Nama : **Dini Tri Widianingsih**

No. Peserta : **FSDO003ONL007**

Link Github : dinni26/DiniFinalProject_PaymentAPI (github.com)

Link Heroku : Biodata (dini-finalproject.herokuapp.com)

PANDUAN PEMBUATAN FINAL PROJECT

CREATE DATABASE

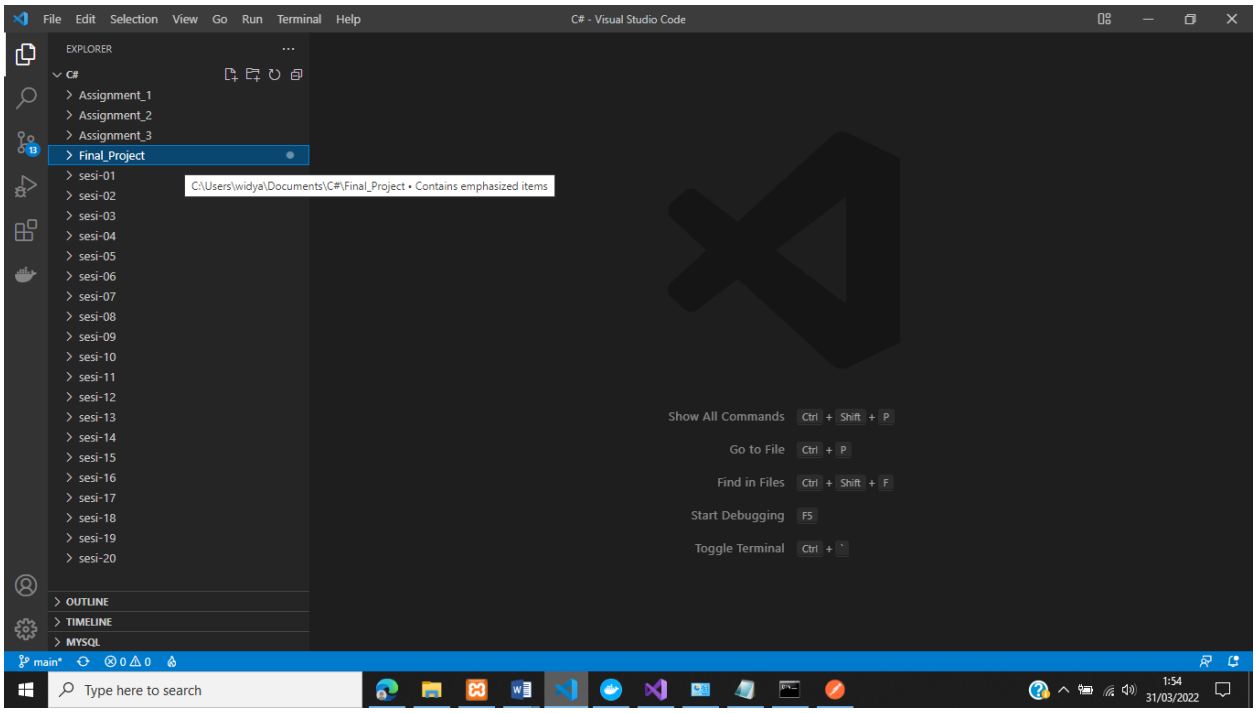
- Belum Online > Jalankan via cmd
  CREATE database payment;

- Jika online, Silahkan buka link https://remotemysql.com/phpmyadmin/
  Lalu masukan :

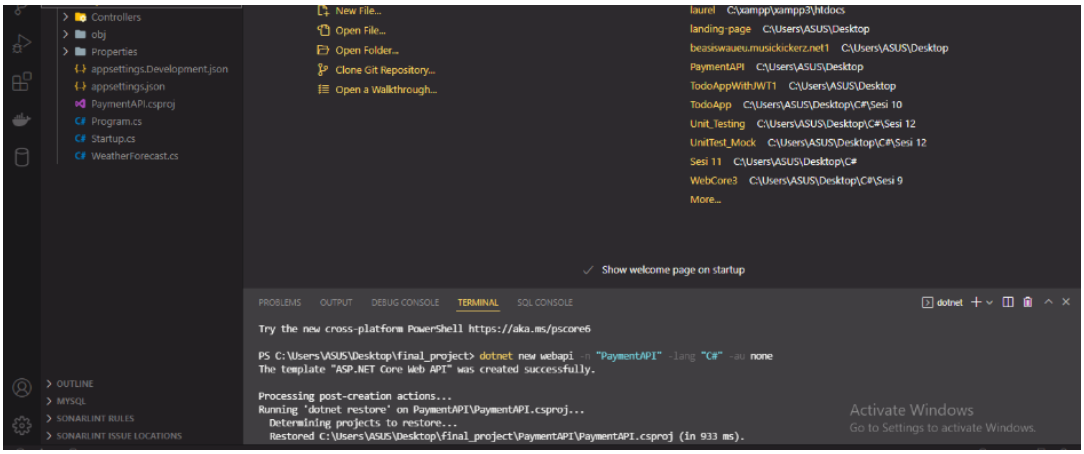
```
- "server=remotemysql.com; port=3306; database=qqfRbkF42v;
  username=qqfRbkF42v; password=obO8WIfgL9"
```

PEMBUATAN

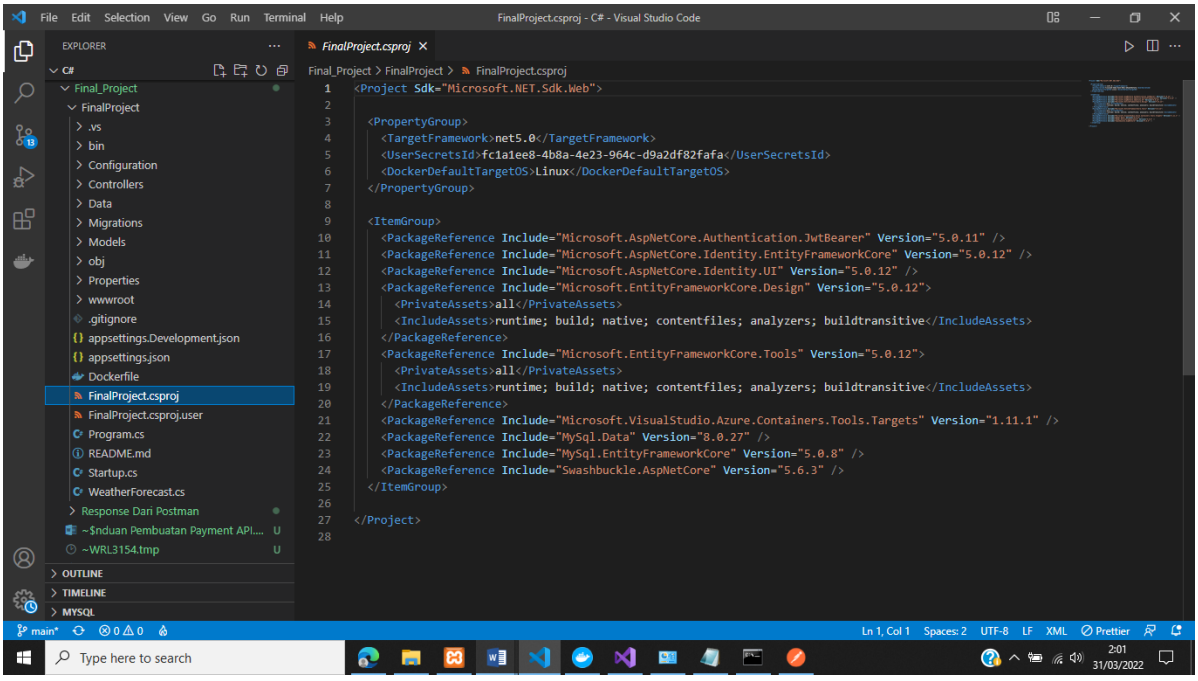
- Buka Visual Studio Code > Create folder Final_Poject



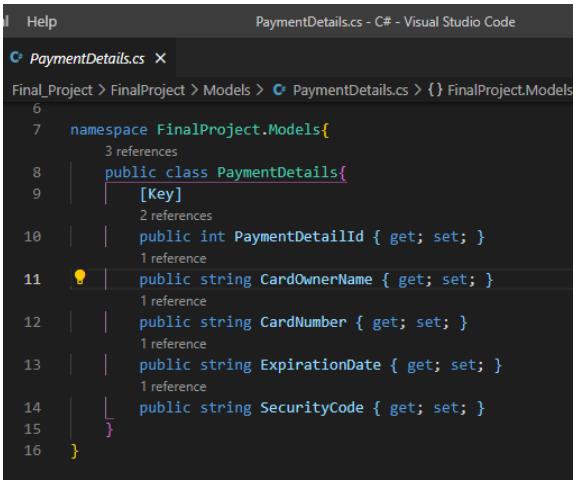
- Inisiasi project di dalam folder tersebut > ***dotnet new webapi -n "Final_Project" -lang "C#" -au none*** lalu akan mucul folder – folder baru



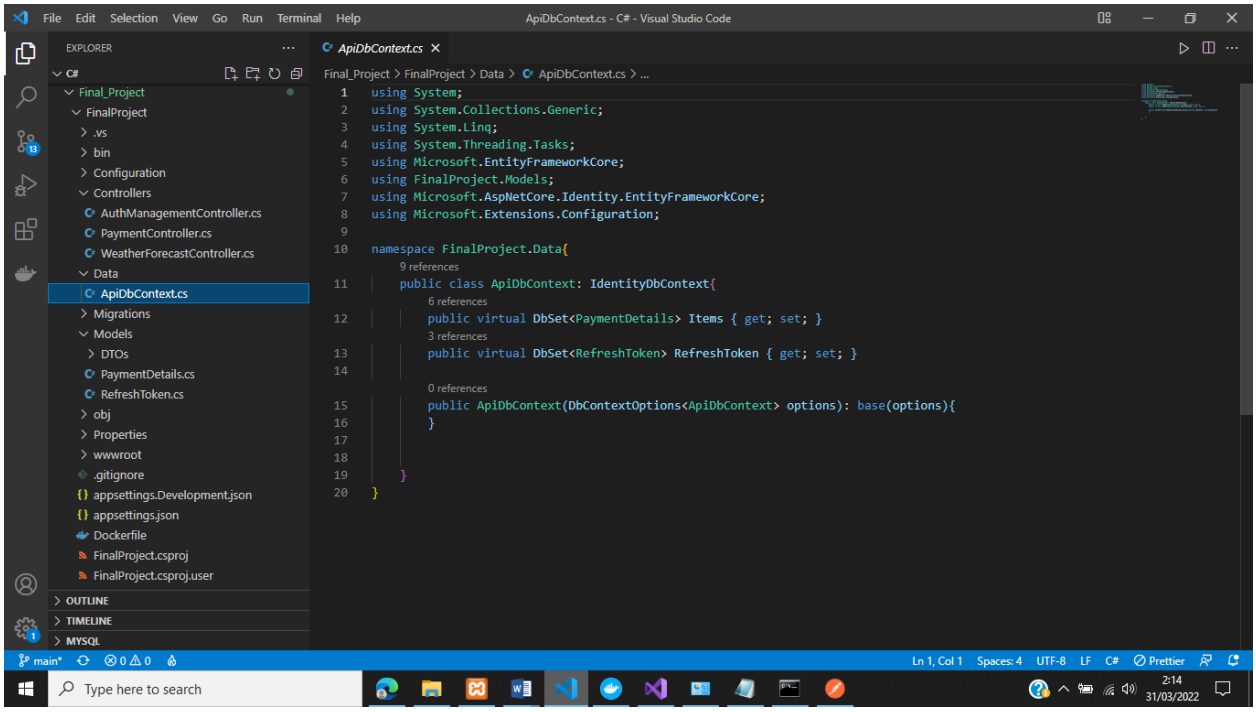
- Install package yang dibutuhkan
  - dotnet add package NuGet.CommandLine.XPlat --version 5.11.0
  - dotnet add package Microsoft.EntityFrameworkCore.Tools --version 5.0.11
  - dotnet add package Microsoft.EntityFrameworkCore.Design --version 5.0.11
  - dotnet add package Microsoft.AspNetCore.Authentication.JwtBearer --version 5.0.11
  - dotnet add package Microsoft.AspNetCore.Identity.EntityFrameworkCore --version 5.0.11
  - dotnet add package Microsoft.AspNetCore.Identity.UI --version 5.0.11
  - dotnet add package Microsoft.Mysql.Data --version 8.0.28
  - dotnet add package Microsoft.Mysql.EntityFrameworkCore --version 5.0.10



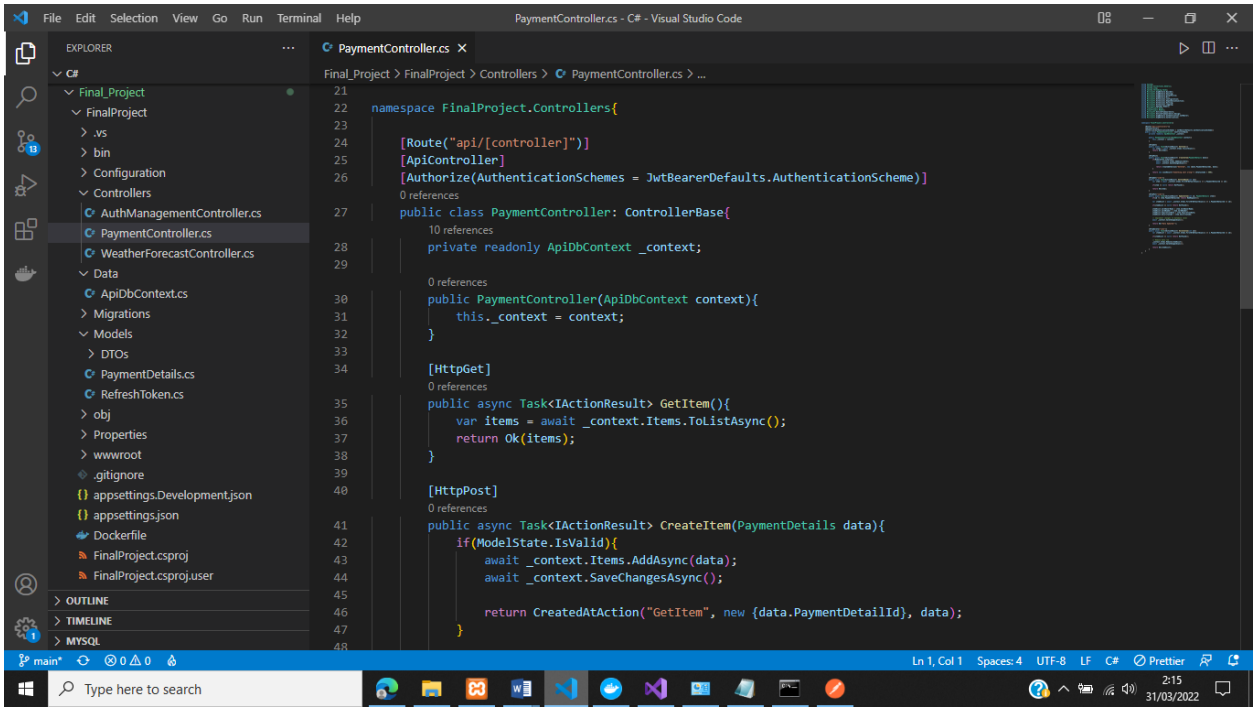
- Buat folder Models, lalu tambahkan file PaymentDetails.cs di dalam folder tersebut
  Location : Final_Project/Models/PaymentDetails.cs



- Buat folder Data, lalu tambahkan file ApiDBContext
  Location : Final_Project/Data/ApiDbContext.cs

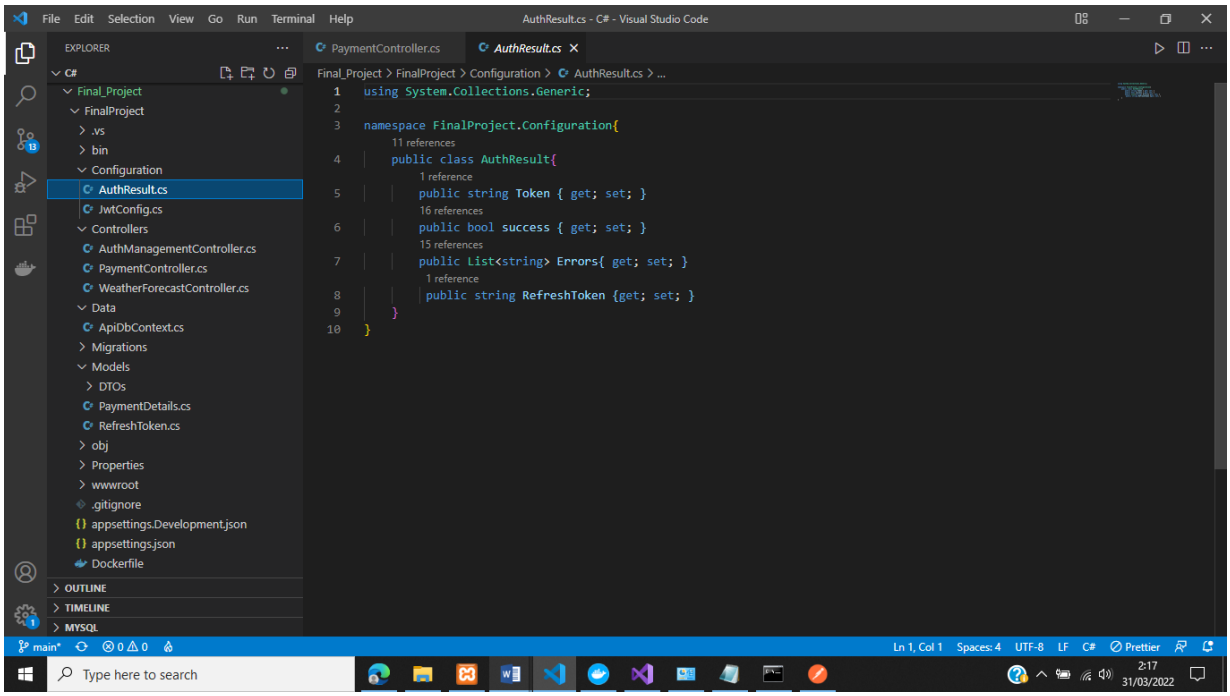


- Buat folder Controllers, lalu lanjutkan build PaymentDetailController untuk method post, get, put, delete
  Location : Final_Project/Controllers/PaymentController.cs

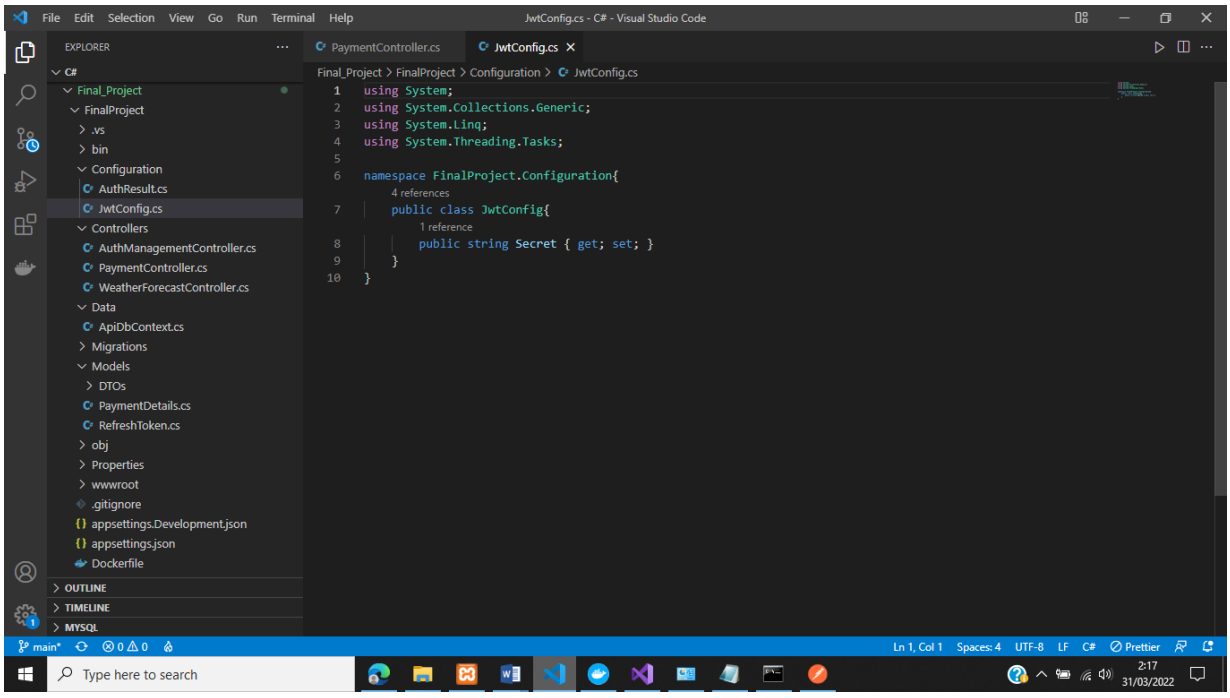


Penambahan JWT Token dan Refresh Token

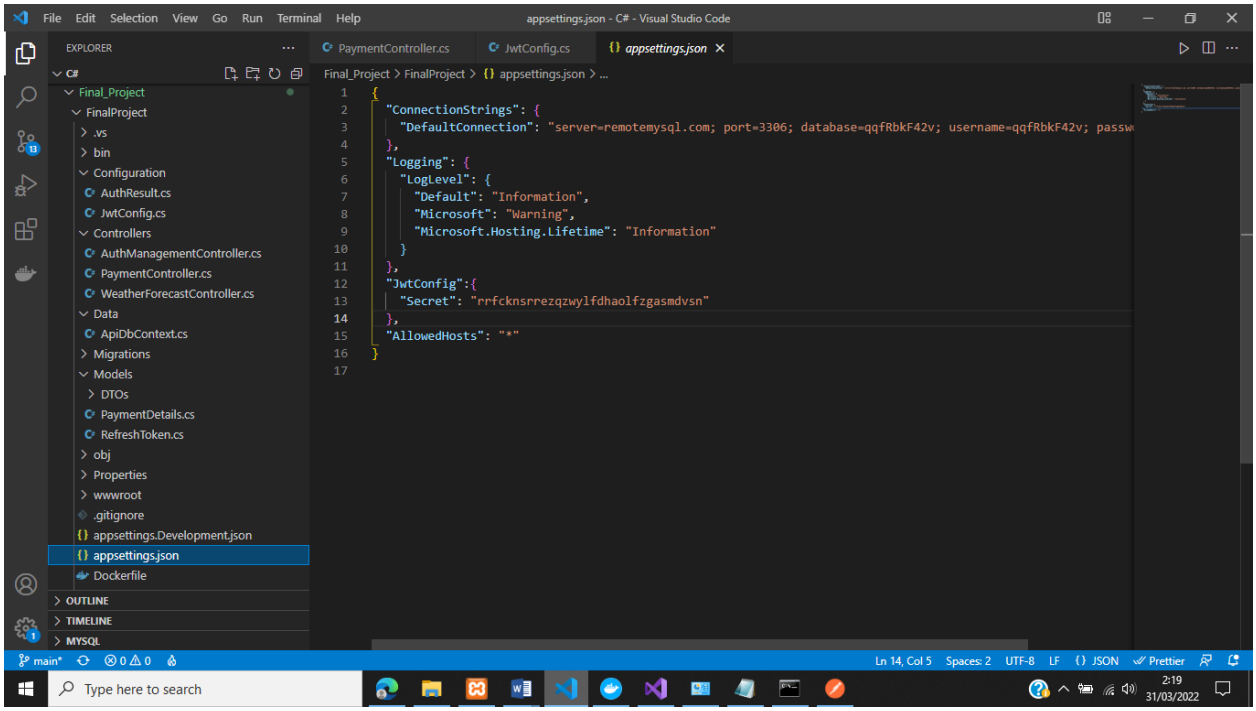
- Buat folder Configuration, lalu tambahkan file AuthResult.cs dan JwtConfig.cs:
  Location : Final_Project/Configuration/AuthResult.cs



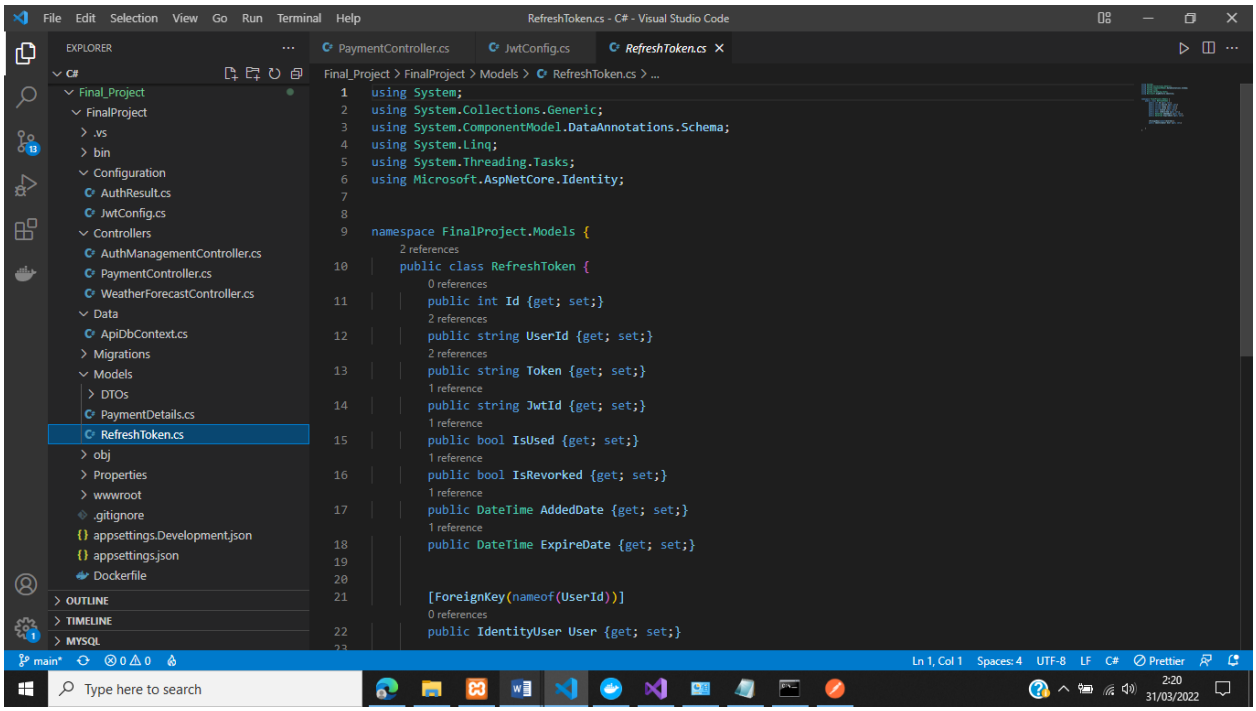
Location : Final_Project/Configuration/JwtConfig.cs



- Pada file appsettings.json tambahkan JwtConfig.Secrets
  Location : Final_Project/appsettings.json

