

**SISTEM APLIKASI KAMUS PENERJEMAH BAHASA INDONESIA
- LONTARA BUGIS DENGAN MENGGUNAKAN METODE
*BINARY SEARCH DAN PARSING TREE***

***THE APPLICATION OF THE INDONESIAN-BUGINESE LONTARA
TRANSLATING DICTIONARY USING THE METHOD OF THE
BINARY SEARCH AND PARSING TREE***

JUMADIL NANGI



**PROGRAM PASCA SARJANA
UNIVERSITAS HASANUDDIN MAKASSAR
MAKASSAR
2013**

SISTEM APLIKASI KAMUS PENERJEMAH BAHASA INDONESIA
- LONTARA BUGIS DENGAN MENGGUNAKAN METODE
BINARY SEARCH DAN PARSING TREE

Tesis

Sebagai Salah Satu Syarat untuk Mencapai Gelar Magister
Program Studi Teknik Elektro

Disusun dan diajukan oleh:

JUMADIL NANGI

Kepada :

PROGRAM PASCA SARJANA
UNIVERSITAS HASANUDDIN MAKASSAR
MAKASSAR

2013

PERNYATAAN KEASLIAN TESIS

Yang betanda-tangan di bawah ini :

Nama : Jumadil Nangi
NIM : P2700211462
Program Studi : Teknik Elektro
Konsentrasi : Teknik Informatika

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan tulisan atau pemikiran orang lain. Adapun kutipan atau rujukan sebagai sumber informasi yang saya gunakan dari penulis lain, telah saya sebutkan namanya pada daftar pustaka tesis ini.

Apabila dikemudian hari ada terbukti bahwa tesis ini adalah hasil karya orang lain maka saya bersedia menerima sanksi apapun sesuai peraturan yang berlaku.

Makassar, Juli 2013

Penulis,

(Jumadil Nangi)

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kehadiran Allah Subhanahu WaTa'ala atas segala limpahan rahmat, taufiq, hidayah serta inayah-Nya kepada penulis sehingga dapat menyelesaikan penelitian ini yang berjudul "**SISTEM APLIKASI KAMUS PENERJEMAH BAHASA INDONESIA-AKSARA LONTARA BUGIS MENGGUNAKAN *BINARY SEARCH* DAN *PARSING TREE***".

Penulis menyadari banyaknya kendala dalam penyusunan tesis ini, namun demikian penulis dapat menyelesaikannya berkat bantuan dari berbagai pihak baik secara moril maupun materil. Karena itu dalam kesempatan ini perkenankan penulis menyampaikan ucapan terima kasih dan penghargaan setinggi - tingginya kepada :

1. Kedua orang tua tercinta yang membesarkan dan mendidik penulis serta senantiasa mendoakan dengan tulus ikhlas dan senantiasa memberikan dukungan moril sehingga penulis mendapatkan kemudahan dalam penulisan tesis ini. Serta tak lupa kepada saudara dan keluarga yang menjadi salah satu motivasi penulis dalam menyelesaikan tesis ini.
2. **Prof.Dr.Ir.H. Nadjamuddin Harun, MS**, sebagai Ketua Komisi Penasihat dan **Dr.Eng. Syafaruddin, ST, M.Eng**, sebagai anggota Komisi Penasihat atas bantuan dan bimbingan yang telah diberikan mulai dari proposal dan pelaksanaan penelitian hingga penulisan tesis

3. Tim Penguji **Dr.Elyas Palantei, ST.,M.Eng, Dr.Ing Faizal Arya Samman, ST, MT** dan **DR. Adnan,ST.,MT** yang telah memberikan saran, kritik dan pertimbangan dalam penyempurnaan Tesis ini.
4. **Prof.Dr.Ir.Salama Manjang,MT** selaku ketua Program Studi S2 Teknik Elektro Universitas Hasanuddin.
5. Kepada semua dosen-dosen Universitas Hasanuddin, yang telah memberikan Ilmunya kepada penulis tanpa pamrih, serta staf pegawai yang sudah banyak membantu secara administrasi hingga selesainya tugas akhir.
6. Teman-teman PascaMelekIT 2011 something special dalam pencarian tuntutan keilmuan di tengah kebersamaan yang tak terhingga.

Makassar, Juli 2013

Jumadil Nangi

ABSTRAK

Jumadil Nangi. Sistem Aplikasi Kamus Penerjemah Bahasa Indonesia-Lontara Bugis Menggunakan Metode *Binary Search* dan *Parsing Tree* (dibimbing oleh **Nadjamuddin Harun** dan **Syafaruddin**)

Di propinsi Sulawesi Selatan terdiri dari 2 (dua) etnis yaitu etnis Bugis dan etnis Makassar yang menggunakan penulisan Aksara Lontara. Pelajaran Bahasa Daerah, dalam hal ini Bahasa Bugis dan penulisan Aksara Lontara Bugis merupakan pelajaran yang kurang menarik dan cenderung sulit untuk dipahami. Sementara itu seringkali dibicarakan masalah pelestarian adat, budaya, tradisi dan lain sebagainya. Penelitian ini merupakan suatu proyek desain untuk membangun sebuah *software* Aplikasi Kamus Penerjemah Bahasa Indonesia-Aksara Lontara Bugis dengan tujuan mengenalkan perbendaharaan kosakata dalam bahasa Bugis beserta penulisan Aksara Lontara pada masyarakat umum

Pada penelitian ini dibuat suatu sistem Aplikasi kamus penerjemah bahasa Indonesia-Aksara Lontara Bugis menggunakan metode *binary search* dan *Parsing tree*. Metode *parsing tree* digunakan untuk memecah kalimat menjadi kata serta menggabungkan kata-kata menjadi kalimat, sedangkan metode *binary search* digunakan untuk melakukan proses pencarian kata-kata. Pembuatan *software* menggunakan pemrograman PHP(*Hypertext Preprocessor*) dan dalam sistem navigasinya dibuat sesederhana mungkin agar mudah dalam pengaksesanya.

Penelitian ini akan menghasilkan aplikasi kamus penerjemah bahasa Indonesia-Aksara Lontara Bugis yang dapat digunakan oleh masyarakat umum sebagai sarana pembelajaran untuk mengenal penulisan Aksara Lontara Bugis, dan pada akhirnya mampu memberikan kontribusi dalam upaya pelestarian budaya asli Indonesia.

Kata Kunci : *aplikasi kamus penerjemah, Aksara Lontara Bugis, binary search, parsing tree*

ABSTRACT

Jumadil Nangi. *Dictionary Translator Application System-Lontara Indonesian Bugis Method Using Binary Tree Search and Parsing (guided by Nadjamuddin Harun and Syafaruddin)*

In the province of South Sulawesi consists of 2 (two) ethnic groups, including ethnic Bugis and Makassar are using Lontara script writing. Regional language lessons, in this case language and writing script Lontara Bugis Bugis is a lesson that is less attractive and tend to be difficult to understand. While it is often talked about the preservation of indigenous issues, culture, traditions and so forth. This study is a design project to build a software application-Indonesian Dictionary Translator Lontara Bugis script with the aim of introducing vocabulary vocabulary in Bugis along with script writing on public Lontara

In this research, a system of application-Indonesian dictionary translator Lontara Bugis script using the binary tree search and parsing. Parsing tree method is used to break the sentence into words and combine words into sentences, whereas binary search methods used to conduct the search process -said. Software development using PHP (Hypertext Preprocessor) and the navigation system as simple as possible for easy in pengaksesanya.

This research will result in application-Indonesian dictionary translator Lontara Bugis script that can be used by the general public as a means of learning to recognize Lontara Bugis script writing, and in the end were able to contribute to the preservation of indigenous cultures of Indonesia.

