

MODULE 12: WORKING WITH TEXT

POKOK BAHASAN:

- Prinsip kerja redirection teks dan file
- Manipulasi standard input, output, dan error di shell Filter output atau file di shell menggunakan regular expression (regex)
- Menampilkan output file dengan paging atau berdasarkan pilihan

OBJEKTIF:

Mahasiswa mampu memahami prinsip kerja redirection

Mahasiswa mampu mengimplementasikan perintah-perintah yang berkaitan dengan standard input, output, dan error

Mahasiswa mampu mengimplementasikan perintah-perintah untuk melakukan filterisasi output atau file menggunakan regex

Mahasiswa mampu menggunakan perintah-perintah untuk menampilkan output sesuai kebutuhan berbasis redirection dan regex

TUGAS PRAKTIKUM:

1. Mengerjakan setiap langkah yang ada di modul praktikum dengan bukti print screen dari output yang didapat dan harus melalui shell dengan nama Anda.
2. Menjawab seluruh soal praktikum
3. Mengumpulkan hasil praktikum di learning pada link yang telah disediakan dengan format file pdf, berikan nama file dengan nama Anda
4. **TIDAK ADA PERPANJANGAN WAKTU PENGUMPULAN**

Jangan lupa copy-kan file vdi hasil instalasi Linux Anda ke Flashdisk Anda agar Anda dapat menggunakan kembali hasil instalasi tersebut di komputer manapun cukup dengan menginstal kembali dari file vdi ini!!

PENJELASAN:

Secara normal umumnya ketika kita menuliskan perintah di shell kemudian dieksekusi maka akan tampil output di layar, (output ini disebut juga dengan istilah channel) output yang tampil di layar

disebut dengan **standard output**, yang dilambangkan dengan istilah/tulisan **stdout**. Nomor **deskriptor file** untuk channel ini adalah **1**.

Standard error terjadi ketika ada sebuah perintah(command) yang salah pada saat dieksekusi, yang dilambangkan dengan istilah/tulisan **stderr**. Nomor **deskriptor file** untuk channel ini adalah **2** dan pesan ini juga akan dikirim ke layar output secara default.

Standard input biasanya pemberian input (perintah/command) di shell oleh user melalui keyboard, namun dengan redirecting (mengarahkan) standard input maka file juga dapat digunakan sebagai standard input. Standard input dilambangkan dengan istilah/tulisan **stdin**. Nomor **deskriptor file** untuk channel ini adalah **0**.

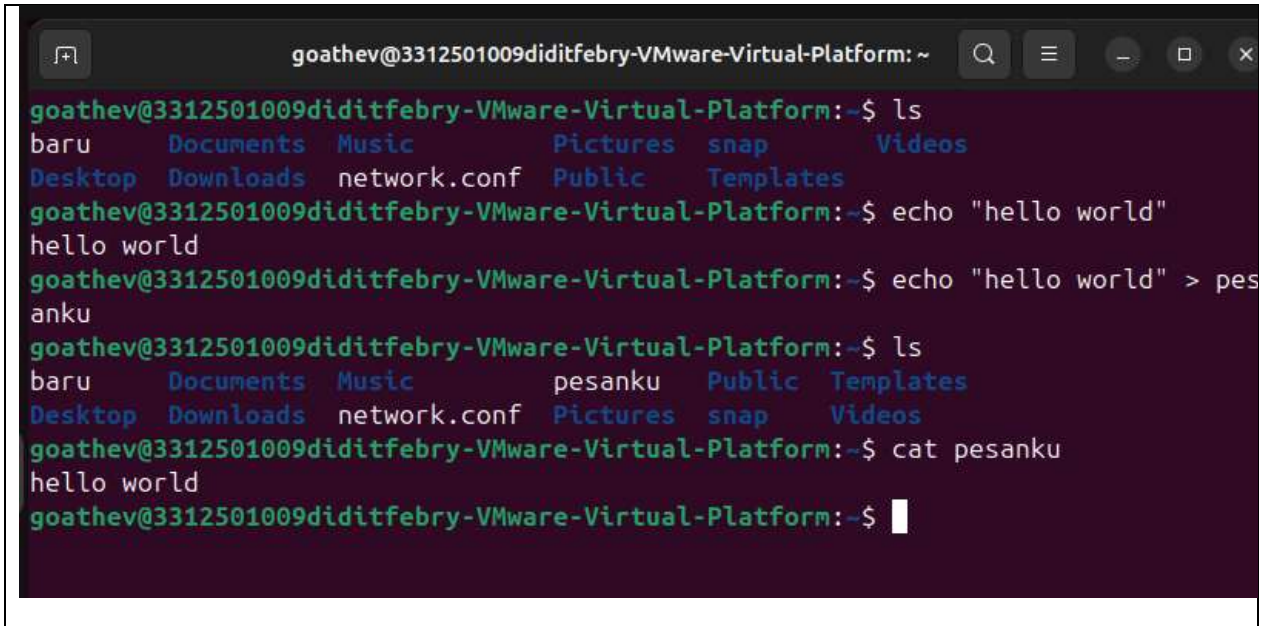
KEGIATAN PRAKTIKUM:

