

# IMPLEMENTASI AZURE MEDIA SERVICES UNTUK VIDEO STREAMING

Yovena Chintya Pujiantoro  
Katon Wijana

## Abstrak

*Video merupakan salah satu bentuk media informasi yang efektif untuk menyebarkan dan menyampaikan informasi. Video dapat merekam gambaran nyata dari suatu peristiwa atau sumber pengetahuan, karena dalam sebuah video, informasi yang bisa disampaikan melibatkan media audio dan visual yang mudah untuk dipahami oleh penerimanya.*

*Web Streaming merupakan salah satu media yang efektif untuk dapat menyampaikan informasi berupa video sehingga informasi dapat disebar luaskan dengan mudah. Azure Storage dan Azure Media Service merupakan teknologi yang memungkinkan suatu video untuk dapat disimpan dan dikelola sehingga dapat kembali kepada penerima informasi dalam bentuk stream.*

*Pada penelitian ini penulis mengimplementasikan Azure Storage dan Azure Media Services pada sebuah sistem informasi web streaming. Penelitian dilakukan dengan menganalisis penerapan Azure Storage dalam penyimpanan data pada sistem informasi dan Azure Media Service sebagai servis untuk mengambil kembali konten video tersebut.*

**Kata Kunci :** *Azure Storage, Azure Media Services, Video Streaming*

## 1. Pendahuluan

Data multimedia, dalam hal ini video memerlukan penanganan khusus untuk penyimpanan dan pengambilan datanya karena ukurannya yang sangat besar.

Pada penelitian ini akan dibangun sebuah website video streaming yang akan menampung data berupa video dan akan menampilkan konten video tersebut berdasarkan kategori yang ada. Dengan adanya website ini diharapkan dapat memudahkan pengguna untuk mengakses informasi berupa video.

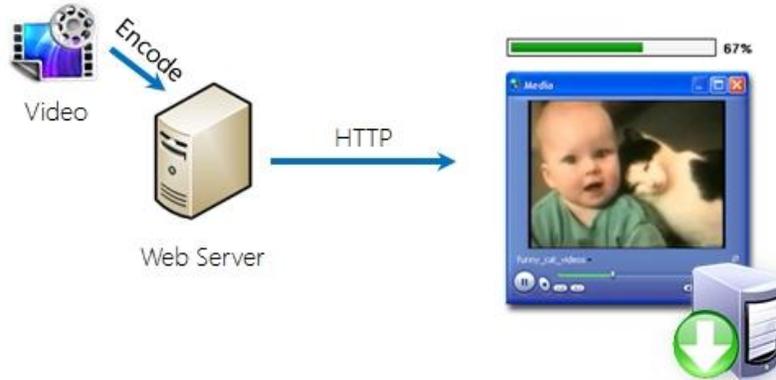
## 2. Landasan Teori

### a. Video Streaming

*Video streaming* adalah istilah yang digunakan ketika pengguna dapat melihat video di internet melalui browser secara langsung tanpa harus mengunduh file video lebih dahulu secara utuh. Ada 3 jenis teknologi streaming yaitu:

### *Progressive Download*

*Progressive Download* adalah teknologi streaming pertama. Pada awalnya cara kerja *progressive download* adalah menaruh file video di *web server* dan user mengunduhnya ke komputer. User hanya dapat menonton setelah unduhan selesai.



*Gambar 1. Cara Kerja Awal Progressive Download*

*Progressive Download* adalah teknologi streaming pertama. Pada awalnya cara kerja *progressive download* adalah menaruh file video di *web server* dan user mengunduhnya ke komputer. User hanya dapat menonton setelah unduhan selesai.



*Gambar 2. Cara Kerja Progressive Download Sekarang*

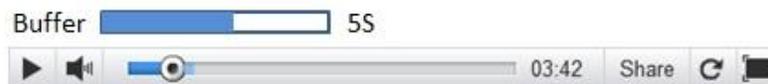
Namun ada beberapa kekurangan di progressive download yaitu:

1. Bandwith terbuang sia-sia
2. Kualitas Video tidak dapat menyesuaikan bandwith

### *Traditional Streaming*

*Traditional Streaming* menggunakan media server dan protokol khusus untuk menyajikan konten. Ada 3 protokol yang digunakan di traditional streaming yaitu RTSP (*Real Time Streaming Protocol*), RTMP (*Real Time Messaging Protocol*) dan RDT (*Real Data Protocol*).

