Patients Data Extraction from DDO Files Extension using Rule-based Classification Algorithm

Patients Data Extraction from DDO Files Extension using Rule-based Classification Algorithm

Nur Alam[1*], Mukhlis Amin[2], Hazriani[3], and Yuyun[4]

[1,2] BBPSDMP Kominfo Makassar, Ministry of Communication and Information Technology Republic of Indonesia, Prof. Abdurrahman Basalamah stret 25, Makassar Indonesia

[3,4] STMIK Handayani, address, Indonesia

Email: nur.alam@kominfo.go.id

Abstract. Arrangement patient's report process of Elim Rantepao hospital is especially complicated because patient data was obtained from FCR Console Prima what saved as DDO file extension as Unicode string. This problem makes understanding and select data manually become difficult. This research tries to give a solution on how to arrange reports easily through application tools developing. This research uses a rule-based classification algorithm to extract Patient Name, Age, ID Number, Date, and time. Applying this algorithm, increase system performance up to 96,6%. This application is very accurate as long as the file DDO has not broken.

1. Introduction

The use of information technology is increasingly penetrating all aspects of human life, including in the field of medicine. The basic example of this utilization is the application of the Hospital Information System to process hospital data in general. On a smaller scale, the use of IT in the medical field is one of which is used in terms of computerized x-ray systems to support the x-ray process of hospital patients. Elim Hospital is one of the hospitals in Rantepao, North Toraja Regency, which also provides a computerized x-ray system for patients.

One of the problems faced by Elim's hospital radiology team is the complexity of providing patient reports using x-ray machines. It is because the built-in X-ray machine application does not yet have a feature that can display/provide accumulative data regarding the profile and number of patients. The data for each patient using each X-ray machine is saved in a different file. So, we need a system to recap patient data from these files. The process of creating reports can be done by opening the files one by one or by retyping them, but it will take a very long time. Another complication is that the patient data stored in the DDO file is mixed up with a very irregular Unicode string (font), making it very difficult to sort patient data from the file structure manually. The X-ray file is read in Unicode string font format when accessed or opened in a programming language or other text processing applications such as Microsoft Word or notepad.

Unicode fonts (also known as Universal Coded Character Set (UCS) fonts and Unicode typefaces) are computer fonts that contain various of characters, letters, numbers, glyphs, symbols, ideograms, logograms, etc., which are collectively mapped to the Universal Character Set standard, come from various languages and scripts from around the world. Unlike most conventional computer fonts, which

are specific to a particular language or legacy character set and contain only a fraction of the UCS characters, these fonts attempt to include thousands of possible glyphs, so that they can be used as a single typeface in multilingual documents. For example, if an image file or .exe file is opened in the Notepad application, the image or .exe file will be displayed with various Unicode characters that are difficult to understand.

One alternative to separate patient data from Unicode characters contained in the DDO file is to extract patient data into a new file with a more regular format and then present it in the form of a report. Information extraction can be defined as a process to find structured information from unstructured or semi-structured documents. The extraction of information is one part of Natural Language Processing [1]. The extraction process is part of text mining, which is a process to retrieve information from the existing text.

Several studies involving the text mining process include research to extract HTML tables on the web by developing algorithms for the extraction of three table forms [2]. Research that is more specific to extract information on web pages [3] or the Development of Wrapper Applications for Data Extraction on Web Pages Using Python [4].

Other research on text extraction was also carried out for automatic text extraction from Indonesian web pages to help speed up corpus development. The extraction process is done by first separating the HTML syntax from the non-HTML syntax [5]. Research that focuses more specifically on algorithms is keyphrase extraction from a site using the Key Exchange Algorithm (KEA) by filtering based on predetermined keywords [6]. The algorithm that most used in string matching is the Brute Force algorithm because of its simplicity and simplicity. The brute force algorithm performs string matching with existing scripts by matching the characters that are searched from left to right or from beginning to end until the character is found or not found. The Brute force algorithm using in [7] and [8].