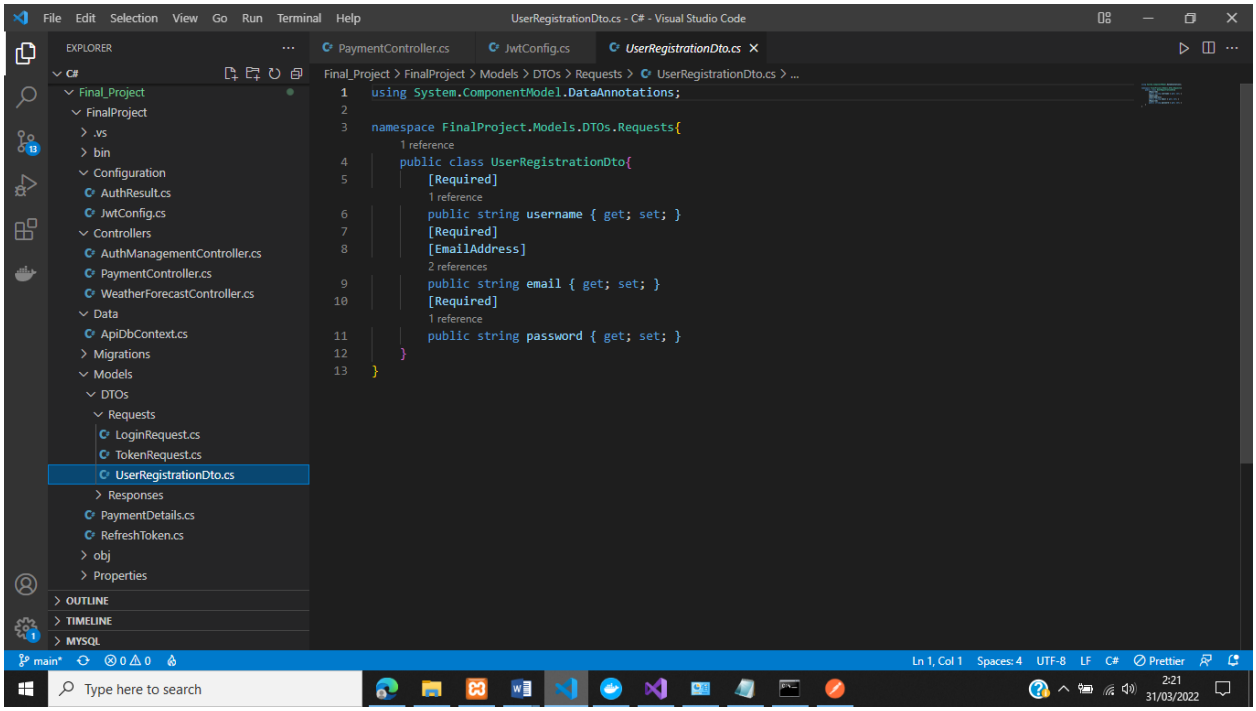


- Buat file baru RefreshToken.cs yang terdapat pada folder Models
  Location : Final_Project/Models/RefreshToken.cs

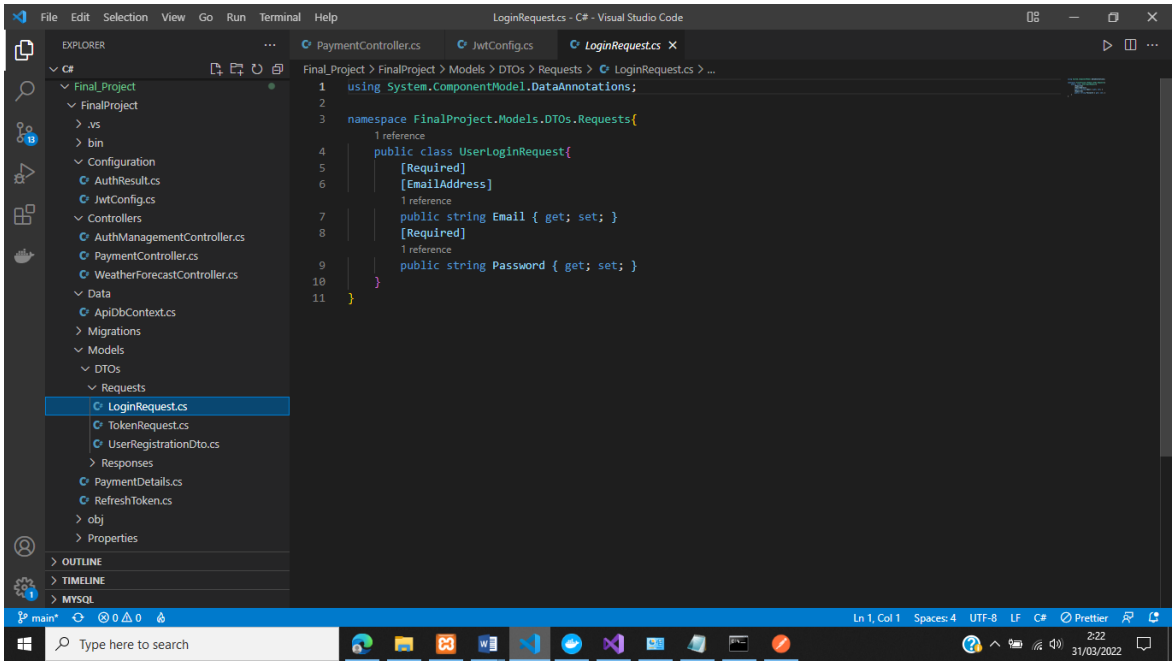


- Buat folder baru bernama DTOs yang di dalamnya terdapat juga folder Request & Response
  - **Folder Request**

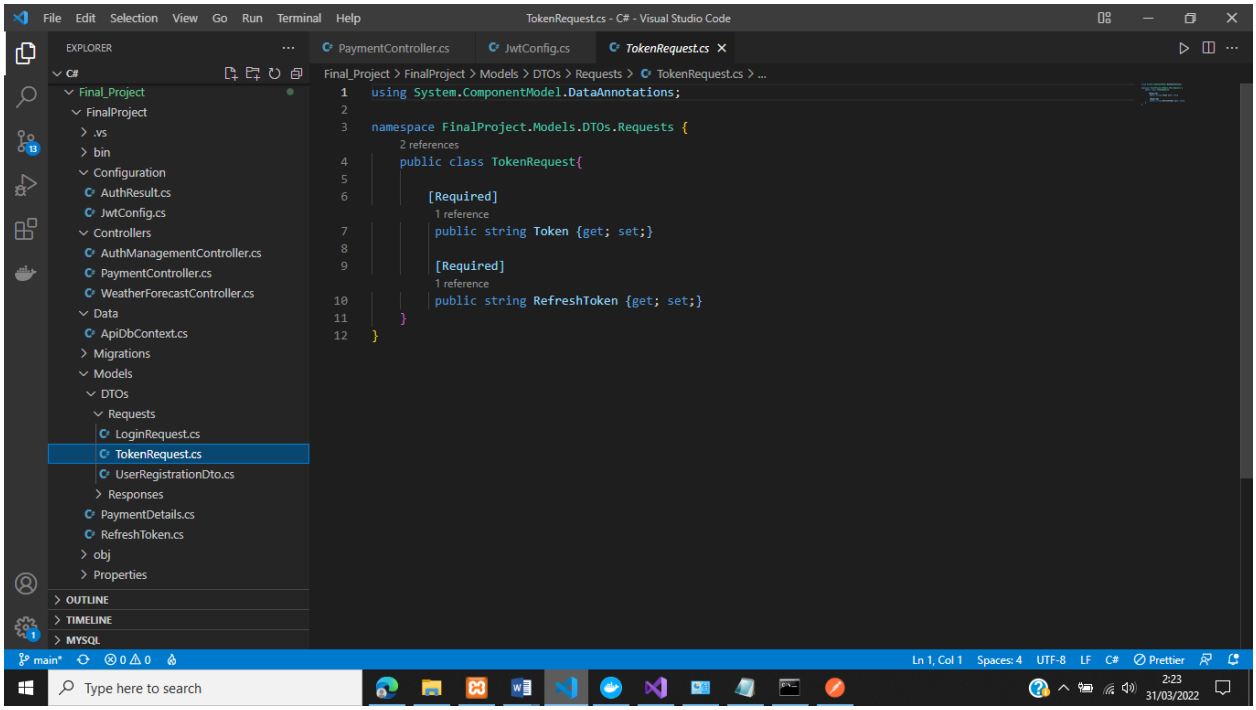
    Location : Final_Project/Models/DTOs/Requests/UserRegistrationDto.cs



Location : Final_Project/Models/DTOs/Requests/LoginRequest.cs

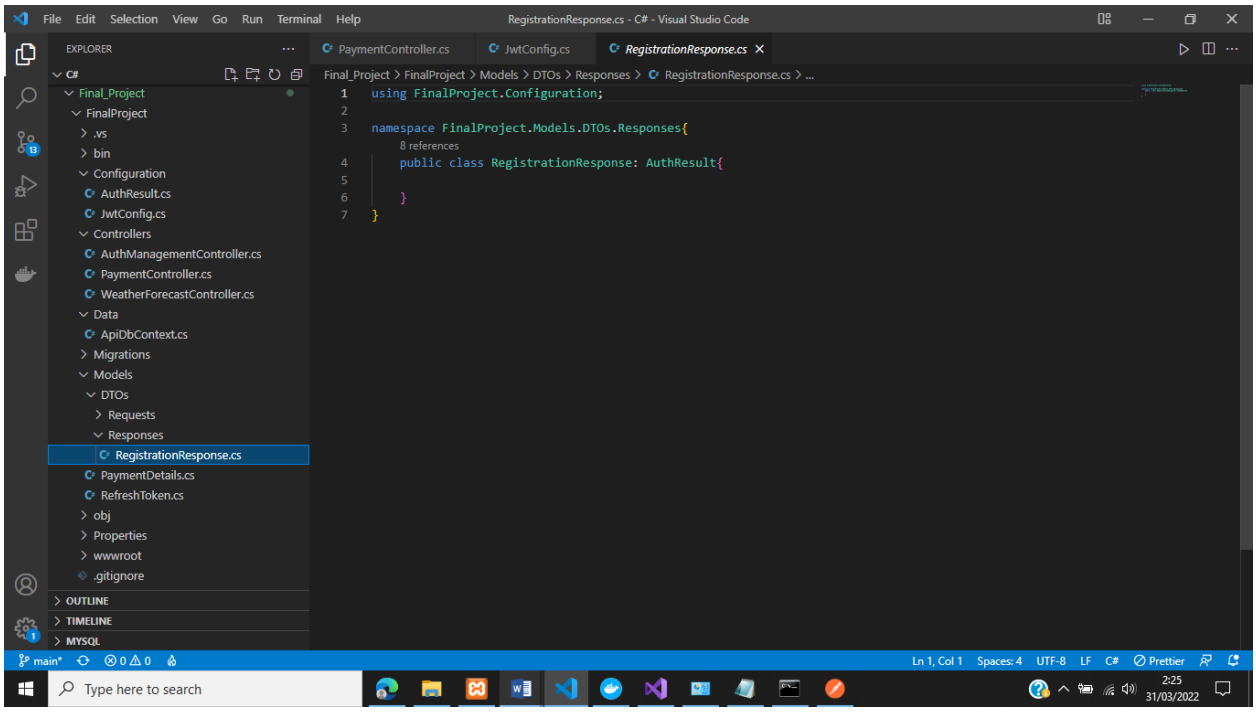


Location : Final_Project/Models/DTOs/Requests/TokenRequest.cs

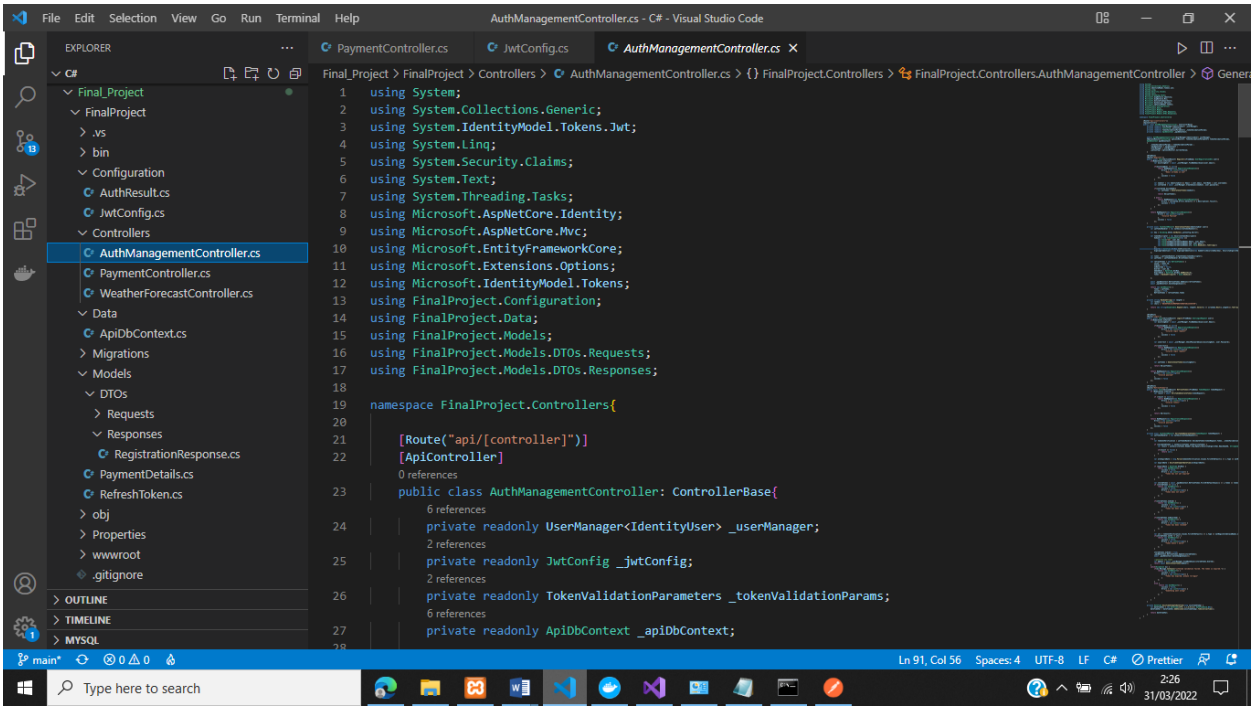


- **Folder Response**

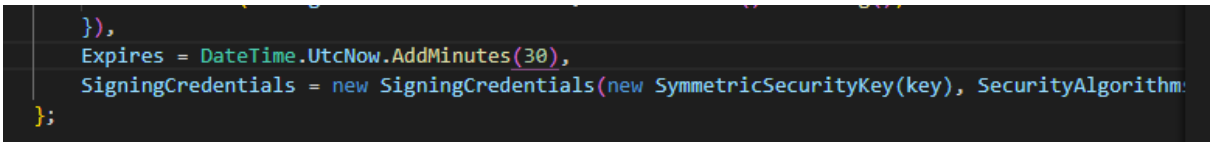
  Location : Final_Project/Models/DTOs/Response/RegistrationResponse.cs



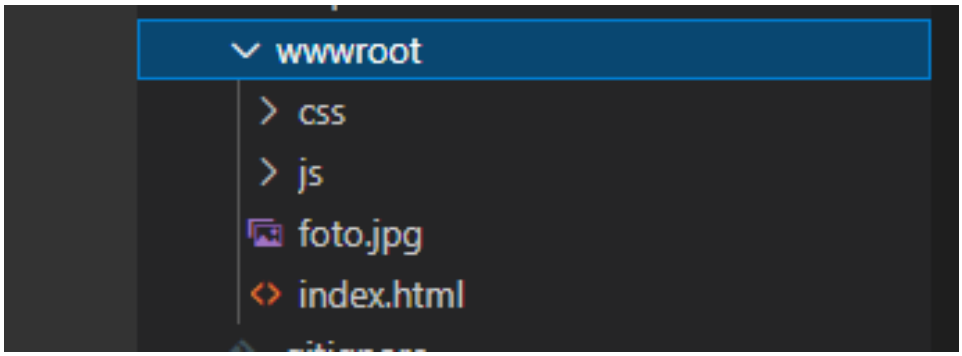
- Tambahkan file AuthManagement pada folder Controllers
  Location : Final_Project/Controllers/AuthManagementController.cs



- JWT dengan batas waktu token 30 menit
  Set waktu expired selama 30 menit pada fungsi GenerateJwtToken di AuthManagementController.cs

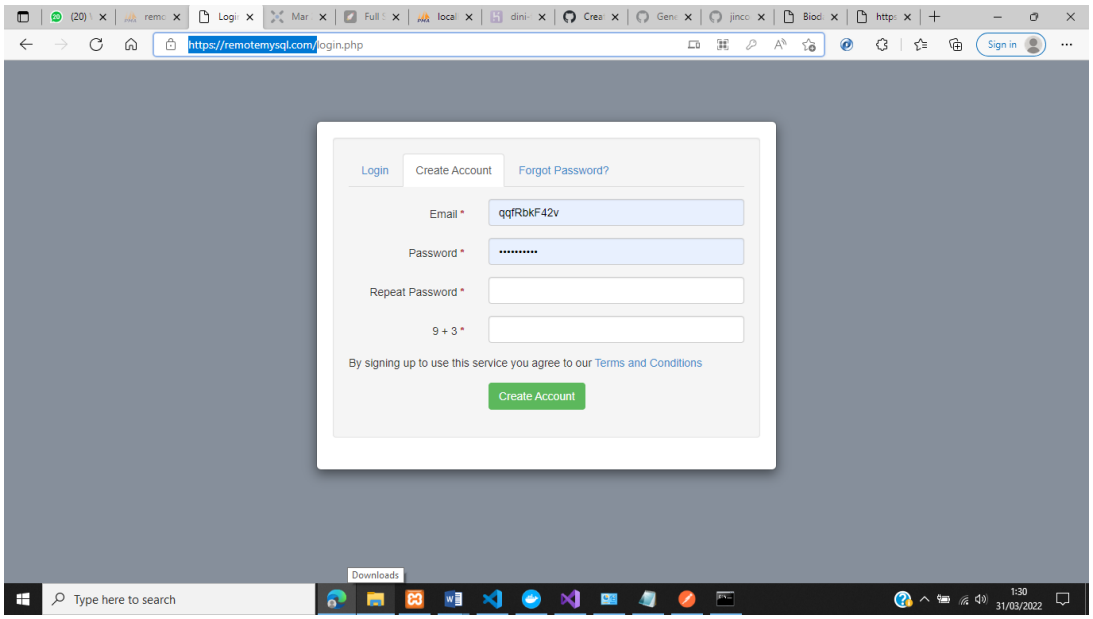


- Pembuatan Landing page
  Pada project Final_Project buat file “wwwroot” yang berisi file index.html sebagai landing page.



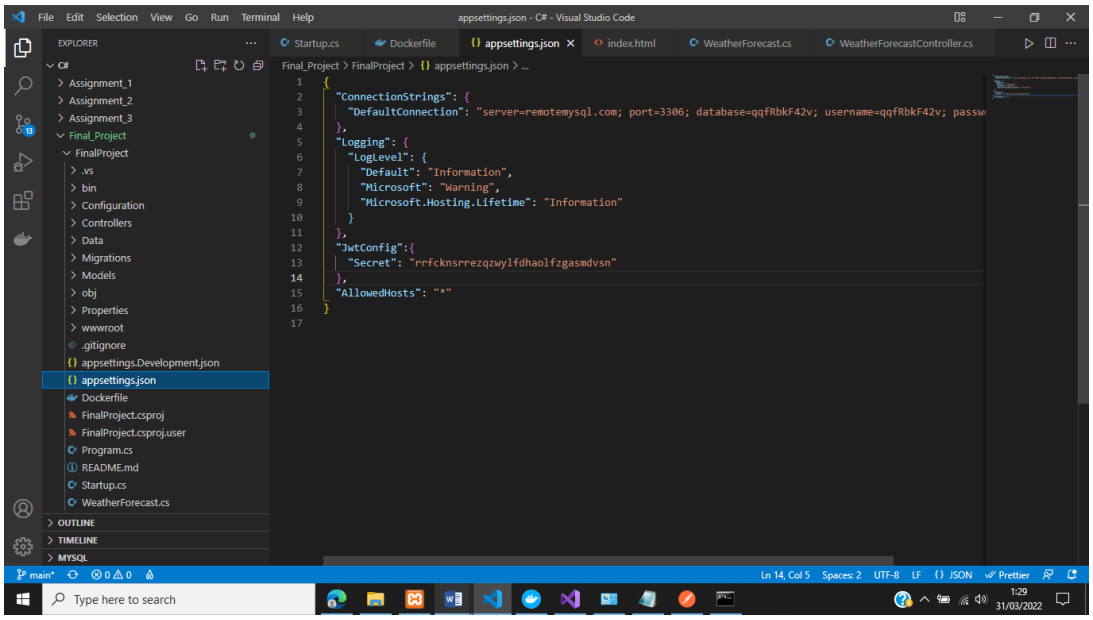
DATABASE

- Buat Akun di https://remotemysql.com/

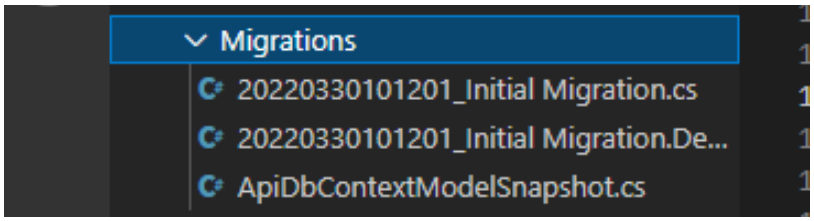


- Setelah masuk dashboard, klik link : https://remotemysql.com/databases.php?action=new untuk membuat database
  ((disini kita akan menerima username, password, dll yang kemudian akan disesuaikan pada file appsettings.json))

- Sesuaikan database didalam appjsettings.json



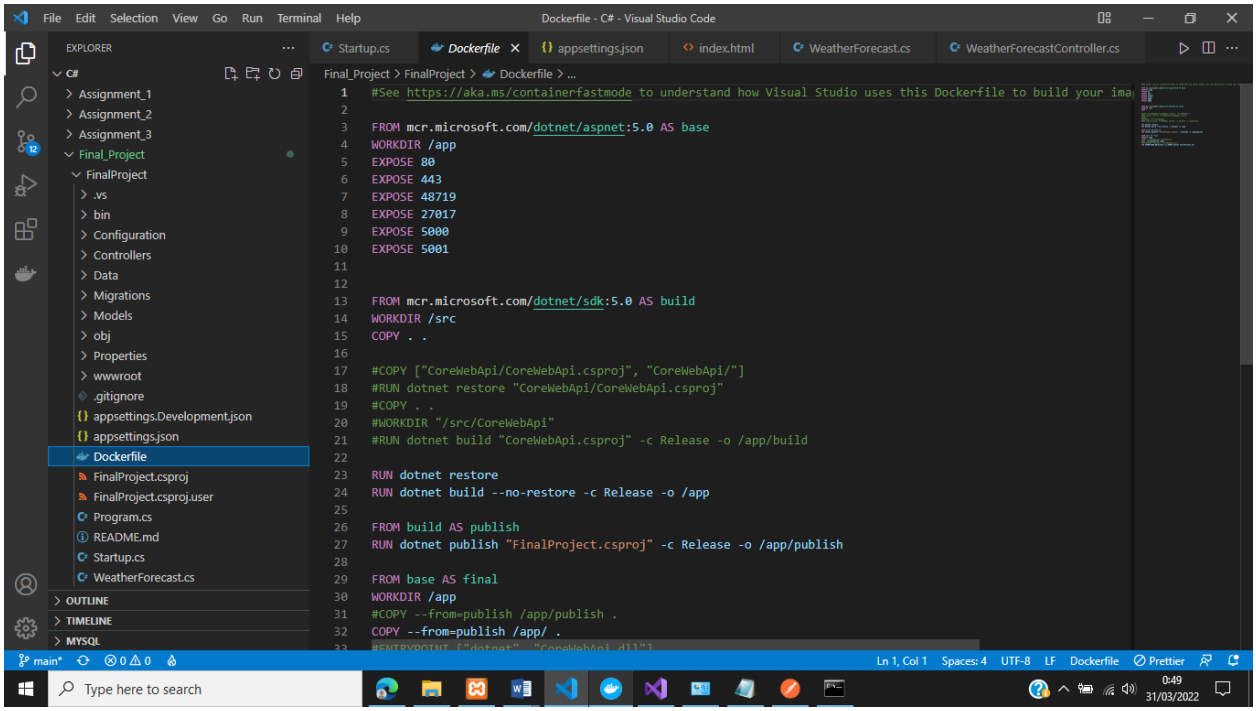
- Lakukan Migrasi ***dotnet ef migrations add "Initial Migrations"***
  Maka akan muncul File Migrations di dalam Folder Final_Project



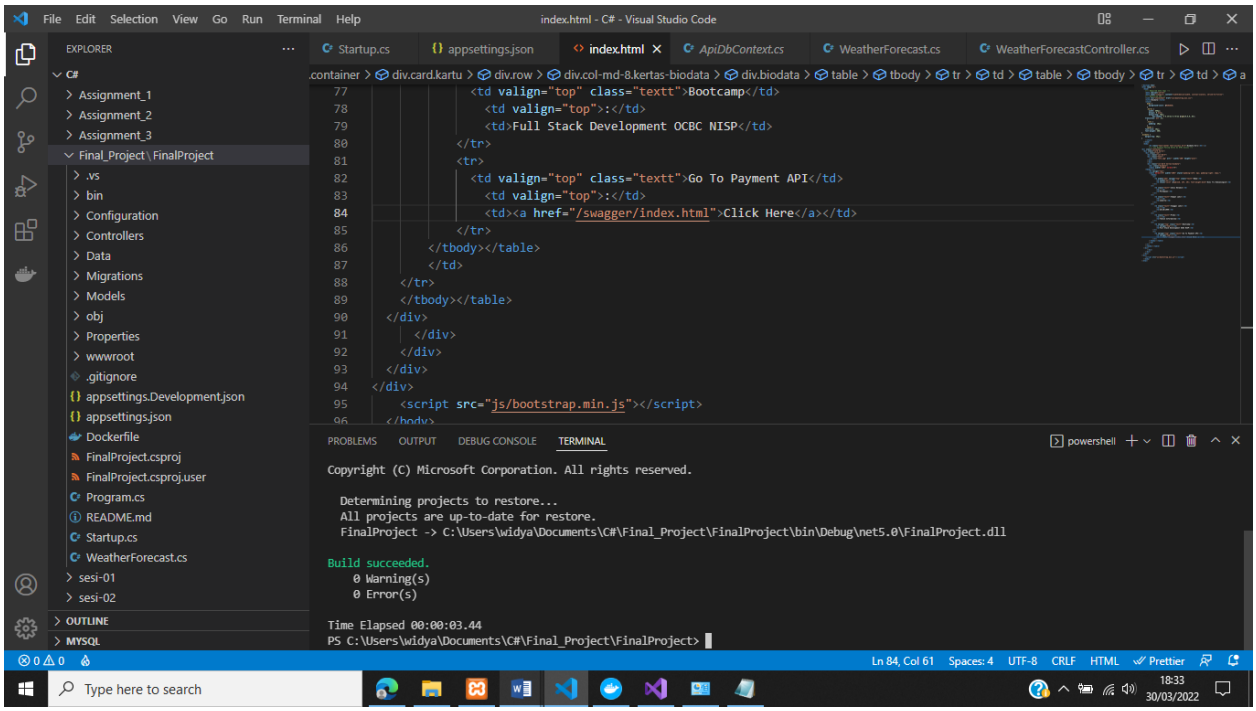
- Lakukan Database Update.

