



# PENGUJIAN PERANGKAT LUNAK

**PENGAJAR**  
ASRI MASPUPAH

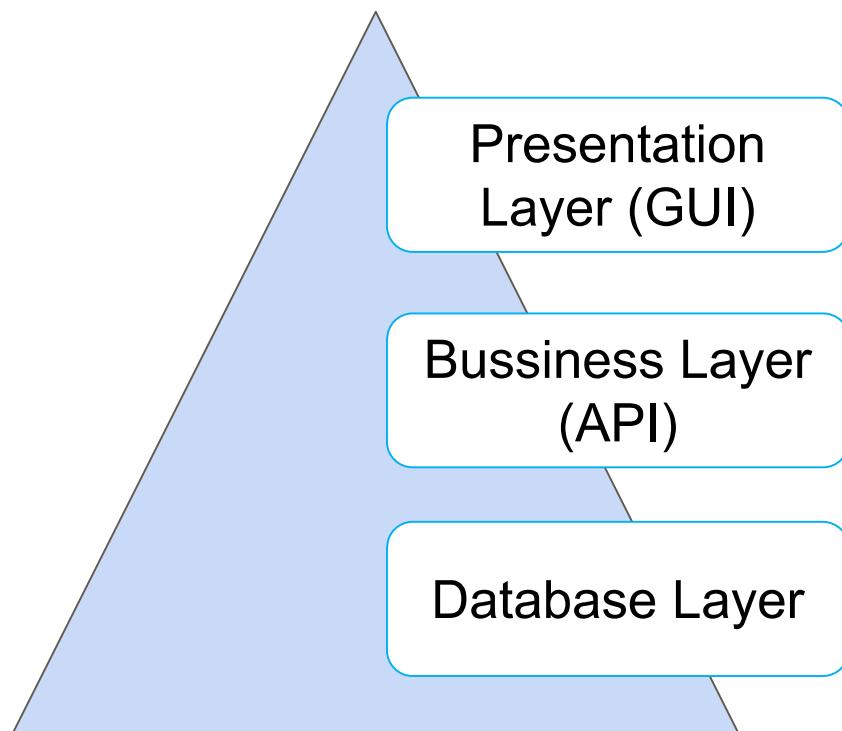
## JADWAL KULIAH

SELASA, 07.00 - 09.30 (3B) & 09.50 - 12.20 (3A)

API Testing Menggunakan Postman

6 & 13 Mei 2025

# Web Application Testing Type



- Umumnya web application memiliki tiga layer: **front-end** (user interface), **business logic**, and a **database**.
- **Pengujian frontend** melibatkan **validasi** bagian aplikasi yang **terlihat oleh end-users**.
- **backend testing**, berurusan dengan semua elemen yang **tidak terlihat oleh end-users**. Hal ini berarti bahwa **validasi** terhadap application's **database** and **API** yang merepresentasikan business logic.



MATERI

# Apa itu API ?



**API** singkatan dari **application program interface**. API sebagai lapisan perantara dalam *software system* yang bertanggung jawab terhadap **data communication** antara the **data source** dan the **Graphical User Interface** (GUI) yang dilihat oleh user

API adalah **business layer** dari software yang membuat **koneksi** antara **presentation layer** dan **data layer**.

Anda dapat menganggap **API** sebagai **mediator** antara user / client dengan resources / web services yang ingin didapatkan

# THE ABCs OF APIs

What are APIs and what's the big deal about them?  
We'll break it down for you in an easy-to-digest way  
that'll leave you hungry for more.



## Analogi API dapat disebut sbg Waiters antara Chef dan Customer

API mengambil *requests* Anda dari device dan mengirimkannya kepada server, kemudian menerima *response* dari server dan mengembalikan respon tersebut kepada Anda.

