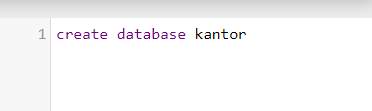
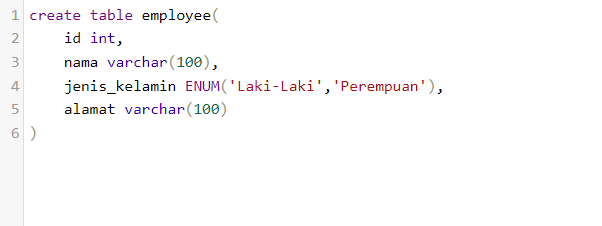
Nama : Alexander Radianta Tarigan

No\_Peserta : FSDO003ONL010

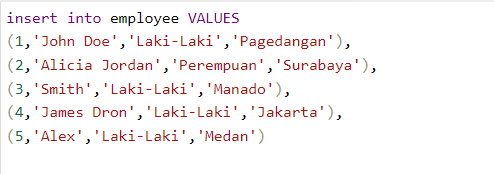
Pertama, buat sebuah database baru Bernama kantor

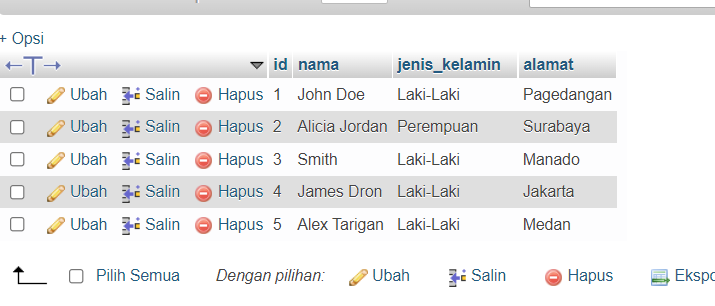


