Analisis Perbandingan Algoritma LCP (Left-Corner-Parsing) Dan Algoritma CYK (Cocke-Younger-Kasami) Untuk Memeriksa Pola Kalimat Baku Bahasa Indonesia

Sri Susanti[1]

[1] Teknik Informatika - Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : srisusanti237@yahoo.com[1]

ABSTRAK

Seorang penulis harus memperhatikan pola kalimat dalam bahasa tulisan, karena pola kalimat akan berpengaruh dalam proses penyampaian dan penerima pesan. Berdasarkan pola dasarnya, bahasa indonesia mengenal beberapa pola kalimat (1) S-P, (2) S-P-O, (3) S-P-Pel, (4) S-P-K, (5) S-P-O-Pel, (6) S-P-O-Pel-K, (7) S-P-O-K, dan (8) S-P-Pel-K. Hal inilah yang mendasari proses penulisan perlu memperhatikan pola kalimat, sehingga perlu diadakan penelitian tentang pemeriksaan pola kalimat.

Algoritma yang digunakan dalam memeriksa kalimat sangat banyak macamnya, dua diantaranya algoritma LCP (*Left Corner Parsing*) dan algoritma CYK (*Cocke-Younger-Kasami*). Algoritma LCP (*Left Corner Parsing*) merupakan algoritma gabungan dari *Top down Parsing* dan *Bottom up Parsing.* Proses dari algoritma LCP (*Left Corner Parsing*) dimulai secara *Bottom Up* dan diakhiri secara *Top Down.* Algoritma CYK menggunakan struktur array dua dimensi dalam pengecekkannya. Setiap kata diperiksa kelompok katanya, kemudian dilakukan pengecekkan struktur kalimatnya. Dari perbedaan cara kerja algoritma, maka akan dipelajari dan dibandingkan kemampuan masing-masing algoritma dalam memeriksa pola kalimat baku bahasa Indonesia. Kedua algoritma tersebut dilakukan analisis sehingga diketahui tingkat akurasi dari masing-masing algoritma dalam memeriksa pola kalimat baku.

Hasil pengujian pemeriksaan pola kalimat baku didapatkan tingkat akurasi dari algoritma CYK (*Cocke-Younger-Kasami*) sebesar 65% dengan laju error 0.35 dan algoritma *left corner parsing* sebesar 60% dengan laju error 0.40. Namun tingkat akurasi yang didapat belum cukup besar karena kamus POS Tag yang digunakan masih terdapat kekurangan yaitu masih banyak kata yang tidak dapat dikenali.

Kata kunci : Algoritma, Left Corner Parsing, Cocke-Younger-Kasami, Pola Kalimat, Bahasa Indonesia.

1. PENDAHULUAN

Suatu tata bahasa harus memenuhi kriteria ilmiah atau empiris. Empiris yaitu tata bahasa harus bisa dibuktikan secara ilmiah oleh setiap orang, disetiap tempat dan pada setiap waktu. Pengajaran fungsi kalimat merupakan pengetahuan standar yang diajarkan dalam kelas-kelas bahasa [1]. Kalimat yang digunakan berasal dari beberapa pola kalimat dasar yang dikembangkan sesuai dengan kebutuhan. Pengembangan pola kalimat tersebut harus didasarkan pada kaidah tata bahasa formal yang berlaku, sehingga ada hubungan timbal balik yang baik dan jelas di antara kata atau kelompok kata yang membentuk kalimat tersebut.

Sebelumnya ada penelitian yang pernah dilakukan untuk memeriksa tata bahasa dengan menggunakan algoritma parsing. Penelitian yang pernah dilakukan yaitu berjudul "Implemetasi *Left Corner Parsing* Pada Perancangan Aplikasi Pemeriksaan Tata Bahasa Dalam Kalimat Bahasa Indonesia" oleh Vita Meriati Pandiangan. Penelitian yang dilakukan hanya untuk menangani kesalahan bahasa Indonesia pada buku wacana untuk anak sekolah dasar (SD). Pengecekkan yang dilakukan berdasarkan pada kesalahan morfologi, dan kalimat yang akan diperiksa sudah ditentukan yaitu berupa kalimat SPOK yang merupakan CFG (*context free grammars*). Hasil dari penelitian yang telah dilakukan mengatakan bahwa algoritma *Left Corner Parsing* dalam pemeriksaan tata bahasa dapat berjalan dengan baik, yaitu dapat menghasilkan pemeriksaan pola kalimat yang sesuai dengan pola yang telah dibuat dan disimpan pada *database* [2]. Namun pada penelitian tersebut tidak dijelaskan secara spesifik ukuran yang digunakan untuk menilai sejauh mana implementasi algoritma *left corner parsing* berhasil dalam pemeriksaan tata bahasa Indonesia baku.

