**LAPORAN PRAKTIKUM MAHASISWA**

**PEMROGRAMAN WEB FRAMEWORK**

**Membuat RESTful API (NODEJS dan MySQL)**

****

**Oleh:**

**NAMA : Haldian**

**NPM :** **20753050**

**KELAS** **: Manajemen Informatika 4B**

**Dosen:**

**Tri Sandhika Jaya, S.Kom., M.Kom.**

**D3 – MANAJEMEN INFORMATIKA**

**JURUSAN EKONOMI DAN BISNIS**

**POLITEKNIK NEGERI LAMPUNG**

**April 2022**

A. Tujuan (Capaian Pembelajaran)

Setelah praktikum ini, praktikan mahasiswa diharapkan dapat:

Membuat Restful Application Programming Interface(API)

B. Peralatan yang digunakan

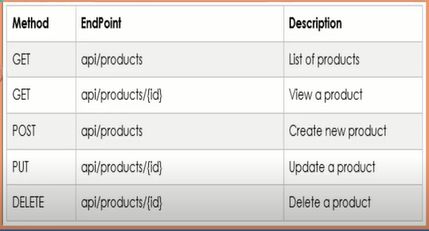
1. Web Editor : Visual Studio Code
2. Web Browser : Google Chrome
3. OS PC : Windows 10
4. XAMPP : versi 7.2
5. NodeJs : 16.14.0 LTS
6. GitBash

C. Hasil Praktikum

**Step #1 Design RESTful API**

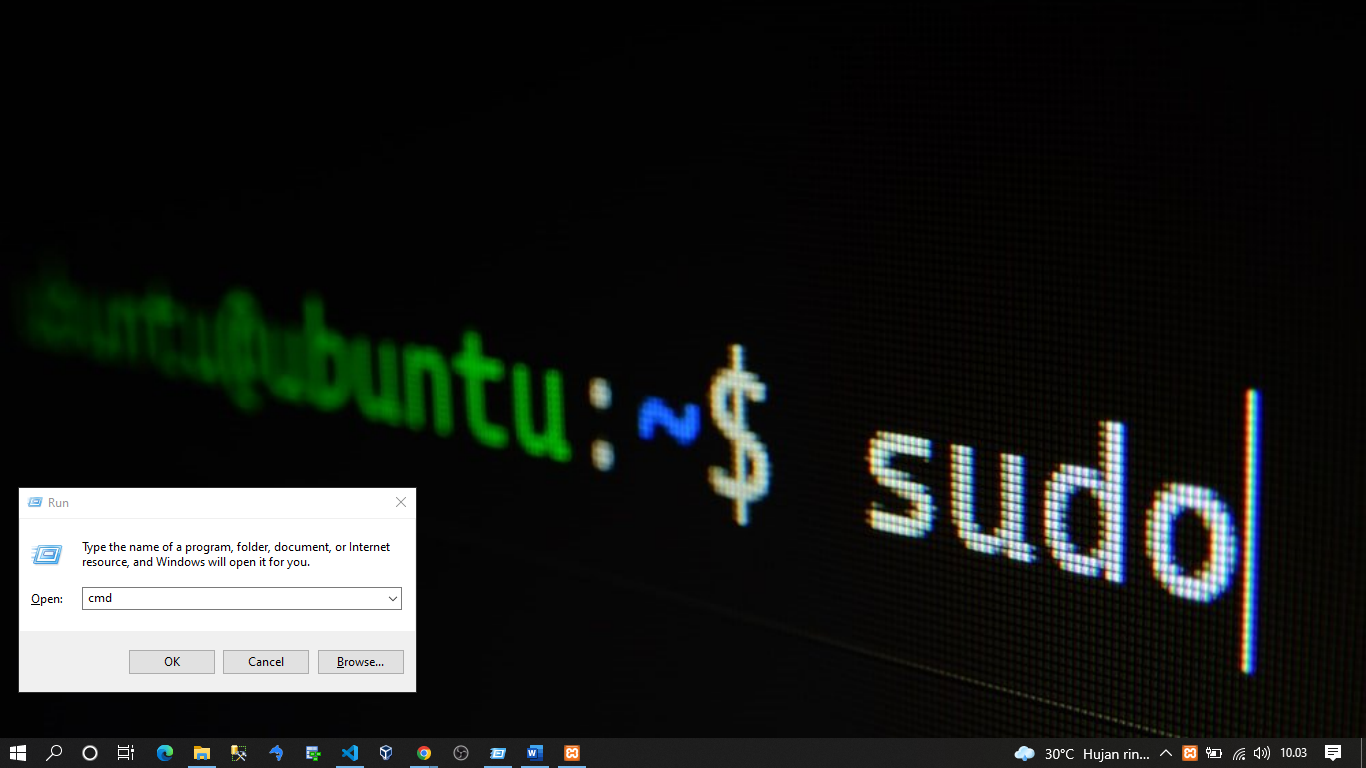
Mendefinisikan dulu EndPoint dari RESTful API yang akan dibuat. EndPoint merupakan routes dari API yang akan kita buat. RESTful API menggunakan HTTP verbs. HTTP verbs yang umum digunakan adalah GET, POST, PUT, dan DELETE. GET untuk mendapatkan data dari server atau lebih dikenal dengan istilah READ, POST untuk meng-CREATE new data, PUT untuk UPDATE data, dan DELETE untuk menghapus data. Atau lebih dikenal dengan istilah CRUD (Create Read Update Deletedari suatu table di database yaitu table

products.

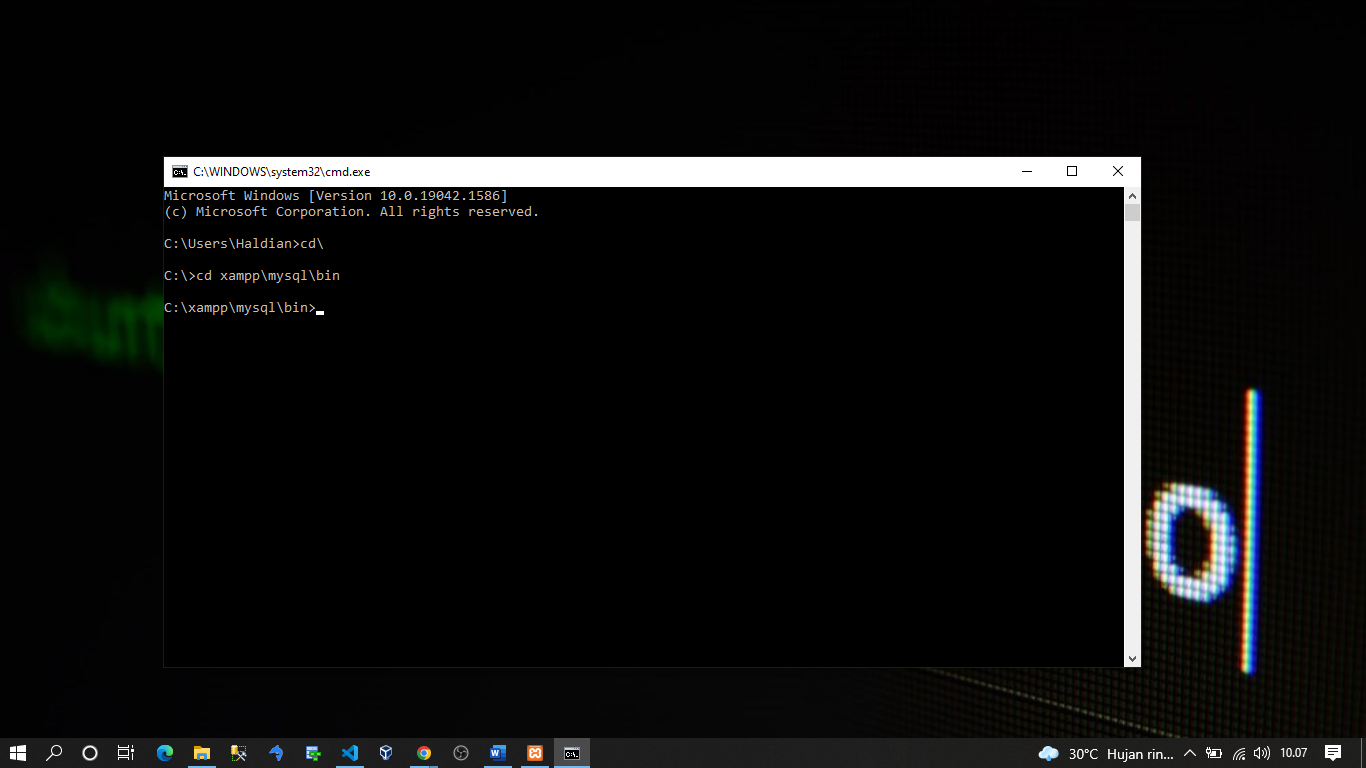


**Step #2. Buat Database dan Table**

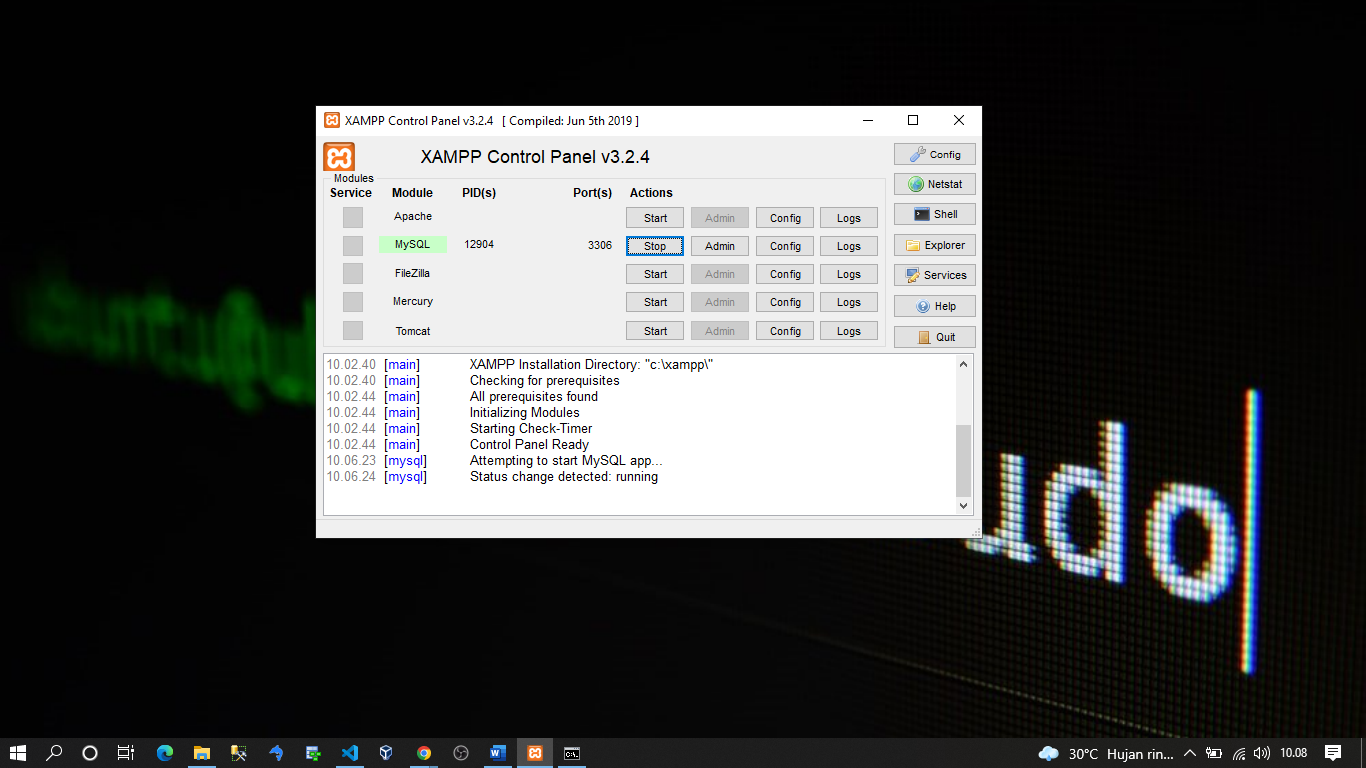
1. Kita buka cmd kita dengan win + r maka akan seperti gambar di bawah ini



