Framework of JawaTeX

Ema Utami[1], Jazi Eko Istiyanto[2], Sri Hartati[3], Marsono[4], Ahmad Ashari[5]

[1]Information System Major of STMIK AMIKOM Yogyakarta
Ring Road Utara ST, Condong Catur, Depok Sleman Yogyakarta
[2,3,5]Doctoral Program in Computer Science of Postgraduate School Gadjah Mada University
Graha Student Internet Center (SIC) 3rd floor
Faculty of Mathematic and Natural Sciences Gadjah Mada University
Sekip Utara Bulaksumur Yogyakarta. 55281
[4] Sastra Nusantara Major of Culture Sciences Gadjah Mada University
Humaniora ST No.1, Bulaksumur, Yogyakarta

Summary
Transliteration is a substitution letter by letter from one alphabet to another, free from how to actually speak those characters or it can be called a letter substitution or transliteration. Currently there are already two Javanese characters of true type font. To use the fonts in writing Javanese characters, users must have knowledge about how to read and write Javanese. No researcher has already developed algorithm to handle writing Javanese character for x and q Latin characters, aritmetic operand, special symbols (except period, coma and double quote), multiple consonant (more than two sequence consonants), also cannot handle roman numbering system.
This paper explains how JawaTeX is designed. The model of transliteration in this paper is not focusing on a font making process but a document transliteration. The transliterator is a media named JawaTeX. JawaTeX project is an initial effort to make typesetting program using Javanese characters. The transliterator system framework include production rule of latin string split pattern browsing, string split pattern models, syntax code pattern models, LaTeX style, Javanese characters Metafont, and JawaTeX program package contain parsing and LaTeX style to write LaTeX syntax code. A JawaTeX program package contains two programs, checking and breaking Latin string to get the string split pattern and LaTeX style to write LaTeX syntax code.
After program testing on JawaTeX is done, the transliteration results are appropiate with the linguistics knowledge of writing Javanese script. The result is proving that the program can be used to transliterated from Latin to Javanese document. JawaTeX can also be used without program of checking and breaking Latin string, but users must have knowledge about how to get tokens and write LaTeX syntax code associated with that tokens. The concept of checking and breaking Latin string to get the pattern and the process to convert it into other characters which are built in this paper can be used as the base to be developed in other cases. Future research, the solution of writing a good Javanese characters still needs to be considered. Javanese characters writing sometimes cannot be justified alignment because writing Java script does not have spaces between words.

Key words
transliteration, Javanese characters, typesetting, framework, LaTeX, JawaTeX

1. Introduction

This paper is influenced by the research of Free/Open Source Software Localization (FOSS) [11] to develop software based on where the software is built. According to The Localization Industry Standards Association (LISA), Localization encloses product building that is appropriate to target culture (region and language) where the poducts are sold [7]. Research in China, Japan dan Korea (CJK) localization is a collaboration of 3 countries to use FOSS as localization, especially in a traditional writing character [4].
In many countries, the research has been done to develop character processing for their local culture. India Ministry of Information Technology also does the research in a local culture localization [7]. ArabTeX is a system for computer based typesetting of texts in the Roman script, which may contain insertions in some right-to-left script as Arabic and Hebrew [6]. TeX/LaTeX based transliteration is a transliteration using TeX/LaTeX. Many researchers do some researches in TeX/LaTeX based on transliteration; such as ArabTeX by Klaus Lagally, ChinaTeX by Shujun Li and ThaiTeX by Manop Wongsaisuwan [11]. Metafont program is used to develop fonts that are used by ArabTeX [5].
A Latin to Javanese character transliteration machine is one of the research fields in linguistic computational. There are just fewer researches on this field in Indonesia than other countries [13]. During this time, the transliteration focuses on making font (true type font) that is used in word processor [2][9]. The application of the word processor that is done is to devide the row space, with the result that the space distance between rows is different. When using those fonts users have to remember several formats. In addition, not all Latin characters writing can be transliterated to Javanese characters. Yet, the true type technology is not sufficient to handle the complexity of JawaTeX rules and represent the complex Latin text document writing.

Manuscript received April 5, 2010
Manuscript revised April 20, 2010

Transliteration process from Latin to Javanese is not simple process. Transliteration research of Latin document to Javanese character using TeX/LaTeX and Metafont has not been done yet [13]. So this paper is a new research in this field, Latin to Javanese character transliteration using TeX/LaTeX.

The research result is expected to facilitate schools, institutions, Department of Culture and Education, museums, tourism, Heritage Protection Department, Institute of Traditional Culture Heritage which need transliretarion from Latin document to Javanese characters for education, promotion, publication or publishing needed. This research result can be an educational tool and as step foward as well in the Java culture inheritance espesialy Javanese characters heritage in information technology era. The existing way to write Javanese characters is often inconsistent in character size and shape, by building the Latin to Javanese text document transliteration supported by the complete production rule to be able to handle the complexity in wrtiting the Javanese characters and can produce the good shape, beautiful and consistent Javanese characters.

