

Tugas Sistem Operasi
Progress UAS Menggunakan Bahasa Shell



Nama : Riovaldo Alfiyan Fahmi Rahman

NIM : 2341720209

Absen : 24

Kelas : 1B

Prodi : Teknik Informatika

FITUR FILE MANAGER KW

1. Melihat Lokasi Direktori

```
1)
pwd
;;
```

Pwd (print working directory), jadi perintah ini berguna untuk menampilkan informasi tentang dimana kita berada dalam struktur direktori, perintah ini juga sangat berguna jika kita berpindah-pindah direktori atau saat kita ingin memeriksa direktori mana yang sedang kita gunakan

```
Masukkan Pilihan: 1
/mnt/d/UAS-SISTEMOPERASI
Ingin kembali ke menu? (y/n) : ☐
Ctrl+K
```

2. Melihat isi dari sebuah direktori

```
2)
read -p "Masukkan nama direktori folder : " dir
ls $dir
;;
```

Ls, jadi kode ini bertujuan untuk menampilkan daftar file dan direktori di dalam direktori saat ini atau yang dicari, namun ada beberapa opsi untuk menyesuaikan output dalam perintah **ls** seperti, **-L**, **-A**, **-H**, **-R**, **-T**

```
Masukkan Pilihan: 2
Masukkan nama direktori folder :
README.md daftar.txt img main.sh tes.sh
Ingin kembali ke menu? (y/n) : ☐
```

3. Melihat detail informasi file atau folder

```
3)
read -p "Masukkan nama direktori file : " dir
ls -l $dir
;;
```

Ls-l, **Ls -l** bisa dibilang singkatan dari **list**, ini merupakan perintah yang digunakan untuk menampilkan daftar file dan direktori pada system,

untuk **-l** ini merupakan opsi dari perintah **ls** yang menginstruksikan **ls** untuk menampilkan hasil dalam format Panjang, dalam format Panjang ini bisa berisi informasi tambahan pada setiap file ataupun folder seperti izin akses, jumlah tautan, pemilik, grup ukuran file dalam byte, dan tanggal terakhir kali dimodifikasi. Jadi perintah ini berguna untuk membaca, menampilkan dari suatu file teks dan menampilkan di terminal yang lebih detail

```
Masukkan Pilihan: 3
Masukkan nama direktori file : main.sh
-rwxrwxrwx 1 riovaldo riovaldo 4617 May 27 09:38 main.sh
Ingin kembali ke menu? (y/n) : █
```

4. Melihat isi dari sebuah file

```
4)
read -p "Masukkan nama file : " file
cat $file
;;
```

Cat, perintah **cat** ini berfungsi banyak dalam berbagai scenario, dari bisa memanipulasi dan menampilkan isi file, Tetapi dalam penggunaan skrip **cat** untuk file besar atau tujuan selain menampilkan isi file mungkin bisa dibilang tidak efisien, tetapi ada Solusi seperti menggunakan skrip **less** atau **more**. Jadi fungsi dari perintah ini untuk menampilkan isi dari file yang dipilih dari pengguna

```
-----
Masukkan Pilihan: 4
Masukkan nama file : rio.sh
echo "hello Duniaaaaaaaaaaaaa"
Ingin kembali ke menu? (y/n) : █
```

5. Menghapus file atau folder

```
5)
  read -p "Masukkan path file/folder yang ingin dihapus : " path
  rm -r $path
  ;;
```

Rm -r, Ketika kita menjalankan skrip `rm -r` dan menuliskan nama file / direktori, maka yang ada didalam direktori akan dihapus, termasuk juga subdirektori dan file di dalamnya. Tapi perlu diperhatikan bahwa perintah `rm -r` ini sangat kuat karena tidak memiliki konfirmasi, sehingga jika kita ingin menggunakan perintah skrip `rm -r` itu harus berhati hati, agar kita bisa menghindari penghapusan yang tidak disengaja. Jadi kode ini berguna untuk penghapusan file atau pun folder

```
-----
Masukkan Pilihan: 5
Masukkan path file/folder yang ingin dihapus : tes.sh
Ingin kembali ke menu? (y/n) : █
```

6. Membuat file baru

```
6)
  read -p "Masukkan nama file baru : " filebaru
  touch $filebaru
  ;;
```

Touch, Perintah ini berguna untuk membuat file kosong, namun tidak membuat file kosong saja perintah `touch` juga bisa berguna untuk memperbarui timestamp untuk file baru yang sudah ada. Tetapi fungsi utama kode ini adalah berguna untuk membuat file kosong

```
Masukkan Pilihan: 6
Masukkan nama file baru : index.ts
Ingin kembali ke menu? (y/n) : █
```

