Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.201856905
Vol. 5, No. 6, November 2018, hlm. 653-658 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 51/E/KPT/2017 e-ISSN: 2528-6579

EKSTRAKSI TABEL HTML BENTUK *COLUMN-ROW WISE* KE DALAM BASIS DATA

Memen Akbar[1], Ardianto Wibowo[2]

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

(Naskah masuk: 05 Juli 2018, diterima untuk diterbitkan: 02 November 2018)

Abstrak

Tabel adalah bagian penting pada sebuah halaman web. Tabel memuat tabulasi data atau informasi yang ingin disampaikan dari halaman web tersebut. Tabulasi data ini dapat digunakan untuk kebutuhan perbandingan dengan tabel serupa atau sebagai pencetus untuk melakukan sebuah aksi. Namun, tabel pada halaman web bersifat independen terhadap pembuat halaman web. Tidak ada bentuk atau layout standar sebuah tabel pada halaman web. Salah satu layout tabel pada halaman web *adalah column-row wise*. Penelitian ini menawarkan pendekatan untuk mengekstraksi isi tabel sedemikian sehingga arti dari keterkaitan antara dua atribut dan data dalam tabel *column-row wise* tidak hilang. Data yang diekstrak disimpan ke dalam basis data yang membentuk tiga tabel, yaitu tabel yang menyimpan atribut pertama, tabel yang menyimpan atribut kedua, dan tabel yang menyimpan atribut pertama, kedua, dan data dari atribut pertama dan kedua. Penelitian ini menghasilkan sebuah algoritma untuk mengekstrak data dari tabel yang berbentuk *column-row wise* pada sebuah halaman web. Algoritma yang dihasilkan dari penelitian ini diharapkan dapat diimplementasikan dalam berbagai bahasa pemrograman. Untuk pengujian, algoritma telah diimplementasikan dengan Bahasa pemrograman Python dan berhasil melakukan ekstraksi tabel dan menyimpannya dalam basis data. Nilai cyclomatic complexity number algoritma yang diusulkan adalah 12. Hal ini berarti, kompleksitas algoritma yang diusulkan masih tinggi.

Kata kunci: *ekstraksi tabel HTML, column-row wise, basis data, halaman web*

EXTRACTION OF COLUMN-ROW WISE HTML TABLE INTO DATABASE

Abstract

Tables are an important part of a web page. The table contains tabulations of data or information that you want to convey from the web page. This data tabulation can be used for comparisons with similar tables or as a trigger for action. However, tables on web pages are independent of webpage makers. There is no standard form or layout for a table on a web page. One of the table layouts on a web page is column-row wise. This study offers an approach for extracting table contents such that the meaning of the linkage between two attributes and a data in the column-row wise table is not disappeared. The extracted data is stored into a database that forms three tables, ie the table that stores the first attribute, the table that stores the second attribute, and the table that stores the first, second, and second attributes of the two attributes. Output of this research is an algorithm to extract data of column-row wise table in a web page. The algorithm generated from this research is expected to be implemented in various programming languages. For testing, the algorithm is implemented in Python and success to extract table and save the data into database. Cyclomatic complexity number of the proposed algorithm is 12. This means that the complexity of the proposed algorithm is still high.

Keywords: *HTML table extraction, column-row wise, database management system, web page*

1. PENDAHULUAN

Sebuah halaman web terdiri dari 3 bagian, yaitu bagian yang berbentuk teks, bagian yang berupa gambar, dan bagian yang berupa tabel. Bagian teks berisikan penjelasan dan cerita mengenai permasalahan yang sedang dibahas pada artikel tersebut. Bagian gambar berisikan ilustrasi sebagai pendukung penjelasan pada bagian teks. Bagian gambar ini juga yang membuat sebuah artikel menarik. Bagian tabel berisikan tabulasi data yang digunakan atau dijelaskan pada halaman web. Pada sebuah halaman web, tabel merupakan bagian penting dari permasalahan yang dijelaskan dalam sebuah artikel. Mengekstrak tabel yang terdapat pada halaman web berarti mengambil inti sari dari halaman web tersebut. Hasil ekstraksi ini dapat digunakan untuk berbagai keperluan, di antaranya untuk membandingkannya dengan tabel lain yang

serupa, seperti untuk perbandingan harga tiket penerbangan dari beberapa halaman web. Hasil ekstraksi ini dapat juga digunakan sebagai pencetus untuk melakukan aksi lain dari website lain, seperti pencetus untuk perhitungan gaji berdasarkan nilai kurs USD yang ditampilkan pada tabel sebuah halaman web.

