

**APLIKASI KAMUS BAHASA MONGONDOW-  
INDONESIA MENGGUNAKAN ALGORITMA  
KNUTH MORRIS PRATT**

**Oleh**

**SITTI NUR AISYAH LAMAMA**

**T3115188**

**SKRIPSI**

**Untuk Memenuhi Salah Satu Syarat Ujian**

**Guna Memperoleh Gelar Sarjana**



**PROGRAM SARJANA  
TEKNIK INFORMATIKA  
UNIVERSITAS ICHSAN GORONTALO  
GORONTALO  
2020**

**PENGESAHAN SKRIPSI**

**APLIKASI KAMUS BAHASA MONGONDOW-  
INDONESIA MENGGUNAKAN ALGORITMA  
KNUTH MORRIS PRATT**


Oleh

**SITTI NUR AISYAH LAMAMA  
T3115188**


**SKRIPSI**

Untuk memenuhi salah satu syarat ujian  
guna memperoleh gelar Sarjana  
dan telah disetujui oleh Tim Pembimbing  
Gorontalo, 3 November 2020

Pembimbing I

  
**Budy Santoso, S.Kom, M.Eng**  
NIDN. 0908048303

Pembimbing II

  
**Husdi, M.Kom**  
NIDN. 0907108701

## PERSETUJUAN SKRIPSI

# APLIKASI KAMUS BAHASA MONGONDOW- INDONESIA MENGGUNAKAN ALGORITMA KNUTH MORRIS PRATT

Oleh  
SITTI NUR AISYAH LAMAMA  
T3115188

Diperiksa Oleh panitia Ujian Strata Satu (S1)  
Univesitas Ichsan Gorontalo

1. Pembimbing I  
Budy Santoso, S.Kom, M.Eng
2. Pembimbing II  
Husdi, M.Kom
3. Penguji I  
Irvan Abraham Salihi, M.Kom
4. Penguji II  
Haditsah Annur, M.Kom
5. Penguji III  
Warid Yunus, M.Kom




Mengetahui :

Dekan Fakultas Ilmu Komputer



**Zohrahayaty, M.Kom**  
NIDN: 0912117702

Ketua Program Studi



**Irvan A Salihi, M.Kom**  
NIDN: 0928028101

## **PERNYATAAN SKRIPSI**

Dengan ini saya Menyatakan bahwa :

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali, arahan Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/situasi dalam naskah dan dicantumkan pula daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyipangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 3 November 2020

Yang Membuat Pernyataan

Sitti Nur Aisyah Lamama

## **ABSTRACT**

*Indonesia has many cultures and varies, one of which is the local language. Language is used to communicate with others. Indonesian people are bilingual people, that is people who know two languages, Indonesian as the national language and the regional language used in their ethnic groups. There are many regional languages in Indonesia, one of which is the Mongondow language. Mongondow is a regional language that we must preserve, remembering that the status of this language itself is almost extinct. Dictionary media can be one way to introduce the diversity of regional languages in Indonesia. The Android platform can be an alternative for creating an attractive electronic dictionary. The mongondow language dictionary application based on Android works by doing string matching. Search for words to be translated using one of the string search algorithms, Knuth Morris Pratt (KMP). This algorithm works by utilizing the maximum possible shift in string matching so that the results of the translation will be displayed more quickly.*

*Keywords: Dictionary, Knuth Morris Pratt (KMP), Android*

## ABSTRAK

Indonesia memiliki banyak budaya dan bervariasi, salah satunya adalah bahasa daerah. Bahasa digunakan untuk berkomunikasi dengan orang lain. Rakyat Indonesia adalah masyarakat bilingual, yaitu masyarakat yang mengetahui dua bahasa, bahasa Indonesia sebagai bahasa nasional dan bahasa daerah yang digunakan di kelompok etnis mereka. Ada banyak bahasa daerah yang ada di Indonesia salah satunya adalah bahasa Mongondow. Bahasa Mongondow adalah Bahasa daerah yang harus kita lestarikan mengingat status Bahasa ini sendiri sudah hampir punah. Media kamus dapat menjadi salah satu cara untuk mengenalkan beragamnya Bahasa daerah yang ada di Indonesia. Platform Android bisa menjadi sebuah alternatif untuk pembuatan kamus elektronik dengan tampilan yang menarik. Aplikasi kamus Bahasa mongondow berbasis Android bekerja dengan melakukan *string matching*. Pencarian kata yang akan diterjemahkan menggunakan salah satu algoritma pencarian string yaitu *Knuth Morris Pratt (KMP)*. Algoritma ini bekerja dengan memanfaatkan pergeseran yang semaksimal mungkin dalam pencocokan string sehingga hasil terjemahan akan ditampilkan dengan lebih cepat.

**Kata Kunci** : Kamus, *Knuth Morris Pratt (KMP)*, Android

## KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan skripsi ini dengan judul **“Aplikasi Kamus Bahasa Mongondow – Indonesia Menggunakan Algoritma Knuth Morris Pratt”**, Sebagai salah satu syarat Ujian Akhir guna memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa skripsi ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materi. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Bapak Muh. Ichsan Gaffar, S.E., M.Ak selaku ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr.Abdul Gaffar La Tjokke, M.Si selaku Rektor Universitas Ichsan Gorontalo;
3. Ibu Zohrahayaty, S.Kom, M.Kom selaku Dekan Fakultas Ilmu Komputer;
4. Sudirman S. Panna M.Kom, selaku Pembantu Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Gorontalo;
5. Ibu Irma Surya Kumala Idris, S.Kom, M.Kom, selaku wakil dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Sudirman Melangi S.Kom, M.Kom, selaku Wakil Dekan III Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Irvan Abraham Salihi, S.Kom, M.Kom selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
8. Bapak Budy Santoso, S.Kom, M.Eng selaku pembimbing I;
9. Bapak Husdi, M.Kom selaku pembimbing II;

10. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
11. Kepada Kedua Orang Tua ,Kakak, dan Adik tercinta yang selalu memberikan dorongan moral maupun materil dari awal hingga akhir perkuliahan;
12. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada saya;
13. Kepada teman-teman saya Risha,Zarah,Mifa,Haza,Quraisya,Rara yang telah banyak memberi dukungan dan semangat kepada penulis.
14. Kepada semua pihak yang ikut membantu dalam menyelesaikan skripsi ini yang tak sempat penulis sebutkan satu – persatu;

Semoga Allah SWT melimpahkan balasan atas jasa – jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, 3 November 2020

Penulis



# DAFTAR ISI

	<b>Halaman</b>
PENGESAHAN SKRIPSI .....	i
PERSETUJUAN SKRIPSI .....	ii
PERNYATAAN SKRIPSI.....	iii
<i>ABSTRACT</i> .....	iv
ABSTRAK .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Identifikasi Masalah .....	3
1.3. Rumusan Masalah .....	3
1.4. Tujuan Masalah .....	4
1.5. Manfaat Penelitian.....	4
BAB II LANDASAN TEORI.....	5
2.1. Tinjauan Studi .....	5
2.2. Tinjauan pustaka.....	7
2.2.1. Bahasa Mongondow.....	7
1.1.1 2.2.2 Algoritma Knuth Morris Pratt.....	9
2.2.3 Android .....	10
2.2.4 Generasi Android .....	11
Tabel 2.2. Nama-Nama Versi <i>Android</i> .....	12
2.2.5 Pengembangan Sistem .....	12

2.2.6	Analisis Sistem.....	12
2.2.7	Desain Sistem.....	17
2.2.8	Konstruksi Sistem .....	18
2.2.9	Pengujian Sistem.....	19
2.3	Kerangka Pikir.....	23
<b>BAB III OBJEK DAN METODE PENELITIAN .....</b>		<b>25</b>
3.1.	Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian.....	25
3.2.	Pengumpulan Data .....	25
3.3.	Pengembangan Sistem.....	26
3.3.1.	Sistem yang di usulkan .....	26
3.3.2.	AnalisisSistem.....	26
3.3.3.	DesainSistem.....	27
3.3.4.	KonstruksiSistem .....	28
3.3.5.	PengujianSisitem.....	28
<b>BAB IV HASIL PENELITIAN .....</b>		<b>29</b>
4.1	Hasil Pengumpulan Data .....	29
4.1.1	Penerapan Metode.....	30
4.2	Hasil Pengembangan Sistem .....	33
4.3	Activity Diagram Login .....	34
4.4	Activity Diagram Search .....	35
4.5	Activity Diagram Tambah Data .....	36
4.6	Activity Diagram Data Proses Data Kamus .....	37
4.7	Sequence Diagram Login Admin .....	38
4.8	Sequence Diagram Proses Cari Kata.....	39
4.9	Sequence Diagram Tambah Data .....	39
4.10	Sequence Diagram Proses Data Kamus .....	40
4.11	Arsitektur Sistem.....	40
4.12	Interface Design .....	41

4.12.1	Mekanisme User.....	41
4.12.2	Mekanisme navigasi Home .....	41
4.12.3	Mekanisme <i>Login</i> .....	42
4.12.4	Mekanisme <i>Input</i> data Kamus.....	43
4.12.5	Mekanisme Detail, Ubah dan Hapus Data Kamus.....	43
4.12.6	Mekanisme Output .....	44
4.13	Data Desain .....	44
4.13.1	Struktur Data .....	44
4.14	Hasil Pengujian Sistem.....	45
4.14.1	Pengujian <i>White Box</i> .....	46
4.14.2	<i>Flowchart</i> .....	48
4.14.3	Pengujian <i>White Box</i> .....	49
4.14.4	Perhitungan CC pada pengujian <i>White Box</i> .....	50
4.14.5	Perhitungan CC pada pengujian <i>White Box</i> .....	50
4.14.6	Pengujian <i>Black Box</i> .....	51
BAB V HASIL DAN PEMBAHASAN.....		51
5.1	Pembahasan Sistem .....	51
5.1.1	Tampilan Halaman Home .....	51
5.1.2	Tampilan Halaman Data Profil .....	52
5.1.3	Tampilan Halaman Knuth Morris Pratt.....	53
5.1.4	Tampilan Halaman Kamus Indonesia - Mongondow .....	54
5.1.5	Tampilan Halaman Kamus Mongondow - Indonesia .....	55
5.1.6	Tampilan Halaman Data Login.....	56
5.1.7	Tampilan Halaman Utama Admin .....	57
5.1.8	Tampilan Halaman Tambah Data Kamus .....	58
5.1.9	Tampilan Halaman Detail Kamus.....	59
BAB VI PENUTUP .....		60
6.1	Kesimpulan.....	60
6.2	Saran.....	60
DAFTAR PUSTAKA .....		61

## DAFTAR GAMBAR

Gambar 2.1. Cara Kerja Algoritma KMP .....	9
Gambar 2.2. Grafik alir[15] .....	21
Gambar 3.1. Sistem yang diusulkan .....	26
Gambar 4.1 <i>Use Case Diagram</i> Kamus Bahasa Mongondow .....	32
Gambar 4.2 <i>Activity Diagram</i> Proses Login .....	33
Gambar 4.3 <i>Activity Diagram</i> Pada Proses Cari Kata.....	34
Gambar 4.4 <i>Activity Diagram</i> Tambah Data .....	35
Gambar 4.5 <i>Activity Diagram</i> Pada Proses Data Kamus .....	36
Gambar 4.6 <i>Sequence Diagram</i> Login Admin.....	37
Gambar 4.7 <i>Sequence Diagram</i> Proses Cari Kata .....	38
Gambar 4.8 <i>Sequence Diagram</i> Tambah Data.....	38
Gambar 4.9 <i>Sequence Diagram</i> Proses Data Kamus .....	39
Gambar 4.10 Mekanisme Navigasi Home User.....	40
Gambar 4.11 Mekanisme Navigasi Home Admin .....	41
Gambar 4.12 Mekanisme <i>Login</i> .....	41
Gambar 4.13 Mekanisme <i>Input</i> data kamus.....	42
Gambar 4.14 Mekanisme Detail, Ubah dan Hapus data Kamus.....	42
Gambar 4.15 Mekanisme Output Kamus.....	43
Gambar 4.16 <i>Flowchart</i> Algoritma KMP .....	46
Gambar 4.17 <i>Flowgraph</i> Algoritma KMP.....	47
Gambar 5.6Tampilan <i>Home</i> User .....	51
Gambar 5.7 Halaman data Profil.....	52
Gambar 5.3 Halaman Knuth Morris Pratt .....	53
Gambar 5.4 Halaman Kamus Indonesia - Mongondow.....	54
Gambar 5.5 Halaman Kamus Mongondow - Indonesia.....	55
Gambar 5.6 Halaman Data Login .....	56

Gambar 5.7 Halaman Utama Admin.....	57
Gambar 5.8 Halaman Tambah Data Kamus .....	58
Gambar 5.9 Halaman Detail Kamus .....	59

## DAFTAR TABEL

Tabel 2.1. Penelitian Terkait .....	6
Tabel 2.2. Nama-Nama Versi Android .....	11
Tabel 2.3. Use Case Diagram[11] .....	13
Tabel 2.4. Activity Diagram[11] .....	14
Tabel 2.5. Sequence Diagram[11] .....	15
Tabel 2.6. Multiplicity Class Diagram[11] .....	17
Tabel 4.1 Hasil Pengumpulan Data.....	29
Tabel 4.2 Mekanisme User .....	40
Tabel 4.3 Tabel kamus .....	43
Tabel 4.4 Kecepatan Pencarian denganAlgoritma KMP .....	44
Tabel 4.5 Basis <i>Path</i> .....	49
Tabel 4.6 Tabel Pengujian <i>Black Box</i> .....	50

## DAFTAR LAMPIRAN

Lampiran 1: Kode Program.....	62
Lampiran 2: Daftar Riwayat Hidup Peneliti .....	64
Lampiran 3: Rekomendasi Penelitian .....	65
Lampiran 4: Bukti Penerimaan Softcopy Skripsi .....	66
Lampiran 5: Rekomendasi Bebas Plagiasi .....	67
Lampiran 6: Hasil Turnitin .....	68

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Indonesia memiliki banyak budaya dan bervariasi, salah satunya adalah bahasa daerah. Bahasa digunakan untuk berkomunikasi dengan orang lain. Rakyat Indonesia adalah masyarakat bilingual, yaitu masyarakat yang mengetahui dua bahasa, bahasa Indonesia sebagai bahasa nasional dan bahasa daerah yang digunakan di kelompok etnis mereka. Bahasa dapat menjadi identitas dan ciri khas suatu negara maupun daerah. Ada banyak bahasa daerah yang ada di Indonesia salah satunya adalah bahasa Mongondow. Bahasa Mongondow adalah bahasa rumpun Filipina yang digunakan oleh Suku mongondow di Sulawesi Utara, yang pada mulanya Bahasa Mongondow merupakan bahasa yang digunakan oleh penduduk Kerajaan Bolaang Mongondow yang kemudian menjadi Kabupaten Bolaang Mongondow saat ini. Suku Mongondow tersebar di Kabupaten Bolaang Mongondow, Kota Kotamobagu, Kabupaten Bolaang Mongondow Utara, Kabupaten Bolaang Mongondow Timur, dan Kabupaten Bolaang Mongondow Selatan. Adapula ciri khas dari bahasa Mongondow ini sendiri, yaitu cara menuturkan kata contohnya huruf 'L'. Huruf L dalam Bahasa Mongondow memiliki tiga jenis yaitu 'L', 'l', 'L'. Jenis huruf L yang pertama bertitik di bawahnya, diucapkan seperti kata *are* dalam Bahasa Inggris. Huruf L jenis kedua diberi garis pendek di bawahnya, diucapkan dengan cara melekatkan lidah pada langit-langit keras, lalu diletupkan sepiintas pada saat mengucapkan kata. Huruf L jenis ketiga terjadi jika L itu terletak di antara vokal: i—I; e—e; atau sesudah vokal i dan e[1].

Bahasa Mongondow adalah Bahasa daerah yang harus kita lestarikan mengingat status Bahasa ini sendiri sudah hampir punah. Hal ini terjadi karena penurunan penggunaan dalam berkomunikasi setiap hari. Di dunia pendidikan, pembelajaran tentang Bahasa daerah hanya diterapkan sampai di tingkat menengah pertama (SMP) saja. Pernah ada satu upaya dari pemerintah untuk tetap melestarikan penggunaan Bahasa mongondow, yaitu mereka membuat suatu



peraturan untuk berkomunikasi menggunakan bahasa mongondow di hari yang telah di tentukan, dan peraturan tersebut di berlakukan di instansi-instansi dan di setiap sekolah yang ada di Kota Kotamobagu. Namun peraturan tersebut tidak berlangsung lama karena banyak juga pegawai dan siswa-siswi yang tidak mahir berbahasa mongondow. Hal ini juga terlihat dari para pelajar lebih menyukai bahasa gaul daripada bahasa daerah mereka, inikarenakurangnya rasa ingin tahu, dan rasa bangga terhadap budaya yang ada di daerah mereka sendiri. Perlu kita ketahui Bersama bahwa kita harus mengetahui asal usul nenek moyang dan cara hidup mereka agar dapat mengenali diri kita dan seperti apa asal usul daerah kita.

