

## ALGORITMA BASTAL: ADAPTASI ALGORITMA NAZIEF & ADRIANI UNTUK STEMMING TEKS BAHASA BALI

I Putu Merta Wirayasa<sup>1</sup>,  
I Made Agus Wirawan<sup>2</sup>, I Made Ardwi Pradnyana<sup>3</sup>

<sup>1,2,3</sup>Jurusan Pendidikan Teknik Informatika  
Universitas Pendidikan Ganesha  
Singaraja, Bali

e-mail: merta.wirayasa@gmail.com<sup>1</sup>, imade.aguswirawan@undiksha.ac.id<sup>2</sup>,  
ardwi.pradnyana@undiksha.ac.id<sup>3</sup>

### Abstrak

Bahasa Bali merupakan salah satu identitas kebudayaan masyarakat Bali. Pengaruh perkembangan globalisasi membuat Bahasa Bali semakin terpinggirkan dan seolah-olah menjadi bahasa “asing” bagi manusia Bali sendiri. Bahasa Bali sebagai bahasa ibu wajib dilestarikan, salah satu upayanya adalah dengan mengembangkan aplikasi penerjemah bahasa Bali. Penerjemahan teks bahasa Bali membutuhkan sebuah algoritma *stemming*. Pada penelitian ini, dikembangkan sebuah algoritma *stemming* untuk menentukan kata dasar pada teks bahasa Bali yang diberi nama algoritma Bastal. Algoritma Bastal merupakan adaptasi dari algoritma Nazief & Adriani. Metode penelitian yang digunakan adalah penelitian dan pengembangan dengan model *waterfall*. Algoritma Bastal berhasil diimplementasikan pada aplikasi penerjemah bahasa Bali yang dapat dipasang pada *smartphone* Android. Berdasarkan hasil pengujian, aplikasi mampu bekerja dengan baik. Aplikasi dinilai sangat akurat dengan persentase keseluruhan penilaian yaitu 96,15%. Aplikasi juga mendapatkan respon positif dari pengguna.

**Kata kunci:** *Stemming*, Bahasa Bali, Algoritma Bastal

### Abstract

*The Balinese language is one of Bali's cultural identity. Globalization makes Balinese languages marginalized and becomes a "foreign" language for the Balinese themselves. The Balinese language must be preserved. One of its efforts is to develop the application of Balinese language translator. The translation of Balinese text requires a stemming algorithm. In this research, a Balinese stemming algorithm was developed. The algorithm named Bastal algorithm. Bastal Algorithm used to determine the root word in Balinese text. Bastal algorithm is an adaptation of the Nazief & Adriani algorithm. The research method used is research and development with waterfall model. Bastal algorithm successfully implemented in application of Balinese translator that can be install on Android smartphones. Based on the test results, the application works well. Applications are rated very accurately with 96.15% percentage. Applications also get a positive response from users.*

**Keywords:** *Stemming, Balinese Language, Bastal Algorithm*

### PENDAHULUAN

Suatu sistem temu balik informasi (*Information Retrieval*) dikatakan ideal jika sistem tersebut dapat menemukan seluruh

dokumen yang relevan dan sistem hanya menemukan dokumen yang relevan saja. Banyaknya varian morfologik pada suatu bahasa sangat berpengaruh terhadap hal

tersebut, untuk itu setiap kata harus diubah kedalam bentuk dasarnya atau yang disebut dengan *stemming* untuk mengurangi perbedaan bentuk dari suatu kata, sehingga proses sistem temu kembali menjadi efektif [1].

*Stemming* adalah proses pemotongan imbuhan (*affix*), baik *prefix* maupun *suffix* dari sebuah *term* untuk mendapatkan kata dasar (*root* atau *stem*) dari kata berimbuhan. Beberapa algoritma *stemming* untuk Bahasa Indonesia telah dikembangkan sebelumnya, diantaranya algoritma Nazief & Adriani [2]. Algoritma Nazief & Adriani dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan menjadi awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*) dan gabungan awalan akhiran (*confixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih. Kamus kata dasar tersebut dibutuhkan untuk memeriksa apakah kata dasar yang melalui proses *stemming* benar dan ditemukan pada kamus saat proses *stemming* dilakukan.

