Building a web-based application for language resources in Indonesia

Arawinda Dinakaramani¹ and Totok Suhardijanto²
Universitas Indonesia, Depok, Indonesia
E-mail: arawinda.dinakaramani@ui.ac.id¹, totok.suhardijanto@ui.ac.id²

Abstract. This paper presents our work on building a web-based application for language resources development and management for Indonesian languages. This web-based application is developed using PHP, following the model-view-controller (MVC) architectural pattern. The features of our web-based application are designed to enable users in utilizing language resources and to facilitate team members in developing and managing language resources collaboratively. Some features are accessible to all types of users, but some others are accessible only to a specific type of users. The application is also intended to cover all Indonesian languages. As a result, corpus files should be encoded in UTF-8 to enable the application analyze a wide range of language characters. At the moment, the implemented corpora are Indonesian, Javanese and Sundanese. The application is designed to bear future improvements.

1. Introduction
The development of language resources in Indonesia has shown progress in recent years. It can be seen from the emergence of Kateglo¹, Tesaurus Tematis Daring², KBBI Daring³, and other web-based dictionaries. In terms of electronic corpus, only few language resources for Indonesian languages are available online, such as Malay Concordance⁴, Leipzig Corpora⁵, SEAlang Indonesia corpus⁶, and SEAlang Javanese corpus⁷.

In this paper, we present our attempt to build a web-based application that provides language resources for Indonesian languages and also enables users to manage the language resources collaboratively. Since there are 719 languages spoken in Indonesia, and it would be very time-consuming and costly to develop language resources for all of those languages, we chose Javanese and Indonesian for our pilot project. Javanese is one of the major languages with large number of speakers in Indonesia. According to Simons and Fennig [1], Javanese has 84.3 million speakers. Indonesian is chosen due to its national language status.

¹ http://kateglo.com/
² http://tesaurus.kemdikbud.go.id/tematis/
³ https://kbbi.kemdikbud.go.id/
⁴ http://mcp.anu.edu.au/
⁵ http://corpora.uni-leipzig.de/en?corpusId=ind_mixed_2013
⁶ http://sealang.net/indonesia/corpus.htm
⁷ http://sealang.net/java/corpus.htm
2. Language resources
According to ELRA (European Language Resources Association), the term language resource refers to a set of speech or language data and descriptions in machine readable form. It is used for building, improving, or evaluating natural language and speech algorithms or systems. It also can be used as core resources for the software localisation and language services industries, for language studies, electronic publishing, international transactions, subject-area specialists and end users. Examples of language resources are terminology databases, computational lexica, speech collection, and written corpora.

This work aims to develop a web-based application for managing and developing under-described language resources. Although working on language resource building, to some extent, this work is quite different from Language Grid [2]. Language Grid is aimed to not only for building a language resource, but also for implementing language services. The idea of the service grid architecture is based on the concepts of fragmentation, which provides various language services, and recombination, which realizes a customizable language environment. The implementation of the Language Grid created a full range of customized language environments for different types of user activities. Unlike Language Grid, this work focuses on building a workspace for developing and managing language resources. The idea does not cover an implementation of the resources for further applications. However, this work created an accommodating workspace which meets a user need for developing minor languages in a multicultural countries like Indonesia.

This work is also different from Nasution’s work [3] which dealt with machine-readable bilingual dictionaries for translation works. The research result developed a generalized constraint-based bilingual lexicon induction framework for closely related low resource languages. The use of generalized constraint based framework makes the method applicable for a wider range of language groups than the one-to-one approach. In this work, we did not work either on bilingual dictionaries or on machine translation. We only focused on developing a corpus management system that allow user to collect, maintain, and analyze a corpus.

The term corpus has existed at least since 1950s [4]. In its development, modern technologies and computers have been involved in corpus management since 1980s. Nowadays, the term corpus is often associated with computer software or analytical techniques that use computer to analyze language data. Sinclair [5] defines corpus as a collection of pieces of language text in electronic form, selected according to external criteria to represent, as far as possible, a language or language variety as a source of data for linguistic research. The emphasis on ”electronic form” or ”machine readable” distinguishes corpus in modern era from corpus in the past.

3. Corpus tool
Lexicographers need a tool to analyze corpus data, which allows lexicographers to search, manipulate, and save corpus data. This kind of tool is referred to with various terms, including corpus query system [6] and corpus tool [7]. Examples of corpus tools are Antconc [8], SketchEngine [9], and WordSmith [10].

A corpus tool can be a stand-alone software installed on the users computer or an online corpus tool that allows user to access corpus, or corpora, from any computer. A corpus tool is designed to have some features. The basic feature is concordance. User can use corpus tool to search for a key word and then the search results will be shown as line of context for each occurrence of the key word. Other features of corpus tool include the ability to extract wordlist and metadata information from the corpus.