LANGKAH 1:

Menggunakan simbol redirection (pengalihan) **>** yang tujuannya untuk mengarahkan output normal yang seharusnya tampil di layar menjadi diarahkan ke sebuah file. Jalankan perintah berikut ini di shell:

1. **ls** ☐ **enter**
2. **echo "Hello World"** ☐ **enter**
3. **echo "Hello World" > pesanku** ☐ **enter**
4. **ls** ☐ **enter**
5. **cat pesanku** ☐ **enter**

screenshot hasilnya dan jelaskan informasi apa yang didapat dari setiap urutan perintahnya!



```
goathev@3312501009diditfebry-VMware-Virtual-Platform: ~  
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ ls  
baru      Documents Music      Pictures snap      Videos  
Desktop  Downloads network.conf Public  Templates  
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ echo "hello world"  
hello world  
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ echo "hello world" > pesanku  
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ ls  
baru      Documents Music      pesanku Public  Templates  
Desktop  Downloads network.conf Pictures snap      Videos  
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ cat pesanku  
hello world  
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$
```

ls menampilkan daftar file, echo "Hello World" hanya mencetak teks ke layar, echo "Hello World" > pesanku membuat file *pesanku* berisi teks tersebut, ls berikutnya menunjukkan file itu sudah muncul, dan cat *pesanku* menampilkan isi file untuk memastikan isinya benar.

LANGKAH 2:

Perintah **find** adalah contoh perintah yang bagus untuk menunjukan kerja dari **stderr**. Find adalah perintah yang digunakan untuk mencari filesystem yang didasarkan pada kriteria pencarian yang diberikan seperti nama file.

Sekarang ketikkan perintah berikut ini di shell: **find /etc -name hosts** ☐ **enter**

screenshot hasilnya dan jelaskan informasi apa yang didapat dan mengapa demikian!

```
goathev@3312501009diditfebry-VMware-Virtual-Platform: ~  
/etc/alternatives/ebtables-restore  
/etc/alternatives/ex.1.gz  
/etc/alternatives/ex  
/etc/alternatives/lzcmp.1.gz  
/etc/alternatives/view.ja.1.gz  
/etc/alternatives/awk.1.gz  
/etc/alternatives/which  
/etc/alternatives/telnet  
/etc/alternatives/vtrgb  
/etc/alternatives/lzegrep  
/etc/alternatives/infobrowser.1.gz  
/etc/alternatives/view.it.1.gz  
/etc/alternatives/pico  
/etc/alternatives/cpp  
/etc/alternatives/nawk  
/etc/alternatives/ftp  
/etc/alternatives/rmt  
/etc/alternatives/netcat  
/etc/alternatives/arptables-save  
/etc/alternatives/lzma.1.gz  
/etc/rygel.conf  
find: '-name': No such file or directory  
find: 'hosts': No such file or directory  
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$
```

Perintah `find /etc -name hosts` mencari file bernama **hosts** di seluruh folder `/etc` dan menampilkan lokasinya jika ditemukan.