This study classified patient data into a more structured format to facilitate the preparation of radiology patient reports at Elim Rantepao Hospital. The information consists of the patient's name, age, ID number, date, and time of the X-ray. This study is an extension of previous studies that extracted patient data with DDO extensions using the brute force algorithm [9]. The research resulted in an accuracy of 91.6%. In the development of this research, we use a rule-based classification algorithm by performing word/string matching based on certain patterns. The purpose of this research is to build an application that can extract ddo files with higher accuracy than previous studies. Rule-based classification is one extraction method commonly used. The use of this algorithm is expected to be suitable for data extraction containing Unicode strings so that it can improve system performance. This algorithms have been applied in extracting the core data of Audit Report document [10]. Its also has a very good performance even though the data has high uncertainty [11]. Several studies have discussed data extraction [12], [13], [14], [15], [16]. However, none of them have the address to the Unicode string issue.

2. Text Mining

Text mining (also called text data mining) is the process of extracting information from existing text. Text mining looks for patterns in the text in unstructured natural languages such as books, emails, articles, web pages, etc. Activities that are usually carried out in text mining are preprocessing and classification. Preprocessing is a process or method that must be done so that the data can be used in the core process of data mining. This preprocessing can be done by a variety of different algorithms that start by examining the document and processing it for patterns by filtering for similarities or by adding other characteristics. Several methods commonly used in preprocessing include POS tagging, stemming, full parsing, or swallow parsing.There are 3 processes that are usually involved in a text mining activity:

1. Characterization of data text, is structured by processes such as parsing and is entered into a database. This stage is also known as preprocessing.
2. Data mining from existing data is then searched with a certain algorithm to get a pattern from that data.

3. Data visualization. The search results will be interpreted and issued as an output in an easily understandable form.

3. Research Method

The extraction of patient data is carried out by building a system that classifies data using a rule-based classification method. The first step is to perform initial processing of the DDO file by searching the patterns for each component (Patient Name, Age, Id Number, Date, and Time of X-ray) based on keywords. The result of this process is a pattern for each part of the DDO file.

3.1 Preprocessing

The Part-of-Speech (POS) Tagging method is used for preprocessing. POS tagging was chosen because it was considered appropriate and could be combined with rule-based classification to obtain the desired pattern. POS Tagging used in this study utilizes the Hidden Markov Model (HMM) method that is an extension of the Markov chain where the state cannot be observed directly (hidden) but can only be observed through a set of other observations. Markov Chain is usually used to calculate the probability of an observable sequence of events. HMM itself is useful for getting a sequence of events but cannot be observed.

In HMM, the tag sequence cannot be observed directly or is commonly called the hidden state. A direct observation (observed state) can only be made of word order. From the word order, the most appropriate sequence of tags must be found. HMM allows modeling of a system containing an interrelated observed state and hidden state.

3.2 Classification

After getting the pattern, the next process is to find the position of the part in the file. If the position of each section is known (based on keywords that mark the beginning and end of a part of the DDO file), then classification is carried out using rule-based classification by utilizing the intended pattern in carrying out learning and extracting training data and test data. Classification is the learning process of a function or model against a set of training data so that the model can be used to predict the class from the test data [17].

This method was chosen because it was considered the most appropriate with the structure of the DDO file. DDO file is a semi-structured file. According to Feldman, semi-structured documents can be defined as documents that have elements or a consistent format where each type of part of the document can be recognized easily. Other examples of semi-structured documents are HTML, pdf files, and word files that have template or style sheet restrictions. DDO files are categorized as semi-structured because each file contains information in the form of patient data needed.

3.3 Application Design

Preprocessing and classification stages are outlined in the application design and then translated into a programming language to create extraction tools. To facilitate application creation, the application workflow to be built is designed in advance. This workflow will make it easier to design and determine the module or function to be used. The module or function is first designed in the form of a programming algorithm so that it is easy to interpret in a programming language. To facilitate application development, another design that needs to be done is the application interface and the output that will be produced.

