

LAPORAN SIMULASI API LIFECYCLE

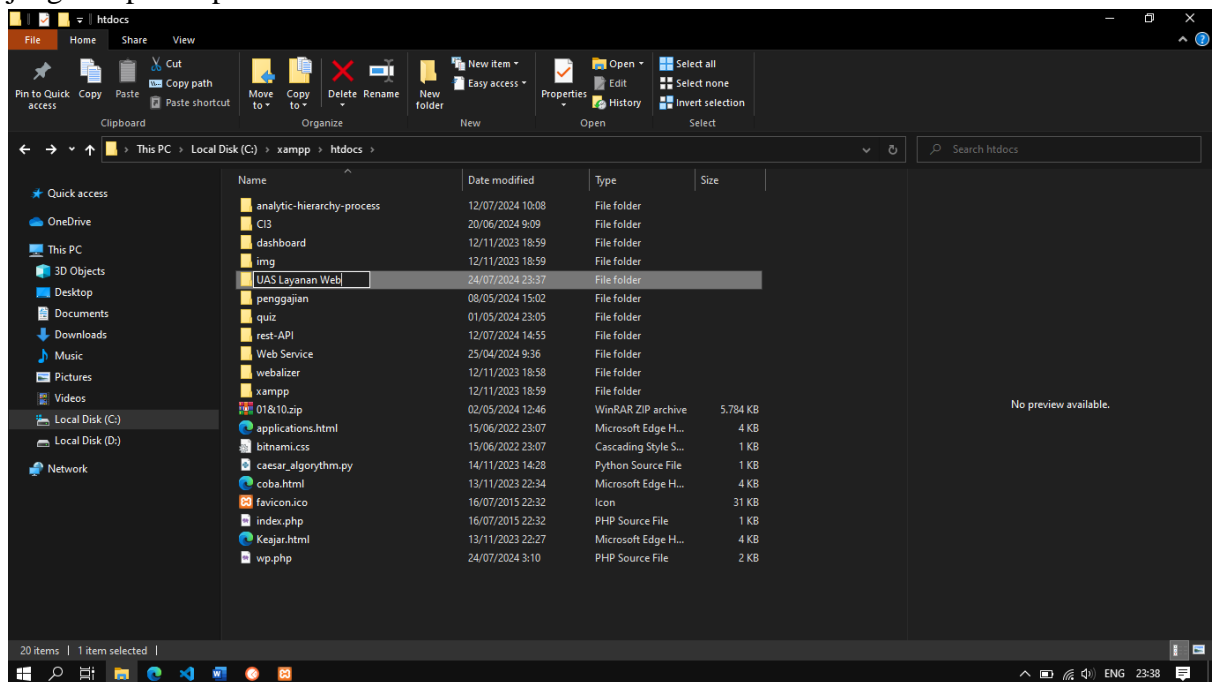
UAS LAYANAN WEB

Nama Kelompok :

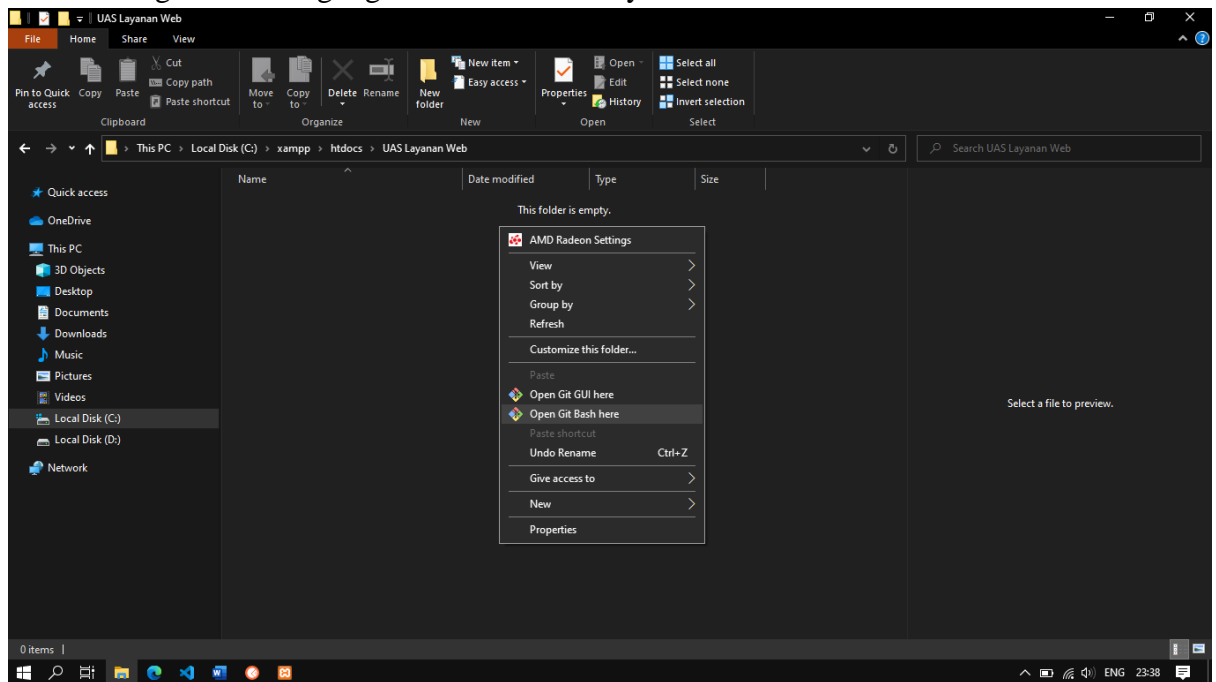
1. Amanah Wismarta Pangesti / 211111101
2. M. Kausari Kaidani / 211111110
3. Muhammad Ramdeni / 211111126

Berikut langkah-langkah simulasi API :

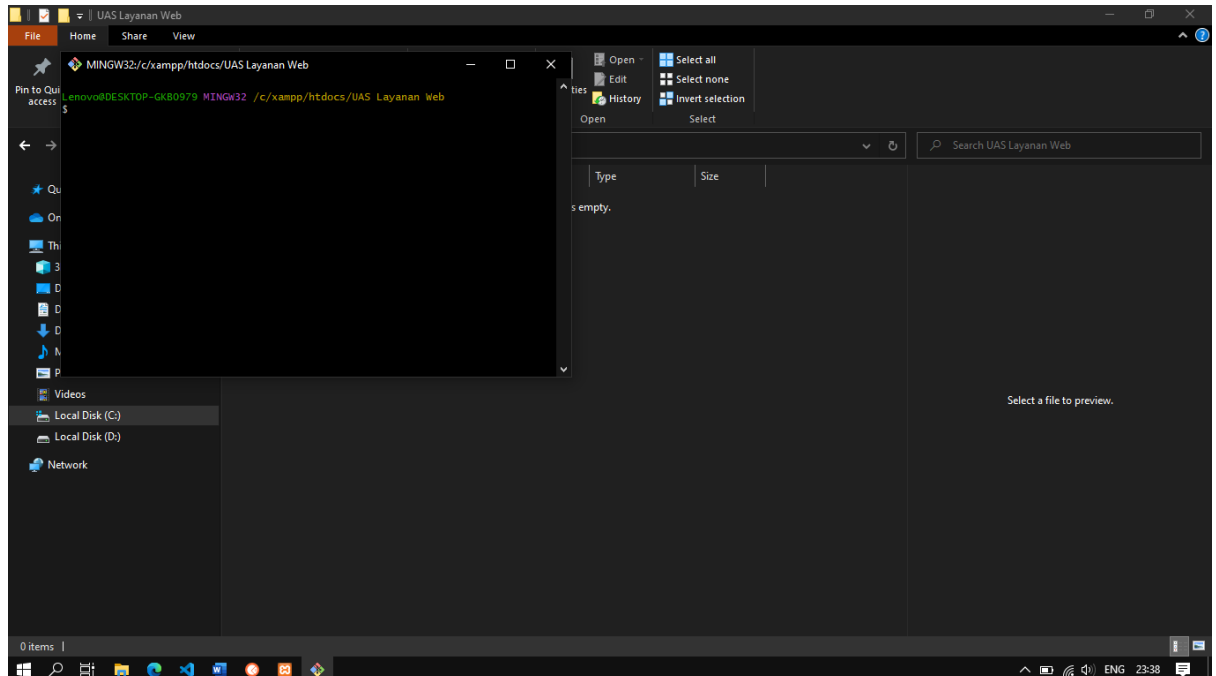
1. Buat folder baru sebagai wadah untuk melakukan beberapa persiapan simulasinya dan jangan lupa tempatkan di “**htdocs**”.



2. Jika sudah, karena kami menggunakan bantuan Node.js dengan express untuk membuat API sederhana, langkah selanjutnya