

Nama : Dini Tri Widianingsih

No. Peserta : FSDO003ONL007

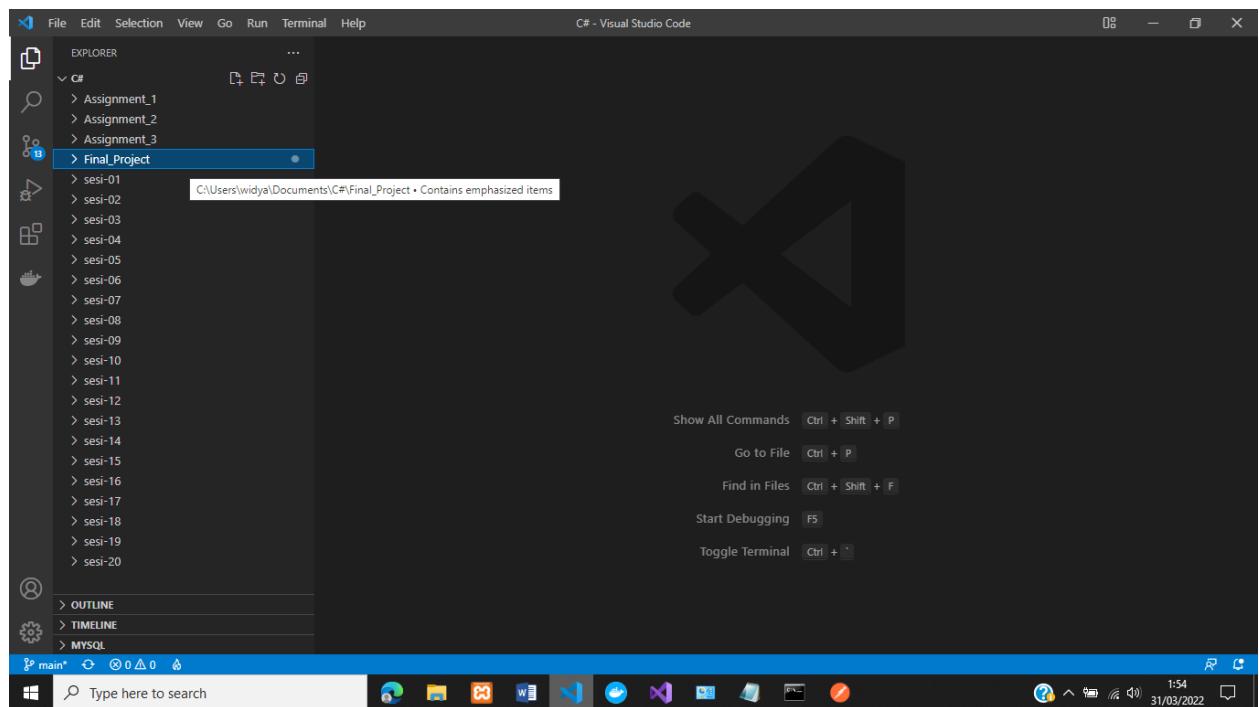
Link Github : [dinni26/DiniFinalProject_PaymentAPI \(github.com\)](https://github.com/dinni26/DiniFinalProject_PaymentAPI)

Link Heroku : [Biodata \(dini-finalproject.herokuapp.com\)](https://biodata.dini-finalproject.herokuapp.com)

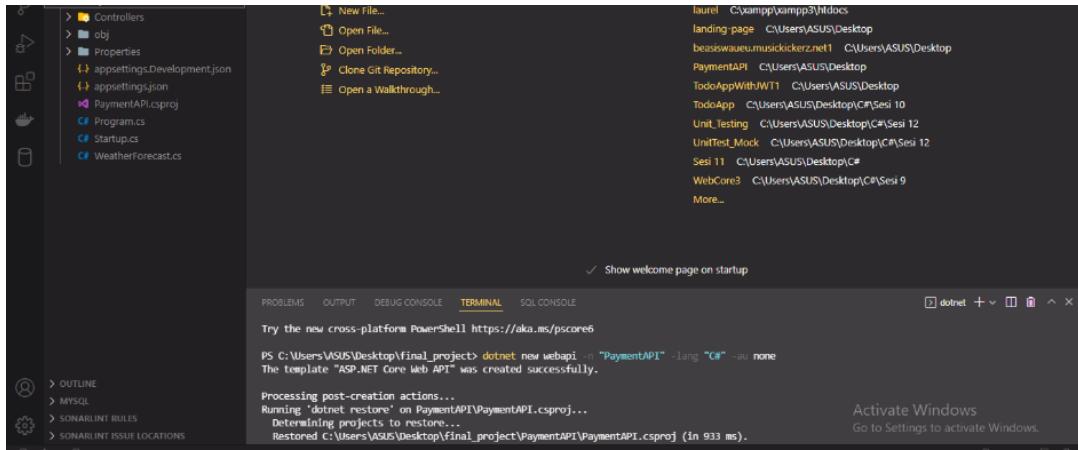
PANDUAN PEMBUATAN FINAL PROJECT

PEMBUATAN

- Buka Visual Studio Code > Create folder Final_Poject

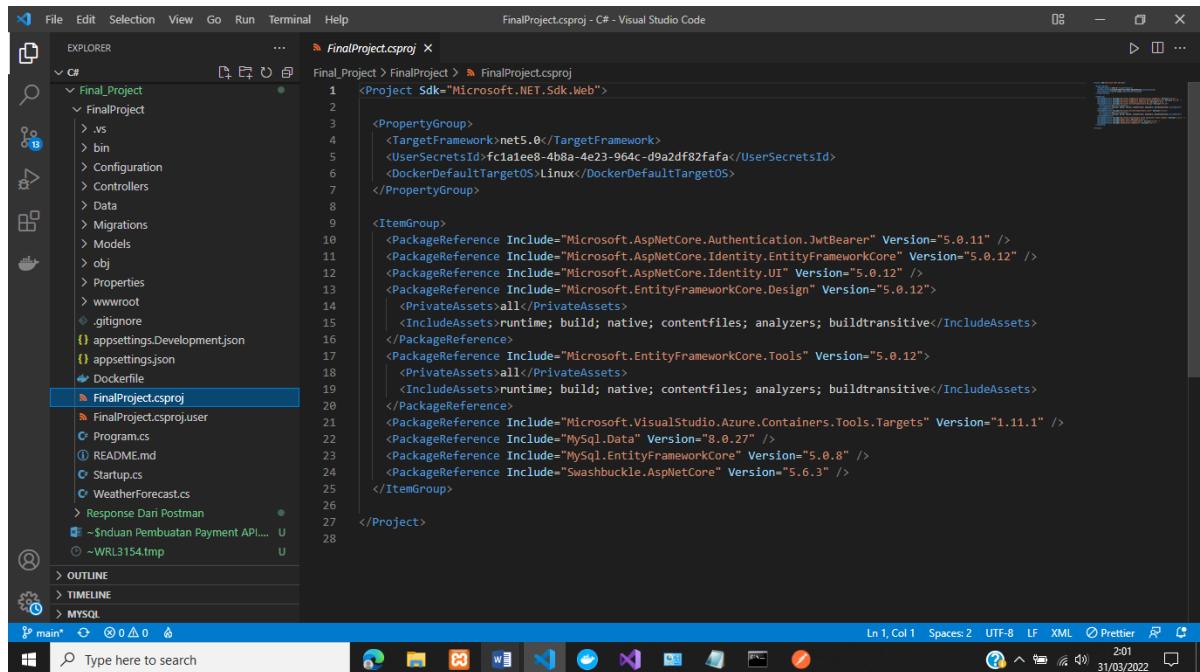


- Inisiasi project di dalam folder tersebut > **dotnet new webapi -n "Final_Project" -lang "C#" -au none** lalu akan mucul folder – folder baru



- 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

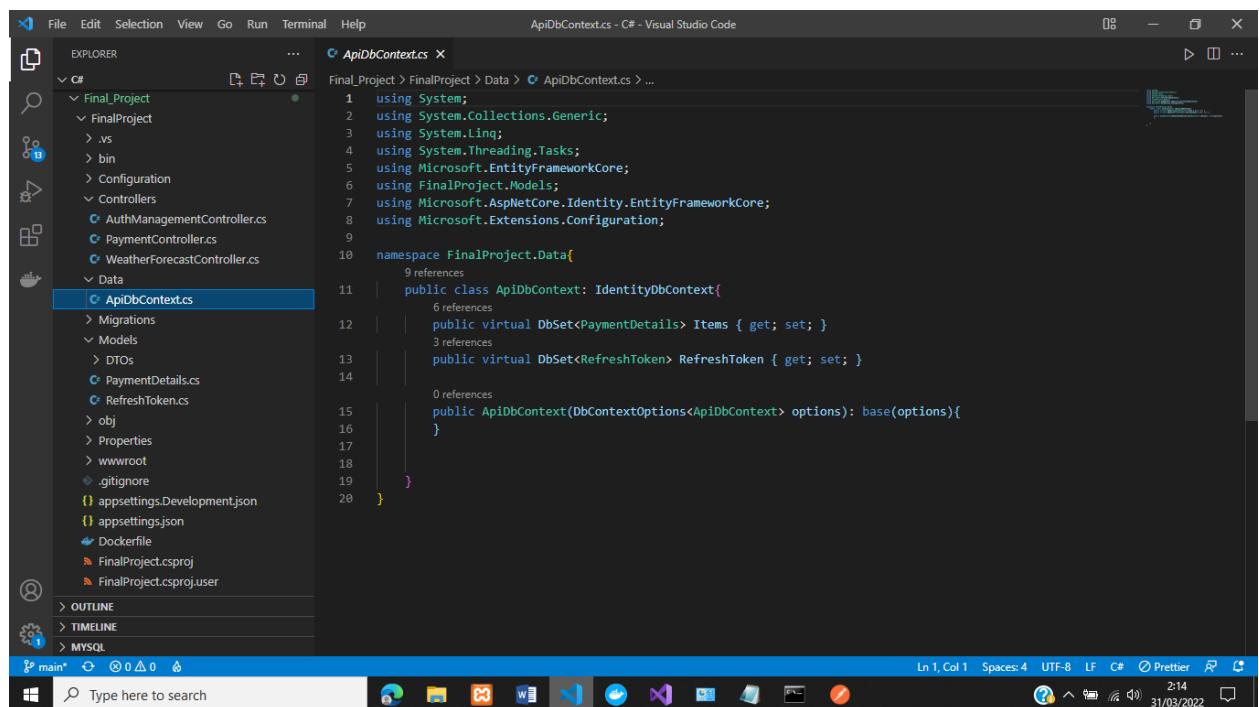


- Buat folder Models, lalu tambahkan file PaymentDetails.cs di dalam folder tersebut
Location : Final_Project/Models/PaymentDetails.cs

Final_Project > FinalProject > Models > **C PaymentDetails.cs** > {} FinalProject.Models

```
6
7     namespace FinalProject.Models{
8         3 references
9         public class PaymentDetails{
10             [Key]
11             2 references
12             public int PaymentDetailId { get; set; }
13             1 reference
14             public string CardOwnerName { get; set; }
15             1 reference
16             public string CardNumber { get; set; }
17             1 reference
18             public string ExpirationDate { get; set; }
19             1 reference
20             public string SecurityCode { get; set; }
21         }
22     }
```

- 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

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "Final_Project". The "Controllers" folder is expanded, and "PaymentController.cs" is selected.
- Code Editor:** Displays the code for "PaymentController.cs". The code defines a controller class "PaymentController" that inherits from "ControllerBase". It includes methods for GET and POST operations, both of which interact with an "ApiDbContext" context to handle data items.
- Bottom Status Bar:** Shows information such as "Ln 1, Col 1", "Spaces: 4", "UTF-8", "LF", "C#", "Prettier", and the date and time "31/03/2022 2:15".

```
PaymentController.cs - C# - Visual Studio Code

namespace FinalProject.Controllers;
[Route("api/[controller]")]
[ApiController]
[Authorize(AuthenticationSchemes = JwtBearerDefaults.AuthenticationScheme)]
public class PaymentController : ControllerBase
{
    private readonly ApiDbContext _context;

    public PaymentController(ApiDbContext context)
    {
        this._context = context;
    }

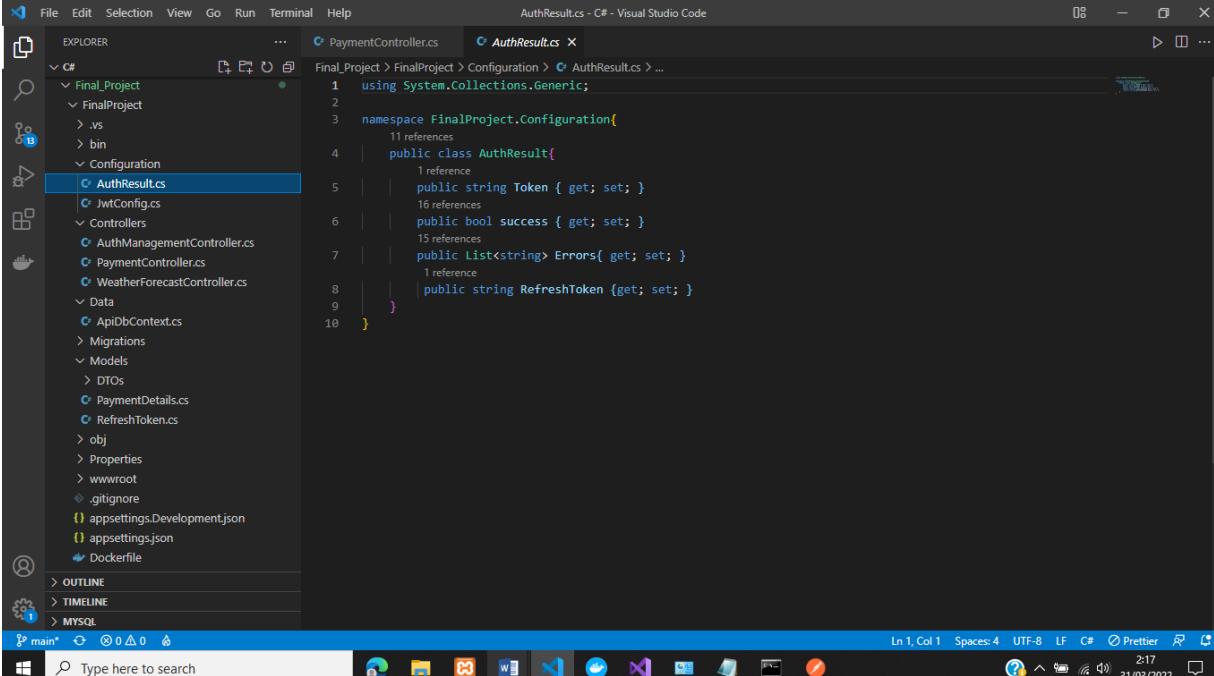
    [HttpGet]
    public async Task<IActionResult> GetItem()
    {
        var items = await _context.Items.ToListAsync();
        return Ok(items);
    }

    [HttpPost]
    public async Task<IActionResult> CreateItem(PaymentDetails data)
    {
        if(ModelState.IsValid)
        {
            await _context.Items.AddAsync(data);
            await _context.SaveChangesAsync();

            return CreatedAtAction("GetItem", new {data.PaymentDetailId}, data);
        }
    }
}
```

Penambahan JWT Token dan Refresh Token

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



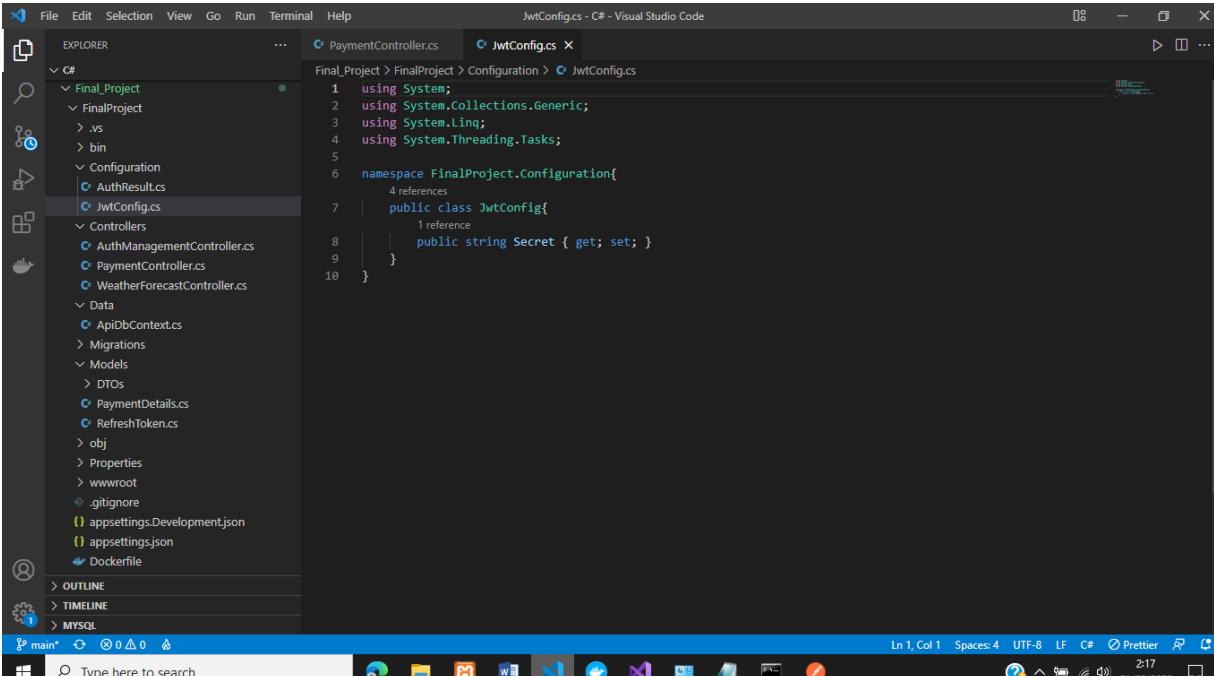
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "Final_Project". The "Configuration" folder is expanded, showing "AuthResult.cs" and "JwtConfig.cs".
- Code Editor:** The file "AuthResult.cs" is open, displaying the following C# code:

```
1 using System.Collections.Generic;
2
3 namespace FinalProject.Configuration{
4     public class AuthResult{
5         public string Token { get; set; }
6         public bool success { get; set; }
7         public List<string> Errors{ get; set; }
8         public string RefreshToken { get; set; }
9     }
10 }
```

The status bar at the bottom indicates "Ln 1, Col 1" and "217 31/03/2022".

