IMPLEMENTASI METODE REST REQUEST PADA YOUTUBE WEB SERVICE UNTUK REPRESENTASI INFORMASI BERBASIS TIMELINE

Ryan Peterzon Hadjon[1]
22043527@students.ukdw.ac.id

Restyandito[2]
dito@ukdw.ac.id

Willy Sudiarto Raharjo[3]
willysr@ti.ukdw.ac.id

Abstract

Developing video search application using the service of a third party data (resources) provider is a complex process. Therefore to obtain the desired data, an efficient communication method is needed. This research explored the use of REST request communication method by using URL to identif video resources through YouTube API data. Video data that were identified will then be processed to get more relevant video search. Relevancy of the search result was determined by applying the mechanism of relevance-based system and user-based system. Based on the experiment analysis, it is concluded that key word has an important role in determining the relevance of the search result, as it is needed for timeline based representation.

Keywords: *web service, rest request, youtube data API*

1. Pendahuluan

Pemanfaatan data yang disediakan oleh pihak luar sebagai *resource* menjadi salah satu keunggulan dalam pengembangan aplikasi web yang saat ini telah memasuki generasi kedua dan dikenal dengan nama Web 2.0. Proses pemanfaatan data dari pihak luar menggunakan *web service* sebagai sarana penghubung untuk mempermudah pengembangan aplikasi web.

Dalam mengakses data dari *web service*, aplikasi web yang dibangun membutuhkan teknik komunikasi yang memungkinkan sumber (*resource*) dapat teridentifikasi melalui pemanggilan sebuah *request*. Salah satu teknik komunikasi yang dapat digunakan untuk memenuhi tujuan tersebut adalah REST *(Representational State Transfer)*. REST merupakan teknik komunikasi web yang menggunakan HTTP *(Hyper Text Transfer Protocol)* sebagai metode transfer data serta XML *(eXtensible Markup Languange)* sebagai bentuk representasi data.

Dalam penelitian ini, penulis mencoba membangun sebuah aplikasi pencarian video berbasis web dengan mengimplementasikan metode *REST request* melalui layanan YouTube Web Service untuk merepresentasikan kembali informasi yang didapat kedalam bentuk timeline. Agar dapat direpresentasikan dalam bentuk timeline, maka seluruh data video yang berada didalam timeline harus relevan dengan *keyword* masukan. Mekanisme yang digunakan untuk menentukan relevansi adalah dengan menerapkan relevansi berbasis sistem

[1]Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana
[2]Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana
[3]Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana

dan relevansi berbasis pengguna. Penentuan relevansi didasarkan pada kemampuan internal sistem serta *feedback* eksternal dari pengguna. Dengan demikian relevansi hasil pencarian video dengan kata kunci masukan yang ditampilkan oleh sistem akan menjadi lebih obyektif.

2. Tinjauan Pustaka

2.1. REST (*Representational State Transfer*)

Representational State Transfer (REST) merupakan model arsitektur untuk perangkat lunak *hypermedia* terdistribusi. Beberapa aspek penting yang mendasari model arsitektur REST ini antara lain :

- **Resources**
  Spesifikasi URI dalam RFC 2396 menyebutkan *resource* sebagai segala sesuatu yang memiliki identitas. Allamaraju mendefinisikan r*esource* sebagai segala sesuatu yang dapat diidentifikasi oleh URI (Allamaraju, 2010). Dengan demikian URI (*Uniform Resource Identifier*) merupakan penyedia referensi kepada sesuatu yang memiliki identitas.

- **Representation**
  Secara umum, resource dapat diidentifikasi dan diakses melalui konstruksi URL ataupun URI. Semua hal dalam web, baik itu berupa halaman, gambar, dan lainnya pada dasarnya dapat dikategorikan sebagai resource (Glover, 2008). Namun tidak semua resource dapat diakses. Oleh karena itu dalam konteks web, bukanlah resource yang akan diakses, melainkan representasi dari resource itu sendiri. Sebuah resource dapat direpresentasikan dalam banyak model.

