

# API TESTING MENGUNAKAN POSTMAN

---

ARDINI YUANITA LUBIS - 714222073



# API (Application Programming Interface)

API merupakan antarmuka yang dapat menghubungkan satu aplikasi dengan aplikasi lainnya. Perantara bagi beberapa aplikasi atau klien dan server, baik pada satu platform yang sama maupun lintas platform, agar bisa saling berkomunikasi.



# Manfaat API

1 Mempermudah Pembuatan Aplikasi Fungsional

---

2 Efisiensi Pengembangan Aplikasi

---

3 Meringankan Beban Server

---

# Cara Kerja API



- Aplikasi Mengakses API
- API Melakukan Request ke Server
- Server Memberi Respon ke API
- API Menyampaikan Respon ke Aplikasi

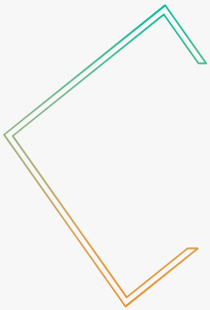


**POSTMAN**

---



## Postman adalah



Sebuah platform kolaboratif yang digunakan oleh pengembang perangkat lunak untuk membangun, menguji, dan mendokumentasikan API (Application Programming Interface)



# ELEMEN POSTMAN

Permintaan (Request)

01

05

Koleksi (Collection)

02

Respon (Response)

Variabel (Variables)

04

Lingkungan (Environments)

03

06


Skrip (Script)



# Method HTTP pada Request


 **GET** => Mendapatkan Informasi


 **DELETE** => Menghapus Informasi

 **POST** => Menambahkan Informasi

 **HEAD** => Memeriksa Informasi

 **PUT** => Mengganti Informasi

 **OPTIONS** => Mengembalikan informasi

 **PATCH** => Memperbaharui informasi spesifik



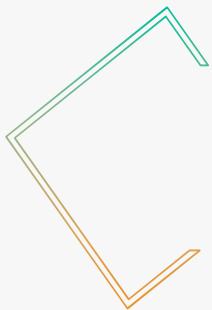


## Kode Respon di POSTMAN

- **200 OK** : Request berhasil dan response berisi data yang diminta.
- **201 Created** : Request berhasil dan server telah membuat sumber daya baru.
- **400 Bad Request** : Request tidak valid atau tidak dapat diproses oleh server karena kesalahan klien.
- **401 Unauthorized** : Request membutuhkan autentikasi, dan kredensial yang diberikan tidak valid.
- **403 Forbidden** : Server memahami Request , tetapi tidak mengizinkan akses ke sumber daya yang diminta
- **404 Not Found** : Sumber daya yang diminta tidak ditemukan di server.
- **500 Internal Server Error** : Terjadi kesalahan pada server yang mencegah pemenuhan Request.



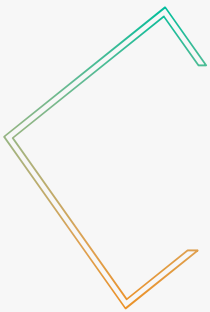
## Environments (Lingkungan)



Merujuk pada pengaturan variabel yang dapat disesuaikan untuk digunakan dalam permintaan API. Environments memungkinkan pengguna untuk mengelola nilai-nilai yang dapat berubah, seperti URL, token otentikasi, atau parameter lainnya, tergantung pada environments yang sedang digunakan (misalnya, pengembangan, pengujian, produksi).



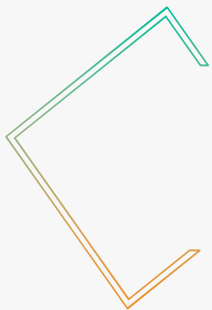
## Variabels



Variabel digunakan untuk menyimpan dan mengganti nilai-nilai yang dapat digunakan dalam permintaan dan respons API. Variabel memungkinkan pengguna untuk mengelola data dinamis, seperti URL, token otentikasi, atau parameter lainnya, sehingga mempermudah penggunaan dan pengujian API.



# Collection



Merupakan kumpulan request yang digunakan untuk mengorganisir dan menjalankan dengan mudah. Collection merupakan salah satu fitur utama yang memungkinkan pengguna untuk mengelompokkan request API dalam satu tempat dan mengaturnya sesuai kebutuhan.



## Script

- **Pre-request Script** : adalah skrip yang dieksekusi sebelum request dikirimkan ke server. Skrip ini memungkinkan pengguna untuk memodifikasi request, menetapkan nilai variabel, menambahkan header custom, menghitung tanda tangan otentikasi, atau melakukan tugas-tugas lainnya sebelum request dikirimkan.
- **Test Script**: adalah skrip yang dieksekusi setelah menerima respons dari server. Skrip ini memungkinkan pengguna untuk memeriksa dan memvalidasi respons API, menguraikan data respons, menguji kondisi, memeriksa kode status, dan melakukan tugas-tugas lainnya untuk memverifikasi bahwa request berjalan dengan benar.



## KELEBIHAN

- Kemudahan Penggunaan
- Pengujian dan Debugging yang kuat
- Automatisasi dengan skrip
- Pengelolaan dan Organisasi yang Mudah
- Kolaborasi Tim
- Integrasi dengan Alat dan Layanan Lain
- Dokumentasi Interaktif



## KEKURANGAN

- Ketergantungan pada Antarmuka Pengguna
- Pembatasan Fitur Gratis
- Kurangnya Dukungan Tersedia
- Keterbatasan lingkungan dan integrasi
- Kecepatan Respons API Eksternal

The image shows the Postman web interface with several annotations pointing to specific features:

- Collections:** Points to the left sidebar where API collections are listed.
- Environment:** Points to the 'Environments' section in the left sidebar.
- Method:** Points to the HTTP method dropdown (currently set to 'GET') in the request editor.
- Variable:** Points to the URL input field, specifically to a variable placeholder like `http://localhost:9999/api/products/2`.
- Kode Resoonse:** Points to the 'Test Results' tab at the bottom, which displays the JSON response from the API.

The main interface shows a 'GET' request to `http://localhost:9999/api/products/2` with a status of 200 OK. The response body is displayed in JSON format:

```
1 {
2   "code": 200,
3   "data": [
4     {
5       "id": 17,
6       "code": "TNK834",
7       "name": "Open Systems Interconnection",
8       "price": "100000"
9     },
10    {
11      "id": 19,
12      "code": "TNK834",
13      "name": "Open Systems Interconnection",
14      "price": "100000"
15    },
16    {
17      "id": 20,
```





**SHARING**

---



**TERIMA KASIH**