STEMMING ENGINE FOR BAHASA MELAYU WORDS

Muhammad Aizat bin Abdul Rahman

2006

ACKNOWLEDGEMENT

First of all, my most grateful to ALLAH, the most merciful, the most compassionate, by all HIS mercy letting me finish my final project for my study here in University of Malaya. HE had given me so much opportunity and convenience for me to make this project done even though I had to struggle everyday and stayed up all night long to make sure that this projects fulfill the requirements to graduate for the University of Malaya Bachelor of Science Computer Degree.

My second gratitude is to my supervisor, Dr. Rukaini Hj Abdullah for her thought and consideration. She had been very helpful all the time. She always gave me chances to improve my project and gave me the ideas, comments and her wishful thinking in order to ensure that my project will be perfect and valuable.

Next, my warmest thank you to Dr. Roziati Zainudin and Dr. Sapiyan Baba as my panels for the viva presentation for evaluating my project and gave me a bit idea on how to write my report. Not forgetting all the individuals who had contributed much or less to the completion of this project.

Last but not least, my deep love to my parents, family and friends who had gave me so much support and encouraged me to do my best for whatever I am doing. They had been very caring and affectionate in times when my days went tough. Thank you, your love will always be in my heart.

ABSTRACT

Despite its top rank in the world of the most speakers for a language, Bahasa Melayu is lacked in its computer software particularly in its language and grammar processing software. This situation had put much excite on me to build a system for stemming the words in Bahasa Melayu which can be used for further development of projects that will use Bahasa Melayu for its domain or in its interface. The purpose of this stemmer is to stem any given Bahasa Melayu words to its pure stem. The steps are, first, the system will take the input from user. Then, it will make a comparison in its database which is the dictionary of Bahasa Melayu stems. After that, in its stemming engine, the words will discard all the possible affixes that attached to it. Lastly, the pure stem of the word will be displayed as the output to the user. Another reason for me to build this system is to explore the power of programming in prolog which is one of the common languages for Artificial Intelligence. It is my hope that this project will give much help to those who concern about making Bahasa Melayu a prestigious language in the world.

TABLE OF CONTENT

LIST OF FIGURES & TABLE

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Bahasa Melayu or Malay language is becoming more and more popular each day as our government encourages the practice of Malay language in everyday situations such as in government departments and schools. In fact it is one of the ten most used languages in the world.

Despite of its popularity, surprisingly, only a few studies and researches had been carried out by those who are concern in making Malay as a prestigious language of the world. There are not many computer programs which are designed to entertain the needs of using and studying Malay language and grammar or at least apply it as the interface of their programs.

The lack of Malay computer programs has been the idea of developing such a system which concentrates on one of the least studied fields: the stemming of Malay words. This system is supposed to stem any Malay word to its original root. This system can assist anyone who wishes to learn Malay in a way that it gives the root of the affixed Malay words so that they can easily search the meaning in dictionaries. It also helps children to learn the various patterns of affixes in Malay language and the proper to use them.

All in all, this project is one of the efforts to make sure that Malay will become one of the most researched languages like the ever popular English, French and Chinese.

The problems stated above had made much consideration to improve the system being developed. For the project motivation, the project will overcome all of the problems mentioned. It will have added functionality and not just a program that can stem but does nothing else. The interface will be as attractive as it could so that users will find it is enjoyable and pleasurable to use.

1.2 PROJECT OBJECTIVES

The main objective for this project is to develop an engine of stemming which can stem any Malay words in paragraphs. This means that the system can stem every single word in a given Malay text. Among with the objective stated above, there are several other objectives to complement this project:

- To enhance and improve the efficiency and effectiveness of the existing Malay stemmers
- To build a sensible program which is not only reliable but up to date with new words or phrases entering Malay lexicon.
- To create an easy algorithm in Prolog for a Malay stemming engine that can be upgraded as requirements change.
- To act as a customized dictionary for people who want to look up the correct spelling of Malay words as well as to confirm whether or not a particular word is a Malay word.

1.3 PROJECT SCOPES

Every project should have its scopes in order to meet the main objectives of the project. The scopes are essential for project developers as they set the fundamental boundaries so that it would not go further beyond the objectives. Listed below are the scopes that act as the guidelines for this project development:

- The stemmer should only stem words exist in Malay language; both the pure and the loanwords.
- It would be able to stem any words to their very exact root. This includes 'imbuhan', 'kata kerja' and 'kata ganda'.
- This stemmer would deal with stripping of affixes, to be precise prefixes and suffixes.
- It is developed to be a stand-alone system and not a web-based one.
- It applies a user-friendly approach in its interface.
- It must be upgradeable for further development.

1.4 PROBLEM STATEMENT

After doing some individual researches and analysis, there are several problems that need to be overcome by the proposed system. Below are the problems found:

1. Lack of existing system

   There are many stemmers that can be found in the internet but they are for mainly for the European languages such as French, Italian, Spanish and German while others are Russian, Arabic and Japanese. So far, there is only some similar stand-alone system for stemming of Malay words existed while none of them were found available in the internet. This is very disappointing for those who whish to learn Malay language.

2. Does not stem certain words

   The existing stemmer was created to stem some kind of words in Bahasa Melayu. The proposed system should overcome these problem.

