

# Sistem Pemeriksa Pola Kalimat Bahasa Indonesia berbasis Algoritme *Left-Corner Parsing* dengan *Stemming*

Hartanto Kusuma Wardana<sup>1</sup>, Iswara Swanita<sup>2</sup>, Banu Wirawan Yohanes<sup>3</sup>

**Abstract**—In recent years, human computer interaction has been attracted many researchers and applied in various applications. In order to understand human language, the computer is required to learn human language grammar, one of the languages is Bahasa Indonesia. Left-corner parsing algorithm is used extensively to check English grammar. However, only several implementations are found in Bahasa Indonesia. By combining Left-Corner Parsing algorithm with stemming algorithm, to reproduce basic word and word type before the sentence is parsed according to production rules defined, this article discusses Bahasa Indonesia sentence pattern checking system testing for four single sentence patterns, i.e., Subject-Predicate, SP-Object, SP-Adverb, and SPOA. The testing result shows the accuracy of sentence pattern recognition of 95.3%.

**Intisari**—Pada beberapa tahun terakhir ini, interaksi manusia dengan sistem komputer menggunakan bahasa alami manusia semakin banyak diminati dan diterapkan pada berbagai aplikasi. Agar komputer dapat memahami makna dari bahasa yang diberikan, komputer perlu mempelajari tata bahasa manusia, salah satunya adalah bahasa Indonesia. Algoritme *Left-Corner Parsing* banyak digunakan untuk memeriksa tata cara penulisan bahasa Inggris, tetapi belum banyak yang diterapkan pada bahasa Indonesia. Dengan menggabungkan algoritme *Left-Corner Parsing* dengan algoritme *stemming*, yang menentukan kata dasar dan jenis kata sebelum diumpungkan kepada *parsing* untuk diuraikan kalimatnya sesuai aturan produksi yang ditetapkan, pada artikel ini dibahas pengujian sistem pemeriksa pola kalimat bahasa Indonesia pada empat pola kalimat tunggal, yaitu Subjek-Predikat, SP-Objek, SP-Keterangan, dan SPOK. Hasil yang diperoleh menunjukkan tingkat keberhasilan pengenalan pola kalimat sebesar 95,3%.

**Kata kunci**—Pemeriksa pola kalimat bahasa Indonesia, *Left-corner parsing*, *stemming*.

## I. PENDAHULUAN

Pada era teknologi informasi dan komunikasi ini banyak kemajuan dalam bidang bahasa. Salah satunya adalah interaksi manusia dengan sistem komputer menggunakan bahasa alami manusia. Agar komputer dapat memahami makna dari bahasa yang diberikan, komputer perlu mempelajari tata bahasa manusia. Ada tiga tahapan yang perlu dilakukan, yaitu analisis sintaksis yang berfungsi untuk memeriksa susunan kalimat dari bahasa yang diberikan berdasarkan tata bahasa yang ada,

interpretasi semantik yang berguna untuk merepresentasikan arti dari kalimat secara konteks, dan interpretasi kontekstual yang berguna untuk menentukan makna dari penggunaan suatu kalimat [1].

Analisis sintaksis dapat diterapkan pada sistem pemeriksa tata bahasa Indonesia. Tata bahasa merupakan sekumpulan kaidah tentang struktur gramatikal bahasa, meliputi fonologi, morfologi, dan sintaksis [2]. Kaidah fonologi berkaitan dengan tata bunyi suatu bahasa, seperti penggunaan huruf vokal, konsonan, diftong (vokal rangkap), dan konsonan rangkap (*consonant cluster*). Sementara, kaidah morfologi berkaitan dengan aturan pembentuk suatu kata. Aturan ini mengatur perubahan suatu kata setelah mendapatkan imbuhan, misalnya imbuhan me-, -kan, di-, dan ber-.

Kaidah sintaksis berkaitan dengan aturan pembentuk suatu kalimat. Misalnya, pada kalimat “Seorang pelajar sedang belajar matematika di perpustakaan.”, kalimat tersebut terbentuk dari kategori pola Subjek-Predikat-Objek-Keterangan (S-P-O-K) dengan struktur kalimat sebagai berikut. “Seorang pelajar” (subjek) sebagai frasa nomina, “belajar” (predikat) sebagai frasa verba, “matematika” (objek) sebagai nomina, dan “di perpustakaan” (keterangan tempat) sebagai frasa adverbial [3].

Penelitian sebelumnya pada analisis tata bahasa penerjemah bahasa Indonesia ke Jawa Ngoko, dan sebaliknya, menggunakan algoritme *Left-Corner Parsing* (LCP) telah dilakukan [4]. Selain itu, terdapat penerapan LCP pada pembelajaran tata bahasa Inggris pada permainan petualangan tiga dimensi “Go to London” [5]. LCP digunakan untuk memeriksa sintaksis pada permainan papan tersebut, tetapi belum dapat memberi makna kata. Serupa dengan penelitian tersebut, LCP juga telah digunakan pada pemeriksa tata bahasa Inggris, tetapi menggunakan parser probabilistik dan *part-of-speech* (POS) *tag* [6]. Sebagai tambahan, telah dianalisis pula perbandingan algoritme LCP dan Cocke-Younger-Kasami (CY) untuk memeriksa delapan pola kalimat baku bahasa Indonesia, tetapi baru mencapai akurasi 65% dan 60%, untuk CYK dan LCP secara berurutan [7]. Hal ini disebabkan banyak kata yang tidak dikenali dalam kamus *POS tag*. Perancangan aplikasi pemeriksa tata bahasa Indonesia menggunakan LCP berbasis *rational unified process* pada *Unified Modeling Language* (UML) juga telah dikembangkan [8]. Tidak ketinggalan, telah diteliti pula penerapan metode LCP pada penerjemah bahasa Indonesia ke bahasa Bima [9].