We need a system workflow and its stages to facilitate translation of the processes carried out in the system. The data extraction process begins by collecting all files with the DDO extension generated from the radiological examination process into a specific folder. Furthermore, each file in the folder is identified one by one by accessing/opening the file using a programming language based on their respective characteristics. The Rule-based classification algorithm plays a role in the identification process. This algorithm matches these characteristics with the text content of each file accessed. By accommodating the search results and the process of eliminating some unnecessary characters from

the search results, it will produce radiological patient data that will be collected in the application to facilitate the extraction process into a file (.doc or .txt). The system workflow is shown in Figure 1.

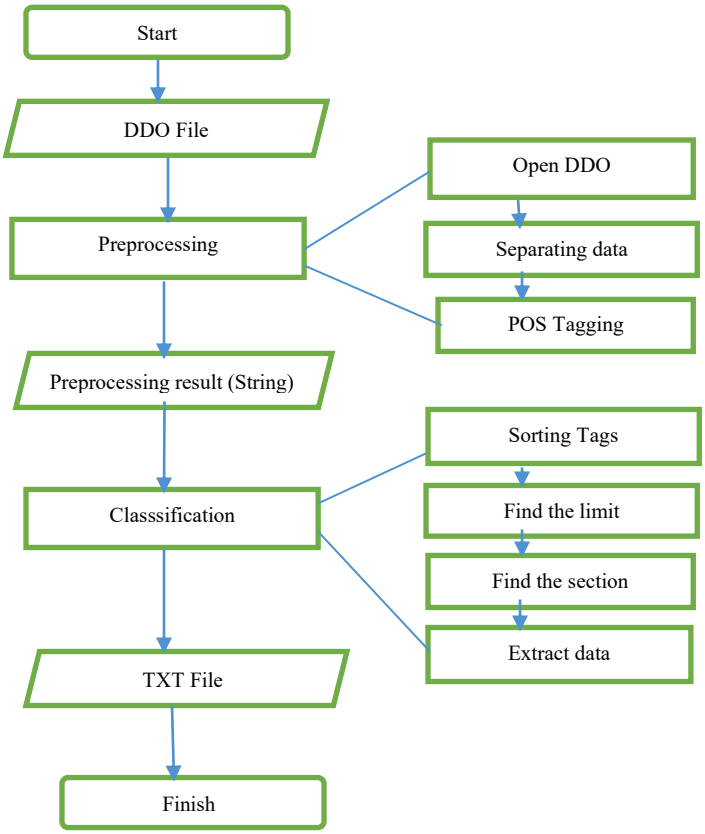

Figure 1. System workflow

4. Result and Discussion

4.1. Aplication Interface

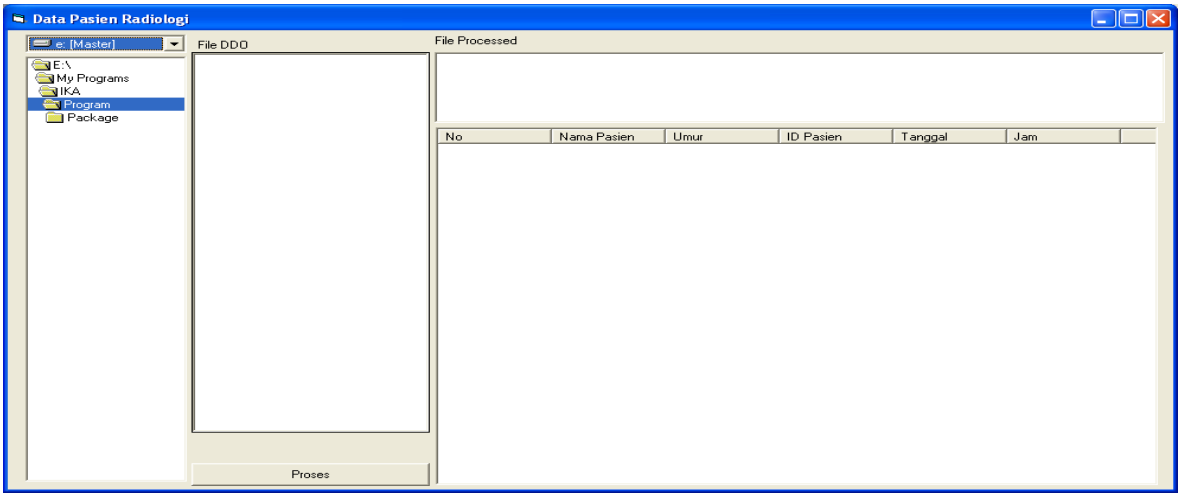

Figure 2. User Interface

The application interface structure consists of several features developed in particular to support all extraction processes. It starts from selecting the folder used to collect DDO files, List of DDO files, lists of extracted patient data, and the process of storing patient data into new files in TXT format. The design is designed in a blank form, as shown in Figure 2.