Location : Final_Project/Configuration/JwtConfig.cs



The screenshot shows the Visual Studio Code interface with the following details:

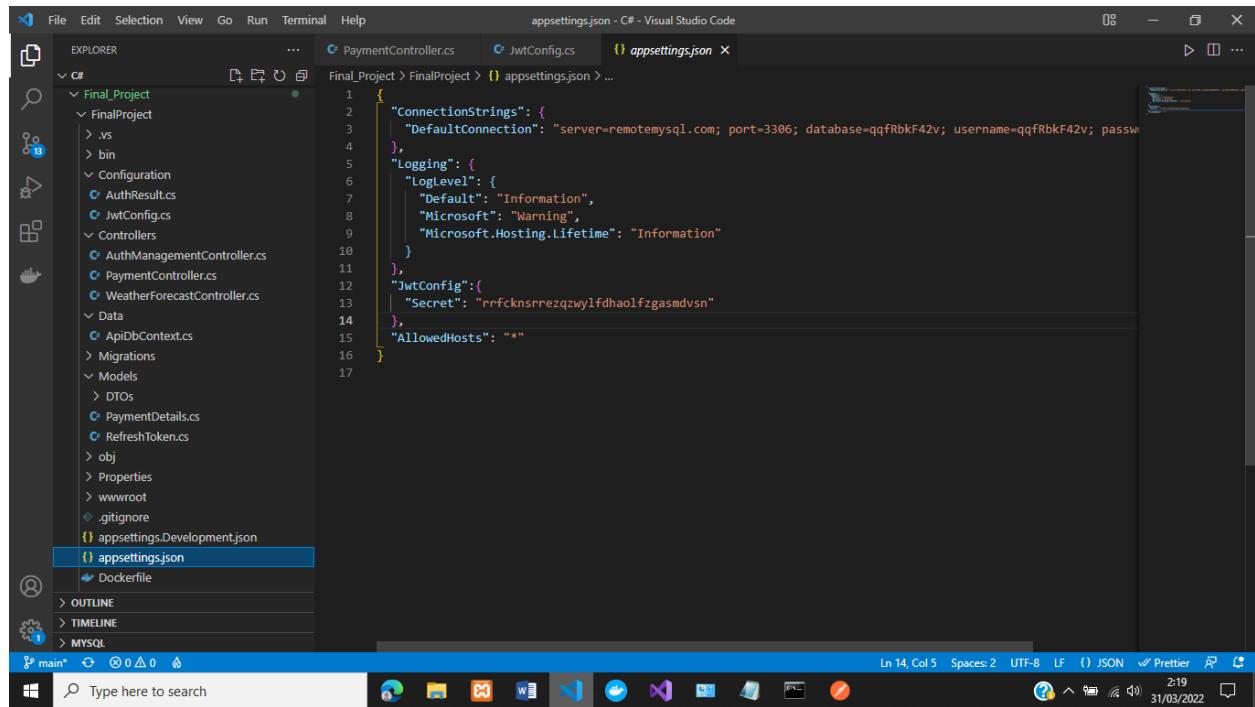
- File Explorer:** Shows the project structure under "Final_Project". The "Configuration" folder is expanded, showing "AuthResult.cs" and "JwtConfig.cs".
- Code Editor:** The file "JwtConfig.cs" is open, displaying the following C# code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5
6 namespace FinalProject.Configuration{
7     public class JwtConfig{
8         public string Secret { get; set; }
9     }
10 }
```

The status bar at the bottom indicates "Ln 1, Col 1" and "217 31/03/2022".

- Pada file appsettings.json tambahkan JwtConfig.Secrets

Location : Final_Project/appsettings.json



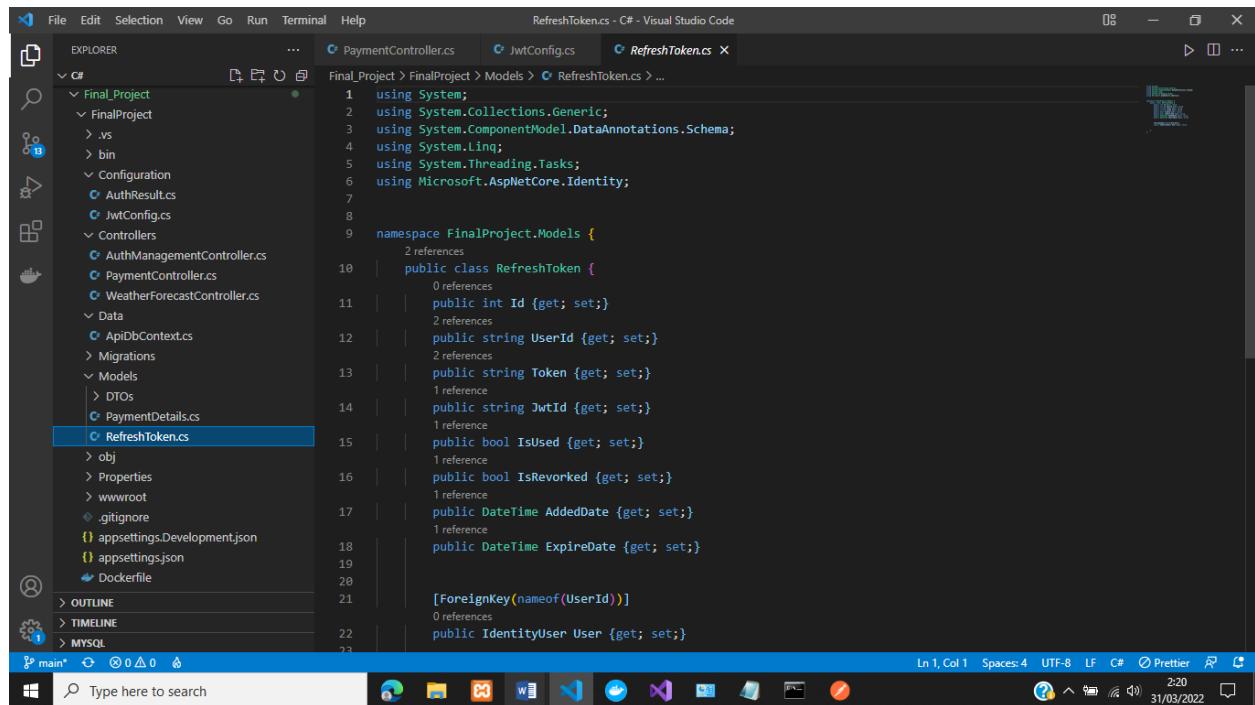
```

{
  "ConnectionStrings": {
    "DefaultConnection": "server=remotemysql.com; port=3306; database=qqrRbkF42v; username=qqrRbkF42v; password=qqrRbkF42v"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "JwtConfig": {
    "Secret": "rrfcKnsrrezqzwylfdhaolfgzgasmdvsn"
  },
  "AllowedHosts": "*"
}

```

- Buat file baru RefreshToken.cs yang terdapat pada folder Models

Location : Final_Project/Models/RefreshToken.cs



```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Identity;

namespace FinalProject.Models {
  public class RefreshToken {
    public int Id { get; set; }
    public string UserId { get; set; }
    public string Token { get; set; }
    public string JwtId { get; set; }
    public bool IsUsed { get; set; }
    public bool IsRevoked { get; set; }
    public DateTime AddedDate { get; set; }
    public DateTime ExpireDate { get; set; }

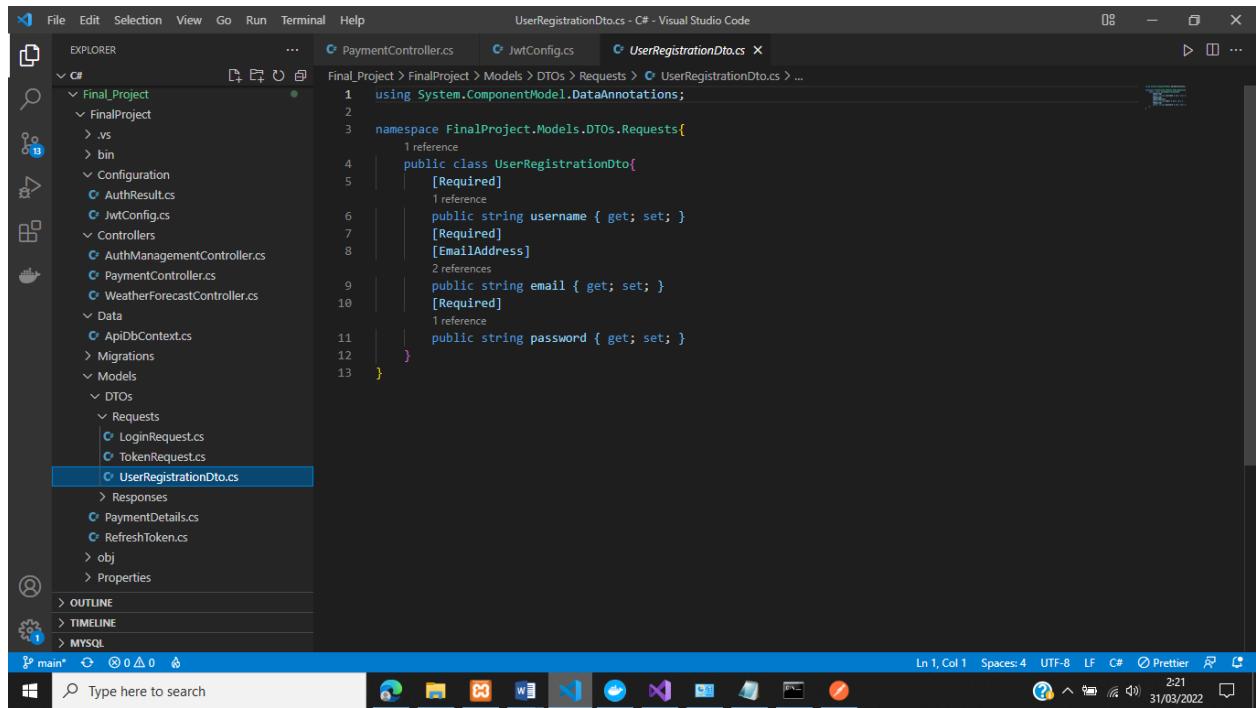
    [ForeignKey(nameof(UserId))]
    public IdentityUser User { get; set; }
  }
}

```

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

- **Folder Request**

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

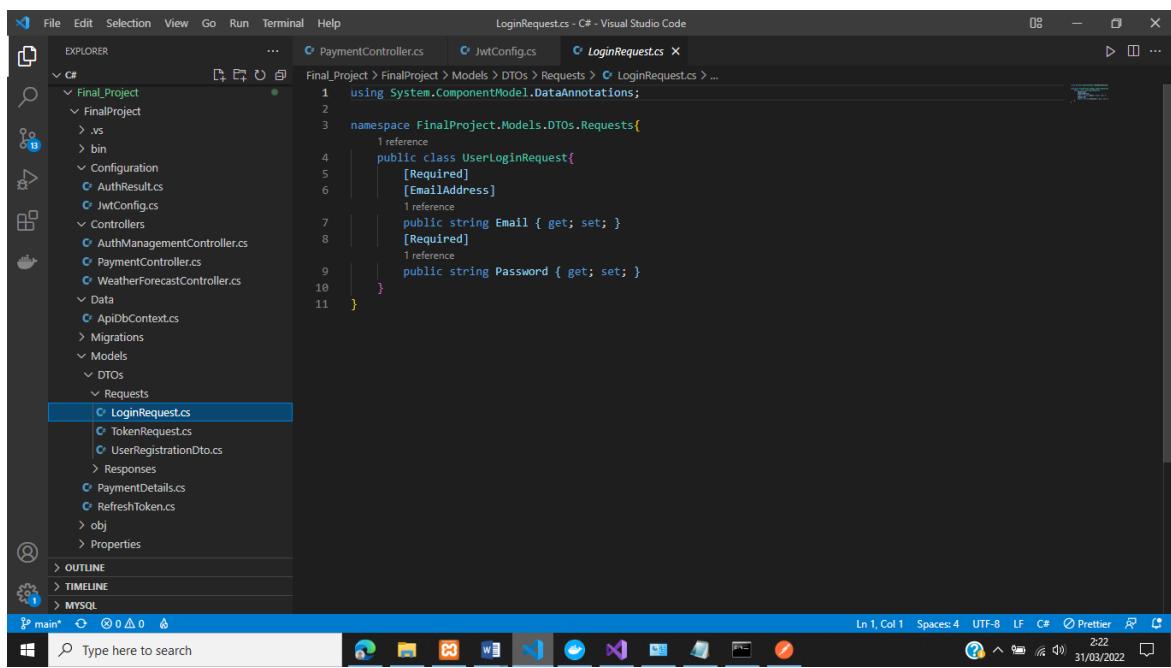


