

Postman, Software untuk pengujian Restfull API

Saya ulas Kembali bahwa REST merupakan singkatan dari Representational State Transfer. Yang merupakan gaya arsitektur untuk merancang aplikasi yang saling terhubung. Dengan menggunakan HTTP sederhana untuk memungkinkan komunikasi antar mesin. Jadi, anda dapat menggunakan URL untuk memanipulasi beberapa informasi pengguna, REST mengirimkan permintaan HTTP seperti GET, POST, DELETE, dll ke URL untuk memanipulasi data.

Sebagai contoh, sebagai ganti membuat sebuah permintaan GET ke URL seperti:

`/deleteuser?id=10`

permintaannya akan menjadi seperti:

`DELETE /user/10`

Untuk tahu lebih apa itu Rest API, silahkan and baca modul praktikum lain yang telah saya upload pada CIS.

Postman merupakan salah satu alat paling populer untuk melakukan pengujian API. Postman muncul pada tahun 2012 sebagai proyek sampingan oleh Abhinav Asthana untuk menyederhanakan alur kerja API dalam pengujian dan pengembangan. API memungkinkan aplikasi perangkat lunak untuk berkomunikasi satu sama lain melalui panggilan API.

Dalam praktikum ini, Anda akan belajar:

- Mengapa menggunakan postman?
- Cara menggunakan postman
- Bekerja dengan GET Requests
- Bekerja dengan POST Requests
- Cara Parameterisasi Permintaan
- Bagaimana membuat testing dengan postman
- Bagaimana membuat collections

Mengapa Menggunakan Postman?

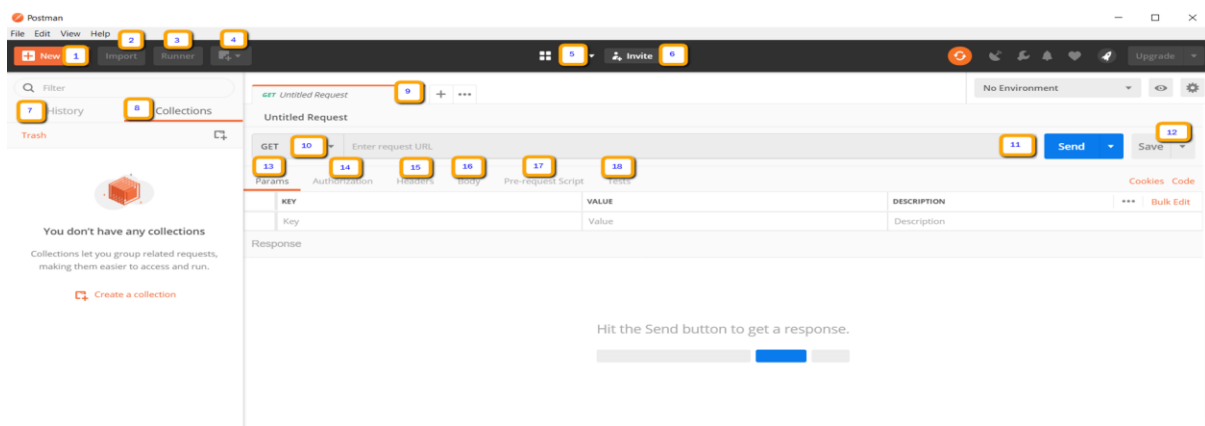
Dengan lebih dari 4 juta pengguna saat ini, Postman telah menjadi alat pilihan karena alasan berikut:

- ✓ **Aksesibilitas** - Untuk menggunakan Postman, seseorang hanya perlu masuk ke akun mereka sendiri sehingga memudahkan untuk mengakses file kapan saja, di mana saja selama aplikasi Postman dipasang di komputer.
- ✓ **Fitur collection** - Postman memungkinkan pengguna membuat collection untuk panggilan API mereka. Setiap collection dapat membuat subfolder dan beberapa permintaan. Ini membantu dalam mengatur suite tes Anda.

- ✓ **Kolaborasi** - Collection dan lingkungan dapat diimpor atau diekspor sehingga mudah untuk berbagi file. Tautan langsung juga dapat digunakan untuk berbagi collection.
- ✓ **Menciptakan environments** - Memiliki beberapa lingkungan bantu dalam pengulangan API tes yang lebih sedikit karena seseorang dapat menggunakan collection yang sama tetapi untuk lingkungan yang berbeda. Di sinilah parameterisasi akan terjadi yang akan kita bahas dalam pelajaran selanjutnya.
- ✓ **Pembuatan Tes** - Pos pemeriksaan seperti memverifikasi untuk status respons HTTP yang berhasil dapat ditambahkan ke setiap panggilan API yang membantu memastikan cakupan pengujian.
- ✓ **Pengujian Otomasi** - Melalui penggunaan Runner Collection atau Newman, pengujian dapat dijalankan dalam beberapa iterasi menghemat waktu untuk pengujian berulang.
- ✓ **Debugging** - Konsol Postman membantu untuk memeriksa data apa yang telah diambil sehingga memudahkan untuk men-debug tes.
- ✓ **Integrasi Berkelanjutan** - Dengan kemampuannya untuk mendukung integrasi berkelanjutan, praktik pengembangan dipertahankan.

Cara menggunakan postman

Di bawah ini adalah workspace Postman. Mari kita jelajahi berbagai fitur tools ini:



1. **New** - This is where you will create a new request, collection or environment.
2. **Import** - This is used to import a collection or environment. There are options such as import from file, folder, link or paste raw text.
3. **Runner** - Automation tests can be executed through the Collection Runner. This will be discussed further in the next lesson.
4. **Open New** - Open a new tab, Postman Window or Runner Window by clicking this button.
5. **My Workspace** - You can create a new workspace individually or as a team.
6. **Invite** - Collaborate on a workspace by inviting team members.
7. **History** - Past requests that you have sent will be displayed in History. This makes it easy to track actions that you have done.