Keywords: *dictionary translator, Lontara Bugis script, binary search, parse tree*

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN	iv
KATA PENGANTAR	iv
ABSTRAK	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Rumusan Masalah	3
C. Tujuan Penelitian	4
D. Manfaat Penelitian	4
E. Batasan Masalah	5
BAB II TINJAUAN PUSTAKA	
A. Road Map Penelitian.....	6
B. Tinjauan Bahasa	9
1. Kamus.....	9
2. Lontara Bugis.....	10
3. Morfologi Bahasa Bugis	12

C. Metode Penerjemah	13
1. <i>Parsing Tree</i>	13
2. <i>Binary Search</i>	15
D. Proses Penerjemah	22
1. Proses <i>Parsing</i> Kalimat.....	22
2. Proses <i>Searching</i> (pencarian).....	23
3. Proses Penggabungan kata.....	31
4. Mekanisme Penerjemah Lontara Bugis	32
E. Model Perancangan Sistem	34
 BAB III METODE PENELITIAN	
A. Tahapan Penelitian	41
B. Jenis Penelitian	42
C. Teknik Pengumpulan Data.....	42
D. Perancangan Sistem.....	43
E. Indikator Output.....	49
F. Kerangka Pikir	50
G. Jadwal Penelitian	51
 BAB IV HASIL DAN PEMBAHASAN	
A. Analisis Sistem.....	54
B. Perancangan Aplikasi	56
1. <i>Use Case Diagram</i> sistem Penerjemah	57
2. <i>Sequence Diagram</i>	58
3. <i>Class Diagram</i>	60

4. <i>Activity Diagram</i>	60
5. <i>Statechart Diagram</i>	62
C. Implementasi Sistem	65
1. <i>Interface User</i>	65
2. <i>Interface Admin</i>	67
D. Pengujian Sistem.....	69
1. Pengujian <i>White Box Testing</i>	69
2. Pengujian <i>Black Box Testing</i>	72
3. Analisa Pengenalan Pola dan Probabilitas	74
4. Pengujian Pengenalan Kata dan Kalimat.....	76
5. Pengujian Kuisisioner	78
6. <i>Upload Online Aplikasi Kamus</i>	79
BAB V KESIMPULAN DAN SARAN	
A. Kesimpulan	81
B. Saran	82
DAFTAR PUSTAKA.....	83
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
1. Gambar 2.1 Aksara Lontara Dengan Penyebutannya	12
2. Gambar 2.2 <i>Top Down Parser</i>	14
3. Gambar 2.3 <i>Bottom-Up Parser</i>	15
4. Gambar 2.4 Kumpulan Data	18
5. Gambar 2.5 <i>Looping Pertama Binary Search</i>	18
6. Gambar 2.6 <i>Looping Kedua Binary Search</i>	19
7. Gambar 2.7 Kumpulan Kata Terurut	24
8. Gambar 2.8 Proses Pencarian <i>Looping Pertama</i>	25
9. Gambar 2.9 Proses <i>Looping Kedua</i>	26
10. Gambar 2.10 Nilai Tengah <i>Looping Kedua</i>	26
11. Gambar 2.11 Proses <i>Looping Ketiga</i>	27
12. Gambar 2.12 Proses Penemuan Kata “Ibu”	27
13. Gambar 2.13 Proses <i>Looping Pertama</i> Kata “Makan”	28
14. Gambar 2.14 Proses <i>Looping Kedua</i> Kata “Makan”	29
15. Gambar 2.15 Proses Penemuan Kata “Makan”	29

16. Gambar 2.16 Proses <i>Looping</i> Pertama Kata “Nasi”	30
17. Gambar 2.17 Proses <i>Looping</i> Kedua Kata “Nasi”	31
18. Gambar 2.18 Proses Penggabungan Kata Menjadi Kalimat.....	32
19. Gambar 2.19 Rangkaian State Kata “manre”	34
20. Gambar 3.1 Proses Penerjemahan Kalimat.....	45
21. Gambar 3.2 <i>Flowchart</i> Pemecah Kalimat	46
22. Gambar 3.3 <i>Flowchart</i> Pencarian <i>Binary Search</i>	47
23. Gambar 3.4 <i>Flowchart</i> Penggabungan Kata.....	48
24. Gambar 3.5 Kerangka Pikir.....	50
25. Gambar 4.1 Rancangan Kamus <i>Database</i>	54
26. Gambar 4.2 Alur Penerjemahan Kalimat	55
27. Gambar 4.3 Alur Proses Penggunaan Sistem	57
28. Gambar 4.4 <i>Use Case Diagram</i> Aksara Lontara Bugis.....	57
29. Gambar 4.5 <i>Sequence Diagram</i> Input Kosakata.....	58
30. Gambar 4.6 <i>Sequence Diagram</i> Terjemahan Kata	59
31. Gambar 4.7 <i>Class Diagram</i>	60
32. Gambar 4.8 <i>Activity Diagram</i> Input Kata	61

33. Gambar 4.9 <i>Activity Diagram</i> Terjemahan Kata/Kalimat.....	62
34. Gambar 4.10 <i>Statechart Diagram</i> Input Kata.....	63
45. Gambar 4.11 <i>Statechart Diagram</i> Terjemahan	64
36. Gambar 4.12 <i>Interface</i> Terjemahan Indonesia-Aksara Bugis	65
37. Gambar 4.13 <i>Interface</i> Terjemahan Aksara Bugis-Indonesia	66
38. Gambar 4.14 <i>Interface</i> Penambahan Kosakata.....	67
39. Gambar 4.15 <i>Interface</i> Edit Kosakata.....	68
40. Gambar 4.16 <i>Interface</i> Hapus Kosakata.....	69
41. Gambar 4.17 <i>Flowchart</i> Pencarian <i>Binary Search</i>	70
42. Gambar 4.18 <i>Flowgraph</i> pengujian <i>white box testing</i>	70
43. Gambar 4.19 Proses Upload Template.....	78
44. Gambar 4.20 Proses <i>Upload Database</i>	79

DAFTAR TABEL

	Halaman
1. Tabel 2.1 Penamaan Jenis Kata	12
2. Tabel 2.2 Algoritma <i>Binary Search</i>	19
3. Tabel 2.3 Simulasi <i>Binary Search</i>	23
4. Tabel 2.4 Hasil Pencarian Kata.....	31
5. Tabel 2.5 Proses <i>Array</i> Kata	32
6. Tabel 2.6 Database Kamus.....	33
7. Tabel 2.7 Proses Kata <i>Manre</i>	33
8. Tabel 2.8 Simbol-Simbol <i>Use Case Diagram</i>	35
9. Tabel 2.9 Simbol-Simbol <i>Class Diagram</i>	27
10. Tabel 2.10 Simbol-Simbol <i>Sequence Diagram</i>	28
11. Tabel 2.11 Simbol-Simbol <i>Activity Diagram</i>	29
12. Tabel 2.12 Simbol-Simbol <i>statechart Diagram</i>	30
13. Tabel 3.6 Jadwal Penelitian	51
14. Tabel 4.1 Pengujian Penginputan Kosakata	72
15. Tabel 4.2 Pengujian Terjemahan Kalimat	73

16. Tabel 4.3 Rekap Kosakata Berdasarkan awalan huruf	74
17. Tabel 4.4 Pengujian Bahasa Indonesia – Lontara Bugis	75
18. Tabel 4.5 Pengujian Bahasa Bugis – Bahasa Indonesia	76
19. Tabel 4.6 Hasil Kuisisionere	77

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Era informasi merupakan periode yang melibatkan informasi dalam pengambilan keputusan, baik individu, perusahaan, instansi pemerintah maupun pendidikan. Informasi sudah semakin mudah diperoleh, sudah semakin bervariasi bentuknya dan semakin banyak pula kegunaannya. Teknologi informasi merupakan salah satu produk teknologi yang mengalami perkembangan yang sangat pesat saat ini, contohnya penggunaan komputer sebagai salah satu sarana penunjang dalam aplikasi yang dapat memberikan hasil lebih akurat untuk *output* sebuah sistem, tentunya bila sistem didalamnya telah berjalan dengan baik.

Pada hakikatnya, kemajuan teknologi dan pengaruhnya dalam kehidupan adalah hal yang tak dapat dihindari. Akan tetapi teknologi informasi juga dapat dipakai untuk melestarikan budaya Indonesia, saat ini sedikit sekali informasi yang berisi pengenalan budaya-budaya Indonesia yang sangat beragam, sehingga kebudayaan daerah kurang diketahui oleh masyarakat luas.

Berbagai macam suku dan ras yang ada di Indonesia, masing-masing memiliki kebudayaan yang berbeda, baik pakaian

khas, adat istiadat, maupun bahasa. Salah satu aset budaya yang perlu dilestarikan yaitu bahasa daerah sebab bahasa daerah merupakan bagian dari kebudayaan daerah dan sekaligus menjadi unsur kebudayaan nasional. Bahasa daerah harus tetap dipertahankan dan salah satu bahasa daerah itu adalah Bahasa Bugis.

Bahasa Bugis digunakan sebagai bahasa sehari-hari di beberapa daerah di Sulawesi Selatan seperti Kabupaten Bone, Soppeng, Wajo, Sidenreng Rappang, Sinjai, Bulukumba, Pinrang dan beberapa daerah lainnya. Walaupun pengucapan atau dialek terdapat perbedaan, namun juga memiliki banyak kesamaan sehingga dapat dimengerti atau diketahui maknanya, sehingga selain bahasa Indonesia, bahasa bugis menjadi salah satu media komunikasi yang umum digunakan di Sulawesi Selatan, apalagi orang-orang suku bugis tersebar dan dapat dijumpai di hampir setiap daerah di Sulawesi Selatan.