Buat table baru bernama employee



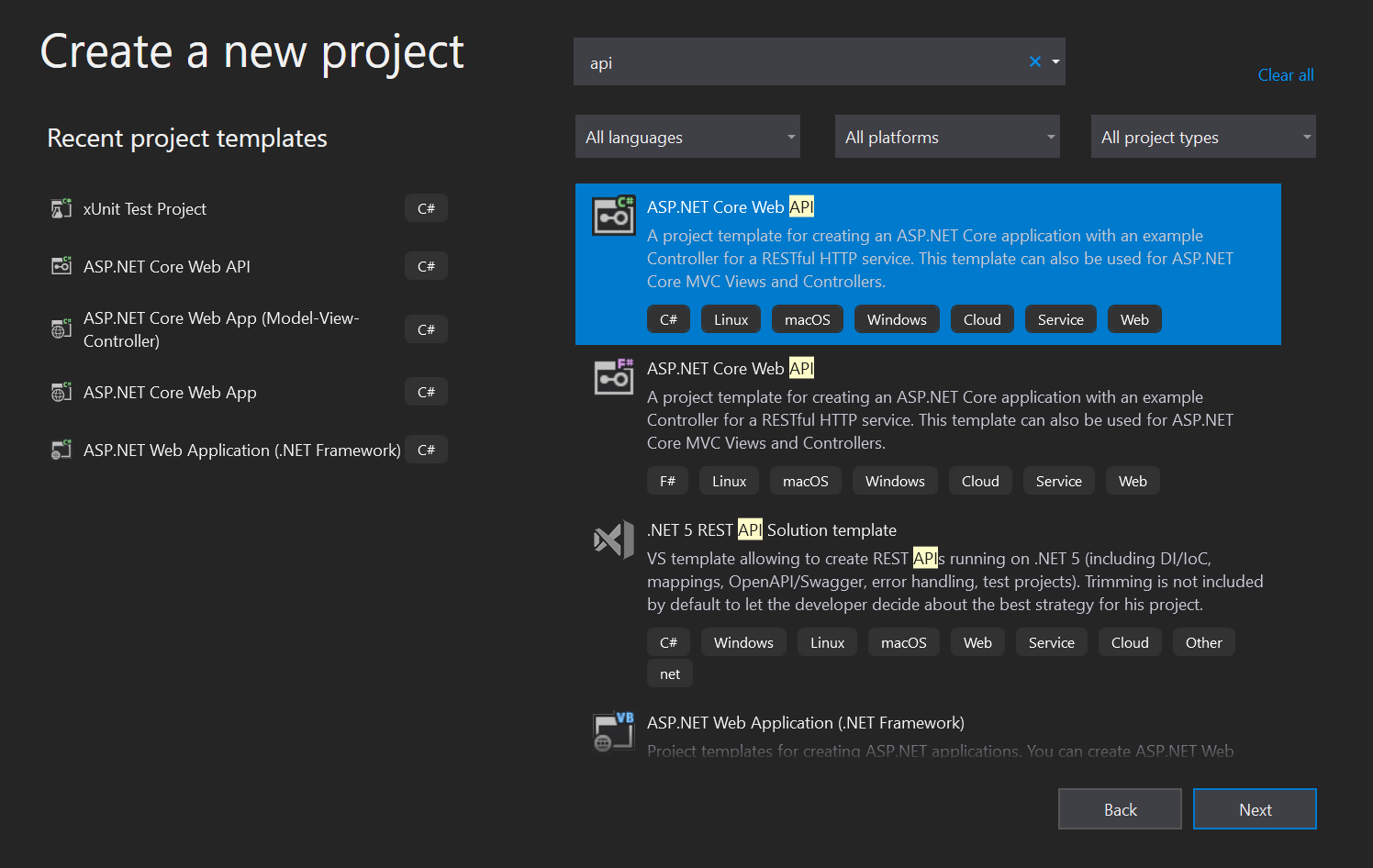
Tambahkan beberapa data



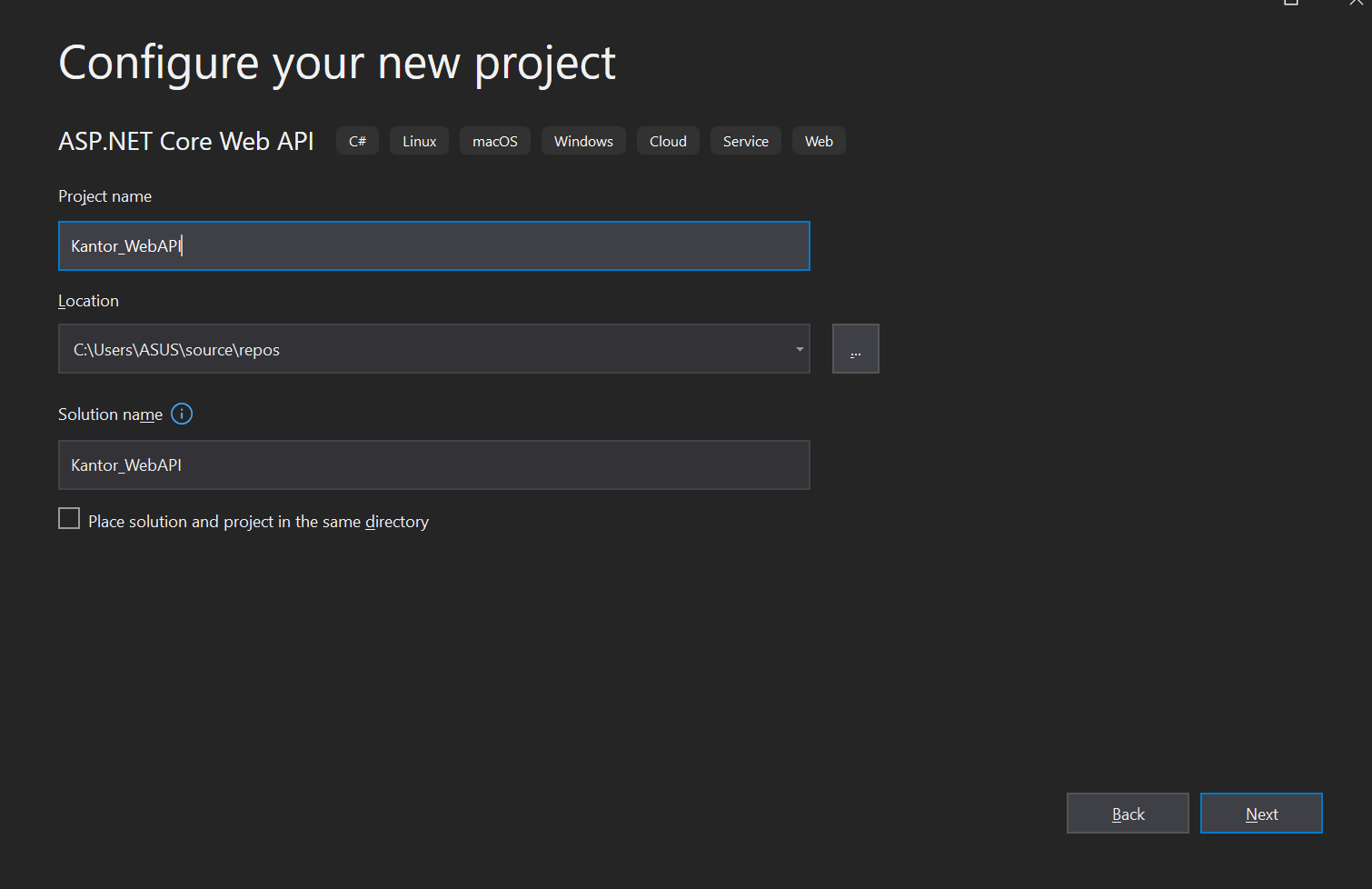


Buka visual studio

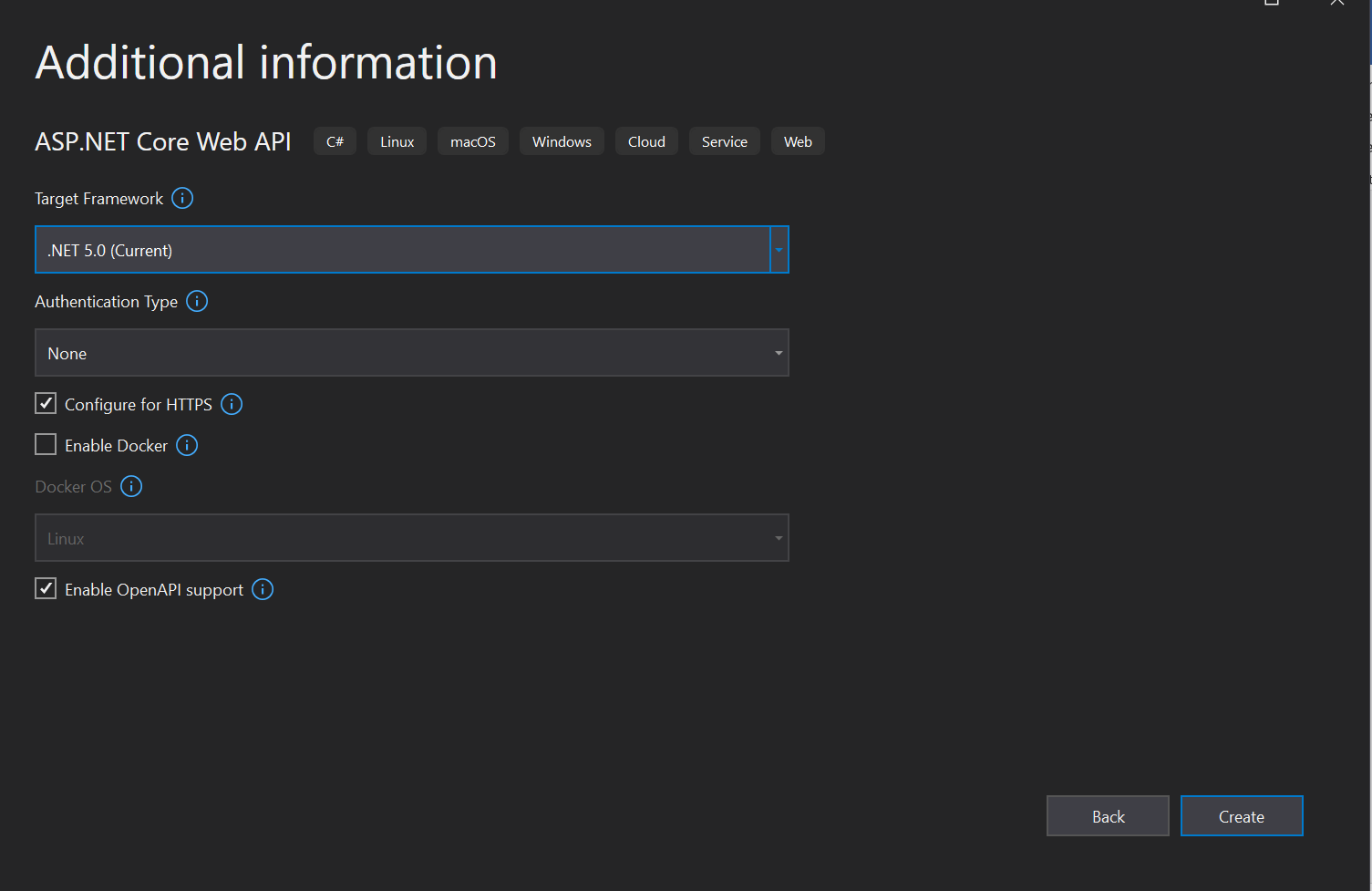
Buat project baru menggunakan ASP.NET Core Web API,lalu klik tombol next



