Jurnal Politeknik Caltex Riau
http://jurnal.pcr.ac.id

Ekstraksi Tabel HTML ke *Database Management System* dengan Pendekatan *Service Oriented Architecture*

Memen Akbar[1] dan Ardianto Wibowo[2]

[1]Politeknik Caltex Riau, email: memen@pcr.ac.id
[2]Politeknik Caltex Riau, email: ardie@pcr.ac.id

Abstrak

Seiring dengan perkembangan berbagai bisnis proses, keberadaan data yang ada semakin berkembang dalam berbagai bentuk. Salah satu bentuk data adalah tabel di dalam sebuah halaman HTML. Berbeda dengan tabel pada database, tabel HTML memiliki struktur yang beragam. Nama atribut, pada tabel database selalu berada pada baris pertama. Sedangkan, pada tabel HTML, nama atribut dapat berada pada kolom pertama (row wise table) atau pada baris dan kolom pertama (column-row wise table). Sehingga, dalam proses ekstraksi, terlebih dahulu harus dikenali bagian pada tabel yang berperan sebagai nama kolom dan bagian tabel yang berperan sebagai data. Setelah diekstraksi, agar dapat digunakan untuk kebutuhan yang lebih lanjut, tabel HTML tersebut disimpan dalam sebuah database. Proses ini dilakukan dengan pendekatan Service Oriented Architecture sehingga ekstraksi dapat dilakukan secara otomatis. Penelitian ini mengembangkan sebuah model ekstraksi tabel HTML dengan pendekatan semantic tree dan memanfaatkan SOA dalam proses loading data ke database. Model yang dihasilkan berhasil mengekstrak tabel dari halaman web dengan 3 jenis bentuk layout, yaitu column wise, row wise, dan column-row wise dan menyimpannya dalam DBMS.

Kata kunci: *tabel HTML, SOA, DBMS, ekstraksi*

Abstract

Along with the development of various business processes, the existence of existing data is growing in various forms. One form of data is a table inside an HTML page. In contrast to tables in a database, HTML tables have diverse structures. The name of the attribute, in the database table is always on the first line. Whereas, in the HTML table, the attribute name can be in the first column (row wise table) or on the first row and column (column-row wise table). Thus, in the extraction process, it must first be identified in the table that acts as the column name and the table part that acts as the data. Once extracted, in order to be used for further needs, the HTML table is stored in a database. This process is done with Service Oriented Architecture approach so that extraction can be done automatically. This research develops an extraction model of HTML table with semantic tree approach and uses SOA in process of loading data to database. The resulting model successfully extracted the table from the web page with 3 types of layout, ie column wise, row wise, and column-row wise and stored it in DBMS.

Keywords: *HTML table, SOA, DBMS, extraction*

Dokumen diterima pada 30 Oktober, 2017
Dipublikasikan pada 15 November, 2017

1. Pendahuluan

Seiring dengan perkembangan berbagai bisnis proses, keberadaan data yang ada semakin berkembang dalam berbagai bentuk. Salah satu bentuk data adalah tabel di dalam sebuah halaman HTML. Berbeda dengan tabel pada database, tabel HTML memiliki struktur yang beragam. Nama atribut, pada tabel database selalu berada pada baris pertama. Sedangkan, pada tabel HTML, nama atribut dapat berada pada kolom pertama (*row wise table*) atau pada baris dan kolom pertama (*column-row wise table*). Sehingga, dalam proses ekstraksi, terlebih dahulu harus dikenali bagian pada tabel yang berperan sebagai nama kolom dan bagian tabel yang berperan sebagai data. Untuk itu, diperlukan sebuah solusi untuk menanganinya. Salah satu solusi yang diharapkan bisa dikembangkan adalah dengan membuat sebuah struktur standar yang menjadi acuan untuk memisahkan bagian nama atribut dan data.

