MENGENAL .NET MAUI

VIDEO: <https://www.youtube.com/watch?v=Hh279ES_FNQ&list=PLdo4fOcmZ0oUBAdL2NwBpDs32zwGqb9DY&index=1>

|  |  |
| --- | --- |
| **ENGLISH** | **INDONESIA** |
| If you've been looking for a way to kickstart your journey on building native cross-platform applications for desktop and mobile with.NET MAUI and C#.  Well, you are in the right place. This is the.NET MAUI beginner series.  In this video series, I'm going to take you from start to finish on what.NET MAUI is and how you can start building cross-platform native applications  for desktop and mobile today.  The first thing you might be asking yourself, well, what is.NET MAUI?    What is the multi-platform app UI?  Which is a brand new framework from Microsoft for building beautiful native and performance cross-platform desktop and mobile apps for iOS, Android, Mac, and Windows.  You can do this all from a shared single code base.  The best part of this is that there's great tooling in Visual Studio to make you  super productive right from the beginning.  In this beginner series, I'm going to show you not only what is in a.NET MAUI project, but we're going to walk through the main parts of creating user interface, doing advanced MVVM architecture and data binding.  We're going to take a look at platform integrations, navigation and so much more.  Now, one of my favorite parts of.NET MAUI is that you can access native APIs completely in C#.  You can also build native cross-platform user interfaces directly in XAML, which is an XML-based markup, which has nice features like data binding or completely in C#.  It's great because.NET MAUI is flexible, enabling you to build these  user interfaces in many ways.  When your application starts up, whatever code you create, let's say a button, .NET MAUI will generate and render the native control.  This means on iOS you're getting UIkit, on Android you're getting Android Widgets, on Mac you're leveraging Mac Catalyst, enabling you to run your iOS applications directly on Mac devices, but also light up native Mac  features and of course on Windows leveraging the latest Windows App SDK and WinUI 3.  But you as a developer get to write everything in one shared code base.  Additionally, besides creating user interface, you'll have all of your  business logic that shared too. Such as Models, View Models, RESTful Service Calls, Databases, and so much more.  Now, like I mentioned earlier, you can access those native APIs directly in C# for the different platforms which I'm going to show you in this beginner series, but more importantly, .NET MAUI introduces a whole bunch of platform APIs that have been abstracted into a common surface for you to code against.  This means common things such as connectivity, geolocation, sensors and so much more are available in one common API, just like the user interface is as well.  Now, like I said, there's great tooling in Visual Studio and I'm going to walk you through from start to finish getting installed up and running and good to go.  I hope you join me on this adventure of building native cross-platform desktop and mobile apps with .NET MAUI | Jika anda ingin membuat aplikasi cross-platform untuk desktop dan mobile dengan .NET MAUI dan C#.  Nah, di sini tempatnya. Ini adalah seri pemula .NET MAUI.  Dalam video ini, saya akan menjelaskan dari awal hingga akhir seputar .NET MAUI dan bagaimana membuat aplikasi lintas platform untuk desktop dan seluler.  Yang pertama. Apa itu .NET MAUI?  Apa yang dimaksud dengan UI aplikasi multi-platform?  .NET MAUI itu framework baru dari Microsoft untuk membuat aplikasi desktop dan seluler lintas platform yang indah dan berkinerja tinggi untuk iOS, Android, Mac, dan Windows.  Anda dapat melakukan ini semua berdasarkan kode tunggal yang digunakan bersama.  Bagian terbaik dari ini adalah fitur yang hebat di Visual Studio untuk membuat Anda sangat produktif sejak awal.  Dalam seri pemula ini, saya akan menunjukkan kepada Anda tidak hanya apa yang ada di proyek .NET MAUI, tetapi kita akan membahas bagian-bagian utama dalam membuat antarmuka pengguna, melakukan arsitektur MVVM tingkat lanjut, dan pengikatan data.  Kita akan melihat integrasi platform, navigasi, dan banyak lagi.  Sekarang, salah satu bagian favorit saya dari .NET MAUI adalah Anda dapat mengakses API asli sepenuhnya di C#.  Anda juga dapat membuat antarmuka pengguna lintas platform secara langsung di XAML, yang merupakan markup berbasis XML, yang memiliki fitur-fitur bagus seperti pengikatan data atau sepenuhnya di C#.  Ini bagus karena .NET MAUI fleksibel, memungkinkan Anda untuk membangun antarmuka pengguna ini dengan berbagai cara.  Ketika aplikasi Anda dijalankan, kode apa pun yang Anda buat, katakanlah sebuah tombol, .NET MAUI akan menghasilkan dan merender kontrol asli.  Ini berarti di iOS Anda mendapatkan UIkit, di Android Anda mendapatkan Android Widget, di Mac Anda memanfaatkan Mac Catalyst, yang memungkinkan Anda untuk menjalankan aplikasi iOS Anda secara langsung di perangkat Mac, tetapi juga menggunakan fitur-fitur asli Mac dan tentu saja di Windows yang memanfaatkan Windows App SDK dan WinUI 3 terbaru.  Tetapi Anda sebagai pengembang dapat menulis semuanya berdasarkan satu kode yang digunakan bersama.  Selain itu, selain membuat antarmuka pengguna, Anda juga akan berbagi semua logika bisnis Anda. Seperti Model, View Model, Panggilan Layanan RESTful, Database, dan banyak lagi.  Sekarang, seperti yang saya sebutkan sebelumnya, Anda dapat mengakses API asli tersebut secara langsung di C# untuk berbagai platform yang akan saya tunjukkan kepada Anda dalam seri pemula ini, tetapi yang lebih penting lagi, .NET MAUI memperkenalkan sejumlah besar API platform yang telah dirangkum untuk digunakan ketika mengkode.  Ini berarti hal-hal umum seperti konektivitas, geolokasi, sensor, dan banyak lagi tersedia dalam satu API yang sama, seperti halnya antarmuka pengguna.  Sekarang, seperti yang saya katakan, ada banyak sekali fitur keren di Visual Studio dan saya akan memandu Anda dari awal sampai akhir untuk menginstal dan menjalankannya dan siap digunakan.  Saya harap Anda bergabung dengan saya dalam petualangan membangun aplikasi desktop dan seluler lintas platform dengan .NET MAUI. |

Note:

1. .Net MAUI Project

2. Creating UI

3. Advanced MVVM Architecture & Data Binding

4. Platform Integration

5. Navigation

Shared Business Logic

1. Models

2. View Models

3. RESTful Service Calls

4. Databases