



INDONESIA  
.NET DEVELOPER COMMUNITY



Seri Belajar

# Microsoft Azure

Pengenalan Azure Web Apps

Erick Kurniawan, Microsoft MVP Microsoft Azure

## Kata Pengantar

Penulis mendedikasikan buku ini kepada Komunitas Microsoft User Group Indonesia yang selalu memberi dukungan dan dorongan kepada penulis untuk terus memiliki semangat untuk berkarya dan berbagi. Tidak lupa penulis juga mengucapkan terima kasih kepada Microsoft Indonesia atas dukungan dan dedikasinya pada komunitas MUGI sehingga kami bisa selalu terus berkarya.

Penulis juga mengucapkan terima kasih kepada sahabat penulis M.Reza Faisal yang sudah menyumbangkan desain cover yang sangat indah pada buku ini.

Yogyakarta, 2 Maret 2016

Erick Kurniawan

## Tentang Penulis



Penulis menyelesaikan pendidikan S1 jurusan Teknik Informatika di Universitas Kristen Duta Wacana, kemudian melanjutkan pendidikan S2 di jurusan Ilmu Komputer Universitas Gajah Mada.

Saat ini penulis bekerja sebagai dosen, konsultan dan professional trainer. Selain mengajar, menulis buku dan menjadi konsultan, penulis juga aktif sebagai pembicara seminar, kuliah umum dan memberikan training dengan topik seputar teknologi terbaru terutama yang berhubungan dengan teknologi web, cloud dan aplikasi mobile.

Penulis juga merupakan founder dari Actual Training (<http://actual-training.com>) dan Cludemia (<http://cludemia.co>). Actual Training menyediakan jasa pelatihan professional di bidang Teknologi Informasi yang berlokasi di Yogyakarta. Cludemia merupakan portal pembelajaran online berupa video tutorial yang menyediakan konten pembelajaran terbaik seputar teknologi informasi berbahasa indonesia.

Saat ini penulis aktif di komunitas MUGI (Microsoft User Group). Penulis juga memiliki beberapa sertifikasi professional seperti MCTS (Microsoft Certified Technology Specialist), MCPD (Microsoft Certified Professional Developer), dan MCT (Microsoft Certified Trainer). Penulis juga pernah mendapatkan penghargaan Microsoft MVP (Most Valuable Professional) pada tahun 2009 sd 2015.

Untuk menghubungi penulis anda dapat mengirimkan email ke [erick@actual-training.com](mailto:erick@actual-training.com).  
Anda dapat mengunjungi website penulis pada alamat <http://erickkurniawan.net> dan <http://actual-training.com>.

## **Daftar Isi**

Kata Pengantar.....	2
Tentang Penulis.....	3
Daftar Isi.....	4
BAB 1 Pengenalan Azure Website / Azure App Services .....	1
Azure Website / Azure Web App .....	2
Azure Mobile Services.....	2
Bab 2 - Medaftar Akun Azure dan Memasang Web CMS pada Layanan Azure Web Apps .....	4
Bagaimana Memperoleh Akun Microsoft Azure?.....	4
Memasang Aplikasi CMS Open Source di Azure .....	5
Langkah 1: Membuat MySQL Database .....	5
Langkah 2: Memasang WordPress pada layanan Azure Web.....	7
Bab 3 – PHP & MySQL dengan Layanan Azure Web Apps.....	10
Membuat Database MySQL di Azure .....	10
Membuat Aplikasi Web dengan PHP .....	14
Bab 4 – ASP.NET MVC & Microsoft Azure.....	22
Membuat Project ASP.NET MVC.....	22
Membuat Web Site Instance di Windows Azure .....	24
Upload Project di Visual Studio ke Microsoft Azure .....	26
Menggunakan SQL Azure dan ASP.NET MVC.....	29
Membuat Database pada Windows Azure .....	31
Autentikasi Aplikasi ASP.NET MVC dengan Azure Active Directory .....	34
Membuat Azure Active Directory .....	34
Bab 5 - Windows Azure Mobile Services.....	41
Memulai Membuat Azure Mobile Services .....	41
Windows Store Client .....	45

Mengakses REST Services dari Table di Mobile Services .....	50
Menambahkan Data dan Table Baru pada Azure Mobile Services.....	53
Menambahkan Script Pada Proses CRUD .....	58
Menggunakan Objek Request pada Script.....	60
Menambahkan Field baru dengan Script.....	62
Menggunakan Query Object pada SCRIPT .....	64
Bab 6 -Identity dengan Azure Mobile Services .....	66
Membuat Twitter Identity Provider.....	66
Membuat Facebook Identity Provider .....	69
Menambahkan Restriction Permission pada Table .....	70
Bab 7 - Menggunakan Push Notification .....	72

## BAB 1 Pengenalan Azure Website / Azure App Services

Untuk membangun infrastruktur teknologi informasi dibutuhkan banyak biaya dari sisi biaya pengadaan perangkat keras, lisensi perangkat lunak, biaya pemantauan, pengaturan, dan perawatan infrastruktur teknologi informasi. Cloud Computing merupakan solusi dari masalah tersebut. Teknologi Cloud Computing yang ada sekarang menawarkan layanan infrastruktur teknologi informasi untuk perusahaan, karena itu perusahaan tidak perlu lagi membangun infrastruktur sendiri, cukup dengan menyewa layanan yang disediakan oleh perusahaan penyedia layanan seperti Amazon, Microsoft, atau Google.

Kelebihan penggunaan layanan Cloud Computing dibandingkan dengan *on-premise* atau infrastruktur yang dibangun sendiri adalah dalam hal biaya, skalabilitas, perawatan, dan kehandalan layanan. Dengan menggunakan layanan Cloud Computing, sumber daya yang dapat digunakan menjadi tidak terbatas. Layanan Cloud Computing menyediakan fasilitas untuk menambah atau mengurangi sumber daya yang digunakan sesuai dengan kebutuhan tanpa harus menambahkan atau mengganti infrastruktur fisik yang digunakan.

*Cloud Computing* sendiri dibedakan menjadi tiga jenis layanan yang mempunyai karakteristik yang berbeda yaitu:

- *Infrastructure as a Services* (IaaS): IaaS menyediakan layanan infrastruktur dasar. Jadi pengguna membuat server berupa VM (Virtual Machine) yang spesifikasi perangkat keras *server* tersebut dapat ditentukan sendiri seperti jumlah *processor*, RAM, Hardisk, Sistem Operasi, dan *bandwidth* yang digunakan. Biaya dari sewa *server* VM yang dibuat berdasarkan fitur perangkat keras dan perangkat lunak yang dipilih oleh pengguna. Contoh dari layanan IaaS adalah Microsoft Azure VM, Amazon EC2, Rackspace Cloud, dan Open Stack.
- *Platform as a Service* (PaaS): PaaS menyediakan layanan pada level platform, seperti layanan *hosting* aplikasi, ataupun layanan basis data. Dengan PaaS pengguna tidak lagi direpotkan dengan instalasi sistem operasi, web server, server basis data, dan konfigurasi level mesin yang harus dilakukan. Pengguna dapat mengunggah aplikasi melalui panel kontrol yang disediakan, dan dapat memilih paket layanan sesuai dengan kebutuhan. Contoh layanan PaaS adalah Microsoft App Services, Google App Engine, Amazon Elastic Beanstalk, Cloud Foundry, dan Heroku.

- Software as a Services (SaaS): SaaS menyediakan layanan dalam bentuk aplikasi yang ditujukan untuk pengguna akhir (*end-user*). Ada banyak layanan aplikasi SaaS yang tersedia saat ini baik yang berbayar maupun aplikasi gratis. Contoh dari aplikasi SaaS diantaranya: Gmail, Office 365, Google Drive, Google Doc, Dropbox, Google Classroom, Onenote Classnote, dan masih banyak lagi.

Microsoft Azure sebagai salah satu penyedia layanan komputasi awan yang handal menyediakan banyak sekali layanan yang dapat membantu pengembang aplikasi membangun aplikasi secara mudah, cepat, produktif, dan mendukung skalabilitas yang handal.

Pada buku ini akan dibahas penggunaan salah satu layanan Microsoft Azure yang termasuk dalam kategori PaaS yaitu penggunaan Azure Website untuk memasang aplikasi web dan Azure Mobile Services yang dapat digunakan untuk membuat backend services dari aplikasi web atau aplikasi web yang anda buat.

### [Azure Website / Azure Web App](#)

Azure Website atau pada portal yang lebih baru dikenal sebagai Azure Web App merupakan layanan cloud PaaS untuk memasang aplikasi web pada platform Azure. Layanan ini cocok untuk membuat *mission critical app* yang mendukung skalabilitas bisnis, sehingga cocok digunakan oleh perusahaan startup atau SMB (*small medium business*) yang mempunyai produk layanan dengan banyak pengguna. Azure Web App juga memiliki banyak fitur pasti disukai oleh developer diantaranya:

- Mendukung berbagai macam teknologi seperti .NET, Java, PHP, Node.js, dan Python.
- Built-in autoscale dan load balancing.
- High availability dan Reliable.
- Mendukung metode Continuous deployment dengan Git, TFS, GitHub, dan Visual Studio Team Services.
- Mendukung berbagai layanan CMS seperti WordPress, Umbraco, Joomla, dan Drupal.

### [Azure Mobile Services](#)

Selain Websites, Cloud Services, dan VM, Microsoft Azure juga menyediakan layanan khusus bernama Azure Mobile Services. Azure Mobile Services sendiri menyediakan solusi lengkap

untuk layanan backend aplikasi mobile. Azure Mobile Services dapat melayani client dengan platform iOS, Android, Windows Phone, Windows Store, dan HTML. Adapun layanan yang disediakan oleh Azure Mobile Services adalah :

- Penyimpanan Data di Cloud tanpa harus membuat aplikasi backend sendiri, data juga dapat diakses via REST dengan output format JSON.
- Autentikasi : dapat menggunakan akun Facebook, Microsoft, Google, dan Twitter untuk autentikasi pada saat login.
- Push Notification.
- Monitor, Alert, Auto Scale.
- Scheduler.

Dengan berbagai fitur yang disediakan diatas, azure mobile services sangat cocok digunakan sebagai backed dari aplikasi mobile atau bahkan aplikasi web SPA (single page application).

## Bab 2 - Medaftar Akun Azure dan Memasang Web CMS pada Layanan Azure Web Apps

### Bagaimana Memperoleh Akun Microsoft Azure?

Jika anda belum memiliki akun Microsoft Azure, maka anda dapat melakukan pendaftaran terlebih dahulu. Untuk mencoba menggunakan layanan Microsoft Azure, ada beberapa solusi yang dapat anda gunakan.

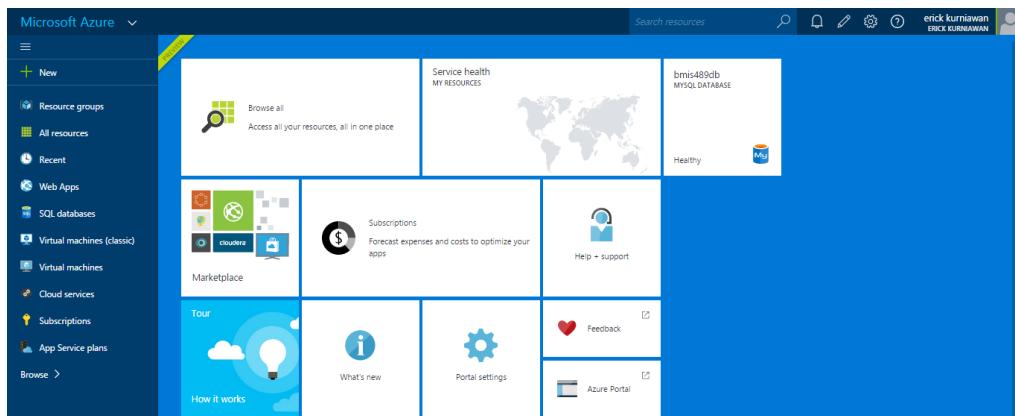
Yang pertama adalah dengan mendaftar program [free trial](#). Dengan mendaftar pada program ini, anda akan mendapatkan kredit senilai 200USD yang dapat anda belanjakan pada semua layanan Azure dalam waktu 1 bulan.

Alternatif kedua berlaku untuk mahasiswa yang memiliki akun [DreamSpark](#). Untuk mahasiswa yang memiliki akun DreamSpark anda dapat mendaftar pada [tautan berikut](#) untuk mendapatkan akun Azure gratis yang dapat digunakan untuk beberapa layanan seperti Azure Web Apps, MySQL database, dan platform kolaborasi Visual Studio Online.

Bagi anda yang memiliki startup dapat mendaftarkan startup anda pada program [BizSpark](#). Program ini menawarkan lisensi Visual Studio Ultimate + MSDN yang didalamnya termasuk fasilitas penggunaan layanan Microsoft Azure.

Atau bagi kalangan akademisi khususnya dosen yang mengajar matakuliah yang menggunakan platform Azure dapat juga mengajukan program [Azure in Education](#). Kalangan akademisi dapat memanfaatkan program ini untuk kepentingan riset atau pengajaran.

Setelah anda mendapatkan akun Microsoft Azure, anda dapat menggunakan layanan tersebut melalui [Microsoft Azure Portal](#).



## Memasang Aplikasi CMS Open Source di Azure

Setelah anda berhasil masuk ke halaman Azure Portal, maka sekarang saatnya untuk memasang aplikasi CMS (Content Management System) pada Azure. Azure sendiri menyediakan banyak pilihan aplikasi yang dikelompokan dalam layanan Marketplace.

Pada artikel ini saya akan menggunakan WordPress sebagai contoh. Seperti kita sudah ketahui bersama bahwa WordPress adalah platform CMS untuk membuat website yang berbasis teknologi open source seperti PHP dan MySQL. Untuk itu anda membutuhkan layanan cloud yang mendukung teknologi tersebut.

### *Langkah 1: Membuat MySQL Database*

Sebelum membuat layanan Azure Web Apps, pertama kali anda harus menyiapkan MySQL database yang nanti akan digunakan oleh aplikasi WordPress yang akan anda buat.

Untuk membuat MySQL database baru, pilih tombol “New” pada Azure Portal, kemudian pilih “Data + Storage”, dan pilih “MySQL Database”.

The screenshot displays two windows from the Microsoft Azure portal. The left window is a sidebar titled 'New' under 'Data + Storage'. It lists various service categories: Compute, Web + Mobile, Data + Storage (which is selected and highlighted in blue), Data + Analytics, Internet of Things, Networking, Media + CDN, Hybrid Integration, Security + Identity, Developer Services, Management, Container Apps, and Marketplace. The right window is titled 'Data + Storage' and shows a list of services: In-Memory Cache, Search, StorSimple, Mongolab, and MySQL Database. The MySQL Database item is highlighted with a red box. The bottom window is titled 'New MySQL Database' and contains the following form fields:

- \* Database Name: actualwpdb
- Database Type: Shared
- \* Location: Southeast Asia
- \* Resource Group: actualwordpress
- Select Existing
- \* Pricing Tier: (dropdown menu)
- Pin to dashboard
- OK button

Pada jendela “New MySQL Database” tambahkan beberapa informasi sebagai berikut:

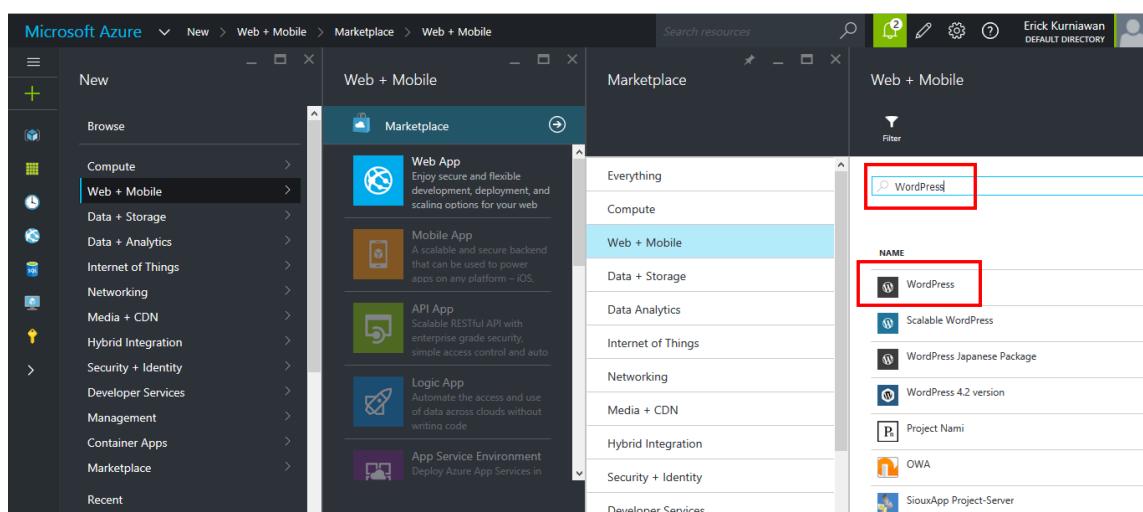
- Masukan nama database yang akan anda buat pada kotak “Database Name”. Ingat nama database yang anda masukan karena akan kita gunakan pada langkah selanjutnya ketika memasang aplikasi WordPress di layanan Azure Web.
- Untuk tipe database pilih “Shared”. Pilihan ini menunjukan bahwa database anda akan disimpan pada server dengan arsitektur multi-tenant, yang berarti server tersebut digunakan juga oleh pengguna lain. Secara default pilihan ini akan aktif karena kita menggunakan free MySQL database.
- Untuk pilihan lokasi anda dapat memilih lokasi *data center* terdekat dengan target pengguna anda. Sebagai contoh saya mengasumsikan bahwa sebagian besar pengakses website yang saya buat berasal dari Indonesia, maka saya akan memilih

South East Asia sebagai lokasi dimana aplikasi saya akan diletakan. Untuk pilihan pricing tier anda dapat menyesuaikan sesuai kebutuhan anda, sebagai permulaan kita dapat memilih paket yang paling murah untuk mengetes aplikasi kita. Selanjutnya pada fase produksi anda dapat meningkatkan paket disesuaikan dengan kebutuhan (banyak pengguna, beban kerja server, dll).

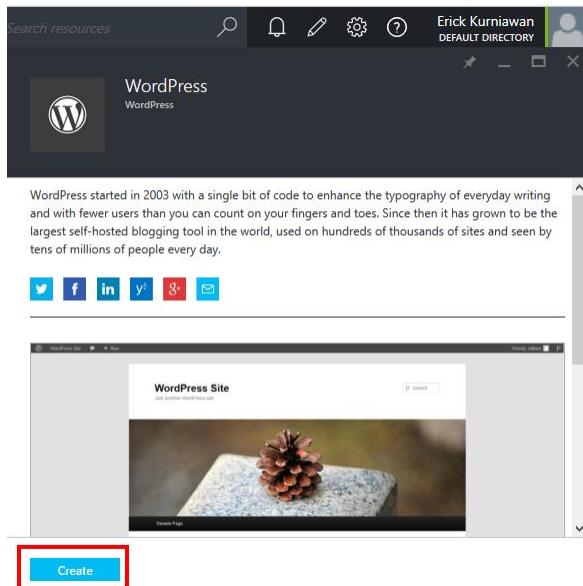
- “Resource group” digunakan untuk mengelompokan sumber daya yang anda buat pada layanan Azure agar anda lebih mudah mengatur, dan mencari sumber daya tersebut.
- Untuk pilihan “Pricing Tier” pilih paket “Mercury” yang merupakan paket gratis dengan spesifikasi paling minimal dari layanan MySQL database di Azure. Untuk mengetahui paket-paket yang disediakan beserta harganya lihat [tautan berikut](#).
- Untuk pilihan “Legal Terms”, pilih tombol “OK”.
- Langkah terakhir, tekan tombol “Create” untuk membuat database baru.

### *Langkah 2: Memasang WordPress pada layanan Azure Web*

Setelah database selesai dipersiapkan, langkah selanjutnya adalah memasang WordPress pada layanan Azure Web. Buka Azure portal, kemudian pilih tombol “New” akan muncul pilihan layanan yang disediakan oleh Azure, anda dapat memilih “Web + Mobile”, kemudian “Marketplace”. Anda dapat mencari aplikasi CMS yang akan anda pasang pada Azure, sebagai contoh anda dapat mengetikkan WordPress pada kotak pencarian. Selain WordPress masih banyak aplikasi open source yang dapat anda pasang seperti Joomla, Moodle, Magento, dan lain sebagainya.

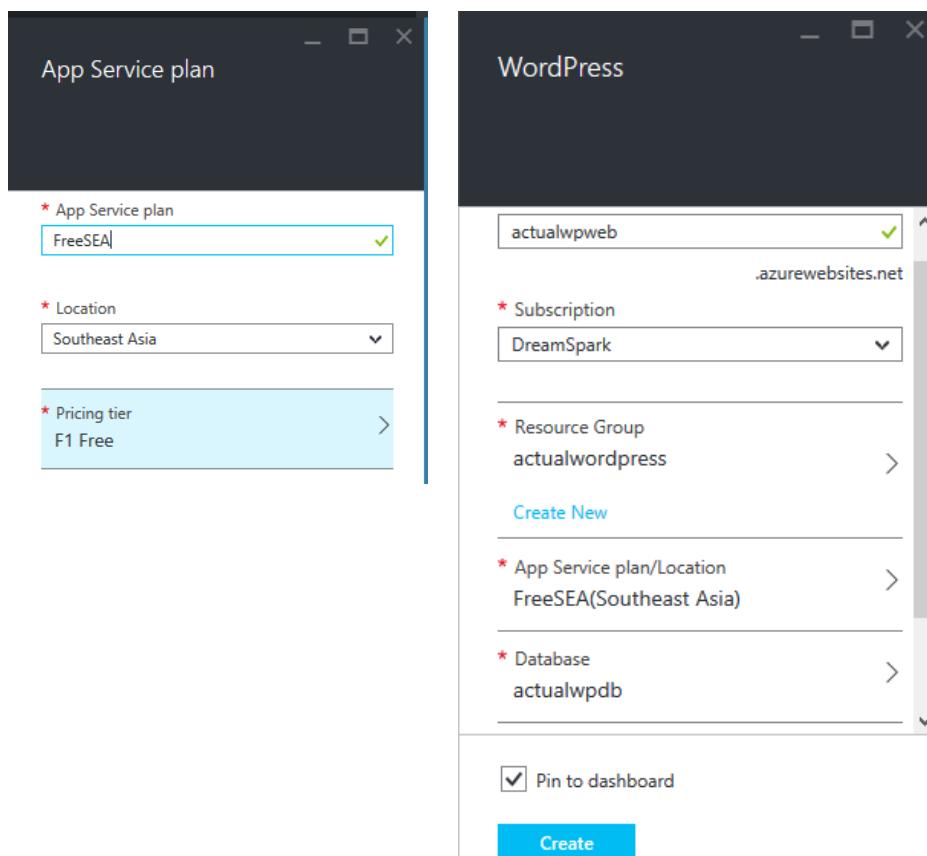


Setelah anda menemukan WordPress pada halaman pencarian selanjutnya pilih aplikasi tersebut, kemudian pilih tombol “Create”.

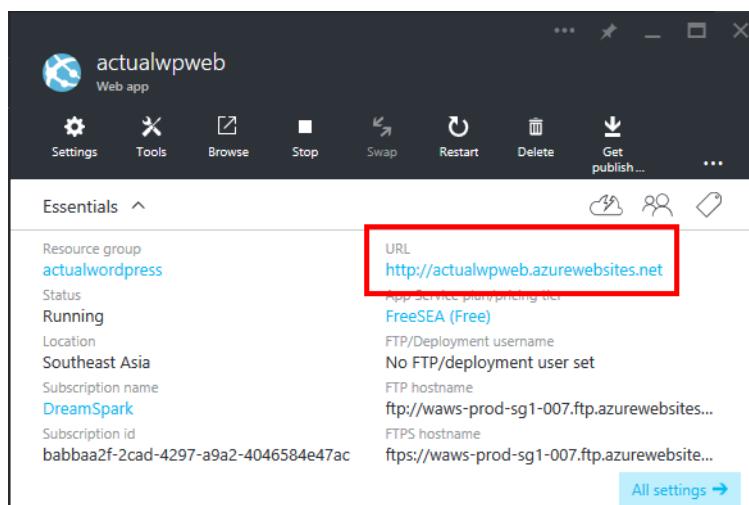


Untuk membuat layanan Azure Web dan memasang aplikasi WordPress pada layanan tersebut tambahkan informasi berikut ini:

- Alamat URL dari aplikasi WordPress yang anda buat. Microsoft Azure menyediakan domain default dengan akhiran `azurewebsites.net`, anda dapat menggunakan alamat domain ini atau menggunakan alamat domain anda sendiri.
- Untuk pilihan subscription anda dapat memilih sesuai dengan subscription yang anda miliki misalnya DreamSpark, atau Visual Studio MSDN.
- Pada pilihan berikutnya masukan resource group yang sama dengan yang sudah anda buat pada saat anda membuat MySQL database.
- Untuk pilihan lokasi dan paket, pilih lokasi data center yang sama dengan database yang sudah anda buat sebelumnya yaitu “South East Asia”, dan pilih paket yang gratis yaitu “F1”.
- Untuk database pilih MySQL database yang sudah anda buat sebelumnya.



