

IMPLEMENTASI AZURE TABLE STORAGE UNTUK PENGELOLAAN KUIS BERBASIS CLOUD STUDI KASUS: UJIAN NASIONAL TINGKAT SLTA

Obed Kharistian Marsudi
Djoni Dwijono

Abstrak

Ujian Nasional (UN) SMA adalah tahap penentuan akhir untuk siswa SMA untuk dinyatakan lulus, maka UN ini perlu dipersiapkan dengan serius dan dilatih berulang kali agar kelulusan siswa di jenjang SMA dapat tercapai. Di era mobile saat ini, maka dibutuhkan hadirnya suatu aplikasi yang dapat menyediakan dengan cepat dan mudah diakses, untuk digunakan sebagai latihan ujian nasional.

Dalam penelitian ini, Azure Table Storage akan digunakan untuk membuat media penyimpanan untuk latihan kuis ujian nasional. Azure Table Storage adalah salah satu basisdata non relasional yang mampu menyimpan data non relasional untuk manajemen kuis. Penyimpanan data yang dibangun dengan Azure Table Storage, dapat diakses melalui layanan web yang berjalan di atas platform Azure.

Produk dari penelitian ini adalah aplikasi dekstop untuk aplikasi manajemen pelatihan dan kuis UN berbasis web dengan fitur layar ponsel. Aplikasi tersebut dapat mengambil data dari layanan web yang dibangun oleh Azure Table Storage.

Kata Kunci : Azure Table Storage, Basisdata Non Relasional, Kuis

1. Pendahuluan

Ujian Nasional merupakan ujian yang menjadi salah satu penentu kelulusan siswa SMA. Siswa SMA sering menilai Ujian Nasional (UN) sebagai ujian yang tidak mudah dan harus dipersiapkan dengan baik sehingga mereka membutuhkan banyak latihan soal UN. Untuk menjawab kebutuhan latihan soal UN bagi Siswa SMA, dibutuhkan aplikasi yang dapat menyediakan soal latihan secara cepat dan mudah diakses dimana saja. Aplikasi harus dapat diakses dengan menggunakan perangkat komputer, *tablet*, maupun *mobile phone*.

Melalui penelitian ini akan dihasilkan implementasi Azure Table Storage dalam pembuatan aplikasi kuis berbasis *cloud* yang mampu menyimpan manajemen pelatihan dan kuis soal-soal latihan UN serta dapat diakses dengan mudah oleh siswa SMA melalui *Mobile Phone* mereka.

2. Dasar Teori

2.1. Tinjauan Pustaka

Pada jurnal *Cloud Computing with Azure Paas for Educational Institutions* (Mirza Zainab, 2014), Mirza menuliskan penggunaan Azure Table Storage dalam mengatasi *server crash* yang terjadi pada sistem ujian *online* saat banyak mahasiswa yang menggunakannya. Mirza menyebutkan tiga alasan penggunaan Azure untuk institusi pendidikan. Pertama skalabilitas Azure mampu menangani perubahan jumlah *data request* pada masa ujian dan pembagian nilai studi. Kedua, Azure menggunakan *web role* untuk menyelesaikan pemrosesan data secara simultan. Ketiga, *web role* milik Azure dapat berjalan pada saat pemrosesan *background*.

2.2. Non Relational Database

RDBMS (*Relational Database Management System*) yang merupakan basisdata relasional yang menggunakan prinsip ACID dalam penerapan transaksinya. ACID merupakan kependekan dari: (Tiwari, 2011)

1. *Atomicity*: Setiap transaksi yang dilakukan berhasil dan jika tidak maka dikembalikan.
2. *Consistency*: Transaksi tidak dapat meninggalkan *database* dalam kondisi yang tidak konsisten.

3. *Isolation*: Sebuah transaksi tidak dapat mengganggu transaksi yang lain.
4. *Durability*: Sebuah transaksi yang sudah terjadi akan tetap meskipun aplikasi dijalankan kembali.

