Ekstraksi Data pada Tabel dari Halaman *Web* Menggunakan Pohon *Document Object Model*

Memen Akbar[1], Cici Patmala[2], Dini Nurmalasari[3]

***Abstract*— Data on the web page can be available in various formats, such as table. With the growing of web pages, the need to extract data from tables is increasing. Results of the extraction can be used for integration with other web tables or stored in a database. This study discusses the extraction of data from a table on a web page using a Document Object Model (DOM) tree. The initial step of this extraction process is to transform the HTML document into a DOM tree. Then, by applying search methods Depth First Search (DFS), part of the data in the table is extracted and stored in a CSV file. An engine has been developed using Visual Basic. The results show that the engine can automatically extract data from the table that has the following characteristics: the number of rows and columns are not limited, able to handle all of the table orientation layout, and able to handle tables that are merged cells.**

***Intisari*— Data pada halaman *web* dapat tersedia dalam berbagai format, salah satunya adalah tabel. Pada tabel, informasi ditampilkan dalam baris dan kolom. Tabel pada halaman *web* ditampilkan dengan menggunakan *tag* HTML. Dengan semakin banyaknya halaman *web*, kebutuhan untuk mengekstrak data dari tabel tersebut semakin meningkat. Hasil ekstraksi dapat digunakan untuk integrasi dengan tabel *web* lainnya atau untuk disimpan dalam suatu basis data. Makalah ini membahas ekstraksi data dari tabel pada halaman *web* dengan menggunakan pohon *Document Object Model* (DOM). Langkah awal proses ekstraksi ini adalah mentransformasi dokumen HTML menjadi pohon DOM. Kemudian, dengan menerapkan metode penelusuran *Depth First Search* (DFS), bagian data pada tabel diekstrak dan disimpan ke dalam *file* CSV. Sebuah aplikasi telah dikembangkan dengan bahasa Visual Basic. Hasil pengujian terhadap sejumlah halaman *web* menunjukkan bahwa aplikasi secara otomatis dapat meng-ekstrak data dari tabel yang memiliki karakteristik: jumlah baris dan kolom yang tidak terbatas, semua orientasi *layout* tabel, serta dapat menangani tabel yang terdapat *merged cell*.**

***Kata Kunci*— Ekstraksi tabel, DOM, *Website*, HTML, DFS**

I. PENDAHULUAN

Saat ini, halaman *web* merupakan sumber informasi dengan volume yang sangat besar. Sebuah halaman *web* dapat terdiri atas teks, gambar, daftar (*list*), dan tabel. Tabel merupakan salah satu cara yang digunakan untuk menampilkan data dalam bentuk baris dan kolom. Tabel menampilkan ringkasan data dari isi halaman *web*. Ekstraksi data pada tabel menjadi penting untuk dapat melihat keseluruhan isi halaman *web*. Selain itu, ekstraksi tabel juga penting sebagai langkah awal dalam proses integrasi data tabel dari beberapa halaman *web* [1].

Tabel pada halaman *web* ditampilkan dengan menggunakan *tag-tag* Hypertext Markup Language (HTML). Oleh karena itu, untuk melakukan ekstraksi tabel dari halaman HTML dibutuhkan langkah-langkah yang dapat mempelajari dan menelusuri *tag-tag* penyusun sebuah halaman *web*, khususnya tabel. Makalah ini membahas tentang ekstraksi tabel pada halaman *web* dengan menelusuri struktur *tag* HTML yang membangun sebuah tabel.

Pada makalah ini, *tag* penyusun sebuah tabel terlebih dahulu dimodelkan dalam pohon *Document Object Model* (DOM). Pada struktur pohon DOM ini, bagian data terletak pada daun paling ujung dari setiap ranting. Struktur data dalam bentuk pohon DOM ini kemudian ditelusuri untuk mendapatkan bagian data pada tabel. Penelusuran pohon yang digunakan adalah *Depth First Search* (DFS). Hasil ekstraksi kemudian disimpan dalam *file* berbentuk csv.

Pada bagian selanjutnya, diuraikan beberapa penelitian terdahulu yang membahas mengenai ekstraksi data dari tabel pada halaman *web*. Bagian III membahas perancangan aplikasi untuk menerapkan teknik yang digunakan. Kemudian, bagian IV membahas pengujian aplikasi terhadap beberapa tabel dari sejumlah halaman *web*.

II. EKSTRAKSI DATA DARI HALAMAN *WEB*

Penelitian mengenai ekstraksi data dari halaman *web* telah banyak dilakukan. Sebuah penelitian melakukan ekstraksi data yang ditampilkan dalam bentuk *list* dan tabel pada sebuah halaman *web* [2]. Penelitian ini menerapkan *unsupervised learning algorithms* untuk memperoleh struktur dari *list* dan tabel yang ada di *web* dengan mengacu pada format halaman dan data yang ada di dalamnya. Dalam penelitian ini, dibutuhkan analisis dari beberapa halaman *web* sebelum akhirnya dapat dilakukan ekstraksi data menjadi kolom dan baris. Padahal bisa saja pada halaman *web* hanya ada satu halaman, dan dari satu halaman tersebut terdapat *list* dan tabel yang butuh diekstrak.

Penelitian yang lain mengembangkan metode untuk deteksi dan ekstraksi tabel dengan melakukan analisis *graphic lines* [3]. Tabel sederhana terdiri atas sel-sel matriks. Semua sel dalam satu baris memiliki tinggi yang sama dan semua sel dalam satu kolom memiliki lebar yang sama. Semua sel tersebut dibatasi oleh *graphic lines.* Penelitian ini dapat menjadi salah satu acuan dalam mengembangkan metode ekstraksi tabel dari *file* dalam format PDF.