The problems stated above had made much consideration to improve the system being developed. For the project motivation, the project will overcome all of the problems mentioned. It will have added functionality and not just a program that can stem but does nothing else. The interface will be as attractive as it could so that users will find it is enjoyable and pleasurable to use.

1.5 PROPOSED SOLUTION

The proposed system will have these specification to meet the obejectives and overcome the pre-mentioned problems:

- The system can stem words of pure Malay or loanwords. It should be able to stem most of the common Malay words and probably not to some uncommon Malay words.
- The system must be user-friendly.
- It will most definitely have some other added function besides the main function of stemming words.

1.6 SUMMARY

This system is one of the researches done to make the learning of Malay language and its usage are easy and enjoyable. Stemming is very important in Malay language since it serves many purposes as mentioned in the previous parts. The methodology, technology and tools had been chosen carefully to make sure the system are developed to meet its objectives.

It is hopefully can benefit those who are desperate of references in Malay in terms of computer software. May this project boost the eagerness of software developers to develop software similar to its kind or if not use Malay language as their main domain of research.

Chapter 2

Literature Review

CHAPTER 2

LITERATURE RIVIEW

2.1 REQUIREMENT TECHNIQUES

To gather the requirements needed to build the system, some common techniques of requirements gathering had been applied gradually as the development matures. Among of the techniques used to gain information on digital cameras and users' requirements are internet-surfing, interviews, observation and reading material. Whereas for report writing, discussion with supervisor and thesis reviewing were the techniques applied to.

2.1.1 Internet Surfing

This is the most effective way to get the information about the previous systems. Even though there are not much similar system in Bahasa Melayu, there is a bundle of similar system in other language such as English, Russian, Arabic and Japanese. However, the grammars of these systems are different from Bahasa Melayu. So, more researches had to be done is other kind of way

2.1.2 Interviews

A few interviews had been conducted with some of the people that make the Bahasa Melayu stemming as the major domain in their researches. One of them is developing the same system but in another language. I gained much information from her to further my research during the design process.

2.1.3 Reading material and thesis reviewing

Revisions and reviews on some major books of AI were done to understand the concepts of Artificial Intelligence (AI) and Natural Language Processing (NLP) that can be applied on the development of the proposed system. The main concept meant here is the NLP which is the techniques used in the system development. The process of documenting the report was analyzed by reviewing prior theses related to AI and NLP.

2.1.4 Discussion with supervisor

Discussions were made frequently during the development of the system. The supervisor gave many useful ideas and comments for the project. She guided the way and methods on doing the project. She also suggested how the system would be turned out in the end. This is important to ensure the development of the system is always on the right path.

2.1.2 Interviews

A few interviews had been conducted with some of the people that make the Bahasa Melayu stemming as the major domain in their researches. One of them is developing the same system but in another language. I gained much information from her to further my research during the design process.

2.1.3 Reading material and thesis reviewing

Revisions and reviews on some major books of AI were done to understand the concepts of Artificial Intelligence (AI) and Natural Language Processing (NLP) that can be applied on the development of the proposed system. The main concept meant here is the NLP which is the techniques used in the system development. The process of documenting the report was analyzed by reviewing prior theses related to AI and NLP.

2.1.4 Discussion with supervisor

Discussions were made frequently during the development of the system. The supervisor gave many useful ideas and comments for the project. She guided the way and methods on doing the project. She also suggested how the system would be turned out in the end. This is important to ensure the development of the system is always on the right path.

2.2 Artificial Intelligence Technique

Natural Language Processing

Humans communicate among themselves using natural language. Computers have some languages they use to communicate so they can link to each other. Humans do understand the languages used by computers. However, computers do not.

During these past few decades, researchers of Artificial Intelligence (AI) have been doing various researches on how computers would be able to understand the languages used by humans. So, then came the new technology called Natural Language Processing (NLP).

Basically, NLP has been categorized into five components. They are as follows:

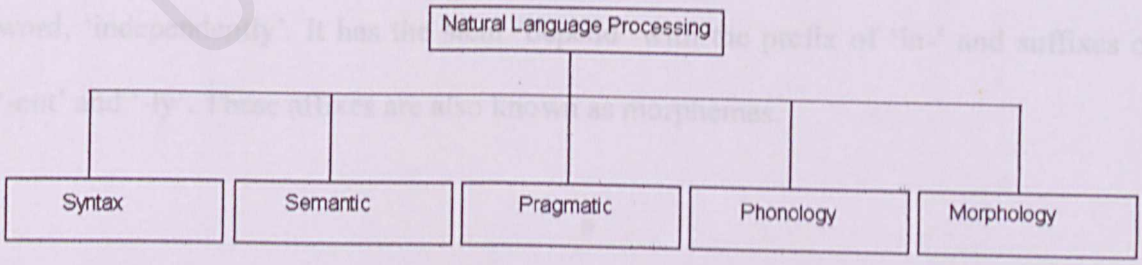

Figure 2.1 Natural Language Processing Components

The components we are focusing here is morphology as stemming is a part of the subcomponents of morphology.