Pengolahan bahasa alamiah, khususnya analisis dari korpus paralel, meneliti tentang penggunaan metode *crowd-sourcing* untuk mengumpulkan kosakata bahasa Inggris, Sunda, dan Jawa tentang *sustainable development goals* dari UNDP, dan *machine learning* statistik untuk menerjemahkan bahasa Inggris menjadi bahasa Sunda dan Jawa [10]. Telah diteliti

<sup>1,3</sup> Program Studi Sistem Komputer, Fakultas Teknik Elektronika dan Komputer, Universitas Kristen Satya Wacana, Jln. Diponegoro 52-60 Salatiga 50711 INDONESIA (e-mail: <sup>1</sup>hwardana1@gmail.com, <sup>3</sup>bonayo@gmail.com; email korespondensi)

<sup>2</sup> Program Studi Teknik Elektro, Fakultas Teknik Elektronika dan Komputer, Universitas Kristen Satya Wacana, Jln. Diponegoro 52-60 Salatiga 50711 INDONESIA (e-mail: iswaraswanita@gmail.com)

juga metode penerjemahan bahasa Jawa ke aksara Jawa, dan sebaliknya, menggunakan karakter *Unicode*, berbasis *Finite State Automata* (FSA) [11].

Sebelumnya juga telah dikembangkan sistem penelusuran web berbasis algoritme genetik [12]. Penelusuran web tersebut menggunakan sebuah dokumen contoh dan kata kunci yang terdapat di dalam dokumen tersebut. Kemudian dibangun koleksi halaman web yang mirip dengan halaman web contoh berdasarkan dua kategori, yaitu jumlah kemunculan kata kunci dan banyaknya *link* yang merujuk dokumen tersebut. Namun, penelusuran web tersebut belum menyertakan analisis teks pada pengolahan bahasa alamiah secara menyeluruh.

Makalah ini membahas penerapan algoritme LCP untuk pemeriksa tata bahasa Indonesia yang dilengkapi dengan algoritme *stemming* karya Nazief dan Adriani. LCP dipilih karena sesuai dengan konteks struktural [13], seperti pola tata bahasa Indonesia dasar. Sedangkan *stemming* dipilih karena diharapkan dapat meningkatkan akurasi dibanding algoritme *stemming* Indonesia lainnya [14], [15].

Sistematika makalah ini dimulai dengan latar belakang persoalan perlunya sistem pemeriksa tata bahasa Indonesia yang telah dilengkapi dengan *stemming* kata. Kemudian penjelasan singkat tentang tata bahasa Indonesia, algoritme LCP, dan algoritme *stemming* yang digunakan. Dilanjutkan dengan perancangan sistem pemeriksa tata bahasa Indonesia berupa arsitektur sistem dan algoritme pemeriksa tata bahasa Indonesia berbasis LCP. Lalu, disajikan hasil pengujian penerjemahan pada beberapa pola kalimat tunggal yang telah ditentukan, beserta analisisnya. Bagian terakhir menyajikan kesimpulan dari pengembangan algoritme pemeriksa tata bahasa Indonesia.

## II. TATA BAHASA INDONESIA, ALGORITME LEFT CORNER PARSING, DAN STEMMING

### A. Tata Bahasa Indonesia

Jenis kata dalam bahasa Indonesia merupakan golongan kata yang memiliki kesamaan bentuk, fungsi, dan perilaku sintaksisnya [16]. Jenis kata yang digunakan pada penelitian ini dibedakan menjadi sebagai berikut.

- Kata benda/nomina (N), misalnya naturalis, bahasa, makanan.
- Kata kerja/verba (V), misalnya alih, lari, bersemangat.
- Kata sifat/adjektiva (Aj), misalnya besar, luas.
- Kata ganti/pronomina (Po), misalnya dia, kamu, aku.
- Kata keterangan/adverbia (Ad), misalnya belum.
- Kata bilangan/numeralia (Nu), misalnya seribu, seratus.
- Kata penghubung/konjungsi (Ko), misalnya dan, tapi.
- Kata depan/preposisi (Pe), misalnya di, ke, dari.

Imbuhan merupakan bubuhan yang berupa awalan, sisipan, atau akhiran pada suatu kata dan menghasilkan kata baru [16]. Misalnya, pe- + catur = pecatur, peN- + daftar = pendaftar, per- + satu + -an = persatuan, dan temu + -an = temuan.

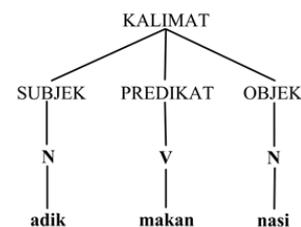
Frasa adalah gabungan dua atau lebih kata yang tidak mengandung unsur predikat [17]. Frasa dibedakan menjadi lima jenis, yaitu frasa verbal yang menyatakan tindakan atau perbuatan, frasa adjektival yang menyatakan sifat atau

keadaan, frasa adverbial yang menerangkan predikat, frasa nominal yang menunjuk suatu benda, dan frasa preposisional yang menunjukkan arah, tempat, dan waktu.