Telah diteliti pula ekstraksi data pada tabel HTML dengan lima jenis *layout* [4]. Tabel pada halaman *web* tidak memiliki *layout* standar seperti tabel pada basis data. *Layout* tabel pada

[1, 3] *Program Studi Teknik Komputer Politeknik Caltex Riau, Jl. Umbansari 1 Rumbai-Pekanbaru 28265, INDONESIA (email: memen@pcr.ac.id, dini@pcr.ac.id)*

[2] *Program Studi Teknik Informatika Politeknik Caltex Riau, Jl. Umbansari 1 Rumbai-Pekanbaru 28265, INDONESIA (email: cici12ti@mahasiswa.pcr.ac.id)*

halaman *web* sangat bergantung pada kebutuhan dan selera *programmer*. Penelitian ini berkontribusi besar dalam ekstraksi kelima jenis *layout* yang ditentukan.

Pada penelitian lain, ekstraksi merupakan bagian dari proses integrasi data dari beberapa halaman *web* [1], [5]-[8]. Dalam proses ekstraksi, penelitian-penelitian ini menggunakan pendekatan pengenalan *tag* HTML pembentuk sebuah tabel. Karena kebutuhan hasil ekstraksi adalah untuk integrasi data dengan tabel lain, proses ekstraksi dilakukan hingga dapat memisahkan bagian atribut dan bagian data dari tabel. Hasil ekstraksi diubah menjadi pohon semantik [1] dan bentuk XML [5]-[8].

Penelitian lain menerapkan DOM dalam proses ekstraksi konten dari sebuah halaman *web* [9]. Penelitian ini berhasil menghilangkan iklan, gambar, dan *link* yang tidak berhubungan dengan isi *web*.

Telah diterapkan pula algoritme berbasis aturan untuk mengekstrak informasi dari laporan audit pada pernyataan keuangan pemerintahan lokal [10].

Dibandingkan dengan penelitian lain, pada makalah ini dilakukan ekstraksi tabel dari halaman *web* menggunakan metode yang berbeda, yakni menggunakan DOM dalam proses ekstraksi. Pohon DOM merupakan pohon sintaktik untuk melihat struktur dokumen HTML. Penerapan pohon DOM juga memungkinkan ekstraksi untuk kelima jenis orientasi *layout* tabel [4]. Keluaran hasil ekstraksi akan disimpan dalam *file* csv. Bentuk ini dapat digunakan untuk proses berikutnya, untuk disimpan ke basis data maupun diintegrasikan dengan tabel lainnya.

III. HTML, DOM, DAN DFS

A. Hypertext Markup Language (HTML)

HTML adalah sekumpulan simbol-simbol atau *tag-tag* yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan alamat pada *web browser* [11]. *Tag-tag* HTML selalu diwakili dengan `<x>` dan diakhiri dengan `</x>`. Setiap dokumen HTML harus diawali dan ditutup dengan *tag* HTML `<HTML>` dan `</HTML>`. *Tag* HTML memberitahu *browser* bahwa yang ada di dalam kedua *tag* tersebut adalah dokumen HTML.

Tabel digunakan untuk menyajikan data dalam bentuk kolom dan baris. Umumnya setiap kolom menunjukkan data yang sejenis, dan setiap baris menunjukkan kelompok data dalam satu kesatuan. Beberapa *tag* HTML dalam pembuatan tabel beserta fungsinya dapat dilihat pada Tabel I [11].

B. Document Object Model (DOM)

DOM merupakan sebuah ketentuan yang dikembangkan oleh W3C untuk berinteraksi dengan objek-objek yang ada di dalam HTML, XML, maupun XHTML. DOM bersifat *cross-platform* dan *language-independent*. DOM dapat digunakan dengan bahasa pemrograman apapun dan pada sistem operasi manapun. Pengembang bahasa pemrograman atau sistem operasi harus mengimplementasikan antarmuka DOM terlebih dahulu sebelum dapat digunakan pada aplikasi.

TABEL I
DAFTAR REFERENSI *TAG* HTML

Nama Tag	Keterangan/Kegunaan
`<table>...</table>`	*Tag* untuk membuat tabel
`<th>...</th>`	*Tag* untuk membuat sebuah sel *header* tabel
`<tr>...</tr>`	*Tag* untuk membuat baris dalam sebuah tabel
`<td>...</td>`	*Tag* untuk membuat sel dalam sebuah tabel
`<thead>...</thead>`	Mengelompokkan isi *header* dalam sebuah tabel
`<tbody>...</tbody>`	Mengelompokkan isi tubuh dalam sebuah tabel
`<tfoot>...</tfoot>`	Mengelompokkan isi *footer* dalam sebuah tabel

Sebuah dokumen HTML direpresentasikan oleh DOM dalam bentuk struktur hirarki pohon. Dengan kata lain, cara DOM melihat dokumen HTML seperti struktur hirarki pohon.

C. Depth First Search (DFS)

DFS adalah salah satu algoritme penelusuran struktur graf/pohon berdasarkan kedalaman. Simpul ditelusuri dari *root* kemudian ke salah satu simpul anaknya (misalnya prioritas penelusuran berdasarkan anak pertama [simpul sebelah kiri]), maka penelusuran dilakukan terus melalui simpul anak pertama dari simpul anak pertama level sebelumnya hingga mencapai level terdalam [12].

Setelah sampai di level terdalam, penelusuran akan kembali ke satu level sebelumnya untuk menelusuri simpul anak kedua pada pohon biner [simpul sebelah kanan] lalu kembali ke langkah sebelumnya dengan menelusuri simpul anak pertama lagi sampai level terdalam dan seterusnya. Dalam pencarian menggunakan algoritme DFS, simpul-simpul yang paling dalam pada pohon yang akan dicari paling awal, simpul yang telah dikunjungi disimpan dalam suatu tumpukan (*stack*).