- **State**

  Dalam pemrograman berorientasi objek, *state* dari sebuah objek merupakan data yang dibawa oleh variabel anggota dari objek tersebut. Hal ini sangat penting jika terdapat operasi yang memiliki banyak langkah, maka state sangat dibutuhkan oleh objek untuk mengingat nilai-nilai yang dibawa pada setiap langkah.

- **Transfer (HTTP methods)**

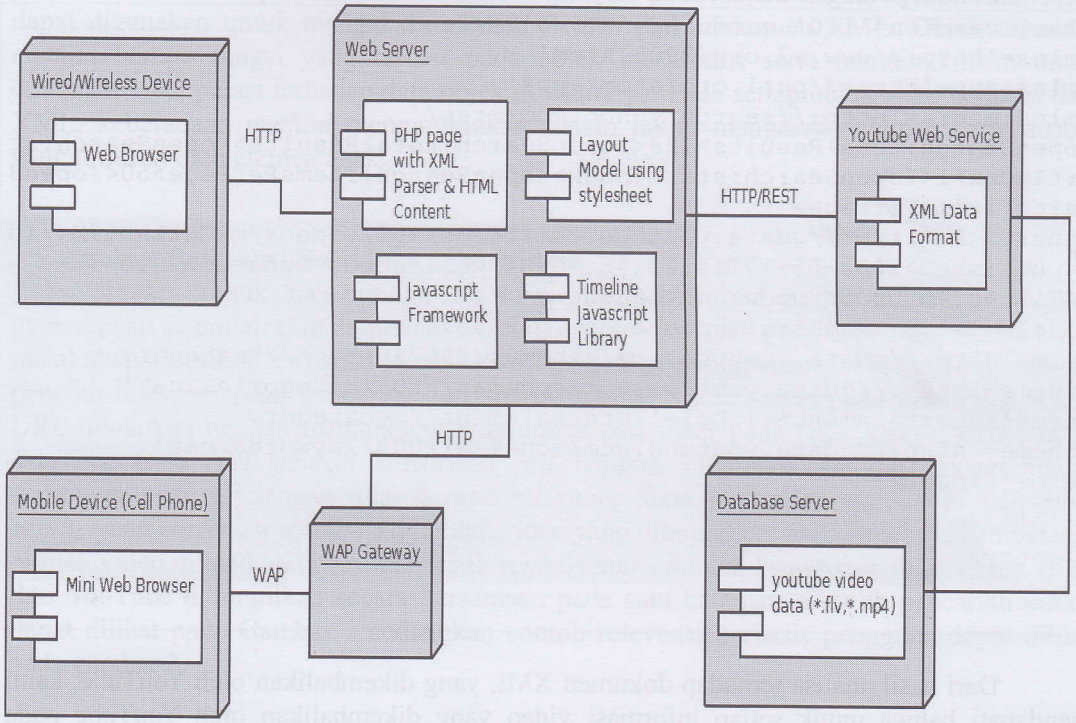
  Salah satu karakteristik RESTful Web Service seperti yang disampaikan oleh Rodriguez adalah penggunaan metode transfer dan produksi data yang ditujukan bagi aplikasi klien untuk melakukan pengambilan resource, pemanggilan data dari Web Server, maupun melakukan eksekusi terhadap *query* yang nantinya diharapkan akan direspon oleh Web Server dengan mengembalikan resource yang tepat sesuai dengan *request* yang dikirimkan (Rodriguez, 2008). Adapun metode transfer yang dilakukan dalam model arsitektur web REST ini menggunakan metode yang sama pada protokol web HTTP. Arsitektur sistem berbasis REST dapat dilihat pada Gambar 1.

2.2. YouTube Data API

Raymond Yee mendefinisikan *public API* sebagai "...suatu channel resmi yang disediakan oleh sebuah website untuk memberikan akses secara terprogram terhadap data maupun service yang ada dalam website tersebut." (Yee, 2008, hlm. 23). YouTube Data API merupakan sebuah *public API* yang disediakan oleh YouTube untuk dapat dimanfaatkan dalam membangun sebuah *client application* guna berinteraksi dengan video-video resource yang ada didalamnya. Model interaksi tersebut berupa suatu *request-response* yang dapat dibentuk melalui pemanggilan sebuah URL disertai dengan parameter-parameter pendukung yang dapat disesuaikan dengan kebutuhan pengguna.