DOCKER

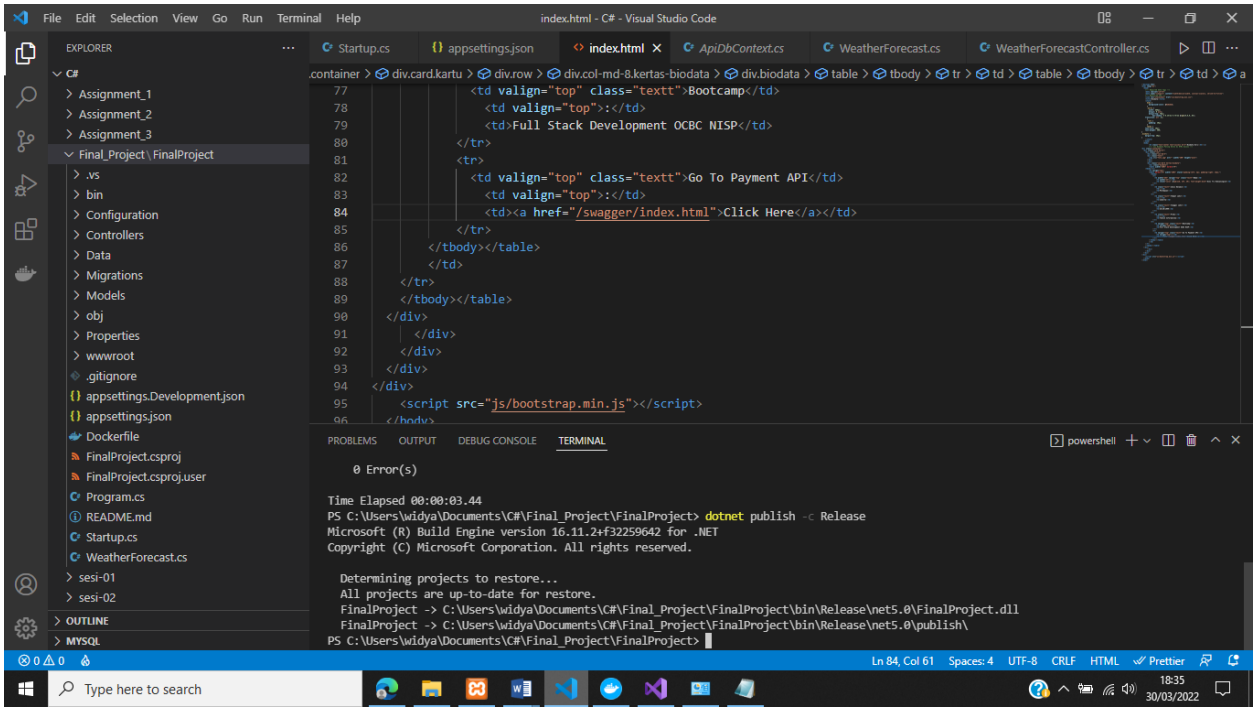
- Buat file Dockerfile di dalam folder Final_Project



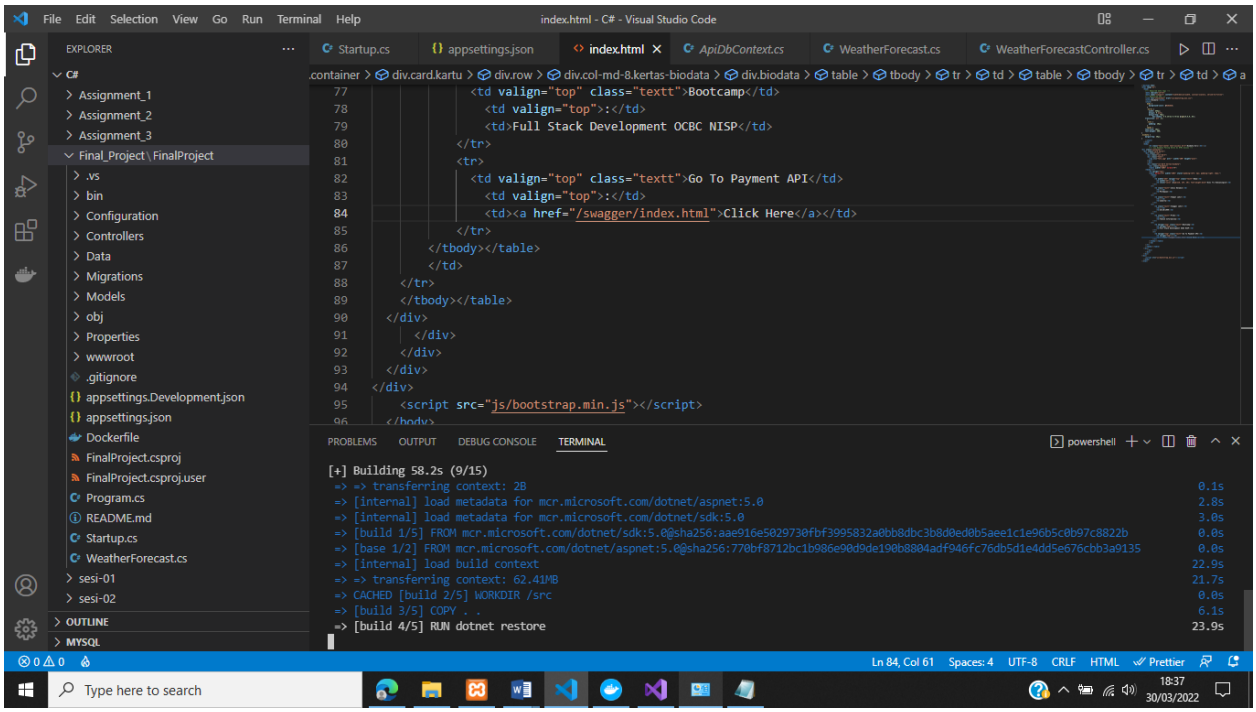
- Lakukan build dengan perintah dotnet build pada terminal

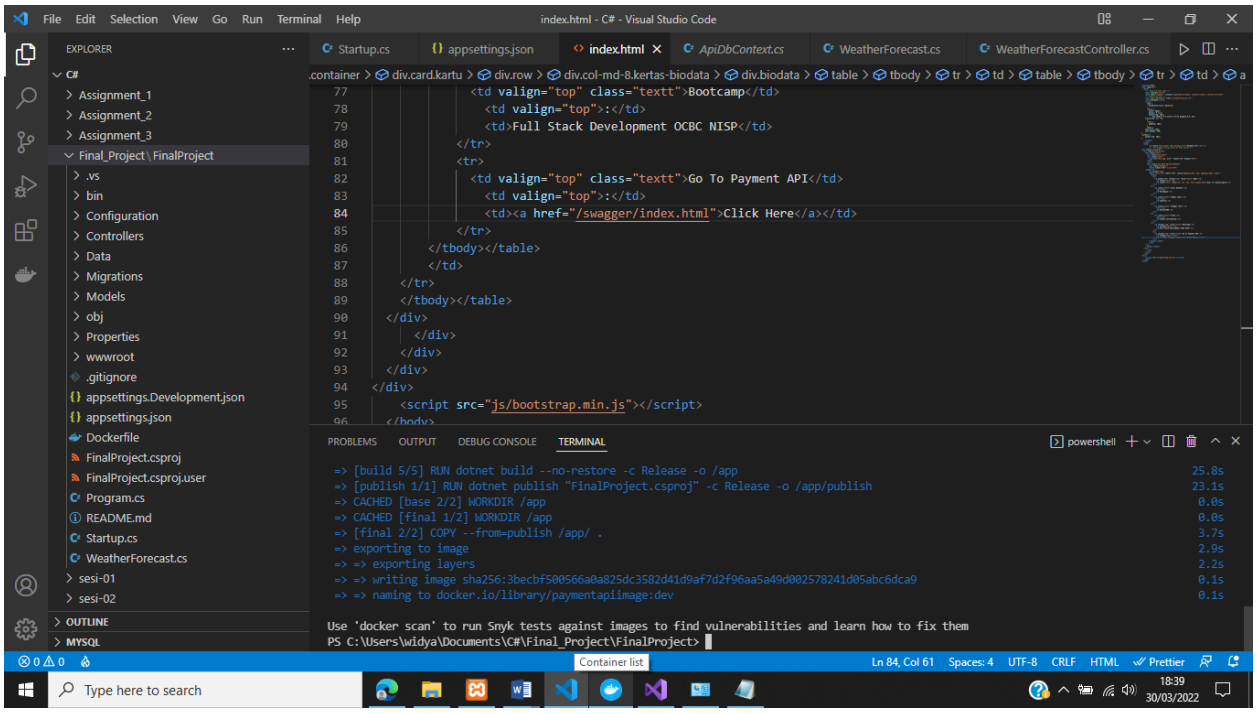


- Lakukan Release dengan perintah ***dotnet publish –c release***

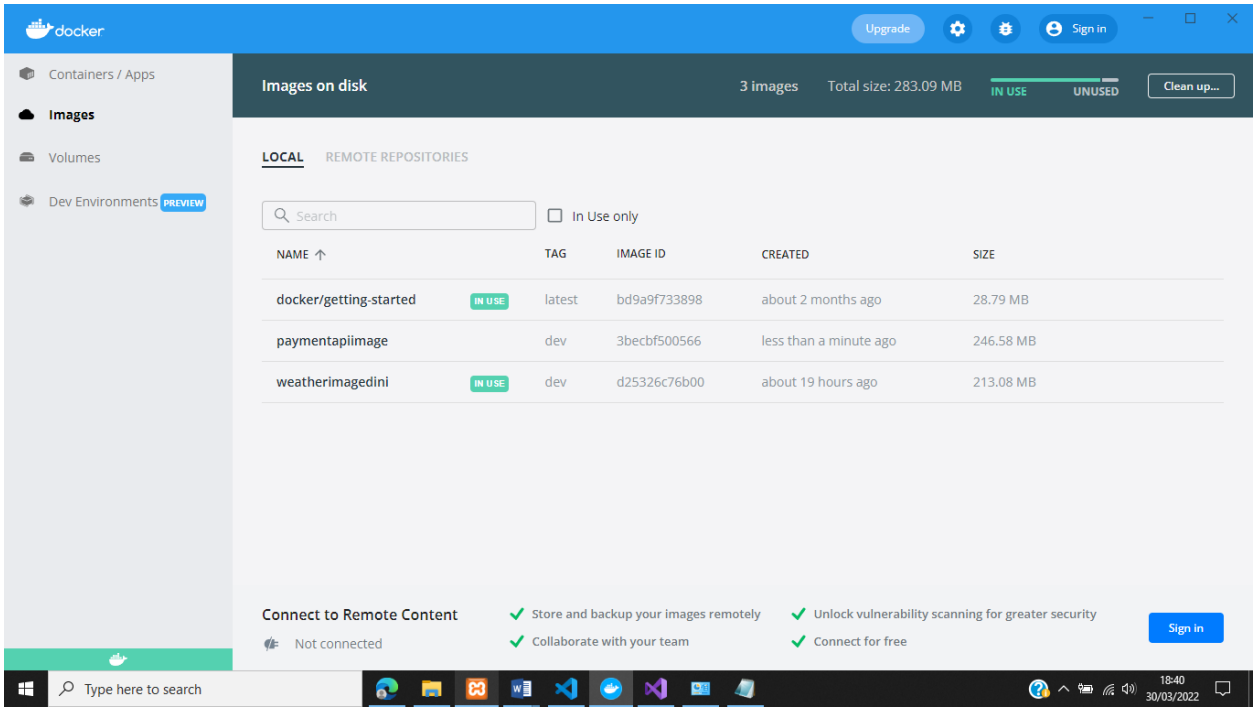


- Pembuatan docker image dengan syntax ***docker build –t paymentapiimage:dev .***

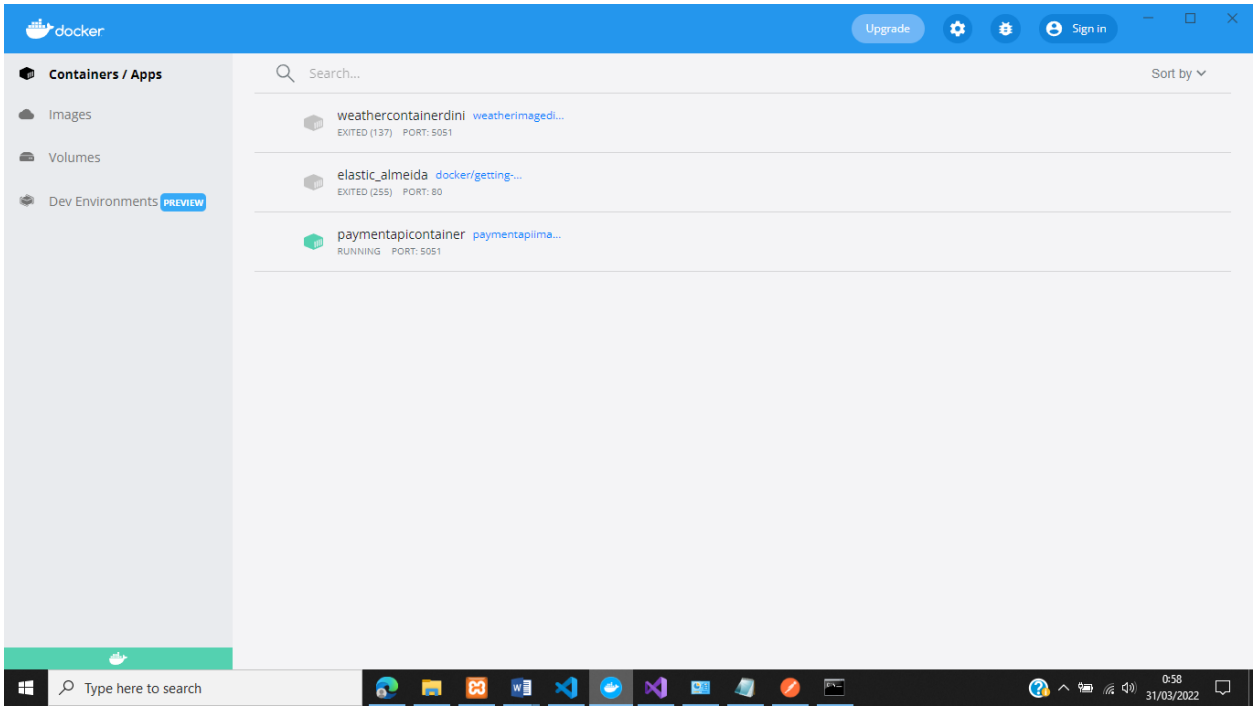




((Apabila berhasil, akan muncul paymentapiimage pada Docker))

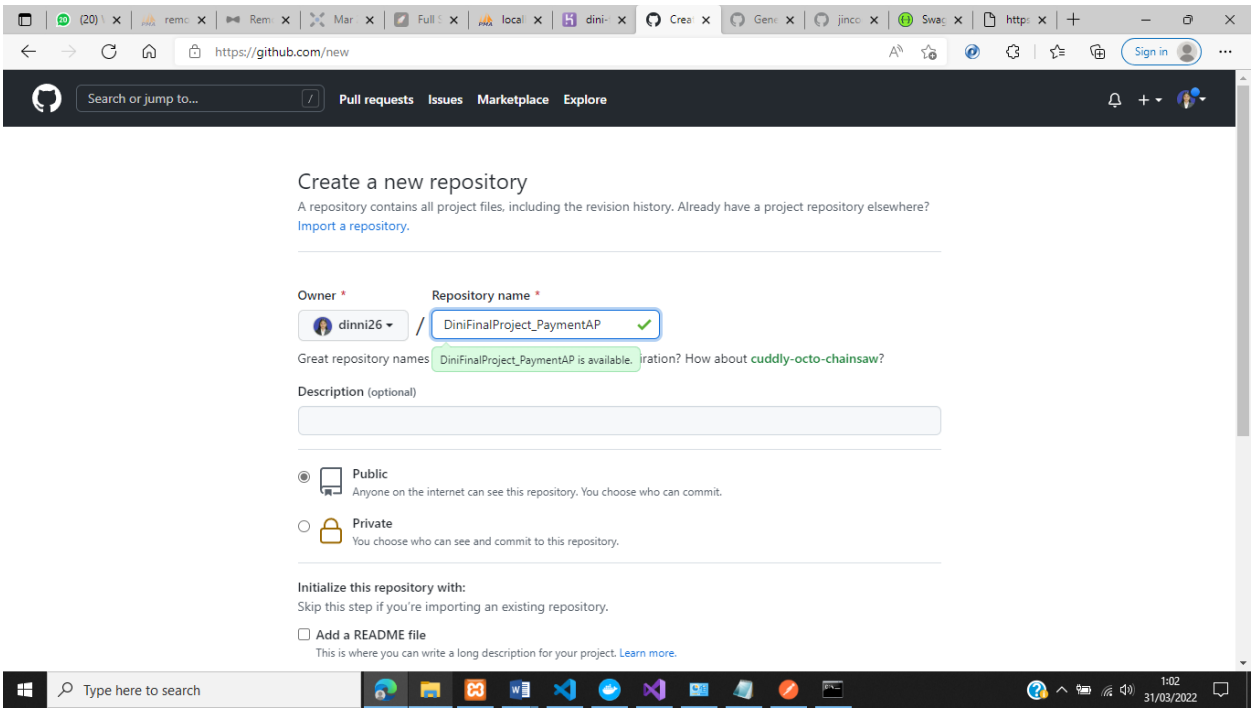


- Membuat container dengan menjalankan perintah ***docker run –d –p 5051:80 –name paymentapicontainer paymentapiimage:dev*** (aplikasi akan berjalan pada port 5051)

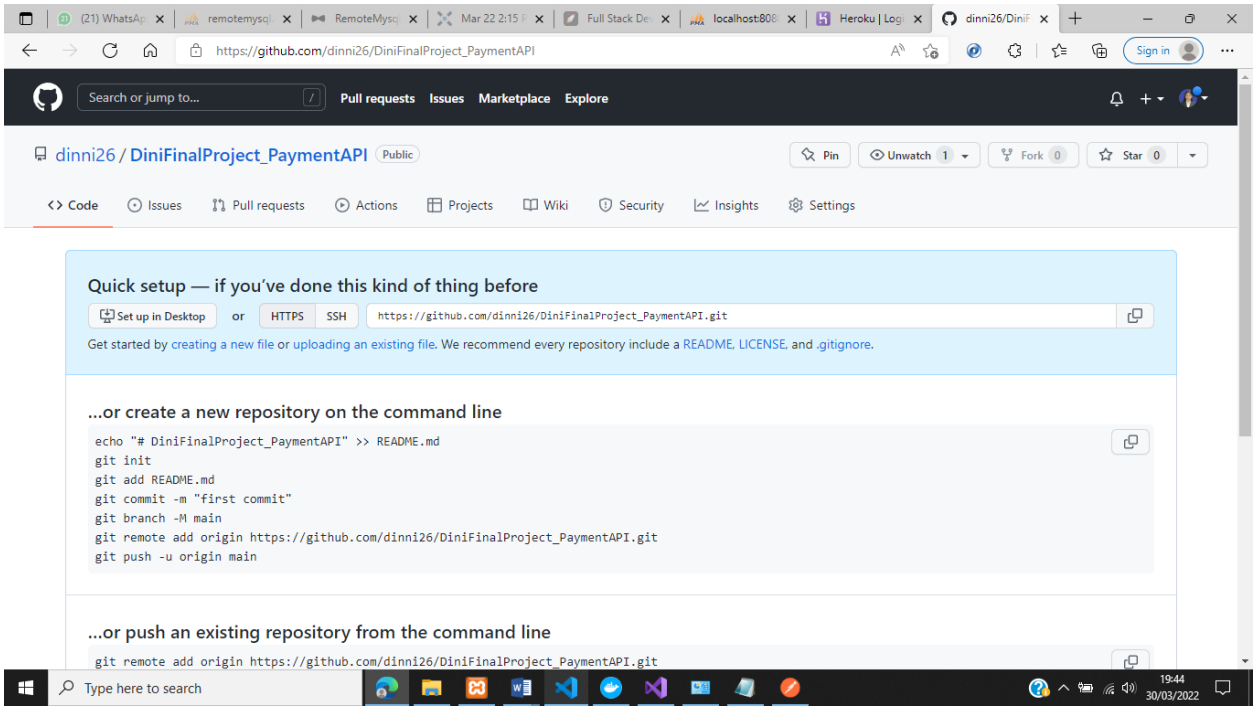


PEMBUATAN REPO BARU DI GITHUB

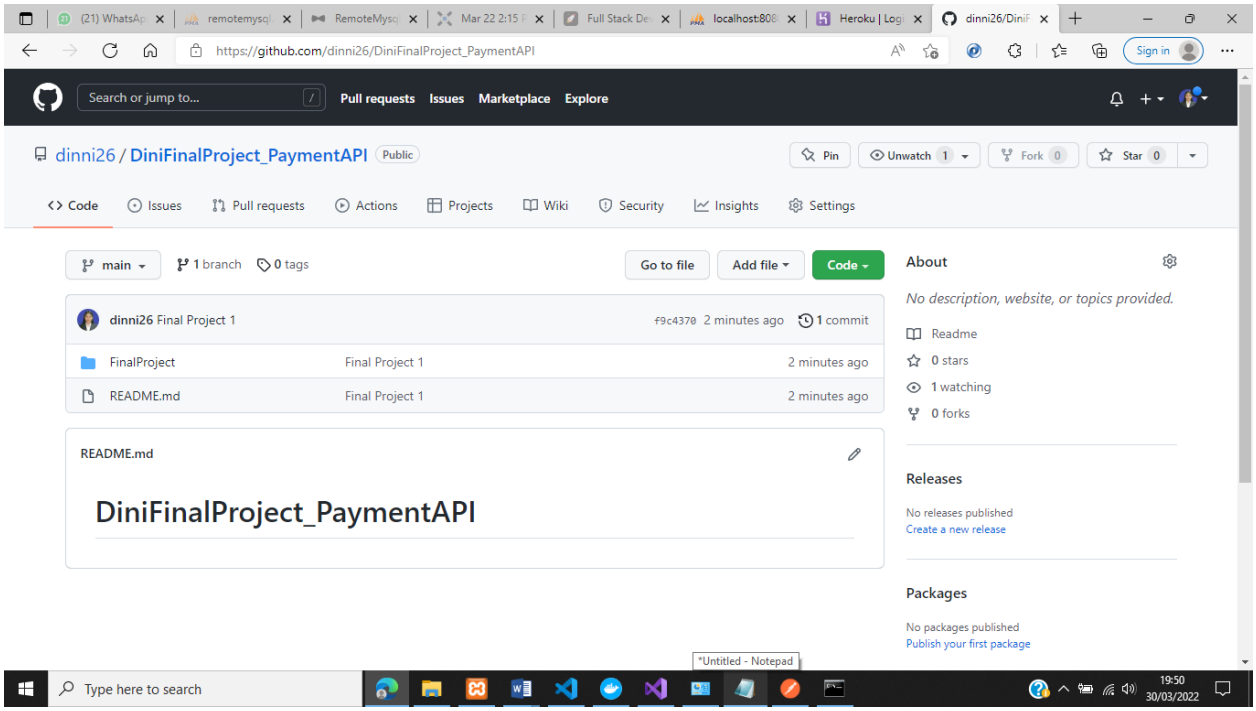
- Create new repo lalu beri nama repo yang baru dibuat



- Setelah terbuat, akan muncul code yang akan kita jalankan di cmd

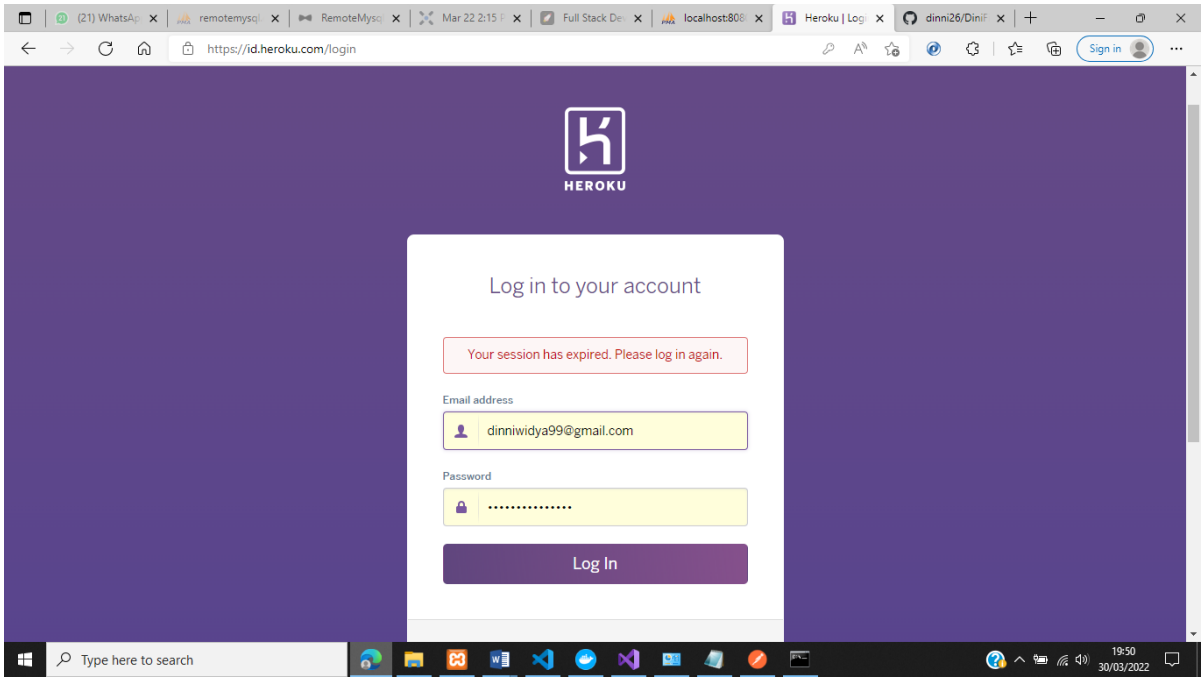


- Setelah dijalankan melalui cmd, makan repository baru akan tampil folder FinalProject yang sudah dibuat