Berbeda dengan frasa, klausa merupakan kelompok kata yang mengandung fungsi subjek dan predikat [17]. Fungsi tersebut dapat diisi oleh kata atau frasa, misalnya “saya” (subjek) “menulis” (predikat) “makalah” (objek). Kalimat adalah bagian tulisan yang mempunyai struktur minimal subjek dan predikat, serta intonasi akhir yang menunjukkan bagian tulisan itu sudah lengkap dengan makna, yaitu bernada berita, tanya, atau perintah [17].

### B. Algoritme Left-Corner Parsing (LCP)

*Parsing* adalah cara memetakan suatu kalimat menjadi suatu pohon uraian (*parse tree*). *Parsing* mengubah kalimat menjadi model struktur tata bahasa [18], seperti ditunjukkan oleh Gbr. 1.



Gbr. 1 Contoh pohon uraian.

Algoritme LCP menerima kata yang dimasukkan terlebih dahulu, lalu menentukan jenis konstituen yang sesuai dengan jenis kata dalam kalimat. Algoritme ini bekerja secara *bottom-up* pada awal proses dan diakhiri secara *top-down*.

Algoritme *bottom-up parsing* membaca tiap kata dalam kalimat, kemudian dicari konstituen terbesar berupa kalimat masukan. Kelebihannya adalah untuk penanganan rekursi-ke kiri kiri (*left-recursion*), tetapi tidak dapat mengatasi permasalahan *empty production* [1].

Sebaliknya, algoritme *top-down parsing* menguraikan kalimat, dimulai dari konstituen terbesar menjadi yang lebih kecil, hingga didapatkan konstituen terkecil yang berupa kata. Kelebihannya adalah dalam mengatasi permasalahan *empty production*, tetapi memiliki kekurangan dalam mengatasi *left recursion* [1].

Algoritme LCP menggunakan tiga *stack* dalam proses memeriksa kalimat. *Stack* pertama berisi kalimat yang akan diperiksa. Kedua, kategori yang berisi daftar aturan ruas kanan yang akan diperiksa. Dan ketiga, konstituen yang berisi hasil pengelompokan aturan ruas kanan dalam aturan produksi.

Selain itu, algoritme LCP juga menggunakan tiga operasi, yaitu pindahkan (*move*, M), hapus (*remove*, Rm), dan kurangi (*reduce*, Rd). Operasi pindahkan dikerjakan ketika nilai paling kiri dari *stack* kategori bernilai \$. Lalu nilai paling kiri dari *stack* konstituen dipindahkan ke dalam *stack* kalimat, dan nilai \$ pada *stack* kategori akan dihapus. Operasi hapus dikerjakan ketika nilai paling kiri dari kedua *stack*, kalimat dan kategori, bernilai sama. Lalu, kedua nilai tersebut akan dihapus. Operasi kurangi dikerjakan ketika syarat kedua operasi, pindahkan dan hapus, tidak terpenuhi, sehingga dalam proses ini, nilai masukan paling kiri dari *stack* kalimat akan dicari

aturan ruas kirinya, dan sisa aturan ruas kanan disimpan ke dalam *stack* kategori [13]. Misalnya, diketahui aturan produksi pada algoritme LCP sebagai berikut:

$S \rightarrow ASB \mid d$   
 $A \rightarrow a$   
 $B \rightarrow b$

Ketika menerima masukan: adb, maka algoritme LCP akan bekerja seperti pada langkah kerja di Tabel I. Apabila masukan sesuai dengan aturan produksi, maka pengecekan berhasil, ditandai dengan tidak ada nilai tersisa pada ketiga *stack*.

TABEL I  
CARA KERJA ALGORITME LEFT-CORNER PARSING

Stack kalimat	Stack kategori	Stack konstituen	Operasi
adb	\$	-	-
db	\$\$	A	Kurangi A $\rightarrow$ a
Adb	S	-	Pindahkan
db	SB\$\$	S	Kurangi S $\rightarrow$ ASB
b	\$B\$\$	SS	Kurangi S $\rightarrow$ d
Sb	SB\$\$	S	Pindahkan
b	B\$\$	S	Hapus
-	\$B\$\$	BS	Kurangi B $\rightarrow$ b
B	B\$\$	S	Pindahkan
-	\$\$	S	Hapus
S	S	-	Pindahkan
-	-	-	Hapus

### C. Algoritme Stemming dari Nazief dan Adriani [19]

*Stemming* merupakan proses untuk mendapatkan kata dasar dari kata asli pada suatu kalimat. Kata asli dapat mengandung imbuhan yang dipisahkan berdasarkan aturan tertentu, misalnya kata makanan, dimakan, memakan yang memiliki satu kata dasar yang sama, yaitu makan.

Dalam bahasa Indonesia, salah satu algoritme *stemming* dikembangkan oleh Nazief dan Adriani. Algoritme ini menggunakan kamus kata dasar sebagai acuan dalam proses *stem* kata. Jika semakin banyak kata dasar yang tersedia dalam kamus, maka algoritme ini akan semakin akurat [14]. Berikut ini urutan cara kerja algoritme *stemming*.

