txt
223 echo ini baris ketiga > test2.txt
224 cat test2.txt
225 history
226 history 10
```

!n

- **!n** digunakan untuk memerintahkan Shell menampikan history nomor n dan kemudian mengeksekusinya

- Contoh

!222

cat test.txt

ini baris pertama

ini baris kedua

ini baris ketiga

Ctrl r

- Kombinasi Tombol ini digunakan untuk mencari perintah tertentu yang telah ada di dalam history.
- Contoh: Mencari perintah berbunyi 'get' yang telah dijalankan sebelumnya

Tekan Ctrl r dan masukkan kata kunci **get**

(reverse-i-search)`get': sudo apt-get install python-orange

\$HISTSIZE

- Variabel ini berisi jumlah perintah yang dapat disimpan oleh history. Beberapa distro Linux memberikan default 500 atau 1000.
- Variabel ini juga digunakan untuk mengubah jumlah tersebut
- Contoh

```
echo $HISTSIZE
```

```
1000
```

```
HISTSIZE=12350
```

```
!e
```

```
echo $HISTSIZE
```

```
12350
```

```
HISTSIZE=1000
```

```
echo $HISTSIZE
```

```
1000
```

\$HISTFILE

- Variabel ini menunjukkan file yang menyimpan history. Shell bash menyimpan history dalam file `~/.bash_history`
- Contoh
`echo $HISTFILE`
`/home/husni/.bash_history`
- History akan disimpan ke file tersebut setelah keluar sesi dengan `exit`
- Jika keluar terminal gnome dengan mouse atau `reboot`, history tidak disimpan ke dalam file tersebut

\$HISTFILESIZE

- Variabel ini menunjukkan jumlah perintah yang dapat disimpan dalam file history. Beberapa distro memberikan nilai default 2000
- Contoh

```
echo $HISTFILESIZE
```

```
2000
```

Ekspresi Reguler

- Ekspresi reguler (Regex) dapat digunakan bersama dengan bang (!)
- Contoh: Mengganti 2 dengan 7 memanfaatkan Regex

```
echo Test 1 2 3 > test.txt
```

```
cat test.txt
```

```
Test 1 2 3
```

```
!e:s/2/7
```

```
echo Test 1 7 3 > test.txt
```

```
!c
```

```
cat test.txt
```

```
Test 1 7 3
```

```
touch file1.txt; touch file2.txt
```

```
echo Halo000 >> file1.txt
```

```
cat file1.txt
```

```
Halo000
```

```
cat file2.txt
```

```
!e:s/1/2
```

```
echo Halo000 >> file2.txt
```

```
!c
```

```
cat file2.txt
```

```
Halo000
```

Korn Shell (ksh)

- Perintah history digunakan untuk menampilkan history

history 10 → menampilkan history mulai nomor 10

- Huruf r digunakan untuk mengerjakan history tertentu

r e → mengeksekusi perintah terakhir dalam history yang berawalan e

r → mengeksekusi perintah terakhir dalam history

Latihan

1. Jalankan perintah `echo The answer to the meaning of life, the universe and everything is 42.`
2. Ulangi perintah sebelumnya menggunakan hanya dua karakter (ada 2 cara!)
3. Tampilkan lima perintah terakhir yang ada dalam history.
4. Hasilkan seperti pertanyaan pertama tetapi memanfaatkan nomor baris yang dihasilkan pertanyaan ketiga
5. Berapa banyak perintah yang dapat berada di memory untuk sesi shell sekarang?
6. Dimana perintah-perintah ini disimpan ketika keluar Shell?
7. Berapa banyak perintah yang dapat dituliskan ke file history saat anda keluar dari sesi Shell sekarang?
8. Pastikan Shell bash anda nanti akan mampu mengingat 5000 perintah yang dijalankannya
9. Bukan *console* baru (Ctrl Shift t) dengan akun yang sama. Kapan history perintah dituliskan ke file history?

Pembangkitan Nama File

- * → Asterisk
- ? → Tanda Tanya
- [] → kurung siku
- Range a-z dan 0-9
- \$LANG dan kurung siku
- Pencegahan
- Latihan

* Asterisk

- * dianggap cocok dengan rangkaian (satu atau lebih) karakter. Digunakan untuk membangkitkan nama file.
- Jika tidak diberikan *path*, maka dianggap direktori aktif
- Manual: man glob(7)
- Contoh