**APIs** memberikan kemudahan kepada software developer dalam pengembangan produk

Silahkan tonton tautan youtube berikut untuk lebih memahami analogi API sebagai Waiters antara Cheft dan Customer <https://youtu.be/81p4v5J9FqE>

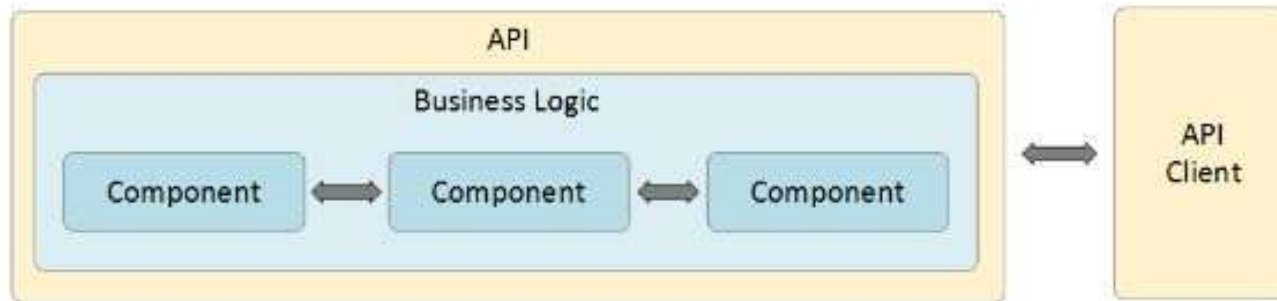
Sumber : <https://toolsqa.com/postman/api-testing-with-postman/>

# Lebih Lanjut tentang API

API secara teknis memiliki arti seperangkat definisi dan protokol untuk membangun dan mengintegrasikan aplikasi. Sering disebut sebagai kontrak antara penyedia informasi dan pengguna informasi.

API dapat **menghubungkan berbagai komponen** dalam aplikasi, seperti feature, data source, dll. Dalam arsitektur **microservices**, **APIs menyatukan berbagai modules berbeda** yang masing-masing memiliki **database** dan **interface terpisah**.

Hal ini berarti bahwa API dapat berfungsi secara **independent** dan menyediakan **higher uptime**.



# Contoh API

Bayangkan sebuah aplikasi perbankan dimana pengguna dapat melihat daftar rekening bank di antarmuka depan. Di balik itu, bisa ada beberapa API yang digunakan aplikasi untuk menampilkan data di halaman.

## **Misalnya:**

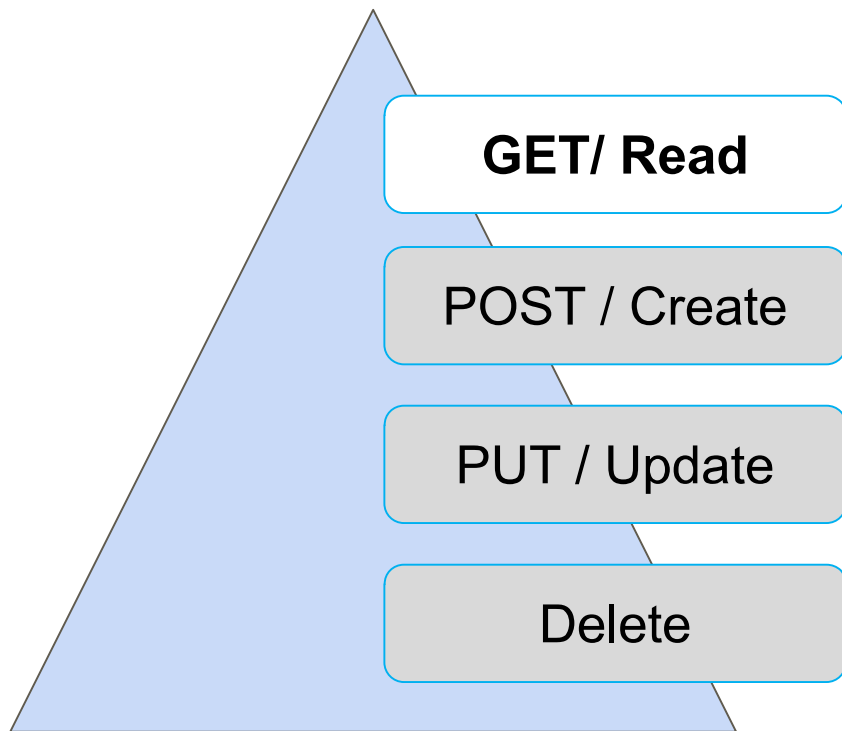
- 1 API untuk mengambil semua rekening pengguna yang aktif
- 1 API untuk mengambil rekening pengguna yang tidak aktif
- 1 API untuk menghapus sebuah akun

# Rest API

**RESTful API** adalah Jenis arsitektur API yang menggunakan HTTP request untuk mengakses dan menggunakan data. Manipulasi data menggunakan request GET, PUT, POST, dan DELETE yang berarti membaca, memperbaharui, membuat baru dan menghapus resource/data yang diinginkan.