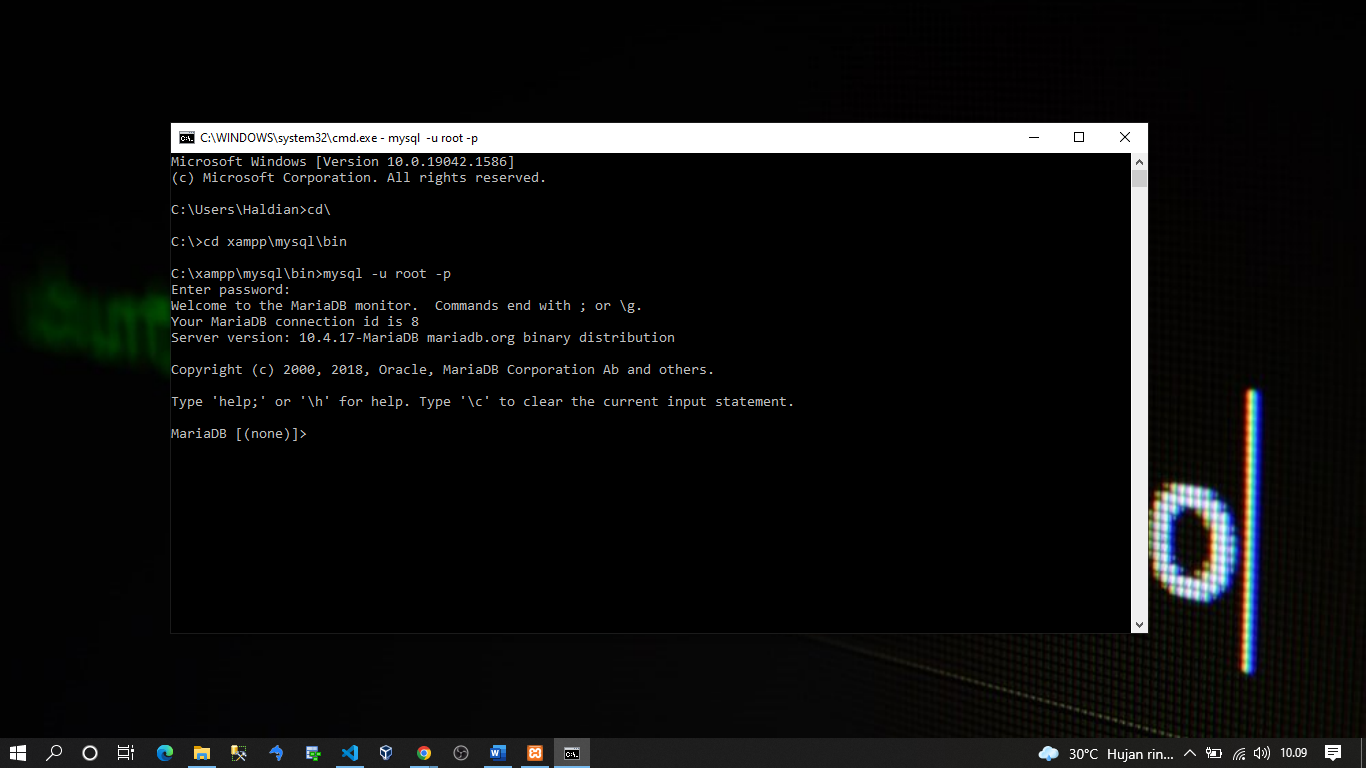
1. Kemudian kita masuk ke cd xampp\mysql\bin seperti gambar di bawah ini



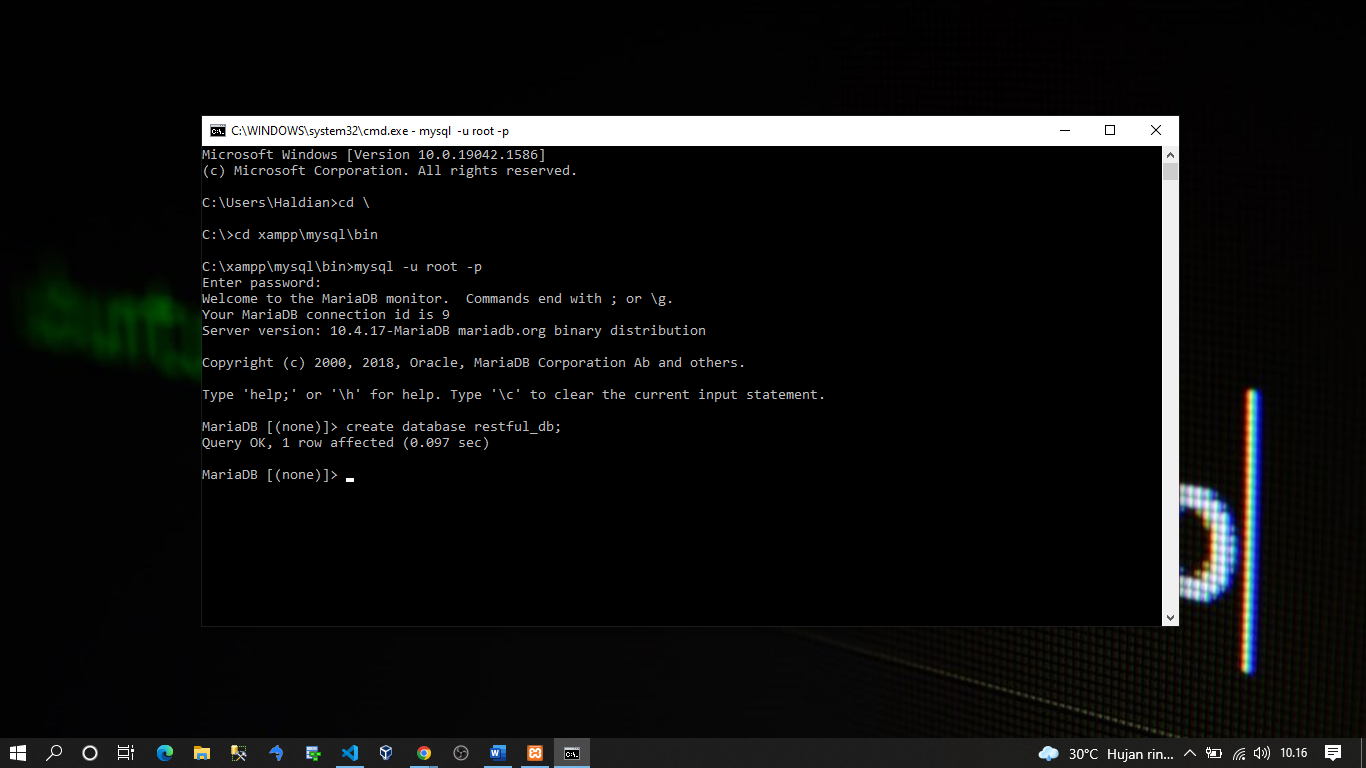
1. Selanjutnya kita buka xampp kita dan kita hidupkan mysql servernya seperti gambar di bawah ini



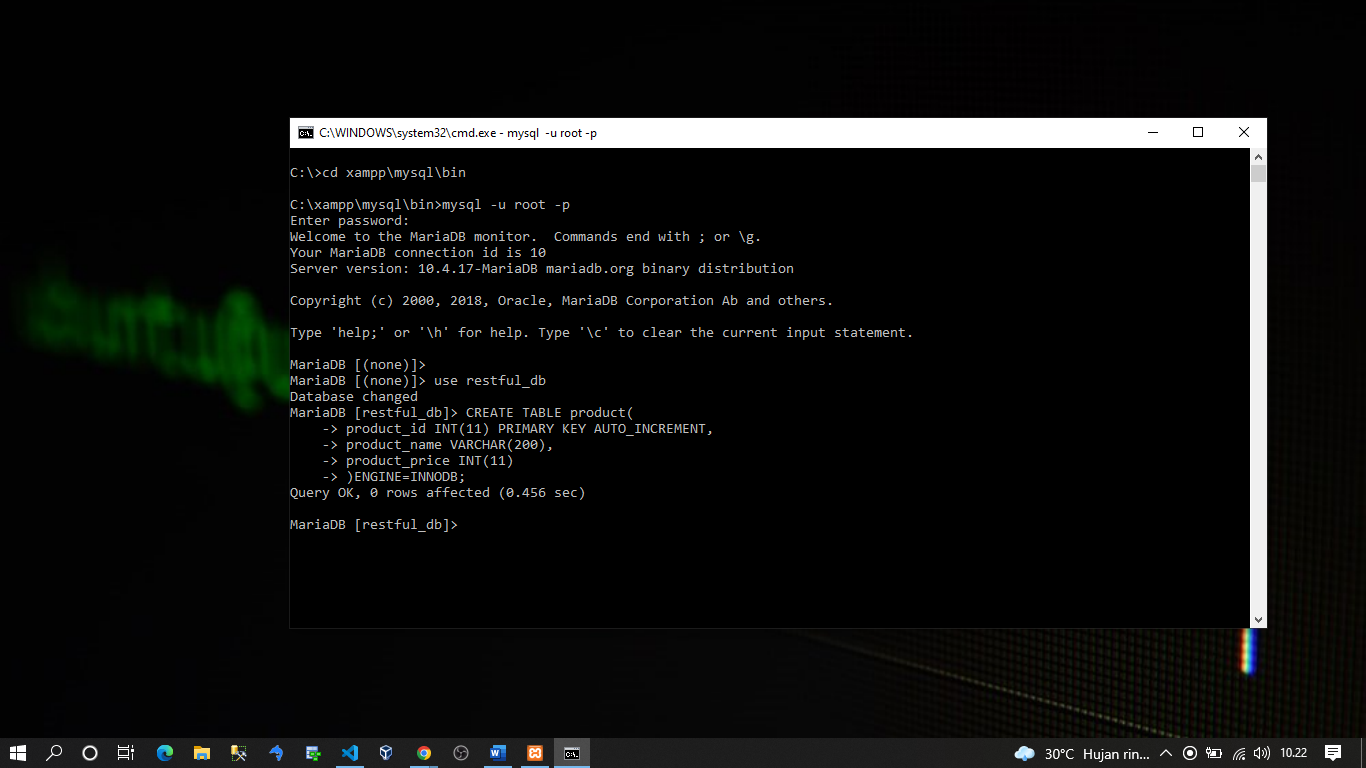
1. Untuk mengakses mysql server kita kita ketikan mysql -u root -p kemudian enter pada console kita seperti gambar di bawah ini



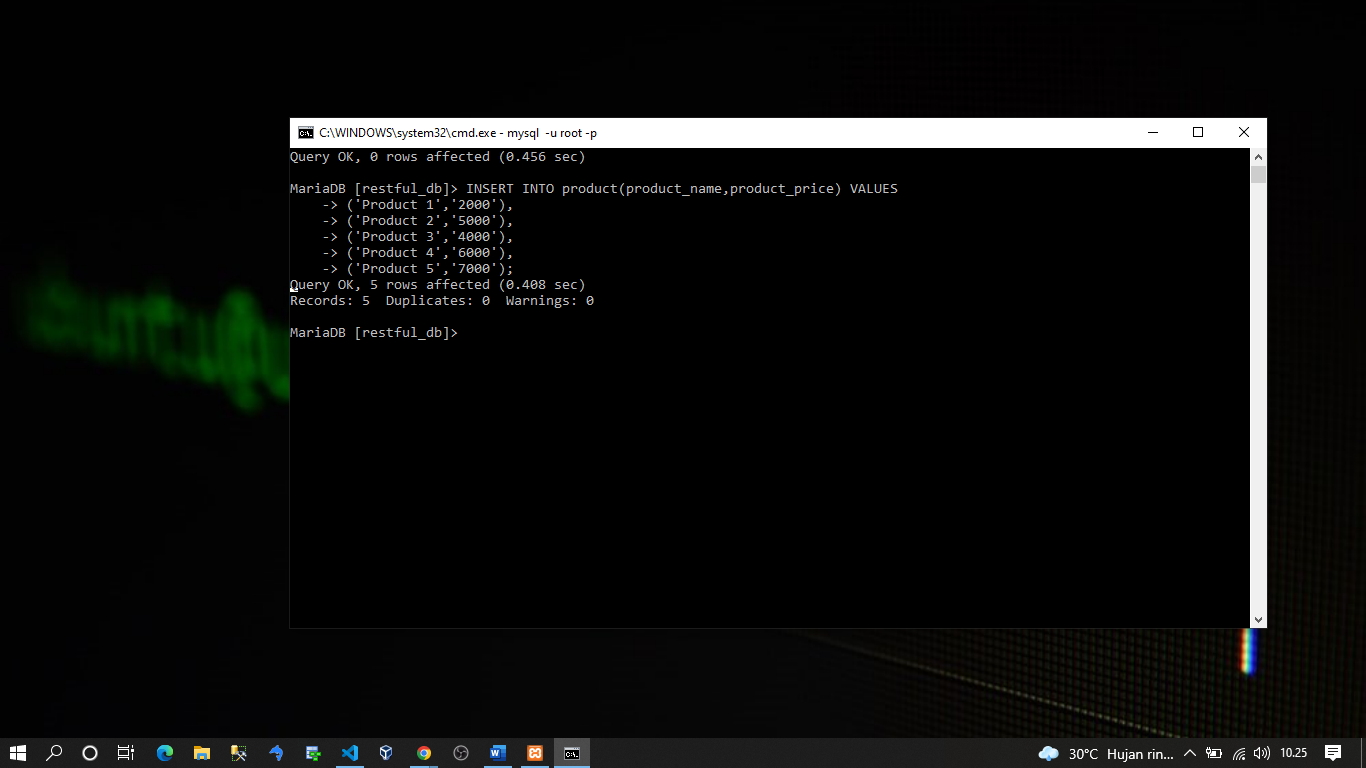
1. kemudian membuat database dengan cara create database restful\_db seperti gambar di bawah ini