Melestarikan budaya bahasa daerah dapat dilakukan dengan berbagai macam cara di tengah kemajuan teknologi yang sangat pesat pada saat ini. Media kamus dapat menjadi salah satu cara untuk mengenalkan beragamnya Bahasa daerah yang ada di Indonesia. Aplikasi kamus sangat populer terutama dikalangan generasi muda, mengingat juga saat ini semua orang tidak akan pernah lepas dari teknologi. Kemudahan yang didapatkan dalam melakukan pencarian makna kata di aplikasi kamus dapat meningkatkan peminat pengguna gadget saat ini. Selain fungsi utamanya untuk berkomunikasi, handphone juga bisa digunakan sebagai alat untuk belajar dan memperoleh pengetahuan. Dibandingkan dengan kamus yang berbentuk buku, kamus elektronik juga memudahkan pengguna untuk belajar dimanapun mereka inginkan. Platform Android bisa menjadi sebuah alternatif untuk pembuatan kamus elektronik dengan tampilan yang menarik. Hingga saat ini Android terus berkembang, baik secara system maupun aplikasinya. Pada akhirnya pada penelitian ini, saya ingin mengangkat sistem operasi Android sebagai smartphone yang akan dijadikan media untuk perancangan Aplikasi kamus Bahasa Mongondow.

Dalam penelitian ini aplikasi kamus Bahasa mongondow berbasis Android bekerja dengan melakukan *string matching* atau pencocokan string berdasarkan kata-kata yang terdapat dalam database yang merupakan aplikasi kamus tersebut[2]. Pencarian kata yang akan diterjemahkan menggunakan salah satu algoritma pencarian string yaitu *Knuth Morris Pratt(KMP)*. Algoritma ini merupakan pencocokkan atau pencarian string yang merupakan pengembangan dari algoritma *Brute Force*. Algoritma ini bekerja dengan memanfaatkan

pergeseran yang semaksimal mungkin dalam pencocokan string sehingga hasil terjemahan akan ditampilkan dengan lebih cepat. Dengan demikian pengguna dapat dengan mudah mempelajari Bahasa Mongondow secara mandiri[3]. Selanjutnya sebelum mengimplementasikan algoritma ke dalam aplikasi maka perlu dilakukan perancangan atau desain aplikasi terlebih dahulu. Pada tahap desain aplikasi kamus ini, tool yang digunakan adalah *Unified Modelling Language (UML)* yang terdiri dari Diagram *Use Case*, Diagram *Activity* dan *Sequence* Diagram. Setelah tahap desain selesai dilanjutkan pada tahap pembuatan program aplikasi[3].

Berdasarkan uraian diatas, maka dianggap perlu untuk melakukan penelitian mengenai proses yang berjalan diatas, dengan judul ***“Aplikasi Kamus Bahasa Mongondow-Indonesia Menggunakan Algoritma Knuth Morris Pratt”***.

### **1.2. Identifikasi Masalah**

Berdasarkan latar belakang masalah yang telah dijelaskan di atas maka identifikasi masalah dalam penelitian ini adalah:

1. Saat ini status Bahasa Mongondow ini sendiri sudah hampir punah, dan kurangnya penggunaan Bahasa daerah dalam berkomunikasi sehari-hari
2. Menerapkan Algoritma Knuth Morris Pratt dalam Aplikasi Kamus Bahasa Mongondow Indonesia.

### **1.3. Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah dijelaskan di atas, dapat dirumuskan masalah pokok yang berkaitan yaitu :

1. Bagaimana merekayasa Aplikasi Kamus Bahasa Mongondow Indonesia, dapat dimanfaatkan oleh masyarakat Bolaang Mongondow.
2. Bagaimana hasil penerapan algoritma Knuth Morris Pratt dalam Aplikasi Kamus Bahasa Mongondow-Indonesia.

#### **1.4. Tujuan Masalah**

1. Dapat merekayasa Aplikasi Kamus Bahasa Mongondow-Indonesia, dan bisa dimanfaatkan oleh masyarakat Bolaang Mongondow.
2. Mengetahui hasil penerapan Algoritma *Knuth Morris Pratt* pada aplikasi Kamus Bahasa Mongondow – Indonesia.

#### **1.5. Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut :

1. Membantu melestarikan Bahasa daerah Mongondow.
2. Memberikan kemudahan bagi para pengguna aplikasi kamus dalam menerjemahkan setiap kata ke Bahasa Mongondow.
3. Dapat mengetahui hasil penerapan Algoritma Knuth Morris Pratt dalam Aplikasi Kamus Bahasa Mongondow – Indonesia.
4. Bagi peneliti sendiri dapat memperdalam pengetahuan Bahasa Mongondow.

## BAB II

### LANDASAN TEORI

#### 2.1. Tinjauan Studi

Berikut ini adalah penelitian yang terkait dengan algoritma *Knuth Morris Pratt*, yaitu :

**Tabel 2.1.** Penelitian Terkait.

No	Peneliti	Judul	Tahun	Metode/Algoritma	Hasil
1	Haris Triono Sigit	Aplikasi Android Kamus Bahasa Jawa Serang – Indonesia Mengguna kan Algoritma <i>Knutt Morris Pratt.</i>	2015	<i>Knutt Morris Pratt</i>	Aplikasi Kamus Bahasa Jawa Serang – Indonesia ini bisa menjadi pilihan pengguna yang ingin mempelajari Bahasa Jawa Serang dengan mudah dan mandiri. Aplikasi ini telah di uji coba dan mendapatkan hasil responden dengan bentuk grafik, yaitu 93% tingkat kemudahannya, 85% tampilannya yang menarik, dan 90% manfaat aplikasi tersebut. Pencarian katanya menggunakan pencocokan String

					dengan algoritma KMP.
2	Rivalri Kristian to Hondro	Implementasi Algoritma <i>Knuth Morris Pratt</i> Pada Aplikasi Penerjemahan Bahasa Mandailing Indonesia	2016	Knuth Morris Pratt	Kegunaan Algoritma <i>Knuth Morris Pratt</i> sangat penting dalam aplikasi Penejemahan Bahasa Mandailing – Indonesia karena dapat mempermudah pencarian kata yang ingin diterjemahkan ke dalam Bahasa Indonesia. Aplikasi ini juga dapat membantu pengguna yang ingin belajar Bahasa Mandailing.
3	Fince Tinus Wawuru	Implementasi Algoritma <i>Knuth Morris Pratt</i> Pada Aplikasi Kamus Istilah Latin Flora dan	2014	Knuth Morris Pratt	Algoritma <i>Knuth Morris Pratt</i> di gunakan untuk mempermudah pengguna untuk pencarian kata yang ingin diterjemah. Aplikasi ini menggunakan Bahasa pemograman java,

		Fauna Berbasis Android			dan menggunakan <i>software editor Eclipse Galelio</i> dan untuk menjalankan program aplikasi ini menggunakan emulator Eclipse.
--	--	------------------------	--	--	---

## 2.2. Tinjauan pustaka

### 2.2.1. Bahasa Mongondow

Bahasa mongondow adalah Bahasa yang digunakan suku Mongondow yang berada di Sulawesi Utara (sekarang Bolaang Mongondow, Bolaang Mongondow Utara, Bolaang Mongondow Selatan, dan Bolaang Mongondow Timur)[1]. Bahasa Mongondow Asli dipergunakan untuk penyampaian bahasa adat, dengan mempergunakan kiasan, perumpamaan, sindirannya tolibang, Aimbu (upacara pengobatan orang sakit), tangki-tangkian (teka-teki) dan sa-la-mat (tanda selamat). Bahasa mongondow merupakan bahasa dengan jumlah penutur peringkat ke-10 terbanyak di pulau sulawesi. Bahasa mongondow adalah bahasa lisan murni, bahkan aspek budaya pengetahuan suku mongondow dipelihara secara lisan (oral) secara turun temurun tanpa bantuan tulisan sama sekali. Bahasa mongondow juga dituturkan sebagai bahasa kedua atau ketiga bagi etnis-etnis asli bolaang mongondow raya. Bahasa ini sekarang berstatus terancam punah karena semakin tergeser posisinya oleh bahasa melayu manado sebagai bahasa lisan utama etnis mongondow.

Bahasa Mongondow sehari-hari sebagai pengantar komunikasi masyarakat banyak menyerap bahasa asing, seperti[4] :

Ba'l = Bola

Masina = Mesin

Bangko = Bangku

Gałas = Gelas

Kadera = Kursi

Garagaji = Gergaji

Usup (1986) mengidentifikasi 2 kelompok besar dialek bahasa mongondow yaitu :

1. Dialek Passi, dituturkan di kampung-kampung di sekitar bukit Passi sebelah utara daratan Mongondow, dan
2. Dialek Lolayan, dituturkan di kampung-kampung di daratan Mongondow bagian selatan serta lembah Dumoga.

Interaksi intens etnis mongondow dengan etnis lain pada masa lampau memang mengakibatkan sejumlah adopsi dan adaptasi kosa kata antar bahasa, namun tidak dominan. Interaksi pertama yaitu dengan etnis Minahasa dan Sangir. kemudian dengan Ternate dilanjutkan dengan Belanda dan Spanyol. adopsi bahasa Ternate sebagian besar terkait gelar-gelar pemerintahan (contohnya kapita laut, sangadi). Sedangkan adopsi bahasa Eropa umumnya ialah kosa kata nama barang, institusi, atau flora-fauna. Sepanjang sejarah interaksi tersebut, adopsi kata kedalam bahasa Mongondow pasti di ikuti dengan adaptasi bunyi disesuaikan dengan kaidah bunyi asli bahasa Mongondow, sehingga kata-kata adopsi tersebut tidak kentara berasal dari bahasa asing.

Tulisan berbahasa Mongondow tertua (abad 19) ditulis dengan huruf latin dengan ejaan Belanda. Sebelumnya di antara para raja, jogugu, atau bangsawan Bolaang Mongondow lainnya ada yang bisa baca tulis huruf latin, namun mereka tidak menggunakannya untuk menulis dalam bahasa mongondow, melainkan menulis dalam bahasa melayu tinggi untuk berkorespondensi dengan penguasa-penguasa negeri tetangga. Adapun non-bangsawan mulai dari pemuka adat hingga rakyat biasa tiada yang merasa memerlukan keterampilan baca tulis karena tradisi lisan yang sangat kuat dipandang sudah mencukupi kebutuhan transmisi informasi.

### 1.1.1 2.2.2 Algoritma Knuth Morris Pratt

Algoritma *Knuth-Morris-Pratt* adalah salah satu algoritma pencarian *string*, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977[5]. Cara kerja algoritma ini adalah :

- a. Algoritma *Knuth Morris Pratt* mulai mencocokkan *pattern* (pola) pada awal teks.
- b. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter ***Pattern*** dengan karakter di **Teks** yang bersesuaian, sampai salah satu kondisi berikut terpenuhi :
  - 1) Karakter di ***Pattern*** dan di **Teks** yang dibandingkan tidak cocok (*mismatch*).
  - 2) Semua karakter di ***Pattern*** cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
- c. Algoritma kemudian menggeser ***Pattern*** berdasarkan *table next* (lompat), lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks.

Contoh penghitungan algoritma *Knuth Morris Pratt* adalah sebagai berikut.

***Pattern***: formasi

**Teks** : info inform diinformasikan

Dalam algoritma pencarian *string*, teks diasumsikan berada di dalam memori, sehingga bila kita mencari *string* di dalam sebuah arsip, maka semua isi arsip perlu dibaca terlebih dahulu, kemudian disimpan di dalam memori. Jika *pattern* muncul lebih dari sekali di dalam teks, maka pencarian hanya akan memberikan keluaran berupa lokasi *pattern* ditemukan pertama kali[6]:

	I	N	F	O	I	N	F	O	R	M	D	I	I	N	F	O	R	M	A	S	I	K	A	N	
1	F	O	R	M	A	S	I																		
2		F	O	R	M	A	S	I																	
3			F	O	R	M	A	S	I																
4				F	O	R	M	A	S	I															
5					F	O	R	M	A	S	I														
6						F	O	R	M	A	S	I													
7							F	O	R	M	A	S	I												
8								F	O	R	M	A	S	I											
9									F	O	R	M	A	S	I										
10										F	O	R	M	A	S	I									
11											F	O	R	M	A	S	I								

**Gambar 2.1** Cara kerja algoritma KMP.



- 1) Mula-mula *pattern* dan teks sejajar pada posisi paling kiri, dan dibandingkan karakter pertamanya.
- 2) Langkah 1 dan 2 masing – masing terjadi pergeseran 1 karakter ke karakter kanan karena karakter pertama dan kedua terjadi kecocokkan.
- 3) Pada langkah 3, ketidakcocokan terjadi saat membandingkan karakter ke-3 dari *pattern* (“R”) dan karakter ke-5 dari teks (spasi). Metode KMP akan memeriksa apakah pada teks yang dilewati pada langkah ini (yaitu F-O-spasi) terdapat karakter awal *pattern* (yaitu F), dan ternyata cocok di karakter pertama yang sudah dibandingkan pada langkah ini. Karena itu tidak perlu menggeser *pattern* satu per satu, *pattern* dapat langsung digeser sejauh 3 karakter pada langkah selanjutnya (langkah 4).
- 4) Kasus yang sama ditemui lagi pada langkah 6, di mana *pattern* dapat digeser sejauh 5 karakter ke kanan pada langkah 7. Demikian seterusnya, hingga pada akhirnya hanya diperlukan 11 langkah untuk menemukan *pattern* pada teks.

### 2.2.3 Android

Berapa tahun belakangan ini seluruh dunia telah dihebohkan dengan adanya platform baru yang kurang lebih telah menguasai pasar global dan sudah lebih dari setengah persen pengguna ponsel telah menggunakan sistem informasi tersebut. Ya platform yang di maksud adalah Android. Android berfungsi sebagai penghubung antara pengguna dan perangkat keras pada *smartphone*. Android baru di rilis pada bulan Oktober tahun 2003 oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White di bawah sebuah perusahaan bernama Android Inc. di Palo Anthom, California. Sebelum akhirnya diakuisisi oleh Google pada tahun 2005[7].

Tujuan awal platform ini adalah untuk mengembangkan sebuah sistem operasi yang lebih canggih bagi kinerja dari sebuah kamera digital. Namun, keberadaan pasar global mengubah arus Andy dan kawan-kawan untuk membawa Android Inc. beralih fungsi sebagai perusahaan yang bergerak pada pengembangan sistem operasi *smartphone*. Pada tanggal 5 November 2007 android meluncurkan

versi beta pertama kali. Setelah satu minggu peresmian versi beta android meluncurkan *Software Development Kit* atau SDK pada tanggal 12 November 2007. SDK memungkinkan pengguna untuk dapat berkontribusi, membuat dan mengembangkan sendiri aplikasi Android mereka[7].

#### 2.2.4 Generasi Android

Android ini juga akan berusaha memperbaharui sistem operasinya. Berikut ini adalah daftar versi android dari awal terbentuknya android hingga saat ini.

Versi	Nama	Tanggal Liris
1.0 (API level 1)	-	23 September 2008
1.1 (API level 2)	-	9 Februari 2009
1.5 (API level 3)	Cupcake	27 April 2009
1.6 (API level 4)	Donut	15 September 2009
2.0 (API level 5)	Éclair	26 Oktober 2009
2.0.1 (API level 6)	Éclair	3 Desember 2009
2.1 (API level 7)	Éclair	12 Januari 2010
2.2-2.2.3(API level 8)	Froyo	20 Mei 2010
2.3-.2.3.2 (API level 9)	Gingerbread	6 Desember 2010
2.3.3-2.3.7 (API level 10)	Gingerbread	9 februari 2011
3.0 (API level 11)	Honeycomb	22 Februari 2011
3.1 (API level 12)	Honeycomb	10 Mei 2011
3.2 (API level 13)	Honeycomb	15 Juli 2011
4.0-4.0.2 (API level 14)	Ice cream sandwich	19 Oktober 2011
4.0.3-4.0.4 (API level 15)	Ice cream sandwich	16 Desember 2011
4.1 (API level 16)	Jelly Bean	27 Juni 2012
4.2 (API level 17)	Jelly Bean	29 Oktober 2012
4.3 (API level 18)	Jelly Bean	24 Juli 2013
4.4 (API level 19)	KitKat	31 Oktober 2013
5.0 (API level 21)	Lollipop	12 November 2014
6.0 (API level 23)	Marshmallow	5 Oktober 2015

7.0 (API level 24)	Nama-Nama Versi	Tanggal
7.1 (API level 25)	Nougat	19 Oktober 2016
8.0 (API level 26)	Oreo	21 Maret 2017
9.0	Pie	2 juli 2018

## 2.2.5 Pengembangan Sistem

### 2.2.5.1 SDK (*Software Development Kit*)

Ada terdapat sebuah software atau perangkat lunak yang dapat digunakan untuk membangun aplikasi android. SDK (*software development kit*) adalah sebuah kit yang berfungsi untuk mengembangkan aplikasi android. Dalam SDK terdapat berbagai tools yang berfungsi untuk proses pengembangan, aplikasi seperti proses debugger, emulator, software libraries, dokumentasi, tutorial maupun *sample code*[7].

