

Nama: Kevin Elvio C. P

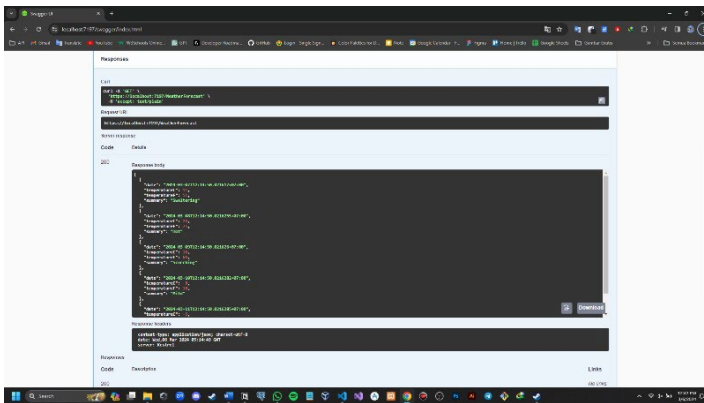
NIM: 222410102088

Link Github: <https://github.com/KevinElvio/Pemrograman-Antarmuka-Aplikasi.git>

Percobaan 1: Hello World API

Hasil:

- Berhasil menampilkan terminal untuk memonitoring aplikasi
- Berhasil menampilkan web browser data JSON Weather Forecast



Analisis: Dari percobaan 1, bisa dilihat bahwa untuk menampilkan sebuah data json di Swagger bisa menggunakan method get, yang dimana fungsi dari method ini digunakan untuk mengambil data, dalam kasus ini mengambil data dari Swagger.

Percobaan 2: Membuat model dan kontroler baru

Hasil :

- Berhasil membuat struktur project dengan menambahkan folder Models dan file Person.cs, PersonContext.cs, dan PersonController.cs.
- Berhasil menampilkan data person
- Berhasil menampilkan data person menggunakan POSTMAN

Analisis:

- Percobaan ini adalah percobaan yang membuat controller dan model baru.
- Model person mendefinisikan data
- Controller PersonController menghubungkan API untuk mendapatkan data Person.