Stemming

Stemming as the reaserchers define it:

1. A process of identifying morphological variantsof words which have similar semantic intrepetation and reducing it to its stem or root form. [Hoyle,1978]
2. An algorithm which reduces all words with the same root to a single form by stripping each word of its derivational and inflectional affixes. [Lovins, 1969]
3. Truncation of words, disregarding of prefixes or suffixes. [McCord, 1999]

The main purpose of stemming is to reduce the lengthy affixed words (words that have prefixes, suffixes and infixes) to their very own stem or root word. Like the English word, 'independently'. It has the stem 'depend' with the prefix of 'in-' and suffixes of '-ent' and '-ly'. These affixes are also known as morphemes.

Morphology can be subcategorized into 2 subcomponents and they are inflectional and derivational. Inflectional is a way to make new words from the same word without changing the full meaning of the words. Example:

Membuat
Membuatkan
Dibuat
Diperbuat

root 'buat' and the meaning are almost the same

In inflectional, like the example above, the morphemes (affixes) are added to the stem or root to make new words but the meanings do not change very much. All have the same idea of do, doing or being done.

Derivation also involves suffixes or prefixes that are not independent words. As an example the word 'makanan' is derived from the word 'makan' by prefixing it with the derivational prefix '–an', but this changes the whole meaning of it: 'makanan' is food while 'makan' is eat.

Chapter 3

Methodology

CHAPTER 3

METODOLOGY

3.1 SOFTWARE DEVELOPMENT LIFE CYCLE

Software products are always oriented on users like any product available in the market. The most important thing in developing any product prominently is customer satisfaction. To ensure this important thing will be on target, a model of system development must be applied to every system project being developed.

Software Development Life Cycle (SDLC) is a sequence of processes that anyone or any organization in a team of developing a system must follow in order to make sure that their system will come out to be as expected. It is actually an informal sequence of processes of developing a system used by the team but many considered it as a standardized method inherited from previous projects of the same team or others.

Many organizations used it to describe their approaches in the development of their projects. Some common SDLC today are Spiral model, Waterfall model, Extreme Programming, Incremental and Iteration model and so on. They are quite different from one another as each serves for certain types of project. For an instance, a project to develop a huge system that will be involved by many individuals and organization is suited with Spiral model while a small, in-house project is well matched with Incremental and Iteration model.

Detailed study and analysis must be brought up to selected the best model to be applied in the system development. Managers, stakeholders, programmers and all individuals involved in the development team must sit in a round table and discuss in for the short and long term scope and finally decide the model they should use which will be the main guide for the whole development processes.

3.2 DEVELOPMENT OF METHODOLOGY

The SDLC selected for the basis of the development of this system project is the linear model or some might call it waterfall model. Linear model is the traditional model of software development is commonly used in the development of previous AI projects.

Basically, the linear model consist of 6 phases of development to be go through. The phases are, in sequence,

- Planning
- Knowledge Definition
    - Source Identification & Selection
    - Acquisition Analysis & Extraction
- Knowledge Design
    - Definition
    - Detailed Design
- Code & Checkout
- Knowledge Verification
    - Formal Test
    - Test Analysis
- System Evaluation

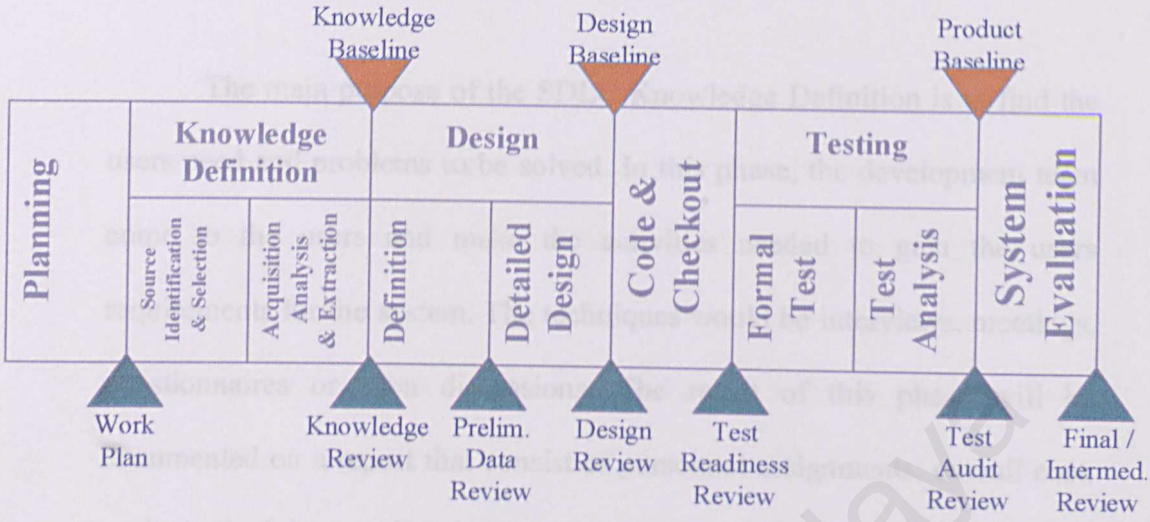

Figure 3.1: Linear model of SDLC

3.2.1 Planning

This is the first phase of developing a system. Here, the developers must establish the requirements for the system elements. There is the necessity of the system preview as it will interface with other elements such as hardware, people and other resources. Sometimes, the system must be re-engineered to improve its functionality. When the perfect proposed system came up, the development team will study the software requirement for the system.