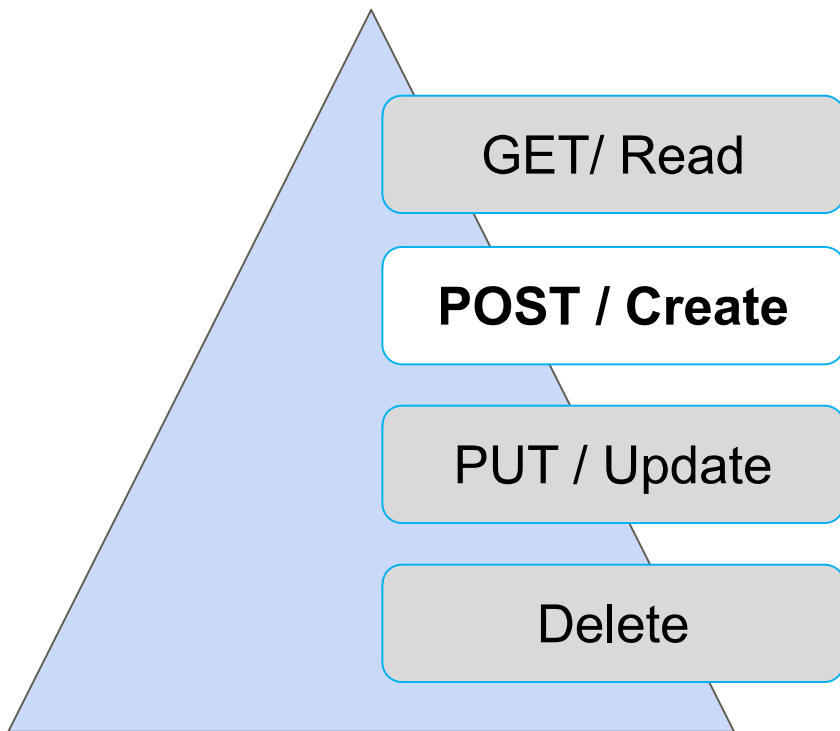


# HTTP METHOD TYPE pada API



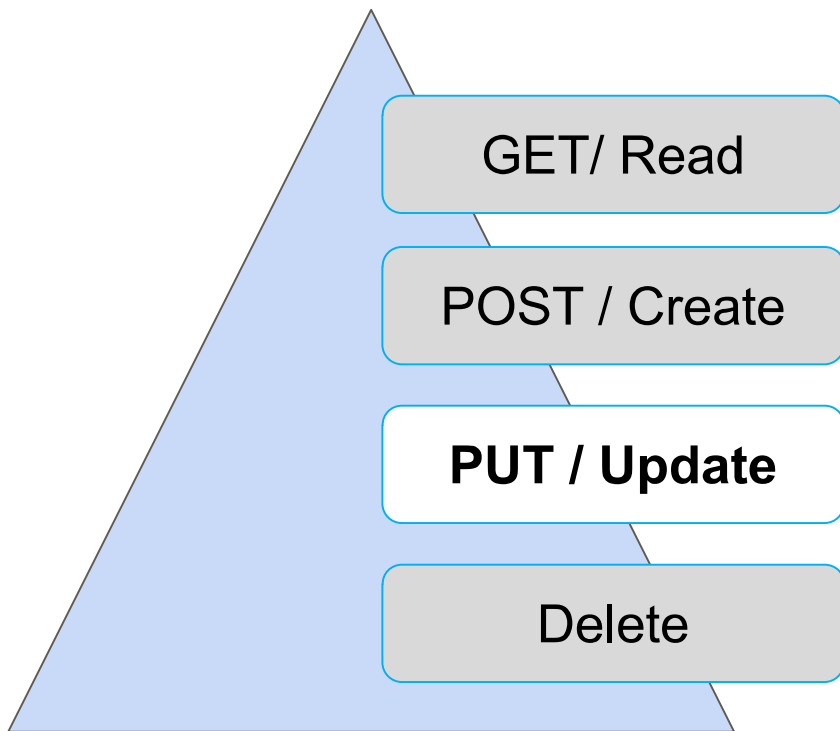
- ❏ HTTP GET method digunakan untuk **read** (atau retrieve) sebuah representasi dari resource.
- ❏ Tujuannya untuk mengambil atau mendapatkan informasi dari server melalui URL yang telah ditentukan
- ❏ Pada case “happy” (atau non-error) path, GET return representasi informasi dalam bentuk XML atau JSON dan HTTP response code bernilai 200 (OK).
- ❏ Pada error case, sebagian besar return response code 404 (NOT FOUND) atau 400 (BAD REQUEST) dengan tidak mengembalikan informasi data yg diinginkan.

