Journal of Computer Science 7 (2): 129-142, 2011
ISSN 1549-3636
© 2011 Science Publications

Proposing the new Algorithm and Technique Development for Integrating Web Table Extraction and Building a Mashup

Rudy A.G. Gultom, Riri Fitri Sari and Bagio Budiardjo
Department of Electrical Engineering, Faculty of Engineering,
University of Indonesia, Depok 16424, Indonesia

Abstract: Problem statement: Nowadays, various types of data in web table can be easily extracted from the Internet, although not all of web tables are relevant to the users. As we may know, most web pages are in unstructured HTML format, making web table extraction process very time consuming and costly. HTML format only focuses on the presentation, not based on the database system. Therefore, users need a tool in dealing with that process. **Approach:** This research proposed an approach for implementing web table extraction and making a Mashup from HTML web pages using Xtractorz application. It is also discussed on how to collaborate and integrate a web table extraction process in the stage of building a Mashup, i.e., Data Retrieval, Data Source Modeling, Data Cleaning/ Filtering, Data Integration and Data Visualization. The main issue lies in stage of data modeling creation, in which Xtractorz must be able to automatically render Document Object Model (DOM) tree in accordance to HTML tag or code of the web page from which the table is extracted. To overcome that, the Xtractorz is equipped with algorithm and rules so it can enable to specifically analyze the HTML tags and to extract the data into a new table format. The algorithm is created by using recursive technique within a user-friendly GUI of Xtractorz. **Results:** The approach was evaluated by conducting experiment using Xtractorz and other similar applications, such as RoboMaker and Karma. The result of experiment showed that Xtractorz is more efficient in completing the experiment tasks, since Xtractorz has fewer steps to complete the whole tasks. **Conclusion:** Xtractorz can give a positive contribution in terms of algorithm technique and a new approach method to web table extraction process and making a Mashup, where the core algorithm can extracts web data tables using recursive technique while rendering the DOM tree model automatically.

Key words: Web table extraction, mashup stages, recursive algorithm, Document Object Model (DOM), HTML format, Integrated Development Environment (IDE), data integration

INTRODUCTION

Nowadays, various types of data can be easily extracted from the Internet web pages, although not all of the data is relevant to the users. We know that most web pages are in unstructured HTML format, making the data extraction or query time consuming and costly. HTML format only focuses on the presentation, not based on the database system. Although at present we can find a number of new formats such as XML or XHTML which can separate or distinguish data structure from its layout, making it easier to exchange data via web pages.

To solve that problem, users need a tool for web table extraction process and making Mashup stages. Web table extraction is a technique which Internet users directly use to extract a data table from a web page with an unstructured format (HTML) (Baumgartner *et al*., 2001a; 2001b; Cafarella *et al*., 2008; Gatterbauer *et al*., 2007; Singh *et al*., 2010; Tuchinda *et al*., 2008;), where Mashup is a web-based application which integrates data extracted from multiple web pages in several stages: Data Retrieval, Data Source Modeling, Data cleaning/filtering, Data Integration and Data Visualization (Tuchinda *et al*., 2008).

Mashup is needed to integrate a series of data extraction processes from multiple web pages which are available in an HTML format into a new format such as XML or XHTML. Once Mashup is created, more relevant data or data tables can automatically be extracted from multiple sources in the Internet.

However, extracting a data table in the web table extraction process which is combined with a Mashup building is not an easy task to do. A lot of problems

Corresponding Author: Rudy. A.G. Gultom, Department of Electrical Engineering, Faculty of Engineering,
University of Indonesia, Depok 16424, Indonesia

may occur; for instance, it is difficult to understand the content of unstructured HTML, consisting of thousands of HTML tags or codes which need to be sorted or indexed and also analyzed in order to determine which belongs to Parent, Child, Sibling, or Leaf Node group. Therefore, it makes modeling the HTML structure into a DOM tree becoming an essential aspect. For that purpose, the need of an artificial intelligence is required in the formulation of its algorithm (Dehuri *et al*., 2006; Mamat *et al*., 2006; Sleit *et al*., 2007).

Another problem is the dominance of human factor (users) in the web table extraction process. In several similar applications such as RoboMaker (OpenKapow), YahooPipes, or Karma, that problem may occur because users search and select data table from a single web page manually. Since it is time consuming and costly, the process becomes less effective and efficient.

Currently there are millions even billions of Internet webpages containing potential data tables for extraction. To overcome the problems mentioned above, we proposed research on a web table extraction technique which is combined with a Mashup building in the Xtractorz application. In this study our Algorithm and rules for web table extraction technique and building Mashup stages are introduced.

We used one scenario to test the Xtractorz application system which was "assigned" to search and extract data in the data tables on the web page of 2009 National General Election Result by downloading them from http://partai.info/pemilu2009.

In the first stage, Data Extraction, the Xtractorz system would retrieve and download relevant data from the targeted tables (Table 1 and 2). In the next stage, Data Modeling, Xtractorz will model the data extracted by referring to the DOM tree concept. In testing the scenario, selected data in the columns of data tables would be modeled, such as columns of "Political Party," "Number of Votes", "Percentage", "Number of Seats Won" and others.

After the DOM tree was created or rendered, the next stage performed was Data Cleaning/Data Filtering, in which Xtractorz did the cleaning or filtering on the raw data having been successfully extracted, yet in a format different from what the users may need. The next stage was Data Integration, in which Xtractorz would integrate the newly extracted data and the previously extracted data in a single table in the XML format in the data repository.

The last stage in making a Mashup stages was Data Visualization, in which all the data in the data repository was presented or visualized in various different formats such as XML table, map, worksheet (excel).

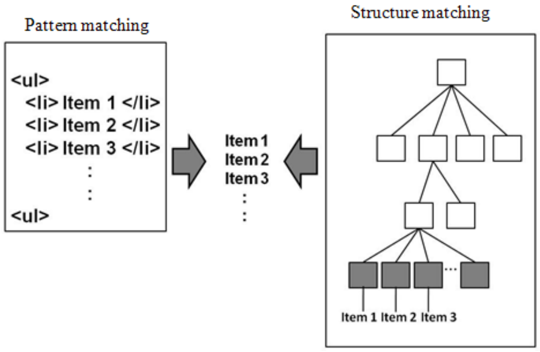


Fig. 1: Pattern matching and structure matching

Table 1: Rules in Xtractorz algorithm

Rules notion	Rules explanation
HTML tag = <tag>^</tag>	Each HTML tag must begin with <tag> and end with </tag>
Tag <script>^</script> = f	Tag <script> and </script> may be ignored
HTML tag ≡ <CSS code =f	HTML tag containing CSS code may be ignored
Focus tag ≡ <table>^</table>	Tag with becomes the main focus is tag <table></table>
$\supseteq a, b \notin V_e^1 \lvert x_1^a < x_1^0 \wedge x_2^a <$	A table does not contain overlapping columns. If there are two or more columns, each table with the same cell in each table will be a primary key
Set of table found ≥2 $\Rightarrow$ first table $\cong$ primary table	If there are two or more tables, the first found is the main table, while the rest of the tables are the supporting ones
Set of tables extracted $\subseteq$ a DOM tree	Tables witch are extracted from a DOM tree
$\lvert x_1^3 \rvert, \lvert x_2^3 \rvert \geq 1, \lvert y_1^3 \rvert, \lvert y_2^3 \rvert \geq 2$	The table is two dimensional and has at last 2 columns
$\supseteq_e, \notin V_e^f \lvert\ \lvert \{w \lvert e$ contains w	The content cell in each column in the table should not be more than 10 words

Table 2: Existing mashup tools comparison

	Data retrieval	Data source	Data modeling clearing	Data integration
MS's popfly	Widgets	Manual	Widget	Widget
CMU's marmite	Widgets	Manual	Widget	Widget
Yahoo's pipes	Widgets	Manual	Widget	Widget
Intel's mashmaker	Dapper	Manual	Widget	Export
MIT's simile	DOM	Manual	N/A	N/A
Dapper	DOM	Manual	Manual	Manual
Xtractorz	DOM	Auto	Manual	Manual

The main objective of this study is to introduce the Xtractorz system which can be used to perform a web table extraction and a Mashup building from different types of web pages containing data tables from the simple table to the complex one (nested table).