1. Selanjutnya kita create table product seperti gambar di bawah ini

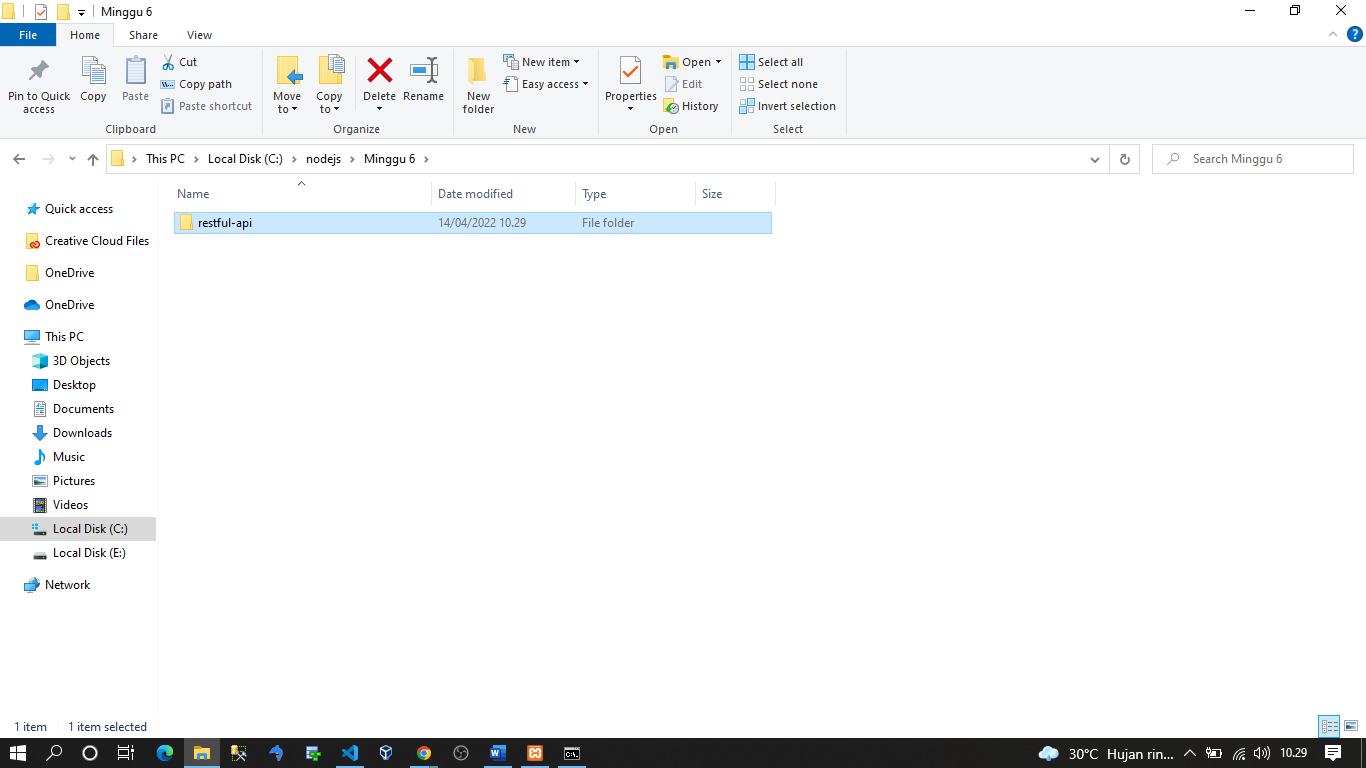


1. Selanjutnya kita insert data kedalam table product dengan mengeksekusi query dimana akan menginsert 5 data pada table product seperti gambar di bawah ini



**Step #3. Install Dependencies**

1. Sebelum menginstall dependencies kita perlu membuat sebuah folder dengan nama restful-api seperti gambar di bawah ini.



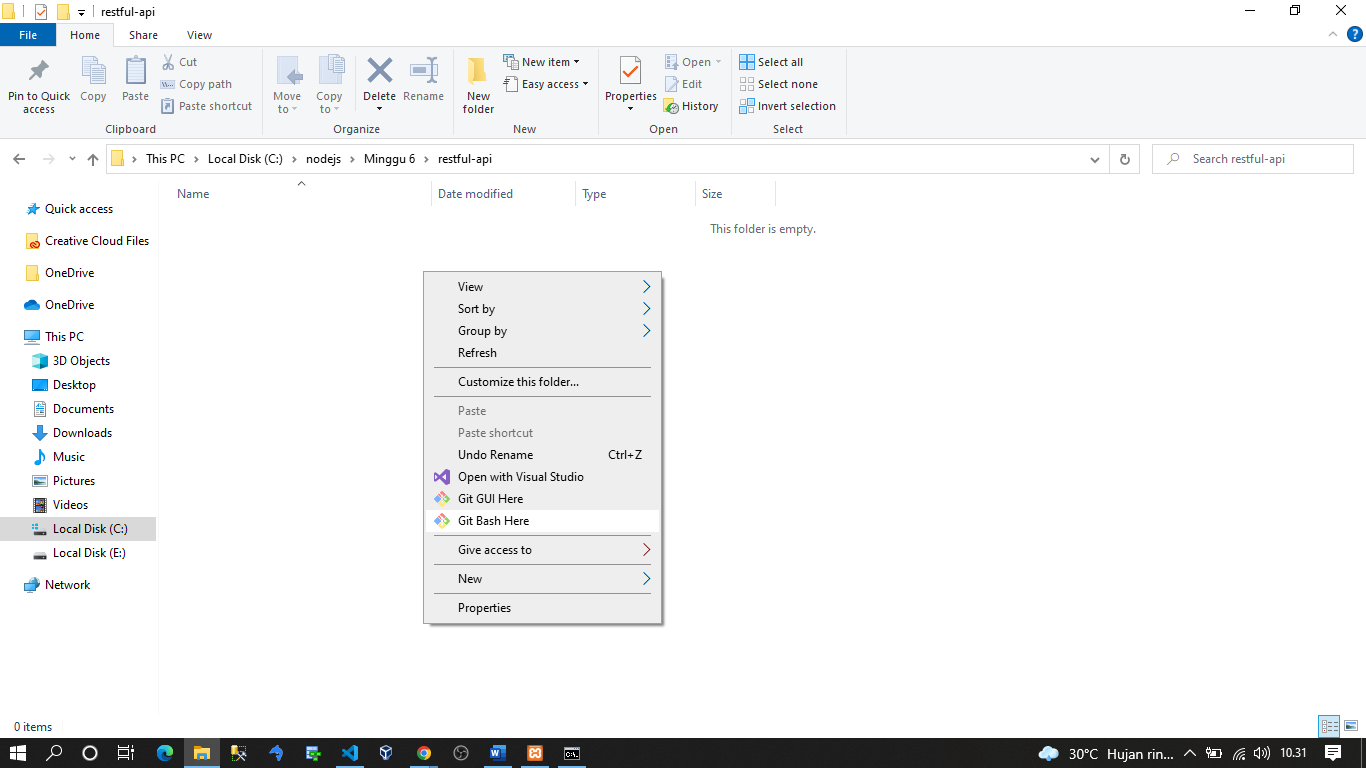
kita membutuhkan 3 dependencies yaitu:

1. Express (node.js framework)

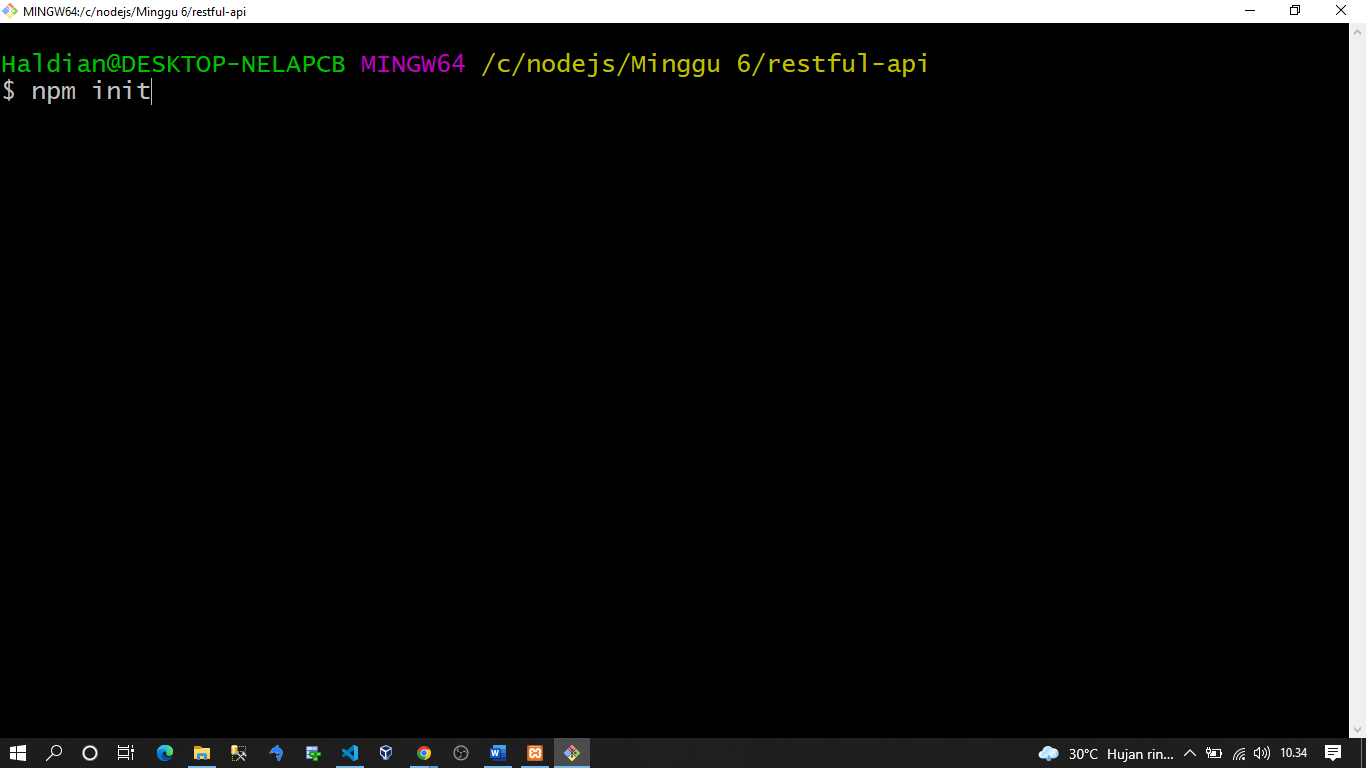
2. MySQL (driver mysql untuk node.js)

3. Body-parser (middleware untuk menghandle post body request)

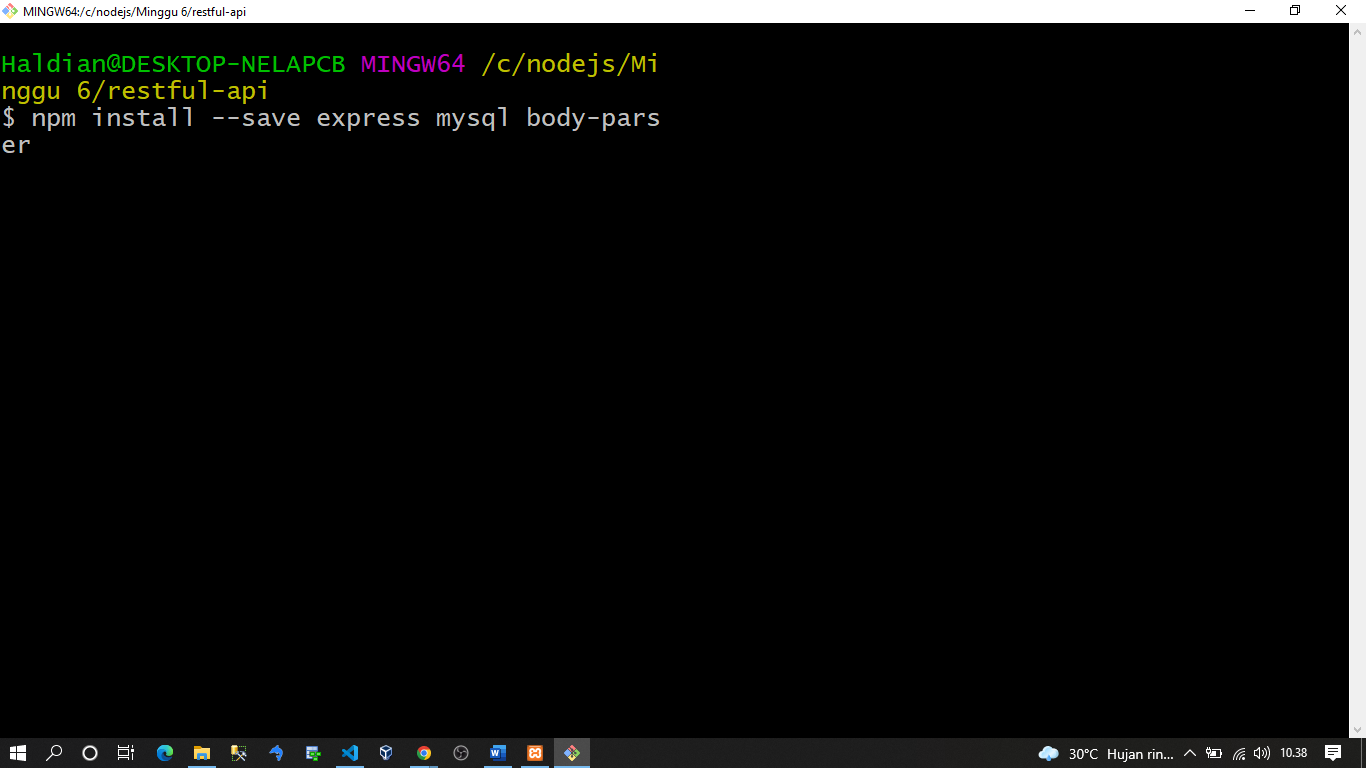
1. Kemudian kita masuk pada foler restful-api lalu klik kanan dan pilih Gitbash here seperti gambar di bawah ini