8. **Collections** - Organize your test suite by creating collections. Each collection may have subfolders and multiple requests. A request or folder can also be duplicated as well.
9. **Request tab** - This displays the title of the request you are working on. By default, "Untitled Request" would be displayed for requests without titles.
10. **HTTP Request** - Clicking this would display a dropdown list of different requests such as GET, POST, COPY, DELETE, etc. In testing, the most commonly used requests are GET and POST.
11. **Request URL** - Also known as an endpoint, this is where you will identify the link to where the API will communicate with.
12. **Save** - If there are changes to a request, clicking save is a must so that new changes will not be lost or overwritten.
13. **Params** - This is where you will write parameters needed for a request such as key values.
14. **Authorization** - In order to access APIs, proper authorization is needed. It may be in the form of a username and password, bearer token, etc.
15. **Headers** - You can set headers such as content type JSON depending on the needs of the organization.
16. **Body** - This is where one can customize details in a request commonly used in POST request.
17. **Pre-request Script** - These are scripts that will be executed before the request. Usually, pre-request scripts for the setting environment are used to ensure that tests will be run in the correct environment.
18. **Tests** - These are scripts executed during the request. It is important to have tests as it sets up checkpoints to verify if response status is ok, retrieved data is as expected and other tests.

Bekerja dengan GET Requests

GET Requests digunakan untuk mengambil informasi dari URL yang diberikan. Tidak akan ada perubahan yang dilakukan ke **endpoint**.

Mahasiswa akan menggunakan URL berikut untuk semua contoh dalam praktikum ini:

`https://jsonplaceholder.typicode.com/users`

JSONPlaceholder adalah:

Fake Online REST API untuk Testing dan Prototyping yang melayani ~350M requests per bulan. Kita menggunakan resource ini karena pada praktikum ini kita fokus bagaimana mengakses/menguji Rest API yang gratis(opensource) dari client menggunakan tools Postman. Anda dapat melihat kumpulan Resource dan route Rest API pada link berikut:

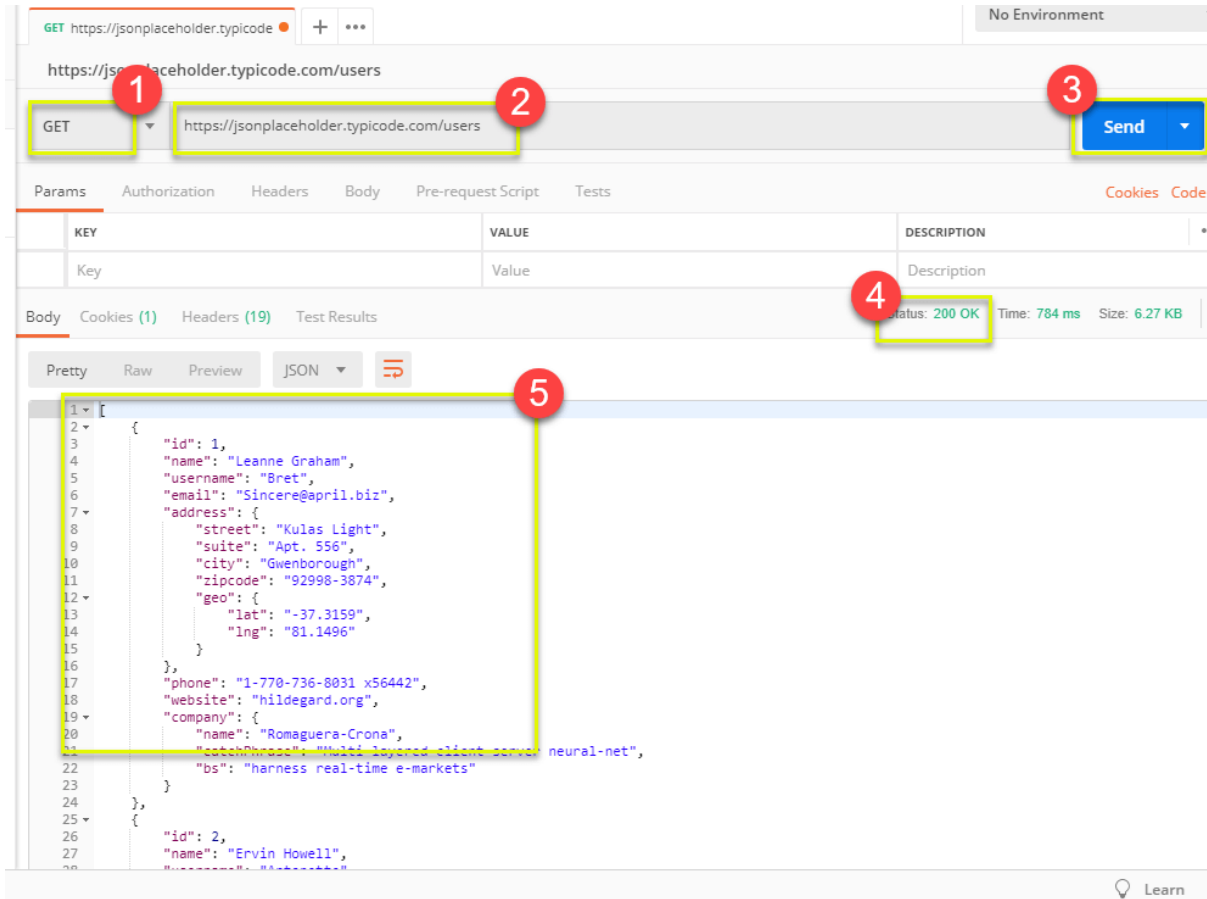
<https://jsonplaceholder.typicode.com/>

Langkah-langkah pada Postman:

- ✓ set permintaan HTTP Anda ke **GET**.

- ✓ Pada request URL Field, masukkan link berikut
- ✓ Klik **Send/Kirim**
- ✓ Anda akan melihat **200/Pesan OK**

Seharusnya ada 10 hasil pengguna pada body yang menunjukkan bahwa tes Anda telah berhasil dijalankan.

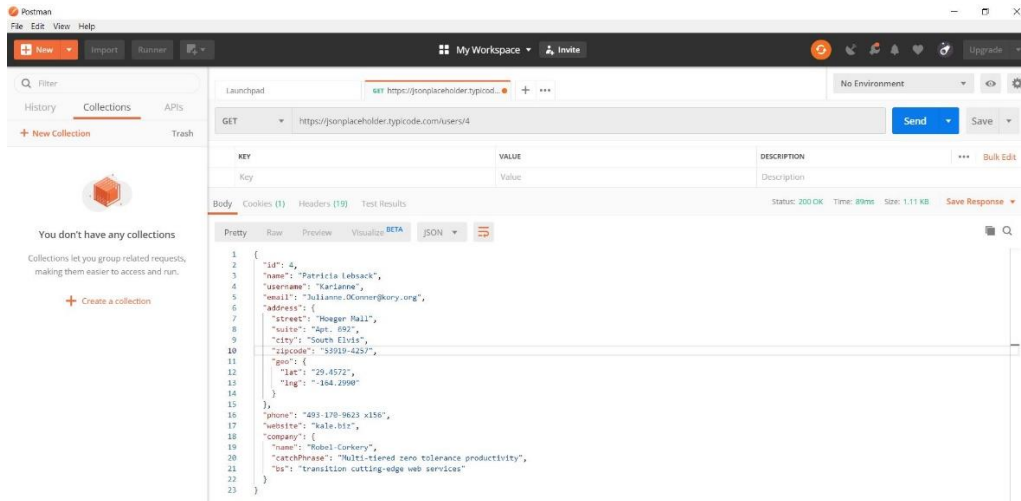


***Catatan:** Mungkin ada beberapa kasus yang GET Request nya mungkin tidak berhasil. Itu bisa karena request URL yang tidak valid atau diperlukan otentikasi.

