

Kode Peserta : FSDO001ONL017

Langkah – Langkah penggunaan nya adalah sebagai berikut :

1. Import db_final_project.sql kedalam database local yang ada di perangkat anda.
2. Aktifkan xampp agar aplikasi dapat mengakses database.
3. Buka folder FSD0001ONL017_BayuAjiNurmansah_FinalProject dengan menggunakan visual studio atau vs code.
4. Lalu ketikkan dotnet run.
5. Buka aplikasi pada browser atau postman dan ketikkan url <http://localhost:5000> atau <https://localhost:5000> jika menggunakan swagger ketikkan <https://localhost:5001/swagger/index.html>

Pada aplikasi ini terdapat 13 endpoint yang terbagi atas 3 endpoint pada controller AuthManagement, 5 endpoint pada controller OwnerDetails, dan 5 endpoint pada controller PaymentDetails

1. Endpoint AuthManagement Register

- Buka url <https://localhost:5001/api/AuthManagement/Register>
- Pastikan method yang digunakan adalah POST
- Lalu ketikkan data dengan format json untuk melakukan register. Contoh Formatnya adalah seperti berikut.

Example Value | Schema

Hal – hal yang perlu diperhatikan:

- Apabila username sudah ada di database maka sistem akan memberikan reponse, jika data dengan username tersebut sudah ada.
- Apabila email sudah ada di database maka sistem akan memberikan response jika data dengan email tersebut sudah ada.
- Apabila password kurang dari 6 character, tidak menggunakan minimal 1 huruf besar, tidak menggunakan minimal 1 symbol, dan tidak menggunakan 1 digit, maka sistem akan memberikan response untuk *user* melakukan pengetikan ulang pada password.

- a. Buka url <https://localhost:5001/api/AuthManagement/Login>

- b. Pastikan method yang digunakan adalah POST
- c. Lalu ketikkan data dengan format json untuk melakukan login. Contoh Formatnya adalah seperti berikut.

Example Value	Schema
	<pre>{ "username": "string", "password": "string" }</pre>

- d. Apabila login sukses akan memberikan response seperti berikut. Lalu salin token nya untuk digunakan sebagai Authentication.

Response body

```
{
  "result": {
    "token":
      "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZCIGImQWYtNDIwTEZlY2I0NTBjOTg0ODFhbnVtZWlsIjoiaWF5MDF5bDUBbnVmbWpCb2o1LkZjczdWIoOjI1YXNjOGdtYWlsImVubSIsImp0eSI6ImNDYSZDEwLTNjMjgtNDU3NC1iMGUzLTQxNTcxOWZmZm11ZiIsIm5iZiI6MTkyZDg4NTEucMcwIzXWhIjoXNjM0ODg4NzEwLkZpYXQiojE2MzQ4ODUxMTB9.8zdPIIClbblkqN-4TzaK9uvvSEs7312Vx2EXmxFWog",
    "refreshToken": "AEPJH7XRKIF6477XIHM3090Y0YDW7N1MSAdse2931e1-6664-4023-8edf-6db098cf147a",
    "success": true,
    "errors": null
  },
  "id": 1,
  "exception": null,
  "status": 5,
  "isCanceled": false,
  "isCompleted": true,
  "isCompletedSuccessfully": true,
  "creationOptions": @,
  "asyncState": null,
  "isFaulted": false
}
```

- Buka url <https://localhost:5001/api/AuthManagement/RefreshToken>

- b. Pastikan method yang digunakan adalah POST

c. Lalu ketikkan data dengan format json untuk melakukan login. Contoh Formatnya adalah seperti berikut.

Example Value | Schema

```
{
  "token": "string",
  "refreshToken": "string"
}
```

d. Apabila token dan refresh token belum expired maka akan diberikan response seperti berikut

Response body

```
{
  "token": null,
  "refreshToken": null,
  "success": false,
  "errors": [
    "Token has not yet expired"
  ]
}
```

B. Endpoint pada Controller OwnerDetails [Improve]

Ini merupakan table improve yang saya gunakan agar dapat menghandle insert atau put data yang nanti akan dilakukan pada controller PaymentDetails. Dengan **skema** seperti berikut.

- Apabila insert data / update data dilakukan pada table PaymentDetails tetapi card number belum tersedia pada table OwnerDetails maka data akan gagal di input
- Apabila insert data / update data dilakukan pada table PaymentDetails tetapi ownerName untuk cardNumber tertentu tidak sesuai maka data akan gagal di input. Contoh :
 - o cardNumber 1 milik owner Bayu, cardNumber 2 milik owner Ubay, apabila insert data pada table PaymentDetails menggunakan cardNumber 1 tetapi owner name nya adalah Ubay maka data akan gagal di input
- Apabila insert data / update data dilakukan pada table PaymentDetails tetapi field status pada table OwnerDetails adalah inactive maka data akan gagal di input.