Sebuah tabel terdiri dari dua bagian, yakni bagian atribut dan bagian data. Bagian atribut merupakan sebuah nama yang menjadi representasi dari data yang dilingkupinya. Tabel 1 merupakan contoh tabel sebagai ilustrasi kedua bagian tersebut. Baris pertama pada tabel yang berisikan data C_1, ..., C_n merupakan bagian atribut dari tabel. Sedangkan, baris kedua dan seterusnya merupakan bagian data dari masing-masing atribut, seperti $a_{1,1}$, ..., $a_{m,1}$ merupakan bagian data dari atribut C_1 dan $a_{1,n}$, ..., $a_{m,n}$ merupakan bagian data dari atribut C_n.

Tabel yang terdapat pada sebuah halaman web berbeda dengan tabel pada basis data. Tabel pada halaman web cenderung tidak memiliki aturan atau standar bentuknya. Jika pada basis data, bagian atribut selalu berada pada baris pertama dan didefinisikan sebelum tabel dibuat. Pada sebuah halaman web, bagian atribut tidak selalu berada pada baris pertama. (Kim & Lee, 2007) mendefinisikan 5 jenis tabel berdasarkan letak atribut terhadap data, yaitu: (1) tabel *colum wise*, (2) tabel *row wise*, (3) tabel *column-row wise*, (4) tabel *composite*, dan (5) tabel *mixed-cell*.

Penelitian mengenai ekstraksi bagian tabel dari sebuah halaman web telah banyak dilakukan. Namun, sedikit sekali yang membahas mengenai ekstraksi tabel yang berbentuk *column-row wise*, di antaranya adalah (Akbar, et al., 2015) dan (Akbar, et al., 2016). Tabel yang berbentuk *column-row wise* diekstrak menjadi sebuah tabel baru dengan mengasumsikan bagian atribut yang terdapat pada baris pertama sebagai atribut dan menambahkan satu nama atribut untuk bagian ini. Model ekstraksi seperti ini menghilangkan makna bagian data pada tabel. Bagian data pada tabel dengan bentuk *column-row wise* merupakan nilai dari kombinasi dua atribut pada baris dan kolomnya. Pada Tabel 2, data 1.30 merupakan nilai kombinasi untuk dept1 dan Q1, data 1.35 merupakan nilai untuk dept1 dan Q2, dan seterusnya. Oleh karena itu, pada penelitian ini ditawarkan sebuah model ekstraksi baru yang tidak mengakibatkan makna data pada tabel yang berbentuk *column-row wise* hilang.

Tabel 1. Contoh Tabel yang Berbentuk Column Wise

C_1	**...**	**C_n**
$a_{1,1}$	...	$a_{1,n}$
...	...	...
$a_{m,1}$	...	$a_{m,n}$

Tabel 2. Contoh Tabel dengan Bentuk Column-Row Wise

	Q1	**Q2**	**Q3**	**Q4**
dept1	1.3	1.32	1.3	1.35
dept2	1.4	1.35	1.15	1.2

STATE OF THE ART

Penelitian yang membahas mengenai ekstraksi bagian tabel pada sebuah halaman web telah banyak dilakukan. (Lim, et al., 2002) melakukan ekstraksi tabel pada halaman web yang berbentuk *colum wise* dan *row wise*. Penelitian ini mengekstrak tabel dengan mengubah tabel menjadi bentuk standar yang berbentuk *column wise*. Sedangkan (Embley, et al., 2004) hanya melakukan ekstraksi untuk tabel yang berbentuk *column wise*. Namun, penelitian ini tidak fokus pada posisi atribut terhadap data.