Aksara Bahasa Bugis secara tekstual dituliskan dalam bentuk huruf-huruf yang dikenal dengan istilah Aksara Lontara. Namun harus dipahami bahwa aksara lontara digunakan pada 2 (dua) etnis besar di Sulawesi Selatan yakni etnis Bugis dan etnis Makassar. Aksara Lontara juga merupakan lambang identitas daerah dan alat transformasi nilai-nilai luhur yang sangat berharga atau *indigenous knowledge*. Aksara Lontara adalah salah satu aset

kekayaan budaya yang berpotensi untuk dikembangkan sebagai objek wisata budaya daerah. Selain itu, dapat menjadi aset dan sumber pengembangan budaya nasional.

Melestarikan Aksara Lontara diperlukan berbagai sumber yang dapat menyediakan informasi tentang Aksara Lontara. Salah satu sarana yang dapat digunakan adalah aplikasi kamus lontara, khususnya lontara bugis. Oleh sebab itu, penelitian ini bermaksud untuk menyusun suatu aplikasi aksara lontara bugis dengan tema **“Sistem Aplikasi Kamus Penerjemah Indonesia-Lontara Bugis Dengan Menggunakan Metode *Binary Search* dan *Parsing Tree*”**.

B. Rumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan dalam penelitian ini dirumuskan sebagai berikut:

1. Bagaimana merancang Aplikasi Kamus Indonesia - Lontara Bugis.
2. Bagaimana menggabungkan metode *binary search* dan *parsing tree* dalam pencarian dan penulisan Aksara Lontara Bugis.
3. Bagaimana mengimplementasikan Aplikasi Kamus Indonesia-Lontara Bugis.
4. Bagaimana membentuk kalimat subjek, predikat dan objek (SPO).

C. Tujuan Penelitian

Beranjak dari latar belakang permasalahan, tujuan penelitian ini adalah sebagai berikut:

1. Untuk merancang Aplikasi Kamus Penerjemah Indonesia-Lontara Bugis.
2. Untuk mengimplementasikan Aplikasi Kamus Penerjemah Indonesia-Lontara Bugis.
3. Mengupload Aplikasi Kamus Penerjemah Indonesia-Lontara Bugis ke jaringan internet.
4. Untuk membentuk terjemahan dalam kalimat subjek, predikat dan objek.

D. Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini antara lain :

1. Menambah khasanah dan variasi dalam mengembangkan budaya lokal yang antara lain Aksara Lontara Bugis.
2. Mengingatkan generasi muda bahwa aksara Lontara adalah salah satu aksara nusantara yang dapat dikembangkan.
3. Sebagai sarana pembelajaran untuk mengetahui penulisan Aksara Lontara Bugis.
4. Sebagai referensi bagi peneliti lain untuk pengembangan sistem selanjutnya.

E. Batasan Masalah

Berikut ini adalah batasan-batasan yang digunakan untuk menyederhanakan masalah yang ada:

1. Aplikasi dapat menerjemahkan perkata dan kalimat dalam aksara latin.
2. Melakukan pencarian kata menggunakan metode *binary search*.
3. Memisahkan kata dari kalimat dengan menggunakan metode *parsing tree*.
4. Tidak membahas kalimat pasif dan bentuk lampau dalam proses penerjemahan bahasa Indonesia – Aksara Lontara Bugis.

BAB II

TINJAUAN PUSTAKA

Tinjauan pustaka dimaksudkan untuk memberikan gambaran atau batasan tentang teori-teori yang dipakai sebagai landasan teori. Teori yang melandasi kajian terjemahan bahasa Indonesia-Aksara Lontara Bugis ini tidaklah bersifat tunggal tetapi plural dan bersifat eklektik. Eklektik bermaksud kajian ini menganut berbagai konsep acuan dan didasarkan atas berbagai teori sepanjang konsep dan teori tersebut memiliki relevansi terhadap substansi kajian. Terjemahan adalah bidang ilmu terapan yang kajiannya melibatkan dua bahasa sehingga terjemahan tidak bisa dilepaskan dari konsep-konsep kebahasaan dan teori *linguistik*.

A. Road Map Penelitian

Beberapa penelitian terkait sebelumnya mengenai aplikasi kamus penerjemah dengan metode yang berbeda-beda, antara lain :

1. **“Pembuatan Kamus Elektronik Kalimat Bahasa Indonesia dan Bahasa Jawa Menggunakan Markov Model”**. Nur Afifah, Tri Budi Santoso. Jurusan Telekomunikasi, PENS-ITS. Pada penelitian di atas dititikberatkan pembuatan kamus elektronik Bahasa Indonesia dan Bahasa Jawa menggunakan *Markov Model* untuk mempercepat pada saat pemrosesan penerjemahan kata. Kelemahan menggunakan metode *Markov Model* adalah diperlukan lebih dari 1 kali proses *markov* untuk tiap kata,

sehingga memerlukan waktu lebih banyak dan semakin banyak jumlah kata yang ada maka pencariannya semakin lama dibandingkan dengan menggunakan *binary search*^[1].

2. **“Aplikasi Kamus Bahasa Indonesia – Jawa – Jawa Krama Berbasis Android”**. Ibnu Prihariyanto, Jurusan Teknik Informatika, STMIK AMIKOM Yogyakarta. Pada penelitian di atas dititikberatkan pada pembuatan kamus Bahasa Indonesia-Jawa dan aksara Jawa yang berbasis Android. Keuntungan dari aplikasi ini adalah dapat menampilkan arti dalam bahasa Jawa, Jawa Krama beserta Aksara Jawa dan berbasis mobile. Kekurangannya adalah aplikasi ini hanya melakukan penerjemahan untuk satu kata dan pencarian katanya hanya bisa digunakan untuk Bahasa Indonesia ke Bahasa Jawa^[2].
3. **“Penggunaan Metode Binary Search Pada Translator Bahasa Indonesia – Bahasa Jawa”**. Mike Yuliana. Jurusan Telekomunikasi, PENS-ITS. Pada penelitian di atas dititikberatkan pembuatan kamus penerjemahan kata dari Bahasa Indonesia ke Bahasa Jawa. Pembuatan tampilan menggunakan pemrograman Java dan dalam sistem navigasinya dibuat sederhana mungkin agar mudah dalam pengaksesannya. Diberikan gambaran bahwa pengguna hanya perlu memasukkan input berupa *text* dengan mengetik melalui *keyboard* komputer dan *software* akan menerjemahkannya ke dalam bahasa Jawa. Kekurangan aplikasi

ini hanya melakukan penerjemahan dan pencarian untuk 1 kata bukan untuk sebuah kalimat ^[3].

4. **“Aplikasi Kamus Dwibahasa Aceh-Indonesia Berbasis Java”.**

Viska Mutiawani, Juwita, Irvanizam. Program Studi Informatika, Jurusan Matematika, FMIPA, Universitas Syiah Kuala. Pada penelitian ini membahas mengenai pembuatan kamus penerjemah kata dwibahasa Aceh-Indonesia. Pembuatan aplikasinya menggunakan bahasa pemrograman java dan aplikasi yang dibuat sederhana mungkin agar mudah dalam pengaksesannya, dan proses penerjemahannya menggunakan *binary search* untuk mempermudah pencarian kata dalam basis datanya. Apabila kata dalam proses pencariannya tidak ditemukan maka akan ditampilkan pilihan kata-kata yang dianggap sepadan. Kekurangan dalam aplikasi ini adalah proses penerjemahan Dwibahasa Aceh-Indonesia sebatas dalam penerjemahan kata dan bukan mengenai penulisan Aksara Aceh ^[4].

Dalam perkembangannya bahasa Indonesia mengambil unsur atau kata dari bahasa lain, seperti bahasa daerah. Sudah banyak kosa kata dari bahasa daerah yang digunakan dalam bahasa Indonesia. Terlebih dahulu kata-kata itu disesuaikan dengan kaidah yang berlaku dalam bahasa Indonesia, baik itu dalam hal pengucapan maupun penulisannya.

B. Tinjauan Bahasa

Bahasa merupakan mediasi pikiran, perasaan dan perbuatan sekaligus alat untuk berinteraksi sesama manusia. Dalam teori bahasa untuk penerjemah bahasa Indonesia – Aksara Lontara Bugis ada beberapa hal yang digunakan antara lain :

1. Kamus

Menurut Jalius (2010) Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata yang berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mempunyai pedoman sebutan, asal-usul (*etimologi*) sesuatu perkataan dan juga contoh penggunaan bagi suatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus. Kamus adalah kitab yang berisi kata-kata dan arti atau keterangan yang disusun secara alfabetik.

Kata kamus diserap dari bahasa Arab *qamus*, dengan bentuk jamaknya *qawamis*. Kata Arab itu sendiri berasal dari kata Yunani '*okeanos*' yang berarti 'lautan'. Sejarah kata itu jelas memperlihatkan makna dasar yang terkandung dalam kata kamus, yaitu wadah pengetahuan yang tidak terhingga dalam dan luasnya.