Hal ini penting untuk menjaga integritas data yang ada pada table PaymentDetails.

Catatan Penting: Hal yang perlu diperhatikan agar dapat melakukan CRUD (Create, Read, Update dan Delete) pada table OwnerDetails adalah kita harus memasukkan token yang didapat dari proses login pada bagian headers (**Postman**) pada key isikan Authorization dan value nya isi dengan Bearer *Token*, atau pada **Swagger** dengan melakukan klik pada button Authorize dan pastekan token yang didapat dari proses login.

4. Endpoint OwnerDetails Get all owner details data

- a. Buka url [https://localhost:5001 /api/OwnerDetails](https://localhost:5001/api/OwnerDetails)
- b. Pastikan method yang digunakan adalah GET
- c. Apabila request berhasil maka akan ditampilkan data sebagai berikut

Response body

```
[
  {
    "ownerId": 1,
    "ownerCardNumber": "1234-5678-9-0",
    "ownerName": "Bayu Aji Nurmansah",
    "ownerAddress": "Medan",
    "status": "active",
    "createdDate": "2021-10-21T16:27:57.553"
  },
  {
    "ownerId": 2,
    "ownerCardNumber": "1234-5678-9-1",
    "ownerName": "Jonh Snow",
    "ownerAddress": "Winterfell",
    "status": "active",
    "createdDate": "2021-10-21T16:27:57.553"
  },
  {
    "ownerId": 4,
    "ownerCardNumber": "1234-5678-9-2",
    "ownerName": "Robert Baratheon",
    "ownerAddress": "King's Landing",
    "status": "active",
    "createdDate": "2021-10-21T16:27:57.553"
  }
]
```

5. Endpoint OwnerDetails Insert owners data

- a. Buka url [https://localhost:5001 /api/OwnerDetails](https://localhost:5001/api/OwnerDetails)
- b. Pastikan method yang digunakan adalah POST
- c. Lalu ketikkan data dengan format json untuk melakukan insert data. Contoh Formatnya adalah seperti berikut.

Example Value | Schema

```
{
  "ownerId": 0,
  "ownerCardNumber": "string",
  "ownerName": "string",
  "ownerAddress": "string",
  "status": "string",
  "createdAt": "2021-10-22T07:06:14.701Z"
}
```

Hal – hal yang perlu diperhatikan:

- Apabila ownerCardNumber sudah terdapat di dalam database maka akan memberikan response jika ownerCardNumber sudah ada. Maka, ownerCardNumber harus diganti dengan data yang belum terdapat didalam database.

6. Endpoint OwnerDetails Get data by id

- a. Buka url [https://localhost:5001 /api/OwnerDetails/id](https://localhost:5001/api/OwnerDetails/id)
- b. Pastikan method yang digunakan adalah Get
- c. Ganti id yang ada di URL dengan id data yang ingin anda lihat
- d. Apabila request berhasil maka akan ditampilkan data sebagai berikut:

Response body

```
{
  "ownerId": 2,
  "ownerCardNumber": "1234-5678-9-1",
  "ownerName": "Jonn Snow",
  "ownerAddress": "Winterfell",
  "status": "active",
  "createdAt": "2021-10-21T16:27:57.553"
}
```

7. Endpoint OwnerDetails Update Owners Data

- a. Buka url [https://localhost:5001 /api/OwnerDetails/id](https://localhost:5001/api/OwnerDetails/id)
- b. Pastikan method yang digunakan adalah Put
- c. Ganti id yang terdapat di url dengan id data yg ingin anda ubah
- d. Lalu ketikkan data dengan format json untuk melakukan update data. Contoh Formatnya adalah seperti berikut.