Penelitian yang dilakukan oleh (Kim & Lee, 2007) fokus pada proses ekstraksi untuk beberapa jenis bentuk tabel. Penelitian ini memberikan cara-cara untuk mengekstraksi tabel dengan kelima bentuk yang telah dijelaskan pada bagian pertama. Untuk tabel dengan bentuk *column-row wise*, penelitian ini memberikan solusi dengan menganggap atribut yang terdapat pada kolom pertama sebagai data dari sebuah atribut yang didefinisikan. Sama halnya yang dilakukan oleh (Akbar, et al., 2015). Mekanisme ekstraksi seperti ini menghilangkan makna data dari tabel yang berbentuk *column-row wise*.

Penelitian yang dilakukan oleh (Kerui, et al., 2011) fokus pada proses ekstraksi untuk tabel yang tidak terstruktur. Posisi atribut dan data pada tabel tidak berbentuk seperti tabulasi. Karena itu, penelitian ini juga fokus pada proses pemberian nama dari kumpulan data yang dilihat kemiripannya.

Dari semua penelitian yang membahas mengenai ekstraksi tabel dari sebuah halaman web, belum ada yang membahas hingga data hasil ekstraksi disimpan ke dalam sebuah basis data. Selain itu, belum ada juga sebuah mekanisme yang melakukan ekstraksi tabel yang berbentuk *column-row wise* sehingga makna data tidak hilang.

Mengekstrak data pada tabel dari halaman web dan kemudian disimpan dalam sebuah basis data dapat dimanfaatkan pada data ware house. Proses ekstraksi ini dapat melengkapi ragam tipe dan jenis dari sumber data pada data ware house. Pengembangan lebih lanjut, proses ini juga dapat dianalisis untuk melihat perubahan data pada tabel di sebuah halaman web, seperti yang dilakukan oleh (Winnetou, et al., 2017) pada data ware house.

3. METODOLOGI PENELITIAN

Metodologi penelitian yang dilakukan di antaranya:

1) Studi literatur

Pada langkah pertama ini dilakukan studi terhadap beberapa buku dan artikel yang membahas mengenai ekstraksi bagian tabel pada halaman web. Tahapan ini dilakukan untuk menemukan fokus persoalan yang akan dibahas pada penelitian ini. Studi literatur ini juga dilakukan untuk merumuskan solusi yang akan ditawarkan pada penelitian ini.

2) Analisis permasalahan dan perumusan hipotesis

Setelah semua bahan literatur dipelajari, pada tahap ini dilakukan analisis untuk merumuskan hipotesis solusi yang akan ditawarkan. Salah satu hipotesis sebagai hasil yang diperoleh pada tahapan ini adalah bahwa untuk tidak menghilangkan makna data, maka tabel yang berbentuk *column-row wise* diekstrak dan difaktorisasi menjadi tiga tabel. Hal ini akan dibahas pada bagian berikutnya.

3) Perancangan solusi

Pada tahap ini, mulai dirancang algoritma sebagai solusi dari permasalahan yang diangkat. Algoritma yang dikembangkan dapat membagi tabel yang berbentuk *column-row wise* menjadi tiga bagian. Tahap ini akan dibahas pada bagian berikutnya.

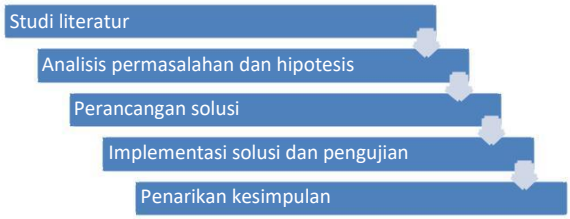
4) Implementasi solusi dan pengujian

Solusi yang telah dirancang kemudian diujikan dengan mengimplementasikannya ke dalam kode program. Pada penelitian ini, digunakan bahasa pemrograman PHP. Kemudian, kode program diuji dengan melakukan ekstraksi bagian tabel pada sebuah halaman web yang berbentuk *column-row wise*.

5) Penarikan kesimpulan

Hasil pengujian ini kemudian dianalisis untuk merumuskan beberapa kesimpulan yang dapat diambil dari penelitian.

Gambar 1 merupakan tahapan metodologi penelitian yang dilakukan.



Gambar 1. Metodologi Penelitian

4. PROSES EKSTRAKSI TABEL

Untuk mengekstraksi tabel dari sebuah halaman web, digunakan pendekatan yang disampaikan oleh (Akbar, et al., 2015). Tahapan ekstraksi ini dapat dilihat pada blok diagram Gambar 2. Yang menjadi input pada proses ini adalah dokumen HTML dari halaman web dan menghasilkan keluaran berupa basis data yang berisi data tabel hasil ekstraksi. Bagian berikut akan dijelaskan proses yang terjadi dari masing-masing tahapan.