Setelah diekstraksi, agar dapat digunakan untuk kebutuhan yang lebih lanjut, tabel HTML tersebut disimpan dalam sebuah database. Agar dapat disimpan dalam database, struktur tabel dibuat dalam bentuk XML. Setelah menjadi bentuk XML, maka tabel dapat disimpan ke database. Proses ini dilakukan dengan pendekatan *Service Oriented Architecture* sehingga ekstraksi dapat dilakukan secara otomatis.

Salah satu teknologi yang sedang berkembang sebagai mekanisme komunikasi antar aplikasi adalah *Service Oriented Architecture (SOA). SOA* merupakan sebuah istilah untuk merepresentasikan sebuah model dimana terdapat serangkaian proses logika otomatisasi yang di pecah kembali ke dalam bagian-bagian logika yang lebih kecil [1]. *SOA* dapat dilihat sebagai sebuah paradigma untuk menyelesaikan masalah integrasi dari berbagai level aplikasi. *SOA* mengirimkan fungsionalitas dari sebuah aplikasi ke aplikasi yang lain [2].

Sesuai dengan uraian di atas, pada penelitian ini akan dikembangkan sebuah model ekstraksi tabel HTML dengan pendekatan semantic tree dan memanfaatkan *SOA* dalam proses loading data ke database. Dengan adanya model ekstraksi ini, diharapkan tabel pada halaman web dapat disimpan ke dalam data base secara otomatis.

Berdasarkan kajian dan rumusan masalah pada poin-poin sebelumnya, ditentukanlah ruang lingkup yang dilakukan di dalam penelitian ini. Adapun ruang lingkup penelitian tersebut adalah sebagai berikut.

1. Fokus dari penelitian bukan untuk pengembangan sistem baru, melainkan kepada permodelan konseptual ekstraksi data semi terstruktur berupa tabel pada halaman HTML ke dalam bentuk data terstruktur pada tabel *DBMS.*
2. Aplikasi sederhana terkait studi kasus akan tetap dikembangkan sebatas sebagai sarana validasi / pengujian model yang dihasilkan.

Bagian berikut tulisan ini berisikan tinjauan pustaka yang terkait dengan ekstraksi tabel HTML. Kemudian diikuti dengan metodologi penelitian pada bagian berikutnya. Hasil implementasi dan pengujian sistem ditempatkan pada bagian keempat sebagai penutup tulisan ini.

2. TINJAUAN PUSTAKA

2.1 Layout Tabel pada Halaman Web

Tabel tersusun atas tiga bagian utama, yaitu *caption*, atribut, dan nilai data. *Caption* merupakan bagian identitas tabel, biasanya dinamakan dengan judul tabel. Atribut dan nilai data merupakan bagian data pada tabel. Atribut merupakan bagian yang menjadi identitas sekelompok nilai data pada suatu baris atau kolom. Bentuk umum tabel dapat dilihat pada

Gambar 1. Tabel ini memiliki n buah atribut (C_1, C_2, ..., C_n) dengan masing-masingnya memiliki m buah nilai data.

C		
C_1	...	C_n
$a_{1,1}$	...	$a_{1,n}$
⋮	...	⋮
$a_{m,1}$	...	$a_{m,n}$

Gambar 1. Bentuk Umum Tabel

Atribut biasanya berada pada baris pertama sebuah tabel. Namun, pada banyak penggunaannya, posisi atribut dan nilai data pada suatu tabel bervariasi. Posisi atribut dan nilai data ini yang dinamakan dengan *layout* tabel. [3] membagi *layout* tabel menjadi 2 kategori, yakni *layout* sederhana dan kompleks. [4] menambahkan satu kategori lagi, yakni tidak terstruktur.

Tabel dengan *layout* sederhana memiliki atribut yang berada pada baris dan atau kolom pertama. Terdapat 3 *layout* yang termasuk kategori ini, yaitu: (1) *Column wise table*, (2) *Row wise table*, dan (3) *Column-row wise table.* Tabel 1 dan Tabel 2 merupakan contoh untuk tabel dengan *layout row wise* dan *column-row wise*.