Some corpus tools are designed for a particular corpus, while other corpus tools are designed to enable user to upload and analyze any corpus. Most corpus tools are used to access a prepared corpus, while some corpus tools are used to access web as corpus. Prepared corpus is a corpus that has been compiled with linguistic research in mind and specifically designed for linguists’
purposes [7]. The web can be viewed as corpus with vast quantities of texts for many languages that covers a wide range of text types and domains [9].

The users of corpus tools can be categorized into several types, such as lexicographers, linguistics researchers and students, and language teachers and learners. Corpus tools are designed to meet the needs of its target users. A corpus tool with particular target users can be different from corpus tool with different target users.

4. Design and architecture

4.1. Corpus data

The design of corpus data for our web-based application is decided by considering (1) selection criteria, (2) corpus size, (3) data authenticity, (4) storage media, and (5) data manipulation. In terms of selection criteria, we determine the main purpose of our corpora development. We aim to provide language resources for indigenous languages in Indonesia that can be utilized for all purposes. Therefore, we design corpus that represents various texts or genres.

In terms of corpus size, we have collected at least 2.5 million tokens for Javanese corpus and 18 million tokens for Indonesian corpus. This size is still growing since we are still developing our corpora both in terms of quality and quantity as well as in terms of the practicality of storage and management. In terms of data authenticity, materials for our corpus data are originated from real data. There are no artificial data that specially composed to complete our corpus.

In terms of storage media, each corpus data is digitalized, especially in the form of text file, that is file with txt filename extension, and the text file is encoded in UTF-8. Each text file contains text of a corpus entry that has been manually annotated by providing additional metadata using markup language format. There are six metadata tags that must be added to each text file. These metadata tags mark entry title, the year when the entry was published, entry type, entry subtype, dialect that is used in the entry, and the author name of the entry. Figure 1 shows an example of metadata tags on an entry text file.

In terms of data manipulation, to access and manage corpus data, we build a web-based application that we present in this paper.

4.2. Corpus tool

We designed a web-based application that could provide many language resources and should also be able to facilitate team members to develop and manage language resources collaboratively. We named our web-based application Korpus Universitas Indonesia. This web-based application is developed using PHP, following the model-view-controller (MVC) architectural pattern. Architecture of our web-based application can be seen in Figure 2.

4.3. User classification

Our web-based application users are classified into seven types, namely admin, chief editor, editor, data contributor, pending member, pending user, and uncategorized. An admin is a
web administrator that can access all features of the application. Chief editor, editor, and data contributor are members of a corpus team. A pending member is a user that has already submitted a request to contribute to a corpus, but is still pending the approval from the chief editor or an editor of said corpus. A pending user is a user that has already registered for a new account, but is still pending the approval from admin. An uncategorized user is a user accessing the application without login or one that logs in using an account not classified as admin and not classified as a member of corpus team.

4.4. Database
Language resources along with user data of our web-based application are stored in a database that is built using MySQL. The entities in our database can be classified into two main groups, namely user entities group and corpus entities group. The user entities group contains entities that are related to users, including role, user, user activity, and user role. The corpus entities group contains entities that are related to the corpus, including corpus, corpus log, entry, entry meta, entry text, language, type meta, word, and word count.

5. Web-based application interface and features
Indonesian is set as the default language for the application interface page, considering that the targeted visitors and application users are Indonesian people and Indonesian speaking people. Nonetheless, an option is made available for international users to change the interface page language into English.

There are two main modules of our web-based application, namely the user module and the corpus module. The user module deals with features related to users, such as Sign Up, Reset Password, Login, Edit Profile, and Edit Account features. The corpus module deals with features related to the corpus, such as Select Corpus, View Word List, and View Concordance Search Result features.

The features of our web-based application are designed to enable users in utilizing language resources and to facilitate team members in developing and managing language resources.
5.1. The home page
The Select Corpus feature is made available as the main feature on the home page. Figure 3 shows this feature on the English version home page. The Select Corpus feature allows user to choose any available corpus. For now, our web-based application contains three corpora, namely Korpus Jawa (Javanese corpus), Korpus Indonesia (Indonesian corpus), and Korpus Sunda (Sundanese corpus). However, it is expected that some other corpora will be developed and integrated into our web-based application in the future.

5.2. The corpus page
Each corpus is presented on a corpus page. After selecting a corpus from the Select Corpus feature on the home page, user will be redirected to the corpus page. Information about the selected corpus will be shown on the corpus page. All types of users can view basic information of a corpus, including description, language, total number of entries, and total number of words. A logged in user can view additional information of a corpus, including date of creation, corpus creator, and corpus status, depending on the type of user.