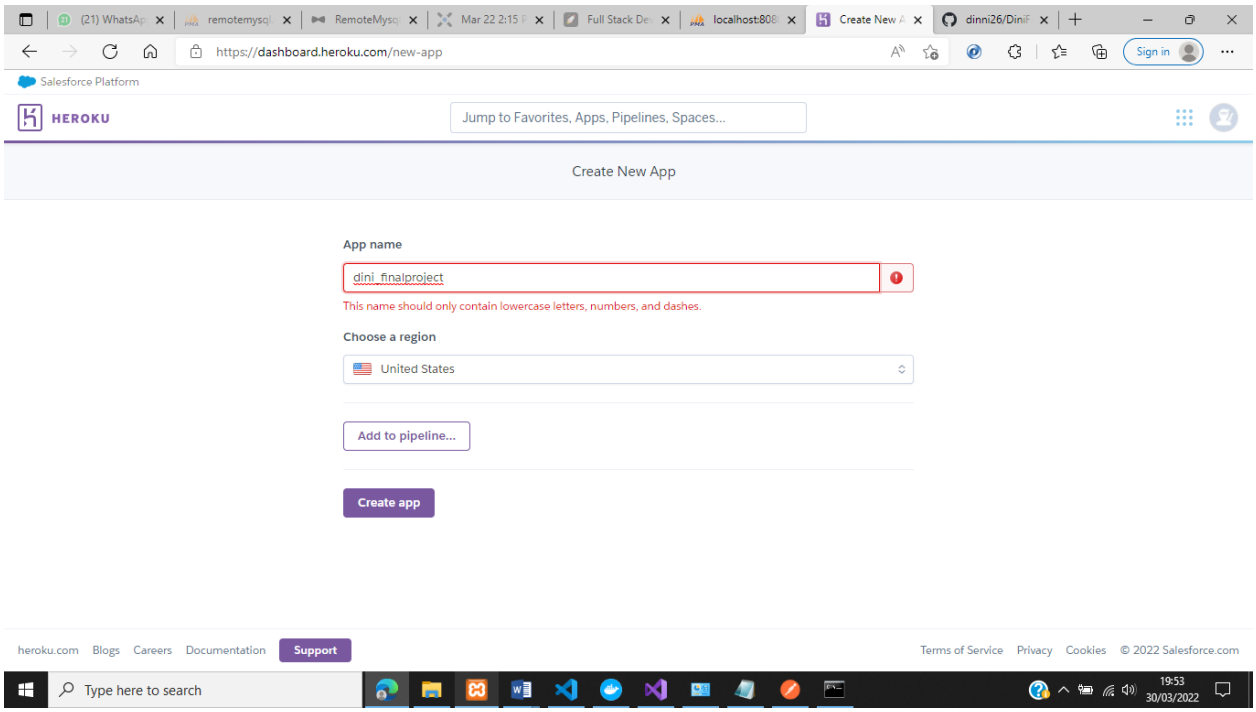


HEROKU

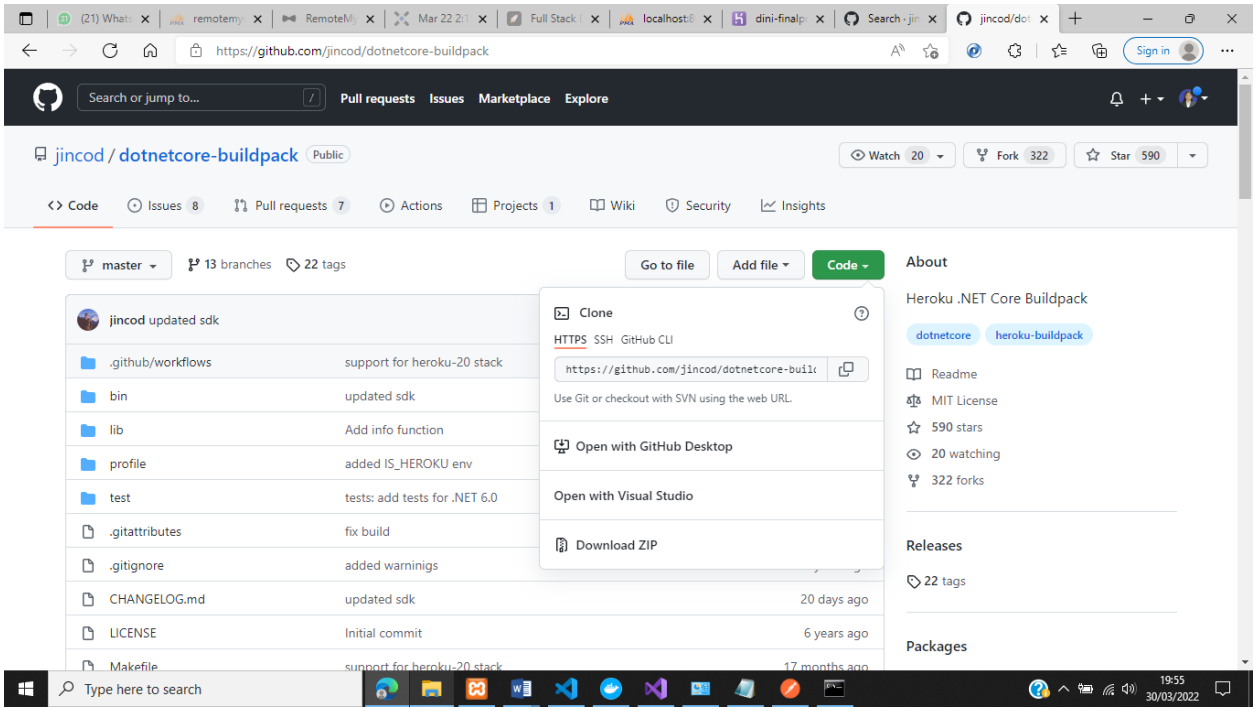
- Login Akun Heroku



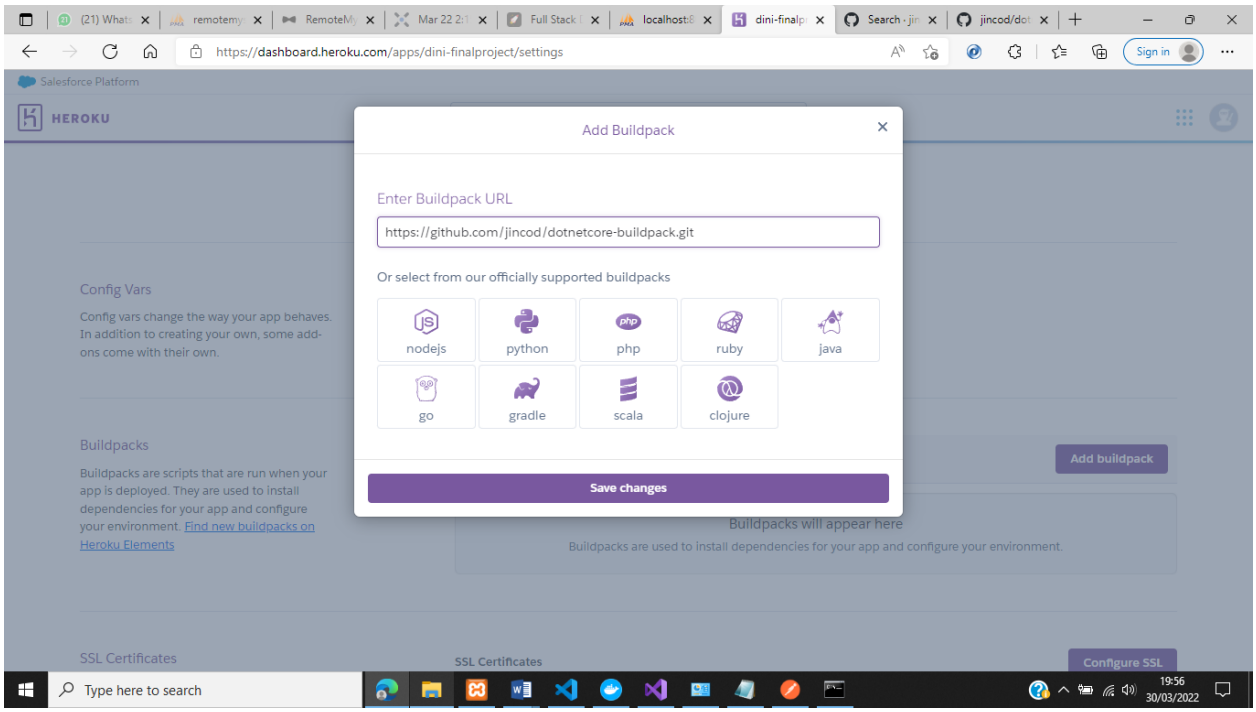
- Create New App lalu masukan nama app yang akan dibuat lalu create app.



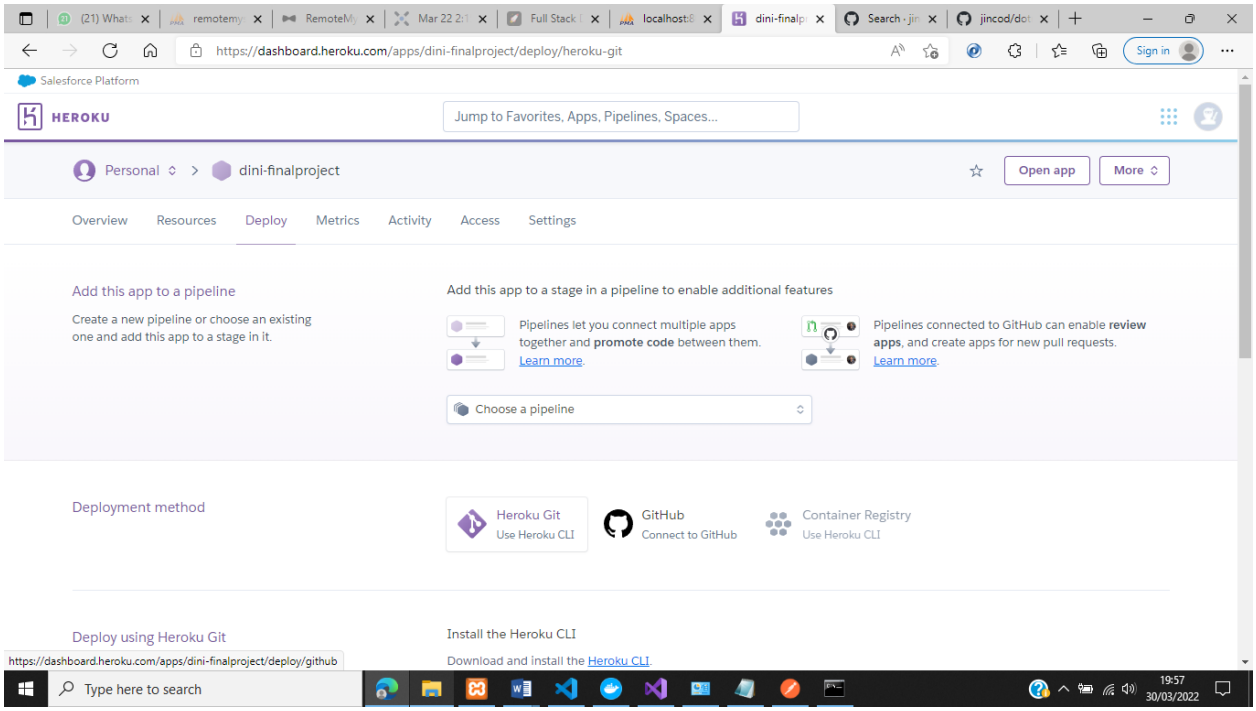
- Setelah Create App > Settings > Add Buildpack > Copy Link Jincod dari github



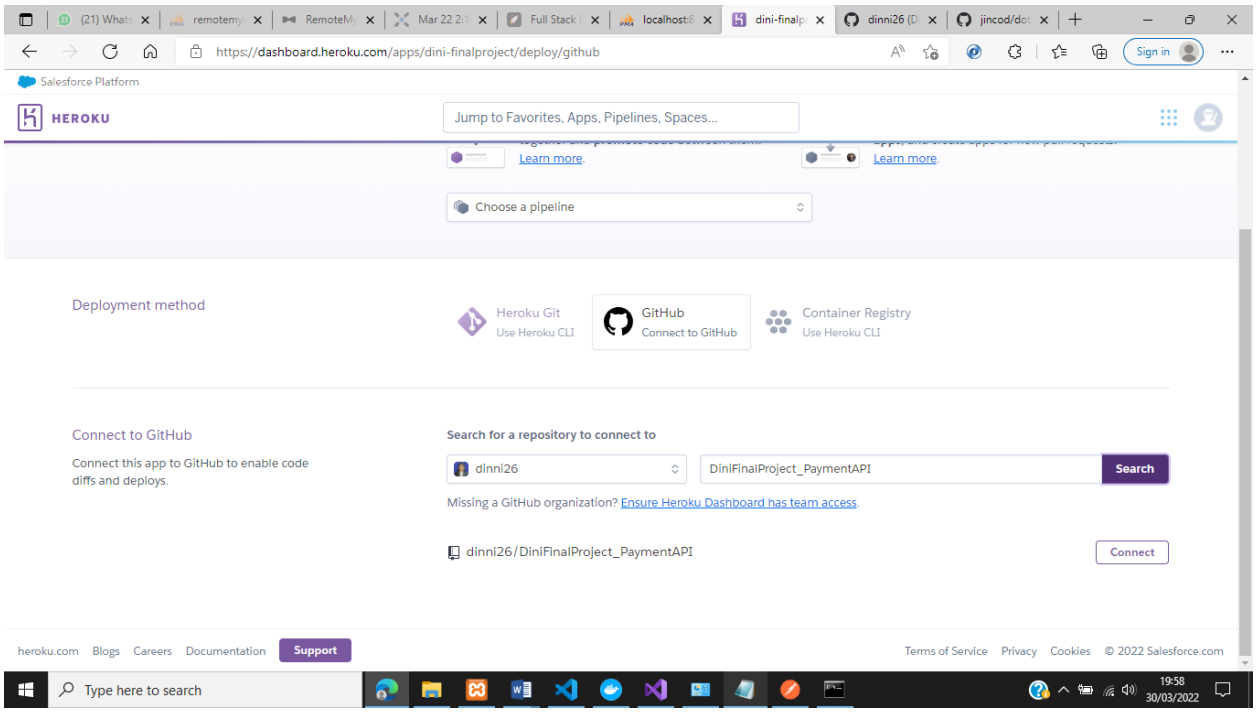
- Paste link Jincod Kedalam Buildpack URL



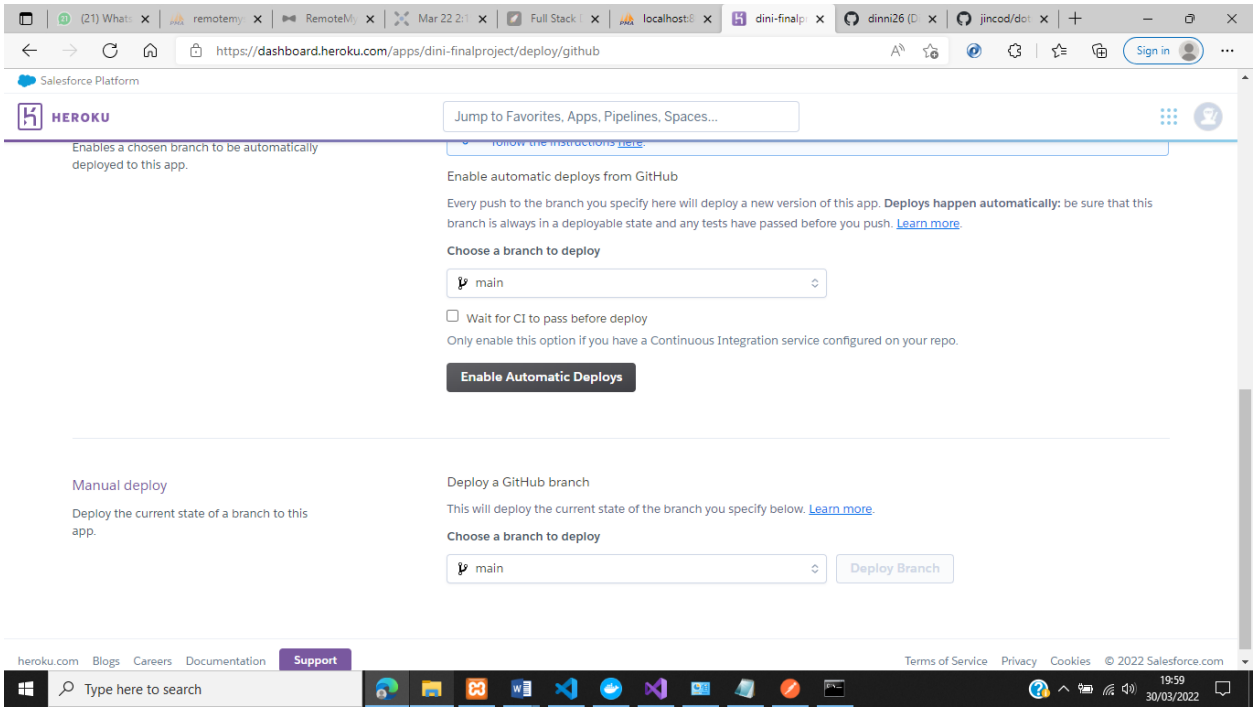
- Lalu Ke Deploy > Connect to Github



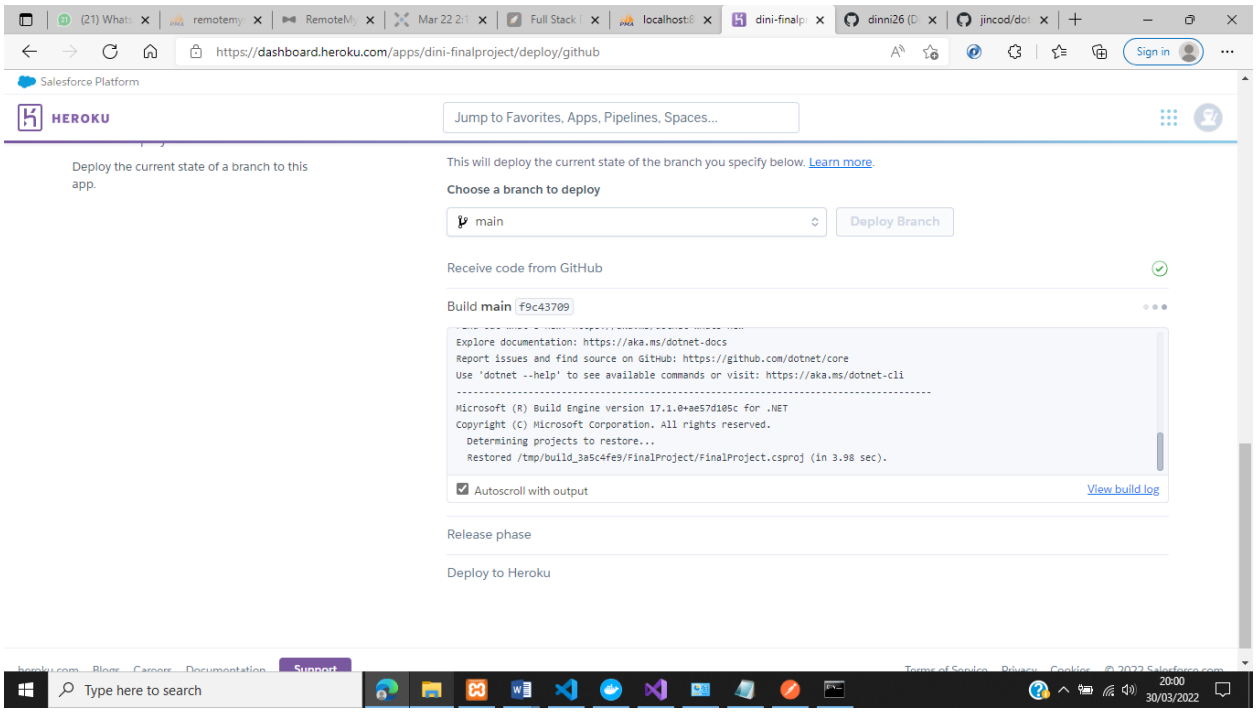
- Masukan Repo Github lalu Connect



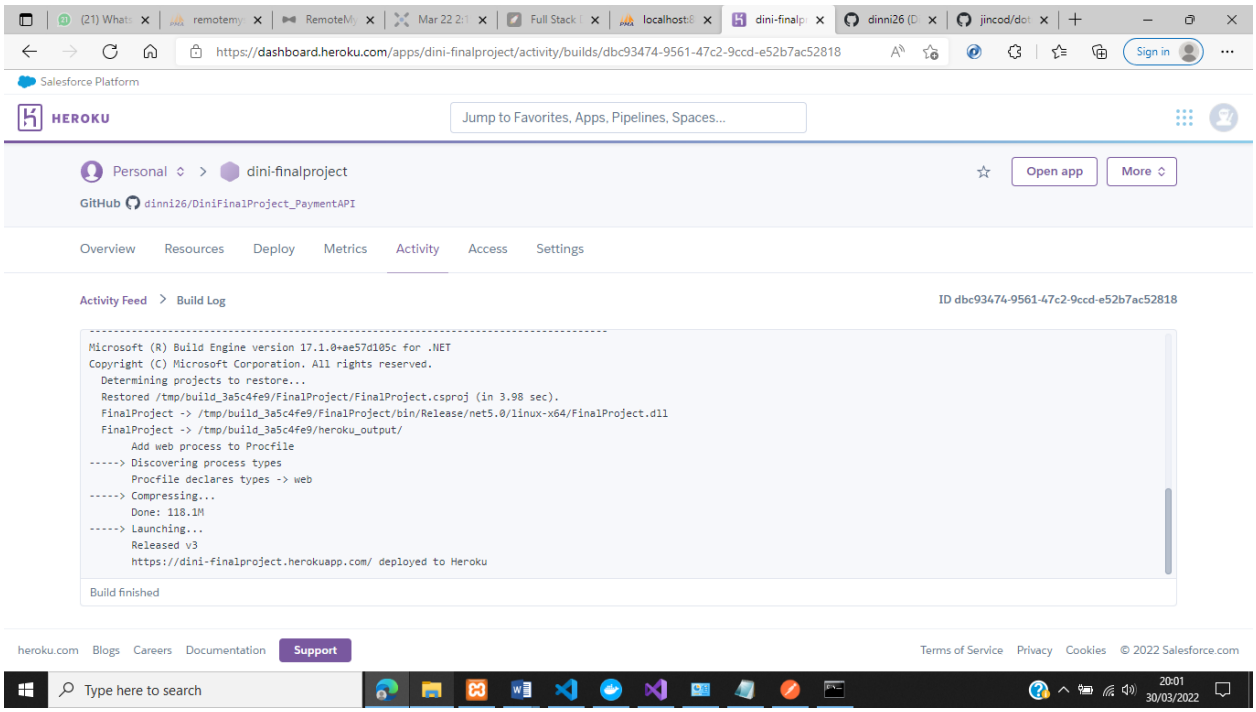
- Setelah itu scroll ke bawah lalu klik “Deploy Branch”



- Tunggu sampai proses selesai

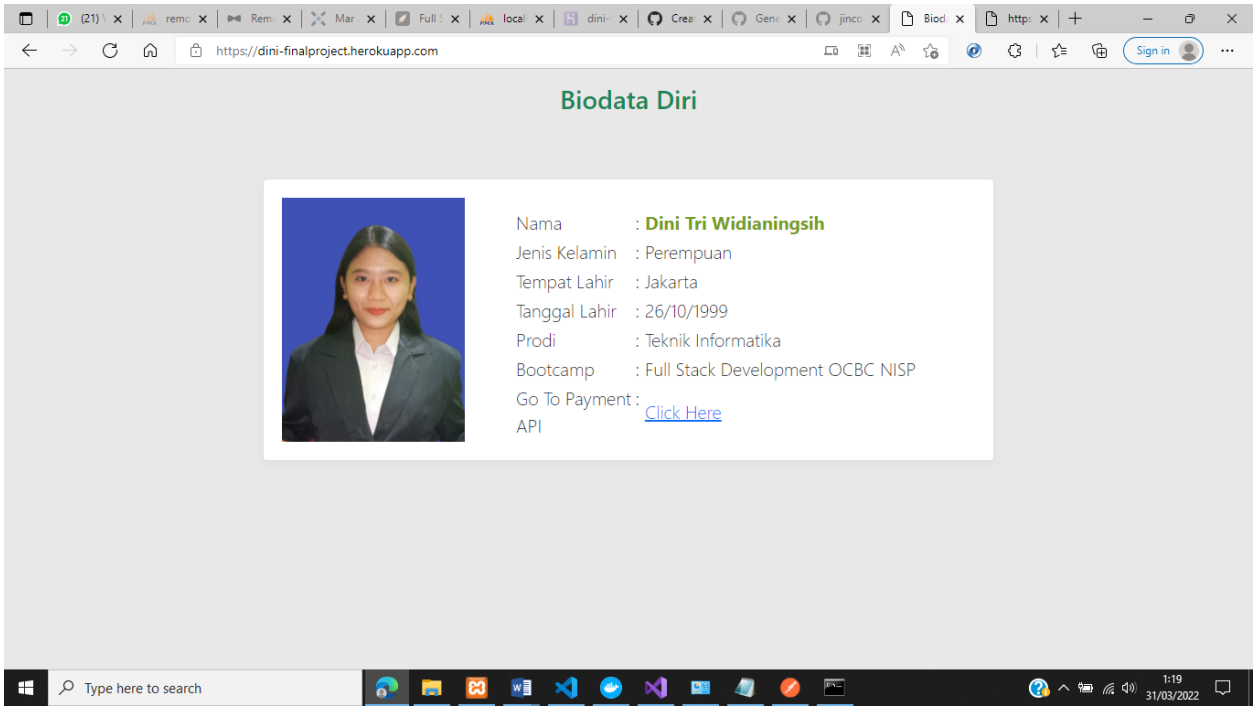


- Setelah selesai maka akan ada link heroku yang nantinya akan kita jalankan untuk penggunaan ProjectAkhir



- Salin Link > Browser untuk menjalankannya.
  Link : [Biodata (dini-finalproject.herokuapp.com)](https://dini-finalproject.herokuapp.com)

Tampilan Awal



((Lanjut ke panduan penggunaan aplikasi))

Nama : **Dini Tri Widianingsih**

No. Peserta : **FSDO003ONL007**

Link Github : dinni26/DiniFinalProject_PaymentAPI (github.com)

Link Heroku : Biodata (dini-finalproject.herokuapp.com)