By knowing all the problems of the existing research, the researcher decides to make the font based on Metafont and transliterator from the Latin to Javanese character by using TeX/LaTeX that is called JawaTeX (Javanese Typesetting and Transliteration Text Document). JawaTeX is intended to write Javanese character more simple and easier based on complex transliteration algoritm. By using several linguistic computational methods, weaknesses of the research can be handled. Transliteration model in this paper did not focus in font making but more in document transliteration. The input text that was used in this transliteration was native text characters written in Latin. As a result the scope is not only Javanese community that know Javanese character but also community that uses Latin character.

2. Framework of JawaTeX

Parsing is a process for spliting the natural language [10]. There are 3 kinds of parsers, The State Machine Parser, The Context Free Recursive Descent Parser and The Noise Diposal Parser. String searching involves two points, text (string with n characters length) and pattern (string with m characters length (m <n)) that will be searched in the text [8]. Pattern Matching can be devided into two models, there are exact string matching and inexact string matching or fuzzy string matching [3]. String matching is a process to find string characters form pattern in other string or text document content. There are several string searching algorithms, for example Brute Force Algorithm, Knuth Morris Pratt and Bayer Moore [12].

The framework of Latin to Javanese character transliteration with LaTeX is in figure 1.

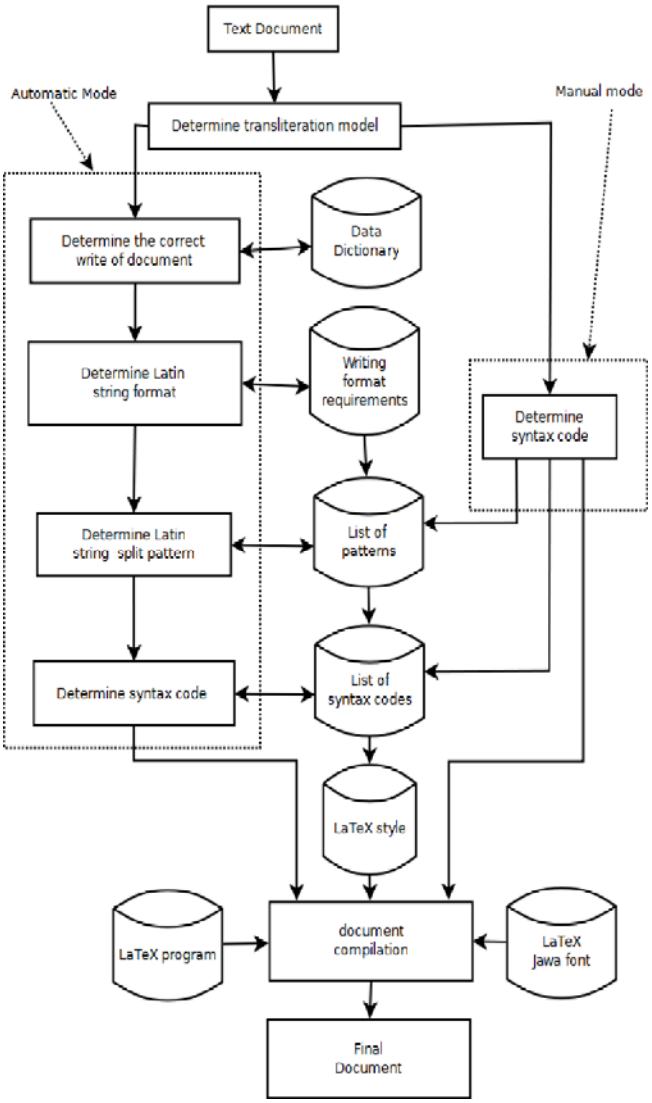


Figure 1: Framework of Latin to Javanese character transliteration with LaTeX

Processing the Latin text documents into the TeX-based Javanese characters has two modes of transliteration, namely: automatic mode and manual mode. Automatic mode designed for users who do not have the knowledge to determine Latin string split and syntax coding patterns. Most of the stages of the process performed by the system, include:

Determining the correct writing of Latin string in the source text by matching the source text and dictionary. This stage will result the Latin text documents that the writing of Latin string has been corrected.

Determining the formatting of Latin string by read, examining, modifying, altering, converting, inserting or adding other characters to meet the requirements of writing formats. The modification is only in the writing format form and does not change the meaning.

Determining the split string Latin pattern refer to 177 split pattern models that will produce 280490 Latin string patterns. This stage produce a list of Latin string split patterns that compose text document. The list of Latin string split pattern that has been obtained and then determined in the pattern of the relevant mapping transliteration to replace any Latin string split pattern into Javanese.

Determining the pattern syntax code which refers to the 57 coding syntax models. At the stage of correct pattern mapping, the first is to determine the position of Javanese characters as the scheme of Javanese characters writing. Every split of the Latin string pattern can site the alphabet blocks consisting of 5 rows and n columns [15]. This stage produces a list of syntax codes that will be used for transliterated split of the Latin string pattern which pattern layout has been obtained using the TeX/LaTeX format, are called the intermediate text.