3.2.2 Knowledge Definition

The main purpose of the SDLC Knowledge Definition is to find the users need and problems to be solved. In this phase, the development team come to the users and make the activities needed to gain the users requirements for the system. The techniques would be interviews, meetings, questionnaires or even discussions. The result of this phase will be documented on a report that consist of personnel assignments, overall cost, project schedule as well as its target and holds specific recommendations for the proposed system. The system analyst must study the domain of the system to be developed and also its required functionality, interface and overall performance too.

3.2.3 Knowledge Design

This phase is all about defining the whole software development process, the overall software structure and the cost to developed the system. To expand the points, the system architecture, the database design, the data structure and so on are all define here completely. It is definitely a very important phase as any fault created here will be hard to solved in the next processes of software development.

3.2.4 Coding & Checkout

Coding in the term of SDLC means the programmers decode the design into a form that computer can read. Complication would not exist too much if the design of the system was planned in detail. Programming tools such as compilers, interpreters and debuggers will be used to generate the codes. Depended on the system being developed, various types of high level programming languages such as Prolog, C++, Java or even Basic can be used for the coding purposes.

3.2.5 Knowledge Verification

This phase is contributed to test the system overall functionality. Testing begins during the coding of the system design progresses. This is especially done to detect the bugs committed from previous phase of coding. Different testing methods using several types of testing tools are available for testing purpose.

3.2.6 Maintenance

Maintenance phase is where the developers have to make the change after the system has been used by the target users. Some reason for the system to be change is it could be that unpredictable data being input into the system. Furthermore, maybe the users requirements changed after the system is created. The software should be set to accommodate post-maintenance changes whenever it needed.

3.3 JUSTIFICATION OF METHODOLOGY

This system picked the linear model as its SDLC model. The linear model is commonly used in any AI development. This linear model has some differences compared to the other linear models as it is specifically design to adapt to the AI development. The major differences that can be seen here is the phases with knowledge prefix to show that the system is predominantly related to knowledge that is the fundamental of AI.

3.4 CHAPTER SUMMARY

The selection of a good methodology is a fundamental task in any system development. The right methodology selected can be the best guide for developers to follow the sequence of phases in order to make sure that the system will result as expected perfectly in a good manner. The methodology chosen in this project development is the most suitable for an AI project based on the successfulness of previous similar AI projects. Linear model was proven for its ability to synchronize with knowledge-based system processes.

Chapter 4

System Design

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

The most essential phase of any system development is the system design as it is the main guidance of all the works involved during the development processes. System design is a process of creating the basis of the system's components, technologies being used, system tasks workflows, database design etc.

The main idea is to ensure the actual output should not be too altered from the system scopes and design proposed during the early phases of the development so that the main objectives will be achieved.

4.2 SYSTEM"S ALGORITHM

This algorithm is based from the former Porter algorithm:

- Step 1
  - Check input
    - ***IF*** word existed in dictionary ***THEN***
      - Output as stem
    - ***ELSE***
      - Proceed to Step 2

- Step 2
  - Check circumfix
    - ***IF*** word has circumfix ***THEN***
      - Send the word to exceptional word's part and back to Step 1
    - ***ELSE***
      - Proceed to Step 3

- Step 3
  - Check the suffix
    - ***IF*** word has suffix ***THEN***
      - Stem the word for its suffix and back to Step 1
    - ***ELSE***

- Proceed to Step 4

- Step 4
  - Check the prefix
    - ***IF*** word has prefix ***THEN***
      - Stem the word for its prefix and back to Step 1
    - ***ELSE***
      - Proceed to Step 4

- Step 4
  - Check the word for its first letter
    - ***IF*** the first letter is 'M' ***THEN***
      - Change to letter 'P' or and back to Step 1
    - ***IF*** the first letter is 'N' ***THEN***
      - Change to letter 'T' and back to Step 1
    - ***IF*** the first letter is 'Y' ***THEN***
      - Change to letter 'S' and back to Step 1
    - ***IF*** the first letter is a vowel ***THEN***
      - Add 'K' as the first letter and back to Step 1
    - ***ELSE***
      - Output same as Input

4.3 INPUT & OUTPUT DESIGN

This system has a workflow as below:

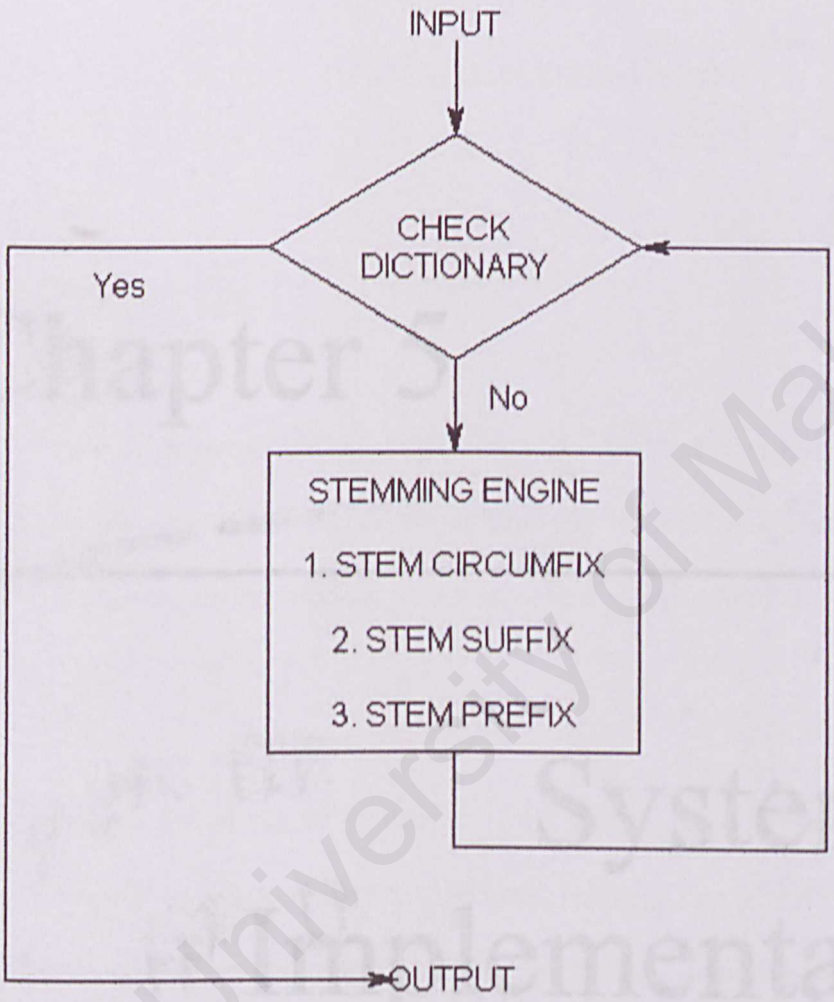

Figure 4.1 Input and Output Flow Diagram

Chapter 5

System Implementation

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

System implementation is a process to ensure the developed system is functioning and the required components of the system can be used and customized. This can be further translated as the process to create a new system from the existing one or even from a scratch.

Knowledge of the environment of the development is the fundamental thing to consider before any system can be developed. The selection of the software as well as the hardware requirements is as important as the former. In this section, the most notable things are the coding of the system design and the database development.

5.2 SYSTEM DEVELOPMENT

This is the most essential phase is any project. The purpose of the system development is to create the system using the tools required to build the system and make all the components of the system mentioned in the system design section are fully constructed.

Before the system implementation started, the development environment of the system must be tested and consider in order to determine the good flow of movement during the process. The pros and contras of the system should be corrected and updated and any occurrence of mistakes must be discarded immediately. Besides, the good selection of tools for the project will guarantee the best result in the end.

5.2.1 DEVELOPMENT TOOLS

5.2.1.1 Hardware Requirements

The followings are the minimal hardware requirements in order to run the system efficiently:

- 1 GHz Processor
- 10 MB of Hard Disk Storage
- 128 MB of RAM

And the recommended hardware requirements are as follow:

- 1.8 GHz Processor

- 10 MB of Hard Disk Storage
- 255 MB of RAM

5.2.1.2 Software Requirements

Below is the list of the software needed to develop and run the system:

- Microsoft Windows XP as the Operating System
- WIN PROLOG 4.040 and above
- Text Editor such as Notepad

5.2.2 Database Development

To make sure the system can always refer to the correct word after it has been stemmed, a database is a compulsory. This system uses a simple text data file to represent the collection of words existed in Bahasa Melayu. The database was created using Notepad, a text editor from Microsoft Corporation.

However, the database used in the system is a self created one and it does not comprise all the words existed in Bahasa Melayu. This means the developer and the user can always add new words to the database file as known here as 'kamus'. To make sure the system run exceptionally correct to the grammar of the Bahasa Melayu, it must use the database from 'Dewan Bahasa dan Pustaka' which is the most complete database of Bahasa Melayu words for the time being.

5.2.3 Coding System

Coding is an interactive process where it is done to the level of the developer's satisfactory. Coding is a series of command written in certain programming languages so that the programming codes can be executed to make the system functioning. A good coding will make the system easier to understand by the future developers especially for the systems that have a higher degree of difficultness.

5.2.3.1 Coding Approach

There are two approaches of coding, and they are the top-down and the bottom-up approaches. Both of these approaches encourage the coding process of the higher level to be done first rather than the lower ones which will be coded later in the process. Only after the higher level of modules have been coded, there are some references need to be done in order to code the rest of the modules in particular the lower level ones.

For the developed system, the approach used here is the bottom-up coding because it covers the execution of the higher level and the major modules. This can avoid the repetition in coding process that will result in the changing a series of codes if requirements change. This will effect the project timeline in the implementation phase and increase the system operational costs.

The advantages of this approach:

1) The testing will be executed in some functions that are used after they have been completed.
2) The critical function can be coded first to test the effectiveness. This approach also has a good interaction in providing the effect of the coding to obtain the program stability.