4.2. Analysis of Data Characteristics

As previously discussed, the patient data in the DDO file is composed of Unicode string fonts which are very difficult to understand. However, each patient data in the file structure has characteristics that allow us to identify it with the help of a specially designed application. We can see the patient data structure in the DDO file in Figure 3.

Figure 3. Example of a file structure with a DDO extension.

The patient data in these images looks quite complicated to understand and sort out manually. Therefore, the identification of the data characteristics must be made for each data. Before identifying the data characteristic, we have to separate the data first. We split the data core with the Unicode string. The result of the separating data process is shown in Figure 4. After separating data, We identified the special characteristics of the data by determining the initial characteristics, final characteristics, or character length (data width).

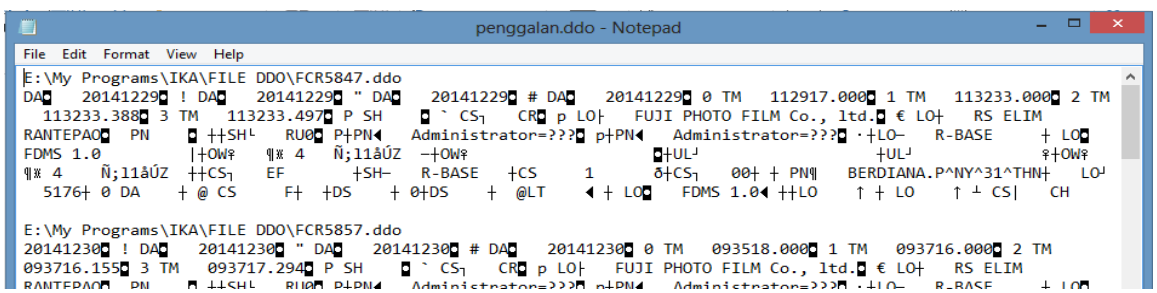

Figure 4. Example of a file after separating data

4.3. Data Training

The training data used in this study were 50 files. After doing the training data, we know the characteristics of each patient identity attribute in the DDO file, as shown in Table 1. This atribute becomes a classification rule.

Tabel 1. Characteristics of patient data in DDO files

No	Type of Data	Initial characteristics (begins with a character)	Final characteristics (ends with a character)	Data Length
1	Name	"PN"	"LO————————"	~
2	Age	"LO————————"	"^"	~
3	ID	"LO———————— "		4 character
4	Date	"DA "		8 character
5	Times	"TM "		6 character

4.4. Data Testing Result

The name and age of the patient are dynamic long data, so it is necessary to identify the initial and final character of the data. Patient ID, date, and time of X-rays are length static data. So, it is enough to identify it by determining the character as the marker of each of these data. We can see the identification results based on the rule, as shown in Table 1. The results of this identification can be known by looking Figure 5. These results are displayed through the interface that is shown in Figure 6.