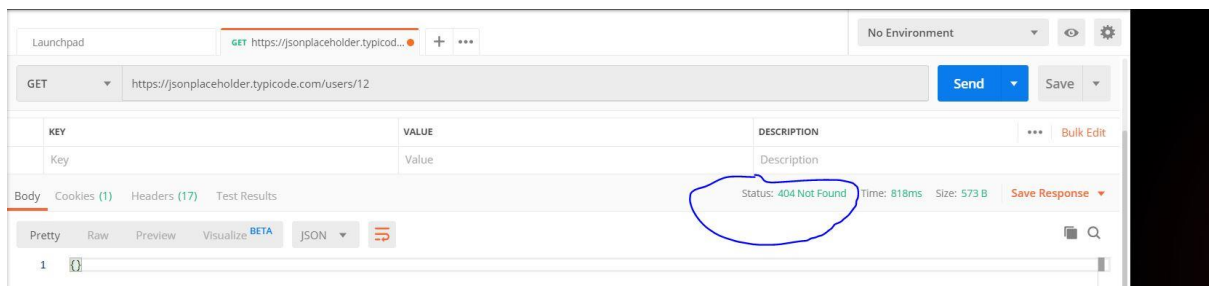
Bagaimana jika anda ingin mengambil data **users** dengan ID: 4?

Anda hanya perlu mengganti URL field menjadi seperti berikut:

```
https://jsonplaceholder.typicode.com/users/4
```



Bagaimana jika anda ingin mengambil data **users** dengan **ID: 12**? Apa yang akan terjadi?



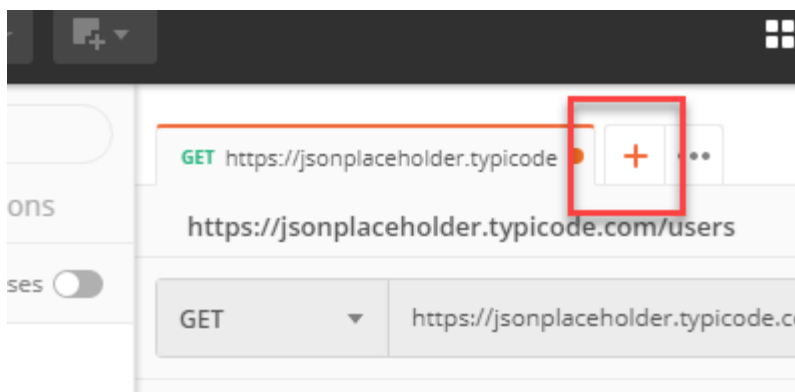
Result:

Pada workspace akan tertampil status code **404: Not Found**, artinya data yang kita **GET** tidak ada/tidak ditemukan. Ini dikarenakan **jsonplaceholder** endpoint hanya memiliki data users sejumlah 10, sementara data users yang kita request adalah data users pada **ID ke 12**.

Bekerja dengan POST Requests

Post Request berbeda dari GET Request karena pada POST Request ada manipulasi data dengan pengguna/anda menambahkan data ke endpoint. Menggunakan data yang sama dari langkah sebelumnya di GET Request, mari sekarang tambahkan pengguna dengan data anda sendiri.

Langkah 1) Klik tab baru untuk membuat **new Request**.

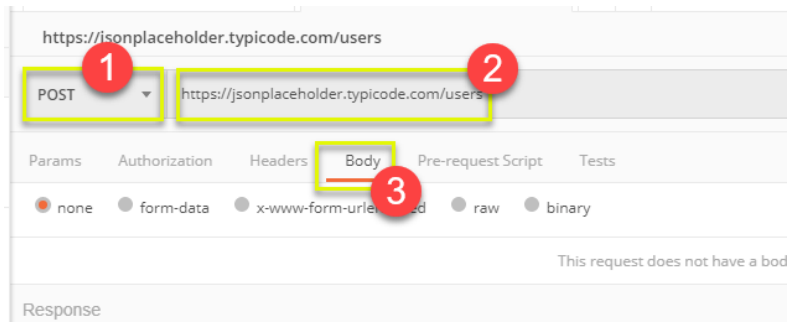


Langkah 2) Di tab baru

- Setel HTTP Request anda ke POST.
- Masukkan tautan yang sama di url permintaan:

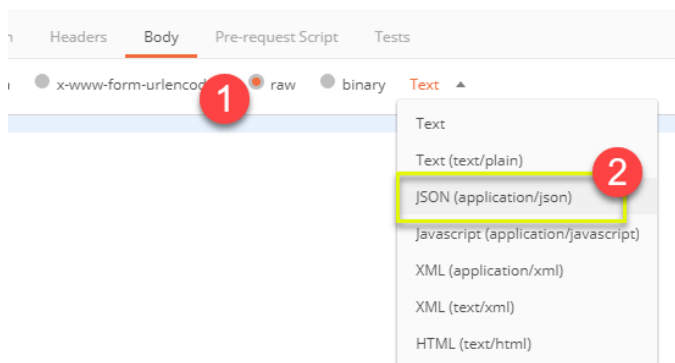
`https://jsonplaceholder.typicode.com/users`

- beralih ke tab body.



Langkah 3) Di Body,

- Klik Raw
- Pilih JSON



Langkah 4) Copy dan paste hanya satu hasil pengguna dari permintaan sebelumnya seperti di bawah ini.