5.2.3.2 Coding Style

The style of coding is an important attribute to the source code to determine the understandability of the coding. It is used in the development of the system for the reason of these criteria:

1) The variable naming that does not repeat
2) Functions and declaration that is easy to understand
3) The standardization of the coding so that it looks neat and assembled.
4) The complexity of words should be simplified so that it will not look confusing.

5.3 Summary

The system Implementation is one of the most important phase in the development of any system. Some of the works like database building, system coding of all the components are important to make sure the smoothness of the system execution. The good selection of tools, the requirements of hardware and software is fundamental so that the result will meet the objectives stated.

Chapter 6

System Testing

CHAPTER 6

SYSTEM TESTING

6.1 INTRODUCTION

System testing is the best way to detect the error and test the usability of the system. Every new hardware, software and procedure must be tested consistently and must be carried throughout the development process and not only during the final stage of the development process. The successfulness of the system testing will produce a high quality system and meets the requirements. This will cut the total of time, costs and work. The testing strategy used in this projects are unit testing, module testing and system testing.

6.2 TESTING OBJECTIVES

The objectives for the testing used in the development of this project are listed as follows:

1. to achieve a high quality in the successfulness of the system that covers the perfection, accurateness, stability and strength of the system in all its documentation.
2. to ensure that the system can execute all of its functions correctly as being expected.
3. to cut the overall costs in the maintenance process of the system.
4. a way to detect and fix any possible errors.

6.3 UINT TESTING

Unit testing is the testing design to cater the smallest components of the developed system that are tested individually without the emergence of other components. An example is to make sure the system can read input from the user. Another example is to make sure that it can print the output from the system.

Some of the strategy that can be obtained from the testing are:

1. Code testing, from here we can detect the errors occured in the algorithm, data and syntax.
2. compare the code with the specifications as well as the design to ensure the relevant cases are considered.

6.4 MODUL TESTING

This is the next step after the unit testing had been done. It covers the module the user and the developer. Data is being input to the system in order to test each code in every module is working when all the module are called during the integration testing. If there are some errors occurred in any the module, that particular module must be tested to detect them.

6.5 INTEGRATION TESTING

After making all of the modules working properly and achieved the objectives, they are all gathered in a working system. In other words, integration testing is a module verification process that work altogether as been described in the design specification.

To test all the module simultaneously is quite difficult. It must be performed consecutively. The testing will ensure that relation or interaction will be done appropriately. The approach that had been used is 'non-incremental' where all the modules were gathered before being tested. The integration testing had been chosen as in the sense of the system in developed by a developer and only he or she knows the modules created.

6.6 SYSTEM TESTING

The purpose of this testing is to verify that the system has fulfill the user requirements. It ensures all the subsystem can be combine to generate a full functioning system. The testing can be seen as the platform of error detection that might be occurred such as the interaction between the system and the hardware.

The other thing is that from here we can know the user-friendliness and usability of the program and see how easy the system to be used without referring to the user manual. A good program will always have the user using the system without referring to the manual frequently.

6.7 TESTING SAMPLE

Here are some of the sample of test data given to the system on the phase of system testing:

INPUT	OUTPUT
mengalahkan	kalah
bukuku	buku
menterbalikkan	balik
penarik	tarik
ketidakupayaan	tidak upaya
terjatuh	jatuh
dibiakkan	biak

Table 6.1 Sample testing data

6.8 THE IMPORTANCE OF TESTING

Listed here are the major importance of testing:

1. the usability where it is measured based on the simple development of the system interface.
2. the trust ability of the system
3. to test the response time of the system for the final user purpose.

6.9 SUMMARY

Testing is the phase need to be done before delivering the system to its final user. It should be able to detect errors. The user interface must be clean and slate so that the user can use the system from the moment he or she sees the system. The system functionality must be working. And finally it must all achieves its objectives and scopes.

Chapter 7

System Evaluation

CHAPTER 7

SYSTEM EVALUATION

7.1 INTRODUCTION

The system evaluation can be considered complete if the system is working, that is to be used in the real world application. The post-work after the delivery of the system to the user is actually the maintenance of the system. One of the difference between the hardware system and the software system is software system is build to adapt with future changes. This means that the system will always evolutes from time to time due to the changing of the users requirements.

This chapter will evaluate the system from the point view of the developer. This evaluation will include the problems and limitations as well as the further recommendation of the system .

7.2 SYSTEM MAINTENANCE

The activities in the system maintenance is focused on four type in the current evolution system:

7.2.1 Corrective Maintenance

This type of maintenance is performed after testing the output in the system. The errors that might be found by the user and he or she will report it to the developer. This kind of maintenance is normally associated with the coding error, design error or while analyzing the functional and non-functional requirements.

7.2.2 Adaptive Control

The maintenance is performed for the components or the parts that are networked in the system application. This means that if there is something to fix on some of the modules in the system, all the parts that have connections among them must be fixed as well.

7.2.3 Perfective Maintenance

It might be useful in the future as this type of maintenance is not based on the error factor. Normally, it performed when there is some changes in the requirements so that the system will work better. Some changes must also be made to the software documentation to suits the maintenance.

7.2.4 Preventive Maintenance

The purpose of this maintenance is similar to the former but more on changes for the system to prevent errors and mistakes. This can involve the upgrading on the error control so that the system can control any probability occurred. This can be performed if the developer can trace error or bug that cannot give any effect to the system.