1. Mengecek kata masukan, jika terdapat di dalam kamus kata dasar, maka selesai. Jika tidak ada di kamus, maka dilanjutkan ke langkah kedua.
2. Menghapus akhiran infleksif. Akhiran-akhiran -kah, -lah, -tah, dan -pun akan dihapus. Selanjutnya, dihapus juga akhiran kepemilikan dalam kata benda, yaitu -ku, -mu, dan -nya.
3. Menghapus akhiran turunan, seperti -i, -kan, dan -an.
4. Menghapus awalan turunan, seperti be-, di-, ke-, me-, pe-, se-, dan te-. Proses akan berhenti jika ditemukan kombinasi imbuhan awalan dan akhiran yang tidak diperbolehkan, misalnya ber-, di-, ke-, -i, -an, -i dan -kan, dan -an. Proses akan berhenti juga jika awalan yang akan dihapus identik dengan awalan yang telah dihapus sebelumnya, atau awalan telah dihapus sebanyak tiga kali.

Identifikasi tipe awalan dilakukan dengan aturan yang ditetapkan [19].

5. Perekaman. Misalnya, kata “menyapu”, menggunakan aturan pada [19], akan diuraikan menjadi meny- + sapu.
6. Apabila kelima langkah telah dikerjakan dan kata masih tidak ditemukan di dalam kamus kata dasar, maka kata akan dikembalikan menjadi kata asli sebelum di-*stem*.

### III. METODE

Metodologi penelitian yang digunakan meliputi studi literatur untuk perbandingan algoritme pemeriksa tata bahasa, analisis persyaratan sistem perangkat lunak, perancangan dan implementasi sistem, serta pengujian sistem.

Pemeriksa tata bahasa yang baik memperhatikan penggunaan pola penyusunan kalimat. Pada sistem pemeriksa tata bahasa Indonesia ini, digunakan pola dasar kalimat berupa: S-P, S-P-O, S-P-K, dan S-P-O-K, dengan struktur jenis kata yang dapat menempati pola tersebut ditampilkan pada Tabel II.

TABEL II  
JENIS KATA PADA KATEGORI POLA KALIMAT

Subjek (S)	Predikat (P)	Objek (O)	Keterangan (K)
N	N	N	Ad
FN	FN	FN	FD
	V		N
	FV		
	Aj		
	FAj		
	Nu		
	FD		

Keterangan: F = frasa, D = depan.

Sedangkan aturan pola kalimat ditampilkan pada aturan produksi sebagai berikut.

$Q \rightarrow SP \mid SPO \mid SPK \mid SPOK$   
 $K \rightarrow Ad \mid Fd \mid N$   
 $O \rightarrow N \mid FN \mid PRON$   
 $P \rightarrow N \mid FN \mid V \mid FV \mid Aj \mid FAj \mid Nu \mid FD$   
 $S \rightarrow N \mid FN \mid Po$

Sebuah basis data dibentuk untuk menyimpan data kata dasar dan jenis katanya. Kumpulan kata dasar tersebut diperoleh dari web [20]. Basis data ini digunakan dalam proses *stemming* dan hasilnya digunakan untuk menentukan jenis kata.

Antarmuka pengguna dibuat menggunakan *Windows form*, seperti ditunjukkan oleh Gbr. 2. Isi dokumen yang akan diperiksa tata bahasa Indonesianya dan hasil pemeriksaan ditampilkan pada *text box* yang sesuai. Fungsi yang disediakan oleh sistem pemeriksa tata bahasa meliputi hal-hal sebagai berikut.

1. *Mode input*, yaitu pilihan cara memasukkan teks;
  - Dokumen: untuk masukan berupa file Word 2007 dengan ekstensi *.doc/.docx*.
  - Manual: mengetik teks menggunakan *keyboard*.
2. *FileName* untuk menampilkan nama dokumen.

3. Tombol *open* untuk memilih dan membuka dokumen yang akan diperiksa, dan tombol *check* untuk menjalankan pemeriksaan tata bahasa.
4. *Input document* untuk menampilkan isi dokumen yang sedang digunakan.
5. *Result* untuk menampilkan hasil pemeriksaan yang melalui tahapan *tokenisasi*, pengelompokan jenis kata, dan status hasil pemeriksaan.



Gbr. 2 Antarmuka pengguna sistem pemeriksa tata bahasa.

Tahapan kerja dari sistem pemeriksa tata bahasa Indonesia ini dimulai dari menerima masukan kalimat, tokenisasi, *stemming* kata, menentukan jenis kata, mengecek struktur kalimat dengan algoritme LCP, dan menampilkan hasil pemeriksaan tata bahasa.

Kalimat yang akan diperiksa merupakan kalimat berita tanpa format penomoran dan daftar. Bahasa pada kalimat dibatasi pada bahasa baku tanpa simbol, singkatan, dan angka. Setiap kata diharapkan memiliki arti tunggal untuk menghindari ambiguitas.

Tokenisasi merupakan pemecahan paragraf atau kalimat menjadi satuan yang lebih kecil yang disebut token. Awalnya, proses pemecahan dilakukan berdasarkan pemisah tanda titik. Kemudian dilakukan pemecahan lagi menggunakan pemisah spasi. Tokenisasi dapat mempermudah penentuan jenis kata pada *stemming*, yang telah dijelaskan pada bagian sebelumnya.

*Stemming* dapat menghasilkan kata dasar, jenis kata dasar, dan imbuhan yang menyusun kata tersebut. Pembentukan kata dasar dilakukan menggunakan aturan morfologis dan sintaksis.