Kamus memiliki berbagai macam jenis, sesuai dengan isi yang terkandung di dalamnya. Ada kamus bahasa baik dwibahasa

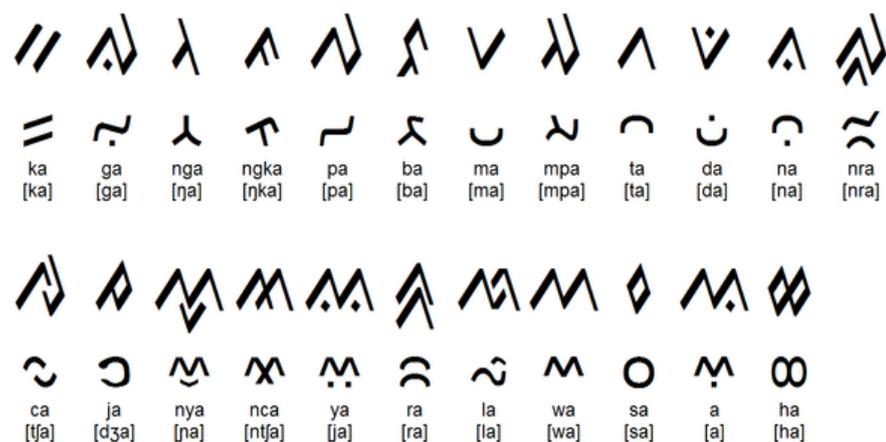
atau tribahasa, ada juga kamus istilah, misalnya kamus ilmiah populer yang menerangkan kata-kata ilmiah. Serta ada juga jenis kamus lain yang menjadi pedoman disiplin ilmu tertentu, misalnya kamus komputer dan kamus seperti yang akan penulis paparkan yakni kamus Lontara Bugis.

2. Lontara Bugis

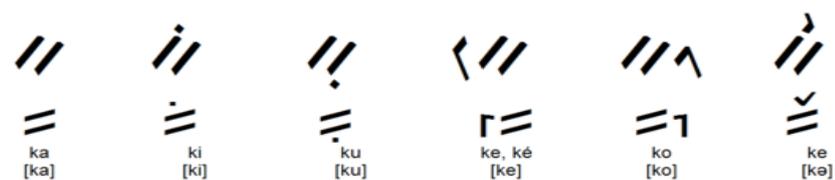
Menurut Wawan supriadi (2010) Lontara ialah aksara asli masyarakat bugis-makassar. Jadi bukan asimilasi apalagi pengaruh budaya lain termasuk india. Bentuk Aksara Lontara menurut budayawan Prof Mattulada (alm) berasal dari "*sulapa eppa wala suji*". *Wala suji* berasal dari kata *wala* = pemisah/pagar/penjaga dan *suji* = putri. *Wala Suji* adalah sejenis pagar bambu dalam acara ritual yang berbentuk belah ketupat. *Sulapa eppa* (empat sisi) adalah bentuk mistis kepercayaan Bugis-Makassar klasik yang menyimbolkan susunan semesta, api-air-angin-tanah. Dari segi aspek budaya, suku bugis menggunakan dialek tersendiri dikenal dengan "Bahasa *Ugi*" dan mempunyai tulisan huruf bugis yang dipanggil "Aksara Lontara Bugis". Aksara ini telah ada sejak abad ke-12 sejak melebarnya pengaruh Hindu di Indonesia. Aksara bugis berjumlah 23 huruf yang semuanya disusun berdasarkan aturan tersendiri. Perbedaan utama antara "Aksara Lontara Bugis" dengan Akasara Nusantara lainnya yaitu walaupun pada Aksara Lontara Bugis ada

beberapa huruf yang namanya sama dengan aksara nusantara lainnya, tetapi bukan hasil asimilasi dari budaya lain seperti India dan Arab dan yang kedua Aksara Lontara Bugis tidak mengenal huruf atau lambang untuk mematikan hurup misalnya “ka” menjadi “k”, sehingga cukup membingungkan bagaimana menuliskan huruf mati. Adapun Aksara Lontara Bugis yang digunakan dapat dilihat pada Gambar 2.1 sebagai berikut (Mattulada, 1991: 4-85)^{[5][6]}.

a) Huruf Lontara Bugis



b) Huruf Vokal



Gambar 2.1 Aksara Lontara Dengan Penyebutannya^[7]

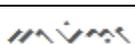
(Sumber: Wawan Supriadi :2010)

Gambar 2.1 menggambarkan penulisan huruf Aksara Lontara Bugis beserta penyebutannya sekaligus tanda baca huruf vokal pada penulisan Aksara Lontara Bugis.

3. Morfologi Bahasa Bugis

Morfologi adalah cabang ilmu bahasa yang mengidentifikasi satuan-satuan dasar bahasa sebagai satuan makna yang dapat berubah-ubah bergantung pada urutan kata, intonasi, bentuk dan kata tugas penentu kalimat. Kata adalah satuan makna dalam tutur kalimat^[8]. Daftar penamaan jenis kata dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penamaan Jenis Kata

Jenis Kata	Indonesia	Bugis	Aksara Lontara
Kata Sifat	malu	Masiri	
Kata keterangan	Disini	Komaie	
Kata kerja	Makan	Manre	

Penjelasan selengkapnya mengenai jenis-jenis kosa kata dapat dilihat di Lampiran 1.

Proses penerjemahan bahasa Indonesia – Aksara Lontara Bugis dilakukan dengan beberapa metode dimana metode tersebut sebagai proses pemecahan kalimat dan pencarian kata yang akan diterjemahkan.

C. Metode Penerjemah

Penerjemahan bahasa Indonesia – Aksara Lontara Bugis dilakukan dengan beberapa metode antara lain :

1. *Parsing Tree*

Parsing adalah suatu cara memecah-mecah suatu rangkaian masukan (misalnya dari berkas atau *keyboard*) yang akan menghasilkan suatu pohon uraian (*parse tree*). *Parse Tree* berfungsi untuk menggambarkan bagaimana memperoleh suatu

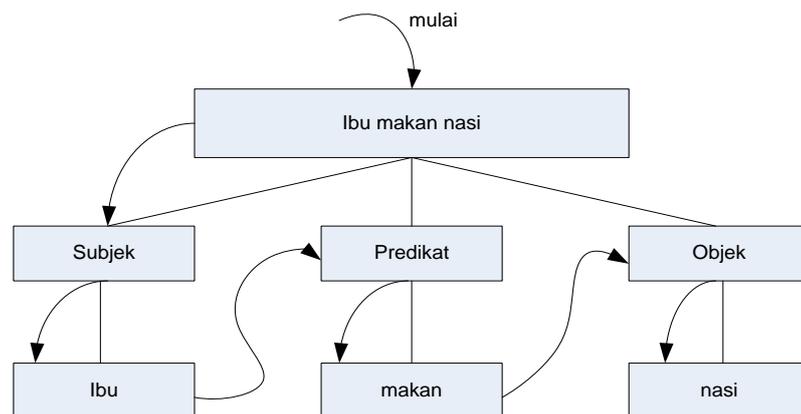
string dengan cara menurunkan simbol-simbol variabel menjadi simbol-simbol terminal, sampai tidak ada simbol yang belum tergantikan.

Algoritma untuk proses *parsing* banyak digunakan adalah sebagai berikut :

a) *Top-down parsing*

Top-down parsing bekerja dengan cara menguraikan sebuah kalimat mulai dari *constituent* yang terbesar yaitu sampai menjadi *constituent* yang terkecil. Hal ini dilakukan terus-menerus sampai semua komponen yang dihasilkan adalah *constituent* terkecil dalam kalimat, yaitu kata.

Sebagai contoh dapat dilakukan proses *top-down parsing* untuk kalimat “*ibu makan nasi*”



Gambar 2.2 : *Top Down Parser*

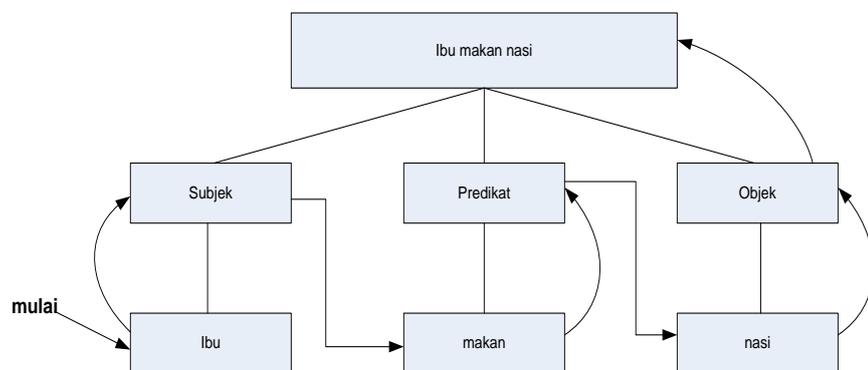
Dari Gambar 2.2 di atas terlihat bahwa *top-down parsing* menelusuri setiap node pada *parsing tree* secara *pre-order*.