After all 4 steps are performed automatically, the next stage is to compile the document. Intermediate text that has been obtained is compiled then JawaTeX.sty and Jawa.tfm are used by TeX to compile the document. JawaTeX.sty contains a Javanese script writing rules in a style TeX form, which includes:

1. The word mastery which is different for example in a name.
2. The rule to combine the characters merger and define how to place and combine the characters.
3. Determining the shape of characters that is required in the merger because the Javanese characters have a lot of variety. A character in Javanese will have a different shape if it is placed in different positions despite having the same sound. A character in Javanese can also be possible to be paired with some Java characters depending on the surrounding characters and the placement is not always in the back of the previous characters, but sometimes it must be inserted between the previous character. In addition, there are some characters that should not be paired with other characters, so that should replace the previous character. Jawa.tfm is font codes known by TeX and is a result of Metafont compilation.

Manual mode is intended for users who have the knowledge to determine the Latin string split patterns and syntax coding. There are 3 stages, the first is the correct of source text writing, the second is the writing of the Latin format string and the third is the split of Latin string patterns in which all of these have been in the mind of users, who then arrange in the intermediate text that is ready to be compiled.

The schema process of Latin to Javanese character transliteration with LaTeX is in figure 2.

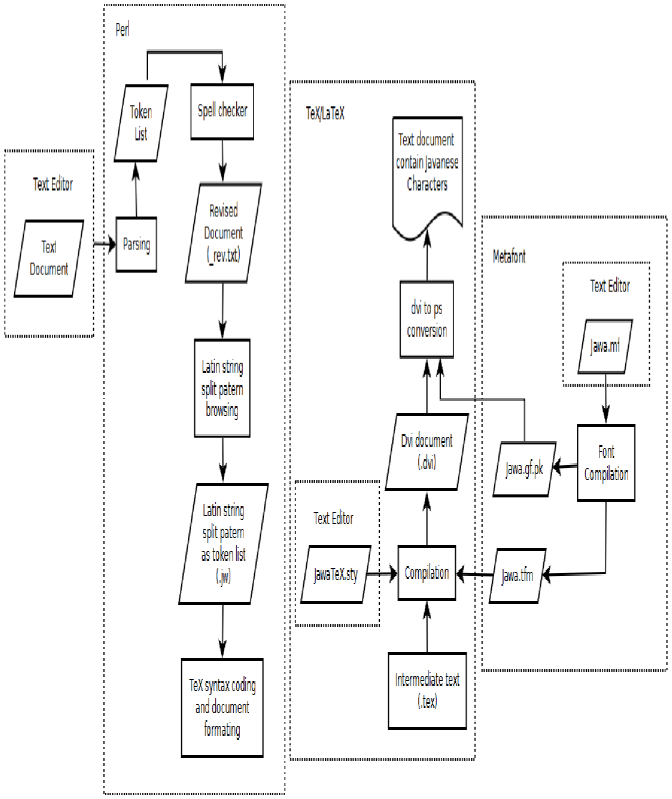


Figure 2: The schema process of Latin to Javanese character transliteration with LaTeX

Text document that will be transliterated to Javanese character was written using text editor. Before the browsing of the split of Latin string pattern, the spelling status from every word must be known. Error checking possibility that happened on writing of the text source is done by matching with dictionary. Thus the sensitivity or accuracy depends on the completeness of words list in the dictionary. Words searching in the dictionary is done by using the Brute Force algorithm, every time the system finds an unmatching pattern with text the pattern shifted one character to right. This spell checker facility is made to find word similarity to correct error spelling word. Word checking process consists of two word checking, Indonesian language consist of 7962 words and English language consists of 29759 words. If the words have similarity with the Indonesian words then they will be saved on temporary and then the checking is done in English. If they were not in the database, then the words can still be replaced by new input.

The Context of Free Recursive Descent Parser algorithm is used to browse and split the text document [14]. A Rule-based method is used to develop Latin string split pattern list which result from Latin text document processing. The Rule-based method is used to build several rules to handle the problems that can not be handled on previous researches. The Pattern Matching method is used to match each Latin

string split pattern into LaTeX mapping format forms. The rule in transliteration model is made according to linguistic knowledge from a book guidance of writing Javanese script written by Darusuprata and published by Pustaka Nusatama Yogyakarta Foundation that cooperate with the government of the special region province of Yogyakarta, the government of central Java, and government of West Java [1].

The Rule-based method is used to build several rules to handle the problems that can not be handled on previous researches. The developed transliterator system includes the rule of Latin string split pattern browsing, string split pattern models, syntax code pattern models, LaTeX style, Javanese characters Metafont, and JawaTeX program package which contain parsing and LaTeX style to write LaTeX syntax code.

An intermediate text is a document with TeX/LaTeX code and syntax. The code and syntax will be used to transliterate split Latin string pattern that the Javanese character positions are known. The intermediate text is a text document with extention .tex that follows right rule to write Javanese character [13].

