

A. TUJUAN

1. Menggunakan perintah **sed** untuk modifikasi file
2. Menggunakan perintah **sed** untuk operasi cyclically.
3. Menggunakan perintah **sed** sebagai control flow untuk eksekusi didalam perintah script

B. DASAR TEORI

Sed adalah editor teks noninteraktif untuk melakukan routine modifikasi file teks. Sed menyediakan kunci modifikasi untuk isi teks dalam sebuah file dan operasi pembacaan file forward. Sebagai contoh, sed dapat mendelete isi semua baris serta memberikan sebagai pattern teks, mengganti satu pattern dengan yang lainnya pada sebuah baris, membaca satu file ke dalam file lainnya dalam tempat yang ditentukan, atau menyebarkan bagian input file ke output file. Sed tidak dapat melakukan tugas menambah sebuah kolom dalam sebuah file, melakukan perubahan file yang rumit, atau menyimpan bagian dari sebuah file yang akan digunakan belakangan. Operasi rumit adalah tugas ideal untuk awk, yang mana sangat diperlukan dalam pemrograman sed. Sintak umum dari perintah sed seperti dibawah ini.

Syntax:

sed {expression} {file}

Sed mirip dengan *ed*, yaitu digunakan sebagai desain control untuk sebuah script. Perbedaanya sed untuk operasi internal, sedangkan ed untuk edit buffer

setiap waktu sebuah perintah yang masuk. Sed mencari perintah script setiap saat pada baris baru yang dibaca dari input teks.

Sed merupakan operasi cyclically, yang mana sebuah cycle harus berisi :

1. membaca baris input kedalam pattern space
2. mengeksekusi edit script
3. men-copy pattern space untuk dikeluarkan/output

Seperti halnya ed, sed juga dipergunakan untuk edit buffer dan menahannya sebagai teks edit. Perbedaananya sed edit buffer, memanggil pattern space yang berisi satu baris teks, sedangkan ed edit buffer berisi segala file. Meskipun demikian perintah sed fleksibel untuk baris pertama dalam pattern space. Konsep dari baris pertama yaitu dapat menahan isi dari baris pertama dan menjadikannya baris baru. Disamping pattern space sed juga berisi hold buffer. Beberapa perintah untuk pertukaran teks akhir dan ke empat antara pattern dan hold buffer dan opsi pada perintah sed adalah sebagai berikut:

Tabel 13-1. Opsi pada perintah sed

Option	Meaning
-n	Don't output the pattern space at the end of each cycle. When -n is given output is produced only when of the print commands is encountered
-e	<i>script</i> The -e argument specifies that the following argument is an the script
-f	<i>scriptfile</i> The -f argument specifies that the following argument is the name of a file that contains an editing script

Sed memberi supply dengan sebuah script, salah satunya adalah pada command line atau didalam sebuah file. Sed juga menggunakan baris alamat dan beberapa operator untuk mencetak, menghapus dan sebagainya. Penggunaan baris alamat pada sed (sed addresses line) dan beberapa operator yang digunakan seperti ditunjukkan pada tabel 13-2 hingga 13-4. Perintah sed dapat menerima "nol", "satu", atau "dua" address. Nol address artinya melakukan perintah setiap baris/perbaris.

Tabel 13-2. Penggunaan Address pada sed

Address code	Meaning
n	An absolute line number n. The line counter is not reset each time a new file is processed. First line = 20 lines, line 21 is the first line of the second file.
\$	The last line of the input
/pat/	A context address matches any line containing the pat regular expression.

Tabel 13-3. Basic sed operators

Operator	Name	Effect
[address-range]/p	print	Print [specified address range]
[address-range]/d	delete	Delete [specified address range]
s/pattern1/pattern2/	substitute	Substitute pattern2 for first instance of pattern1 in a line
[address-range]/s/pattern1/pattern2/	substitute	Substitute pattern2 for first instance of pattern1 in a line, over <i>address-range</i>
[address-range]/y/pattern1/pattern2/	transform	replace any character in pattern1 with the corresponding character in pattern2, over <i>address-range</i> (equivalent of tr)
g	global	Operate on <i>every</i> pattern match within each matched line of input

Tabel 13-4. Examples of sed operators

Notation	Effect
8d	Delete 8th line of input.
/^\$/d	Delete all blank lines.
1,/^\$/d	Delete from beginning of input up to, and including first blank line.
/Jones/p	Print only lines containing "Jones" (with -n option).
s/Windows/Linux/	Substitute "Linux" for first instance of "Windows" found in each input line.
s/BSOD/stability/g	Substitute "stability" for every instance of "BSOD" found in each input line.
s/ *\$//	Delete all spaces at the end of every line.
s/00*/0/g	Compress all consecutive sequences of zeroes into a single zero.
/GUI/d	Delete all lines containing "GUI".
s/GUI//g	Delete all instances of "GUI", leaving the remainder of each line intact.