# HTTP METHOD TYPE pada API



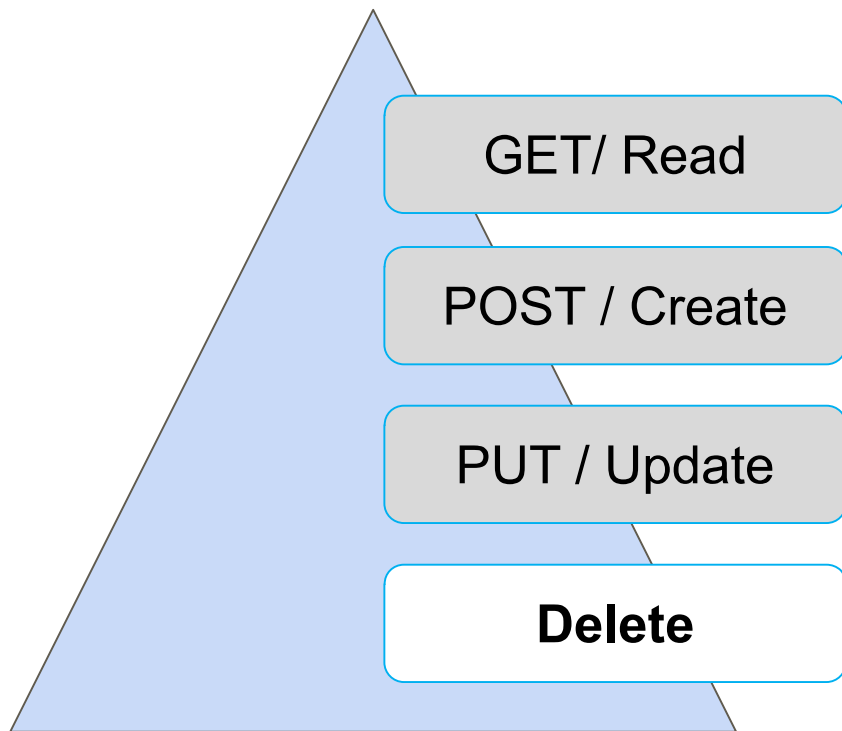
- ❑ Istilah POST paling sering digunakan untuk **create** new resource. Secara khusus, digunakan untuk membuat subordinate resources.
- ❑ Dengan kata lain, ketika membuat new resource, POST kepada parent dan service menangani asosiasi new resource dengan parent, penetapan ID (new resource URI), dll.
- ❑ Tujuannya untuk mengirimkan data ke server seperti upload file atau transfer data atau menambahkan baris data baru ke tabel backend dari input web form.
- ❑ Pada case successful creation, API return HTTP status code 201, mengembalikan Location header dengan tautan ke newly created resource with 201 HTTP status.

# HTTP METHOD TYPE pada API



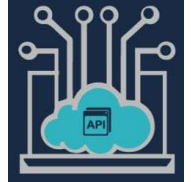
- PUT paling sering digunakan untuk **update** capability, PUT-ing ke resource URI yang diketahui dengan request body berisi newly-update representation dari original resource.
- Pada case successful update, return 200 (atau 204 if tidak mengembalikan konten apa pun dalam body) from a PUT.