Example Value | Schema

```
{
  "ownerId": 0,
  "ownerCardNumber": "string",
  "ownerName": "string",
  "ownerAddress": "string",
  "status": "string",
  "createdAt": "2021-10-22T07:38:27.363Z"
}
```

Hal – hal yang perlu diperhatikan:

- Apabila user mencoba mengupdate ownerCardNumber tetapi ownerCardNumber sudah terdapat pada table ownerDetails maka server akan memberikan reponse jika data tidak dapat di update dan harus mengganti dengan ownerCardNumber yang belum terdapat di database

8. Endpoint OwnerDetails Delete Owners Data

- a. Buka url [https://localhost:5001 /api/OwnerDetails/id](https://localhost:5001/api/OwnerDetails/id)
- b. Pastikan method yang digunakan adalah Delete
- c. Ganti id yang terdapat di url dengan id data yang ingin anda hapus
- d. Apabila request berhasil server akan memberikan response 200 dengan message data berhasil dihapus, akan tetapi apabila id yang akan di delete tidak tersedia maka akan memberikan response bad request

C. Endpoint pada Controller PaymentDetails

Ada beberapa skema yang akan digunakan pada proses POST dan PUT didalam table payment details, untuk lebih jelasnya dapat kembali ke point **B. Endpoint Pada Controller OwnerDetails**.

Catatan Penting: Hal yang perlu diperhatikan agar dapat melakukan CRUD (Create, Read, Update dan Delete) pada table OwnerDetails adalah kita harus memasukkan token yang didapat dari proses login pada bagian headers (**Postman**) pada key isikan Authorization dan value nya isi dengan Bearer *Token*, atau pada **Swagger** dengan melakukan klik pada button Authorize dan pastekan token yang didapat dari proses login.

9. Endpoint PaymentDetails Get All Data

- a. Buka url [https://localhost:5001 /api/PaymentDetails](https://localhost:5001/api/PaymentDetails)
- b. Pastikan method yang digunakan adalah GET

c. Apabila request berhasil maka akan ditampilkan data sebagai berikut

Response body

```
[
  {
    "paymentDetailId": 1,
    "cardOwnerName": "Bayu Aji Nurmansah",
    "cardNumber": "1234-5678-9-0",
    "expirationDate": "2021-10-21T16:30:45.697",
    "securityCode": "1234",
    "totalTransactions": 150500
  }
]
```

10. Endpoint PaymentDetails Insert Data

- a. Buka url <https://localhost:5001 /api/PaymentDetails>
- b. Pastikan method yang digunakan adalah POST
- c. Lalu ketikkan data dengan format json untuk melakukan insert data. Contoh Formatnya adalah seperti berikut.

Example Value | Schema

```
{
  "paymentDetailId": 0,
  "cardOwnerName": "string",
  "cardNumber": "string",
  "expirationDate": "2021-10-24T04:54:41.508Z",
  "securityCode": "string",
  "totalTransactions": 0
}
```

d. Apabila data berhasil insert maka akan muncul tampilan seperti dibawah ini

Response body

```
{
  "contentType": "application/json",
  "serializerSettings": null,
  "statusCode": 200,
  "value": "Data has been insert"
}
```

Terdapat beberapa persyaratan agar data dapat di input kedalam database, untuk lebih jelasnya silahkan Kembali pada point **B. Endpoint Pada Controller OwnerDetails.**

11. Endpoint PaymentDetails Get Data By Id

- a. Buka url <https://localhost:5001/api/PaymentDetails/id>
- b. Pastikan method yang digunakan adalah Get
- c. Ganti id yang ada di URL dengan id data yang ingin anda lihat
- d. Apabila request berhasil maka akan ditampilkan data sebagai berikut:

Response body

```
{
  "paymentDetailId": 7,
  "cardOwnerName": "Jonn Snow",
  "cardNumber": "1234-5678-9-1",
  "expirationDate": "2021-10-24T04:51:50.784",
  "securityCode": "123456",
  "totalTransactions": 30000
}
```

12. Endpoint PaymentDetails Update Data

- a. Buka url <https://localhost:5001/api/PaymentDetails/id>
- b. Pastikan method yang digunakan adalah Put
- c. Ganti id yang terdapat di url dengan id data yg ingin anda ubah
- d. Lalu ketikkan data dengan format json untuk melakukan update. Contoh Formatnya adalah seperti berikut.

Example Value | Schema

```
{
  "paymentDetailId": 0,
  "cardOwnerName": "string",
  "cardNumber": "string",
  "expirationDate": "2021-10-24T05:04:11.560Z",
  "securityCode": "string",
  "totalTransactions": 0
}
```

13. Endpoint PaymentDetails Delete Data

- a. Buka url <https://localhost:5001/api/PaymentDetails/id>
- b. Pastikan method yang digunakan adalah Delete
- c. Ganti id yang terdapat di url dengan id data yang ingin anda hapus
- d. Apabila request berhasil server akan memberikan response 200 dengan message data berhasil dihapus, akan tetapi apabila id yang akan di delete tidak tersedia maka akan memberikan response bad request