Beda *traditional streaming* dengan *progressive download* adalah cara mengatur ukuran buffer ada di sisi pengguna. Misalnya server tidak akan memuat lebih dari 5 detik dari file yang sudah ditonton meskipun sudah dibiarkan beberapa saat.



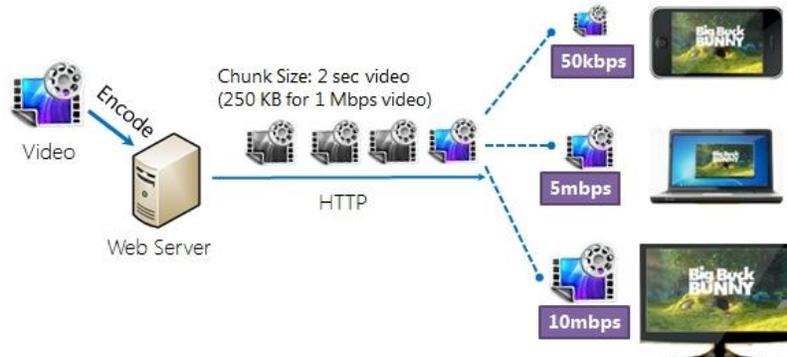
*Gambar 3. Cara Kerja Buffer pada Traditional Streaming*

Beberapa kekurangan traditional streaming:

1. Harus membeli media server khusus yang mahal.
2. Kualitas video tidak bisa menyesuaikan bandwith

### *Http-based Adaptive Streaming*

*Http-based adaptive streaming* menggunakan web server biasa dan menyalurkan data melalui protokol HTTP. File video dipotong menjadi bagian yang lebih kecil, biasanya 2 detik. Untuk 1Mbps video akan dipotong menjadi 250 KB.



Gambar 4. Cara Kerja Http-based Adaptive Streaming

Pada Gambar 4. dapat dilihat bahwa di Http-based adaptive streaming video di-encoded menjadi 3 tipe yaitu: 5mbps, 10mbps dan 15mbps. Pengkategorian ini berdasarkan dari kondisi jaringan dan kemampuan CPU. Misalnya pada 2 detik pertama server akan mencoba memuat video dengan kecepatan 5mbps dan jika koneksinya memungkinkan untuk memuat dengan kecepatan 10mbps maka server akan dengan sendirinya menyesuaikan. Begitu juga sebaliknya, jika koneksi buruk maka server akan memuat video dengan bit-rate yang paling rendah.

Kelebihan dari *Http-based adaptive streaming* adalah pengguna dapat men-streaming video dengan sangat lancar. Selain itu *Http-based adaptive streaming* menggunakan protokol HTTP yang sudah dipakai dalam jaringan internet.

#### **b. Player Framework HTML 5**

HTML5 adalah bahasa markup yang distandarkan oleh konsorsium W3C (*World Wide Web Consortium*) dan WHATWG (*Web Hypertext Application Technology Working Group*). HTML5 adalah pengembangan dari versi HTML sebelumnya. HTML5 memiliki fitur yang lebih lengkap untuk membuat website yang interaktif dan modern. Saat ini HTML5 masih dalam proses pengembangan secara terus-menerus. Sebagian besar browser versi terbaru sudah mendukung sebagian fitur dari HTML5 diantaranya Firefox, Chrome, IE9, Safari, dan Opera.

Player Framework HTML5 digunakan dalam aplikasi berbasis browser baik browser pada komputer dan browser pada mobile yang mendukung tag video HTML5. W3C saat ini tidak memiliki standar untuk memutar video secara adaptive streaming, sehingga sebagian besar browser hanya mendukung pemutaran video secara progressive download. Player Framework HTML5 yang dikembangkan oleh Microsoft, sudah mampu memutar video secara adaptive streaming sehingga pada kasus ini digunakan sebagai video player untuk memutar video.