Beberapa metode yang bekerja secara top down adalah *top-down parsing* biasa, *recursive-descent parsing*, *transition-network parsing* dan *chart parsing*^[9].

b) **Bottom-up parsing**

Bottom-up parsing bekerja dengan cara mengambil satu demi satu kata dari kalimat yang diberikan, untuk dirangkaikan menjadi *constituent* yang lebih besar. Hal ini dilakukan terus-menerus sampai *constituent* yang terbentuk ialah kalimat.

Sebagai contoh dilakukan proses *bottom-up parsing* dengan kata “ibu”, “makan”, “nasi”. Cara kerja *bottom-up parsing* ditunjukkan pada Gambar 2.3.



Gambar 2.3 : *Bottom-Up Parser*

Dari Gambar 2.3 di atas terlihat bahwa *bottom-up parsing* menelusuri setiap node terkecil (kata) hingga sampai ke node terbesar (kalimat)^[9].

2. *Binary Search*

Algoritma pencarian (*searching algorithm*) adalah algoritma yang menerima sebuah argumen kunci dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*)

Binary Search dilakukan dengan cara menebak apakah data yang dicari berada di tengah-tengah data, kemudian membandingkan data yang dicari dengan data yang ada di tengah. Apabila data yang di tengah sama dengan data yang dicari berarti data ditemukan. Namun, bila data yang di tengah lebih besar dari data yang dicari, maka dapat dipastikan bahwa data yang dicari kemungkinan berada di sebelah kiri dari data tengah dan data di sebelah kanan data tengah dapat diabaikan. *Upper bound* dari bagian data kiri yang baru adalah *indeks* dari data tengah itu sendiri. Sebaliknya, bila data yang di tengah lebih kecil dari data yang dicari, maka dapat dipastikan bahwa data yang dicari kemungkinan besar berada di sebelah kanan dari data tengah. *Lower bound* data di sebelah kanan dari data tengah adalah *indeks* dari data tengah itu sendiri ditambah satu.

Salah satu syarat agar pencarian biner dapat dilakukan adalah data sudah dalam keadaan terurut. Dengan kata lain,

apabila data belum dalam keadaan terurut, pencarian biner tidak dapat dilakukan. Dalam kehidupan sehari-hari, sebenarnya kita juga sering menggunakan pencarian biner. Misalnya saat ingin mencari suatu kata dalam kamus.

Prinsip dari pencarian biner dapat dijelaskan sebagai berikut:

(1) Menentukan nilai awal = 0

(2) Menentukan nilai akhir

$$\text{Nilai akhir} = N - 1 \quad \text{.....(2.1)}$$

Keterangan :

N = banyaknya data

(3) Menentukan nilai data tengah

$$\text{Nilai Tengah} = (\text{nilai awal} + \text{nilai akhir}) / 2 \quad \text{.....(2.2)}$$

(4) Data yang dicari dibandingkan dengan data tengah. Jika lebih kecil, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah - 1. Jika lebih besar, proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah + 1.

Pada intinya, algoritma ini menggunakan prinsip *divide and conquer*, di mana sebuah masalah atau tujuan diselesaikan dengan cara mempartisi masalah menjadi bagian yang lebih kecil. Algoritma ini membagi sebuah tabel menjadi dua dan memproses satu bagian dari tabel itu saja. Algoritma ini bekerja dengan cara memilih *record* dengan *indeks* tengah dari tabel dan

membandingkannya dengan *record* yang hendak dicari. Jika *record* tersebut lebih rendah atau lebih tinggi, maka tabel tersebut dibagi dua dan bagian tabel yang bersesuaian akan diproses kembali secara rekursif^[10].

Adapun ide dasar *binary search* yaitu memulai pencarian dengan membagi dua ruang pencarian. Misalnya kita memiliki array A, dan kita ingin menemukan lokasi dari spesifik target integer K dalam array. Ada 4 kemungkinan kondisi pada *binary search* yaitu:

- a) Jika data target K langsung ditemukan, maka proses pembagian ruangan berhenti. Kemudian *print out indeks* data elemen pada *array*.
- b) Jika data target $K < A[middle]$, maka pencarian dapat dibatasi hanya dengan melakukan pencarian pada sisi kiri *array* dari $A[middle]$. Seluruh elemen yang berada di sebelah kanan dapat diabaikan.
- c) Jika data target $K > A[middle]$, maka akan lebih cepat jika pencarian dibatasi hanya pada bagian sebelah kanan saja.
- d) Jika seluruh data telah dicari namun tidak ada, maka diberi nilai seperti -1^[11].

Misalnya ingin mencari data 17 pada sekumpulan data, perhatikan Gambar 2.4 berikut:

3	9	11	12	15	17	20	23	31	35
awal				tengah			akhir		

Gambar 2.4 Kumpulan Data

Perhatikan Gambar 2.4 di atas, mula-mula dicari data tengah, dengan rumus $(0 + 9) / 2 = 4$. Berarti data tengah adalah data ke-4, yaitu 15. Data yang dicari, yaitu 17, dibandingkan dengan data tengah ini. Karena $17 > 15$, berarti proses dilanjutkan tetapi kali ini posisi awal dianggap sama dengan posisi tengah + 1 atau 5.

3	9	11	12	15	17	20	23	31	35
					awal		tengah		akhir

Gambar 2.5 *Looping Pertama Binary Search*

Gambar 2.5 di atas, menunjukkan data tengah yang baru didapat dengan rumus $(5 + 9) / 2 = 7$. Berarti data tengah yang baru adalah data ke-7, yaitu 23. Data yang dicari yaitu 17 dibandingkan dengan data tengah. Karena $17 < 23$, berarti proses dilanjutkan tetapi kali ini posisi akhir dianggap sama dengan posisi tengah - 1 = 6.

3	9	11	12	15	17	20	23	31	35
				awal=tengah		akhir			

Gambar 2.6 *Looping Kedua Binary Search*

Gambar 2.6 di atas, menunjukkan data tengah yang baru didapat dengan rumus $(5 + 6) / 2 = 5$. Berarti data tengah yang

baru adalah data ke-5, yaitu 17. Data yang dicari dibandingkan dengan data tengah dan ternyata sama, jadi data ditemukan pada indeks ke-5.

Pencarian biner ini akan berakhir jika data ditemukan atau posisi awal lebih besar daripada posisi akhir. Jika posisi sudah lebih besar daripada posisi akhir berarti data tidak ditemukan.^[3]

Tabel 2.2 : Algoritma *Binary Search*

<p><u>Kamus</u></p> <p><u>Const</u> N : integer = 8 { misalkan jumlah elemen array maksimum = 8 }</p> <p><u>Type</u> A = <u>array</u> [1 N] <u>of</u> <u>integer</u></p> <p>Cari, BatasAtas, BatasBawah, Tengah : <u>Integer</u></p> <p>Ketemu : <u>boolean</u></p> <p><u>ALGORITMA</u></p> <p><u>Input</u> (cari) { meminta nilai data yang akan dicari }</p> <p>BatasAtas ← 1 { indeks array dimulai dari 1 }</p> <p>BatasBawah ← N</p> <p>Ketemu ← False</p> <p><u>While</u> (BatasAtas < BatasBawah) <u>and</u> (<u>not</u> ketemu) <u>do</u></p> <p style="padding-left: 2em;">Tengah ← (BatasAtas + BatasBawah) <u>div</u> 2</p> <p style="padding-left: 2em;"><u>If</u> A [Tengah] = cari <u>then</u></p> <p style="padding-left: 4em;">Ketemu ← <u>true</u></p> <p style="padding-left: 4em;"><u>Else</u></p> <p style="padding-left: 6em;">BatasAtas ← Tengah + 1</p> <p style="padding-left: 2em;"><u>Else</u></p> <p style="padding-left: 2em;">BatasBawah ← Tengah – 1 { cari di bagian kiri }</p> <p><u>Endif</u></p> <p style="padding-left: 4em;"><u>Endif</u></p> <p><u>EndWhile</u></p> <p><u>If</u> (ketemu) <u>then</u></p> <p style="padding-left: 2em;"><u>Output</u> ('Data berada di index nomor',Tengah')</p>

```

Else   Output ( 'Data tidak ditemukan' )
Endif

```

Sebagai contoh pencarian kata dengan menggunakan metode *binary search*. Dicari kata "Ibu" yang berada pada *index* ke-232 dengan seluruh data sejumlah 1000 kata.