Literature review: Web Table Extraction: Web Table Extraction is usually performed on structured document sources such as data table. The documents are usually in the format of markup languages (for instance: HTML, XHTML, XML) to express the structure of the data in it.The currently available methods to extract data table focus only on the representation of text documents (table context, text label) or the structure of the document tree (Barinka and Jelinek, 2009). Basically, web Table Extraction is divided into two categories, namely: a) Pattern matching and b) Structure matching (Fig. 1).

Pattern matching is a method of accessing documents in the form of text using text patterns such as pattern matching and regular expression. This approach normally only discusses the local data structure (at the line level), rather than specifically the whole document structure.

Structure matching is a method of accessing documents available in a single structure of document tree. The method uses an approach based on paths and a relation between the available nodes (XPath, Xquery) (Chamberlin, 2003; Vijayalakshmi and Mohan, 2010). In this method, the activity of accessing documents can be done all at once or one at a time on each individual subtree. The selected relevant subtrees are presented based on the result of the visual selection (Baumgartner *et al*., 2001b; Liu *et al*., 2002; Vijayalakshmi and Mohan, 2010).

Now there are some web table extraction tools in the market both the open source one and the paid one such as Kapow Mashup Server 6.3 Robomaker, Lixto Visual Developer, Yahoo.Pipes, Marmite and many more.

Mashup: Mashup is a web application which combines data and functions of two or more external sources (web pages) to create a single web page service. An example of Mashup application is the use of cartography in Google Map facility which can add information about the location of real estate data, by creating a single web service which originally is not related to the available relevant data. These days, the community of programmers-creators of Mashup web application rapidly developed with its orientation to combine the available web-based contents with the element of services to create a new web application service.

The main problem in the Mashup building is that it requires some expertise in the field of computer language programming, including in the fields of web crawling, text parsing, pattern matching, pattern matching, databases and HTML tag codes (Wong and Hong, 2007). The standardization in the Mashup building consists of five stages (Tuchinda *et al*., 2008), they are:

- Data Retrieval is the first stage in which an application will extract data or a data table from an unstructured web page (HTML) into a structured data source (XML, XHTML). In this stage, there are also rules which regulate the manner (algorithm) in extracting relevant/specific data from multiple web page sources, since without the algorithm the extraction will be more difficult and complex (Dehuri *et al*., 2006; Huynh *et al*., 2005; Knoblock *et al*., 2003; Mamat *et al*., 2006; Sleit *et al*., 2007)
- Data Source Modeling is a stage for the the process of determining HTML tag codes and assigning the attribute names in modeling the extracted data in each column in the table, creating a relation between the recently extracted data or data table or the previously extracted one
- Data Cleaning/ Filtering is the stage in a Mashup building which requires corrections or rearrangements to be made on the extracted data or the content of data tables, such as, corrections on misspelled words, right/left alighment, or transformation of data from one data format into a more structured data format
- Data Integration is the stage in which the application system specifies how to combine two or more sources of the previously extracted data or content of data table and the currently extracted one, using the database joint operation
- Data Visualization is the last stage in the creation of Mashup, in which the extracted data is visually presented by the users into various different formats as necessary such as XML table, web page, map, graph and other formats

RoboMaker: RoboMaker is a RoboSuite application from OpenKapow which functions to create or to debug various types of robots. In the RoboMaker, users can create any robot according to the task the users want it to do, for instance, a robot which is assigned to collect data by extracting data or objects from various different web pages in the Internet, or a clipping robot which is assigned to clip part by part in a webpage in an HTML format and those parts will be presented in another format such as Portal or a new web page. RoboMaker also creates a medium for the Integrated Development Environment (IDE) for various types of robots. It means that the users should be able to understand the concept of assignment programming of each robot which has its own unique syntax and semantics. To facilitate the users in creating various kinds of robots, RoboMaker provides complete features in its GUI, from the interactive visual programming, capability of full debugging to easy access online assistance on a sensitive context issue (Heier, 2008).

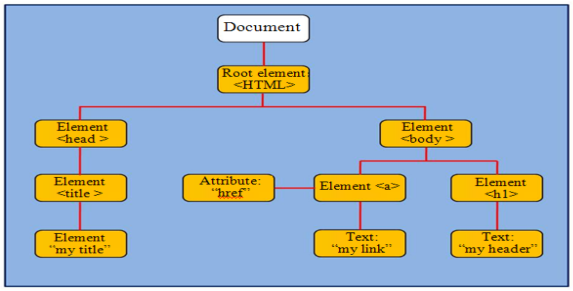


Fig. 2: Overview of the DOM tree

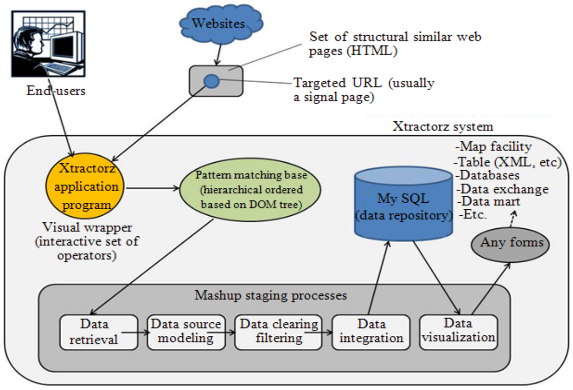


Fig. 3: Design and Architecture of Xtractorz System

Karma: Karma is a Mashup building application which includes or combines the concept of programming by demonstration (Tuchinda *et al*., 2008). Karma helps the users to implicitly solve the problem relating to the integration of information by providing various simple examples. To build a Mashup, the users can combine data or data tables from a web page which becomes the source and they can present the final result in a map. In Karma, there are four stages in the process of building Mashup, namely: Data Retrieval, source modeling, data cleaning and Data Integration. The contribution of Karma in this study is its approach in building Mashup by combining four techniques in the integration of information, which normally is done separately, into a joint framework (Tuchinda *et al*., 2008).

In that framework, the users can build a Mashup, without having to have the expertise in computer programming or understand the programming concept thoroughly. They can simply use some examples of the final results which have been prepared by Karma for each form of operations the users want to use.

DOM tree: DOM Tree (Document Object Model) is a cross-platform and a language-independent convention which is used to represent and create interactions between different document objects into an HTML or XML web page as well as to present them in a tree structure which is called a node-tree. All the nodes in the webpage can be accessed through that tree and the content can also be modified or deleted and a new element can be created.

The DOM tree is usually employed as an approach for the process of web information extraction and wrapper generation to determine the pattern of information from the HTML tag or code structure in a web page (Gatterbauer *et al*., 2007).

In the concept of DOM tree, Root is the highest Node in which each Node, except for Root, has a single Parent and one Parent can have several Children. Leaf is a Node without Children, while Siblings are Nodes from the same Parent.

In a DOM tree or a node tree presented in Fig. 2, we can see a set or a group of nodes which describes the connections between nodes, in which a series of those trees from their root node then go down to the text node which represents the lowest level of the tree.

MATERIALS AND METHODS

The objective of this study is to to implement web table extraction process in collaboration with building a Mashup, this mean the users can easily extract the relevant data tables from targeted web pages followed by building a Mashup, without having to burden them with the necessary expertise in computer programming.

Therefore, we proposed a new approach for algorithm, rules and technique development based on recursive technique to create a DOM tree (Document Object Model) automatically, so the users can do web table extraction process and also building a Mashup via Internet more easily.