Satu address adalah berarti melakukan perintah untuk semua baris yang sesuai dengan address. Dan dua address berarti melakukan perintah dalam range baris. Jika baris kedua kurang dari baris pertama, maka melakukan perintah hanya untuk baris pertama saja. Untuk mengeksekusi sebuah group dapat menggunakan perintah dengan tanda kurung kurawal `{ }`. Sintak satu atau dua address dinyatakan dengan `{ perintah, perintah pernyataan baris, dan akhir }` untuk memberi batas akhir dari sebuah group. Dibawah ini adalah contoh perintah pembacaan script dari sebuah terminal.

```
$ sed -n -f /dev/tty/ usr/dict/words
```

Modifikasi Teks

Perintah sed modifikasi teks dapat ditambahkan insert, atau delete line sebagai berikut:

- mengganti satu group line dengan lainnya
- mensubstitusi satu pattern teks dengan lainnya
- mentranslate satu group karakter kedalam group lainnya

contoh:

```
addr a\  
text\  
text
```

Perintah penambahan tempat *text* pada output, sebelum baris input berikutnya dibaca.

```
addr1,addr2 c\  
text\  
text\  
text\  
text
```

Perintah pengganti delete setiap address pattern space, output text, dan memulai cycle baru.

```
Addr1, addr2 d
```

Perintah delete untuk menghapus pattern space dan memulai cycle baru.

```
Addr1, addr2 D
```

Perintah delete variant untuk menghapus inisial segmen pada pattern pace dan memulai cycle baru. D adalah ekuivalen dari d jika pattern space berisi hanya satu baris.

```
Addr i\  
text\  
text
```

Perintah insert untuk segera menempatkan text pada output.

Tetapi semua baris akhir menggunakan \ dan menuju baris baru.

addr1,addr2 s/expr/repl/f

Perintah substitusi repl dengan expr pada semua baris address

addr1,addr2 y/string1/string2/

Perintah translate karakter pada string1 dan string2

Perintah penambahan dan insert hanya untuk satu address, sedangkan delete, mengganti baris, substitusi pattern teks, dan translate untuk satu atau dua address. Contoh perintah sed script editing:

```
$ sed -f script remind
```

dimana : script adalah nama file script/program

remind adalah nama file yang berisi data yang akan diakses dengan perintah sed dengan opsi -f atau -e.

Control Flow

Control flow pada sed ada dua macam yaitu:

1. **branch**

mengganti point untuk eksekusi didalam perintah script. Pernyataan branch dengan sebuah label atau untuk akhir dari script. Sed mempunyai unconditional branch yang merupakan/selalu branch dan conditional branch, dimana branch jika berupa bentuk substitusi.

2. **Control flow** yang biasa digunakan untuk mengganti operasi cycle.

Contoh control flow untuk operasi cycle:

:label Membuat label dengan nama symbol untuk lokasi dalam script

addr1,addr2 b label

Branch pada label atau akhir dari script jika label tidak ada

addr1,addr2 n

Menulis pattern space pada output dan membaca baris input berikutnya.

addr1,addr2 N

Penambahan baris berikutnya pada input untuk pattern space.

addr q

Keluar dari proses penulisan pattern space ke output, dan proses dihentikan.

```
addr1,addr2 t label
```

Branch pada label jika terdapat substitusi yang dibuat dari akhir t atau sampai akhir dari baris input.

C. PROSEDUR

1. Masuk ke sistem operasi Linux.
2. Login sebagai user yang telah ditentukan.
3. Gunakan perintah sed pada pemrograman shell.
4. Catatlah hasil dari setiap program pada prosedur no.3 sebagai laporan sementara.

D. PERCOBAAN

1. Menggunakan perintah script sed dari terminal.

Buat file baru dengan nama word sebagai berikut :

```
$ cat > word
```

```
agus
irma
afiludin
candra
^d
```

simpan file word ini pada direktori sbb:

```
/home/nama user/word
```

```
$ sed -n -f /home/nama user/word
```

2. Menggunakan perintah sed dari terminal dengan opsi -f dan -e.

```
$ cat > remind
```

```
Jan today at 4
Call DEC
Add serial line for Brad
Home at five sharp
^d
```

```
$ cat > script
s/DEC/Dept. Environ. Cons./
/Jan/s/today/tomorrow/
1i\
Werner and Raquel this weekend\
Feed polly for adam
4d
^d
```

```
$ sed -f script remind
```

```
$ sed -e /DEC/d -e s/five/four/ remind
```