## 2.2.6 Analisis Sistem

### 2.2.6.1 *Unified Modelling Language (UML)*

Menurut Windu Gata, Grace (2013:4), *Unified Modelling Language (UML)* adalah Bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, memspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat pendukung pengembangan sistem[8].

### 2.2.6.2 *Diagram Dasar UML*

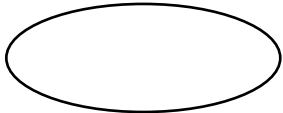
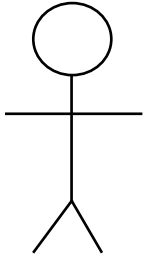


Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut[8]:


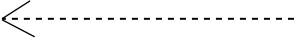
#### 1. *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan

fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram* yaitu [8]:

**Tabel 2.3.** *Use case diagram.*


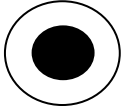
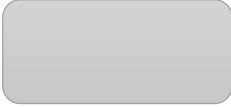
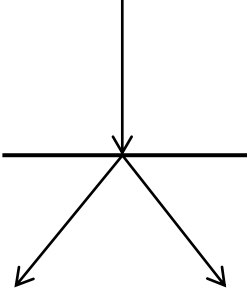
SIMBOL	KETERANGAN
	<p>Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan actor, yang dinyatakan dengan menggunakan kata kerja.</p>
	<p><i>Actor</i> adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi actor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.</p>
	<p>Asosiasi antara actor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p>
	<p>Asosiasi antara actor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila actor berinteraksi secara pasif dengan sistem.</p>

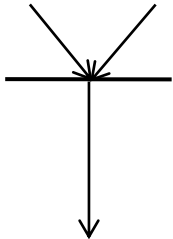
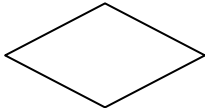

	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

## 2. Activity Diagram

*Activity diagram* menggambarkan workflow (alir kerja) atau aktifitas dari sebuah sistem atau proses sebuah bisnis. Symbol-simbol yang digunakan adalah sebagai berikut[8].

**Tabel 2.4.** *activity diagram*.


SIMBOL	KETERANGAN
	<p><i>Start point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.</p>
	<p><i>End point</i>, akhir aktivitas.</p>
	<p><i>Activities</i>, menggambarkan suatu proses/kegiatan bisnis.</p>
	<p><i>Fork</i>/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua</p>

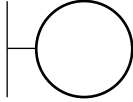
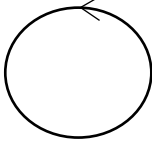


	kegiatan parallel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision point</i> , menggambarkan pilihan untuk pengambilan keputusan <i>true</i> atau <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

### 3. Sequence diagram

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu[8]:

**Tabel 2.5.** *sequence diagram*.

SIMBOL	KETERANGAN
	<i>Entity class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal

	sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan form cetak.
	Control class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Message, symbol mengirim pesan antar class.
	Lifeline, garis titik-titik yang terhubung dengan objek sepanjang lifeline terdapat activation

#### 4. *Multiplicity class diagram*

*Multiplicity class diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan[8].

Class diagram secara khas meliputi : kelas (*class*), relasi associations, *generalisation*, *aggregation*, atribut (*attributes*), operasi (*operation/method*), dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality*[8].

**Tabel 2.6.** *multiplicity class diagram.*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	Satu atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4.

## 2.2.7 Desain Sistem

### 2.2.7.1 Pengertian Object Oriented Analysis And Design (OOAD)

OOAD adalah pendekatan yang dilakukan untuk mengembangkan suatu sistem baru yang responsif terhadap lingkungan bisnis yang cenderung berubah. Unified Modeling Language (UML) merupakan standar industri yang digunakan untuk pemodelan sistem yang menggunakan pendekatan OOAD. UML mencakup diagram yang menggambarkan pembuatan sistem yang berguna untuk meningkatkan kualitas analisis dan desain sistem (Kendall & Kendall, 2011)[9].



## 2.2.8 Konstruksi Sistem

### 2.2.8.1 *Android Studio*

Android studio merupakan integrated development environment (IDE) atau dalam artian lain adalah sebuah lingkungan pengembangan terintegrasi resmi yang memang dirancang khusus untuk pengembangan sistem operasi Google Android. Aplikasi yang satu ini, dibangun di atas sebuah perangkat lunak yang dinamakan IntelliJ IDEA milik JetBrains. Bisa juga dibilang bahwa android studio merupakan pengganti dari Eclipse Android Development Tools atau ADT sebagai IDE utama dalam pengembangan aplikasi android yang asli. Diluncurkan pada 16 Mei 2013 dalam konferensi Google I/O yang pada saat itu masih dalam tahap pratinjau akses versi 0.1 sebagai perintis. Hingga pada akhirnya versi stabil 3.0 yang rilis pada pertengahan bulan Oktober 2017 ini menjadi software terlaris dikalangan development muda[7].

### 2.2.8.2 *Java Development Kit (JDK)*

*Java development kit* (JDK) adalah sekumpulan perangkat lunak yang dapat digunakan untuk mengembangkan perangkat lunak yang berbasis java, sedangkan JRE adalah implementasi dari dari *Java Virtual Machine* yang benar benar digunakan untuk menjalankan program java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber *compiler java*, *bundling*, *debuggers*, *development*, *libraries* dan lain sebagainya[10].

### 2.2.8.3 *Database*

Menurut Asrianda dalam Urva, Gellysa (2008) *database* sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu *database* menunjukkan satu lingkup perusahaan atau instansi. *Database* juga merupakan kumpulan data yang umumnya menggambarkan aktifitas-aktifitas dan pelakunya dalam suatu organisasi[8].

#### 2.2.8.4 *Realm*

*Realm* merupakan *library database* yang digunakan untuk perangkat *mobile smartphone* yang menggunakan konsep tanpa skema atau biasa disebut dengan konsep NO SQL. Biasanya *realm* banyak digunakan untuk penyimpanan *database* yang bersifat sementara pada *platform* Android maupun iOS, karena kemudahannya serta keahlian baca dan tulis yang bisa dibilang lebih baik dari *database* lainnya. Tapi, kekurangan *database* ini adalah hanya bisa digunakan untuk penyimpanan secara *offline*, atau pada sisi klien aplikasi *mobile* saja[11].

#### 2.2.9 Pengujian Sistem

Pengujian sistem adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean. Tujuan dari pengujian ini adalah diharapkan dengan minimal tenaga dan waktu untuk menemukan berbagai potensi kesalahan dan cacat. Pengujian yang dilakukan pada penelitian ini yaitu Kompleksitas algoritma, *white box*, dan pengujian *black box*.

##### 2.2.9.1 Kompleksitas Algoritma KMP

Algoritma yang baik adalah algoritma yang efisien. Efisiensi suatu Algoritma di lihat dari beberapa jumlah waktu yang dibutuhkan dan juga ruang yang dibutuhkan untuk menjalankannya. Kebutuhan algoritma terhadap waktu dan ruang bergantung pada ukuran masukan, yang menyatakan jumlah data yang di proses[12]. Kompleksitas waktu algoritma KMP cukup cepat, yaitu  $O(m+n)$  yang di peroleh dari waktu untuk menghitung fungsi pinggiran  $O(m)$  dan proses pencarian string  $O(n)$ . Fungsi pinggiran karena menghitung semua kemungkinan  $J$  dan nilai  $J$  maksimal pasti panjang karakter yaitu  $m$ . sedangkan proses pencarian string selalu bergerak maju dan tidak pernah mengulang seperti brute force. Jadi batas atasnya adalah  $O(n)$ .

### 2.2.9.2 *White Box Testing*

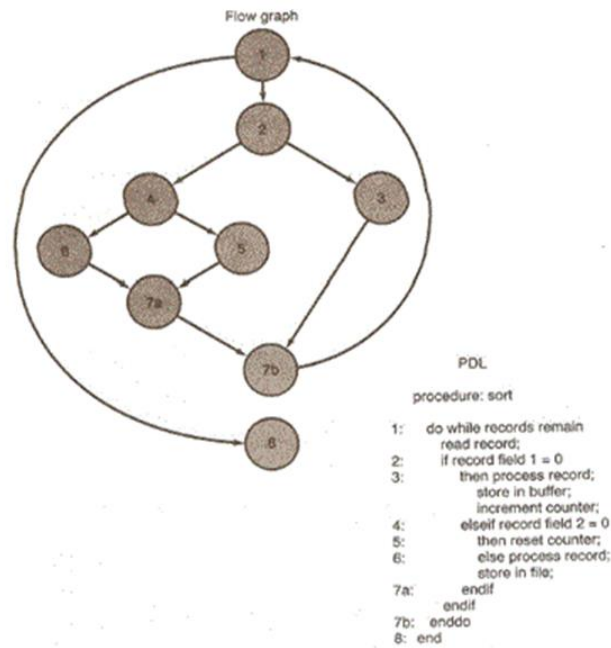
White box testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara *procedural* untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *white box testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100% [13].

Tujuan *white box testing* yaitu untuk mengetahui cara kerja suatu perangkat lunak secara internal, dan untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang [13].

Pelaksanaan White box Testing :

1. Menjamin seluruh *independent path* dieksekusi paling sedikit satu kali. *Independent path* adalah jalur dalam program yang menunjukkan paling sedikit satu kumpulan proses ataupun kondisi baru.
2. Menjalani *logical decision* pada sisi dan false.
3. Mengeksekusi pengulangan (*looping*) dalam batas-batas yang ditentukan.
4. Menguji struktur internal.

Notasi yang digunakan untuk menggambarkan jalur eksekusi adalah notasi diagram alir atau grafik program, yang menggunakan notasi lingkaran (simpul atau *node*), dan anak panah (*link* atau *edge*). Setiap representasi rancangan prosedural dapat diterjemahkan kedalam *flow graph*. Gambar dibawah ini merupakan bagian dari PDL (*Program Design Language*) dan *flow graph*-nya.



**Gambar 2.2.** Grafik alir.

Berdasarkan contoh PDL yang pertama, maka jalur *independent* yang di dapat adalah :

Jalur 1 : 1 – 11

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Kompleksitas siklomatis dapat dicari dengan salah satu dari 3 cara berikut :

1. Jumlah region dari grafik alir mengacu kepada kompleksitas siklomatis.
2. Kompleksitas siklomatis  $V(G)$  untuk grafik alur  $G$  didefinisikan sebagai :  

$$V(G) = E - N + 2$$
dimana  $E$  = jumlah edge,  $N$  = jumlah node
3. Kompleksitas siklomatis  $V(G)$  untuk grafik alur  $G$  didefinisikan sebagai :

$$V(G) = P + 1,$$
dimana  $P$  = jumlah predicates nodes yang diisikan dalam grafik alir  $G$

Simpul predikat adalah penggambaran suatu node yang memiliki satu atau lebih inputan, dan lebih dari satu output. Berdasarkan flow graph pada gambardiatas, maka kompleksitas siklomatisnya dapat dihitung sebagai berikut:

GrafikAlir di atas mempunyai 4 region

$$V(G) = 11 \text{ edges} - 9 \text{ nodes} + 2 = 4$$

$$V(G) = 3 \text{ predicates nodes} + 1 = 4$$

Hasil kompleksitas siklomatis menggambarkan banyaknya *path* dan batas atas sejumlah uji coba yang harus dirancang dan di eksekusi untuk seluruh perintah dalam program.

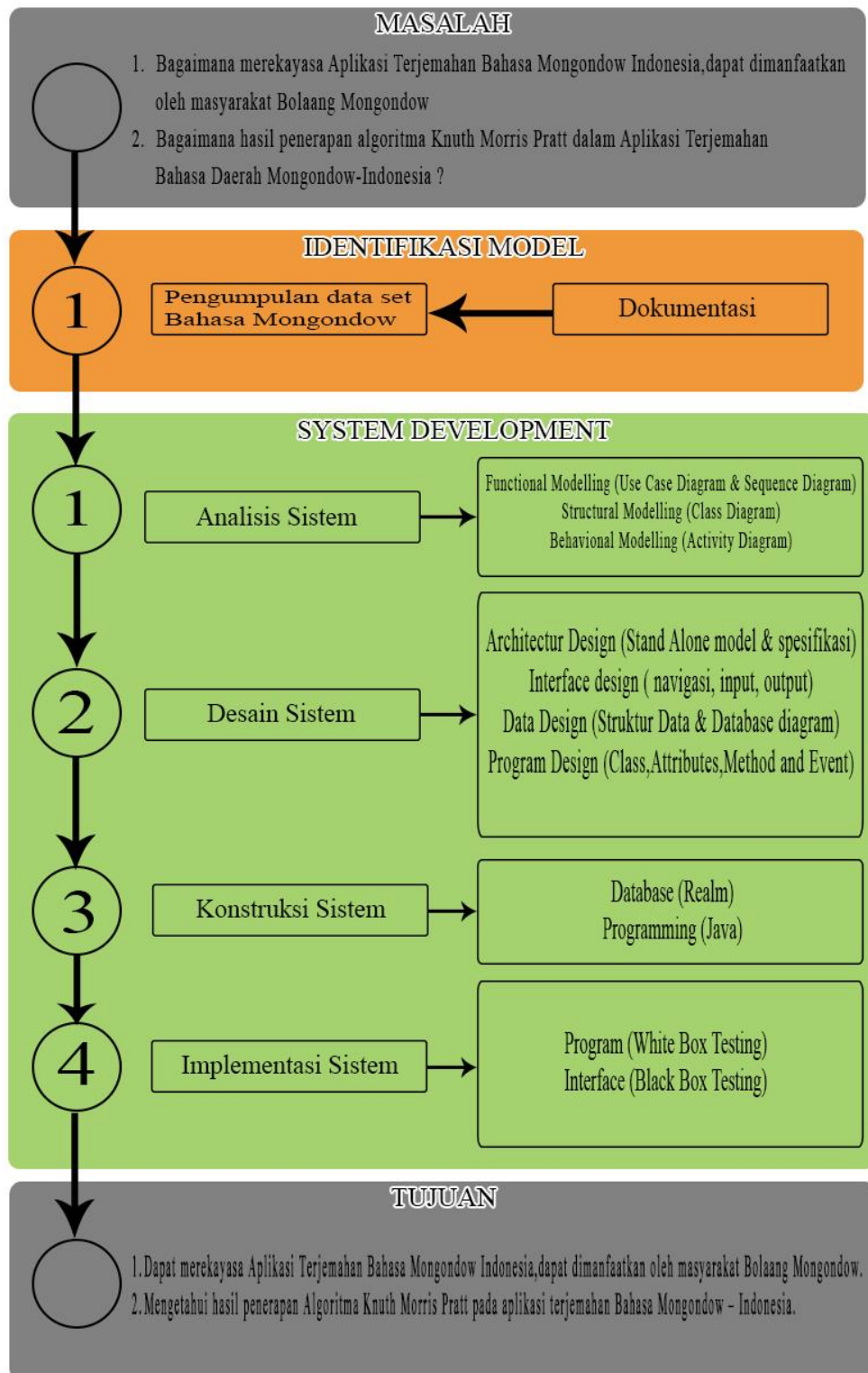
### 2.2.9.3 Black Box Testing

*Black box testing* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat di artikan sebagai kumpulan kondisi input dan melakukan pengetesan pada program. *Black box testing* adalah pelengkap *white box* untuk menguji hal-hal yang tidak di cakup oleh *white box testing*[14].

*Black box testing* cenderung untuk menemukan hal-hal berikut :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

### 2.3 Kerangka Pikir



## **BAB III**

### **OBJEK DAN METODE PENELITIAN**

#### **3.1. Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian**

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data, maka penelitian ini merupakan penelitian konfirmatori.

Penelitian ini menggunakan metode penelitian tindakan. Dengan demikian jenis penelitian ini adalah penelitian deskriptif.

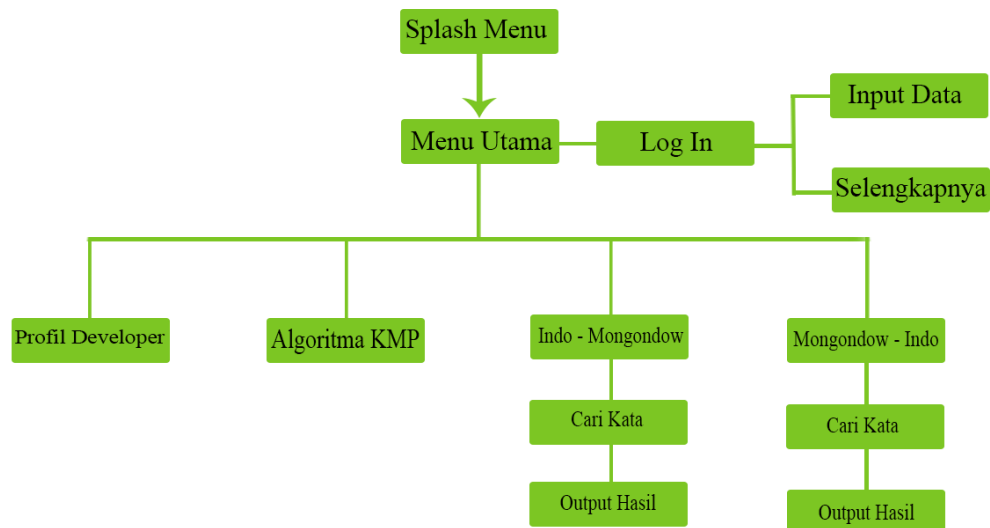
Subjek penelitian ini adalah pencarian pada objek Bahasa daerah Bolaang Mongondow. Penelitian ini di mulai dari bulan Mei 2019 s/d Agustus 2019 yang berlokasi di Daerah Bolaang Mongondow, Kota Kotamobagu.