Algoritme penelusuran pohon menggunakan DFS dapat dijelaskan sebagai berikut.

1. Memasukkan simpul ujung (akar) ke dalam tumpukan.
2. Mengambil simpul dari tumpukan teratas, lalu memeriksa apakah simpul merupakan solusi
3. Jika simpul merupakan solusi, pencarian selesai dan hasil dikembalikan.
4. Jika simpul bukan solusi, seluruh simpul yang bertetangga dengan simpul tersebut (simpul anak) dimasukkan ke dalam tumpukan.
5. Jika tumpukan kosong dan setiap simpul sudah diperiksa, pencarian selesai dan hasil solusi tidak ditemukan dikembalikan.
6. Mengulangi pencarian dari langkah kedua.

IV. PERANCANGAN DAN ANALISIS

Bagian ini menjabarkan perancangan yang digunakan untuk membuat aplikasi ekstraksi data tabel dari halaman *web*. Terdapat dua perancangan yang digunakan, yaitu diagram blok dan diagram *use case*.

A. *Diagram Blok*

Gbr. 1 merupakan diagram blok proses ekstraksi data dari sebuah tabel pada halaman *web*. Yang menjadi *input* pada proses ini adalah alamat *URL* atau kode HTML halaman *web* yang akan diekstraksi. Langkah awal yang dilakukan adalah mengubah dokumen HTML dari halaman *web* ke dalam bentuk teks. Pengolahan data selanjutnya menggunakan pengolahan variabel yang berbentuk *string*.

Gbr. 1 Diagram blok.

Pada sebuah halaman *web* terdapat beberapa bagian yang dapat berupa teks, gambar, *list*, atau tabel. Oleh karena itu, langkah pertama yang dilakukan adalah menentukan bagian yang merupakan tabel dari dokumen HTML. Tabel dideteksi dengan memanfaatkan fungsi pencocokan *string*. Tabel pada dokumen HTML dapat dikenali dengan *tag* `<table>` pada bagian awal dan `<\table>` pada bagian akhir tabel. Jika pada sebuah halaman *web* terdapat dua atau lebih tabel, aplikasi menyimpan tabel tersebut pada indeks kedua dari variabel yang berbentuk *array*.

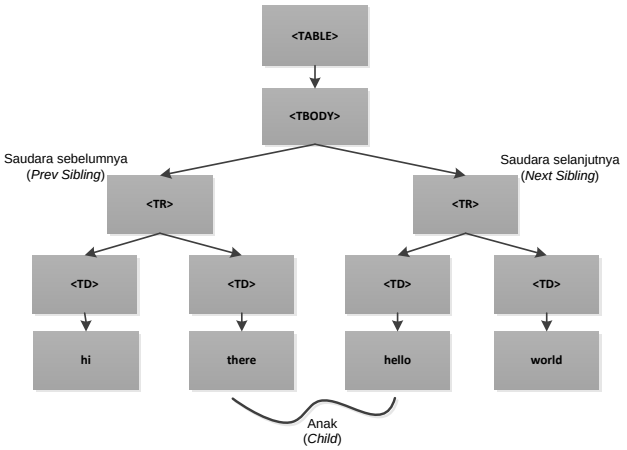
Langkah selanjutnya adalah membuat pohon DOM dari tabel yang dideteksi. Pohon DOM disusun dari *tag* pembentuk tabel, yaitu *tag* `<table>`, `<tr>`, `<th>`, dan `<td>`. Keempat *tag* ini dideteksi untuk menentukan bagian data pada tabel. Bagian data terletak di luar *tag* `<td>` atau `<th>`. Pada pohon DOM, bagian data ini merupakan daun dengan kedalaman paling rendah dari semua ranting. Sebagai contoh, Gbr. 2 merupakan pohon DOM dari Tabel II. Tampak bahwa data tabel berada pada daun terendah dari setiap ranting.

Langkah ketiga adalah mengekstrak atau mengambil bagian data dari pohon DOM. Untuk menelusuri daun terbawah dari pohon DOM, metode penelusuran yang digunakan adalah DFS. Metode ini menelusuri pohon diawali dari akar kemudian dilanjutkan ke daun sebelah kiri di bawahnya hingga ditemukan daun terbawah dari ranting paling kiri. Kemudian penelusuran dilanjutkan ke ranting di sebelahnya hingga ditemukan daun terbawah dari ranting tersebut. Penelusuran seperti ini dilakukan berulang kali hingga penelusuran mencapai daun terendah pada ranting paling kanan. Algoritme yang digunakan adalah algoritme DFS yang telah dijelaskan pada bagian II.

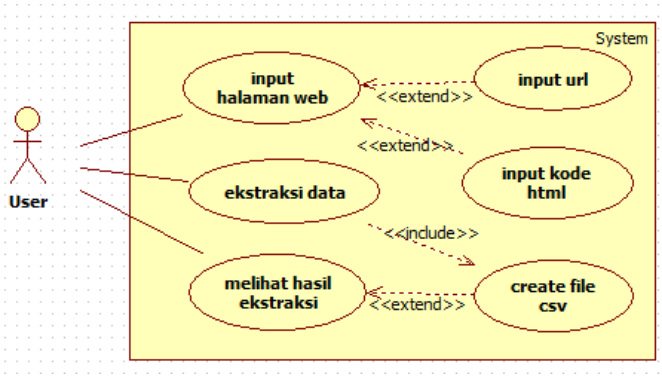
Setelah bagian data pada tabel diperoleh, langkah berikutnya adalah menyimpan data dalam bentuk *file* csv. Pemilihan bentuk *file* penyimpanan ditujukan agar hasil ekstraksi dapat digunakan untuk kebutuhan berikutnya, seperti integrasi dengan data dari tabel lain atau untuk disimpan dalam sebuah basis data.