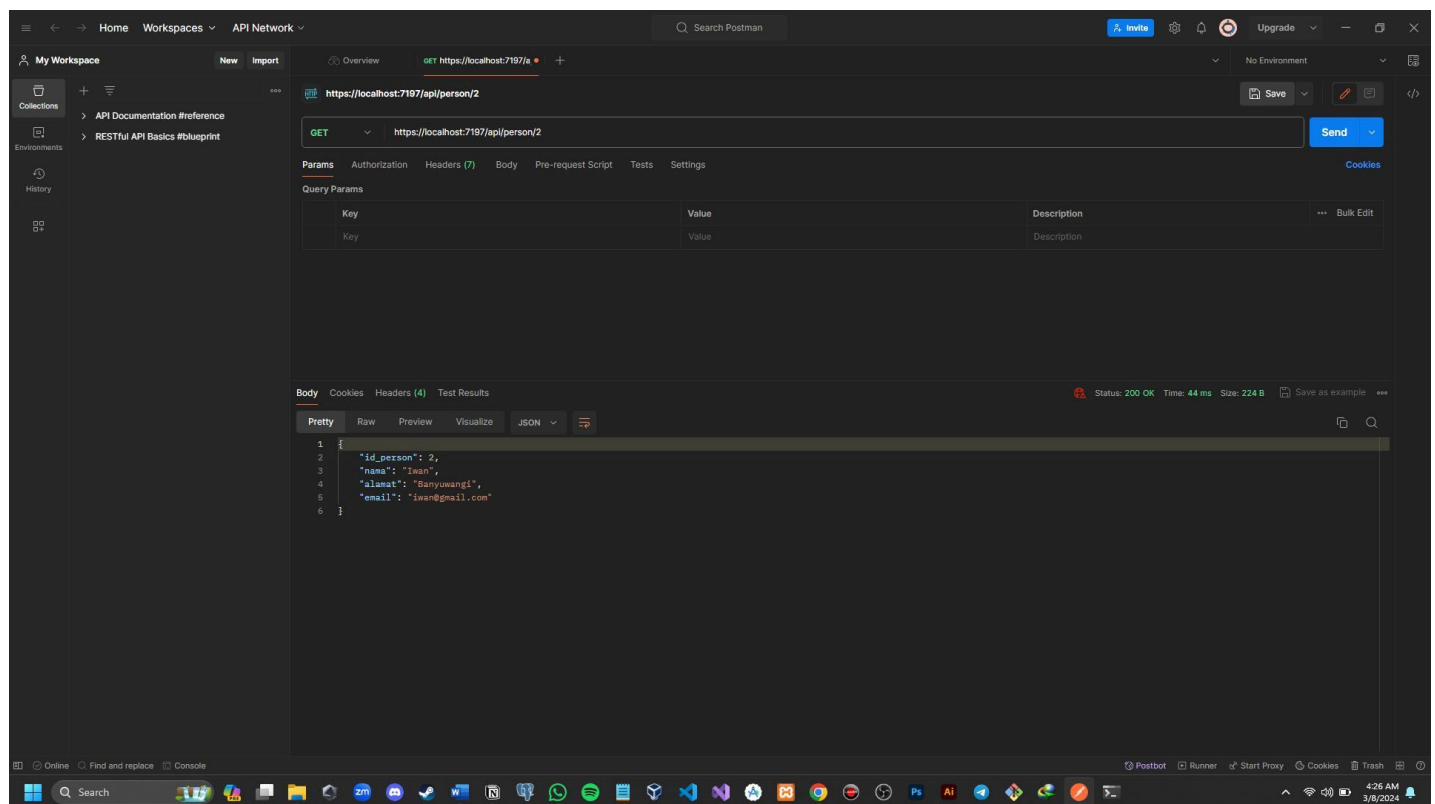
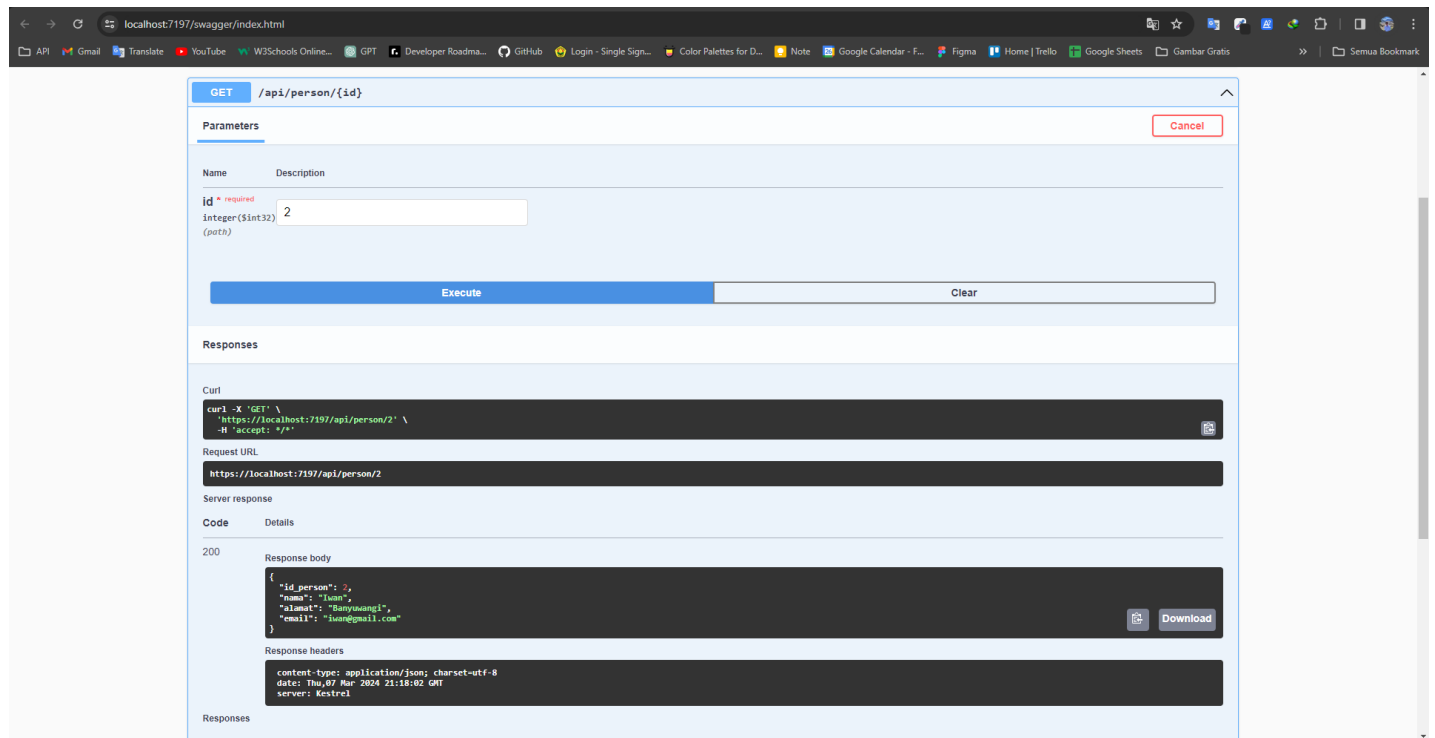
Kesimpulan:

Percobaan ini bertujuan untuk membuat API sederhana dengan ASP.NET menggunakan bahasa c#, belajar untuk membuat model, controller yang menghubungkan data dan bisa digunakan sesuai kebutuhan.

Tugas mandiri

Buatlah program untuk menampilkan satu data Person dengan menambahkan filter id_person dengan menggunakan method GET. Tampilkan hasilnya pada Swagger dan Postman.

Hasil:



Berhasil membuat program untuk menampilkan satu data person, dan menambahkan filter ide_person dengan menggunakan method GET, dan berhasil ditampilkan di Swagger dan Postman

Analisis:

Untuk membuat program yang menampilkan satu data person, harus terlebih dahulu membuat method seperti gambar di bawah ini

```
1 reference
public Person GetPerson(int id)
{
    var person = ListPerson().Find(x => x.id_person == id);
    return person;
}
```

Tambahkan method baru dengan nama GetPerson pada file PersonContext.cs dengan berisikan parameter “id” yang bertipe data integer, pada method ini kita mengambil data dari method ListPerson dan menggunakan fungsi find untuk mencari id.

```
[HttpGet("api/person/{id}")]
0 references
public IActionResult GetPerson(int id)
{
    PersonContext context = new PersonContext();
    Person id_person = context.GetPerson(id);

    if (id_person != null)
    {
        return Ok(id_person);
    }
    else
    {
        return NotFound();
    }
}
```

Selanjutnya buat routing seperti gambar di atas, menggunakan method get dengan path “api/person/{id}”

Pada method GetPerson memiliki parameter id yang bertipe data integer, di dalam method tersebut ada kode untuk membuat sebuah objek baru dan disimpan dengan variabel id_person, yang dimana memiliki percabangan jika id_person tidak sama dengan kosong maka akan mengembalikan status ok dan sebaliknya, jika id_person sama dengan kosong maka akan mengembalikan nilai tidak tersedia.