

# REFERENSI CEPAT REGULAR EXPRESSION

Pola, quantifier, grup, lookahead, flag

## Pola Dasar

### Metakarakter

- `.` Karakter apa pun (kecuali newline)
- `^` Awal string / baris
- `$` Akhir string / baris
- `*` 0 atau lebih dari sebelumnya
- `+` 1 atau lebih dari sebelumnya
- `?` 0 atau 1 dari sebelumnya (opsional)
- `\` Escape metakarakter

### Pencocokan Literal

```
hello # matches "hello" exactly
a.c   # matches "abc", "alc", "a-c", etc.
\\.txt # matches literal ".txt"
```

## Character Class

### Ekspresi Bracket

- `[abc]` Cocok dengan a, b, atau c
- `[^abc]` Cocok dengan apa pun kecuali a, b, c
- `[a-z]` Huruf kecil
- `[A-Z]` Huruf besar
- `[0-9]` Digit
- `[a-zA-Z0-9]` Alfanumerik

### Class Singkatan

- `\d` Digit `[0-9]`
- `\D` Non-digit `[^0-9]`
- `\w` Karakter kata `[a-zA-Z0-9_]`
- `\W` Karakter non-kata
- `\s` Whitespace `[\t\n\r\f]`
- `\S` Non-whitespace

## Quantifier

### Quantifier Greedy

- `*` 0 atau lebih (greedy)
- `+` 1 atau lebih (greedy)
- `?` 0 atau 1 (greedy)
- `{n}` Tepat n kali
- `{n,}` n kali atau lebih
- `{n,m}` Antara n dan m kali

### Quantifier Lazy

- `*?` 0 atau lebih (lazy / non-greedy)
- `+?` 1 atau lebih (lazy)
- `??` 0 atau 1 (lazy)
- `{n,m}?` Antara n dan m (lazy)

Quantifier lazy mencocokkan sesedikit mungkin karakter

### Greedy vs Lazy

```
<.+> # greedy: "<b>bold</b>"
<.+?> # lazy: " <b>"
```

## Anchor

- `^` Awal string (atau baris dengan flag ``m``)
- `$` Akhir string (atau baris dengan flag ``m``)
- `\b` Batas kata
- `\B` Bukan batas kata
- `\A` Awal string (tidak terpengaruh ``m``)
- `\Z` Akhir string (tidak terpengaruh ``m``)

### Contoh Anchor

```
"Hello" # starts with "Hello"
world$  # ends with "world"
\bword\b # "word" as whole word
\bword\B # "word" inside another word
```

## Grup & Alternasi

### Capturing Group

```
(abc) # capture group: match "abc"
(a|b|c) # alternation: a or b or c
(cat|dog) # match "cat" or "dog"
\d{3}-\d{4} # groups: "123-4567"
```

### Jenis Grup

- `(pattern)` Capturing group
- `(?:pattern)` Non-capturing group
- `(?P<name>pat)` Named group (Python)
- `(?<name>pat)` Named group (JS, .NET)
- `\1 \2` Backreference ke grup 1, 2
- `a|b` Alternasi: a atau b

## Lookahead & Lookbehind

- `(?=pattern)` Positive lookahead
- `(?!pattern)` Negative lookahead
- `(?<=pattern)` Positive lookbehind
- `(?<!pattern)` Negative lookbehind

### Contoh Lookaround

```
\d+(?= USD) # digits followed by " USD"
\d+(?! USD) # digits NOT followed by " USD"
(?<=\$)\d+ # digits preceded by "$"
(?<!\$)\d+ # digits NOT preceded by "$"
```

Lookaround mencocokkan posisi tanpa mengonsumsi karakter

## Pola Umum

- `\d{1,3}(\.\d{1,3}){3}` Alamat IPv4 (dasar)
- `[w.+-]+@[w.-]+\.[w.]+` Email (dasar)
- `https?://[w./\-?&#]=1+` URL (dasar)
- `\((?[\d3]{1})?[-.\s]?[\d3]{1}[-.\s]?[\d4]{1}` Nomor telepon AS
- `\d{4}-\d{2}-\d{2}` Tanggal (YYYY-MM-DD)
- `#[0-9a-fA-F]{6}` Kode warna hex

Ini adalah pola yang disederhanakan; penggunaan produksi mungkin perlu validasi lebih ketat

## Flag

- `g` Global: temukan semua kecocokan, bukan hanya yang pertama
- `i` Pencocokan case-insensitive
- `m` Multiline: `^` / `$` cocok batas baris
- `s` Dotall: `.` juga cocok newline
- `x` Verbose: abaikan whitespace, izinkan komentar
- `u` Unicode: dukungan Unicode penuh

## Penggunaan Flag per Bahasa

```
/pattern/gi # JavaScript
re.compile(r"pat", re.I | re.M) # Python
grep -iE "pattern" # grep (extended)
```