TABEL II
CONTOH TABEL

hi	There
hello	World



Gbr. 2 Pohon DOM dari Tabel II.



Gbr. 3 Diagram *use case*.

B. *Diagram Use Case*

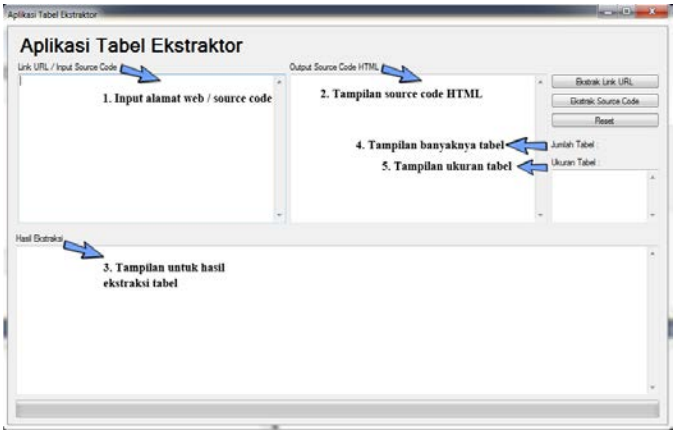
Perancangan diagram *use case* ditujukan untuk menggambarkan fungionalitas aplikasi yang dibangun. Terdapat seorang aktor secara umum yang dinamakan dengan *user*. Selain itu, terdapat tiga fungsionalitas yang dapat dilakukan oleh *user* terhadap aplikasi. Yang pertama, *user* dapat memasukkan halaman *web* dalam bentuk alamat *URL* atau kode HTML dari halaman *web*. Dengan inisialisasi dari *user*, sistem dapat melakukan ekstraksi data dengan proses yang telah dijelaskan pada bagian sebelumnya. Selain itu, *user* juga dapat melihat data hasil ekstraksi langsung pada aplikasi atau pada *file* csv yang telah disimpan. Diagram *use case* dari aplikasi dapat dilihat pada Gbr. 3.

V. HASIL DAN PENGUJIAN

A. *Tabel Ekstraktor*

Aplikasi tabel ekstraktor ini dikembangkan dengan menggunakan bahasa pemrograman Visual Basic. Masing-masing fungsionalitas aplikasi yang terdapat pada diagram *use*

case diimplementasikan menjadi beberapa *method*. Ada *method* untuk mengolah *input* hingga diperoleh bagian tabel yang akan diekstrak, ada pula *method* untuk mengubah kode HTML menjadi pohon DOM yang kemudian ditelusuri dengan menggunakan *method* DFS. Kemudian ada juga *method* untuk menampilkan hasil ekstraksi dan mengubahnya menjadi *file* csv.



Gbr. 4 Tampilan aplikasi.

Gbr. 4 merupakan tampilan aplikasi yang dihasilkan. Terdapat satu bagian dari aplikasi untuk memasukkan alamat *URL* atau kode HTML halaman *web* yang ingin diekstrak. Setelah menekan tombol Ekstrak *Link URL* atau Ekstrak *Source code*, aplikasi akan menampilkan kembali *source code* HTML pada bagian Output *Source code* HTML, kemudian menampilkan hasil ekstraksi pada bagian Hasil Ekstraksi. Selain ditampilkan pada salah satu bagian pada aplikasi, hasil ekstraksi juga secara otomatis tersimpan dalam *file* csv. Aplikasi dapat berjalan jika terhubung dengan koneksi internet.

B. Pengujian

Pengujian dibagi menjadi dua bagian, yaitu pengujian fungsionalitas aplikasi dan pengujian hasil ekstraksi. Kedua pengujian menerapkan prinsip pengujian *black box*.

Pada pengujian fungsionalitas, prosedur pengujian menggunakan *suitability metrics* pada ISO 9126-2. Terdapat tiga metrik yang digunakan, yaitu:

1. *functional adequacy* (FA);
2. *functional implementation completeness* (FICM); dan
3. *functional implementation coverage* (FIC).

Berdasarkan ketiga metrik tersebut diperoleh hasil bahwa semua kebutuhan pengguna telah terpenuhi dan telah lengkap diimplementasikan.

Pengujian kedua, yakni pengujian untuk melihat kebenaran hasil ekstraksi. Pada pengujian ini, aplikasi tabel ekstraktor diujikan pada beberapa halaman *web* dengan beberapa karakteristik. Karakteristik yang diperhatikan pada pengujian ini antara lain eksistensi dan banyak tabel pada halaman *web*, ukuran tabel, orientasi *layout* tabel, serta ada atau tidaknya *merging cell* pada tabel. Tabel III merupakan rekapitulasi data uji yang digunakan.