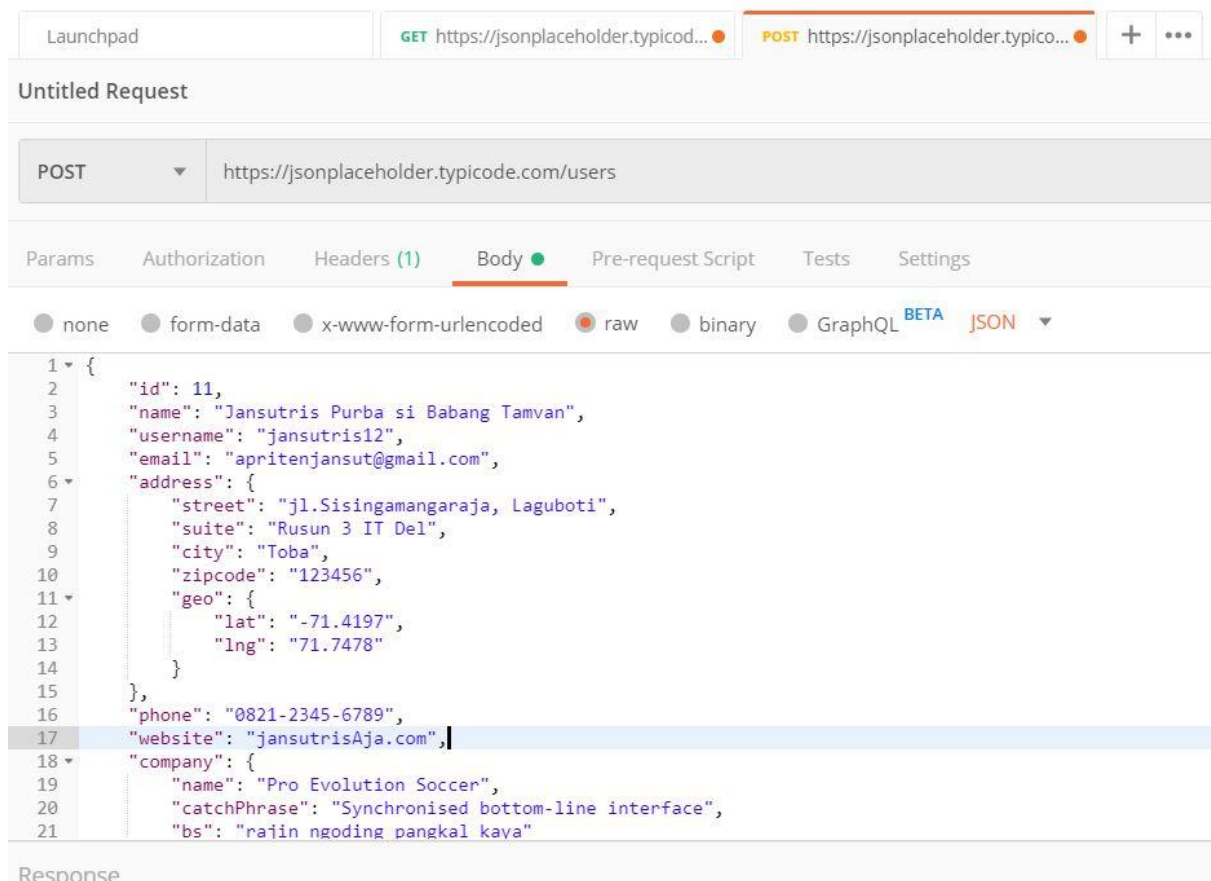
- Pastikan kode telah disalin dengan benar dengan kurung kurawal dan kurung.
- Ubah id ke 11 dan beri nama ke nama yang diinginkan. Anda juga dapat mengubah detail lainnya seperti alamat.

```
{
  "id": 11,
  "name": "Jansutris Purba si Babang Tamvan",
  "username": "jansutris12",
  "email": "apritenjansut@gmail.com",
  "address": {
    "street": "Jl.Sisingamangaraja, Laguboti",
```

```

    "suite": "Rusun 3 IT Del",
    "city": "Toba",
    "zipcode": "123456",
    "geo": {
      "lat": "-71.4197",
      "lng": "71.7478"
    },
    "phone": "0821-2345-6789",
    "website": "jansutrisAja.com",
    "company": {
      "name": "Pro Evolution Soccer",
      "catchPhrase": "Synchronised bottom-line interface",
      "bs": "rajin ngoding pangkal kaya"
    }
  }
}

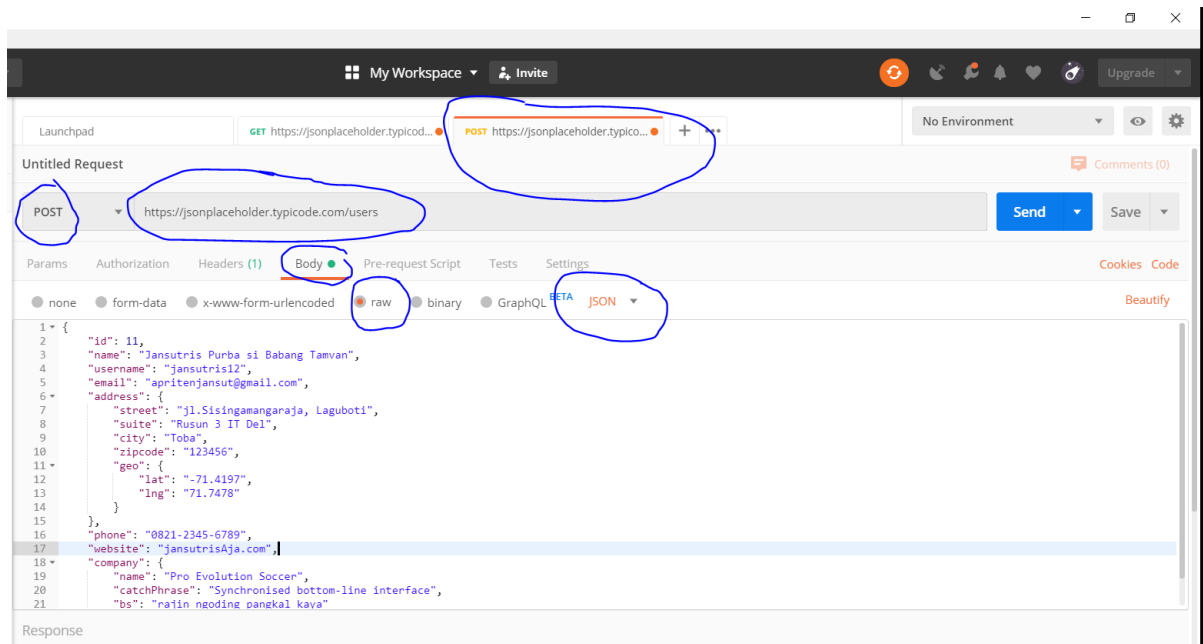
```