UAS-SISTEMOPERASI

```
> img U
  daftar.txt U
  index.ts U
  main.sh M
  README.md M
```

7. Membuat folder baru

```
7)
read -p "Masukkan nama folder baru : " folderbaru
mkdir $folderbaru
;;
```

Mkdir (make directory), perintah ini biasanya digunakan untuk membuat folder / direktori baru.

```
Masukkan Pilihan: 7
Masukkan nama folder baru : testing
Ingin kembali ke menu? (y/n) : 
```

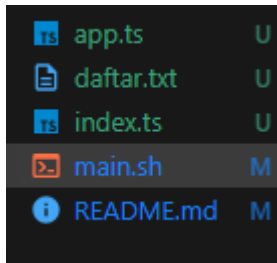
```
UAS-SISTEMOPERASI
> img
> testing
  daftar.txt  U
  index.ts   U
  main.sh    M
  README.md  M
```

8. Menyalin File

```
8)
read -p "Masukkan nama file yang ingin disalin : " sebelum
read -p "Masukkan nama file yang sudah disalin : " sesudah
cp $sebelum $sesudah
;;
```

Cp (copy), perintah ini berguna untuk menyalin sebuah file atau direktori dari satu lokasi ke lokasi lainnya, tetapi fungsi utamanya adalah membuat Salinan file baru atau direktori yang sudah ada , dengan memberikan nama baru atau menempatkannya di lokasi yang berbeda

```
Masukkan Pilihan: 8
Masukkan nama file yang ingin disalin : index.ts
Masukkan nama file yang sudah disalin : app.ts
Ingin kembali ke menu? (y/n) : 
```

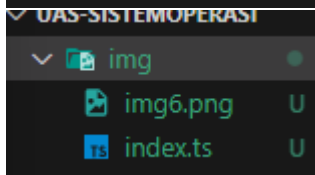


9. Memindah File

```
9)
read -p "Masukkan nama file yang ingin dipindah : " sebelum
read -p "Masukkan tujuan file yang ingin dipindah : " sesudah
mv $sebelum $sesudah
;;
```

Mv, Fungsi utama dari perintah ini merupakan memindah file dari lokasi default ke lokasi lainnya, tetapi tidak hanya memindah file saja perintah mv juga bisa melakukan mengubah nama file atau direktori, pemindahan direktori, jadi secara kesimpulan perintah mv berfungsi untuk mengatur file dan direktori yang mungkin lebih efisien

```
Masukkan Pilihan: 9
Masukkan nama file yang ingin dipindah : index.ts
Masukkan tujuan file yang ingin dipindah : img
Ingin kembali ke menu? (y/n) : 
```

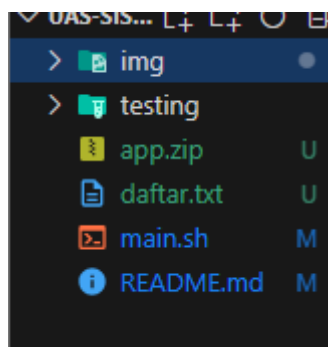


10. Kompres file atau folder

```
10)
read -p "Masukkan nama file atau folder yang ingin dikompres : " sebelum
read -p "Masukkan nama file atau folder yang sudah dikompres : " sesudah
tar -czvf $sesudah $sebelum
;;
```

Tar -czvf, untuk perintah tar sendiri berfungsi sebagai mengompres atau bisa juga mengekstrak arsip, dan juga tar merupakan alat serbaguna yang digunakan untuk membuat mengelola dan mengekstrak arsip, kemudian untuk perintah -czvf, untuk **-C** sendiri berguna untuk create atau membuat arsip baru, **-Z** sendiri berguna untuk gzip, **-V** sendiri berguna sebagai verbose atau menampilkan detail tentang file yang sedang diarsipkan dan dikompresi saat proses itu berlangsung, kemudian untuk **-F** berguna untuk mengikuti nama file arsip yang kita buat

```
Masukkan Pilihan: 10
Masukkan nama file atau folder yang ingin dikompres : app.ts
Masukkan nama file atau folder yang sudah dikompres : app.zip
app.ts
Ingin kembali ke menu? (y/n) : 
```



11. Extract file atau folder

```
11)
read -p "Masukkan file yang akan diekstrak (file.zip) : " file
tar -xzf $file
;;
```