Setelah semua informasi dimasukan, tekan tombol “Create”, maka Azure akan menyiapkan semua sumber daya yang dibutuhkan seperti database dan layanan Azure web apps, kemudian dilakukan proses deployment aplikasi WordPress kedalam layanan Azure web apps. Tunggu beberapa saat sampai proses deploying selesai dilakukan, anda akan dapat melihat aplikasi WordPress pada halaman dashboard dari Azure Portal.



Untuk melakukan konfigurasi pada WordPress, pilih URL dari aplikasi yang sudah anda buat pada layanan Azure. Sebagai contoh alamat URL yang sudah saya buat adalah

<http://actualwpweb.azurewebsites.net>. Maka akan muncul tampilan awal untuk instalasi aplikasi WordPress. Anda dapat mengikuti proses instalasi sampai aplikasi WordPress siap. Dari artikel diatas dapat dilihat bahwa sangat mudah untuk memasang aplikasi open source CMS seperti WordPress pada layanan Microsoft Azure, dan ketika pengguna aplikasi anda bertambah dan anda merasa perlu untuk menambah kapasitas layanan anda, maka anda dapat melakukan proses scaling-up dengan relatif mudah. Anda dapat mengakses panel kontrol yang ada di Azure untuk meningkatkan paket layanan Azure Web atau layanan database anda sesuai dengan kebutuhan tanpa harus melakukan proses migrasi server yang merepotkan.

## Bab 3 – PHP & MySQL dengan Layanan Azure Web Apps.

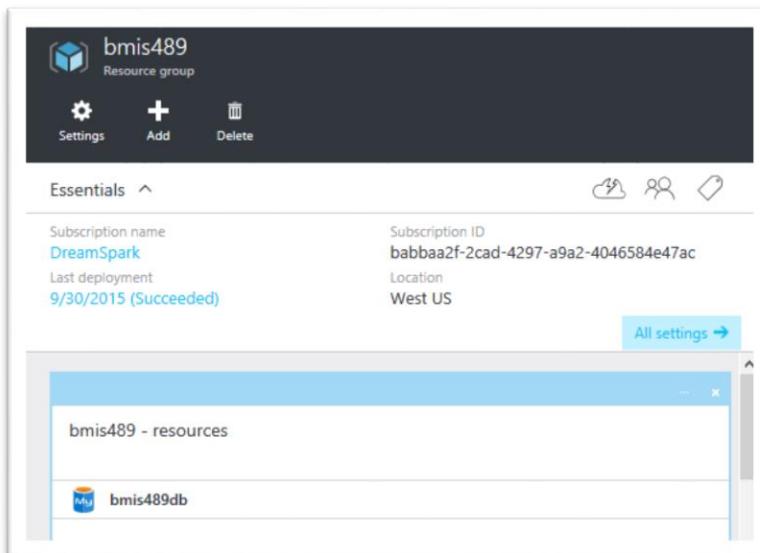
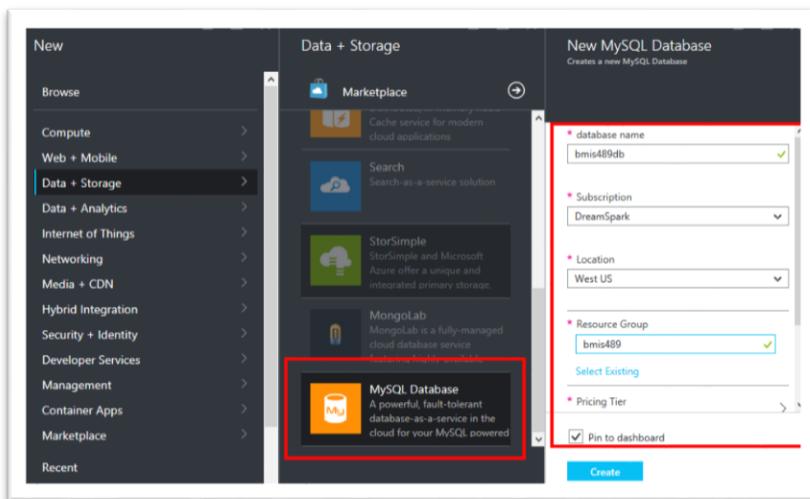
Pada bab ini kita akan mencoba untuk membuat aplikasi web dengan menggunakan PHP dan MySQL kemudian memasang aplikasi tersebut pada layanan Azure Web App.

Untuk menjalankan tutorial yang akan kita buat pada bab ini, maka anda harus menginstal beberapa aplikasi sebagai berikut.

- [XAMPP Lite](#) (jika anda ingin menjalankan aplikasi di server lokal).
- [MySQL Workbench](#)
- [Visual Studio Code for Editor](#) (atau editor lain seperti [sublimetext](#))
- [Git](#)
- [GitHub account](#)
- [GitHub desktop for Windows/Mac](#)

### Membuat Database MySQL di Azure

1. Login kedalam azure portal.
2. Pilih new icon, kemudian pilih Data + Storage, kemudian MySQL Database.
3. Buat resource group baru, sebagai contoh ‘bmis489’.
4. Pilih tombol create, dan sekarang database MySQL sudah dapat digunakan.

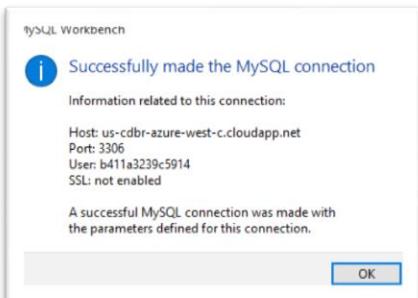
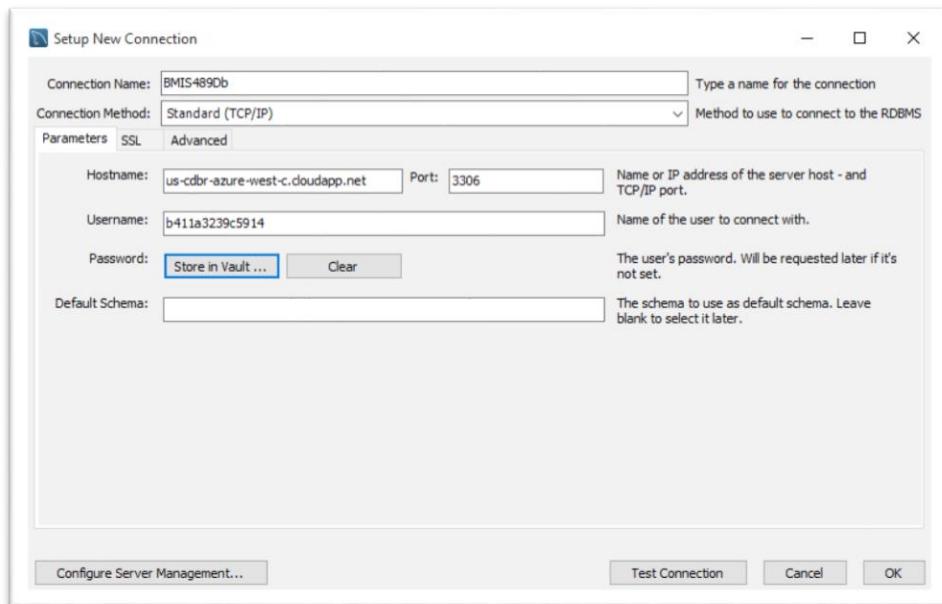


The screenshot shows the Azure portal interface for managing a MySQL database named 'bmis489db'. At the top, there are 'Delete' and 'Manage Database' buttons. Below that, the 'Essentials' section provides basic information: Resource group 'bmis489', Status 'Healthy', Location 'West US', Subscription Id 'babbaa2f-2cad-4297-a9a2-4046584e47ac', Hostname 'us-cdbr-azure-west-c.cloudapp.net', Plan 'Mercury', Max Size (in Mb) '20', and Subscription Name 'DreamSpark'. A 'All settings' button is located at the top right of the essentials section. Below this, the 'Usage' section displays two metrics: 'Database Size BMIS489DB' (CURRENT 0B, THRESHOLD 0%) and 'Connections Used BMIS489DB' (0%).

1. Anda dapat melihat database properties. Kita akan menggunakan property tersebut ketika akan mengakses MySQL database dari server side script.

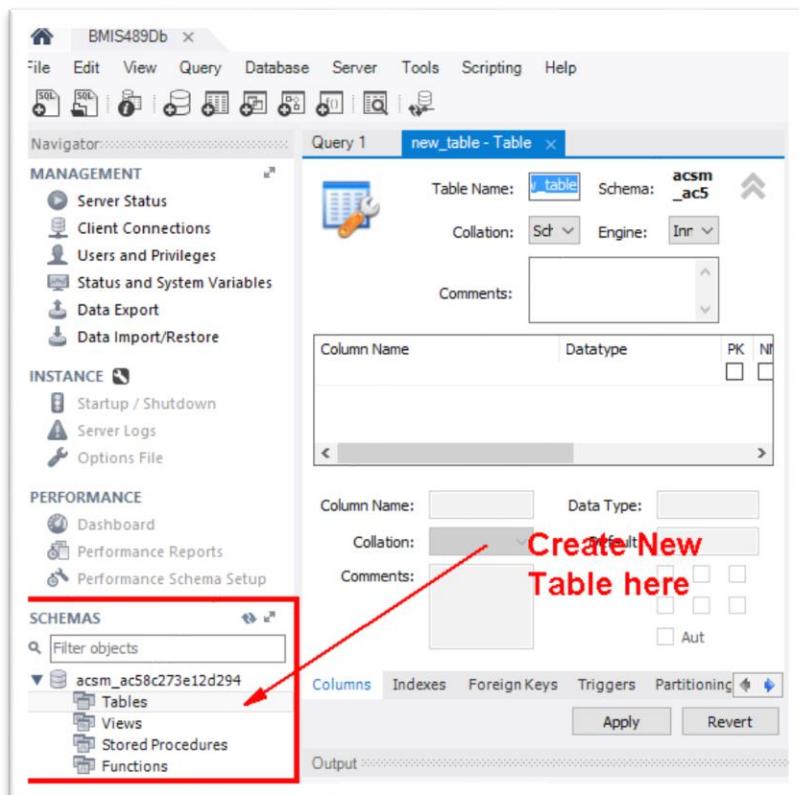
The screenshot shows the 'Properties' page for the MySQL database 'bmis489db'. On the left, there is a 'Search settings' input field and a 'Properties' section with a dropdown menu. Below that is a 'RESOURCE MANAGEMENT' section with 'Users' and 'Tags' options. On the right, the properties are listed: HOSTNAME 'us-cdbr-azure-west-c.cloudapp.net', PORT '3306', USERNAME 'b411a3239c5914', PASSWORD '7aee971d', CONNECTION STRING 'Database=acs...;Data...', and SUBSCRIPTION ID 'babbaa2f-2cad-4297-a9a2-4046584e47ac'.

2. Setelah database dibuat, anda dapat melakukan pengaturan database cloud tersebut pada komputer lokal. Buka tool MySQL Workbench dan masukan informasi hostname, username, dan password. Click Pilih Connection button untuk memastikan bahwa koneksi anda lancar.

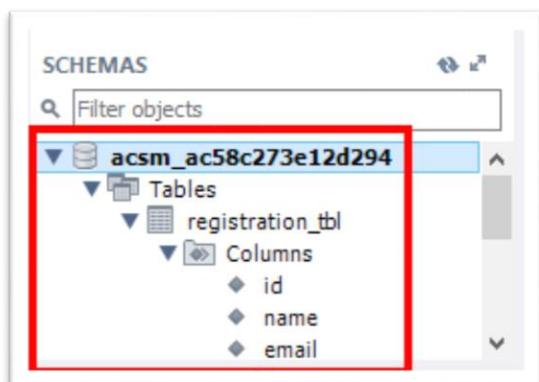


3. Buka MySQL Workbench, pada jendela query tambahkan script berikut, dan kemudian jalankan script untuk membuat tabel ‘registration\_tbl’.

```
CREATE TABLE registration_tbl(id INT NOT NULL AUTO_INCREMENT, PRIMARY KEY(id), name  
VARCHAR(30), email VARCHAR(30), date DATE);
```

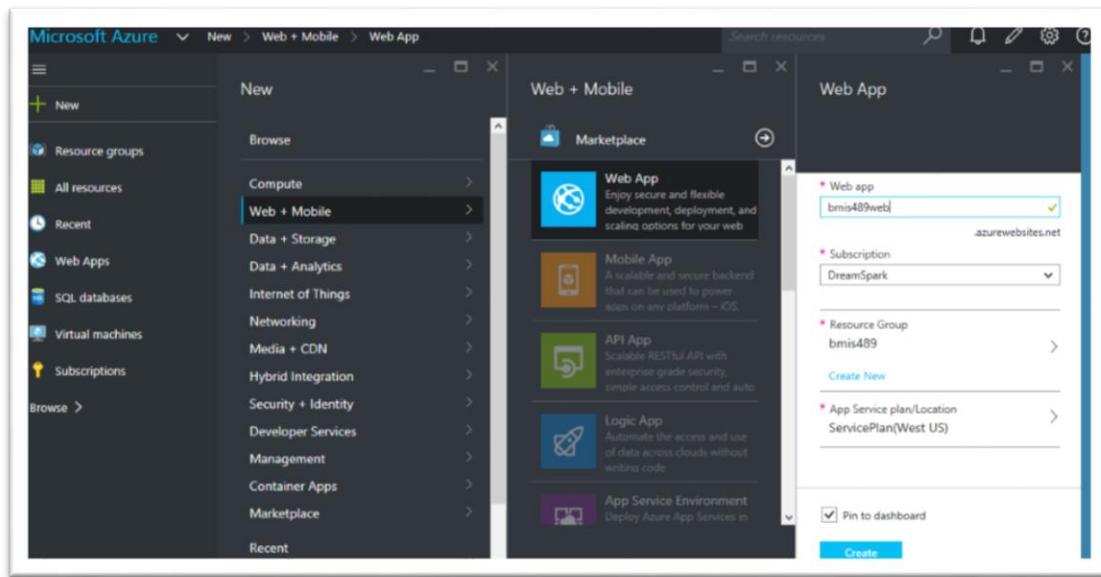


1. Anda dapat melihat tabel “registration\_tbl” berhasil dibuat.



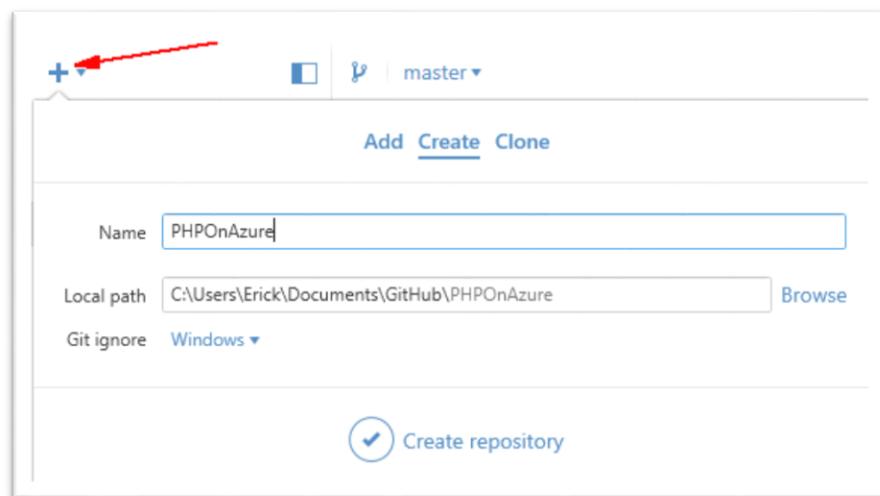
## Membuat Aplikasi Web dengan PHP

1. Langkah selanjutnya adalah membuat aplikasi web.
2. Pilih New Icon, kemudian pilih Web + Mobile, kemudian pilih Web App.
3. Masukan nama untuk aplikasi web yang anda buat.
4. Pilih “bmis489” sebagai nama resource group.
5. Pilih tombol create.



6. Ketika aplikasi web sudah berhasil dibuat, anda akan melihat tampilan resource group. Pilih nama pada aplikasi web untuk melakukan konfigurasi.

1. Buka aplikasi GitHub Desktop pada komputer lokal.



2. Buat file PHP dengan nama “index.php” dengan editor favorit anda. Simpan pada folder “PHPOnAzure”.
3. Buka file “index.php” dan tambahkan script berikut untuk terbuhung dengan MySQL database pada layanan Azure, menampilkan data peserta, dan menambahkan peserta pada sistem.

```
<html>
<head>
<Title>Registration Form</Title>
<style type="text/css">
body { background-color: #fff; border-top: solid 10px #000;
       color: #333; font-size: .85em; margin: 20; padding: 20;
       font-family: "Segoe UI", Verdana, Helvetica, Sans-Serif;
     }
h1, h2, h3,{ color: #000; margin-bottom: 0; padding-bottom: 0; }
h1 { font-size: 2em; }
h2 { font-size: 1.75em; }
h3 { font-size: 1.2em; }
table { margin-top: 0.75em; }
th { font-size: 1.2em; text-align: left; border: none; padding-left: 0; }
td { padding: 0.25em 2em 0.25em 0em; border: 0 none; }
</style>
```

```
</head>

<body>

<h1>Register here!</h1>

<p>Fill in your name and email address, then click <strong>Submit</strong> to register.</p>

<form method="post" action="index.php">

    Name <input type="text" name="name" id="name"/><br>
    Email <input type="text" name="email" id="email"/><br>
    <input type="submit" name="submit" value="Submit" />

</form>

<?php

// DB connection info

//TODO: Update the values for $host, $user, $pwd, and $db

//using the values you retrieved earlier from the portal.

$host = "us-cdbr-azure-west-c.cloudapp.net";
$user = "b411a3239c5914";
$pwd = "7aee971d";
$db = "acsm_ac58c273e12d294";

// Connect to database.

try {

    $conn = new PDO( "mysql:host=$host;dbname=$db", $user, $pwd);
    $conn->setAttribute( PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION );
}

catch(Exception $e){

    die(var_dump($e));
}

// Insert registration info

if(!empty($_POST)) {

    try {

        $name = $_POST['name'];
```

```
$email = $_POST['email'];

$date = date("Y-m-d");

// Insert data

$sql_insert = "INSERT INTO registration_tbl (name, email, date)
VALUES (?, ?, ?)";

$stmt = $conn->prepare($sql_insert);

$stmt->bindValue(1, $name);

$stmt->bindValue(2, $email);

$stmt->bindValue(3, $date);

$stmt->execute();

}

catch(Exception $e) {

    die(var_dump($e));

}

echo "<h3>Your're registered!</h3>";

}

// Retrieve data

$sql_select = "SELECT * FROM registration_tbl";

$stmt = $conn->query($sql_select);

$registrants = $stmt->fetchAll();

if(count($registrants) > 0) {

    echo "<h2>People who are registered:</h2>";

    echo "<table>";

    echo "<tr><th>Name</th>";

    echo "<th>Email</th>";

    echo "<th>Date</th></tr>";

    foreach($registrants as $registrant) {

        echo "<tr><td>".$registrant['name']."</td>";

        echo "<td>".$registrant['email']."</td>";

        echo "<td>".$registrant['date']."</td></tr>";

    }

}
```

```
}

echo "</table>";

} else {

    echo "<h3>No one is currently registered.</h3>";

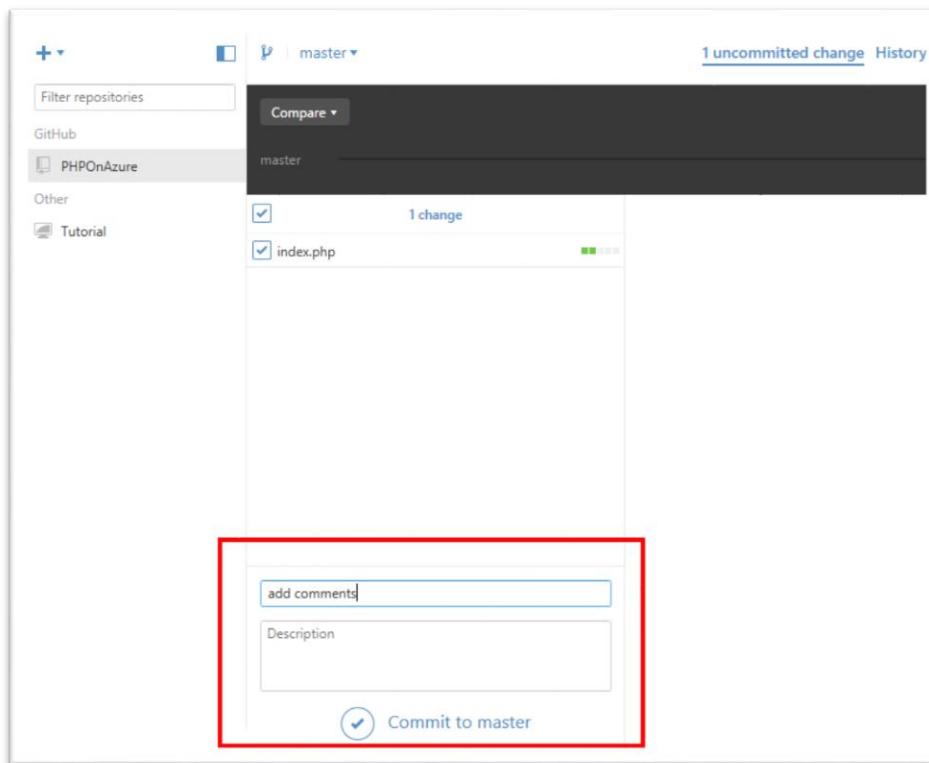
}

?>

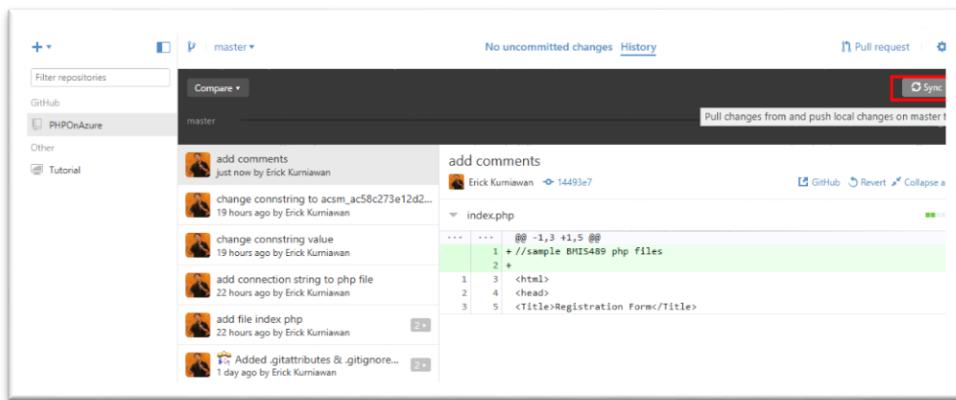
</body>

</html>
```

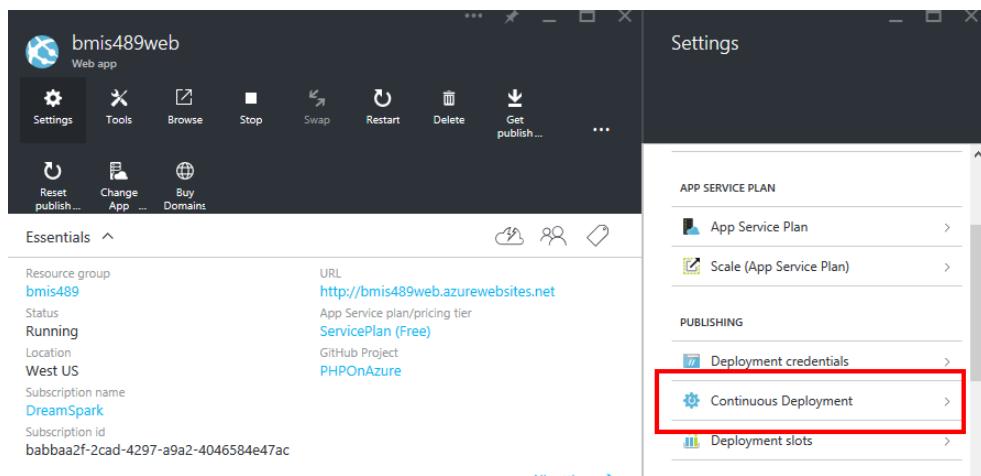
1. Buka aplikasi GitHub Desktop, dan kemudian pilih tombol commit to master untuk melakukan aksi push scripts anda kedalam Git repository.



2. Pilih tombol sync pada pojok kanan atas untuk melakukan aksi push local changes ke master yang ada di remote server.



3. Kembali ke pengaturan aplikasi web pada Azure. Pilih “Continuous Deployment” untuk melakukan automate build, test, and deploy aplikasi web yang telah dibuat ketika anda melakukan check-in ke GitHub.