#### **c. Azure Blob Storage**

*Azure Blob Storage* adalah servis untuk menyimpan data yang tidak terstruktur dalam jumlah yang besar yang dapat diakses dari seluruh dunia melalui HTTP atau HTTPS. Satu blob dapat menyimpan ratusan Gigabyte sedangkan satu storage dapat menyimpan sampai dengan 200 Terabyte.

**d. Azure Media Services**

(Yan, 2012:http://mingfeiy.com/what-windows-azure-media-services/).

“Windows Azure Media Services offers a collection of components and technologies from Microsoft and third parties to enable end-to-end media solutions, leveraging on Windows Azure platform. If you are looking for end-to-end media solution with the power of cloud computing, Windows Azure Media Services is the right service you should consider.”

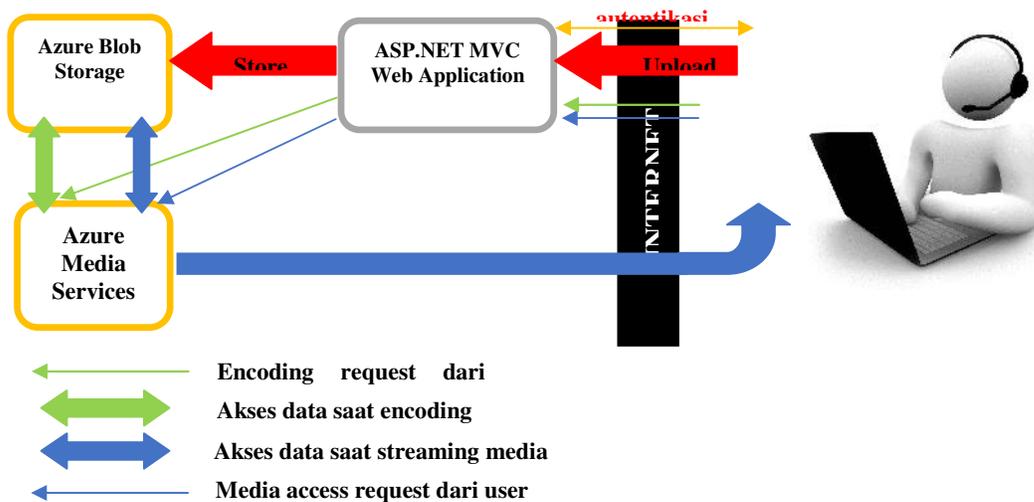
Berdasarkan kutipan diatas, dapat dipahami bahwa Windows Azure Media Services adalah sebuah services dari Microsoft yang menawarkan berbagai komponen dan teknologi untuk streaming berbagai media berbasis cloud computing.

**e. ASP.NET MVC 4**

ASP.NET MVC adalah implementasi arsitektur MVC pada lingkungan pemrograman web ASP.NET. ASP.NET MVC membantu mengurangi kompleksitas sebuah aplikasi web dengan membagi aplikasi ke dalam tiga layer, yaitu Model, View dan Controller. Pemisahan ini membuat performa website menjadi lebih baik dan juga mudah dikelola.