adalah dengan menyiapkan terminal untuk menginstal *dependencies*-nya. Lebih mudahnya kami menggunakan terminal GIT Bash agar tidak bingung mencari direktorinya.



Seperti ini, kalian akan langsung di arahkan ke direktori yang diinginkan.



3. Untuk menginstal *dependencies*-nya, berikut beberapa command yang dijalankan.

```
MINGW32:/c/xampp/htdocs/UAS Layanan Web

Lenovo@DESKTOP-GKB0979 MINGW32 /c/xampp/htdocs/UAS Layanan Web
$ npm init -y
Wrote to C:\xampp\htdocs\UAS Layanan Web\package.json:

{
  "name": "uas-layanan-web",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Kemudian *command* selanjutnya:

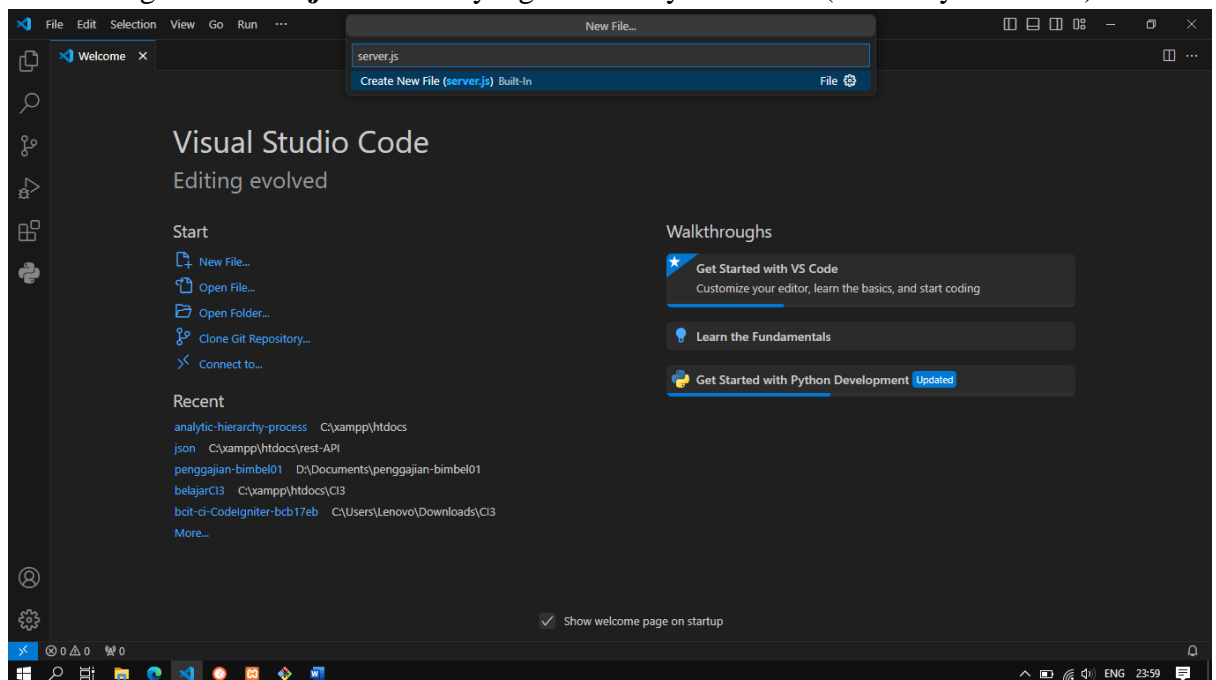
```
Lenovo@DESKTOP-GKB0979 MINGW32 /c/xampp/htdocs/UAS Layanan Web
$ npm install express body-parser

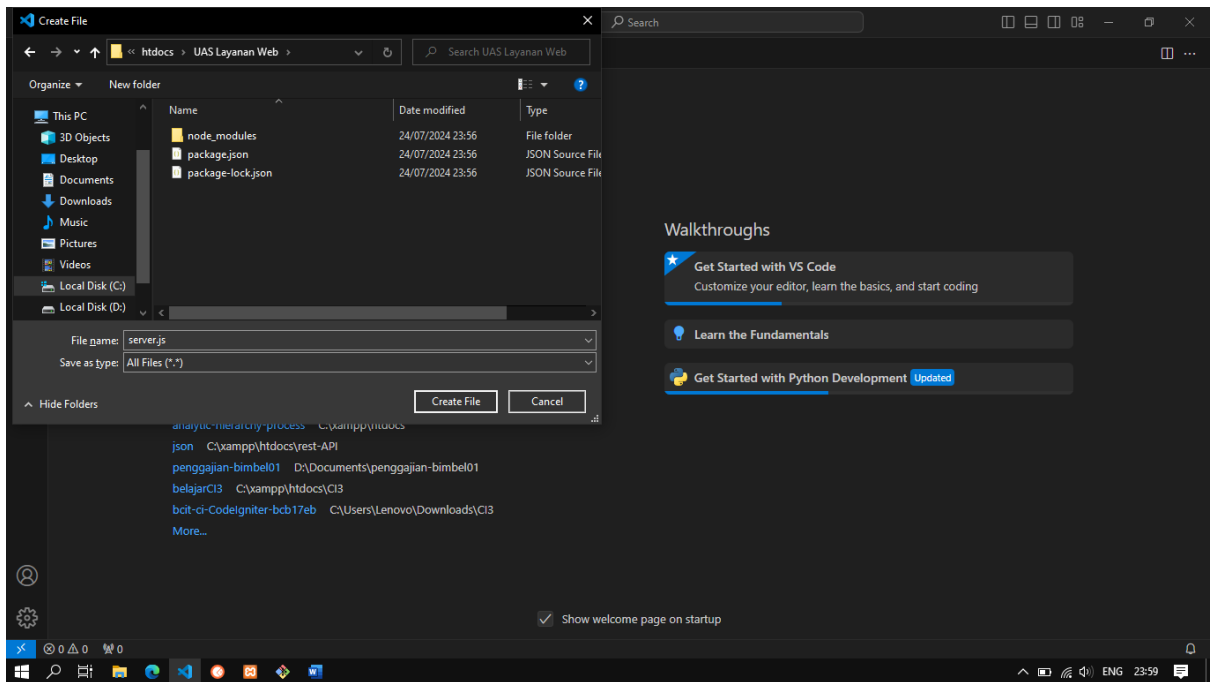
added 64 packages, and audited 65 packages in 8s

12 packages are looking for funding
  run `npm fund` for details

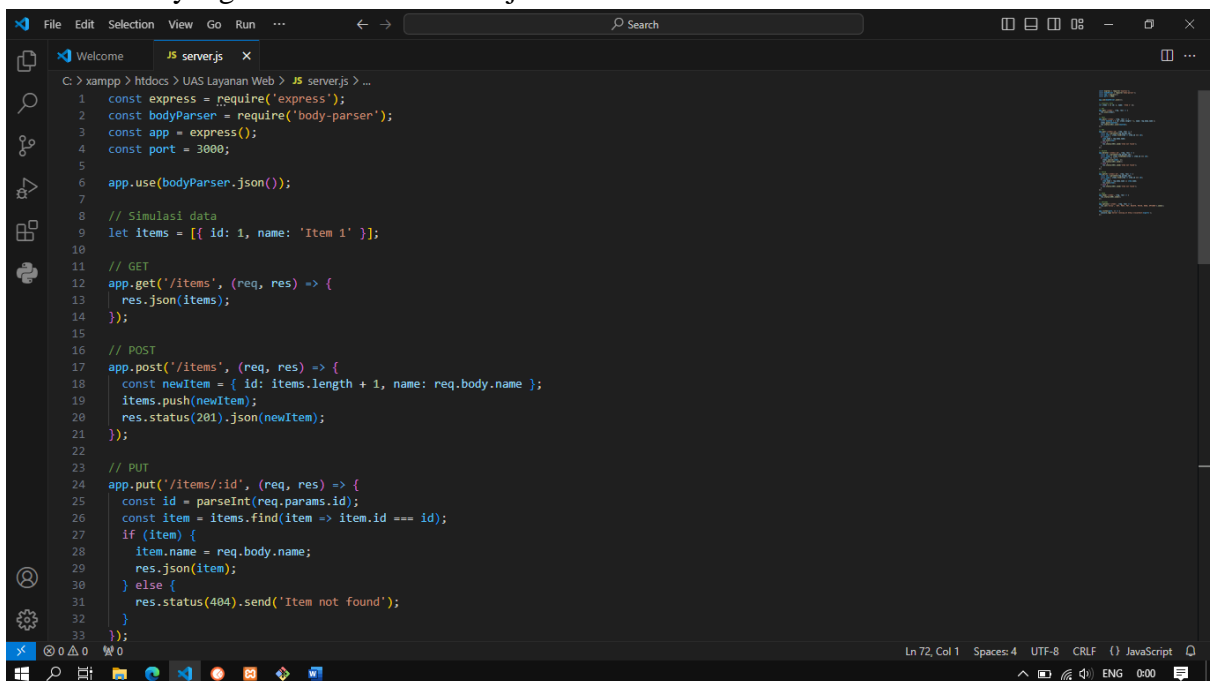
found 0 vulnerabilities
```

4. Setelah berhasil melakukan langkah nomor tiga, selanjutnya adalah membuat file server dengan format “**.js**” di folder yang sebelumnya kita buat (UAS Layanan Web).