LANGKAH 3:

Pesan output yang terjadi baik output normal, error, maupun gabungan dari keduanya dapat dialihkan (redirect) ke sebuah file tertentu. Sekarang ketikkan perintah berikut ini di shell:

1. **find /etc -name hosts > output_normal.txt** ☐ enter
2. **ls** ☐ enter
3. **cat output_normal.txt** ☐ enter
4. **find /etc -name hosts 2> output_error.txt** ☐ enter
5. **ls** ☐ enter
6. **cat output_error.txt** ☐ enter

7. **find /etc -name hosts > output_bersama.txt 2>& 1** ☐ enter
8. **ls** ☐ enter
9. **cat output_bersama.txt** ☐ enter

screenshot hasilnya dan jelaskan informasi apa yang didapat dari setiap urutan perintahnya dan mengapa demikian!

```
find: '/etc/polkit-1/rules.d': Permission denied
find: '/etc/cups/ssl': Permission denied
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ ls
baru      Downloads  output_normal.txt  Public  Videos
Desktop   Music      pesanku            snap
Documents network.conf Pictures            Templates
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ cat output_normal.txt
/etc/avahi/hosts
/etc/hosts
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ find /etc -name hosts 2> output_error.txt
/etc/avahi/hosts
/etc/hosts
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ ls
baru      Downloads  output_error.txt  Pictures  Templates
Desktop   Music      output_normal.txt Public     Videos
Documents network.conf pesanku           snap
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ cat output_error.txt
find: '/etc/sss': Permission denied
find: '/etc/credstore.encrypted': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/credstore': Permission denied
find: '/etc/polkit-1/rules.d': Permission denied
find: '/etc/cups/ssl': Permission denied
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ find /etc -name hosts > output_bersama.txt 2>& 1
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ ls
baru      Downloads  output_bersama.txt  pesanku  snap
Desktop   Music      output_error.txt    Pictures  Templates
Documents network.conf output_normal.txt   Public    Videos
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ cat output_bersama.txt
find: '/etc/sss': Permission denied
find: '/etc/credstore.encrypted': Permission denied
find: '/etc/ssl/private': Permission denied
/etc/avahi/hosts
find: '/etc/credstore': Permission denied
find: '/etc/polkit-1/rules.d': Permission denied
find: '/etc/cups/ssl': Permission denied
/etc/hosts
```

Urutan perintah ini menunjukkan bahwa output normal bisa diarahkan ke file, error bisa dipisahkan ke file lain, dan keduanya bisa digabung ke satu file. File *output_normal.txt* berisi hasil pencarian, *output_error.txt* berisi pesan error, dan *output_bersama.txt* berisi keduanya karena diarahkan dengan `2>&1`.

LANGKAH 4:

Melakukan pengalihan *stdin* yang di simbolkan dengan `<` dapat juga dilakukan dari sebuah file, gunakan file *pesanku* seperti yang telah Anda buat sebagai input file nya.

Sekarang ketikkan perintah berikut ini di shell: **tr a-z A-Z < pesanku** ☐ **enter**
(Anda dapat membaca simbol < dengan cara dikirim dari ...)

screenshot hasilnya dan jelaskan informasi apa yang didapat dari isi file **pesanku** dan mengapa demikian!

LANGKAH 5:

Jalankan perintah berikut ini di shell:

1. **cat /etc/passwd** ☐ **enter**
2. **cut -d: -f1 /etc/passwd** ☐ **enter**
3. **cut -d: -f1 /etc/passwd | sort** ☐ **enter**

screenshot hasilnya dan jelaskan informasi apa yang didapat dari setiap urutan perintahnya!

```
goathev@3312501009diditfebry-VMware-Virtual-Platform:~$ cut -d: -f1 /etc/passwd | sort
_apt
avahi
backup
bin
colord
cups-browsed
cups-pk-helper
daemon
dhcpcd
dnsmasq
fwupd-refresh
games
gdm
geoclue
gnome-initial-setup
gnome-remote-desktop
goathev
hplip
irc
kernoops
list
lp
mail
man
messagebus
news
nm-openvpn
nobody
polkitd
proxy
root
rtkit
saned
```

`cat /etc/passwd` menampilkan seluruh data akun sistem, `cut -d: -f1 /etc/passwd` mengekstrak hanya daftar username, dan menambahkan `sort` mengurutkan username tersebut secara alfabetis.

