1. **Add New Users**

   Endpoint ini akan menambahkan user baru pada database berdasarkan data yang dikirim via request body json

   URL: /users/add
   Method: POST

   Example:

   Request Body:

```
{
        "email": "userdummy@gmail.com",
        “username”: ”dummykoboy”
}
```

   Response(Expected):

```
{
        "status": 200,
         "msg": "Success Add User",
         "data": null
}
```

2. **Get All Users**

   Endpoint ini mengambil seluruh data users pada db. Di bawah hanya mereturn 3 data user karna hanya contoh.

   URL: /users
   Method: GET

   Example:

   Response(Expected):

```
{
        "status": 200,
        "msg": "Success",
        "data": [
                {
                  "id": 0,
                  "email": "user0@gmail.com",
                  “username”: ”user123”
                  "isActive": false,
                },{
                  "id": 1,
                  "email": "user1@gmail.com",
                  “username”: ”user456
                  "isActive": true,
                },{
                  "id": 2,
                  "email": "user2@gmail.com",
                  “username”: ”user789”
                  "isActive": false,
                },
        ]
}
```

3. **Get User**

   Endpoint ini hanya mengambil 1 user berdasarkan id yang dikirim dalam pathparam **:userId**

   URL: /users/:userId
   Method: GET

   Example:
   Request URL: /users/1

   Response(Expected):

```
{
        "status": 200,
        "msg": "Success",
        "data": {
                  "id": 1,
                  "email": "user1@gmail.com",
                  “username”: ”user456
                  "isActive": true,
            }
}
```

4. **Update User Active Status**

Endpoint ini akan mengupdate active status dari user terpilih berdasarkan **userId** yang dikirim pada request body json.

URL: /users/update/active
Method: PUT

Example:

Request Body:

```
{
        id: 1,
        isActive: false
}
```

Response(Expected):

```
{
          "status": 200,
          "msg": "Success Update Status",
          "data": null
}
```

5. **Get All Users with Pagination (Optional)**
   Endpoint ini mengambil data users pada db berdasarkan limit dan page yang dikirim pada path param **:limit** dan **:page**

   URL: /users/:limit/:page
   Method: GET

   Example:
   Request URL: /users/5/1

   Response(Expected):

```
{
 "status": 200,
 "msg": "Success",
 "data": {
  "users": [
   {
    "id": 0,
    "email": "user0@gmail.com",
    "username": "user123",
    "isActive": false
   },
   {
    "id": 1,
    "email": "user0@gmail.com",
    "username": "user123",
    "isActive": false
   },
   {
    "id": 2,
    "email": "user0@gmail.com",
    "username": "user123",
    "isActive": false
   },
      {
        "id": 3,
        "email": "user0@gmail.com",
        "username": "user123",
        "isActive": false
      },
      {
        "id": 4,
        "email": "user0@gmail.com",
        "username": "user123",
        "isActive": false
      }
    ],
    "max_page": 5,
    "totalData": 25
  }
}
```

*Di atas merupakan bare minimum penilaian, jika mau menambahkan negatif case seperti respon ketika error/request tidak sesuai/data tidak ada. Semakin baik.