

Pencarian dan Pemrosesan Teks

Praktikum Sistem Operasi

Ilmu Komputer IPB

2017

Pencarian

man

Mencari halaman manual suatu program, fungsi, dan sebagainya.

`man [SECTION] PAGE`

- ▶ `q`: *quit*; keluar
- ▶ `/`: pencarian kata
 - ▶ `n`: *next*; lanjutkan pencarian kata
 - ▶ `N`: *next-reverse*; lanjutkan pencarian kata mundur



Gambar 1: Bagian halaman manual

`which`

Mencari lokasi *file* program.

`which` COMMAND

locate

Mencari lokasi *file* berdasarkan namanya pada *database*.

```
locate [OPTION] 'PATTERN'
```

- ▶ *-c*: *count*; tampilkan jumlah *file* yang ditemukan
- ▶ *-i*: *ignore-case*

find

Mencari *file* langsung pada sebuah hierarki direktori.

```
find [PATH] [TEST]...
```

- ▶ `-name 'PATTERN'`
- ▶ `-iname 'PATTERN'`
- ▶ `-size [+_]N[kMG]`
- ▶ `-atime [+_]N`
- ▶ `-mtime [+_]N`
- ▶ `-empty`
- ▶ `-type [dfl]`

xargs

Mengubah tiap baris masukan menjadi argumen suatu perintah.

```
... | xargs COMMAND
```

Contoh

```
# temukan semua file backup (.bak), lalu hapus  
find . -name '*.bak' | xargs rm
```


grep

Mencetak baris teks yang cocok dengan suatu pola.

```
grep [OPTION] 'PATTERN' FILE
```

- ▶ `-c`: *count*; tampilkan jumlah baris yang cocok
- ▶ `-i`: *ignore-case*
- ▶ `-r`: rekursif
- ▶ `-v`: *invert*; kebalikan dari pola yang diberikan

Editor Teks

pager (less)

Menampilkan *file* teks per halaman layar.

`pager [FILE] ...`

- ▶ `q`: *quit*; keluar
- ▶ `/`: pencarian kata
 - ▶ `n`: *next*; lanjutkan pencarian kata
 - ▶ `N`: *next-reverse*; lanjutkan pencarian kata mundur

editor (nano)

Membuat dan mengedit *file* teks.

```
editor [OPTION] [FILE] ...
```

- ▶ `-i`: *indent*; indentasi otomatis
- ▶ `-u`: *undo*; aktifkan fitur *undo*
- ▶ `^O`: *write-out*; simpan *file*
- ▶ `^X`: *exit*

micro

Editor teks yang intuitif dan mudah dipakai, penerus 'nano'.

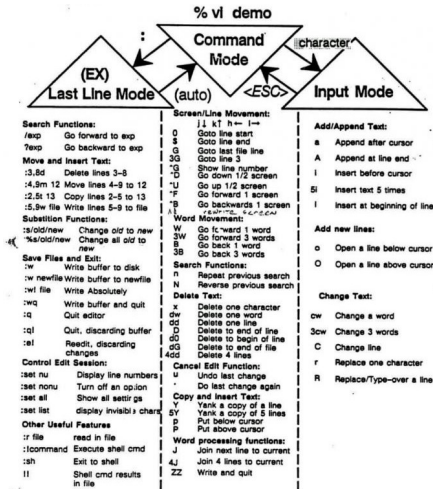
`micro [FILE]...`

- ▶ `^X`, `^C`, `^V`: *cut, copy, paste*
- ▶ `^Z`: *undo*
- ▶ `^S`: *save*
- ▶ `^Q`: *quit*

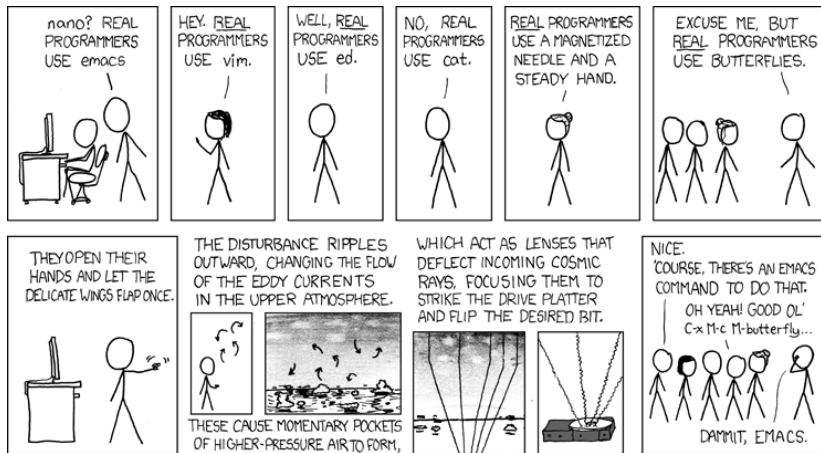
vi

Editor teks untuk *programmer*.

```
vi [OPTION] [FILE]...
```



Gambar 2: Petunjuk singkat vi



Gambar 3: *Real programmers*

Pemrosesan Teks

cat

Menggabungkan *file* dan menampilkannya ke layar.

```
cat [FILE] ...
```

split

Memecah *file* menjadi beberapa bagian.

```
split [OPTION] FILE [PREFIX]
```

- ▶ `-b`: *bytes*; pecah per sekian *byte*
- ▶ `-l`: *lines*; pecah per sekian baris

head

Menampilkan bagian awal *file* (*default*: 10 baris).