LANGKAH 6:

Perintah `find` adalah perintah yang sangat fleksibel dengan sejumlah opsi yang memungkinkan pengguna menemukan file berdasarkan nama file, ukuran, tanggal, jenis, dan hak akses.

Format perintah `find` adalah:

`find <lokasi pencarian> -opsi <argumen>`

Jalankan perintah berikut ini di shell:

1. **find ~ -name "*bash*" □ enter**
2. **find ~ -mmin -10 □ enter**
3. **find ~ -size -1M □ enter**
4. **find ~ -type d □ enter**
5. **find ~ -type f □ enter**

screenshot hasilnya dan jelaskan informasi apa yang didapat dari setiap urutan perintahnya!


```
goathev@3312501009diditfebry-VMware-Virtual-Platform: ~  
2Ftracker%23Video.db  
/home/goathev/.cache/tracker3/files/meta.db  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Software.db-wal  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Software.db-shm  
/home/goathev/.cache/tracker3/files/meta.db-wal  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Pictures.db-shm  
/home/goathev/.cache/tracker3/files/ontologies.gvdb  
/home/goathev/.cache/tracker3/files/meta.db-shm  
/home/goathev/.cache/tracker3/files/first-index.txt  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Software.db  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Video.db-wal  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Documents.db  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Pictures.db-wal  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Audio.db-wal  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Pictures.db  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Documents.db-shm  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23Video.db-shm  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23FileSystem.db  
/home/goathev/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology  
2Ftracker%23FileSystem.db-wal  
/home/goathev/.cache/event-sound-cache.tdb.f0a2a093a679465c92a1e2b96e45b8c8.x86_  
linux-gnu  
/home/goathev/.cache/update-manager-core/meta-release-lts  
/home/goathev/.cache/gstreamer-1.0/registry.x86_64.bin  
/home/goathev/output_normal.txt
```

Perintah `find ~ -name "*bash"` menampilkan semua file atau folder di home yang namanya mengandung "bash". `find ~ -mmin -10` menunjukkan file yang dimodifikasi dalam sepuluh menit terakhir. `find ~ -size -1M` menampilkan file berukuran kurang dari satu megabita. `find ~ -type d` menampilkan semua direktori di home, sedangkan `find ~ -type f` menampilkan semua file biasa di home.

PENGANTAR GREP:

Selanjutnya Anda akan diajak untuk menggunakan perintah dari keluarga **grep** dengan regular expression (regex) untuk mencari/mencocokkan string dari karakter tertentu dalam aliran data misalnya file teks.

Grep sendiri merupakan satu perintah di Linux yang cukup populer, grep sendiri adalah singkatan dari **g**lobal **r**egular **e**xpression **p**rint sehingga dapat kita artikan bahwa perintah grep adalah perintah untuk memproses teks baris demi baris dan mencetak setiap baris yang cocok dengan pola tertentu.

Variasi perintah dalam keluarga grep diantaranya: **grep**, **egrep**, **fgrep**, dan **rgrep**. Penjelasan dari masing-masing fungsi perintah tersebut seperti terlihat pada tabel 12.1.

Tabel 12.1 Variasi perintah grep

grep (biasa)	: Perintah grep dan tanpa karakter khusus ketika melakukan pencocokan stringnya
egrep	: Perintah egrep atau dapat menggunakan perintah grep dengan opsi -E dan membutuhkan karakter khusus ketika melakukan pencocokan stringnya seperti: ? , + ,
fgrep	: Perintah fgrep atau dapat menggunakan perintah grep dengan opsi -F adalah perintah untuk melakukan pencocokan string yang mengandung karakter-karakter khusus seperti \$, ^ , . , atau yang lainnya
rgrep	: Perintah rgrep atau dapat menggunakan perintah grep dengan opsi -r untuk melakukan pencocokan string secara rekursif mulai dari awal direktori yang ditentukan

Variasi kategori bentuk grep diantaranya: **grep**, **basic regular expression (BRE)**, **extended regular expression (ERE)**, dan **perl regular expression (PCRE)** untuk PCRE sendiri tidak masuk dalam pembahasan. Penjelasan mengenai perbedaan dari masing-masing kategori seperti terlihat pada tabel 12.2.