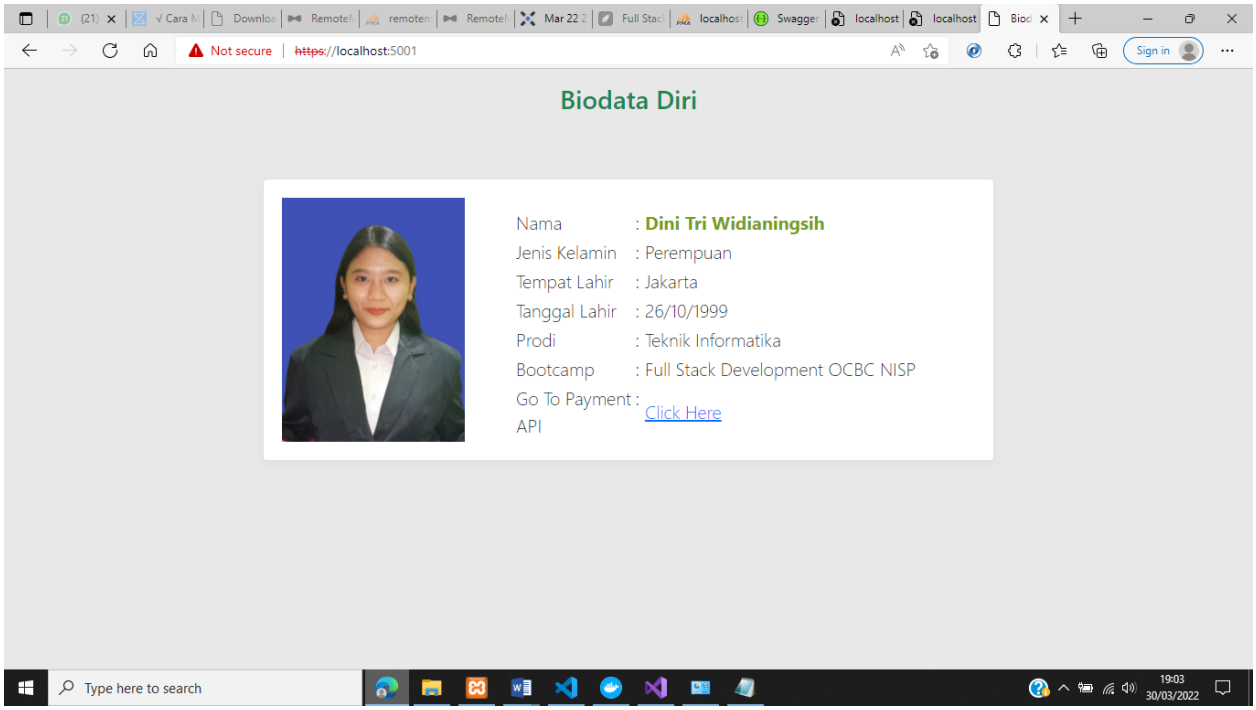
PANDUAN PENGGUNAAN FINAL PROJECT

Uji Coba Mengunakan Swagger dan Postman

Jalankan Link Berikut : https://dini-finalproject.herokuapp.com/

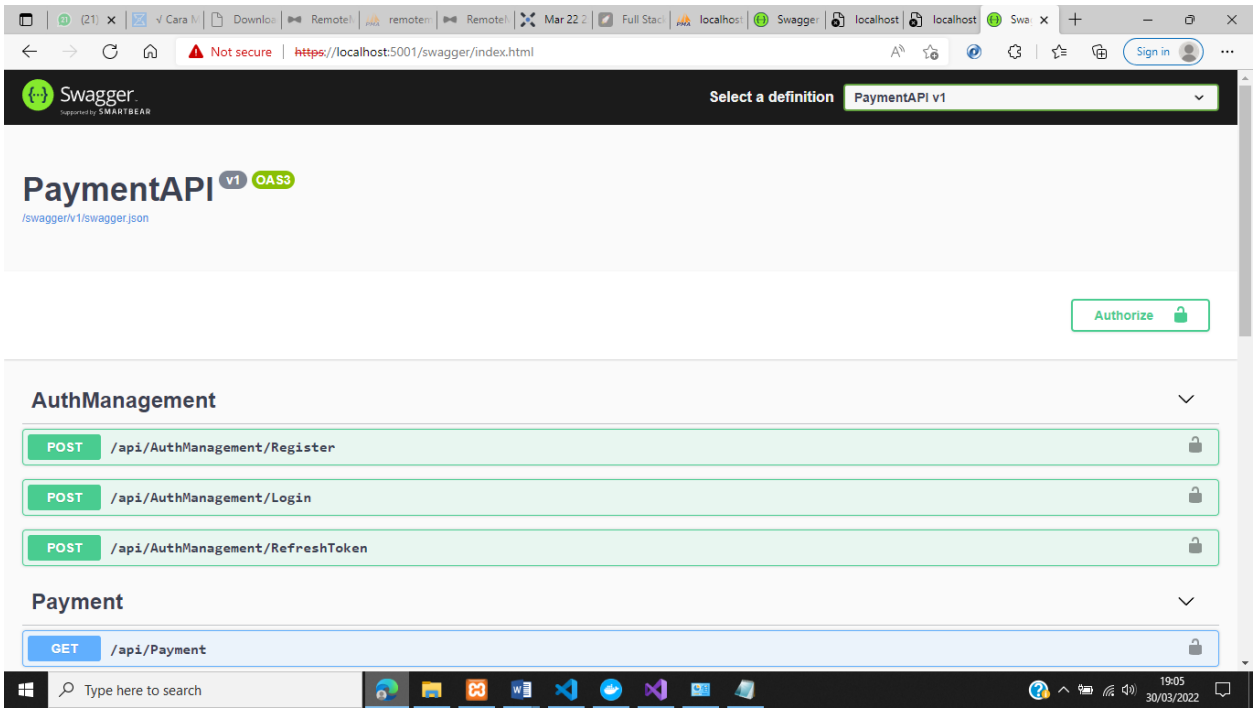
SWAGGER

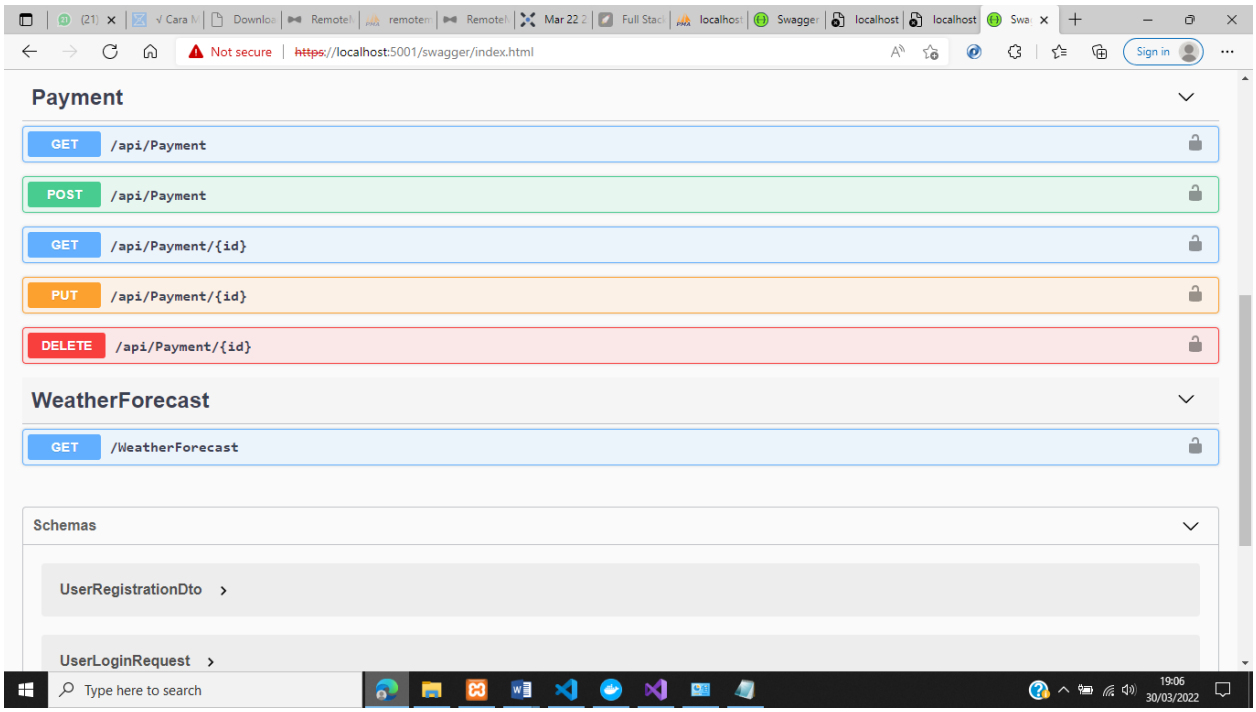
1. Tampilan Awal Saat di Jalankan



((Click Here Untuk Beralih Ke Laman Swagger))

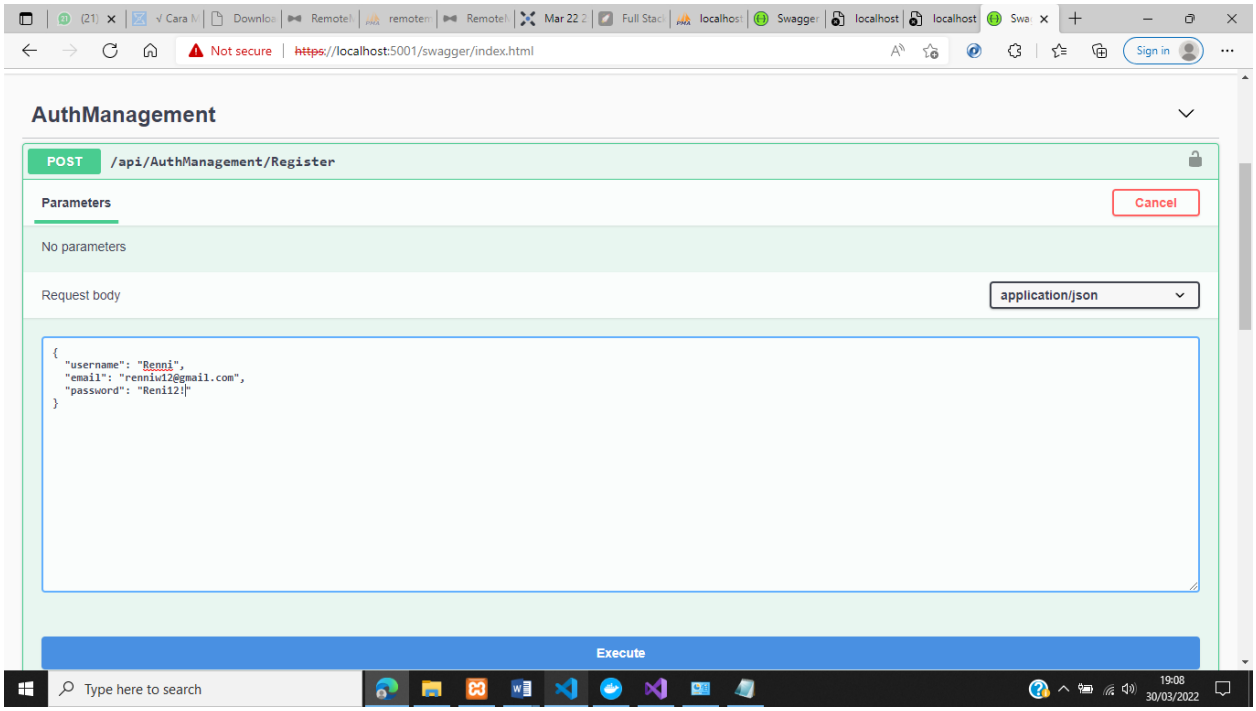
Halaman Swagger



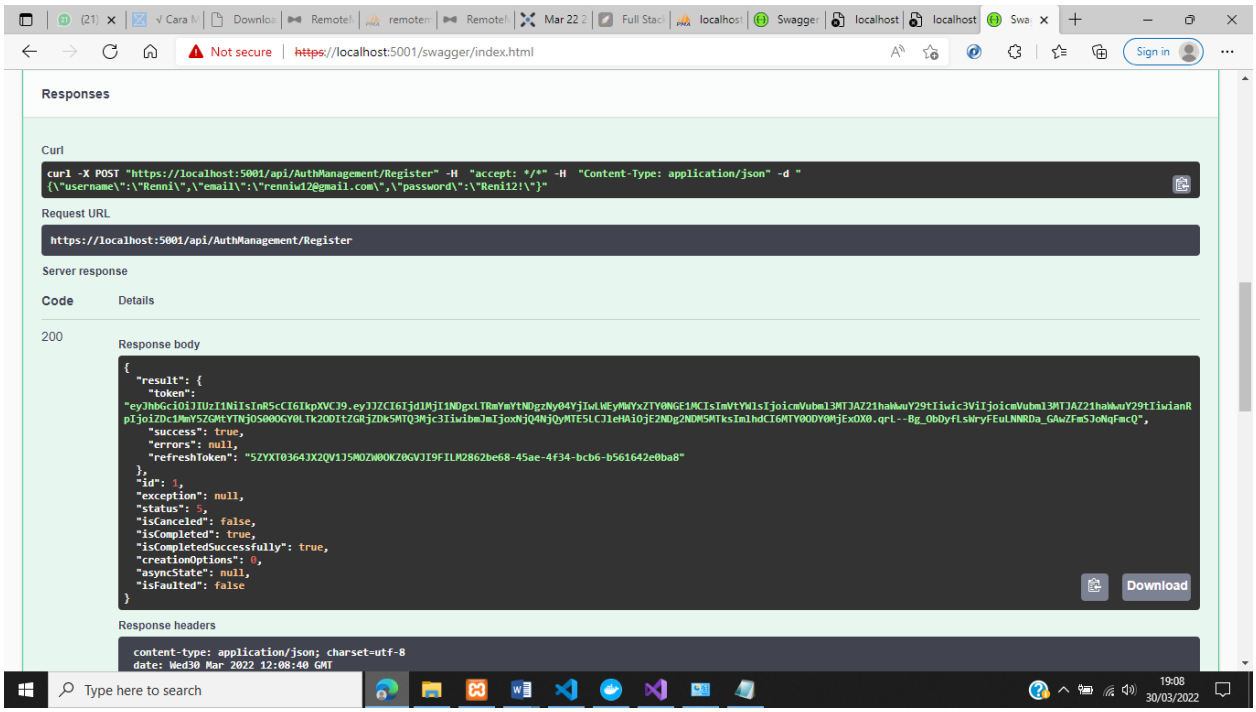


Register

- Lakukan Register dengan menekan tombol “Try It Out”.
- Register dengan memasukan username, email, dan password baru.

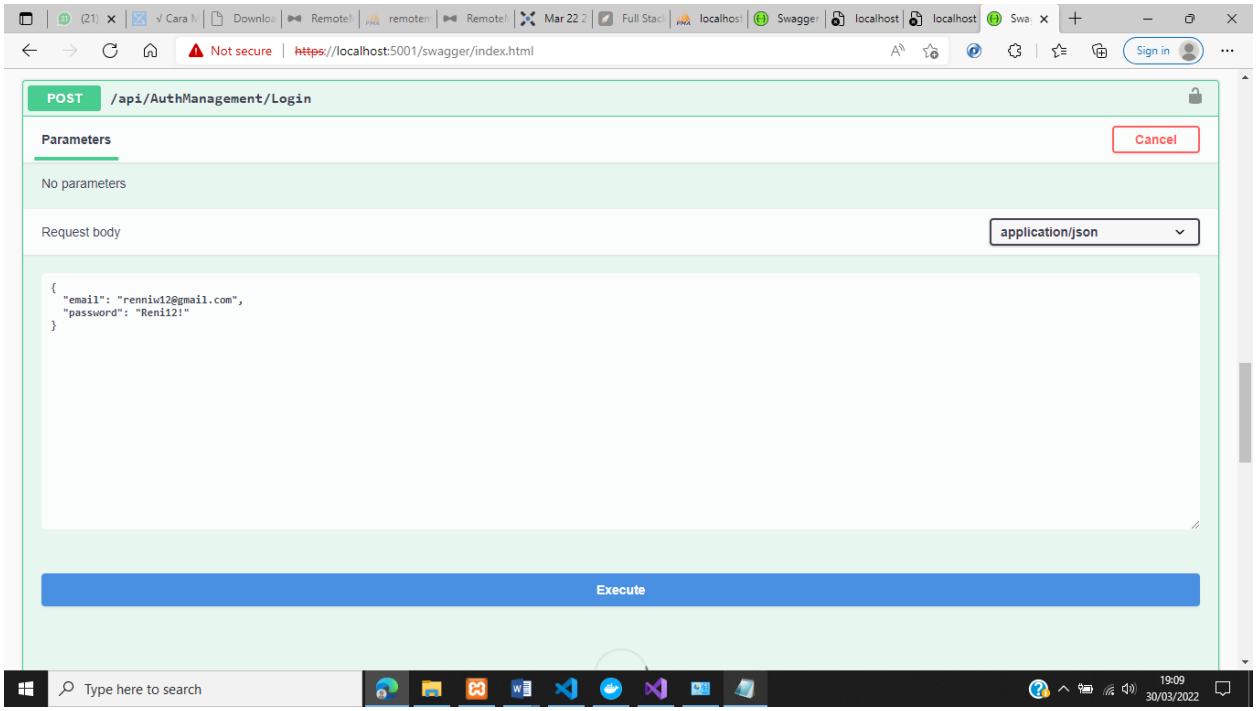


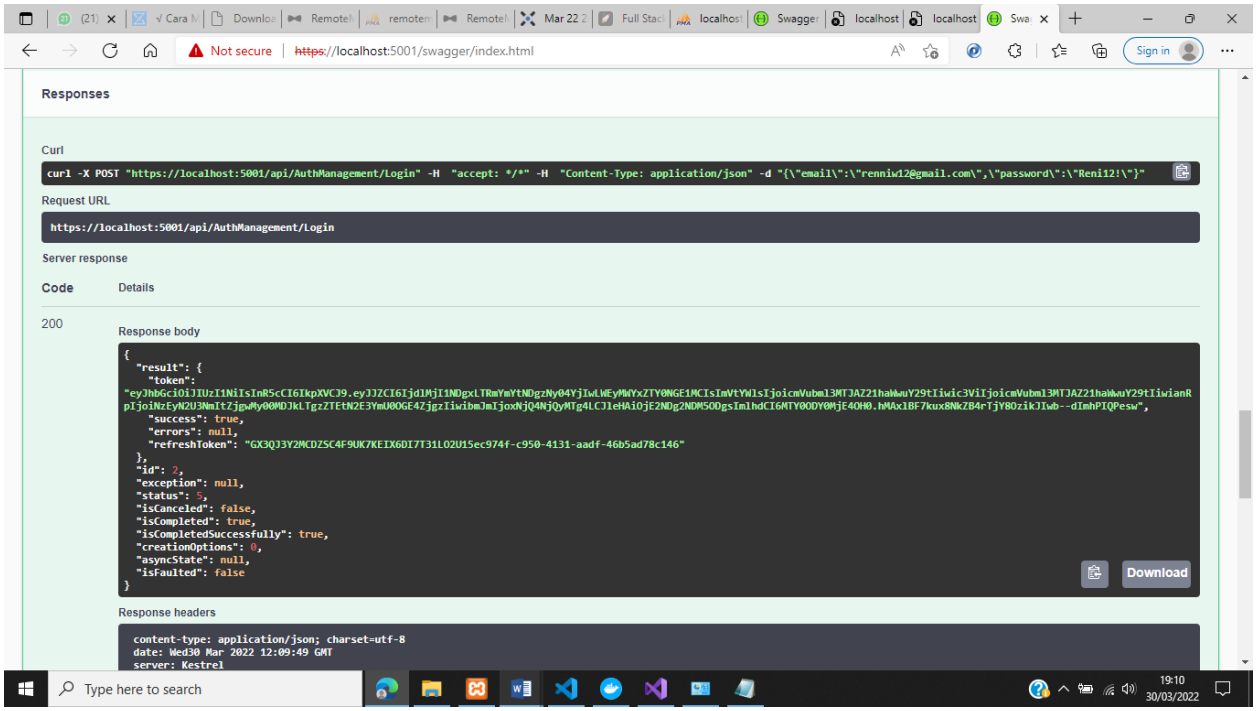
Execute, lalu respon akan menampilkan token.



Login

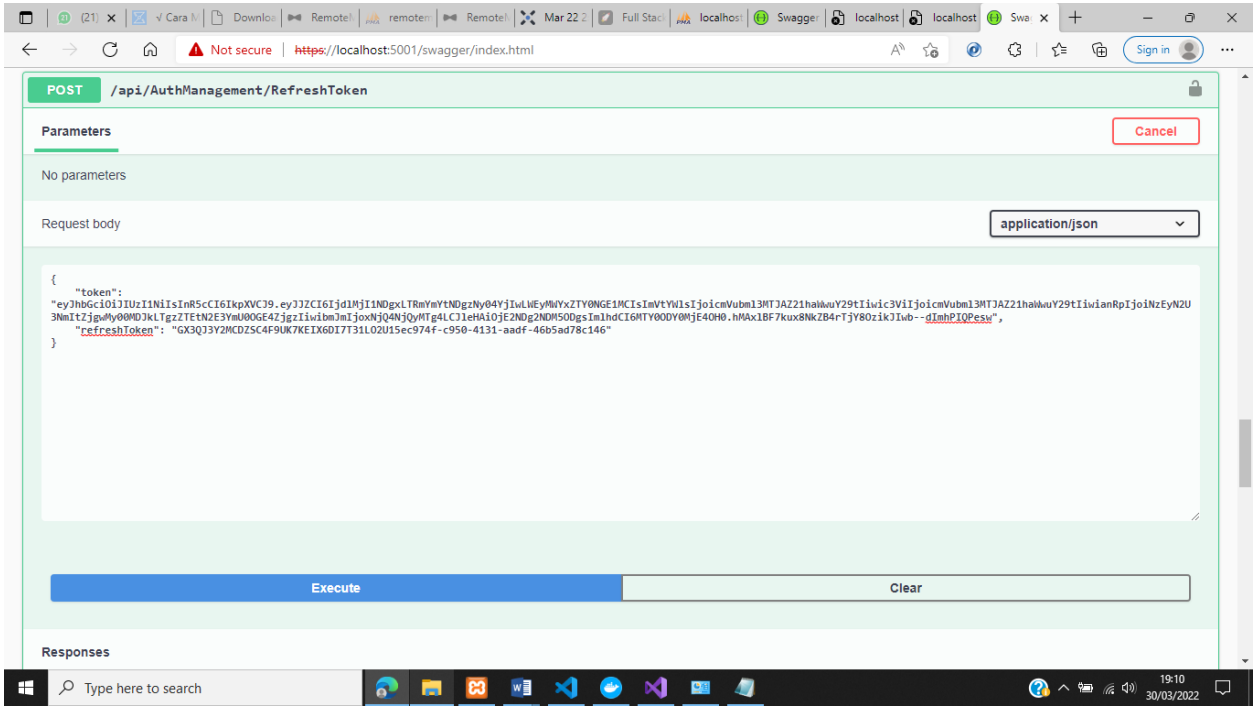
- Login menggunakan akun yang sebelumnya sudah didaftarkan dengan memasukan email dan password. Perlu di ingat, token hanya bisa digunakan dalam waktu 30 menit saja.

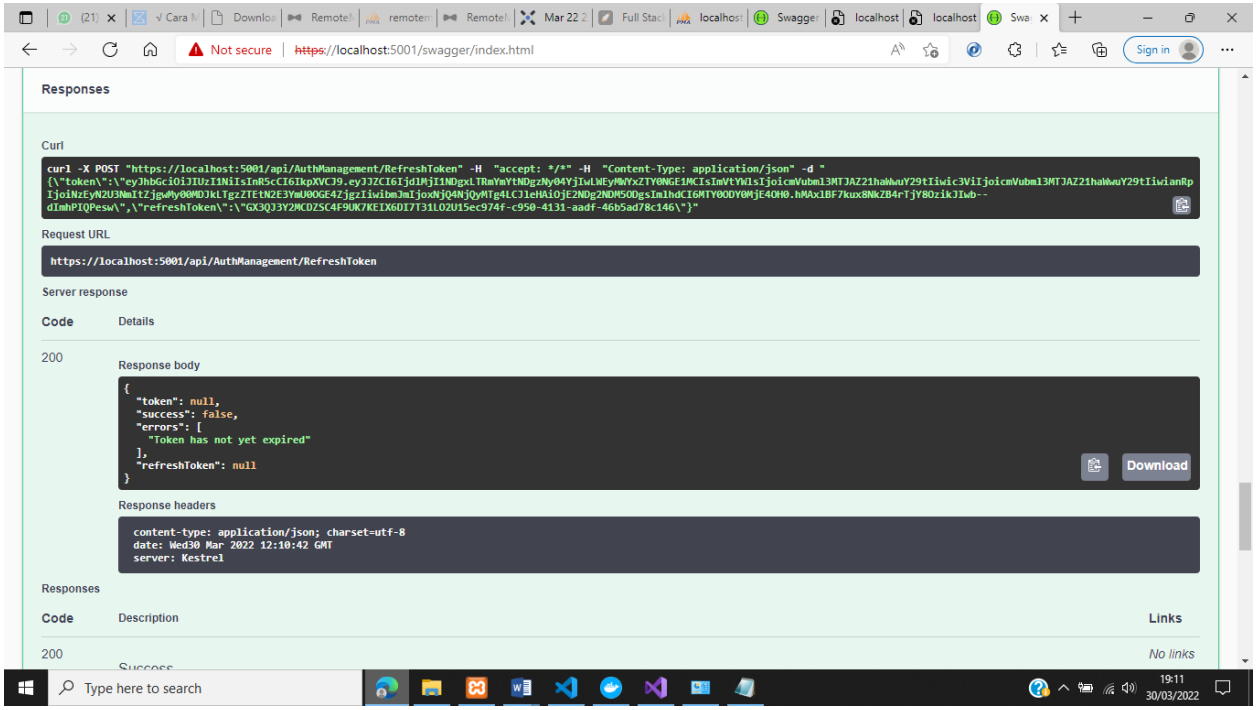




Refresh Token

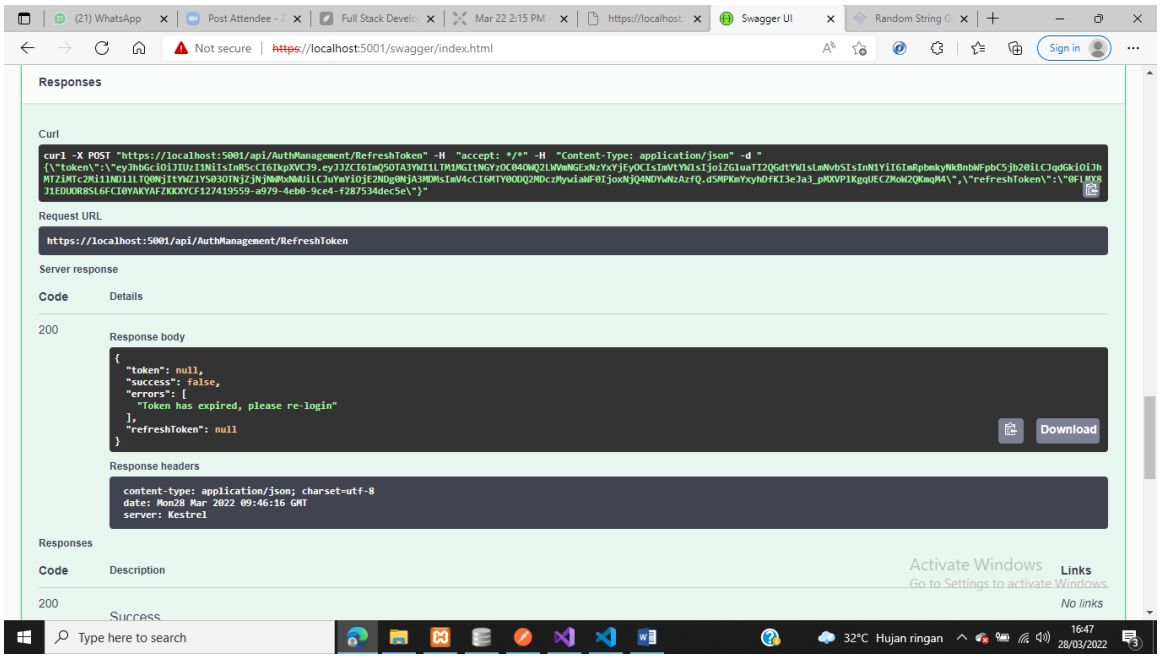
- Percobaan Refresh Token sebelum 30 Menit





((Muncul pesan bahwa token masih bisa digunakan))

