

## PEMROGRAMAN LANJUT

### MVC Pattern dan REST API (Web Service)

Kamarudin, M.Kom

[kamarudin@amikom.ac.id](mailto:kamarudin@amikom.ac.id)

<http://coding4ever.net/>

# Persiapan Mengakses REST API Menggunakan C#

- ✓ Buka project *PerpustakaanAppMVC* (folder sample project)
- ✓ Jalankan project *PerpustakaanAppMVC* (shortcut F5)
- ✓ Menginstall library *RestSharp* menggunakan NuGet Package.

Apa itu Nuget? NuGet adalah paket manager untuk platform pengembangan menggunakan Microsoft .NET.

Dengan menggunakan NuGet akan mempermudah kita menginstall library siap pakai yang dikembangkan oleh Microsoft atau pihak ketiga.

# Apa Itu RestSharp?

RestSharp adalah library pihak ketiga yang akan mempermudah kita untuk mengakses REST API (web service).

Dengan menggunakan RestSharp data/informasi yang dihasilkan oleh REST API yang secara default menggunakan format JSON bisa dengan mudah di konversi ke dalam sebuah objek.

## Objek JSON

```
{  
  "npm": "15.02.9011",  
  "nama": "ABDUL HARI SUHUD",  
  "alamat": "Jl. Pengadilan Lama No. 253"  
}
```



## Objek C#

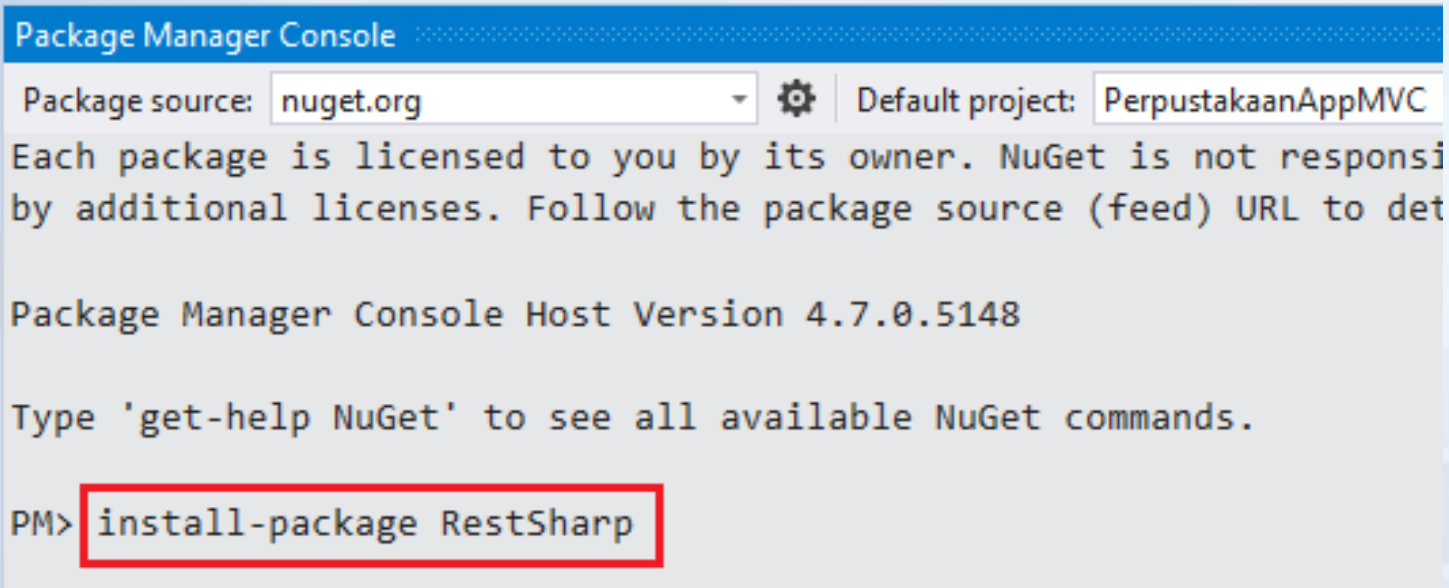
```
// membuat objek mahasiswa  
var mhs = new Mahasiswa();  
mhs.npm = "15.02.9011";  
mhs.nama = "ABDUL HARI SUHUD";  
mhs.alamat = "Jl. Pengadilan Lama No. 253";
```