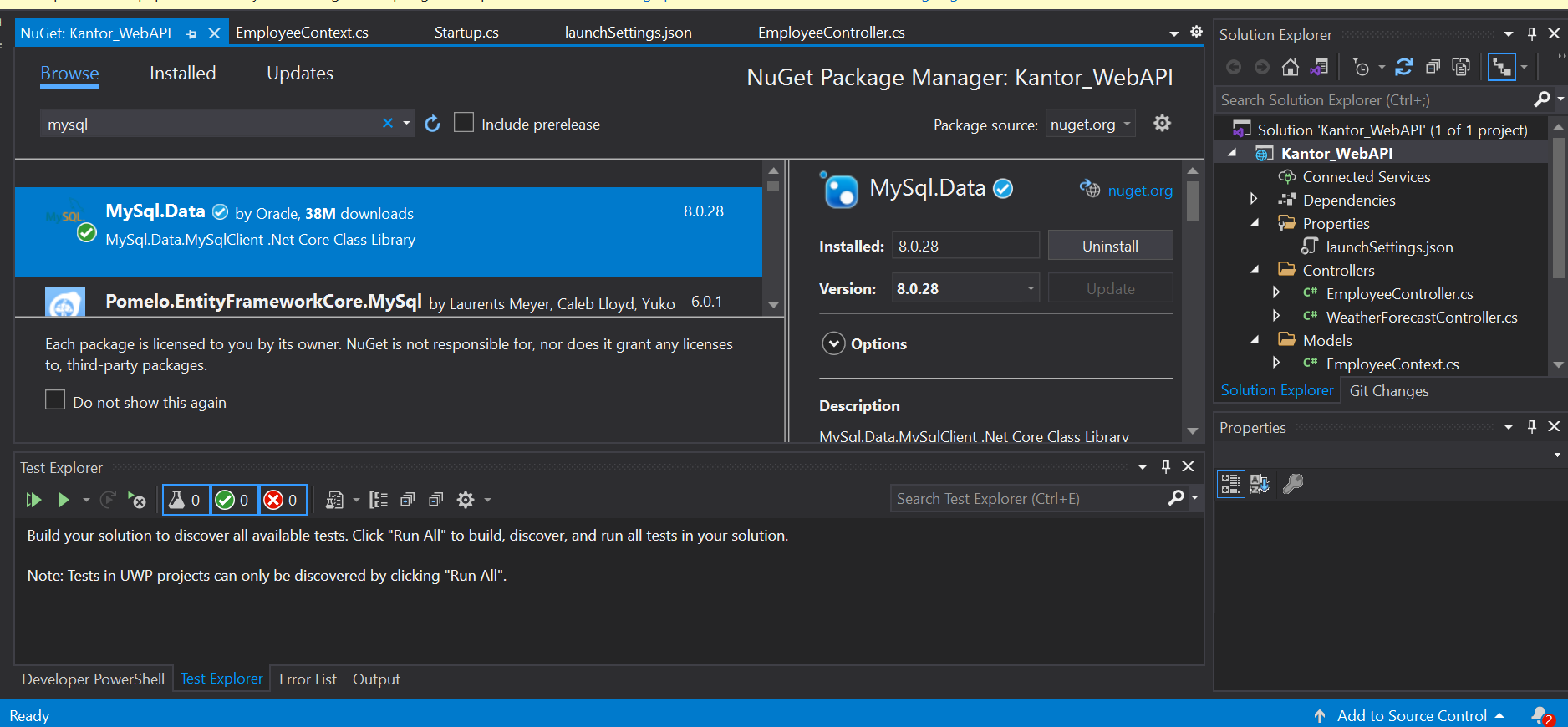
Tulis nama project, lalu klik next



Pilih .NET 5.0 framework,ceklis configure for https dan Enable OpenAPI support, lalu klik create.



Install package Mysql di Nuget project.

Selanjutnya kita perlu membuat sebuah folder bernama Models, lalu pada folder Models buat sebuah class bernama EmployeeItem.cs. class EmployeeItem.cs berfungsi menampung atribut-atribut yang diterima dari database.

public class EmployeeItem

    {

        private Models.EmployeeContext context;

        public int id { get; set; }

        public string nama { get; set; }

        public string jenisKelamin { get; set; }

        public string alamat { get; set; }

    }

Selanjutnya kita perlu membuat sebuah class yang berperan dalam melakukan komunikasi dan request kepada database. Klik kanan pada folder Models lalu klik add class dan beri nama EmployeeContext.cs

public class EmployeeContext

    {

        public string ConnectionString { get; set; }

        public EmployeeContext(string connectionString)

        {

            this.ConnectionString = connectionString;

        }

        private MySqlConnection GetConnection()

        {

            return new MySqlConnection(ConnectionString);

        }

        public List<EmployeeItem> GetAllEmployee()

        {

            List<EmployeeItem> list = new List<EmployeeItem>();

            using (MySqlConnection conn = GetConnection())

            {

                conn.Open();

                MySqlCommand cmd = new MySqlCommand("SELECT \* FROM employee", conn);

                using (MySqlDataReader reader = cmd.ExecuteReader())

                {

                    while (reader.Read())

                    {

                        list.Add(new EmployeeItem()

                        {

                            id = reader.GetInt32("id"),

                            nama = reader.GetString("nama"),

                            jenisKelamin = reader.GetString("jenis\_kelamin"),

                            alamat = reader.GetString("alamat")

                        });

                    }

                }

            }

            return list;

        }

        public List<EmployeeItem> GetEmployee(string id)

        {

            List<EmployeeItem> list = new List<EmployeeItem>();

            using (MySqlConnection conn = GetConnection())

            {

                conn.Open();

                MySqlCommand cmd = new MySqlCommand("SELECT \* FROM employee WHERE id =@id", conn);

                cmd.Parameters.AddWithValue("@id", id);

                using (MySqlDataReader reader = cmd.ExecuteReader())

                {

                    while (reader.Read())

                    {

                        list.Add(new EmployeeItem()

                        {

                            id = reader.GetInt32("id"),

                            nama = reader.GetString("nama"),

                            jenisKelamin = reader.GetString("jenis\_kelamin"),

                            alamat = reader.GetString("alamat")

                        });

                    }

                }

            }

            return list;

        }

    }

Selanjutnya membuat Sebuah Class Controller yang digunakan untuk mengolah data yang masuk melalui EmployeeItem dan EmployeeContext.klik kanan folder Controllers lalu add class dan beri nama EmployeeController.cs

[Route("api/[controller]")]

    [ApiController]

    public class EmployeeController : ControllerBase

    {

        private EmployeeContext \_context;

        public EmployeeController(EmployeeContext context)

        {

            this.\_context = context;

        }

        //GET : api/user

        [HttpGet(Name = "Get Employee")]

        public ActionResult<IEnumerable<EmployeeItem>> GetEmployeeItem()

        {

            \_context = HttpContext.RequestServices.GetService(typeof(EmployeeContext)) as EmployeeContext;

            //return new stringg[]

            return \_context.GetAllEmployee();

        }

        //Get : api/user/{id}

        [HttpGet("{id}", Name = "Get Employee Where")]

        public ActionResult<IEnumerable<EmployeeItem>> GetEmployeeItem(String id)

        {

            \_context = HttpContext.RequestServices.GetService(typeof(EmployeeContext)) as EmployeeContext;

            return \_context.GetEmployee(id);

        }

    }

Dan selanjutnya buka file launchsettings.json (folder properties), lalu ubah atribut launchUrl dari api/values jadi api/employee

{

  "$schema": "http://json.schemastore.org/launchsettings.json",

  "iisSettings": {

    "windowsAuthentication": false,

    "anonymousAuthentication": true,

    "iisExpress": {

      "applicationUrl": "http://localhost:29423",

      "sslPort": 44343

    }

  },

  "profiles": {

    "IIS Express": {

      "commandName": "IISExpress",

      "launchBrowser": true,

      "launchUrl": "api/employee",

      "environmentVariables": {

        "ASPNETCORE\_ENVIRONMENT": "Development"

      }

    },

    "Kantor\_WebAPI": {

      "commandName": "Project",

      "dotnetRunMessages": "true",

      "launchBrowser": true,

      "launchUrl": "api/employee",

      "applicationUrl": "https://localhost:5001;http://localhost:5000",

      "environmentVariables": {

        "ASPNETCORE\_ENVIRONMENT": "Development"