Jika file sudah dibuat, buat *code* yang akan digunakan untuk simulai API di Postman, berikut *code* yang kami buat untuk di ujikan:



Sebelum simulai di Postman jangan lupa untuk menjalankan *server* dengan *command* sebagai berikut :

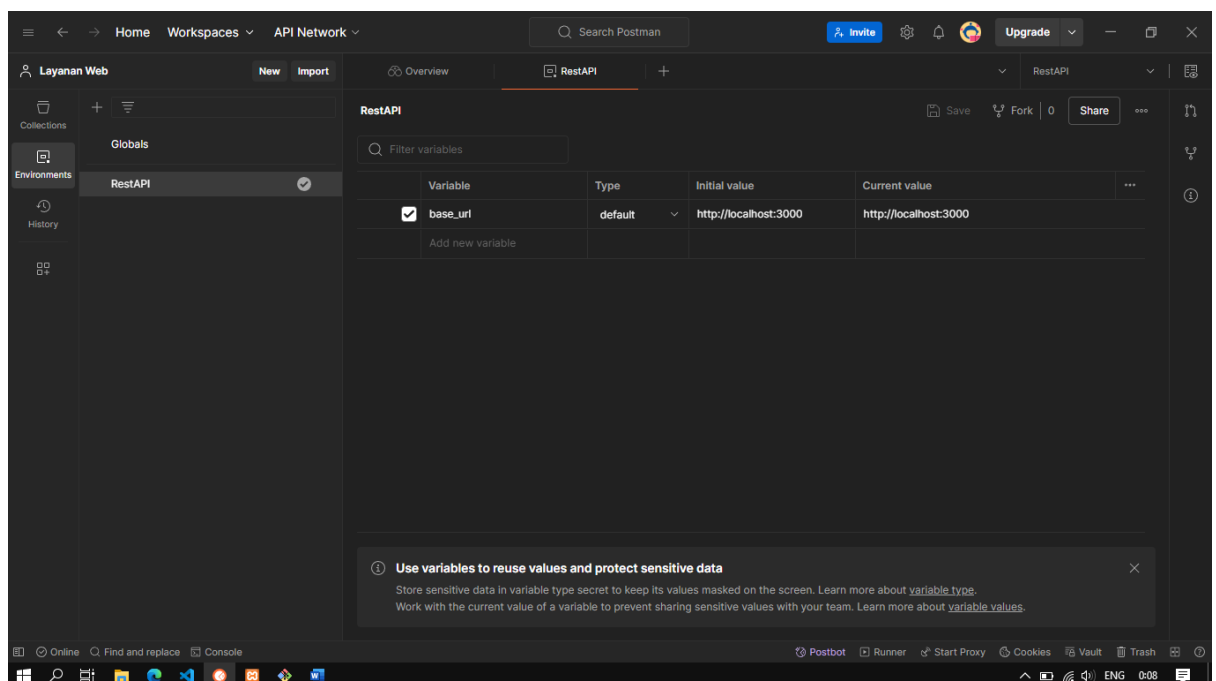
```

Lenovo@DESKTOP-GKB0979 MINGW32 /c/xampp/htdocs/UAS Layanan Web
$ node server.js
Server running at http://localhost:3000

```

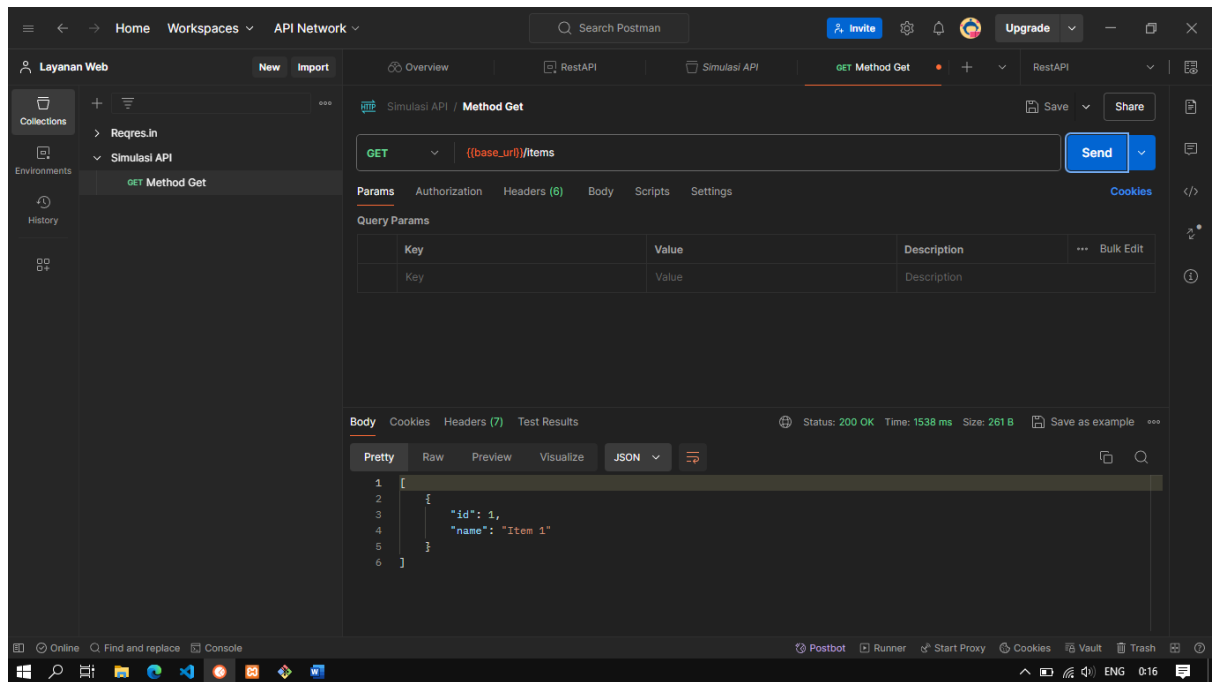
5. Baik, setelah file-file yang dibutuhkan untuk mendukung simulasi API sudah selesai langkah selanjutnya adalah dengan menyiapkan *environment*-nya di Postman. Jika kamu belum pernah membuat environment, berikut langkah yang harus dilakukan :
- Klik menu “Environments”
 - Kemudian klik tulisan “*create new environments*”
 - Beri nama sesuai dengan keinginan, kemudian isikan beberapa hal seperti :
 - Variabel-nya
 - Value-nya
 - Jangan lupa di sebelah pojok kanan ubah ke environment yang sudah kamu setting dan jangan lupa klik simpan.