```

File Edit Selection View Go Run Terminal Help UserRegistrationDto.cs - C# - Visual Studio Code
EXPLORER Final_Project > FinalProject > Models > DTOs > Requests > UserRegistrationDto.cs > ...
Final_Project > FinalProject > Models > DTOs > Requests > UserRegistrationDto.cs > ...
1 using System.ComponentModel.DataAnnotations;
2
3 namespace FinalProject.Models.DTOs.Requests{
4     public class UserRegistrationDto{
5         [Required]
6             public string username { get; set; }
7             [Required]
8                 [EmailAddress]
9                     public string email { get; set; }
10                    [Required]
11                         public string password { get; set; }
12
13 }

```

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

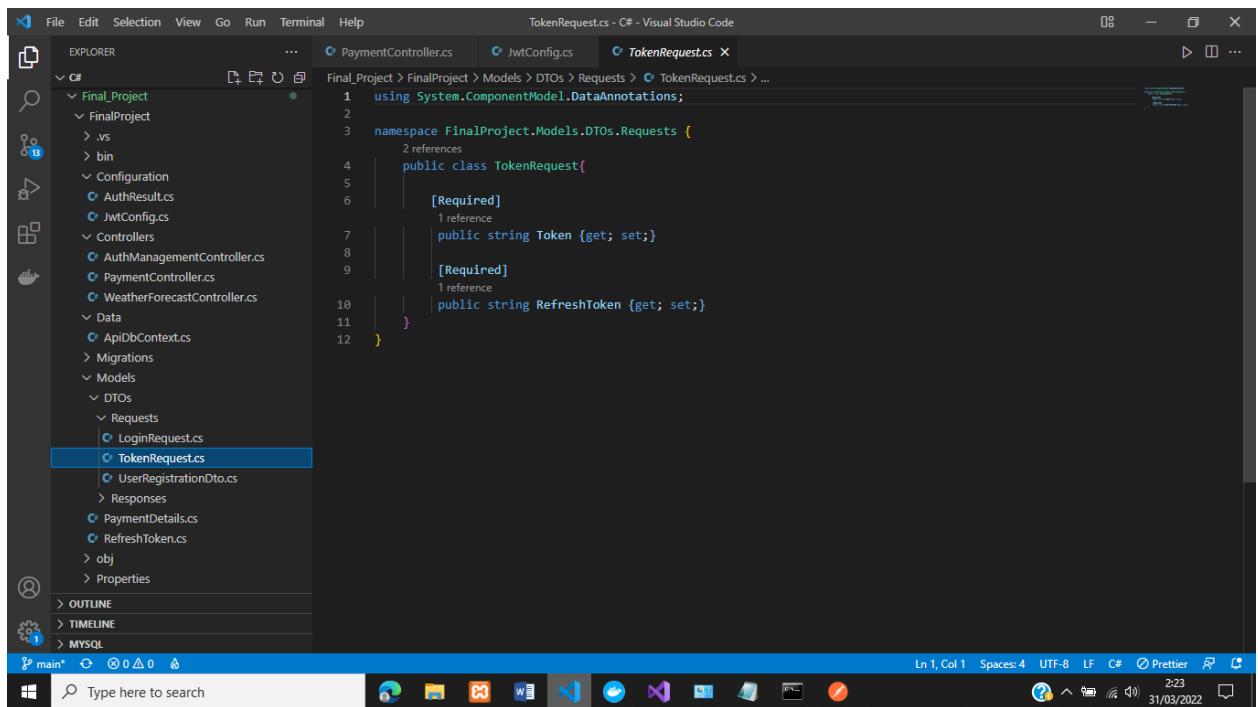


```

File Edit Selection View Go Run Terminal Help LoginRequest.cs - C# - Visual Studio Code
EXPLORER Final_Project > FinalProject > Models > DTOs > Requests > LoginRequest.cs > ...
Final_Project > FinalProject > Models > DTOs > Requests > LoginRequest.cs > ...
1 using System.ComponentModel.DataAnnotations;
2
3 namespace FinalProject.Models.DTOs.Requests{
4     public class UserLoginRequest{
5         [Required]
6             [EmailAddress]
7                     public string Email { get; set; }
8             [Required]
9                     public string Password { get; set; }
10
11 }

```

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



The screenshot shows the Visual Studio Code interface with the following details:

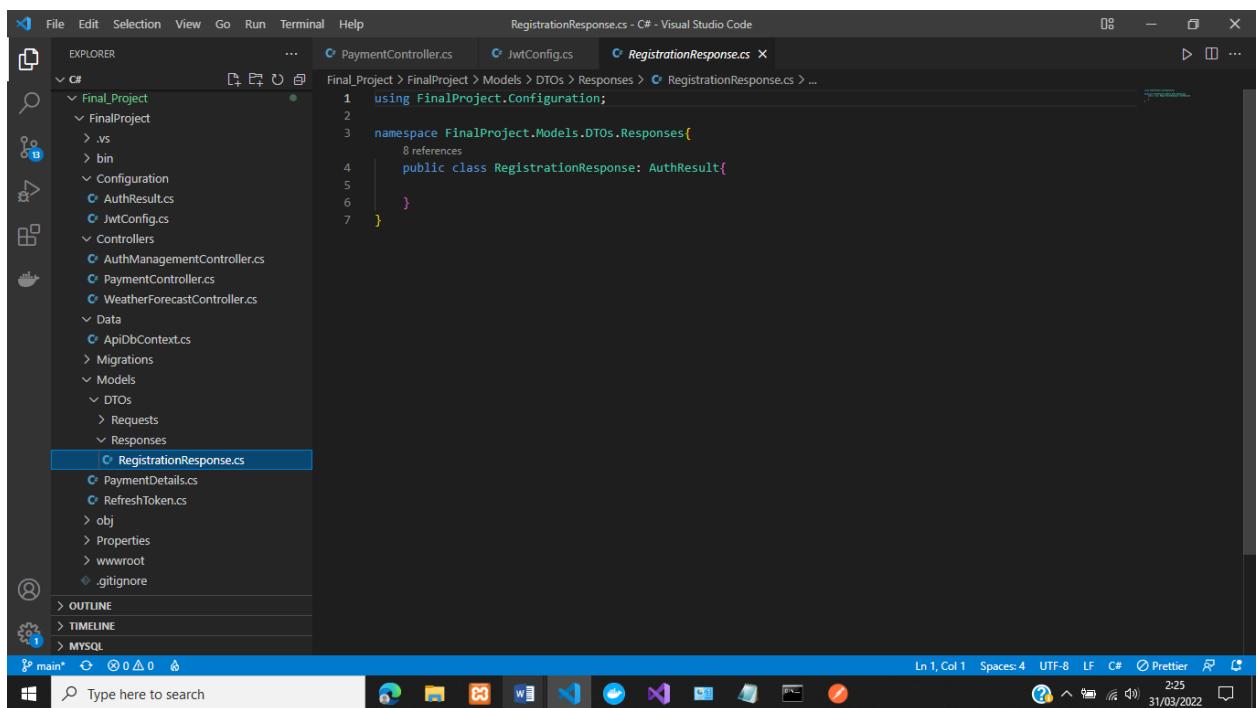
- File Path:** Final_Project/FinalProject/Models/DTOs/Requests/TokenRequest.cs
- Code Content:**

```
1 using System.ComponentModel.DataAnnotations;
2
3 namespace FinalProject.Models.DTOs.Requests {
4     public class TokenRequest{
5         [Required]
6             public string Token {get; set;}
7
8         [Required]
9             public string RefreshToken {get; set;}
10    }
11 }
```

- Explorer View:** Shows the project structure with the current file selected in the Requests folder.
- Bottom Bar:** Includes icons for file operations, search, and various extensions like Prettier.

- **Folder Response**

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



The screenshot shows the Visual Studio Code interface with the following details:

- File Path:** Final_Project/FinalProject/Models/DTOs/Responses/RegistrationResponse.cs
- Code Content:**

```
1 using FinalProject.Configuration;
2
3 namespace FinalProject.Models.DTOs.Responses{
4     public class RegistrationResponse: AuthResult{
5     }
6 }
```

- Explorer View:** Shows the project structure with the current file selected in the Responses folder.
- Bottom Bar:** Includes icons for file operations, search, and various extensions like Prettier.

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

```
File Edit Selection View Go Run Terminal Help AuthManagementController.cs - C# - Visual Studio Code

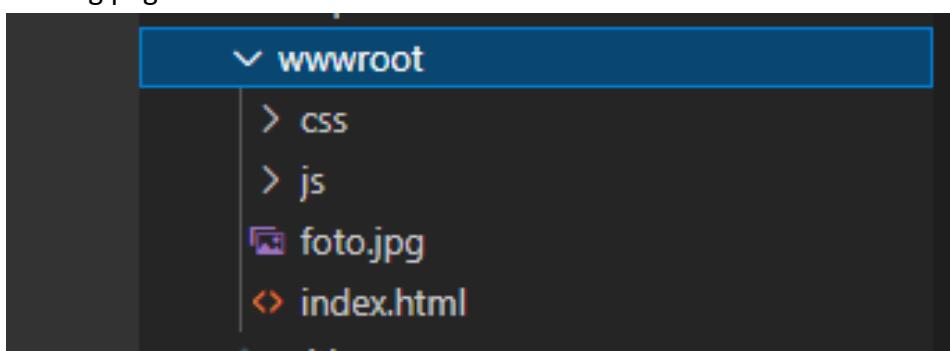
EXPLORER PaymentController.cs JwtConfig.cs AuthManagementController.cs
Final_Project FinalProject > Controllers AuthManagementController.cs > () FinalProject.Controllers > FinalProject.Controllers.AuthManagementController > Gener...
Final_Project vs bin Configuration AuthResult.cs JwtConfig.cs Controllers AuthManagementController.cs PaymentController.cs WeatherForecastController.cs Data ApiDbContext.cs Migrations Models DTOs Requests Responses RegistrationResponse.cs PaymentDetails.cs RefreshToken.cs obj Properties wwwroot .gitignore
OUTLINE TIMELINE MYSQL

Ln 91 Col 56 Spaces: 4 UTF-8 LF C# Prettier
main* Type here to search
PaymentController.cs
JwtConfig.cs
AuthManagementController.cs
1 using System;
2 using System.Collections.Generic;
3 using System.IdentityModel.Tokens.Jwt;
4 using System.Linq;
5 using System.Security.Claims;
6 using System.Text;
7 using System.Threading.Tasks;
8 using Microsoft.AspNetCore.Identity;
9 using Microsoft.AspNetCore.Mvc;
10 using Microsoft.EntityFrameworkCore;
11 using Microsoft.Extensions.Options;
12 using Microsoft.IdentityModel.Tokens;
13 using FinalProject.Configuration;
14 using FinalProject.Data;
15 using FinalProject.Models;
16 using FinalProject.Models.DTOS.Requests;
17 using FinalProject.Models.DTOS.Responses;
18
19 namespace FinalProject.Controllers{
20
21     [Route("api/[controller]")]
22     [ApiController]
23     public class AuthManagementController: ControllerBase{
24         private readonly UserManager<IdentityUser> _userManager;
25         private readonly JwtConfig _jwtConfig;
26         private readonly TokenValidationParameters _tokenValidationParams;
27         private readonly ApiDbContext _apiDbContext;
28
29 }
```

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

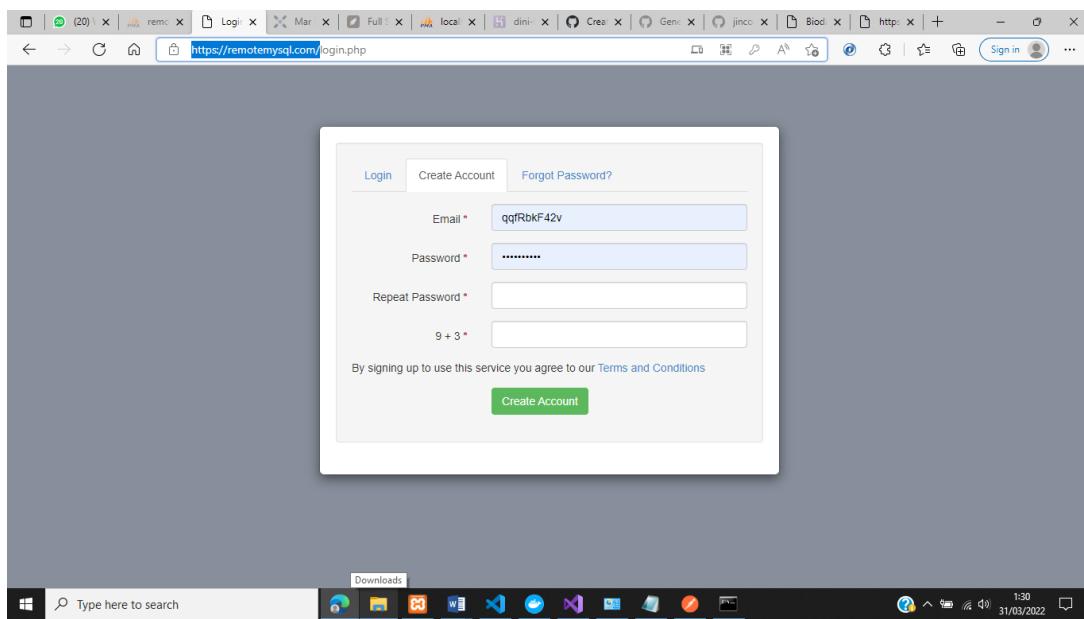
```
        },
        Expires = DateTime.UtcNow.AddMinutes(30),
        SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithm),
    };
}
```

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

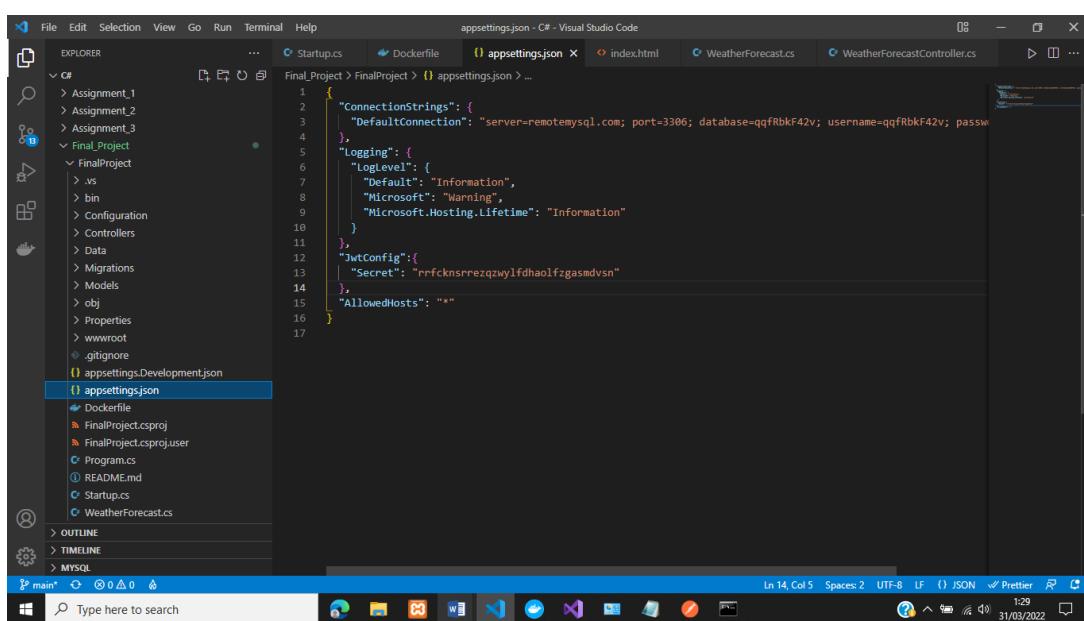


DATABASE

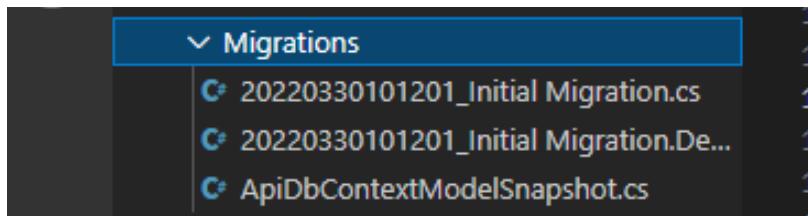
- Buat Akun di <https://remotemysql.com/>



- 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 appsettings.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

```

Dockerfile - C# - Visual Studio Code
File Edit Selection View Go Run Terminal Help Dockerfile - C# - Visual Studio Code
EXPLORER Dockerfile appsettings.json index.html WeatherForecast.cs WeatherForecastController.cs
Assignment_1
Assignment_2
Assignment_3
Final_Project
FinalProject
.vs
bin
Configuration
Controllers
Data
Migrations
Models
obj
Properties
wwwroot
.gitignore
appsettings.Development.json
appsettings.json
Dockerfile
FinalProject.csproj
FinalProject.csproj.user
Program.cs
README.md
Startup.cs
WeatherForecast.cs
OUTLINE
TIMELINE
MYSQL
main* Type here to search
1 #See https://aka.ms/containerfastmode to understand how Visual Studio uses this Dockerfile to build your image
2
3 FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
4 WORKDIR /app
5 EXPOSE 80
6 EXPOSE 443
7 EXPOSE 48719
8 EXPOSE 27017
9 EXPOSE 5000
10 EXPOSE 5001
11
12
13 FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
14 WORKDIR /src
15 COPY .
16
17 #COPY ["CoreWebApi/CoreWebApi.csproj", "CoreWebApi/"]
18 #RUN dotnet restore "CoreWebApi/CoreWebApi.csproj"
19 #COPY .
20 #WORKDIR "/src/CoreWebApi"
21 #RUN dotnet build "CoreWebApi.csproj" -c Release -o /app/build
22
23 RUN dotnet restore
24 RUN dotnet build --no-restore -c Release -o /app
25
26 FROM build AS publish
27 RUN dotnet publish "FinalProject.csproj" -c Release -o /app/publish
28
29 FROM base AS final
30 WORKDIR /app
31 #COPY --from=publish /app/publish .
32 COPY --from=publish /app/
33 ENTRYPOINT ["dotnet", "CoreWebApi.dll"]
Ln 1, Col 1 Spaces: 4 UTF-8 LF Dockerfile Prettier 0:49 31/03/2022

```

- Lakukan build dengan perintah dotnet build pada terminal

index.html - C# - Visual Studio Code

```

EXPLORER   Startup.cs  appsettings.json  index.html  ApiDbContext.cs  WeatherForecast.cs  WeatherForecastController.cs
            Assignment_1
            Assignment_2
            Assignment_3
            Final_Project:FinalProject
            .vs
            bin
            Configuration
            Controllers
            Data
            Migrations
            Models
            obj
            Properties
            wwwroot
            .gitignore
            appsettings.Development.json
            appsettings.json
            Dockerfile
            FinalProject.csproj
            FinalProject.csproj.user
            Program.cs
            README.md
            Startup.cs
            WeatherForecast.cs
            sesi-01
            sesi-02
            OUTLINE
            MYSQL
            powershell +  □  ^  ×
            Copyright (C) Microsoft Corporation. All rights reserved.