      }

    }

  }

}

Lalu buka file appsetings.json, lakukan konfigurasi pada defaultconnection

{

  "ConnectionStrings": {

    "DefaultConnection": "Server=localhost;Port=3306;Database=kantor;Uid=root;Pwd=;SSL Mode=none;"

  },

  "Logging": {

    "LogLevel": {

      "Default": "Information",

      "Microsoft": "Warning",

      "Microsoft.Hosting.Lifetime": "Information"

    }

  },

  "AllowedHosts": "\*"

}

Lalu pada file startup.cs tambahkan method ConfigureServices()

public void ConfigureServices(IServiceCollection services)

        {

            services.AddControllers();

            services.Add(new ServiceDescriptor(typeof(Models.EmployeeContext), new Models.EmployeeContext(Configuration.GetConnectionString("DefaultConnection"))));

            services.AddSwaggerGen(c =>

            {

                c.SwaggerDoc("v1", new OpenApiInfo { Title = "Kantor\_WebAPI", Version = "v1" });

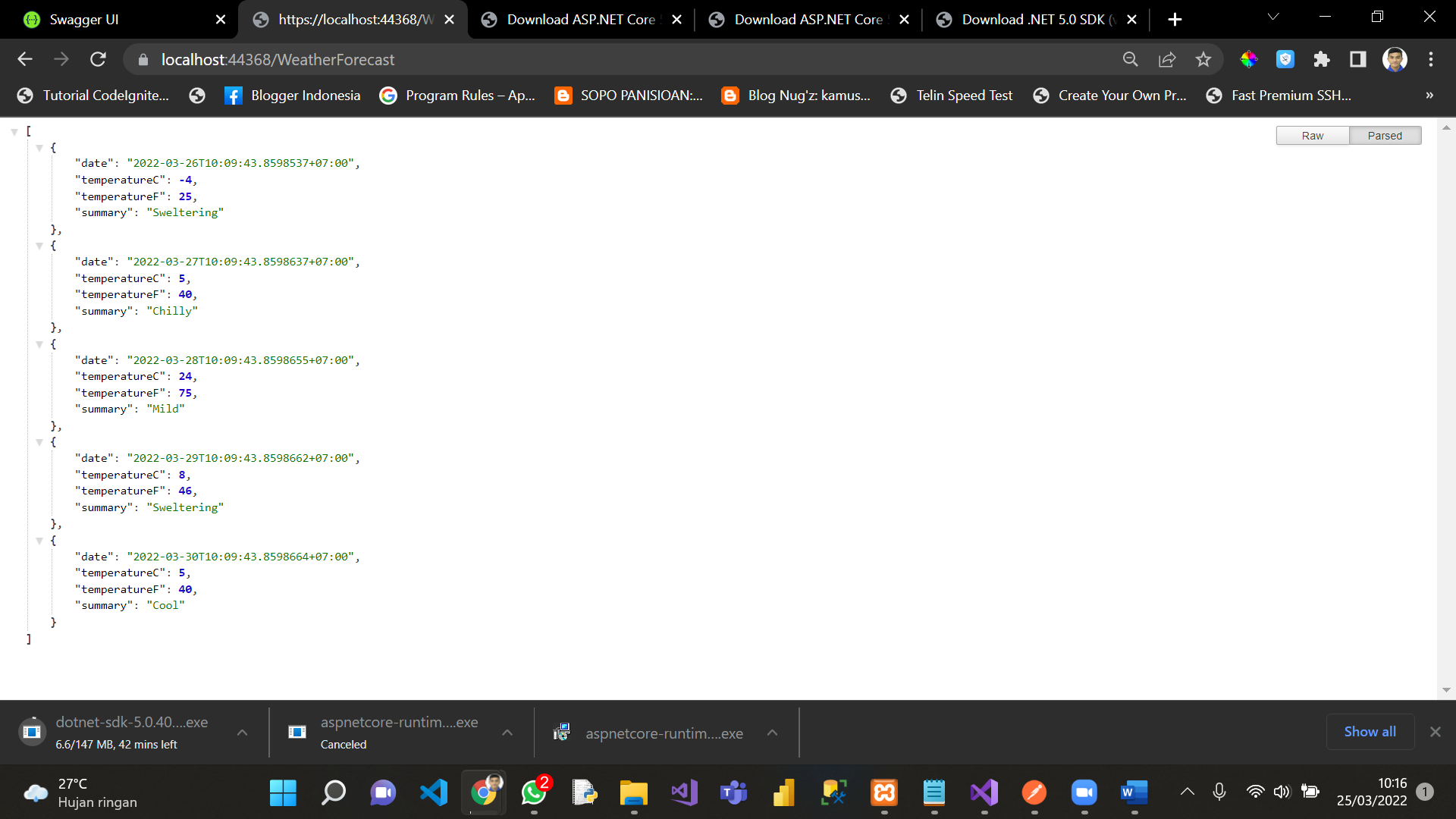
            });