Dengan menggunakan YouTube Data API, kami dapat melakukan pencarian

terhadap video-video tertentu serta dapat menampilkannya kembali dalam format yang diinginkan berdasarkan parameter masukan pada URL yang dibentuk. Contoh pembentukan URL pada YouTube API menggunakan *api projection* yang merespon *request* pencarian video dengan kata kunci "***skateboarding dog***" dapat dilihat pada potongan URL berikut:
`http://gdata.youtube.com/feeds/api/videos?q=skateboarding+dog&start-index=21&max-results=10&v=2`



Gambar 1. Arsitektur Sistem Berbasis REST

3. Hasil dan Pembahasan

3.1. Konstruksi URL Berbasis REST

Proses konstruksi *URL* berbasis *REST* oleh sistem dilakukan dengan menggunakan *base URL* yang telah disediakan oleh YouTube Data API untuk memudahkan pembentukan *REST request* selanjutnya. Berikut ini adalah *base URL* yang digunakan: `http://gdata.youtube.com/feeds/api/`. Base URL tersebut merupakan konstruksi URL dasar yang nantinya dapat ditambahkan sesuai dengan kebutuhan untuk mengakses video-video tertentu pada YouTube. Konstruksi URL untuk setiap jenis video dengan menggunakan REST request pada YouTube berbeda-beda namun base URL yang digunakan tetap sama.

3.2. Response URL Berbasis REST

Hasil pengiriman request URL berbasis REST yang dikembalikan oleh YouTube berupa dokumen XML yang mendeskripsikan video-video sesuai dengan kata kunci masukan pada sistem. Dokumen XML ini tidak secara eksplisit mengembalikan video yang diminta melainkan hanya memuat informasi mendetail tentang video yang diminta. Oleh karena itu, dibutuhkan proses *parsing* lebih lanjut pada dokumen XML yang dikembalikan tersebut untuk menampilkan video yang sesuai dan relevan dengan kata kunci masukan.

3.3. Parsing dokumen XML dari YouTube

Proses *parsing* dokumen XML yang dikembalikan oleh YouTube penting dilakukan untuk mendapatkan video yang diinginkan secara eksplisit sesuai dengan kata kunci masukan pada form pencarian sistem. Secara sederhana, proses *parsing* ini dikerjakan oleh

sistem dengan terlebih dahulu melakukan pembacaan tag-tag XML pada dokumen XML yang dikembalikan oleh YouTube, kemudian dari proses pembacaan tersebut akan dipilih tag-tag XML yang disesuaikan dengan kebutuhan sistem dalam melakukan pencarian terhadap kata kunci masukan. Namun, untuk dapat melakukan pembacaan terhadap dokumen XML tersebut, kami terlebih dahulu harus mampu memahami dan menganalisa struktur dokumen XML yang dikembalikan oleh YouTube secara mendetail.

Contoh potongan dokumen XML yang dimaksudkan adalah seperti berikut:

```
<?xml version='1.0' encoding='UTF-8'?><feed
xmlns='http://www.w3.org/2005/Atom'
xmlns:app='http://purl.org/atom/app#'
xmlns:media='http://search.yahoo.com/mrss/'
<openSearch:totalResults>128</openSearch:totalResults><openSearch:st
artIndex>1</openSearch:startIndex><openSearch:itemsPerPage>50</openS
earch:itemsPerPage>
<entry><id>http://gdata.youtube.com/feeds/api/videos/mwGHkLhD67M</id
><published>2008-05-17T15:20:56.000Z</published><updated>2010-08-
24T17:00:45.000Z</updated><category
scheme='http://schemas.google.com/g/2005#kind'
term='http://gdata.youtube.com/schemas/2007#video'/><category
scheme='http://gdata.youtube.com/schemas/2007/categories.cat'
term='Entertainment' label='Entertainment'/><category
scheme='http://gdata.youtube.com/schemas/2007/keywords.cat'
term='final'/>
......
</entry>
</feed>
```