Tabel 1. Contoh Tabel dengan *layout row wise*

Temperature	66°
Humidity	75%
Precip. Today	n/a
Wind	SE at 13 mph
Barometer	30.02 in
Dewpoint	58°
Visibility	10 mi
Sunrise	5:31 a.m
Sunset	8:15 p.m

Tabel 2. Contoh Tabel dengan *layout column-row wise*

	Q1	**Q2**	**Q3**	**Q4**
dept1	1.30	1.32	1.30	1.35
dept2	1.40	1.35	1.15	1.20

Tabel dengan *layout* yang lebih kompleks memiliki atribut dan nilai data dengan posisi yang lebih bervariasi. Terdapat dua jenis *layout* yang termasuk dalam kategori ini, yaitu: (1) *composite table*, dan (2) *mixed-cell table.*

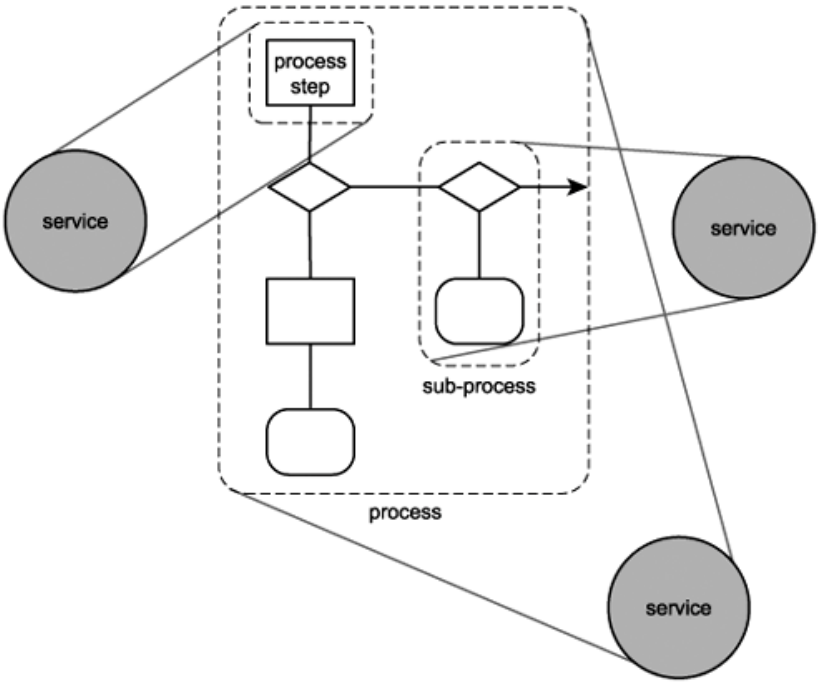
2.2 Service Oriented Architecture (SOA)

Service Oriented Architecture(SOA) merupakan sebuah istilah untuk merepresentasikan sebuah model dimana terdapat serangkaian proses logika otomatisasi (*automationlogic*) yang di

dekomposisikan ke dalam bagian-bagian logika yang lebih kecil. Apabila dilihat secara kolektif, unit-unit logika tersebut dapat meliputi otomatisasi suatu bisnis proses yang besar. Sedangkan apabila dilihat secara individu, maka unit-unit logika tersebut dapat didistribusikan secara terpisah. [1]

Untuk mengontrol dan mengelola unit-unit logika tersebut, *SOA* membungkusnya (mengenkapsulasi) ke dalam kelompok-kelompok logika yang berbeda-beda. Kelompok tersebut dapat berupa *business task*, *business entity*, atau jenis kelompok yang lain. Ukuran dan kompleksitas antar kelompok logika yang satu dengan kelompok logika yang lain bisa jadi sangat bervariasi, tergantung dari bisnis proses yang ditanganinya. Selain itu, sebuah kelompok logika bisa mengambil logika-logika unit dari kelompok logika yang lainnya.