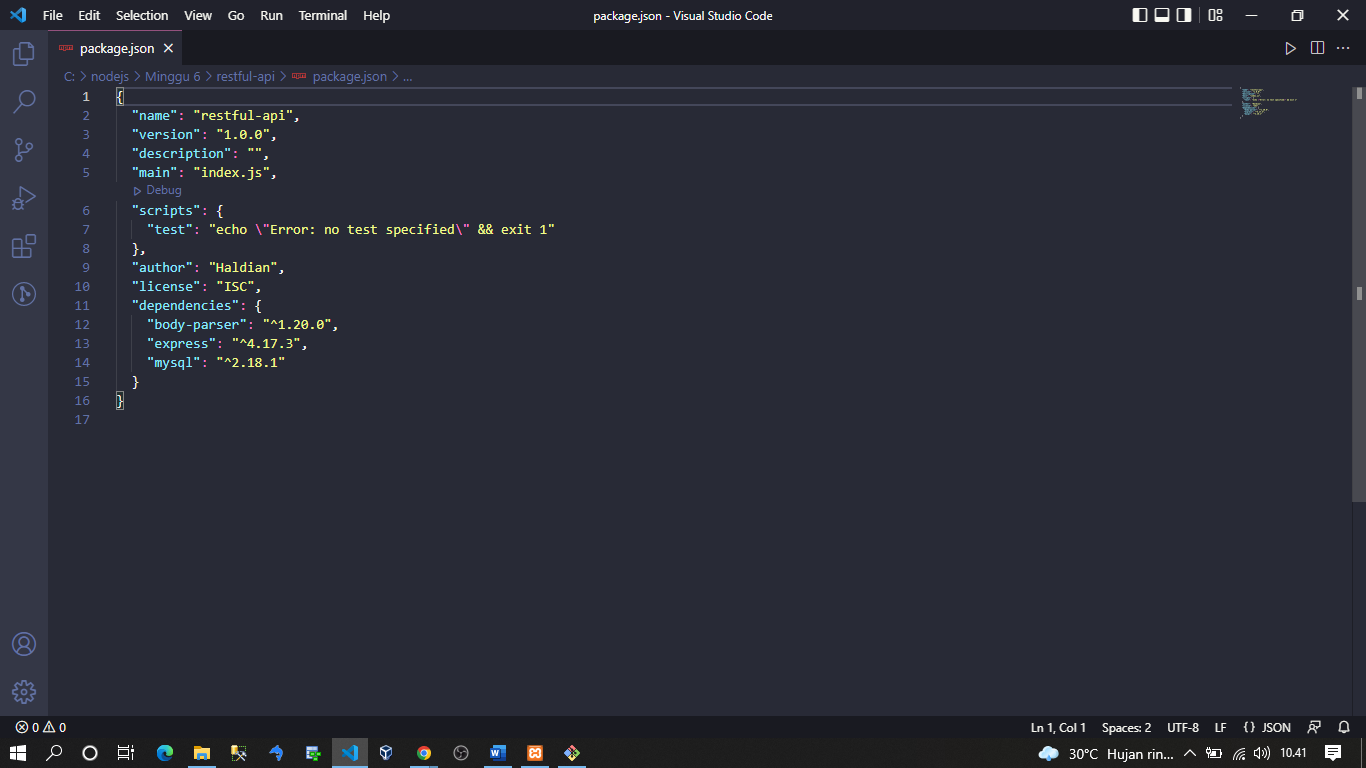
1. Kemudian kita perlu membuat package json dengan otomatis pada gitbash kita dengan mengetikan npm init seperti gambar di bawah ini



1. Selanjutnya kita Install semua dependencies denagan perintah npm install –save express mysql body-parser yang dibutuhkan dengan mengetikkan perintah seperti gambar di bawah ini

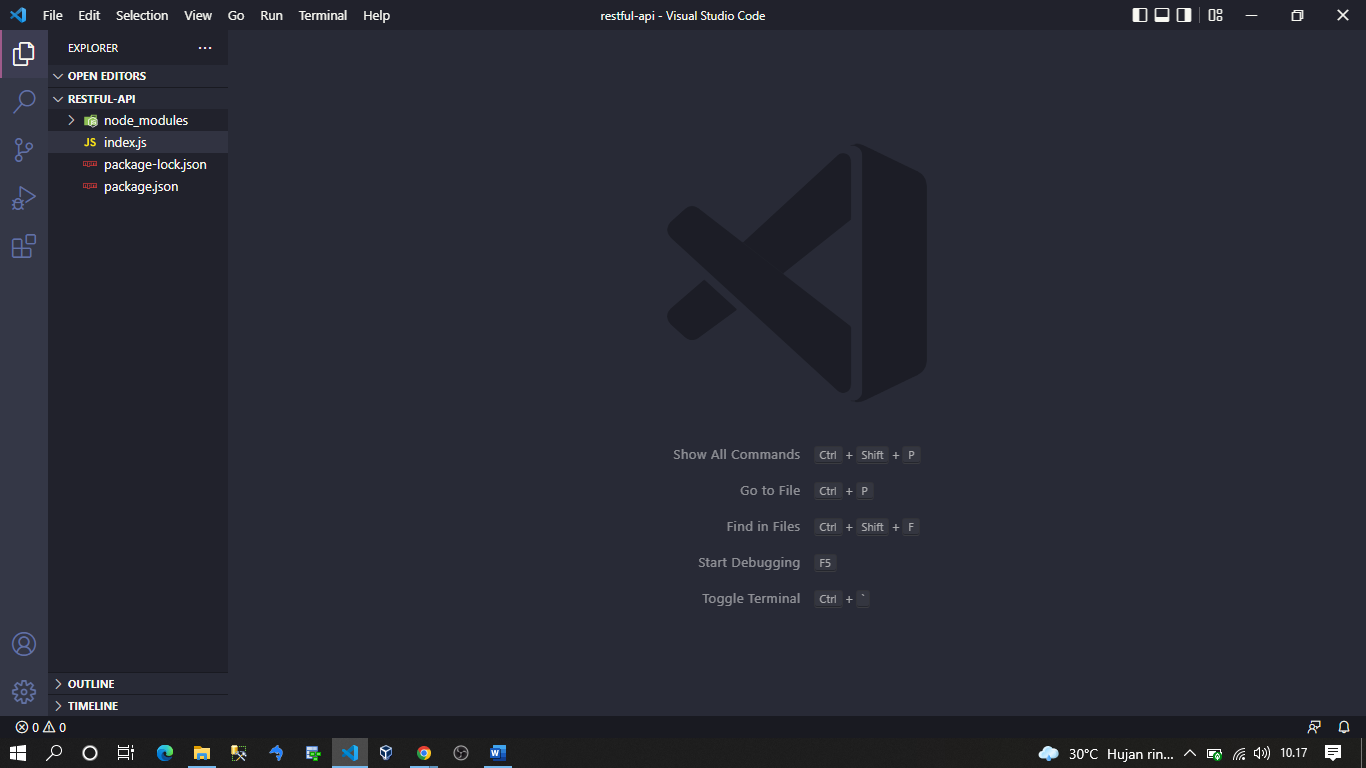


1. Jika file package.json di buka maka kan seperti gambar di bawah ini

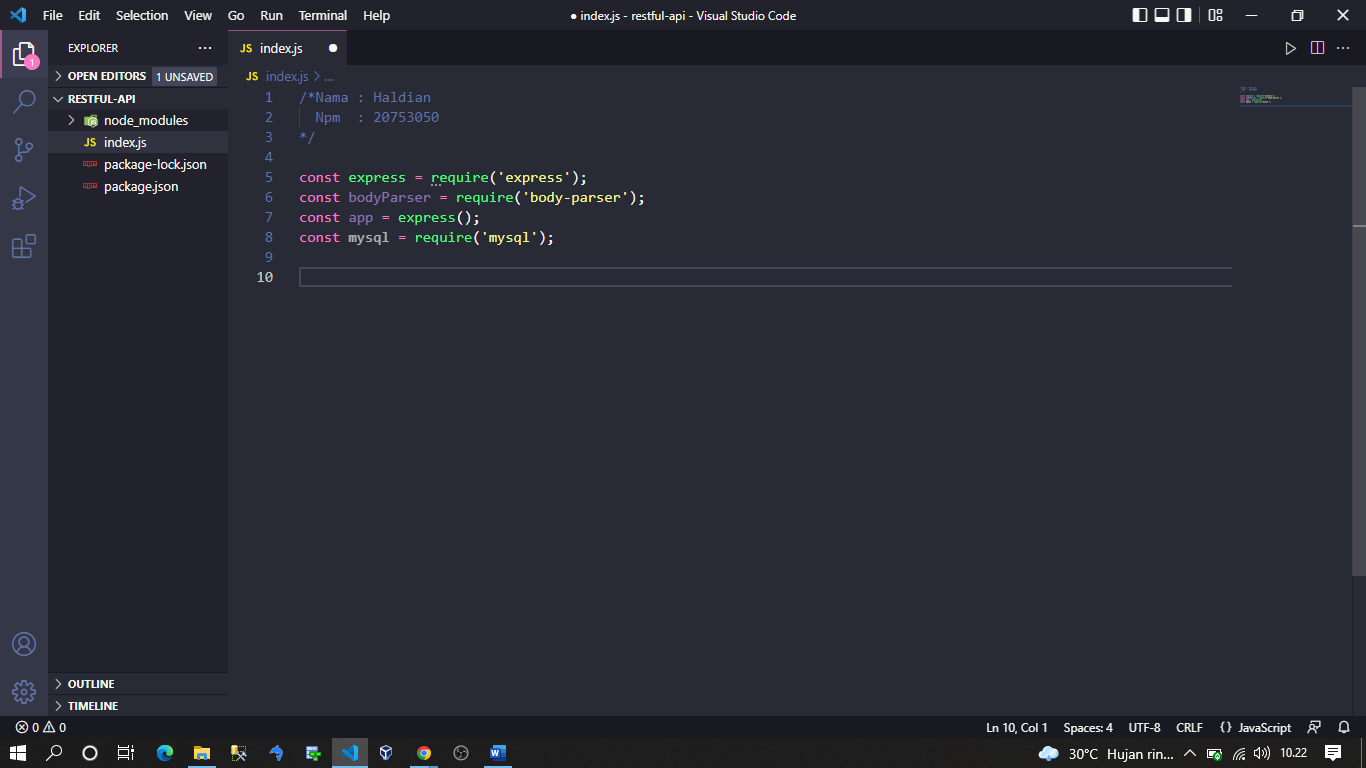


**Step #4. Buat file Index.js**

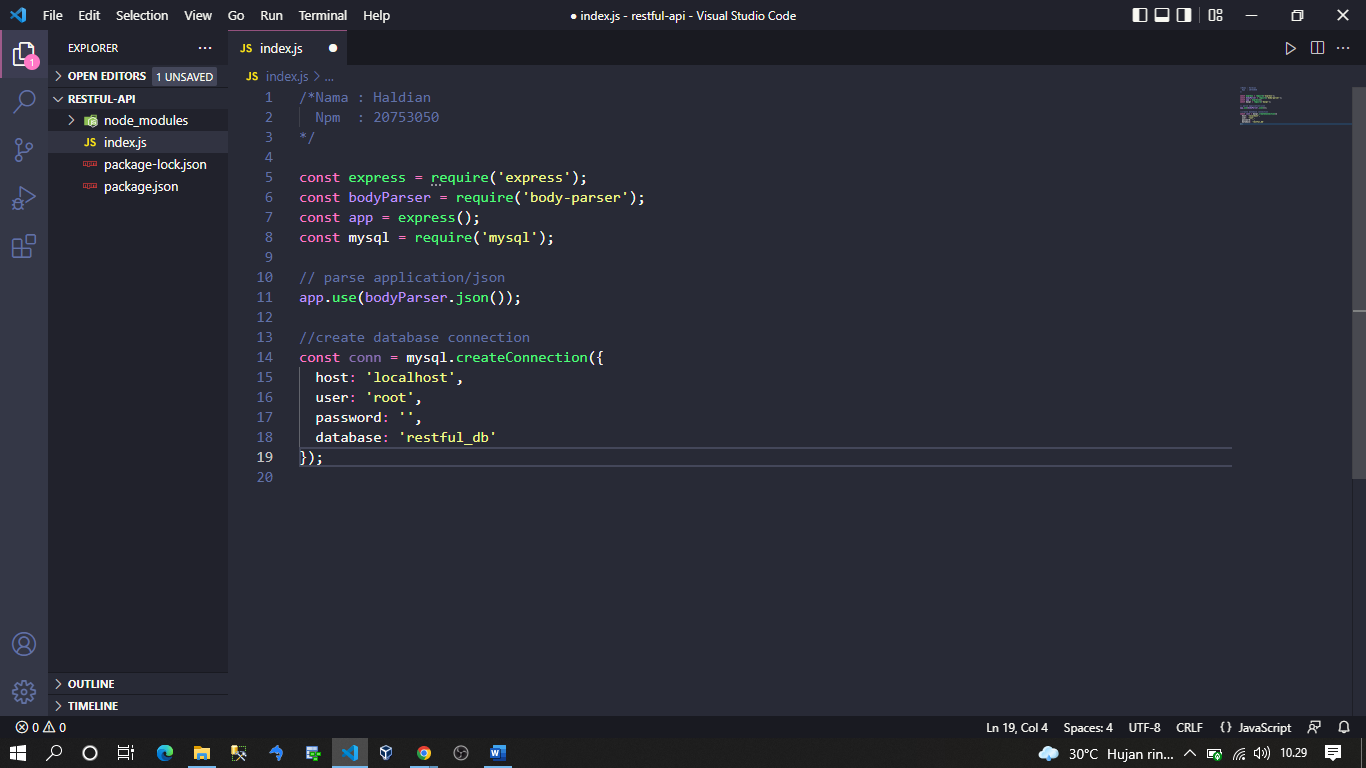
1. Pertama kita buat sebuah file dengan nama indeks seperti gambar di bawah ini