Penelitian lain mengenai pemeriksaan tata bahasa yang pernah dilakukan yaitu dengan menggunakan algoritma CYK (Cocke-Younger-Kasami) yang berjudul "Aplikasi Program Dinamis dalam Algoritma Cocke-Younger-Kasami" oleh Inas Luthfi. Algoritma CYK pada penelitian tersebut melakukan pengecekan *validitas* sebuah untai dimensi. Algoritma CYK merupakan penerapan program dinamis yang cukup mudah diterapkan

meski memiliki kompleksitas kasus terburuk *O*(n3). Hasil dari penelitian yang telah dilakukan mengatakan bahwa algoritma CYK cukup efisien dalam hal mengenali aturan-aturan yang telah dibuat dalam bentuk CNF (*Chomsky Normal Form*) [3].

Dari penelitian-penelitian di atas mengatakan bahwa masing-masing algoritma berjalan dengan baik dalam pemeriksaan tata bahasa. Namun pada penelitian-penelitian yang telah dilakukan tidak terdapat parameter atau ukuran penilaian untuk melihat sejauh mana implementasi dari kedua algoritma digunakan dalam pemeriksaan tata bahasa Indonesia. Algoritma *Left Corner Parsing* baik dalam pemeriksaan tata bahasa bebas konteks (CFG), sedangkan Cocke-Younger-Kasami (CYK) dapat mengenali kalimat dengan baik sesuai aturan pola kalimat dalam bentuk CNF (*Chomsky Normal Form*)**.** Dengan melakukan analisis perbandingan dari kedua algoritma maka akan dapat diketahui cara kerja dan akurasi dalam ketepatan pemeriksaan pola kalimat baku dari masing-masing algoritma. Agar selanjutnya algoritma yang lebih baik dapat digunakan pada pemeriksaan pola kalimat baku bahasa Indonesia.

Berdasarkan permasalahan yang telah dipaparkan, maka dalam penelitian ini dilakukan perbandingan antara algoritma *Left Corner Parsing* dan algoritma CYK (*Cocke-Younger-Kasami*) dalam mengenali pola kalimat baku bahasa Indonesia untuk menganalisis apakah algoritma *Left Corner Parsing* lebih baik dari algoritma CYK (*Cocke-Younger-Kasami*) dari segi tingkat keakuratan dalam memeriksa pola kalimat baku.

1.1 Sintaksis Bahasa Indonesia

Sintaksis merupakan cabang ilmu bahasa yang membicarakan penataan dan pengaturan kata-kata ke dalam satuan yang lebih besar, yang disebut satuan-satuan sintaksis, yakni kata, frase, klausa, kalimat, dan wacana.

1.1.1 Fungsi Sintaksis

Fungsi sintaksis adalah semacam kotak-kotak dalam struktur sintaksis yang kedalamnya diisikan kategori-kategori tertentu. Kotak-kotak itu bernama subjek (S), predikat (P), objek (O), komplemen (Kom), dan keterangan (Ket). Namun di dalam praktik berbahasa urutannya tidak sama. Secara formal fungsi S dan P harus selalu adadalam setiap klausa karena keduanya saling berkaitan dalam hal ini bisa dikatakan, bahwa S adalah bagian klausa yang menandai apa yang dinyatakan oleh pembicaraan mengenai S.

Objek adalah bagian dari verba yang menjadi predikat dalam klausa. Kehadirannya sangat ditentukan oleh ketransitifan verba tersebut. Artinya jika verba bersifat transitif maka objek akan muncul, tetapi kalau verbanya bersifat tak transitif maka objek tidak akan ada. Terdapat dua macam objek yaitu objek afektif dan objek efektif. Objek afektif adalah objek yang bukan merupakan hasil perbuatan predikat. Sebaliknya objek efektif adalah objek yang merupakan hasil perbuatan predikat.

Komplemen (komp) atau pelengkap adalah bagian dari P verba yang menjadikan P itu menjadi pelengkap. Kedudukannya mirip dengan O, hanya perbedanya jika O keberadaannya ditentukan oleh sifat verbanya yang transitif. Sedangkan komp keberadaannya bukan ditentukan oleh faktor ketransitifan, melainkan oleh faktor keharusan untuk melengkapi P. Unsur S, P, O dan komplemen merupakan inti klausa, sedangkan unsur keterangan merupakan bagian luar inti klausa. Hal tersebut karena kedudukan keterangan di dalam klausa lebih fleksibel, artinya dapat berada pada awal klausa maupun pada akhir klausa [6].

1.1.2 Kategori Sintaksis

Kategori sintaksis merupakan jenis atau tipe kata atau frase yang menjadi pengisi fungsi-fungsi sintaksis. Kategori sintaksis berkenaan dengan istilah nomina (N), verba (V), ajektifa (A), adverbia (Adv), numeralia (Num), preposisi (Prep), konjungsi (Konj), dan pronominal (Pron). Dalam hal ini N, V, dan A merupakan kategori utama, sedangkan yang lain merupakan kategori tambahan. Pengisi fungsi sintaksis dapat berupa kata dapat pula berupa frase, sehingga di samping ada kata nomina ada pula frase nomina (FN), di samping kata verba ada pula frase verba (FV), dan di samping ada kata ajektifa ada pula frase ajektifa (FA). Selain itu di samping ada kata berkategori adverbial ada pula frase adverbia (FAdv), di samping kata berkategori numeralia ada pula frase numeral (FNum), dan di samping kata berkategori preposisi ada pula frase preposisional (FProp).