Penyelesaian

$$(0 + 1000) / 2 = 500$$

Angka 500 lebih besar dari 232, maka pencarian dilakukan lagi.

$$(0 + 500) / 2 = 250$$

Angka 250 lebih besar dari 232, maka pencarian dilakukan lagi.

$$(0 + 250) / 2 = 125$$

Angka 125 lebih kecil dari 232, maka pencarian dilakukan lagi.

$$(125 + 250) / 2 = 187$$

Angka 187 lebih kecil dari 232, maka pencarian dilakukan lagi.

$$(187 + 250) / 2 = 218$$

Angka 218 lebih kecil dari 232, maka pencarian dilakukan lagi.

$$(218 + 250) / 2 = 234$$

Angka 234 lebih besar dari 232, maka pencarian dilakukan lagi.

$$(219 + 233) / 2 = 226$$

Angka 226 lebih kecil dari 232, maka pencarian dilakukan lagi.

$$(227 + 233) / 2 = 230$$

Angka 230 lebih kecil dari 232, maka pencarian dilakukan lagi.

$$(231 + 233) / 2 = 232$$

Sudah ditemukan pada *index* ke-232.

Metode yang dilakukan dalam proses penerjemahan bahasa Indonesia – Aksara Lontara Bugis, merupakan suatu cara untuk memisahkan kalimat menjadi kata, melakukan pencarian kata dan melakukan penggabungan kata menjadi kalimat.

D. Proses Penerjemah

Beberapa tahapan dalam proses penerjemahan bahasa Indonesia – Aksara Lontara Bugis antara lain :

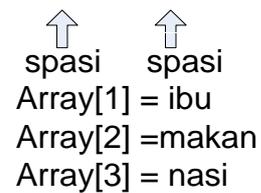
1. Proses Parsing Kalimat

Proses penguraian (*parsing*) merupakan penguraian kalimat menjadi kata-kata, dimana input yang berupa kalimat dicek berapa jumlah kata dan kemudian akan dipisahkan antarkata tersebut. Pemisahan kata menggunakan split yaitu berdasarkan tanda spasi (' ') di depannya. Proses tersebut diulang sampai kata terakhir dan kata disimpan pada *array* dengan pemberian *index* sementara.

Simulasi parsing kalimat

Syarat dari sebuah kalimat adalah terdiri dari 2 kata atau lebih. Sebagai contoh kalimat yang akan dilakukan proses *parsing* misalnya “Ibu makan nasi”.

Ibu makan nasi



 spasi spasi

 Array[1] = ibu

 Array[2] = makan

 Array[3] = nasi

Dari hasil proses *parsing* di atas diperoleh kata “Ibu” sebagai array pertama, kata ”makan” sebagai *array* kedua dan kata “nasi” sebagai *array* ketiga. Setelah proses parsing kalimat menjadi kata-kata selesai maka dilanjutkan pada tahap pencarian kata-kata tersebut.

2. Proses *Searching* (Pencarian)

Proses *seraching* (pencarian) kata dengan menggunakan metode *binary search*. *Binary Search* merupakan algoritma pencarian yang lebih efisien, dikarenakan algoritma ini tidak perlu menjelajahi setiap elemen dari tabel. Kerugian algoritma ini hanya bisa digunakan pada tabel yang elemennya sudah terurut baik menaik maupun menurun.

Simulasi *Binary Search*

Cara kerja pencarian *binary search* dapat disimulasikan sebagai berikut, dimisalkan di *database* memiliki data terurut seperti Tabel 2.3 di bawah ini.

Tabel 2.3 Simulasi *Binary Search*

index	Konversi	Indonesia	Bugis	Lontara
0	1	Bapak	Ambo	ᨁᨆᨗᨆᨗ
1	2	Bermain	Maccule	ᨁᨆᨗᨆᨗ
2	3	Ibu	Indo'	ᨁᨆᨗᨆᨗ
3	4	Kepala	Ulu	ᨁᨆᨗᨆᨗ
4	5	Makan	Manre'	ᨁᨆᨗᨆᨗ
5	6	Nasi	Nanre	ᨁᨆᨗᨆᨗ
6	7	Padi	Ase	ᨁᨆᨗᨆᨗ
7	8	Tidur	Tinro	ᨁᨆᨗᨆᨗ

Kata-kata yang dicari dengan metode *binary search* adalah hasil dari proses *parsing* kalimat menjadi kata-kata.

Misalkan hasil dari proses parsing kalimat "Ibu makan nasi" yaitu

Ibu makan nasi


 spasi spasi
 Array[1] = ibu
 Array[2] = makan
 Array[3] = nasi

Adapun cara kerja proses pencarian *binary search* berdasarkan hasil parsing kalimat di atas adalah sebagai berikut:

a) Pencarian kata "ibu"

Pencarian kata "ibu" yang berada pada *index* ke-2

Konversi Kata Bugis Aksara Lontara	1	2	3	4	5	6	7	8
	Bapak Ambo	Bermain Maccule	Ibu Indo'	Kepala Ulu	Makan Manre	Nasi Nanre	Padi Ase	Tidur Tinro
index	0	1	2	3	4	5	6	7
	↑ Awal							↑ Akhir

Gambar 2.7 Kumpulan Kata Terurut

Loop(1)

Perhatikan Gambar 2.7, dimana nilai awal = 0 dan akhir = 7 dari nilai index masing-masing setiap kosa kata. Adapun langkah-langkah untuk mencari kata “ibu” adalah sebagai berikut :

(1) Menentukan nilai awal dan nilai akhir

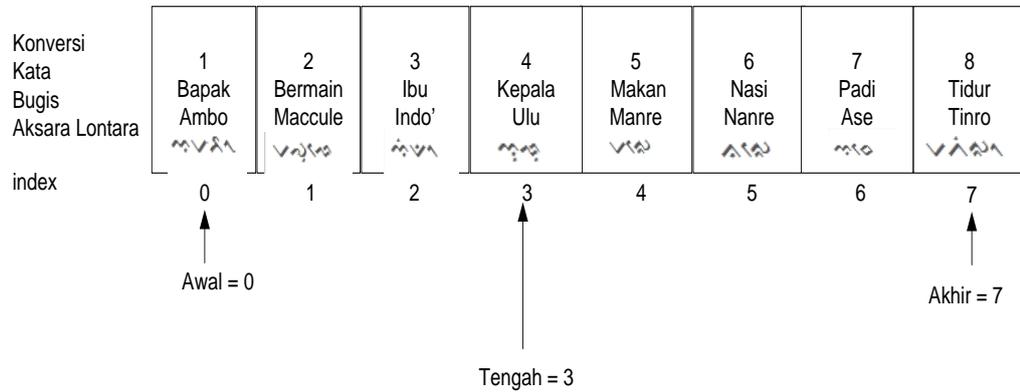
Nilai awal = 0, Nilai akhir = 7

(2) Mencari nilai tengah

Tengah = (awal + akhir)/2

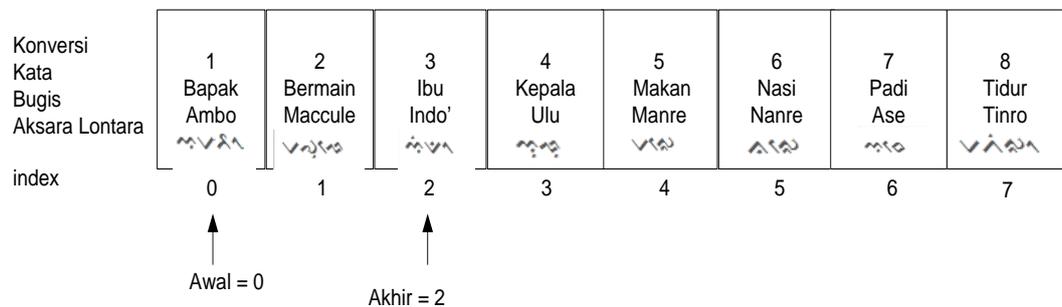
Tengah = $(0+7)/2 = 3,5$ dibulatkan kebawah menjadi 3

(3) **Nilai tengah = indeks** yang dicari maka pencarian berhasil ditemukan, apabila nilai **Tengah < indeks** yang dicari maka nilai **Tengah = Tengah + 1**. Jika nilai **Tengah > indeks** yang dicari maka nilai **Tengah = Tengah - 1**.

Gambar 2.8 Proses Pencarian *Looping* Pertama

Perhatikan Gambar 2.8 nilai awal = 0, nilai tengah = 3 dan nilai akhir = 7. Nilai tengah [3] > Indeks yang dicari [2], maka
 Tengah = $3 - 1 = 2$

Loop(2)

Gambar 2.9 Proses *Looping* Kedua

Pada Gambar 2.9 nilai awal = 0 dan nilai akhir = 2, nilai akhir berubah karena nilai Tengah pada loop(1) > *index* kata yang dicari, sehingga nilai akhir = $3 - 1 = 2$

(1) Nilai awal = 0, nilai akhir = 2

(2) Nilai Tengah = $(0 + 2)/2 = 1$

Konversi Kata Bugis Aksara Lontara	1	2	3	4	5	6	7	8
	Bapak Ambo	Bermain Maccule	Ibu Indo'	Kepala Ulu	Makan Manre	Nasi Nanre	Padi Ase	Tidur Tinro
index	0	1	2	3	4	5	6	7

Awal = 0 Tengah = 1 Akhir = 2

Gambar 2.10 Nilai Tengah *Looping* Kedua

- (3) Perhatikan Gambar 2.10 di atas, nilai tengah = 1, setelah itu membandingkan nilai tengah dengan *index* kata yang dicari. $Tengah[1] < Index[2]$ maka $tengah = 1 + 1 = 2$.