Pemeriksaan tata bahasa baku bahasa Indonesia dilakukan menggunakan algoritme LCP, dimulai dengan menempatkan nilai inisial pada ketiga *stack*. Nilai inisial berupa kumpulan jenis kata dari sebuah kalimat dan simbol terminal pada *stack* masukan dan kategori. Selanjutnya, dilakukan proses pindahkan, hapus, dan kurangi pada ketiga data *stack* tersebut.

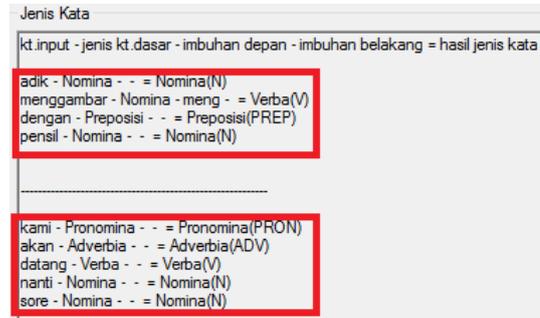
IV. HASIL PENGUJIAN DAN ANALISIS

Pengujian sistem pemeriksa tata bahasa Indonesia dilakukan dengan beberapa skenario pada masukan teks yang diambil dari beberapa sumber [17], [21]-[23].

Jumlah kata maksimal yang dapat diperiksa oleh sistem yang dibuat sebanyak 5.000 kata. Jika total kata yang akan diperiksa kosong atau melebihi kapasitas yang diperbolehkan, maka sistem akan mencegah proses pemeriksaan dan menampilkan pesan kesalahan kepada pengguna.

Hasil dari tokenisasi diperiksa menggunakan algoritme *stemming* untuk menentukan imbuhan pada kata dan juga jenis kata dasarnya. *Stemming* ini menggunakan kamus kata yang tersimpan pada basis data. Jika kata pada basis data semakin

lengkap, maka hasil *stemming* akan lebih baik dan sesuai. Format keluaran algoritme *stemming* ini adalah [kata yang diperiksa] – [jenis kata dasar] – [awalan] – [akhiran]. Contoh keluaran dari algoritme *stemming* ditampilkan pada Gbr. 3. Dalam pembentukan jenis kata, ditemukan beberapa kesulitan dalam penentuan aturan morfologisnya, seperti perbedaan informasi mengenai aturan morfologis yang didapat dari acuan yang berbeda. Oleh karena itu, permasalahan tersebut diselesaikan dengan mengacu pada salah satu sumber saja. Dalam hal ini, digunakan dua acuan dalam pembentukan aturan jenis kata untuk saling melengkapi [16], [17].



Gbr. 3 Keluaran algoritme *stemming* kata.

TABEL III  
HASIL PEMERIKSAAN POLA KALIMAT

Kalimat	Pola pada Buku	Pola pada Sistem	Keterangan
Adik menggambar dengan pensil.	S P K	S P K	BERHASIL
Kami akan datang nanti sore.	S P K	S P O K	GAGAL
Kucing itu gemuk sekali.	S P	S P	BERHASIL
Ibu menanak nasi.	S P O	S P O	BERHASIL

Aturan pola kalimat yang digunakan dalam pemeriksaan tata bahasa ini masih relatif sederhana. Misalnya, sistem belum dapat menentukan beberapa gabungan kata yang membentuk suatu frasa. Hal ini menyebabkan kata yang seharusnya benar menjadi salah akibat pola yang tidak dikenali. Contoh hasil pemeriksaan ditampilkan pada Tabel III.

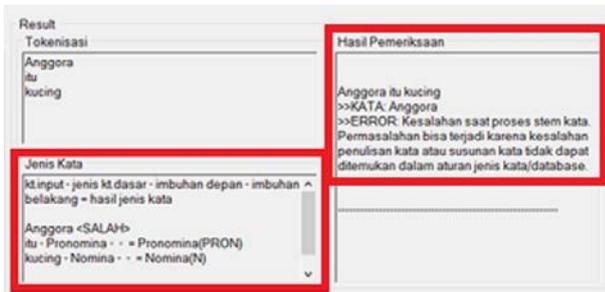
Permasalahan lain yang muncul adalah kesalahan pola yang dideteksi oleh sistem. Hal ini disebabkan oleh sistem yang tidak dapat mengidentifikasi frasa dalam suatu kalimat, sehingga jika kata yang seharusnya tergabung dalam frasa justru berdiri sendiri dan jenis katanya sesuai dengan pola yang diharapkan dalam aturan, sistem mendeteksinya sebagai hasil benar meskipun pola yang dipilih salah.

Cuplikan hasil pengujian sistem secara keseluruhan pada pola kalimat SP ditampilkan dalam Tabel IV. Kalimat masukan diambil dari buku Jenis Kalimat dalam Bahasa Indonesia [16] dengan jumlah tujuh belas kalimat yang akan diperiksa dan terdapat 52 kata. Berikutnya, kalimat diambil dari buku Komposisi Bahasa Indonesia untuk Mahasiswa Nonjurusan Bahasa [17] dengan jumlah enam kalimat yang diperiksa dan terdapat sembilan belas kata. Kemudian, kalimat juga diambil dari buku Tata Bahasa Baku Bahasa Indonesia [22] dengan jumlah enam kalimat yang diperiksa dan delapan belas

kata. Selain itu, kalimat juga diambil dari buku Sintaksis Bahasa Indonesia (Pendekatan Proses) [23] sebanyak 24 kalimat dan 73 kata.