Metafont is used to design and create Javanese character. The Javanese font is written and saved using text editor in file Jawa.mf. The Metafont program is used to converte Jawa.mf to become TeX font codes, .gf and .tfm. Metafont is a language program to define vektor font. Metafont is also a compiler that executes Metafont code, converts source code (vector font) .mf into bitmap font with .gf and and .tfm extention.

JawaTeX.sty is a style file or a class file that contains macro to define all rules that will be used to write Javanese characters [13]. TeX documents that are compiled to compose text document become output that can be seen in monitor screen. TeX Font Metrics (TFM) is used by TeX to compile document. dvi document (.dvi) is compiled in result document. Generic font (.gf) is compresed into packed font (.pk) using GftoPK program to obtain smaller size. Process to convert (dvi document becomes document that is ready to print (ps document) by using dvips program. The result of this transliterator is a document that contains Javanese characters as a result of complex rule using LaTeX and Perl program.

3. Result

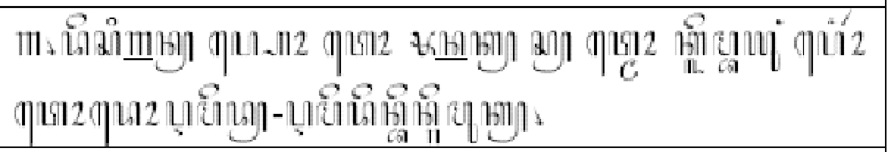
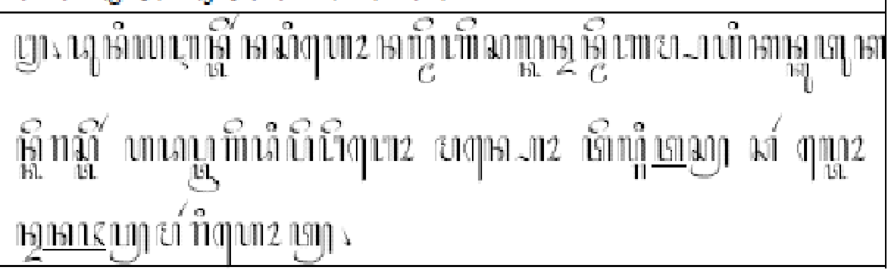
The testing of application is done by giving Latin text document input which contains: the sequence of string that has character combination which is possible to be written in syllable of Javanese character, and the sequence of string Latin that has character combination which is impossible to be written in syllable of Javanese character and previously cannot be transliterated to Javanese character.

There are 2 kinds of mechanisms for transliteration using JawaTeX: automatic mode and manual mode. The automatic mode, every source text written uses text editor and saved in *.txt* format. Source text is processed using Perl to produce correct syllable split pattern according to linguistics knowledge of writing Javanese script and mapping process into LaTeX code. Thus the general text source cultivation includes: the file writing and reading process, the formatting process, the Roman number checking process, the spelling checker for filtering wrong words process, and the LaTeX syntax code writing process. This process produces 3 files which contains: the corrected text document after the spelling checker process, the list of split Latin string pattern, and the list of syntax code pattern. The .tex document is then processed in LaTeX by calling syntax code that has been written in the LaTeX file style called *JawaTeX.sty*. The result is *.dvi* file that can be processed to be *.ps* and *.pdf* file. The system of text document split is tested by using text based file as shown in table l.
Three examples of Latin text in table 1 which have never been successfully transliterated using true type font, because this text contains character combination which is impossible to happen in Javanese. The result of the testing on table 1 shows that the model of split browsing can form the split pattern which has been in a line with the existing linguistic knowledge. The writing of Javanese character rule must follow the rule because in Javanese character writing a character depends on another character and influences the writing of other characters. No deletion or remove all in the input character will be transliterated. The Latin string is transliterated as it is without transcription (substitution of writing which is suitable by pronunciation and word). Ambiguity problem can be avoided by handling of space character and be expected to solve complex problem without any problems.

The manual mode, a user writes the LaTeX syntax code. The user must have knowledge of how to split the syllabic base on writing Javanese script and remember the syntax code based on JawaTeX. Figure 3 is an example of how to write a document that not all part will be transliterated. The user writes the

source text using the text editor and saves it in .tex format.