We also proposed a new application system, called Xtractorz, to implement this new approach into a GUI (Graphical User Interface). We designed Xtractorz so it can process the rendering and parsing of HTML codes within web pages and extracted automatically into the DOM tree form.

During the process, all of HTML codes has been analysed and indexed its data table structures by the Xtractorz algorithm and grouped into Root, Parent, Child, Sibling and Leaf Nodes. It showed that the Xtractorz is capable to complete web table extraction and making a Mashup process and also contributing a new algorithm in rendering DOM tree automatically.

Design and architecture:

Xtractorz: Xtractorz is a web table extraction application system prototype and a Mashup builder which we continuosly develop (Gultom *et al*., 2010). Xtractorz is made in the Hypertext Preprocessor (PHP) computer language and Asynchronous Javascript And XML (AJAX). The reason why PHP is used is because it is one of the most popular web programming languages nowadays and PHP is a programming language specifically designed to create websites. AJAX is not a new programming language, but AJAX technically uses XMLHttpRequest object with javascript to communicate with the server asynchronously. By using XMLHttpRequest object in AJAX can make a process run at the background while the users can interact with the existing web pages.

Basically, Xtractorz is designed to implement the web table extraction process and the Mashup building stages, so the users can easily extract the relevant data from different sources of websites in the Internet and build a Mashup, without having to burden them with the necessary expertise in computer programming. The design and architecture of Xtractorz system can be seen in Fig. 3.

The design and architecture of Xtractorz application system includes several steps:

- In the early stage, Xtractorz will search the web page (URL) which becomes the targeted data source and the relevant data or data table will be extracted but in the unstructured HTML format
- And then, the data or a data table which has been extracted will be processed for data restructured by Xtractorz, by performing Parsing on HTML tag code for each element of data in the table data columns and then performing modeling of those data for the users' purpose
- In the next stage, Data Cleaning/ Filtering is performed on the data or content of the data table successfully extracted and that in the existing data tables in the data repository in order to integrate the data for the subsequent process, the stage of Data Integration
- In the Data Integration stage, a process of integration takes place as Xtractorz will collect and store all of the currently extracted data and the previously extracted one into a single new structured table in the data repository, which uses MySQL database system, for further computation
- In the last stage, Data Visualization, in the Mashup building, all of the data which has been integrated in the data repository can be presented in various visualization formats such as table, graph, web browser, website, map

The proposed algorithm: The Xtractorz application system is equipped with the proposed algorithm so it can enable to specifically analyze the HTML tags and to extract the data table into a new table format (Gultom *et al*., 2010). A number of algorithms related to Web Table Extraction and building a mashup can be seen in (Cafarella *et al*., 2008), where some algorithms are also inspiring (Dehuri *et al*., 2006; Hergli *et al*., 2005). The proposed algorithm is created by using recursive technique, as follows:

```
DomTree($tag,$CodeHtml,$Parent,$Index) {
  // Parsing $CodeHtml
 $ResultParsing=ParsingCode($Tag,$CodeHtml);
  // To Stop Recursive Condition
 If (NodeLeaf($HasilParsing)) {
     Exit;
 } Else {
     // Find Tag Child
     $TagChild=Array();
     $TagChild=FindTagChild($ResultParsing)
     For (i=0;i<count($TagChild),i++) {
     DomTree($TagChild[$i],$ResultParsing,
     $Index,$Index++)
     }
   }
}
```

In principle, this proposed algorithm analyzes the HTML tags or codes from the web data table which becomes the targetted data source from a website. The analysis of HTML tags or codes is conducted for indexing the group structure of the web table, such as which node is the Parent, Child, Sibling and Leaf. The core Xtractorz algorithm above can be described in the following steps:

- The First Step is to determine the initial HTML tag and HTML codes, in which the indexing is created for the first time Parent: -1 dan Index: 0, or as the first in the order since there is no parent.
- The Second Step is to perform Parsing on the HTML codes, in order that the <tag> and </tag> in them can be retrieved, resulting in:
  <head>..................</head><body></tbody>
- The Third Step determines a stop condition of a recursive process in the core algorithm, or a condition in which the HTML codes no longer has <tag> and </tag> and it means that the Leaf node has been obtained. Before the Leaf node is obtained, the algorithm will continue searching other tags which are the Children of the tags, just like the initial step, in order to get:
  <head></head>
  <body></body>
- The Fourth Step is when the recursive process towards the first step for each Field from the perviously obtained and analyzed:

- Recurcive process for tag for Head
  <head><title>.........</title></head>
- Recurcive process for tag for Body
  <body><table width="100%" border="0">
  <tbody><tr><td>..</td><td>..</td><td>
  ...etc...</td></tr></tbody></body>
- Recurcive process for tag for Table
  <tablewidth="100%"border="0"><tbody>
  <tr><td>...</td><td>...dst...</td></tr>
  </tbody></table>
- Recurcive process for tag for Tbody
  <tbody><tr><td>...</td><td>...dst..</td>
  </tr></tbody>
- Recurcive process for tag for Tr part
  (Tr is for a header column in the table)
  <tr><td>No</td><td>...</td><td>..dst..
  </td></tr>
- Recurcive process for tag for Td part
  <td>No</td><td>...</td><td>..dst..</td>
  <tr><td>...</td><td>...dst..</td></tr>

The recursive of Xtractorz algorithm will stop the searching after it finds leaf nodes in the HTML code. Then, all of the findings will be indexed for recognition and identification of Parent, Child, Sibling in each Leaf node which is found. The Leaf nodes which are found in those HTML codes are grouped into a form of Array. Here, the recurcive-based core algorithm is implemented in the Loading Page and Data Retrieval stages in the stage of building a Mashup.

To implement this recursive-based core algorithm, Xtractorz must follow several rules which are created by referring to the Parents, Children, Sibling and Leaf structures which exist in each HTML code in a website. Using the method in writing mathematical notion (Gatterbauer *et al*., 2007; Ilieva and Ormandjieva, 2005), in Table 1 we can see some notions and the explanations of Xtractorz algorithm rules.

The design of DOM Tree in Xtractorz: In this study, the DOM tree is used as the basic structure in the web table extraction process in the targeted website. The DOM tree is formed based on the organizing of HTML structure consisting of codes or tags in it. Figure 4 and 5 illustrate how the selected web tables from the targeted website(http://partai.info/pemilu2009/)are automatically simplified in the form of DOM tree.

The DOM tree approach is the most effective way to identify the HTML tags or codes before the web data table extraction. Using the DOM tree, users can easily find the Xpath path (http://www.w3.org/TR/xpath) by identifying tags from the highest level paths of Root, Parent, Child, Siblings to the Leaf, while at the same time performing Parsing.