Loop(3)

Konversi Kata Bugis Aksara Lontara	1	2	3	4	5	6	7	8
	Bapak Ambo	Bermain Maccule	Ibu Indo'	Kepala Ulu	Makan Manre	Nasi Nanre	Padi Ase	Tidur Tinro
index	0	1	2	3	4	5	6	7

Awal = 2 Akhir = 2

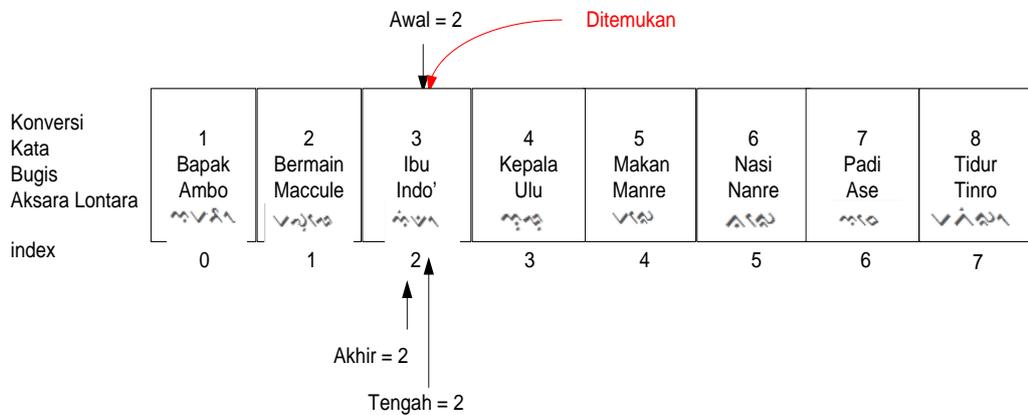
Gambar 2.11 Proses *Looping* Ketiga

Perhatikan Gambar 2.11 di atas, nilai awal berubah karena nilai Tengah < *index* data yang dicari. Sehingga nilai tengah berubah menjadi nilai awal = 2 dan nilai akhir = 2

- (1) Nilai Awal = 2, nilai Akhir = 2

- (2) Nilai Tengah = $(2 + 2)/2 = 2$

(3) Tengah [2] = Indeks [2], maka kata ditemukan



Gambar 2.12 Proses Penemuan Kata “Ibu”

Perhatikan Gambar 2.12 di atas, proses pencarian kata “ibu” ditemukan karena nilai $tengah[2] = index[2]$ data yang dicari.

b) Pencarian kata “makan”

Pencarian kata “makan” berada pada *index* ke-4, adapun langkah-langkan untuk pencarian kata “makan” sama dengan proses pencarian kata “ibu”. Perhatikan Gambar 2.6 di atas, dimana semua kata sudah dalam kondisi terurut.

Loop(1)

(1) Nilai awal = 0, nilai akhir = 7

(2) Nilai tengah = $(0 + 7)/2 = 3,5$ dibulatkan ke bawah = 3

Konversi Kata Bugis Aksara Lontara	1 Bapak Ambo 	2 Bermain Maccule 	3 Ibu Indo' 	4 Kepala Ulu 	5 Makan Manre 	6 Nasi Nanre 	7 Padi Ase 	8 Tidur Tinro 
index	0	1	2	3	4	5	6	7
	↑ Awal							↑ Akhir

Gambar 2.13 Proses Looping Pertama Kata “Makan”

Perhatikan Gambar 2.13 di atas, nilai tengah = 3
bandingkan dengan *index* kata yang dicari = 4. Jika nilai
 $Tengah[3] < index[4]$, maka nilai awal = tengah[3] + 1 = 4

Loop(2)

(1) Nilai awal = 4 dan nilai akhir = 7

(2) Nilai tengah = $(4+7)/2 = 5,5$ dibulatkan ke bawah = 5

Konversi Kata Bugis Aksara Lontara	1 Bapak Ambo 	2 Bermain Maccule 	3 Ibu Indo' 	4 Kepala Ulu 	5 Makan Manre 	6 Nasi Nanre 	7 Padi Ase 	8 Tidur Tinro 
index	0	1	2	3	4	5	6	7
					↑ Awal = 4	↑ Tengah = 5		↑ Akhir = 7

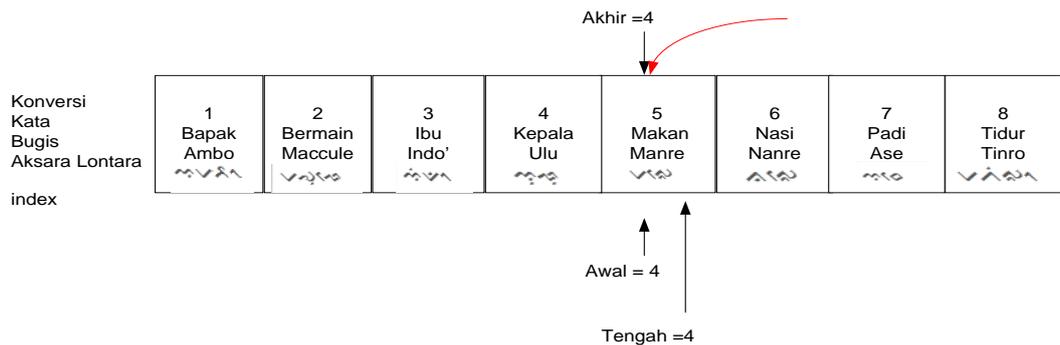
Gambar 2.14 Proses *Looping* Kedua Kata “Makan”

Perhatikan Gambar 2.14 di atas, nilai tengah = 5
bandingkan dengan *index* kata yang dicari = 4. Jika nilai
 $Tengah[5] > index[4]$, maka nilai akhir = tengah[5] - 1 = 4

Loop(3)

(1) Nilai awal = 4 dan nilai akhir = 4

(2) Nilai tengah = $(4+4)/2 = 4$



Gambar 2.15 Proses Penemuan Kata “Makan”

Perhatikan Gambar 2.15 di atas nilai tengah = 4, nilai awal = 4 dan nilai akhir = 4. Bandingkan nilai tengah dengan nilai *index* yang dicari, $tengah[4] = index[4]$, maka kata makan yang dicari telah ditemukan.

c) Pencarian kata “nasi”

Pencarian kata “nasi” berada pada *index* ke-5, adapun langkah-langkah untuk pencarian kata “nasi” sama dengan proses pencarian kata “ibu”. Perhatikan Gambar 2.6 di atas, dimana semua kata sudah dalam kondisi terurut.

Loop(1)

(1) Nilai awal = 0 dan nilai akhir = 7

(2) Nilai tengah = $(0 + 7)/2 = 3,5$ dibulatkan ke bawah = 3

Konversi Kata Bugis Aksara Lontara	1	2	3	4	5	6	7	8
	Bapak Ambo	Bermain Maccule	Ibu Indo'	Kepala Ulu	Makan Manre	Nasi Nanre	Padi Ase	Tidur Tinro
index	0	1	2	3	4	5	6	7
	↑ Awal							↑ Akhir

Gambar 2.16 Proses *Looping* Pertama Kata “Nasi”

Perhatikan Gambar 2.16 di atas, nilai tengah = 3
bandingkan dengan *index* kata yang dicari = 5. Jika nilai
 $Tengah[3] < index[5]$, maka nilai awal = $tengah[3] + 1 = 4$

Loop(2)

(1) Nilai awal = 4, nilai akhir = 7

(2) Nilai tengah = $(4+7)/2 = 5,5$ dibulatkan ke bawah = 5

Konversi Kata Bugis Aksara Lontara	1	2	3	4	5	6	7	8
	Bapak Ambo	Bermain Maccule	Ibu Indo'	Kepala Ulu	Makan Manre	Nasi Nanre	Padi Ase	Tidur Tinro
index	0	1	2	3	4	5	6	7
					Awal = 4	↑ Tengah = 5		↑ Akhir = 7

Gambar 2.17 Proses *Looping* Kedua Kata “Nasi”

Perhatikan Gambar 2.17 di atas, nilai tengah = 5
bandingkan dengan *index* kata yang dicari = 5. Jika nilai
 $Tengah[5] = index[5]$, maka kata yang dicari berhasil
ditemukan.