1) Menentukan bagian tabel dari sebuah halaman web

Bagian tabel dikenali dengan mengenali tag `<table>` pada dokumen HTML. Bagian-bagian lain dari sebuah tabel juga dikenali dengan beberapa tag yang merepresentasikannya, seperti: tag `<tr>` untuk mengenali baris, tag `<th>` untuk mengenali bagian header dari tabel atau dalam penelitian ini dinamakan atribut, tag `<td>` untuk mengenali isi dari masing-masing cell.

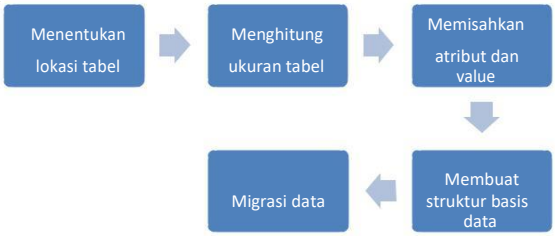
2) Mengenali ukuran tabel

Ukuran tabel dihitung berdasarkan jumlah tag `<th>`, `<tr>`, dan `<td>`. Banyak kolom sebuah tabel dihitung dari jumlah tag `<tr>` yang dikenali, sedangkan banyak baris dihitung dari jumlah tag `<th>` atau `<td>` yang dikenali. Ukuran minimal tabel yang ditangani pada penelitian ini adalah 2 x 3 atau 3 x 2.

3) Memisahkan bagian atribut dan bagian data

Atribut pada tabel dapat dikenali dengan dua cara, yaitu: (1) Jika pada tabel yang dikenali terdapat tag `<th>` maka bagian ini menjadi atribut dari tabel. (2) Jika pada tabel tidak ada tag `<th>` atau semua data ditulis dengan tag `<td>` maka terdapat 4 karakteristik visual yang dapat digunakan untuk membedakan atribut dan data, yaitu: jenis huruf (*bold*, *italic*, *underline*), ukuran huruf (*big size*, *font size*), warna huruf, dan *background*. Keempat karakteristik visual tersebut dapat dikenali dari tag atau komponen visual yang terdapat di dalam tag.

Dari pemisahan kedua bagian ini, dapat ditentukan apakah tabel berbentuk *column-row wise* atau bukan. Tabel dengan bentuk ini dikenali dari indeks atribut yang dikenali. Jika atribut yang dikenali terdapat pada baris pertama dan kolom pertama maka tabel berbentuk *column-row wise*.



Gambar 2. Tahapan Ekstraksi Tabel

4) Membuat struktur basis data

Pada penelitian ini, tabel yang berbentuk *column-row wise* disimpan ke dalam basis data menjadi 3 tabel. Tabel pertama dan kedua untuk menyimpan bagian atribut dan tabel ketiga menyimpan data dan atribut yang terkait dengannya.

Bagian atribut yang terdapat pada baris pertama disimpan ke dalam basis data menjadi satu tabel baru dengan nama tabel `Tabel1` yang memiliki satu atribut, diberi nama `Atribut1`. Bagian atribut yang terdapat pada kolom pertama dismpan ke dalam basis data menjadi satu tabel baru dengan nama tabel `Tabel2` yang memiliki satu atribut, diberi nama `Atribut2`. Kombinasi kedua atribut disimpan ke dalam basis data menjadi satu tabel baru dengan nama `Tabel3` dan terdiri dari 3 kolom, yakni `Atribut1`, `Atribut2`, dan sebuah kolom baru `Data`.

Ilustrasi tabel yang terbentuk pada basis data dapat dilihat pada Tabel 3. Tabel ini terbentuk dari ekstraksi tabel *column-row wise* pada Tabel 2.

5) Migrasi seluruh isi tabel ke dalam basis data

Setelah semua komponen telah disiapkan maka seluruh isi tabel siap dipindahkan ke basis data. Pada bagian ini dijalankan *query insert* data ke dalam basis data.