# HTTP METHOD TYPE pada API



- ❏ Delete paling sering digunakan untuk **menghapus data** sesuai dengan parameter yang didefinisikan pada URI
- ❏ Pada case successful deletion, API return HTTP status 200 (OK) bersama dengan response body yang mungkin merupakan representasi dari item yang dihapus.
- ❏ Dengan demikian, jika kamu ingin menghapus sebuah akun user dalam aplikasi, maka aplikasi akan melakukan API call DELETE method dan akun user akan terhapus dalam database.

# API Testing



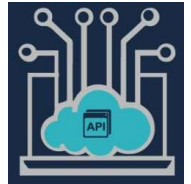
Setiap kali Anda melakukan panggilan ke server, hal ini dianggap sebagai **API request**.

API testing melibatkan **pengiriman API request** dan **pemeriksaan response-nya** dengan menggunakan software untuk call API. Terdapat metode spesifik dalam **API testing**.

Pada **HTTP method**, setiap panggilan memiliki **response dan status code**.

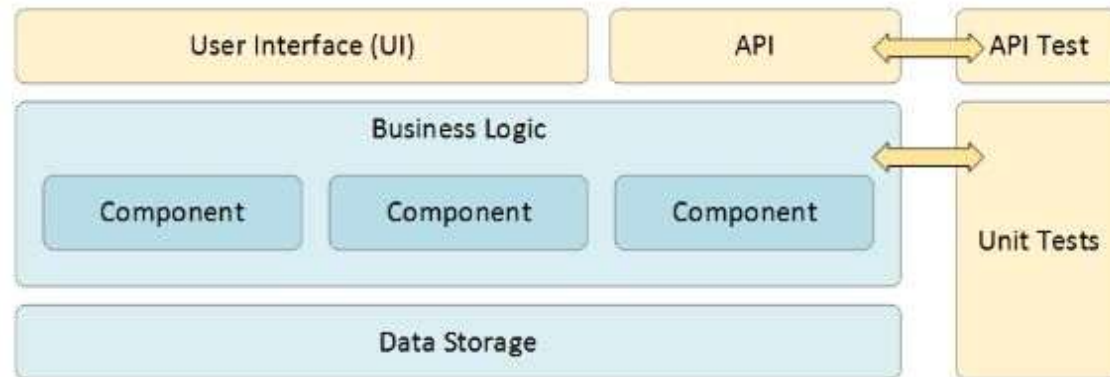
**HTTP status codes membantu mengidentifikasi penyebab masalah** ketika web page atau resource lainnya tidak di-load dengan benar

Ketika developer mendefinisikan API, dia mengidentifikasi HTTP method mana yang dapat dilakukan.



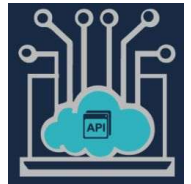
# API Testing

API testing dapat dilakukan sebelum GUI application siap dipakai. Hal ini karena, **API Contract** dapat dibuat sebelum proses development selesai dikerjakan.



API Testing seharusnya dilakukan dengan baik agar dapat mencegah isu yang terjadi di masa akan datang, karena dapat mengarahkan kinerja aplikasi yang sesuai harapan.

Pendekatan pengujian dapat dilakukan dengan manual dan automation. Alat API testing yang sering digunakan adalah POSTMAN (manual) dan Rest Assured (Automation)



# API Testing

API Testing pada dasarnya menggunakan **black box testing** dengan fokus pengujian terhadap final output dari system under test. Pengujian dilakukan ketika API telah selesai dibangun dan siap berinteraksi dengan user interface.

Tujuan API testing untuk menguji **business logic layer dari system's architecture** dan dilakukan oleh tim QA dengan menjalankan setiap API di atas bangunan tertentu dengan tujuan untuk melayani tujuan tertentu.

API Testing termasuk **pengujian unit** dari sebuah sistem dengan fokus kepada fungsionalitas modul tertentu, karena kinerja API dapat terisolasi dari bagian sistem lain.