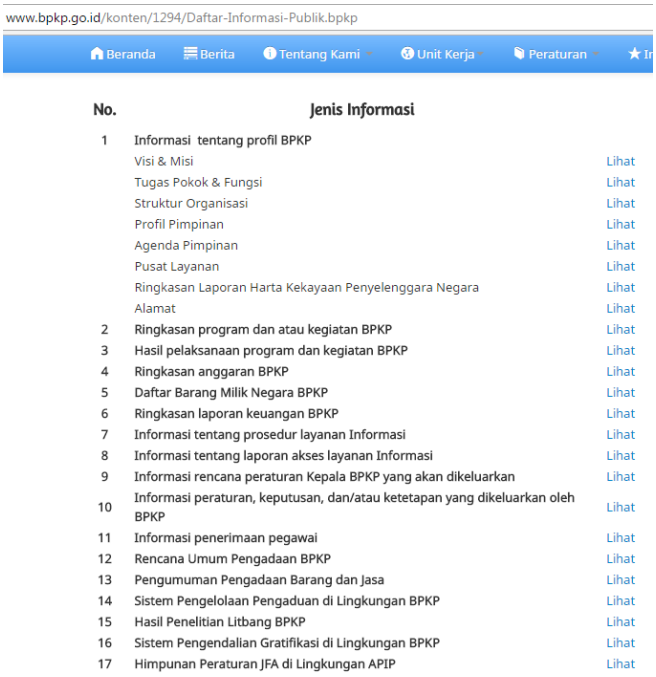
TABEL III
REKAPITULASI DATA PENGUJIAN

No.	Alamat Web	Jumlah Tabel	Ukuran Tabel	Kondisi Tabel
1.	http://www.bpkp.go.id/konten/1294/Daftar-Informasi-Publik.bpkp	1 (satu)	- Baris: 26 - Kolom: 3	- *Column wise Table* - Tidak ada *merging cell*
2.	http://www.bengkaliskab.go.id/statis-93-Bidang-Energi-dan-Sumber-Daya-mineral.html	1 (satu)	- Baris: 2 - Kolom: 3	- *Column wise Table* - Tidak ada *merging cell*
3.	http://diskominfo.riau.go.id/hal-daftar-ppid-provinsi-riau.html	2 (dua)	Tabel 1 - Baris: 58 - Kolom: 4 Tabel 2 - Baris: 13 - Kolom: 4	- *Column wise Table* - Terdapat *merging cell*
4.	https://lpse.lkpp.go.id/eproc4	1 (satu)	- Baris: 18 - Kolom: 4	- *Column wise Table* - Terdapat *merging cell*
5.	http://diskominfo.pemkomedan.go.id/statis-6-sumberdayamanusia.html	4 (empat)	Tabel 1 - Baris: 13 - Kolom: 4 Tabel 2 - Baris: 7 - Kolom: 4 Tabel 3 - Baris: 8 - Kolom: 10 Tabel 4 - Baris: 5 - Kolom: 4	- *Column wise Table* - Tidak ada *merging cell*
6.	http://www.pemkomedan.go.id/hal-kependudukan.html	4 (empat)	Tabel 1 - Baris: 19 - Kolom: 4 Tabel 2 - Baris: 13 - Kolom: 4 Tabel 3 - Baris: 24 - Kolom: 5 Tabel 4 - Baris: 22 - Kolom: 9	- *Column wise Table* - Terdapat *merging cell*
7.	http://diskominfo.riau.go.id/subbidang-3-jarkom.html	1 (satu)	- Baris: 7 - Kolom: 2	- *Row wise Table* - Terdapat *merging cell*

Pada bagian berikut akan dijelaskan beberapa data uji sebagai gambaran hasil pengujian aplikasi tabel ekstraktor.

1) Data Uji 1: Gbr. 5 merupakan tampilan halaman *web* data uji nomor 1 yang diakses pada 11 September 2016. Terdapat sebuah tabel yang berukuran 26 x 3 dengan l*ayout column wise*. Pada tabel tidak terdapat kondisi *merging cell*.

Alamat URL dimasukkan pada aplikasi tabel ekstraktor. Hasil ekstraksi dapat dilihat pada Gbr. 6 dan *file* csv yang dihasilkan dapat dilihat pada Gbr. 7. Aplikasi berhasil mengekstrak data pada tabel dan menyimpannya dalam file csv.

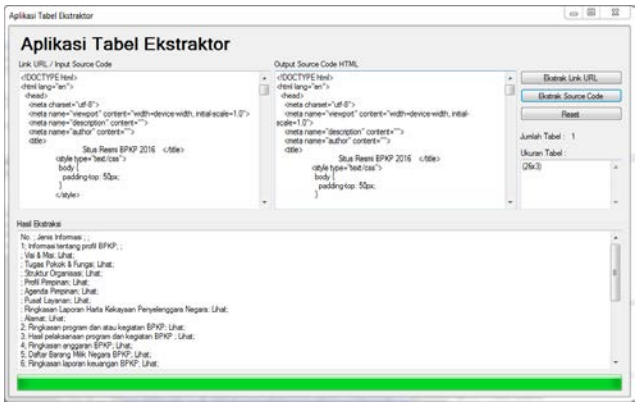


www.bpkp.go.id/konten/1294/Daftar-Informasi-Publik.bpkp

Beranda | Berita | Tentang Kami | Unit Kerja | Peraturan

No.	Jenis Informasi	
1	Informasi tentang profil BPKP	
	Visi & Misi	Lihat
	Tugas Pokok & Fungsi	Lihat
	Struktur Organisasi	Lihat
	Profil Pimpinan	Lihat
	Agenda Pimpinan	Lihat
	Pusat Layanan	Lihat
	Ringkasan Laporan Harta Kekayaan Penyelenggara Negara	Lihat
	Alamat	Lihat
2	Ringkasan program dan atau kegiatan BPKP	Lihat
3	Hasil pelaksanaan program dan kegiatan BPKP	Lihat
4	Ringkasan anggaran BPKP	Lihat
5	Daftar Barang Milik Negara BPKP	Lihat
6	Ringkasan laporan keuangan BPKP	Lihat
7	Informasi tentang prosedur layanan Informasi	Lihat
8	Informasi tentang laporan akses layanan Informasi	Lihat
9	Informasi rencana peraturan Kepala BPKP yang akan dikeluarkan	Lihat
10	Informasi peraturan, keputusan, dan/atau ketetapan yang dikeluarkan oleh BPKP	Lihat
11	Informasi penerimaan pegawai	Lihat
12	Rencana Umum Pengadaan BPKP	Lihat
13	Pengumuman Pengadaan Barang dan Jasa	Lihat
14	Sistem Pengelolaan Pengaduan di Lingkungan BPKP	Lihat
15	Hasil Penelitian Litbang BPKP	Lihat
16	Sistem Pengendalian Gratifikasi di Lingkungan BPKP	Lihat
17	Himpunan Peraturan JFA di Lingkungan APIP	Lihat

Gbr. 5 Tampilan halaman *web* data uji 1.