Dari hasil analisa terhadap dokumen XML yang dikembalikan oleh YouTube, kami mendapati bahwa untuk setiap informasi video yang dikembalikan oleh YouTube pada dokumen XML tersebut masing-masing dipisahkan oleh tag <entry></entry> sebagai penanda antara satu video dengan video lainnya yang ditemukan melalui proses pencarian sebelumnya. Hal ini akan lebih memudahkan dalam melakukan pembacaan terhadap dokumen XML dari Youtube karena fokus pembacaan hanya akan mengarah pada semua tag-tag yang berada dalam tag <entry></entry> tersebut. Seluruh tag XML yang berada dalam tag <entry></entry> kemudian akan dibaca oleh sistem dengan mengidentifikasi *namespace* dan entitas-entitas XML yang berada didalamnya. Sekumpulan nama tag-tag XML yang terkumpul menjadi satu akan membentuk *namespace* sementara entitas XML sendiri merupakan kode simbol yang digunakan untuk melakukan proses *encode* terhadap karakter-karakter spesial yang berada dalam dokumen XML.

Namespace digunakan untuk mengorganisir kumpulan-kumpulan tag dalam XML menjadi bentuk logis yang dapat dipahami oleh sistem ketika akan melakukan pembacaan terhadap dokumen XML.

3.4. Rekonstruksi Skema Data XML dari YouTube

Dengan menganalisa skema data XML hasil kembalian YouTube, kami mendapati bahwa struktur XML tersebut pada dasarnya merupakan susunan elemen atau hirarki elemen yang didalamnya memuat tipe hubungan atau relasi induk-anak (*parent-child relationship*).

Elemen yang dimaksud dalam hal ini merupakan konsep abstrak dari tag-tag XML dan data yang termuat didalamnya. Hubungan antara elemen ini sangat berpengaruh dalam proses ekstrasi terhadap data yang dibutuhkan. Setiap elemen yang ditutupi oleh elemen lainnya merupakan elemen anak sedangkan elemen yang menutupinya merupakan elemen induk. Oleh karena itu, proses ekstrasi terhadap skema data XML YouTube harus memperhatikan susunan hirarki serta hubungan antar elemen tersebut.

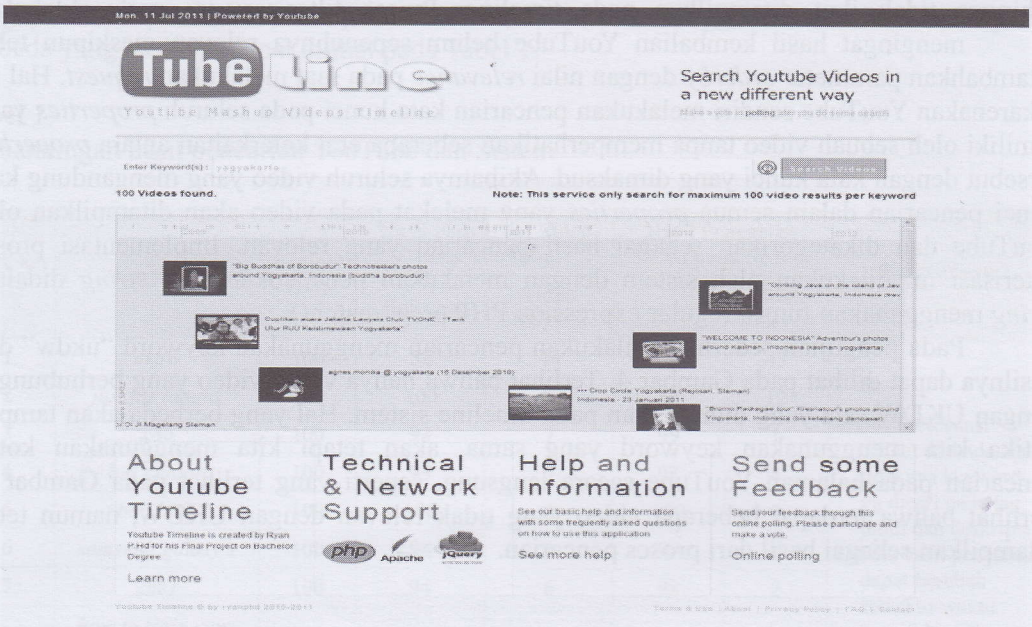
Dengan melihat konsep tersebut di atas, maka kendala pertama yang dihadapi dalam penelitian ini adalah bagaimana menerapkan konsep tersebut pada sistem untuk mengakses