Seperti ini misalnya :



6. Langkah selanjutnya adalah menyiapkan beberapa *request* di Postman dengan buat “Collections” baru dan tambahkan beberapa request yang ingin kamu ujikan. Ada 7 *method* yang di sediakan di Postman, berikut cara pengujiannya :
- Get Request**

Yang perlu di *setting* :
Methodnya : GET
URLnya : {{base_url}}/items (sesuai yang kamu isikan di code server)



Setelah itu klik “Send” dan akan muncul seperti pada gambar di atas atau kamu berhasil mendapatkan item yang ada di code server kamu, dilihat dari statusnya 200 OK yang mana menandakan pengujian berhasil.

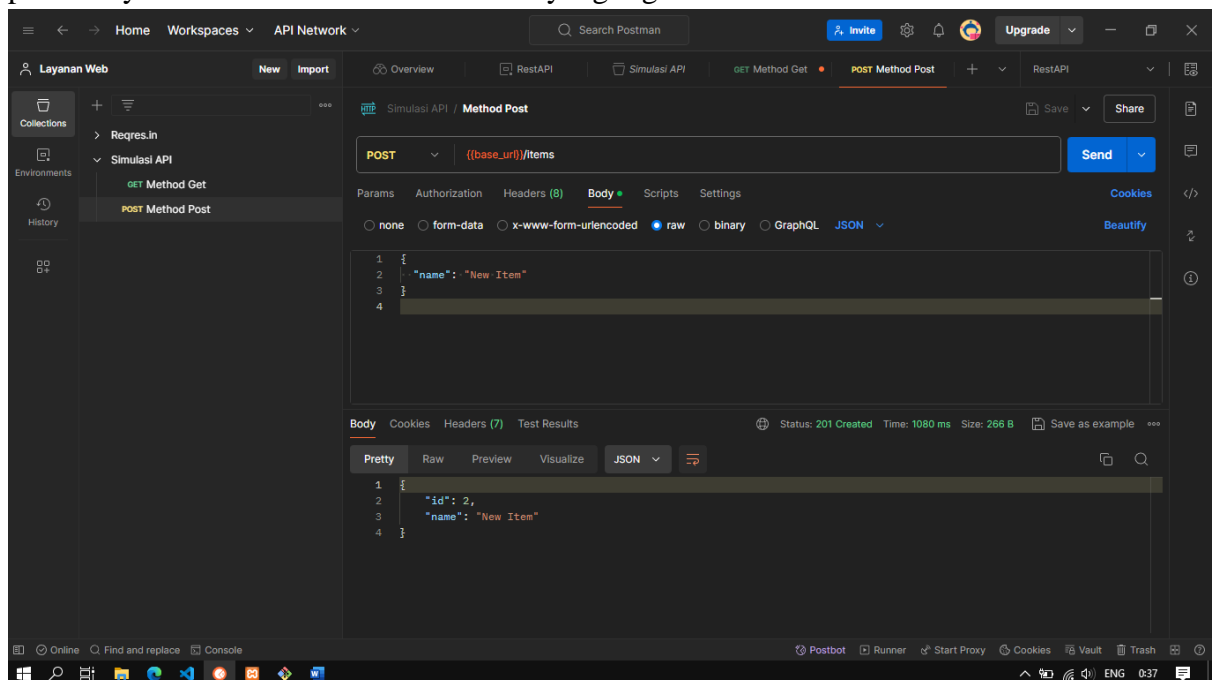
2. POST Request

Yang perlu di *setting* :

Methodnya : POST

URLnya : `{{base_url}}/items` (sesuai yang kamu isikan di code server)

Karena di *method* ini kita akan menambahkan item baru, maka di bagian Postman pilih body -> raw -> JSON -> isikan item yang ingin kamu tambahkan.



Setelah itu klik “Send” dan akan muncul seperti pada gambar di atas, dilihat dari

statusnya 200 OK yang mana menandakan pengujian berhasil dan item baru berhasil di tambahkan sesuai dengan yang kamu *request*-kan.

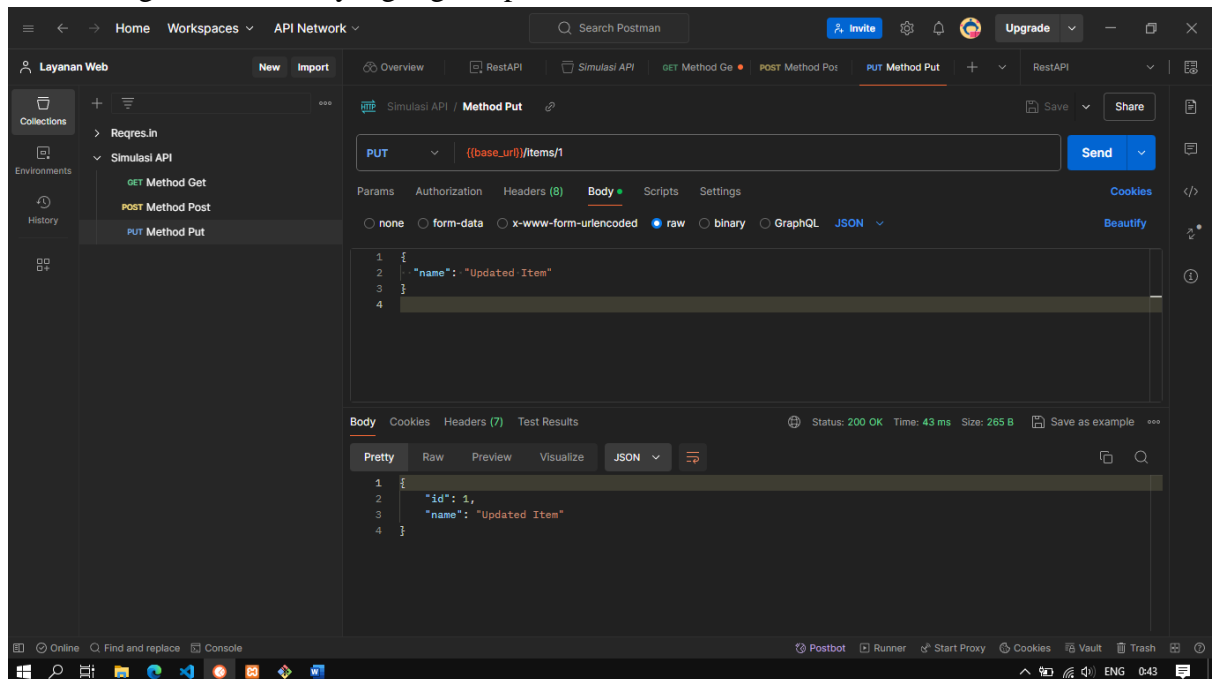
3. PUT Request

Yang perlu di *setting*:

Methodnya : PUT

URLnya : `{{base_url}}/items/1`

Karena di *method* ini kita akan memperbarui item pada ID 1 maka jangan lupa tambahkan /1 di URLnya, lalu di bagian Postman pilih body -> raw -> JSON -> isikan dengan nama item yang ingin diperbarui.