Tar -xzf, kurang lebih sama dengan proses compress namun yang berbeda adalah di perintah **-X** yang dimana berguna untuk mengekstrak arsip, dan juga perintah ini memberi tahu kepada tar untuk mengesktrak file dari arsip yang dipilih pengguna, bukan membuat arsip baru

```
Masukkan Pilihan: 11
Masukkan file yang akan diekstrak (file.zip) : app.zip
app.ts
Ingin kembali ke menu? (y/n) : 
```

```
> testing
  app.ts      U
  app.zip     U
  daftar.txt  U
  main.sh     M
  README.md  M
```

12. Mengganti nama file

```
12)
read -p "Masukkan nama file yang ingin diganti : " filelama
read -p "Masukkan nama file baru : " filebaru
mv $filelama $filebaru
::
```

Mv, Jadi selain memindah file perintah mv juga berfungsi sebagai mengganti nama file ataupun bisa mengganti nama folder / direktori

```
Masukkan Pilihan: 12
Masukkan nama file yang ingin diganti : app.ts
Masukkan nama file baru : Navbar.ts
Ingin kembali ke menu? (y/n) : 
```

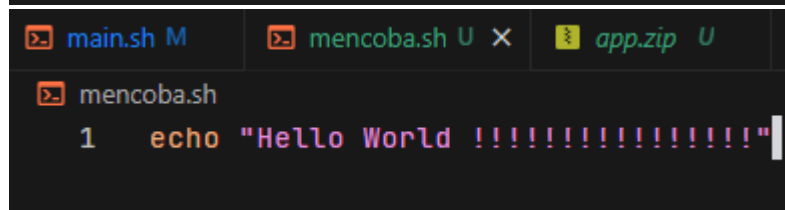
```
  app.zip     U
  daftar.txt  U
  main.sh     M
  Navbar.ts   U
  README.md  M
```


13. Menjalankan Program

```
13)
read -p "Masukkan nama file yang ingin dijalankan : " file
source ./ $file
::
```

Source, jadi perintah ini berfungsi untuk menjalankan sebuah perintah atau skrip yang ada didalam file skrip yang berbeda pada file dalam shell saat ini, dan memuat fungsi-fungsi shell dan variable-variabel yang didefinisikan dalam skrip tersebut

```
Masukkan Pilihan: 13
Masukkan nama file yang ingin dijalankan : mencoba.sh
Hello World !!!!!!!!!!!!!!!
Ingin kembali ke menu? (y/n) : 
```



```
main.sh M  mencoba.sh U x  app.zip U
mencoba.sh
1  echo "Hello World !!!!!!!!!!!!!!!"
```

14. Mengunci File

```
14)
read -p "Masukkan nama file yang ingin dikunci : " file
read -sp "Masukkan kata sandi untuk mengunci file : " password
echo ""
zip -P $password $file.zip $file
echo "File berhasil dikunci dengan kata sandi."
::
```

Jadi script diatas berguna untuk mengunci sebuah file dengan menggunakan bantuan program zip dan juga memberikan kata sandi untuk mengakses file tersebut, untuk letak proses penguncian file ada di skrip **zip -P \$password \$file.zip \$file**

```
Masukkan Pilihan: 14
Masukkan nama file yang ingin dikunci : rio.sh
Masukkan kata sandi untuk mengunci file :
  adding: rio.sh (deflated 28%)
File berhasil dikunci dengan kata sandi.
Ingin kembali ke menu? (y/n) : 
```

rio.sh.zip	U
------------	---

15. Membuka File yang disandi

```
15)
read -p "Masukkan nama file yang ingin diekstrak : " file
read -sp "Masukkan kata sandi untuk membuka file : " password
echo ""

if [ -f "$file" ]; then
    if unzip -t -P "$password" "$file" 2>/dev/null; then
        unzip -P "$password" "$file"
        echo "File berhasil diekstrak."
    else
        echo "Kata sandi salah atau file tidak dapat diekstrak."
    fi
else
    echo "Kata Sandi Salah."
fi
;;
```

Script diatas merupakan berfungsi sebagai membuka file zip yang dikunci seperti sebelumnya, dan letak proses membuka sebuah file yang dikunci ada di skrip **if [-f "\$file"]; then sampai else**

```
Masukkan Pilihan: 15
Masukkan nama file yang ingin diekstrak : rio.sh.zip
Masukkan kata sandi untuk membuka file :
Archive:  rio.sh.zip
  testing: rio.sh                      OK
No errors detected in compressed data of rio.sh.zip.
Archive:  rio.sh.zip
  inflating: rio.sh
File berhasil diekstrak.
Ingin kembali ke menu? (y/n) : 
```

rio.sh	U
rio.sh.zip	U

FITUR PENGELOLA KONTAK TELEPON

Note : jadi ada logika tambahan yang dimana data nama dan telepon disimpan di daftar.txt

```
# File untuk menyimpan kontak  
PHONEBOOK_FILE="daftar.txt"
```