Secara formal kategori N atau FN mengisi fungsi S dan atau O pada klausa verba. Bisa juga mengisi fungsi P pada klausa nominal. Kategori V atau FV secara formal mengisi fungsi P pada klausa verba, dan kategori A atau FA mengisi fungsi P pada klausa ajektifa [6]. Maka, secara formal pengisi fungsi-fungsi sintaksis dapat dilihat pada Tabel 1.1 berikut:

Tabel 1.1 Pengisi Fungsi-Fungsi Sintaksis

S	P	O
N	N	N
FN	FN	FN
	V	
	FV	
	A	
	FA	

1.1.3 Frase

Frase adalah satuan sintaksis yang tersusun dari dua buah kata atau lebih, yang di dalam klausa menduduki fungsi-fungsi sintaksis. Dilihat dari kedudukan kedua unsurnya, dibedakan adanya frase koordinatif dan frase subordinatif. Frase koordinatif yaitu yang kedudukan kedua unsurnya sederajat, sedangkan frase subordinatif yaitu yang kedudukan kedua unsurnya tidak sederajat. Ada yang berkedudukan sebagai unsur atasan yang disebut inti frase dan ada yang berkedudukan sebagai bawahan yang disebut sebagai tambahan penjelas frase [6].
Frase dibagi menjadi beberapa kelompok:

1. Frase nomina koordinatif (FNK)
2. Frase nomina subordinatif (FNS)
3. Frase verba koordinatif (FVK)
4. Frase verba subordinatif (FVS)
5. Frase ajektifa koordinatif (FAK)
6. Frase ajektifa subordinatif (FAS)
7. Frase preposisional (Fprep)

1.1.4 Klausa

Klausa adalah satuan sintaksis yang bersifat prediktif. Artinya di dalam satuan atau konstruksi itu terdapat sebuah predikat, bila di dalam satuan tidak terdapat predikat, maka satuan itu bukan sebuah klausa. Kedudukan predikat ini sangat penting, sebab jenis dan kategori dari predikat itulah yang menentukan hadirnya fungsi subjek (S), fungsi objek (O), fungsi pelengkap, dan sebagainya [6]. Dalam analisis fungsional klausa dianalisis berdasarkan fungsi unsur-unsurnya menjadi S, P, O, PEL, KET, dan dalam analisis kategorial telah dijelaskan bahwa pengisi fungsi S terdiri dari N, fungsi P terdiri dari N, V, Num, FP, fungsi O terdiri dari N, fungsi PEL terdiri dari N, V, Num, dan fungsi KET terdiri dari FP, N [11].

Berdasarkan kategori yang mengisi fungsi P dapat dibedakan menjadi:

1. Klausa verba
2. Klausa nomina
3. Klausa ajektifa
4. Klausa preposisional
5. Klausa numerial

1.1.5 Kalimat

Kalimat adalah satuan sintaksis yang disusun dari konstituen dasar, yang biasanya berupa klausa, dilengkapi dengan konjungsi bila diperlukan, serta disertai dengan intonasi *final*. Intonasi *final* merupakan syarat penting dalam pembentukan sebuah kalimat dapat berupa intonasi deklaratif (ang dalam bahasa ragam tulis diberi tanda titik), intonasi interogatif (yang dalam bahasa ragam tulis diberi tanda tanya), intonasi imperatif (yang dalam bahasa ragam tulis diberi tanda seru), dan intonasi interjektif (yang dalam bahasa ragam tulis diberi tanda seru). Tanpa intonasi *final* ini sebuah klausa tidak akan menjadi sebuah kalimat.

1.2 *Chomsky Normal Form* (CNF)

CNF merupakan salah satu bentuk normal yang sangat berguna untuk CFG yang telah mengalami penyederhanaan. Aturan produksi dalam bentuk CNF ruas kanannya tepat berupa sebuah terminal atau dua variabel. Dalam CNF, ruas kanan hanya boleh berupa sebuah simbol terminal atau dua buah simbol variable. Jika terdapat lebih dari satu simbol terminal maka harus dilakukan penggantian dan juga jika terdapat lebih dari dua buah simbol variable maka harus dilakukan perubahan.

Contoh aturan produksi dapat dilihat sebagai berikut:

$A \rightarrow BC$
$A \rightarrow b$
$B \rightarrow a$
$C \rightarrow BA \mid d$

1.3 POS Tag

POS (*Part-of-Speech*) Tag merupakan suatu cara pengkategorian kelas kata, seperti kata benda, kata kerja, kata sifat, dan lain-lain. POS Tagger merupakan sebuah aplikasi yang mampu melakukan proses anotasi *part-of-Speech* tag untuk setiap kata di dalam dokumen secara otomatis. POS Tag yang digunakan sebagai bantuan dalam mengenali token/tag setiap kata diambil dari POS Tag Indonesia yang dibuat oleh Arawinda Dinakaramani, Fam Rashel, Andry Luthfi, dan Ruli Manurung [8].