Table 1: Latin teks document test result

Example	Transliteration stage	File content
1	Latin string source text(.txt) Latin string split pattern (.jw)	1. Design photo Bank Stmok Lembayung Fortodo remed-remeden ben remuk. \1\.\de\si\gn\pho\to\Ba\nk\s\tmo\kle\mba\yung\for\to\do\re \me\d\-\re\me\de\nbe\n re\mu\k\.\
	Syntax code (.tex)	\1\ttk\de\si\gb{\ga}\n\pE\Ho\to\mba\gb{\na}\k\s\tE\Mo\ka \Le\ma\Ba\yung\for\to\dO\re\me\d\min\re\me\de\na\Be\na\Re\mu \k\ttk
	Javanese characters (.ps, .dvi, .pdf)	
2	Latin string source text (.txt)	2. Jembatan Trenggolong Krapyak ambrol remuk hancur.
	Latin string split pattern (.jw)	\2\.\je\mba\ta\nTreng\go\long\Kra\pya\k ha\mbro\l re\mu\k ha\ncur\.\
	Syntax code (.tex)	\2\ttk\je\ma\Ba\ta\na\mTreng\go\lOng\mkra\pya\ka\Ha\mE \Bro\la\Re\mu\ka\Ha\na\Cur\ttk
	Javanese characters (.ps, .dvi, .pdf)	
3	Latin string source text (.txt)	3. Dunia Internasional menyesalkan dan menyampaikan kutukan keras terhadap tragedi pengeboman Hotel Rits Carlton dan JW Marriot.
	Latin string split pattern (.jw)	\3\.\du\ni\ya\I\nter\na\si\ho\na\lme\nye\sa\lka\nda\nme \nya\mpa\hi\ka\nku\tu\ka\nke\ra\ster\ha\da\ptra\ge\di \pe\nge\bo\ma\nho\te\l ri\ts\car\lto\nda\njw\mar\ri\ho\t\.\
	Syntax code (.tex)	\3\ttk\du\ni\ya\I\na\Ter\na\si\ho\na\la\Me\nye\sa\la\Ka\na \Da\na\Me\nya\ma\Pa\hi\ka\na\Ku\tu\ka\na\Ke\ra\sa\Ter\ha\da \pa\Tra\ge\di\pe\nge\bo\ma\nE\Ho\te\la\Ri\gb{\ta}\s\car\lE\To \na\Da\gb{\na\ja}\w\mar\ri\ho\t\ttk
	Javanese characters (.ps, .dvi, .pdf)	

```
\documentclass{article}
\usepackage{JawaTeX}
\begin{document}
In many countries research has been done to develop character computeration for
their local culture. Right now Javanesse society write and read using Latin
characters and not familiar using Javanesse characters. This is happened because of
they did not have enough education to be able write and read Javanesse characters.
Writing Javanesse characters also not simple as Latin so make other problem. There
is only less research on digital Javanesse character. One of reason is that there
is only few researcher that expert in Javanesse grammar and has capability in
digital technology. Latin text document can be transliterated to Javanesse
character. That will be transliterated is parsed or broke to get token list
(sylabic). \begin{jw}
\mte\gb{\ksa}\t\dO\cu\me\gb{\na}\t\msa\Pa\Li\ta\mPa\ta\Ter\nE\mBro\wa\Sing\mba\se\d
E\Ho\na\Li\ngu\hi\sa\Ti\c\mkE\No\wa\Le\da\Ge\ho\fa\Wri\ting\ja\va\ne\sa\Se\msa\Cri\
gb{\pa}\t\hu\sing\mna\tu\ra\la\La\ngu\ha\ge\mpro\ce\sa\Sing\ttk\The\ha\lE\Go\ri\gb{
