

# Referensi Cepat curl

HTTP request, header, auth, form, debugging

## Penggunaan Dasar

### Request Sederhana

```
curl https://example.com # GET request
curl -o file.html https://url # save to file
curl -O https://url/file.tar.gz # save with remote name
curl -L https://url # follow redirects
```

### Flag Umum

<b>-s</b>	Mode silent (tanpa progress)
<b>-S</b>	Tampilkan error dalam mode silent
<b>-f</b>	Gagal diam-diam pada error HTTP
<b>-L</b>	Ikuti redirect
<b>-o file</b>	Tulis output ke file
<b>-O</b>	Simpan dengan nama file remote
<b>-C -</b>	Lanjutkan unduhan yang terputus
<b>--max-time 30</b>	Timeout setelah 30 detik

## HTTP Method

### GET & HEAD

```
curl https://api.example.com/users
curl -I https://example.com # HEAD (headers only)
curl -i https://example.com # include response headers
```

### POST

```
curl -X POST https://api.example.com/users \
-H "Content-Type: application/json" \
-d '{"name":"Jo","email":"jo@ex.com"}
```

### PUT & PATCH & DELETE

```
curl -X PUT https://api.example.com/users/1 \
-d '{"name":"Updated"}'
curl -X PATCH https://api.example.com/users/1 \
-d '{"email":"new@ex.com"}'
curl -X DELETE https://api.example.com/users/1
```

## Header

### Mengatur Header

```
curl -H "Content-Type: application/json" URL
curl -H "Accept: text/html" URL
curl -H "X-Custom: value" URL
curl -H "Header1: v1" -H "Header2: v2" URL
```

### Header Response

<b>-i</b>	Sertakan header response dalam output
<b>-I</b>	Ambil header saja (HEAD)
<b>-D file</b>	Dump header response ke file
<b>-w '%{http_code}'</b>	Cetak HTTP status code

## Autentikasi

### Basic & Token Auth

```
curl -u user:pass https://api.example.com
curl -H "Authorization: Bearer TOKEN" URL
curl -u user:pass --digest URL
curl --negotiate -u : URL # Kerberos/SPNEGO
```

### Metode Auth

<b>-u user:pass</b>	Autentikasi Basic
<b>--digest</b>	HTTP Digest auth
<b>--negotiate</b>	Kerberos/SPNEGO auth
<b>--ntlm</b>	Autentikasi NTLM
<b>-n</b>	Gunakan kredensial ~/.netrc

## Data & Form

### Mengirim Data

```
curl -d "key=val&key2=val2" URL # form urlencoded
curl -d @data.json URL # data from file
curl --data-raw '{"raw":"json"}' URL
curl --data-urlencode "q=hello world" URL
```

### Upload File

```
curl -F "file=@photo.jpg" URL
curl -F "file=@doc.pdf;type=application/pdf" URL
curl -F "field=value" -F "file=@img.png" URL
```

### Multipart vs URL-Encoded

<b>-d</b>	application/x-www-form-urlencoded
<b>-F</b>	multipart/form-data
<b>--json</b>	Shorthand: set Content-Type + Accept ke JSON
<b>-T file</b>	Upload file via PUT

## SSL/TLS

### Opsi Sertifikat

```
curl --cacert ca.pem URL # custom CA bundle
curl --cert client.pem URL # client certificate
curl --cert client.pem --key key.pem URL
curl -k URL # skip TLS verify (dev only)
```

### Flag TLS

<b>-k / --insecure</b>	Lewati verifikasi sertifikat TLS
<b>--cacert file</b>	Gunakan CA sertifikat kustom
<b>--cert file</b>	Sertifikat client
<b>--key file</b>	Private key client
<b>--tlsv1.2</b>	Paksa minimum TLS 1.2
<b>--tlsv1.3</b>	Paksa minimum TLS 1.3

## Output & Debugging

### Verbose & Trace

```
curl -v URL # verbose output
curl --trace dump.txt URL # full trace to file
curl --trace-ascii - URL # trace to stdout
curl -w "\n%{http_code}\n" URL # custom output format
```

### Variabel Write-Out

<b>%{http_code}</b>	HTTP response status code
<b>%{time_total}</b>	Total waktu dalam detik
<b>%{time_connect}</b>	Waktu untuk membuat koneksi
<b>%{size_download}</b>	Byte yang diunduh
<b>%{speed_download}</b>	Kecepatan unduhan rata-rata
<b>%{redirect_url}</b>	URL redirect (jika ada)
<b>%{ssl_verify_result}</b>	Hasil verifikasi SSL (0 = OK)

### Contoh Write-Out

```
curl -s -o /dev/null -w \
"code: %{http_code}\ntime: %{time_total}s\n" \
https://example.com
```

## Pola Umum

### Alur Kerja API

```
# GET JSON and pipe to jq
curl -s https://api.example.com/data | jq '.items[]'
# POST JSON with auth
curl -s -H "Authorization: Bearer $TOKEN" \
--json '{"key":"val"}' https://api.example.com
```

## Pola Unduhan

```
# Download with progress bar
curl -# -O https://releases.example.com/v2.tar.gz
# Resume interrupted download
curl -C - -O https://releases.example.com/v2.tar.gz
# Download multiple files
curl -O https://url/file1 -O https://url/file2
```

## Helper Scripting

```
# Check if URL is reachable
curl -sf -o /dev/null https://example.com && echo OK
# Save cookies and reuse
curl -c cookies.txt -b cookies.txt URL
# Rate-limit request
curl --limit-rate 100k URL
```

## Proxy & Jaringan

### Pengaturan Proxy

```
curl -x http://proxy:8080 URL
curl -x socks5://proxy:1080 URL
curl --proxy-user user:pass -x http://proxy:8080 URL
curl --noproxy "*.local,localhost" URL
```

## DNS & Resolve

<b>--resolve host:port:addr</b>	Paksa resolusi DNS ke addr
<b>--dns-servers 8.8.8.8</b>	Gunakan server DNS kustom
<b>--interface eth0</b>	Gunakan interface jaringan tertentu
<b>-4 / -6</b>	Paksa IPv4 / IPv6

## Konfigurasi & Lanjutan

### File Konfigurasi

```
# ~/.curlrc - default options
--silent
--location
--max-time 30
```

```
# Use config file explicitly
curl -K myconfig.txt URL
```

### Flag Berguna

<b>--retry 3</b>	Coba ulang pada error transien
<b>--retry-delay 2</b>	Jeda antar percobaan ulang (detik)
<b>--compressed</b>	Minta dan dekompresi gzip/br
<b>--limit-rate 100k</b>	Throttle kecepatan transfer
<b>-Z</b>	Transfer paralel (curl 7.66+)
<b>--create-dirs</b>	Buat direktori path untuk -o