            Determining projects to restore...
            All projects are up-to-date for restore.
            FinalProject -> C:\Users\widya\Documents\C#\Final_Project\FinalProject\bin\Debug\net5.0\FinalProject.dll

            Build succeeded.
            0 Warning(s)
            0 Error(s)

            Time Elapsed 00:00:03.44
            PS C:\Users\widya\Documents\C#\Final_Project\FinalProject>
        
```

Ln 84, Col 61 Spaces: 4 UTF-8 CRLF HTML ✕ Prettier ⌂ 18:33 30/03/2022

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

index.html - C# - Visual Studio Code

```

EXPLORER   Startup.cs  appsettings.json  index.html  ApiDbContext.cs  WeatherForecast.cs  WeatherForecastController.cs
            Assignment_1
            Assignment_2
            Assignment_3
            Final_Project:FinalProject
            .vs
            bin
            Configuration
            Controllers
            Data
            Migrations
            Models
            obj
            Properties
            wwwroot
            .gitignore
            appsettings.Development.json
            appsettings.json
            Dockerfile
            FinalProject.csproj
            FinalProject.csproj.user
            Program.cs
            README.md
            Startup.cs
            WeatherForecast.cs
            sesi-01
            sesi-02
            OUTLINE
            MYSQL
            powershell +  □  ^  ×
            0 Errors

            Time Elapsed 00:00:03.44
            PS C:\Users\widya\Documents\C#\Final_Project\FinalProject> dotnet publish -c Release
            Microsoft (R) Build Engine version 16.11.2+f32259642 for .NET
            Copyright (C) Microsoft Corporation. All rights reserved.

            Determining projects to restore...
            All projects are up-to-date for restore.
            FinalProject -> C:\Users\widya\Documents\C#\Final_Project\FinalProject\bin\Release\net5.0\FinalProject.dll
            FinalProject -> C:\Users\widya\Documents\C#\Final_Project\FinalProject\bin\Release\net5.0\publish\
            PS C:\Users\widya\Documents\C#\Final_Project\FinalProject>
        
```

Ln 84, Col 61 Spaces: 4 UTF-8 CRLF HTML ✕ Prettier ⌂ 18:35 30/03/2022

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

The screenshot shows the Visual Studio Code interface. The code editor displays the `index.html` file with some HTML code. The terminal at the bottom shows the output of a build command:

```
[+] Building 58.2s (9/15)
-> => transferring context: 28
-> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:5.0
-> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:5.0
-> [build 1/5] FROM mcr.microsoft.com/dotnet/sdk:5.0@sha256:aae1e5029730fbf3995832a0bb8dbc3b8d0ed0b5aee1c1e96b5c0b97c8822b
-> [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:5.0@sha256:770bf871bc1b986e90d9de190b8804adf946fc76db5d1e4dd5e676ccb3a9135
-> [internal] load build context
-> => transferring context: 62.41MB
-> CACHED [build 2/5] WORKDIR /src
-> [build 3/5] COPY .
-> [build 4/5] RUN dotnet restore
```

The screenshot shows the Visual Studio Code interface. The code editor displays the `index.html` file. The terminal at the bottom shows the output of a build command followed by Docker image creation commands:

```
[+] Building 58.2s (9/15)
-> => transferring context: 28
-> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:5.0
-> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:5.0
-> [build 1/5] FROM mcr.microsoft.com/dotnet/sdk:5.0@sha256:aae1e5029730fbf3995832a0bb8dbc3b8d0ed0b5aee1c1e96b5c0b97c8822b
-> [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:5.0@sha256:770bf871bc1b986e90d9de190b8804adf946fc76db5d1e4dd5e676ccb3a9135
-> [internal] load build context
-> => transferring context: 62.41MB
-> CACHED [build 2/5] WORKDIR /src
-> [build 3/5] COPY .
-> [build 4/5] RUN dotnet restore
```

PS C:\Users\Widya\Documents\C#\Final_Project\FinalProject>

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

Images on disk

3 images Total size: 283.09 MB IN USE UNUSED Clean up...

LOCAL REMOTE REPOSITORIES

NAME	TAG	IMAGE ID	CREATED	SIZE
docker/getting-started	latest	bd9a9ff733898	about 2 months ago	28.79 MB
paymentapiimage	dev	3becbf500566	less than a minute ago	246.58 MB
weatherimagedini	dev	d25326c76b00	about 19 hours ago	213.08 MB

Connect to Remote Content

Not connected ✓ Store and backup your images remotely ✓ Unlock vulnerability scanning for greater security
✓ Collaborate with your team ✓ Connect for free Sign in

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

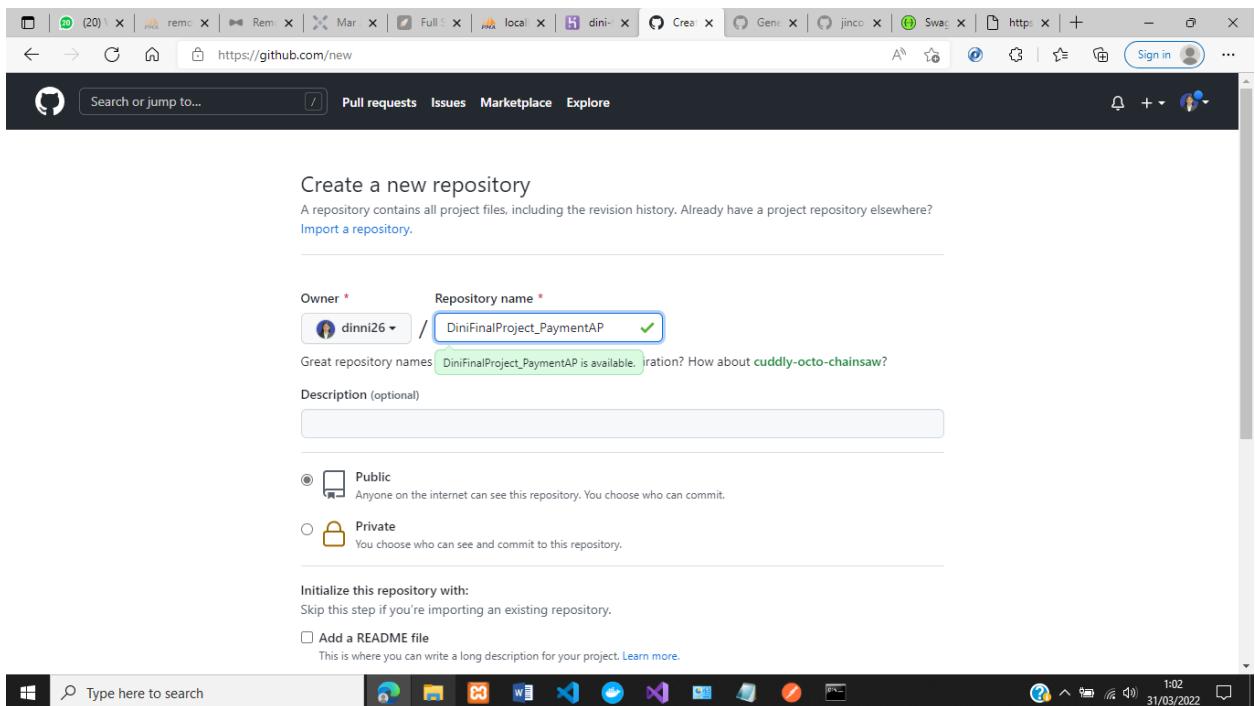
Containers / Apps

Search...

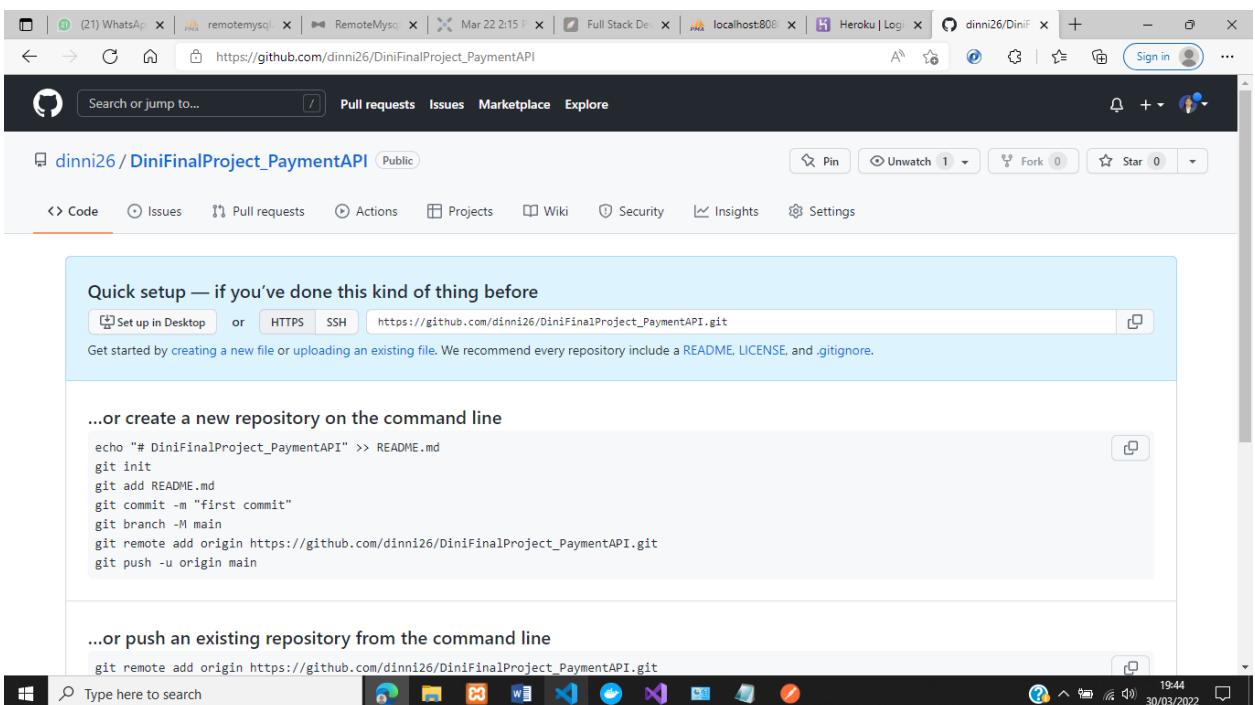
NAME	STATUS	PORTS
weathercontainererdini	EXITED (137)	PORT: 5051
elastic_almeida	EXITED (255)	PORT: 80
paymentapicontainer	RUNNING	PORT: 5051

PEMBUATAN REPO BARU DI GITHUB

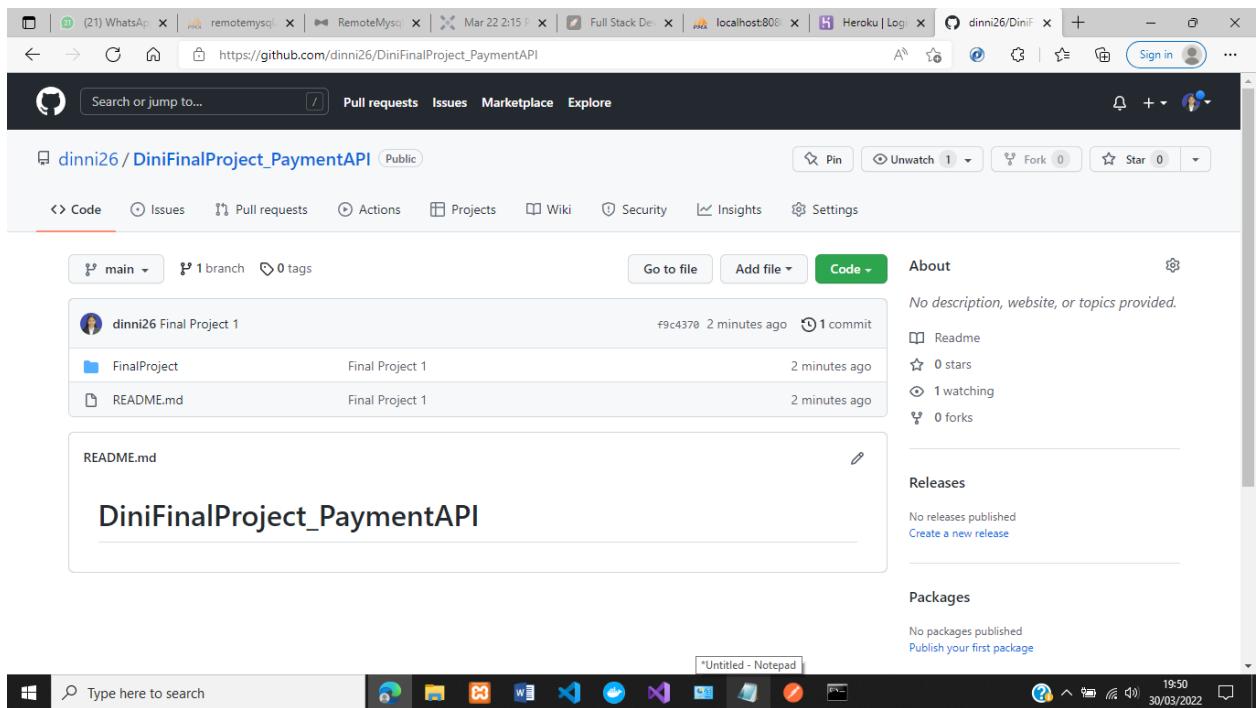
- Create new repo lalu beri nama repo yang baru dibuat



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

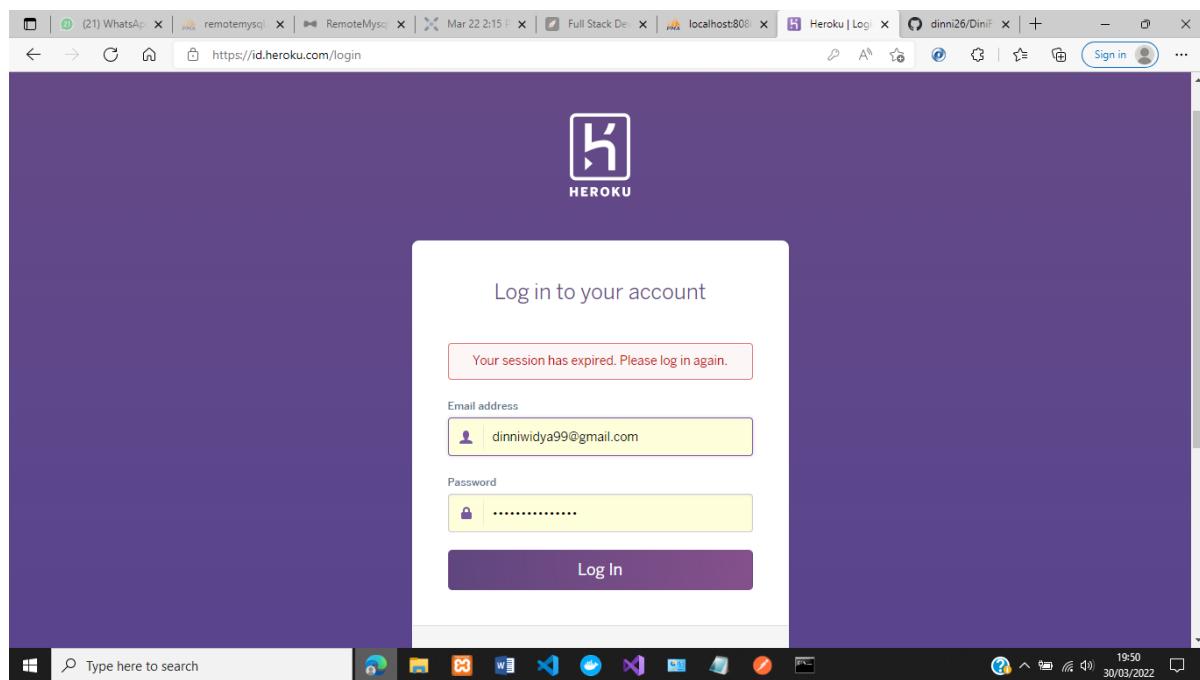


- Setelah dijalankan melalui cmd, maka repository baru akan tampil folder FinalProject yang sudah dibuat

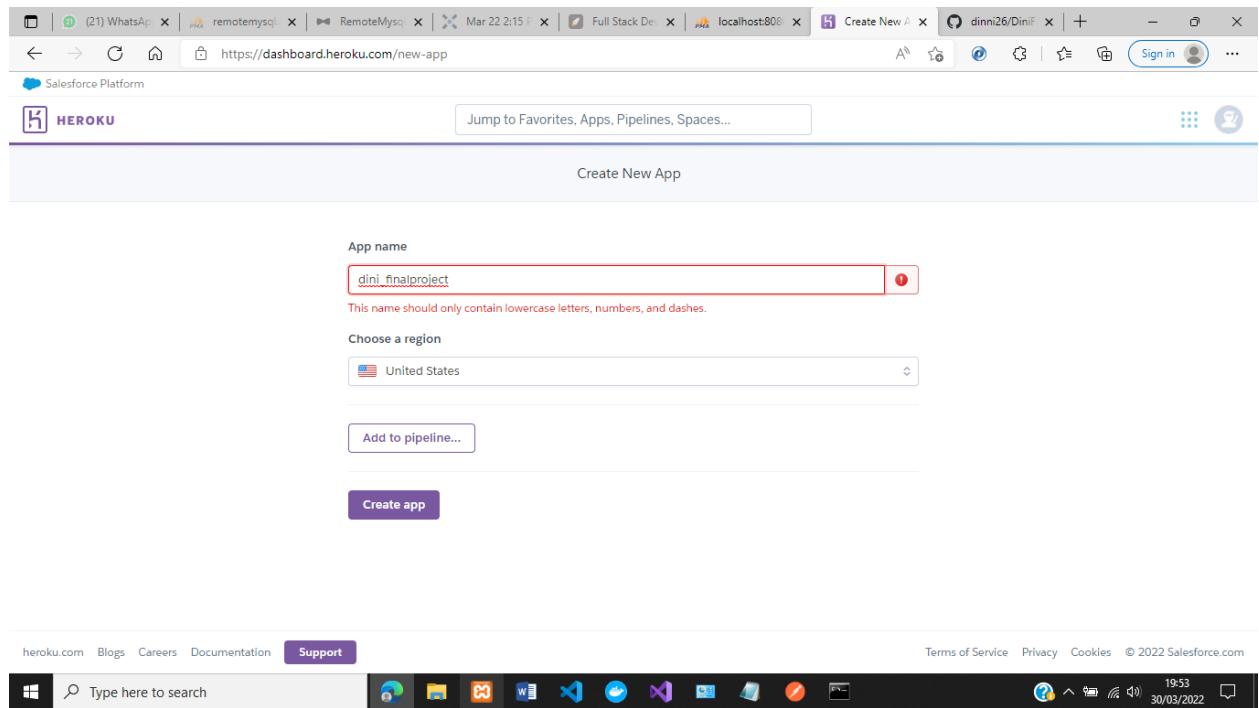


HEROKU

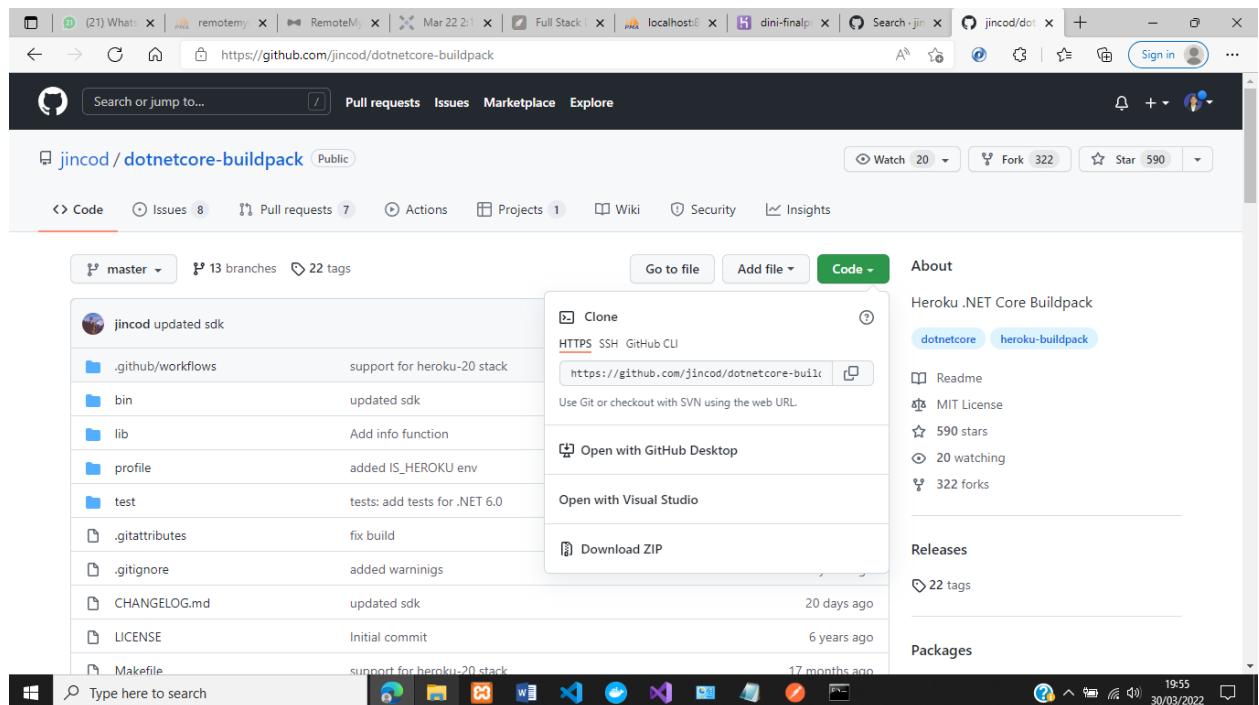
- Login Akun Heroku



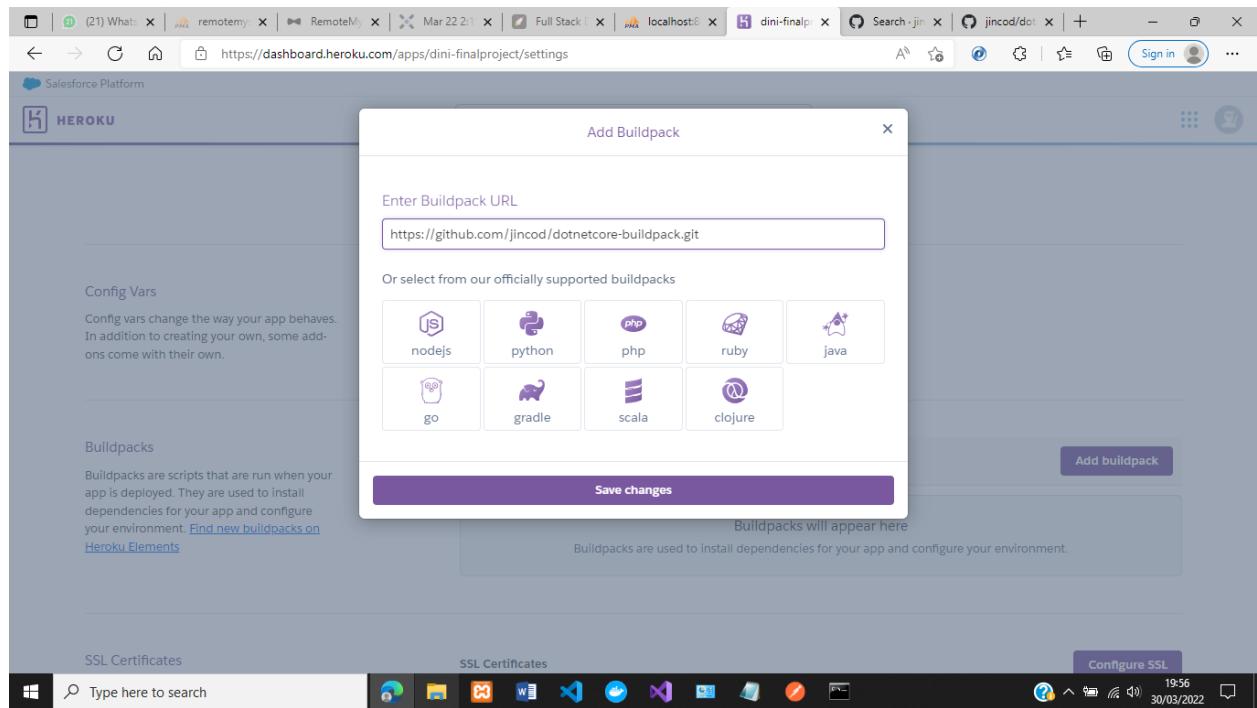
- Create New App lalu masukan nama app yang akan dibuat lalu create app.



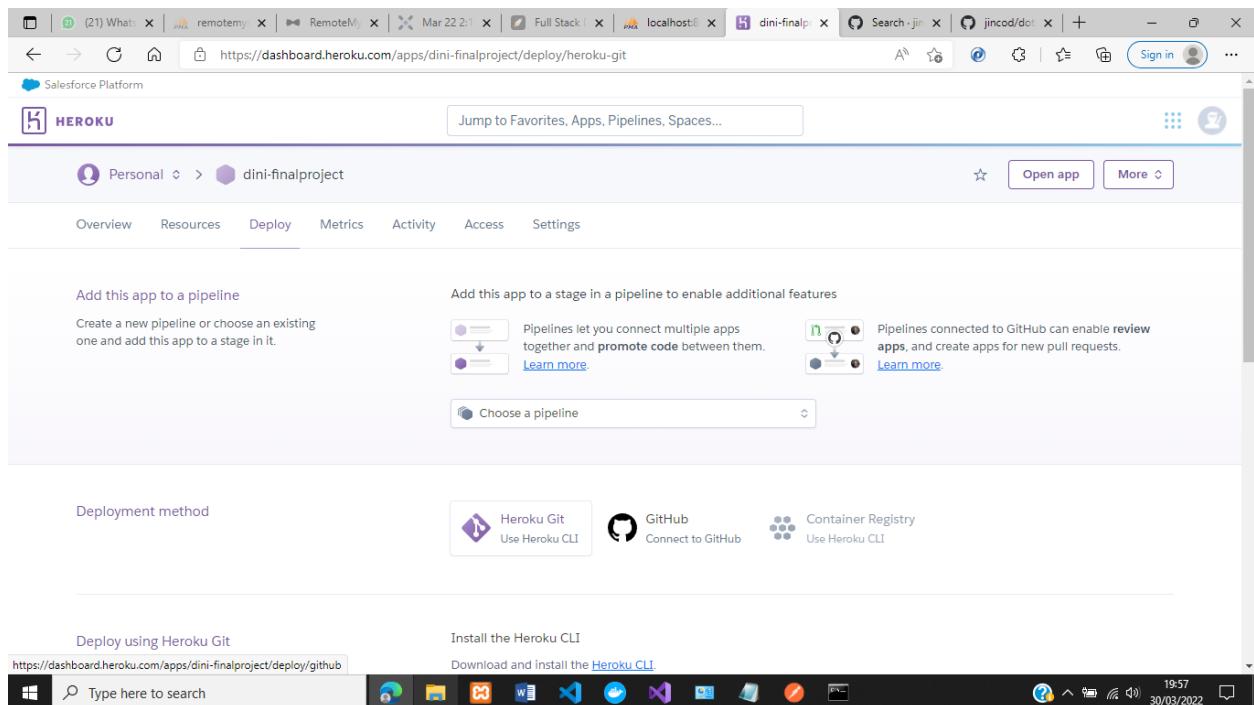
- Setelah Create App > Settings > Add Buildpack > Copy Link Jincod dari github



- Paste link Jincod Kedalam Buildpack URL



- Lalu Ke Deploy > Connect to Github



- Masukan Repo Github lalu Connect

Deployment method

Heroku Git Use Heroku CLI GitHub Connect to GitHub Container Registry Use Heroku CLI

Connect to GitHub

Search for a repository to connect to

dinni26 DiniFinalProject_PaymentAPI Search

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

dinni26/DiniFinalProject_PaymentAPI Connect

- Setelah itu scroll ke bawah lalu klik "Deploy Branch"

Follow the instructions [here](#).

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#)

Choose a branch to deploy

main

Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Manual deploy

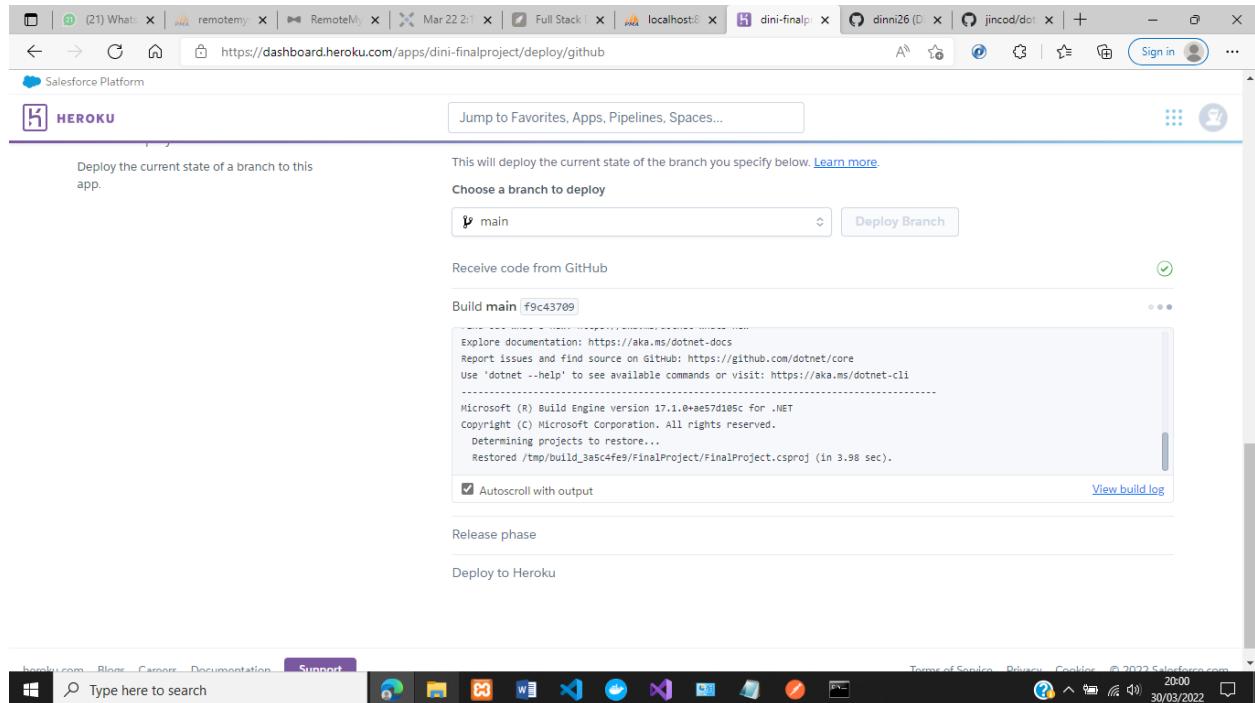
Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#).

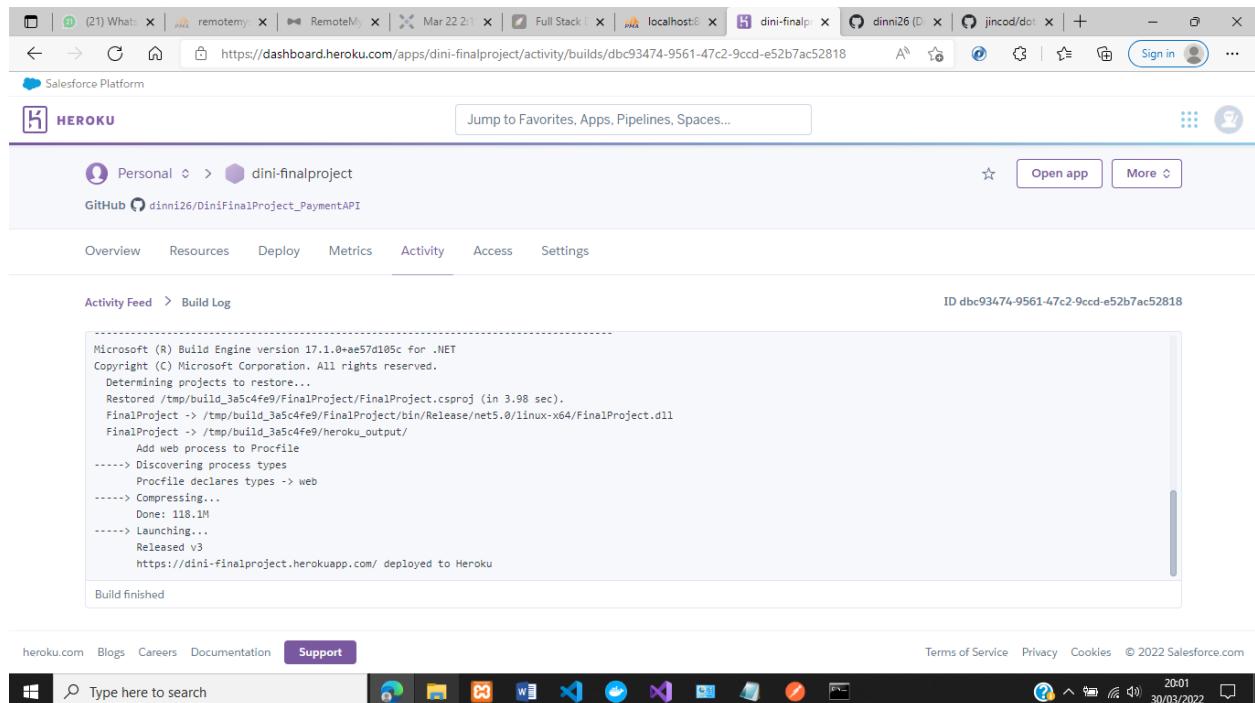
Choose a branch to deploy

main Deploy Branch

- Tunggu sampai proses selesai



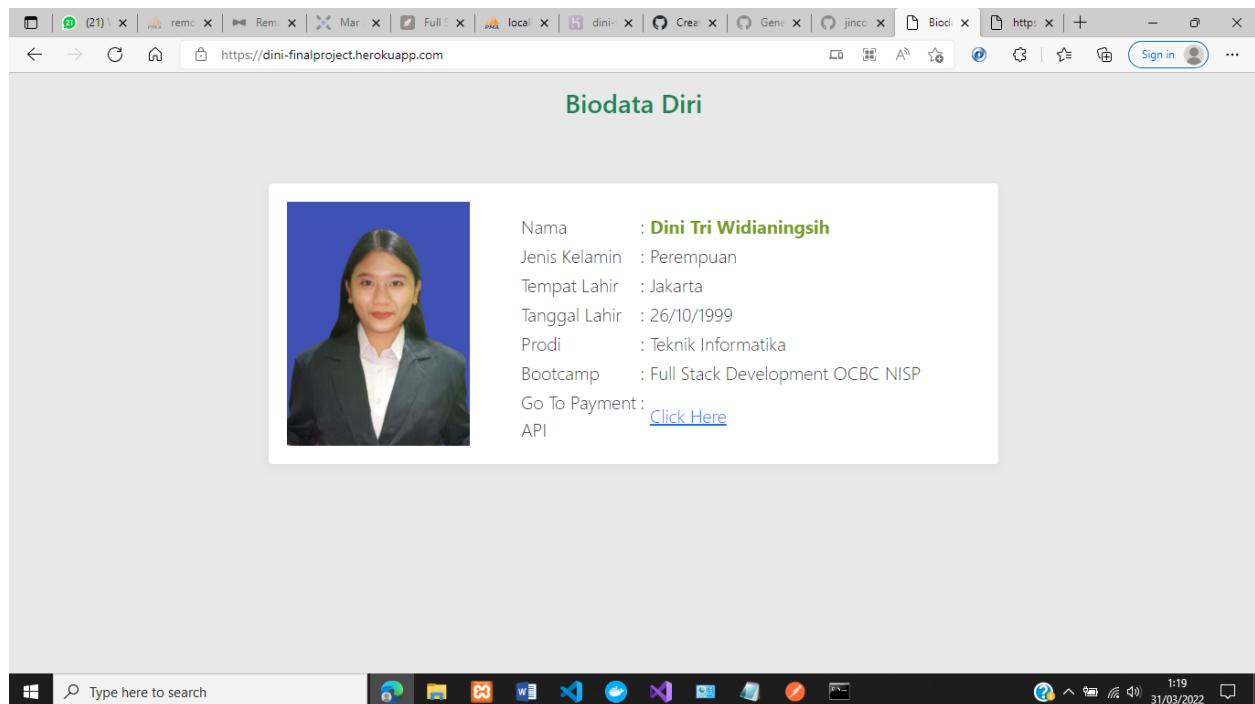
- Setelah selesai maka akan ada link heroku yang nantinya akan kita jalankan untuk penggunaan ProjectAkhir



- 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 \(github.com\)](https://dinni26/DiniFinalProject_PaymentAPI)

Link Heroku : [Biodata \(dini-finalproject.herokuapp.com\)](https://dini-finalproject.herokuapp.com/)

PANDUAN PENGGUNAAN FINAL PROJECT

Uji Coba Menggunakan 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

The screenshot shows the Swagger UI interface for a Payment API version 1. The top navigation bar includes tabs for 'Select a definition' (PaymentAPI v1) and 'Sign in'. Below the header, the title 'PaymentAPI' is displayed with 'v1' and 'OAS3' badges. A link to '/swagger/v1/swagger.json' is shown. On the right, there is a green 'Authorize' button with a lock icon.

AuthManagement

- POST /api/AuthManagement/Register** (green button)
- POST /api/AuthManagement/Login** (green button)
- POST /api/AuthManagement/RefreshToken** (green button)

Payment

- GET /api/Payment** (blue button)
- POST /api/Payment** (green button)
- GET /api/Payment/{id}** (blue button)
- PUT /api/Payment/{id}** (orange button)
- DELETE /api/Payment/{id}** (red button)

The screenshot shows the Swagger UI interface for a Payment API version 1. The top navigation bar includes tabs for 'Select a definition' (PaymentAPI v1) and 'Sign in'. Below the header, the title 'Payment' is displayed with 'v1' and 'OAS3' badges. A link to '/swagger/v1/swagger.json' is shown. On the right, there is a blue 'Authorize' button with a lock icon.

Payment

- GET /api/Payment** (blue button)
- POST /api/Payment** (green button)
- GET /api/Payment/{id}** (blue button)
- PUT /api/Payment/{id}** (orange button)
- DELETE /api/Payment/{id}** (red button)

WeatherForecast

- GET /WeatherForecast** (blue button)

Schemas

- UserRegistrationDto** >
- UserLoginRequest** >

The screenshot shows the Swagger UI interface for a Payment API version 1. The top navigation bar includes tabs for 'Select a definition' (PaymentAPI v1) and 'Sign in'. Below the header, the title 'Payment' is displayed with 'v1' and 'OAS3' badges. A link to '/swagger/v1/swagger.json' is shown. On the right, there is a blue 'Authorize' button with a lock icon.

Payment

- GET /api/Payment** (blue button)
- POST /api/Payment** (green button)
- GET /api/Payment/{id}** (blue button)
- PUT /api/Payment/{id}** (orange button)
- DELETE /api/Payment/{id}** (red button)

WeatherForecast

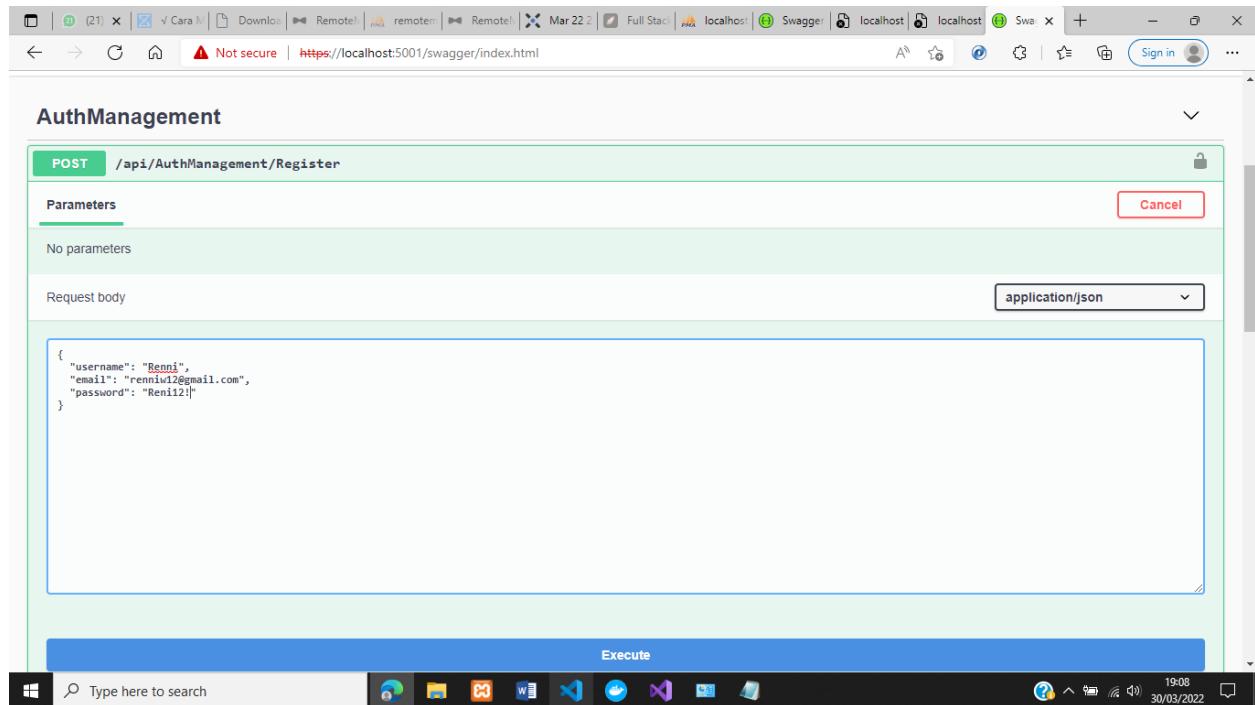
- GET /WeatherForecast** (blue button)

Schemas

- UserRegistrationDto** >
- UserLoginRequest** >

Register

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

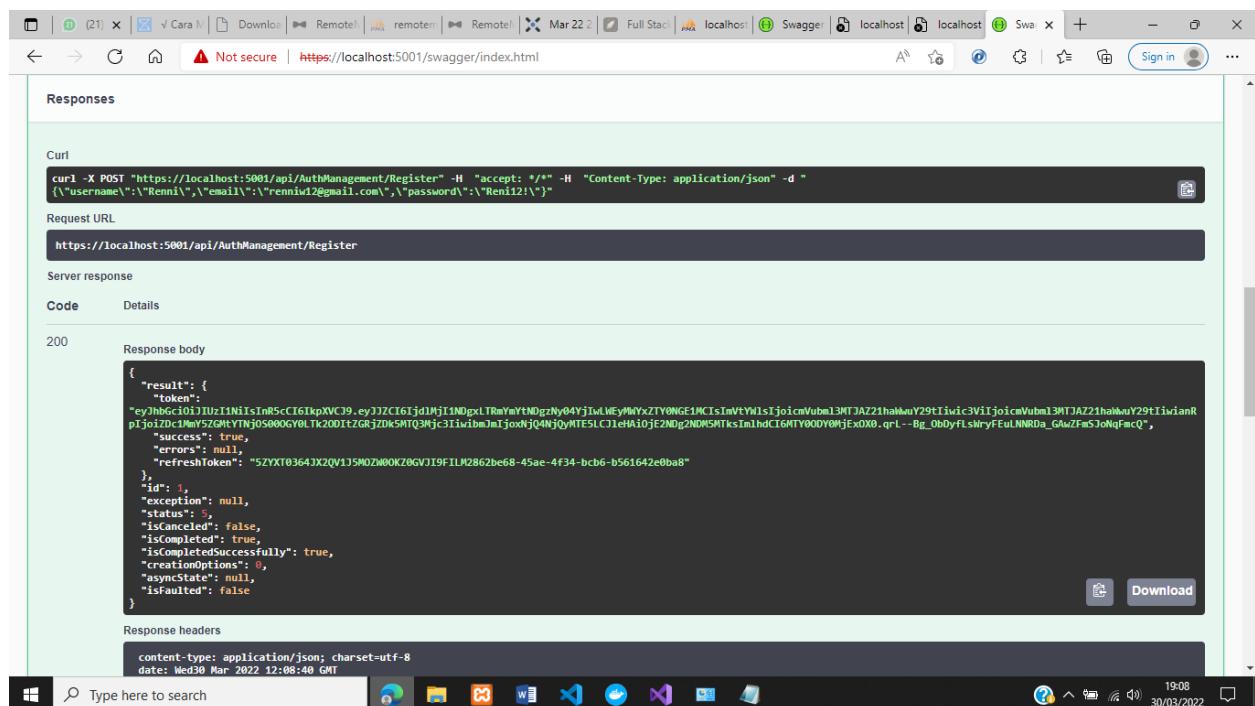


The screenshot shows the Swagger UI interface for a .NET Core application. The URL is <https://localhost:5001/swagger/index.html>. The endpoint being tested is `/api/AuthManagement/Register`. The method is `POST`. The request body is set to a JSON object:

```
{ "username": "Renni", "email": "renniid12@gmail.com", "password": "Reni12!" }
```

The 'Execute' button is located at the bottom of the request form. The status bar at the bottom right shows the date and time as 30/03/2022 19:08.

Execute, lalu respon akan menampilkan token.



The screenshot shows the Swagger UI interface after executing the `/api/AuthManagement/Register` endpoint. The response code is 200. The response body is a JSON object:

```
{ "result": { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjZC16IjdjI1NDgxLTRmYmVtNDgzNy04YjIwLWEyMWYxZTY0NGE1NCIsImVtYnlsIjoicmVubm13MTJAZ221haKwuY29tIiwickVubm13MTJAZ221haKwuY29tIiwicknRpiJo1ZDc1Mm5ZGMhYTINj0S000Gy0Ltk20Dit2GRjDk5MTQmjc3IiwbmJnTjoxnj04nj0yMTE5LC1eHai0je2Ng2NDMSMTksImhdCI6NTY0ODY0MjExOX0.4g--Bg_0bDyFlsWryFeulNNRDa_GwZfmsJnQfmcQ", "success": true, "errors": null, "refreshToken": "S2YXT0364JX2QV13MOZn80KZ0GVJ19FILM2862be68-45ae-4f34-bcb6-b561642e0ba8" }, "id": 1, "exception": null, "status": 5, "isCancelled": false, "isCompleted": true, "isCompletedSuccessfully": true, "creationOptions": 0, "asyncState": null, "isFaulted": false }
```

The response headers are:

```
content-type: application/json; charset=utf-8
date: Wed, 30 Mar 2022 12:08:40 GMT
```

The status bar at the bottom right shows the date and time as 30/03/2022 19:08.

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.

The screenshot shows the Swagger UI interface for a POST request to the endpoint `/api/AuthManagement/Login`. The request body is defined as follows:

```
{ "email": "renniw12@gmail.com", "password": "Reni12!" }
```

The 'Execute' button is visible at the bottom of the request panel. The browser's address bar shows `https://localhost:5001/swagger/index.html`. The Windows taskbar at the bottom indicates the date and time as 30/03/2022 19:09.

The screenshot shows the Swagger UI interface displaying the response details for the login request. The response code is 200, and the response body is a JSON object containing the following data:

```
{ "result": { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZC16Ijd1MjI1NDgxLTRmYmYtNDg2Ny04YjIwLWEyMjYzTjY0NGE1MCIsImVtYWIslSjoicnVubm13MTJAZ21haMuY29tIiwiic3ViIjoiwmVubm13MTJAZ21haMuY29tIiwanRpjo1NzEyN2l3UmItZjpMy00MDjkIlgzZETn2E3mlJ00GE4ZjgziwibmJmIjoxNjQ4njdyMtg4LC1leHai0je2NDg2NDM500gsImhdI6MTY0ODY0MjE4OH0.hMaX1BF7kux8NkZB4rTjy80zikJiwb--d1mhPIQew", "success": true, "errors": null, "refreshToken": "GX3QJ3Y24CDZSC4F9UK7KE1X6017T31L02U5ec974f-c950-4131-aadf-46b5ad78c146" }, "id": 2, "exception": null, "status": 5, "isCancelled": false, "isCompleted": true, "isCompletedSuccessfully": true, "creationOptions": 0, "asyncState": null, "isAudited": false }
```

The response headers section shows:

```
content-type: application/json; charset=utf-8
date: Wed, 30 Mar 2022 12:09:49 GMT
server: Kestrel
```

The browser's address bar shows `https://localhost:5001/swagger/index.html`. The Windows taskbar at the bottom indicates the date and time as 30/03/2022 19:10.

Refresh Token

- Percobaan Refresh Token sebelum 30 Menit

POST /api/AuthManagement/RefreshToken

Parameters

No parameters

Request body

application/json

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ2Cl6IjdmJi1NDgxLTRmYmYtNDg2Ny04Yj1wLiEwMjYxTY0NGE1MCIsImVtYnlsIjoicmVubm13MTJAZ21haWuY29tIiwi3ViijoicmVubm13MTJAZ21haWuY29tIiwiianRpIjoInzEyN2U3NmItZjgwly00MDJkLTgtZTET2E3yJmU00GE42jgzIiwiIbmJmJoxhj04NjQyMTg4LC1eHA0jE2NDg2NDM5DgsImIhdC16MTY00DY0MjE4OH0.hMax1BF7ku8NkZB4rTjY80zikIwb--dImhPIOPesu",
  "refreshToken": "GX3Q3Y2MC0ZC4F9UK7KEIX6D17T31L02U15ec974f-c950-4131-aadf-46b5ad78c146"
}
```

Execute Clear

Responses

Responses

Curl

```
curl -X POST "https://localhost:5001/api/AuthManagement/RefreshToken" -H "accept: */*" -H "Content-Type: application/json" -d "{
  \"token\": \"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ2Cl6IjdmJi1NDgxLTRmYmYtNDg2Ny04Yj1wLiEwMjYxTY0NGE1MCIsImVtYnlsIjoicmVubm13MTJAZ21haWuY29tIiwi3ViijoicmVubm13MTJAZ21haWuY29tIiwiianRpIjoInzEyN2U3NmItZjgwly00MDJkLTgtZTET2E3yJmU00GE42jgzIiwiIbmJmJoxhj04NjQyMTg4LC1eHA0jE2NDg2NDM5DgsImIhdC16MTY00DY0MjE4OH0.hMax1BF7ku8NkZB4rTjY80zikIwb--dImhPIOPesu\",
  \"refreshToken\": \"GX3Q3Y2MC0ZC4F9UK7KEIX6D17T31L02U15ec974f-c950-4131-aadf-46b5ad78c146\"
}"
```

Request URL

https://localhost:5001/api/AuthManagement/RefreshToken

Server response

Code Details

200 Response body

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

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 30 Mar 2022 12:10:42 GMT
server: Kestrel
```

Responses

Code Description Links

200 Success No links

((Muncul pesan bahwa token masih bisa digunakan))

- Percobaan melakukan Refresh Token Setelah 30 Menit

The screenshot shows a browser window with multiple tabs open. The active tab is 'Swagger UI' at <https://localhost:5001/swagger/index.html>. A warning message 'Not secure' is displayed above the address bar. The main content area shows a 'Responses' section for a 'RefreshToken' endpoint. It includes a 'Curl' command, a 'Request URL' of <https://localhost:5001/api/AuthManagement/RefreshToken>, and a 'Server response' block. The response code is 200, and the response body is a JSON object:

```
{
  "token": null,
  "success": false,
  "errors": [
    "Token has expired, please re-login"
  ],
  "refreshToken": null
}
```

Below the response, there are 'Download' and 'Copy' buttons. The status bar at the bottom right shows the date as 28/03/2022 and the time as 16:47.

((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

The screenshot shows a browser window with multiple tabs open. The active tab is 'Swagger UI' at <https://localhost:5001/swagger/index.html>. A warning message 'Not secure' is displayed above the address bar. The main content area shows a 'PaymentAPI v1' definition. A modal dialog titled 'Available authorizations' is open. It contains a 'Bearer (apiKey)' section with instructions to enter a valid token. A text input field contains the value 'Bearer eyJhbGciOiJIUzI1NilsInR5C16IkpxVCJ0'. At the bottom of the dialog are 'Authorize' and 'Close' buttons. The status bar at the bottom right shows the date as 30/03/2022 and the time as 19:11.

PaymentAPI v1 OAS3

AuthManagement

POST /api/AuthManagement/Register

Parameters

No parameters

Request body

application/json

Logout Close

Value: ****

Authorizations

Bearer (apiKey)

Authorized

Enter 'Bearer' [space] and then your valid token in the text input below.
Example: "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9"
Name: Authorization
In: header
Value: ****

Authorize

PAYMENT POST

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

Payment

GET /api/Payment

POST /api/Payment

Parameters

No parameters

Request body

application/json

```
{ "paymentDetailId": 1, "cardHolderName": "Reni Widiastuty", "cardNumber": "54453", "expirationDate": "2022/12/23", "securityCode": "Renii123" }
```

The screenshot shows a browser window with the URL <https://localhost:5001/swagger/index.html>. The page displays a 'Responses' section for a POST request to `/api/Payment`. The response code is 201 (Undocumented), and the response body is:

```
{
  "paymentDetailId": 1,
  "cardOwnerName": "Reni Widiastry",
  "cardNumber": "54453",
  "expirationDate": "2022/12/23",
  "securityCode": "Renii23"
}
```

The response headers are:

```
content-type: application/json; charset=utf-8
date: Wed, 30 Mar 2022 12:13:13 GMT
location: https://localhost:5001/api/Payment?PaymentDetailId=1
server: Kestrel
```

Below the responses, there is a 'Responses' section for a 200 status code, which is described as 'Success'. The 'Links' section indicates 'No links'.

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

GET PAYMENT Untuk mengecek data yang sudah di input.

The screenshot shows a browser window with the URL <https://localhost:5001/swagger/index.html>. The page displays a 'Payment' section with a 'GET /api/Payment' operation. The 'Parameters' section shows 'No parameters'. The 'Execute' button is highlighted. Below it is a 'Responses' section for a 200 status code. The response body is:

```
[
```

The response headers are:

```
content-type: application/json; charset=utf-8
date: Wed, 30 Mar 2022 12:13:13 GMT
location: https://localhost:5001/api/Payment?PaymentDetailId=1
server: Kestrel
```

Below the responses, there is a 'Responses' section for a 200 status code, which is described as 'Success'. The 'Links' section indicates 'No links'.

Request URL
https://localhost:5001/api/Payment

Server response

Code	Details
200	Response body <pre>[{ "paymentDetailId": 1, "cardOwnerName": "Reni Widiastuty", "cardNumber": "54453", "expirationDate": "2022/12/23", "securityCode": "Renil23" }, { "paymentDetailId": 2, "cardOwnerName": "Bayu Widi Prayogo", "cardNumber": "65543", "expirationDate": "2022/11/10", "securityCode": "Bayul23" }, { "paymentDetailId": 3, "cardOwnerName": "Miwik Supiyati", "cardNumber": "2345", "expirationDate": "2022/09/21", "securityCode": "Miwil123" }]</pre> Response headers <pre>content-type: application/json; charset=utf-8 date: Wed, 30 Mar 2022 12:15:24 GMT server: Kestrel</pre>

Responses

Code	Description
GET	/api/Payment/{id}

Type here to search

((Data sudah masuk ke dalam database))

GET PAYMENT BERDASARKAN ID

Co: Mamanggil ID = 2

GET /api/Payment/{id}

Parameters

Name	Description
id <small>required</small>	integer(\$int32) (path)

Execute Clear

Responses

Curl

```
curl -X GET "https://localhost:5001/api/Payment/2" -H "accept: */*" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ejYjC16j1Mj1N0gxL1RwYmTN0gZy04YjIwLWeY/WMyZT0NGE1MCisImVtYmIsIjoicmVubm13MTJAZ21haWwY29tliwi3ViijoicmVubm13MTJAZ21haWwY29tlixiianRpIjoINzEyN2l3NmtZjpdMy00MDjkLTgzZTEtn2E3YmU0GE47jgzIiwbmJmjoxNjQ4NjQyMtg4lC1eHaiojE2NDg2N0M500gsImhdC16MTY0ODY0MjE4OH0.hMaX1BF7kux8NkZB4rTjy80zikJiwb--d1mhPIQPesw"
```

Request URL
https://localhost:5001/api/Payment/2

Server response

Code	Details
200	Response body

Type here to search

Curl
curl -X GET "https://localhost:5001/api/Payment/2" -H "accept: */*" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjdjMjIzNDgxLTBmYnYtNGzNy04YjIwLWEyMwYxZTY0NGE1MCIsImVtbWlzcjoicmVubm13MTJAZ21haWwuY29tIiwi3V1IjoicmVubm13MTJAZ21haWwuY29tIiwiianRpIjoINzEyNzU3NzIzJgdy00MDJkLTgzZTETn2E3YmJ00GE42jgzIiwibmJmIjoxNjQ4NjQyMTg4LC1eHA10jE2Ng2NDw50DgsImhdC16MTY0ODY0MjE4OH0.hAx1BF7kux8NkZB4rTjY80zikIwb--d1mhPIQPesw"
Request URL
<https://localhost:5001/api/Payment/2>
Server response
Code Details
200 Response body
{
 "paymentDetailId": 2,
 "cardOwnerName": "Bayu Widi Prayogo",
 "cardNumber": "65543",
 "expirationDate": "2022/11/10",
 "securityCode": "Bayu123"
}
Response headers
content-type: application/json; charset=utf-8
date: Wed, 30 Mar 2022 12:16:08 GMT
server: Kestrel
Responses
Code Description Links
200 Success No links
wwwroot
Type here to search 19:16 30/03/2022

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

PUT / UPDATE

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

PUT /api/Payment/{id}
Parameters
Name Description
id * required integer(\$int32) 2
(path)
Request body
application/json
{
 "paymentDetailId": 2,
 "cardOwnerName": "Dini Tri Widianingsih",
 "cardNumber": "22213",
 "expirationDate": "2022/10/26",
 "securityCode": "Dini123"
}
wwwroot
Type here to search 19:18 30/03/2022

Responses

Curl

```
curl -X PUT "https://localhost:5001/api/Payment/2" -H "accept: */*" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.evJZC1G1jdMjI1Nm0gxlTmYeYtDgzN04YJiwLWEyMwzTYVNGE1MCisImVtYlsIjoicmVu bm13MTJAZ21haWw0Y29tIiwi3ViTjjoicmVu bm13MTJAZ21haWw0Y29tIiwiianRpIjoiNzEyN2U3NmtZjg0yj0w0DJkltgZTETn2E3Y0BGE4ZjgzIxiibmJmljoxNj04Nj0yMTg4LC1jeHA10je2NDMS00gS1mlhdIGMTY0ODY0MjE4OH0.bMAX1BF7kuxBNkZB4rTjYB0zikJlub-dIMbPIQPesw" -H "Content-Type: application/json" -d "{\"paymentDetailId\":2,\"cardOwnerName\":\"Dini Tri Widianingsih\",\"cardNumber\":\"22213\",\"expirationDate\":\"2022/10/26\",\"securityCode\":\"Dini123\"}"
```

Request URL

<https://localhost:5001/api/Payment/2>

Server response

Code	Details
200	<p>Response body</p> <p>Data Updated!</p> <p>Download</p> <p>Response headers</p> <p>content-type: text/plain; charset=utf-8 date: Wed, 30 Mar 2022 12:17:59 GMT server: Kestrel</p>

Responses

Code	Description	Links
200	Success	No links

((Data Berhasil Di Update))

DELETE

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

The screenshot shows a browser window with multiple tabs open, including 'Cara M', 'Downloads', 'Remote1', 'remote1', 'Mar 22 2', 'Full Stack', 'localhost', 'Swagger', 'localhost', 'localhost', 'Swag', and 'Sign in'. The main content area is a Swagger UI interface for a 'DELETE /api/Payment/{id}' endpoint.

DELETE /api/Payment/{id}

Parameters

Name	Description
id <small>* required</small>	integer(\$int32) (path)

Responses

Curl

```
curl -X DELETE "https://localhost:5001/api/Payment/1" -H "accept: */*" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInRS...[redacted]"
```

Request URL

```
https://localhost:5001/api/Payment/1
```

Server response

Code **Details**

200 Response body

Curl

```
curl -X DELETE "https://localhost:5080/api/Payment/1" -H "accept: */*" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCIkIkpXVC9...eyJ7ZC16JdMjI1NDg1T8wYmYzNDgNy04Yj1wLWEyMmYzTY8wGE3MCIsImtYhlsIjoicmVubm13MTAZ21haWuY29tTixic3ViijoicmVubm13MTAZ21haWuY29tTiwianRpIjoizNzEyNzU3NmItZjpaW0D0jKLtgZTEtn2E3mJ00GE47jgzLiwbmJmIjoxNjQ4NjQyMtg4LC1eHai0jE2Nd2NDm5DgsIm1hdC16MTY0ODY0je4OH0.hNxz1BF7kux8NkZB4rTjY80zikJlw--dImhPIQPesw"
```

Request URL

<https://localhost:5080/api/Payment/1>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "paymentDetailId": 1, "cardOwnerName": "Reni Midiaستuty", "cardNumber": "54453", "expirationDate": "2022/12/23", "securityCode": "Ren1123" }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Wed, 30 Mar 2022 12:18:53 GMT server: Kestrel</pre>

Responses

Code	Description	Links
200	Success	No links

((Respons data telah berhasil dihapus))

CEK KEMBALI MELALUI GET PAYMENT

The screenshot shows a browser window with the address bar displaying 'Not secure | https://localhost:5001/swagger/index.html'. The main content area is titled 'Payment' and shows a 'GET /api/Payment' operation. Under 'Parameters', it says 'No parameters'. Below are 'Execute' and 'Clear' buttons. Under 'Responses', there is a 'Curl' code block and a 'Request URL' input field containing 'https://localhost:5001/api/Payment'. The 'Server response' section shows a 'Code' dropdown set to '200' and a 'Details' section for 'Response body' which contains '[{ }]'. The bottom of the screen shows a taskbar with various icons and a search bar.

((ID = 1 telah berhasil dihapus))

SWAGGER/WEATHERFORECAST

- Klik Try It Out lalu Execute

Not secure | <https://localhost:5001/swagger/index.html>

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, "temperatureF": 122}],
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Wed 30 Mar 2022 12:20:22 GMT
server: Kestrel
```