Catatan:

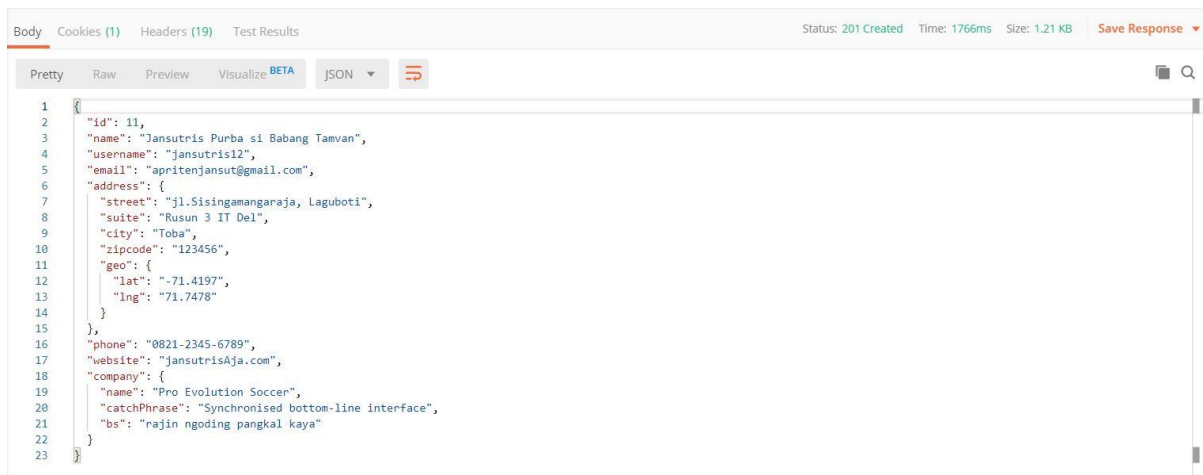
Request POST harus memiliki format yang benar untuk memastikan bahwa data yang diminta akan dibuat. Ini adalah praktik yang baik untuk menggunakan `Dapatkan dulu` untuk memeriksa format permintaan JSON. Anda dapat menggunakan tools online seperti:

<https://jsonformatter.curiousconcept.com/>



Langkah 5) Next,

- Klik Send/Kirim.
- Status: 201 Dibuat harus ditampilkan
- Data yang diposting muncul di Body.



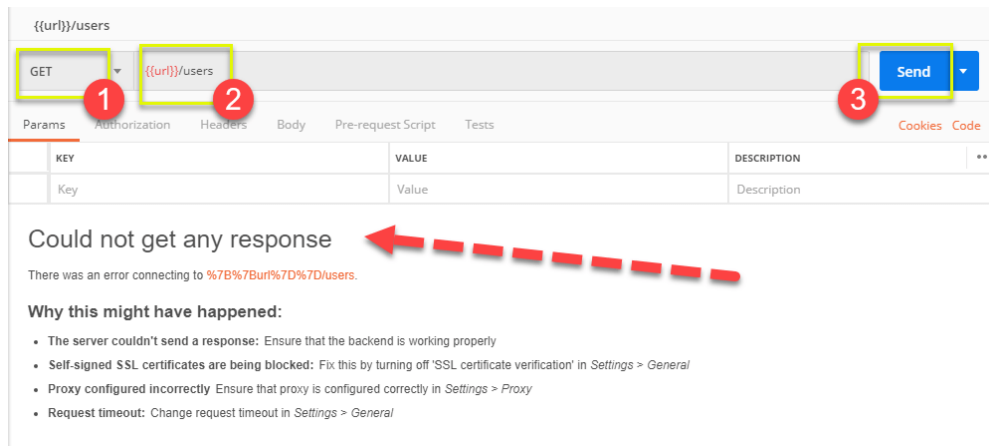
Cara Parameterisasi Permintaan

Parameterisasi Data adalah salah satu fitur paling berguna dari Postman . Alih-alih membuat permintaan yang sama dengan data yang berbeda, Anda bisa menggunakan variabel dengan parameter. Data ini dapat dari file data atau variabel lingkungan. Parameterisasi membantu untuk menghindari pengulangan tes yang sama dan iterasi dapat digunakan untuk pengujian otomatis.

Parameter dibuat melalui penggunaan *double curly brackets*/kurung keriting ganda: **{{sample}}**. Mari kita lihat contoh penggunaan parameter dalam Request sebelumnya:

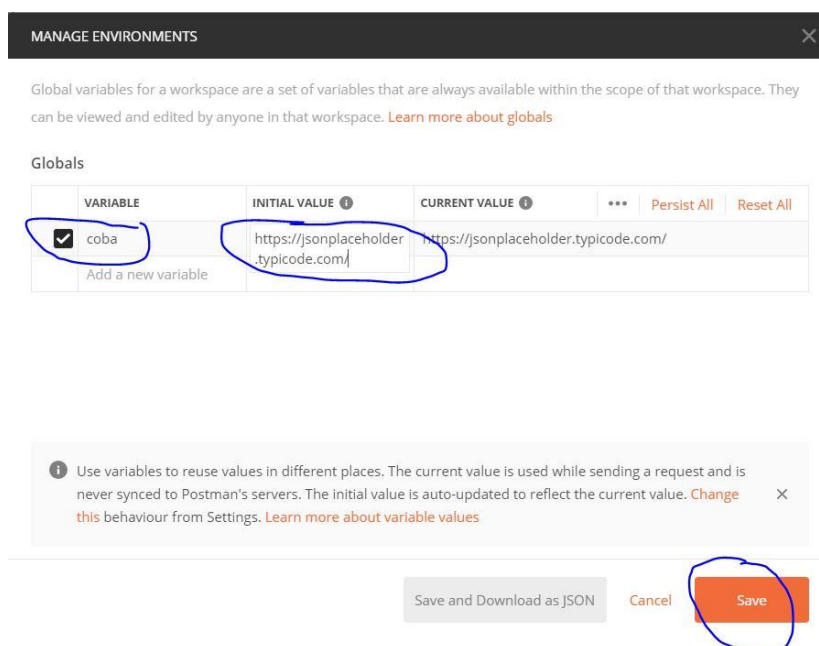
Langkah 1)

- Setel permintaan HTTP Anda ke GET
- Masukkan tautan ini: <https://jsonplaceholder.typicode.com/users>. Ganti bagian pertama tautan dengan parameter seperti {{url}}. Url permintaan sekarang harus {{url}} / users.
- Klik Send/kirim.
- Pasti tidak ada respon karena kita belum menetapkan sumber parameter kita.

