

REFERENSI CEPAT JSON

Sintaks, tipe data, objek, array, jq

Sintaks

Aturan

{ } Objek (pasangan key-value tidak berurutan)

[] Array (daftar nilai berurutan)

"key": value Key harus berupa string dengan tanda kutip ganda

No trailing comma Item terakhir tidak boleh memiliki koma

No comments JSON tidak mengizinkan komentar

Contoh Minimal

```
{
  "name": "Alice",
  "age": 30,
  "active": true
}
```

Tipe Data

Enam Tipe Nilai

"string" Teks UTF-8 dengan tanda kutip ganda

42 / 3.14 Number (integer atau bilangan pecahan)

true / false Boolean

null Null (tidak ada nilai)

{ } Objek

[] Array

Urutan Escape String

\" Tanda kutip ganda

**** Backslash

\n \t Newline, tab

\uXXXX Unicode escape (hex)

Objek

Sintaks Objek

```
{
  "id": 1,
  "name": "Widget",
  "tags": ["new", "sale"]
}
```

Aturan

Keys Harus berupa string dengan tanda kutip ganda yang unik

Values Tipe JSON valid mana pun

Order Urutan key tidak dijamin

Nesting Objek bisa mengandung objek lain

Array

Sintaks Array

```
[1, "two", true, null, {"key": "val"}]
```

Array Tipe Campuran

```
{
  "matrix": [[1, 2], [3, 4]],
  "empty": []
}
```

Aturan

Ordered Elemen mempertahankan urutan penyisipan

Mixed types Item array bisa berbeda tipe

Indexing Berbasis nol (di sebagian besar bahasa)

Nesting

Struktur Bersarang

```
{
  "user": {
    "name": "Alice",
    "address": { "city": "Boston" },
    "scores": [95, 88, 72]
  }
}
```

Pola Akses

obj.user.name Notasi titik (JavaScript)

obj["user"]["name"] Notasi kurung

obj.user.scores[0] Indeks array dalam objek bersarang

Validasi Schema

Contoh JSON Schema

```
{
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "age": { "type": "integer", "minimum": 0 }
  },
  "required": ["name"]
}
```

Kata Kunci Schema

type string, number, integer, boolean, object, array, null

required Array nama properti yang wajib ada

properties Mendefinisikan properti objek yang diharapkan

enum Batasi ke kumpulan nilai tetap

minLength / maxLength Batasan panjang string

minimum / maximum Batasan rentang angka

Dasar jq

Filter Umum

. Identity — lewatkan input

.key Akses key objek

.key.nested Akses key bersarang

[0] Elemen array pertama

[] Iterasi semua elemen array

select(.age > 20) Filter berdasarkan kondisi

map(.name) Transformasi setiap elemen

length Panjang array atau string

keys Key objek sebagai array

Contoh jq

```
echo '{"a":1}' | jq '.a' # 1
echo '[1,2,3]' | jq 'map(. * 2)' # [2,4,6]
cat data.json | jq '.users[].name'
cat data.json | jq '.[] | select(.active)'
```

Pola Umum

Respons API

```
{
  "status": 200,
  "data": [{"id": 1, "name": "Alice"}],
  "meta": {"total": 42, "page": 1}
}
```

File Konfigurasi

```
{
  "host": "localhost",
  "port": 8080,
  "debug": false,
  "features": ["auth", "logging"]
}
```

Tips

Validate Gunakan jsonlint atau python -m json.tool

Pretty print jq .file.json atau python -m json.tool

JSONL Satu objek JSON per baris (newline-delimited)

JSON5 / JSONC Ekstensi yang mengizinkan komentar dan trailing comma