Disamping pemahaman basisdata relasional, ada juga basisdata yang disebut *Non Relational Database*. *Non Relational Database* ini dikenal dengan nama NoSQL.

“NoSQL secara harafiah merupakan kombinasi dari dua kata: No dan SQL. Implikasinya adalah NoSQL merupakan teknologi atau produk yang bertentangan dengan SQL. Pembuat dan pemakai mula-mula dari kata “NoSQL” mungkin ingin mengatakan *No RDBMS* atau *Non Relational* namun begitu terpesona dengan istilah NoSQL yang lebih enak didengar sehingga terjebak menggunakan istilah tersebut.” (Tiwari, 2011: 4)

NoSQL berasal dari kata *not only SQL*. Pada penerapannya NoSQL juga mengijinkan adanya bahasa yang menyerupai SQL. Istilah NoSQL pertama kali digunakan pada tahun 1998 untuk *Database* relasional yang menghilangkan penggunaan SQL. Kemudian pada tahun 2009 istilah ini muncul kembali dalam pertemuan di San Fransisco yang diorganisir oleh Jon Oskarsson. Berbeda dengan RDBMS yang menerapkan ACID, NoSQL menerapkan konsep BASE sebagai berikut: (Tiwari, 2011:10)

1. *Basic availability*: *Each request is guaranteed a response—successful or failed execution.*
2. *Soft state*: *The state of the system may change over time, at times without any input (for eventual consistency).*
3. *Eventual consistency*: *The Database may be momentarily inconsistent but will be consistent eventually.*

Berikut ini akan disajikan perbandingan antara konsep SQL dengan NoSQL, atau antara ACID dengan BASE seperti pada tabel berikut ini:

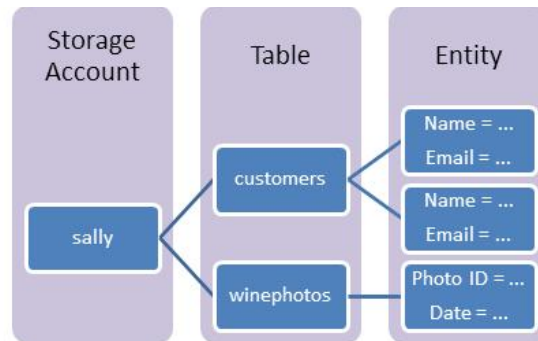
Tabel 1.
Perbandingan Konsep SQL (ACID) dan NoSQL (BASE)

SQL (ACID)	NoSQL (BASE)
Konsistensi kuat	Konsistensi lemah
Susah Berevolusi	Mudah Berevolusi
Pesimis	Optimis
Cepat	Lebih Cepat
Perintah kompleks	Lebih Sederhana

2.3. Azure Table Storage

Windows Azure merupakan komputasi awan yang diciptakan oleh Microsoft. Windows Azure menyediakan tiga macam layanan yaitu *platform as a service (paas)*, *software as a service (saas)*, dan *infrastructure as a service (iaas)*. *Azure Table Storage* merupakan salah satu product dari Windows Azure yang menyediakan media penyimpanan data berbasis NoSQL. *Azure Table Storage* menyediakan batas ukuran *database* hingga 100 Tera Byte tiap akun. (Calder, 2012)

Konsep: *Azure Table Storage* merupakan penyimpanan yang bersifat *Non Relational Database* yang tergolong dalam tipe Key-Value Store. Layanan tabel ini mengandung komponen sebagai berikut:



Gambar 1. Komponen Table Storage

(<https://acomdpsstorage.blob.core.windows.net/dpsmedia-prod/azure.microsoft.com/en-us/documentation/articles/storage-nodejs-how-to-use-table-storage/20140918053042/includes/howto-table-storage/table1.png>)

3. Implementasi Sistem

Implementasi sistem dilakukan dengan *database* Azure Table Storage, pemrograman ASP.Net MVC dan C#, serta JQuery Mobile sebagai *framework* untuk tampilan *mobile*.