4. Tambahkan akun GitHub anda, dan pilih kode anda pada code repository.

The screenshot shows the 'Deployments' section for the 'bmis489web' project. At the top, there are 'Sync' and 'Disconnect' buttons. Below this, a deployment history is listed:

- FRI 10/02**: A successful deployment is shown with a green checkmark icon. Details: 'add comments', 'GitHub', 'Active', '12:02 PM'.
- THU 10/01**: Two failed deployments are listed, each with a red minus sign icon. Details:
  - 'change connstring to acsm\_ac58c273e12d294', 'GitHub', 'Inactive', '5:02 PM'
  - 'add file index.php', 'GitHub', 'Inactive', '4:59 PM'

5. Setelah anda mempublish aplikasi anda, maka anda dapat mengedit kode dari aplikasi tersebut jika dibutuhkan, anda cukup melakukan code check-in pada akun GitHub anda, dan Azure akan mengambil kode anda secara otomatis kemudian mempublish kode anda.
6. Untuk melihat aplikasi, anda dapat mengakses tautan berikut  
<http://bmis489web.azurewebsites.net/index.php>.

The screenshot shows a registration form titled 'Register here!'. It includes fields for 'Name' and 'Email', and a 'Submit' button. Below the form, a success message says 'Your're registered!'. A heading 'People who are registered:' is followed by a table listing one registered user:

Name	Email	Date
Erick	kurniawan@gonzaga.edu	2015-10-01

## Bab 4 – ASP.NET MVC & Microsoft Azure

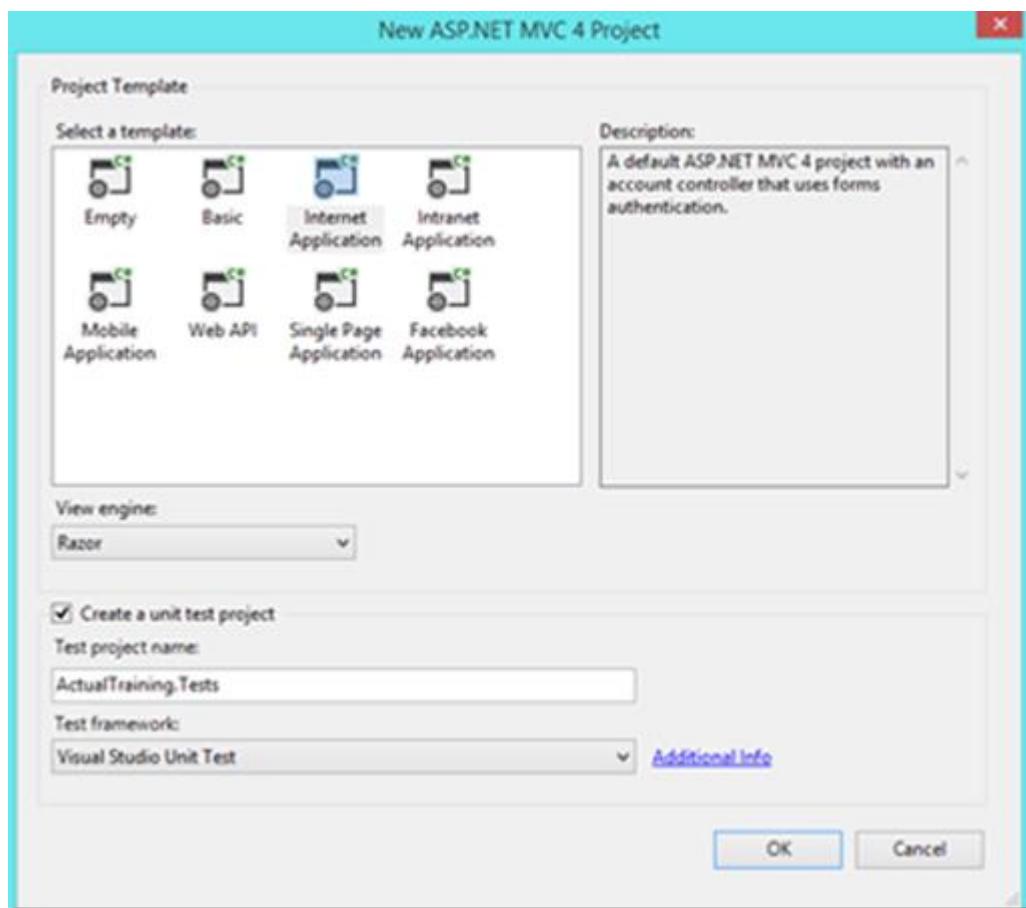
Pada bab ini saya akan membahas bagaimana cara membuat aplikasi ASP.NET MVC dan kemudian memasang aplikasi tersebut pada layanan Azure Web App.

### Membuat Project ASP.NET MVC

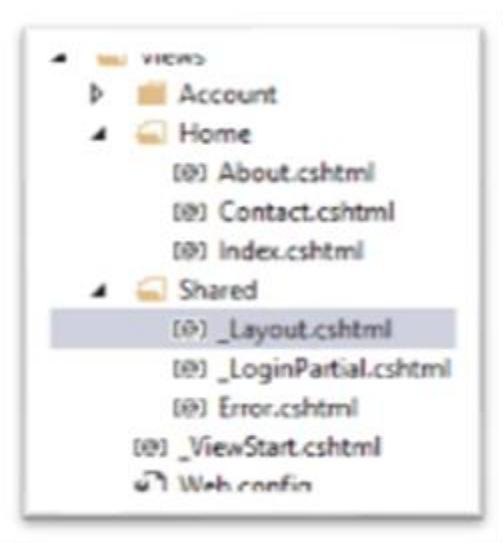
1. Pilih File – New Project – Visual C# - Web – Visual Studio 2012 – ASP.NET MVC 4  
- beri nama projectnya ActualTraining.



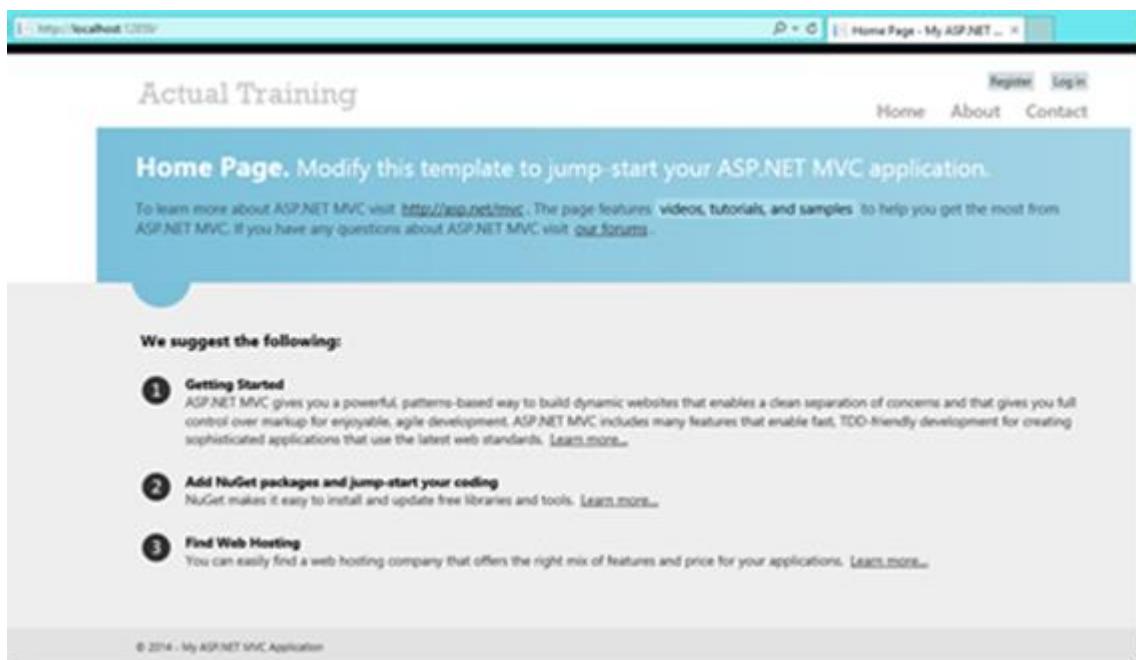
2. Pilih Internet Application, dan check pada pilihan ‘Create a unit test project’.



3. Pada solution explorer anda dapat melihat banyak file yang sudah secara otomatis di generate oleh Visual Studio ketika kita memilih template Internet Application.
4. Tekan F5 untuk mencoba menjalankan project yang sudah anda buat.
5. Anda dapat melakukan modifikasi template yang digunakan dengan cara merubah file \_layout.cshtml yang ada pada folder View.



6. Hasil akhir setelah melakukan sedikit modifikasi pada layout adalah sebagai berikut.



### Membuat Web Site Instance di Windows Azure

1. Login ke Microsoft Azure.
2. Klik pada Web Sites – NEW – Quick Create – beri nama sesuai keinginan anda.  
Nama ini akan digunakan sebagai nama domain dari aplikasi anda  
<nama>.azurewebsites.net. (pada contoh ini saya menggunakan nama **Actual**).
3. Pilih **East Asia** sebagai lokasi region. Ini adalah yang data center terdekat dengan lokasi pengguna aplikasi anda.



4. Kemudian pilih Create Website.

5. Tunggu sebentar karena Windows Azure akan membuatkan instance website baru untuk anda.

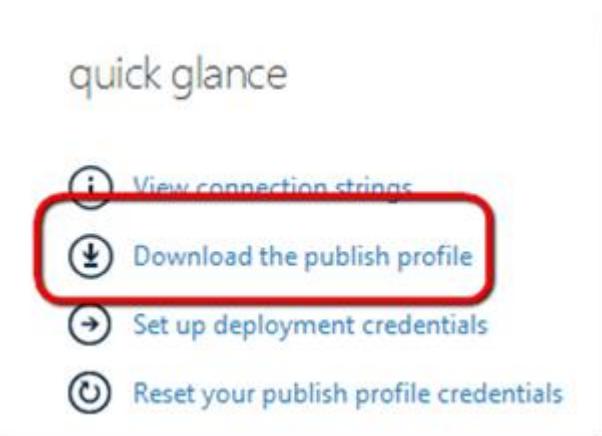
## web sites

NAME	STATUS	SUBSCRIPTION	LOCATION	MODE	URL
Actual	Running	Windows Azure MSD...	East Asia	Free	actual.azurewebsites.net

6. Setelah proses selesai maka anda dapat melihat nama domain dari aplikasi anda yaitu **actual.azurewebsites.net**.
7. Pilih website instance yang sudah kita buat yaitu Actual.

The screenshot shows the Windows Azure portal interface for the 'actual' website. At the top, there's a navigation bar with links for DASHBOARD, MONITOR, WEB JOBS, CONFIGURE, SCALE, and LINKED RESOURCES. Below the navigation, a large banner displays the message 'Your site has been created!' with a 'Skip Quick Start' checkbox. Underneath the banner, there's a section titled 'Get the tools' with links to 'Install WebMatrix' and 'Install a Windows Azure SDK'. The main content area is labeled 'quick glance' and contains four items: 'View connection strings', 'Download the publish profile' (which is highlighted with a red box), 'Set up deployment credentials', and 'Reset your publish profile credentials'.

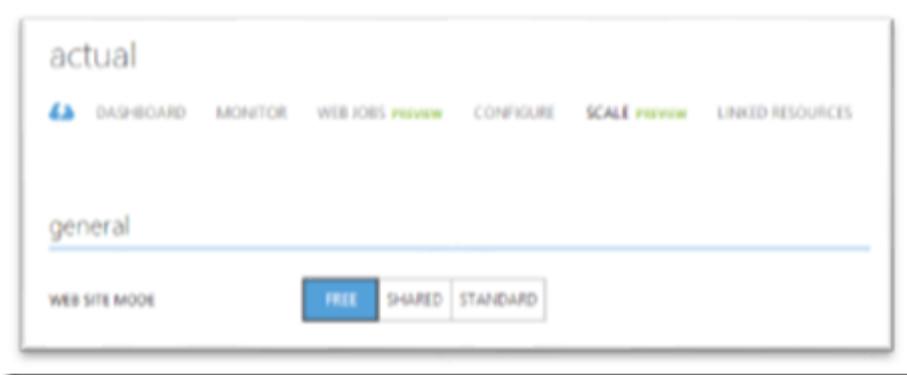
8. Kemudian pilih Dashboard. Pada menu pilih **Download Publish Profile**.



9. Simpan file setting hasil download ke komputer anda. (pada contoh ini saya mendapatkan file dengan nama Actual.azurewebsites.net.PublishSettings). file ini

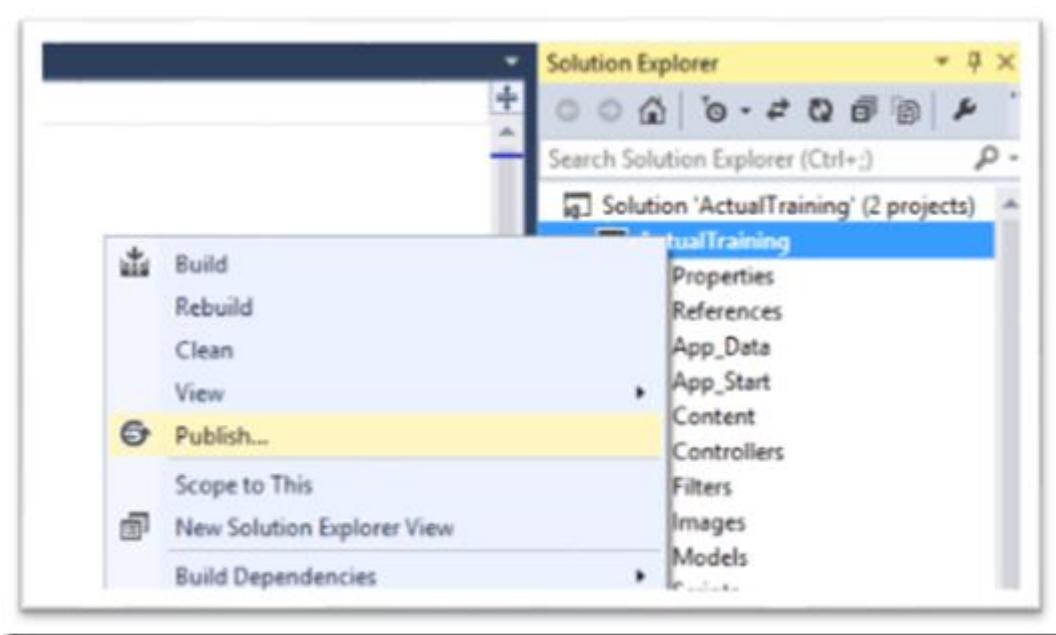
akan kita gunakan di Visual Studio untuk autentikasi ketika akan mengupload aplikasi yang kita buat ke Windows Azure.

10. Anda juga dapat melihat bahwa paket Web Site yang anda pilih adalah FREE pada menu SCALE.

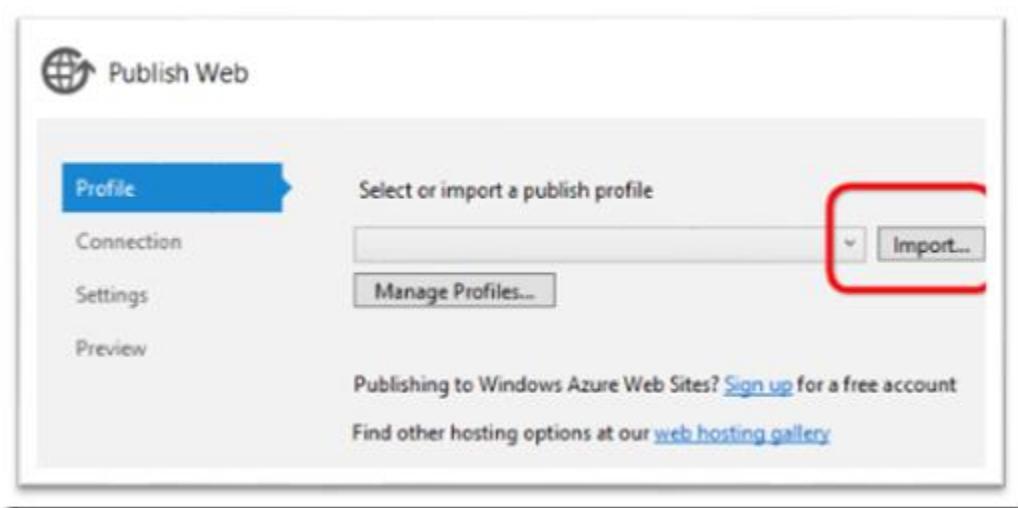


#### Upload Project di Visual Studio ke Microsoft Azure

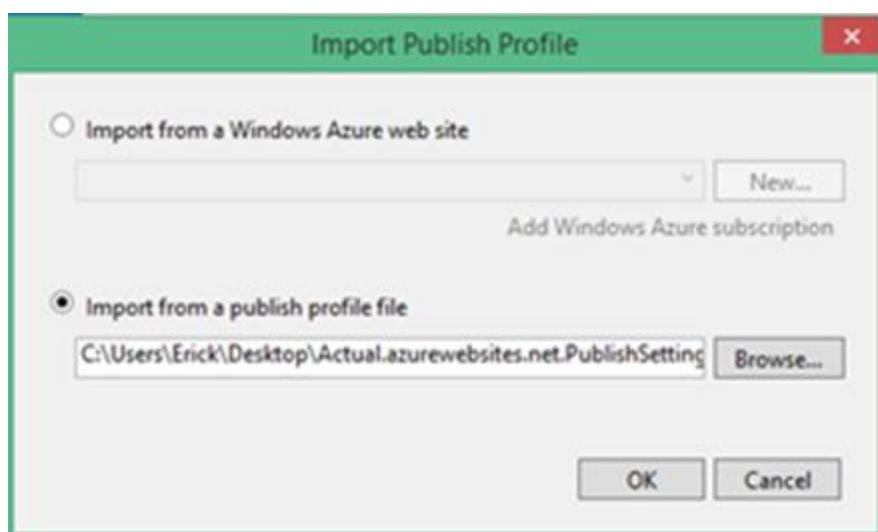
1. Buka project ActualTraining yang sebelumnya sudah anda buat.
2. Klik kanan pada Project, kemudian pilih **Publish**.

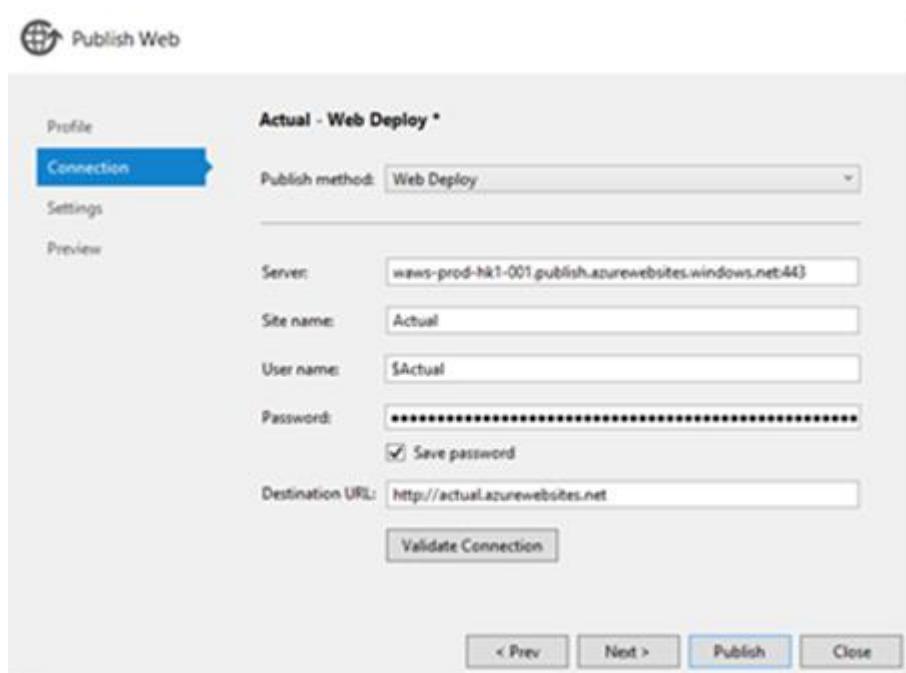


3. Kemudian pilih tombol Import.

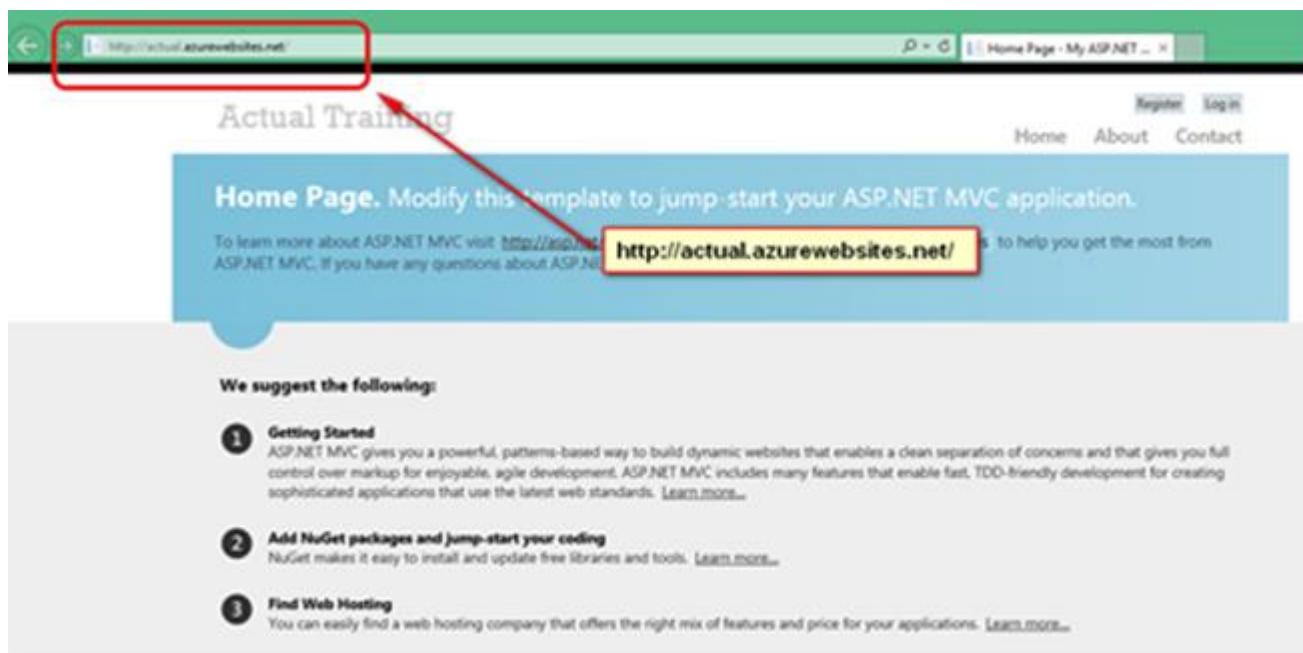


4. Pilih **publish profile** yang telah anda download sebelumnya.





5. Kemudian tekan tombol Publish.
6. Setelah project web anda selesai dipublish ke windows azure, maka akan ditampilkan pada browser sebagai berikut. (url sesuai dengan nama domain yang kita pilih sebelumnya)



Jadi dengan langkah yang cukup sederhana dan cepat anda dapat membuat website, memilih nama domain dan publish aplikasi tersebut di Microsoft Azure.