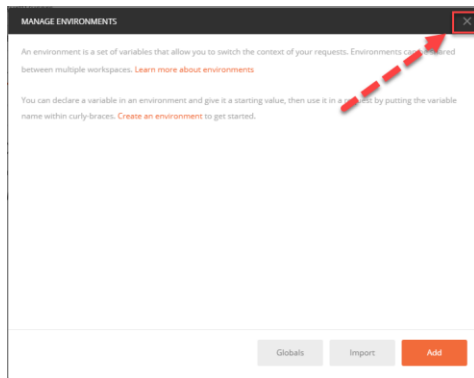


Langkah 2) Untuk menggunakan parameter yang Anda butuhkan untuk mengatur lingkungan

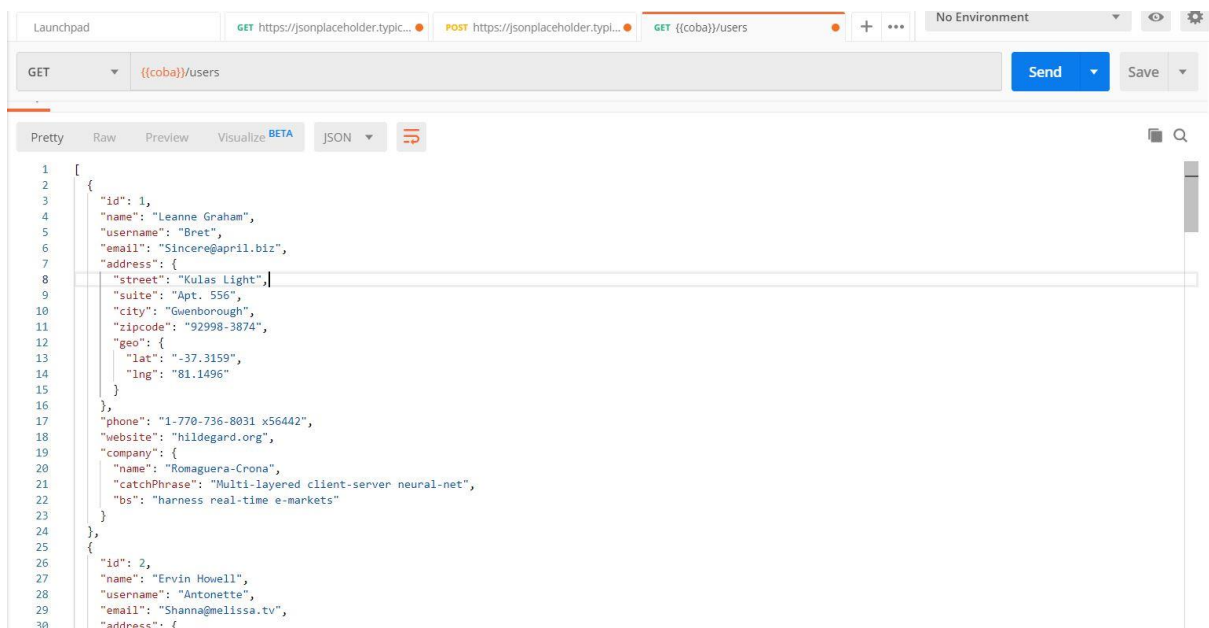
- Klik **ikon mata**
- Klik **edit** untuk mengatur variabel ke lingkungan global yang dapat digunakan di semua collection/collection.
- Isi sesuai dengan gambar pada Langkah 2 berikut
- Klik **Persist All**



Langkah 4) Klik tutup jika Anda melihat layar berikutnya.



Langkah 5) Kembali ke Request GET, lalu klik **Send/kirim**. Seharusnya sekarang ada hasil untuk permintaan Anda.



Catatan:

Selalu pastikan bahwa parameter Anda memiliki sumber seperti environment variable atau file data untuk menghindari kesalahan.

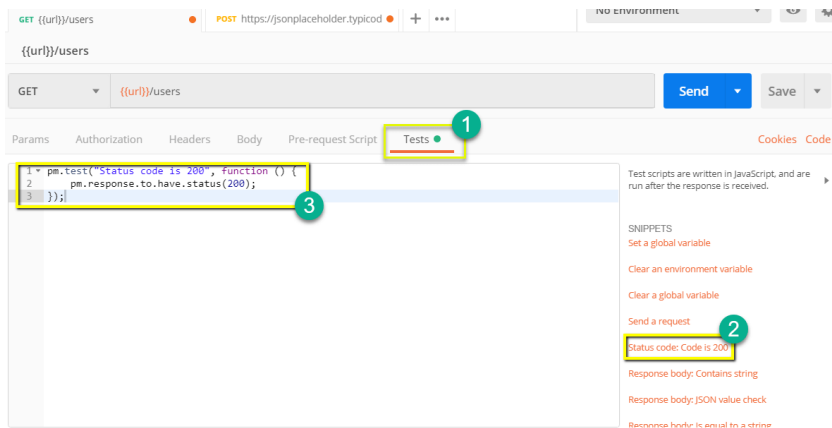
Cara Membuat Tes Postman

Postman Tests adalah kode JavaScript yang ditambahkan ke permintaan yang membantu Anda memverifikasi hasil seperti status berhasil atau gagal, perbandingan hasil yang diharapkan, dll. Biasanya dimulai dengan **pm.test**. Itu dapat dibandingkan dengan menegaskan, memverifikasi perintah yang tersedia di alat lain.

