Langkah2 Integrasi BackEnd

Sebelum kita membicarakan tentang bagaimana untuk membuat project BackEnd API dengan memanfaatkan BackEnd yang lama, maka lebih baik kita melihat hubungan secara keseluruhan antara backend dan frontend yang nantinya akan kita bangun

# Relasi antara Project Common, BackEnd dan FrontEnd

## Alur Data dari FrontEnd ke BackEnd

Semua program FrontEnd akan memanggil BackEnd melalui API service yg diwakili oleh masing2 project Service yg berisi controller untuk setiap program nya dan akan diteruskan ke Class yang ada di project Back untuk masing2 program nya

BackEnd

FrontEnd

* Class 001
* Class 002
* Class 003
* Controller 001
* Controller 002
* Controller 003
* Program 001
* Program 002
* Program 003

## Hubungan secara Class antara BackEnd dan FrontEnd

Controller

Back

Using

Using

Using

Using

Common

Front

# Project yang terbentuk untuk setiap BusinessProcess

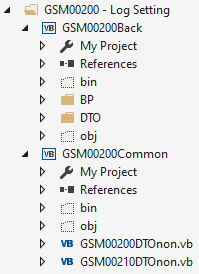
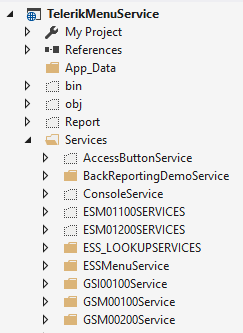
Untuk memberikan contoh terhadap perubahan yg terjadi dari tehnologi lama menggunakan WCF menjadi tenologi baru dengan menggunakan API, maka kita dapat melihat terlebih dahulu project yg dibentuk untuk yg lama dan yang baru dengan mengambil contoh sebuah program yaitu GSM00200.

Secara umum setiap program di back end akan terbagi menjadi 3 project

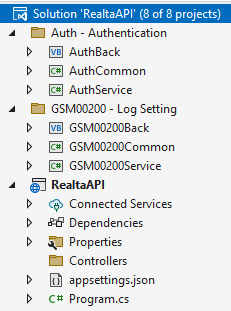
1. Project Common
   1. Perupakan project yg berisi semua hal yang dipakai bersama antara project service dan project front
   2. Project tsb biasanya berisi DTO dari parameter dan result, juga perjanjian enumeration yg dipakai antara FrontEnd dan BackEnd
2. Project Back
   1. Merupakan project yg berisi class2 bisnis proses, dimana pada bagian ini akan terhubung dengan database secara langsung
3. Project Service
   1. Merupakan project yang akan dipanggil oleh FrontEnd, project service ini merupakan gabungan semua service, jadi project service tidak akan berdiri sendiri untuk setiap program, tetapi semua program service akan bergabung menjadi satu project.

## Perbandingan antara project tehnologi lama dan baru

Project tehnologi lama (WCF)



Project tehnologi baru (API)

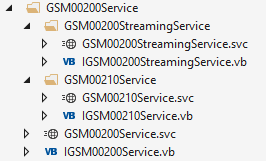


1. GSM00200Common
   1. Merupakan project yg berisi
      1. EntityModel (DTO)
      2. Parameter Model
      3. Result Model
      4. Interface
         1. Dipakai untuk mempermudah bagian front untuk memanggil API yg ada di Back
   2. Project ini merupakan project yg mempunyai class2 yg digunakan oleh Back API dan Front
2. GSM00200Back
   1. Merupakan project yg berisi Business Process
   2. Merupakan project Back yg digunakan oleh Controller
3. GSM00200Service
   1. Merupakan project Back yg berisi Controller, dimana di dalam controller tsb kita mengimplementasikan interface yg ada di GSM00200Common

Controller nantinya akan memanggil class Back yg ada di GSM00200Back

# Langkah2 yg dibuat untuk melakukan konversi Back

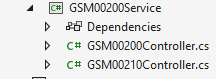
## Lihat Services yg sdh ada dan rancang project service yg baru



Service yg lama terdiri dari 2 penomoran service yaitu gsm00200 dan gsm00210, dimana masing2 penomoran mempunyai service yg normal dan service streaming

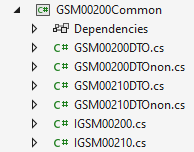
Untuk pembuatan Service yg baru, maka kita akan membuat

1. Sebuah Service
2. Dua buah controller



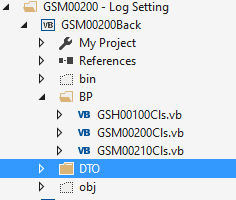
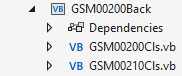
## Pembuatan Project Common

Untuk membuat project service yg baru, maka kita harus membuat terlebih dahulu project common nya, sehingga nanti nya interface yg kita buat dapat dipakai oleh masing2 controller.