POS Tag akan mengenali kata mana yang termasuk kata benda, kata kerja, kata sifat, kata keterangan, kata depan, kata sambung, kata ganti benda, dan kata bilangan. POS Tag ini menggunakan pendekatan *Rule-Based* berdasarkan aturan tata bahasa Indonesia. Pertama-tama, POS Tag akan melakukan tokenisasi terhadap teks menggunakan kamus bahasa Indonesia. Selanjutnya kata-kata yang termasuk ke dalam jenis *closed-class word* diproses. Lalu setiap kata yang ambigu diproses menggunakan aturan-aturan yang sudah didefinisikan untuk menemukan kelas kata yang tepat.

1.4 Algoritma Left Corner Parsing (LCP)

Algoritma *left corner parsing* merupakan gabungan dari *top-down parsing* dan *bottom-up parsing* yaitu merupakan strategi yang menggunakan data secara *bottom-up parsing* dan prediksi dari *top-down parsing.* Cara kerja algoritma ini yaitu dengan mula-mula menerima sebuah kata, menentukan jenis *constituent* apa yang dimulai dengan jenis kata tersebut. Kemudian akan dilakukan proses *parsing* terhadap sisa dari *constituent* secara *top-down.* Dengan demikian proses *parsing* pada algoritma *left corner parsing* dimulai secara *bottom-up* dan diakhiri secara *top-down* [9].

Dalam proses pemeriksaan algoritma *left corner parsing* operasi yang digunakan adalah *reduce*, *move*, dan *remove.* Operasi *reduce* digunakan untuk

mengetahui ruas kiri aturan produksi dari suatu terminal atau variabel. Untuk operasi *move* digunakan ketika ruas paling kiri dari *categories* terdapat simbol $, dimana simbol tersebut merupakan pembatas. Operasi *move* yaitu perpindahan dari *stack constituent* ke *stack categories.* Sedangkan operasi *remove* yaitu operasi yang digunakan untuk menghapus jika terdapat variabel atau terminal yang sama pada *stack sentence* dan *categories* (*stack* di pop).

Sebagai contoh terdapat suatu aturan produksi sebagai berikut:

S → ASB | d
A → a
B → b

Langkah-langkah dari cara kerja algoritma *left corner parsing* dapat dilihat pada tabel 1.2 sebagai berikut:

Tabel 1.2 Langkah-langkah *Left Corner Parsing*

Sentence	Categories	Constituent	Operation
adb	S	ε	
db	$S	A	reduce A →a
Adb	S	ε	move
db	SB$S	S	reduce S →ASB
b	SBS	SS	reduce S →d
Sb	SB$S	S	move
b	B$S	S	remove
ε	BS	BS	reduce B →b
B	B$S	S	move
ε	S	S	remove
S	S	ε	move
ε	ε	ε	remove

1.5 Algoritma *Cocke-Younger-Kasami* (CYK)

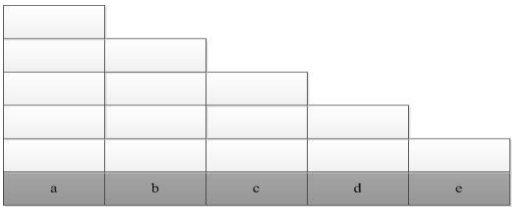
Algoritma CYK merupakan algoritma *parsing* dan keanggotaan untuk tata bahasa bebas konteks. Algoritma CYK diciptakan oleh J. Cocke, DH. Younger, dan T. Kasami. Syarat untuk penggunaan algoritma CYK adalah tata bahasa harus berada dalam bentuk CNF. Tujuan dari algoritma ini adalah untuk menunjukkan apakah suatu *string* dapat diperoleh dari suatu tata bahasa [7].

Proses *parsing* untai dengan algoritma CYK menggunakan struktur data sebuah *array* dua dimensi dengan jumlah baris dan kolom n x n. Nilai n ditentukan dari jumlah kata dalam suatu kalimat. Proses *parsing* dari algoritma CYK memanfaatkan hasil *parsing* sebelumnya untuk menentukan apakah proses yang sedang berlangsung dapat diterima atau tidak.

Sebagai contoh terdapat suatu aturan produksi dengan variabel awal s.

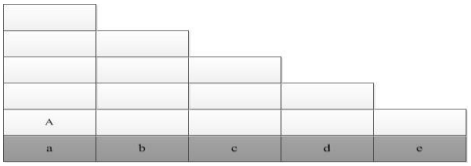
S→AB | EC
F→CA
G→CD
H→EF
A→a
B→b
C→c
D→d
E→e

String yang akan diperiksa misalnya “abcde”. Langkah pertama yang dilakukan adalah dengan membuat tabel dua dimensi seperti gambar 1.1.