        }

Lalu jalankan project

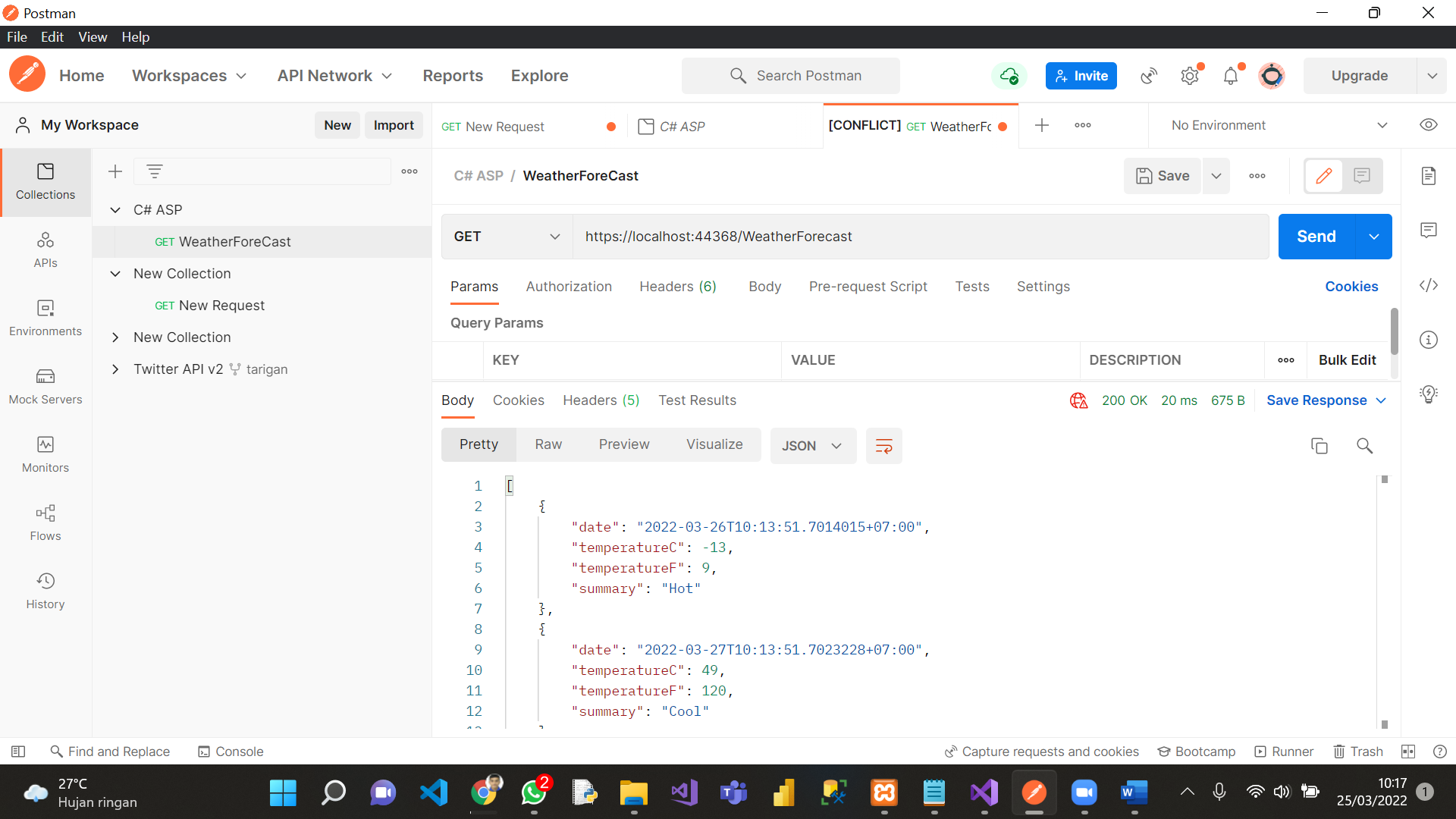
Swagger

Response json swagger memanggil data WeatherForeCast