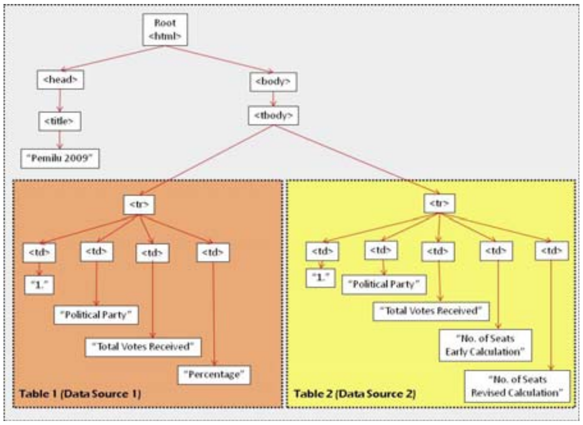
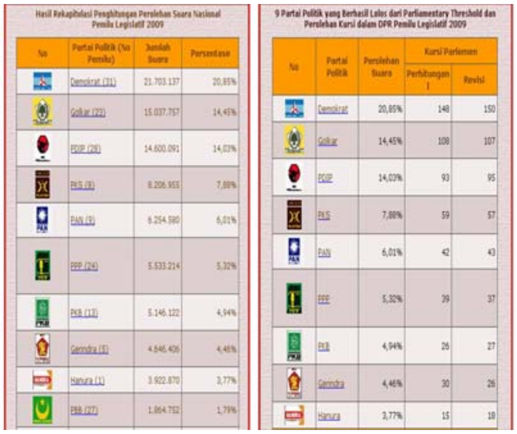


Fig. 4: The DOM Tree model structure in Xtractorz



Hasil Rekapitulasi Penghitungan Perolehan Suara Nasional Pemilu Legislatif 2009

No	Partai Politik (No Pemilu)	Jumlah Suara	Persentase
	Demokrat (31)	21.703.137	20,85%
	Golkar (23)	15.037.757	14,45%
	PDIP (28)	14.600.091	14,03%
	PKS (8)	8.206.955	7,88%
	PAN (9)	6.254.580	6,01%
	PPP (24)	5.533.214	5,32%
	PKB (13)	5.146.122	4,94%
	Gerindra (5)	4.646.406	4,46%
	Hanura (1)	3.922.870	3,77%
	PBB (27)	1.864.752	1,79%

9 Partai Politik yang Berhasil Lolos dari Parliamentary Threshold dan Perolehan Kursi dalam DPR Pemilu Legislatif 2009

No	Partai Politik	Perolehan Suara	Kursi Parlemen	
			Perhitungan I	Revisi
	Demokrat	20,85%	148	150
	Golkar	14,45%	108	107
	PDIP	14,03%	93	95
	PKS	7,88%	59	57
	PAN	6,01%	42	43
	PPP	5,32%	39	37
	PKB	4,94%	26	27
	Gerindra	4,46%	30	26
	Hanura	3,77%	15	18

Fig. 5: Selected web tables to be extracted

For instance, XPath path for the "Political Party" (/tbody/tr[1]/td[2]) means that the available paths start from path:tbody, which is the first tr tag, followed by the second td tag and taking all the available Leaf nodes. To find the parallel paths, generalization of paths is performed by ignoring the existing number of nodes, such as */tbody/tr/td/* consisting of two nodes: /tbody/tr[1]/td[2] and /tbody/tr[2]/td[2].

In this stage, Xtractorz helps the users to perform the extraction of data tables containing the relevant data, such as "Name of Political Party", "Number of Votes", "Percentage", "Number of Seats Won in the Parliament". After completing the extraction of the first column, Xtractorz continues to perform the extraction of the next column based on the position of the nodes in the first column.

In the recursive-based Xtractorz algorithm, the extraction of new data will stop after the conditions for such a stop are met and the condition is when the Leaf node in one XPath path is found. Meanwhile, the matching structure approach is represented in the form of DOM tree, in which a set of nodes in the first column are used as a marker in performing computation of extraction rules which are based on the relation between a single marker and the most recent extracted node.

Flow diagram in Xtractorz: The flow diagram of the implementation of web table extraction and the stage in Mashup building can be seen in Fig. 6. The main elements in the flow diagram are the Xtractorz GUI in the process of web table extraction, from the selection of targeted URLs which become the data source, extraction of the selected data or data table from a website, to completion of stages in balding a Mashup. Meanwhile, the UML sequence diagram (Sari and Ayuningtyas, 2010) of the order of Xtractorz system process can be seen in Fig. 7, for the first time the users perform the web table extraction process by searching the determining of the targeted website (URL) where the data source of which (its data table) will be extracted.

Xtractorz implementation: In this study, we seek to implement the process of web table extraction and Mashup building, by carrying out a test on the Xtractor application system by going through the five standard stages in building a Mashup.

Data retrieval: Before entering the Data Retrieval stage, the users first download the targeted website at a real-time basis (http://partai.info/pemilu2009). The Loading Page process shows the visual form of the targeted website along with its HTML codes in the GUI screen. The next step is the Data Retrieval stage, in which the URL target which becomes the data source from which two data tables will be extracted.

The tables contain the data on the recapitulation of the result of National General Election in Indonesia in 2009. The content of the tables extracted consists of columns with attribute names “Name of Political Party”, “Number of Seats Won,” “Percentage”. Various data contents in those columns are extracted in the next stage (Data Retrieval).

In Data Retrieval stage, Xtractorz GUI presents all HTML codes of the data tables retrieved in the form of a DOM tree. Using the modified recursive algorithm, Xtractorz automatically sorts the codes (tags) which contain data tables or the ones which do not contain them.

The result is a DOM tree which is formed automatically in accordance with the data tables in the website (Fig. 8). The DOM tree makes it easier for Xtractorz to perform Data Retrieval on the columns in the tables containing data which will be extracted from the website. The algorithm in Data Retrieval stage as follow:

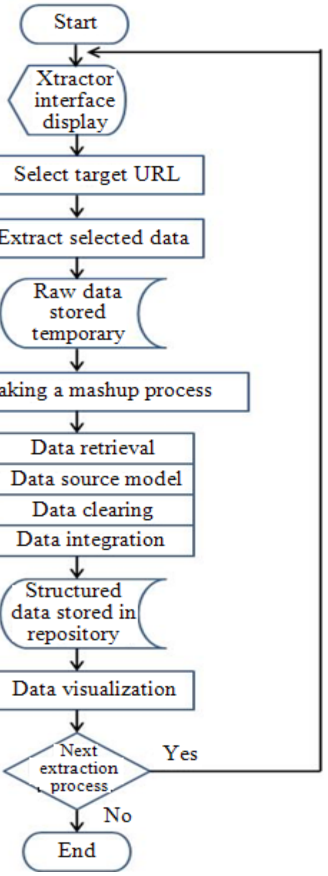


Fig. 6: Flowchart of Xtractorz system

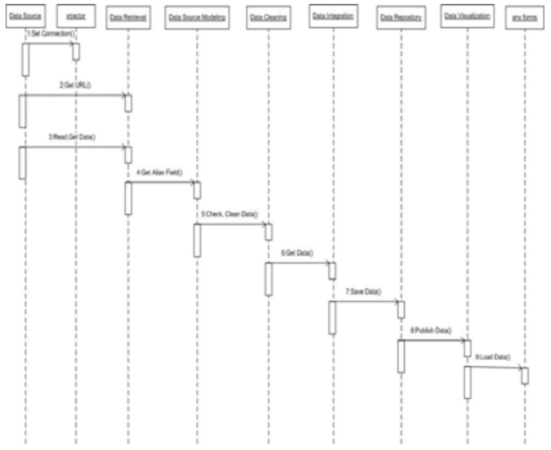


Fig. 7: Sequence diagram in building a mashup stages

```
DomTree($tag,$CodeHtml,$Parent,$Index) {
    // Parsing $CodeHtml
 $ResultParsing=ParsingCode($Tag,$CodeHtml);
   // To Stop Recursive Condition
 If (NodeLeaf($HasilParsing)) {
       Exit;
 } Else {
       // Find Other Tag Child
       $TagChild=Array();
       $TagChild=FindTagChild($ResultParsing)
       For (i=0;i<count($TagChild),i++) {
       DomTree($TagChild[$i],$ResultParsing,
       $Index,$Index++)
       }
   }
}
```

Data (source) modeling: In Data Modeling a more structured form of the target website (data sources) which has been transformed into a DOM tree is presented. This form is recognized by the users merely as a table in Excel. In the Database approach, Parent, Child, Sibling and Leaf nodes are already recognized in this stage.

In its implementation, Xtractorz does not require the users' intervention, if there are two or more tables which share at least one column, Xtractorz will present the information in a single table by placing different table columns in a single table. Using the method in writing mathematical notion (Tuchinda *et al*., 2008), the formula used in the data modeling are as follows:

$$\{a \mid \forall a,s : a \in att(s) \wedge (val(a,s) \subset V)\}$$

Where:

V = A set of values from a new column.