Chief editor and editor of a corpus can change corpus status from not published to published online or from published online to not published. Corpus that is published online can be viewed by everyone on the Internet, while corpus that is not published can be viewed only by corpus team members and admins. Chief editor and editor can also change corpus information, such as corpus title, corpus description in English, and corpus description in Indonesian.

On the corpus page, there is a menu that contains options to view list of corpus team members, view list of entries, search for a word, and view list of words. In addition to those options, specific type of users can also choose some additional options on the corpus page. Additional options

---

**Figure 3.** The home page of the web-based application.
on the corpus page include options to submit request to contribute, change corpus status, edit corpus information, add new entry, manage corpus, and view activity log.

5.3. The entry page
If user chooses option to view list of entries on the corpus page, entries of the corpus will be displayed in alphabetical order. Each entry is displayed in the form of its title with link to its entry page. User can click the title of an entry and view the entry page of said entry.

The entry page contains information and menu related to entry. All types of users can view basic information of an entry, including title, year, type, subtype, language, dialect, author, and total number of words. Menu related to entry contains options to search for a word and view list of words. Specific type of users can choose some additional options on the entry page, including options to edit entry and view activity log.

5.4. Entry text file processing
In our web-based application, a corpus can contain many entries, one entry, or no entry at all. A new entry can be added to the corpus by uploading a file that contains the entry. At the moment, our web-based application can accept a new entry in the form of text file only.

The entry text file must match required criteria. Our web-based application will validate the filename extension, the file encoding, and the contents of the file. The filename extension of entry text file must be txt. If the filename extension is not txt, entry text file will be rejected. The file encoding of entry text file must be UTF-8. If the file encoding is not UTF-8, entry text file will be rejected.

Each entry text file must contain all required metadata tags and the entry text. The required metadata tags must exist in the correct format and order as shown in the example in Figure 1. Entry text file will be rejected if required metadata tags are not in the correct order, a required metadata tag is not in the correct format, or a required metadata tag is not found. Entry text file will also be rejected if the entry text is empty.

If entry text file matches required criteria, our web-based application will extract the contents of the file. The metadata will be stored in the entry_meta table in database. The entry text will be stored in the entry_text table in database. If entry text file is uploaded by chief editor or editor, the entry text will also be tokenized and taken into account in calculating total number of entries, total number of words, word frequency, and document frequency. The resulted tokens and calculation will be stored in the word table and the word_count table in database.

Our web-based application currently uses whitespace delimiter for tokenization. Punctuation, HTML tag, token that contains only numbers, and token that contains only one character are excluded from the resulted tokens. The resulted tokens of that entry can be viewed by choosing option to view list of words on the entry page. For example, if user uploads an entry text that is written in Javanese script as shown in Figure 4, the resulted tokens will be as shown in Figure 5.

All members of a corpus team can add new entry to the corpus by uploading entry text file. If entry text file is uploaded by data contributor, the status of uploaded entry will be pending entry. A pending entry needs approval from chief editor or editor. The metadata of pending entry will be stored in the entry_meta table in database. The entry text of pending entry will be stored in the entry_text table in database. The entry text of pending entry will not be
Figure 4. The entry text that is written in Javanese script.

Figure 5. The resulted tokens of the entry text that is written in Javanese script.

tokenized and taken into account in calculating total number of entries, total number of words, word frequency, and document frequency.

Chief editor and editor can view list of pending entries by choosing option to manage corpus on the corpus page. Chief editor and editor can view the contents of a pending entry before approving or rejecting it. Chief editor and editor can also edit a pending entry before approving it. If a pending entry is edited, its metadata and entry text in database will be updated. If a pending entry is rejected, its metadata and entry text will be deleted from database. If a pending entry is approved, the approved entry will be removed from the list of pending entries and its entry text will be tokenized and taken into account in calculating total number of entries, total number of words, word frequency, and document frequency. The resulted tokens and calculation will be stored in the word table and the word_count table in database.

Unlike chief editor and editor, by choosing option to manage corpus on the corpus page, data contributor can only view list of his/her pending entries. Data contributor can view and edit a pending entry from the list of his/her pending entries. If data contributor edits a pending entry, its metadata and entry text in database will be updated.
5.5. The word list
User can view list of words by choosing option to view list of words on the corpus page or on the entry page. If user chooses option to view list of words on the corpus page, the application will select all words in the corpus from the word_count table in database. After that, the application will calculate frequency of appearance of each word and total number of document entries where the word appears. The application will also calculate total number of word types, that is total number of unique words in the corpus, and total number of word tokens, that is total number of words in the corpus. List of words on the corpus page can be sorted by frequency, by document frequency, and in alphabetical order.