Bahasa Bali memiliki morfologi yang berbeda dengan Bahasa Indonesia atau bahasa negara lain dalam hal penyusunan kata. Proses penyusunan kata dalam bahasa Bali pada proses penyusunannya akan menggunakan *pangater* (awalan), *seselan* (sisipan) dan *pangiring* (akhiran). Jika sebuah kata dasar dalam bahasa Bali sudah mendapatkan awalan, sisipan, akhiran atau kombinasinya maka kata tersebut disebut *kruna tiron* (kata berimbuhan). Morfologi bahasa Bali yang berbeda dengan Bahasa Indonesia memiliki keunikan dan kesulitan tersendiri dalam proses *stemming*-nya. Perbedaan pada awalan, sisipan dan akhiran dalam bahasa Bali membuat tingkat kesulitan yang semakin kompleks dalam proses pembuatan *stemmer* bahasa Bali.

Berdasarkan paparan tersebut, penulis termotivasi untuk mengembangkan sebuah algoritma *stemming* bahasa Bali bernama algoritma Bastal. Algoritma ini dikembangkan dengan mengadaptasi

algoritma Nazief & Adriani dan dibuat berdasarkan aturan morfologi bahasa Bali. Algoritma Nazief & Adriani dipilih karena algoritma tersebut menggunakan kamus kata dasar sebagai acuan dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih.

## KAJIAN TEORI

### A. Pengolahan Bahasa Alami (*Natural Language Processing*)

Tujuan dalam bidang *Natural Language* ini adalah melakukan proses pembuatan model komputasi dari bahasa, sehingga dapat terjadi suatu interaksi antara manusia dengan komputer dengan perantara bahasa alami. Pada prinsipnya Pemrosesan bahasa alami (*Natural language Proccessing*) adalah kemampuan suatu komputer untuk memproses bahasa, baik lisan maupun tulisan yang digunakan oleh manusia dalam percakapan sehari-hari.

Pengolahan Bahasa Alami tidak bertujuan untuk mentransformasikan bahasa yang diterima dalam bentuk suara menjadi data digital dan/atau sebaliknya, melainkan bertujuan untuk memahami arti dari teks/tulisan yang diberikan dalam format bahasa alami dan memberikan respon yang sesuai, misalnya dengan melakukan suatu aksi tertentu atau menampilkan data tertentu [3].

Sebuah *Natural Language System* harus memperhatikan pengetahuan terhadap bahasa itu sendiri, baik dari segi kata yang digunakan, bagaimana kata-kata tersebut digabung untuk menghasilkan suatu kalimat, apa arti sebuah kata, apa fungsi sebuah kata dalam sebuah kalimat dan sebagainya. Bahasa dapat dibedakan menjadi 2: Bahasa Alami dan Bahasa Buatan. Bahasa alami adalah bahasa yang biasa digunakan untuk berkomunikasi antar manusia, misalnya bahasa Indonesia, Sunda, Jawa, Inggris, Jepang, dan sebagainya. Bahasa buatan adalah bahasa yang dibuat secara khusus untuk memenuhi kebutuhan tertentu, misalnya bahasa pemodelan atau bahasa pemrograman komputer.

## 1. Bahasa Alami

Bahasa alami merupakan fenomena yang kompleks, melibatkan berbagai proses, misal: pengenalan suara atau huruf tercetak, *parsing* sintak, inferensi semantik tingkat tinggi dan komunikasi dari isi emosional melalui ritme serta *inflection* (perubahan nada suara). Untuk mengatasi kompleksitas ini maka para ahli bahasa telah mendefinisikan tingkat analisis yang berbeda untuk bahasa alami [4]:

- a) *Prosody*, mengerjakan ritme & intonasi bahasa.
- b) *Phonology*, menguji suara yang dikombinasikan untuk membentuk bahasa.
- c) *Morphology*, mengenai komponen-komponen (morfem-morfem) yang membentuk kata-kata. Termasuk aturan-aturan pengembangan formasi kata-kata seperti efek dari prefix (un-, non-, anti-, dll), & sufiks (-ing, -ly, dll) yang memodifikasi arti dari akar kata.
- d) *Syntax*, pemahaman tentang urutan kata dalam pembentukan kalimat dan hubungan antar kata tersebut dalam proses perubahan bentuk dari kalimat menjadi bentuk yang sistematis. Selain itu dapat pula dikenali bagian-bagian kalimat dalam suatu kalimat yang lebih besar.
- e) *Semantics*, pemetaan bentuk struktur sintaksis dengan memanfaatkan tiap kata ke dalam bentuk yang lebih mendasar dan tidak tergantung struktur kalimat. Semantik mempelajari arti suatu kata dan bagaimana dari arti kata-kata tersebut membentuk suatu arti dari kalimat yang utuh.
- f) *Pragmatics*, merupakan studi cara dimana bahasa digunakan dan pengaruhnya pada pendengar.
- g) *World Knowledge*: mencakup arti sebuah kata secara umum dan apakah ada arti khusus bagi suatu kata dalam suatu percakapan dengan konteks tertentu.