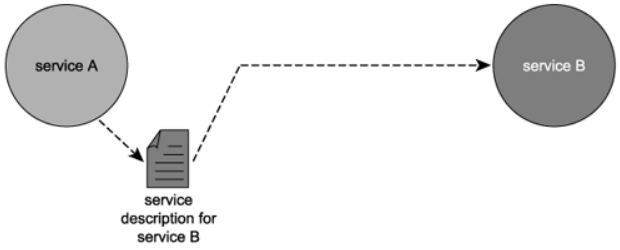
Di dalam *SOA*, unit-unit logika dan kelompok logika tersebut digunakan oleh *service*. Penggunaannya sendiri bisa bersifat dinamis, artinya adalah kelompok logika yang terlibat di dalam *service* bisa jadi berbeda-beda, tergantung dari task yang sedang dijalankan oleh *service* tersebut. Penjelasan diatas dapat digambarkan seperti Gambar 2 [1].



Gambar 2. Diagram Konsep *Service* & Logika (Erl, 2005)

Dari Gambar 2 dapat dilihat bahwa sebuah *service* dapat digunakan ulang oleh *service* lain yang terlibat dalam sebuah task, ataupun langsung digunakan oleh task yang lain. Bentuk hubungan antar *service* ini didasarkan dari mekanisme komunikasi saling mengenali yang dapat diketahui dari deskripsi masing-masing *service* (*service description*), untuk dapat menentukan *service* mana yang perlu dilibatkan ataupun tidak.

Secara umum, sebuah *service description* berisi nama dari *service*, data input yang diharapkan, serta data output yang akan dikembalikan ke pemanggil *service* tersebut. Sebagai contoh, Gambar 3 berikut menunjukkan bahwa *service* A mengenali *service* B dengan cara mengakses *service* description milik *service* B.



Gambar 3. Mekanisme Komunikasi Antar *Service* (Erl, 2005)

2.3 State of The Art

Pembahasan mengenai ekstraksi tabel HTML sudah banyak dilakukan. [5] melakukan ekstraksi tabel dari halaman web dengan memanfaatkan document object model (DOM) yang disediakan pada halaman HTML. Penelitian ini berhasil mengekstrak tabel HTML dan mengubahnya menjadi file Microsoft Excel. Namun, penelitian ini tidak mampu mengekstrak tabel dengan layout composite dan mixed-cell.

Penelitian yang dilakukan oleh [6] berhasil membuat sebuah engine dengan bahasa python untuk mengekstrak tabel dari halaman web dengan beberapa layout yang disampaikan oleh [3] dan [4]. Keluaran dari hasil ekstraksi pada engine yang dihasilkan berupa tabel standar berbentuk column wise. Ekstraksi menjadi bagian dari engine yang dihasilkan karena fungsi utama engine ini adalah untuk mengintegrasikan beberapa tabel dari beberapa halaman web. Tabel hasil integrasi langsung ditampilkan kepada pengguna dalam bentuk halaman HTML.

Kebutuhan untuk integrasi tabel dari beberapa halaman web tiap pengguna berbeda. Oleh karena itu, pemisahan mesin ekstraksi dan mesin integrasi dapat menjadi solusi untuk menghadapi kebutuhan pengguna yang berubah-ubah. Penelitian ini khusus membahas bagian mesin ekstraksi yang dapat mengekstrak tabel dari halaman web dengan memanfaatkan dokumen HTML halaman tersebut. Mesin ekstraksi dikembangkan menggunakan pendekatan service oriented architecture (SOA) sehingga pengguna dapat menggunakan service yang disediakan oleh mesin untuk melakukan ekstraksi tabel HTML dengan beraneka ragam layout.