If user chooses option to view list of words on the entry page, the application will select all words in the entry from the word_count table in database. After that, the application will calculate frequency of appearance of each word, total number of word types, and total number of word tokens. List of words on the entry page can be sorted by frequency and in alphabetical order.

5.6. The concordance
User can search for a word in the corpus by choosing option to search for a word on the corpus page and enter the search keyword. User can set search criteria to narrow the search results. The application will search in database for the entries that match the search criteria and contain the search keyword. If such entries are found, the application will select the search keyword and its context from those entries. At the moment, the context is eight words before and eight words after the search keyword.

The search results will be displayed as concordance. Each line of the search results shows context for each occurrence of the word, with the word centred, and provides link to the entry from which the context originated. The link to the entry is displayed in the form of line number at the beginning of each line of the search results.

Option to search for a word is available not only on the corpus page, but also on the entry page. However, there is no search criteria for searching on the entry page. As on the corpus page, the search results on the entry page are displayed as concordance. Each line of the search results shows context for each occurrence of the word, with the word centred. Unlike on the corpus page, the line number at the beginning of each line of the search results is not a link.

5.7. Corpus team members
The corpus team consists of chief editor, editor, and data contributor. By choosing option to view list of corpus team members, user can view list of members of corpus team. Each member of corpus team is displayed in the form of his/her full name with link to his/her profile page.

User that is not a member of corpus team can request to contribute by choosing option to submit request to contribute on the corpus page. If a user submits request to contribute to a corpus, the user will be a pending member of that corpus and the request will need to be reviewed by chief editor or editor. If chief editor or editor approves a pending request, the request submitter will be added to the corpus team as data contributor.

In addition to approving pending request, data contributor can be added to the corpus team by using the Add Data Contributor feature. The privilege of adding and removing data contributor is given to chief editor and editor, while the privilege of adding and removing editor is given only to chief editor.

5.8. Activity log
Our web-based application logs user activities, such as upload file of entry, approve new entry, edit entry, request to contribute, approve request to contribute, add editor, and remove editor. Corpus team member can view list of user activities by choosing option to view activity log.
This option is available both on the corpus page and the entry page. The option on the corpus page is for viewing activity log related to the corpus. If user wants to view activity log related only to a single entry of the corpus, user can choose option to view activity log on said entry page.

6. Conclusion and future work
We have built a web-based application that provides language resources for Indonesian languages and also enables users to manage the language resources collaboratively. We named our web-based application Korpus Universitas Indonesia. It contains three corpora, namely Korpus Jawa (Javanese corpus), Korpus Indonesia (Indonesian corpus), and Korpus Sunda (Sundanese corpus).

We plan to expand language resources and add more features in our web-based application. In the near future, we will implement annotation features into the application. Since part of speech (POS) annotation is language specific, at the first step, we will apply a POS tagging for Indonesian corpus. After this attempt, we will also implement POS tagging for Javanese and Sundanese corpora. For further implementation, the application will also be equipped with a name-entity annotation, word sense disambiguation (WSD) annotation, and syntax parser.

References
[1] Simons G F and Fennig C D (eds) 2018 Ethnologue: Languages of the World 21st ed (Dallas: SIL International)
[2] Ishida T, Murakami Y, Lin D, Nakaguchi T and Otani M 2018 Computer 51 72–81 ISSN 0018-9162
[3] Nasution A H, Murakami Y and Ishida T 2018 ACM Trans. Asian & Low-Resource Lang. Inf. Process. 17 9:1–9:29 URL http://doi.acm.org/10.1145/3138815
[4] McEnery T, Xiao R and Tono Y 2006 Corpus-based Language Studies: An Advanced Resource Book (London/New York: Routledge)
[5] Sinclair J 2005 Developing Linguistic Corpora: a Guide to Good Practice ed Wynne M (Oxford: Oxbow Books) pp 1–16
[6] Atkins S B T and Rundell M 2008 The Oxford Guide to Practical Lexicography (Oxford: Oxford University Press)
[7] Kilgarriff A and Kosem I 2012 Electronic Lexicography ed Granger S and Paquot M (Oxford: Oxford University Press) pp 83–106
[8] Anthony L 2006 Proceedings of the JACET 45th Annual Convention pp 218–219
[9] Kilgarriff A, Baisa V, Buta J, Jakubek M, Kov V, Michelfeit J, Rychl P and Suchomel V 2014 Lexicography 7–36
[10] Scott M 2016 WordSmith Tools version 7 (Stroud: Lexical Analysis Software)