1. Kita akan menggabungkan StreamingService(GSM00200StreamingService) dengan Service yg normal (GSM00200Service) menjadi controller yg sama untuk masing2 nomor misalkan (GSM00200 berbeda dengan GSM00210)
2. Kita catat semua Entity dan kita pisahkan interface nya berdasarkan class back yg dipakai
   1. Interface
      1. IGSM00200 (gabungan dari IGSM00200Services dan IGSM00200StreamingServices)
      2. IGSM00210 (dari ISM00210Services)
   2. Entity
      1. GSM00200DTOnon
      2. GSM00200DTO
      3. GSM00210DTOnon
      4. GSM00210DTO
3. Kita buat Common nya yg akan berisi semua Entity dan interface
   1. Buat Project Common
      1. Nama project disesuaikan dengan business process nya
         1. GSM00200Common
      2. dengan menggunakan C# Library
      3. dengan menggunakan Standard 2.1
      4. Reference Assembly R\_CommonFrontBackAPI
         1. Digunakan untuk Result DTO
         2. Digunakan untuk Interface inherit dari
   2. Entity
      1. Semua Entity kalau ada inherits R\_DTOBase tidak dipakai lagi
      2. GSM00200DTOnon
         1. Ubah internal class menjadi public class
         2. Dipakai oleh GSM00200ServiceStreaming
      3. GSM00200DTO
         1. Ubah internal class menjadi public class
         2. Dipakai oleh GSM00200Service
      4. GSM00210DTO
         1. Ubah internal class menjadi public class
         2. Dipakai oleh gsm00210Service
      5. GSM00210DTOnon
         1. Ubah internal class menjadi public class
         2. Dipakai oleh gsm00200ServiceStreaming (kita akan masukkan kedalam interface GSM00210)
   3. Parameter DTO
      1. Kita dapat melihat dari Interface Service Regular (IGSM00200Service dan IGSM00210Service) biasanya akan meng-inherits dari R\_IServicebase
      2. Untuk semua Result dari method yg meng-implement R\_IServicebase tidak perlu kita buatkan Parameter DTO nya
         1. Method2
            1. Svc\_R\_GetRecord
            2. Svc\_R\_Save
            3. Svc\_R\_Delete
      3. Nantinya akan di handle oleh implementasi R\_IServiceCRUDBase
      4. Nama Parameter DTO disesuaikan dengan nama method nya
   4. Result DTO
      1. Kita dapat melihat dari Interface Service Regular (IGSM00200Service dan IGSM00210Service) biasanya akan meng-inherits dari R\_IServicebase
      2. Untuk semua Result dari method yg meng-implement R\_IServicebase tidak perlu kita buatkan DTO Result nya
         1. Method2
            1. Svc\_R\_GetRecord
            2. Svc\_R\_Save
            3. Svc\_R\_Delete
         2. Nantinya akan di handle oleh implementasi R\_IServiceCRUDBase
      3. Result DTO penamaan nya mengikuti Entity nya ditambahkan dengan ResultDTO
      4. Untuk hasil dari streaming kita akan langsung memakai entity nya
   5. Interface
      1. Menggabungkan Interface berdasarkan class Back nya
      2. Semua yg bersifat streaming akan memakai method asynch
      3. IGSM00200
         1. Ubah internal menjadi public
         2. Inherits R\_IServiceCRUDBase<GSM00200DTO>
         3. IAsyncEnumerable<GSM00200DTOnon> GetTableHDList()
            1. Nantinya akan diimplementasikan secara async di controller
      4. IGSM00210
         1. Ubah internal menjadi public
         2. Inherits R\_IServiceCRUDBase<GSM00210DTO>
         3. IAsyncEnumerable<GSM00210DTOnon> GetTableDTList()
            1. Nantinya akan diimplementasikan secara async di controller
4. 

## Pembuatan Project Back

Kita buat project Back yang berisi semua business process dengan menggunakan project back yg lama

1. Buat Project Back
   1. Nama project disesuaikan dengan business process nya
      1. GSM00200Back
   2. Menggunakan VB Library Linux (untuk menampung back class yg mempunyai language VB)
   3. Menggunakan net 6.0
   4. Reference Assembly yg digunakan oleh class backend yg lama
      1. R\_APICommon
      2. R\_APIBackEnd
      3. R\_CommonFrontBackAPI
         1. Digunakan untuk enumeration dari CRUD method yg kita inherit dari R\_BusinessObject
   5. Reference Project Common
      1. GSM00200Common
2. Copy-kan semua class yang ada di folder BP
   * 1. 
     2. Dalam hal ini GSM00200Cls.vb dan GSM00210Cls.vb
   1. Adjust GSM00200Cls.vb maupun GSM00210Cls.vb
      1. Import R\_CommonFrontBackAPI
         1. Dipakai untuk enumeration CRUD
      2. Import System.Data
         1. Ada perubahan namespace untuk ConnectionState dari System.Data.Common menjadi System.Data
      3. Ubah enumeration R\_Common.eCRUDMode menjadi eCRUDMode
   2. Restore NuGet
      1. Seringkali kalau kita meng-copy class dari project yg lain, maka pada waktu compile akan terjadi error Assets file “....project.assets.json not found”
      2. Buka Nuget Console
         1. Tools > NuGet Package Manager > Package Manager Console
      3. Restore dengan perintah dotnet restore di Nuget Console
3. 