3. METODOLOGI

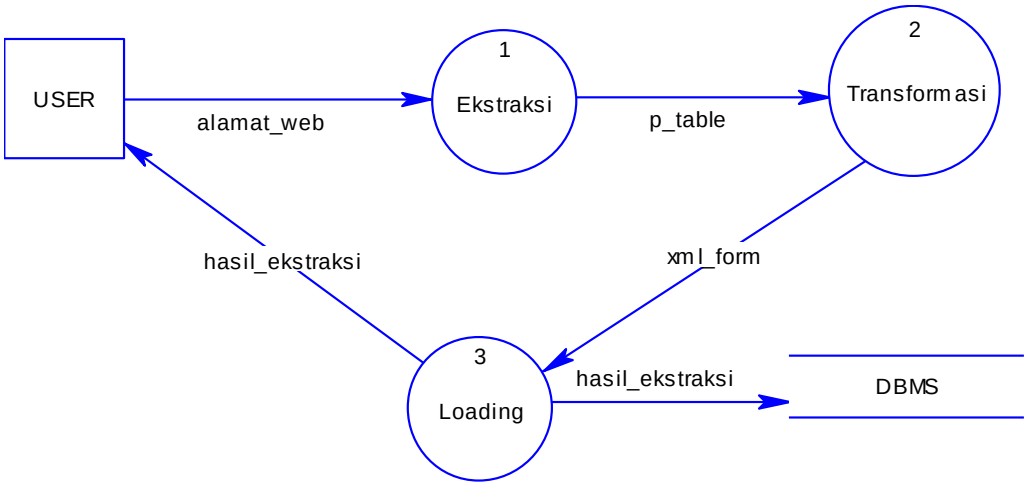
Dalam penyusunan penelitian ini digunakan tahapan umum sebagai berikut :

1. Eksplorasi dan studi literatur
   Dilakukan dengan mempelajari literatur-literatur yang mendukung pelaksanaan penelitian dalam bentuk buku *(textbook)*, jurnal, artikel ilmiah maupun website. Materi yang dipelajari terkait dengan *Enterprise Application Integration, Service Oriented Architecture,* dan *Data Integration*
2. Analisis karakteristik & permasalahan
   Analisis penyelesaian masalah dilakukan dengan menganalisis karakteristik dari *EAI, SOA,* dan *data integration.* Selanjutnya, karakteristik serta permasalahan yang akan dijadikan studi kasus juga akan dianalisa.
3. Perancangan dan implementasi model integrasi
   Berdasarkan karakteristik yang didapatkan pada masing-masing komponen, model integrasi menggunakan *EAI* dan *SOA* akan dirancang dan diimplementasikan.
4. Pengujian model
   Pengujian model akan diuji dengan cara mengimplementasikan pada studi kasus, dalam hal ini adalah integrasi Dokumen Pengembangan Karir Dosen di Indonesia.

Bagian berikut akan membahas satu per satu tahapan yang dilakukan.

3.1 Perancangan Mesin Ekstraksi

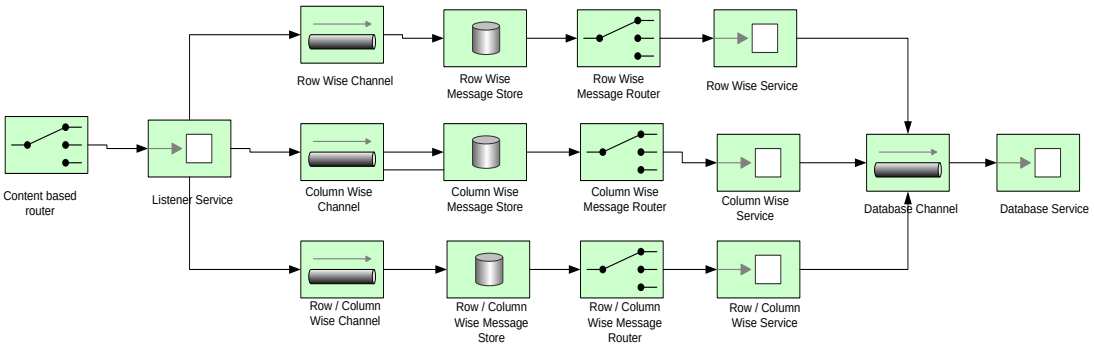
Gambar 4 merupakan perancangan DFD Level 1 aplikasi ekstrakstor yang dikembangkan. Terdapat 3 proses utama yang dapat dilakukan aplikasi ini, yaitu proses ekstraksi, proses transformasi, dan proses loading. User sebagai entitas eksternal aplikasi ini, memasukkan alamat url sebuah website. Tabel pada alamat url yang dimasukkan kemudian diekstraksi dan diubah menjadi bentuk tabel standar yang dinamakan p-table. P-table adalah tabel dalam bentuk column wise yang mana atribut terdapat pada satu baris pertama dari tabel ini. P-table kemudian ditransformasi menjadi bentuk XML sebagai input pada proses loading ke datastore DBMS. Bentuk XML dijadikan skema perantara yang akan dikenali pada service yang disediakan.