#### **3.2. Pengumpulan Data**

Data primer penelitian ini adalah kata Bahasa Mongondow, Kata Bahasa Indonesia, cara pengucapan 'L', 'l', dan 'l' yang dikumpulkan menggunakan teknik dokumentasi. Sedangkan data sekunder juga dikumpulkan dengan Teknik dokumentasi.

### 3.3.Pengembangan Sistem

#### 3.3.1. Sistem yang di usulkan



**Gambar 3.1** Sistem yang di usulkan.

#### 3.3.2. Analisis Sistem

Analisis sistem menggunakan pendekatan berorientasi objek yang digambarkan dalam bentuk:

(a) *Functional Modelling*, menggunakan alat bantu UML,dalam bentuk :

*Use case diagram*

*Actifity diagram*

(b) *Structural Modelling*, menggunakan alat bantu UML,dalam bentuk :

*Class diagram*

(c) *Behavior Modelling*, menggunakan alat bantu UML dalam bentuk :

*Sequence diagram*



### 3.3.3. Desain Sistem

Desain sistem menggunakan pendekatan berorientasi objek yang digambarkan dalam bentuk:

(a) *Architecture Design*, menggunakan alat bantu Android Studio, dalam bentuk :

- Model jaringan dari sistem stand alone
- Spesifikasi Smartphone dengan sistem Operasi Android versi 5.1 (Lolipop)

(b) *Interface Design*, menggunakan alat bantu Android Studio, dalam bentuk:

- Mekanisme Navigasi
- Mekanisme Input
- Mekanisme Output

(c) *Data Design*, menggunakan alat bantu Android Studio, dalam bentuk :

- Format data yang digunakan SQLite
- Struktur Data
- Database Diagram

(d) *Program Design*, menggunakan alat bantu Android Studio, dalam bentuk:

- *Class*
- *Attributes*
- *Method*
- *Event*

### 3.3.4. Konstruksi Sistem

Pada tahap ini menerjemahkan hasil pada tahap analisis dan desain ke dalam kode – kode program komputer kemudian membangun sistemnya. Alat bantu yang digunakan pada tahap ini adalah Android Studio dengan Bahasa Pemrograman Java. Alat bantu *database* yang digunakan Realm.

### 3.3.5. Pengujian Sistem

#### a. White Box Testing

*Software* yang telah direkayasa kemudian diuji dengan metode *White Box Testing* pada kode program proses penerapan metodenya/modelnya. Kode program tersebut dibuatkan *flowchart* programnya, kemudian dipetakan kedalam bentuk *flowgraph* (bagan alir kontrol) yang tersusun dari beberapa *node* dan *edge*. Berdasarkan *flowgraph*, ditentukan jumlah *Region* dan *Cylomatic complexity* (CC). Apabila *Independent path* =  $V(G) = (CC) = Region$ , dimana setiap *path* hanya di eksekusi sekali dan sudah benar, maka sistem dinyatakan efisien dari segi kelayakan logika pemrograman[15].

#### b. Black Box Testing

Selanjutnya, *software* diuji pula dengan metode *Black Box Testing* yang fokus pada keperluan fungsional dari *software* dan berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya : (1) Fungsi – fungsi yang salah atau hilang; (2) Kesalahan *interface*; (3) Kesalahan dalam struktur data atau akses basis data eksternal; (4) Kesalahan performa; (5) Kesalahan inisialisasi dan terminasi. Jika sudah tidak ada kesalahan – kesalahan tersebut, maka sistem dinyatakan efisien dari segi kesalahan komponen – komponen sistem[15].

## BAB IV HASIL PENELITIAN

### 4.1 Hasil Pengumpulan Data

Hasil pengumpulan data primer dalam penelitian ini adalah sebagai berikut :

**Tabel 4.1** Hasil Pengumpulan Data.

No.	Abjad	Bahasa Indonesia	Bahasa Mongondow
1	A	Ada	Oyu'on
2		Adik	Ai-ai
3		Air	Tubig
4		Anak	Adi'
5		Angkut	Akut
6		Alas	Aip
7		Akhir	Aher
8		Ayah	Ama'
9	B	Bantal	Olunan
10		Bantu	Abang
11		Banyak	Barong
12		Baru	Bagu
13		Bakar	Anag
14		Bengkak	Baga'
15		Bersin	Ba'an
16		Boleh	Bali'
17	C	Cacar air	Ronto
18		caci	Pa'ad
19		cahaya	Ulinag
20		Celana	Solana
21		Campur	Ba'uk
22		Coba	Ba'at
23	D	Dagu	Obaga

24		Dan	Bo
25		Dangkal	Babow
26		Dapur	Abu
27		Dukung	Baba
28	E	Ekor	Iput
29		Empat	Opat
30		Enam	Onom
31		Eram	O'ob
32	...	...	...
597		kurus	Yutuk

#### 4.1.1 Penerapan Metode

Algoritma KMP akan mengenali kata “BIBIR” dalam kalimat “BIBBIBIR” dengan proses sebagai berikut :

T[i]	B	I	B	B	I	B	I	R
P[j]	B	I	B	I	R			

Algoritma ini melakukan pergeseran dari kiri ke kanan, dan tidak lagi di lakukan satu persatu seperti pada algoritma *brute force*. Sebelum menentukan seberapa banyak pergeseran yang di lakukan maka yang pertama kita lakukan adalah menentukan *prefix* (awalan) dan *suffix* (akhiran). untuk menentukan seberapa banyak pergeseran yang dilakukan, maka mekanismen umum algoritma knuth morris pratt adalah jika terjadi ketidakcocokkan teks T pada T[i] dan *pattern* P pada P[j] yaitu  $T[i] \neq P[j]$  maka pergeseran yang kita lakukan sebanyak nilai *prefix*  $P[0..j-1]$  terbesar pada karakter *pattern* yang sudah sama sebelumnya yang juga merupakan *suffix*  $P[1..j-1]$  dari *pattern* tersebut.

T[i]	B	I	B	B	I	B	I	R
P[j]	B	I	B	I	R			

↗ i  
↘ J=3

Terjadi ketidakcocokan

Dari pattern P kita bisa tahu :

*Prefix* P[0..j-1] = B, BI, BIB

*Suffix* P[1..j-1] = B, IB, BIB

Dari kemungkinan dari *prefix* dan *suffix* tersebut, kita lihat nilai terbesar yang sama dengan pattern P untuk *prefix* dan *suffix*nya adalah BIB. Jadi disini kita mendapatkan nilai 3 dari banyaknya karakter yang sama pada *prefix* dan *suffix*. Nilai 3 yang diperoleh dari nilai yang terbesar pada *prefix* dan *suffix* digunakan sebagai indeks J yang baru untuk memulai pemeriksaan selanjutnya. Sedangkan indeks untuk memulai pemeriksaan baru pada T adalah indeks I, di mana I adalah indeks saat terjadi ketidakcocokkan sebelumnya.

T[i]	B	I	B	B	I	B	I	R
P[j]				B	I	B	I	R

Kita harus tahu pada saat terjadi ketidakcocokkan pada indeks ke 3 pada karakter P berapakah nilai *prefix* terbesar mulai dari indeks 0 hingga j-1 yang sama dengan *suffix* untuk karakter P mulai indeks ke 1 hingga j-1. Karena nilai P ketika pencocokkan string di lakukan maka kita bisa menghitung di awal nilai *prefix* terbesar yang sama dengan *suffix* untuk semua kemungkinan nilai j. Penentuan nilai ini disebut dengan KMP *Border Function*. *Border function* adalah fungsi yang di panggil ketika karakter P telah di input dalam program, namun sebelum proses pencocokkan terhadap teks T dilakukan. Dengan mencari semua nilai *prefix* terbesar pada P[0..j-1], yang sama dengan *suffix* pada P[1..j-1] ketika terjadi ketidakcocokkan pada P[j].

$$J = 0 \ggg K = 0 \ggg B(k) = 0$$

$$J = 1 \ggg K = 0 \ggg B(k) = 0$$

$$J = 2 \ggg K = 1 \ggg B(k) = 1$$

$$J = 3 \ggg K = 2 \ggg B(k) = 1$$

$$J = 4 \ggg K = 3 \ggg B(k) = 2$$

J	0	1	2	3	4
P[j]	B	I	B	I	R
K	0	0	1	2	3
B(k)	0	0	1	1	2

Terlihat pada tabel di atas, J adalah posisi ketidakcocokan terjadi. Dan K adalah posisi J-1. B(k) adalah besaran *prefix* terbesar pada P[0..k] yang menjadi *suffix* dari P[1..k]. cara pengerjaannya adalah sebagai berikut.

$$J = 2 = \text{prefix}[0..1] = BI = \mathbf{B}, BI$$

$$\text{suffix}[1] = B = \mathbf{B}$$

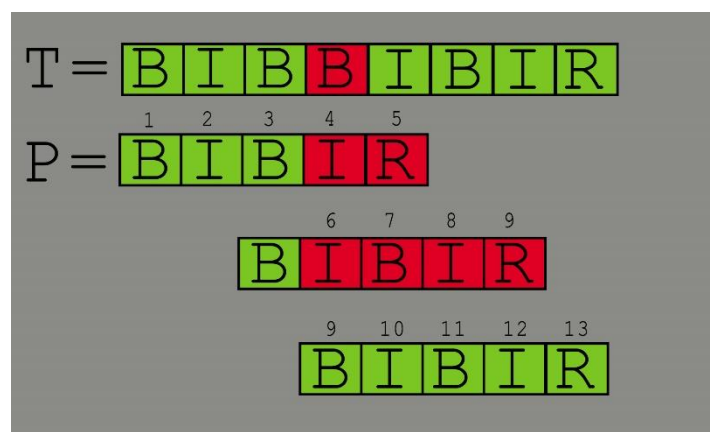
$$J = 3 = \text{prefix}[0..2] = BIB = \mathbf{B}, BI, BIB$$

$$\text{suffix}[1..2] = IB = \mathbf{B}, IB$$

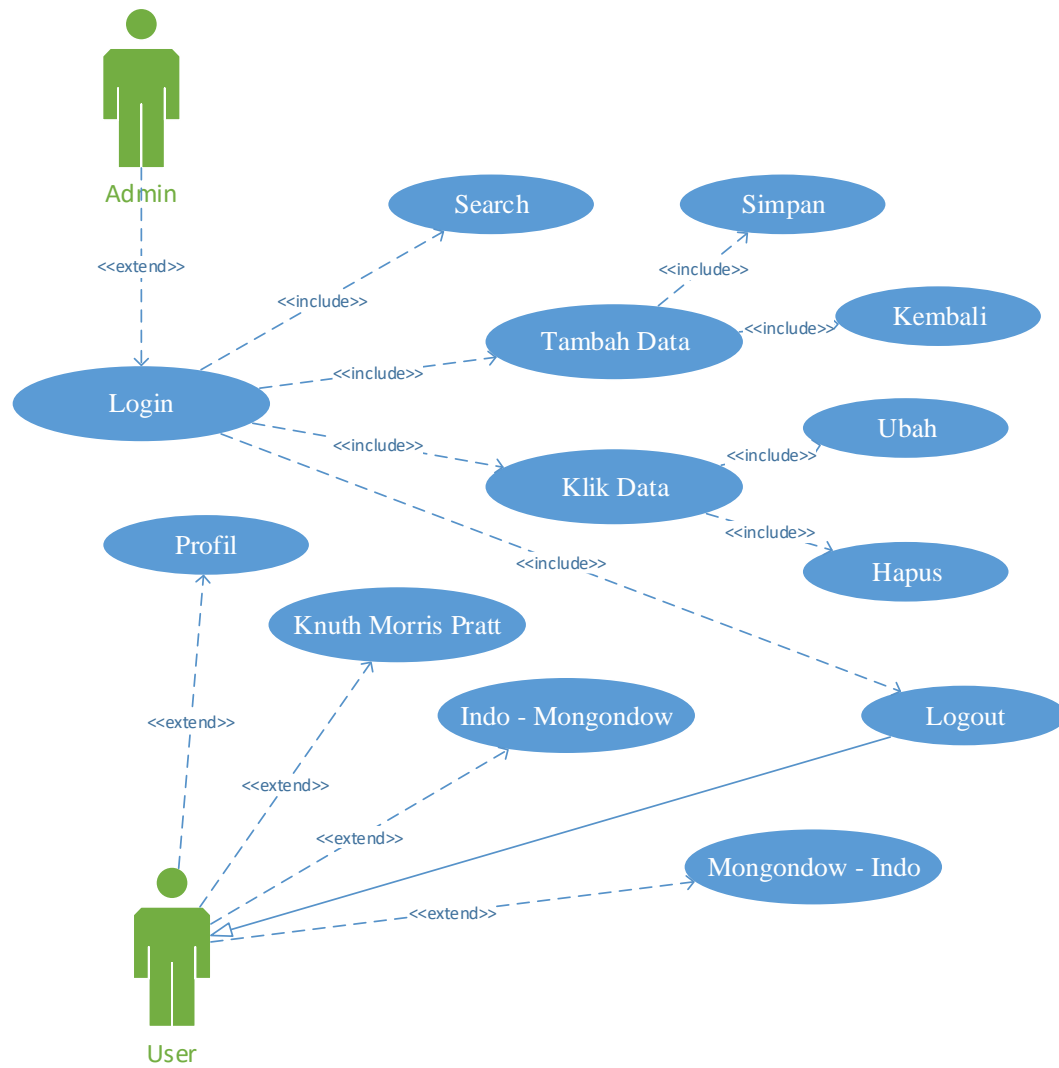
$$J = 4 = \text{prefix}[0..3] = BIBI = \mathbf{B}, \mathbf{BI}, BIB, BIBI$$

$$\text{suffix}[1..3] = IBI = \mathbf{I}, \mathbf{BI}, IBI$$

Dan hasilnya adalah :