**3. Perancangan Sistem**

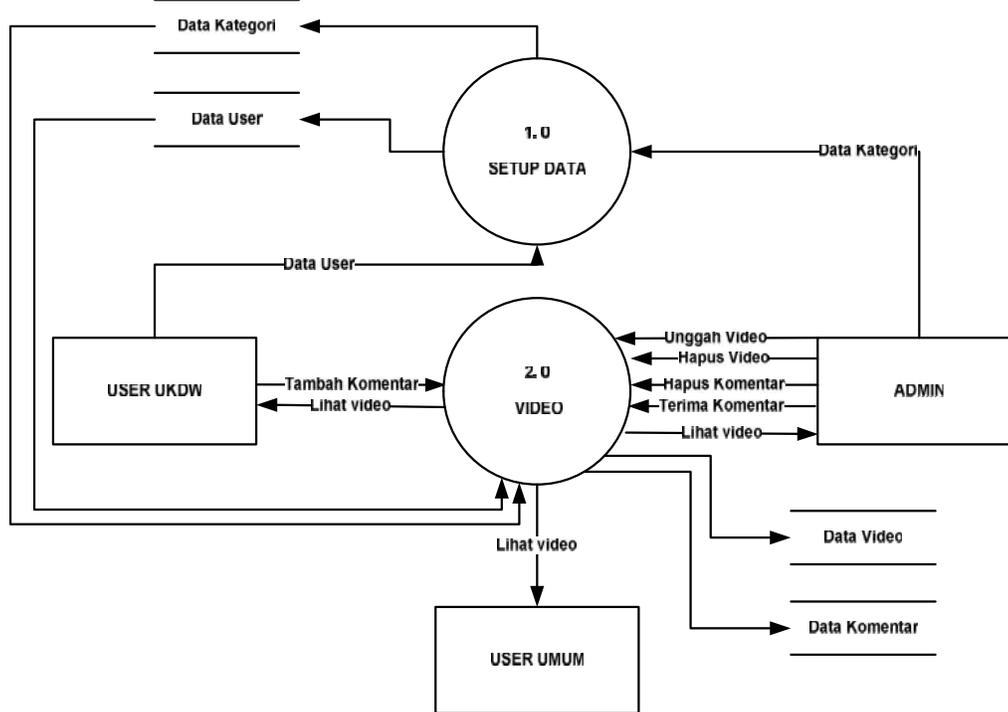
**a. Arsitektur Aplikasi**



Gambar 5. Rancangan Arsitektur Aplikasi

Gambar 5. menunjukkan rangkaian kerja sistem secara menyeluruh dan aliran data yang terjadi saat menggunakan Azure Media Services dan Azure Blob. Pertama, pengguna akan diminta login ke dalam sistem yang berupa ASP.NET MVC Web Application. Lalu file video yang diunggah pengguna akan disimpan ke dalam Azure Blob Storage. Setelah diunggah, file akan diencoding supaya saat streaming dilakukan tidak ada kendala. Ketika user mengirimkan permintaan video, maka data video itu ditampilkan secara streaming kepada user.

**b. Rancangan Sistem**

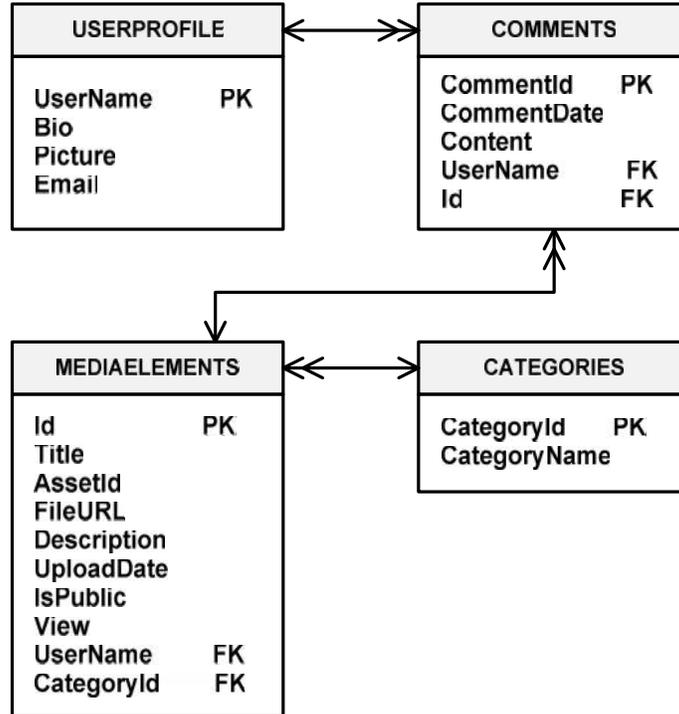


Gambar 6. DFD Level 1

Rancangan proses dari sistem yang akan dibangun dapat dilihat dari *Data Flow Diagram* (DFD). DFD menggambarkan aliran data, penyimpanan data dan dokumentasi dari sistem yang ada. Aliran tersebut termasuk input, proses dan output pada sistem.