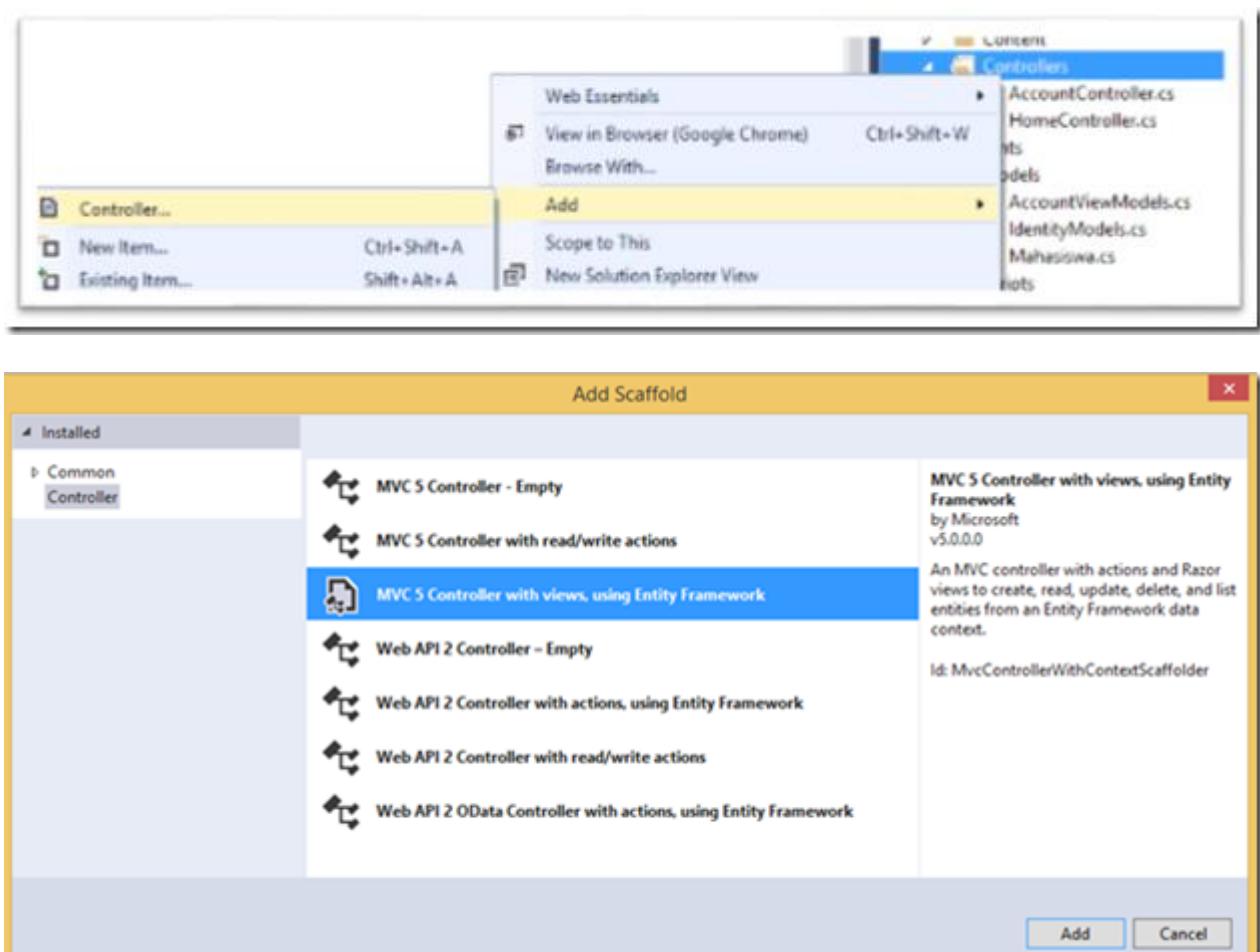
## Menggunakan SQL Azure dan ASP.NET MVC

Pada subbab sebelumnya kita telah berhasil membuat project ASP.NET MVC kemudian memasang website tersebut pada Microsoft Azure. Pada subbab kali ini kita akan menggunakan Entity Framework Code First untuk membuat database di SQL Azure dan kemudian mengakses database tersebut dari Microsoft Azure.

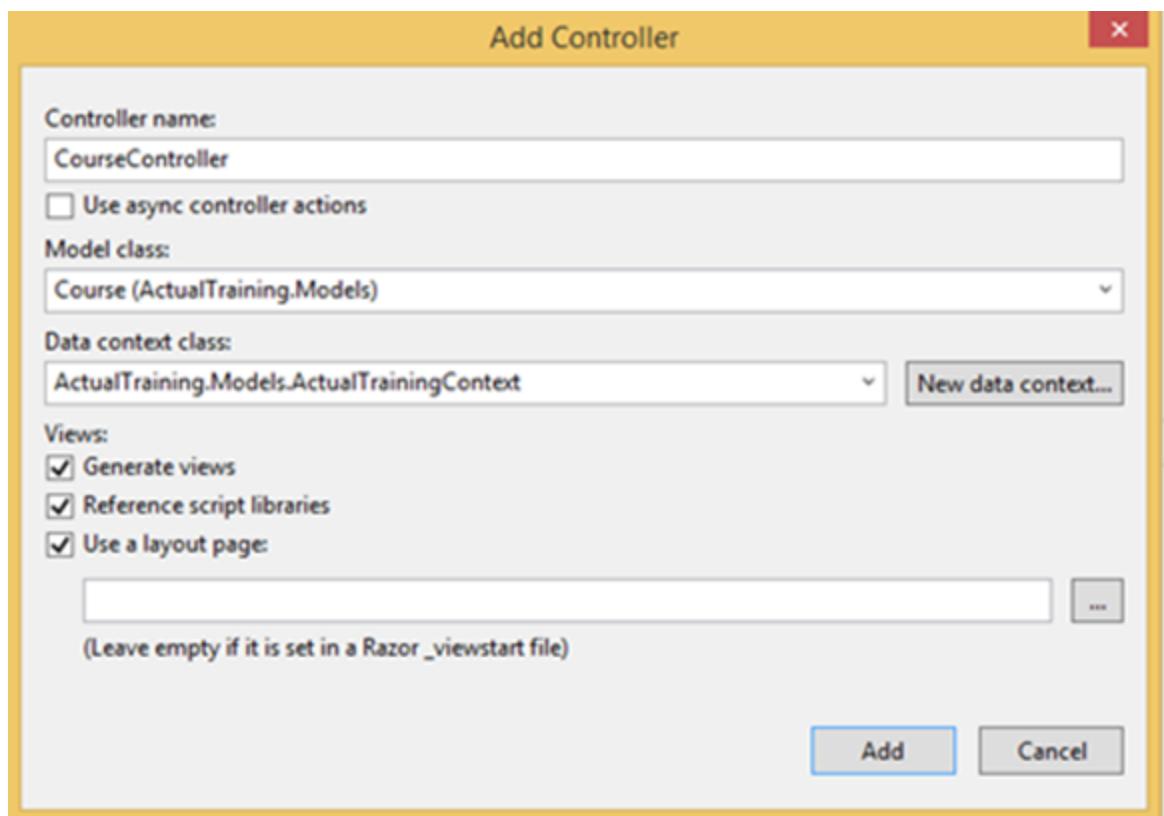
1. Buka project **ActualTraining** yang sebelumnya sudah kita buat.
2. Kemudian kita akan menambahkan data baru yang akan ditampilkan. Sebagai contoh kita akan membuat database dengan ‘Course’ untuk menyimpan data mahasiswa.
3. Tambahkan class baru pada table model dengan nama ‘Course’.

```
public class Course { public int CourseID { get; set; } public string Title { get; set; } public string Description { get; set; } public decimal Prices { get; set; } }
```

4. Buat Controller baru dengan nama CourseController.



5. Tambahkan detail berikut pada controller yang anda buat. Jangan lupa untuk memilih model Course yang sudah anda buat sebelumnya, kemudian tambahkan juga class DataContext baru.



6. Kemudian pada file ActualTrainingContext tambahkan kode berikut untuk menambahkan data contoh.

```
public class ActualTrainingContextInitialize : DropCreateDatabaseIfModelChanges<ActualTrainingContext> {protected override void Seed(ActualTrainingContext context) { context.Courses.Add(new Course { Title = "C# Fundamental", Description = "Course ini berisi materi dasar pemrograman C#", Prices = 100000 }); context.Courses.Add(new Course { Title = "ASP.NET MVC", Description = "Course ini berisi materi tentang pemrograman Web berbasis ASP.NET MVC", Prices = 150000 }); }}
```

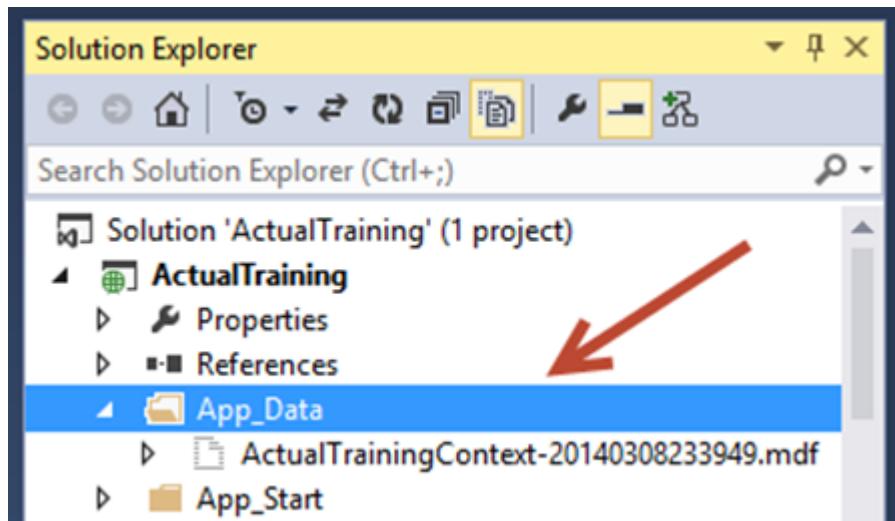
7. Tambahkan juga kode berikut pada file global.asax untuk membuat database ketika aplikasi pertama kali dijalankan.

```
protected void Application_Start() { .....  
Database.SetInitializer<ActualTrainingContext>(new ActualTrainingContextInitialize r()); }
```

8. Jalankan aplikasi anda, maka database akan dibuat secara otomatis.

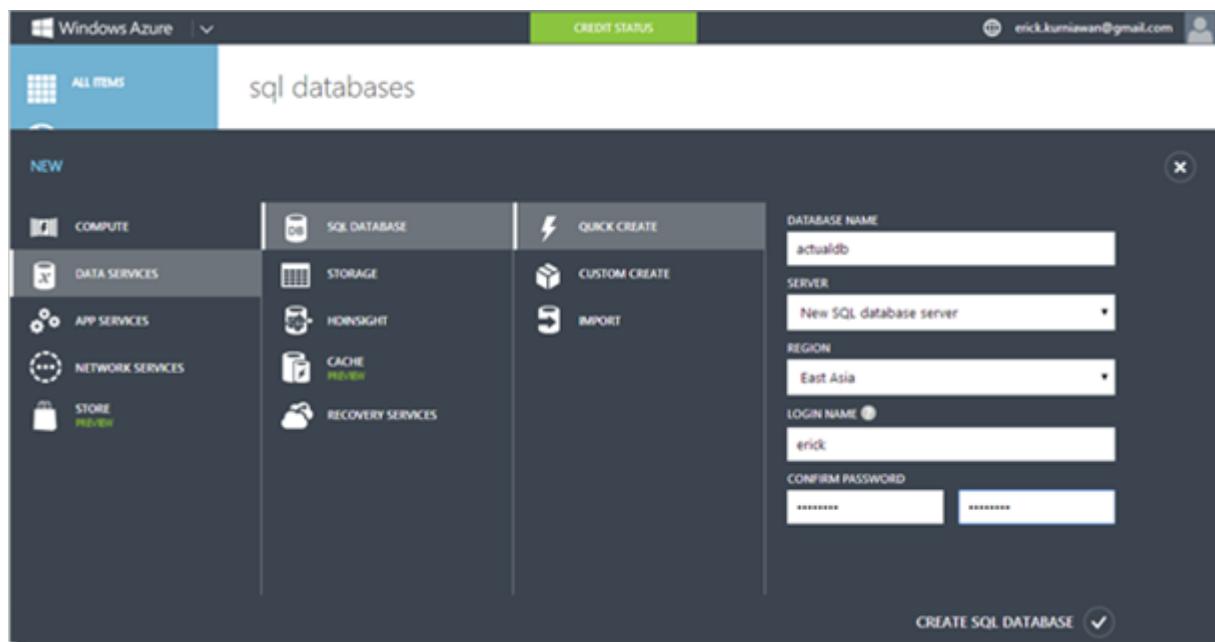
Course List			
Create New			
Title	Description	Prices	
C# Fundamental	Course ini berisi materi dasar pemrograman C#	100000.00	Edit   Details   Delete
ASP.NET MVC	Course ini berisi materi tentang pemrograman Web berbasis ASP.NET MVC	150000.00	Edit   Details   Delete

9. Jika anda lihat pada solution explorer di aplikasi anda maka dapat dilihat bahwa ada database yang secara otomatis dibuat yaitu ActualTrainingContext-20140308233949.mdf



#### Membuat Database pada Windows Azure

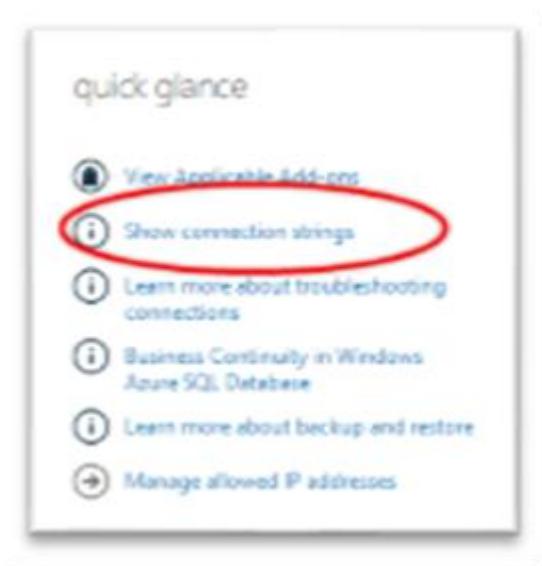
1. Buka Windows Azure Management untuk membuat database baru.
2. Pilih SQL Database – NEW, untuk membuat database baru.
3. Masukan nama database, pilih new sql database server, dan masukan username dan password untuk mengakses database server tersebut.



- Setelah database selesai dibuat maka hasilnya dapat dilihat pada gambar dibawah sebagai berikut.

sql databases						
DATABASES		SERVERS				
NAME	STATUS	LOCATION	... SERVER	EDITION	MAX SIZE	P
actualdb	✓ Online	East Asia	... muc40k4b2i	Web	1 GB	

- Langkah selanjutnya adalah kita akan mengcopy connection string untuk melakukan koneksi dengan database yang sudah kita buat di windows azure.



- Pada menu dashboard di database pilih ‘Show connection string’. Kemudian copy conn



7. Kemudian tambahkan konfigurasi connection string tersebut ke file web.config pada aplikasi anda.

```
<connectionStrings>
  <add name="ActualTrainingContext"
    connectionString="Server=tcp:muc40k4b21.database.windows.net,1433;
    Database=actualdb;User ID=erick@actualdb;Password=*****;
    Trusted_Connection=False;Encrypt=True;Connection Timeout=30;" 
    providerName="System.Data.SqlClient"/>
</connectionStrings>
```

8. Kemudian publish kembali aplikasi yang sudah anda buat ke Windows Azure (langkah untuk publish aplikasi ASP.NET MVC ke Windows Azure dapat dilihat pada artikel sebelumnya).
9. Setelah aplikasi anda berhasil dipublish di Windows Azure, coba untuk akses url [http://<url\\_app\\_anda>/Course](http://<url_app_anda>/Course) untuk menampilkan daftar Course.

The screenshot shows a web browser window with the URL <http://actualtraining.azurewebsites.net/Course>. The page title is "Course List". There is a "Create New" button. Below it is a table with the following data:

Title	Description	Prices	Action
C# Fundamental	Course ini berisi materi dasar pemrograman C#	100000.00	Edit   Details   Delete
ASP.NET MVC	Course ini berisi materi tentang pemrograman Web berbasis ASP.NET MVC	150000.00	Edit   Details   Delete

## Autentikasi Aplikasi ASP.NET MVC dengan Azure Active Directory

Pada subbab ini akan dibahas bagaimana cara autentikasi dengan melakukan login dari ASP.NET MVC menggunakan Azure AD (Active Directory). Beberapa tahapan yang akan kita lakukan adalah sebagai berikut:

- Membuat akun pada Azure Active Directory
- Membuat aplikasi ASP.NET MVC baru yang menggunakan Azure AD untuk autentikasi

### *Membuat Azure Active Directory*

Langkah pertama yang harus dilakukan adalah membuat akun di Azure Active Directory.

The screenshot shows the 'Add directory' page in the Azure portal. The form has several fields: 'DIRECTORY' (dropdown menu showing 'Create new directory'), 'NAME' (text input field containing 'Cloudemia'), 'DOMAIN NAME' (text input field containing 'cloudemia' with a '.onmicrosoft.com' suffix), and 'COUNTRY OR REGION' (dropdown menu showing 'Select country or region'). A red box highlights the 'DOMAIN NAME' field. In the bottom right corner of the form area, there is a circular button with a checkmark icon.

Setelah anda berhasil membuat Azure Active Directory, tambahkan user yang akan digunakan untuk login kedalam aplikasi ASP.NET anda.

The screenshot shows the cloudemia web application. At the top, there is a navigation bar with links: USERS (which is highlighted with a red box), GROUPS, APPLICATIONS, DOMAINS, DIRECTORY INTEGRATION, CONFIGURE, REPORTS, and LICENSES. Below the navigation bar is a large blue diamond icon containing a white network graph symbol. To the right of the icon, the text "Your directory is ready to use. Here are a few options to get started." is displayed. There is also a checkbox labeled "Skip Quick Start the next time I visit". At the bottom of the main content area, there is a horizontal bar with four buttons: "I WANT TO", "Set Up Directory" (which is highlighted in blue), "Manage Access", and "Develop Applications".

Tambahkan user baru kedalam Active Directory. Pada contoh dibawah ini akan ditambahkan user baru yaitu [jovan@cloudemia.onmicrosoft.com](mailto:jovan@cloudemia.onmicrosoft.com)

The screenshot shows the "ADD USER" form. The title "ADD USER" is at the top left. Below it is the heading "Tell us about this user". Under "TYPE OF USER", there is a dropdown menu set to "New user in your organization". In the "USER NAME" field, the value "jovan" is entered. To the right of the user name field is a dropdown menu showing "cloudemia.onmicrosoft.com". A vertical blue bar is on the right side of the form.

Tambahkan role dari user yang baru saja anda buat sebagai **Global Administrator**.

The screenshot shows the 'ADD USER' interface for creating a new user profile. The form includes fields for First Name ('Jovan'), Last Name ('Christoval'), Display Name ('Jovan Christoval'), Role ('Global Administrator'), Alternate Email Address ('jovan@actual-training.com'), and Multi-Factor Authentication ('Enable Multi-Factor Authentication').

FIRST NAME: Jovan  
LAST NAME: Christoval  
DISPLAY NAME: Jovan Christoval  
ROLE: Global Administrator  
ALTERNATE EMAIL ADDRESS: jovan@actual-training.com  
MULTI-FACTOR AUTHENTICATION:  Enable Multi-Factor Authentication

Kemudian anda dapat menggenerate password untuk user tersebut.

The screenshot shows the 'Get temporary password' step. It displays a message stating that a temporary password will be assigned and must be changed on first sign in. A green 'create' button is highlighted with a red border.

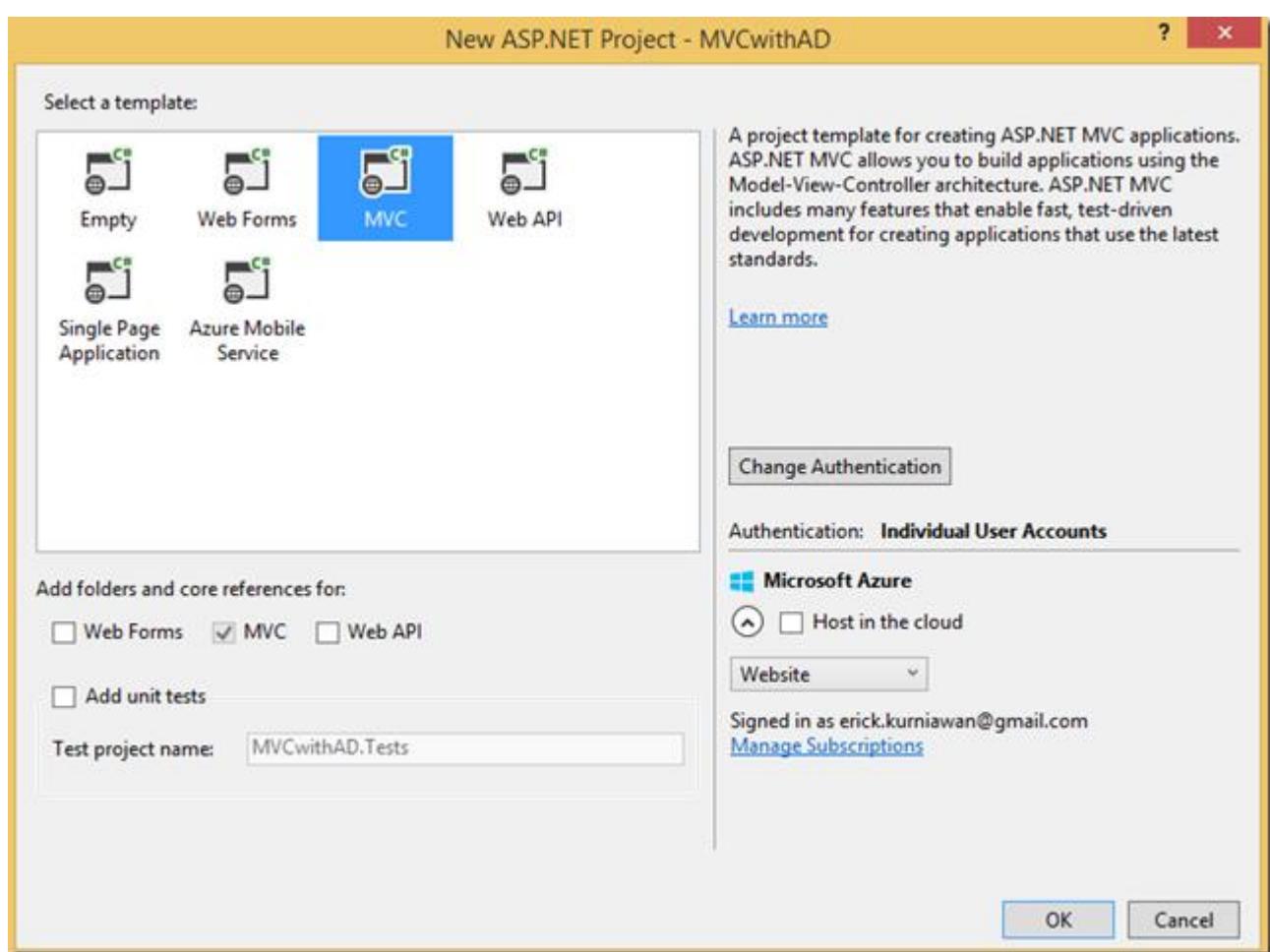
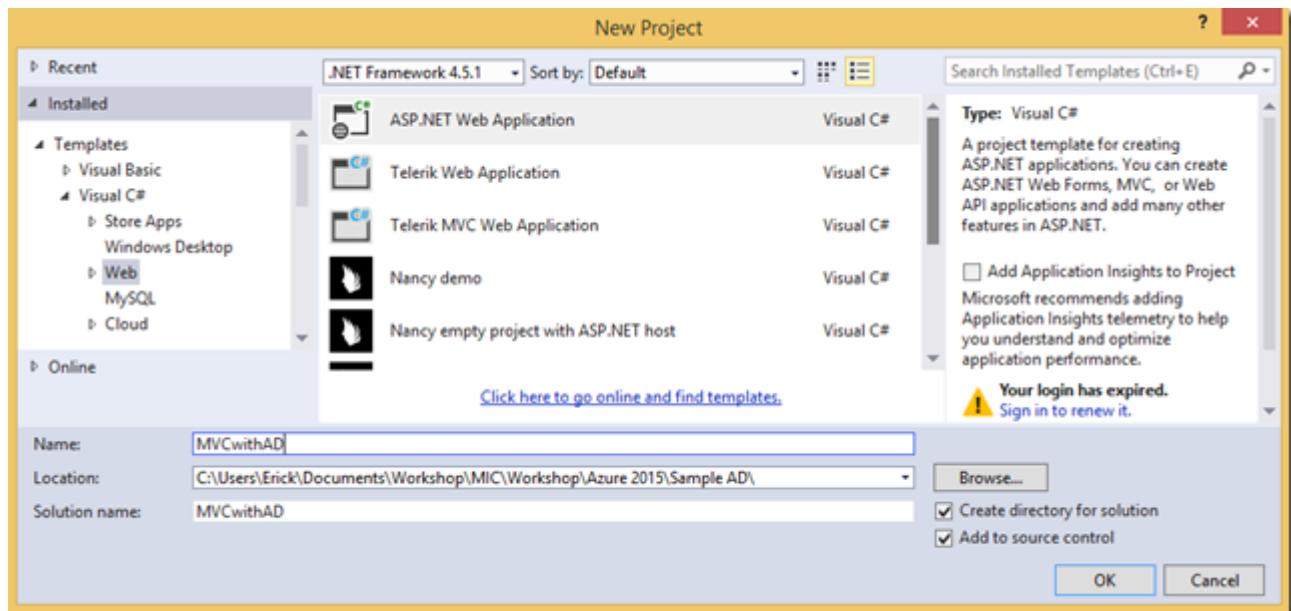
The new user 'budi@cloudemia.onmicrosoft.com' will be assigned a temporary password that must be changed on first sign in. To display the temporary password and to create the account, click Create.

create

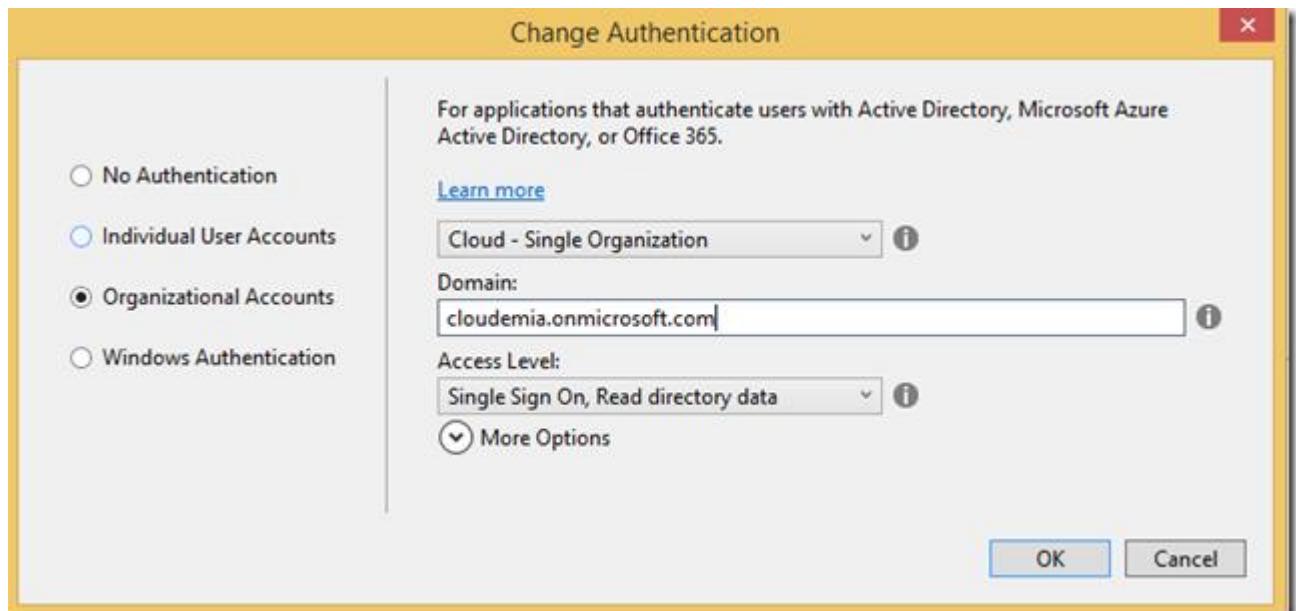
Anda dapat login sebagai user **jovan** dan mengganti password dengan login terlebih dahulu pada halaman berikut <https://account.activedirectory.windowsazure.com/>.