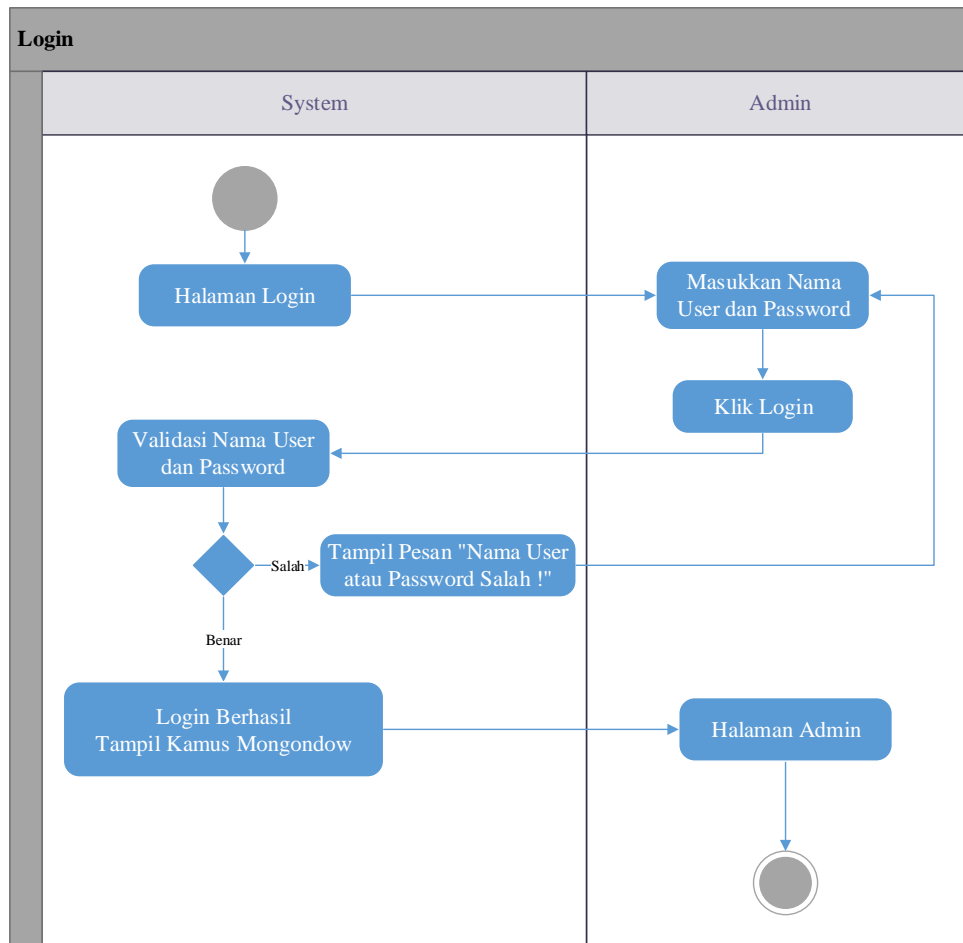


## 4.2 Hasil Pengembangan Sistem



**Gambar 4.1** Use Case Diagram Kamus Bahasa Mongondow.

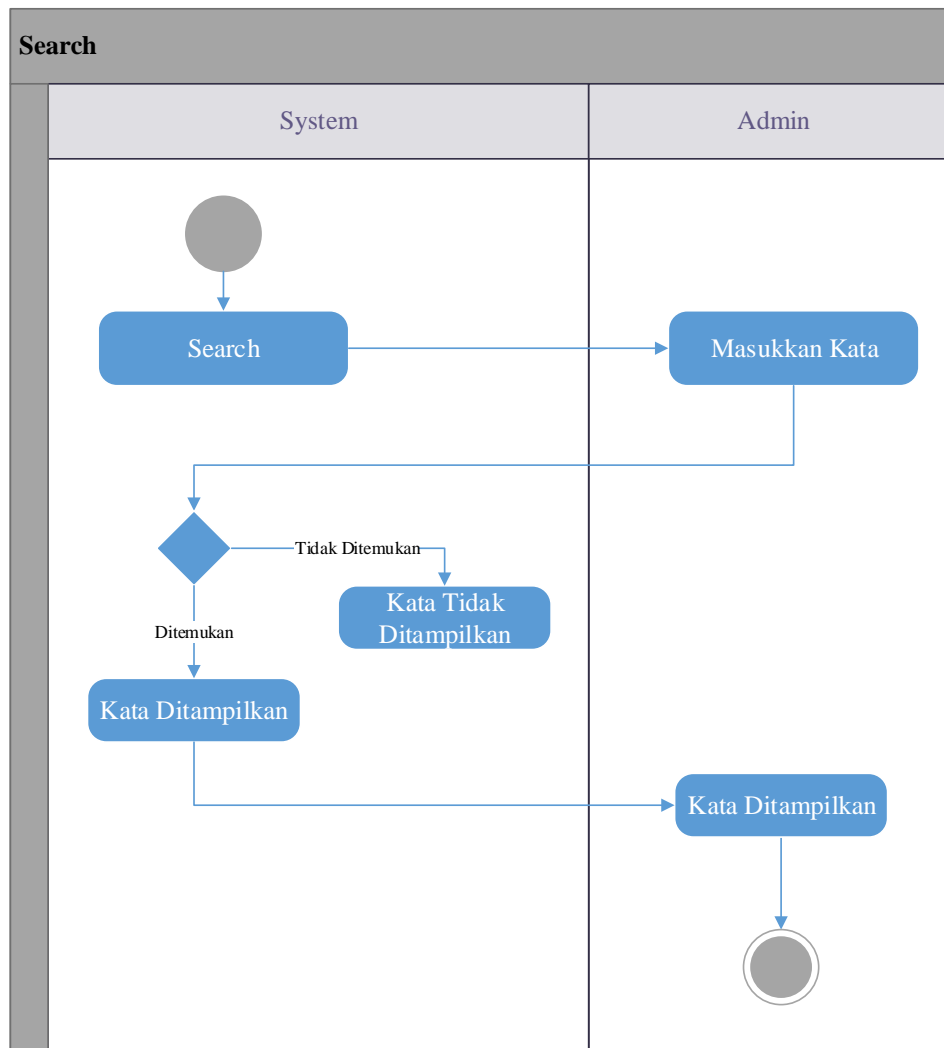
### 4.3 Activity Diagram Login



**Gambar 4.2** Activity Diagram Proses Login.

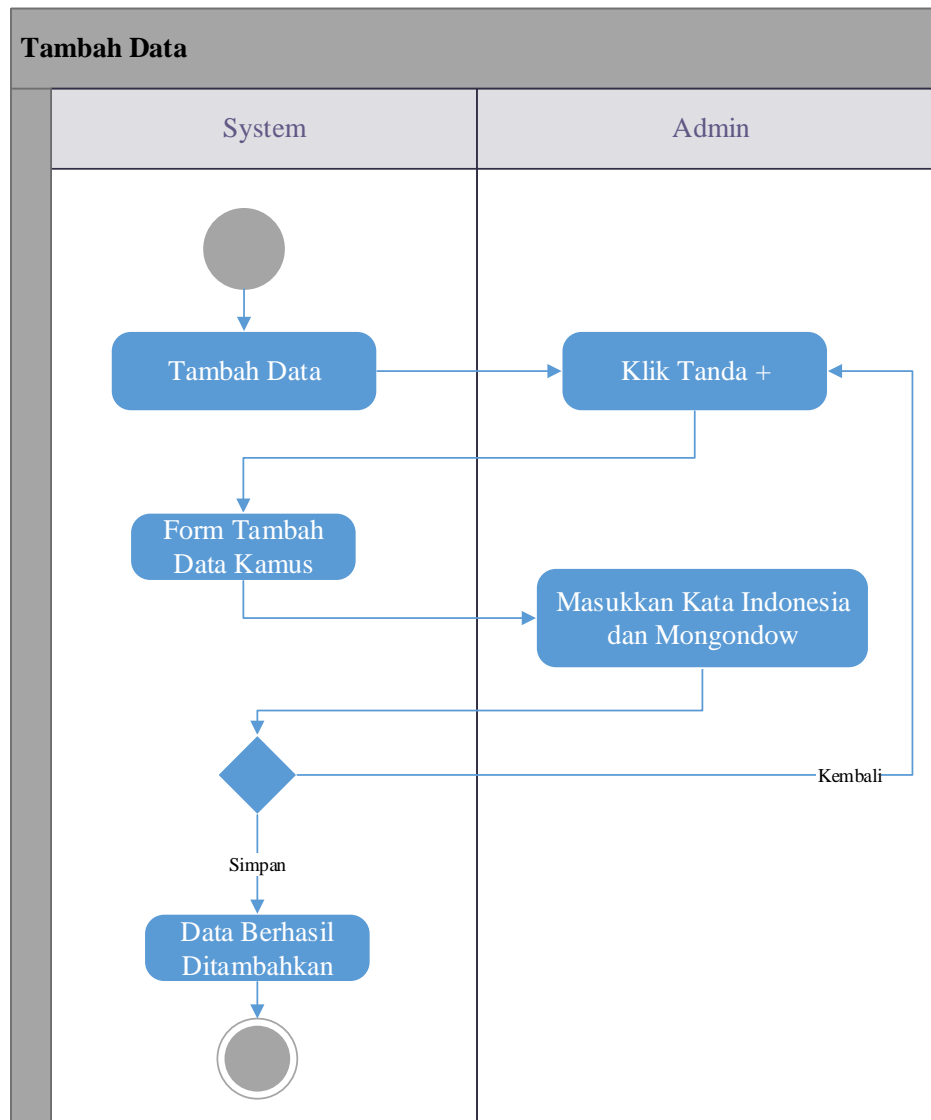


#### 4.4 Activity Diagram Search



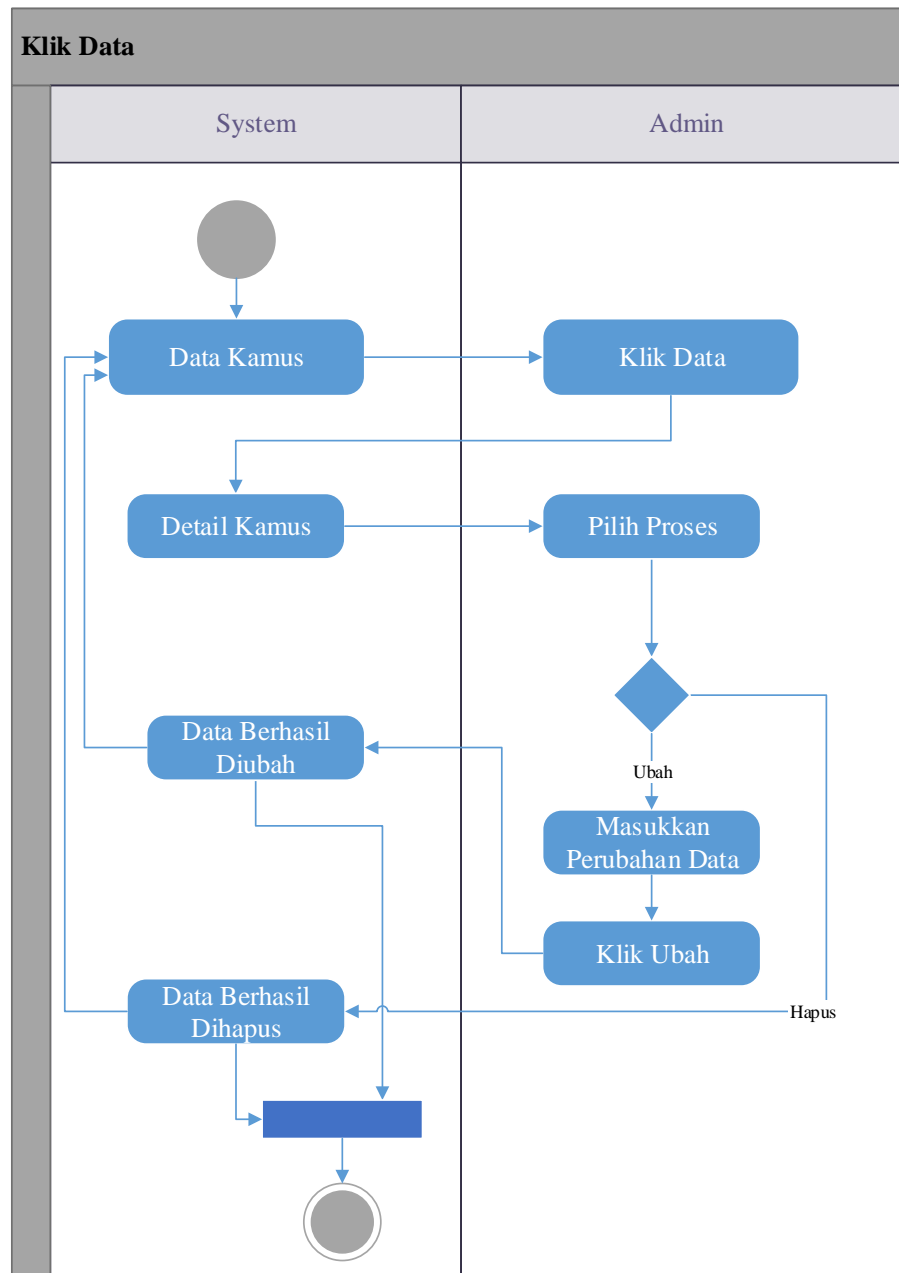
**Gambar 4.3** Activity Diagram Pada Proses Cari Kata.

#### 4.5 Activity Diagram Tambah Data



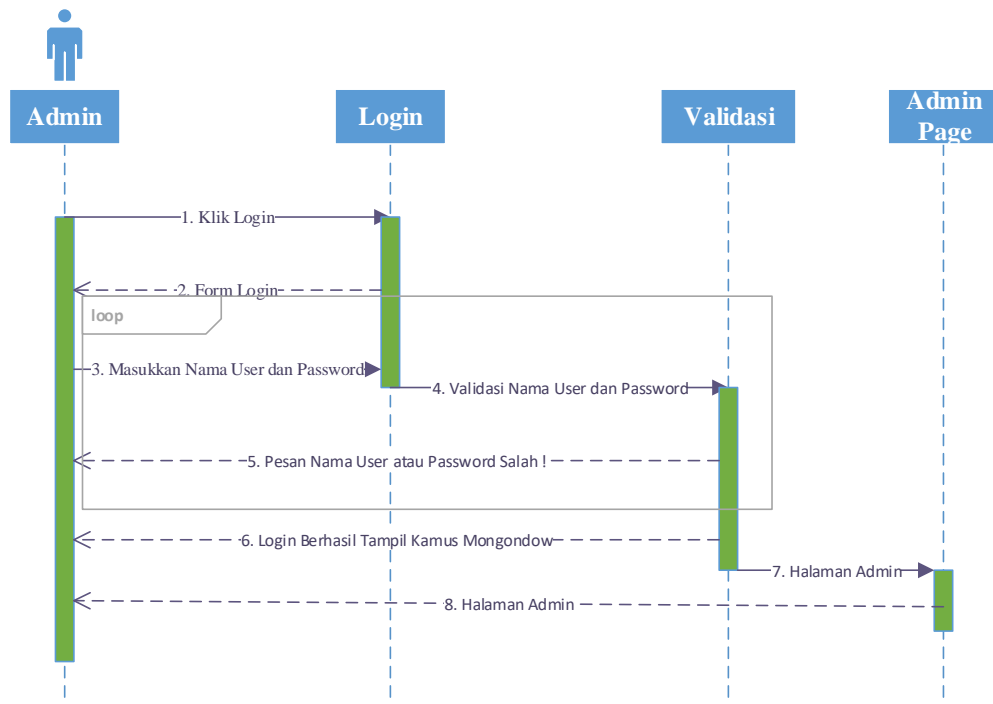
**Gambar 4.4** Activity Diagram Tambah Data.

#### 4.6 Activity Diagram Data Proses Data Kamus



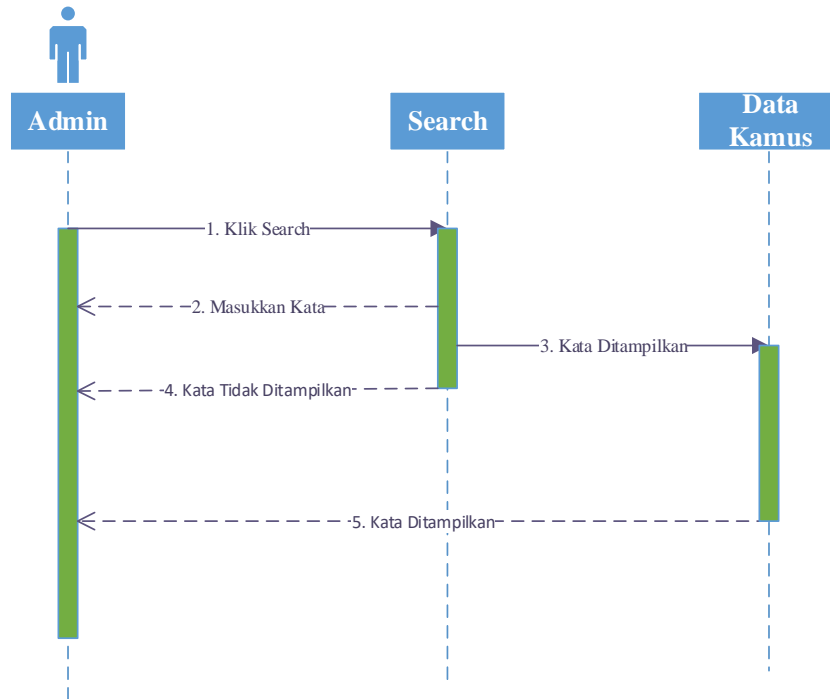
**Gambar 4.5** Activity Diagram Pada Proses Data Kamus.