Dengan adanya gambaran tersebut maka dapat diketahui dari mana data berasal dan proses yang terjadi di dalamnya. Penggambaran DFD dimulai dari diagram konteks yang merupakan gambaran umum dari sistem. Dari konteks diagram tersebut dapat diuraikan menjadi DFD level 0, 1, dan seterusnya hingga tingkatan paling detail yang mewakili alur sistem. Gambar 6 merupakan gambar DFD level 1 sistem yang akan dibangun.

c. Rancangan Basis Data

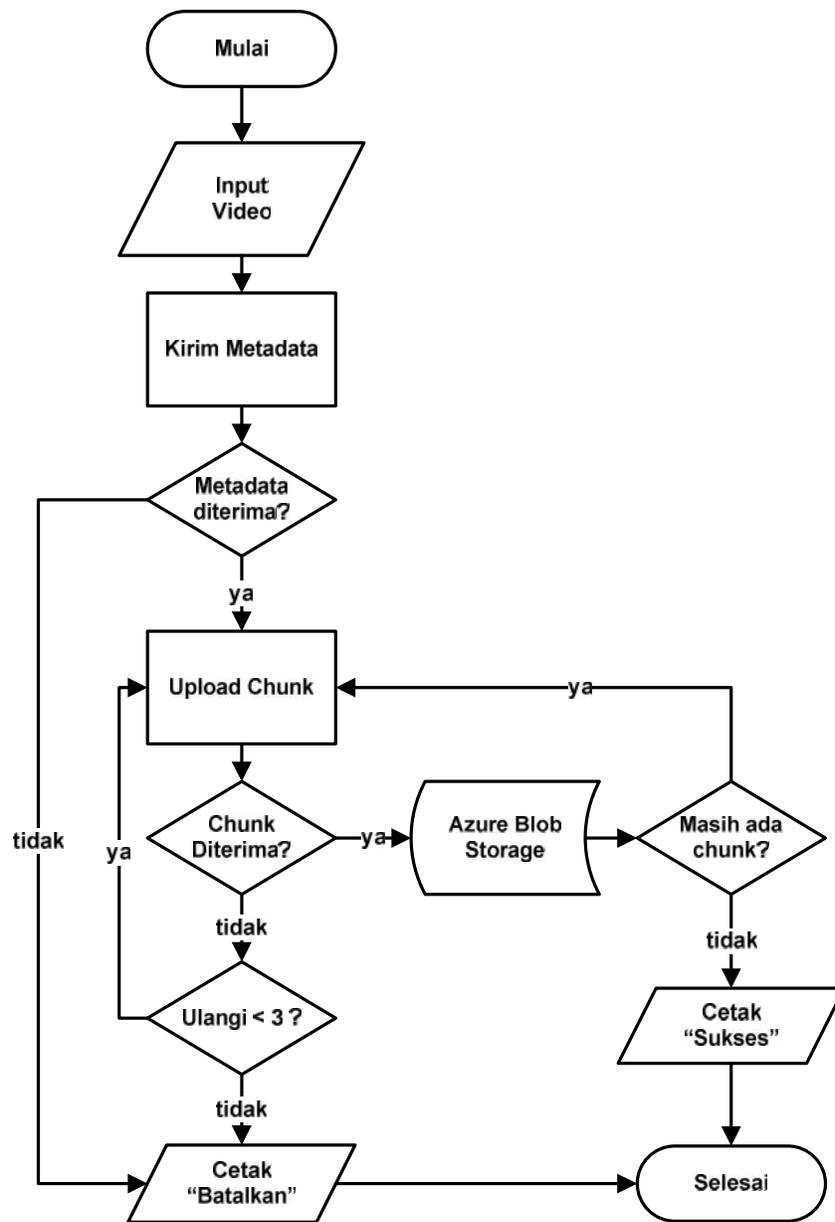


Gambar 7. Rancangan Basis Data

Gambar 7. merupakan rancangan basis data yang digunakan untuk membangun sistem yang terdiri dari 4 tabel yaitu tabel *UserProfile*, *MediaElements*, *Comments* dan *Categories*.