## 2. Tahap Analisis Bahasa Alami

Secara umum, ada tiga tahapan untuk analisis pemahaman bahasa alami [4] yaitu:

- a) *Parsing* (penguraian) merupakan tindakan menganalisa struktur sintaks kalimat. *Parsing* tidak hanya membuktikan bahwa kalimat dapat dibentuk secara sintaksis, tetapi juga dapat menentukan struktur lingustiknya. Dengan mengidentifikasi hubungan-hubungan lingustik utamanya seperti *subjek-verb*, *verb-objek*, dan *noun-modifier*, maka penguraian (*parser*) memberikan suatu kerangka/bentuk diagram bagi interpretasi semantik dan hal ini sering kali digambarkan dalam pohon pengurai (*parse tree*). Penguraian menggunakan pengetahuan sintaks bahasa, morfologi dan beberapa semantik bahasa.
- b) Interpretasi semantik, pada tahap ini menghasilkan suatu representasi arti dari teks. Representasi lain umumnya digunakan, meliputi ketergantungan konseptual, bingkai dan representasi berbasis logika. Interpretasi semantik menggunakan pengetahuan tentang arti kata dan struktur linguistiknya, seperti peranan kata benda atau transitivitas kata kerja. Tahap ini juga melakukan pengujian konsistensi semantik.
- c) Struktur-struktur dari basis pengetahuan ditambahkan pada representasi kalimat untuk menghasilkan representasi perluasan dari arti kalimat. Struktur yang dihasilkan merepresentasikan makna dari teks bahasa alami dan digunakan oleh sistem untuk pemrosesan lebih jauh.

## 3. Proses Bahasa Alami

Pemrosesan bahasa alami dapat dibagi atas beberapa proses, yaitu:

- a) *Morphological analysis*, dimana kata secara individu dianalisis ke dalam komponen-komponennya dan yang bukan kata (seperti tanda baca) dipisahkan dari kata.
- b) *Syntactic analysis*, urutan linier dari kata ditransformasikan ke dalam

struktur yang menunjukkan bagaimana kata saling berhubungan. Urutan kata yang melanggar aturan bahasa dalam pengkombinasian kata, akan ditolak.

- c) *Semantic analysis*, dibuat *mapping* antara struktur sintaksis & objek dalam *domain* kerja. Struktur dimana tidak ada *mapping* yang memungkinkan, akan ditolak.
- d) *Discourse integration*, dimana arti dari kalimat secara individu bergantung pada kalimat-kalimat yang mendahului dan mungkin dapat mempengaruhi arti dari kalimat-kalimat yang mengikutinya.
- e) *Pragmatic analysis*, dimana struktur yang merepresentasikan apa yang diucapkan, diinterpretasikan kembali untuk menentukan apa arti sebenarnya.

#### 4. Aplikasi Dalam Bidang Bahasa Alami

Aplikasi yang dapat dibuat pada bidang bahasa alami (*Natural Language*) yaitu *text-based application* dan *dialogue-based application*. *Text-based application* mencakup segala macam aplikasi yang melakukan proses terhadap teks tertulis seperti buku, berita di surat kabar, dan lain sebagainya. Contoh penggunaan dari *text-based application* adalah:

- a) Mencari topik tertentu dari buku yang ada pada perpustakaan
- b) Mencari isi dari surat kabar atau *e-mail*
- c) Menerjemahkan dokumen dari satu bahasa ke bahasa yang lain

Bentuk berikutnya adalah bentuk *dialog-based application*. Idealnya pendekatan ini melibatkan bahasa lisan atau pengenalan suara, akan tetapi bidang ini juga memasukkan interaksi dengan cara memasukkan teks pertanyaan melalui keyboard. Aplikasi yang sering ditemui untuk bidang ini antara lain:

- a) Sistem tanya jawab, dimana *natural language* digunakan dalam mendapatkan informasi dari satu *database*.
- b) Sistem otomatis pelayanan melalui telpon.