```
ls te*
```

```
test2.txt test.txt
```

```
ls *.txt
```

```
file1.txt file2.txt hitung.txt test2.txt test.txt
```

```
ls *e*.txt
```

```
file1.txt file2.txt test2.txt test.txt
```


? Tanda Tanya

- Digunakan untuk membangkitkan nama file. ? dianggap tepat satu karakter.

- Contoh

Is test*

test2.txt test3.txt test6.odp test.txt

test3.docx test5.doc test_masuk.docx

Is test?.*

test2.txt test3.docx test3.txt test5.doc test6.odp

Is test?.doc

test5.doc

Is test?.doc*

test3.docx test5.doc

Is tes?? .txt

test2.txt test3.txt

[] Kurung Siku

- Mencocokkan apapun yang ada dalam kurung siku tanpa melihat urutannya. Digunakan untuk membangkitkan nama file.
- Pemberian tanda ! sebelum karakter menunjukkan NEGASI.
- Contoh

test[35]*

test3.docx test3.txt test5.doc

ls test[3]*

test3.docx test3.txt

ls test[56]*[d]*

test5.doc test6.odp

ls test*[dp]*

test3.docx test5.doc test6.odp test_masuk.docx

ls test[!5]*

test2.txt test3.docx test3.txt test6.odp test_masuk.docx test.txt

Range a-z dan 0-9

- Range karakter dalam [] dapat digunakan
- Contoh

ls test[0-9]*

test2.txt test3.docx test3.txt test5.doc test6.odp

ls te[a-z]*

terserah.docx test3.docx test5.doc test_masuk.docx
test2.txt test3.txt test6.odp test.txt

ls te[a-z][a-z][!0-9]*

terserah.docx test_masuk.docx test.txt

ls test[0-9]?[d]*

test3.docx test5.doc

ls test[0-9]?[!d]*

test2.txt test3.txt test6.odp

\$LANG dan Kurung Siku

- Beberapa bahasa (\$LANG) tidak *case-sensitive*, misal en_US.UTF-8

```
echo $LANG
```

```
en_US.UTF-8
```

```
ls [a-z]est[0-9]*
```

```
test2.txt test3.txt test6.odp Test7.txt
```

```
test3.docx test5.doc Test7.odp Test8.doc
```

```
ls [A-Z]est[0-9]*
```

```
test2.txt test3.txt test6.odp Test7.txt
```

```
test3.docx test5.doc Test7.odp Test8.doc
```

```
LANG=C
```

```
echo $LANG
```

```
C
```

```
ls [A-Z]est[0-9]*
```

```
Test7.odp Test7.txt Test8.doc
```

Pencegahan...

- Apa yang dihasilkan baris berikut?

```
echo *
```

```
echo '*'
```

```
echo \*
```

```
echo "*"
```

```
mkdir dir1; cd dir1; echo *
```

```
touch file1.txt; touch file2.txt
```

```
echo *
```

- Apa yang dilakukan untuk mencegah pembuatan nama file secara otomatis?

Latihan

- Buatlah direktori bernama TestDir dan masuk ke dalamnya.
- Buat file bernama berkas1, berkas10, berkas11, berkas2, Berkas3, Berkas3, berkasAB, berkasa, berkasAAA, berkas(, berkas 2
- Tampilkan semua file berawalan berkas
- Tampilkan semua file berawalan Berkas
- Tampilkan semua file yang dimulai dengan berkas dan diakhiri bilangan
- Tampilkan semua file yang dimulai dengan berkas dan diakhiri huruf
- Tampilkan semua file yang diawali dengan Berkas dan membuat satu digit sebagai karakter kelima
- Tampilkan semua file yang dimulai dengan Berkas dan mempunyai suatu digit sebagai karakter kelima dan tidak lain ada lainnya
- Tampilkan semua file yang dimulai huruf dan diakhir bilangan

Latihan

- Tampilkan semua file yang panjangnya tepat tujuh karakter.
- Tampilkan semua file yang berawalan B atau b dan berakhiran 3 atau A
- Tampilkan semua file yang berawalan dengan b yang mempunyai e atau R sebagai karakter kedua dan diakhiri bilangan
- Tampilkan semua file yang tidak dimulai huruf F
- Salin nilai dari variabel \$LANG ke \$MyLANG.
- Perhatikan pengaruh dari \$LANG dalam menampilkan range A-Z atau a-z.
- Jika server anda dihack seseorang dan menghilangkan perintah ls. Apakah anda menggunakan perintah lain yang berfungsi serupa ls? Bagaimana?
- Adakah perintah selain cd yang dapat digunakan untuk berpindah direktori?

Redireksi dan Pipe

- stdin, stdout dan stderr
- Redireksi Output
- Redireksi Error
- Redireksi Input
- Redireksi yang Membingungkan
- Pembersihan File Cepat
- swapping stdout dan stderr
- Pipe
- Latihan

stdin, stdout dan stderr

- Shell mengambil input dari stdin (stream 0) dan mengirimkan output ke stdout (stream 1) dan pesan error ke stderr (stream 2)
- Keyboard berfungsi sebagai stdin. Display (monitor) berfungsi sebagai stdout dan stderr

Redireksi Output

- Stdout
- Standarnya adalah Display. Diredirect dengan tanda lebih besar > atau 1>
- Contoh