However, because of the system had just been developed, and has not being tested by the final user, the maintenance could not be performed for the time being. Nevertheless, the recommendation pre-mentioned before is hoped to give the sketch on how the system can be maintain in the future.

7.3 PROBLEMS FACED

Some of the problems faced during the development of the systems are:

1) The difficulty to determine the projects objectives and scopes.
2) The design and the whole system concept
3) The coding process
4) The database of the pure Bahasa Melayu
5) Lack of similar system in Bahasa Melayu

a. SYSTEM'S STRENGTH

The strength of the system that can be seen:

1) The procedure of the coding is easy to understand and follow.
2) It can stem some difficult words that cannot be stemmed by previous system.
3) Easy to use by user.
4) The system does not require a large amount of space.

b. SYSTEM'S LIMITATION

Some of the limitations that make the system is not the most complete stemmer for Bahasa Melayu:

1) It is unable to stem some kind of words in Bahasa Melayu such as 'kata ganda' for the verb in Malay, combined nouns, misspelled words and 'kata ganda separa'.
2) The database, 'kamus', in provided by the developer is not the original complete stem of Bahasa Melayu words.
3) It cannot handle strings of words or sentences.

7.6 FUTURE RECOMMENDATIONS

It is hoped that the future developer of the stemming engine of Bahasa Melayu words can build a system that can deal with all the limitations pre-mentioned above. The developer can also include domain in science related words and the words from classical Bahasa Melayu.

7.9 SUMMARY

This project of the stemming engine for Bahasa Melayu words is one of the several researches to generate a stable, reliable and most complete Bahasa Melayu stemmer ever. Even though this system has limitations that is not a limitation for the previous system, it has shown a new algorithm of designing a stemmer for Bahasa Melayu in Prolog language. It is hoped that the future researches of similar of system can build a more complete stemmer using Prolog as Prolog is one of the most powerful language in Artificial Intelligence field.

REFFERENCE

Internet:

1. http://tartarus.org/~martin/PorterStemmer/
2. http://tartarus.org/~martin/PorterStemmer/prolog.txt
3. http://snowball.tartarus.org/texts/quickintro.html
4. http://snowball.tartarus.org/algorithms/porter/stemmer.html
5. http://ms.wikipedia.org/wiki/Tatabahasa
6. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
7. http://wotan.liu.edu/docis/dbl/coling/

Books:

1. Zainal Abidin Ahamd (Za'ba) (2000). *Pelita Bahasa Melayu Penggal I – III.* Dewan Bahasa dan Pustaka.
2. Hajah Noresah bt Baharom (2005). *Kamus Dewan edisi keempat.* Dewan Bahasa dan Pustaka.
3. Ismail bin Dahaman & Manshoor bin Hj Ahmad (2004). *Daftar Kata Bahasa Melayu.* Dewan Bahasa dan Pustaka.
4. Koh, Boh Boon (1978). *The Teaching Of Malay Affixes.* Fajar Bakti.
5. Rogers, Jean B. (1986). *A Prolog Primer.* Addison-Wesley Publishing Company.
6. Sterling, Leon & Shapiro, Ehud (1994). *The Art of Prolog.* The MIT Press.
7. Covington, Michael A. & Nute, Donald & Vellino, Andre (1997). *Prolog Programming In Depth.* Prentice Hall.

USER MANUAL

Step 1 Open WINPROLOG program.

Step 2_ Open File 'stemmer.pl'

Step 3 Open dictionary ('kamus.txt') or you can create your own dictionary by creating a text file using notepad. Make entry every single line and put a full stop after it. Save it as '*name*.txt', where name is of your choice. Then, change the code for '*name*.txt' in the program.

Step 4 Compile all the files.

Step 5 Type 'start.' at the console window and use the program.