d. Rancangan Metode Upload Video

Gambar 8. menjelaskan alur pengaturan sistem saat video diupload. Setelah user menginputkan video, maka metadata video akan dikirim ke server untuk mengetahui berapa banyak chunk/potongan file yang akan dikirim dan besarnya masing-masing chunk. File video harus dipotong menjadi beberapa chunk karena IIS mempunyai batas maksimal upload yaitu 4MB. Sedangkan file yang diupload besarnya dapat lebih dari itu. Jika metadata berhasil dikirim, server akan melanjutkan ke tahap berikutnya yaitu mengupload chunk. Jika metadata tidak berhasil dikirim, maka upload video akan dibatalkan. Chunk akan dikirim satu per satu bagian. Jika dalam pengiriman ada kegagalan maka chunk akan dikirim ulang maksimal 3 kali pengiriman ulang dan jika tetap gagal maka upload akan dibatalkan. Sedangkan ketika chunk berhasil diupload akan langsung disimpan kedalam Azure Blob Storage dan melanjutkan pengiriman selanjutnya sampai tidak ada lagi chunk yang tersisa.



Gambar 8. Flowchart Pengaturan Upload Video

#### 4. Implementasi dan Analisa Sistem

Sistem dilengkapi dengan fasilitas progress bar pada saat proses *upload* yang berfungsi untuk mengetahui sampai di mana video berhasil di-*upload* dan berapa persen video telah berhasil di-*upload*.

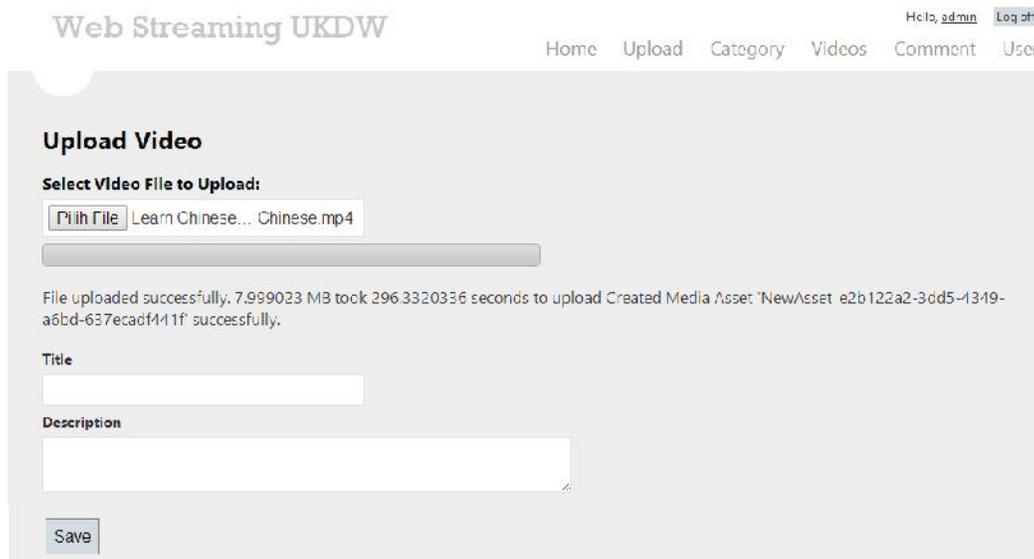


Gambar 9. Tampilan Progress Bar pada Saat Upload

Pada saat *upload* akan dilakukan beberapa proses di antaranya:

1. Mengidentifikasi besarnya *file* lalu membaginya ke dalam beberapa *chunk*/ potongan file dengan *slice* API pada *browser* karena batas maksimal IIS untuk mengupload file adalah 4 MB. Berikut ini adalah potongan kode program untuk mengidentifikasi besarnya file yang diupload dan membaginya ke dalam *chunk* yang tiap *chunk*-nya berukuran 1 MB lalu mengirimkannya sebagai metadata ke *server*. *Metadata* yang dikirim berupa jumlah *block*, nama *file* dan ukuran *file*.
2. Setelah metadata berhasil dikirim ke *server*, akan dilakukan pengiriman *file* yang akan dilakukan pada fungsi *sendFile* yang akan mengirim *file* satu persatu sesuai jumlah *block*. Jika *block* gagal dikirim, maka akan dilakukan pengiriman ulang setelah 10 detik dengan batas maksimal pengiriman ulang adalah 3 kali.
3. Setiap *chunk* yang berhasil dikirim, progress bar akan diperbaharui nilainya dengan fungsi *updateProgress* dan akan ditampilkan pesan sesuai dengan berapa persen file yang berhasil dikirim seperti terlihat pada Gambar 9.
4. Jika *chunk* tetap gagal setelah 3 kali dikirim ulang, maka pengiriman akan dihentikan dan akan ditampilkan pesan bahwa file gagal diupload.
5. Setelah semua *chunk* berhasil dikirim ke dalam *Azure Storage*, akan dibuatkan asset di *Azure Media Services* untuk menyajikan kembali video.

Setelah semua proses *upload* berhasil, sistem akan menampilkan panel untuk menginputkan data video. Data video yang diinputkan berupa: *Title*, *Description*, *Category* dan hak akses video seperti dapat dilihat pada Gambar 10.



Gambar 10. Tampilan Panel Input Data Video

## 5. Penutup

Berdasarkan hasil penelitian yang dilakukan, maka kesimpulan yang dapat diambil adalah :

- Penggunaan *Azure Media Services*, pembuatan sebuah *web* dapat menghasilkan sebuah *web streaming* yang menyajikan konten video secara lancar menggunakan teknologi *Adaptive Streaming* sehingga konten yang disajikan hampir tidak mengalami buffering. Dengan teknologi adaptive streaming, video akan dikirimkan ke *client* sedikit demi sedikit menyesuaikan dengan bandwidth yang dimiliki client begitu juga dengan kualitas video yang juga akan menyesuaikan secara otomatis.
- Website* mampu mengelola konten video dan data-data lain akan yang ditampilkan dalam website dengan menggunakan ASP.NET MVC 4. ASP.NET MVC 4 juga memudahkan dalam penulisan kode program dan *maintenance* program karena terbagi menjadi tiga layer yang memiliki fungsi masing-masing.
- Azure Storage* mampu memenuhi kebutuhan penanganan data dalam penyimpanan data besar berupa video karena data disimpan dalam bentuk *blob*. Pada proses *upload*, video harus dibagi menjadi beberapa chunk karena batas maksimal *upload* pada IIS hanya 4 MB sehingga video dibagi dengan menggunakan Slice API pada browser sebesar 1 MB untuk setiap *chunk*nya.
- Konten video dikelompokkan berdasarkan kategori tertentu dengan menyimpan data kategori pada video. Dengan adanya penyimpanan data kategori inilah yang membuat sistem dapat menampilkan video berdasarkan kategori tertentu untuk mempermudah pencarian.

## Daftar Pustaka

- Dykstra, T. (2012, July 30). *ASP.NET Tutorial*. Retrieved January 2014, from ASP.NET: <http://www.asp.net/mvc/tutorials/getting-started-with-ef-5-using-mvc-4>.
- Microsoft. (2012). *What is storage account?* Retrieved 11 2013, from Windows Azure: <http://www.windowsazure.com/en-us/documentation/articles/storage-what-is-account/>.

- Yan, M. (2012, Juni 10). *How does video streaming work? [Part 1: Progressive Download]*. Retrieved Agustus 2013, from Mingfey Yan Website: <http://www.mingfey.com/video-streaming-part-1/>.
- Yan, M. (2012, Juni 10). *How does video streaming work? [Part 2: Traditional Streaming]*. Retrieved Agustus 2013, from Mingfey Yan Website: [www.mingfey.com/traditional-streaming-video-streaming/](http://www.mingfey.com/traditional-streaming-video-streaming/).
- Yan, M. (2012, Juni 10). *How does video streaming work? [Part 3: Http-based Adaptive Streaming]*. Retrieved September 2013, from Mingfey Yan Website: <http://www.mingfey.com/video-streaming-part-3>.