

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

```
-s Mode silent (tanpa progress)
-S Tampilkan error dalam mode silent
-f Gagal diam-diam pada error HTTP
-L Ikuti redirect
-o file Tulis output ke file
-O Simpan dengan nama file remote
-C Lanjutkan unduhan yang terputus
--max-time 30 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 -X "Custom: value" URL
curl -H "Header1: v1" -H "Header2: v2" URL
```

Header Response

```
-i Sertakan header response dalam output
-I Ambil header saja (HEAD)
-D file Dump header response ke file
-w '%{http_code}' 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

```
-u user:pass Autentikasi Basic
--digest HTTP Digest auth
--negotiate Kerberos/SPNEGO auth
--ntlm Autentikasi NTLM
-n 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

```
-d application/x-www-form-urlencoded
-F multipart/form-data
--json Shorthand: set Content-Type + Accept ke JSON
-T file 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

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

```
%{http_code} HTTP response status code
%{time_total} Total waktu dalam detik
%{time_connect} Waktu untuk membuat koneksi
%{size_download} Byte yang diunduh
%{speed_download} Kecepatan unduhan rata-rata
%{redirect_url} URL redirect (jika ada)
```

```
%{ssl_verify_result} 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 socks://proxy:1080 URL
curl --proxy-user user:pass -x http://proxy:8080 URL
curl --noproxy "*" local,localhost URL
```

DNS & Resolve

```
--resolve host:port:addr Paksa resolusi DNS ke addr
--dns-servers 8.8.8.8 Gunakan server DNS kustom
--interface eth0 Gunakan interface jaringan tertentu
-4 / -6 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

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