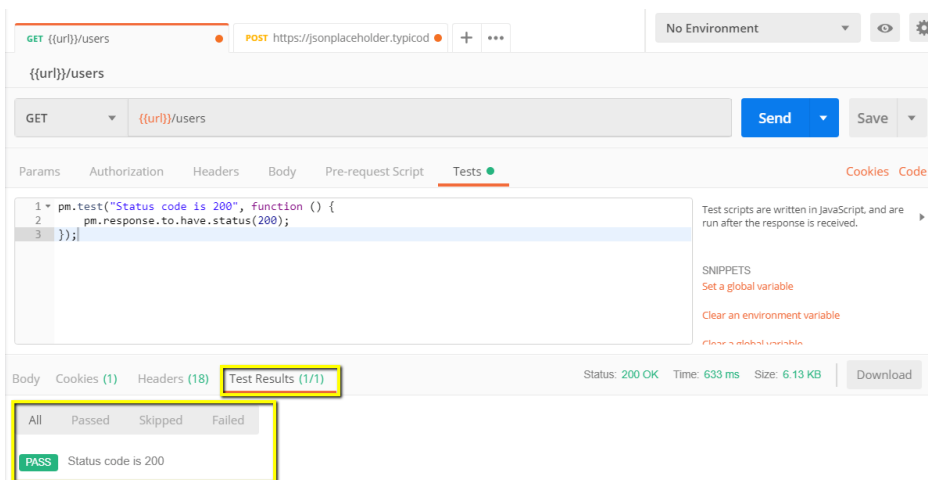
Mari kita buat beberapa tes dasar untuk permintaan parameterisasi dari pelajaran sebelumnya.

Langkah 1) Buka permintaan pengguna GET Anda dari tutorial sebelumnya.

- Beralih ke tab tes. Di sebelah kanan adalah kode snippet.
- Dari bagian cuplikan, klik "Kode status: Kode adalah 200".
- Panel diisi secara otomatis.

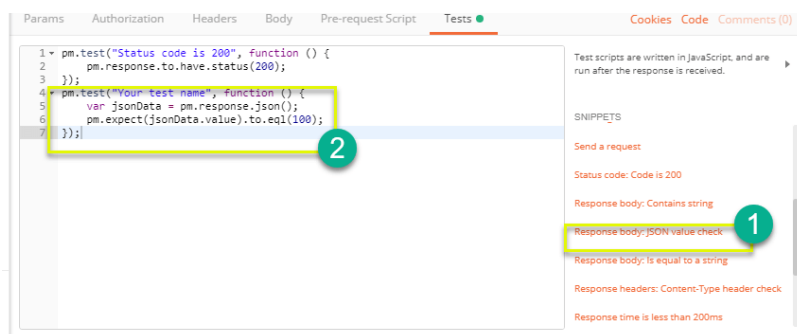


Langkah 2) Sekarang klik **Send/Kirim**. Hasil tes sekarang harus ditampilkan.



Langkah 3) Kembali ke tab tes dan mari kita tambahkan tes lain.

- Kali ini kita akan membandingkan hasil yang diharapkan dengan hasil yang sebenarnya.
- Dari bagian cuplikan, klik **"Response body:JSON value check"**. Kami akan memeriksa apakah **Leanne Graham** memiliki **userid 1**.



Langkah 4)

Ganti **"Your Test Name"** dari kode dengan **"Check if user with id1 is Leanne Graham"** sehingga nama tes menentukan dengan tepat apa yang ingin kami uji.

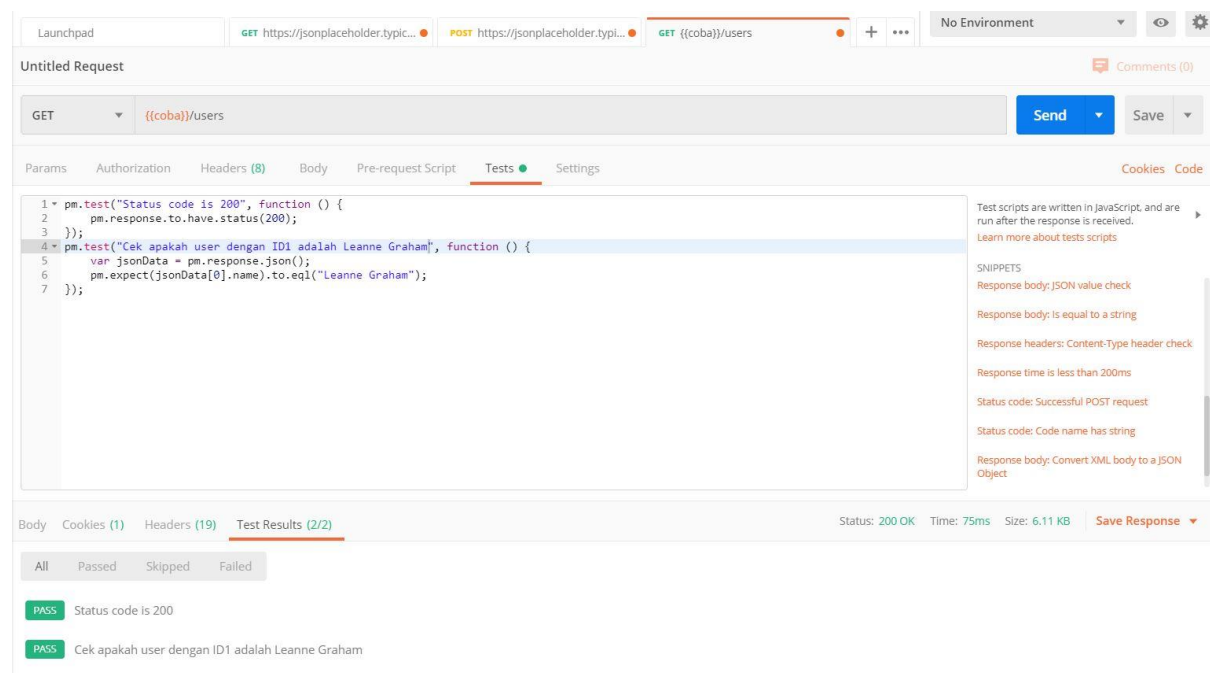
Ganti `jsonData.value` dengan `jsonData [0] .name`. Untuk mendapatkan jalur, periksa badan di Dapatkan hasil lebih awal. Karena Leanne Graham adalah userid 1, `jsonData` berada di hasil pertama

yang harus dimulai dengan 0. Jika Anda ingin mendapatkan hasil kedua, gunakan jsonData [1] dan seterusnya untuk hasil selanjutnya.

Dalam to eql, masukkan "Leanne Graham"

```
pm.test("Cek apakah user dengan ID1 adalah Leanne Graham", function () {  
    var jsonData = pm.response.json();  
    pm.expect(jsonData[0].name).to.eql("Leanne Graham");  
});
```

Langkah 5) Klik Send/kirim. Sekarang harus ada dua hasil tes yang lulus untuk permintaan Anda.