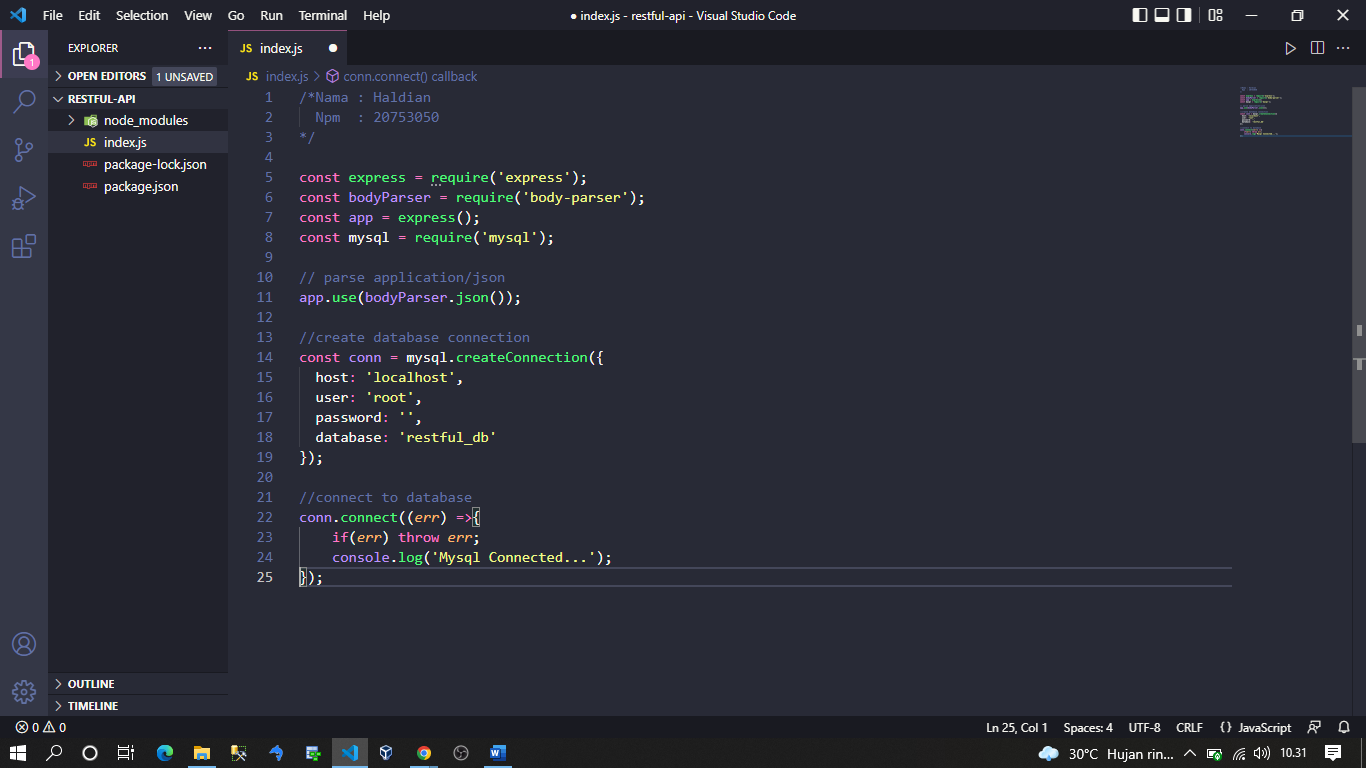
1. Kemudian kita ketikan script pada file index sepereti gambar di bawah ini
2. Yang pertama dimana kita memerlukan file kita dengan semua dependencies yang telah kita insatll sebelumnya seperti gambar di bawah ini



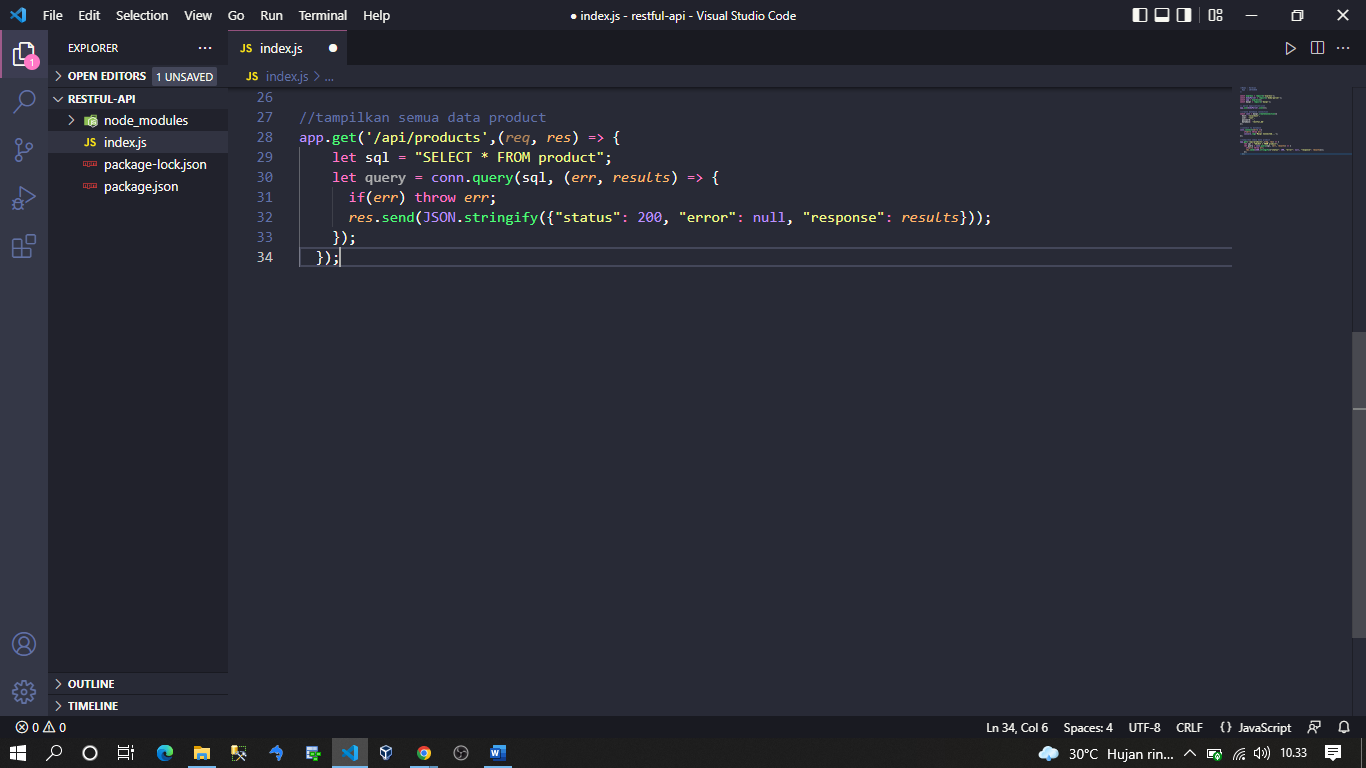
1. Kemudian kita buat script untuk uraikan json dengan app.use(bodyParser.json()) serta kita lakukan pembuatan script database untuk koneksi seperti gambar di bawah ini



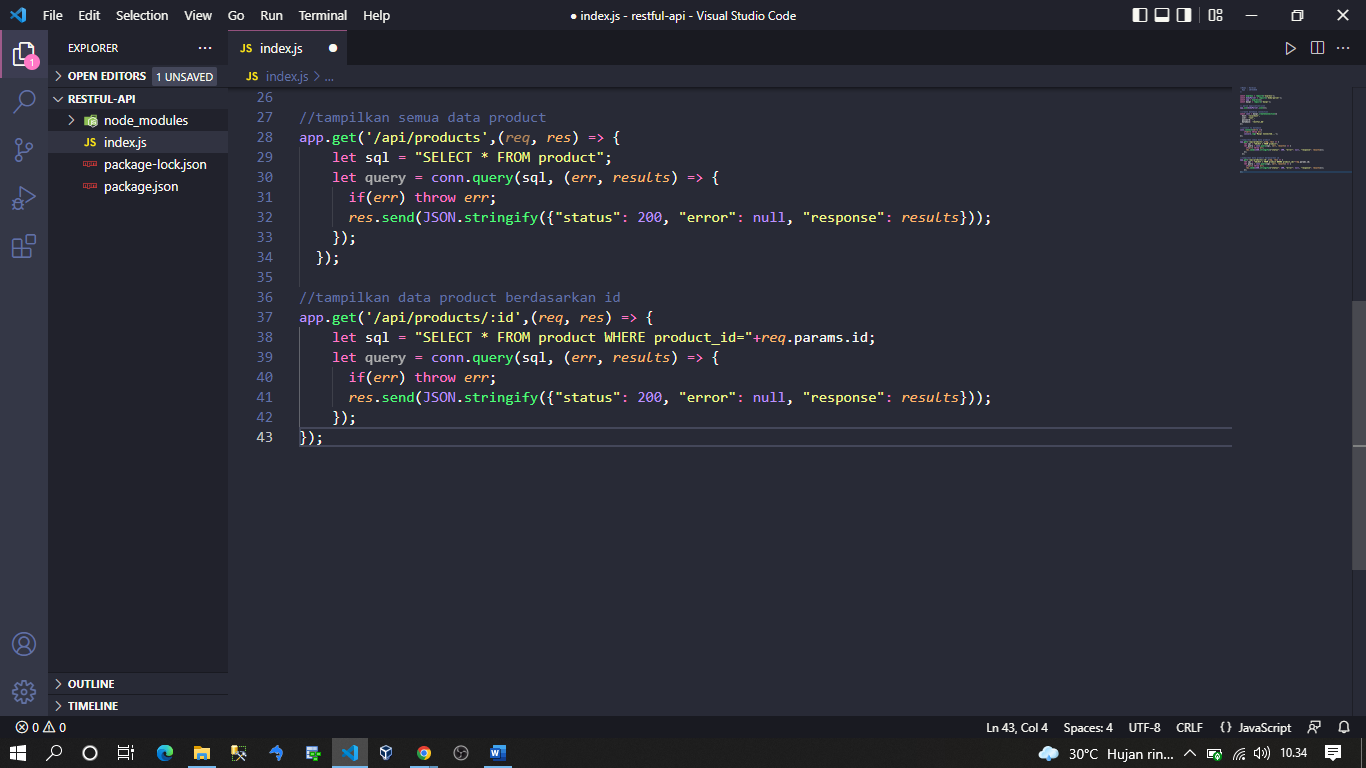
1. Kemudian buat script untuk konek ke database seperti gambar di bawah ini