\tha}\m\ho\fa\La\ti\na\Te\gb{\ksa}\t\dO\cu\me\gb{\na}\t\sa\Pa\Li\ta\Ba\se\dE\Ho\na\
Li\ngu\hi\sa\Ti\c\kE\No\wa\Le\da\Ge\ho\fa\Wri\ting\ja\va\me\sa\Se\sa\Cri\gb{\pa}\t\
hu\sing\The\co\na\Te\gb{\ksa}\t\min\fre\he\re\cur\si\ve\min\de\sa\Ce\gb{\na}\t\mpar
\ser\ha\lE\Go\ri\gb{\tha}\m\co\na\Si\gb{\sa\ta}\s\ho\f\tda\ffi\le\wri\ting\ha\gb{\n
a}\d\re\ha\ding\pro\ce\gb{\sa}\s\kma\for\ma\ta\Ting\pro\ce\gb{\sa}\s\tda\ro\ma\na\N
u\ma\Ber\ca\He\ca\King\pro\ce\gb{\sa}\s\kma\sa\Pe\gb{\la}\l\ca\He\ca\Ker\for\ffi\la
\Te\ring\wrong\wor\gb{\da}\s\pro\ce\gb{\sa}\s\kma\ha\gb{\na}\d\la\ti\n\sa\Tring\sa\
Pa\Li\ta\Pa\ta\Ter\nE\Bro\wa\Sing\pro\ce\gb{\sa}\s\ttk\E\ver\yE\Pro\ce\gb{\sa}\s\ha
\sE\Ho\gb{\wa}\n\pro\du\ca\Ti\ho\na\Ru\le\ttk\gb{\ba}\y\bu\hi\la\Ding\ha\se\tE\Co\m
a\Pa\Le\ksE\Ho\fa\Ru\le\kma\la\ti\n\sa\Tring\se\khu\he\na\Ce\wa\Hi\gb{\ca}\h\hi\sa\
Wri\ta\Te\na\Hi\na\La\ti\n\ca\Ha\ra\ca\Ter\ca\na\Be\pro\ce\sa\Se\dE\So\tha\gb{\ta\s
a}\y\la\La\ba\Le\sa\Pa\Li\ta\Pa\ta\Ter\gb{\na}\s\ca\na\Be\pro\du\ce\dE\Cor\re\gb{\c
a\ta\la}\y\ttk\A\ma\Bi\gu\hi\gb{\ta}\y\pro\ba\Le\ma\Ca\na\Be\ha\vo\hi\de\gb{\da\ba}
\y\ha\n\da\Ling\ho\fa\Sa\Pa\ce\ca\Ha\ra\ca\Ter\ha\gb{\na}\d\be\he\ksa\Pe\ca\Te\dE\T
o\so\la\Ve\co\ma\Pa\Le\ksE\Pro\ba\Le\ma\Wi\tho\hu\ta\Ha\nyE\Pro\ba\Le\gb{\ma}\s\ttk
\thi\sa\La\ti\na\Te\gb{\ksa}\t\dO\cu\me\gb{\na}\t\sa\Pa\Li\gb{\ta}\s\ha\re\re\ha\gb
{\da}\y\to\be\pro\ce\sa\Se\da\Hi\nE\To\The\ne\gb{\ksa}\t\sa\Te\pa\THa\ta\Hi\sa\THe\
pro\ce\gb{\sa}\s\ho\fE\Co\ma\Ver\ting\gb{\sa}\y\la\La\ba\Le\sa\Pa\Li\ta\Pa\ta\Ter\g
b{\na}\s\to\be\ja\va\ne\sa\Se\ca\Ha\ra\ca\Ter\ha\gb{\na}\d\The\ma\pa\Ping\ho\fa\THe
\hir\wri\ting\s\ca\He\me\ttk\end{jw} This research belongs to Linguistics Computing
area, conducting the natural language processing using symbol by computer
technology. The concept of this text document split model can improve the existing
machine of Latin string split which was made before. From the testing it is seen
that by using The Context-Free Recursive-Descent Parser algorithm, text document in
Latin writing can be processed so that syllable split patterns can be produced
correctly. Honestly this research has not come to the final result, it has only
succeeded to produce syllable split patterns. The produced syllable split patterns
are ready to be processed into the next step that is the process of converting
syllable split patterns to be Javanesse character and the mapping of their writing
scheme. The concept of the next document split built in this paper can be used as
the base to be developed in other case.
\end{document}
```