Setelah itu klik “Send” dan akan muncul seperti pada gambar di atas, dilihat dari statusnya 200 OK yang mana menandakan pengujian berhasil dan perubahan item pada ID 1 berhasil di lakukan yang sebelumnya nama ID 1 adalah item-1 berubah menjadi Update Item.

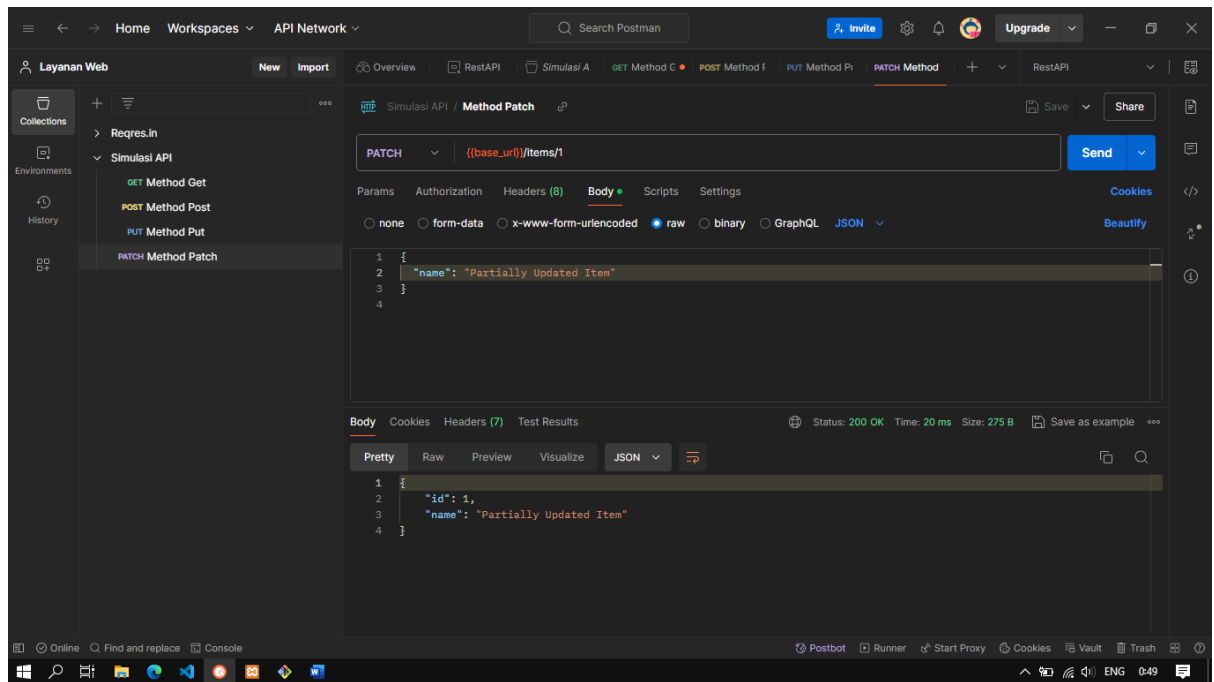
4. PATCH Request

Yang perlu di *setting* :

Methodnya : PATCH

URLnya : `{{base_url}}/items/1`

Karena di *method* ini kita akan memperbarui sebagian item pada ID 1 maka jangan lupa tambahkan /1 di URLnya, lalu di bagian Postman pilih body -> raw -> JSON -> isikan dengan nama item yang ingin diperbarui.



Setelah itu klik “Send” akan muncul seperti pada gambar di atas, dilihat dari statusnya 200 OK yang mana menandakan pengujian berhasil dan perubahan item pada ID 1 berhasil di lakukan sesuai dengan apa yang kita isikan.