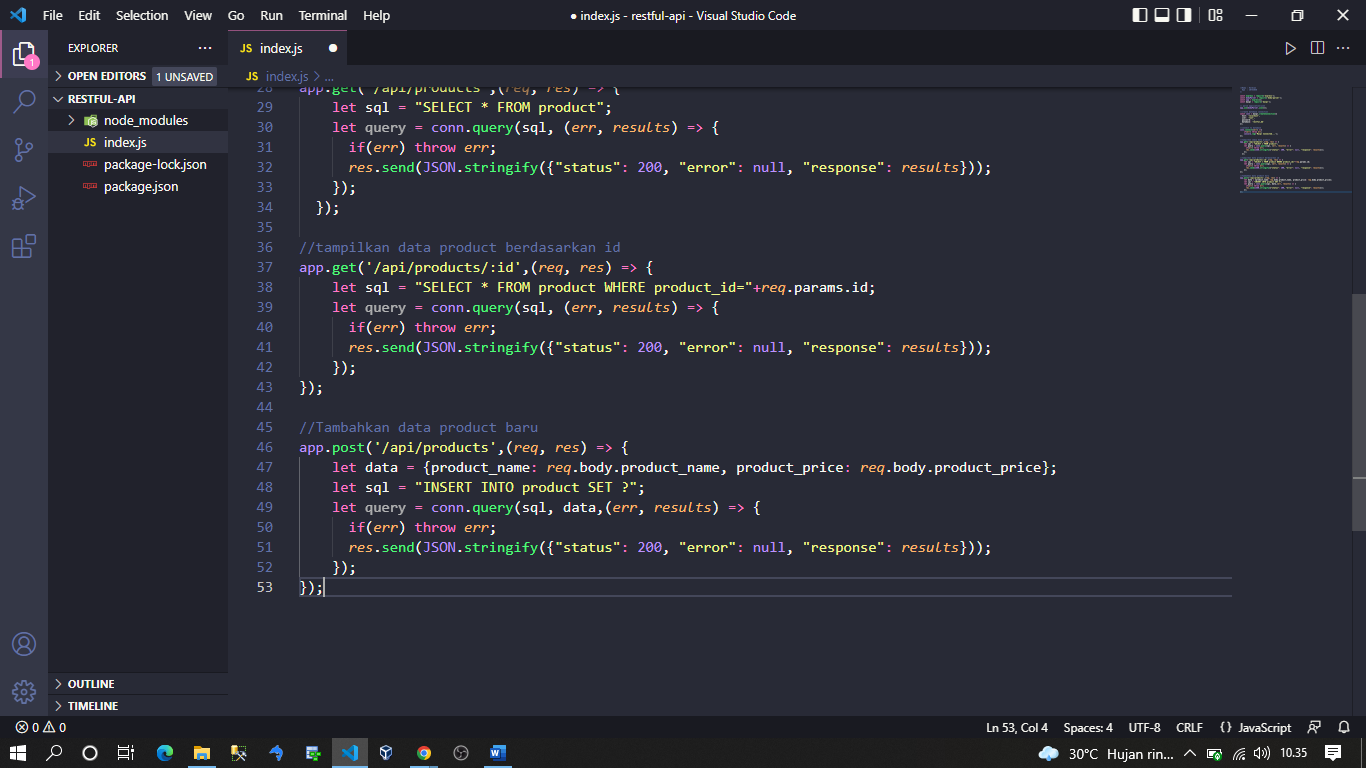
1. Kemudian kita buat script query untuk mengambil semua data product pada database kita dengan api seperti gambar di bawah ini



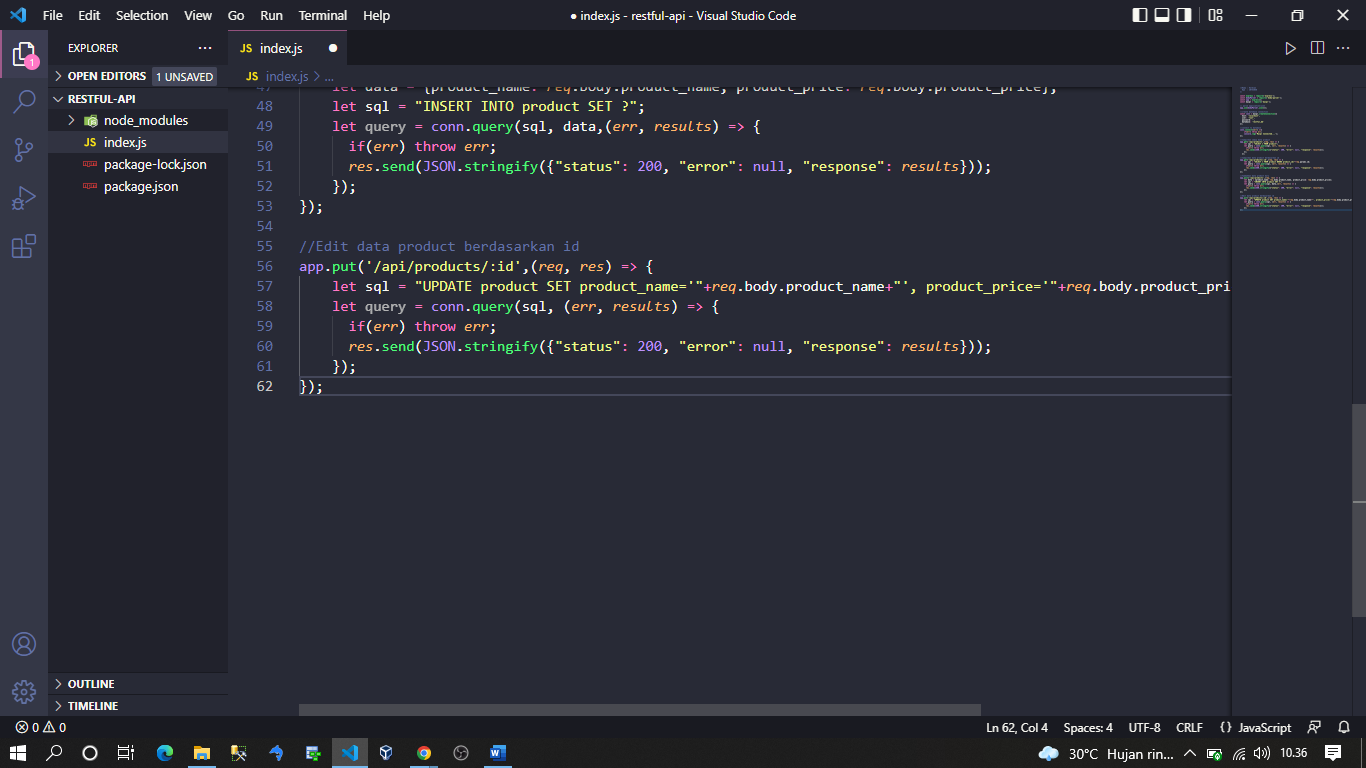
1. Kemudian kita buat script query untuk menampilkan semua data product berdasarkan ide pada database kita dengan api seperti gambar di bawah ini



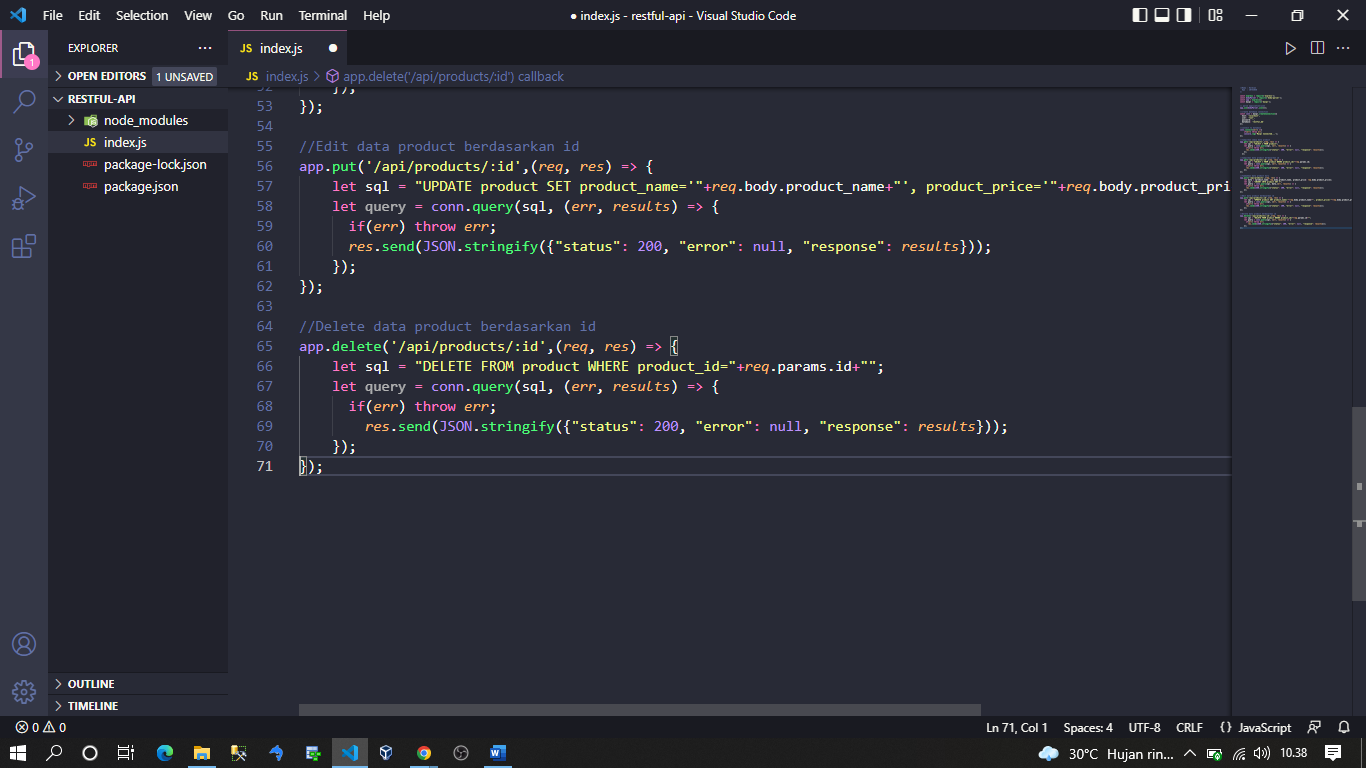
1. Kemudian kita buat script untuk menambah data product baru ke dalam database kita dengan api seperti gambar di bawah ini



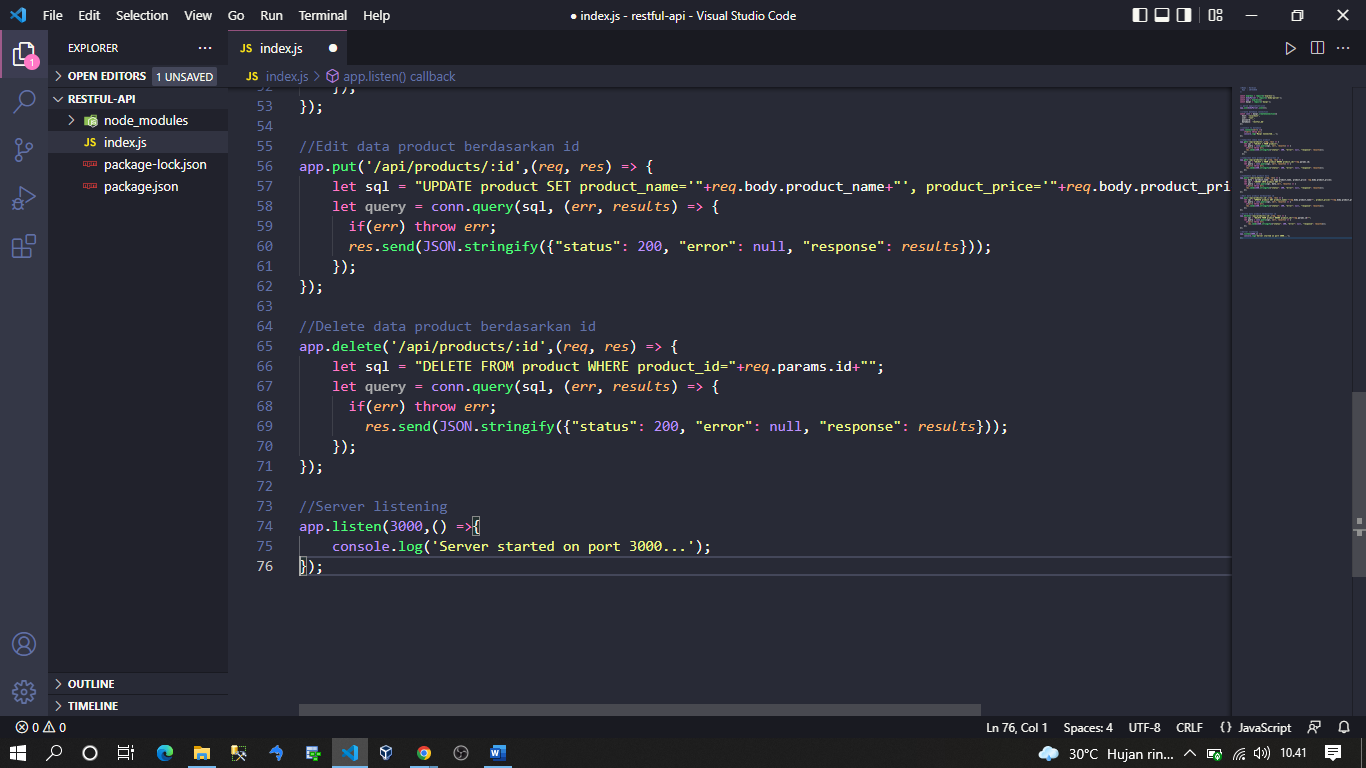
1. Kemudian kita buat script untuk mengedit data product berdasarkan ide pada database kita dengan api seperti gambar di bawah ini



1. Kemudian kita buat script untuk delete data product berdasarkan ide pada database kita dengan api seperti gambar di bawah ini