TABEL IV  
HASIL PEMERIKSAAN POLA KALIMAT SUBJEK-PREDIKAT (SP)

Kalimat	Keterangan
Gubernur itu diperiksa.	BERHASIL
Kami tidak berangkat.	BERHASIL
Ibu sedang ke pasar.	GAGAL (Pola tidak terdaftar)
Anak itu sedang sekolah.	BERHASIL
Ayahnya dari Bali.	BERHASIL
Ketua komite itu rapat.	BERHASIL
Kuda merumput.	BERHASIL
Orang itu sedang tidur.	BERHASIL
Dia dari medan.	BERHASIL
Pencuri itu lari.	BERHASIL
Kesehatannya sudah membaik.	BERHASIL
Pesawat itu akan mendarat.	BERHASIL
Nenek ke Medan.	BERHASIL
Kakek dari pasar.	BERHASIL
Anjing itu binatang.	BERHASIL
Kakap itu ikan.	BERHASIL
Anggora itu kucing.	GAGAL (kata tidak dikenali )



Gbr. 4 Pesan kesalahan pengenalan pola kalimat SP.

TABEL V  
HASIL PEMERIKSAAN POLA KALIMAT SUBJEK-PREDIKAT-OBJEK (SPO)

Kalimat	Keterangan
Para demonstran mendatangi KPUD.	BERHASIL
Ayah menulis surat itu.	BERHASIL
Dia mendapat piagam.	BERHASIL
Ayah memanggil orang itu.	BERHASIL
Mereka kejatuhan pohon.	BERHASIL
Rudi menendang bola.	BERHASIL
Ibu menanak nasi.	BERHASIL

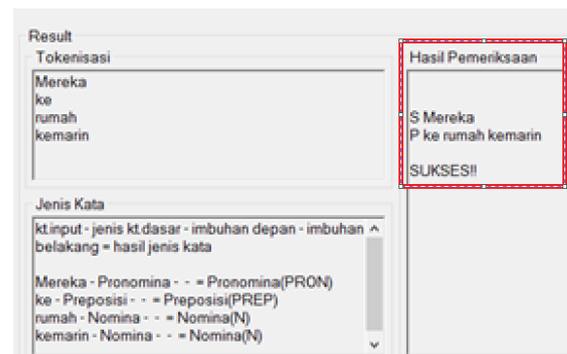
Kalimat “Ibu sedang ke pasar.” memiliki pola kalimat SP dengan subjek “ibu” dan predikatnya adalah “sedang ke pasar”. Namun, sistem tidak dapat menemukan pola yang sesuai dengan pola kalimat yang disediakan pada sistem, sehingga menyebabkan hasil pemeriksaan untuk kalimat tersebut menjadi salah. Selain itu, sistem belum dapat mengenali nama orang, kota, negara, angka, atau singkatan kata. Misalnya, kalimat “Anggora itu kucing.” tidak dapat dikenali polanya karena terdapat nama jenis kucing “Anggora.” Tampilan kesalahan pengenalan pola kalimat SP ditunjukkan pada Gbr. 4.

Cuplikan hasil pengujian sistem secara keseluruhan pada pola kalimat SPO ditampilkan pada Tabel V. Kalimat diambil

dari buku Jenis Kalimat dalam Bahasa Indonesia [21] sejumlah empat kalimat dan enam belas kata yang diperiksa. Lalu, kalimat juga diambil dari buku Komposisi Bahasa Indonesia untuk Mahasiswa Nonjurusan Bahasa [17] dengan jumlah satu kalimat dan tiga kata yang diperiksa. Tambahan kalimat diambil dari buku Tatabahasa Baku Bahasa Indonesia [22] dengan jumlah sembilan kalimat dan 34 kata. Selain itu, kalimat diambil dari buku Sintaksis Bahasa Indonesia (Pendekatan Proses) [23] dengan jumlah delapan kalimat dan 25 kata. Untuk pola kalimat SPO, semua masukan kalimat berhasil diperiksa dengan benar.

TABEL VI  
HASIL PEMERIKSAAN POLA KALIMAT SUBJEK-PREDIKAT-KETERANGAN (SPK)

Kalimat	Keterangan
Mereka ke rumah kemarin.	GAGAL (kesalahan pengenalan pola)
Benteng itu diratakan dengan tanah.	BERHASIL
Polisi membawanya ke rumah sakit.	BERHASIL
Kami tinggal di Jakarta.	BERHASIL
Ibu sedang makan di dapur.	BERHASIL
Kami tinggal di Jatinegara.	BERHASIL
Dia berjalan cepat.	GAGAL (pola tidak terdaftar)
Kami membaca perlahan-lahan.	GAGAL (kesalahan pengenalan pola)
Ibu pergi ke pasar.	BERHASIL



Gbr. 5 Pesan kesalahan pengenalan pola kalimat SPK.

Cuplikan hasil pengujian sistem secara keseluruhan pada pola kalimat SPK ditampilkan pada Tabel VI. Kalimat diambil dari buku Jenis Kalimat dalam Bahasa Indonesia [21] dengan jumlah dua kalimat dan sembilan kata yang diperiksa. Kalimat juga diambil dari buku Komposisi Bahasa Indonesia untuk Mahasiswa Nonjurusan Bahasa [17] dengan jumlah tiga kalimat dan tiga belas kata yang diperiksa. Lalu, kalimat diambil dari buku Tatabahasa Baku Bahasa Indonesia [22] dengan jumlah lima kalimat dan 23 kata yang diperiksa. Tambahan kalimat juga diambil dari buku Sintaksis Bahasa Indonesia (Pendekatan Proses) [18] dengan jumlah sepuluh kalimat dan empat puluh kata yang diperiksa.