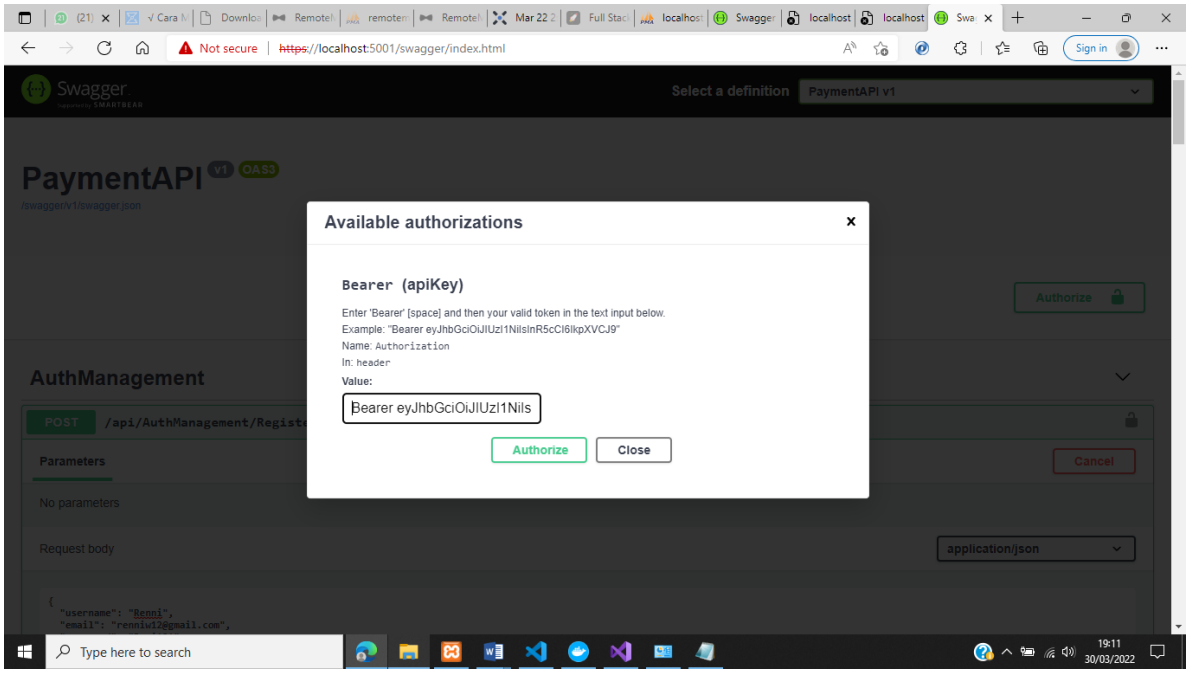
- Percobaan melakukan Refresh Token Setelah 30 Menit

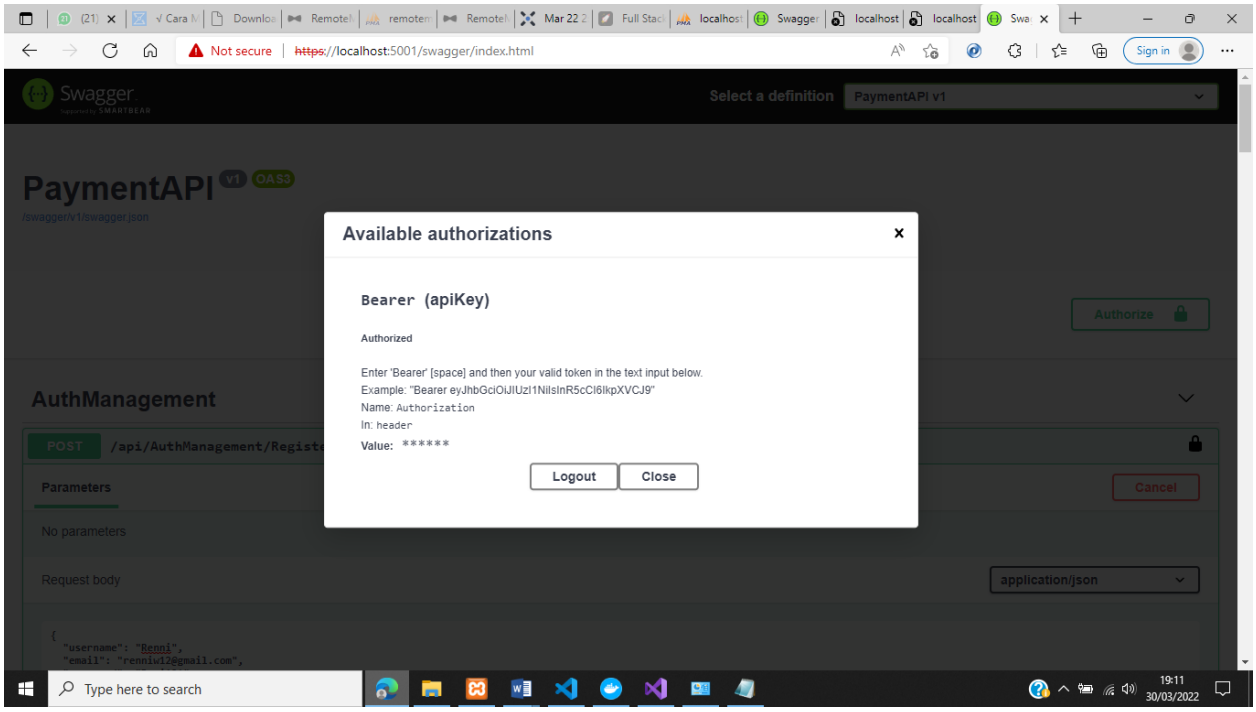


((terdapat pesan bahwa token sudah expired dan harus melakukan login kembali))

AUTHORIZE

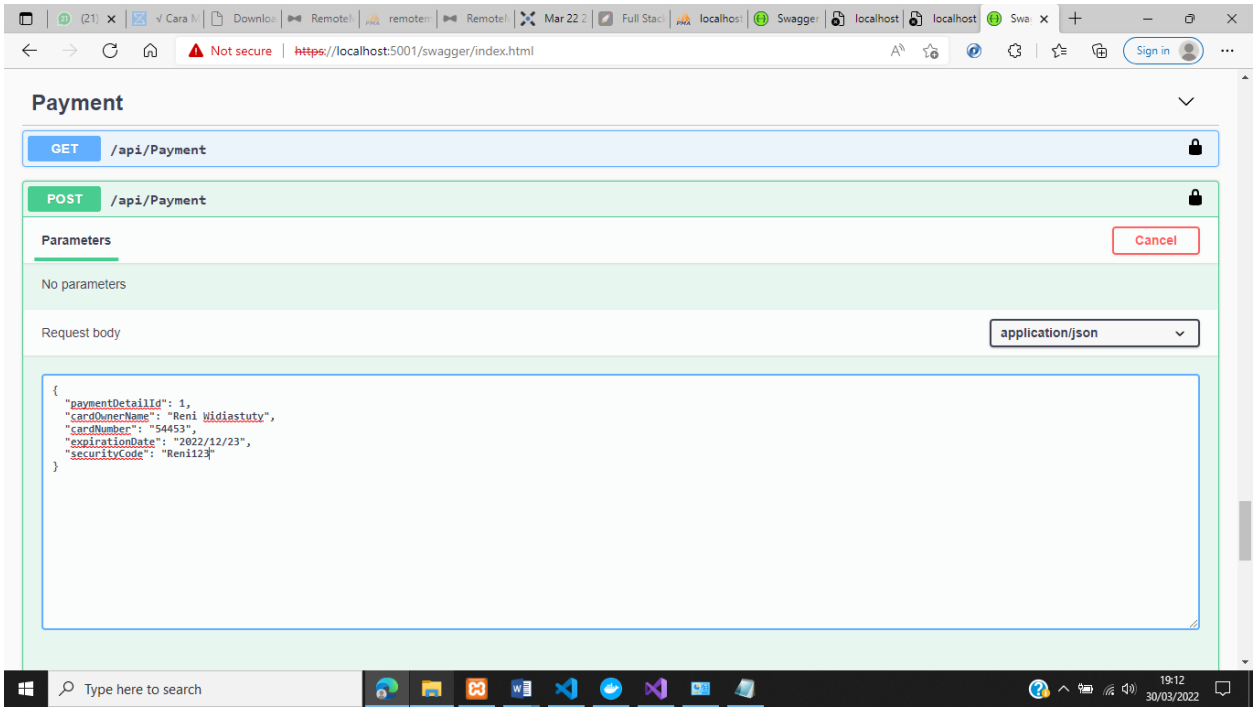
Masukan Bearer lalu disambungkan dengan token yang didapatkan untuk bisa lanjut ke proses CRUD

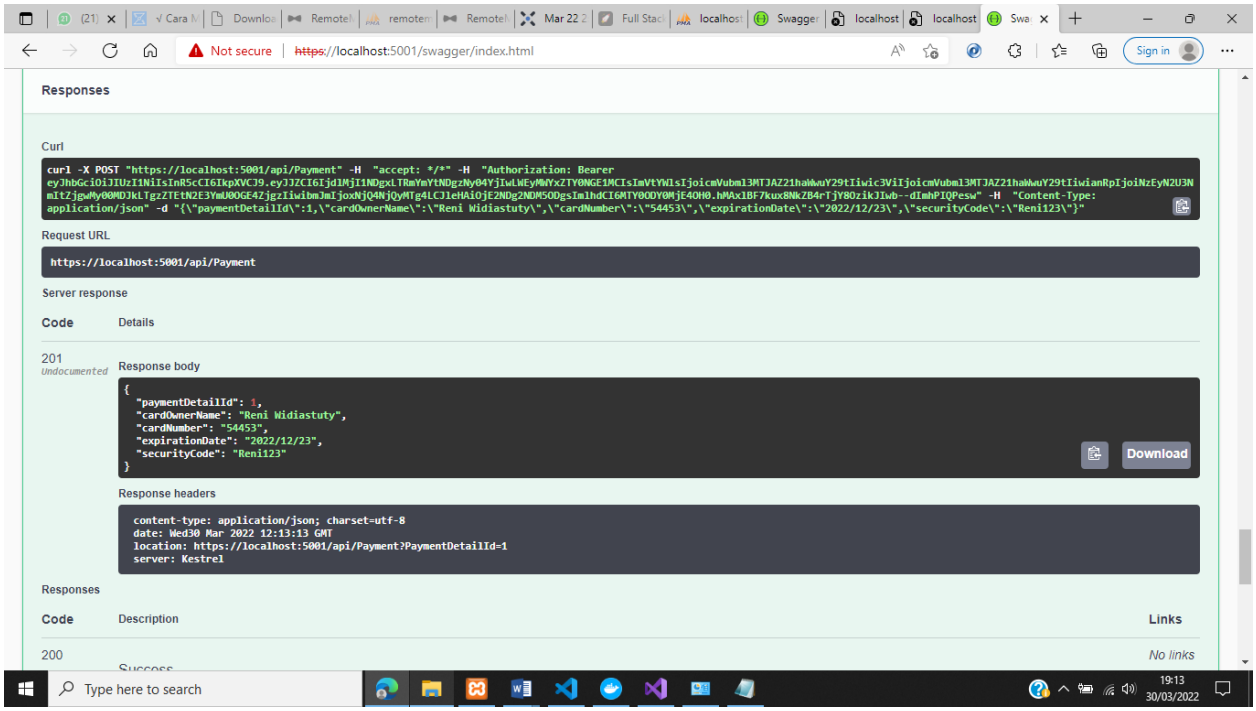




PAYMENT POST

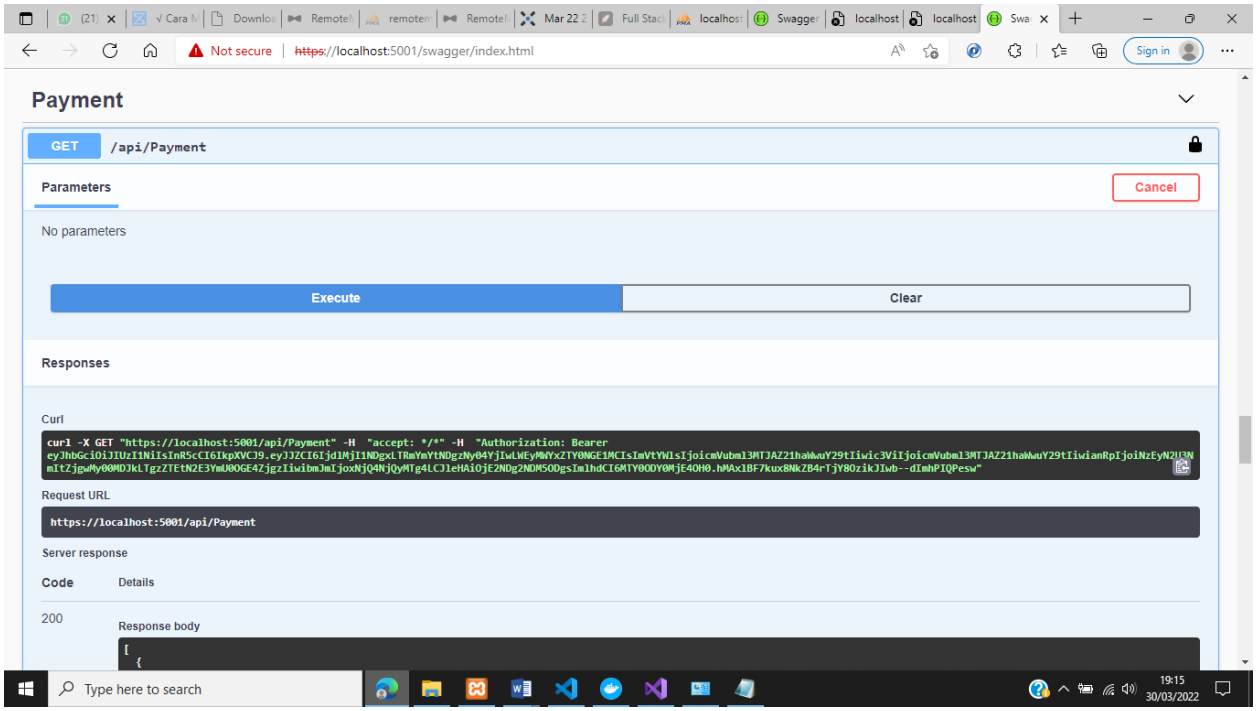
Membuat detail payment baru dengan memasukan data yang diminta. (Masukan 3 data)

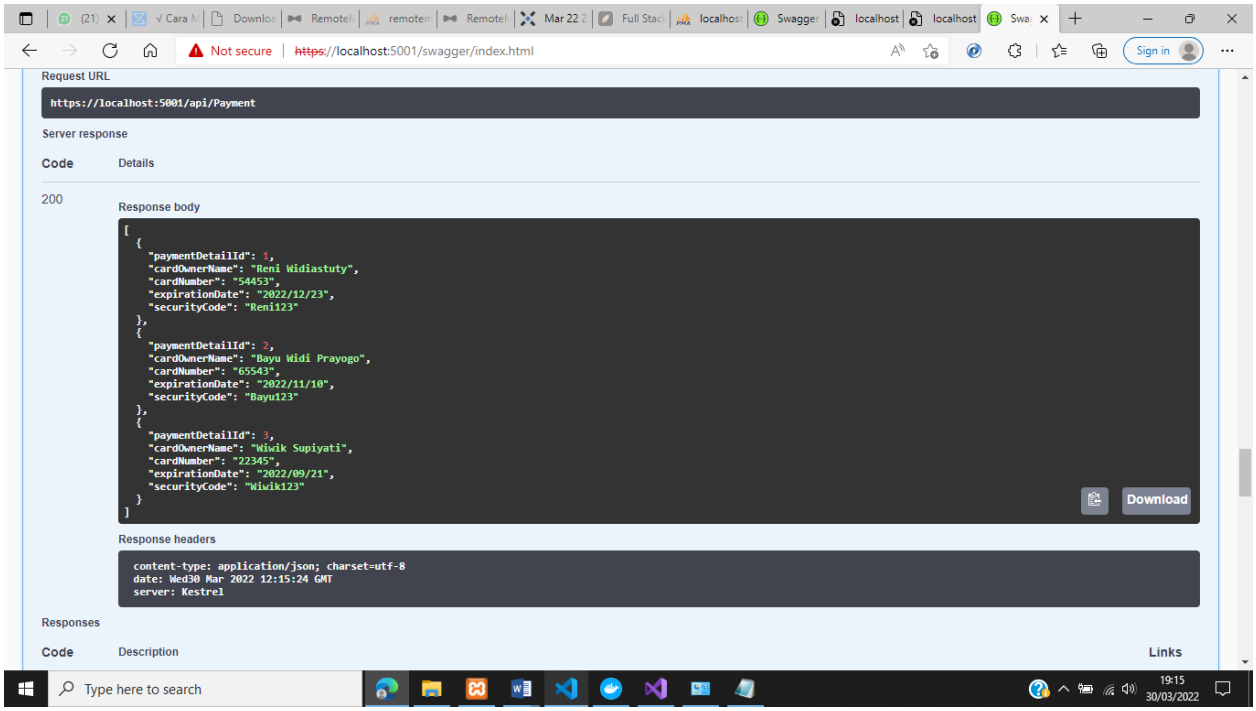




((Setelah berhasil, maka data yang sudah di input akan masuk ke dalam database))

GET PAYMENT Untuk mengecek data yang sudah di input.

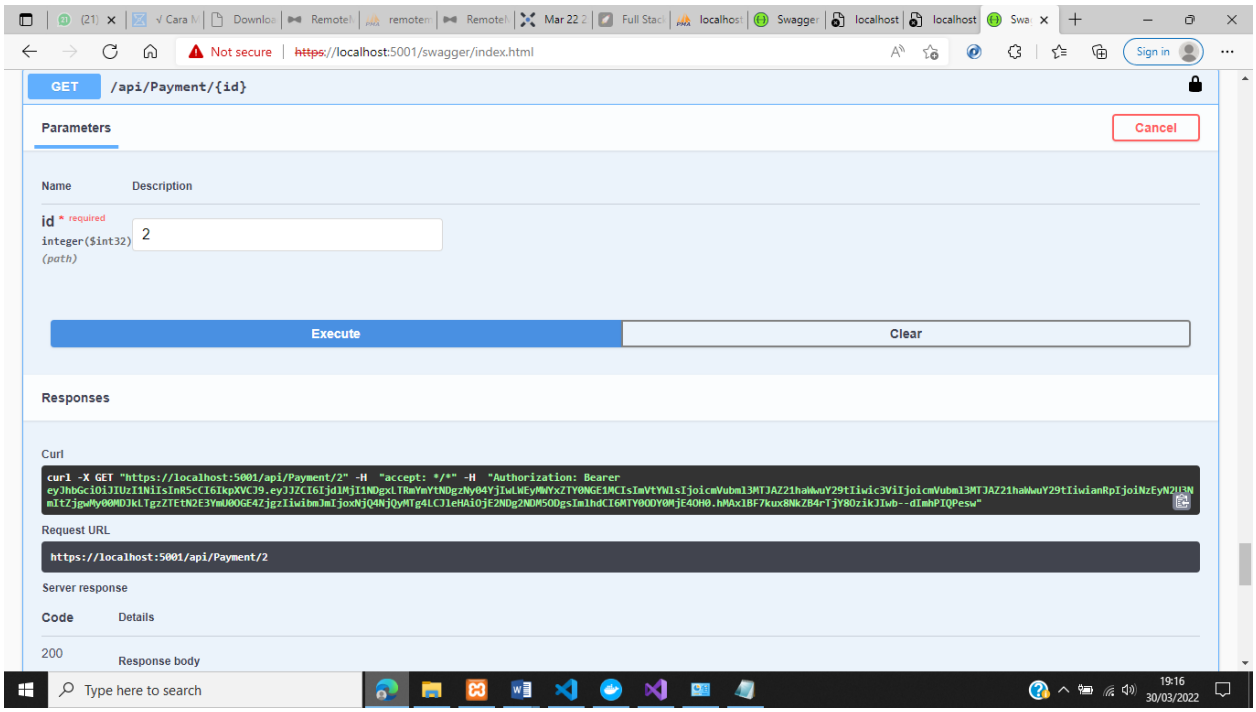


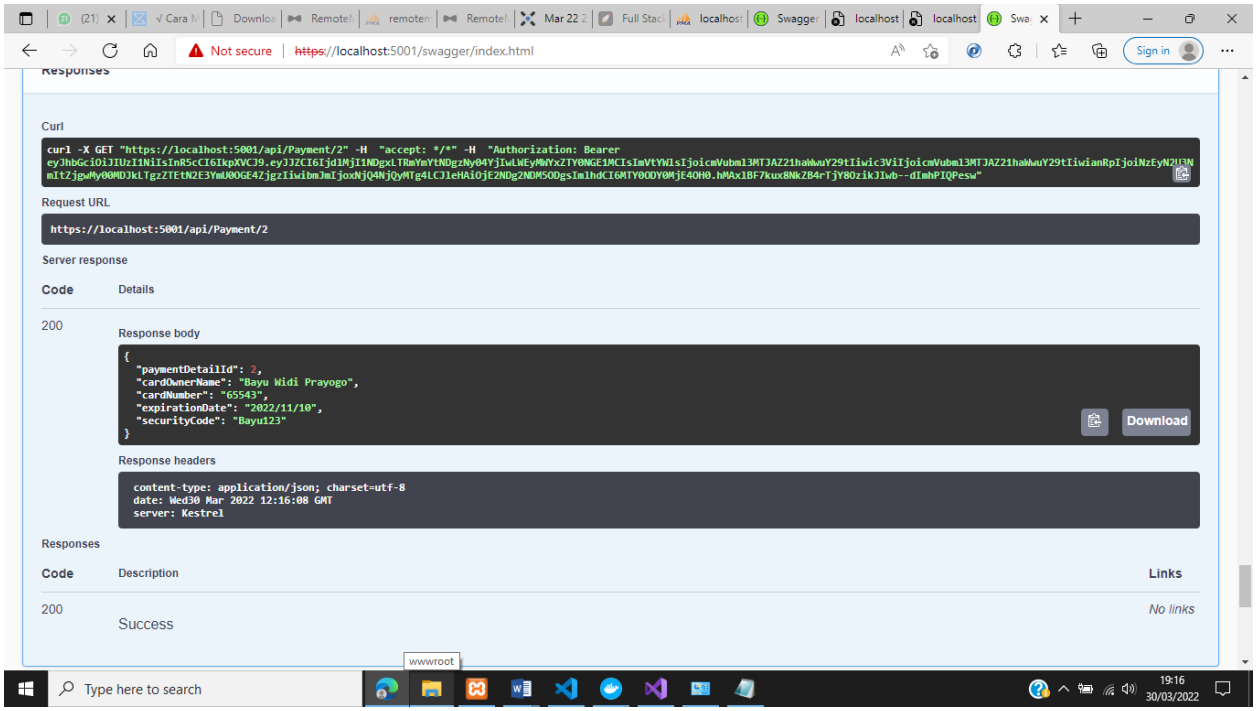


((Data sudah masuk ke dalam database))

GET PAYMENT BERDASARKAN ID

Co: Mamanggil ID = 2

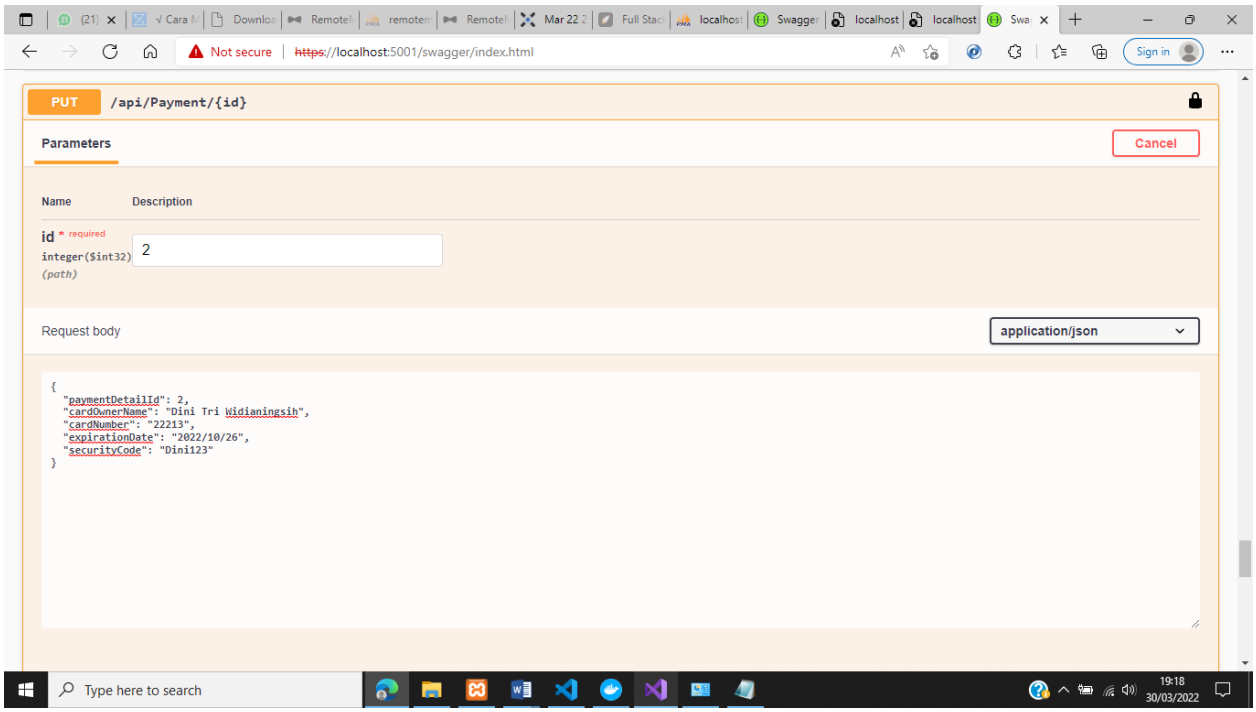


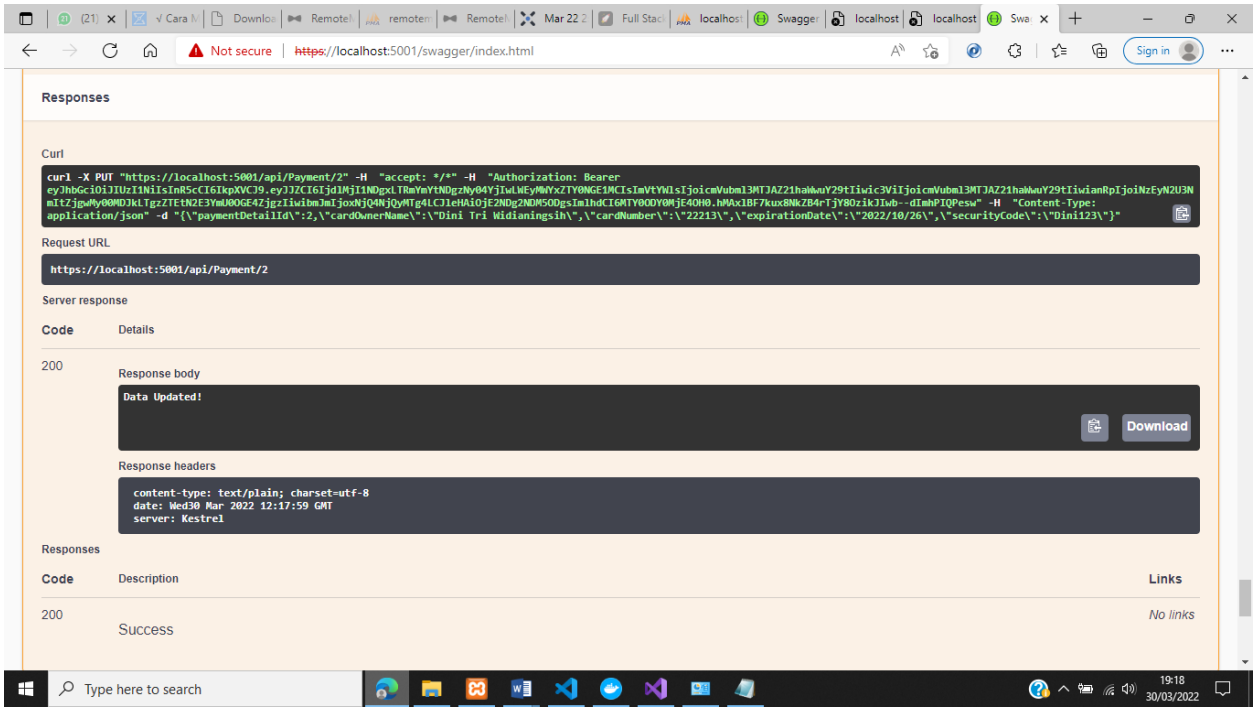


((Maka akan muncul data dengan ID = 2))

PUT / UPDATE

Disini, kita akan mengubah data berdasarkan ID. Co. ID = 2 Lalu masukan data terbaru.