- c) Contoh suara pada peralatan elektronik.
- d) Sistem *problem-solving* yang membantu untuk melakukan penyelesaian masalah yang umum dihadapi dalam suatu pekerjaan.

#### B. Rule Based Machine Translation

Penerjemah mesin (*machine translation*) atau biasa disingkat MT adalah cabang linguistik komputasional yang mempelajari penggunaan perangkat lunak komputer untuk menerjemahkan teks atau ucapan dari satu bahasa alami ke bahasa lainnya. Sekarang ini ada beberapa jenis MT, yang umumnya ditemui adalah *Statistical Machine Translator*, *Rule Base Machine Translator*, dan *Hybrid Machine Translator*.

*Rule Based Machine Translation* menggunakan aturan-aturan bahasa baku dalam menerjemahkan. Selain aturan-aturan, diperlukan juga data kamus untuk tiap kata dalam dua bahasa. Jadi tiap kata diterjemahkan satu persatu, kemudian diatur lagi berdasarkan aturan bahasa baku. Contoh aplikasi rule based ini adalah *rekso translator*. Namun karena aturan bahasa tidak selalu baku, dan data kamus terbatas, maka penerjemahan terkadang tidak sesuai dengan yang diharapkan.

#### C. Information Retrieval

*Information retrieval* pada dasarnya merupakan proses untuk menentukan dokumen dalam koleksi yang harus ditemubalikkan untuk memenuhi keinginan pengguna akan informasi. Suatu sistem temu balik informasi dikatakan ideal jika sistem tersebut dapat menemukan seluruh dokumen yang relevan dan sistem hanya menemukan dokumen yang relevan saja. Banyaknya varian morfologik pada suatu bahasa sangat berpengaruh terhadap hal tersebut, untuk itu setiap kata harus diubah kedalam bentuk dasarnya atau yang disebut dengan *stemming* untuk mengurangi perbedaan bentuk dari suatu kata, sehingga proses sistem temu kembali menjadi efektif [1].

#### D. Stemming

*Stemming* adalah proses untuk menemukan akar kata (*root*) atau kata dasar dengan memisahkan semua *affix* atau imbuhan yang melekat pada kata tersebut [3]. *Affix* (imbuhan) bisa terdiri dari awalan (*prefix*), akhiran (*suffix*), sisipan (*infix*), dan gabungan awalan-akhiran (*confix*). Pada banyak bahasa, kata-kata biasanya dihasilkan dengan menambahkan imbuhan pada kata dasarnya (*root*). Hasil dari *stemming* adalah *stem* (akar kata) yang merupakan bagian kata yang tersisa setelah dihilangkan imbuhan.

#### E. Algoritma Nazief & Adriani

Algoritma Nazief & Adriani dikembangkan pertama kali oleh Bobby Nazief dan Mirna Adriani. Algoritma ini berdasarkan pada aturan morfologi bahasa Indonesia yang mengelompokkan imbuhan, yaitu imbuhan yang diperbolehkan atau imbuhan yang tidak diperbolehkan. Pengelompokan ini termasuk imbuhan di depan (awalan), imbuhan dibelakang (akhiran), imbuhan di tengah (sisipan) dan kombinasi imbuhan awalan dan akhiran (konfiks).

Algoritma Nazief & Adriani yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut [5]:

- 1) Pertama cari kata yang akan di-*stemming* dalam kamus kata dasar. Jika ditemukan maka diasumsikan kata adalah *root word*. Maka algoritma berhenti.
- 2) *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa *particles* (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”), jika ada.
- 3) Hapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
  - a) Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan

dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.

- b) Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
- 4) Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
    - a) Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
    - b) For  $i = 1$  to 3, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
  - 5) Melakukan *Recoding*.
  - 6) Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

Kelebihan pada algoritma ini adalah memperhatikan kemungkinan adanya partikel-partikel yang mungkin mengikuti suatu kata berimbuhan. Selain itu algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih.

#### F. Afiksasi Bahasa Bali

Morfologi bahasa pada umumnya menunjuk pada afiksasi pada sebuah kata, begitupun halnya dengan Bahasa Bali. Di bawah ini penjelasan afiks tersebut.

##### 1. Prefiks (Awalan)

Bahasa Bali memiliki 17 macam prefiks. Ketujuh belas macam prefiks terdiri atas prefiks yang sering digunakan (produktif), prefiks yang kurang produktif, dan prefiks yang jarang digunakan.