Kalimat “Mereka ke rumah kemarin.” seharusnya memiliki pola SPK. Namun, sistem mengidentifikasinya sebagai pola SP seperti ditunjukkan pada Gbr. 5. Kemudian, kalimat “Kami membaca perlahan-lahan.” dikenali sebagai pola SPO karena kata dasar dari perlahan-lahan menjadi lahan. Hal ini disebabkan oleh kondisi aturan dalam sistem. Apabila suatu

kata memiliki jenis kata yang sesuai untuk mengisi suatu cabang pohon uraian dalam aturan yang disediakan, maka kata tersebut akan mengisi posisi tersebut meskipun menghasilkan makna yang kurang sesuai. Selain itu, terdapat kesalahan pengenalan pola kalimat akibat pola aturan tidak terdaftar di dalam sistem. Misalnya pada kalimat “Dia berjalan cepat.” tidak berhasil dikenali karena tidak terdapat kata “cepat” pada basis data.

TABEL VII  
HASIL PEMERIKSAAN POLA KALIMAT SUBJEK-PREDIKAT-OBJEK-  
KETERANGAN (SPOK)

Kalimat	Keterangan
Wartawan itu mengirimkan berita ke luar negeri.	BERHASIL
Adik menimba air di sumur.	BERHASIL
Bapak menyimpan uang di Bank.	BERHASIL
Beliau memperlakukan kami dengan baik.	BERHASIL
Anak-anak bermain layang-layang di halaman kampus.	BERHASIL
Dia memotong kue itu dengan garpu.	BERHASIL
Nenek membaca komik di kamar.	BERHASIL

Cuplikan hasil pengujian sistem secara keseluruhan pada pola kalimat SPOK ditampilkan pada Tabel VII. Kalimat diambil dari buku Jenis Kalimat dalam Bahasa Indonesia [21] dengan jumlah tiga kalimat dan tujuh belas kata yang diperiksa. Lalu, kalimat diambil dari buku Komposisi Bahasa Indonesia untuk Mahasiswa Nonjurusan Bahasa [17] dengan jumlah lima kalimat dan 28 kata yang diperiksa. Kalimat juga diambil dari buku Tatabahasa Baku Bahasa Indonesia [22] dengan jumlah dua kalimat dan sebelas kata yang diperiksa. Tambahan kalimat diambil dari buku Sintaksis Bahasa Indonesia (Pendekatan Proses) [18] dengan jumlah satu kalimat dan lima kata yang diperiksa. Untuk pola kalimat SPOK, semua masukan kalimat berhasil diperiksa dengan benar.

## V. KESIMPULAN

Berdasarkan hasil pengujian sistem dan analisis algoritme pemeriksa pola kalimat bahasa Indonesia, diperoleh beberapa kesimpulan. Pertama, kelengkapan kata dalam kamus yang digunakan memengaruhi hasil pembentukan jenis kata. Misalnya, belum terdapatnya nama orang, tempat, dan benda lain menyebabkan sistem tidak mampu mengenali pola kalimat secara lengkap. Selain itu, jumlah aturan yang disediakan dalam pembentukan jenis kata memengaruhi hasil pembentukannya. Dalam sistem ini hanya terdapat lima aturan produksi dan empat pola kalimat. Kebanyakan kesalahan pengenalan pola disebabkan oleh sistem yang tidak dapat mengidentifikasi pola kalimat.

Untuk pengujian pengenalan pola kalimat SP, diperoleh tingkat keberhasilan 96,2% dari sejumlah 53 kalimat dan 162 kata. Sedangkan untuk pola kalimat SPO, SPK, dan SPOK, masing-masing sebesar 100% dari sejumlah 22 kalimat dan 78 kata, 85% dari sejumlah dua puluh kalimat dan 85 kata, dan 100% dari sejumlah sebelas kalimat dan 61 kata. Akan tetapi, perlu diperhatikan bahwa jumlah total data uji kalimat dan

kata yang digunakan relatif kecil, yaitu seratus enam kalimat dan 188 kata.

Dibandingkan dengan hasil perbandingan algoritme LCP dan CYK, dengan algoritme LCP digunakan untuk memeriksa delapan pola kalimat baku bahasa Indonesia, baru 60%. Maka, hasil pengujian sistem pemeriksa empat pola kalimat bahasa Indonesia ini, dengan akurasi 96,2%, secara umum dapat dikatakan memperoleh hasil yang lebih baik.

Diharapkan sistem pemeriksa pola kalimat pada tata bahasa Indonesia ini dapat membantu peneliti dan praktisi bidang ilmu/teknik komputer dan juga linguistik. Disarankan untuk menggunakan Kamus Besar Bahasa Indonesia (KBBI) secara *online* untuk menyediakan data pengecek kata dan jenis kata yang lebih lengkap. Selain itu, diperlukan untuk menetapkan aturan tata bahasa yang lebih luas agar dapat mengenali berbagai macam pola kalimat lain.