3.1. Windows Azure Account dan Koneksi

Implementasi Azure diawali dengan empat tahap yaitu pembuatan akun Azure, pemasangan Azure SDK, pembuatan koneksi menuju Azure dengan *Account Name* dan *Account Key*, serta pembuatan kelas-kelas *entity*.

3.2. Perancangan Table Storage

Perancangan *database* Azure dilakukan melalui tiga tahapan yaitu menentukan *query* yang akan dipanggil, menentukan transaksi yang akan terjadi pada sistem, dan menentukan *partitionkey* pada setiap tabel. Ada pun hasil perancangannya sebagai berikut:

Tabel 2.
Player Entity

Player Entity		
<i>Partition Key</i>	: <i>PlayerID</i>	String
<i>Row Key</i>	: “ <i>Players</i> ”	String
<i>Properties</i>	: - NamaLengkap	String
	-NamaPanggilan	String
	-Sekolah	String
	-Kelas	String
	-TanggalLahir	DateTime
	-Level	Int
	-Pengalaman	Int

Tabel 3.
Player Account Entity

Player Account Entity		
<i>Partition Key</i>	: <i>PlayerID</i>	String
<i>Row Key</i>	: “ <i>Account</i> ”	String
<i>Properties</i>	: - Email	String
	-Password	String
	-DateCreated	DateTime
	-DateUpdated	DateTime

Tabel 4.
 Player Log Entity

Player Log Entity		
<i>Partition Key</i>	: <i>PlayerID</i>	String
<i>Row Key</i>	: "Log"	String
<i>Properties</i>	: - VisitDate	DateTime
	-LeaveDate	DateTime

Tabel 5.
 PermainanEntity

Permainan Entity		
<i>Partition Key</i>	: <i>PlayerID</i>	String
<i>Row Key</i>	: KuisID	String
<i>Properties</i>	: - NamaKuis	String
	-MataPelajaran	String
	-Tahun	String
	-Score	Double
	-JawabanBenar	Int
	-JumlahSoal	Int

Tabel 6.
 Kuis Entity

Kuis Entity		
<i>Partition Key</i>	: <i>KuisID</i>	String
<i>Row Key</i>	: "KeteranganKuis"	String
<i>Properties</i>	: - NamaKuis	String
	-MataPelajaran	String
	-Tahun	String
	-JumlahSoal	Int
	-Kode	String

Tabel 7.
 Detail KuisEntity

DetailKuis Entity		
<i>Partition Key</i>	: KuisID	String
<i>Row Key</i>	: SoalID	String
<i>Properties</i>	: -Soal	String
	-Gambar	String
	-PilihanA	String
	-PilihanB	String
	-PilihanC	String
	-PilihanD	String
	-PilihanE	String
	-KunciJawaban	String

3.3. Implementasi Aplikasi Desktop

Aplikasi dekstop yang dihasilkan bertujuan untuk menjadi aplikasi bagi admin untuk mengelola data soal dan data user.

ID	Nama Kuis	Mata Pelajaran	Tahun	Jumlah Soal	Kode	Tanggal	Keterangan
IPA	UN Biologi 2012	Biologi	2012	10	1	8/17/2014 4:33	A_Biologi_2012
IPA	UN Biologi 2012	Biologi	2012	10	2	8/17/2014 4:33	A_Biologi_2012_2
IPA	UN Biologi 2012	Biologi	2012	10	3	8/17/2014 4:33	A_Biologi_2012_3
IPA	UN Fisika 2012	Fisika	2012	10	1	8/17/2014 4:33	A_Fisika_2012_1
IPA	UN Fisika 2012	Fisika	2012	10	2	8/17/2014 4:33	A_Fisika_2012_2
IPA	UN Fisika 2012	Fisika	2012	10	3	8/17/2014 4:33	A_Fisika_2012_3
IPA	UN Kimia 2012	Kimia	2012	10	1	8/20/2014 4:20	A_Kimia_2012_1
IPA	UN Kimia 2012	Kimia	2012	10	2	8/20/2014 4:20	A_Kimia_2012_2
IPA	UN Kimia 2012	Kimia	2012	10	3	8/20/2014 4:20	A_Kimia_2012_3

Gambar 2. Form Soal

Detail Soal

Kuis: UN Fisika 2012, Tahun: 2012, Jumlah Soal: 10, Kode: 7

ID: Kurus Jemberan C, Gambar

Soal: Sebuah benda bergerak dari titik A tanpa kecepatan awal. Jika selama gerakan tidak...