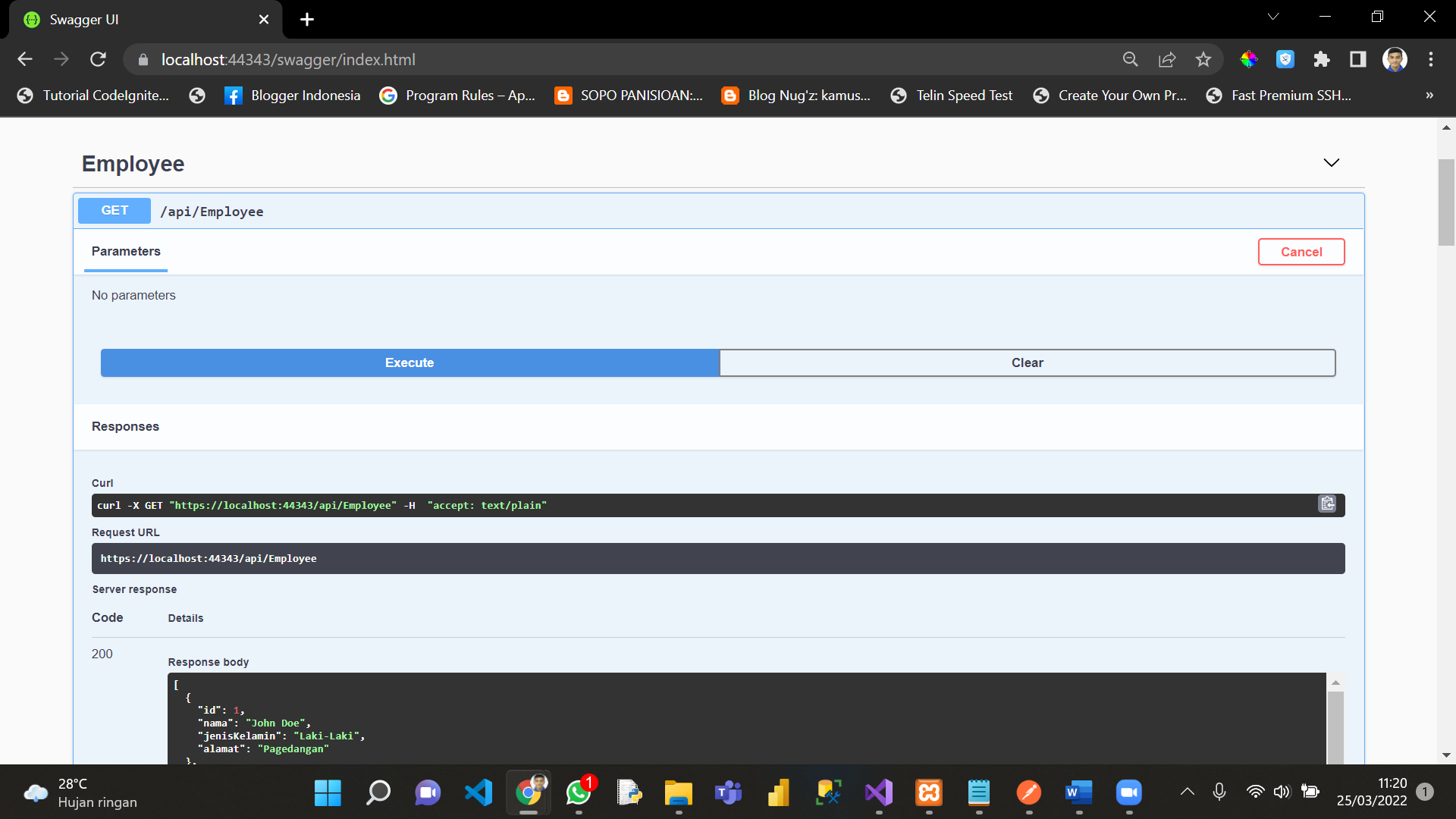
Postman

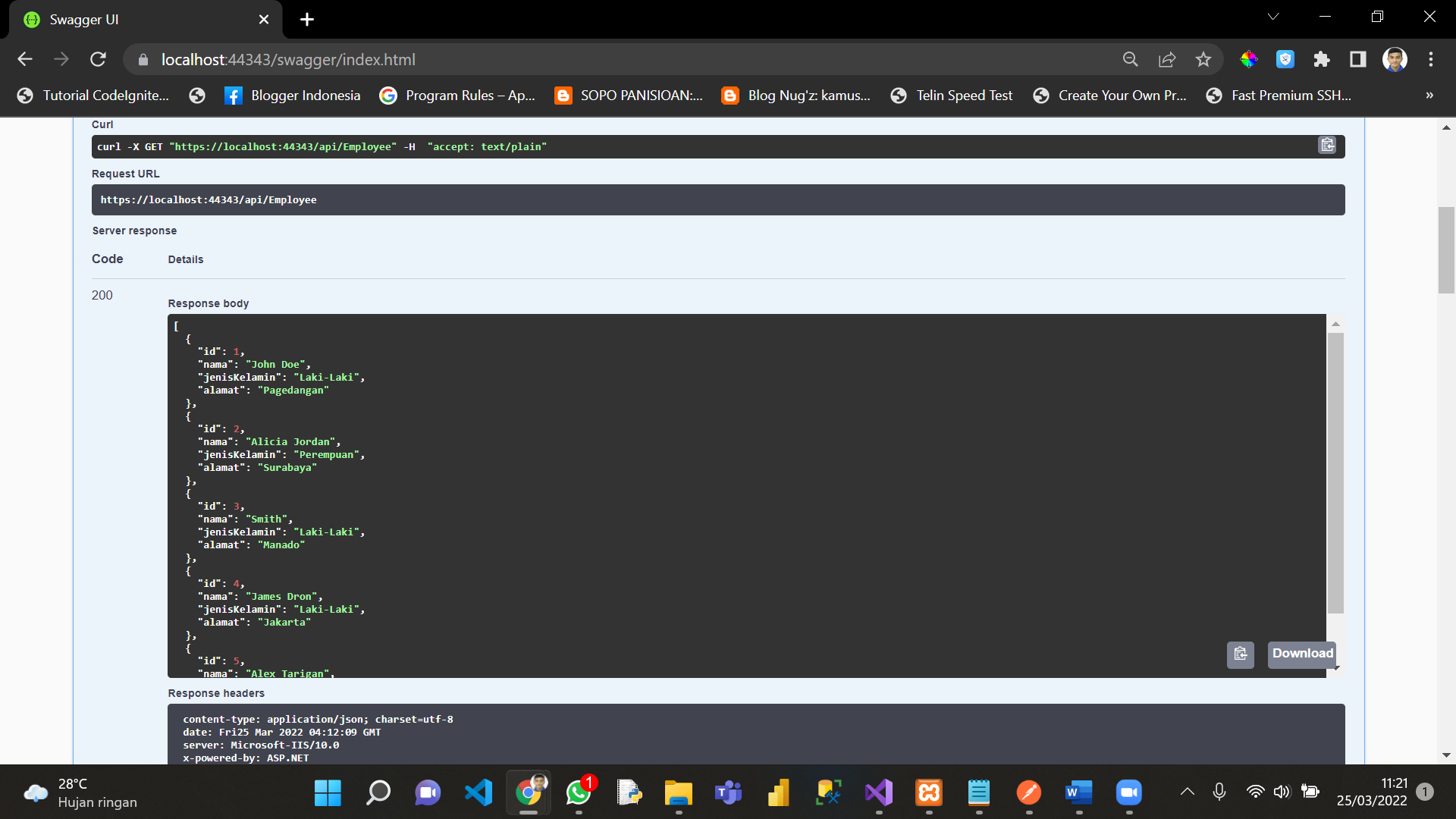
Postman Response json memanggil data WeatherForeCast