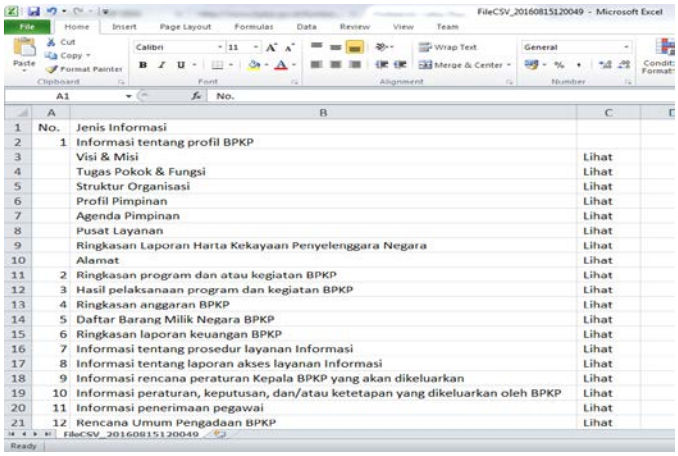


Gbr. 6 Hasil ekstraksi tabel pada data uji 1.

2) Data Uji 2: Gbr. 8 merupakan tampilan halaman *web* pada data uji nomor 3 yang diakses pada tanggal 11 September 2016. Terdapat dua buah tabel pada halaman *web* dengan ukuran masing-masing 58 x 4 dan 13 x 4. Orientasi *layout* kedua tabel adalah *column wise*. Pada kedua tabel terdapat *cell* yang merupakan *merging* dari beberapa *cell*.

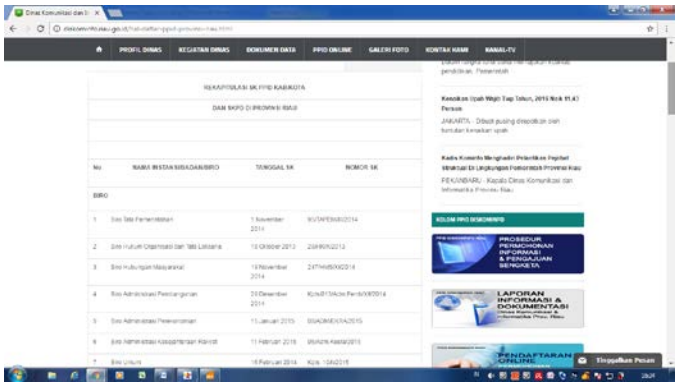
Gbr. 9 merupakan tampilan *file* csv hasil ekstraksi tabel pada data uji 2. Tampak bahwa aplikasi tabel ekstraktor berhasil mengekstrak data pada kedua tabel dan menyimpannya ke dalam *file* csv.

3) Data Uji 3: Gbr. 10 merupakan tampilan halaman *web* pada data uji nomor 7 yang diakses pada tanggal 11 September 2016. Pada halaman *web* terdapat sebuah tabel yang berukuran 7 x 2 dengan orientasi *layout row wise*. Pada tabel juga terdapat kondisi *cell* yang merupakan *merging* dari beberapa *cell*.

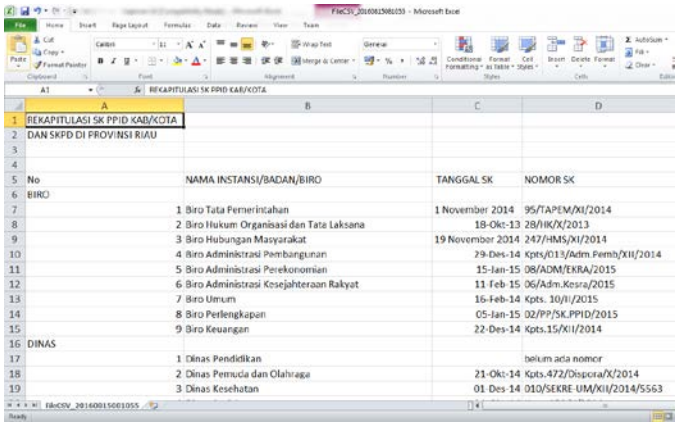


	A	B	C
1	No.	Jenis Informasi	
2	1	Informasi tentang profil BPKP	
3		Visi & Misi	Lihat
4		Tugas Pokok & Fungsi	Lihat
5		Struktur Organisasi	Lihat
6		Profil Pimpinan	Lihat
7		Agenda Pimpinan	Lihat
8		Pusat Layanan	Lihat
9		Ringkasan Laporan Harta Kekayaan Penyelenggara Negara	Lihat
10		Alamat	Lihat
11	2	Ringkasan program dan atau kegiatan BPKP	Lihat
12	3	Hasil pelaksanaan program dan kegiatan BPKP	Lihat
13	4	Ringkasan anggaran BPKP	Lihat
14	5	Daftar Barang Milik Negara BPKP	Lihat
15	6	Ringkasan laporan keuangan BPKP	Lihat
16	7	Informasi tentang prosedur layanan Informasi	Lihat
17	8	Informasi tentang laporan akses layanan Informasi	Lihat
18	9	Informasi rencana peraturan Kepala BPKP yang akan dikeluarkan	Lihat
19	10	Informasi peraturan, keputusan, dan/atau ketetapan yang dikeluarkan oleh BPKP	Lihat
20	11	Informasi penerimaan pegawai	Lihat
21	12	Rencana Umum Pengadaan BPKP	Lihat

Gbr. 7 Tampilan *file* csv data uji 1.