# Latihan 1:

## Menginstall RestSharp

Langkah-langkah:

1. Klik menu Tools -> NuGet Package Manager -> Package Manager Console.
2. Kemudian ketik perintah *install-package RestSharp* pada console NuGet Package Manager.



```
Package Manager Console
Package source: nuget.org
Default project: PerpustakaanAppMVC
Each package is licensed to you by its owner. NuGet is not responsible for
by additional licenses. Follow the package source (feed) URL to det
Package Manager Console Host Version 4.7.0.5148
Type 'get-help NuGet' to see all available NuGet commands.
PM> install-package RestSharp
```

# Latihan 1 (Lanjutan): Menginstall RestSharp

## Package Manager Console

Package source:

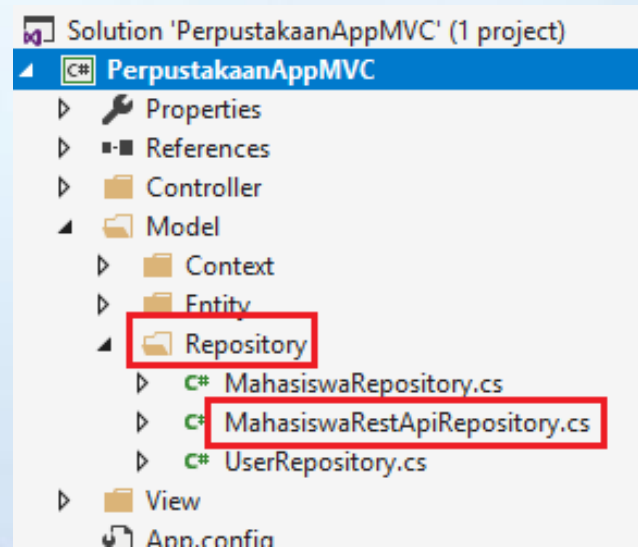


Default project:

```
Attempting to resolve dependencies for package 'RestSharp.106.6.10' with  
Resolving dependency information took 0 ms  
Resolving actions to install package 'RestSharp.106.6.10'  
Resolved actions to install package 'RestSharp.106.6.10'  
Retrieving package 'RestSharp 106.6.10' from 'nuget.org'.  
Adding package 'RestSharp.106.6.10' to folder 'E:\Praktikum\01 Pemrograman  
Added package 'RestSharp.106.6.10' to folder 'E:\Praktikum\01 Pemrograman  
Added package 'RestSharp 106.6.10' to 'packages.config'  
Successfully installed 'RestSharp 106.6.10' to PerpustakaanAppMVC  
Executing nuget actions took 1.97 sec  
Time Elapsed: 00:00:02.2410146  
PM> |
```

## Latihan 2: Menambahkan Class Repository

Tambahkan class baru dengan nama *MahasiswaRestApiRepository* di folder Repository.



Kemudian tambahkan beberapa method untuk operasi CRUDnya.

# Latihan 2 (Lanjutan): Menambahkan Class Repository

```
public class MahasiswaRestApiRepository
{
    public int Create(Mahasiswa mhs)
    {
        throw new NotImplementedException();
    }

    public int Update(Mahasiswa mhs)
    {
        throw new NotImplementedException();
    }

    public int Delete(Mahasiswa mhs)
    {
        throw new NotImplementedException();
    }

    public List<Mahasiswa> ReadAll()
    {
        throw new NotImplementedException();
    }

    public List<Mahasiswa> ReadByNama(string nama)
    {
        throw new NotImplementedException();
    }

    public Mahasiswa ReadByNpm(string npm)
    {
        throw new NotImplementedException();
    }
}
```

# Latihan 3:

## Melengkapi Kode untuk Method ReadAll

Method *ReadAll* digunakan untuk menampilkan semua data mahasiswa. Adapun informasi alamat REST API yang diakses sebagai berikut:

✓ Base Url

- <http://latihan.buatdevelop.com:5000/>

✓ Endpoint

- api/mahasiswa

✓ HTTP Method

- GET



# Latihan 3 (Lanjutan): Melengkapi Kode untuk Method ReadAll


```
public List<Mahasiswa> ReadAll()
{
    string baseUrl = "http://latihan.buatdevelop.com:5000/";
    string endpoint = "api/mahasiswa";

    // membuat objek rest client
    var client = new RestClient(baseUrl);

    // membuat objek request
    var request = new RestRequest(endpoint, Method.GET);

    // kirim request ke server
    var response = client.Execute<List<Mahasiswa>>(request);

    return response.Data;
}
```