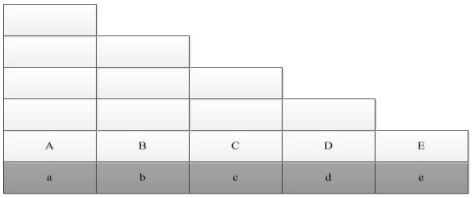
Gambar 1.1 Pembentukan Tabel *Parsing* CYK

Setiap kata yang ada pada suatu *string* ditempatkan pada masing-masing blok seperti gambar 1.1. Setiap blok yang kosong diisi berdasarkan variabel yang sesuai dengan aturan produksi dari CNF yang sudah dibentuk sebelumnya. Sebagai contoh pada baris pertama dan kolom pertama diisi dengan variabel yang dapat menghasilkan huruf a, yaitu variabel A. Hasilnya dapat dilihat pada gambar 1.2.



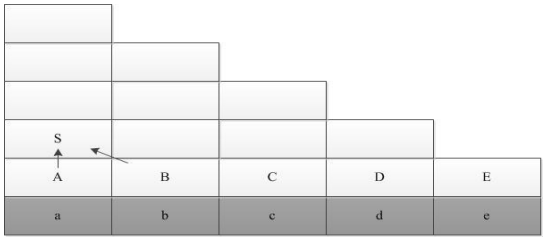
Gambar 1.2 Pengisian Tabel dengan Variabel A

Blok selanjutnya diisi dengan langkah yang sama, sehingga dapat dilihat baris pertama isiannya seperti gambar 1.3.



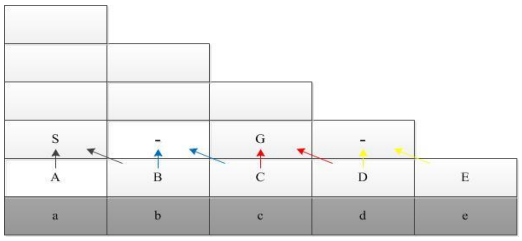
Gambar 1.3 Pengisian Tabel Baris Pertama

Pada baris selanjutnya cara pengisian dilakukan dengan melihat variabel pada dua blok, kemudian dilihat variabel mana yang menghasilkan kedua variabel tersebut. Sebagai contoh baris kedua kolom pertama diisi dengan melihat variabel pada blok pertama dan blok dua, yaitu variabel A dan B yang termasuk ke dalam variabel S, sehingga cara pengisiannya terlihat seperti pada gambar 1.4.



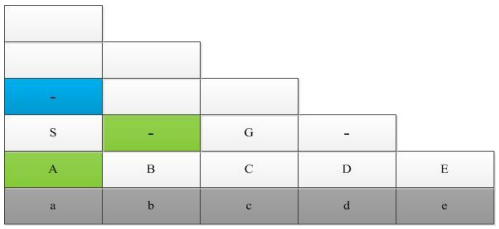
Gambar 1.4 Pengisian Tabel dengan Variabel B

Blok selanjutnya diisi dengan langkah yang sama sesuai dengan blok yang akan diisi. Apabila pengisian dilakukan pada blok kedua, variabel diambil dari kolom kedua dan ketiga. Untuk lebih jelasnya dapat dilihat pada gambar 1.5.



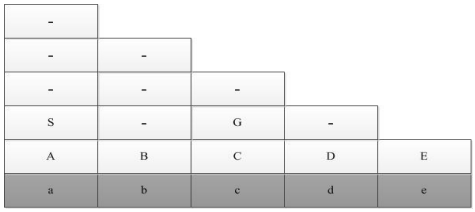
Gambar 1.5 Pengisian Tabel Baris Kedua

Pengisian pada baris ketiga dan selanjutnya hampir sama seperti baris kedua, namun terdapat perbedaan dari cara pengecekkan yang dilakukan. Pengisian baris ketiga dilakukan dari baris pertama hingga sebelum baris ketiga (n-1). Cara pengecekkan dapat dilihat seperti gambar 1.6.



Gambar 1.6 Cara Pengecekan Langkah Pertama

Warna biru menunjukkan blok yang akan diisi, sedangkan warna hijau merupakan blok yang dicek variabelnya. Pada kedua gambar diatas dapat dilihat untuk baris ketiga cara pengecekkan dilakukan secara diagonal dan vertikal. Cara pengisian selanjutnya sama seperti gambar 1.6. Hasil akhir dari pengecekan *string* diatas dapat dilihat pada gambar 1.7.



Gambar 1.7 Hasil *Parsing String* dengan Algoritma CYK

2. ISI PENELITIAN

2.1 Analisis Data Masukan

Data masukan untuk simulator berupa dokumen artikel berita dengan format .docx. Setiap kalimat dalam dokumen diperiksa satu persatu dan setiap kata yang ada pada kalimat digolongkan ke dalam kelompok kata. Pengelompokkan kata diambil dari kamus menggunakan POS Tag Indonesia.

2.2 Analisis Sintaksis untuk Aturan Produksi

Berdasarkan pembahasan tentang sintaksis bahasa Indonesia dan pengelompokkan kata dengan POS Tag Indonesia sebelumnya, maka selanjutnya pada bagian ini akan dijelaskan bagaimana membuat aturan produksi dalam bentuk CNF. Sebelumnya, seluruh penamaan kelompok kata harus diubah ke dalam aturan POS Tag agar tidak sulit dalam mengecek aturan produksinya.
Berikut perubahannya dapat dilihat pada tabel 2.1.