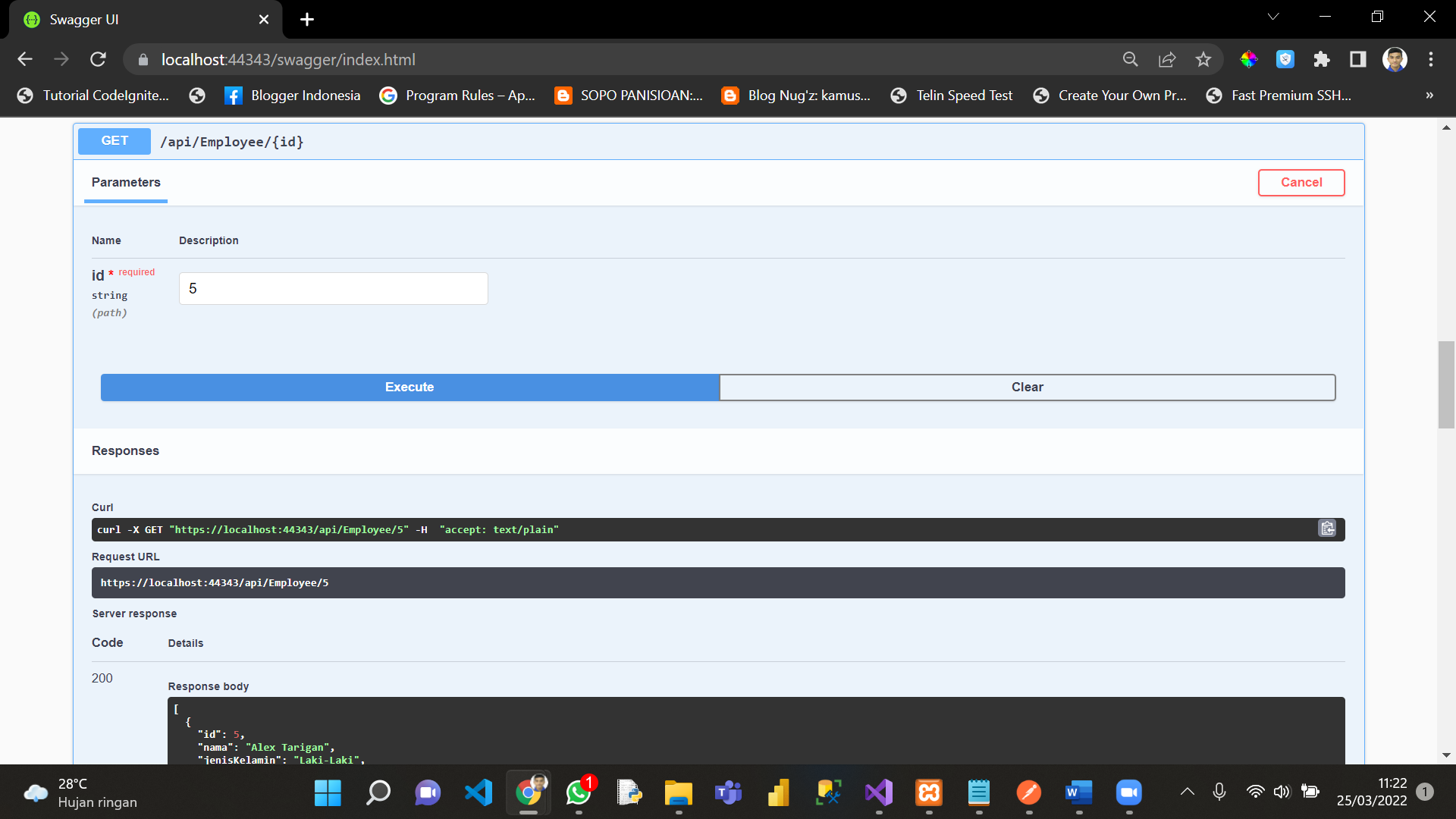
Kantor WebAPI

1. Response swagger memanggil semua data di table employee

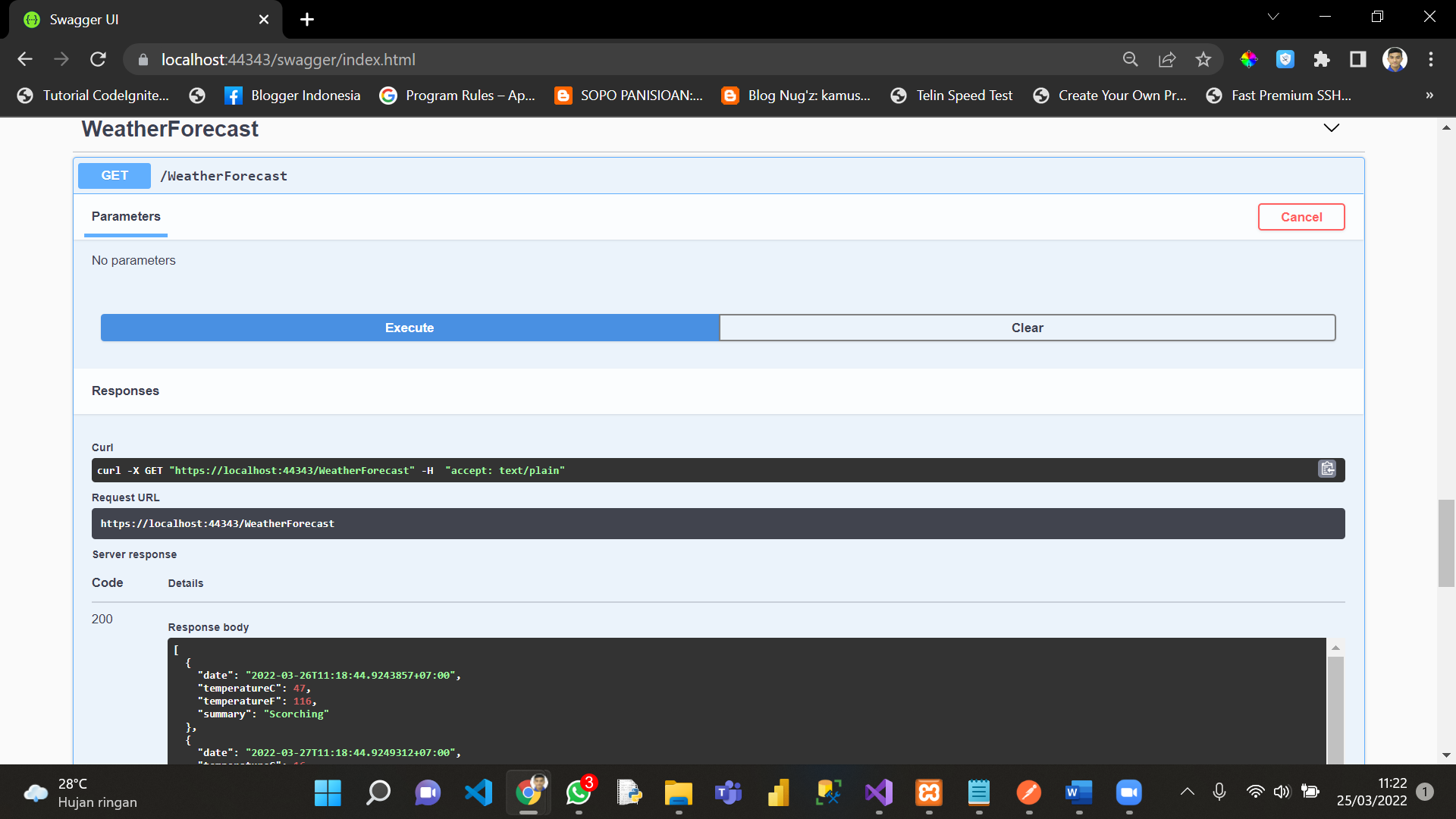




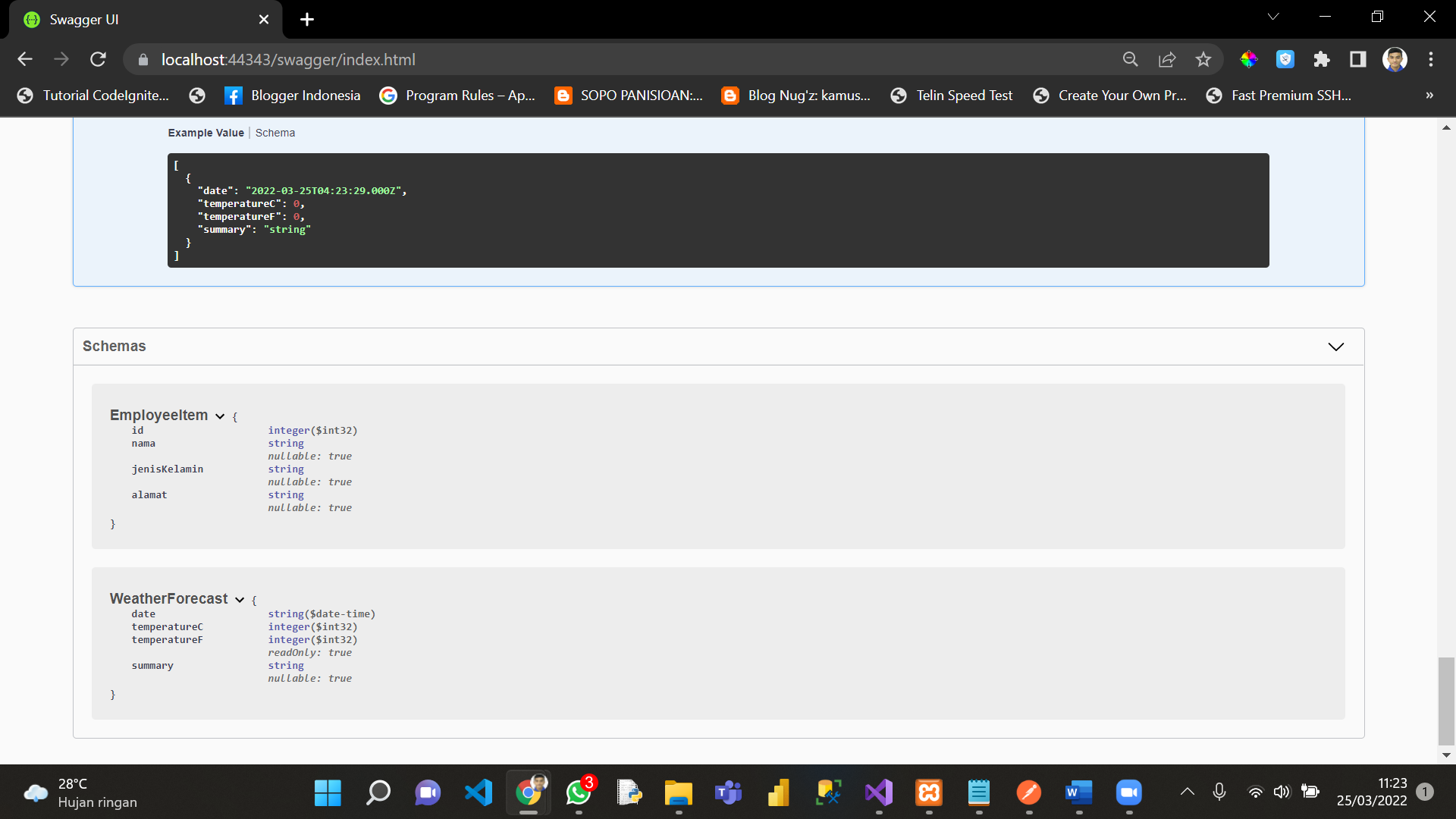
2. Response swagger memanggil semua data di table employee WHERE id = 5