Figure 3: An example of how to write a document that not all part will be transliterated

File in .tex format which is shown in figure 3 is then ready to be compiled using instructions:
ema@debian:~/JawaTeX/$ latex double.tex
ema@debian:~/JawaTeX/$ dvips double.dvi
ema@debian:~/JawaTeX/$ ps2pdf double.ps
This system can produce file in .pdf format which contains the result of transliteration as shown in figure 4.

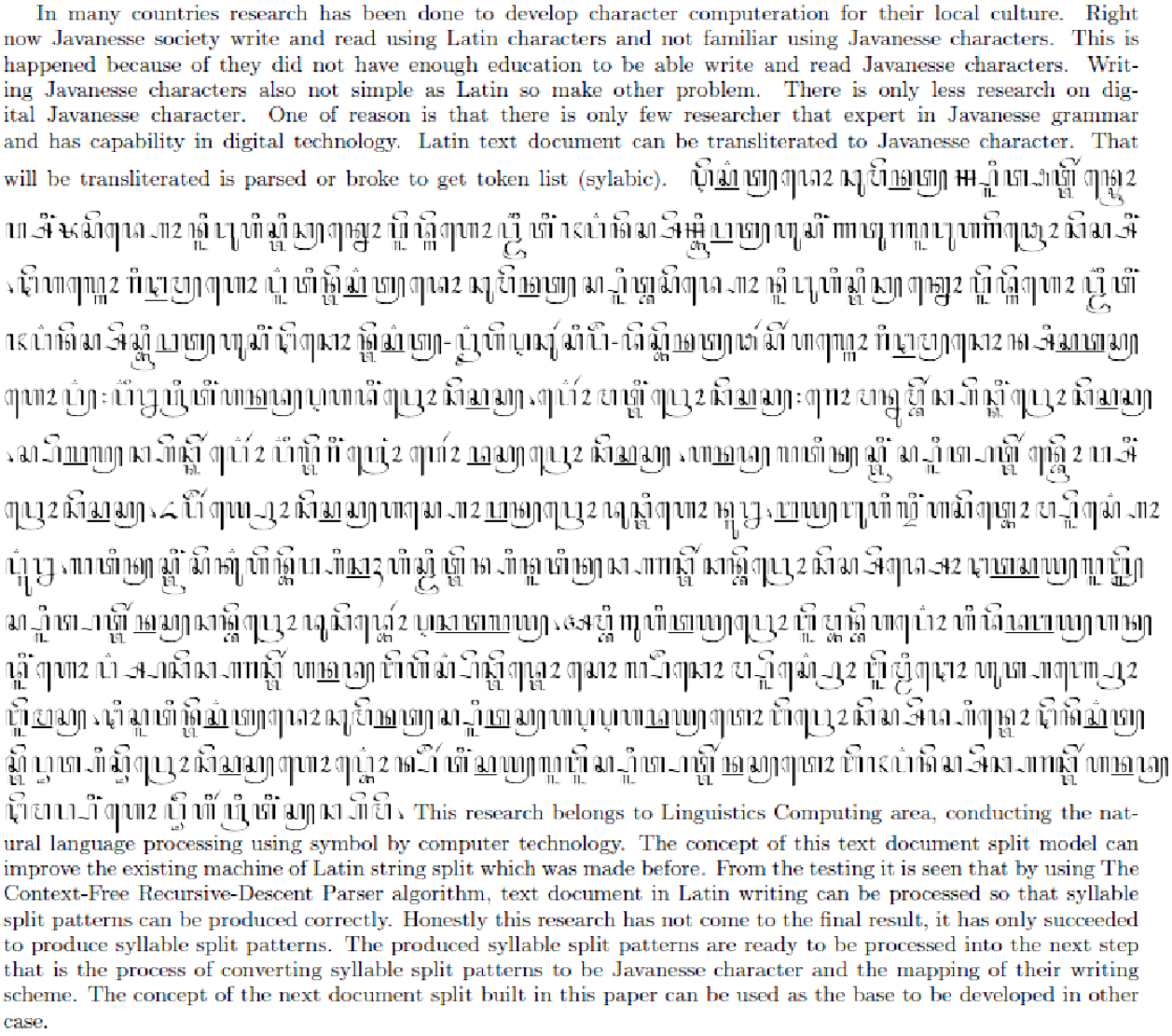


Figure 4: The part of text document processing

Because users do split Latin string in accordance with their knowledge or understanding, it is possible if a user makes a transcript of the split pattern which is based on the Latin string formation, or the sound of the word pronunciation of a word or pronunciation of sounds corresponding the language elements.

4. Discussion

The experimental results shows that by using The Context Free Recursive Descent Parser algorithm, the text document in Latin writing can be processed so that syllable split patterns can be produced correctly. The produced syllable split patterns are ready to be processed into the next step that is the process of converting split patterns to be the Javanese character and the mapping of their writing scheme using the Pattern Matching.

Teguh Budi Sayoga [9] created his own program to transliterate Latin to Javanese character based on the true type font called Hanacaraka font. These fonts are used by word processor such as Microsoft Word or Open Office. Jason Glavy also made Javanese font using fontograper program [2] called JG Aksara Jawa font. Using those fonts, users have to use word processor such as Microsof Word to write Javanese character. When using those fonts users have to remember several formats. For example, sentence

aksara jawi punika taksih kathah kekiranganipun thus the Latin characters that we have to write are ?aksrjwipunika [tkSih kqh kekirqnNipun$\backslash$. Another example is when we will show up Javanese character of script kraton jogjakarta, the Latin character that have to write is k][to[nJogJk/t. Hanacaraka and JG Aksara Jawa font also cannot write Javanese character that has multiple consonant Latin document, can not write q and x Latin character into Javanese character and cannot handle Roman numbering.

This research focuses on the document transliteration. Users write Latin characters text as input text and do not need to have knowledge of how to write Javanese characters. This transliteration algorithm has two main processes, checking and breaking Latin string into split Latin string pattern, and converting into Javanese characters. Each process has its own algorithm. List of split Latin string is transliterated without any transcription. The result of the testing shows that JawaTeX has some capabilities:

1. Able to find the word similarity to correct the word spelling mistakes.
2. Able to read, modify, and insert other characters into the character of input string in order to fulfill writing format requirement.
3. Able to handle the writing of Latin characters which have no equalization in Javanese alphabets.
4. Able to handle the writing of diphthong (multiple vocal characters).
5. Able to handle the writing of roman numbering systems.
6. Able to accomodate period to avoid ambiguity because period also has possibility as an indication of finishing sentences, decimal or abbreviation, thus it will influence how that character builds split Latin string pattern.
7. Able to accomodate space to avoid ambiguity, because Javanese character does not recognize space to divide words.
8. Able to accomodate an acute accent to avoid ambiguity.
9. Able to handle the writing of more than three multiple consonant characters.
10. Able to handle the sequence of string Latin that has character combination which is not possible to be written in syllable of Javanese character and previously cannot be transliterated to Javanese character.

This transliteration model is simulated for the text editor software, and not for the word proceccor or TeX as compiler, with the result that can add a number of TeX based transliterator. By developing JawaTeX class or style in TeX, then the Javanese characters are expexted to be equal with other etnic characters such as ArabTeX, ChinaTeX, dan ThaiTeX and are most likely to be recognized by the global community.

Model formulation of this text document transliteration can improve the existing Latin to Javanese characters machine transliteration. By constructing a complete production rules, transliteration models can be created to handle the problems that occurr in previous studies. This transliteration model can transliterate all possible combinations of characters that make up the Latin text of a document, without limiting the natural language used to create the Latin text documents. In addition to that transliteration model is also facilitated with the spelling checker that is expected to increase quality of transliteration.