Membuat Aplikasi ASP.NET MVC

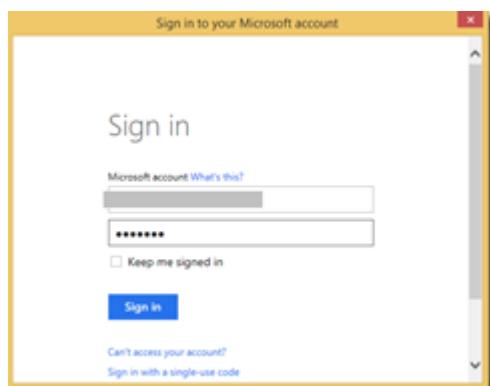
Buat project ASP.NET MVC, kemudian pilih tombol **Change Authentication**, kemudian pilih **Organizational Account**



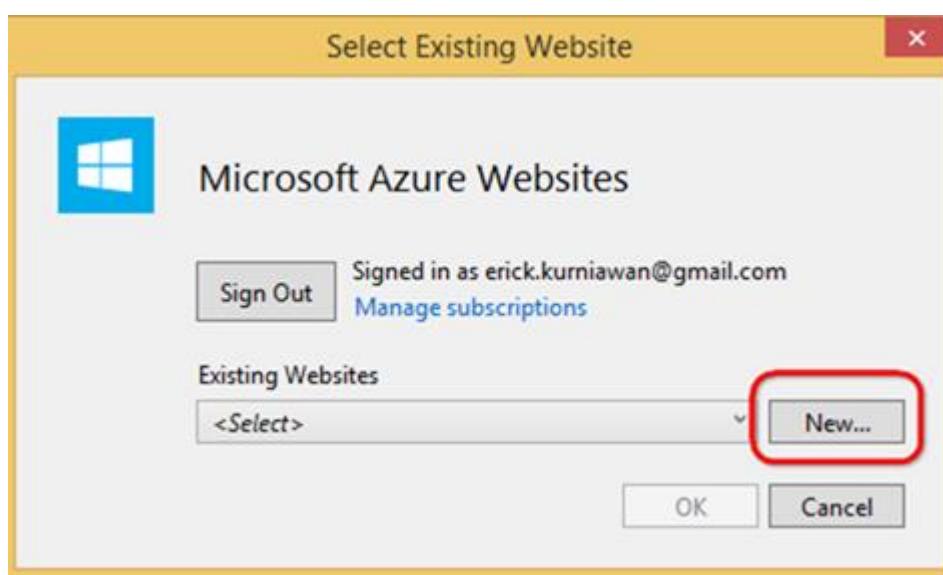
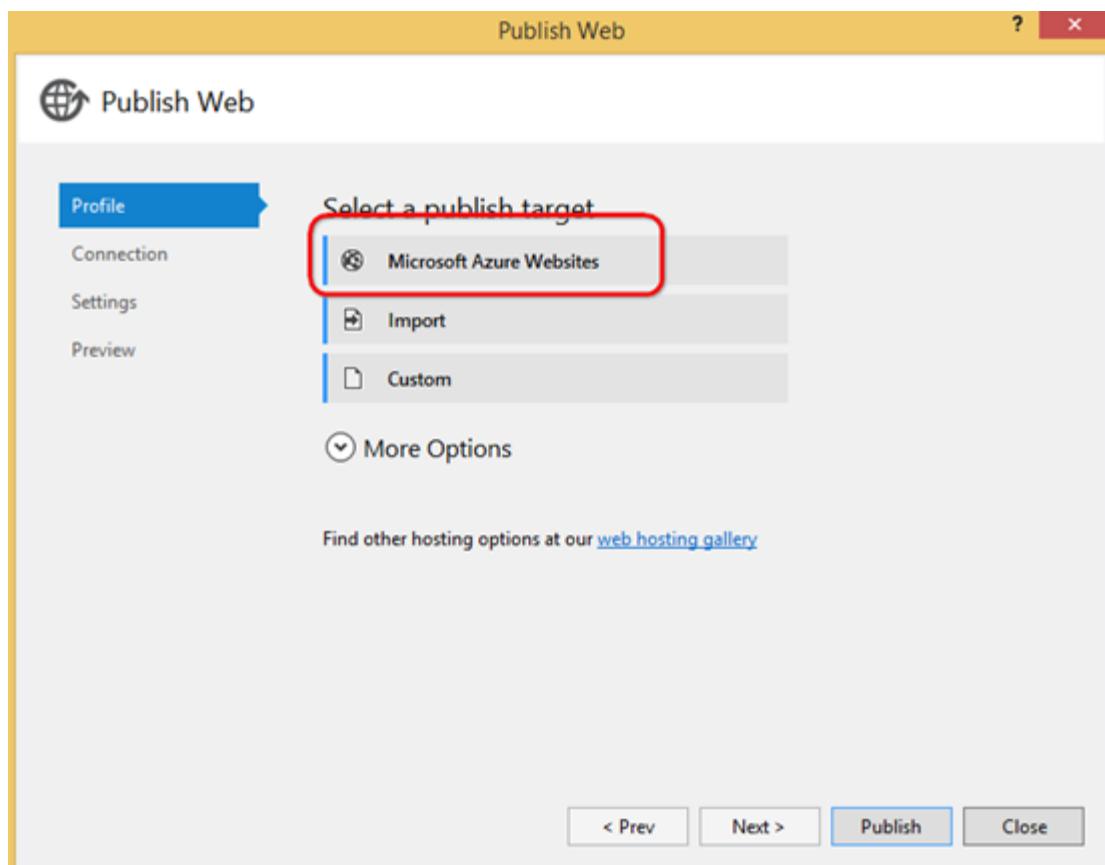
Tambahkan domain Azure Active Directory anda pada isian Domain, dan pilih Access Level: Single Sign On, Read derectory data.

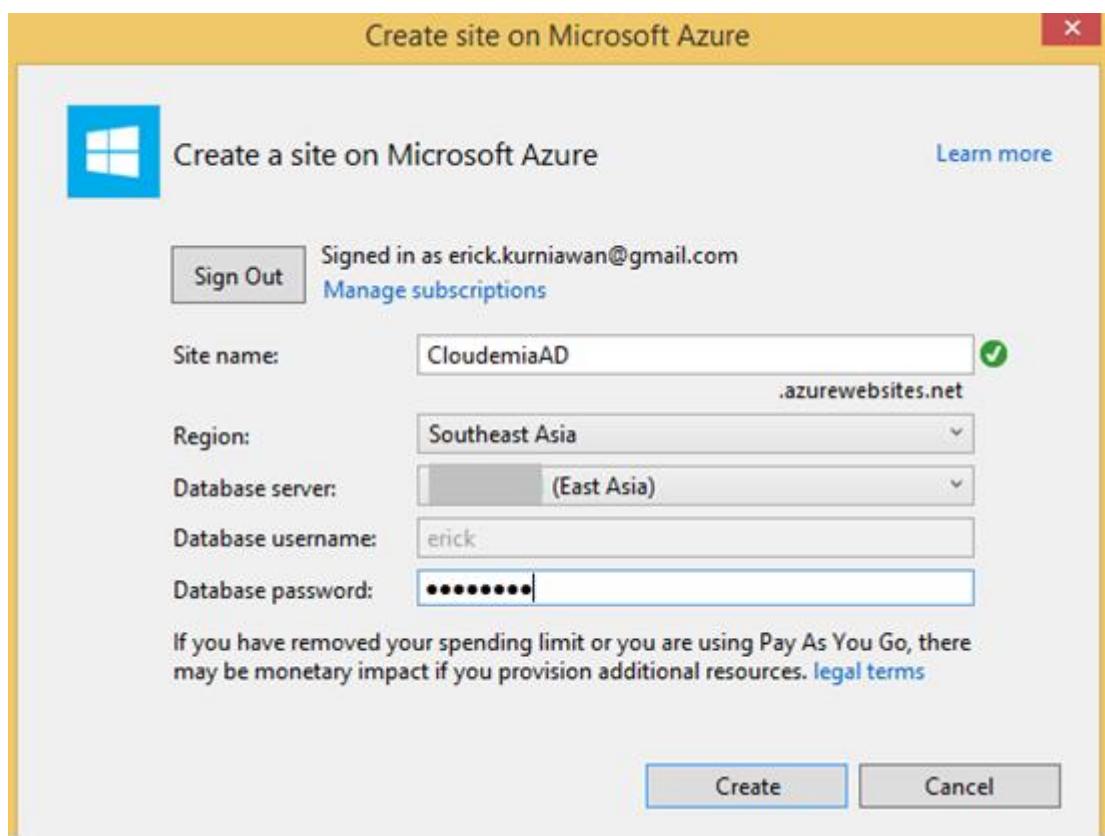


Jika muncul halaman sign-in, anda dapat memasukan user Azure Active Directory anda.

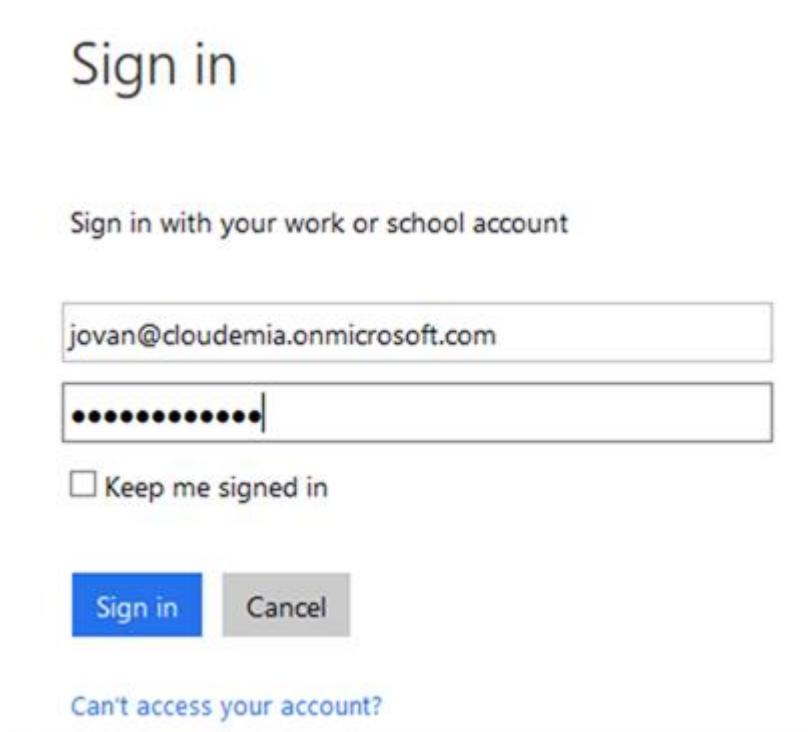


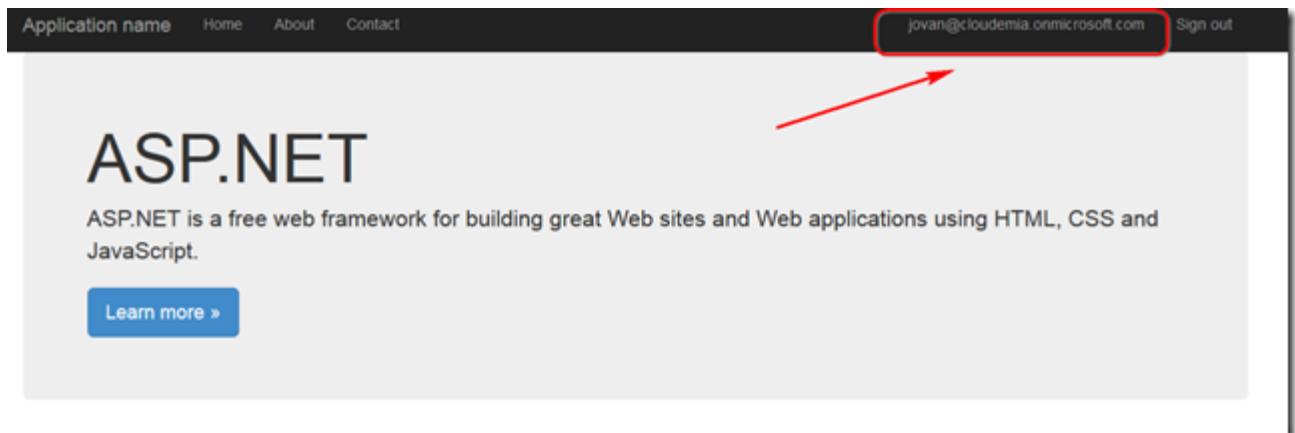
Anda dapat mempublish aplikasi ASP.NET MVC yang anda buat ke Microsoft Azure Website.





Anda diharuskan melakukan login ke Azure AD terlebih dahulu sebelum dapat mengakses aplikasi web.





## Bab 5 - Windows Azure Mobile Services

Pada bab ini kita akan mempelajari penggunaan Windows Azure Mobile Services yang dapat digunakan sebagai backend untuk aplikasi web dan aplikasi mobile.

Azure Mobile Services menyediakan layanan penting yang dibutuhkan oleh setiap developer aplikasi mobile seperti :

- Data : stuctured storage, sql database, dynamic schema
- Push Notification : yang dapat diakses dari Windows, Windows Phone, iOS, dan Android.
- Identity : Oauth Providers, Notification Hub.
- Scheduled Task : Server-side scripting, API framework

Karena Azure Mobile Services berjalan di platform Windows Azure yang merupakan layanan berbasis cloud, maka Azure Mobile Services juga bersifat elastis sehingga dapat dengan mudah di *scale-up* dan *scale-down*.

### Memulai Membuat Azure Mobile Services

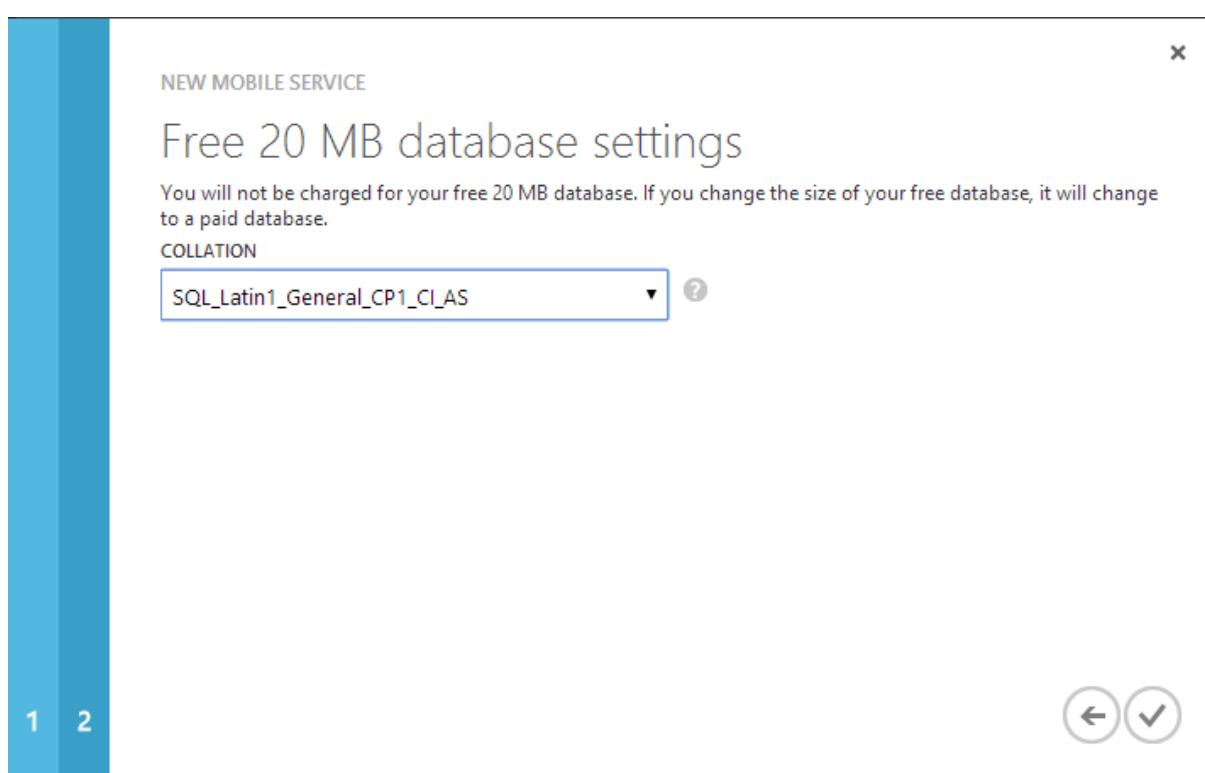
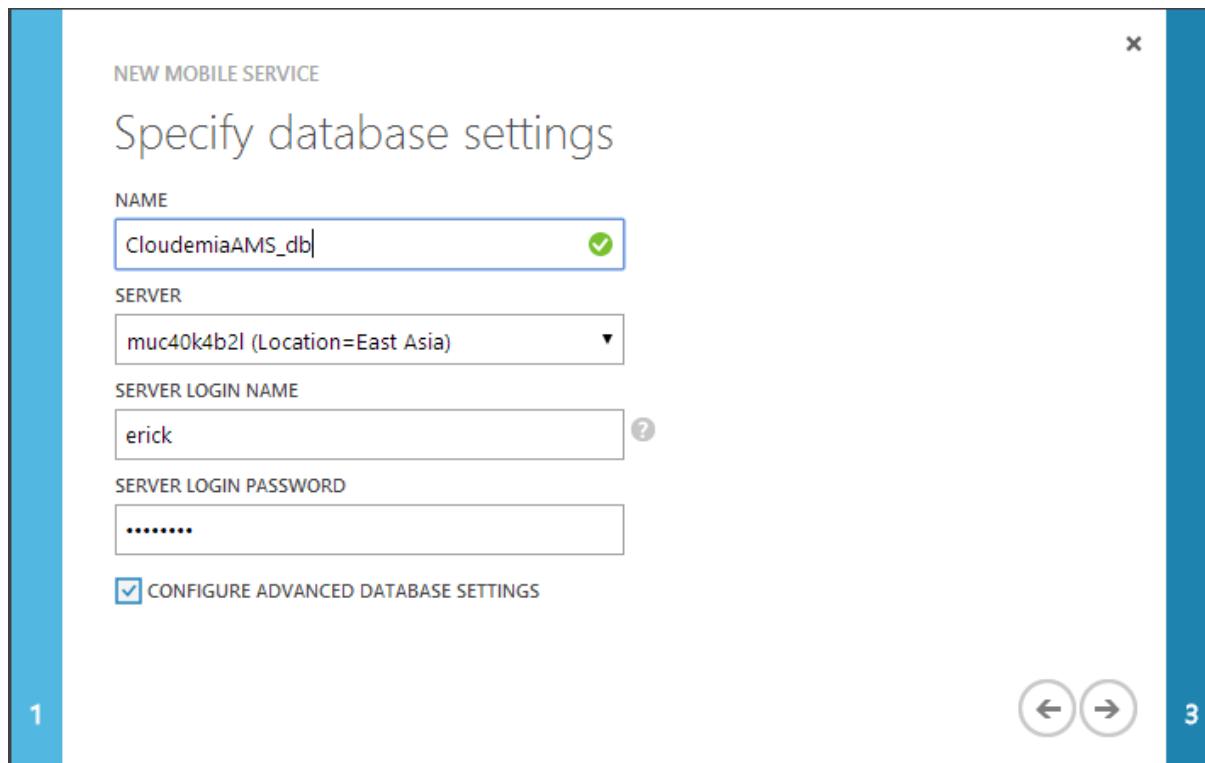
1. Buka Windows Azure manajemen portal, kemudian pilih Mobile Services.

The screenshot shows the Windows Azure dashboard. On the left, there's a sidebar with icons for All Items, Web Sites, Virtual Machines, Mobile Services (which has a red arrow pointing to it), Cloud Services, and SQL Databases. The main area is titled "mobile services" and displays the message "You have no mobile services. Create one to get started!" Below this is a "CREATE A NEW MOBILE SERVICE" button.

2. Tambahkan new mobile service baru dengan detail sebagai berikut (untuk penamaan anda dapat menentukan nama sendiri). Anda dapat memilih free 20mb database yang disediakan oleh Azure.

The screenshot shows the "Create a Mobile Service" dialog box. It contains fields for URL (CloudemiaAMS.azure-mobile.net), DATABASE (Create a free 20 MB SQL database), REGION (East Asia), and BACKEND (JavaScript). At the bottom right is a large "Next" button with a circular arrow icon and the number "2".

3. Pilih tombol next. Kemudian anda dapat memilih menyimpan database tersebut pada database server yang sudah ada, atau membuat database server yang baru.



4. Setelah selesai maka anda dapat melihat azure mobile services baru dengan nama CloudemiaAMS.

NAME	STATUS	SUBSCRIPTION	LOCATION
CludemiaAMS	Ready	Windows Azure MSDN ...	East Asia

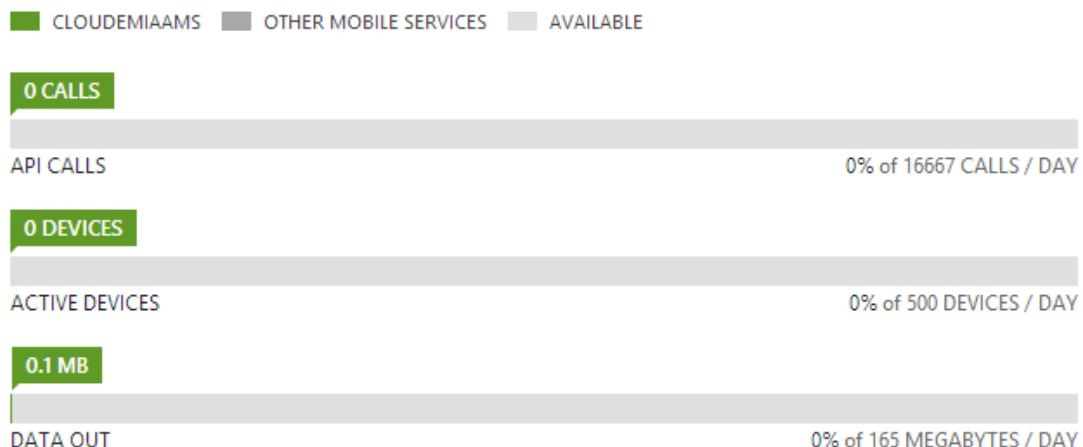
5. Sampai tahap ini kita sudah berhasil membuat azure mobile services baru.
6. Bila kita pilih ke menu SQL Database anda akan dapat melihat database CludemiaAMS\_db yang akan digunakan untuk menyimpan data.
7. Jika anda perhatikan maka Azure Mobile Services dan SQL Database yang sudah kita buat akan memiliki region yang sama.