Jawaban A: 8 m/s
 Jawaban B: 12 m/s
 Jawaban C: 20 m/s
 Jawaban D: 24 m/s
 Jawaban E: 30 m/s

Tombol: Tambah, Hapus, Ubah, Detail, Skip

Gambar 3. Form Detil Soal

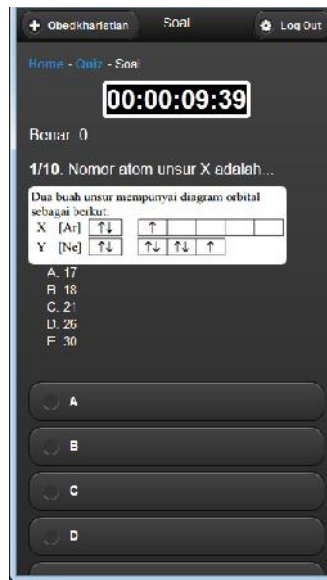
Form Detil Soal ini berisi paket soal Ujian Nasional. Form ini berfungsi untuk menambah, mengubah, maupun menghapus paket soal. Form ini dilengkapi dengan tombol untuk membuka form detail soal.

```
private void IsiData()
{
    IEnumerable<KuisEntity> entitie =
    TableHelper.QueryEntities<KuisEntity>("TableSoal").Where(e =>
    e.RowKey == "KeteranganKuis").AsTableServiceQuery<KuisEntity>();
    bs.DataSource = entitie;
    dgvKuis.DataSource = bs;
    BindingData();
}
```

Gambar 4. Kode Program Isi Data Soal

3.4. Implementasi Aplikasi Website Kuis

Halaman ini merupakan halaman utama dari transaksi di dalam sistem. Gambar soal yang ditampilkan diambil dari penyimpanan blob. Kemudian setiap halaman berisi satu soal saja. Setiap kali user menjawab, sistem akan melakukan pengecekan terhadap kunci jawaban, dan apabila jawaban benar maka poin Benar akan bertambah satu. Dengan demikian user dapat melihat secara langsung apakah jawaban yang mereka berikan benar atau salah. Kemudian ketika pengguna merasa belum dapat menjawab, pengguna dapat menekan tombol skip untuk melempar soal tersebut ke urutan belakang.



Gambar 5. Halaman Soal Kuis

```
ViewBag.Pkey = PartitionKey;  
ViewBag.Rkey = RowKey;  
ViewBag.Skor = 0 + Skor; //Simpan Nilai Skor  
ViewBag.JumlahSoal = Session["Jumlah"];  
string storageConnectionString =  
(string)HttpContext.Application["conn"];  
TableHelper TableHelper =  
new TableHelper(storageConnectionString);  
DetailKuisEntity detailkuis = new DetailKuisEntity();  
try  
{  
    detailkuis =  
    TableHelper.QueryEntities<DetailKuisEntity>("TableSoal").Where  
(e => e.PartitionKey == PartitionKey && e.RowKey ==  
    RowKey).SingleOrDefault();  
    return View(detailkuis);  
}  
catch (Exception)  
{  
    //Kirim ke Hasil untuk merekap hasil Quiz  
    return RedirectToAction("Hasil", "Quiz", new { PartitionKey =  
    PartitionKey, RowKey = RowKey, Skor = Skor });  
}
```

Gambar 6. Kode Soal Kuis

4. Analisis Sistem

4.1 Analisis Implementasi Azure Table Storage sebagai Non Relational Database

Implementasi Azure Table Storage memenuhi konsep BASE yang merupakan sifat dari *non relational database* yaitu konsistensi lemah, mudah berevolusi, optimis, lebih cepat, dan lebih sederhana.