Gbr. 8 Tampilan halaman *web* data uji 2.



	A	B	C	D
1	REKAPITULASI SK PPID KAB/KOTA			
2	DAN SKPD DI PROVINSI RIAU			
3				
4				
5	No	NAMA INSTANSI/BADAN/BIRO	TANGGAL SK	NOMOR SK
6	BIRO			
7	1	Biro Tata Pemerintahan	1 November 2014	95/TAPEM/XI/2014
8	2	Biro Hukum Organisasi dan Tata Laksana	18-Okt-13	28/HK/X/2013
9	3	Biro Hubungan Masyarakat	19 November 2014	247/HMS/XI/2014
10	4	Biro Administrasi Pembangunan	29-Des-14	Kpts/013/Adm.Pemb/XII/2014
11	5	Biro Administrasi Perekonomian	15-Jan-15	08/ADM/EKRA/2015
12	6	Biro Administrasi Kesejahteraan Rakyat	11 Feb-15	06/Adm.Kesra/2015
13	7	Biro Umum	16-Feb-14	Kpts. 10/II/2015
14	8	Biro Perlengkapan	05-Jan-15	02/PP/SK.PPID/2015
15	9	Biro Keuangan	22-Des-14	Kpts.15/XII/2014
16	DINAS			
17	1	Dinas Pendidikan		belum ada nomor
18	2	Dinas Pemuda dan Olahraga	21-Okt-14	Kpts.472/Dispora/X/2014
19	3	Dinas Kesehatan	01 Des-14	010/SEKRE-UM/XII/2014/5563

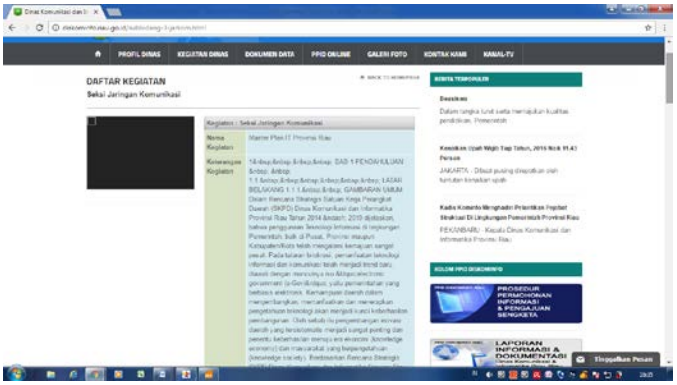
Gbr. 9 *File* csv hasil ekstraksi data uji 2.

Gbr. 11 merupakan tampilan *file* csv hasil ekstraksi data uji 3. Tampak bahwa aplikasi tabel ekstraktor berhasil mengekstrak data pada tabel dan menyimpannya dalam *file* csv.

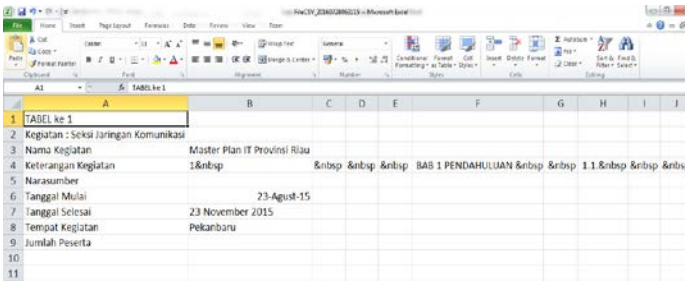
C. Analisis Hasil Pengujian

Berdasarkan hasil pengujian fungsionalitas aplikasi, implementasi fitur pada aplikasi tabel ekstraktor telah sesuai

dengan fungsinya. Hal ini menunjukkan bahwa fitur pada aplikasi berfungsi dengan baik sesuai kegunaannya. Setiap operasi pada sistem dapat berfungsi sesuai dengan yang diharapkan. Operasi pada sistem dapat digunakan untuk proses lebih lanjut, seperti mengambil bagian yang dianggap penting sesuai dengan input alamat *web* yang mengandung bentuk tabel.



Gbr. 10 Tampilan halaman *web* data uji 3.



	A	B
1	TABEL ke 1	
2	Kegiatan : Seksi Jaringan Komunikasi	
3	Nama Kegiatan	Master Plan IT Provinsi Riau
4	Keterangan Kegiatan	1
5	Narasumber	
6	Tanggal Mulai	23-Agust-15
7	Tanggal Selesai	23 November 2015
8	Tempat Kegiatan	Pekanbaru
9	Jumlah Peserta	
10		
11		

Gbr. 11 Tampilan *file* csv hasil ekstraksi tabel pada data uji 3.

Berdasarkan hasil pengujian keluaran hasil ekstraksi, aplikasi berhasil mengekstrak data dari satu atau beberapa tabel yang terdapat pada halaman *web*. Aplikasi juga berhasil mengekstrak data dengan beberapa karakteristik tabel sebagai berikut.