Responses

Type here to search

19:20 30/03/2022

Not secure | <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"}]
```

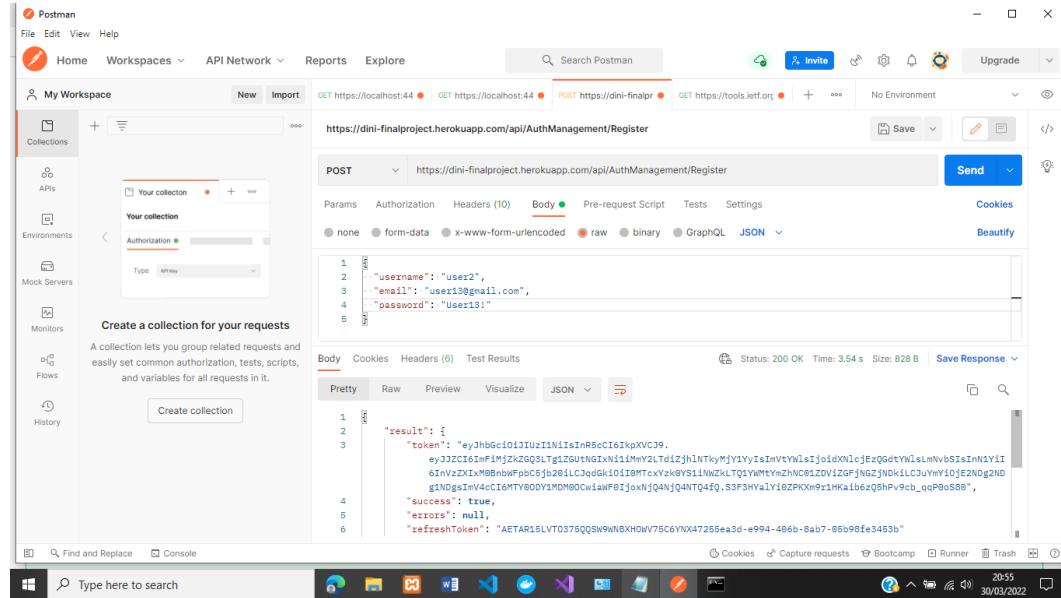
23:35 30/03/2022

POSTMAN

REGISTER

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

- Masukan username, email, dan password baru



The screenshot shows the Postman interface with a POST request to <https://dini-finalproject.herokuapp.com/api/AuthManagement/Register>. The request body is set to JSON and contains the following data:

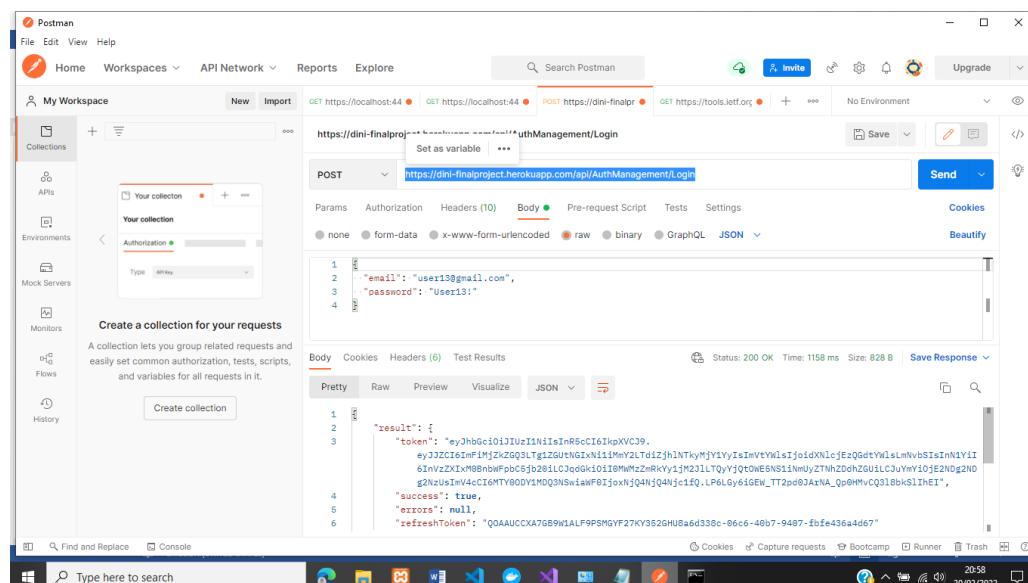
```
1 "username": "userx2",
2 "email": "user13@gmail.com",
3 "password": "User13!"
```

The response status is 200 OK, and the response body is a large JSON token.

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>



The screenshot shows the Postman interface with a POST request to <https://dini-finalproject.herokuapp.com/api/AuthManagement/Login>. The request body is set to JSON and contains the following data:

```
1 "email": "user13@gmail.com",
2 "password": "User13!"
```

The response status is 200 OK, and the response body is a large JSON token.

REFRESH TOKEN

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

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

The screenshot shows the Postman application interface. A collection named 'Your collection' is selected. A POST request is made to <https://dini-finalproject.herokuapp.com/api/AuthManagement/RefreshToken>. The response status is 200 OK, time 1532 ms, size 277 B. The response body is:

```
1 "token": null,
2 "success": false,
3 "errors": [
4     "Token has not yet expired"
5 ],
6 "refreshToken": null
```

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>

The screenshot shows the Postman application interface. A collection named 'My first collection' is selected. A POST request is made to <https://dini-final-project.herokuapp.com/api/AuthManagement/Login>. The response status is 200 OK, time 2.13 s, size 833 B. The response body is:

```
1 "result": {
2     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
3         eyJzIzC161jd1Mj1lNDgxLTRmYyNDg2Ny04Yj1wLMyWmYyZTV0NGE1MCIsImVtYWhsIjoicmVubml3MTJAZ21haWwUY29tIiwic3V
4         iijoiicVubm13MTJAZ21haWwUY29tIiwanRoiIoiMw1NTi4MTMtnjYxNC00NDU4LwI4YjgtYmY3ZWRI0dg0MTVkiIwibmJiIoxkj
5         Q4NjUSMjMyLC1leHai0je2N0g2NjEwMzIsImlhdCI6MTY0ODY1OTizMn0.pk9hGvHzkzMBGGNuObkLex0X71ZYNQHs5n19mMsq8",
6     "success": true,
```

PAYMENT POST

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

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

The screenshot shows the Postman application interface. In the center, there is a request configuration window for a POST method to the URL <https://dini-finalproject.herokuapp.com/api/payment>. The 'Body' tab is selected, showing a JSON payload:

```
1   ...
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8   ...
9   ...
10  ...
11  ...
12  ...
13  ...
14  ...
15  ...
16  ...
17  ...
18  ...
19  ...
```

The payload contains fields: cardOwnerName, cardNumber, expirationDate, and securityCode. The response status is 201 Created, Time: 2.36 s, Size: 399 B. Below the body, the response JSON is displayed:

```
1   {
2     "paymentDetailId": 6,
3     "cardOwnerName": "Mega Utami",
4     "cardNumber": "22383",
5     "expirationDate": "2022/10/10",
6     "securityCode": "Mega123"
7   }
```

PAYMENT GET

Mendapatkan Semua data yang ada didalam database.

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

The screenshot shows the Postman application interface. In the center, there is a request configuration window for a GET method to the URL <https://dini-finalproject.herokuapp.com/api/payment>. The 'Body' tab is selected, showing a JSON response:

```
1   [
2     {
3       "paymentDetailId": 2,
4       "cardOwnerName": "Dini Tri Widianingsih",
5       "cardNumber": "22213",
6       "expirationDate": "2022/10/26",
7       "securityCode": "Dini123"
8     },
9     {
10       "paymentDetailId": 3,
11       "cardOwnerName": "Wiwik Supiyati",
12       "cardNumber": "2346",
13       "expirationDate": "2022/09/25",
14       "securityCode": "Wiwik123"
15     },
16     {
17       "paymentDetailId": 4,
18       "cardOwnerName": "Dicky Sastro",
19       "cardNumber": "22234",
20     }
21   ]
```

PAYMENT GET WHERE ID

Mendapatkan data payment details dengan ID yang dimasukan.

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

The screenshot shows the Postman application interface. In the center, there is a request card for a GET request to `https://dini-finalproject.herokuapp.com/api/payment/2`. The response status is 200 OK, and the response body is displayed in JSON format:

```
1 "paymentDetailId": 2,
2 "cardOwnerName": "Dini Tri Widianingsih",
3 "cardNumber": "22213",
4 "expirationDate": "2022/10/26",
5 "securityCode": "Dini123"
```

PAYMENT PUT

Mengubah data dengan ID tertentu lalu menampilkan respons perubahan.

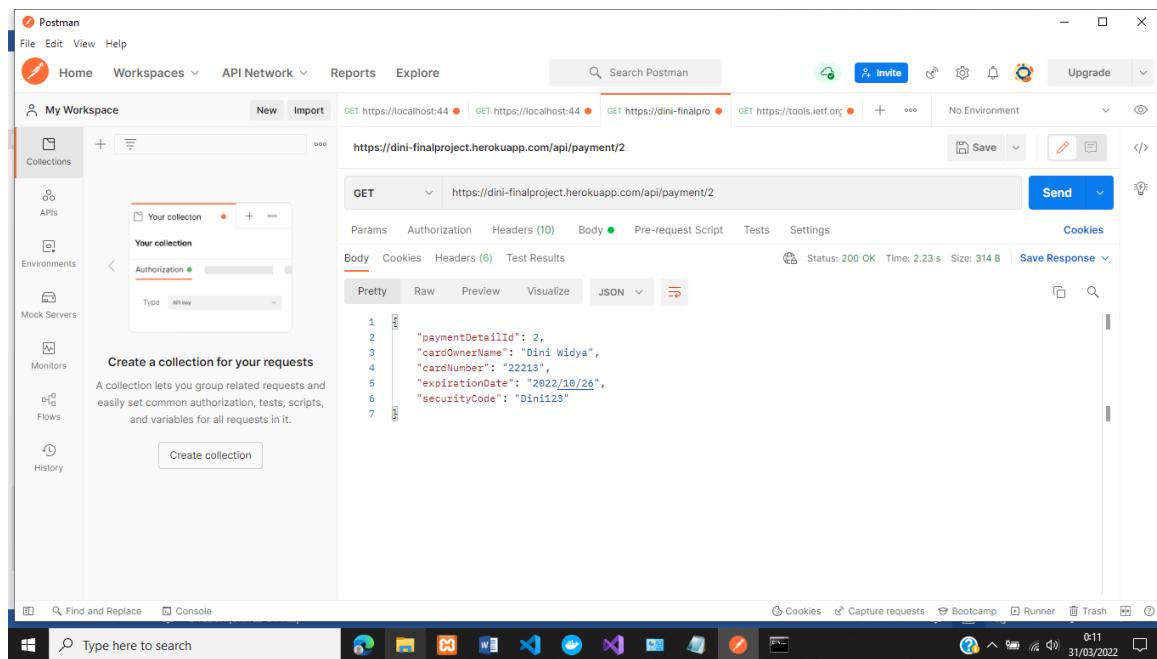
Link : <https://dini-finalproject.herokuapp.com/api/payment/2>

The screenshot shows the Postman application interface. In the center, there is a request card for a PUT request to `https://dini-finalproject.herokuapp.com/api/payment/2`. The response status is 200 OK, and the response body is displayed in JSON format:

```
1 "paymentDetailId": 2,
2 "cardOwnerName": "Dini Widya",
3 "cardNumber": "22213",
4 "expirationDate": "2022/10/26",
5 "securityCode": "Dini123"
```

Below the response, the text "Data Updated!" is visible.

((Cek Get Payment Kembali))



The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace displays a collection named "Your collection". A specific request is selected, showing a GET method directed at <https://dini-finalproject.herokuapp.com/api/payment/2>. The response status is 200 OK, with a response body containing JSON data:

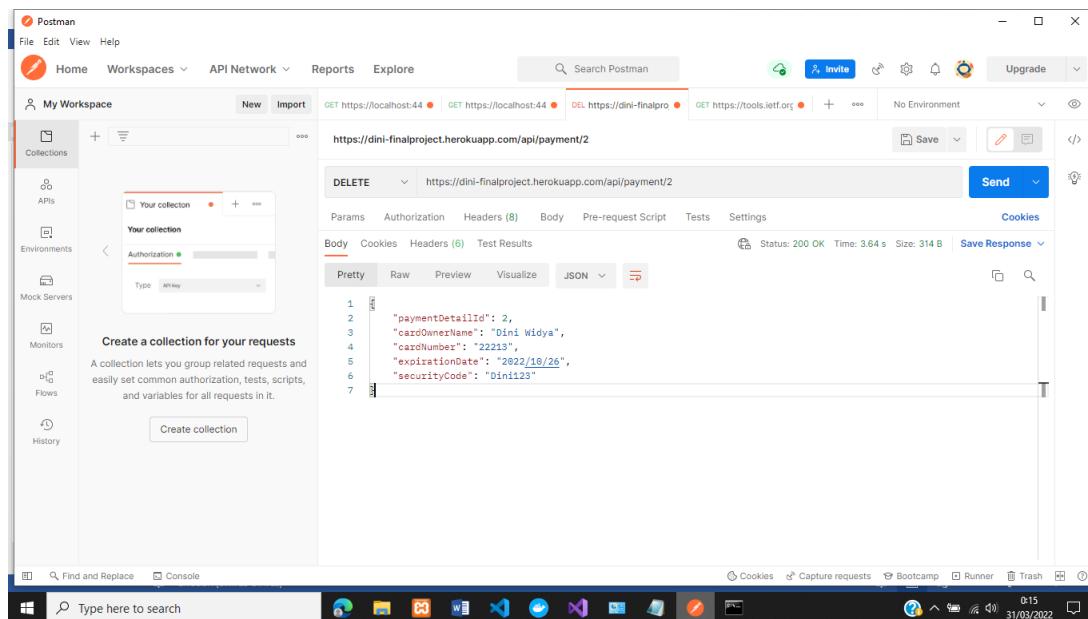
```
1  {
2      "paymentDetailId": 2,
3      "cardOwnerName": "Dini Widya",
4      "cardNumber": "22213",
5      "expirationDate": "2022/10/26",
6      "securityCode": "Dini123"
7 }
```

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>



The screenshot shows the Postman application interface. The sidebar and collection structure are identical to the previous screenshot. A new request is selected, showing a DELETE method directed at <https://dini-finalproject.herokuapp.com/api/payment/2>. The response status is 200 OK, with a response body containing the same JSON data as the previous screenshot:

```
1  {
2      "paymentDetailId": 2,
3      "cardOwnerName": "Dini Widya",
4      "cardNumber": "22213",
5      "expirationDate": "2022/10/26",
6      "securityCode": "Dini123"
7 }
```

((Cek GET Kembali))

The screenshot shows the Postman application interface. On the left, the sidebar includes 'My Workspace' with 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. A 'Create a collection for your requests' section is present. In the main area, a collection named 'Your collection' is expanded, showing a single item 'Your collection'. A request card for 'https://dini-finalproject.herokuapp.com/api/payment' is selected, showing a GET method. The response body is displayed in JSON format:

```
1  {
2   "paymentDetailId": 3,
3   "cardOwnerName": "Wiwik Supiyati",
4   "cardNumber": "22348",
5   "expirationDate": "2022/09/21",
6   "securityCode": "Wiwik123"
7 },
8 [
9   {
10    "paymentDetailId": 4,
11    "cardOwnerName": "Dicky Sastro",
12    "cardNumber": "22234",
13    "expirationDate": "2021/10/12",
14    "securityCode": "Dicky123"
15 },
16 {
17   "paymentDetailId": 6,
18   "cardOwnerName": "Mega Utami",
```

The status bar at the bottom indicates 'Status: 200 OK Time: 2.48 s Size: 578 B' and 'Save Response'.

ID = 2 sudah tidak ada di dalam database.

GET WEATHERFORECAST

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

The screenshot shows the Postman application interface. The sidebar is identical to the previous screenshot. In the main area, a collection named 'My first collection' is expanded, showing two folders: 'First folder inside collection' and 'Second folder inside collection', each containing several requests. A request card for 'https://dini-finalproject.herokuapp.com/WeatherForecast' is selected, showing a GET method. The response body is displayed in JSON format:

```
1 [
2   {
3     "date": "2022-03-31T14:21:44.4435776+00:00",
4     "temperatureC": 19,
5     "temperatureF": 66,
6     "summary": "Cool"
7   },
8   {
9     "date": "2022-04-01T14:21:44.4435817+00:00",
10    "temperatureC": 3,
11    "temperatureF": 37,
12    "summary": "Freezing"
13   },
14   {
15     "date": "2022-04-02T14:21:44.4435821+00:00",
16     "temperatureC": 45,
17     "temperatureF": 112,
18     "summary": "Chilly"
19   }
20 ],
```

The status bar at the bottom indicates 'Status: 200 OK Time: 258 ms Size: 689 B' and 'Save Response'.

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