1. Konsistensi Lemah

Lemahnya konsistensi Azure Table Storage disebabkan oleh konsep *non relational* dari Azure Table Storage. Azure Table Storage mengizinkan adanya *null value* pada sebuah tabel karena setiap tabel mengandung entitas yang berbeda-beda. Pada pengelolaan kuis

ini juga ditemukan nilai kosong pada TableEntity maupun TableSoal. Pada TableEntity ada tiga buah entitas yaitu entitas Log, entitas Account, dan entitas Player. Pada entitas Player, seorang pengguna dengan partition key A memiliki nilai pada kolom nama lengkap, nama panggilan, tanggal lahir, dan sebagainya. Namun, entitas ini akan memiliki nilai kosong pada kolom password, email, tanggal datang, tanggal pergi.

2. Mudah Berevolusi

Konsep yang membedakan Azure Table Storage dari SQL adalah kemudahan berevolusi. Azure Table Storage tidak memiliki aturan ketat akan *durability* dimana data akan tetap saat ia disimpan. Azure Table Storage mengizinkan perubahan jumlah property pada tiap entitas. Pada pembuatan aplikasi pengelolaan kuis ada penambahan field Kode pada KuisEntity untuk menampung kode paket soal yang berguna dalam melakukan pengacakan soal kuis.

3. Optimis

Azure Table Storage optimis akan kepastian data yang diinput ke dalamnya. Tidak akan ada data yang dianggap tidak konsisten dan menyebabkan sistem melakukan *roll back*. Data yang diinput akan selalu mendapatkan respon baik itu berhasil meskipun tidak konsisten ataupun gagal.

4. Lebih Cepat

Hal yang menyebabkan Azure Table Storage memenuhi sifat NoSQL yaitu lebih cepat adalah adanya *partition key*. *Partition Key* pada setiap entitas menjadi kunci untuk mengelompokkan data. Saat data dipindah ke dalam server lain, data akan tetap dalam kelompok sesuai dengan *PartitionKey* yang dimiliki setiap entitas.

5. Lebih Sederhana

Azure Table Storage mengizinkan terjadinya duplikasi sehingga sebuah field yang sama dapat dijumpai di beberapa tempat seperti NamaKuis, Tahun, dan JumlahSoal yang terdapat pada PermainanEntity dan KuisEntity. Dengan adanya duplikasi, pemanggilan data menjadi lebih sederhana. Data dipanggil berdasarkan tabel dan kuncinya tanpa melakukan penggabungan tabel atau join. Proses yang lebih sederhana ini juga meningkatkan kecepatan penemuan kembali data dari Table Storage.

4.2. Analisis Pembuatan Web Service dengan Azure Table Storage

Berikut adalah penggunaan metode REST yang digunakan dalam pengelolaan kuis:

1. Create Table

Pada saat pembuatan tabel kode program `TableHelper.CreateTable("TableSoal");` dipanggil. Kode ini akan memanggil metode yang sudah tersedia pada `tableHelper` `TableClient.CreateTable (tableName);` Metode ini merupakan perintah POST menuju `https://eduquiz.table.core.windows.net/Tables`.

2. Insert Entity

Pada saat penambahan entity kode program akan memanggil metode yang sudah tersedia pada `tableHelper`. Metode ini merupakan perintah POST menuju `https://eduquiz.table.core.windows.net/TableSoal`.