APPENDIX

Source code:

```
start:-nl, write('Sila masukkan perkataan (Taip `exit` untuk tamat)>> '),
    go,!,nl.

go:-read(Word),Word \= exit,atom_string(Word,List),test(List).

test(List):-see('special.txt'),find(List,3,Found),List==Found-
>Found=List,write(List),seen;stem(List),seen.

conc([],L,L).
conc([X|L1],L2,[X|L3]):-conc(L1,L2,L3).

%circumfix
%---------

stem(Newlist):-conc("keber",After,Newlist),conc(Before,"ankah",After),cetak(Before).
stem(Newlist):-conc("keber",After,Newlist),conc(Before,"anlah",After),cetak(Before).
stem(Newlist):-conc("keber",After,Newlist),conc(Before,"annya",After),cetak(Before).
stem(Newlist):-conc("keber",After,Newlist),conc(Before,"anku",After),cetak(Before).
stem(Newlist):-conc("keber",After,Newlist),conc(Before,"anmu",After),cetak(Before).
stem(Newlist):-conc("keber",After,Newlist),conc(Before,"an",After),cetak(Before).

stem(Newlist):-conc("keter",After,Newlist),conc(Before,"ankah",After),cetak(Before).
stem(Newlist):-conc("keter",After,Newlist),conc(Before,"anlah",After),cetak(Before).
stem(Newlist):-conc("keter",After,Newlist),conc(Before,"annya",After),cetak(Before).
stem(Newlist):-conc("keter",After,Newlist),conc(Before,"anku",After),cetak(Before).
stem(Newlist):-conc("keter",After,Newlist),conc(Before,"anmu",After),cetak(Before).
stem(Newlist):-conc("keter",After,Newlist),conc(Before,"an",After),cetak(Before).

stem(Newlist):-conc("ketidak",After,Newlist),conc(Before,"ankah",After),cetak(Before).
stem(Newlist):-conc("ketidak",After,Newlist),conc(Before,"anlah",After),cetak(Before).
stem(Newlist):-conc("ketidak",After,Newlist),conc(Before,"annya",After),cetak(Before).
stem(Newlist):-conc("ketidak",After,Newlist),conc(Before,"anku",After),cetak(Before).
stem(Newlist):-conc("ketidak",After,Newlist),conc(Before,"anmu",After),cetak(Before).
stem(Newlist):-conc("ketidak",After,Newlist),conc(Before,"an",After),cetak(Before).

stem(Newlist):-conc("ke",After,Newlist),conc(Before,"ankah",After),cetak(Before).
stem(Newlist):-conc("ke",After,Newlist),conc(Before,"anlah",After),cetak(Before).
stem(Newlist):-conc("ke",After,Newlist),conc(Before,"annya",After),cetak(Before).
stem(Newlist):-conc("ke",After,Newlist),conc(Before,"anku",After),cetak(Before).
stem(Newlist):-conc("ke",After,Newlist),conc(Before,"anmu",After),cetak(Before).
stem(Newlist):-conc("ke",After,Newlist),conc(Before,"an",After),cetak(Before).
```

```
%prefix
%-------

stem(Newlist):-conc("bel",After,Newlist),cetak(After).
stem(Newlist):-conc("ber",After,Newlist),cetak(After).
stem(Newlist):-conc("be",After,Newlist),cetak(After).
stem(Newlist):-conc("diper",After,Newlist),cetak(After).
stem(Newlist):-conc("diter",After,Newlist),cetak(After).
stem(Newlist):-conc("di",After,Newlist),cetak(After).
stem(Newlist):-conc("memper",After,Newlist),cetak(After).
stem(Newlist):-conc("menge",After,Newlist),cetak(After).
stem(Newlist):-conc("meng",After,Newlist),cetak_k(After).
stem(Newlist):-conc("meny",After,Newlist),cetak_s(After).
stem(Newlist):-conc("mem",After,Newlist),cetak_p(After).
stem(Newlist):-conc("men",After,Newlist),cetak_t(After).
stem(Newlist):-conc("me",After,Newlist),cetak(After).
stem(Newlist):-conc("penge",After,Newlist),cetak(After).
stem(Newlist):-conc("peny",After,Newlist),cetak_s(After).
stem(Newlist):-conc("peng",After,Newlist),cetak_k(After).
stem(Newlist):-conc("pen",After,Newlist),cetak_t(After).
stem(Newlist):-conc("pem",After,Newlist),cetak_p(After).
stem(Newlist):-conc("per",After,Newlist),cetak(After).
stem(Newlist):-conc("pel",After,Newlist),cetak(After).
stem(Newlist):-conc("pe",After,Newlist),cetak(After).
stem(Newlist):-conc("se",After,Newlist),cetak(After).
stem(Newlist):-conc("ter",After,Newlist),cetak(After).
stem(Newlist):-conc("te",After,Newlist),cetak(After).
stem(Newlist):-conc("kau",After,Newlist),cetak(After).
stem(Newlist):-conc("ku",After,Newlist),cetak(After).


%suffix
%------

stem(Newlist):-conc(Before,"ankah",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"anlah",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"annya",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"anku",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"anmu",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"an",Newlist),cetak(Before).

stem(Newlist):-conc(Before,"kankah",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"kanlah",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"kannya",Newlist),cetak(Before).
```

```
stem(Newlist):-conc(Before,"kanku",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"kanmu",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"kan",Newlist),cetak(Before).

stem(Newlist):-conc(Before,"ikah",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"ilah",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"inya",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"imu",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"iku",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"i",Newlist),cetak(Before).

stem(Newlist):-conc(Before,"kah",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"lah",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"pun",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"ku",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"mu",Newlist),cetak(Before).
stem(Newlist):-conc(Before,"nya",Newlist),cetak(Before).


cetak(Wordlist):-atom_chars(Rootword,Wordlist),
                 write('Perkataan asal ialah '),
                 write(Rootword),
                 go,nl.

cetak_k(Wordlist):-atom_chars(Rootword,Wordlist),
                 write('Perkataan asal ialah'),
                 write('k'),write(Rootword),
                 go,nl.

cetak_sWordlist):-atom_chars(Rootword,Wordlist),
                 write('Perkataan asal ialah'),
                 write('s'),write(Rootword),
                 go,nl.

cetak_t(Wordlist):-atom_chars(Rootword,Wordlist),
                 write('Perkataan asal ialah'),
                 write('t'),write(Rootword),
                 go,nl.

cetak_p(Wordlist):-atom_chars(Rootword,Wordlist),
                 write('Perkataan asal ialah'),
                 write('p'),write(Rootword),
                 go,nl.
```