

Referensi Cepat AWK

Pattern, field, array, fungsi, pemrosesan teks

Dasar

Menjalankan AWK

```
awk '{ print }' file.txt # print every line
awk '{ print $1 }' file.txt # print first field
awk -F: '{ print $1 }' /etc/passwd # custom delimiter
awk -f script.awk file.txt # run from file
cmd | awk '{ print $2 }' # pipe input
```

Struktur Program

awk 'pattern { action }' Bentuk dasar — action berjalan saat pattern cocok

BEGIN { ... } Berjalan sekali sebelum memproses input

END { ... } Berjalan sekali setelah semua input diproses

Tanpa pattern Action berjalan untuk setiap baris

Tanpa action Action default adalah **{ print }**

Pattern & Action

Jenis Pattern

```
awk '/error/' file.txt # regex match
awk '$3 > 100' file.txt # comparison
awk 'NR >= 5 && NR <= 10' file.txt # line range
awk '/start/,/end/' file.txt # range pattern
```

Referensi Pattern

/regex/	Cocokkan baris dengan regex
\$1 ~ /pat/	Field cocok dengan regex
\$1 !~ /pat/	Field tidak cocok dengan regex
expr1, expr2	Range: dari kecocokan pertama ke kedua
expr1 && expr2	Logical AND
expr1 expr2	Logical OR
!expr	Logical NOT

Variabel

Variabel Bawaan

NR	Nomor record (baris) saat ini
NF	Jumlah field dalam record saat ini
FS	Separator field input (default: whitespace)
OFS	Separator field output (default: spasi)
RS	Separator record input (default: newline)
ORS	Separator record output (default: newline)
FILENAME	Nama file input saat ini
FNR	Nomor record dalam file saat ini

Variabel Pengguna

```
awk '{ total += $1 } END { print total }' file.txt
awk -v threshold=50 '$1 > threshold' file.txt
awk 'BEGIN { count = 0 } /pat/ { count++ }
END { print count }' file.txt
```

Field

Akses Field

\$0	Seluruh baris saat ini
\$1, \$2, ...	Field pertama, kedua, ...
\$NF	Field terakhir
\$(NF-1)	Field kedua dari terakhir

Separator Field

```
awk -F, '{ print $2 }' data.csv # comma
awk -F'\t' '{ print $1 }' data.tsv # tab
awk 'BEGIN { FS = "[,;]" } { print $1 }' f # multi-char
awk 'BEGIN { OFS = "," } { print $1, $3 }' f # output sep
```

Control Flow

Kondisional & Loop

```
awk '{ if ($1 > 50) print "high"; else print "low" }' f
awk '{ for (i = 1; i <= NF; i++) print $i }' f
awk '{ i = 1; while (i <= NF) { print $i; i++ } }' f
awk '/skip/ { next } { print }' f # skip matching lines
```

Pernyataan Kontrol

if (cond) { ... } else { ... }	Kondisional
for (i = 0; i < n; i++) { ... }	For loop gaya C
for (key in array) { ... }	Iterasi key array
while (cond) { ... }	While loop
do { ... } while (cond)	Do-while loop
next	Lanjut ke record input berikutnya
exit	Hentikan pemrosesan, jalankan blok END

Fungsi

Fungsi Buatan Pengguna

```
awk 'function max(a, b) {
    return (a > b) ? a : b
}
{ print max($1, $2) }' file.txt
```

Fungsi Numerik

int(x)	Potong ke bilangan bulat
sqrt(x)	Akar kuadrat
sin(x), cos(x)	Fungsi trigonometri
log(x), exp(x)	Logaritma natural dan eksponen
rand()	Float acak antara 0 dan 1
srand(seed)	Inisialisasi generator angka acak

Array

Array Asosiatif

```
awk '{ count[$1]++ }
END { for (k in count) print k, count[k] }' f
awk '{ arr[NR] = $0 }
END { for (i = NR; i >= 1; i--) print arr[i] }' f
```

Operasi Array

arr[key] = val	Set elemen
arr[key]	Ambil elemen (dibuat otomatis saat diakses)
key in arr	Cek apakah key ada
delete arr[key]	Hapus satu elemen
delete arr	Hapus seluruh array
for (k in arr)	Iterasi key (tidak berurutan)
length(arr)	Jumlah elemen (gawk)

Fungsi String

Referensi String

length(s)	Panjang string
substr(s, start, len)	Substring (indeks mulai dari 1)
index(s, target)	Posisi target dalam s (0 jika tidak ditemukan)
split(s, arr, sep)	Pisah string ke array
sub(pat, repl, s)	Ganti kecocokan pertama
gsub(pat, repl, s)	Ganti semua kecocokan
match(s, pat)	Posisi kecocokan regex (set RSTART, RLENGTH)
tolower(s) / toupper(s)	Konversi huruf
sprintf(fmt, ...)	Format string (seperti printf di C)

Contoh String

```
awk '{ gsub(/old/, "new"); print }' f # sed-like replace
awk '{ print toupper($0) }' f # uppercase all
awk '{ print substr($0, 1, 40) }' f # truncate to 40
```

I/O

Output

```
awk '{ print $1, $2 }' f # space-separated
awk '{ printf "%s,%d\n", $1, $2 }' f # formatted output
awk '{ print $1 > "out.txt" }' f # redirect to file
awk '{ print $1 >> "out.txt" }' f # append to file
```

Referensi I/O

print	Cetak dengan ORS (newline secara default)
printf fmt, ...	Cetak terformat (tanpa newline di akhir)
print > file	Redirect output ke file
print >> file	Tambahkan output ke file
print cmd	Pipe output ke perintah
getline < file	Baca satu baris dari file
cmd getline var	Baca output perintah ke variabel
close(file)	Tutup file atau pipe

Pola Umum

One-Liner

```
awk '{ sum += $1 } END { print sum }' f # sum column
awk 'END { print NR }' f # count lines
awk '!seen[$0]++' f # remove dupes
awk 'NF' f # remove blanks
awk '{ print NF }' f # fields per line
```

Resep

CSV ke TSV	awk -F, 'BEGIN{OFS="\t"} {\$1=\$1; print}'
Jumlahkan kolom 2 N baris teratas	awk '{ s += \$2 } END { print s }'
Hitung frekuensi	awk 'NR <= 10' (seperti head) awk '{ c[\$1]++ } END { for (k in c) print k, c[k] }'
Di antara penanda	awk '/BEGIN/,/END/'
Cetak field ke-N	awk '{ print \$N }' (ganti N)