Selain itu, API testing juga dapat dilakukan secara **end-to-end testing** dengan menguji setiap API yang berhubungan dengan bisnis secara lengkap dari ujung ke ujung

# Manfaat API Testing

- Anda dapat mulai menguji sebelum front-end siap dan memperbaiki bug yang terkait dengan logika dengan biaya yang lebih murah.
- Anda dapat menemukan bug yang dapat merusak produk lain dan proses bisnis, jika API Anda bersifat publik.
- Dapat meningkatkan cakupan pengujian dan akurasi.
- Anda dapat menjalankan **pengujian regresi** untuk memastikan bahwa perubahan terbaru tidak merusak API Anda dan sistem yang menggunakannya.

Pengujian regresi tentu dengan menggunakan otomasi pengujian



# Kemampuan Software Tester yang harus dimiliki pada API Testing

1. harus bisa menggunakan semua web methods, seperti GET, POST, DELETE, dll
2. Validasi response, response time, dan error code
3. Harus dapat memvalidasi XML and JSON body dengan Json parsers
4. Harus tau cara menggunakan mekanisme otentikasi OAuth dan OAuth2
5. Pahami Load dan Security testing pada web services
6. Harus dapat membaca dan memahami API documentations
7. Harus dapat membuat sejumlah kasus uji dan skenario yang baik
8. Harus mahir SQL queries untuk memvalidasi API dan DB data elements
9. Menjadi ahli penggunaan alat pada pilihan Anda. Seperti SOAPUI dan **Postman** bukanlah alat otomatisasi, atau juga **Rest Assured**, Rest Sharp, Node modules yang merupakan open source libraries untuk API testing.

# What Does an API Test look like?

1. **The API URL** → This is the HTTP request for invoking a particular API.  
E.g. API for login functionality. If it's URL structure is Mobile/User/Login and your domain URL is <https://domain.com>, then API URL to invoke the API from browser/tools will be:  
`http://domain.com/Mobile/user/Login`
2. **HTTP Method** → in the case RESTfull API's, they use HTTP method to classify the APIs based on type of call made to the server  
For example, POST (send data), GET (get data), DELETE (remove data) etc.
3. **Payload** → this defines the structure or the model of input data which is to be supplied to the API. For example, if we can use the below structur for login API:

```
{
  username: string
  password: string
  device_id: string
  token: string
  app_language: string
}
```

# What Does an API Test look like?

4. **Request Sample Data** → this contains the actual input data which is passed on to the API as payload. You can have as many test cases as you want based in this sample data. For login scenario, the sample data might look like:

```
{  
  "username": "andi",  
  "password": "tes123",  
  "device_id": "F4325-C3423-32453",  
  "token": "f5kl54jkl54g4oi23443",  
  "app_language": "en"  
}
```

4. **Expected - Response Code** → This represents the response code of the API request For Example: 200 OK should be the response code of successful request, 400 as Bad request, 401 is Unauthorized etc. For more detail about response code you can see link <https://www.softwaretestinghelp.com/rest-api-response-codes/>. It would be useful if you could note down the Response Code of the API from the output console of the API testing tools.

# What Does an API Test look like?

## 6. Expected - Response Result Message

For each input, there might be different API output success message. You need to figure out the corresponding response message and document then in your test cases. Some common message for Login API will be:

SUCCESS, INACTIVE\_ACCOUNT, INVALID\_PASSWORD, USER\_NOT\_FOUND,  
INVALID\_DEVICE\_ID, ERROR

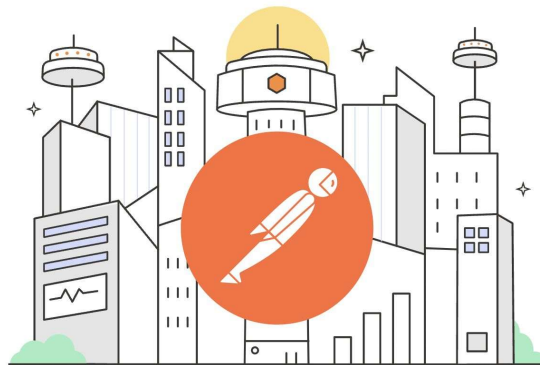
## 7. Expected - Response Result Sample → This is the output data for each of the input data combinations. You will need this data to be able to validate against the actual output from the API result. A sample for a successful login operation is:

```
{  
  "result": "SUCCESS",  
  "data": { "id": 7093,  
            "email": "ozan@test.com",  
            "create_at": "2023-03-20"}  
}
```

# Pengenalan Alat Pengujian API - Postman

Postman adalah platform API yang digunakan untuk membangun dan menggunakan API. Postman menyederhanakan setiap langkah dari siklus API dan mempermudah kolaborasi sehingga dapat create API yang lebih baik dan lebih cepat performanya.