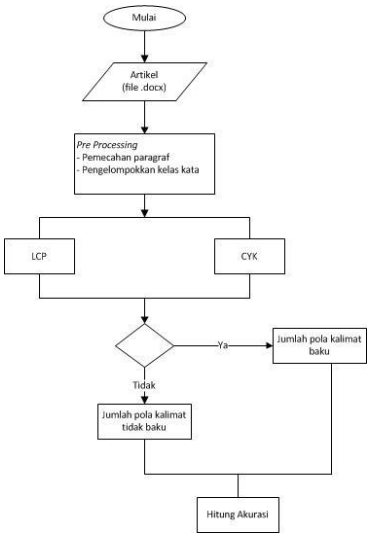
Tabel 2.1 Penamaan Kelompok Kata ke Dalam POS Tag

Nama Kelompok Kata	Diubah Menjadi
Noun / Kata Benda (N)	NN, NNP (FW), NND, PRP, atau WH
Adjective / Kata Sifat (A)	JJ
Verb / Kata Kerja (V)	VB
Adverb / Kata Keterangan (Adv)	RB
Numeralia (Num)	OD dan CD
Preposisi (P)	IN
Demonstrative (Dem)	PR
Konjungsi Koordinatif	CC
Artikula	DT
Verba Bantu	MD
Kata Ingkar	NEG
Partikel	RP
Konjungsi Subordinatif	SC
Simbol	SYM

2.3 Analisis Pemrosesan

Analisis pemrosesan merupakan proses awal yang menggambarkan alur sistem yang akan

dibangun. Tahapan awal sistem adalah pengecekkan data masukan. Data masukan yang diperbolehkan merupakan dokumen berformat .docx. Kemudian sistem akan melakukan tahap *pre processing*, yaitu dokumen yang telah dimasukkan akan dipecah menjadi satuan terkecil (kata). Lalu kata tersebut akan dikelompokkan ke dalam kelas kata berdasarkan POS Tag, selanjutnya akan dilakukan proses pemeriksaan pola kalimat menggunakan kedua algoritma. Setelah itu akan keluar hasil dari pemeriksaan algoritma, yang nantinya akan dihitung tingkat akurasinya. Perancangan proses dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Flowchart* Sistem

2.4 Analisis Metode

Analisis metode merupakan analisis penerapan metode atau algoritma dalam menyelesaikan suatu permasalahan yang ada. Metode dalam penelitian ini adalah algoritma *left corner parsing* dan algoritma CYK (*Cocke-Younger-Kasami*).

2.4.1 Tahapan Algoritma *Left Corner Parsing*

Terdapat rangkaian proses pada algoritma *left corner parsing* yang dilakukan dalam pemeriksaan pola kalimat baku bahasa Indonesia, proses tersebut harus dilakukan secara terurut, setiap tahap menghasilkan suatu nilai sebagai hasil proses, kemudian akan diolah dan digunakan kembali untuk tahap berikutnya, proses algoritma *left corner parsing* tersebut meliputi:

1. Pemecahan paragraf

Pada tahap awal dokumen artikel berita yang akan diperiksa pola kalimat baku akan dipecah paragrafnya menjadi kalimat kemudian kalimat menjadi kata. Pemecahan paragraf mejadi kalimat menggunakan *regex*. *Regex* yaitu sebuah teknik yang digunakan untuk mencocokkan *string* teks, seperti karakter, kata-kata, atau pola karakter. Tahap ini akan diketahui ada berapa jumlah kalimat dalam satu paragraf artikel berita yang diuji. Titik menjadi tanda antar kalimat dan menjadi penanda akhir pada kalimat. Namun, tidak semua titik merupakan penanda akhir sebuah kalimat. Pola yang dibentuk *regex* adalah [karakter] [tanda_titik] [spasi]. Tahap awal pengecekan dilakukan dengan menelusuri karakter demi karakter yang ada pada paragraf artikel berita *online*.

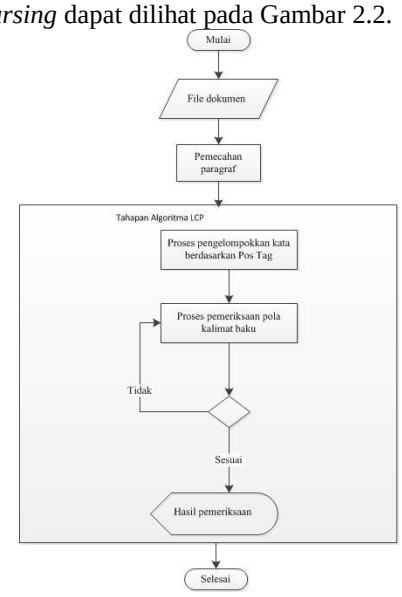
2. Pengelompokkan kelas kata

Pada tahap ini setiap kata dikelompokkan sesuai dengan kamus yang diambil berdasarkan aturan dari POS Tag. Tabel POS Tag tersebut dapat dilihat pada lampiran A. Tanda titik pada akhir kalimat harus disertakan juga untuk proses pembentukan pola kalimat.