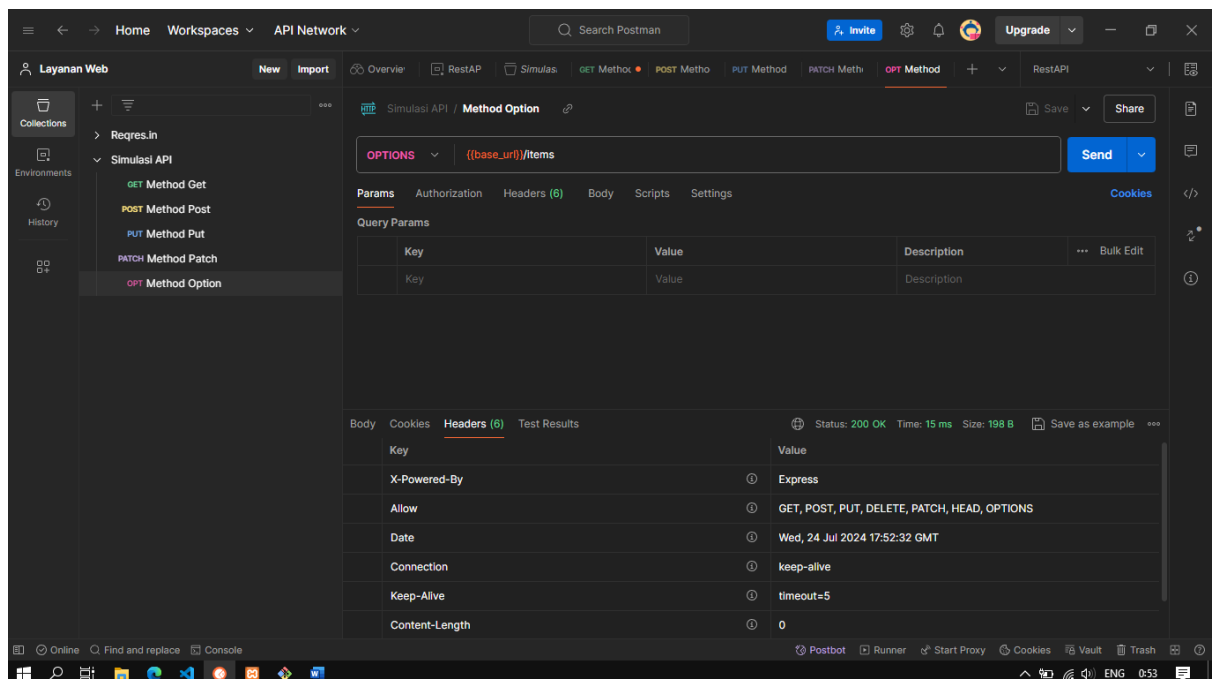
5. OPTION Request

Yang perlu di *setting* :

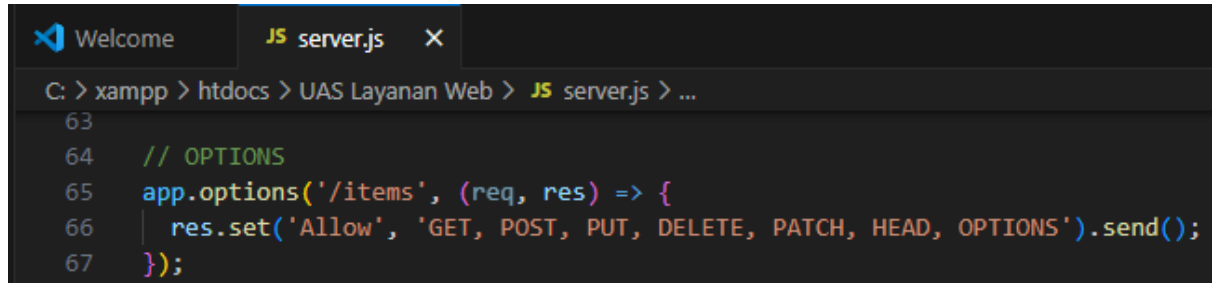
Methodnya : OPTION

URLnya : `{{base_url}}/items`

Pada method ini kita ingin mengetahui method apa saja yang diizinkan di ujikan, tentunya sesuai dengan apa yang kita masukkan pada code server yang sudah kita buat tadi.



Setelah itu klik “Send” akan muncul seperti pada gambar di atas, dilihat dari statusnya 200 OK yang mana menandakan pengujian berhasil dan terlihat method apa saja yang diizinkan untuk diujikan sesuai dengan yang kita isikan di *code server*, seperti berikut :



```
63
64 // OPTIONS
65 app.options('/items', (req, res) => {
66   res.set('Allow', 'GET, POST, PUT, DELETE, PATCH, HEAD, OPTIONS').send();
67 });
```

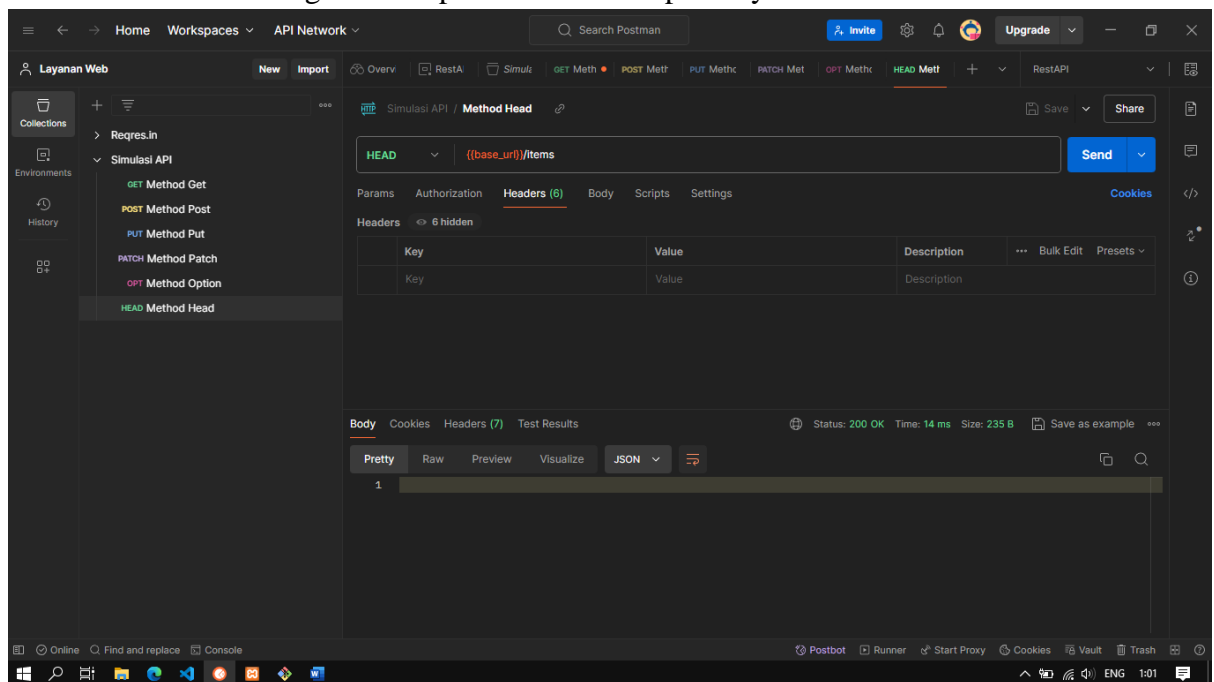
6. HEAD Request

Yang perlu di *setting* :

Methodnya : HEAD

URLnya : `{{base_url}}/items`

Pada method ini kita ingin mendapatkan header tanpa body



Setelah itu klik “Send” akan muncul seperti pada gambar di atas, dilihat dari statusnya 200 OK yang mana menandakan pengujian berhasil.