Tabel 3. Hasil Ekstraksi Tabel yang Berbentuk Column-Row Wise

Tabel 1	Tabel 2	Tabel3		
Atrbt 1	**Atrbt 2**	**Atrbt 1**	**Atrbt 2**	**Data**
dept1	Q1	dept1	Q1	1.3
dept2	Q2	dept1	Q2	1.32
	Q3	dept1	Q3	1.3
	Q4	dept1	Q4	1.35
		dept2	Q1	1.4
		dept2	Q2	1.35
		dept2	Q3	1.15
		dept2	Q4	1.2

Keseluruhan proses ekstraksi pada metode ini dapat diterapkan untuk HTML versi 4 ataupun HTML versi 5. Proses ekstraksi tabel yang dilakukan berdasarkan tekstual yang tertulis pada dokumen HTML sehingga independen terhadap versi HTML yang digunakan. Hanya saja, jika terdapat perubahan jenis tag yang digunakan pada HTML versi terbaru, maka perlu ada penyesuaian pada proses ekstraksinya untuk memperbaharui atau mengenalkan jenis tag yang baru dari dokumen HTML tersebut.

Sebagai contoh, pada HTML 5, tag <big> sudah tidak digunakan atau dihapus. Dalam proses ekstraksi yang diajukan, tag ini digunakan untuk mengenali bagian atribut, seperti yang dijelaskan pada proses ke-3. Penggunakan tag <big> di dalam proses ekstraksi menjadi sesuatu yang tidak berguna, sehingga perlu dihilangkan atau tidak digunakan lagi. Meskipun secara keseluruhan tidak mengganggu proses ekstraksi.

5. ALGORITMA EKSTRAKSI TABEL

Solusi yang ditawarkan pada tulisan ini adalah sebuah algoritma dalam bentuk semi *pseudocode* sehingga dapat diterapkan dalam bahasa pemrograman yang berbeda. Berikut ini adalah algoritma yang ditawarkan.

Input: Halaman HTML
Output: Tabel pada DBMS
Algoritma:

```
1.  Find <table>
2.  Find <tr>
3.  m ← Count(<tr>)
4.  for <tr>[1]
       Find <th>, Find <td>
       Count(<th>), Count(<td>)
5.  if Count(<th>) = 0
       n ← Count(<td>)
    else n ← Count(<th>)
6.  atrr = 0, atrc = 0
7.  for <tr>[1]
       atrr = count(<td>.isBold)
       atrr = count(<td>.isItalic)
       atrr = count(<td>.isUnderline)
       atrr = count(<td>.isBig)
8.  for i = 0; i < m; i++
       for <tr>[i]
          if <td>[1].isBold:
             atrc += 1, break
9.  if atrr = m and atrc =
       n isColRowWise = True
10. if isColRowWise
       for i = 0; i < m; i++
          insert into Tabel1 atribut
          ke-i
       for i = 0; i < n; i++
          insert into Tabel2 atribut
          ke-i
       for i = 0; i < m; i++ for
          j = 0; j < n; j++
             insert into Tabel3
             atribut ke-i, atribut ke-
             j, and data ke-(i,j)
```

Fungsi Find <table>, Find <tr>, Find <th>, dan Find <td> diimplementasikan dengan menerapkan perbandingan string atau karakter. Di dalam dokumen HTML diperiksa apakah terdapat string dari setiap tag. Tabel diawali dengan tag <table> dan diakhiri dengan tag </table>, baris diawali dengan tag <tr> dan diakhir dengan tag <\tr>, dan begitu seterusnya.

Namun, harus diperhatikan bahwa di dalam tag tersebut, seringkali terdapat atribut dari tag, seperti colour, colspan, scope, class, dan lainnya, misalkan <th scope="col">. Oleh karena itu, masing-masing tag cukup dikenali dengan string "<table", "<th", "<tr", "<td" tanpa kurung penutup. Kurung penutup dikenali kemudian untuk menandakan bahwa tag berakhir pada bagian tersebut. Setiap satu pencarian berakhir jika mengenali string "</table", "</th", "</tr", dan "</td".

6. IMPLEMENTASI DAN PENGUJIAN

1) IMPLEMENTASI

Algoritmayangdihasilkankemudian diimplementasikan dalam bahasa pemrograman untuk kemudian diuji dengan beberapa bahasa pemrograman. Bahasa pemrograman yang dipilih adalah bahasa Python. Pengguna memasukkan alamat URL halaman web yang akan diekstrak pada *console* dari IDE yang digunakan.