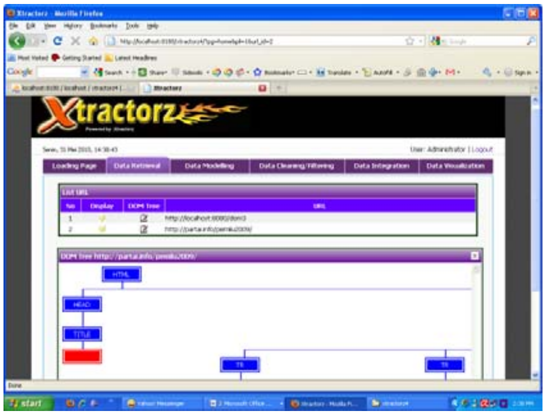
S = A set of all available data table sources in data repository.

att(s) = A procedure which returns a set of attributes from s source, where $s \in S$

val(a,s) = A procedure which returns a set of values related to attribute a in s source.

R = Candidate sets which have been ranked:

After the users extract the data table from Table 1, Xtratorz will fill the columns which have been previously given attribute names, such as "Political Party", "Number of Votes" and "Percentage", then use the values in those columns as a start to determine the mapping of attributes. For each initial value, Xtractorz performs a query to the data repository to determine whether the existing value has been stored in the table of result from the previous extraction process. If yes, Xtractorz will extract related attributes. The algorithm and result of Data Modeling in the Xtractorz GUI (Fig. 9) is as follows:

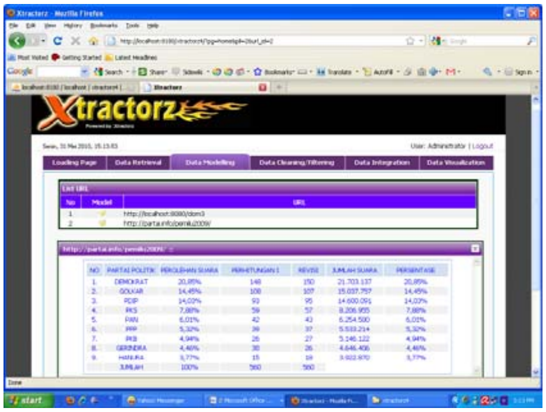

Xtractorz - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Most Visited Getting Started Latest Headlines
Xtractorz
Xtractorz
Loading Page
Data Retrieval
Data Modelling
Data Cleaning/Filtering
Data Integration
Data Visualization
User: Administrator | Logout
HTML
HEAD
TITLE
TR
TR
Done
start

List URL

No	Display	DOM Tree	URL
1			http://localhost:8080/dom3
2			http://partai.info/pemilu2009/

DOM Tree http://partai.info/pemilu2009/

Fig. 8: Data retrieval stage, Xtractorz renders a DOM tree based on the structure of extracted HTML tags



List URL

No	Model	URL
1		http://localhost:8080/dom3
2		http://partai.info/pemilu2009/

http://partai.info/pemilu2009/

NO	PARTAI POLITIK	PEROLEHAN SUARA	PERHITUNGAN 1	REVISI	JUMLAH SUARA	PERSENTASE
1.	DEMOKRAT	20,85%	148	150	21.703.137	20,85%
2.	GOLKAR	14,45%	108	107	15.037.757	14,45%
3.	PDIP	14,03%	93	95	14.600.091	14,03%
4.	PKS	7,88%	59	57	8.206.955	7,88%
5.	PAN	6,01%	42	43	6.254.580	6,01%
6.	PPP	5,32%	39	37	5.533.214	5,32%
7.	PKB	4,94%	26	27	5.146.122	4,94%
8.	GERINDRA	4,46%	30	26	4.646.406	4,46%
9.	HANURA	3,77%	15	18	3.922.870	3,77%
	JUMLAH	100%	560	560		

Fig. 9: Result of data (source) modeling stage

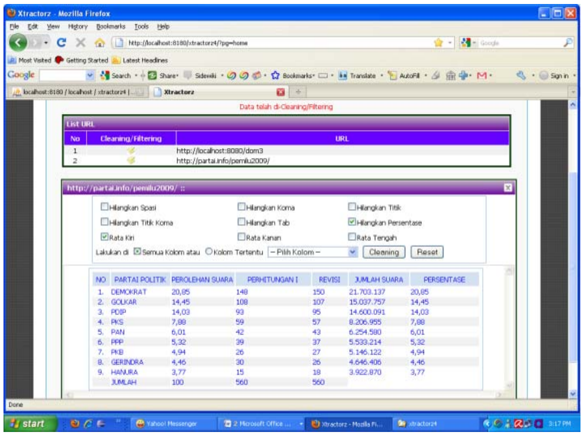
```
DataModelling($Url) {
   // To Take Data Title
 $Title=DataTitle($Url);
    // To Take Data Header Table
 $Header=Array();
 $Header=DataHeader($Url);
    // To Take and Count Columns
 $RowCells=Array();
 $Row=0;
 While (Not EOF()) {
    // To Take and Fill One Row
 for ($i=0;$i<&Column;$i++) {
     $RowCells[$Row,$Column]=DataRow($Row,
     $Column);
     // To Visualize
     Display($FillRow[$Row,$Column]);
     }
    $Row++;
  }
}
```

Data Cleaning/ Filtering: Data Cleaning in the process of Mashup building is performed to make corrections to the extracted data by correcting misspelling or establishing the format of data content which is required by the users (Fig. 10).

Data Cleaning/Data Filtering is used in order that the users can specify how the extracted data will be cleaned or adjusted to the format the users need. To implement this stage, Xtractorz refers to an algorithm and rules for cleaning which have been previously created.

In the Data Cleaning stage, the users can apply the cleaning rules provided in Xtractorz GUI for application in the columns in table. In its application, the users may apply the cleaning rules provided in Xtractor GUI for application in the columns in the table. The users may apply them by selecting the available options for cleaning. In this case, the cleaning options include deletion of space, deletion of semicolon, deletion of percentage symbol and left/right alignment.

```
DataCleaningFiltering($Url,$CleaningType) {
   // To Take Data Title
   $Title=DataTitle($Url);
   // To Take Data Header Table
   $Header=Array();
   $Header=DataHeader($url);
   // To Take and Count Columns
   $Column=Count($Header);
    // To Take Row Cells
   $RowCells=Array();
   $Row=0;
 While(Not EOF()) {
       // To Take and Fill One Row
 for ($i=0;$i<&Column;$i++) {
       $RowCells[$Row,$Column]=DataRow($Row,
       $Column);
       // To Clean and Filter
       $CleaningResult=CleaningFiltering
       ($RowCells($Row,$Column],$CleaningType);
       Display($CleaningResult);
       }
       $Row++;
    }
}
```


Xtractorz - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8180/xtractorz4/?pg=home
Most Visited Getting Started Latest Headlines
Google Search Share Sidewiki Bookmarks Translate AutoFill Sign in
localhost:8180 / localhost / xtractorz4 |...
Xtractorz
Done
start Yahoo! Messenger 2 Microsoft Office ... Xtractorz - Mozilla Fi... xtractorz4 3:17 PM

Data telah di-Cleaning/Filtering

List URL

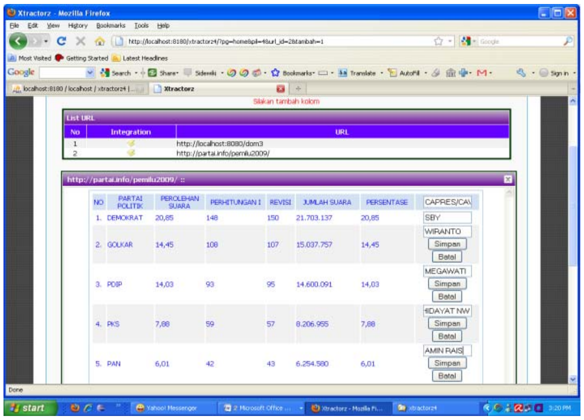
No	Cleaning/Filtering	URL
1		http://localhost:8080/dom3
2		http://partai.info/pemilu2009/

http://partai.info/pemilu2009/ ::