3. Menggunakan perintah sed dari terminal dengan beberapa opsi.

```
$ cat > f1
.CX 3
.DS
Twas Sol and solstice
Ad and Astra
Til Hic saw Sum
In Corporate
.DE
^d

$ cat > script1
/DS/b fail
/KS/b fail
/EQ/,/EN/p
b
: fail
s/./No displays allowed at start/p
q
^d

$ sed -n -f script1 f1
$ sed -e s/DS/EQ/ -e s/DE/EN/ -n -f script1 f1
```

4. Menggunakan perintah sed untuk menampilkan bilangan random 10 digit.

```
$ cat > sed4.sh
head -c4 /dev/urandom | od -N4 -tu4 | sed -ne '1s/.*/p'
exit 0

$ ./sed4.sh
```

5. Menggunakan perintah sed untuk mencari data string.

```
$ vi sed5.sh
#!/bin/bash
directory=/usr/bin/
fstring="Free Software Foundation"
for file in $( find $directory -type f -name '*' | sort )
do
    strings -f $file | grep "$fstring" | sed -e
"s%$directory%"
done
exit 0

$ ./sed5.sh
```

6. Perintah sed untuk menampilkan data baris.

```
$cat > lines
10
20
30
```

```
^d
$ sed -n -e 'p (enter)
>g
>p' lines (enter)
```

7. Perintah sed untuk mengambil data dari suatu file dan disimpan ke file lain.

```
$cat >script
s/CUXBASIC/3/g
s/SCMDARGS/CSHONE.2/g
s/SMETA/CSHONE.8/g
s/CSHONE/4/g
^d

$cat >doc
Chapter CUXBASIC covers the fundamental
Feature of the URC language. However two
important features - - arguments and meta-
arguments - - aren't discussed until
Section SCMDARGS and section SMETA
^d

$ sed -f script doc
```

8. Sed untuk membaca semua baris pada sebuah file dan *delete* baris tertentu

```
$ vi sed8.sh
#!/bin/bash
sed -e /^s/d "$1"
exit 0

$ ./sed8.sh file1.txt
```

9. Menggunakan perintah sed untuk substitusi pattern

```
$ vi sed9.sh
#!/bin/bash
ARGS=3
E_BADARGS=65

if [ $# -ne "$ARGS" ]
then
    echo "Usage: `basename $0` old-pattern new-pattern
filename"
    exit $E_BADARGS
fi

old_pattern=$1
new_pattern=$2

if [ -f "$3" ]
then
    file_name=$3
```

```

else
    echo "File \"$3\" does not exist."
    exit $E_BADARGS
fi

sed -e "s/$old_pattern/$new_pattern/g" $file_name
exit 0

$ cat word
$ ./sed9.sh agus candra word
$ ./sed9.sh candra agus word

```

10. Menggunakan perintah sed sebagai implementasi perintah grep

```

$ vi sedb.sh

#!/bin/bash
E_BADARGS=65

if [ -z "$1" ]
then
    echo "Usage: `basename $0` pattern"
    exit $E_BADARGS
fi
echo
for file in *
do
    output=$(sed -n /"$1"/p $file)
    if [ ! -z "$output" ]
    then
        echo -n "$file: "
        echo $output
    fi
    echo
done
echo
exit 0

$ ./sedb.sh afileudin      (afileudin => isi file word prog. No.1)
$ ./sedb.sh 200            (200 => isi file sdata di praktikum 12 prog. No.1)

```

11. Menggunakan perintah sed untuk pembuatan kolom

```

$ sedc.sh

#!/bin/bash
(printf "PERMISSIONS LINKS OWNER GROUP SIZE MONTH DAY
HH:MM PROG-NAME\n" \;
ls -l | sed 1d) | column -t
exit 0

$ ./sedc.sh

```

Bandingkan hasilnya dengan perintah ls -l.

E. Latihan

1. Gunakan perintah `sed` untuk membuat program aplikasi **"Message filter"** yang berfungsi membantu penyaringan kata-kata yang tidak diinginkan agar tidak muncul atau diganti dengan karakter lain.

Misalnya kata ***porno*** dan ***seks*** seperti contoh berikut:

Meskipun sudah ada sangsi hukumnya, orang yang sudah kebiasaan ***pornoaksi*** ya tetap saja melakukan ***seks*** bebas.

Hasil penyaringan:

Meskipun sudah ada sangsi hukumnya, orang yang sudah kebiasaan #####aksi ya tetap saja melakukan ##### bebas.

2. Pada sebuah toko elektronika menjual barangnya dengan beberapa nomor item dan harga, untuk membeli barang pembeli harus memasukkan nomor item barang dan harga barang tersebut. Selanjutnya akan muncul informasi tentang tipe, spesifikasi, dan harga barang yang telah dipilih oleh pembeli. Dari kasus diatas buatlah programnya dengan menggunakan perintah `sed` untuk menampilkan informasi tipe, spesifikasi dan harga barang yang terdapat pada sebuah file teks (`.txt`) berdasarkan masukan nomor item dan harga barang.