#### 4.7 Sequence Diagram Login Admin



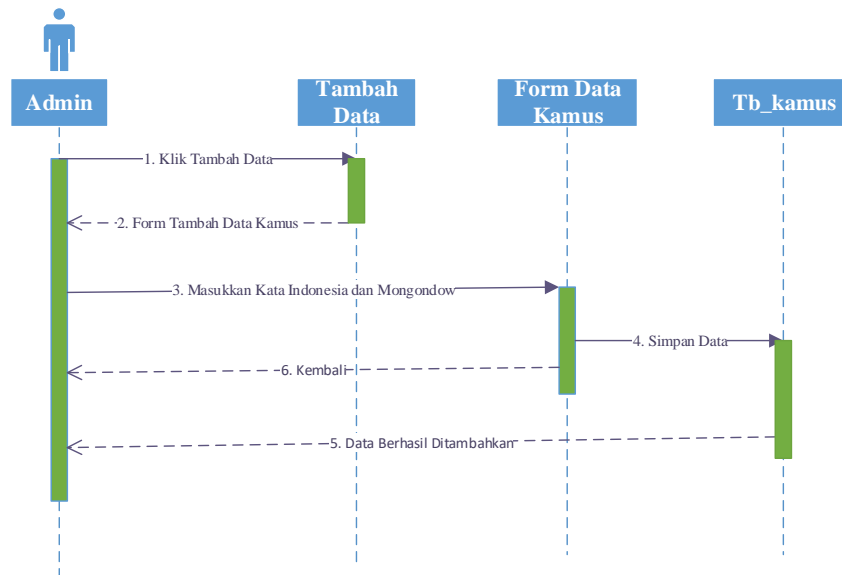
**Gambar 4.6** *Sequence Diagram Login Admin.*

#### 4.8 Sequence Diagram Proses Cari Kata



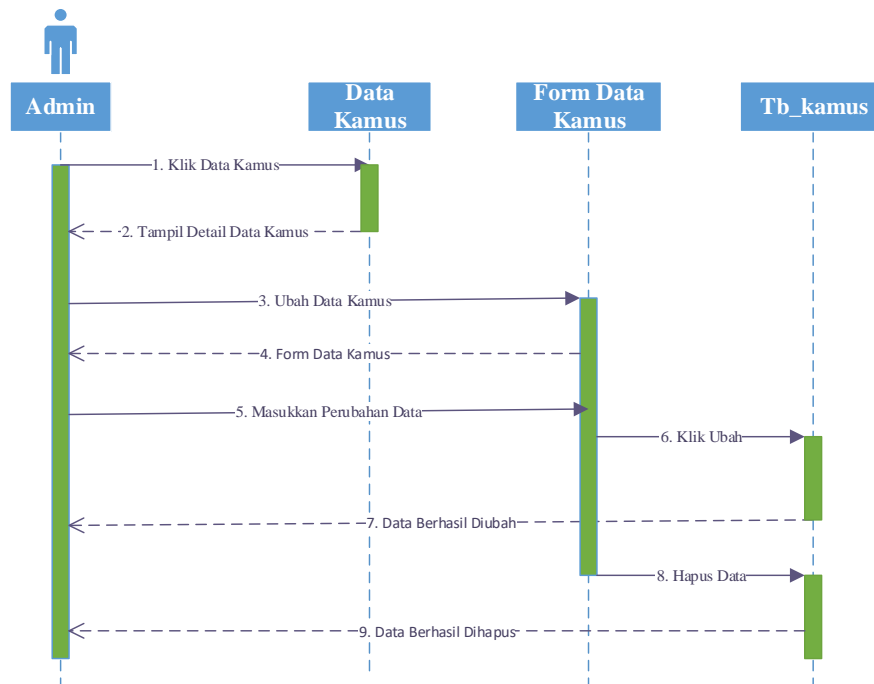
Gambar 4.7 Sequence Diagram Proses Cari Kata.

#### 4.9 Sequence Diagram Tambah Data



Gambar 4.8 Sequence Diagram Tambah Data.

#### 4.10 Sequence Diagram Proses Data Kamus



**Gambar 4.9** *Sequence Diagram* Proses Data Kamus.

#### 4.11 Arsitektur Sistem

Untuk kinerja sistem yang optimal, sebaiknya gunakan perangkat keras dan perangkat lunak sebagai berikut:

1. Processor : Minimal Core i5
2. RAM : 2GB
3. VGA : 16 Bit
4. Hardisk : 500GB
5. Operating System : Windows 8
6. Tools : Android Studio

## 4.12 Interface Design

### 4.12.1 Mekanisme User

**Tabel 4.2** Mekanisme User.

User	Kategori	Akses Input	Akses Output
User	User	-	Kamus
Admin	Administrator	All	All

### 4.12.2 Mekanisme navigasi Home



**Gambar 4.10** Mekanisme Navigasi Home User.

Kamus Indo - Mongondow

Search...

Bahasa Indonesia	Bahasa Mongondow
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....

**Gambar 4.11** Mekanisme Navigasi Home Admin.

#### 4.12.3 Mekanisme *Login*

Selamat Datang

Nama User

Password

<- Kembali      Login

**Gambar 4.12** Mekanisme *Login*.



#### 4.12.4 Mekanisme *Input* data Kamus



<- Tambah Data Kamus

Kata Bahasa Indonesia

Kata Bahasa Mongondow

Simpan

**Gambar 4.13** Mekanisme *Input* data kamus.

#### 4.12.5 Mekanisme Detail, Ubah dan Hapus Data Kamus



<- Detail Kamus - Ubah / Hapus

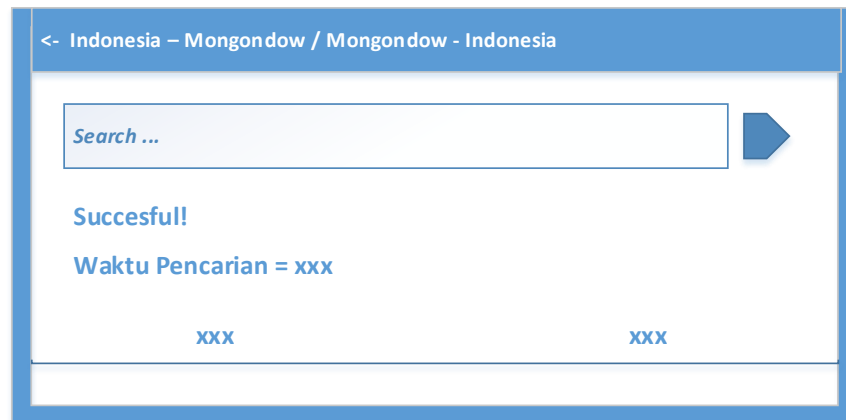
Kata Bahasa Indonesia

Kata Bahasa Mongondow

Ubah Hapus

**Gambar 4.14** Mekanisme Detail, Ubah dan Hapus data Kamus.

#### 4.12.6 Mekanisme Output



**Gambar 4.15** Mekanisme Output Kamus.

#### 4.13 Data Desain

##### 4.13.1 Struktur Data

**Tabel 4.3** Tabel kamus.

Nama File	: kamus			
Primary key	: id			
Media	: hardisk			
fungsi	: menyimpan data kamus			
struktur data	:			
No	Field Name	Type	Width	Keterangan
1.	Id	integer	5	Id kamus
2.	kata	Varchar	200	Data kata
3.	hasil	Varchar	200	Data hasil

#### 4.14 Hasil Pengujian Sistem

Hasil pengujian kecepatan pencarian kosa kata menggunakan Algoritma Knuth Morris Pratt yang dilakukan terhadap beberapa kosakata Indonesia-Mongondow atau Mongondow-Indonesia pada tiap kata yang berbeda. Maka hasilnya adalah sebagai berikut :

**Tabel 4.4** Kecepatan Pencarian dengan Algoritma KMP.

NO	Kata	Running Time(NanoTime)	Arti Kata	Kejadian
1	Saya	2.8183	Aku'oi	Kata ditemukan
2	Topi	10.9152	Sapoyo	Kata ditemukan
3	Sapu	3.0257	Tosimpat	Kata ditemukan
4	Kerja	2.9480	Aid	Kata ditemukan
5	Kamu	2.8655	Ikow	Kata ditemukan
6	Bibir	3.2229	Bibig	Kata ditemukan
7	Kaki	2.9197	Si'ol	Kata ditemukan
8	Baju	29.589	Lambung	Kata ditemukan
	<b>Total</b>	58.3043		
	<b>Rata-rata</b>	7,2880375		

Tabel di atas menunjukkan terdapat 8 kosakata dengan kecepatan menggunakan Algoritma *Knuth Morris Pratt*, pada masing-masing kosakata terdapat nilai kecepatan dalam satuan *Nano second* yang menunjukkan waktu pencarian kata. Data di atas menghasilkan kecepatan rata-rata 7,2880375 Nano atau sama dengan 0,07288 detik. Nilai kecepatan pencarian pada tabel di atas tidak selalu mendapatkan hasil yang sama, hasil ini hanya menunjukkan sebagaimana cepat cara kerja algoritma KMP ini memperoleh kata yang di cari.

#### 4.14.1 Pengujian *White Box*

<b>int</b> index = <i>NOT_FOUND_INDEX</i> .....	1
<b>int</b> patternLength = pattern.length(); .....	1
<b>int</b> textLength = text.length();.....	1
<b>char</b> [] textCharArray = text.toCharArray();.....	1
<b>char</b> [] patternArray = pattern.toCharArray(); .....	1
<b>int</b> [] suffixArray = <i>generateSuffixArray</i> ( patternLength, pattern); .....	1
<b>int</b> ® = 0;.....	1
<b>boolean</b> isMatched = <b>true</b> ; .....	1
<b>int</b> j = 0;.....	1
<b>while</b> (j < patternLength && ® < textLength){ .....	2
<b>if</b> (textCharArray[i] == patternArray[j]){ .....	3
j = j + 1;.....	4
® =® + 1; .....	4
<b>else</b> { .....	5
<b>int</b> newIndex = ® – j + 1;.....	5
isMatched = <b>false</b> ; .....	5
j = j – 1;.....	5
<b>if</b> (j < 0) .....	6
j = 0; .....	7
j = suffixArray[j]; .....	8
<b>if</b> (j==0) .....	9
® =newIndex; .....	10
<b>else</b> .....	11
® = ® + 1;.....	12
}	
<b>if</b> (j == patternLength) {.....	13
isMatched = <b>true</b> ;.....	14
}	

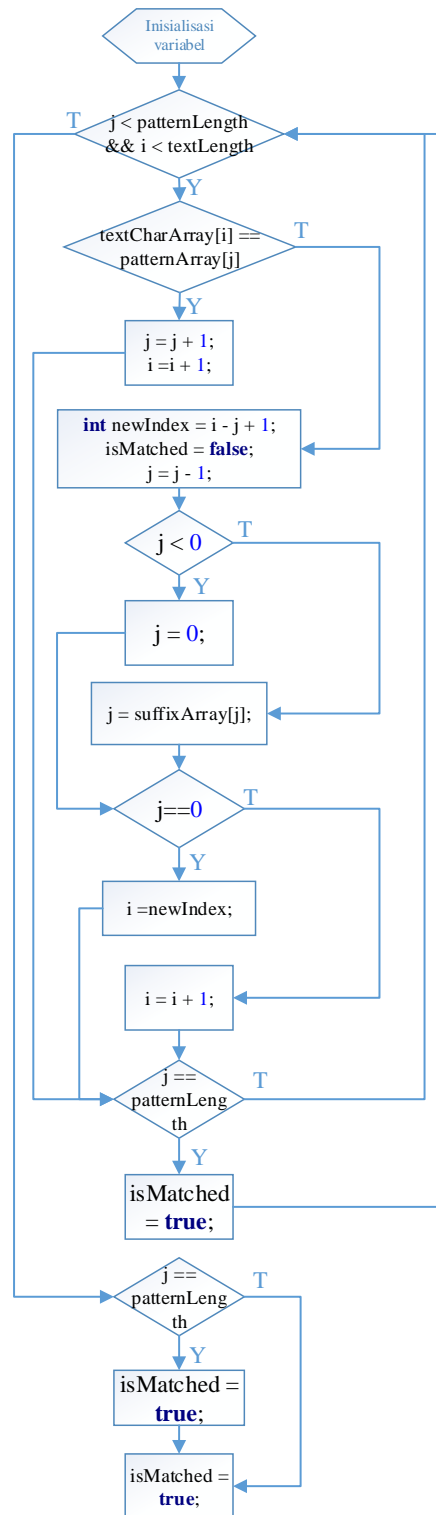
```
}
```

```
if(isMatched).....15
```

```
    index = ® - patternLength;.....16
```

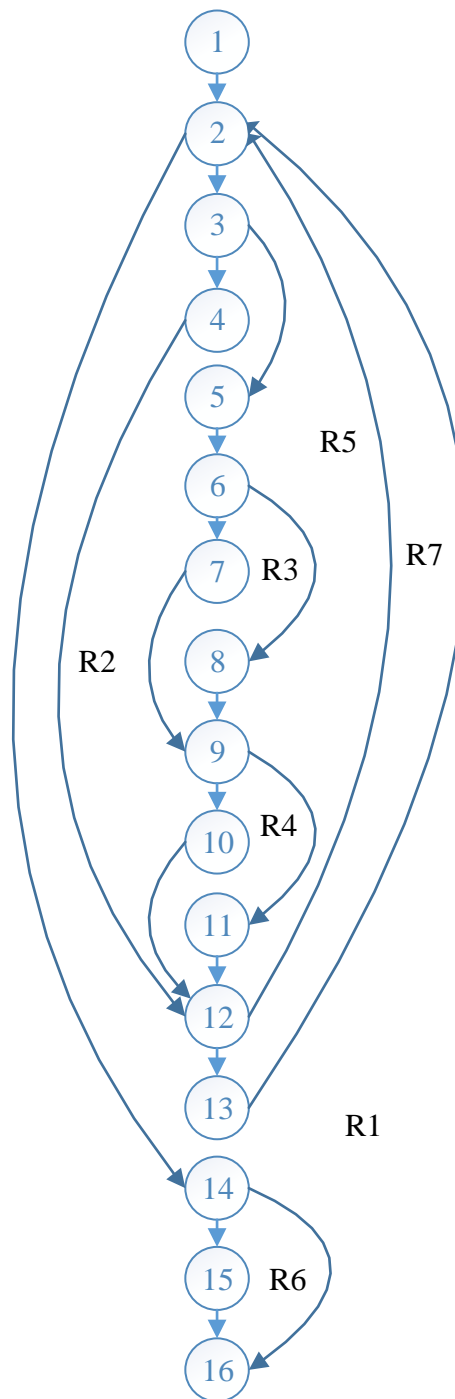
```
return index;.....17
```

## 4.14.2 Flowchart



Gambar 4.16 Flowchart Algoritma KMP.

### 4.14.3 Pengujian *White Box*



**Gambar 4.17** *Flowgraph* Algoritma KMP.

#### 4.14.4 Perhitungan CC pada pengujian *White Box*

Dari *Flowgraph* tersebut, didapatkan :

$$\text{Diketahui} \quad \text{Region}^{\text{R}} = 7$$

$$\text{Node}(N) = 16$$

$$\text{Edge}(E) = 22$$

$$\text{Predicate Node}(P) = 6$$

$$\text{Rumus: } V(G) = E - N + 2 \text{ dan } V(G) = P + 1$$

$$\text{Penyelesaian: } V(G) = 21 - 16 + 2 = 7$$

$$V(G) = 6 + 1 = 7$$

$$(R1, R2, R3, R4, R5, R6, R7)$$

#### 4.14.5 Perhitungan CC pada pengujian *White Box*

Tabel 4.5 Basis *Path*.

NO	PATH	KET
1.	1-2-14-15-16	OK
2.	1-2-3-4-12-2-...	OK
3.	1-2-3-5-6-8-9-10-12-2-...	OK
4.	1-2-3-5-6-7-9-11-12-2-...	OK
5.	1-2-3-5-6-8-9-10-12-2-...	OK
6.	1-2-14-15-16	OK
7.	1-2-3-5-6-8-9-11-12-13-2-...	OK

Ketika aplikasi dijalankan, maka terlihat bahwa semua basis path yang dihasilkan telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem ini telah memenuhi syarat.