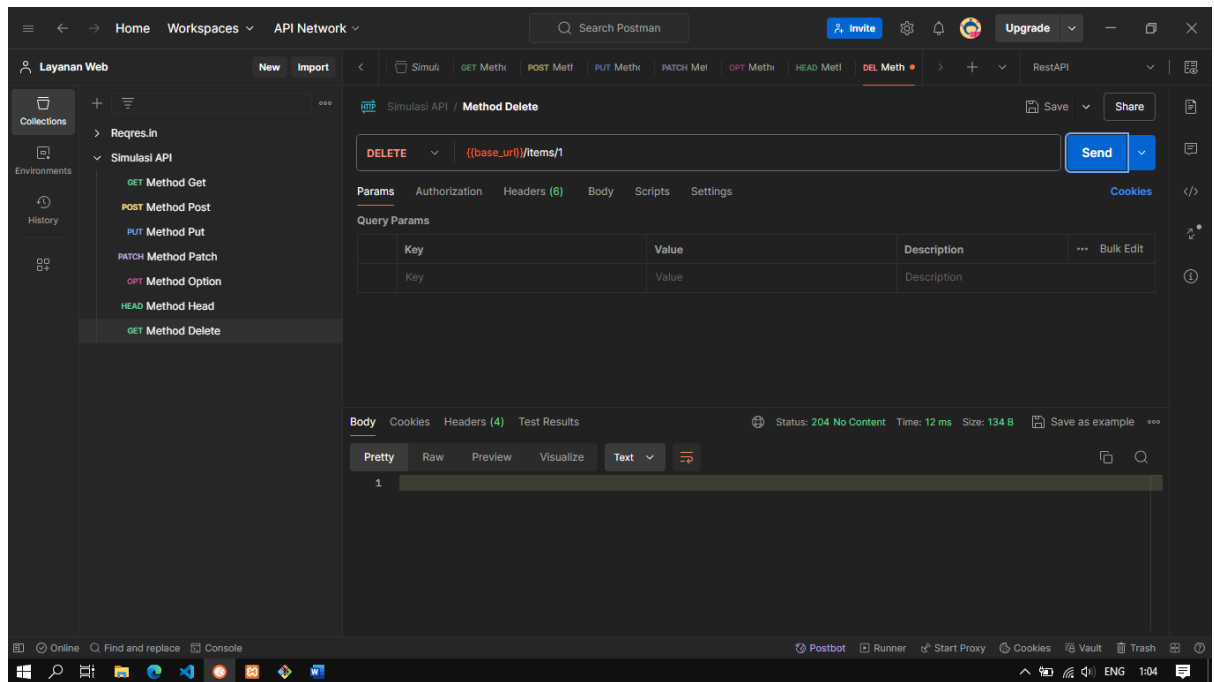
7. DELETE Request

Yang perlu di *setting* :

Methodnya : DELETE

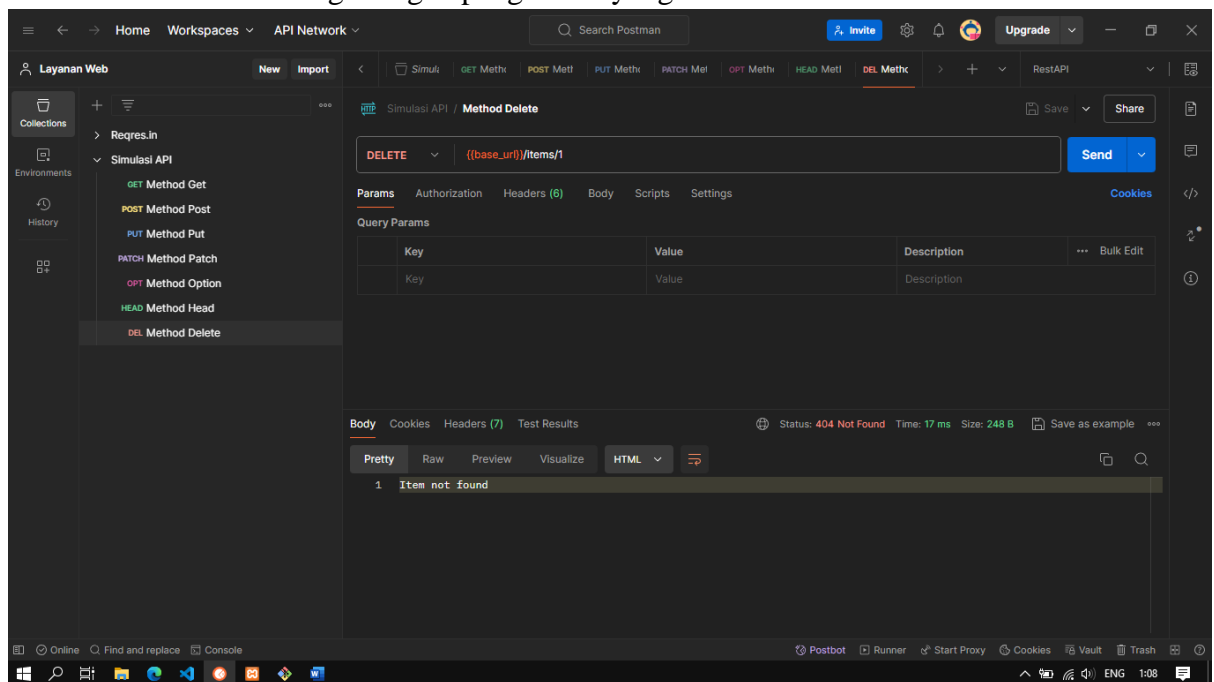
URLnya : `{{base_url}}/items/1`

Karena di method ini kita akan menghapus item pada ID 1 maka jangan lupa tambahkan /1 di URLnya



Setelah itu klik “Send” akan muncul seperti pada gambar di atas, dilihat dari statusnya 204 No Content yang mana menandakan pengujian berhasil.

Jika kamu klik “Send” lagi dengan pengaturan yang sama maka akan muncul



Seperti pada gambar di atas, dengan status 404 Not Found yang artinya ID 1 tidak ada, karena sebelumnya sudah kita hapus dan pengujian berhasil.