NAME	STATUS	LOCATION	... SERVER
actualdb	Online	East Asia	... muc40k4b2l
samplewebdb	Online	East Asia	... muc40k4b2l
CludemiaAMS_db	Ready	Online	East Asia ... muc40k4b2l

8. Kemudian pilih azure mobile services yang sudah anda buat. Akan muncul beberapa menu seperti dashboard, data, api, scheduler, push, identity, configure, dll.

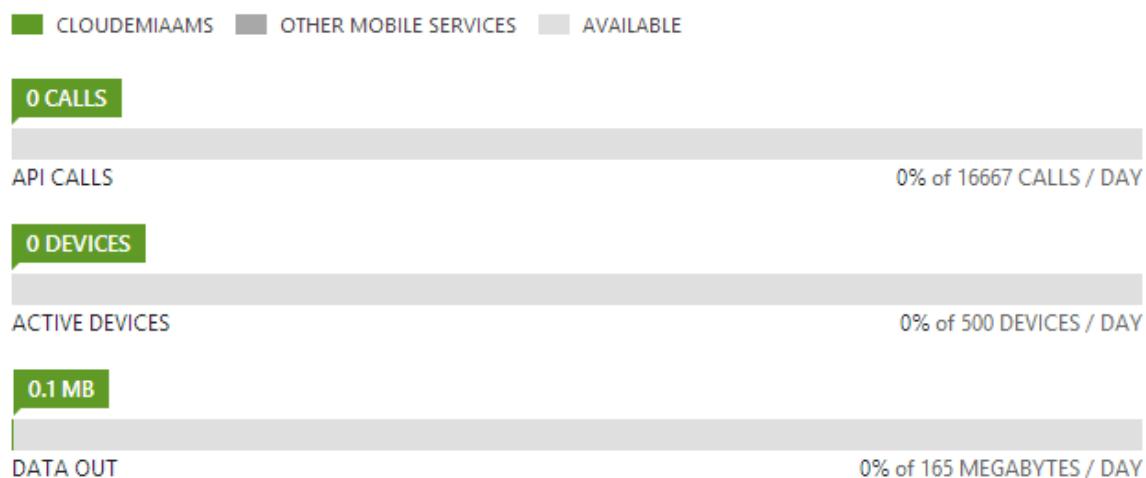
9. Pada dashboard kita dapat melihat limitasi dari services yang kita buat. Karena kita menggunakan versi gratis maka ada batasan pemanggilan API per hari, dan batasan devices yang mengakses per hari. Kita dapat melakukan scaling jika kebutuhan kita sudah melebihi batas yang ditentukan.

#### usage overview



10. Untuk melakukan scaling, anda dapat melakukan pengaturan di menu SCALE.

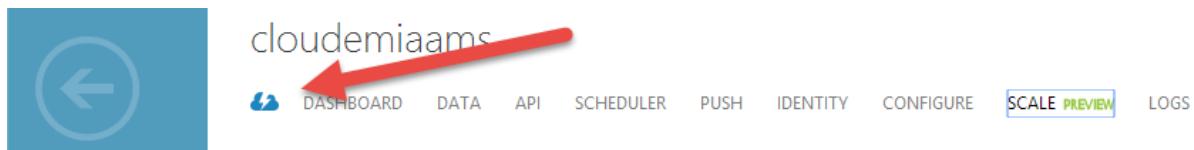
#### usage overview



#### Windows Store Client

Pada langkah selanjutnya kita akan membuat client berupa Windows Store App yang akan mengakses azure mobile services yang sebelumnya sudah kita buat.

1. Pilih logo biru di sebelah kiri atas dari menu



2. Maka akan muncul halaman berikut untuk memilih platform dari aplikasi client yang akan kita gunakan. Anda dapat memilih aplikasi Windows Store, Windows Phone, iOS, maupun aplikasi Android.

## cludemiaams

A screenshot of a web page titled "Your mobile service was created. Now let's connect it to an app." It features a large icon of a smartphone with an envelope on its screen. Below the icon is a checkbox labeled "Skip Quick Start the next time I visit". At the bottom, there is a row of buttons for "CHOOSE A PLATFORM": Windows Store (highlighted in blue), Windows Phone 8, iOS, Android, HTML/JavaScript, Xamarin, and PhoneGap.

## GET STARTED

- ▶ CREATE A NEW WINDOWS STORE APP
- ▶ CONNECT AN EXISTING WINDOWS STORE APP

3. Pilih Windows Store, kemudian pilih menu Create New Windows Store App.
4. Karena sebelumnya kita sudah menginstal Visual Studio 2012/2013, maka langkah selanjutnya adalah membuat contoh table. Pilih 'Create TodoItem Table'.

2

### Create a table

To store data in your mobile service, you need a table. Click the button below to create a TodoItem table for your starter project. To add additional tables later, click the Create button on the Data tab.

[Create TodoItem Table](#)

We have created the TodoItem table for you.

5. Kemudian pilih Javascript sebagai bahasanya, dan pilih download.

3

### Download and run your app

Download your personalized starter project, extract it, and then open it in Visual Studio. The app is pre-configured to work with your new mobile service.

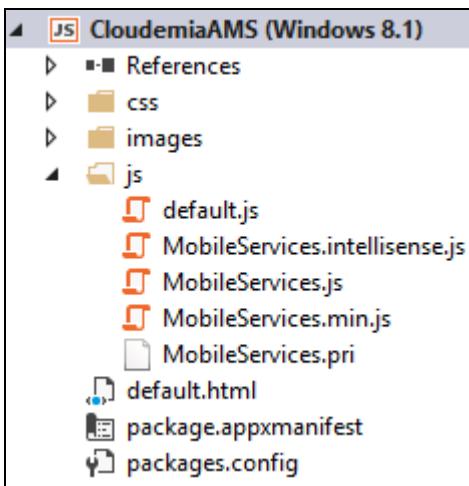
LANGUAGE:

[Download](#)

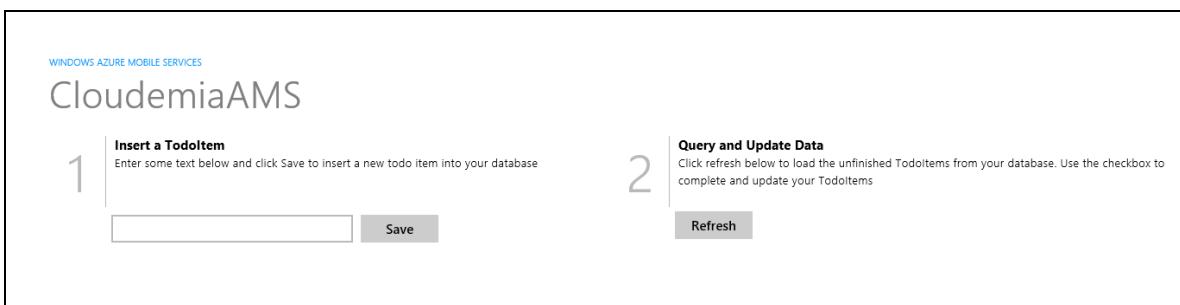
Run the project to start working with data in your mobile service.

6. Setelah itu ekstrak file yang sudah anda download, kemudian buka solusinya. (Misal pada aplikasi yang saya buat nama solusinya adalah CludemiaAMS.sln).

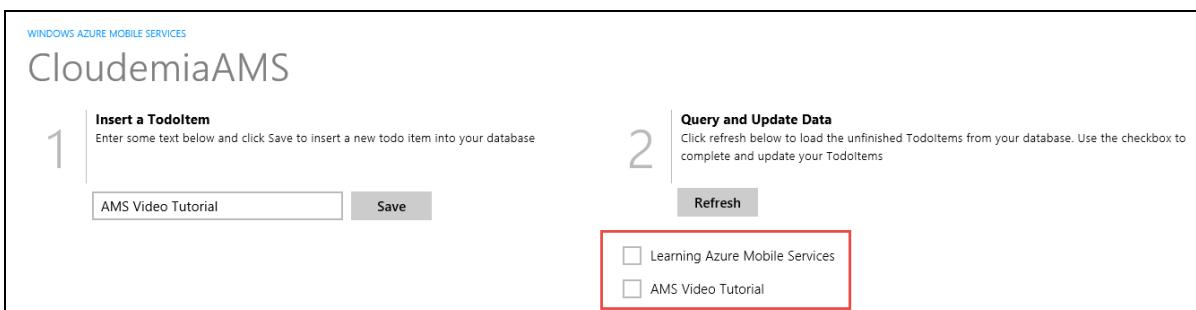
7. Pada folder 'js' di solution explorer anda dapat melihat file-file yang digunakan untuk mengakses azure mobile services.



8. Kemudian coba jalankan aplikasi tersebut dengan menekan tombol F5. Maka anda akan dapat melihat tampilan aplikasi Windows Store App sebagai berikut.



9. Pada aplikasi 'todo' tersebut anda dapat menambahkan beberapa contoh data. Anda juga dapat mencontreng todo yang sudah selesai dikerjakan, maka data tersebut secara otomatis akan dihilangkan.



10. Ketika anda menambahkan data todo pada aplikasi, maka data tersebut akan disimpan di azure mobile services.
11. Anda dapat membuka kembali windows azure management untuk melihat apakah data yang anda tambahkan sudah disimpan.
12. Pada mobile services yang sudah anda buat sebelumnya, pilih menu DATA.

The screenshot shows the 'DATA' section of the Azure Mobile Services interface. A red arrow points to the 'DATA' tab. Below it, the 'TodoItem' table is selected. The table has columns: id, text, complete, \_\_createdAt, \_\_updatedAt, and \_\_version. Two rows of data are visible:

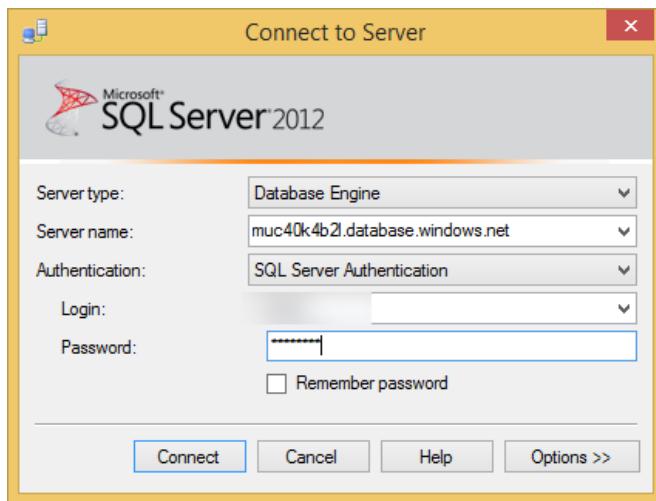
id	text	complete	__createdAt	__updatedAt	__version
048D41E1-58C1-4B7B-94...	AMS Video Tutorial	false	2014-03-26T17:19:40.408...	2014-03-26T17:19:40.408...	AAAAAAAAAB9Q=
FDB66FF5-E51B-4B10-B5...	Learning Azure Mobile S...	false	2014-03-26T17:19:05.769...	2014-03-26T17:19:05.769...	AAAAAAAAAB9I=

13. Pilih table TodoItem, maka anda akan dapat melihat data yang sudah anda masukan sebelumnya.

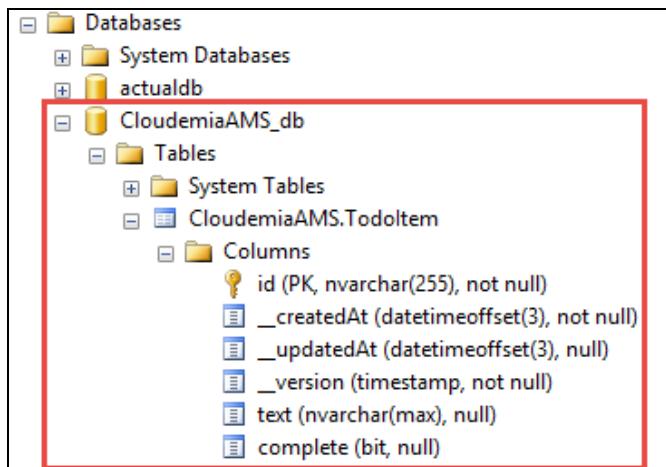
The screenshot shows the 'todoitem' table details view. It includes tabs for BROWSE, SCRIPT, COLUMNS, and PERMISSIONS. The BROWSE tab is selected, displaying the table structure and data. A red box highlights the table structure and the data rows.

id	text	complete	__createdAt	__updatedAt	__version
048D41E1-58C1-4B7B-94...	AMS Video Tutorial	false	2014-03-26T17:19:40.408...	2014-03-26T17:19:40.408...	AAAAAAAAAB9Q=
FDB66FF5-E51B-4B10-B5...	Learning Azure Mobile S...	false	2014-03-26T17:19:05.769...	2014-03-26T17:19:05.769...	AAAAAAAAAB9I=

14. Selain menggunakan cara diatas, anda juga dapat langsung mengakses database dengan cara pilih menu SQL Database, kemudian pilih database, pilih manage untuk mengijinkan ip komputer kita mengakses SQL Azure.
15. Kemudian buka SQL Management Studio dan masukan nama server, username, dan password.



16. Anda dapat melihat bahwa table TodoItem sudah dibuatkan beserta field.



17. Anda juga dapat mencermati file 'default.js' untuk melihat bagaimana cara untuk menambahkan data todoitem.
18. Selain menggunakan SQL Server Management studio untuk mengakses database, anda juga dapat menggunakan Sql Azure Manage Portal. Untuk menampilkan portal tersebut anda harus menginstal silverlight plugin.

SERVER  
muc40k4b2l

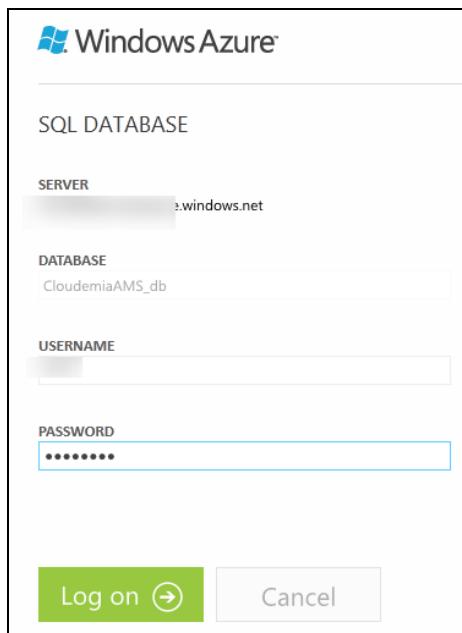
STATUS  
Online

COLLATION  
SQL\_Latin1\_General\_CI\_AS

EDITION  
Web

MANAGE URL  
[https://muc40k4b2l.database.windows.net/#\\$database=CloudemiaAMS\\_db](https://muc40k4b2l.database.windows.net/#$database=CloudemiaAMS_db)

19. Kemudian anda dapat login ke database pada portal tersebut.



20. Setelah login anda akan mendapatkan tampilan sebagai berikut. Anda dapat menambahkan table baru, membuat view, stored procedure, dll.

A screenshot of the Microsoft Azure portal showing the `CloudemiaAMS_db` database. The top navigation bar has tabs for **Tables**, **Views**, and **Stored Procedures**. The **Tables** tab is selected. Below the tabs is a search bar labeled `Search by table name`. A table lists one table:

Schema Name	Table Name	Table Size	Row Count	Actions
CloudemiaAMS	Todolitem	16.00 KB	3	Edit  Dependencies

At the bottom of the table are buttons for **+ New table** and **- Drop table**.

### Mengakses REST Services dari Table di Mobile Services

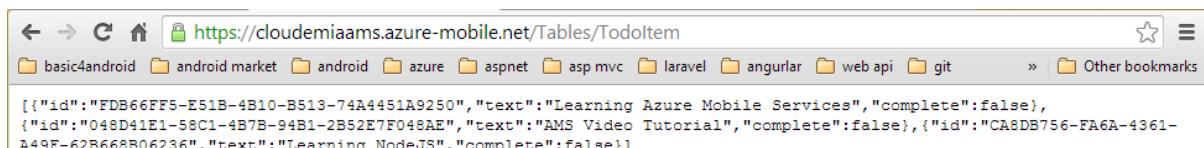
Anda dapat mengakses table `Todolitem` yang sudah anda buat secara REST. Untuk mengakses table tersebut anda harus memeriksa permission dari table tersebut, caranya anda dapat memilih table `Todolitem` kemudian pilih menu 'Permission'.

A screenshot of the Azure portal showing the `todoitem` table permissions. The top navigation bar has tabs for **BROWSE**, **SCRIPT**, **COLUMNS**, and **PERMISSIONS**. The **PERMISSIONS** tab is highlighted with a red box. Below the tabs is a link labeled `table permissions`.

Agar semua orang dapat mengakses dan menquery data, anda harus mengubah pengaturan READ PERMISSION menjadi Everyone.

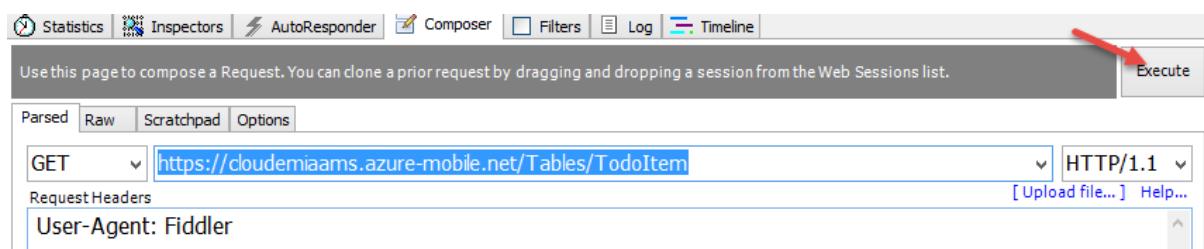


Setelah pengaturan selesai tekan SAVE, kemudian anda bisa mencoba untuk membaca data menggunakan REST. Pada browser tambahkan url berikut :

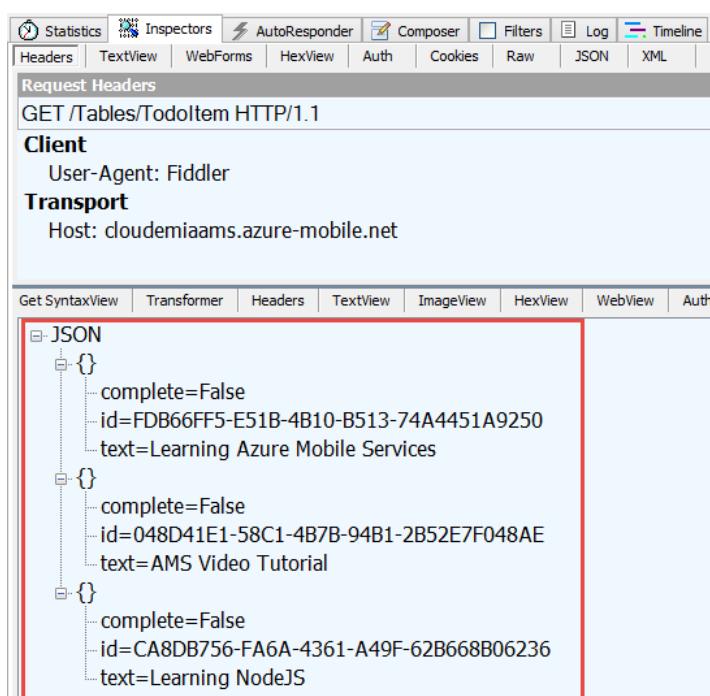


Maka anda dapat melihat bahwa data dengan format JSON yang berisi semua data dari table Todolitem akan ditampilkan.

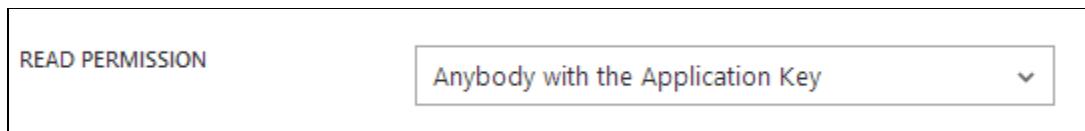
Anda juga dapat menggunakan program seperti fiddler untuk mengakses REST services tersebut. Berikut contoh penggunaan fiddler.



Dapat dilihat file JSON yang akan dihasilkan sebagai berikut :



Cara seperti diatas dapat dilakukan namun kurang aman karena semua orang dapat mengakses table TodoItem secara REST. Coba anda kembalikan pengaturan READ PERMISSION menjadi 'Anybody with the Application Key'.

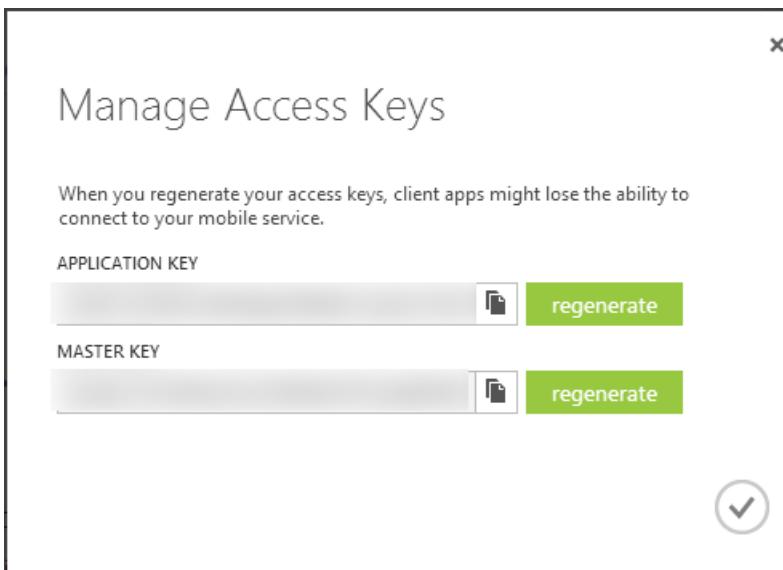


Kemudian buka fiddler kembali, dan akses REST url seperti sebelumnya.

A screenshot of the Fiddler web debugger. On the left, a list of network requests shows item 4 as a 401 Unauthorized response to a GET request for '/Tables/TodoItem'. The main pane displays the request details: 'Request Headers' (GET /Tables/TodoItem HTTP/1.1), 'Client' (User-Agent: Fiddler), 'Transport' (Host: cloudemiaams.azure-mobile.net), and 'Response Headers' (HTTP/1.1 401 Unauthorized). A red arrow points to the 'Unauthorized' status code in the response headers. Below the headers, the 'Cache' section shows Cache-Control: no-cache and Date: Thu, 27 Mar 2014 03:35:36 GMT.

Sekarang pesannya adalah Unauthorized karena permission sudah dirubah, anda harus menambahkan application key untuk melakukan request. Untuk mendapatkan application key anda dapat memilih tombol MANAGE KEYS.

A screenshot of the Azure mobile service endpoint status page. It shows a message: 'mobile service endpoint status PREVIEW You have not configured mobile service endpoint monitoring.' Below this is a 'CONFIGURE MOBILE SERVICE ENDPOINT MONITORING' button with a right-pointing arrow. At the bottom, there are two buttons: 'MANAGE KEYS' with an 'i' icon and 'DELETE' with a trash can icon. A large red arrow points to the 'MANAGE KEYS' button.

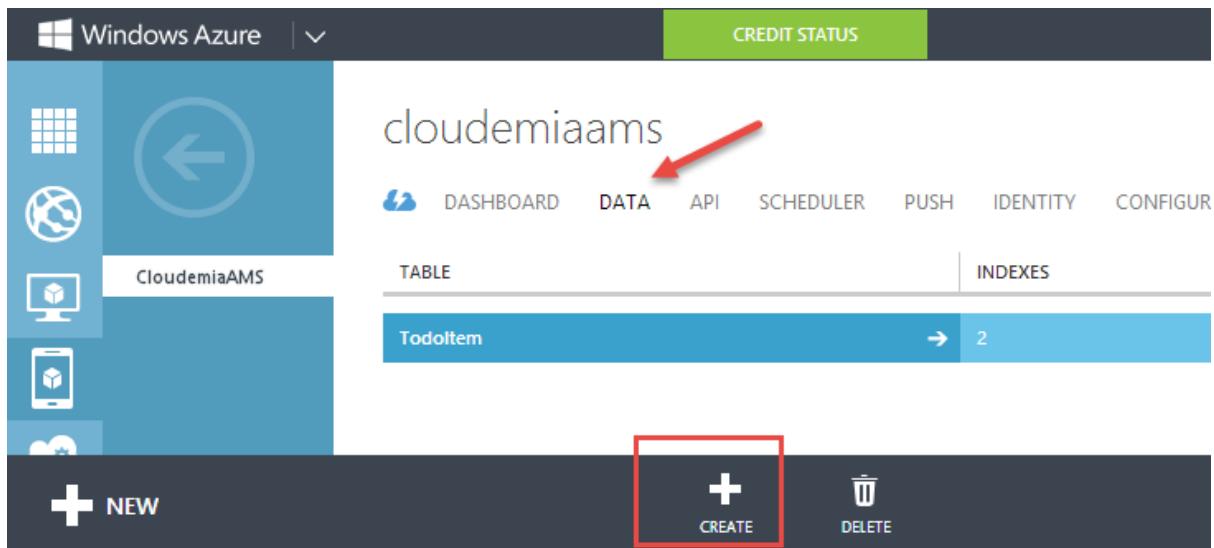


Copy application key, kemudian tambahkan application key tersebut ketika kita melakukan request ke server.