Tabel 12.2 Variasi kategori grep

grep (biasa)	: penggunaannya tanpa memerlukan karakter khusus
BRE	: penggunaannya memerlukan karakter khusus seperti: . , [] , ^ , * , \$
ERE	: penggunaannya memerlukan karakter khusus seperti: ? , + , , {}
PCRE	: penggunaannya memerlukan karakter khusus dan lebih banyak dari yang dimiliki BRE dan ERE

LANGKAH 7:

Penggunaan perintah **grep** dalam bentuk yang sederhana untuk mencocokkan rangkaian karakter tertentu. Misal kita ingin mencari kata **mail** di file **/etc/passwd**, maka perintahnya adalah: **grep mail**

/etc/passwd ☐ **enter** output seperti terlihat pada gambar 12.1



```
agusf123@agusf123-VirtualBox:~$ grep mail /etc/passwd
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

Gambar 12.1 Pencocokan Kata Mail Menggunakan Perintah Grep

Jalankan perintah berikut ini di shell:

1. **grep ^mail /etc/passwd** ☐ **enter**
2. **grep sh\$ /etc/passwd** ☐ **enter**
3. **grep .h /etc/passwd** ☐ **enter**
4. **grep ..h /etc/passwd** ☐ **enter**
5. **grep -E 'mail|sys' /etc/passwd** ☐ **enter**
6. **grep -E 'sy (s|n)' /etc/passwd** ☐ **enter**
7. **grep '[0-9]' /etc/passwd** ☐ **enter**
8. **grep -E '[0-9]{3}' /etc/passwd** ☐ **enter**

screenshot hasilnya dan jelaskan informasi apa yang didapat dari setiap urutan perintahnya serta sebutkan mana perintah grep yang masuk kategori **grep**, **BRE**, atau **ERE**!


```
goathev@3312501009diditfebry-VMware-Virtual-Platform: ~  
sync:x:4:65534:sync:/bin:/bin/sync  
_apt:x:42:65534:/:nonexistent:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
systemd-network:x:998:998:systemd Network Management:/:usr/sbin/nologin  
systemd-timesync:x:996:996:systemd Time Synchronization:/:usr/sbin/nologin  
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false  
messagebus:x:101:101:/:nonexistent:/usr/sbin/nologin  
syslog:x:102:102:/:nonexistent:/usr/sbin/nologin  
systemd-resolve:x:991:991:systemd Resolver:/:usr/sbin/nologin  
uidd:x:103:103:/:run/uidd:/usr/sbin/nologin  
usbmux:x:104:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin  
tss:x:105:105:TPM software stack,,,:/var/lib/tpm:/bin/false  
systemd-oom:x:990:990:systemd Userspace OOM Killer:/:usr/sbin/nologin  
kernoops:x:106:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin  
whoopsie:x:107:109:/:nonexistent:/bin/false  
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin  
avahi:x:108:111:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin  
tcpdump:x:109:112:/:nonexistent:/usr/sbin/nologin  
sssd:x:110:113:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin  
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false  
cups-pk-helper:x:112:114:user for cups-pk-helper service,,,:/nonexistent:/usr/sbin/nolo  
gin  
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin  
saned:x:113:116:/:var/lib/saned:/usr/sbin/nologin  
geoclue:x:114:117:/:var/lib/geoclue:/usr/sbin/nologin  
cups-browsed:x:115:114:/:nonexistent:/usr/sbin/nologin  
hplip:x:116:7:HPLIP system user,,,:/run/hplip:/bin/false  
gnome-remote-desktop:x:988:988:GNOME Remote Desktop:/var/lib/gnome-remote-desktop:/usr/  
sbin/nologin  
polkitd:x:987:987>User for polkitd:/:usr/sbin/nologin  
rtkit:x:117:119:RealtimeKit,,,:/proc:/usr/sbin/nologin  
colord:x:118:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin  
gnome-initial-setup:x:119:65534:/:run/gnome-initial-setup:/bin/false  
gdm:x:120:121:Gnome Display Manager:/var/lib/gdm3:/bin/false  
nm-openvpn:x:121:122:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologi  
n  
goathev:x:1000:1000:Goathev:/home/goathev:/bin/bash
```

erintah `grep ^mail /etc/passwd` menampilkan baris yang dimulai dengan *mail* dan termasuk BRE. `grep sh$ /etc/passwd` mencari baris yang berakhir dengan *sh*, juga BRE. `grep .h` dan `grep ..h` menampilkan baris yang mengandung pola satu karakter lalu *h*, atau dua karakter lalu *h*; keduanya BRE. Perintah `grep -E 'mail|sys'` mencocokkan *mail* atau *sys*, sedangkan `grep -E 'sy(s|n)'` mencocokkan pola dengan alternation; keduanya ERE. `grep '[0-9]'` menampilkan baris yang mengandung digit dan termasuk BRE, sedangkan `grep -E '[0-9]{3}'` mencari tiga digit berurutan dan termasuk ERE.

LANGKAH 8:

Xargs adalah salah satu perintah di Linux yang fungsinya membantu kumpulan perintah pipeline yang kompleks menjadi lebih efisien dalam pengeksekusiannya. **Xargs** sendiri singkatan dari **extended arguments**, artinya di sini dapat menggunakan banyak argumen sekaligus dalam pemberian perintahnya dan mencegah eksekusi perintah untuk setiap argumen yang diberikan.

Ikuti langkah-langkah ini:

CARA I

1. **mkdir DirBaru** ☐ **enter**
2. **cd DirBaru** ☐ **enter**
3. **mkdir Satu** ☐ **enter**
4. **mkdir Dua** ☐ **enter**
5. **mkdir Tiga** ☐ **enter**
6. **ls** ☐ **enter**

CARA II

pastikan direktori aktif Anda masih berada di direktori **DirBaru**!

1. **echo Empat Lima Enam | xargs mkdir** ☐ **enter**
2. **ls** ☐ **enter**

screenshot hasilnya dan jelaskan informasi apa yang didapat dan lihat isi dari direktori **DirBaru**!