elemen-elemen anak dalam dokumen XML yang dikembalikan oleh YouTube. Penggunaan tipe data biasa untuk menangani kendala tersebut tidak dimungkinkan karena memiliki keterbatasan dalam penerapan konsep yang bersifat abstrak.

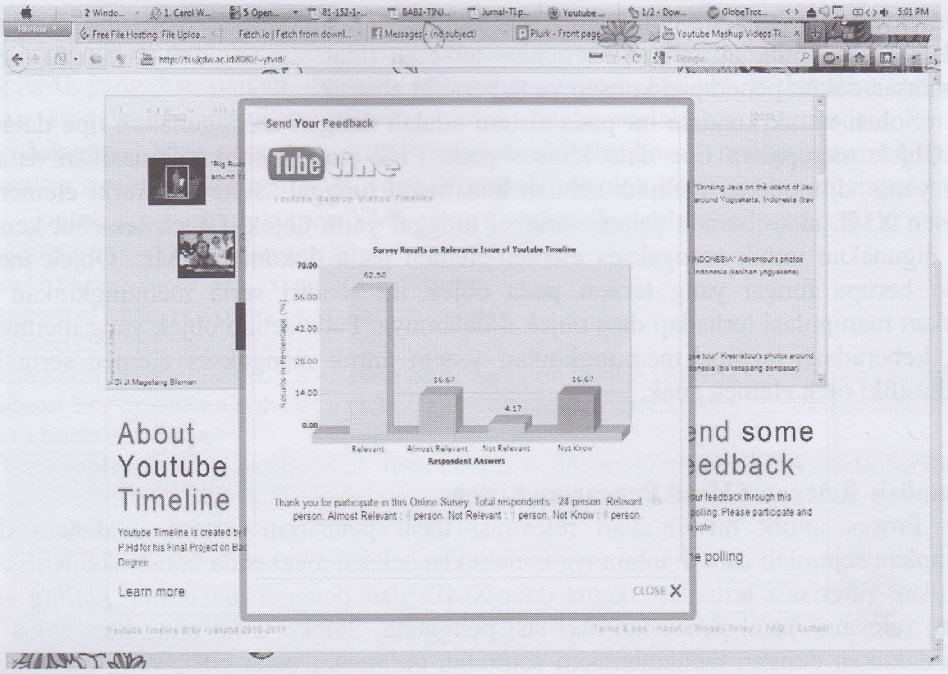
Solusi untuk kendala ini pada sistem adalah dengan menggunakan tipe data objek PHP. Objek merupakan tipe data khusus pada PHP yang mengkombinasikan data serta fungsi yang dibutuhkan menjadi sebuah konstruksi tunggal. Seluruh hirarki elemen pada dokumen XML akan berada dalam variabel tunggal yaitu objek. Objek tersebut kemudian dapat digunakan untuk mengakses elemen-elemen pada dokumen XML. Objek memiliki *method* berupa fungsi yang terikat pada objek itu sendiri serta memungkinkan untuk dilakukan manipulasi terhadap data objek didalamnya. Pada setiap objek yang memuat data XML, keberadaan *method* memungkinkan sistem untuk mengakses elemen serta atribut yang dimiliki oleh elemen anak.