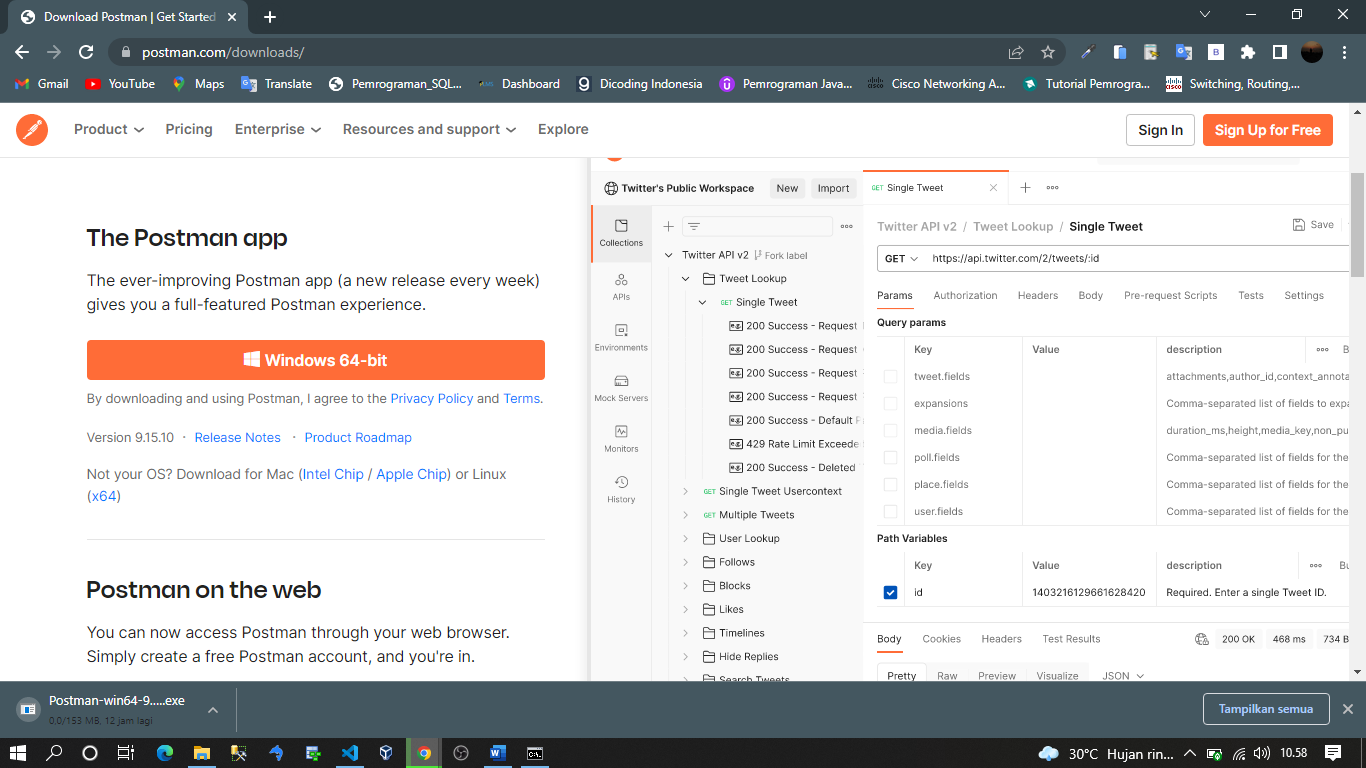


1. Selanjutnya kita buat script untuk server local untuk menjalankan file kita seperti gambar di bawah ini

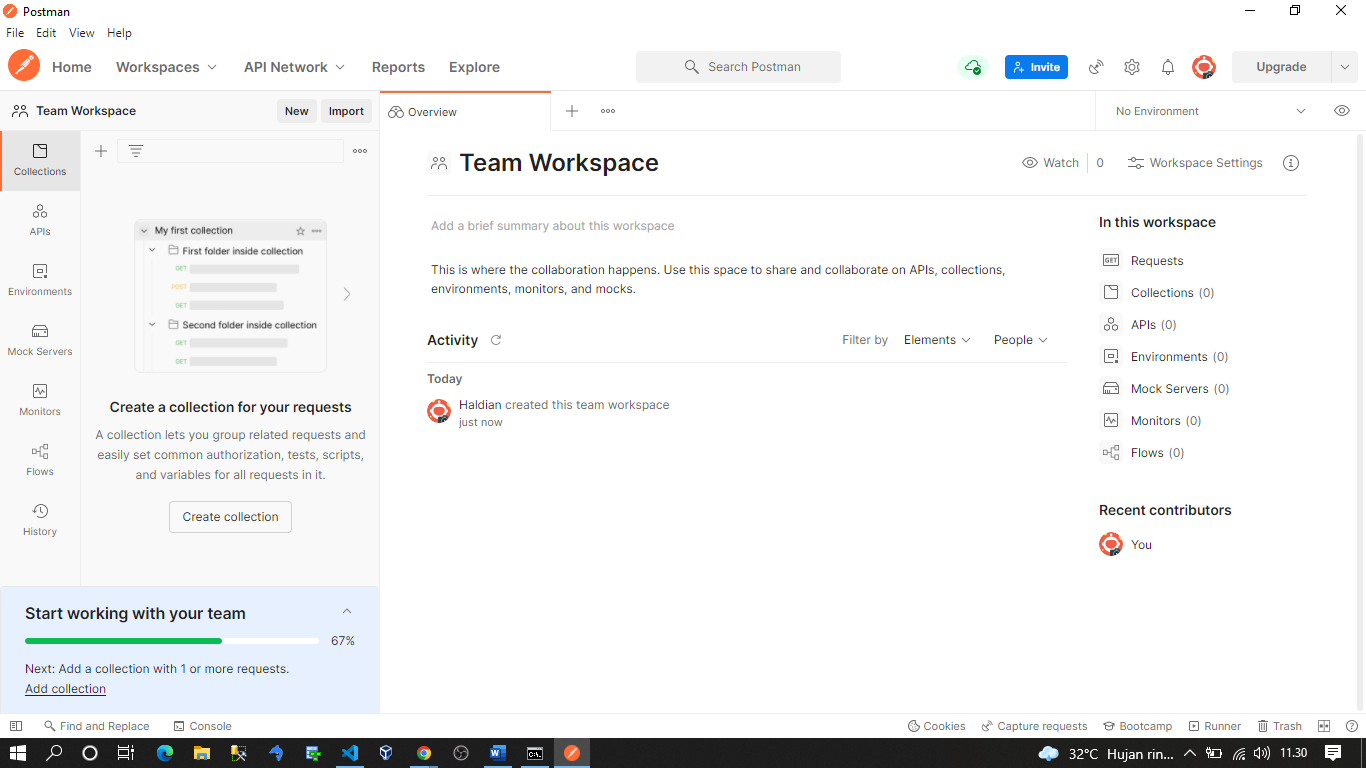


**Step #5. Testing**

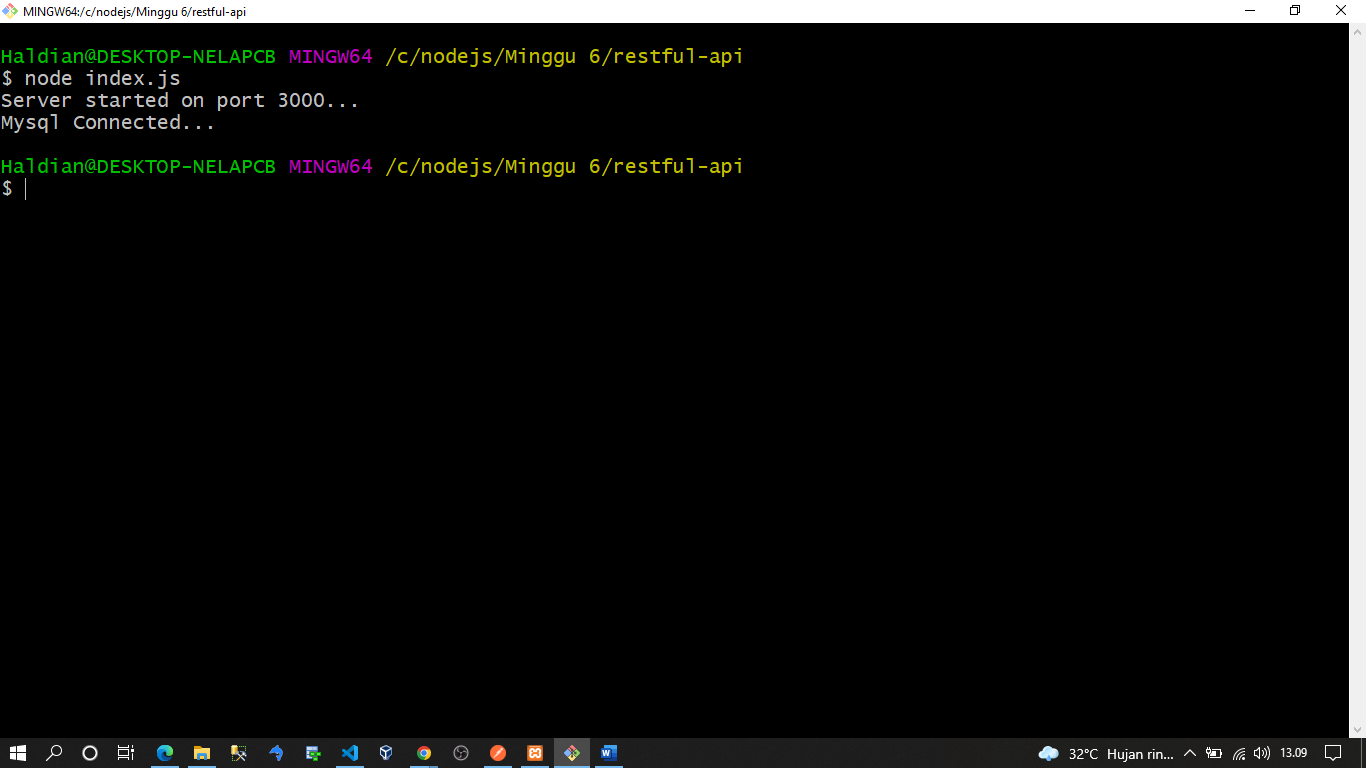
1. Untuk melakukan uji coba API yang telah kita buat yang pertama yang harus kita lakukan yaitu mendownload POSTMAN pada situs resminya yaitu https://www.getpostman.com/ kemudian kita download seperti gambar di bawah ini



1. Setelah kita download dan install maka ini dia tampilan dari POSTMAN kita seperti gambar di bawah ini



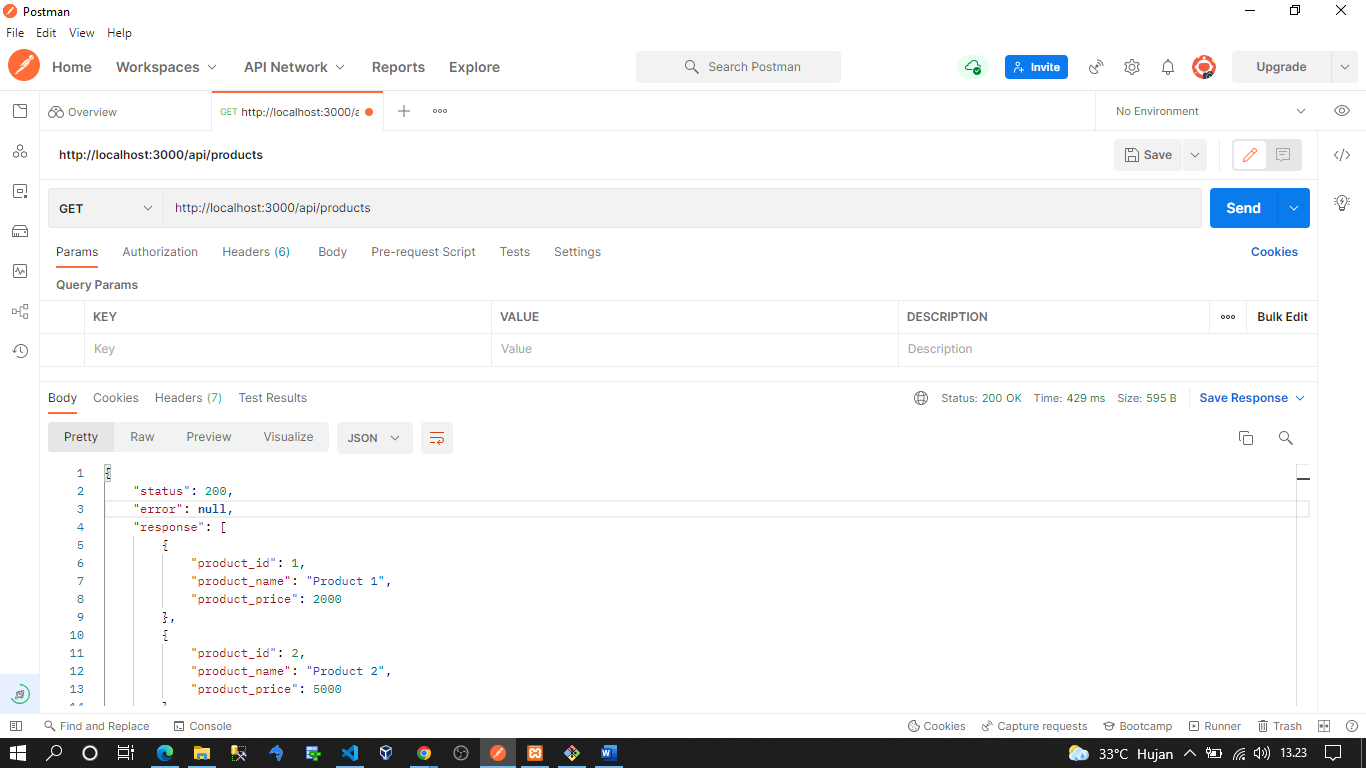
1. Selanjutanya kita coba run di gitbash kita dengan cara node index.js jika seperti gambar di bawah ini



1. selanjutnya kita uji EndPoint-nya satu per satu.