Catatan:

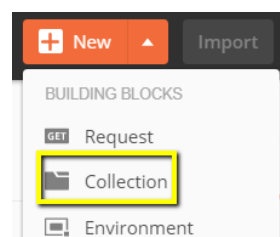
Ada beberapa jenis tes yang dapat dibuat di Postman . Coba jelajahi alat ini dan lihat tes apa yang sesuai dengan kebutuhan Anda.

Cara Membuat Collection

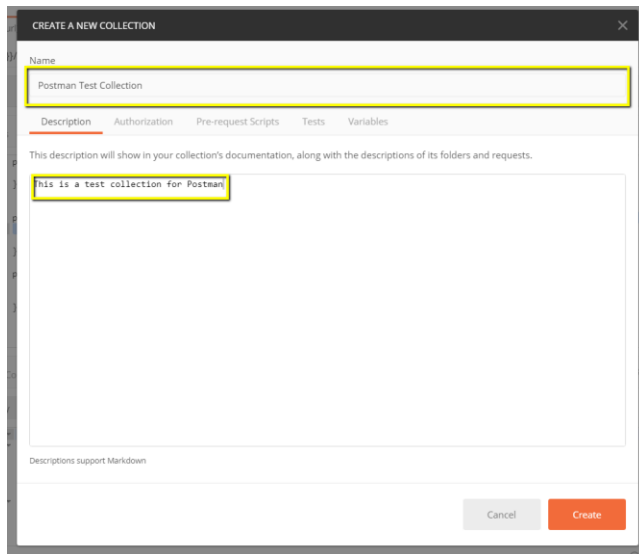
Collection memainkan peran penting dalam mengatur suite tes. Dapat diimpor dan diekspor sehingga mudah untuk berbagi collection di antara tim. Dalam praktikum ini, kita akan belajar cara membuat dan menjalankan collection.

Mari mulai membuat collection:

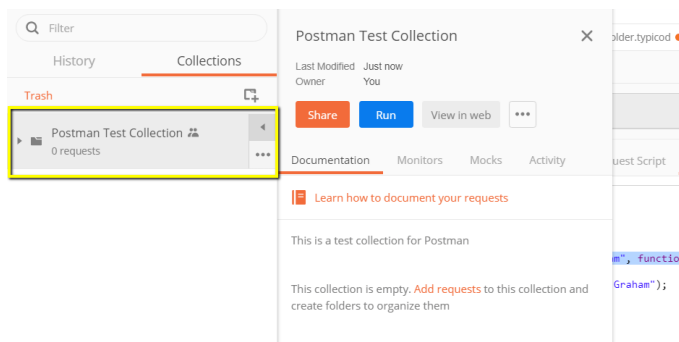
Langkah 1) Klik tombol Baru di sudut kiri atas halaman.



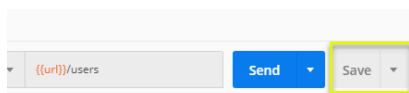
Langkah 2) Pilih Collection. Buat jendela collection harus muncul.



Langkah 3) Masukkan nama dan deskripsi collection yang diinginkan kemudian klik buat. Collection sekarang harus dibuat.

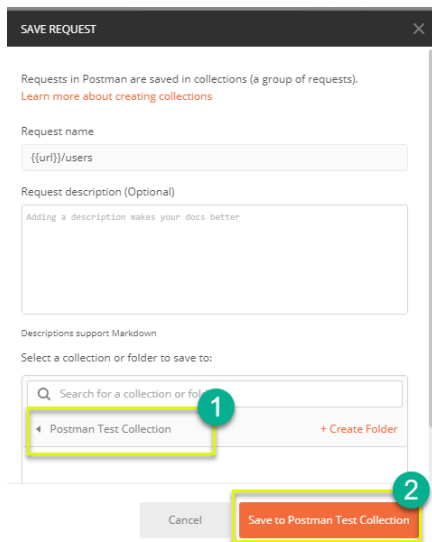


Langkah 4) Kembali ke Get request Sebelumnya. Klik **Save/Simpan**

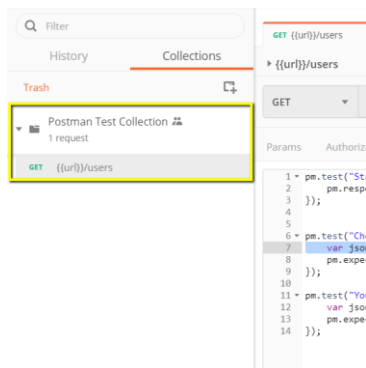


Langkah 5)

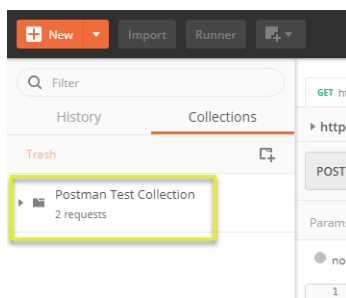
- Pilih Postman test Collection.
- Klik Save to Postman test Collection.



Langkah 6) Postman Test Collection sekarang pasti sudah berisi satu permintaan.



Langkah 7) Ulangi langkah 4-5 untuk Request POST sebelumnya sehingga collection sekarang akan memiliki dua permintaan.



Summary

- Postman saat ini adalah salah satu alat paling populer yang digunakan dalam pengujian API
- Aksesibilitas, Penggunaan Collection, Kolaborasi, Integrasi Berkelanjutan, adalah beberapa fitur utama untuk dipelajari di Postman
- Disarankan Anda membuat akun di Postman, sehingga collection Anda tersedia secara online
- Anda dapat membuat parameter permintaan di Postman
- Anda dapat membuat Tes untuk memverifikasi permintaan Postman
- Collection dapat dijalankan menggunakan Newman atau Collection Runner

Tugas:

1. Bagaimana cara menjalankan Collection menggunakan **Collection Runner di Postman?**
(Praktikkan dan dokumentasikan)
2. Bagaimana cara menjalankan Collection menggunakan **Newman di Postman?**
(Praktikkan dan dokumentasikan)
3. Akseslah website <https://regres.in/> untuk mengambil sample REST API. Dalam website ini ada sample-sample untuk melakukan API Test.
4. Lalu coba melakukan perintah-perintah seperti : request get, put, post dan delete dan terapkan pada Postman kesayangan anda dengan memanfaatkan <https://regres.in/> sebagai Rest API endpoint. (manfaatkanlah internet anda untuk eksplorasi)
5. Dokumentasikanlah Langkah demi Langkah praktikum dan tugas anda dalam laporan praktikum dengan nama file:
PrakW10S02_KodeKelas_NIM Lengkap.pdf

EOF