3.5. Analisis Relevansi Hasil Pencarian Sistem

Proses untuk menentukan relevansi hasil pencarian sistem dilakukan dengan menerapkan sejumlah cara diantaranya melakukan seleksi awal pada konstruksi REST URL, melakukan filterisasi terhadap skema data XML, dan penggunaan *online polling* sebagai penentu relevansi hasil pencarian dari sisi pengguna. Seleksi awal pada konstruksi REST URL dilakukan dengan menambahkan sejumlah parameter pada URL yang dikirimkan ke YouTube Data API sebagai konstruksi *rest request*. Parameter yang dimaksud adalah parameter *orderby* dengan nilai berupa *relevance* serta parameter *start-index* dan *max-results* yang berfungsi membatasi jumlah video yang ditampilkan pada timeline. Pembatasan jumlah video dilakukan untuk mencegah *request time out* oleh browser apabila semua video dari YouTube ditampilkan secara bersamaan pada satu halaman. Contoh pencarian sistem dapat dilihat pada Gambar 2 sedangkan contoh relevansi berbasis pengguna dapat dilihat pada gambar 3.



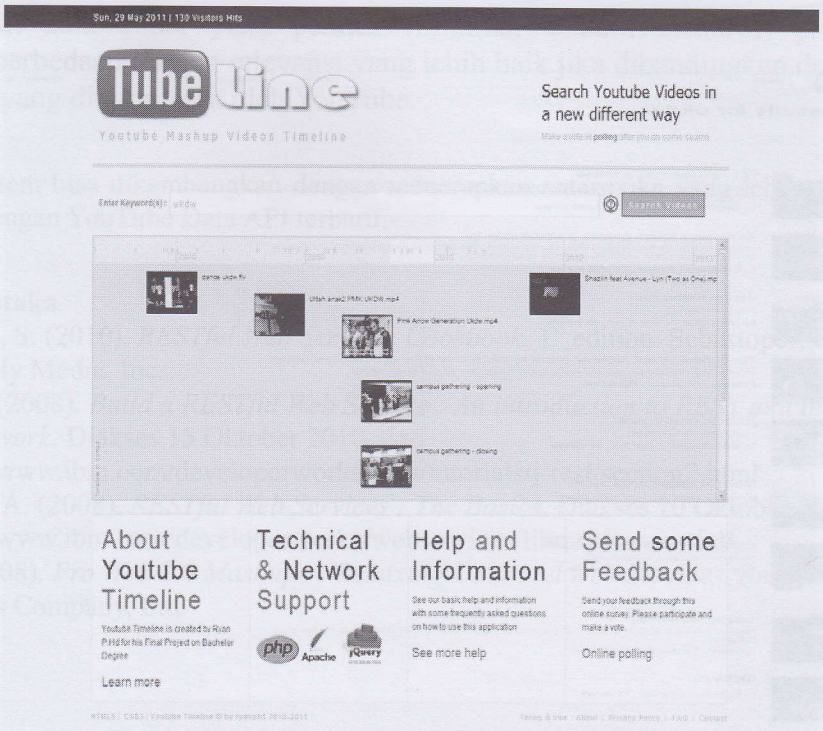
Gambar 2. Contoh Pencarian Sistem



Gambar 3. Relevansi Berbasis Pengguna Menggunakan Online Polling

Filterisasi terhadap skema data XML diimplementasikan pada sistem dengan melakukan pencarian kata kunci masukan hanya pada bagian judul serta deskripsi dari video pada skema data XML yang dikembalikan oleh YouTube. Jika kata kunci masukan ditemukan pada salah satu dari kedua bagian tersebut, maka video terkait akan dikategorikan sebagai video yang relevan oleh sistem sehingga dapat ditampilkan pada timeline. Jika tidak ditemukan maka video akan dianggap sebagai video yang tidak relevan sehingga tidak ikut ditampilkan pada timeline. Proses filterisasi ini perlu dilakukan mengingat hasil kembalian YouTube belum sepenuhnya relevan meskipun telah ditambahkan parameter *orderby* dengan nilai *relevance* pada saat melakukan *request*. Hal ini dikarenakan YouTube sendiri melakukan pencarian kata kunci pada seluruh *properties* yang dimiliki oleh sebuah video tanpa memperhatikan seberapa erat keterkaitan antara *properties* tersebut dengan kata kunci yang dimaksud. Akibatnya seluruh video yang mengandung kata kunci pencarian dalam semua *properties* yang melekat pada video akan ditampilkan oleh YouTube dan dikategorikan sebagai hasil pencarian yang relevan. Implementasi proses filterisasi ini dilakukan oleh sistem dengan melakukan pencocokan pola *string* didalam *string* menggunakan fungsi regular expression PHP preg_match().