Hasil implementasi kemudian diujikan untuk mengekstrak tabel pada Gambar 3. Tabel ini merupakan data kependudukan menurut kepadatan penduduk DKI Jakarta. Data ini didapatkan dari http://data.jakarta.go.id/dataset.

Tabel pada halaman web ini *berbentuk column-row wise*. Pada bagian kolom, terdapat 1 buah atribut, yakni Nilai. Pada bagian baris terdapat 8 buah atribut, yakni Last Updated, Dibuat, Sumber, Frekuensi Penerbitan, Tahun, Cakupan, Penyajian, dan Kontak. Pada algoritma ekstraksi yang dikembangkan, cell yang berisi nilai Field diabaikan karena bukan atribut dari sebuah data. Selain ketiga bagian tersebut merupakan bagian data dari tabel. Bagian ini merupakan nilai dari kedua atribut yang saling bersesuaian. Misal, nilai May 17, 2018, 05:42 merupakan data dari atribut Nilai dan Last Updated.

Field	Nilai
Last Updated	May 17, 2018, 05:42
Dibuat	May 27, 2015, 08:09
Sumber	Dinas Kependudukan dan Pencatatan Sipil
Frekuensi Penerbitan	1 Tahun sekali
Tahun	2014 dan 2016
Cakupan	Provinsi DKI Jakarta
Penyajian	kelurahan
Kontak	bdi.dukcapil@gmail.com

Gambar 3. Data Kependudukan DKI Jakarta yang Berbentuk Tabel Column-row Wise

Aplikasi berhasil mengekstrak tabel dan menyimpan hasil ekstraksi pada basis data menjadi tiga buah tabel baru. Tabel pertama menyimpan atribut pada baris yang memiliki satu atribut, yaitu **atribut1**. Atribut ini menyimpan data pada baris yang dideteksi merupakan atribut dari tabel yang diekstrak. Hal ini dapat dilihat pada Gambar 4. Tabel kedua menyimpan atribut pada kolom yang juga memiliki satu atribut, yaitu **atribut2**. Atribut ini menyimpan data pada kolom yang juga dideteksi sebagai atribut dari tabel yang diekstrak. Hal ini dapat dilihat pada Gambar 5. Tabel ketiga menyimpan bagian data dari dua atribut yang bersesuaian posisinya yang disimpan dalam atribut **data** pada tabel ketiga. Selain menyimpan data, tabel ini juga menyimpan kedua atribut yang melekat pada data tersebut, yakni **atribut1** dan **atribut2**. Hal ini dapat dilihat pada Gambar 6.

atribut1
last updated
dibuat
sumber
frekuensi penerbitan
tahun
cakupan
penyajian
kontak

Gambar 4. Hasil Ekstraksi Atribut pada Baris yang Tersimpan di Dalam Basis Data

atribut2
nilai

Gambar 5. Hasil Ekstraksi Atribut pada Kolom yang Tersimpan di Dalam Basis Data

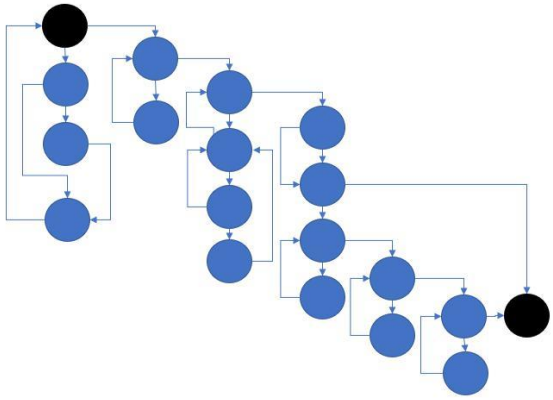
atribut1	atribut2	data
last updated	nilai	may 17, 2018, 05:42
dibuat	nilai	may 27, 2018, 08:09
sumber	nilai	dinas kependudukan dan pencatatan sipil
frekuensi penerbitan	nilai	1 tahun sekali
tahun	nilai	2014 dan 2018
cakupan	nilai	provinsi dki jakarta
penyajian	nilai	kelurahan
kontak	nilai	bdi.dukcapil@gmail.com