Gambar 4. *Data Flow Diagram* Level 1 Mesin Ekstraksi Tabel HTML

3.2 Perancangan Service yang Disediakan

Gambar 5 merupakan arsitektur dari service yang akan digunakan.



Gambar 5. Enterprise Integration Protocol

Gambar 5 merupakan perancangan protokol pada server yang mengatur aliran data dari service ekstraksi sampai dengan service loading ke database management system (DBMS). Pada service loading, aplikasi menyediakan service untuk loading data hasil ekstraksi sesuai jenis layout masing-masing tabel. Terdapat 3 service loading yang disediakan, yakni service loading untuk tabel column wise, service loading untuk tabel row wise, dan service loading untuk tabel column-row wise. Jenis layout didapatkan dari Listener Service, yang mengenali jenis layout tabel yang dimasukkan oleh pengguna.

4. HASIL DAN PEMBAHASAN

4.1 Service Ekstraksi

Sebelum digabungkan dengan aplikasi service yang disediakan, pada bagian ini akan dijelaskan hasil aplikasi tabel ekstraktor.

Yang menjadi input pada aplikasi ini adalah alamat url website yang akan diekstrak tabelnya. Kemudian aplikasi akan mengeluarkan output berupa jenis layout tabel dan bentuk XML dari tabel tersebut. Jenis tabel inilah yang kemudian menentukan lokasi penyimpanan pada DBMS.

Gambar 6. Tampilan Awal Aplikasi

4.2 Service Loading

Service loading yang dimaksudkan pada bagian ini adalah service yang disediakan pada aplikasi untuk menyimpan beragam bentuk layout tabel ke dalam tabel pada database sesuai dengan bentuk layout tabel yang diekstrak. Setelah aplikasi mengenali bentuk layout tabel, service akan memigrasikan bagian data ke dalam database. Sebuah tabel dengan layout column wise atau row wise, masing-masing bermigrasi menjadi sebuah tabel dalam database yang berbentuk column wise. Sedangkan tabel dengan layout column-row wise bermigrasi menjadi tiga tabel yang merupakan kombinasi dari masing-masing atribut pada tabel. Sebagai ilustrasi, jika tabel 2 pada bagian sebelum ini dimigrasikan ke dalam database, maka tabel yang terbentuk seperti pada tabel 3.