□Hilangkan Spasi □Hilangkan Koma □Hilangkan Titik
□Hilangkan Titik Koma □Hilangkan Tab ☑Hilangkan Persentase
☑Rata Kiri □Rata Kanan □Rata Tengah
Lakukan di ● Semua Kolom atau ○ Kolom Tertentu – Pilih Kolom – Cleaning Reset

NO	PARTAI POLITIK	PEROLEHAN SUARA	PERHITUNGAN I	REVISI	JUMLAH SUARA	PERSENTASE
1.	DEMOKRAT	20,85	148	150	21.703.137	20,85
2.	GOLKAR	14,45	108	107	15.037.757	14,45
3.	PDIP	14,03	93	95	14.600.091	14,03
4.	PKS	7,88	59	57	8.206.955	7,88
5.	PAN	6,01	42	43	6.254.580	6,01
6.	PPP	5,32	39	37	5.533.214	5,32
7.	PKB	4,94	26	27	5.146.122	4,94
8.	GERINDRA	4,46	30	26	4.646.406	4,46
9.	HANURA	3,77	15	18	3.922.870	3,77
	JUMLAH	100	560	560		

Fig. 10: Result of data cleaning stage


Xtractorz - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8180/xtractorz4/?pg=home&pil=4&url_id=2&tambah=1
Most Visited Getting Started Latest Headlines
Google Search Share Sidewiki Bookmarks Translate AutoFill Sign in
localhost:8180 / localhost / xtractorz4 |...
Xtractorz
Done
start Yahoo! Messenger 2 Microsoft Office ... Xtractorz - Mozilla Fi... xtractorz4 3:20 PM

Silakan tambah kolom

List URL

No	Integration	URL
1		http://localhost:8080/dom3
2		http://partai.info/pemilu2009/

http://partai.info/pemilu2009/ ::

NO	PARTAI POLITIK	PEROLEHAN SUARA	PERHITUNGAN I	REVISI	JUMLAH SUARA	PERSENTASE	CAPRES/CAV
1.	DEMOKRAT	20,85	148	150	21.703.137	20,85	SBY
2.	GOLKAR	14,45	108	107	15.037.757	14,45	WIRANTO Simpan Batal
3.	PDIP	14,03	93	95	14.600.091	14,03	MEGAWATI Simpan Batal
4.	PKS	7,88	59	57	8.206.955	7,88	HIDAYAT NW Simpan Batal
5.	PAN	6,01	42	43	6.254.580	6,01	AMIN RAIS Simpan Batal

Fig. 11: Results of data integration stage

Data integration: The purpose of Data Integration is to find the easiest way to combine and integrate data in the columns for the recently extracted data table and the data stored from the previous extraction process. A number of problems which occur are among others (a) to precisely determine the relevant data, the recently extracted data or the previously extracted one in the data repository for Data Integration, (b) to find the appropriate query technique to combine the data from the new sources and the previously extracted data (Fig. 11).

In this experiment, Xtractorz tries to solve the problems by using the table (table constraints). The users fill out the empty cells in the available columns, by selecting the values or attributes in the available list. After the users select a value, Xtractorz then stores and counts the constraints of the number of data sources which have just been put in the new columns. To implement Data Integration stage, Xtractorz refers to an algorithm and rules for integrating the extracted data table:

```
DataIntegration($Url,$OtherData) {
```

```
// To Take Data Title
$Title=DataTitle($Url);
// To Take Data Header Table
$Header=Array();
$Header=DataHeader($Url);
// To Take and Count Columns
$Column=Count($Header);
// To Take Row Cells
$RowCells=Array();
$Row=0;
While(Not EOF()) {
// To Take and Fill One Row
for ($i=0;$i<&Column;$i++) {
$RowCells[$Row,$Column]=DataRow($Row,
$Column);
      // To Integrate Data
      $IntegrationResult=Integration($RowCells[$Row,
      $Column],$OtherData[$Column]);
      Display($IntegrationResult);
      }
      $Row++;
  }
}
```

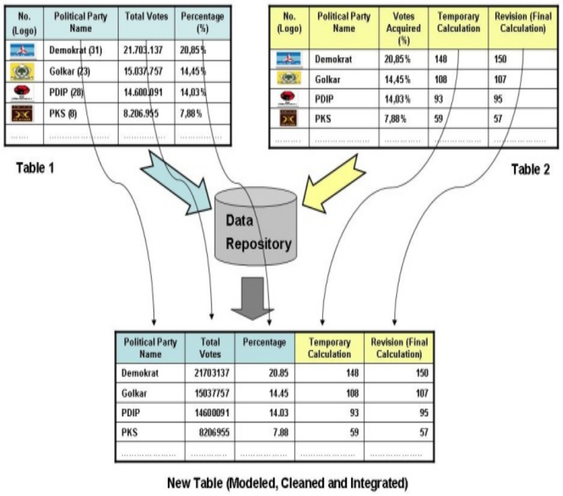
The case study in this study is the extracted table data from the targeted web page http://partai.info/pemilu2009, in which Xtractorz has to integrate two tables as presented below.

To demonstrate how Xtractorz handles the process of Data Integration, the Xtractorz algorithms first show the DOM tree of both tables and then it performs the stages in Mashup building (to extract, model and clean). Later Xtractorz put all the extracted data in the data repository.

Initially, in the data repository there were only 3 (three) extracted data sources from table 1, namely: “Political Party”, “Number of Votes” and “Percentage”. Then, after Xtractorz completes the last stage in Mashup building, the users can do the next process of web table extraction on table 2 until the users obtain a set or a group of data combination from the extraction of tables 1 and 2 stored in the data repository.

In this experiment, Xtractorz also shows to the users that the first lines of columns 1, 2 and 3 contain the data and value {“Demokrat”, “21703137”, “20.85”}. The reason is to maintain the integrity of those columns, such as cell 1 (line 1, column 1) is the association of “Political Party” attribute or tag and cell 2 (line 1, column 2) “Number of Votes” and cell 3 (line 1, column 3) “Percentage”.

In this case, Xtractorz creates the attribute “Political Party” as the primary key and uses it for the subsequent calculation process. The process is called



No. (Logo)	Political Party Name	Total Votes	Percentage (%)
	Demokrat (31)	21.703.137	20,85%
	Golkar (23)	15.037.757	14,45%
	PDIP (28)	14.600.091	14,03%
	PKS (8)	8.206.955	7,88%
....	....	....	....

Table 1

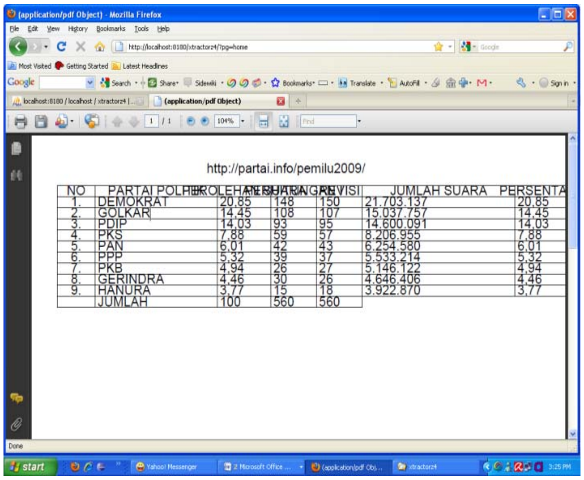
No. (Logo)	Political Party Name	Votes Acquired (%)	Temporary Calculation	Revision (Final Calculation)
	Demokrat	20,85%	148	150
	Golkar	14,45%	108	107
	PDIP	14,03%	93	95
	PKS	7,88%	59	57
....	....	....	....	....