3. Proses pemeriksaan pola kalimat baku

Pada tahap ini dilakukan pemeriksaan apakah pola kalimat yang diperiksa sesuai dengan aturan produksi atau tidak.

Rangkaian proses pada tahapan algoritma *left corner parsing* dapat dilihat pada Gambar 2.2.



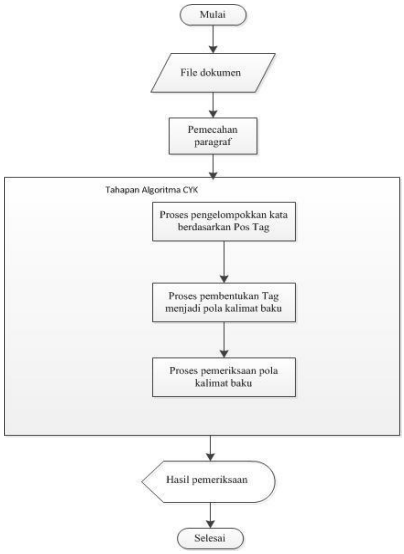
Gambar 2.2 *Flowchart* Implementasi Algoritma *left corner parsing*

2.4.2 Tahapan Algoritma CYK

Algoritma CYK (*Cocke-Younger-Kasami*) juga terdapat rangkaian proses yang dilakukan dalam pemeriksaan pola kalimat baku bahasa Indonesia, proses tersebut harus dilakukan secara terurut, setiap tahap menghasilkan suatu nilai sebagai hasil proses, kemudian akan diolah dan digunakan kembali untuk tahap berikutnya, proses algoritma CYK (*Cocke-Younger-Kasami*) hampir sama seperti tahapan pada proses algoritma *left corner parsing*. Proses algoritma CYK tersebut meliputi:

1. Pemecahan paragraf
2. Pengelompokkan kata berdasarkan POS Tag
3. Pembentukan Tag menjadi pola kalimat baku
4. Proses pemeriksaan pola kalimat baku

Rangkaian proses pada tahapan algoritma CYK (*Cocke-Younger-Kasami*) dapat dilihat pada Gambar 2.3.



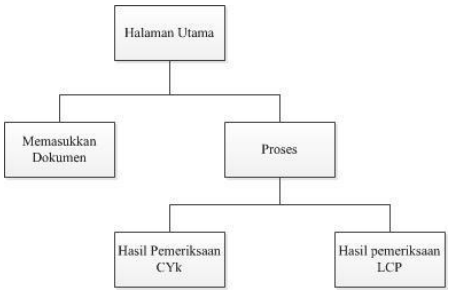
Gambar 2.3 *Flowchart* Implementasi Algoritma CYK

2.5 Perancangan Sistem

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi.

2.5.1 Perancangan Struktur Menu

Perancangan struktur menu merupakan gambar jalur pemakaian sistem sehingga sistem yang dibangun agar mudah dipahami dan mudah digunakan. Gambar perancangan struktur menu dapat dilihat pada Gambar 2.4.



Gambar 2.4 Perancangan Struktur Menu

2.5.2 Perancangan Antarmuka

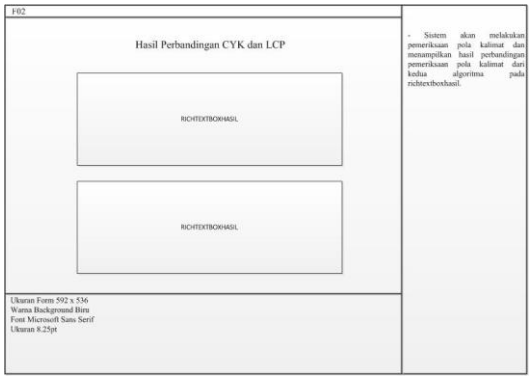
Perancangan antarmuka merupakan rancangan dalam menentukan tata letak dari tampilan sistem yang akan dibuat. Rancangan tersebut dibuat sebagai perancangan antarmuka pada sistem. Perancangan antarmuka disesuaikan dengan pengguna yang mengakses sistem tersebut, berikut adalah perancangan antarmuka untuk *form*.