Setelah ditambahkan application key, maka anda dapat melihat bahwa request tersebut berhasil dan mengembalikan status 200.

## Menambahkan Data dan Table Baru pada Azure Mobile Services

Untuk menambahkan table pada windows azure mobile services, sebenarnya anda tidak perlu menggunakan SQL Server Management studio atau fasilitas managed database di SQL Azure. Anda dapat menambahkan table lewat menu Mobile Services.



Pertama kita akan menambahkan table baru dengan nama 'Courses'.

A screenshot of the 'Create New Table' dialog. The title is 'MOBILE SERVICES: DATA' and 'Create New Table'. The 'TABLE NAME' field contains 'Courses' and is highlighted with a red border. Below it, a message says 'Tables have changed; your existing client code may need to be modified.' A link 'Learn more' is provided. There are four permission dropdowns: 'INSERT PERMISSION' (set to 'Anybody with the Application Key'), 'UPDATE PERMISSION' (set to 'Anybody with the Application Key'), 'DELETE PERMISSION' (set to 'Anybody with the Application Key'), and 'READ PERMISSION' (set to 'Anybody with the Application Key'). In the bottom right corner, there is a circular icon with a checkmark.

Secara default table yang kita buat akan memiliki empat column sebagai berikut :

## COURSES

BROWSE    SCRIPT    COLUMNS    PERMISSIONS

COLUMN NAME	TYPE	INDEX
id	string	✓ Indexed
__createdAt	date	✓ Indexed
__updatedAt	date	
__version	timestamp (MSSQL)	

Jika anda buka table tersebut di SQL Server Management studio, dapat dilihat bahwa column Id akan digunakan sebagai primary key.

```

    [+] CludemiaAMS_db
      [+] Tables
        [+] System Tables
        [+] CludemiaAMS.Courses
          [+] Columns
            [+] id (PK, nvarchar(255), not null)
            [+] __createdAt (datetimeoffset(3), not null)
            [+] __updatedAt (datetimeoffset(3), null)
            [+] __version (timestamp, not null)
  
```

Tambahkan 3 column baru dengan nama Title – string, Description- string, dan Complete – boolean.

COLUMN NAME	TYPE	INDEX
id	string	✓ Indexed
__createdAt	date	✓ Indexed
__updatedAt	date	
__version	timestamp (MSSQL)	
title	string	
description	string	
complete	boolean	



Setelah menambahkan column tersebut, sekarang kita akan mencoba untuk menambahkan data kedalam table yang sudah kita buat menggunakan fiddler. Untuk menambahkan data kita akan menggunakan method POST.

The screenshot shows the Fiddler interface with a red box highlighting the request details. The method is POST, the URL is https://cloudemiaams.azure-mobile.net/Tables/Courses, and the protocol is HTTP/1.1. The Request Headers include User-Agent: Fiddler, X-ZUMO-APPLICATION: cObFFsPHEhzbGXkgIAbBQxrLpeyTLk62, Host: cloudemiaams.azure-mobile.net, and Content-Length: 113. The Request Body contains the JSON payload: {"Title": "ASP.NET MVC", "Description": "Introduction Course tentang penggunaan MVC framework", "Complete": "false"}.

Jika data berhasil ditambah maka akan muncul status kode '201 Created'.

The screenshot shows the Fiddler interface with a red box highlighting the response details. The status code is HTTP/1.1 201 Created. The response body contains the JSON data: {"Title": "ASP.NET MVC", "Description": "Introduction Course tentang penggunaan MVC framework"}. The response headers include Cache-Control: no-cache and Date: Thu, 27 Mar 2014 04:28:54 GMT.

Jika kita lihat data yang kita inputkan pada windows azure adalah sebagai berikut :

courses					
BROWSE	SCRIPT	COLUMNS	PERMISSIONS		
id	title	description	complete	__createdAt	__updatedAt
2A2D8134-3A5A-48CE-95...	ASP.NET MVC	Introduction Course tent...	false	2014-03-27T04:34:50.513...	2014-03-27T04:34:50.513...

Untuk melakukan update data anda dapat menggunakan method PUT atau PATCH. Dengan menggunakan PATCH anda cukup mengirimkan field yang akan diupdate. Sebagai contoh kita akan mengupdate field 'Complete' menjadi 'true' dari data dengan id tertentu.

Use this page to compose a Request. You can clone a prior request by dragging and dropping a session from the Web Sessions list.

Parsed Raw Scratchpad Options

PATCH https://cloudemiaams.azure-mobile.net/Tables/Courses/2A2D8134-3A5A-48CE-95E2-D6D77C84D673

Request Headers

User-Agent: Fiddler  
X-ZUMO-APPLICATION: cObFFsPHEhzBGXkgIAbBQxrLpeyTLk62  
Host: cloudemiaams.azure-mobile.net  
Content-Length: 19

Request Body

```
{"Complete": "true"}
```

Jika anda lihat record setelah data diupdate adalah sebagai berikut :

BROWSE	SCRIPT	COLUMNS	PERMISSIONS				
		id	title	description	complete	__createdAt	__updatedAt
2A2D8134-3A5A-48CE-95...	ASP.NET MVC		Introduction Course tentang pen...		true	2014-03-27T04:34:50.513...	2014-03-27T04:45:42.674...

Untuk mendelete data anda dapat menggunakan method DELETE.

Use this page to compose a Request. You can clone a prior request by dragging and dropping a session from the Web Sessions list.

Parsed Raw Scratchpad Options

DELETE https://cloudemiaams.azure-mobile.net/Tables/Courses/2A2D8134-3A5A-48CE-95E2-D6D77C84D673

Request Headers

User-Agent: Fiddler  
X-ZUMO-APPLICATION: cObFFsPHEhzBGXkgIAbBQxrLpeyTLk62  
Host: cloudemiaams.azure-mobile.net  
Content-Length: 0

Maka record tersebut akan didelete sehingga kita sudah tidak mempunyai record yg tersisa.

Tambahkan beberapa record lagi sebagai data contoh karena kita akan mencoba beberapa variasi query dapat dilakukan.

id	title	description	complete
00DF8ACA-6E5F-4C63-BB...	Azure Website	Course Azure Website	true
65F55806-F11B-41F8-B45...	ASP.NET MVC	Course ASP.NET MVC 5	false
B7A6317E-453A-41A9-A72...	Azure Mobile Services	Course Azure Mobile Serv...	false

Kita dapat menggunakan sintaks ODATA untuk memfilter record sesuai dengan kebutuhan. Untuk menampilkan data yg field 'Complete' bernilai 'true' tambahkan sintaks berikut pada url.

`https://cloudemiaams.azure-mobile.net/Tables/Courses?$filter=Complete eq true`

Parsed Raw Scratchpad Options

GET https://cloudemiaams.azure-mobile.net/Tables/Courses?\$filter=Complete%20eq%20true

Request Headers

User-Agent: Fiddler  
X-ZUMO-APPLICATION: cObFFsPHEhzbgXkgIAbBQxrLpeyTLk62  
Host: cloudemiaams.azure-mobile.net  
Content-Length: 0

setelah di-  
encode

Hasil query dalam format JSON adalah sebagai berikut :

JSON

```
{
    "complete": true,
    "description": "Course Azure Website",
    "id": "00DF8ACA-6E5F-4C63-BBE8-DA0430CD6F3F",
    "title": "Azure Website"
}
```

Untuk contoh2 query pada ODATA dapat dilihat pada alamat berikut :

<http://www.odata.org/documentation/odata-version-2-0/uri-conventions/>

Misal untuk mengurutkan data course berdasarkan title, anda dapat menggunakan keyword

\$orderby sebagai berikut:

Parsed Raw Scratchpad Options

GET https://cloudemiaams.azure-mobile.net/Tables/Courses?\$orderby=title

Request Headers

User-Agent: Fiddler  
X-ZUMO-APPLICATION: cObFFsPHEhzbgXkgIAbBQxrLpeyTLk62  
Host: cloudemiaams.azure-mobile.net  
Content-Length: 0

## Menambahkan Script Pada Proses CRUD

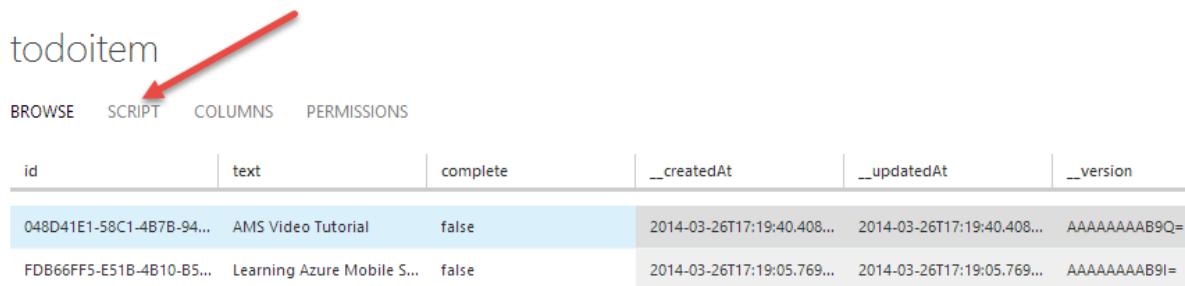
Anda juga dapat menambahkan script untuk mengubah atau memfilter data sebelum atau sesudah perintah CRUD dijalankan. Dengan script anda dapat bekerja dengan bermacam-macam objek data sebagai berikut :

- Request : execute default operation
- Query : filtering, paging, ordering
- Table : CRUD operation dan query

- MSSQL : Direct SQL Statement

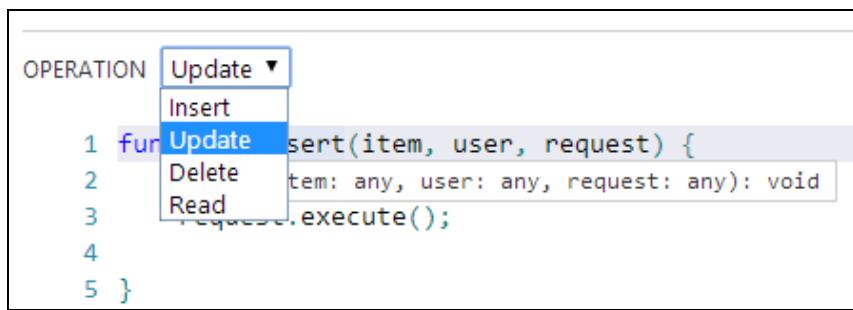
Untuk mencoba menggunakan script ikuti langkah berikut :

Pada services yang sebelumnya anda buat buka menu DATA. Kemudian pilih table Todolitem.



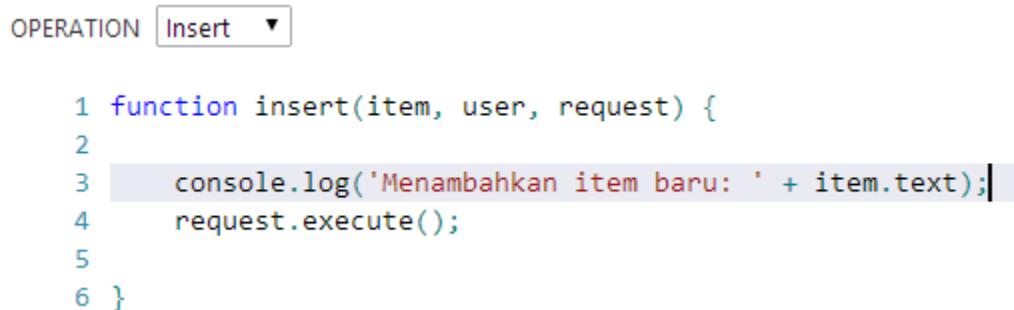
todoitem					
BROWSE	SCRIPT	COLUMNS	PERMISSIONS		
id	text	complete	__createdAt	__updatedAt	__version
048D41E1-58C1-4B7B-94...	AMS Video Tutorial	false	2014-03-26T17:19:40.408...	2014-03-26T17:19:40.408...	AAAAAAAAAB9Q=
FDB66FF5-E51B-4B10-B5...	Learning Azure Mobile S...	false	2014-03-26T17:19:05.769...	2014-03-26T17:19:05.769...	AAAAAAAAAB9I=

Kemudian pilih menu SCRIPT, anda dapat menambahkan script ketika operasi insert, update, delete, dan read.



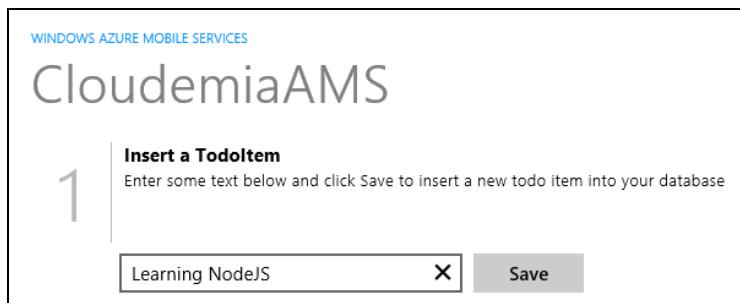
```
OPERATION Update ▾
Insert
1 function insert(item, user, request) {
2     console.log('Menambahkan item baru: ' + item.text);
3     request.execute();
4 }
5 }
```

Disini perintah script yang dijalankan menggunakan nodejs, Sebagai contoh pada script tambahkan kode berikut :



```
OPERATION Insert ▾
1 function insert(item, user, request) {
2
3     console.log('Menambahkan item baru: ' + item.text);
4     request.execute();
5
6 }
```

Kemudian tekan Save, dan jalankan aplikasi Todolitem. Tambahkan record baru kedalam aplikasi anda.



WINDOWS AZURE MOBILE SERVICES

# CloudemiaAMS

Insert a Todolitem

Enter some text below and click Save to insert a new todo item into your database

1 Learning NodeJS

Setelah anda tambahkan, coba lihat menu log untuk melihat log yang dituliskan ketika perintah insert dijalankan.

## cloudemiaams



Maka anda dapat melihat log yang kita create ketika perintah insert di table TodoItem dijalankan.

LEVEL	MESSAGE	SOURCE	TIME STAMP
Information	Menambahkan item baru: Learning NodeJS	TodoItem/insert	Thu Mar 27 2014, 1:07:30 AM

Jika anda sudah menginstall azure cli maka anda juga dapat menggunakan command prompt untuk melihat list daftar mobile services dan script yang sudah anda buat.

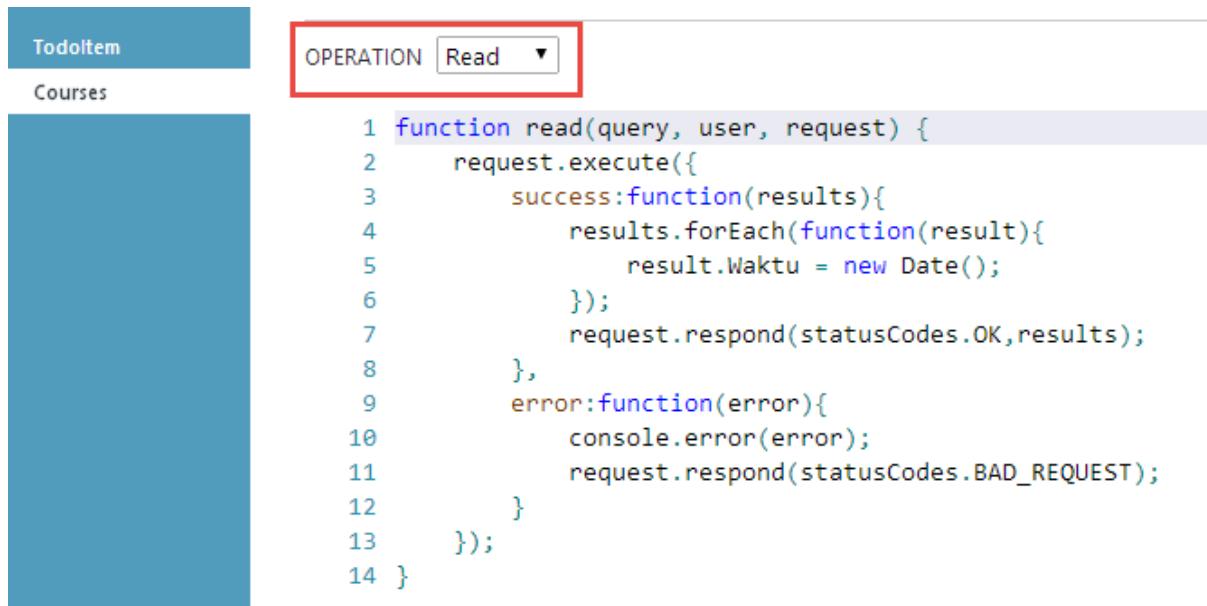
```
PS C:\Users\Erick\Desktop> azure mobile list
info: Executing command mobile list
+ Getting list of mobile services
data:      Name          State    URL
data:      -----        -----
data:      CloudemiaAMS Ready   https://cloudemiaams.azure-mobile.net/
info: mobile list command OK

PS C:\Users\Erick\Desktop> azure mobile script list
info: Executing command mobile script list
+ Retrieving choices
help: Mobile Service:
  1) CloudemiaAMS
  : 1
+ Retrieving script information
info: Table scripts
data:      Name          Size
data:      -----        -----
data:      table/TodoItem.insert 484
info: There are no shared scripts. Create scripts using the
info: There are no scheduled job scripts. Create scheduled jo
info: There are no custom API scripts. Create APIs using the
info: mobile script list command OK
```

## Menggunakan Objek Request pada Script

Pada contoh dibawah ini akan ditunjukan cara penggunaan objek request untuk menambahkan result data dari server.

Untuk itu pilih table 'Course' kemudian pilih SCRIPT dan tambahkan kode berikut :

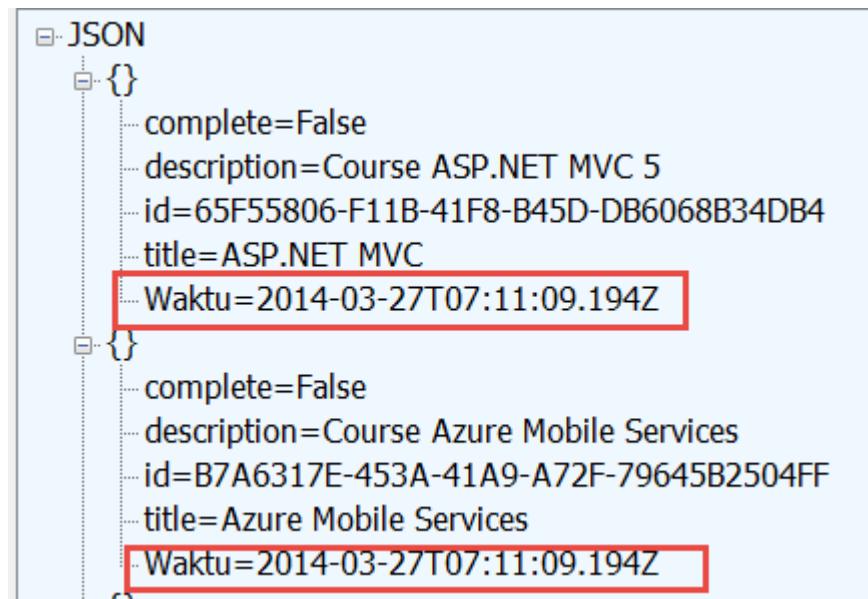


```

1 function read(query, user, request) {
2     request.execute({
3         success:function(results){
4             results.forEach(function(result){
5                 result.Waktu = new Date();
6             });
7             request.respond(statusCodes.OK,results);
8         },
9         error:function(error){
10            console.error(error);
11            request.respond(statusCodes.BAD_REQUEST);
12        }
13    });
14 }

```

Kemudian coba request api dari table Courses menggunakan fiddler. Jika request berhasil maka akan ditambahkan field baru dengan nama ‘Waktu’ yang berisi tanggal sekarang.



```

{
  "id": "65F55806-F11B-41F8-B45D-DB6068B34DB4",
  "title": "ASP.NET MVC",
  "description": "Course ASP.NET MVC 5",
  "complete": false,
  "Waktu": "2014-03-27T07:11:09.194Z"
}

{
  "id": "B7A6317E-453A-41A9-A72F-79645B2504FF",
  "title": "Azure Mobile Services",
  "description": "Course Azure Mobile Services",
  "complete": false,
  "Waktu": "2014-03-27T07:11:09.194Z"
}

```

Seperti anda lihat output yang dikirimkan mempunyai satu field tambahan ‘Waktu’ yang dibuat pada saat request. Cara ini dapat digunakan jika anda ingin menambahkan informasi tambahan yang dikirimkan ke client pada saat request.

Jika anda lihat di struktur table maka dapat terlihat struktur table tersebut tidak berubah, tidak ada tambahan field baru dengan nama ‘Waktu’.

### Menambahkan Field baru dengan Script

Jika konfigurasi dynamic schema diaktifkan, maka kita akan dapat menambahkan field pada table secara dinamis (menggunakan script). Cara ini mirip dengan penggunaan entity framework code first.

The screenshot shows the 'CONFIGURE' tab selected in the top navigation bar. Below it, under 'database settings', there are two sections: 'SQL DATABASE' (CloudemiaAMS\_db) and 'SQL SERVER' (muc40k4b2l). A red arrow points to the 'CONFIGURE' tab. A red box highlights the 'dynamic schema' section, which contains a toggle switch labeled 'ENABLE DYNAMIC SCHEMA' with 'ON' and 'OFF' options.

Pada contoh dibawah ini akan ditunjukkan bagaimana cara untuk menambahkan field 'DateCourseCreated' secara dinamis dan menambahkan nilai kedalam field tersebut saat perintah insert dijalankan.

The screenshot shows the 'TodoItem' application interface. On the left, there's a sidebar with 'Courses'. In the main area, the 'OPERATION' dropdown is set to 'Insert'. Below it, a code editor displays the following JavaScript function:

```
1 function insert(item, user, request) {  
2  
3     item.DateCourseCreated = new Date();  
4     request.execute();  
5  
6 }
```

Buka fiddler, kemudian tambahkan data baru kedalam table Courses.

```
Parsed Raw Scratchpad Options  
POST https://cloudemiaams.azure-mobile.net/Tables/Courses  
Request Headers  
User-Agent: Fiddler  
X-ZUMO-APPLICATION: TLk62  
Host: cloudemiaams.azure-mobile.net  
Content-Length: 91  
  
Request Body  
{"Title": "ASP.NET Web API", "Description": "Course about ASP.NET Web API", "Complete": "false"}
```

Selesai itu jalankan method GET untuk melihat output JSON yang dikirimkan.

```
{  
    ...  
    complete=True  
    DateCourseCreated=(null)  
    description=Course Azure Website  
    id=00DF8ACA-6E5F-4C63-BBE8-DA0430CD6F3F  
    title=Azure Website  
    Waktu=2014-03-27T07:40:48.403Z  
}  
  
{  
    ...  
    complete=False  
    DateCourseCreated=2014-03-27T07:38:56.836Z  
    description=Course about ASP.NET Web API  
    id=CAB76399-4793-4414-A28C-A2CF568C7C47  
    title=ASP.NET Web API  
    Waktu=2014-03-27T07:40:48.403Z  
}
```

Dapat anda lihat bahwa disetiap record akan ada satu field baru dengan nama 'DateCourseCreated'.

Jika anda lihat pada struktur table, maka field tersebut juga sudah ditambahkan secara permanen.

The screenshot shows the database structure for the 'CloudemiaAMS\_db'. It includes a 'Tables' section with 'System Tables' and 'CloudemiaAMS.Courses'. Under 'CloudemiaAMS.Courses', there are 'Columns' listed. One of the columns, 'DateCourseCreated', is highlighted with a red box.

### Menggunakan Query Object pada SCRIPT

Anda juga dapat memanipulasi objek Query dengan menggunakan SCRIPT. Pada contoh dibawah ini akan ditunjukkan bagaimana cara memanipulasi objek query.