`head [OPTION] [FILE]`

- ▶ `-b`: *bytes*; tampilkan sekian *byte* awal
- ▶ `-n`: *lines*; tampilkan sekian baris awal

tail

Menampilkan bagian akhir *file* (*default*: 10 baris).

```
tail [OPTION] [FILE]
```

- ▶ `-b`: *bytes*; tampilkan sekian *byte* terakhir
- ▶ `-n`: *lines*; tampilkan sekian baris terakhir

sort

Mengurutkan baris teks pada *file*.

```
sort [OPTION] [FILE]
```

- ▶ `-n`: *numeric*; urutkan secara numerik
- ▶ `-r`: *reverse*; urutkan terbalik

uniq

Menghilangkan baris teks yang berulang.

`uniq [OPTION] [FILE]`

- ▶ `-c`: *count*; tampilkan jumlah kemunculan
- ▶ `-i`: *ignore-case*

tr

Translasi karakter dari set pertama ke set kedua.

`tr [OPTION] SET1 [SET2]`

- ▶ `-c`: *complement*; gunakan komplemen dari *SET1*
- ▶ `-d`: *delete*; hapus karakter pada *SET1*
- ▶ `-s`: *squeeze*; hapus karakter yang berulang pada *SET1*

sed

Stream editor, manipulasi baris teks dengan *regular expression*.

```
sed [OPTION] 's/SEARCH/REPLACE/' [FILE]
```

- ▶ `-e`: *execute*; tambahkan perintah untuk dieksekusi
- ▶ `-i`: *in-place*; edit *file* langsung

cut

Mengambil karakter/kolom tertentu dari tiap baris teks.

`cut` OPTION [FILE]

- ▶ `-c`: *char*; cetak karakter berikut
- ▶ `-f`: *field*; cetak kolom berikut
- ▶ `-d`: *delimiter*; pemisah antarkolom

paste

Menggabungkan tiap baris dari beberapa *file* per kolom.

```
paste [OPTION] [FILE...]
```

- ▶ `-d:` *delimiter*; pemisah antarkolom
- ▶ `-s:` *serial*; gabungkan isi *file* menjadi sebaris

join

Menggabungkan baris dari dua *file* berdasarkan kolom bersama.

```
join [OPTION] FILE1 FILE2
```

- ▶ `-t`: karakter pemisah antarkolom
- ▶ `-1`: nomor kolom join untuk *FILE1*
- ▶ `-2`: nomor kolom join untuk *FILE2*

diff

Membandingkan antara dua *file* per baris.

```
diff [OPTION] FILE1 FILE2
```

- ▶ `-u`: *unified*; keluaran beserta konteks
- ▶ `-y`: *side-by-side*; keluaran dua kolom

WC

Mencetak jumlah baris, kata, dan karakter dari suatu *file*.

```
wc [OPTION] [FILE...]
```

- ▶ `-c`: *char*; cetak jumlah karakter
- ▶ `-l`: *line*; cetak jumlah baris
- ▶ `-w`: *word*; cetak jumlah kata

Ekspresi Reguler

Ekspresi Reguler (Regex)

Regex adalah susunan karakter yang merupakan pola pencarian. Regex digunakan untuk mencari *string* tertentu pada teks.

Misal, ekspresi reguler `/G64\d{6}/` dapat mencocokkan NIM semua mahasiswa S1 Ilmu Komputer IPB.

Latihan dan informasi lebih lanjut, kunjungi <http://regexr.com>.

Referensi Singkat

► Karakter

`.` karakter apapun selain *newline*

`[abc]` karakter a, b, atau c

`[^abc]` bukan karakter a, b, atau c

`[a-g]` karakter antara a sampai g

► Kelas karakter

`\w` kata, `[A-Za-z0-9_]`

`\d` digit, `[0-9]`

`\s` *whitespace*

► *Anchor*

`^` awal baris

`$` akhir baris

Referensi Singkat

► Jumlah

* 0 atau lebih

+ 1 atau lebih

? 0 atau 1

{3} tepat 3

{3,} 3 atau lebih

{3,5} antara 3 sampai 5

► Grup

(...) membuat grup

\n referensi balik grup ke- n



Gambar 4: *Regex saves the day*

Tugas

UNIX Text Processing Contest 2017¹

¹<https://www.hackerrank.com/unix-text-processing-2017>