Prefiks yang sering digunakan (produktif) yakni: a-; akatih, asibak, abungkul (selembar, sebagian, sebuah), ka-

; katunjel, katugel, kasibak (dibakar, dipotong, dibelah), sa-; saumah, sadados, sadurung (serumah, sebisanya, sebelum), pa-; penampel, pewarah, pengawi (alat untuk menutup, pengumuman, pengarang), ma-; majalan, makeber, matakon (berjalan, terbang, bertanya), pi-; piorah, piolas, pitresna (pemberitahuan, pertolongan, bantuan), m-; maca, mucu, mangunang (membaca, memojokkan, membangun), n-; nunu (membakar), ñ(ny); nyampat, nyate, nyebit (menyapu, membuat sate, membelah kecil-kecil), ŋ(ng); ngabut, ngigel, gengsapang, ngorahang, (mencabut, menari, melupakan, memberitahu).

Prefiks yang kurang produktif yaitu: pra-; pramangkin, prakanggo (seketika, pemuka masyarakat), para-; para sameton, para pamiarsa (para keluarga, para pendengar), pari-; pari polah, pari wangde (kelakuan, diundurkan), maka-; maka jalaran, maka dasar (yang menyebabkan, yang menjadi dasar), pati-; patikaplug, patidelik (saling seruduk, saling mendelik), kuma-; kumalipan, kumalindung (seperti lipan, seperti belut), upa-; upacara, upajiwa (upacara, mata pencaharian).

Prefiks yang jarang digunakan diantaranya: nir-/nis-; niskala, niridon (tidak nyata, tidak berguna), su-; susila (kelakuan baik), swa-; swadaya, swakarya (swadaya, bekerja sendiri).

## 2. Sufiks (Akhiran)

Berbeda dengan bahasa Indonesia, bahasa Bali memiliki jumlah sufiks yang lebih banyak. Bahasa Indonesia hanya memiliki 3 sufiks asli, sedangkan bahasa Bali memiliki lima sufiks, yaitu: -an; gedean, cerikan (lebih besar, lebih kecil), -a; jemaka, tugela (diambil, dipotong), -in; cerikin, gedein (kecilkan, besarkan), -ang; jemakang, tugelang (ambilkan, potongkan), -ne; kenjemne, limane (senyumnya, tangannya).

## 3. Infiks (Sisipan)

Dalam bahasa Bali, infiks itu tidak produktif. Hal itu sama dengan bahasa Indonesia yang hanya beberapa kata mengalami proses infiks ini. Jumlah infiks

dalam bahasa Bali ada dua, yaitu: -um-; gumuyu, sumanggup (besenda gurau, menyanggupi), -in-; sinambung (bersambung).

## 4. Konfiks (Kombinasi Prefiks & Sufiks)

Bahasa Bali memiliki 3 macam konfiks dan sangat kurang produktif digunakan. Konfiks-konfiks itu adalah: pa--an; pagenahan, panadtadan (rumah, alat menjinjing), ma--an; magregotan, macangkriman (sarat, bernyanyi cangkrim), ma--an; maririhan (memperdayakan).

## G. Stemming Bahasa Bali

*Stemming* Bahasa Bali menggunakan algoritma Bastal yang merupakan algoritma *stemming* untuk bahasa Bali yang diadaptasi dari algoritma Nazief & Adriani. *Stemming* yang dilakukan menggunakan metode *rule based* dalam pencarian *root* sebuah kata. Algoritma ini menggunakan kamus kata dasar sebagai acuan.

Algoritma Bastal mengelompokkan imbuhan (*wewehan*), yaitu imbuhan di depan (*pangater*), imbuhan dibelakang (*pangiring*), dan suara nasal (*anusuara*) berdasarkan pada aturan morfologi Bahasa Bali.

Algoritma Bastal memiliki tahap-tahap sebagai berikut:

- 1) Periksa jumlah huruf, kemudian cari kata yang akan di-*stemming* dalam kamus kata dasar. Jika terdiri atas 2(dua) huruf atau lebih sedikit, dan atau ditemukan pada kamus kata dasar maka diasumsikan kata merupakan *root-word*. Maka algoritma berhenti.
- 2) Jika ditemukan, *Pangater* ("a-") dibuang. Jika kata ditemukan dalam kamus, maka algoritma berhenti. Jika tidak maka lanjut ke langkah 2a.
  - a) *Pangater* yang dihapus ("a-") dikembalikan.
- 3) Jika ditemukan, *Pangater* ("ka-", "sa-", "di-", "pa-", "pi-", "ma-") dibuang. Jika kata ditemukan dalam kamus, maka algoritma berhenti. Jika tidak maka lakukan langkah 4 dan lanjutkan ke langkah 3a.