## Pembuatan Project Service

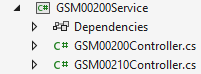
1. Kita buat Service Project yang berisi semua method yg akan di-ekspose ke front
   1. Buat Project Service
      1. Nama project disesuaikan dengan business process nya
         1. GSM00200Service
      2. Menggunakan C# Library
      3. Menggunakan net 6.0
      4. Edit project file dan tambahkan
         1. <ItemGroup>
         2. <FrameworkReference Include="Microsoft.AspNetCore.App" />
         3. </ItemGroup>
      5. Reference Assembly yg digunakan oleh class backend yg lama
         1. R\_APICommon
         2. R\_APIBackEnd
         3. R\_CommonFrontBackAPI
            1. Digunakan untuk enumeration dari CRUD method kalau interface kita inherit dari R\_IServiceCRUDBase
      6. Reference Project Common dan Back
         1. GSM00200Common
         2. GSM00200Back
   2. Buat Class Controller
      1. Class Controller tergantung dari banyak nya class Back
         1. Ubah internal menjadi public
      2. Buat dengan penamaan Business Process ditambahkan dengan Controller
         1. GSM00200Controller
         2. GSM00210Controller
      3. Using
         1. R\_Common
         2. R\_BackEnd
         3. R\_CommonFrontBackAPI
         4. Microsoft.AspNetCore.Mvc
      4. Tambahkan annotation di class
         1. [Route("api/[controller]/[action]")]
         2. [ApiController]
      5. Inherit dan interface
         1. Inherit ControllerBase
         2. Interface sesuai dengan business object interface nya, contohnya
            1. Untuk GSM00200Controller

IGSM00200

* + - * 1. Untuk GSM00210Controller

IGSM00210

* + - 1. Implementasikan semua interface nya sesuai dengan kebutuhan, misalkan tidak ada implementasi R\_ServiceDelete, biarkan isinya “throw new NotImplementedException()”
    1. Tambahkan annotation di method [HttpPost]
    2. Untuk method yg meng-implementasikan IAsyncEnumerable
       1. Tambahkan async di depan nya
       2. Kalau ada pemanggilan untuk context
          1. Ubah dari R\_Utility.R\_GetStreamingContext menjadi R\_Utility.R\_GetStreamingContext<….>
       3. Tidak boleh memakai try block karena akan memakai yied di foreach
  1. GSM00200Controller
     1. Using
        1. GSM00200Common
        2. GSM00200Back
     2. Untuk implementasi nya terjemahkan dari old service
        1. GSM00200Service
        2. GSM00200StreamingService

1. 

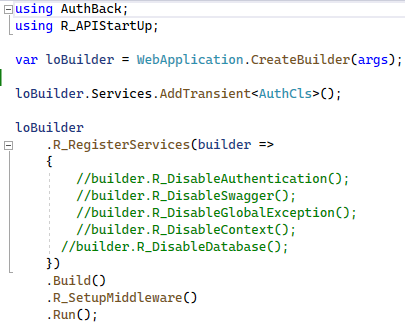
Sebelum kita gabungkan dengan Project API, maka setiap project service dapat kita publish

1. Build dengan command
   1. dotnet build GSM00200Service/GSM00200Service.csproj -c Release
2. Lalu dapat kita publish dengan target output tertentu dengan perintah
   1. dotnet publish GSM00200Service/GSM00200Service.csproj -c release -o ..\dll

Dengan cara ini, maka kita akan mengumpulkan semua service dll dan dependency-nya kedalam suatu folder, dalam hal ini dll folder

## Pembuatan Project API

Kita buat sebuah project API yang akan dipakai untuk menampung semua dll Back project yang nantinya akan di-reference Bersama dengan dll library.

1. Penamaan dan Type Project
   1. Penamaan API\_CRUD
   2. Language C#
   3. Platform Open (Linux)
   4. Type project ASP.NET Core Web API
   5. Framework NET 6.0
   6. HTTP
2. Reference Project
   1. Reference Service Project
      1. AuthService
      2. Project yg diapakai untuk Authentication
3. Reference semua collection assembly baik Back maupun Common
   1. Semua assembly yg ada di back directory sudah termasuk common directory
   2. Reference semua assembly di directory “RuntimeLibrary/Back” semua assembly yg mempunyai awalan R\_ saja
4. Reference Service Assembly
   1. Secara perjanjian ada di directory Dll
   2. Dalam hal ini GSM00200Service.dll
5. Perubahan pada Class Program.cs
   1. using R\_APIStartUp
   2. Untuk penggunaan StartUp ada 5 hal yg dapat di-disable
      1. Authentication
      2. Swagger
      3. GlobalException
      4. Context
      5. Database
   3. Karena kita akan menggunakan semuanya, maka kita tidak akan melakukan disable
   4. 
6. Penambahan pada AppSetting
   1. Karena kita akan berhubungan dengan database, maka kita akan tambahkan konfigurasi database di bagian AppSettings.json
   2. Kita dapat melakukan setting untuk beberapa database di bagian R\_DBConfigs, semua nya akan dijelaskan pada dokumentasi R\_Db
   3. 