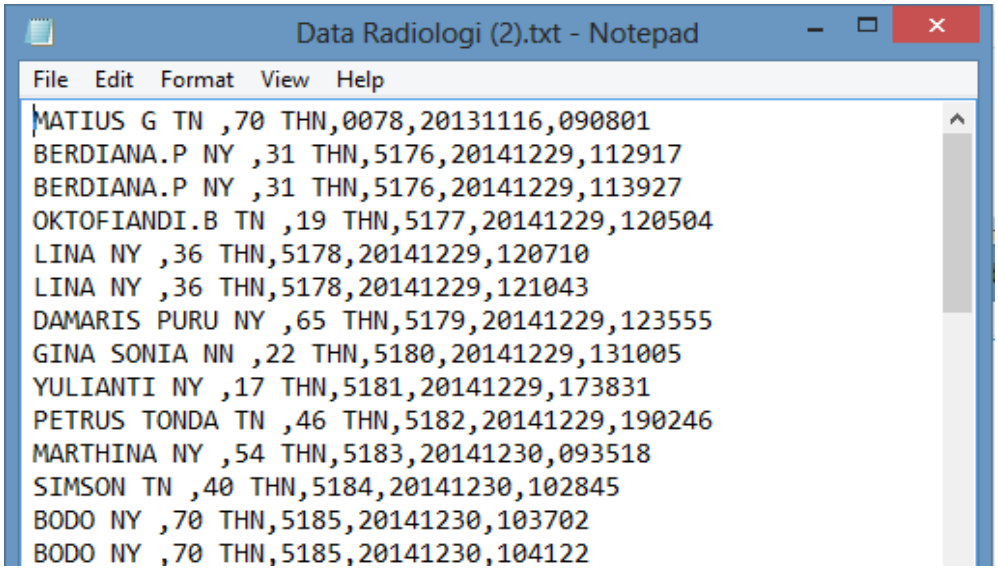

Figure 5. Data extraction result

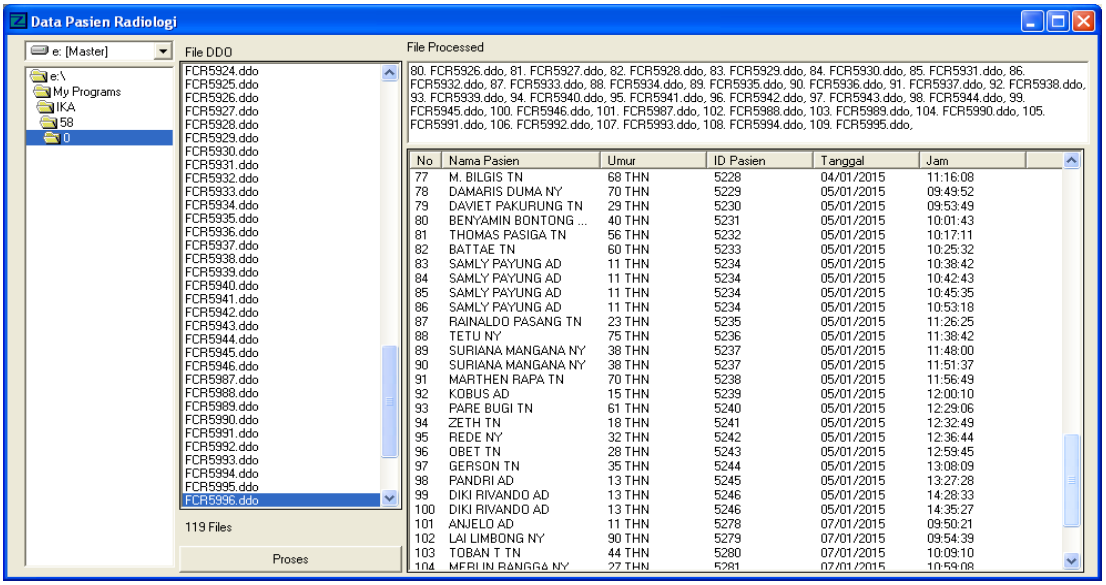

Figure 6. DDO file extraction application output

We use 119 DDO files as test data, 115 data were extracted successfully. In other words, system accuracy is 96.6%. This result is better than the previous work using the brute force algorithm. We have to check the files that failed to extract. So, we check it manually. We found that the data failed to extract because the DDO file does not store patient data. By knowing these causes, accuracy or sensitivity testing is no need to be done using statistical or computational methods. Thus it can be said that this application suitable for use to extract patient data accurately.

5. Conclusion

This study has identified the characteristics and patterns of patient data stored in the DDO file generated by the X-ray machine. From the test results, it can be seen that patient data is stored in a semi-structured manner with a uniform pattern based on the type of data so that the data can be extracted accurately using a rule-based classification algorithm. Finding the patterns and characteristics of each patient data attribute will facilitate the development of a more accurate and efficient patient data extraction application. Applying this algorithm, increase system performance up to 96,6%. This application is very accurate as long as the file DDO has not broken. Nevertheless, this study was limited to patient data extraction only. Further research can develop applications for reading X-ray results.