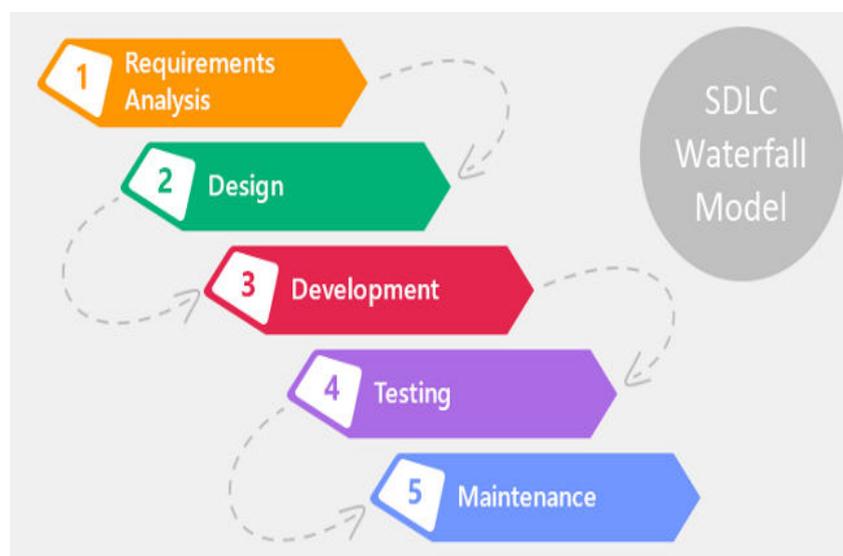
- a) Semua *pangater* (“ka-”, “sa-”, “di-”, “pa-”, “pi-”, “ma-”) yang terhapus dikembalikan.
- 4) Jika ditemukan, *Anusuara* (“ng-”, “ny-”, “n-”, “m-”) digantikan sesuai aturan. Jika kata ditemukan dalam kamus, maka algoritma berhenti. Jika tidak maka *anusuara* (“ng-”, “ny-”, “n-”, “m-”) yang tergantikan dikembalikan.
- 5) Jika ditemukan, *Pangiring* (“-a”) dibuang. Jika kata ditemukan dalam kamus, maka algoritma berhenti. Jika tidak maka lanjut ke langkah 5a.
  - a) *Pangiring* yang dihapus (“-a”) dikembalikan, lanjut ke langkah 3.
- 6) Jika ditemukan, *Pangiring* (“-an”, “-in”, “-ne”) atau pun “-ng” dibuang. Jika kata ditemukan dalam kamus, maka algoritma berhenti. Jika tidak maka lakukan langkah 2-4 dan lanjutkan ke langkah 6a.
  - a) Jika “-ng” telah dibuang dan huruf terakhir merupakan “a”, maka “a” juga ikut dibuang. Jika kata ditemukan dalam kamus, maka algoritma berhenti. Jika tidak maka lakukan langkah 2-4 dan lanjutkan ke langkah 6b.
  - b) Semua *pangiring* (“-an”, “-in”, “-ne”) atau pun “-ng” dan “a” yang terhapus dikembalikan.

- 7) Jika semua langkah (1-6) sudah dilewati tetapi belum menemukan *root-word*, maka kata awal diasumsikan sebagai *root-word*. Proses selesai.

Algoritma ini masih memerlukan penyempurnaan. Kelompok imbuhan (*wewehan*) yang diperhatikan pada algoritma ini hanya meliputi imbuhan di depan (*pangater*), imbuhan dibelakang (*pangiring*), dan suara nasal (*anusuara*). Algoritma ini belum memerhatikan imbuhan sisipan (*seselan*) sehingga belum mampu melakukan *stemming* kata/kalimat Bahasa Bali yang mengandung imbuhan sisipan (*seselan*).

## METODE

Algoritma Bastal diterapkan pada pengembangan Aplikasi Penerjemah Bahasa Indonesia – Bahasa Bali. Metode pengembangan perangkat lunak yang dilakukan peneliti mengacu pada model yang ada pada *Software Development Life Cycle* (SDLC). Model SDLC yang digunakan dalam pengembangan penelitian ini adalah model *waterfall* dan pemodelannya menggunakan diagram UML (*Unified Modelling Language*). Beberapa tahap penelitian yang akan dilakukan dalam penelitian ini yaitu bergerak maju mulai dari tahap analisis, perancangan, implementasi dan pengujian.