1. F01-*Form* Halaman Utama Masuk Program



Gambar 2.5 *Form* Halama Utama

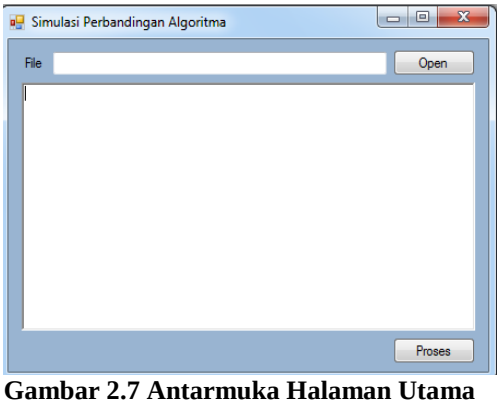
2. F02-*Form* Pemeriksaan Pola Kalimat



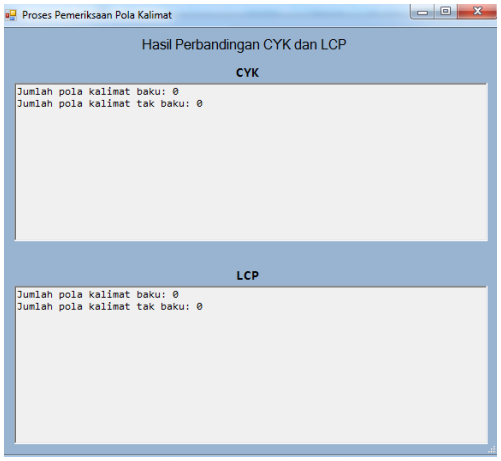
Gambar 2.6 *Form* Pemeriksaan Pola Kalimat

2.5.3 Implementasi Antarmuka

Implementasi antarmuka adalah implementasi dari perancangan antarmuka yang telah dibuat yang disesuaikan dengan navigasi beserta fungsionalitas dari halaman tersebut. Gambar 2.7 dan Gambar 2.8 merupakan tampilan antarmuka sistem untuk implementasi algoritma *left corner parsing* dan algoritma CYK untuk pemeriksaan pola kalimat baku bahasa Indonesia yang telah dibuat.



Gambar 2.7 Antarmuka Halaman Utama



Gambar 2.8 Antarmuka Halaman Hasil Pemeriksaan Pola Kalimat

3 PENUTUP

Berdasarkan hasil yang didapat dalam penelitian dan penyusunan skripsi ini serta disesuaikan dengan tujuan penelitian, maka diperoleh kesimpulan bahwa algoritma CYK dan algoritma *left corner parsing* dapat mengenali kalimat dengan baik sesuai dengan aturan produksi yang telah dibuat. Semakin banyak aturan yang dibuat maka akan semakin banyak pula pola kalimat yang dapat dikenali. Hasil pengujian pemeriksaan pola kalimat baku bahasa Indonesia menyatakan algoritma CYK memiliki tingkat akurasi sebesar 65% dengan dengan laju error sebesar 0.35, sedangkan algoritma *left corner parsing* memiliki tingkat akurasi sebesar 60% dengan dengan laju error sebesar 0.40. Nilai yang didapat belum cukup besar karena masih terdapat kekurangan diantaranya kamus POS Tag yang digunakan *offline* dan memiliki kekurangan masih banyak kata yang belum dikenali. Untuk melengkapi sebagian kata yang belum ada pada kamus POS Tag penulis menggunakan kamus besar bahasa Indonesia (KBBI).

Berdasarkan semua proses dalam membangun aplikasi ini saran-saran yang dapat diberikan dalam pengembangan aplikasi simulasi pemeriksaan pola kalimat baku bahasa indonesia adalah sebagai berikut:

1 Aplikasi simulasi ini dapat dikembangkan lagi dengan menggunakan kamus POS Tag *online* supaya mendapatkan data secara *realtime* dan tingkat akurasi yang didapat akan lebih besar.
2 Dapat menggunakan algoritma atau metode lain untuk dibandingkan, agar mendapatkan algoritma yang lebih baik lagi dalam proses pemeriksaan pola kalimat baku bahasa Indonesia.

DAFTAR PUSTAKA

[1] P. D. Iswara, "Penelitian Variasi Pola Kalimat," 24 februari 2009. [Online]. Available: http://iswara.staf.upi.edu/2009/02/24/penelitian-variasi-pola-kalimat. [Accessed 5 September 2015].

[2] . M. Pandiangan, "Implementasi Left Corner Parser Pada Perancangan Aplikasi Pemeriksaan Tata Bahasa Dalam Kalimat Bahasa Indonesia," vol. v, pp. 1-5, 2015..

[3] I. Luthfi, "Aplikasi Program Dinamis dalam Algoritma Cocke-Younger-Kasami (CYK)," pp. 1-5, 2007.

[4] N. S. Ibrahim, Penelitian dan Penilaian Pendidikan, Bandung: Sinar Baru, 1989.

[5] M. S. Rosa A. S, Rekayasa Perangkat Lunak, Bandung: BI-Obses, 2014.

[6] A. Chaer, Sintaksis Bahasa Indonesia (Pendekatan Proses), Jakarta: Rineka Cipta, 2009.

[7] F. Utdirartatmo, Teori Bahasa dan Otomata, Yogyakarta: Graha Ilmu, 2005.

[8] F. R. A. L. d. R. M. Arawinda Dinakaramani, "Pos Tag Indonesia," 20 Oktober 2014. [Online]. Available: http://bahasa.cs.ui.ac.id/postag/tagger. [Accessed 10 Oktober 2015].

[9] L. Kassner, "Left Corner Parsing," pp. 33-97, 2007.

[10] E. Prasetyo, "Klasifikasi," p. 6, 2012.