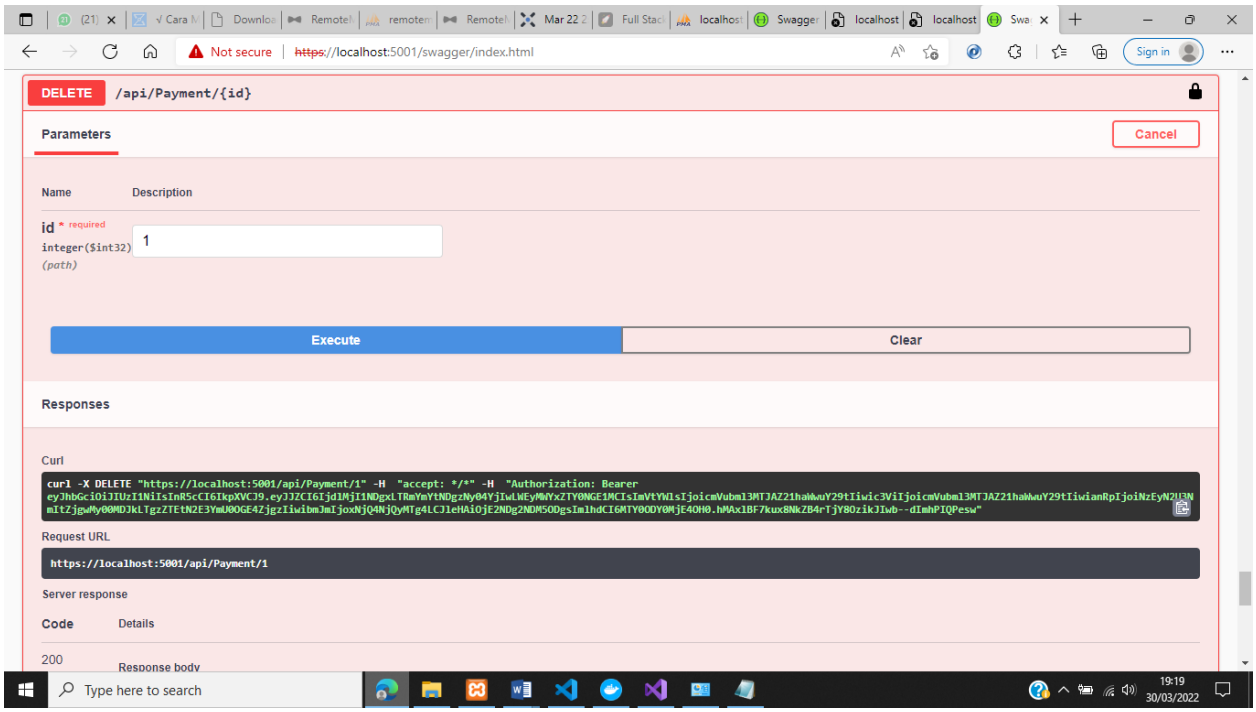


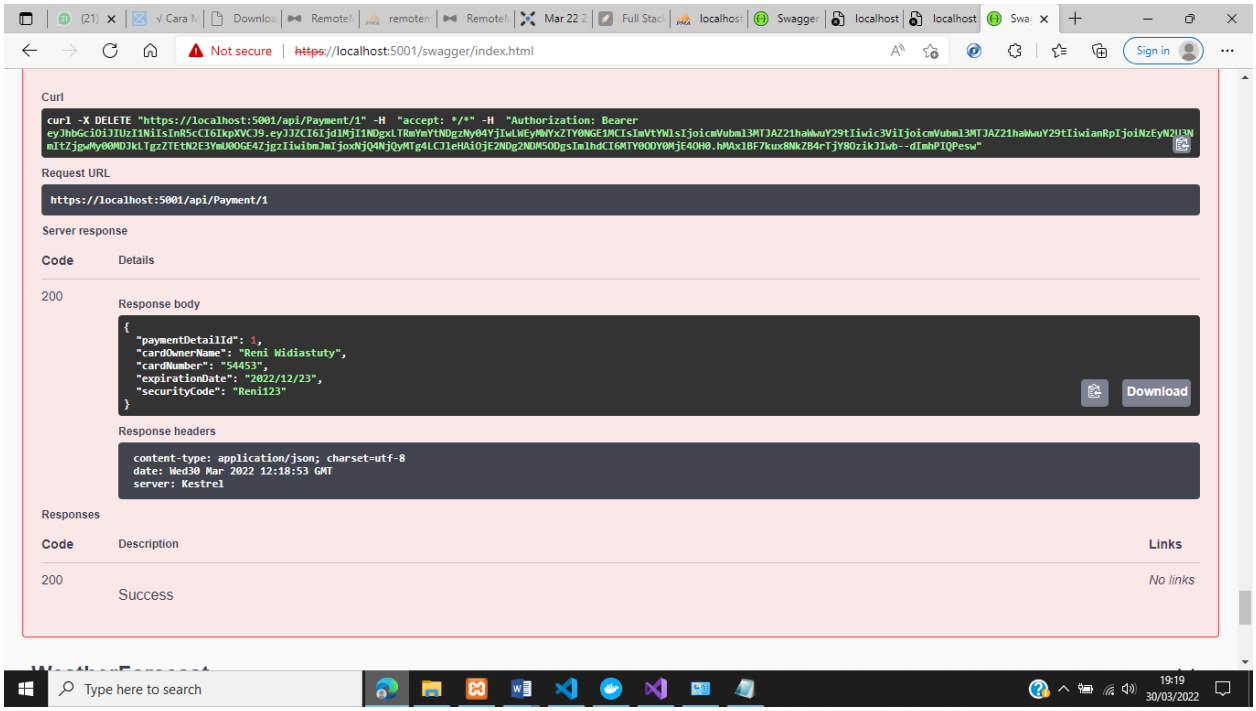


((Data Berhasil Di Update))

DELETE

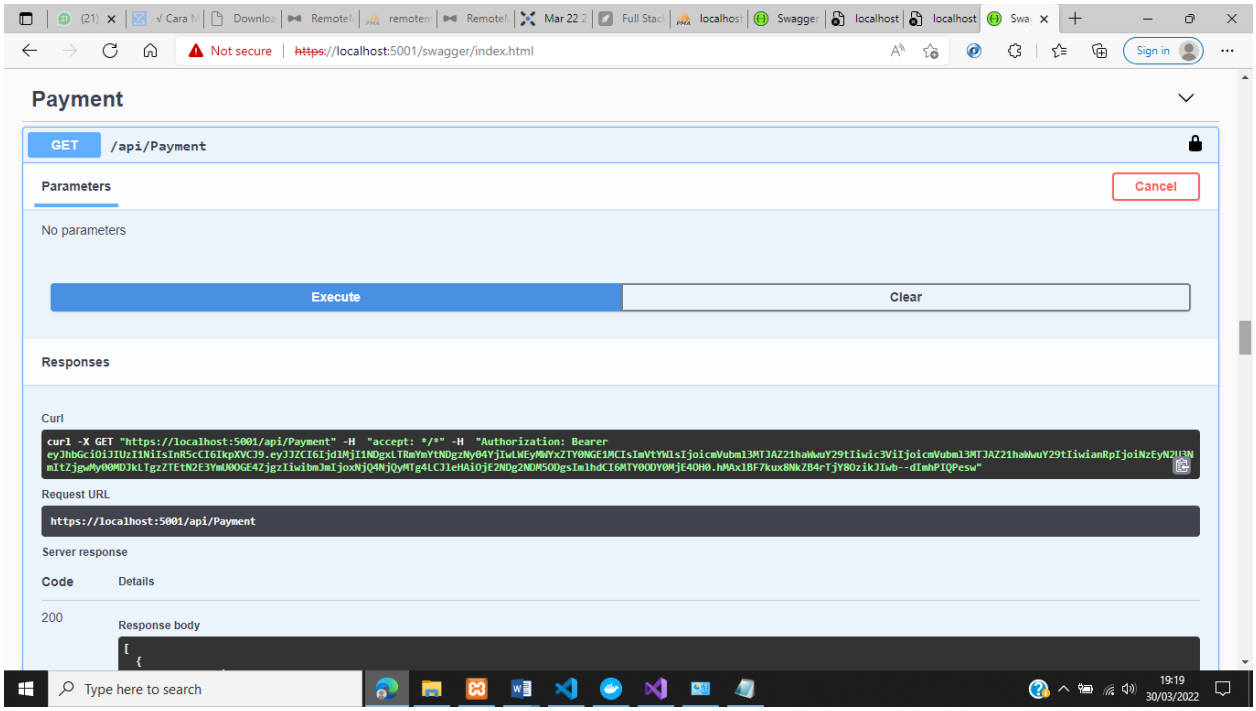
Disini kita akan menghapus data berdasarkan id. Co ID = 1

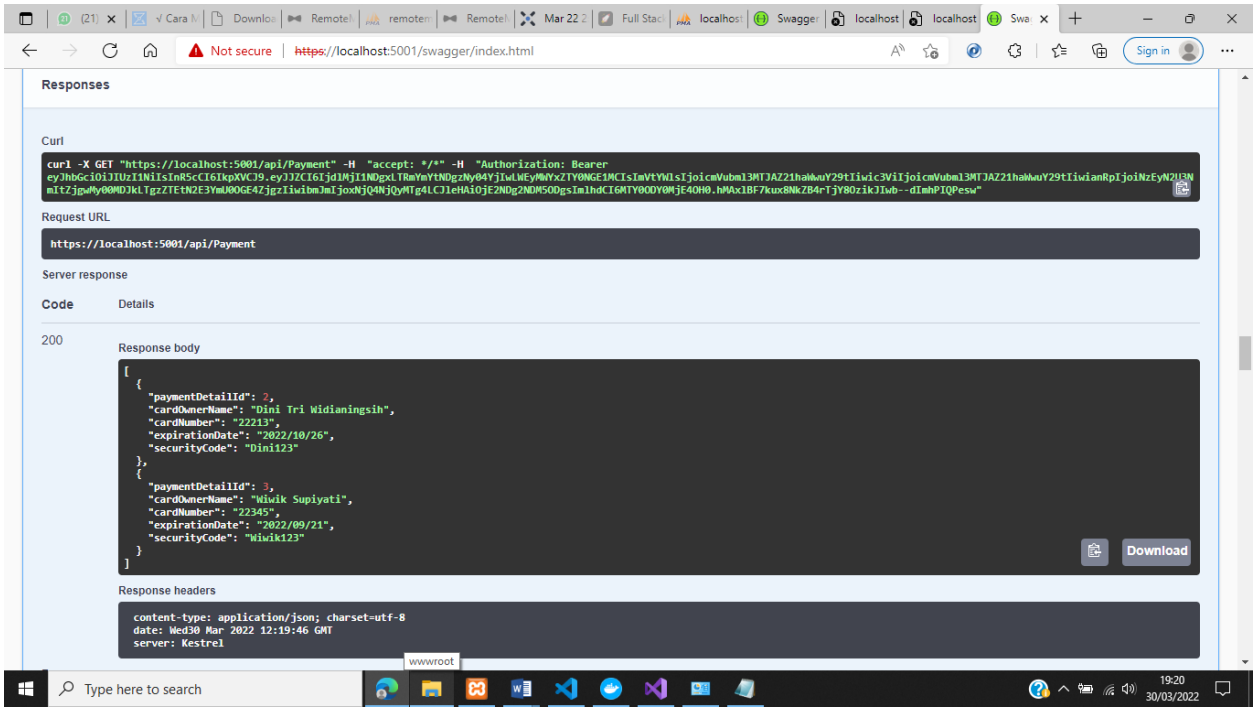




((Respons data telah berhasil di hapus))

CEK KEMBALI MELALUI GET PAYMENT

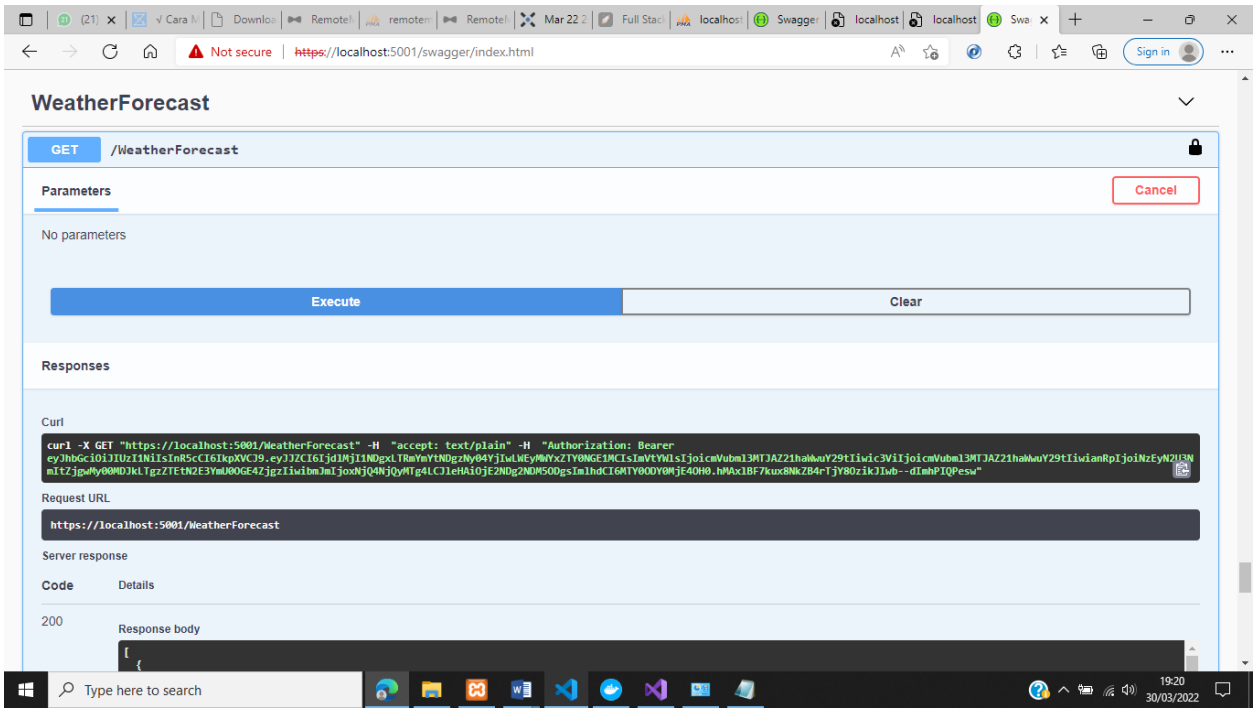


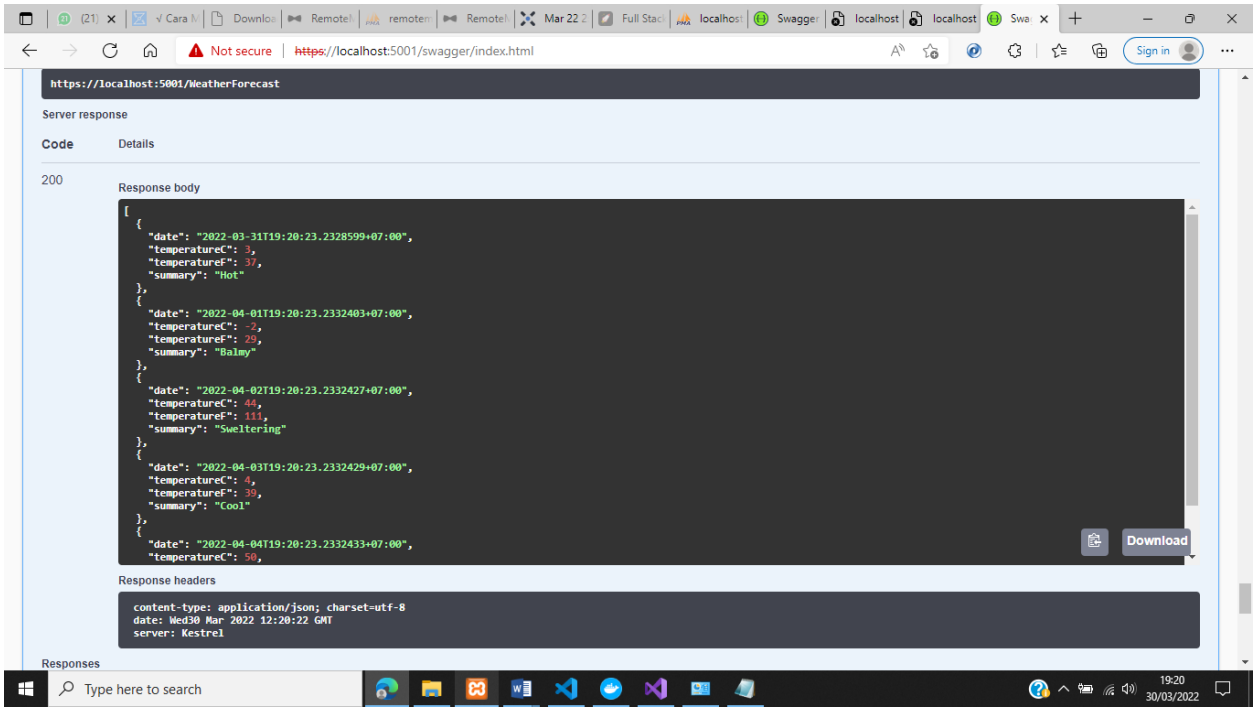


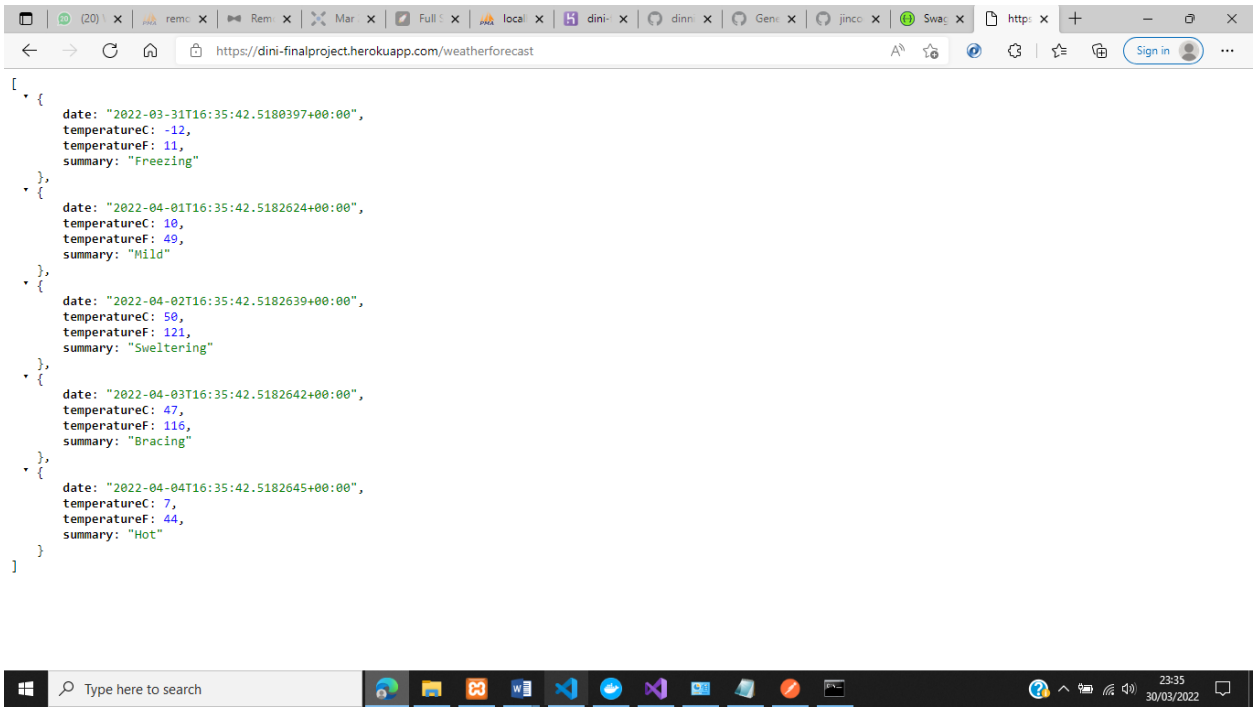
((ID = 1 telah berhasil di hapus))

SWAGGER/WEATHERFORECAST

- Klik Try It Out lalu Execute




Not secure
https://localhost:5001/swagger/index.html
https://localhost:5001/WeatherForecast
Server response
Code
Details
200
Response body
[
{
"date": "2022-03-31T19:20:23.2328599+07:00",
"temperatureC": 3,
"temperatureF": 37,
"summary": "Hot"
},
{
"date": "2022-04-01T19:20:23.2332403+07:00",
"temperatureC": -2,
"temperatureF": 29,
"summary": "Balmy"
},
{
"date": "2022-04-02T19:20:23.2332427+07:00",
"temperatureC": 44,
"temperatureF": 111,
"summary": "Sweltering"
},
{
"date": "2022-04-03T19:20:23.2332429+07:00",
"temperatureC": 4,
"temperatureF": 39,
"summary": "Cool"
},
{
"date": "2022-04-04T19:20:23.2332433+07:00",
"temperatureC": 50,
Download
Response headers
content-type: application/json; charset=utf-8
date: Wed30 Mar 2022 12:20:22 GMT
server: Kestrel
Responses
Type here to search
19:20
30/03/2022

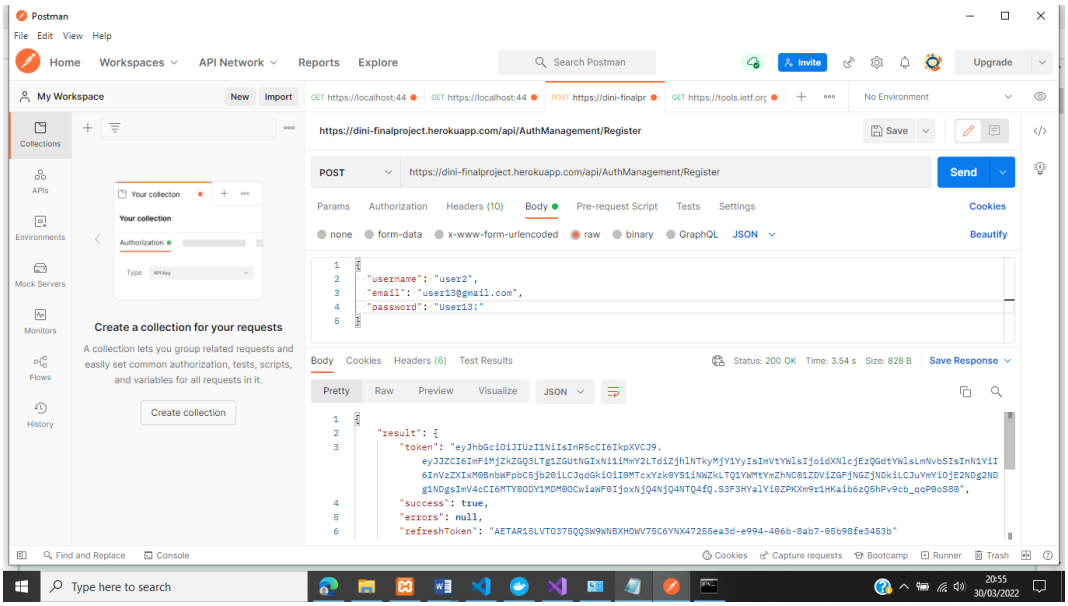

https://dini-finalproject.herokuapp.com/weatherforecast
[
{
date: "2022-03-31T16:35:42.5180397+00:00",
temperatureC: -12,
temperatureF: 11,
summary: "Freezing"
},
{
date: "2022-04-01T16:35:42.5182624+00:00",
temperatureC: 10,
temperatureF: 49,
summary: "Mild"
},
{
date: "2022-04-02T16:35:42.5182639+00:00",
temperatureC: 50,
temperatureF: 121,
summary: "Sweltering"
},
{
date: "2022-04-03T16:35:42.5182642+00:00",
temperatureC: 47,
temperatureF: 116,
summary: "Bracing"
},
{
date: "2022-04-04T16:35:42.5182645+00:00",
temperatureC: 7,
temperatureF: 44,
summary: "Hot"
}
]
Type here to search
23:35
30/03/2022

POSTMAN

REGISTER

Link : https://dini-finalproject.herokuapp.com/api/AuthManagement/Register

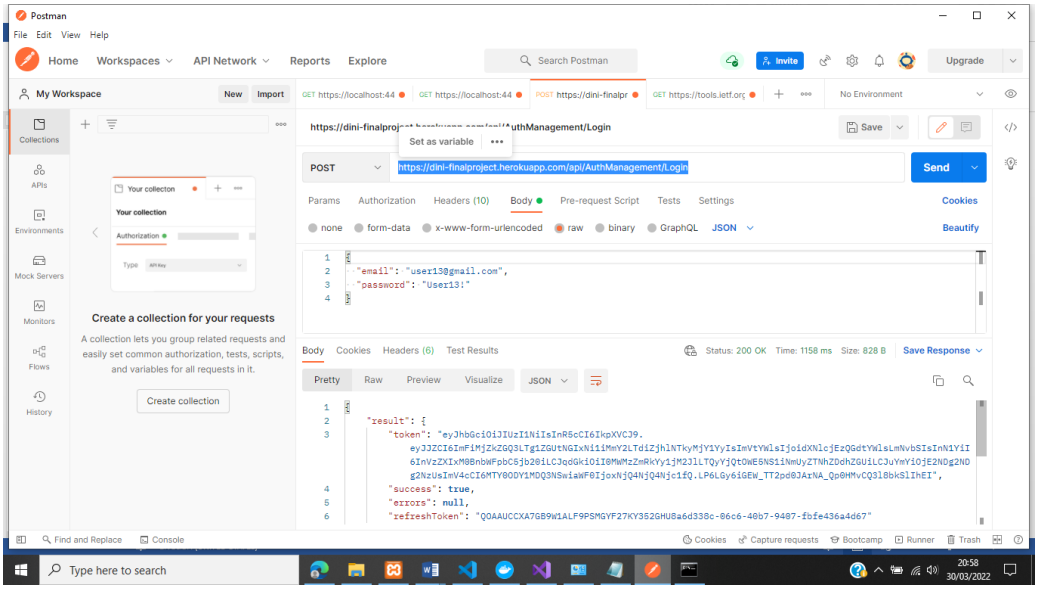
- Masukan username, email, dan password baru



LOGIN

- Silahkan login menggunakan email dan password yang telah dibuat.
- Lalu akan didapatkan token untuk melakukian autentikasi dimana token hanya berlaku selama 30 menit.

