

# REFERENSI CEPAT GREP

Pencocokan pola, regex, pencarian rekursif, konteks, filter

## Penggunaan Dasar

### Menjalankan grep

```
grep "pattern" file.txt # search in file
grep "error" *.log # search multiple files
grep "hello" file1.txt file2.txt # explicit file list
cat file.txt | grep "pattern" # pipe input
dmesg | grep -i "usb" # filter command output
```

### Flag Umum

- i Pencocokan tidak peka huruf besar/kecil
- v Balik pencocokan — tampilkan baris yang tidak cocok
- c Tampilkan jumlah baris yang cocok
- n Tampilkan nomor baris
- l Daftar nama file dengan kecocokan saja
- L Daftar nama file tanpa kecocokan
- w Cocokkan kata utuh saja
- x Cocokkan baris utuh saja

## Pola Regex

### Basic Regular Expressions (BRE)

- [ ] Karakter tunggal apa pun
- \* Nol atau lebih dari elemen sebelumnya
- ^ Awal baris
- \$ Akhir baris
- [abc] Kelas karakter — salah satu dari a, b, c
- ^[abc] Kelas negasi — apa pun kecuali a, b, c
- [a-z] Rentang — huruf kecil
- \<, \> Batas kata (GNU)
- \( \), \1 Grup tangkapan dan back-reference

### Contoh BRE

```
grep '^#' file.conf # lines starting with #
grep 'errors$' file.log # lines ending with error
grep '^$' file.txt # blank lines
grep 'col[ou]r' file.txt # match color or colour
```

## Regex Extended

### Extended Regular Expressions (ERE)

- + Satu atau lebih dari elemen sebelumnya
- ? Nol atau satu dari elemen sebelumnya
- {n} Tepat n pengulangan
- {n,m} Antara n dan m pengulangan
- (a|b) Alternasi — cocokkan a atau b
- () Pengelompokan (tanpa backslash)

### Contoh ERE

```
grep -E '[0-9]{3}-[0-9]{4}' f # phone number pattern
grep -E '(error|warn|fatal)' f # multiple patterns
grep -E '[A-Z][a-z]+' f # capitalized words
grep -P 'd(1,3)\.d(1,3)' f # Perl regex: IP fragments
```

## Baris Konteks

### Contoh Konteks

```
grep -B 3 "error" app.log # 3 lines before match
grep -A 5 "FAIL" test.log # 5 lines after match
grep -C 2 "crash" kern.log # 2 lines before and after
grep --group-separator="---" -C 1 "err" f # custom separator
```

### Flag Konteks

- B N Tampilkan N baris sebelum setiap kecocokan
- A N Tampilkan N baris setelah setiap kecocokan
- C N Tampilkan N baris sebelum dan sesudah (konteks)

--group-separator=stx Pemisah antar grup kecocokan (default --)

--color=auto Sorot kecocokan di terminal

## Pencarian Rekursif

### Contoh Rekursif

```
grep -r "TODO" # recursive from current dir
grep -rn "FIXME" src/ # recursive with line numbers
grep -r --include="*.py" "import" . # only .py files
grep -r --exclude="*.log" "error" # skip .log files
grep -r --exclude-dir=node_modules "require" .
```

### Flag Rekursif

- r / --recursive Cari direktori secara rekursif
- R Seperti -r tetapi mengikuti symlink
- include=glob Cari hanya file yang cocok dengan glob
- exclude=glob Lewati file yang cocok dengan glob
- exclude-dir=dir Lewati direktori yang cocok dengan nama
- include-dir=dir Hanya cari direktori yang cocok dengan nama

## Menghitung & Mendaftar

### Contoh Hitung & Daftar

```
grep -c "error" *.log # count matches per file
grep -l "TODO" src/*.py # list files with TODOs
grep -l "test" src/*.py # files missing "test"
grep -o "http[^"]*" page.html # extract matching parts only
grep -c '' file.txt # count total lines (like wc -l)
```

### Flag Output

- c Tampilkan jumlah baris yang cocok per file
- l Tampilkan hanya nama file dengan kecocokan
- L Tampilkan hanya nama file tanpa kecocokan
- o Tampilkan hanya bagian yang cocok dari baris
- H / -h Tampilkan / sembunyikan prefiks nama file
- Z Output null-delimited (untuk xargs -0)

## Kecocokan Terbalik

### Balik & Kecualikan

```
grep -v '^#' config.conf # remove comment lines
grep -v '^$' file.txt # remove blank lines
grep -v -e "debug" -e "trace" app.log # exclude two patterns
grep -v "pattern" f | grep "other" # chain: NOT A, then B
```

## Strategi Filter

- v Balik pencocokan — pilih baris yang tidak cocok
- v dengan -e Kecualikan beberapa pola
- pipe chain Rangkaikan grep untuk filter kompleks
- grep -v '^\$' | grep -v '^\$' Hapus baris kosong dan komentar
- v dengan -c Hitung baris yang tidak cocok

## Beberapa Pola

### Contoh Beberapa Pola

```
grep -e "error" -e "warning" app.log
grep -E "error|warning|fatal" app.log
grep -f patterns.txt file.txt # patterns from file
grep -v -e "GET" -e "POST" access.log
```

### Opsi Pola

- e pattern Tentukan pola (gunakan beberapa kali)
- f file Baca pola dari file (satu per baris)
- E 'a|b|c' Alternasi ERE untuk beberapa pola
- F String tetap — tanpa regex, pencocokan lebih cepat
- G Regex dasar (mode default)
- P Regex kompatibel Perl (PCRE)

## Performa

### Tips Performa

- F (fgrep) Mode string tetap — tercepat untuk string literal
- LC\_ALL=C grep Bypass locale untuk speedup 2-10x pada data ASCII
- include/--exclude Kurangi file yang dicari sebelum dibuka
- m N Berhenti setelah N kecocokan per file
- q Mode senyap — keluar pada kecocokan pertama (untuk script)
- ripgrep (rg) Pengganti drop-in; lebih cepat di repo besar

### Contoh Performa

```
LC_ALL=C grep -F "exact string" huge.log
grep -r -m 1 "needle" /var/log/ # stop after first hit
grep -rq "pattern" ; && echo "found" # boolean test
grep -r --include="*.go" "func main" .
```

## Pola Umum

### One-Liner

```
grep -rn "TODO\|FIXME\|HACK" src/ # find code markers
grep -oP '(?<=)[^]+(?:=) ' f # extract quoted strings
grep -E '^\\s*$' f | wc -l # count blank lines
grep -c '*.py' | sort -t: -k2 -rn # sort files by line count
grep -rn --include="*.yaml" "password" . # audit for secrets
```

### Resep

Alamat IP grep -oE '[0-9]{1,3}(\.[0-9]{1,3}){3}'

Alamat email grep -oE '[a-zA-Z0-9\_%.+@-z-]+@[-z-]+'

URL grep -oE 'https?://[^\s]+'

Basis di antara penanda grep -A999 'START' f | grep -B999 'END'

Kecocokan unik grep -oE 'pattern' f | sort -u

Hitung per pola grep -c 'pat1' f; grep -c 'pat2' f