By the modification this transliteration model can be developed to become easier and better tool that can be used by everyone as initial effort to transliterate local characters in wider scope and benefits. This research provides wider development materials in linguistic computational field that is needed for future use, thus more community will take advantage of this research result.

5. Conclusion

The text document transliteration framework covers: production rules for splitting Latin string that consists of the spelling checker, the Roman number checking, the text formatting and the Latin string split pattern browsing, the list of the split Latin string pattern models, the list of yhe syntax code pattern models, and the production rules that are used for transliteration the split of Latin string pattern to Javanese character. Every process has its own production rule. By building a set complex of rule, the Latin string sequence which was written in Latin character can be processed so that split patterns can be produced correctly. Ambiguity problem can be avoided by handling the space, dash, and period characters, and be expected to solve complex problems without any problems. The concept of this

transliterator model can improve the existing machine of Latin string split which is made previously.

JawaTeX program package contains two programs, checking and breaking the Latin string to get the string split pattern and LaTeX style to write LaTeX syntax code. JawaTeX can also be used without program of checking and breaking Latin string, but users must have knowledge about how to get tokens and write LaTeX syntax code associated with those tokens. The concept of checking and breaking Latin string to get the pattern and the process of converting it into other characters which are built in this paper can be used as the base to be developed in other cases..

References

[1] Darusuprapta, et all., Pedoman Penulisan Aksara Jawa. Yayasan Pustaka Nusatama, Yogyakarta. 2002.

[2] Glavy, J., Asian Fonts. Online at http://www.geocities.com/jglavy/asian.html. 10 September 2006.

[3] Gusfield, D., Algorithms on Strings, Trees, and Sequences: Computer Science. Cambridge University Press, 1997.

[4] Krikke, J., Linux Revolution: Asian Countries Push Open Source. 2003. Online at www.linuxinsider.com/story/3241.html. 21 Februari 2006.

[5] Lagally, K., ArabTeX: Multilingual Computer Typesetting. Technical Report, Universitat Stuttgart, Faculty of Computer Science. 1991.

[6] Lagally, K., ArabTeX: Typesetting Arabic and Hebrew. Technical Report, Universitat Stuttgart, Faculty of Computer Science. 2004

[7] Lommel, A., LISA The Localization Industry Primer Second Edition. LISA-The Localization Industry Standards Association. 2003. Online at www.lisa.org/interact/LISAprimer.pdf. 21 Februari 2006.

[8] Mohammad, A; Saleh, O; Abdeen, R., Occurrences Algorithm for String Searching Based on Brute-Force algorithm. Journal of Computer Science 2(1): 82-85, 2006. ISSN 1549-3636. Science Publications. 2006. Online at www.scipub.org/fulltext/jcs/jcs2182-85.pdf. 25 Juni 2008.

[9] Sayoga, T., The Official Site of Aksara Jawa 2005. Online pada http://hanacaraka.fateback.com, 10 September 2006.

[10] Schildt; Herbert., Artificial Intelligence Using C. Osborne-McGraw Hill, California, 1987.

[11] Souphavanh, A.; Karoonboonyanan, T., Free/Open Source Software Localization. Elseiver, Reed Elseiver India Private Limited. 2005.

[12] Stephen, G., String Searching Algorithms. World Scientific. 1994.

[13] Utami, E.; Istiyanto, J.; Hartati, S.; Marsono; Ashari, A., JawaTeX: Javanese Typesetting and Transliteration Text Document. Presented at Proceedings of International Conference on Advanced Computational Intelligence and Its Application (ICACIA) 2008, ISBN: 978-979-98352-5-3, 1 September 2008, page 149-153.

[14] Utami, E.; Istiyanto, J.; Hartati, S.; Marsono; Ashari, A., Applying Natural Language Processing in Developing Split Pattern of Latin Character Text Document According to Linguistic Knowledge of Writing Javanese Script. Presented at Proceedings of International Graduate Conference on Engineering and Science (IGCES) 2008, ISSN: 1823-3287, 23-24 December 2008, D5.

[15] Utami, E.; Istiyanto, J.; Hartati, S.; Marsono; Ashari, A., Developing Transliteration Pattern of Latin Character Text Document Algorithm Based on Linguistics Knowledge of Writing Javanese Script. Proceeding of International Conference on Intrumentation, Communication, Information Technology and Biomedical Engineering (ICICI-BME) 2009, ISBN: 978-979-1344-67-8, IEEE: CFPO987H-CDR, 23-25 November 2009.

Ema Utami received the S.Si, M.Kom and Doctoral degrees in Computer Science from Gadjah Mada University in 1997, 1999 and 2010 respectively. Since 1998 she a lecturer in STMIK AMIKOM Yogyakarta, Indonesia. Her areas interest are Natural Language Processing, Computer Algoritms, and Database Programming.

Jazi Eko Istiyanto, Sri Hartati, Marsono, and Ahmad Ashari are the promoters of Ema Utami in Doctoral degrees at Computer Science of Postgraduate School Gadjah Mada University