16. Menambah Kontak

```
14)  
    read -p "Masukkan nama kontak : " nama  
    read -p "Masukkan nomor telepon : " nomor  
    echo "$nama:$nomor" >> $PHONEBOOK_FILE  
    echo "Kontak berhasil ditambahkan."  
    ;;
```

Jadi fitur diatas itu berguna untuk menambahkan kontak baru, dan jika berhasil maka akan disimpan pada daftar.txt, yang dimana proses penyimpanan tersebut ditandai dengan >>

```
Masukkan Pilihan: 14  
Masukkan nama kontak : Danang  
Masukkan nomor telepon : 908723153921  
Kontak berhasil ditambahkan.  
Ingin kembali ke menu? (y/n) : 
```

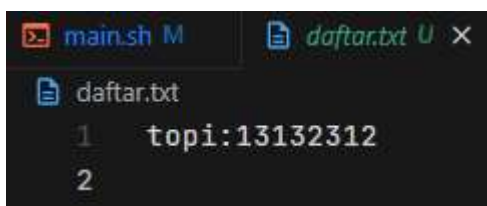
```
main.sh M  daftar.txt U x  
daftar.txt  
1  topi:13132312  
2  Danang:908723153921  
3  Ctrl+L to chat, Ctrl+K to generate
```

17. Menghapus daftar yang ada di kontak

```
15)
read -p "Masukkan nomor kontak yang ingin dihapus : " nomor
if grep -q ":$nomor" $PHONEBOOK_FILE; then
    grep -v ":$nomor" $PHONEBOOK_FILE > temp.txt
    mv temp.txt $PHONEBOOK_FILE
    echo "Kontak berhasil dihapus!"
else
    echo "Kontak dengan nomor HP tersebut tidak ditemukan."
fi
;;
```

Jadi fungsi if yang pertama yaitu untuk mengecek data nomor yang ada di daftar.txt itu ada atau tidak, kemudian untuk grep -v berfungsi sebagai file temp.txt secara sementara, dan untuk mv temp.txt itu berguna untuk mengganti file asli daftar.txt dengan file sementara yaitu temp.txt yang telah diperbarui

```
-----
Masukkan Pilihan: 15
Masukkan nomor kontak yang ingin dihapus : 908723153921
Kontak berhasil dihapus!
Ingin kembali ke menu? (y/n) : █
```



```
main.sh M  daftar.txt U x
daftar.txt
1 topi:13132312
2
```

18. Melihat daftar kontak

```
16)
    if [ -s $PHONEBOOK_FILE ]; then
        cat $PHONEBOOK_FILE
    else
        echo "Buku telepon kosong."
    fi
;;
```

Jadi untuk fungsi if pertama berfungsi sebagai pengecekan data di daftar.txt, jika ada maka fungsi cat akan dijalankan dan akan menampilkan isi dari daftar.txt

```
-----
Masukkan Pilihan: 16
topi:13132312
Ingin kembali ke menu? (y/n) : █

main.sh M  daftar.txt U ×
daftar.txt
1  topi:13132312
2
```

FITUR PENGELOLA PERANGKAT KITA

19. Melihat Informasi CPU

```
17) echo "Informasi CPU Anda : "  
if command -v lscpu > /dev/null; then  
    lscpu  
elif [ -f /proc/cpuinfo ]; then  
    cat /proc/cpuinfo  
else  
    echo "Informasi CPU tidak tersedia."  
fi  
::
```