#### 4.14.6 Pengujian *Black Box*

Pengujian *Black Box* dilakukan untuk memastikan bahwa suatu *event* atau masukan akan menjalankan proses yang tepat dan menghasilkan *Output* Sesuai dengan rancangan. Untuk Contoh pengujian terhadap beberapa proses memberikan hasil sebagai berikut :

**Tabel 4.6** Tabel Pengujian *Black Box*

<b>Input/Event</b>	<b>Fungsi</b>	<b>Hasil Yang Diharapkan</b>	<b>Hasil Uji</b>
Klik Menu login	Login ke halaman admin	Tampil form login	Sesuai
Masukkan Nama User salah, Klik Login	Melakukan validasi nama user	Tampil pesan “Nama User atau Password Salah !”	Sesuai
Masukkan Password Salah, klik Login	Melakukan validasi password	Tampil pesan “Nama User atau Password Salah !”	Sesuai
Masukkan nama user dan password benar, klik login	Melakukan validasi nama user dan password	Tampil halaman Admin	Sesuai
Klik salah satu kata di kamus	Menampilkan detail data kamus	Tampil Detail Kamus – Ubah / Hapus	Sesuai
Masukkan data kamus yang baru, klik Ubah	Mengubah data kamus	Tampil pesan “Data Berhasil Diubah !”	Sesuai
Klik Hapus pada detail Kamus	Menghapus data Kamus	Tampil pesan “Data berhasil dihapus !”	Sesuai
Klik tanda +	Menambah data Kamus	Tampil form Tambah data kamus	Sesuai

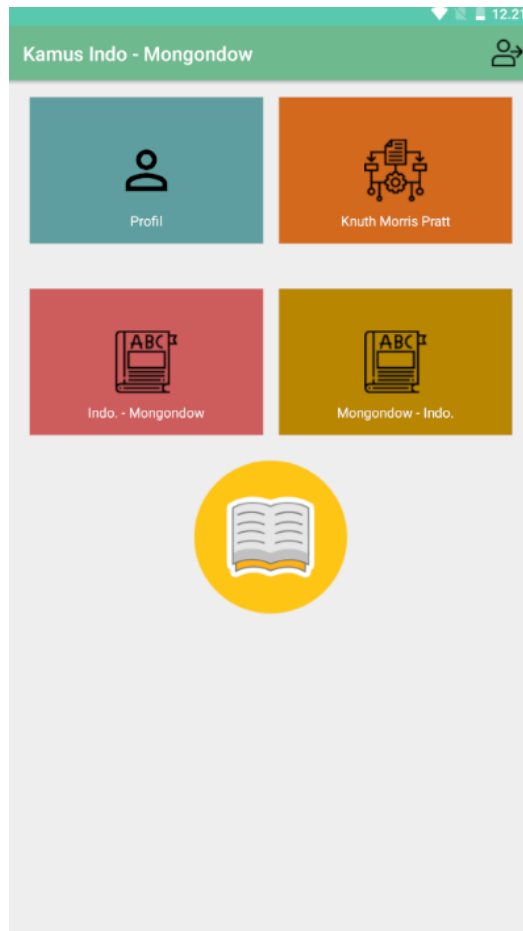
<b>Input/Event</b>	<b>Fungsi</b>	<b>Hasil Yang Diharapkan</b>	<b>Hasil Uji</b>
Masukkan data kamus, klik Simpan	Menambahkan data kamus	Tampil pesan “ Data berhasil disimpan !”	Sesuai
Masukkan kata yang akan dicari	Mencari kata	Jika ditemukan data ditampilkan, jika tidak ditemukan data tidak ditampilkan	Sesuai
Klik Logout	Keluar dari halaman Admin	Tampil Halaman User	Sesuai
Klik Profil	Menampilkan data profil	Tampil Profile Developer	Sesuai
Klik Knuth Morris Pratt	Menampilkan data KMP	Tampil Kotak dialog, tentang Algoritma KMP	Sesuai
Klik Indo – Mongondow	Mencari kata indo – mongondow	Tampil pencarian kata	Sesuai
Klik Mongondow – indo	Mencari kata mongondow – indo	Tampil pencarian kata	Sesuai

## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1 Pembahasan Sistem

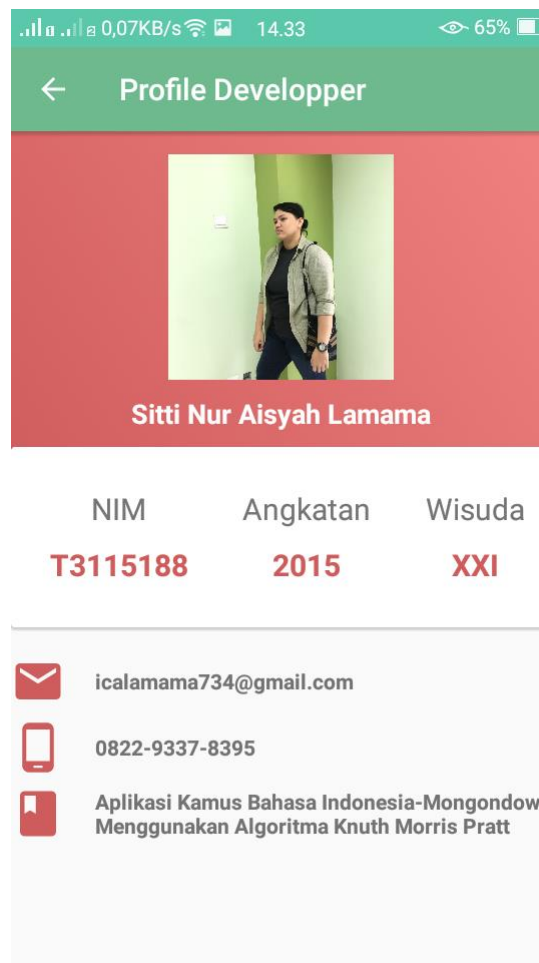
##### 5.1.1 Tampilan Halaman Home



**Gambar 5.1**Tampilan *Home* User.

Halaman ini merupakan halaman utama dari sisi User, halaman ini dapat menampilkan data Profil, data Algoritma KMP, Kamus Indo- Mongondow, dan Kamus Mongondow – Indo.

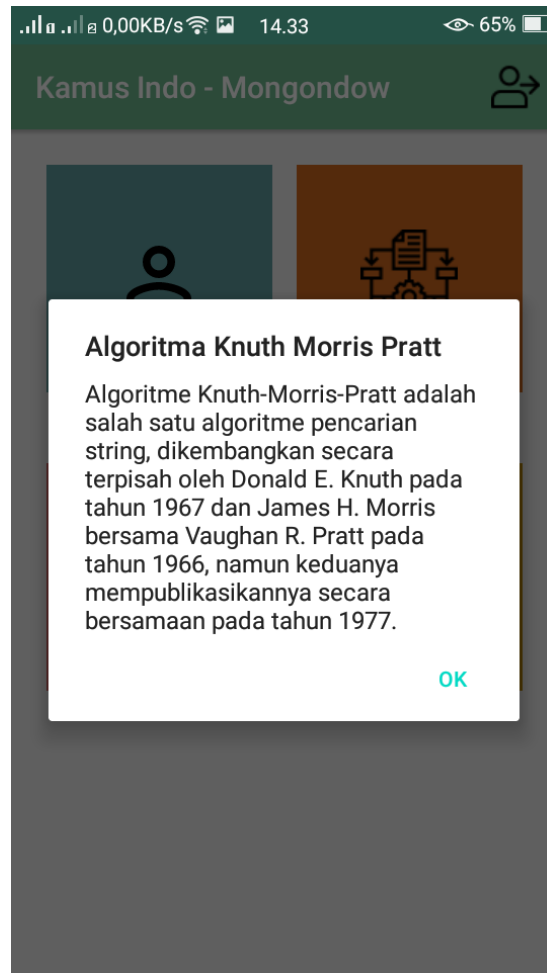
### 5.1.2 Tampilan Halaman Data Profil



**Gambar 5.2** Halaman data Profil.

Halaman ini menampilkan data Profile Developer atau pembuat Sistem, terdiri dari Nama Profil, NIM, Angkatan, Wisuda, Email, No telp, dan Judul Penelitian

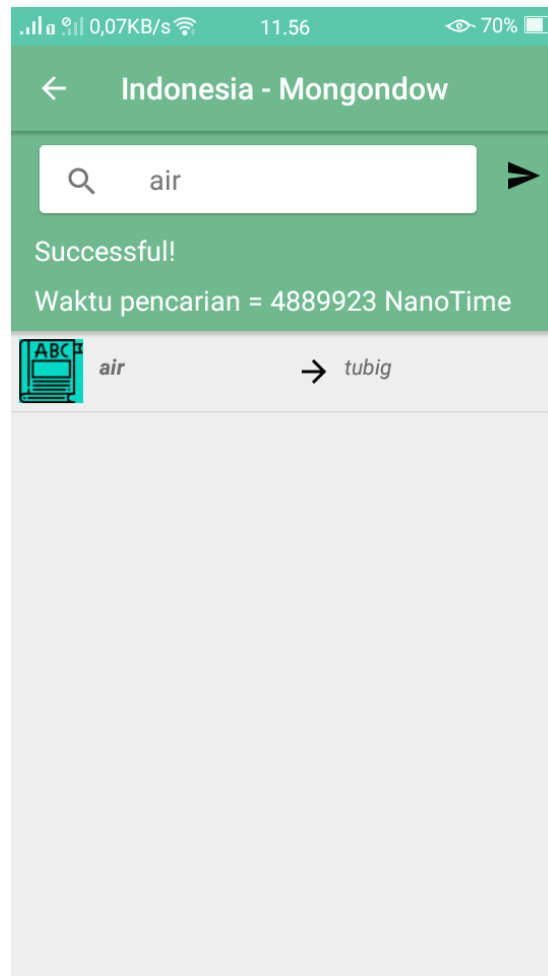
### 5.1.3 Tampilan Halaman Knuth Morris Pratt



**Gambar 5.3** Halaman Knuth Morris Pratt.

Halaman ini menampilkan informasi tentang Algoritma Knuth Morris Pratt. Untuk kembali ke Halaman Utama User, klik OK

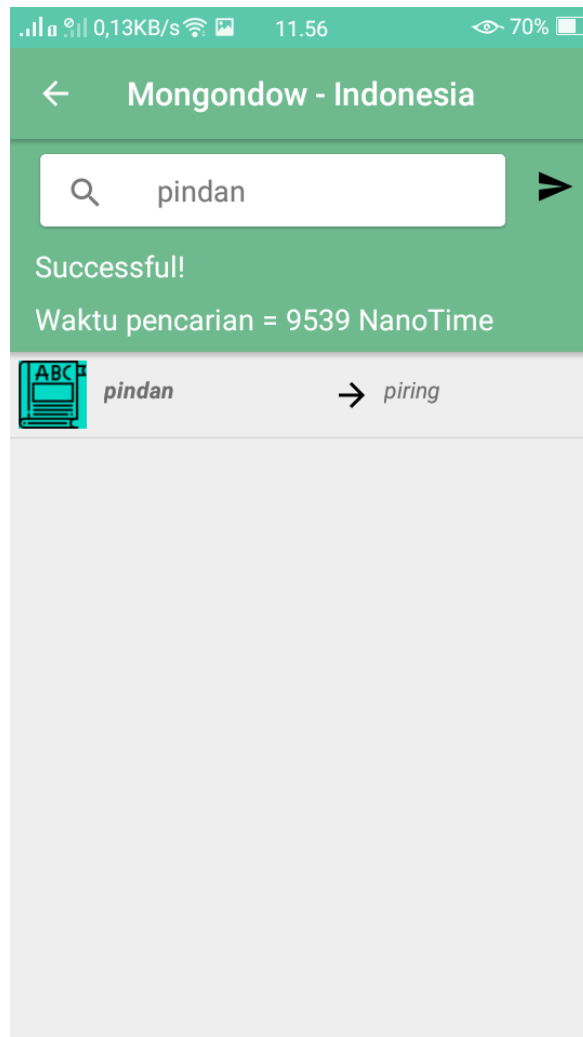
#### 5.1.4 Tampilan Halaman Kamus Indonesia - Mongondow



**Gambar 5.4** Halaman Kamus Indonesia – Mongondow.

Tampilan ini digunakan untuk mencari data Kamus Indonesia – Mongondow, dimulai dengan memasukkan kata yang akan dicari, klik cari > Kata yang di cari akan ditampilkan apabila kata ditemukan.

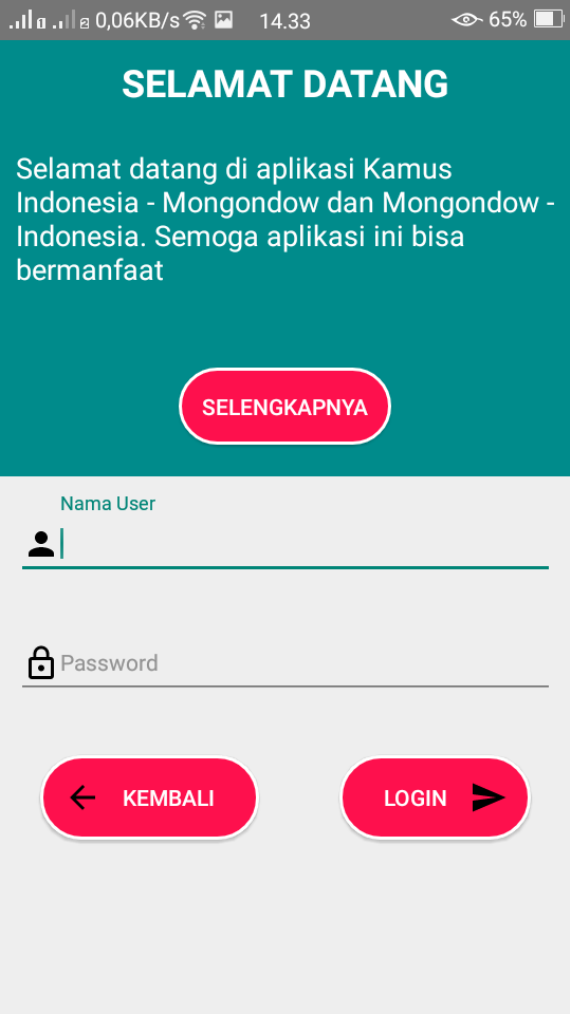
### 5.1.5 Tampilan Halaman Kamus Mongondow - Indonesia



**Gambar 5.5** Halaman Kamus Mongondow – Indonesia.

Tampilan ini digunakan untuk mencari data Kamus Mongondow - Indonesia, dimulai dengan memasukkan kata yang akan dicari, klik cari > Kata yang di cari akan ditampilkan apabila kata ditemukan.

### 5.1.6 Tampilan Halaman Data Login



..ll 0,06KB/s 14.33 65%

## SELAMAT DATANG

Selamat datang di aplikasi Kamus Indonesia - Mongondow dan Mongondow - Indonesia. Semoga aplikasi ini bisa bermanfaat

SELENGKAPNYA

Nama User

Password

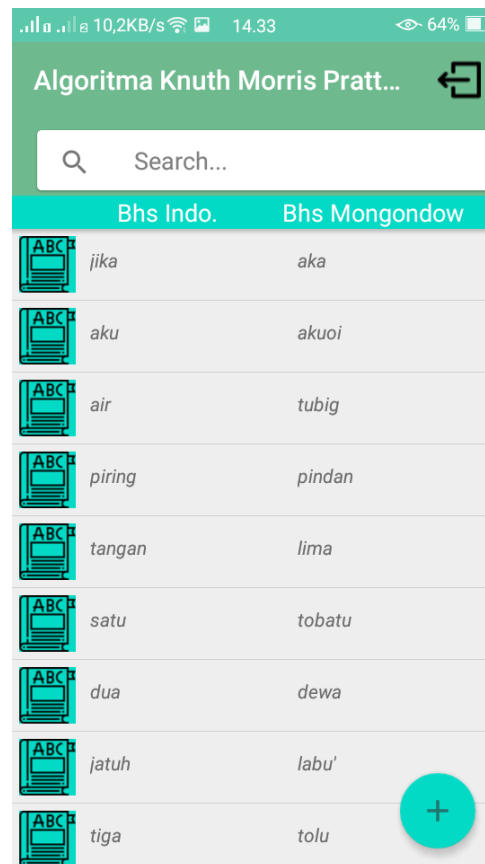
← KEMBALI LOGIN →

**Gambar 5.6** Halaman Data Login.

Halaman ini digunakan untuk masuk ke halaman Admin, dimulai dengan memasukkan Nama User dan Password, Klik Login untuk melanjutkan proses Login, dan klik Kembali untuk kembali ke Halaman Utama User.



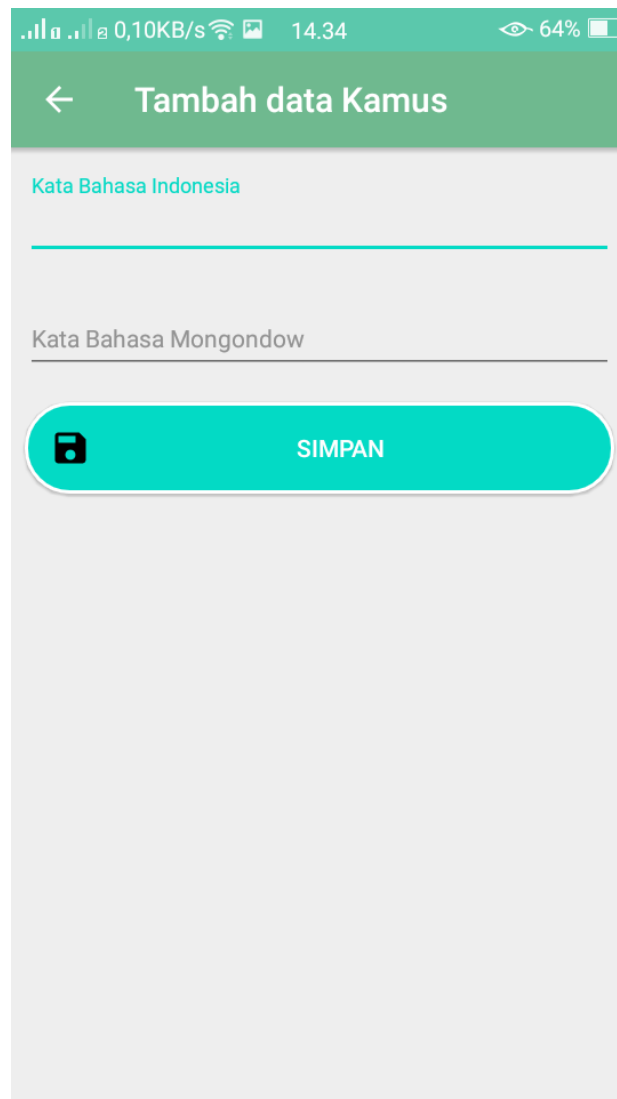
### 5.1.7 Tampilan Halaman Utama Admin



**Gambar 5.7** Halaman Utama Admin.

Tampilan ini merupakan tampilan Utama Admin, data yang ditampilkan yaitu data Kamus yang terdiri dari Bahasa Indonesia dan Bahasa Mongondow. Untuk menambahkan data Kamus klik tanda +. Untuk mencari data kamus, masukkan kata yang akan dicari. Untuk melihat detail data kamus, klik salah satu kata di kamus.