```
echo Belajar Shell Linux
```

```
Belajar Shell Linux
```

```
echo Belajar Shell Linux 1> belajar.txt
```

```
cat belajar.txt
```

```
Belajar Shell Linux
```

Akibat dari `>` atau `1>`

- Isi file yang dijadikan tujuan dibersihkan, walaupun redirect GAGAL

```
echp Belajar Shell Linux 1> belajar.txt
```

```
cat belajar.txt
```

- Dapat dicegah dengan opsi `noclobber`

```
set -o noclobber
```

```
echo Belajar Shell Linux 1> belajar.txt
```

```
bash: belajar.txt: cannot overwrite existing file
```

```
set +o noclobber
```

```
echp Belajar Shell Linux 1> belajar.txt
```

```
cat belajar.txt
```

Menembus noclobber

- Opsi `set -o noclobber` dapat ditembus dengan *redirect* berbentuk `>|`
- Contoh

```
set -o noclobber
```

```
echo Belajar Shell Linux - Pertemuan 04 1>  
belajar.txt
```

```
bash: belajar.txt: cannot overwrite existing file
```

```
echo Belajar Shell Linux - Pertemuan 04 1>|  
belajar.txt
```

```
cat belajar.txt
```

```
Belajar Shell Linux - Pertemuan 04
```

Beda antara > dan >>

- > digunakan untuk membuat file baru dengan isi baru
- >> digunakan untuk menambahkan isi baru ke dalam file yang telah ada
- Contoh

echo ini membuat file > file101.txt

cat file101.txt

ini membuat file

echo ini baris baru dalam file101 >> file101.txt

cat file101.txt

ini membuat file

ini baris baru dalam file101

echo ini baris terakhir dalam file101 >> file101.txt

cat file101.txt

ini membuat file

ini baris baru dalam file101

ini baris terakhir dalam file101

Redireksi Error

- 2> dapat digunakan untuk mengalihkan pesan error ke suatu file dan menghilangkan tampilan tersebut di layar

- Contoh

```
find / > semuafile.txt 2> /dev/null
```

```
cat semuafile.txt
```

- 2>&1 digunakan untuk me-redirect stdout ke stderr ke file yang sama

- Contoh

```
find / > semuafile_plus_error.txt 2>&1
```

Redireksi Input

- Paling simpel adalah menggunakan `<` atau `0<`
- Contoh
- `cat < file101.txt`
- ini membuat file
- ini baris baru dalam file101
- ini baris terakhir dalam file101
-

<< dan <<<

- << dapat digunakan untuk meredirect input sampai ditemukan deretan khusus, misalnya EOF

```
cat <<EOF > file102.txt
```

```
> satu
```

```
> EOF
```

```
cat file102.txt
```

```
cat <<stop > file103.txt
```

```
> satu nol
```

```
> tiga
```

```
> stop
```

- <<< digunakan untuk meredirect string langsung ke suatu perintah

```
base64 <<< husni@mail.ugm.ac.id
```

```
aHVzbmlAbWFpbC51Z20uYWMMuaWQK
```

```
base64 -d <<< aHVzbmlAbWFpbC51Z20uYWMMuaWQK
```

```
husni@mail.ugm.ac.id
```


Kejelasan Redireksi

- Apa maksud 3 baris berikut?

cat winter.txt > snow.txt 2> errors.txt

2> errors.txt cat winter.txt > snow.txt

< winter.txt > snow.txt 2> errors.txt cat

Pipe

- Pipe (|) mengambil stdout dari perintah sebelumnya dan mengirimnya sebagai stdin untuk perintah berikutnya
- Contoh
 - `ls /etc > etcfiles.txt`
 - `tail -4 etcfiles.txt`
- Dapat disingkat menjadi
 - `ls /etc | tail -4`
- Bar vertikal atau pipe diletakkan diantara dua perintah yang akan dieksekusi secara simultan
- `ls /etc | tail -4 | tac`

Latihan

- Gunakan ls untuk meng-output-kan isi dari direktori /etc/ ke file etc.txt.
- Aktifkan opsi Shell noclobber
- Pastikan noclobber aktif dengan mengulang ls terhadap /etc/.
- Non-aktifkan opsi noclobber
- Buka dua console atau shell sekaligus. Buat file kosong bernama tailing.txt. Ketik tail -f tailing.txt. Gunakan Shell kedua untuk menambahkan sebaris teks ke file tersebut. Pada Shell pertama, periksa hasilnya.
- Buat file yang mengandung nama lima orang. Gunakan cat dan redireksi output untuk membuat file tersebut dan gunakan << untuk menandai akhir input