## REFERENSI

- [1] J. Suciadi, “Studi Analisis Metode-Metode Parsing dan Interpretasi Semantik Pada Natural Language Processing,” *Jurnal Informatika*, Vol. 2, No. 1, hal. 13-22, 2001.
- [2] E. Setiawan (2018) “Kamus Besar Bahasa Indonesia (KBBI),” [Online] <http://kbbi.web.id/tata%20bahasa>, tanggal akses: 2-Nov-2018.
- [3] M. Ramlan, *Sintaksis*, Yogyakarta, Indonesia: CV Karyono, 2001.
- [4] K.H.K.Y.S.D. Nastiti, E.B. Cahyono, dan W.A. Kusuma, “Analisis Sintaks Tata Bahasa dengan Menggunakan Algoritma *Left-Corner Parsing* pada Aplikasi Terjemahan Bahasa Indonesia - Jawa (Ngoko), Bahasa Jawa (Ngoko) - Indonesia,” Skripsi, Universitas Muhammadiyah Malang, Malang, Indonesia, 2013.
- [5] F. Khusaini dan F. Kurniawan, “Implementasi *Left-Corner Parsing* untuk Pembelajaran Grammar Bahasa Inggris pada Game 3D Adventure ‘Go To London’,” *Matics Journal*, Vol. 5, No. 3, hal. 166-173, 2013.
- [6] L.A. Muharrom dan S. Sulaiha, “Pemeriksa Tata Bahasa pada Kalimat Bahasa Inggris Menggunakan Algoritma *Left-Corner Parsing*,” Skripsi, Univ. Muhammadiyah Jember, Jember, Indonesia, 2016.
- [7] S. Susanti, “Analisis Perbandingan Algoritma *Left-Corner Parsing* (LCP) dan *Cocke-Younger-Kasami* (CYK) untuk Memeriksa Pola Kalimat Baku Bahasa Indonesia,” Skripsi, Universitas Komputer Indonesia, Bandung, Indonesia, 2016.
- [8] O. Rante, D. Paseru, dan I. Saputro, “Aplikasi Pemeriksaan Tata Bahasa dalam Kalimat Bahasa Indonesia menggunakan *Left-Corner Parsing*,” *Jurnal Elektro Unika Atma Jaya*, Vol. 9, No. 1, hal. 35-44, 2016.
- [9] E. Istiqamah, D. Soyusiawaty, “Penerapan metode *Left-Corner Parsing* dalam aplikasi terjemahan bahasa Indonesia ke bahasa Bima,” Seminar Nasional Aplikasi Teknologi Informasi, 2017, hal. D.1-8.
- [10] A.A. Suryani, I. Ariesianti, B.W. Yohanes, M. Subair, S.D. Budiwati, dan B.S. Rintyarna, “Enriching English into Sundanese and Javanese Translation List Using Pivot Language,” *International Conference on Information & Communication Technology and Systems (ICTS)*, 2016, hal. 167-171.
- [11] B.W. Yohanes, T. Robert, dan S. Nugroho, “Sistem Penerjemah Bahasa Jawa-Aksara Jawa berbasis Finite State Automata,” *JNTETI*, Vol. 6, No. 2, hal. 127-132, 2017.
- [12] B.W. Yohanes, Handoko, dan H.K. Wardana, “Focused Crawler Optimization Using Genetic Algorithm,” *TELKOMNIKA*, Vol. 9, No. 3, hal. 403-410, 2011.
- [13] L. Kassner (2007) “*Left-Corner Parsing*, Computational Linguistics II: Parsing”, [Online], <https://www.english-linguistics.de/fr/teaching/ws06-07/cl2/slides/LeftCorner-laura.pdf>, tanggal akses: 2-Nov-2018.
- [14] L. Agusta, “Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia,” *Konferensi Nasional Sistem dan Informatika*, 2009, hal. 196-201.

- [15] M.L. Rozi, E. Rachmawati, A. Romadhony, "Implementasi dan Analisis Perbandingan Algoritma Stemming Nazief & Adriani dengan Algoritma Stemming Vega dalam Information Retrieval System," Skripsi, Universitas Telkom, Bandung, Indonesia, 2013.
- [16] M. Muslich, *Tatabentuk Bahasa Indonesia*, Jakarta, Indonesia: Bumi Angkasa, 2010.
- [17] L. Finoza, "Komposisi Bahasa Indonesia untuk Mahasiswa Nonjurusan Bahasa," Jakarta, Indonesia: Diksi Insan Mulia, 2009.
- [18] R.A. Sukamto, "Penguraian Bahasa Indonesia dengan Menggunakan Pengurai Collins," Tesis, Institut Teknologi Bandung, 2009.
- [19] J. Asian, "Effective Techniques For Indonesian Text Retrieval," Tesis Ph.D, RMIT University, Melbourne, Australia, 2007.
- [20] (2014) "Algoritma Stemming (Pencarian Kata Dasar) Nazief dan Adriani untuk Bahasa Indonesia di C# (Csharp)," [Online] <http://www.csharp-indonesia.com/2014/07/algoritma-stemming-pencarian-kata-dasar.html>, tanggal akses: 5-Jun-2018.
- [21] I.B. Putrayasa, *Jenis Kalimat dalam Bahasa Indonesia*, Bandung, Indonesia: Refika Aditama, 2009.
- [22] A.M. Moeliono, *Tatabahasa Baku Bahasa Indonesia*, Jakarta, Indonesia: Balai Pustaka, 1988.
- [23] A. Chaer, *Sintaksis Bahasa Indonesia (Pendekatan Proses)*, Jakarta, Indonesia: PT Asdi Mahasatya, 2009.