Acknowledgement

The author thanks Ika Adriana and the entire Elim Rantepao hospital management for the data support provided so that this research can be completed properly. Alam and Amin have a greater contribution than other authors.

References

[1] J. Jiang, Information Extraction from Text. In Mining Text Data,, China: Springer, 2012.

[2] D. Purnamasari, I. W. S. Wicaksana and S. Ruhama , "Algoritma untuk ekstraksi tabel HTML di web," in *Konfrensi Nasional Sistem Informasi 2012*, Bali, 2012.

[3] V. Hutagalung, "Pembuatan aplikasi ekstraksi informasi pada web," Gunadarma, Depok, 2014.

[4] L. Y. Banoasari, D. Pamungkas, I. W. S. Wicaksana and B. A. Mutiara, "Pengembangan Aplikasi Wrapper Untuk Ekstraksi Data Pada Halaman Web Dengan Menggunakan Python," Universitas Gunadarma, Depok, 2008.

[5] F. and Z. Abidin, "Ekstraksi Teks Otomatis dari Halaman Web Berbahasa Indonesia Guna Membantu

Mempercepat Penyusunan Korpus," ejournal.uin-malang.ac.id, Malang, 2010.

[6] G. T. A. Hermawan, D. J. Saputra and J. Santoso, "Ekstraksi Keyphrase dari Suatu Situs Dengan Algoritma KEA," in *Konferensi Nasional Inovasi dalam desain dan Teknologi*, Surabaya, 2011.

[7] B. W. Santoso, F. Sundawa and M. Azhari, "Implementasi Algoritma Brute Force Sebagai Mesin Pencari (Search Engine) Berbasis Web pada Database," *Jurnal Sisfotek Global,* vol. 6, no. 1, pp. 1 - 8, 2016.

[8] F. E. M.A, A. Hanifa and N. Riyanto, "Implementasi Algoritma Brute Force Dan Fitur Location Based Service (Lbs) Pada Aplikasi Kumpulan Doa Harian Berbasis Android," *Pseudocode,* vol. 1, no. 2, pp. 105-115, 2014.

[9] N. Alam, "Application Design of Radiology Patient Data Extraction from DDO Files Extension," in *Prosiding Seminar Nasional Komunikasi dai Informatika ke-2*, Makassar, 2016.

[10] A. Ismaya, "Algoritma Ekstraksi Informasi Berbasis Aturan," *JNTETI,* vol. 3, no. 4, pp. 242 - 246, 2014.

[11] B. Qin, Y. Xia, S. Prabhakar and Y. Tu, "A Rule-Based Classification Algorithm for Uncertain Data," in *IEEE International Conference on Data Engineering*, 2009.

[12] Z. Jensen, E. Kim, S. Kwon, T. Z. H. Gani, Y. Román-Leshkov, M. Moliner, A. Corma and E. Olivetti, "A Machine Learning Approach to Zeolite Synthesis Enabled by Automatic Literature Data Extraction," *ACS Central Science,* vol. 5, pp. 892-899, 2019.

[13] H. Pedder, G. Sarri, E. Keeney, V. Nunes and S. Dias, "Data extraction for complex meta-analysis," *Systematic Reviews,* vol. 5, no. 212, pp. 1-6, 2016.

[14] M. Nasr, H. Fahmy and M. Thabet, "Deep Web Data Extraction," in *14th International Conference on Computer Engineering and Systems (ICCES)*, Cairo, 2019.

[15] D. Jung, W. Kim, H. Song, J.-i. Hwang, B. Lee, B. Kim and J. Seo, "ChartSense: Interactive Data Extraction from Chart Images," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017.

[16] S. S. Paliwal, V. D, R. Rahul, M. Sharma and L. Vig, "TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images," in *International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, 2019.

[17] S. K. Palanisamy, Association Rule Based Classification, Worcester: Worcester Polytechnic Institute, 2006.