Gambar 1. SDLC Waterfall Model

#### A. Analisis Perangkat Lunak

Analisis kebutuhan dilakukan untuk mengetahui kebutuhan yang dibutuhkan dalam proses pengembangan.

##### 1. Kebutuhan Perangkat Lunak

###### a) Kebutuhan Fungsional

Berdasarkan analisis yang dilakukan terhadap proses-proses yang dapat diimplementasikan oleh Aplikasi adalah sebagai berikut.

- 1) Aplikasi dapat menampilkan halaman utama yang menyediakan menu utama pada aplikasi.
- 2) Aplikasi dapat menerjemahkan masukan berupa kalimat tunggal.
- 3) Aplikasi dapat menerjemahkan dari bahasa Bali ke bahasa Indonesia dengan Algoritma Bastal.

###### b) Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dari Aplikasi antara lain:

- 1) Desain antarmuka Aplikasi dirancang sedemikian rupa agar memudahkan pengguna aplikasi untuk menggunakannya (*user friendly*).
- 2) Aplikasi dapat diakses dengan waktu kurang dari 10 detik.

##### 2. Tujuan Pengembangan Perangkat Lunak

Aplikasi diharapkan mampu memenuhi proses-proses sebagai berikut.

- 1) Mampu menampilkan halaman utama yang menyediakan menu utama pada aplikasi.
- 2) Mampu menerjemahkan masukan berupa kalimat tunggal.
- 3) Mampu menerjemahkan dari bahasa Bali ke bahasa Indonesia dengan Algoritma Bastal.

##### 3. Masukan dan Keluaran Perangkat Lunak

Masukan dalam perangkat lunak yaitu berupa teks yang diketik langsung oleh pengguna, baik dalam bentuk kata ataupun kalimat. Keluaran dari perangkat lunak yaitu berupa teks hasil terjemahan, baik dalam bentuk kata ataupun kalimat.

#### HASIL DAN PEMBAHASAN

Implementasi perangkat lunak terdiri dari lingkungan implementasi perangkat

lunak, batasan implementasi perangkat lunak, implementasi arsitektur perangkat lunak, implementasi struktur data perangkat lunak, implementasi layar antarmuka perangkat lunak.

#### A. Lingkungan Implementasi

Aplikasi dikembangkan menggunakan beberapa perangkat lunak sebagai berikut.

1. Sistem Operasi Windows 10 Pro 64-bit
2. Sistem Operasi Android Versi 4.4 (Kitkat)
3. Android Studio
4. Java
5. SQLite Browser
6. Android Emulator
7. Google Firebase
8. Adobe Photoshop
9. Inkscape

Aplikasi dikembangkan menggunakan beberapa perangkat keras sebagai berikut.

##### a) Perangkat Komputer

1. Laptop Asus A43SD
2. Processor Intel(R) Core(TM) i3-2330M CPU 2.20GHz
3. RAM 8.00 GB
4. Dilengkapi alat *input* dan *output*

##### b) Perangkat Android

1. *Smartphone* Asus Zenfone 5
2. Resolusi 720 x 1280 *pixels*
3. Dual-core 2.0 GHz
4. RAM 1 GB
5. Kamera primer 8 MP

#### B. Batasan Implementasi

Batasan implementasi perangkat lunak Aplikasi Penerjemah Bahasa Indonesia - Bahasa Bali Menggunakan Metode *Rule Based* dan *String Matching* Berbasis Android dijabarkan sebagai berikut.

1. Spesifikasi minimal perangkat Android untuk dapat menjalankan aplikasi yaitu Processor ARM-v7a, GPU kelas *mid-end*, RAM 512 MB, OS Android versi 4.4 (Kitkat), Resolusi layar 320 x 480 *pixels*.
2. Aplikasi melakukan penerjemahan dua arah yaitu dari bahasa Indonesia ke bahasa Bali dan dari bahasa Bali ke bahasa Indonesia.
3. Terjemahan untuk kata yang merupakan sebuah nama, istilah

khusus, kata/kalimat khiasan, kata/kalimat ungkapan, dan kata perulangan, hasilnya terbatas dan kemungkinan tidak sesuai.