Berisi collection dari objek mahasiswa

## Latihan 4:

# Revisi Method ReadAll di Class MahasiswaController

```
public List<Mahasiswa> ReadAll()
{
    // membuat objek collection
    List<Mahasiswa> list = new List<Mahasiswa>();

    // membuat objek context menggunakan blok using
    /*
    using (DbContext context = new DbContext())
    {
        // membuat objek dari class repository
        _repository = new MahasiswaRepository(context);

        // panggil method GetAll yang ada di dalam class repository
        list = _repository.ReadAll();
    }
    */

    // membuat objek dari class MahasiswaRestApiRepository
    var repo = new MahasiswaRestApiRepository();
    list = repo.ReadAll();

    return list;
}
```

## Latihan 5: Melengkapi Kode untuk Method ReadByNama

Method *ReadByNama* digunakan untuk melakukan pencarian data mahasiswa berdasarkan nama. Adapun informasi alamat REST API yang diakses sebagai berikut:

- ✓ Base Url
  - <http://latihan.buatdevelop.com:5000/>
- ✓ Endpoint
  - api/mahasiswa?**name**=diisi nama mhs
- ✓ HTTP Method
  - GET

**Silahkan dicoba sendiri 😊**

# Latihan 6:

## Revisi Method ReadByNama di Class MahasiswaController

```
public List<Mahasiswa> ReadByNama(string nama)
{
    // membuat objek collection
    List<Mahasiswa> list = new List<Mahasiswa>();

    // membuat objek context menggunakan blok using
    /*
    using (DbContext context = new DbContext())
    {
        // membuat objek dari class repository
        _repository = new MahasiswaRepository(context);

        // panggil method GetByNama yang ada di dalam class repository
        list = _repository.ReadByNama(nama);
    }
    */

    // membuat objek dari class MahasiswaRestApiRepository
    var repo = new MahasiswaRestApiRepository();
    list = repo.ReadByNama(nama);

    return list;
}
```

## Latihan 7: Melengkapi Kode untuk Method Create

Method *Create* digunakan untuk menambahkan data baru. Adapun informasi alamat REST API yang diakses sebagai berikut:

- ✓ Base Url
  - <http://latihan.buatdevelop.com:5000/>
- ✓ Endpoint
  - api/mahasiswa
- ✓ HTTP Method
  - POST
- ✓ Body
  - Objek mahasiswa

# Latihan 7 (Lanjutan): Melengkapi Kode untuk Method Create

```
public int Create(Mahasiswa mhs)
{
    var baseUrl = "http://latihan.buatdevelop.com:5000/";
    var endpoint = "api/mahasiswa";

    // membuat objek rest client
    var client = new RestClient(baseUrl);


    // membuat objek request
    var request = new RestRequest(endpoint, Method.POST);

    // tambahkan objek mahasiswa ke dalam body
    request.AddJsonBody(mhs);

    // request ke server
    var response = client.Execute(request);

    // cek nilai content, jika 1 berhasil selain itu gagal
    var result = response.Content.IndexOf("1") > 0 ? 1 : 0;

    return result;
}
```



JSON

```
{
  "result": 1
}
```

## Latihan 8:

# Revisi Method Create di Class MahasiswaController

```
// membuat objek context menggunakan blok using
```

```
/*  
using (DbContext context = new DbContext())  
{  
    // membuat objek class repository  
    _repository = new MahasiswaRepository(context);  
  
    // panggil method Create class repository untuk menambahkan data  
    result = _repository.Create(mhs);  
}  
*/
```

```
var repo = new MahasiswaRestApiRepository();  
result = repo.Create(mhs);
```

```
if (result > 0)  
{  
    MessageBox.Show("Data mahasiswa berhasil disimpan !", "Informasi",  
        MessageBoxButtons.OK, MessageBoxIcon.Information);  
}  
else  
    MessageBox.Show("Data mahasiswa gagal disimpan !!!", "Peringatan",  
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
```

**Selesai 😊**