### 5.1.8 Tampilan Halaman Tambah Data Kamus

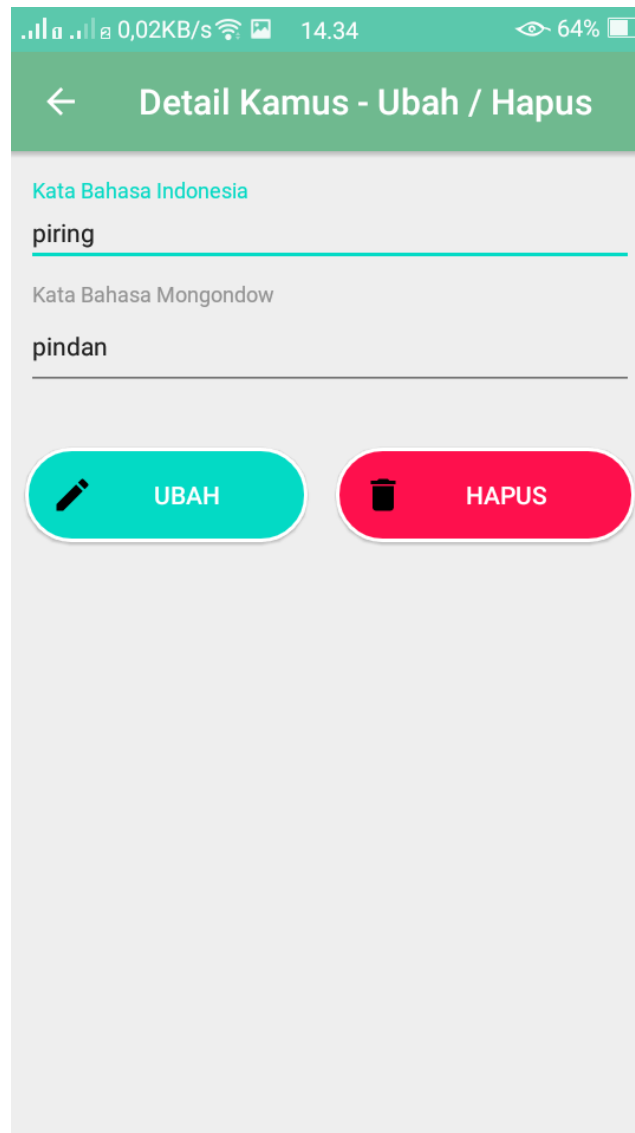


The screenshot shows a mobile application interface for adding dictionary data. The top status bar displays network speed (0,10KB/s), time (14.34), and battery level (64%). The app's header is green with a white back arrow and the text 'Tambah data Kamus'. Below the header, there are two input fields: 'Kata Bahasa Indonesia' and 'Kata Bahasa Mongondow'. At the bottom, there is a large green button with a floppy disk icon and the text 'SIMPAN'.

**Gambar 5.8** Halaman Tambah Data Kamus.

Halaman ini dilakukan untuk menambahkan data Kamus, dimulai dengan memasukkan kata Bahasa Indonesia dan Kata Bahasa Mongondow. Klik Simpan untuk melanjutkan proses penyimpanan data.

### 5.1.9 Tampilan Halaman Detail Kamus



**Gambar 5.9** Halaman Detail Kamus.

Halaman ini menampilkan Detail data Kamus. Detail data yang ditampilkan yaitu Kata Bahasa Indonesia, Kata Bahasa Mongondow. Untuk melakukan perubahan klik Ubah, dan untuk menghapus data klik Hapus.

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Berdasarkan hasil penelitian yang telah dilakukan dan pembahasan yang telah diuraikan sebelumnya, maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Aplikasi Kamus Bahasa Mongondow-Indonesia dapat digunakan oleh masyarakat Bolaang Mongondow sebagai media yang efisien dan efektif untuk menerjemahkan kata bahasa Mongondow ke Indonesia.
2. Algoritma *Knuth Morris Pratt* pada aplikasi Kamus Bahasa Mongondow-Indonesia dapat memangkas waktu lebih cepat dalam menerjemahkan kata, sehingga tidak memakan waktu lebih banyak. Dapat dilihat dari hasil pencarian kosa kata menggunakan Algoritma KMP. Terdapat dalam tabel 4.4 pencarian terhadap beberapa kosa kata menghasilkan kecepatan rata-rata yaitu 7,2880375 Nano atau sama dengan 0,07288 detik.

#### **6.2 Saran**

Dari penelitian ini, maka dianggap perlu adanya saran yang penulis sampaikan antara lain :

1. Aplikasi ini dapat disempurnakan dengan penambahan kata-kata yang lebih banyak lagi.
2. Aplikasi ini hanya bisa digunakan di platform Android. Baiknya dikembangkan menjadi aplikasi yang bisa digunakan di semua platform.

## DAFTAR PUSTAKA

- [1] saad mokoagow, *Kamus Bahasa Mongondow Indonesia*. Publish Yourself, 2015.
- [2] R. K. Hondro, Z. A. Hsb, and R. D. Sianturi, “Aplikasi Penerjemahan Bahasa Mandailing-Indonesia,” *JURIKOM (Jurnal Ris. Komputer)*, vol. 3, no. 4, pp. 49–53, 2016.
- [3] H. T. Sigit and K. Anwar, “Aplikasi Android Kamus Bahasa Jawa Serang – Indonesia Menggunakan Algoritma Knutt Morris Pratt,” *J. PROTEKINFO*, vol. 2, no. September, pp. 30–34, 2015.
- [4] “Etnis dan Bahasa Asli di Bolaang Mongondow,” 2012. [Online]. Available: <http://lipukobayagan.blogspot.com/2012/11/etnis-dan-bahasa-asli-di-bolaang.html>. [Accessed: 12-Mar-2019].
- [5] N. Nursobah, P. P.- Sebatik, and U. 2019, “Penerapan Algoritma Pencarian Knuth-Morris-Pratt (Kmp) Dalam Sistem Informasi Perpustakaan Smk Ti Pratama,” *Jurnal. Wicida.Ac.Id*, pp. 112–115, 2018.
- [6] Fajar Sani, “Algoritma Knuth Morris Pratt,” 2014. [Online]. Available: [https://www.slideshare.net/Fajar\\_Sany/algoritma-knuth-morrispratt](https://www.slideshare.net/Fajar_Sany/algoritma-knuth-morrispratt). [Accessed: 13-Mar-2019].
- [7] Firly Nadia, *Create Your Own Android Application*. Jakarta: PT. Elex Media Komputindo, 2018.
- [8] A. Hendini, “Pemodelan UML sistem informasi Monitoring Penjualan dan stok barang,” *Pemodelan Uml Sist. Inf. Monit. Penjualan Dan Stok Barang (Studi Kasus Distro Zhezha Pontianak)*, vol. IV, no. 2, pp. 107–116, 2016.
- [9] C. R. Dewati, I. Aknuranda, W. Hayuhardhika, and N. Putra, “Analisis dan Perancangan Sistem Informasi Pengarsipan Dokumen Dengan Pendekatan Berorientasi Objek ( Studi Kasus : Dinas Sekretariat Dewan , Pemerintahan Kota Batu ),” vol. 3, no. 5, pp. 5140–5146, 2019.
- [10] A. Juansyah, “Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A-GPS ) Dengan Platform Android,” *J. Ilm. Komput. dan Inform.*, vol. 1, no. 1, pp. 1–8, 2015.
- [11] Ryan Yulianto, “Realm Database,” 2016. [Online]. Available: <https://www.codepolitan.com/realms-database-versi-200-telah-dirilis-dengan-fitur-sinkronisasi-online-57eb19ce804ae-346>.
- [12] Elliana Gautama, “Kompleksitas Algoritma,” 2017. [Online]. Available: <https://dosen.perbanas.id/kompleksitas-algoritma/#:~:text=Ada dua macam kompleksitas algoritma,fungsi dari ukuran masukan n.> [Accessed: 03-Dec-2020].
- [13] R. Syartika, “Testing dan Implementasi Sistem White Box Testing,” 2019.
- [14] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, “( Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN ),” *Penguji. Apl. Menggunakan Black Box Test. Bound. Value Anal. (Studi Kasus Apl. Prediksi Kelulusan SNMPTN)*, vol. I, no. 3, p. 34, 2015.
- [15] A. A. Asmaul Husna Nasrullah, Amiruddin, Azminuddin Azis, Budy Santoso, *Pedoman Penelitian Ilmu Komputer*. Gorontalo, 2018.

## LAMPIRAN 1 KODE PROGRAM

- *Algoritma Knuth Morris Pratt*

```

public class kmp {
    public static final int NOT_FOUND_INDEX = -1;
    //algoritma KMP
    public static int findPatternInText(String pattern,
String text) {
        int index = NOT_FOUND_INDEX;
        int patternLength = pattern.length();
        int textLength = text.length();
        char[] textCharArray = text.toCharArray();
        char[] patternArray = pattern.toCharArray();

        int[] suffixArray =
generateSuffixArray(patternLength, pattern);

        int i = 0;
        boolean isMatched = true;
        int j = 0;
        while(j < patternLength && i < textLength){
            if(textCharArray[i] == patternArray[j]){
                j = j + 1;
                i =i + 1;
            } else{
                int newIndex = i - j + 1;
                isMatched = false;
                j = j - 1;
                if (j < 0)
                    j = 0;
                j = suffixArray[j];
                if(j==0)
                    i =newIndex;
                else
                    i = i + 1;
            }

            if(j == patternLength) {
                isMatched = true;
            }

        }
        if(isMatched)
            index = i - patternLength;
        return index;
    }
}

```

```
public static int[] generateSuffixArray(int
patternLength, String pattern) {
    int[] suffixArray = new int[patternLength];
    char[] patternArray = pattern.toCharArray();
    int i = 1;
    while(i < patternLength){
        for(int j = 0; j < patternLength; j++){
            if((i < patternLength) &&
(patternArray[i] == patternArray[j])){
                suffixArray[i] = j + 1;
                i = i + 1;

            } else{
                i = i + 1;
                break;
            }

            if(i == patternLength - 1)
                break;
        }
    }
    return suffixArray;
}
```

## LAMPIRAN 2 DAFTAR RIWAYAT HIDUP PENELITI

Nama : Sitti Nur Aisyah Lamama  
Nim : T3115188  
Tempat, Tanggal Lahir : Kotamobagu, 29 April 1997  
Agama : Islam  
Email : icalamama734@gmail.com



### Riwayat Pendidikan :

1. Tahun 2009, Menyelesaikan Pendidikan di Sekolah Dasar Negeri 1 Mogolaing, Kecamatan Kotamobagu Barat, Kota Kotamobagu, Provinsi Sulawesi Utara.
2. Tahun 2012, Menyelesaikan Pendidikan di Sekolah Menengah Pertama Negeri 4 Kotamobagu, Kecamatan Kotamobagu Barat, Kota Kotamobagu, Provinsi Sulawesi Utara.
3. Tahun 2015, Menyelesaikan Pendidikan di Sekolah Menengah Kejuruan Negeri 1 Kotamobagu, Kecamatan Kotamobagu Barat, Kota Kotamobagu, Provinsi Sulawesi Utara
4. Tahun 2015, Telah di terima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.



## LAMPIRAN 3 REKOMENDASI PENELITIAN



PEMERINTAH KOTA KOTAMOBAGU  
PENDIDIKAN DINAS KOTA KOTAMOBAGU  
SD NEGERI 4 MOTOBOI KECIL

*Jln. Molayak, Desa Motoboi Kecil, Kec. Kotamobagu Selatan, Kota Kotamobagu*



### SURAT KETERANGAN NO. 58/C.1/SDN4MK/XII/2019

Yang Bertandatangan di bawah ini :

Nama : JONO KOIKIT  
NIP : 19680720 201407 1 002  
Pangkat/Golongan : Pengatur/2B  
Jabatan : Kepala Sekolah

Dengan ini menerangkan bahwa :


Nama : Sitti Nur Aisyah Lamama  
NIM : T3115188

Benar – benar telah melaksanakan penelitian di SD Negeri 4 Motoboi Kecil dalam rangka Penyusunan Skripsi dengan Judul “**Aplikasi Kamus Bahasa Mongondow – Indonesia Menggunakan Algoritma Knuth Morris Pratt**”.

Demikian Surat Keterangan ini dibuat untuk seperlunya.

Motoboi Kecil, 23 Desember 2019

Kepala Sekolah SD Negeri 4 Motoboi Kecil

  
JONO KOIKIT  
NIP.19680720 201407 1 002

## LAMPIRAN 4 BUKTI PENERIMAAN SOFTCOPY SKRIPSI



Pustikom  
Universitas Ichsan Gorontalo

### BUKTI PENERIMAAN SOFTCOPY SKRIPSI PENGECEKAN SIMILARITY TURNITIN

Nama Mahasiswa : SITI NUR AISYAH LAMAMA  
 NIM : T3115188  
 Program Studi : Teknik Informatika (S1)  
 Fakultas : Fakultas Ilmu Komputer  
 Judul Skripsi : Aplikasi Kamus Bahasa Mongondow-Indonesia Menggunakan Algoritma Knuth Mori Pratt

Nama File (Pdf) : skripsi\_T3115188\_SITINUR AISYAH LAMAMA.docx

No. HP/WA : 082293378395

e-Mail : lanlamama939@gmail.com

Tgl. Terima : 

20					
----	--	--	--	--	--

Hasil Pengecekan : 

31 %					
------	--	--	--	--	--

Diterima/Diperiksa Oleh,

  
**Sudirman S. Panna, M.Kom**  
 085340910768

## LAMPIRAN 5 SURAT REKOMENDASI BEBAS PLAGIASI



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS ICHSAN  
(UNISAN) GORONTALO**

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001  
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

### SURAT REKOMENDASI BEBAS PLAGIASI

No. 0641/UNISAN-G/S-BP/XI/2020

Yang bertanda tangan di bawah ini:

Nama : Sunarto Taliki, M.Kom  
NIDN : 0906058301  
Unit Kerja : Pustikom, Universitas Ichsan Gorontalo

Dengan ini Menyatakan bahwa :

Nama Mahasisw : SITI NUR AISYAH LAMAMA  
NIM : T3115188  
Program Studi : Teknik Informatika (S1)  
Fakultas : Fakultas Ilmu Komputer  
Judul Skripsi : Aplikasi Kamus Bahasa Mongondow-Indonesia Menggunakan Algoritma Knuth Morri Pratt

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 31%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/IX/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 28 November 2020

Tim Verifikasi,



**Sunarto Taliki, M.Kom**

NIDN. 0906058301

Tembusan :

1. Dekan
2. Ketua Program Studi
3. Pembimbing I dan Pembimbing II
4. Yang bersangkutan
5. Arsip

## LAMPIRAN 6 HASIL TURNITIN



SKRIPSI\_1\_T3115188\_SITTI NUR AISYAH LAMAMA.docx

Nov 26, 2020

6859 words / 41886 characters

T3115188 SITTI NUR AISYAH LAMAMA

APLIKASI KAMUS BAHASA MONGONDOW-INDONESIA MENGG...

Sources Overview

**31%**

OVERALL SIMILARITY

1	www.coursehero.com INTERNET	4%
2	eprints.walisongo.ac.id INTERNET	3%
3	www.scribd.com INTERNET	2%
4	eprints.umk.ac.id INTERNET	2%
5	www.slideshare.net INTERNET	2%
6	id.123dok.com INTERNET	2%
7	e-jurnal.ppmunswk.org INTERNET	2%
8	agusan.files.wordpress.com INTERNET	2%
9	smk.lembahdempo.ac.id INTERNET	1%
10	repository.upi.edu INTERNET	1%
11	pdfs.semanticscholar.org INTERNET	1%
12	prosiding-snatikomf.stb-maden.ac.id INTERNET	1%
13	e-jurnal.stmik-budidarma.ac.id INTERNET	<1%
14	edoc.pub INTERNET	<1%
15	nulpointexception.com INTERNET	<1%
16	id.wikipedia.org INTERNET	<1%

17	docplayer.info	INTERNET	<1%
18	123dok.com	INTERNET	<1%
19	repository.unimal.ac.id	INTERNET	<1%
20	titonkadit.blogspot.com	INTERNET	<1%
21	jurnal.stmikalahma.ac.id	INTERNET	<1%
22	text4d.123dok.com	INTERNET	<1%
23	wa.scribd.com	INTERNET	<1%
24	pt.scribd.com	INTERNET	<1%
25	Agustian Noor. "Lab IT Security System Dengan QR Code Bebas Web Menggunakan Microcontroller Arduino", Jurnal Sains dan Infor...	CROSSREF	<1%
26	Body Santoso, Azminuddin I. S. Azis, Andi Bode. "Pengendalian Lampu Lalu Lintas Cerdas di Perempatan Empat Ruas yang Komple...	CROSSREF	<1%
27	ejournal.catursekti.ac.id	INTERNET	<1%
28	eprints.unianu.ac.id	INTERNET	<1%

**Excluded search repositories:**

- Submitted Works

**Excluded from Similarity Report:**

- Small Matches (less than 25 words).

**Excluded sources:**

- None