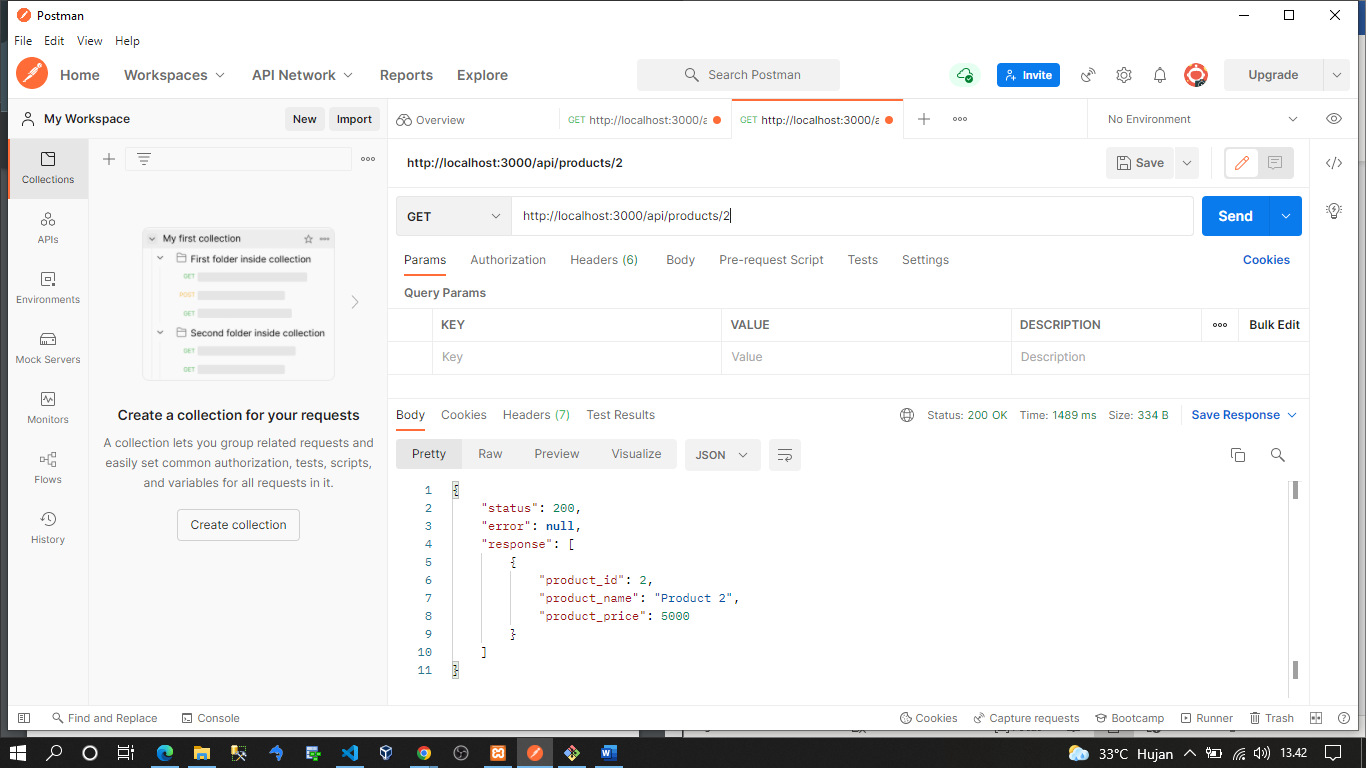
**#1. Get All Product (GET)**

Pertama kita buka POSTMAN kita kemudian kita ketikan URL <http://localhost:3000/api/products> kemudian kita pilih Method GET, dan klik tombol Send kemudian kita pilih JSON, jika seperti gambar di bawah maka data berjalan dengan baik seperti gambar di bawah ini.



**#2. Get Single Product (GET)**

Ketika URL <http://localhost:3000/api/products/2> masukan pada POSTMAN dan kita pilih Methodm GET kemudian kita Send maka terlihat data product yang ditampilkan pada product\_id=’2’ sesui dengan parameter URL seperti gambar di bawah ini.



**#3. Create New Product (POST)**

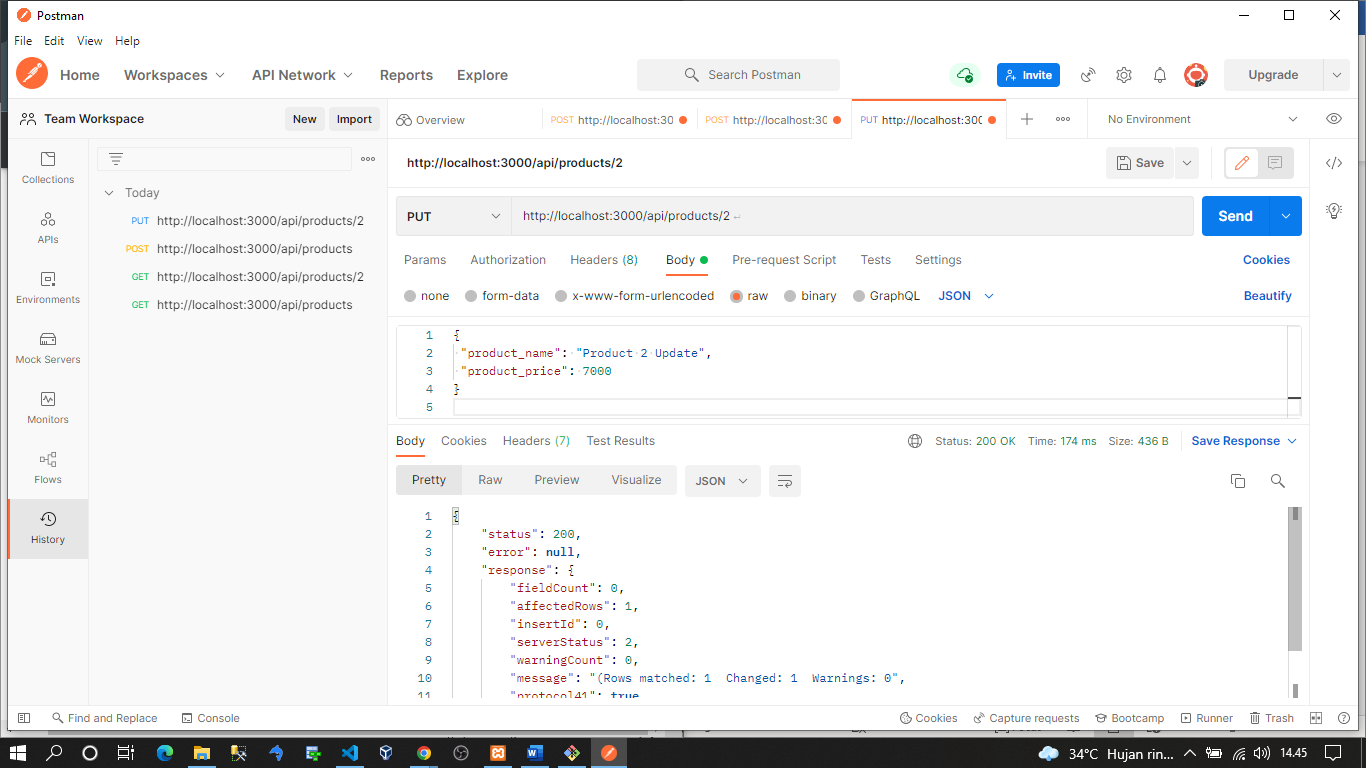
Selanjutnya kita ketikan URL <http://localhost:3000/api/product> pada Method POST kemudian kita masukan data { "product\_name": "Product 6 Added", "product\_price": 6000 } pada kolom JSON(application/json) kemudian kita Send, jika kita perhatikan pada bagian response, terdapat “affectedRows”: 1, dan “insertId”: 6. Itu berarti terdapat satu data yang diinsert ke database dengan product\_id=’6’.



**#4. Update Product (PUT)**

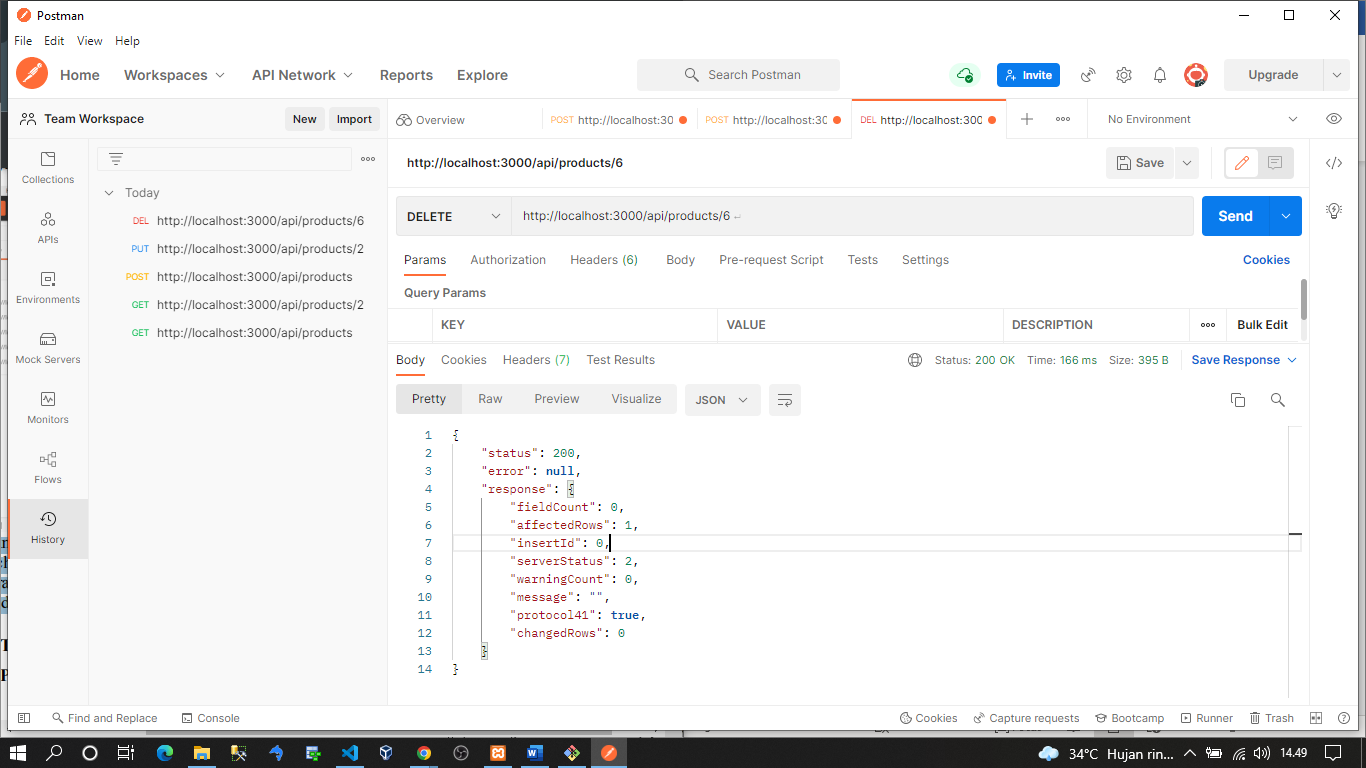
Selanjutnya kita ketikan URL <http://localhost:3000/api/products/2> kemudian kita pilih Method PUT dan kita masukan data pada a kolom JSON(application/json)

{ “product\_name”: “Product 2 Update”, “product\_price”: 7000 } setelah itu kita klik tombol Send. Jika kita perhatikan pada bagian response, terdapat “affectedRows”: 1, dan “changedRows”: 1. Itu berarti terdapat satu data yang diupdate ke database dengan product\_id=’2’ sesuai dengan parameter pada URL.



**#5 Delete Product (DELETE)**

Selanjutnya kita ketikan URL <http://localhost:3000/api/products/6> kemudian kita pilih Method Delete kemudian klik tombol Send, Jika kita perhatikan pada bagian response, terdapat “affectedRows”: 1, “insertId”: 0 ,dan “changedRows”: 0. Itu berarti terdapat satu data yang dihapus (delete) ke database dengan product\_id=’6’ sesuai dengan parameter pada URL maka Akan terlihat hasilnya seperti gambar di bawah ini.



E. Daftar Pustaka

1. TSJ TV

www.youtube.com/c/TSJTvPro