The screenshot shows the 'TodoItem' query object with an 'OPERATION' set to 'Read'. The script content is:

```

1 function read(query, user, request) {
2     query.where({Complete:false}).orderByDescending('Title');
3     request.execute();
4 }

```

Setelah script diatas disimpan, coba jalankan method GET untuk request data dengan menggunakan fiddler. Maka hasil output setelah objek query dimanipulasi adalah sebagai berikut:

The screenshot shows a JSON response with two course objects. Both objects have the 'complete' field set to 'False'. The first object has a title of 'Azure Mobile Services' and the second has a title of 'ASP.NET Web API'.

```

{
    "value": [
        {
            "complete": false,
            "DateCourseCreated": null,
            "description": "Course Azure Mobile Services",
            "id": "B7A6317E-453A-41A9-A72F-79645B2504FF",
            "title": "Azure Mobile Services"
        },
        {
            "complete": false,
            "DateCourseCreated": "2014-03-27T07:38:56.836Z",
            "description": "Course about ASP.NET Web API",
            "id": "CAB76399-4793-4414-A28C-A2CF568C7C47",
            "title": "ASP.NET Web API"
        }
    ]
}

```

Dapat dilihat pada output bahwa data yang ditampilkan adalah data yang field ‘complete’ bernilai ‘false’.

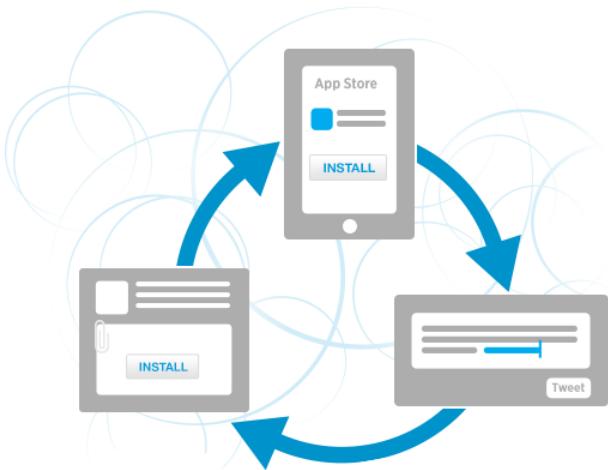
## Bab 6 -Identity dengan Azure Mobile Services

Pada sub bab ini akan dibahas bagaimana cara penggunaan Identity pada Windows Azure Mobile services. Dengan menggunakan services ini user dapat login dengan menggunakan account (username/password) dari provider seperti google, facebook, microsoft.

### Membuat Twitter Identity Provider

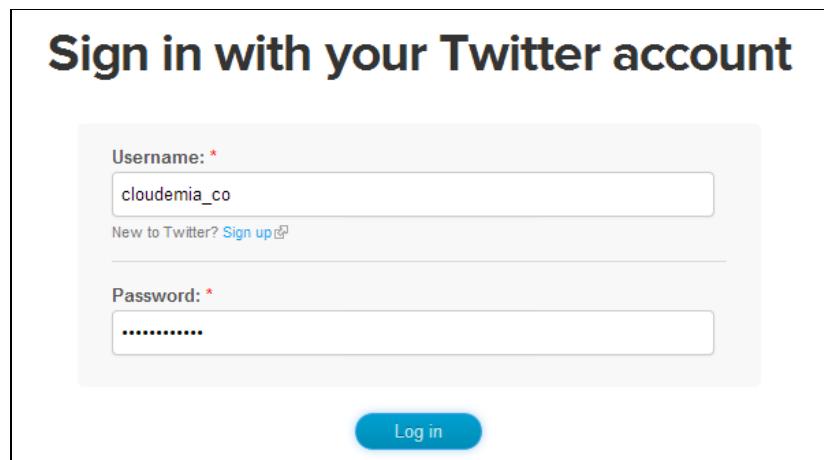
Untuk membuat twitter provider langkah yang pertama kali harus dilakukan adalah :

1. Masuk ke halaman <http://dev.twitter.com>



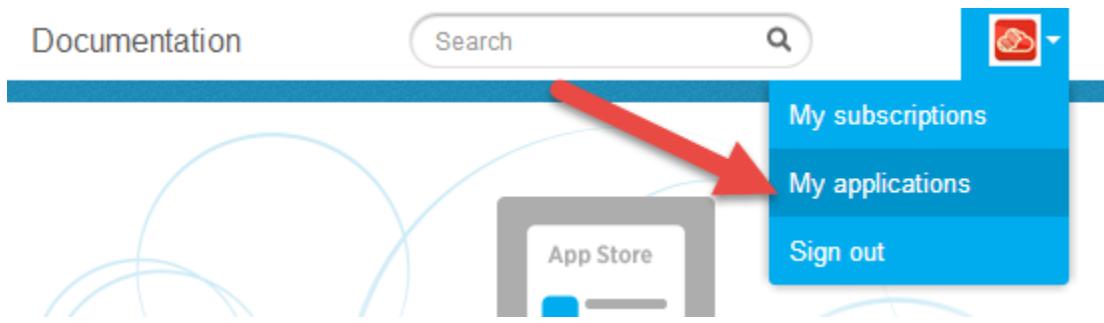
The screenshot shows the Twitter Developers website. At the top, there's a navigation bar with links for Developers, API Health, Blog, Discussions, Documentation, a search bar, and a 'Sign in' button (which is highlighted with a red box). Below the navigation, there's a large heading 'More downloads for your app with Twitter Cards'. A sub-section explains that Twitter Cards offer a fast and easy way to grow your user base for mobile apps by adding markup to pages. It shows three icons: a smartphone labeled 'App Store' with an 'INSTALL' button, a computer screen labeled 'INSTALL', and a tweet card labeled 'Tweet'. Blue arrows connect these three icons in a circular loop, illustrating the process of users viewing tweets and being prompted to download and install the app.

2. Kemudian pilih sign-in untuk login dengan twitter account



The screenshot shows a 'Sign in with your Twitter account' form. It has two input fields: 'Username: \*' containing 'cloudemia\_co' and 'Password:' containing a series of dots. Below the fields is a blue 'Log in' button.

3. Kemudian pilih My Application



4. Kemudian pilih new application untuk membuat app baru.

A screenshot of the Twitter Apps page. The title 'Twitter Apps' is at the top. Below it, a message says 'You don't currently have any Twitter Apps.' A red arrow points to the 'Create New App' button.

5. Tambahkan informasi dari aplikasi yang akan kita buat. Arahkan callback ke mobile services anda.

A screenshot of the 'Application details' form. It has four sections: 'Name' (CludemiaAMS), 'Description' (Sample Cludemia Azure Mobile Services), 'Website' (http://cludemia.azurewebsites.net), and 'Callback URL' (https://cludemiaams.azure-mobile.net/). Each input field is highlighted with a red border. Below each section is a descriptive note.

6. Setelah services selesai dibuat akan muncul halaman yang berisi informasi dari aplikasi yang kita buat.

Your application has been created. Please take a moment to review and adjust your application's settings.

## CludemiaAMS

Test OAuth

Details    Settings    API Keys    Permissions

Sample Cludemia Azure Mobile Services  
http://cludemia.azurewebsites.net

7. Kemudian catat API key dan API secret untuk dimasukan kedalam Azure Mobile Services.

## CludemiaAMS

Details    Settings    **API Keys**    Permissions

*Keep the "API secret" a secret. This key should never be human-readable in your application.*

API key	[REDACTED]
API secret	[REDACTED]
Access level	Read-only (modify app permissions)
Owner	cludemia_co
Owner ID	2261673546

8. Pada Azure Mobile Services, pilih menu Identity, kemudian tambahkan consumer key (API key) dan consumer secret (API secret)

twitter settings

CONSUMER KEY	[REDACTED]	[REDACTED]
CONSUMER SECRET	[REDACTED]	[REDACTED]

## Membuat Facebook Identity Provider

1. Untuk membuat facebook developer account anda dapat mengunjungi url berikut :  
<https://developers.facebook.com/apps>
2. Silahkan login dengan facebook account anda dan buat aplikasi baru.

**Create a New App**

Get started integrating Facebook into your app or website

Display Name  
The name of your app or website

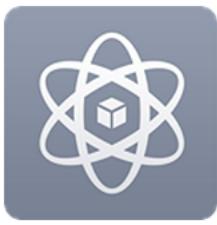
Namespace  
A unique identifier for your app (optional)

Category  
Choose a Category ▾

By proceeding, you agree to the [Facebook Platform Policies](#)

3. Setelah aplikasi berhasil dibuat, masuk ke tampilan dashboard untuk mengambil App ID dan App Secret.

**Dashboard**

 **CloudemiaAMS •**  
This app is public and available to all users [?]

App ID  App Secret

**User Stats**

Active Users Trend

Monthly Facebook Connected Users  Weekly Facebook Connected Users  Daily Facebook Connected Users

4. Kemudian masukan informasi tersebut pada halaman Identity.

facebook settings

---

APP ID/API KEY	1470912636468399
APP SECRET	5db398ef21970d6e2525c4d17f3a031d

### Menambahkan Restriction Permission pada Table

Langkah selanjutnya adalah menambahkan permission pada table yang sudah kita buat sehingga user harus melakukan login terlebih dahulu sebelum dapat mengakses informasi pada table. Pada contoh dibawah ini kita akan menambahkan autentikasi menggunakan facebook login yang sebelumnya sudah anda buat.

Misal pada table ‘Todoitem’ yang sudah anda buat tambahkan permission sebagai berikut agar pengguna harus login terlebih dahulu sebelum dapat mengakses table Todoitem.

### table permissions

---

INSERT PERMISSION	Only Authenticated Users
UPDATE PERMISSION	Only Authenticated Users
DELETE PERMISSION	Only Authenticated Users
READ PERMISSION	Only Authenticated Users

Kemudian buka project Windows Store App dengan nama ClodemiaAMS yang sebelumnya sudah anda buat (contoh aplikasi yang didownload dari sample windows azure).

Tambahkan kode berikut pada file default.js.

Pada event **app.onactivated** hapus atau tandai method **refreshTodoItems()** dan tambahkan kode berikut dibawah method tersebut.

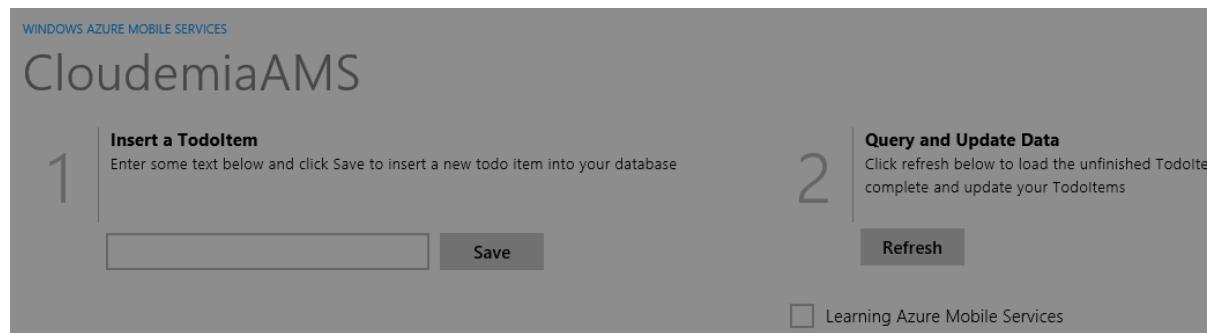
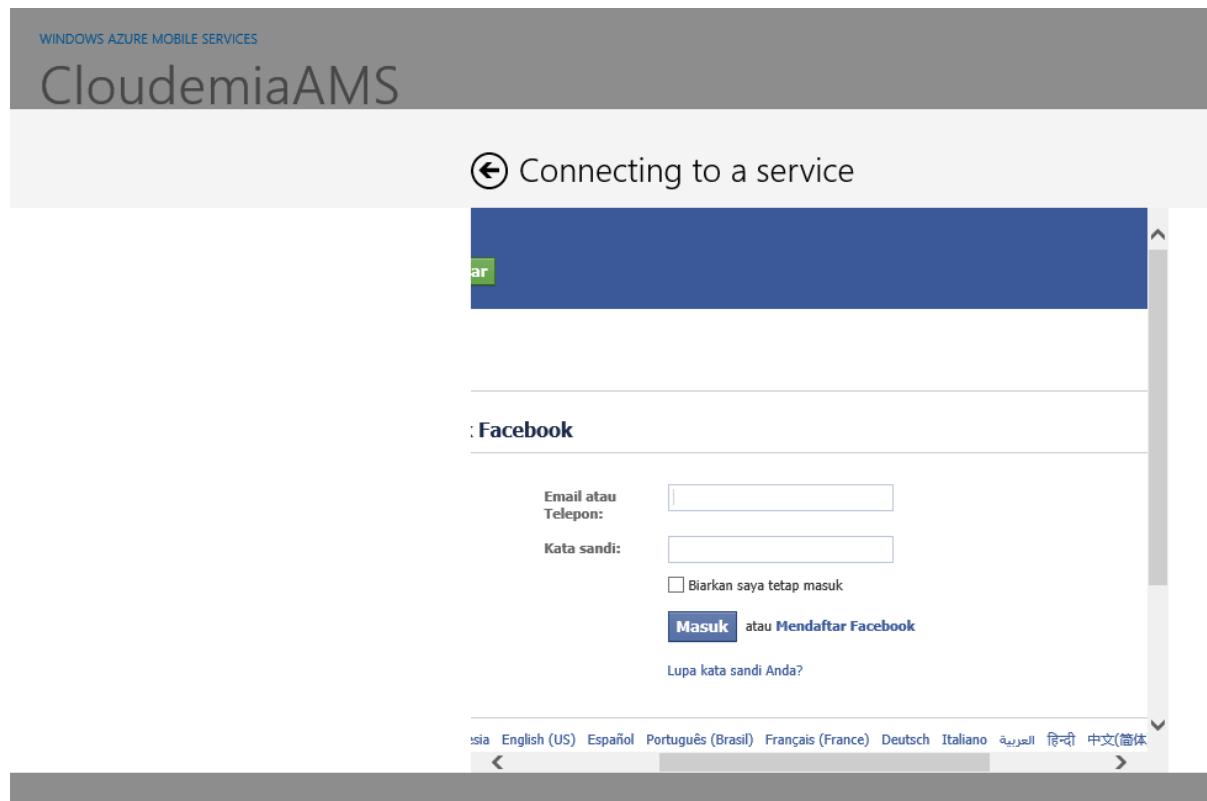
```
var userId = null;
```

```
var login = function() {
    return new WinJS.Promise(function(complete) {
        client.login("facebook").done(function(results) {
            userId = results.userId;
            refreshTodoItems();
            var message = "Anda sekarang login sebagai : " +
userId;
            var dialog = new
Windows.UI.Popups.MessageDialog(message);
            dialog.showAsync().done(complete);
        },
        function(error) {
            userId = null;
            var dialog = new
Windows.UI.Popups.MessageDialog("Terjadi error ketika login", "Login
Required");
            dialog.showAsync().done(complete);
        });
    });
}

var authenticate = function() {
    login().then(function() {
        if (userId == null) {
            authenticate();
        }
    });
}

authenticate();
```

Kemudian jalankan program, maka sebelum anda dapat mengakses aplikasi untuk melihat dan menambahkan todo list, anda wajib untuk login menggunakan account facebook terlebih dahulu. Anda juga dapat bereksperimen dengan provider yang lain seperti microsoft account atau twitter account.

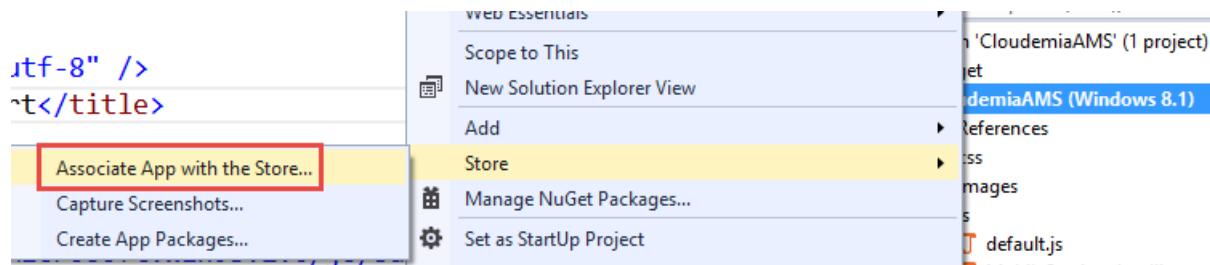


## Bab 7 - Menggunakan Push Notification

Pada bab ini kita akan mencoba menggunakan fitur push notification yang ada pada Azure Mobile Services. Dengan menggunakan fitur ini maka anda dapat mengirimkan push notification ke mobile device. Azure mobile services dapat menggunakan beberapa layanan push seperti APN (Apple Push Notification), GCM (Google), WNS (Windows 8), dan MPNS (Windows Phone).

Buka aplikasi Windows Apps Todo List yang sudah anda buat sebelumnya.

Kemudian klik kanan pada project, pilih 'store' dan 'associate app with the store'. Untuk melakukan langkah ini anda harus sudah punya account untuk Windows Store Developer, kemudian anda dapat menambahkan apps baru di portal Windows Store Developer.



Setelah berhasil login pilih aplikasi Windows Store yang akan diasosiasikan dengan project yang sedang kita buat. Pada contoh dibawah ini saya sudah membuat apps baru dengan nama CludemiaAMS.

The screenshot shows a 'Select an app name' dialog box. It lists existing app names: 'ActTodoList' and 'CludemiaAMS'. The 'CludemiaAMS' entry is highlighted with a red box. Below the list, there is a 'Reserve a new app name:' input field and a 'Reserve' button. At the bottom, there are 'Previous', 'Next' (which is highlighted with a red box), and 'Cancel' buttons.

App Name	Package Identity in the Store
ActTodoList	None
CludemiaAMS	None

Setelah memilih aplikasi tersebut, tekan tombol Next.

Buka portal Windows Store Developer kemudian pilih apps yang sudah anda asosiasikan dengan project anda pada langkah selanjutnya. Pilih menu services untuk mengambil informasi Client Script dan Package SID.

## CloudemiaAMS: Release 1

App name Selling details Services Age rating Cryptography Packages Description Notes to testers	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">   <b>App name</b>                  You reserved an app name.                  You can reserve other names for your app to use in different languages or to change your app's name.  <a href="#">Learn more</a> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">   <b>Selling details</b>                  Pick your app's price, listing categories, and where you want to sell it.  <a href="#">Learn more</a> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">   <b>Services</b>                  Add push notifications, authenticate users, enable cloud storage, and define in-app offers.  <a href="#">Learn more</a> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">   <b>Age rating and rating certificates</b>                  Describe the audience for your app and upload your rating certificates.  <a href="#">Learn more</a> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">   <b>Cryptography</b>                  Declare whether your app uses cryptography and enable package upload.  <a href="#">Learn more</a> </div>
News  Free Phone developer account Add Windows 8.1 packages Increase in app roaming limits Age ratings Latest Windows ACK	

Kemudian pilih ‘Live Services Site’, kemudian catat informasi Client Script dan Package SID.

## CloudemiaAMS: Release 1

App name Selling details Services Age rating Cryptography Packages Description Notes to testers	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Services</b>                  Add services to bring connected, integrated experiences to your app and make it more engaging, dynamic, and appealing to your customers. You can also provide in-app offers to let customers make additional purchases from within your app.             </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Windows Azure Mobile Services</b>                  You can use Mobile Services to send push notifications, authenticate and manage app users, and store app data in the cloud. <a href="#">Learn more</a> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>In-app offers</b>                  You can use in-app offers to sell additional features and products for this app through the Windows Store. <a href="#">Learn more</a> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">                 Enter a unique product ID for each offer. The product ID is the internal reference to the offer that you use in the app's program code. Your customers won't see the product ID, but they will see the offer's description that you enter on the Description page later.             </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">                 You can't change or delete product IDs after you submit the app for certification.             </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">                 Product ID <input type="text"/>                  Price tier <input type="button" value="Pick a price tier"/>      Product lifetime <input type="button" value="Forever"/>      Content type <input type="button" value="Inherit from app"/> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <a href="#">Add another offer</a> </div>
News  Free Phone developer account Add Windows 8.1 packages Increase in app roaming limits Age ratings Latest Windows ACK	

## CloudemiaAMS

Settings  
Basic Information  
API Settings  
**App Settings**  
Localization

To protect your app's security, Windows Push Notification Services (WNS) and Live Connect services use client secrets to authenticate the communications from your server.

Package SID: [REDACTED] This is the unique identifier for your Windows Store app.

Link to different app

Application identity:  
<Identity  
Name="1 [REDACTED]"  
Publisher=" [REDACTED]" />

Client ID: [REDACTED] To set your application's identity values manually, open the AppManifest.xml file in a text editor and set these attributes of the <identity> element using the values shown here.

Client secret: [REDACTED] This is a unique identifier for your application.

If your client secret has been compromised or your organization requires that you periodically change client secrets, create a new client secret here. After you create a new client secret, both the old and the new client secrets will be accepted until you activate the new secret.

[Create a new client secret](#)

Kemudian masuk kedalam portal Azure Mobile Services, pilih menu PUSH, dan pada bagian bawah tekan tombol 'Enable Enhanced PUSH'.



Masukan informasi Client Script dan Package SID yang sudah anda dapatkan.

cloudemiaams

DASHBOARD DATA API SCHEDULER PUSH IDENTITY CONFIGURE SCALE **PREVIEW** LOGS

windows application credentials

CLIENT SECRET [REDACTED]

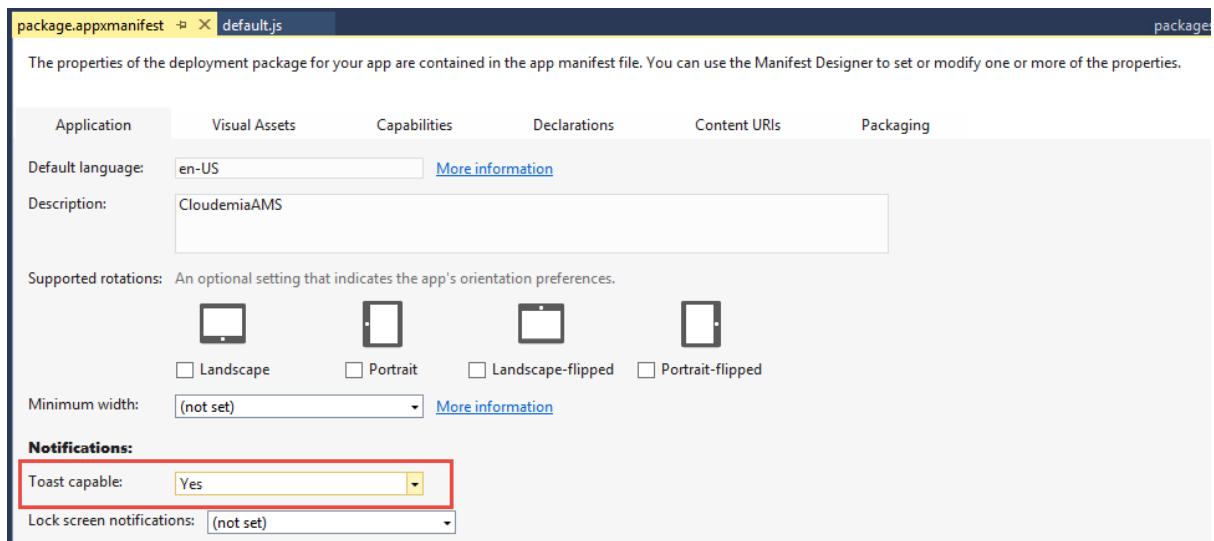
PACKAGE SID [REDACTED]

Kemudian pada default.js tambahkan kode dibawah instansiasi WindowsAzure.MobileServiceClient.

```
// This MobileServiceClient has been configured to communicate
with your Mobile Service's url
// and application key. You're all set to start working with
your Mobile Service!
var client = new WindowsAzure.MobileServiceClient(
    "https://cloudemiaams.azure-mobile.net/",
    "...."
);
```

```
//push notification
Windows.Networking.PushNotifications
    .PushNotificationChannelManager
.createPushNotificationChannelForApplicationAsync().then(
    function(channel) {
        client.push.registerNative(channel.uri);
});
```

Pada solution explorer pilih package.appmanifest kemudian ubah pengaturan Toast capable menjadi 'Yes'.



Kembali ke halaman azure mobile services, pilih menu DATA – Todoitem – SCRIPT. Pada saat perintah insert tambahkan kode javascript berikut untuk mengirimkan notifikasi push ke client.

```
function insert(item, user, request) {

    //console.log('Menambahkan item baru: ' + item.text);

    var payload = '<?xml version="1.0" encoding="utf-8"?><toast><visual>' +
        '<binding template="ToastText01"> <text id="1">' +
        item.text + '</text></binding></visual></toast>';

    request.execute({
        success: function() {
            // If the insert succeeds, send a notification.
            push.wns.send(null,payload, 'wns/toast', {
                success: function(pushResponse) {
                    console.log("Sent push:", pushResponse);
                    request.respond();
                },
            });
        }
    });
}
```

```
error: function (pushResponse) {  
    console.log("Error Sending push:", pushResponse);  
    request.respond(500, { error: pushResponse });  
}  
});  
}  
});  
}
```

Setelah itu save dan jalankan aplikasi Windows Store App anda. Ketika anda jalankan dan menambahkan todolist baru maka push notification akan muncul dalam bentuk toast seperti pada gambar berikut.