Sebagai QA, kita menggunakan Postman untuk memvalidasi dan memverifikasi respon API.



# JSON SCHEMA

JSON Schema adalah kosakata / vocabulary yang memungkinkan kita untuk menambahkan anotasi dan memvalidasi dokumen JSON.

## **Benefit dari JSON Schema :**

- Menjelaskan format data yang ada.
- Menyediakan dokumentasi yang jelas dapat dibaca manusia dan mesin.
- Memvalidasi data yang berguna untuk:
  - Automation Testing
  - Memastikan kualitas data yang dikirimkan oleh klien.

membuat JSON Schema dapat menggunakan website online  
<https://www.liquid-technologies.com/online-json-to-schema-converter>

# JSON SCHEMA - EXAMPLE

## Overview

The example we use in this guide is a product catalog that stores its data using JSON objects, like the following:

```
1  {  
2    "productId": 1,  
3    "productName": "A green door",  
4    "price": 12.50,  
5    "tags": [ "home", "green" ]  
6  }  
7
```

Each product in the catalog has:

- `productId` : an identifier for the product
- `productName` : the product name
- `price` : the cost to the consumer
- `tags` : an optional array of identifying tags

Lebih lanjut : <https://json-schema.org/docs>

# Referensi

1. API Testing di dalam course UDEMY
2. Postman Tutorial from basic to advance pada link berikut  
["https://toolsqa.com/postman/postman-tutorial/"](https://toolsqa.com/postman/postman-tutorial/)
3. Rest API pada tautan ["https://www.redhat.com/en/topics/api/what-is-a-rest-api"](https://www.redhat.com/en/topics/api/what-is-a-rest-api)
4. 7 Ways to Validate that Your APIs Are Working Correctly pada tautan  
["https://www.apiscience.com/blog/7-ways-to-validate-that-your-apis-are-working-correctly/"](https://www.apiscience.com/blog/7-ways-to-validate-that-your-apis-are-working-correctly/)
5. HTTP REQUEST Methods-GET, POST, PUT, PATCH, DELETE. (A Walkthrough With JavaScript's Fetch API) pada tautan  
["https://medium.com/swlh/restful-api-design-get-post-put-patch-delete-a-walkthrough-with-javascripts-fetch-api-e37a8416e2a0"](https://medium.com/swlh/restful-api-design-get-post-put-patch-delete-a-walkthrough-with-javascripts-fetch-api-e37a8416e2a0)





# KEGIATAN PRAKTIKUM

# Praktikum (Pemahaman API Testing dengan pendekatan manual & Otomasi)

- 1) Pahami instruksi praktikum pada tautan <https://drive.google.com/file/d/12etoiYTrgdl84Tyob3sfhKcXgvC0nhdb/view?usp=sharing> yang meliputi persiapan environment API testing dan pelaksanaan API testing dengan pendekatan manual.
- 2) Goals :
  - a. paham melakukan persiapan environment API testing,
  - b. melakukan pengujian API (test case development, test execution dan test reporting)
- 2) Buatlah laporan pemahaman praktikum dalam bentuk slide (individu)
- 3) Tautan pengumpulan:
  - a. Progres (11/04/2025) : folder progress
  - b. Final (18/05/2025) : folder final
  - c. Perbaikan (22/05/2025) : folder progress

# Praktikum(Pemahaman Unit Testing pada White Box dengan menggunakan build tools)

Perangkat praktikum dapat diakses pada folder “TGS. 4 API Testing dengan Pengujian Manual” yang berisi:

1. Detail Instruksi Praktikum
2. Matriks Pengujian
3. Sheet monitoring Tab “W10-W11”

Isi Sheet Monitoring untuk pemantauan progress pada tautan : [//bit.ly/40W84pU](https://bit.ly/40W84pU)