1) Orientasi Layout Column Wise dan Row Wise: Meskipun orientasi *layout* yang diujikan hanya dua dari lima *layout* yang disampaikan pada makalah acuan [4], aplikasi ini diyakini dapat melakukan ekstraksi untuk ketiga jenis *layout* lainnya. Hal ini diyakini karena implementasi proses ekstraksi yang digunakan hanya menerapkan pembacaan *tag* pembangun sebuah tabel pada dokumen HTML. Proses ekstraksi tidak memperhatikan kondisi *layout* tabel pada halaman *web*. Mengenali kondisi *layout* tabel dibutuhkan untuk memisahkan bagian atribut dan bagian data dari tabel. Pemisahan bagian atribut dan data pada tabel berguna untuk proses integrasi dengan tabel lainnya [1]. Namun untuk integrasi suatu tabel pada halaman *web* dengan tabel lain, deteksi bagian atribut dan bagian data tidak diperlukan karena pada aplikasi yang dikembangkan ditambahkan sebuah pengetahuan yang dapat mendeteksi nama atribut dari suatu kolom tanpa harus mengetahui nama atribut yang tertulis pada tabel [5],. Namun, hal ini tidak berlaku untuk tabel dengan orientasi *layout* yang tidak terstruktur [1], [4].

Aplikasi yang dikembangkan saat ini belum dilengkapi dengan pengetahuan untuk pengenalan atribut dan data pada tabel.

2) Terdapat Merging Cell pada Tabel: Aplikasi berhasil mengekstrak data dari tabel dengan kondisi terdapat *merging cell*. Aplikasi tabel ekstrakstor ini menganggap bahwa *cell* yang merupakan *merging* dari beberapa *cell* merupakan satu data. Hal ini dapat terjadi karena proses ekstraksi tidak membaca atribut yang terdapat di dalam sebuah *tag*. Pada *tag* HTML, *merging cell* ditandai dengan nilai *colspan* atau *rowspan* yang lebih besar dari 1. Mengenali *cell* dengan kondisi *merging* dibutuhkan untuk menentukan ukuran tabel. Aplikasi ini telah berhasil menentukan ukuran tabel meskipun tanpa mengenali *cell* dengan kondisi *merging*.

Pengenalan ukuran tabel pada aplikasi ini dihitung berdasarkan jumlah *tag* `<td>` dan `<tr>` pada dokumen HTML. Ukuran baris dapat diketahui dari jumlah *tag* `<tr>` dalam suatu rentang *tag* `<table>` dan ukuran kolom diperoleh dari jumlah *tag* `<td>` dalam suatu rentang *tag* `<tr>`.

VI. KESIMPULAN

Berdasarkan hasil pengujian dan analisis terhadap pengujian yang dilakukan, dapat diambil beberapa kesimpulan sebagai berikut. Proses ekstraksi tabel dengan menggunakan pohon DOM dan metode DFS untuk penelusurannya telah berhasil diimplementasikan pada aplikasi tabel ekstraktor ini. Aplikasi yang dikembangkan dapat mengekstrak data dari satu atau beberapa tabel pada halaman *web* dengan orientasi *layout column wise*, *row wise*, *column-row wise*, dan *layout* yang tidak terstruktur. Aplikasi juga berhasil melakukan ekstraksi terhadap tabel yang di dalamnya terdapat *merging cell*. Selain itu, aplikasi juga berhasil menghitung ukuran tabel yang diekstrak. Selanjutnya, aplikasi dapat menerima *input* halaman *web* dalam bentuk alamat *URL* atau kode HTML. Penambahan *input* berupa kode HTML diharapkan menambah alternatif *input* untuk melakukan ekstraksi tabel.

REFERENSI

[1] Memen Akbar, Fazat Nur Azizah, and G. A. Putri Saptawati, "Pengembangan Engine Integrasi Tabel HTML pada Halaman Web," *JNTETI*, vol. 5, no. 3, pp. 177-183, Agustus 2016.

[2] Kristina Lerman, Craig Knoblock, and Steven Minton, "Automatic data extraction from lists and tables in web sources," *IJCAI-2001 Workshop on Adaptive Adaptive Text Extraction and Mining*, vol. 98, August 2001.

[3] J. Y. Ramel, Michel Crucianu, Nicole Vincent, and Claudie Faure, "Detection, extraction and representation of tables," in *Document Analysis and Recognition*, 2003, pp. 374-378.

[4] 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.

[5] 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.

[6] David W. Embley, Cui Tao, and Stephen W. Liddle, "Automating the extraction of data from HTML tables with unknown structure," *Data & Knowledge Engineering (Elsevier)*, vol. 54, pp. 3-28, November 2004.

[7] Shijun Li, Zhiyong Peng, and Mengchi Liu, "Extraction and integration information in HTML tables," in *Fourth International Conference on Computer and Information Technology (CIT)*, 2004.

[8] Eko Prasetyo, Lukito Edi Nugroho, and Marcus Nurtiantara Aji, "Perancangan Data Warehouse Sistem Informasi Eksekutif untuk Data Akademik Program Studi," *JNTETI*, vol. 1, no. 3, pp. 13-20, November 2012.

[9] Suhit Gupta, Gail Kaiser, David Neistadt, and Peter Grimm, "DOM-based content extraction of HTML documents," in *12th International Conference on World Wide Web*, 2003, pp. 207-214.

[10] Agny Ismaya, "Algoritma Ekstraksi Informasi Berbasis Aturan," *JNTETI*, vol. 3, no. 4, pp. 242-247, November 2014.

[11] Anhar, *Panduan menguasai PHP & MySQL secara otodidak*. Jakarta, Indonesia: Mediakita, 2010.

[12] Nathanael T. Black and Wolfgang Ertel, *Introduction to artificial intelligence*. Germany: Springer science; Business Media, 2011.