Tabel 3. Hasil Migrasi Tabel Column-row Wise pada Tabel 2

Atribut 1
dept1
dept2

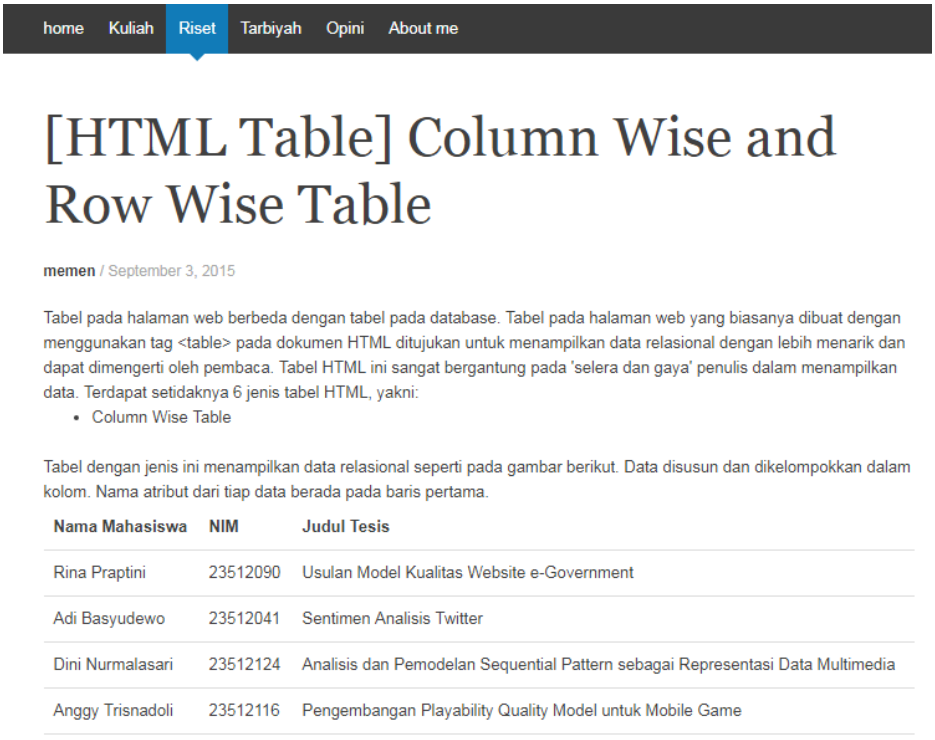
Atribut 2
Q1
Q2
Q3
Q4

Atribut 1	Atribut 2	Atribut 3
dept1	Q1	1.30
dept1	Q2	1.32
dept1	Q3	1.30
dept1	Q4	1.35
dept2	Q1	1.40
dept2	Q2	1.35
dept2	Q3	1.15
dept2	Q4	1.20

4.3 Pengujian Sistem

Gambar 7 merupakan halaman web yang di dalamnya terdapat dua tabel dengan layout yang berbeda. Layout pertama berbentuk column wise dan layout kedua berbentuk row wise.

Dengan memasukkan URL dari halaman web, aplikasi berhasil mengekstrak tabel pertama yang berbentuk column wise. Tabel ini kemudian disimpan dalam database seperti terlihat pada gambar 8.



Gambar 7. Halaman Web Pengujian 1

Gambar 8. Hasil Ekstraksi yang Tersimpan di Database

5. KESIMPULAN DAN SARAN

Berdasarkan pengujian dan analisis yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Terdapat 4 service yang disediakan pada aplikasi, yaitu sebuah service untuk ekstraksi sebagai tampilan awal aplikasi dan 3 buah service transformasi bentuk layout tabel HTML.

2. Dengan beberapa batasan sebagai lingkup penerapan, aplikasi dengan pendekatan service oriented architeture ini berhasil mengekstrak tabel dari halaman web dan menyimpannya ke dalam database.

3. Penyempurnaan aplikasi dan pemanfaatan lebih lanjut data tabel yang disimpan menjadi fokus penelitian berikutnya.

Daftar Pustaka

[1] Thomas Erl, *Service-Oriented Architecture:Comcepts, Technology, and Design*.: Prentice Hall PTR, 2005.

[2] Bogdan Ghlicic Michu, Marian Stoica, and Marinela Mircea, "SOA, SoBI & EDA – Paradigms for Integration Capabilities of BI Platform," *Revista Informatica Economică nr. 2*, 2008.

[3] Yeon-Seok Kim and Kyong-Ho Lee, "Extracting logical structures from HTML tables," *Computer Standards and Interfaces (Elsevier)*, vol. 30, no. 5, pp. 296-308, August 2007.

[4] Chen Kerui, Zhao Jinchao, Zuo Wanli, He Fengling, and Chen Yongheng, "Automatic table integration by domain-specific ontology," *International Journal ofDigital Content Technology and Its Application*, vol. 5, no. 1, pp. 218-226, January 2011.

[5] Goldstone, "Enterprise Application Integration - An Overview,".

[6] Florence Lin, "Enterprise Application Integration (EAI) Techniques," 2005.

[7] Michael Havey, *Modeling Orchestration and Choreography in Service Oriented Architecture*.: Packt Publishing, 2008.

[8] Gregor Hohpe and Boby Woolf, *Enterprise Integration Pattern : Designing, Building, and Deploying Messaging Solution*.: Addison Wesley, 2003.

[9] Mike Rosen, *Orchestration or Choreography?*: Wilton Consulting Group, 2008.