Gambar 6. Hasil Ekstraksi Data yang Tersimpan di Dalam Basis Data Kedua Atribut

Tabel ketiga inilah yang menyimpan keseluruhan nilai dari tabel *column-row wise* sehingga keterhubungan makna antara dua atribut dalam sebuah data tidak hilang. Meskipun dari makna pada tabel yang diekstrak tidak hilang, namun makna pada tabel yang dihasilkan pada *database management system* justru sebaliknya. Makna sebuah data di dalam basis data tidak didapatkan dari tabel hasil ekstraksi. Data pada basis data dapat mengalami penambahan, yaitu dengan penambahan baris pada tabel karena data yang menjadi nilai sebuah atribut adalah data yang memiliki kesamaan karakteristik. Tabel hasil ekstraksi yang disimpan pada basis data menghilangkan makna tersebut. Data pada **atribut1** tidak memiliki kesamaan karakteristik. Demikian juga data pada **atribut2** dan **data**. Sehingga, data pada tabel hasil ekstraksi ini tidak dapat dilakukan penambahan. Namun demikian, untuk hanya sekedar sebagai tempat menyimpan hasil ekstraksi, teknik yang digunakan sudah dapat diterima dengan baik.

2) ANALISIS KOMPLEKSITAS ALGORITMA

Pada bagian ini akan dibahas analisis kompleksitas terhadap algoritma yang diusulkan. Analisis kompleksitas algoritma yang digunakan adalah cyclomatic complexity number.



Gambar 7. Control Flow Graph Usulan Algoritma

Tahap awal untuk menghitung nilai ccn adalah membuat flow graph dari algoritma yang diusulkan. Flow graph dari algoritma yang disampaikan pada bagian 5 dapat dilihat pada Gambar 7. Berdasarkan gambar tersebut, nilai ccn dari algoritma ini adalah 12. Hal ini menunjukkan bahwa kompleksitas algoritma yang diusulkan masih tinggi

7. KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari penelitian ini antara lain:

1) Algoritma yang dihasilkan berhasil diterapkan dengan menggunakan bahasa pemrograman Python.
2) Tabel *column-row wise* diekstraksi menjadi tiga tabel baru pada basis data, yakni tabel untuk menampung atribut pada kolom, tabel untuk menampung atribut pada baris, dan tabel untuk menampung nilai kedua atribut.
3) Ketiga tabel yang dihasilkan diyakini tidak menghilangkan makna (semantik) dari tabel yang diekstrak.
4) Kompleksitas algoritma termasuk kategori tinggi dengan nilai cyclomatic complexity number sebesar 12.

DAFTAR PUSTAKA

AKBAR, M., AZIZAH, F. N. & SAPTAWATI, G. P., 2015. Integration of HTML Tables in Web Pages. Yogyakarta, IEEE, pp. 132-137.

AKBAR, M., PATMALA, C. & NURMALASARI, D., 2016. Ekstraksi Data pada Tabel dari Halaman Web Menggunakan Pohon Document Object Model (DOM). Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI) UGM, November, 5(4), pp. 265-271.

EMBLEY, D. W., TAO, C. & LIDDLE, S. W., 2004. Automating the extraction of data from HTML tables with unknown structure. Data & Knowledge Engineering (Elsevier), November, Volume 54, pp. 3-28.

KERUI, C. ET AL., 2011. Automatic table integration by domain-specific ontology. International Journal ofDigital Content Technology and Its Application, January, 5(1), pp. 218-226.

KIM, Y.-S. & LEE, K.-H., 2007. Extracting logical structures from HTML tables. Computer Standards and Interfaces (Elsevier), August, 30(5), pp. 296-308.

LIM, S.-J., NG, Y.-K. & YANG, X., 2002. Integrating HTML tables using semantic hierarchies and meta-data sets. s.l., s.n.

WINNETOU, A. B., WICAKSONO, S. A. & PINANDITO, A., 2017. Analisis Peningkatan Performa Proses ETL (Extract, Transform, Dan Loading) Pada Data Warehouse Dengan Menerapkan Delta ExtractionMenggunakan Historical Table. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Agustus, 2(4), pp. 1366-1371.