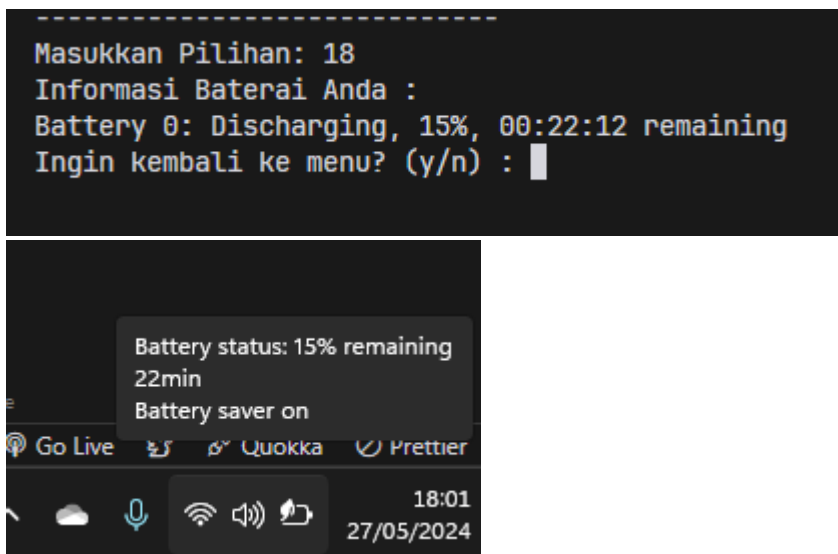
Jadi untuk fungsi if pertama itu berguna untuk mengecek lscpu tersedia di system ada atau tidak, kemudian cat/proc/cpuinfo itu berguna untuk menampilkan informasi detail tentang cpu

```
=====
FITUR PENGELOLA PERANGKAT KITA
=====
17. Informasi CPU
18. Informasi Hardware
=====
Masukkan Pilihan: 17
Informasi CPU Anda :
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:          39 bits physical, 48 bits virtual
Byte Order:             Little Endian
CPU(s):                 12
On-line CPU(s) list:   0-11
Vendor ID:              GenuineIntel
Model name:             12th Gen Intel(R) Core(TM) i5-12450H
CPU family:             6
Model:                 154
Thread(s) per core:    2
Core(s) per socket:    6
Socket(s):              1
Stepping:               3
BogoMIPS:               4992.00
Flags:                  fpu vme de pse tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat ps
e3e clflush
                        mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc r
cp_good nop
                        1 xtopology tsc_reliable nopl tsc cpuid pni pclmulqdq vmx sse3 f
ma cx16 pti
                        d sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx
f16c rdprn
                        d hypervisor lahf_lm abm 3dnowprefetch invpcid_single ssbd ibrs ibpb
stibp ibrs
                        _enhanced tpr_shadow vmx1 ept vpid ept_ad fsgsbase tsc_adjust hle1 a
vx2 smep bmi
                        12 erms invpcid rdseed adx smap clflushopt clwb sha_ni xsaveopt xsave
ec xgetbv1
                        xsavec unlp waitpkg gfni vaes vpclmulqdq rdpid movdiri movdir64b fsr
a rd_clear
                        serialize flush_lid arch_capabilities
Virtualization features:
Virtualization:         VT-x
Hypervisor vendor:      Microsoft
Virtualization type:    full
Caches (sum of all):
L1d:                    288 KiB (6 instances)
L1i:                    192 KiB (6 instances)
L2:                     7.5 MiB (6 instances)
```

20. Melihat Informasi Baterai

```
18)
echo "Informasi Baterai Anda : "
if command -v acpi > /dev/null; then
    acpi -b
else
    echo "acpi tidak tersedia."
fi
;;
```

Pertama untuk fungsi if itu berguna untuk mengecek acpi tersedia di system atau tidak, kemudian jika ada maka akan menjalankan acpi -b, yang dimana acpi itu perintah untuk menampilkan informasi tentang status baterai, dan untuk -b berguna untuk menampilkan informasi hanya tentang baterai



21. Melihat Informasi Waktu

```
21)
current_time=$(date "+%Y-%m-%d %H:%M:%S")
echo "Waktu saat ini adalah: $current_time"
;;
```

Perintah diatas berguna untuk menampilkan waktu secara real time, proses untuk mendapatkan informasi waktu pada perangkat saat ini itu menggunakan perintah date kemudian akan disimpan dalam variable **current_time**


```
-----  
Masukkan Pilihan: 21  
Waktu saat ini adalah: 2024-06-03 11:18:43  
Ingin kembali ke menu? (y/n) : █
```

22. Melihat informasi kecepatan wifi

```
22)  
timestamp=$(date "+%Y-%m-%d %H:%M:%S")  
  
hasil=$(speedtest-cli --simple)  
  
if [ $? -eq 0 ]; then  
    echo "$timestamp - Speedtest Sukses"  
    echo "$hasil"  
    echo "$timestamp - $hasil" >> speedtest.log  
else  
    echo "$timestamp - Speedtest Gagal" >&2  
    echo "$timestamp - Speedtest Gagal" >> speedtest.log  
fi  
::
```

Jadi kode diatas berguna untuk memberi informasi kecepatan wifi dengan menggunakan bantuan speedtest, dan hasilnya akan disimpan dalam berbentuk log file, Bersama juga dengan timestamp

```
-----  
Masukkan Pilihan: 22  
2024-06-03 11:00:20 - Speedtest Sukses  
Ping: 60.59 ms  
Download: 1.02 Mbit/s  
Upload: 21.37 Mbit/s  
Ingin kembali ke menu? (y/n) : █
```