Table 2

Political Party Name	Total Votes	Percentage	Temporary Calculation	Revision (Final Calculation)
Demokrat	21703137	20.85	148	150
Golkar	15037757	14.45	108	107
PDIP	14600091	14.03	93	95
PKS	8206955	7.88	59	57
....	....	....	....	....

New Table (Modeled, Cleaned and Integrated)

Fig. 12: Data Integration scheme from the 2009 general



(application/pdf Object) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8180/xtractorz4/?pg=home
Most Visited Getting Started Latest Headlines
Search Share Sidewiki Bookmarks Translate AutoFill Sign in
localhost:8180 / localhost / xtractorz4 |...
(application/pdf Object)
1 / 1
104%
Find
Done
start

http://partai.info/pemilu2009/

NO	PARTAI POL	PEROLEHAN	PERHITUNGAN	REVISI	JUMLAH SUARA	PERSENTA
1.	DEMOKRAT	20,85	148	150	21.703.137	20,85
2.	GOLKAR	14,45	108	107	15.037.757	14,45
3.	PDIP	14,03	93	95	14.600.091	14,03
4.	PKS	7,88	59	57	8.206.955	7,88
5.	PAN	6,01	42	43	6.254.580	6,01
6.	PPP	5,32	39	37	5.533.214	5,32
7.	PKB	4,94	26	27	5.146.122	4,94
8.	GERINDRA	4,46	30	26	4.646.406	4,46
9.	HANURA	3,77	15	18	3.922.870	3,77
	JUMLAH	100	560	560		

Fig. 13: Result of data visualization stage (.pdf table)

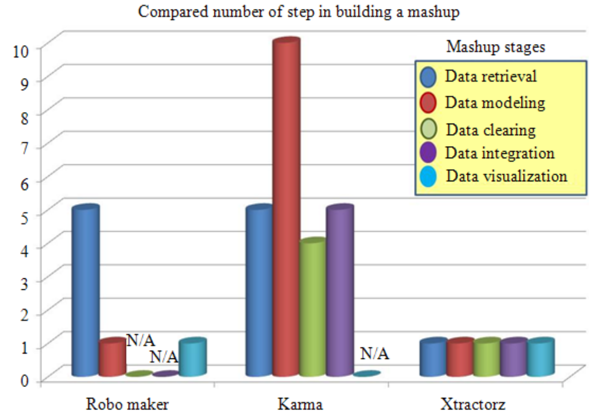


Fig. 14: Result of comparions (N/A: Not Available)

vertical constraint in which the values in the same columns must be associated with the same attributes. In this case, only two table data sources (Table 1 and Fig. 14) with the same columns for "Political Party" attribute, so Xtractorz formulates a vertical constraint-based query to make a list of suggestions in the "Political Party" column (Fig. 12).

Data visualization: Data visualization is the last stage in building a Mashup, presenting the data in the data repository in the visual forms. In this stage, the users take the final data which have been processed or the extracted final data and present the data in various visual formats such as XML table, map, data marts, web services, graphs. Furthermore, the visual presentations can also be used to meet the need of external software and other forms of services. One of the visual forms is a pdf table the content of which can be used for data exchange (Fig. 13).

To implement this stage, Xtractorz refers to an algorithm and rules for displaying the output of the stored data table in the data repository.

```
DataVisualization($Url,$ExportType) {
 // To Take Data Title
 $Title=DataTitle($Url);
 // To Take Data Header Table
 $Header=Array();
 $Header=DataHeader($Url);
 // To Take and Count Columns
 $Column=Count($Header);
 // To Take Row Cells
 $RowCells=Array();
 $Row=0;
 While(Not EOF()) {
    // To Take and Fill One Row
    for ($i=0;$i<&Column;$i++) {
    $RowCells[$Row,$Column]=DataRow($Row,
    $Column);
    // To Visualize
     Export($RowCells,$ExportType);
     }
    $Row++;
  }
}
```

RESULTS

Performance evaluation and results: For this study, we conduct the evaluation by comparing Xtractorz application system and the similar application systems: RoboMaker (OpenKapow) and Karma.

The reason RoboMaker is selected because in its application, the users can create and debug a robot which can be ordered to do data searching and data collection, by extracting various objects or data from websites which are made a target for its data sources. In addition, the robots in RoboMaker can also do "clipping" of one or more parts in one webpage (HTML) to be presented in the context of different presentations, such as a presentation in a website portal.

RoboMaker also provides a programming feature which is very beneficial for its users called interactive programming visual feature which is equipped with the ability to perform full debugging as well as an easy access to the online help system. Therefore, designing an experiment scenario between Xtractorz and RoboMaker is a challenge for us.

However, from the perspective of application system, RoboMaker cannot cover all of the problems in the implementation of web table extraction and stages in Mashup building. The situation results from the fact that RoboMaker mainly focuses on an area of data extraction from a single website source and that the output of the data extraction is stored in the form of RSS feed Using the method in writing mathematical notion.

In addition, RoboMaker is an application system with a high learning curve, so in order to use RoboMaker, the users must first read all of the available tutorials or manuals. The users can also carry out several tryout examples and they also need to understand the concept of computer program creation.

Meanwhile, another application system, Karma, contribution to this evaluation is through an approach to Data Integration technique including: (a) Karma does not require the users to have prior detailed knowledge on which data table in a website which becomes the data source, (b) Karma can make the users believe the value the valid data contained in the data table, (c) Karma can make a query which is consistent with the data in the website and its position always returns to its initial value.

Karma also has an approach in the stages of Mashup building, that is by combining the four techniques for information integration, which are normally done separately, into one unified framework Using the method in writing mathematical notion (Tuchinda *et al*., 2008).

The users can also build Mashup without having neither to write the scripting of the computer language nor to have prior understanding of the concept of computer programming language, by providing examples in the form of presentations of the final output of an operation which is wanted by the users.

On the other side, the Xtractorz application system which we propose also has been equipped with the ability to perform web table extraction activity visually, while at the same time combining it with the process in the stages in Mashup building, through the retrieving, modeling, cleaning and integrating the data sources. In the application of Xtractorz application system and its GUI, the user does not have to know how to create a program using a computer language.

Users' evaluation: For the task given in this evaluation, we assign an expert in computer to understand and comprehend each of the application systems (RoboMaker, Karma and Xtractorz) used to perform the task given. Furthermore, the success in each task can be measured using "step" as a unit of measurement.

All of the application systems, RoboMaker, Karma and Xtractorz, are given the same task, where the outputs are in the form of steps, they are:

- First, the three systems must be used on a real time basis on the website or URL which becomes the target source and in this case the target source is http://partai.info/pemilu2009.
- The next stage is to perform a web table extraction on the available data on the data tables on that website (in this research 2 data tables are used): Political Party, Number of Votes, Percentage, Number of Seats, with the features available in its User Interface
- The final step is all of the application systems complete the stages in building Mashup, from data retrieving, modeling, clearing/filtering, integrating to Data Visualization (displaying)

For that purpose, the computer experts conduct the web table extraction and complete all the stages in Mashup building. Each task is designed to represent each specific problem which occurs in the activity. The result of the measurement of each area of assignment is also presented in "step" as a unit of measurement.

Performance results: The result of the evaluation of web table extraction and Mashup building using the three application systems can be seen in Fig. 14. The tables show the number of steps obtained from each stage in Mashup building.