3. Get Entity

Pada saat pengambilan entity kode program akan memanggil metode yang sudah tersedia pada `tableHelper`. Metode ini merupakan perintah GET menuju `https://eduquiz.table.core.windows.net/TablePlayer(PartitionKey='<partition-key>',RowKey='<row-key>')?$select=<comma-separated-property-names>`

4. Replace Update Entity

Pada saat perubahan entity kode program 4.20 dipanggil. Kode ini akan memanggil metode yang sudah tersedia pada tableHelper. Metode ini merupakan perintah PUT menuju [https://eduquiz.table.core.windows.net/TablePlayer\(PartitionKey='myPartitionKey',RowKey='myRowKey'\)](https://eduquiz.table.core.windows.net/TablePlayer(PartitionKey='myPartitionKey',RowKey='myRowKey'))

5. Delete Entity

Pada saat perubahan entity kode program 4.20 dipanggil. Kode ini akan memanggil metode yang sudah tersedia pada tableHelper. Metode ini merupakan perintah DELETE menuju [https://eduquiz.table.core.windows.net/TablePlayer\(PartitionKey='myPartitionKey',RowKey='myRowKey'\)](https://eduquiz.table.core.windows.net/TablePlayer(PartitionKey='myPartitionKey',RowKey='myRowKey'))

4.3. Kelebihan Sistem

- Sistem dapat menyimpan rekam jejak pengerjaan soal dari pemain.
- Sistem dapat menunjukkan peringkat permainan kepada pemain yang diharapkan dapat meningkatkan daya saing dalam latihan soal.
- Sistem dapat menyimpan data gambar dan notasi rumit pada *blob storage*.
- Sistem dapat diakses dalam bentuk REST API sehingga sistem mampu diakses baik dari *desktop* admin maupun dari *website client* atau bahkan ke depannya untuk aplikasi *mobile* di *platform* Android maupun IOS.
- Sistem menggunakan *JqueryMobile* sehingga dapat diakses dari *mobile phone* sekalipun.
- Sistem memiliki *server* penyimpanan data yang handal dengan Azure Table Storage.

4.4. Kekurangan Sistem

- Sistem hanya dapat menyajikan data kompleks seperti grafik, ekspresi matematika dalam bentuk gambar sehingga desain kuis tidak begitu baik saat banyak data kompleks.
- Sistem belum dapat menambahkan soal *listening* untuk mata pelajaran bahasa inggris.

5. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, penulis mengambil kesimpulan sebagai berikut:

- Implementasi *Azure Table Storage* sebagai *Non Relational Database* mampu mengelola soal-soal UN dengan membuat entitas-entitas yang sesuai untuk aplikasi kuis.
- Sistem dapat menemukan kembali data kuis dari *RESTful Web Service* dengan cara membuat kelas yang memanggil Table Service REST Api yang disediakan oleh *Web Service* dari Azure Table Storage.
- Penggunaan Azure Table Storage sebagai basisdata untuk aplikasi kuis memberikan kemudahan akses, kecepatan akses, dan keamanan bagi data kuis namun perancangan entitas tidak mudah dan proses impor ekspor data dinilai cukup rumit.

Daftar Pustaka

- Calder, B. (2012). Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency. Microsoft, 143-157.
- Gamma, T. W. (2014). Paket Sukses Menghadapi Ujian Nasional dan Ujian Sekolah SMA/MA IPA 2013/2014. Bandung: Yrama Widya.
- Microsoft. (2011, Januari 7). Windows Azure Storage Helper. Retrieved from Nuget: <https://www.nuget.org/packages/WindowsAzureStorage.Helper/>

- Microsoft. (2014, Maret 4). How to use the Table Storage Service. Retrieved from Windows Azure: <http://www.windowsazure.com/en-us/documentation/articles/storage-dotnet-how-to-use-table-storage-20/>
- Mirza Zainab, V. D. (2014). Cloud Computing with Azure PaaS for Educational Institutions. *International Journal of Information and Computation Technology*, 139-144.
- Neudesic. (2014, Maret 5). Get Started with ASP.NET Web Sites. Retrieved from ASP.NET: <http://www.asp.net/get-started/websites>
- Tiwari, S. (2011). *Professional NoSql*. Indianapolis: John Wiley & Sons, Inc.