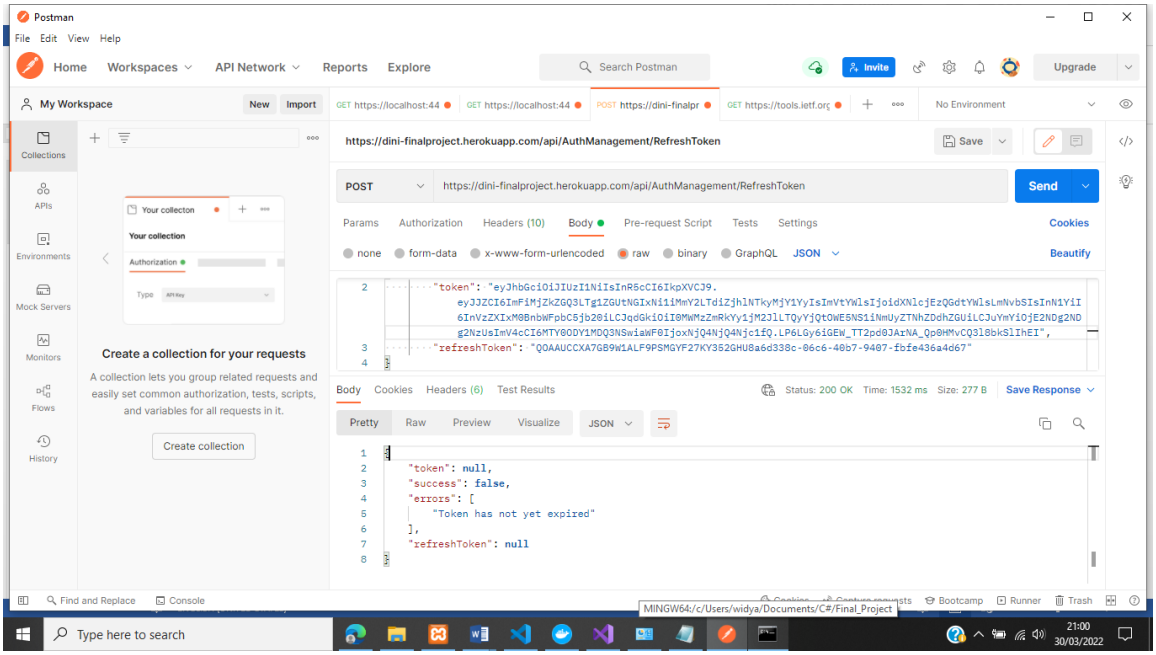
Link : https://dini-finalproject.herokuapp.com/api/AuthManagement/Login



REFRESH TOKEN

Link : https://dini-finalproject.herokuapp.com/api/AuthManagement/RefreshToken

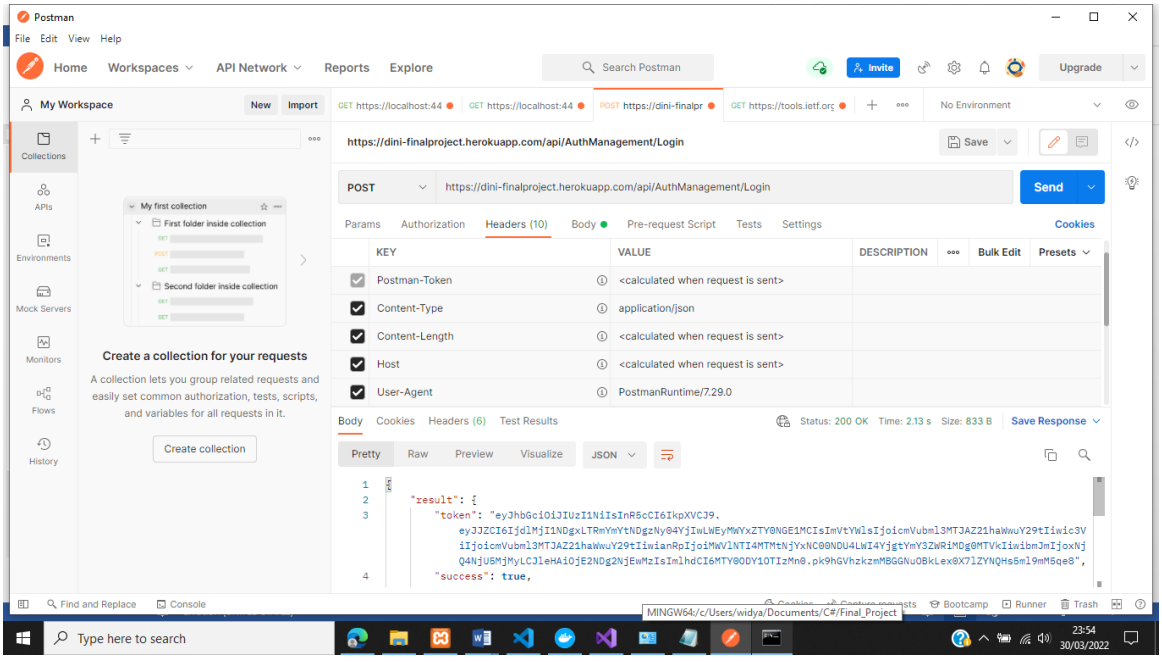
Percobaan melakukan refresh token sebelum 30 menit akan menampilkan pesan gagal.



AUTHORIZE

Melakukan autorisasi dengan menambahkan key baru “Authorization” dan diisi dengan Bearer key dimana key adalah token yang didapatkan dari hasil login.

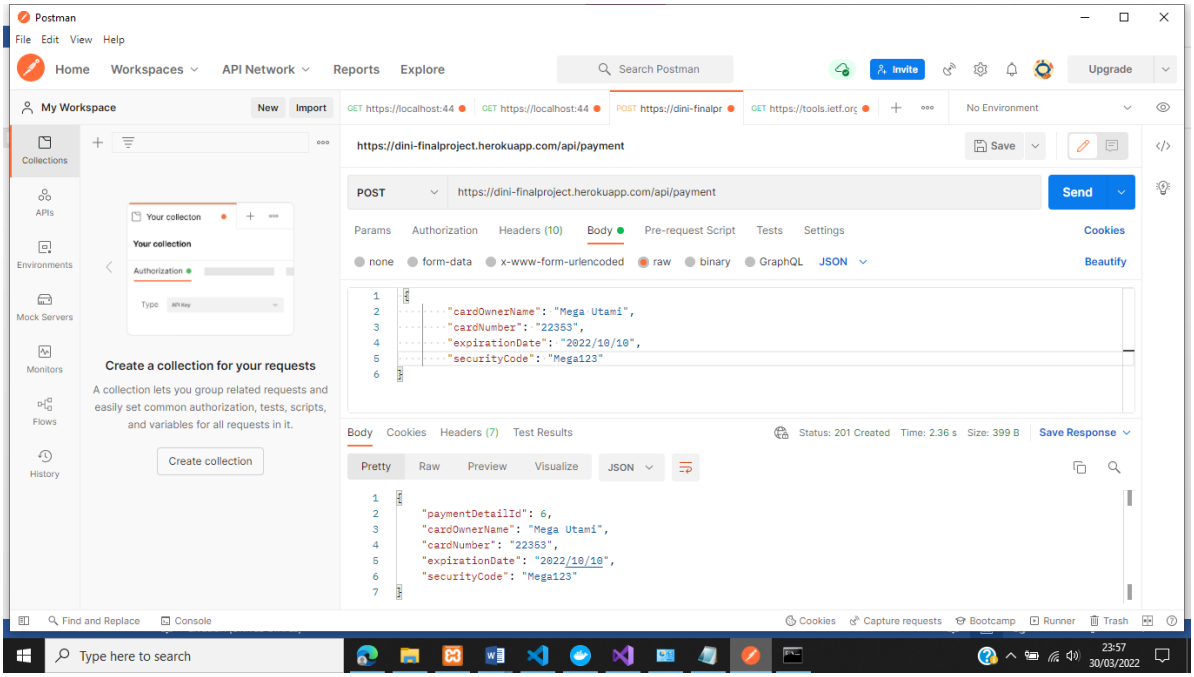
Link : https://dini-finalproject.herokuapp.com/api/Payment



PAYMENT POST

Membuat data payment baru dengan mengisi id, nama, tanggal expired, dan kode keamanan.

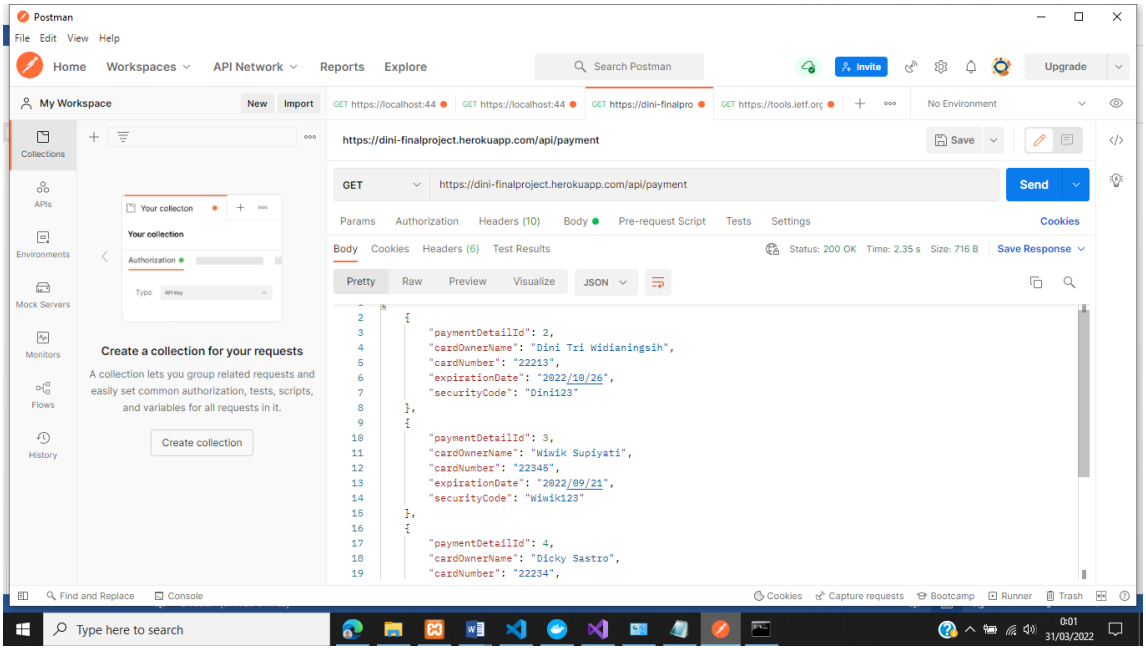
Link : https://dini-finalproject.herokuapp.com/api/payment



PAYMENT GET

Mendapatkan Semua data yang ada didalam database.

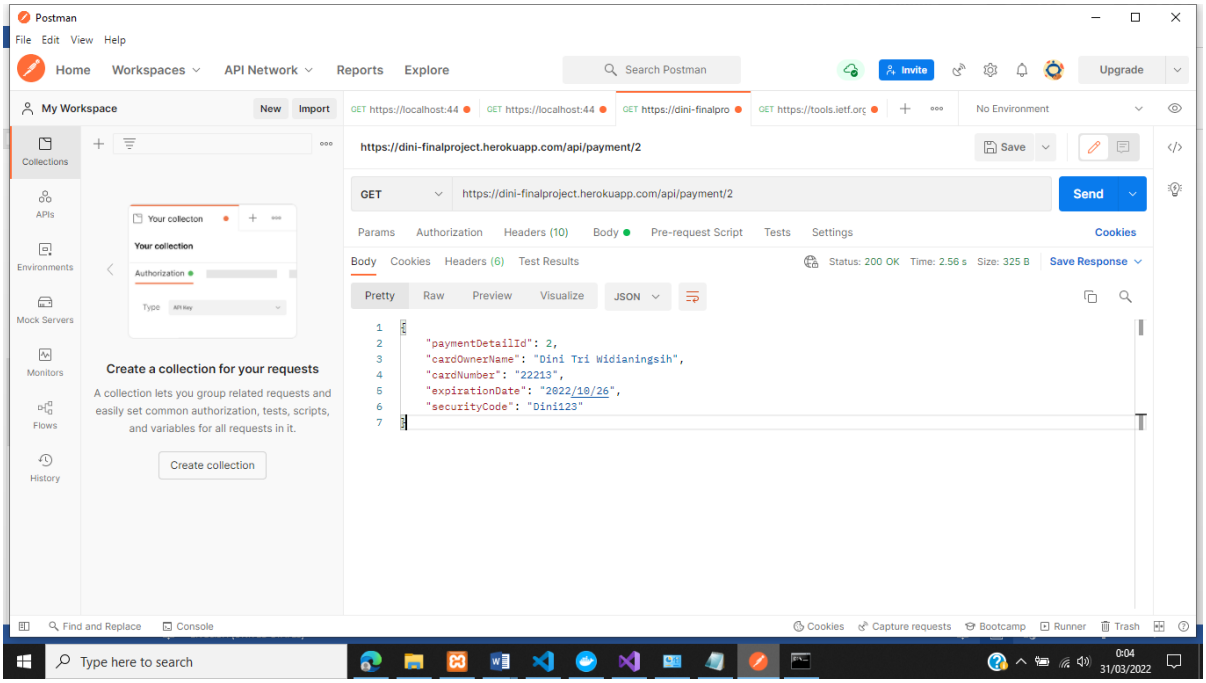
Link : https://dini-finalproject.herokuapp.com/api/payment



PAYMENT GET WHERE ID

Mendapatkan data payment details dengan ID yang dimasukan.

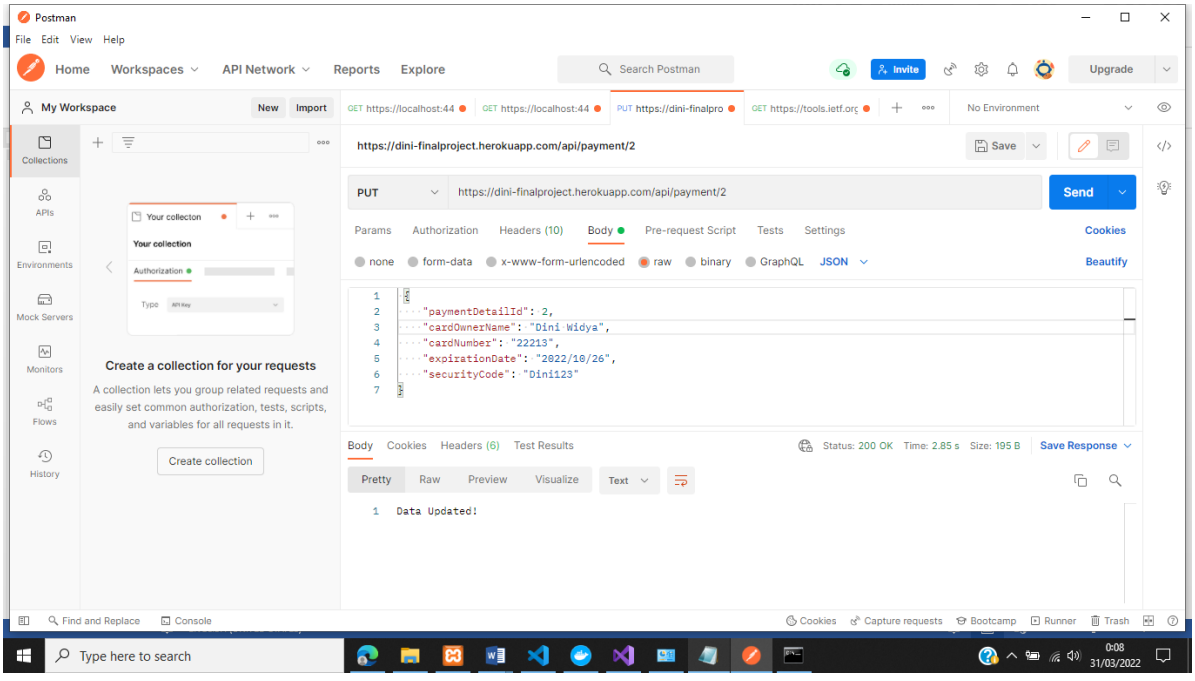
Link : https://dini-finalproject.herokuapp.com/api/payment/2



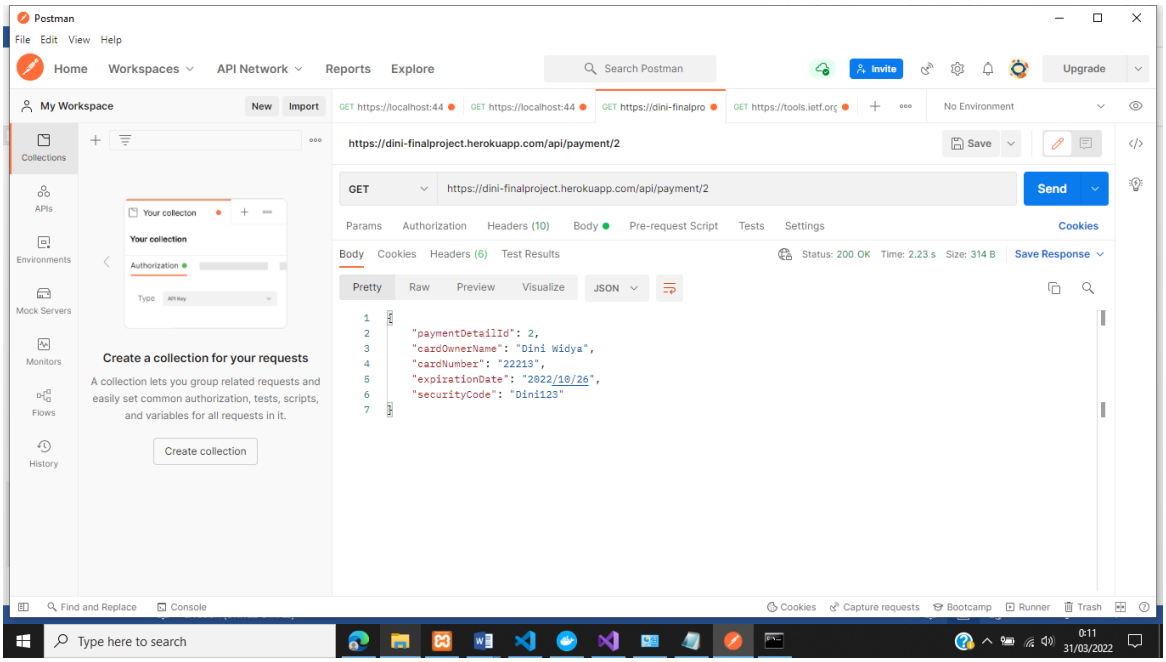
PAYMENT PUT

Mengubah data dengan ID tertentu lalu menampilkan respons perubahan.

Link : https://dini-finalproject.herokuapp.com/api/payment/2



((Cek Get Payment Kembali))

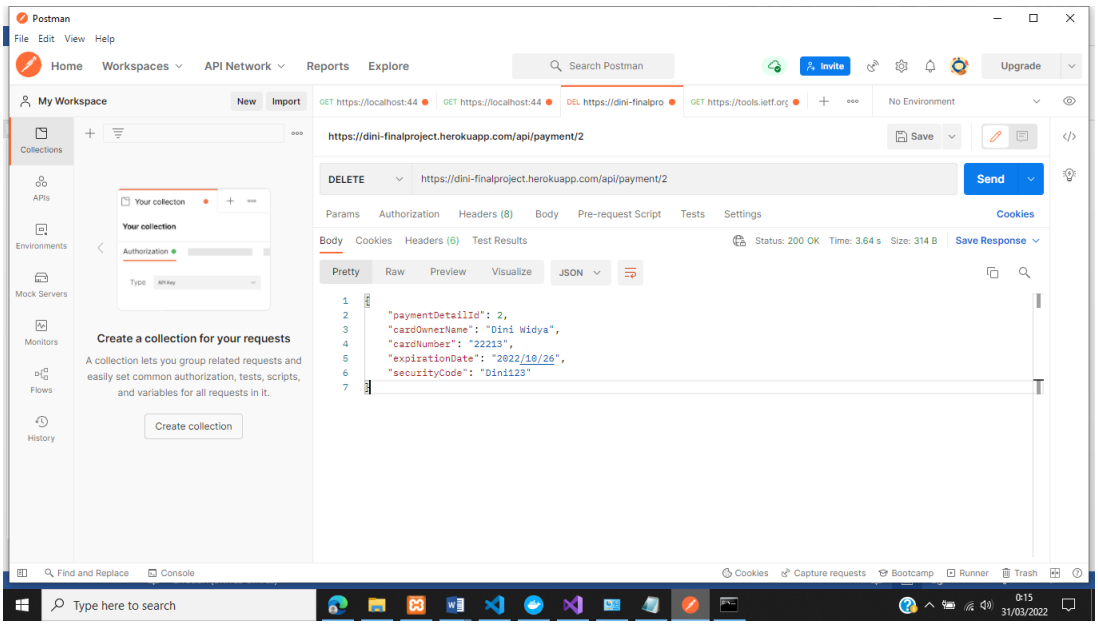


Data berhasil di ubah.

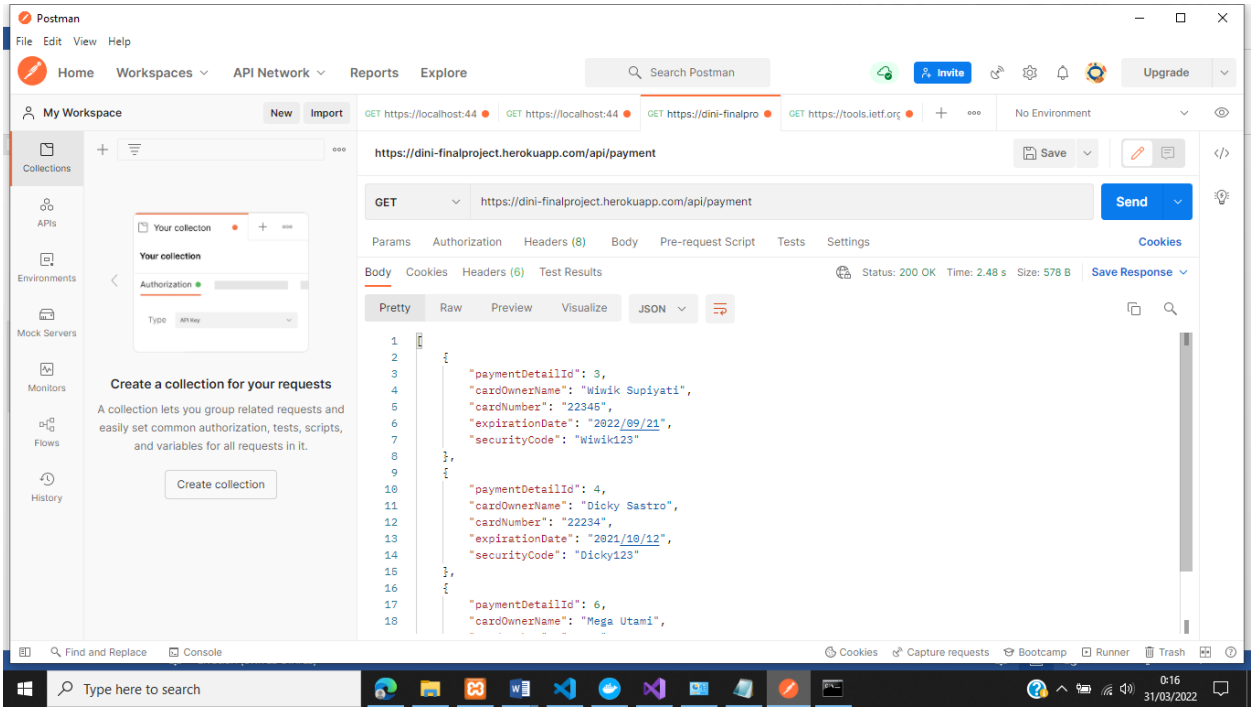
DELETE PAYMENT

Menghapus data dengan ID tertentu dan menampilkan response berupa data yang berhasil dihapus.

Link : https://dini-finalproject.herokuapp.com/api/payment/2



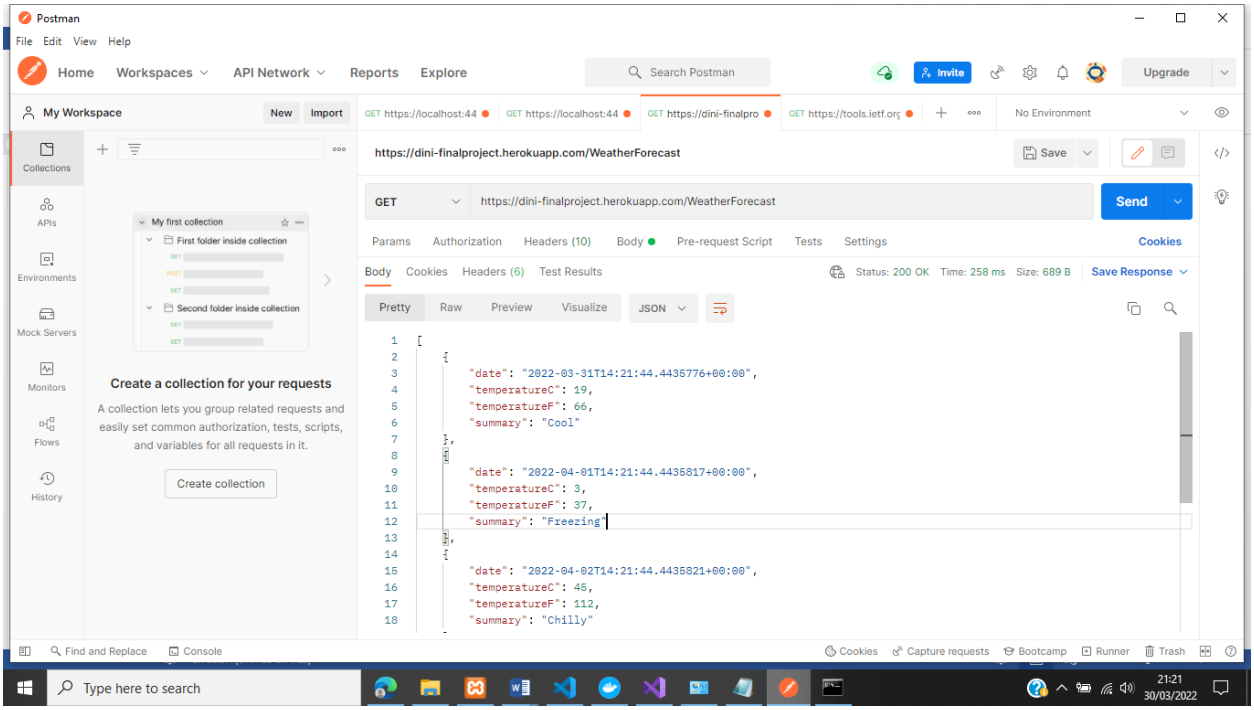
((Cek GET Kembali))



ID = 2 sudah tidak ada di dalam database.

GET WEATHERFORECAST

Link : https://dini-finalproject.herokuapp.com/WeatherForecast



SAVE HASIL CRUD DARI POSTMAN

1. Response Register	31/03/2022 0:19	JSON Source File	1 KB
2. Response Login	31/03/2022 0:20	JSON Source File	1 KB
3. Response Refresh Token	31/03/2022 0:22	JSON Source File	1 KB
4. Response Post Payment	30/03/2022 23:59	JSON Source File	1 KB
5. Respons Get Payment	31/03/2022 0:02	JSON Source File	1 KB
6. Response Get Payment Where ID	31/03/2022 0:05	JSON Source File	1 KB
7. Response Put Payment	31/03/2022 0:10	JSON Source File	1 KB
8. Respons Delete Payment	31/03/2022 0:14	JSON Source File	1 KB
9. Response WeatherForecast	31/03/2022 0:17	JSON Source File	1 KB