The number of "steps" produced by the three application systems is then compared and it can be started by accessing the targeted website http://partai.info/pemilu2009/. Except for the task, Karma uses the data available in a paper Using the method in writing mathematical notion (Tuchinda *et al*., 2008). We have designed that the web data tables (2 tables) on the target website is extracted by the three application systems in the same manner.

DISCUSSION

Our performance experiment shows that our proposed algorithm which is implemented in the Xtractorz application system is more efficient than RoboMaker and Karma. The result indicates that apparently RoboMaker needs 7 "steps" in the process of web table extraction and the stages in building a Mashup, but it has 2 stages with the N/A (Not Available) results, in the Data Cleaning and Data Integration stages. Meanwhile, Karma (*) needs 24 "steps" to complete all the stages in the similar task (one stage with N/A result) as indicated in Karma's evaluation (Tuchinda *et al*., 2008). On the other hand, Xtractorz needs only 5 "steps" with no N/A results, or fewer than the other system do, to complete all the stages in building a Mashup, starts from data retrieval, modelling, cleaning, integration and visualization.

This experiment result relates to the original objectives of this study, which is to perform a web table extraction process and building a Mashup stages, from web pages containing data tables. And, this experiment result is also correlates with the recent reviews, where due to its high complexity, the implementation of web table extraction and building a Mashup stages are normally performed separately (or partially), by different similar application systems available in the market, such as Yahoo Pipes, Mashmaker, RoboMaker, Lixto, Dapper, etc.

The significant or unique finding of this research is the discovery of the proposed algorithm which is implemented in the prototype of Xtractorz. It can be seen that our proposed algorithm is capable to sorts and indexes the HTML codes (tags) of the retrieved web tables into a DOM tree form, which is formed automatically by the Xtractorz.

CONCLUSION

It can be concluded that the Xtractorz can give a positive contribution in terms of algorithm technique and also give a new approach method to web table extraction and building a Mashup stages.

The Xtractorz is also capable to colaborate the process of web table extraction and building a Mashup in a one unified GUI framework, so the users can easily and comfortably perform the extraction of web tables without having to have the expertise in creating a computer program.

Another important conclusion, which is the state-of-the-art of our research, is the discovery of the proposed algorithm implemented in the Prototype of Xtractorz application system as a tools to extracts and indexes web table HTML codes into a DOM tree form automatically.

For further works, we have planned to complete the available features in the Xtractorz GUI with JQuery facility and the NLP (Natural Language Processing) to further collaborate semantic meaning of the attributes or the table contents (cells) which are extracted (Ilieva and Ormandjieva, 2005).

We also try to integrate the semantic modeling with the concept of Ontology (Sari and Ayuningtyas, 2010) as an alternative technique of DOM tree modeling, in order to find the best solution to implement web table extraction process and building a Mashup.

REFERENCES

Barinka, L. and I. Jelinek, 2009. Data Extraction by Visual Matching. Proceedings of the 10th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, June 18-19, ACM New York, NY, USA., pp: 1-653.

Baumgartner, R., S. Flesca and G. Gottlob, 2001a. Declarative Information Extraction, Web Crawling and Recursive Wrapping with Lixto. Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning, (LPNR' 01), Springer-Verlag London, UK., pp: 21-41.

Baumgartner, R., S. Flesca and G. Gottlob, 2001b. Visual Web Information Extraction with Lixto, Proceedings of the 27th International Conference on Very Large Data Bases, (VLDB' 01), Morgan Kaufmann Publishers Inc. San Francisco, pp:119-128.

Cafarella, M.J., A. Halevy, Z.D. Wang, E. Wu and Y. Zhang, 2008. Web tables: Exploring the power of tables on the web. Proceeding of the VLDB Endowment, VLDB Endowment, New Zealand, pp: 538-549. DOI: 10.1145/1453856.1453916

Chamberlin, D., 2003. XQuery: A query language for XML. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, (ACMICMD'06), ACM New York, NY, USA., pp:682-682.

Dehuri, S., A.K. Jagadev, A. Ghosh and R. Mall, 2006. Multi-objective genetic algorithm for association rule mining using a homogeneous dedicated cluster of workstations. Am. J. Applied Sci., 3: 2086-2095. DOI: 10.3844/ajassp.2006.2086.2095

Gatterbauer, W., P. Bohunsky, M. Herzog, B. Krupl and B. Pollak, 2007. Towards DOMain Independent Information Extraction from Web Tables. Proceeding of the International World Wide Web Conference Committee (IW3C2), May 8-12, ACM, Banff, Alberta, Canada, pp: 71-80.

Gultom, R.A.G., R.F. Sari and B. Budiardjo, 2010. Implementing web data extraction and making mashup with xtractorz. Proceedings of 2010 IEEE 2nd International Advance Computing Conference, Feb. 19-20, IEEE Xplore Press, Patiala, pp: 385-393. DOI: 10.1109/IADCC.2010.5422921

Heier, J.E., 2008. Mashup the OODA Loop, Mitre Technical Report, Project No: 0708M290-IT, http://www.dodccrp.org/events/13th_iccrts_2008/CD/html/papers/058.pdf

Hergli, M., J. Baili, F. Bouslama and K. Besbes, 2005. A new compressing ultrasonic data algorithm based on wavelets. Am. J. Applied Sci., 2: 1615-1618. DOI: 10.3844/ajassp.2005.1615.1618

Huynh, D., S. Mazzocchi and D. Karger, 2005. Piggy bank: Experience the semantic web inside your web browser. Lect. Notes Comput. Sci., 3729: 413-430. DOI: 10.1007/11574620

Ilieva, M.G. and O. Ormandjieva, 2005. Automatic transition of natural language software requirements specification into formal presentation. Nat. Lang. Process. Inform. Syst., 3513: 427-434. DOI: 10.1007/11428817_45

Knoblock, C.A., K. Lerman, S. Minton, I. Muslea, 2003. Accurately and reliably extracting data from the web: A machine learning approach., IEEE Data Eng. Bull., 23: 33-41.

Liu, Z., W.K. Ng, F. Li and E.P. Lim, 2002. A visual tool for building logical data models of websites, Proceedings of the 4th international workshop on Web information and data management, Nov. 08-08, ACM New York, NY, USA., pp: 92-95. DOI: 10.1145/584931.584951

Mamat, M.R., M. Rizon and M.S. Khanniche, 2006. Fault detection of 3-phase VSI using wavelet-fuzzy algorithm. Am. J. Applied Sci., 3: 1642-1648. DOI: 10.3844/ajassp.2006.1642.1648

Sari, F.S. and N. Ayuningtyas, 2010. Implementing of web ontology and semantic application for electronic journal citation system. J. Emerg. Technol. Web Intell., 2: 34-41. DOI: 10.4304/jetwi.2.1.34-41

Singh, S.K., S. Sabharwal and J.P. Gupta, 2010. An event-based methodology to generate class diagrams and its empirical evaluation. J. Comput. Sci., 6: 1301-1325. DOI: 10.3844/jcssp.2010.1301.1325

Sleit, A., W. Al-Mobaideen, S. Al-Areqi and A. Yahya, 2007. A dynamic object fragmentation and replication algorithm in distributed database systems. Am. J. Applied Sci., 4: 613-618. DOI: 10.3844/ajassp.2007.613.618

Tuchinda, R., P. Szekely and C.A. Knoblock, 2008. Building Mashup by Example. Proceeding of the2008 International Conference on Intelligent User Interfaces, (ICIUI'08), ACM, New York, pp: 139-148.

Vijayalakshmi, S. and V. Mohan, 2010. Mining sequential access pattern with low support from large pre-processed web logs. J. Comput. Sci., 6: 1293-1300. DOI: 10.3844/jcssp.2010.1293.1300

Wong, J. and J.I. Hong, 2007. Making mashups with marmite: Towards end-user programming for the web. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (CHFCS' 07), ACM New York, NY, USA., pp: 1435-1444. DOI: 10.1145/1240624.1240842