```
goathev@3312501009deditfebry-VMware-Virtual-Platform: ~/Dirbaru
goathev@3312501009deditfebry-VMware-Virtual-Platform:~$ mkdir Dirbaru
goathev@3312501009deditfebry-VMware-Virtual-Platform:~$ cd Dirbaru
goathev@3312501009deditfebry-VMware-Virtual-Platform:~/Dirbaru$ mkdir Satu
goathev@3312501009deditfebry-VMware-Virtual-Platform:~/Dirbaru$ mkdir Dua
goathev@3312501009deditfebry-VMware-Virtual-Platform:~/Dirbaru$ mkdir Tiga
goathev@3312501009deditfebry-VMware-Virtual-Platform:~/Dirbaru$ ls
Dua  Satu  Tiga
goathev@3312501009deditfebry-VMware-Virtual-Platform:~/Dirbaru$ echo Empat Lima Enam |
xargs mkdir
goathev@3312501009deditfebry-VMware-Virtual-Platform:~/Dirbaru$ ls
Dua  Empat  Enam  Lima  Satu  Tiga
goathev@3312501009deditfebry-VMware-Virtual-Platform:~/Dirbaru$
```

Perintah pertama membuat direktori DirBaru, masuk ke dalamnya, lalu membuat tiga folder: Satu, Dua, dan Tiga, yang kemudian terlihat saat ls dijalankan. Pada cara kedua, masih di dalam DirBaru, perintah echo Empat Lima Enam | xargs mkdir membuat tiga folder tambahan (Empat, Lima, Enam), sehingga isi DirBaru setelah ls berisi enam direktori tersebut.

KESIMPULAN:

Berikan kesimpulan Anda dari praktikum ini!

Praktikum ini menunjukkan cara kerja dasar redirection untuk mengatur aliran input, output, dan error, sekaligus bagaimana memisahkan atau menggabungkannya sesuai kebutuhan. Perintah find, grep, dan regex memperjelas teknik pencarian file maupun pencarian pola teks secara lebih terarah. Latihan dengan xargs menampilkan cara mengeksekusi banyak argumen secara efisien. Intinya, praktikum ini memperkuat pemahaman tentang pengolahan data di shell dan bagaimana memanfaatkan perintah dasar Linux secara lebih efektif.

SARAN:

Berikan saran jika ada!