Pada pengujian sederhana dilakukan pencarian menggunakan keyword "ukdw" dan hasilnya dapat dilihat pada Gambar 4. Terlihat bahwa hanya video-video yang berhubungan dengan UKDW saja yang ditampilkan pada timeline sistem. Hal yang berbeda akan tampak ketika kita menggunakan keyword yang sama, akan tetapi kita menggunakan kotak pencarian pada halaman YouTube secara langsung, seperti yang terlihat pada Gambar 5. Terlihat bahwa terdapat beberapa video yang tidak relevan dengan UKDW, namun tetap ditampilkan sebagai hasil dari proses pencarian.



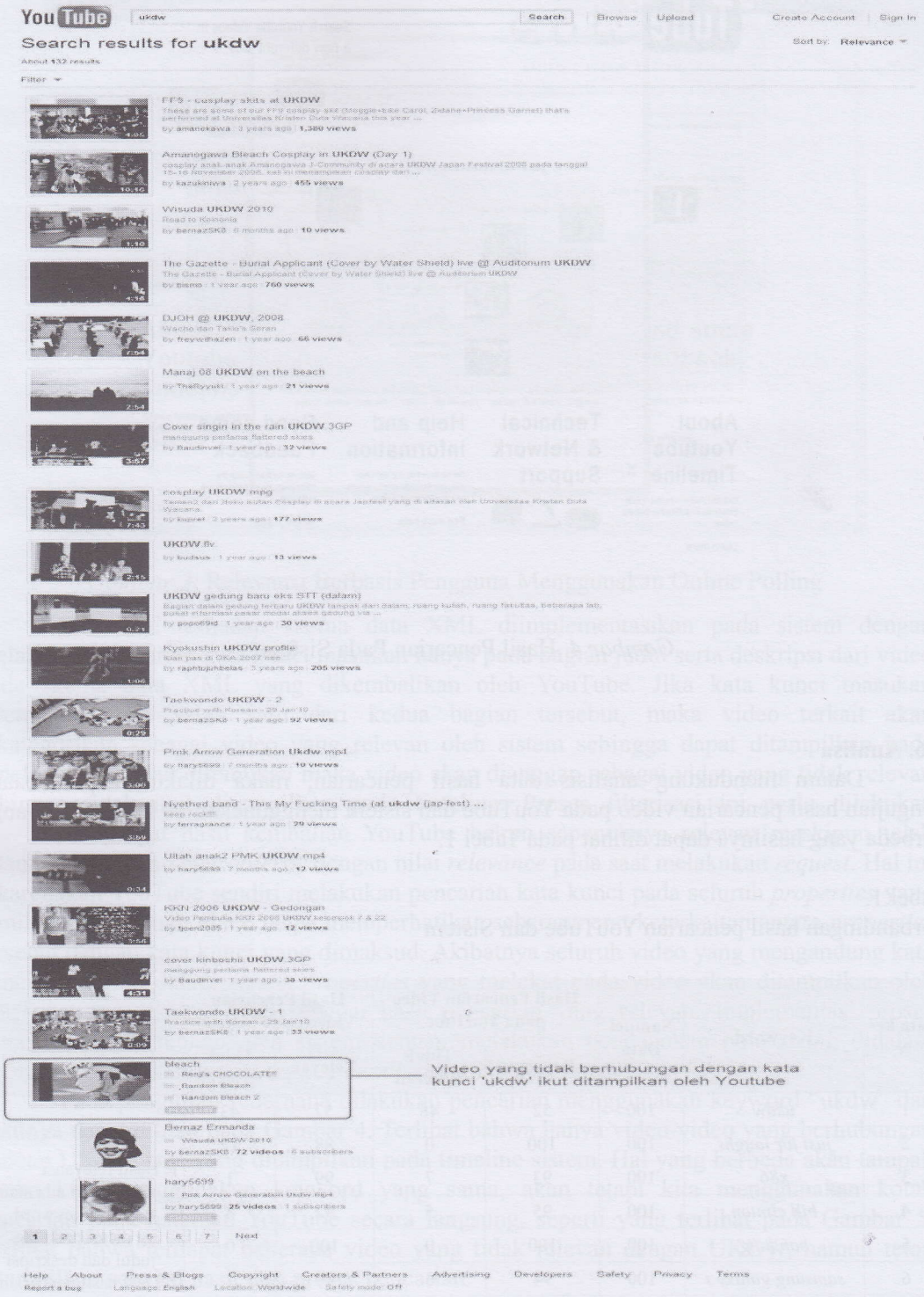
Gambar 4. Hasil Pencarian Pada Sistem

3.6. Analisa

Dalam mendukung analisis data hasil pencarian, maka dilakukan percobaan pengujian hasil pencarian video pada YouTube dan sistem menggunakan 10 kata kunci yang berbeda yang hasilnya dapat dilihat pada Tabel 1.

Tabel 1.
Perbandingan hasil pencarian YouTube dan Sistem

Data ke-N	Keyword	Sampel Data	Hasil Pencarian Video pada YouTube		Hasil Pencarian Video pada Sistem		Keterangan
			Relevan	Tidak Relevan	Relevan	Tidak Relevan	
1	*ukdw*	100	52	48	57	43	1. Hasil relevansi dilihat dari ada atau tidaknya keyword yg dicari pada tag judul dan deskripsi 2. Data hasil pencarian dapat berubah sewaktu-waktu karena bersifat dinamis. 3. Data diambil pada tanggal 14 juni 2011.
2	*just for laughs*	100	100	0	88	12	
3	*x86*	100	93	7	99	1	
4	*bill clinton*	100	95	5	92	8	
5	*bandung*	100	100	0	100	0	
6	*samsung galaxy s*	100	94	6	99	1	
7	*2ne1*	100	94	6	98	2	