4. Hasil terjemahan bersifat apa adanya dan akan terdapat beberapa awalan dan akhiran yang tidak sesuai.
5. Daftar kata yang terdapat pada aplikasi bersumber dari *dataset* berdasarkan buku dan digunakan sebagai *database* sebanyak  $\pm 3.000$  baris kosakata.

### C. Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan melalui beberapa pengujian, yaitu *black box testing*, *white box testing*, uji ahli, dan uji akurasi serta evaluasi dari pengguna. Pengujian dilaksanakan mulai dari hari Senin, 11 Desember 2017 sampai dengan hari Kamis, 14 Desember 2017.

Setelah melakukan pengujian fungsional (*black box testing*) pada aplikasi oleh 5 orang penguji dengan *smartphone* yang berbeda-beda, hasilnya adalah semua proses sesuai atau semua proses berhasil berjalan dengan baik. Hasil tersebut juga menunjukkan bahwa aplikasi mampu berjalan dengan baik di berbagai *smartphone* yang berbeda. Selanjutnya pada pengujian struktural (*white box testing*), hasilnya menyimpulkan bahwa proses kerja perangkat lunak secara internal bekerja dengan baik sesuai dengan spesifikasi yang telah ditetapkan. *Source code* program bekerja dengan baik dan benar sesuai dengan fungsinya.

Pengujian untuk mengetahui tingkat akurasi aplikasi yang telah dilakukan oleh 5 orang penguji, hasilnya menyatakan bahwa aplikasi sangat akurat dengan persentase keseluruhan penilaian yaitu 96,15%. Uji akurasi kategori penerjemahan kata, hasilnya yaitu dari 100 kata yang diujikan terdapat 1 kata yang terjemahannya tidak sesuai sehingga persentase tingkat akurasi pada kategori penerjemahan kata sebesar 99%. Sedangkan untuk kategori penerjemahan kalimat, dari 30 kalimat yang diujikan terdapat 4 kalimat yang hasil terjemahannya tidak sesuai sehingga persentase tingkat akurasi pada kategori penerjemahan kalimat sebesar 86,67%.

Pengujian ahli media yang telah dilakukan oleh 2 orang ahli, hasilnya menyatakan bahwa aplikasi sangat layak dengan persentase keseluruhan penilaian yaitu 100%. Penilaian dari segi tampilan aplikasi berada pada tingkat persentase sebesar 100% dan dari segi kualitas teknis juga sebesar 100%. Hasil tersebut diperoleh setelah menindak lanjut saran perbaikan dari ahli, yaitu perbaikan warna *font* pada aplikasi.

Selanjutnya pengujian untuk mengetahui penilaian pengguna terhadap aplikasi. Pengujian melibatkan 15 orang pengguna dari kalangan masyarakat umum. Pengujian dilakukan menggunakan angket penilaian UEQ (*User Experience Questionnaire*). Berdasarkan perhitungan angket UEQ tersebut diperoleh hasil antara lain: dari segi daya tarik aplikasi tergolong sangat baik, dari segi kejelasan aplikasi sangat baik, dalam hal efisiensinya aplikasi sangat baik, ketepatan aplikasi sangat baik, stimulasi aplikasi sangat baik, dan dari segi kebaruan juga dinyatakan sangat baik.

### SIMPULAN

Algoritma Bastal yang diimplementasikan dalam sebuah aplikasi, mampu melakukan *stemming* teks Bahasa Bali dengan baik. Aplikasi telah berhasil dikembangkan dengan menggunakan model *waterfall* pada metode pengembangan *Software Development Life Cycle (SDLC)*. Aplikasi memiliki tingkat akurasi yang sangat akurat dengan persentase keseluruhan penilaian yaitu 96,15%. Uji akurasi kategori penerjemahan kata, persentase tingkat akurasinya sebesar 99%. Sedangkan untuk kategori penerjemahan kalimat, tingkat akurasinya 86,67%.

Berdasarkan hasil penelitian, pengembangan, dan kesimpulan yang telah dipaparkan sebelumnya, terdapat beberapa saran untuk penelitian selanjutnya antara lain: Algoritma Bastal membutuhkan penyempurnaan dalam hal penerjemahan dengan penggunaan metode serta aturan-aturan yang lebih tepat agar hasil penerjemahan menjadi lebih sempurna. Aplikasi penerjemah yang mengimplementasikan algoritma Bastal