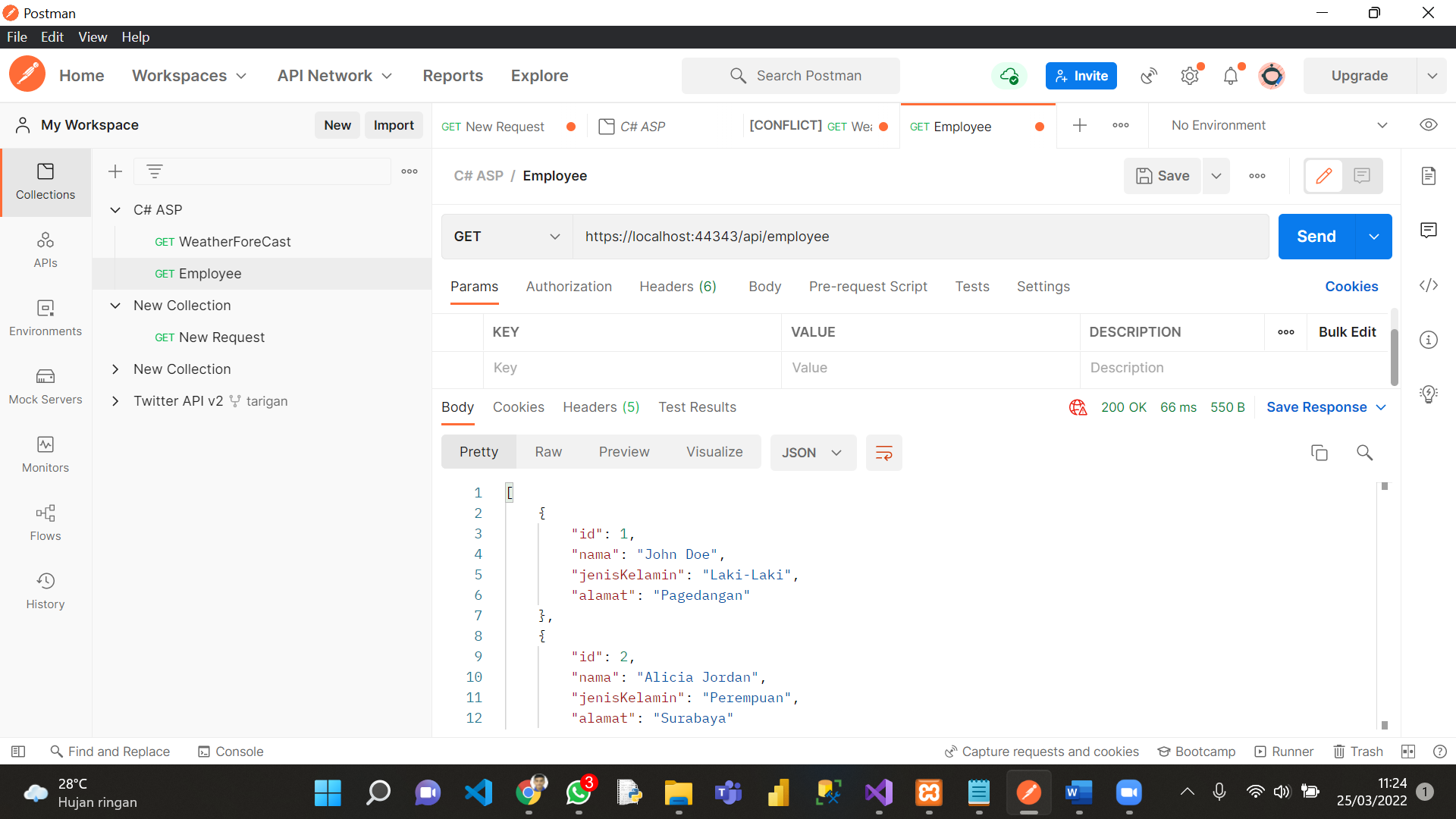
3. Response swagger memanggil data **WeatherForecast**



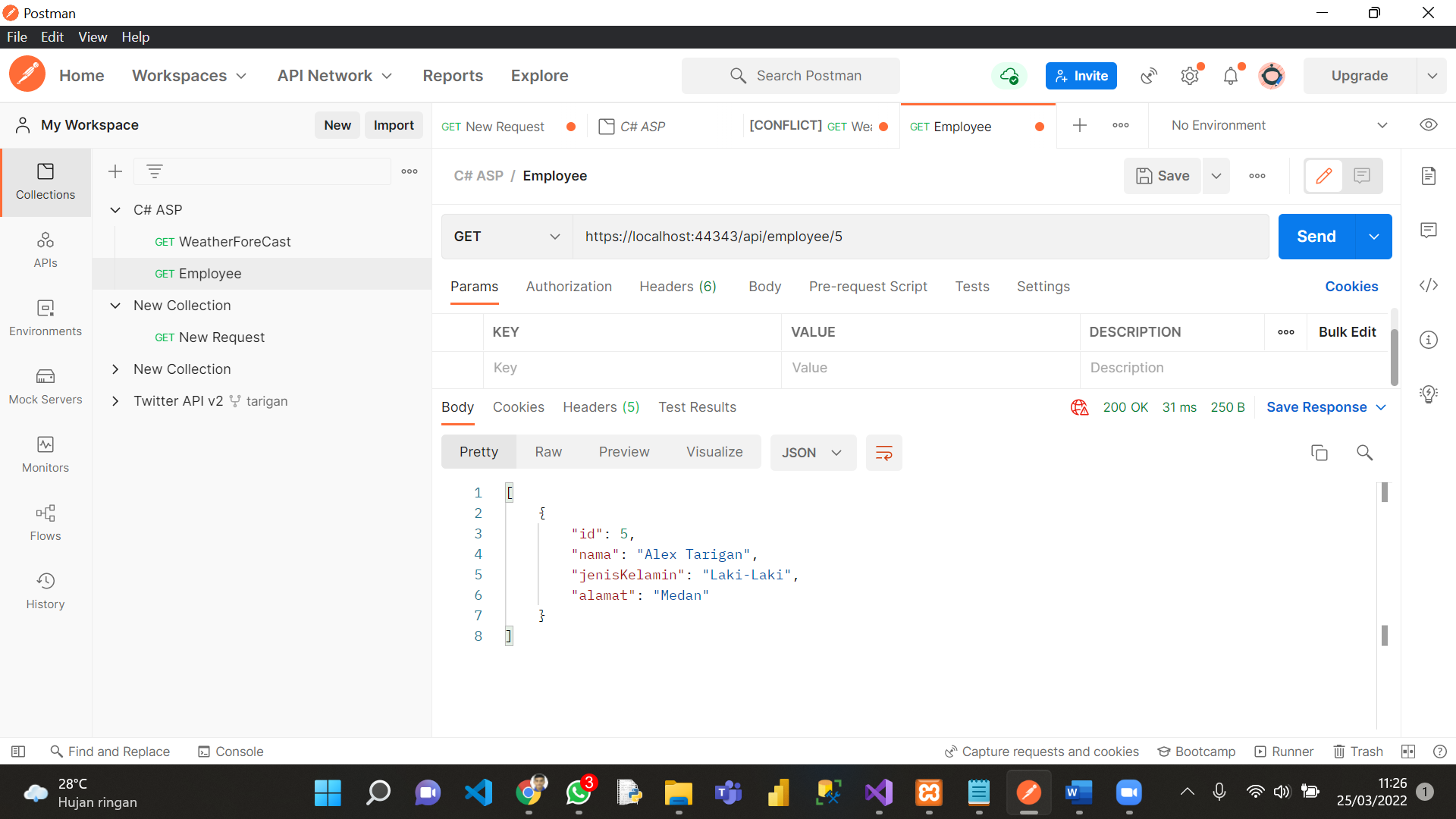
4. Response swagger memanggil schema



5. Response POSTMAN memanggil semua data di table employee



6. Response postman memanggil semua data di table employee WHERE id = 5



Response postman memanggil semua data di table employee WHERE id = 3