8	*how to train your dragon*	100	100	0	100	0	
9	*Kungfu panda 2*	100	96	4	98	2	
10	*my girlfriend is a gumiho*	100	97	3	98	2	



Gambar 5. Pencarian pada halaman YouTube

4. Kesimpulan dan Saran

4.1. Kesimpulan

Proses pengambilan video pada YouTube oleh sistem diimplementasikan melalui konstruksi URL berbasis REST yang dikirimkan kepada YouTube dengan memanfaatkan Web Service yang dimilikinya. Hasilnya kemudian akan disaring untuk mendapatkan video yang relevan. Relevansi pencarian video sangat dipengaruhi oleh pemilihan kata kunci yang tepat. Berdasarkan hasil pengujian,

penggunaan kata kunci yang pendek (minimal 4 karakter/huruf) pada sistem memiliki perbedaan tingkat relevansi yang lebih baik jika dibandingkan dengan hasil pencarian yang ditampilkan oleh YouTube.

4.2. Saran

Sistem bisa dikembangkan dengan menerapkan antarmuka yang lebih dinamis dan diupdate dengan YouTube Data API terbaru.

Daftar Pustaka

Allamaraju, S. (2010). *RESTful Web Services Cookbook*, 1st edition. Sebastopol, CA : O'Reilly Media, Inc.

Glover, A. (2008). *Build a RESTful Web Service : An introduction to REST and the Restless framework.* Diakses 15 Oktober 2010, dari http://www.ibm.com/developerworks/java/tutorials/j-rest/section2.html

Rodriguez, A. (2008). *RESTful Web Services : The Basics*. Diakses 10 Oktober 2010, dari http://www.ibm.com/developerworks/webservices/library/ws-restful/

Yee, R. (2008). *Pro Web 2.0 Mashups : Remixing Data and Web Services. Berkeley*, CA : Apress Company, Ltd.