Hasil proses pencarian kata berdasarkan dari proses *parsing* kalimat menjadi kata dapat dilihat pada Tabel 2.4 di bawah :

Tabel 2.4 Hasil Pencarian Kata

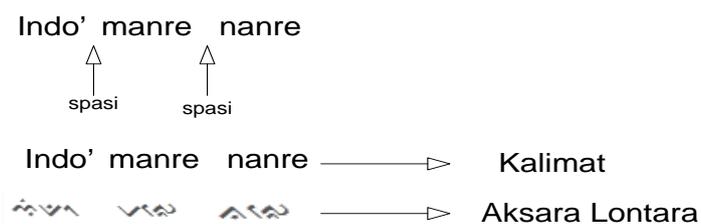
Indonesia	Bugis	Aksara Lontara
Ibu	Indo'	
Makan	Manre	
Nasi	Nanre	

3. Proses Penggabungan Kata

Proses penggabungan kata adalah proses menggabungkan kata menjadi kalimat. Hasil dari terjemahan tiap kata akan dimasukkan ke *array* sesuai dengan *index* masing-masing kata dapat dilihat pada Tabel 2.5 di bawah.

Tabel 2.5 Proses *Array* Kata

	Kata[1]	Kata[2]	Kata[3]
Indonesia	Ibu	Makan	Nasi
Bugis	Indo'	Manre	Nanre
Aksara Lontara			



Gambar 2.18 Proses Penggabungan Kata Menjadi Kalimat

Setelah proses pencarian dan pengisian array selesai sesuai pada Tabel 2.5 di atas, maka dapat dilihat pada Gambar 2.18, kata-kata tersebut akan digabungkan kembali menjadi sebuah kalimat dengan pemisahan tanda spasi (' ').

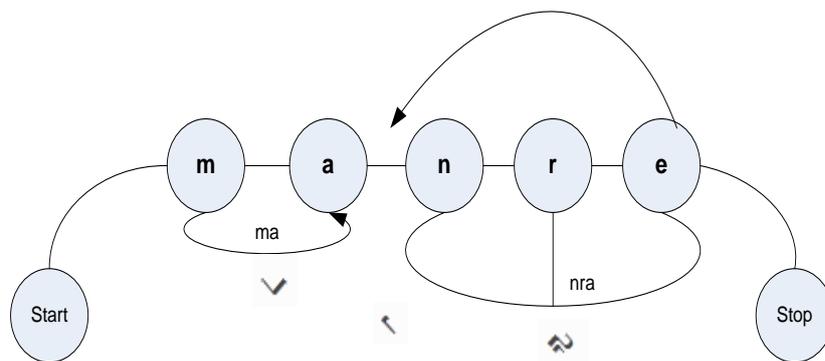
Tabel 2.7 Proses Kata *Manre*

Indonesia	Bugis	Lontara
Makan	Manre	↘ ↗

Nilai probabilitas transisi I :

$$S1 \text{ "Manre"} = (1 / \text{Jumlah Awalan Huruf}) = 1/1 = 1$$

Pada probabilitas transisi II atau *event* kedua diproses sama dengan karakter sebelumnya dan proses *compare* selesai, sehingga dapat dilihat pada Gambar 2.19 di bawah ini :



Gambar 2.19 Rangkaian State Kata "manre"

Pada Gambar 2.19 di atas menjelaskan rangkaian state kata "manre" dimana karakter "ma" diubah ke Lontara Bugis menjadi "↘", dan untuk karakter "nre" diubah menjadi huruf "nra" atau "↘ ↗". Apabila huruf vokal "e" di dalam penulisan Lontara Bugis akan berada pada awalan huruf Lontara sehingga untuk penulisan huruf "nre" akan menjadi "enra" atau "↘ ↗" dalam penyebutan Aksara Lontara Bugis. Untuk penulisan kata "manre" akan diubah

ke dalam penulisan Aksara Lontara Bugis yang menjadi”  ”.

Untuk huruf vokal ”e” atau”  ” dalam penulisan Aksara Lontara Bugis akan berada di depan huruf Lontara Bugis untuk semua jenis kosakata penulisan Aksara Lontara Bugis.

Merancang proses penerjemahan kalimat ke dalam piranti perangkat lunak sangat diperlukan karena sebagai alat untuk visualisasi dan dokumentasi aplikasi yang akan dirancang.

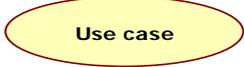
E. Model Perancangan Sistem

Model perancangan sistem menggunakan *Unified Modeling Language* (UML). UML adalah sebuah bahasa yang telah menjadi standar visualisasi, merancang dan mendokumentasi sistem piranti lunak. Pemodelan menggunakan *Unified Modeling Language* (UML) merupakan pemodelan berorientasi objek dan berbasis visual. *Unified Modeling Language* menawarkan diagram yang dikelompokan untuk memodelkan suatu sistem antara lain^[13]:

1. Use Case Diagram

Use Case Diagram menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna (*user*) . Terdapat beberapa simbol dalam menggambarkan *use case diagram* yaitu *use case*, aktor, dan relasi. Simbol-simbol yang terdapat pada diagram *use case* dapat dilihat pada Tabel 2.8 berikut :

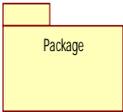
Tabel 2.8 Simbol-Simbol *Use Case Diagram*

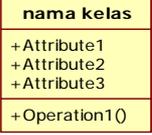
Simbol	Deskripsi
 Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem yang akan dibuat itu sendiri
 Use case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
asosiasi/ asociation 	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor

2. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas yang akan dibuat untuk membangun sistem. Diagram *class* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain. Simbol-simbol yang ada pada *class diagram* dapat dilihat pada Tabel 2.9 di bawah ini :

Tabel 2.9 Simbol-Simbol *Class Diagram*

Simbol	Deskripsi
Package  Package	Package merupakan suatu bungkus dari satu atau lebih kelas

<p>Operasi</p> 	<p>Kelas pada struktur sistem</p>
<p>Asosiasi</p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan multiplicity.</p>
<p>Asosiasi berarah/<i>directed asosiasi</i></p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity</p>
<p>Generalisasi</p> 	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)</p>
<p>Kebergantungan</p> 	<p>Relasi antar kelas dengan makna kebergantungan antar kelas</p>
<p>Agregasi</p> 	<p>Relasi antar kelas dengan makna semua bagian (whole-part)</p>

3. Sequence Diagram

Diagram *sequence* menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Banyaknya *diagram sequence* yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interksi jalanya

pesan. Adapun simbol-simbol sequence diagram pada Tabel 2.10 sebagai berikut :

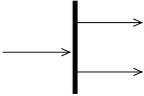
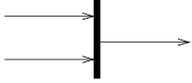
Tabel 2.10 Simbol-Simbol *Sequence Diagram*

Simbol	Deskripsi
Actor  Aktor	Menggambarkan orang yang sedang berinteraksi dengan sistem
Entity class 	Menggambarkan hubungan kegiatan yang akan dilakukan
Boundary class 	Menggambarkan sebuah penggambaran dari form
Control class 	Menggambarkan hubungan antara boundary dengan tabel
a focus of control dan a Life line 	Menggambarkan tempat mulai dan berakhirnya sebuah pesan
Pesan 	Menggambarkan pengiriman pesan

4. Activity Diagram

Activity diagram menggambarkan *workflow* (alur kerja) atau aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Adapun simbol-simbol *activity diagram* pada Tabel 2.11 sebagai berikut :

Tabel 2.11 Simbol-Simbol *Activity Diagram*

Simbol	Deskripsi
Start Point 	Memulai alur kerja
End Point 	Mengakhiri Alur kerja
Aktivitas 	Kegiatan dalam alur kerja
Fork 	Percabangn alur kerja
Join 	Penggabungan alur kerja menjadi satu
Decision 	Suatu kondisi dalam proses alur kerja
Swimlane 	Sebuah cara untuk mengelompokan <i>activity</i> berdasarkan aktor

5. Statechart Diagram

Diagram statechart adalah diagram untuk menggambarkan *behavior*, yaitu perubahan *state* di suatu *class* berdasarkan *event* dan *message* yang dikirimkan dan diterima oleh *class* tersebut. Setiap diagram state hanya boleh memiliki *start state* (*initial state*) dan boleh memiliki satu atau lebih dari satu *stop states* (*final state*). Adapun simbol-simbol *statechart diagram* terdapat pada Tabel 2.12 sebagai berikut :

Tabel 2.12 Simbol-Simbol *statechart Diagram*

GAMBAR	KETERANGAN
<i>Activity</i> 	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
<i>Action</i> 	State dari sistem yang mencerminkan eksekusi dari suatu aksi
<i>Initial Node</i> 	Bagaimana objek dibentuk atau diawali.
<i>Activity Final Node</i> 	Bagaimana objek dibentuk dan diberhentikan
<i>Fork Node</i> 	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran