

ALGORITMA COCKE YOUNGER KASAMI UNTUK DETEKSI STRUKTUR KALIMAT DAN MEREKOMENDASIKANYA MENGGUNAKAN ALGORITMA DAMERAU LEVENSHTAIN DISTANCE

Budi Prabowo⁽¹⁾, Heru Cahya Rustamadji⁽²⁾, Yuli Fauziah⁽³⁾

Jurusan Teknik Informatika Fakultas Teknik Industri UPN "Veteran" Yogyakarta
Jl. Babarsari 2 Tambakbayan Yogyakarta

e-mail : budiprabowo666@gmail.com⁽¹⁾, herucr@upnyk.ac.id⁽²⁾, yuli.if@gmail.com⁽³⁾

Abstract

The use of standard words and sentence structure is one of the requirements in writing research reports, unwittingly writing errors can occur either in the form of typographical errors or in the structure of sentences, some of which are habits when writing short messages, the development of everyday language and the arrangement of keyboards too close. Writing errors will usually be corrected immediately after finishing writing, but to correct them, time and accuracy are needed. CYK algorithm is a membership parsing algorithm for context-free grammar that can be used to examine sentence structure while the DLD algorithm is an algorithm that is able to calculate the distance difference between two strings so that it can be used for word and sentence recommendations. The purpose of this study is to apply the CYK algorithm to detect sentence structures and DLD algorithms to recommend words and sentence structure. Examination of sentences is done by grouping each word contained in the text based on its type, the words that have been grouped are then rearranged into sentences and examined by the CYK algorithm to find out whether the sentence is true or false, if the sentence is wrong then the sentence recommendations are used using the DLD algorithm with calculates the edit distance, besides repairs to the sentence the DLD algorithm also fixes the wrong words. The test results obtained the success rate of the CYK algorithm in detecting sentence structure by 96% and the DLD algorithm in recommending words by 96%, while for recommending sentences by 88%.

Keywords : *Cocke Younger Kasami, Damerau Levenshtein Distance, Sentence Structure, Word, Bahasa Indonesia*

Abstrak

Penggunaan kata baku dan struktur kalimat merupakan salah satu syarat dalam penulisan laporan penelitian, tanpa disadari kesalahan penulisan dapat terjadi baik berupa kesalahan pengetikan maupun pada struktur kalimat, beberapa penyebabnya ialah kebiasaan saat menulis pesan pendek, berkembangnya bahasa yang digunakan sehari-hari dan susunan keyboard yang terlalu dekat. Kesalahan penulisan biasanya akan segera diperbaiki setelah selesai menulis, namun untuk memperbaikinya diperlukan waktu dan ketelitian. Algoritma CYK merupakan algoritma parsing keanggotaan untuk tata bahasa bebas konteks yang dapat digunakan untuk memeriksa struktur kalimat sedangkan algoritma DLD merupakan algoritma yang mampu menghitung jarak perbedaan dari dua buah string sehingga dapat dimanfaatkan untuk rekomendasi kata dan kalimat. Tujuan dari penelitian ini adalah menerapkan algoritma CYK untuk mendeteksi struktur kalimat dan algoritma DLD untuk merekomendasikan kata dan struktur kalimat. Pemeriksaan kalimat dilakukan dengan mengelompokkan setiap kata yang terdapat pada teks berdasarkan jenisnya, kata yang telah dikelompokkan tersebut kemudian disusun kembali kedalam bentuk kalimat dan diperiksa dengan algoritma CYK untuk mengetahui apakah kalimat tersebut benar atau salah, jika kalimat salah maka diberikan rekomendasi kalimat menggunakan algoritma DLD dengan menghitung edit distance-nya, selain perbaikan pada kalimat algoritma DLD juga melakukan perbaikan pada kata yang salah. Hasil pengujian didapatkan tingkat keberhasilan algoritma CYK dalam mendeteksi struktur kalimat sebesar 96% dan algoritma DLD dalam merekomendasikan kata sebesar 96%, sedangkan untuk merekomendasikan kalimat sebesar 88%.

Kata Kunci : *Cocke Younger Kasami, Damerau Levenshtein Distance, Struktur Kalimat, Kata, Bahasa Indonesia*

1. PENDAHULUAN

Penggunaan kata baku dan stuktur kalimat merupakan salah satu syarat dalam penulisan laporan penelitian, namun banyak diantara Mahasiswa Universitas Pembangunan Nasional “Veteran” Yogyakarta (UPN “Veteran” Yogyakarta) yang saat menulis tidak sengaja melakukan kesalahan dalam penggunaannya, seperti penulisan kata “yang” yang hanya ditulis dengan “yg” atau kata “pada” yang hanya ditulis dengan “pd” atau mungkin pada struktur kalimat seperti tertukarnya susunan subjek dengan predikat. Beberapa faktor penyebab kesalahan diantaranya kebiasaan mahasiswa saat mengetik pada pesan pendek seperti *chat* atau *Short Message Service (SMS)* menggunakan kata-kata yang disingkat, selain itu berkembangnya bahasa yang digunakan sehari-hari dikalangan generasi muda juga menjadi penyebab bergesernya bahasa Indonesia, sehingga saat menulis sebuah laporan penelitian yang seharusnya menggunakan kata baku dan struktur kalimat yang sesuai dengan kaidah tatabahasa Indonesia tergantikan oleh bahasa baru yang tidak terdapat dalam tatabahasa Indonesia namun dianggap lebih nyaman dan mudah untuk diungkapkan. Faktor lain disebabkan oleh kesalahan mekanis saat mengetik seperti letak huruf pada *keyboard* yang terlalu dekat sehingga menyulitkan pengetikan atau kesalahan jari yang tanpa sadar salah menekan tombol *keyboard*.

Kesalahan pada pengetikan biasanya akan segera diperbaiki setelah selesai menulis, jika teks yang ditulis sedikit tidak terlalu bermasalah tetapi jika teks sudah mencapai ribuan tentu akan menyulitkan sebab seseorang harus membaca teks berulang-ulang untuk mengetahui letak kesalahannya, dan tentu membutuhkan waktu dan juga ketelitian, selain itu Mahasiswa biasanya akan fokus kepada kesalahan besar terkait isi laporan penelitian sehingga proses deteksi kesalahan pengetikan maupun struktur kalimat tidak diperhatikan, padahal hal yang kecil seperti kesalahan dalam pengetikan dapat berpengaruh terhadap informasi dan pengetahuan yang disampaikan.

Aplikasi *editor* teks seperti *Microsoft Word* memang sudah memerikan fasilitas *spelling check* untuk membantu penulis dalam mengoreksi teks tetapi tidak menjamin jalanya *spelling check* untuk bahasa Indonesia, sebab tidak ada acuan atau sumber yang jelas seperti Kamus Besar Bahasa Indonesia untuk memastikan bahwa kata tersebut terdaftar dalam bahasa Indonesia. Berdasarkan permasalahan diatas untuk membantu mahasiswa dalam memeriksa kata dan struktur kalimat akan dibuat dalam sebuah program aplikasi menggunakan algoritma *Cocke Younger Kasami (CYK)* yang merupakan algoritma *parsing* dan keanggotaan untuk tatabahasa bebas konteks, algoritma *CYK* pada penelitian ini akan digunakan untuk menentukan struktur kalimat, sedangkan dalam memeriksa kata dan memberikan rekomendasi perbaikan kata dan struktur kalimat akan digunakan algoritma *Damerau Levenshtein Distance (DLD)* yang memiliki kemampuan dalam menghitung perubahan yang terjadi pada sebuah *string*.

2. METODOLOGI PENELITIAN

2.1. Analisis Masalah

Menulis merupakan kegiatan seseorang dalam mengutarakan ide dan pikirannya kedalam bentuk tulisan seperti paragraf, kalimat, atau kata. Saat menulis mahasiswa akan fokus untuk mengutarakan ide dan pemikirannya yang tanpa disadari kesalahan dalam penulisanpun terjadi. Penulisan yang salah harus segera diperbaiki agar sebuah tulisan mudah dibaca dan informasi tersampaikan kepada pembaca, dalam memperbaiki penulisan dibutuhkan ketelitian serta pengetahuan mengenai bentuk kata baku dan struktur kalimat yang sesuai dengan kaidah bahasa Indonesia. Bentuk kata baku dapat diketahui dari ada atau tidaknya kata dalam Kamus Besar Bahasa Indonesia, sedangkan struktur kalimat diketahui dari aturan pembentukan kalimat. Sebuah kalimat memiliki syarat pembentukan minimal terdapat unsur subjek dan predikat didalamnya. Pembentukan kalimat didasarkan pada aturan penggunaan subjek, predikat, objek, pelengkap, dan keterangan yang diatur dalam tatabahasa Indonesia.

2.2. Pengumpulan Data

Pengumpulan data dilakukan dengan mencari informasi tentang tatabahasa, Algoritma *Damerau Levenshtein Distance* dan *CYK* yang diperoleh dari jurnal, skripsi, dan *website* yang dapat

dipertanggungjawabkan, selain itu dilakukan pencarian daftar kata dan kelompok kata pada Kamus Besar Bahasa Indonesia (KBBI).

2.3. Rekomendasi Kata Menggunakan Algoritma DLD

Proses ini dilakukan dengan memeriksa setiap kata dan memperbaiki kata yang tidak terdapat pada kamus data. Setiap kata yang diperiksa akan dibandingkan dengan kamus data. Perbaikan kata ditentukan dengan nilai *edit distace* terkecil yang dimiliki kata.

$$d_{a,b}(i, j) = \begin{cases} \max(i, j) \\ \min \begin{cases} d_{a,b}(i-1, j) + 1 \\ d_{a,b}(i, j-1) + 1 \\ d_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \\ d_{a,b}(i-2, j-2) + 1 \end{cases} \\ \min \begin{cases} d_{a,b}(i-1, j) + 1 \\ d_{a,b}(i, j-1) + 1 \\ d_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} \end{cases}$$

Keterangan :

d = matrik

a = *string* sumber

b = *string* target

i = lebar matrik (banyaknya *sting* a)

j = panjang matrik (banyaknya *string* b)

min = minimal

max = maksimal

Pembentukan Tabel Dua Dimensi

Pembentukan tabel dua dimensi pada algoritma *DLD* dibentuk seperti pada Tabel 1.

Tabel 1. Pembentukan Tabel *DLD*

I/J		S	A	Y	A
	0	1	2	3	4
S	1				
Y	2				
A	3				

Menghitung *Edit Distance*

Setiap karakter yang terdapat pada tabel dihitung nilai *edit ditance*-nya, *edit distance* dihitung berdasarkan operasi yang dimiliki *DLD* yaitu *insert*, *delete*, substitusi, dan transposisi.

- Inserting* dan *deleting* dihitung dengan menambahkan nilai 1 (satu) pada sel sebelah kiri [0,1] dan atas [1,0] dari sel yang akan diisi [1,1], nilai yang telah ditambahkan tersebut kemudian dimasukkan kedalam sel yang menjadi pertemuan dari baris dan kolom karakter yang dibandingkan, pada pengisian awal sel yang dimaksud yaitu [1,1].
- Susbstitusi dihitung dengan menambahkan nilai 0 (nol) pada sel sebelah kiri atas [0,0] dari sel yang akan diisi [1,1] jika karakter yang dibandingkan berbeda namun jika karakternya sama maka nilai yang ditambahkan yaitu 1 (satu), nilai yang telah ditambahkan tersebut kemudian dimasukkan kedalam sel yang menjadi pertemuan dari baris dan kolom karakter yang dibandingkan, pada pengisian awal sel yang dimaksud yaitu [1,1].
- Transposisi dihitung saat sel yang akan diisi berada pada baris dan kolom lebih dari satu, saat kondisi tersebut sel ditambahkan 1 (satu).

Tabel 2. Pengisian Tabel Dua Dimensi

I/J		S	A	Y	A
	0	1	2	3	4
S	1	0	1	2	3
Y	2	2	1	1	2

A	3	3	2	1	1
---	---	---	---	---	---

Jumlah perhitungan *edit distance* dari dua buah string diatas adalah 1 (satu), hasil perhitungan tersebut diperoleh dari sel terakhir pada tabel. Hasil tersebut kemudian dijadikan nilai parameter nilai rekomendasi untuk setiap kata yang dianggap salah.

2.4. Perbaikan Struktur Kalimat Menggunakan Algoritma CYK dan Rekomendasi Struktur Kalimat menggunakan Algoritma DLD

Proses pemeriksaan kalimat dilakukan dengan mengenali pola kalimat, dalam algoritma CYK pengenalan pola kalimat dapat dilakukan dengan menganalisis sintaksis dan aturan produksi.

Sintaksis dan Aturan Produksi

Algoritma CYK membutuhkan aturan produksi yang sudah dalam bentuk CNF, dan CNF memiliki aturan produksi yang sudah melalui penyederhanaan CFG sedangkan CFG memiliki aturan produksi berbentuk $\alpha \rightarrow \beta$ dimana β tidak memiliki batasan, sehingga β disini dapat diisi dengan tatabahasa Indonesia.

Sebelum membentuk tatabahasa Indonesia kedalam aturan produksi CFG, seluruh kata dikelompokkan jenis katanya, masing-masing kelompok kata tersebut diberikan label secara acak dan menghasilkan kelompok kata.

Tabel 3. Pengelompokan Kata

Kelompok kata	Label	Keterangan
Nomina	N	Kata benda
Adjektiva	V	Kata sifat
Verba	A	Kata kerja
Adverbia	B	Pemberi keterangan pada kata
Preposisi	R	Kata depan penghubung
Demonstrativa	D	Kata tunjuk
Konjungsi Subordinatif	H	Kata hubung kalimat koordinatif
Negasi	I	Kata ingkar
Modalitas	U	Kata kerja bantu
Adjektiva/Verba Prefiks	A3	Kata kerja yang tidak memerlukan objek (menyatakan paling)
Prefiks-Verba Transitif	A4	Kata kerja memerlukan objek
Prefiks-Verba Intransitif	A5	Kata kerja yang tidak memerlukan objek
Sufiks-Nomina	T	Kata benda pengganti
Prefiks-Nomina Pemakai	N1	Kata benda yang menyatakan kepemilikan
Negasi Nomina	I1	Kata ingkar untuk predikat berkategori nomina
Konjungsi Penjelas	H1	Konjungsi yang memperjelas subjek
Konjungsi Penggambaran	A2	Kata kerja yang memiliki fungsi menggambarkan subjek
Konjungsi Koordinatif	H2	Kata hubung
Verba Deskriptif	A1	Kata kerja untuk mendeskripsikan subjek

Kelompok kata yang terbentuk diatas selanjutnya digunakan sebagai terminal simbol untuk aturan produksi. Kelompok kata tersebut adalah terminal simbol yang mewakili seluruh kata yang telah dikelompokkan. Daftar kata yang telah dikelompokkan tersebut disimpan dalam sebuah *file system* yang dipisah sesuai dengan kelompok katanya.

Kalimat dibentuk dengan minimal terdapat subjek dan predikat, untuk objek dan pelengkap berada dibelakang predikat, terkecuali objek pada kalimat intransitif dapat ditempatkan didepan predikat, sedangkan keterangan karena dapat ditempatkan dimana saja.

Sebuah aturan produksi harus memiliki keterkaitan antara unit satu dengan yang lain, guna fleksibilitas dalam penggunaan variabel simbolnya, sebuah aturan produksi dibentuk dengan susunan sebagai seperti Gambar 1.

G → SP, SO, SQ, GQ, SL, NV, NA, NA3, NA5, NE, NP, N1V, N1A, N1A3, N1A5, N1P, SK
 S → A4X, NA, NV, CP, NN, NP, N1N, N1V, N1A, N1P, KS, SK, SP, CP, NG, N1G, SG, GP, CM, KN, KN1, NK, KG, GN, GN1, N1K, GK, NA3, N1A3, SA3, NA4, N1A4, SA4, NA5, N1A5, SA5, CV, CN
 P → IE, A4E, AN, A3C, IIN, IA, IV, IA5, RN, RV, RA, RE, RC, RO, RQ, RK, H1N, H1V, H1A, H1A4, H1A2, H1A3, H1A5, H1E, H1C, H1O, H1Q, H1K, UV, UA, UN, UC, UA5, UA4, UO, UQ, UK, UE, BN, BC, BV, BA, BA4, BA5, BA3, BE, BO, BQ, BK, PL, A3E, PK, PO, PQ, PN, PN1, A1G, A1N, A1N1, A1O, A1Q, A1L, A1K, A2N, A2E, A2N1, A2O, A2Q, A2K, A2G, A2L, A2V1, A3E, A3N, A3N1, A3O, A3Q, A3K, A3G, A3L, A4N, A4N1, A4O, A4Q, A4K, A4G, A4L, A5E, A5N, A5N1, A5O, A5Q, A5K, A5L, IP, H1F, AA4, NF
 O → NT, ND, N1D, N1T, NV, NA, NQ, N1N, N1V, N1A, HN, HN1, OQ, OK, KO, KQ, HK
 Q → CN, NC, BC, BN, CN, BN, NV, H1V, QK, KQ
 K → RJ, RC, RN, BV, HG, BC, BN, HC, IT, BA, RF, HE, NH, IC, BE, GG, MK, HA, HJ, FN, KF
 L → OQ
 M → HN, HV, HA
 C → NN, NC, CN, NT, N1I, N1I1, S1I, ND, N1D, N1T, CC, NV, NA, CD
 E → VA, AA, AV, AB, AE, A4N, AN, A4V, A4B, A4E, A4C, A5V
 F → RA, RV, RN, RC, RE
 Y → H2N, H2G, H2C, H2S, H2F, H2K, H2G
 J → BA, BV, BN, BC, BE
 V1 → VN, VV, VA, VC
 X → A2N, A2C

Gambar 1. Aturan produksi

Keterangan :

G = tatabahasa

S = subjek

P = predikat

Q = pelengkap

O = objek

K = keterangan

L = gabungan objek dan pelengkap

M = gabungan konjungsi

C = frasa nomina

E = frasa verba

V1 = frasa adjektiva

F = frasa preposisi

Y = gabungan konjungsi setara dengan unsur lain

J = frasa adverbial

X = penurunan dalam penggunaan predikat yang bersifat deskriptif

Variabel unit “G” pada aturan produksi diatas memuat 19 (sembilan belas) pola kalimat meliputi SP, SPO, SPQ, SPOQ, KSP, KSPO, KSPQ, KSPOQ, SKP, SKPO, SKPQ, SKPOQ, SPK, SPKO, SPKQ, SPKOQ, SPOK, SPOKQ, SPOQK dengan masing-masing unsur kalimat seperti subjek, predikat, objek, pelengkap dan keterangan memiliki penurunan berupa kata, frasa maupun klausa pada kalimat luas koordinatif.

2.5. Pemeriksaan pola kalimat menggunakan algoritma CYK

Tahap ini melakukan pemeriksaan pada setiap kalimat untuk mengetahui apakah kalimat tersebut sesuai dengan aturan produksi atau tidak, sebagai contoh terdapat sebuah kalimat “Belajar kakak di kamar”. Setiap kata yang terdapat pada kalimat memiliki unsur pembentukan, unsur-unsur tersebut adalah bagian dari terminal simbol yang terdapat pada aturan produksi.

```

Begin
1. for i :=1 to n do
2.  $V_{i1} := \{A|A \rightarrow a \text{ aturan produksi dimana simbol ke } i \text{ adalah } a\};$ 
3. for j :=2 to n do
4.   for i :=1 to n-j+1 do
     begin
5.        $V_{ij} := \emptyset;$ 
6.       for k:=1 to j-1 do
7.          $V_{ij} := V_{ij} \cup \{A|A \rightarrow BC \text{ adalah suatu produksi ,dimana } B$ 
           di  $V_k$  dan C di  $V_{i+k,j-k}\}$ 
         end
     end
end
    
```

Psudecode 1. Algoritma CYK

Psudecode diatas membentuk matrik berukuran panjang sebanyak simbol yang akan diuji. Setiap sel diisi oleh x. Pembentukan tersebut didasarkan pada for i :=1 to n do , dengan pengisian sel berupa x ditunjukkan pada baris kedua. Baris ketiga yaitu for j :=2 to n do melakukan pengisian dari baris kedua hingga n, pengisian dari for i :=1 to n-j+1 do, jika j=2 maka n-2+1, jika j=3 maka n-3+1. Baris ke enam yaitu for k:=1 to j-1 do adalah pengisian yang bergerak dari 1 hingga j-1 yang diisi dengan baris ke tujuh berupa $V_{ij} \cup \{A|A \rightarrow BC$, vij atau disebut dengan sel diisi dengan non terminal simbol yang memproduksi BC dimana B adalah V_k dan C di $V_{i+k,j-k}$.

Pemeriksaan kalimat menggunakan algoritma CYK yang dilakukan untuk menemukan susunan kalimat "Belajar kakak di kamar" diperoleh dengan cara membentuk tabel matrik dua dimensi dengan jumlah kolom dan baris sebanyak jumlah kata yang akan diperiksa. Setiap kata yang terdapat pada kalimat ditempatkan pada setiap kolom tabel, seperti yang ditunjukkan pada tabel berikut.

Tabel 4. Pemeriksaan pola kalimat dengan algoritma CYK

	Belajar	kakak	di	kamar

Setiap sel yang kosong yang terdapat pada baris pertama diisi dengan nonterminal simbol yang menurunkan terminal simbol "belajar", "kakak", "di", "kamar", sehingga akan terbentuk tabel seperti berikut :

Tabel 5. Pengisian tabel baris pertama

	Belajar	Kakak	di	Kamar
1	V	N	R	N
2				
3				
4				

Pengisian pada baris kedua diperoleh dengan mengambil nonterminal simbol yang berada pada posisi atas dan kanan atas dari sel yang akan diisi, agar mempermudah proses pengisian pengisian dibuat tabel bantuan untuk menyimpan data sementara seperti berikut :

Tabel 6. Tabel bantuan pengisian baris kedua

		Gabungan	Ruas kiri
V	N	VN	E,P

N	R	NR	∅
R	N	RN	F,K,P

Hasil yang diperoleh pada tabel 6 terdapat pada kolom ruas kiri, simbol “∅” merupakan simbol yang menyatakan bahwa tidak ada nonterminal simbol yang menurunkan “NR” atau dapat diartikan produksi kosong. Hasil yang terdapat pada ruas kiri tabel 6 kemudian menjadi data yang diisikan pada tabel CYK, sehingga diperoleh hasil sementara seperti pada Tabel 7.

Tabel 7. Pengisian baris kedua

	Belajar	Kakak	di	Kamar
1	V	N	R	N
2	↓E,P	↓∅	↓F,K,P	↘
3				
4				

Pengisian pada baris ketiga sama seperti sebelumnya, namun nonterminal simbol yang digabungkan harus terurut, dimulai dari penggabungan baris pertama kolom “Belajar” dengan baris kedua kolom “Kakak”, kemudian baris kedua kolom “Belajar” dengan baris pertama kolom “di”, untuk pengisian pada sel dikolom berikutnya juga sama hanya perlu menyesuaikan posisi kolomnya.

Tabel 8. Tabel bantuan pengisian baris ketiga pada tabel CYK

↓	↙	Gabungan	Ruas kiri
V	∅	V∅	∅
E,P	R	ER,PR	∅
N	F,K,P	NF,NK,NP	P,S,G
∅	N	∅N	∅

Tabel diatas pada kolom ruas kiri terdapat warna biru dan hijau, masing-masing warna tersebut mewakili sel yang terdapat pada tabel CYK, pada tabel diatas terdapat 2 (dua) baris pada ruas kiri dengan warna biru memiliki hasil yang sama untuk penempatan yang sama, dalam penempatannya pada tabel CYK cukup diisikan sekali saja, sedangkan pada ruas kiri yang ditandai dengan warna hijau terdapat nonterminal simbol “P,S,G” dan “∅” maka untuk produksi kosong atau “∅” tidak perlu dituliskan.

Tabel 9. Pengisian baris ketiga kolom pertama

	Belajar	Kakak	di	Kamar
1	V	N	R	N
2	↓E,P	↓∅	↓F,K,P	↘
3	∅	P,S,G		
4				

Pengisian baris terakhir sama seperti pada pengisian baris ketiga, isi yang terdapat pada baris terakhir merupakan penentu apakah kalimat dapat dinyatakan benar atau salah.

Tabel 10. Tabel bantuan pengisian baris keempat pada tabel CYK

↓	↙	Gabungan	Ruas kiri
V	P,S,G	VP,VS,VG	∅
E,P	F,K,P	EF,EK,EP,PF,PK,PP	P
∅	N	∅N	∅

Warna hijau pada tabel diatas memiliki 2 (dua) produksi kosong dengan dan 1 (satu) nonterminal simbol, hasil pengisian dapat dilihat pada Tabel 11.

Tabel 11. Pengisian baris keempat

	Belajar	Kakak	di	Kamar
1	V	N	R	N
2	E,P	Ø	F,K,P	
3	Ø	P,S,G		
4	P			

Hasil yang ditunjukkan pada tabel diatas menandakan kalimat tidak dapat dikenali, sebab hasil akhirnya tidak mencapai *start symbol* "G".

2.6. Perbaikan pola kalimat menggunakan algoritma DLD

Bagian ini menjelaskan cara untuk memperbaiki pola kalimat yang keliru. Hasil akhir dari tabel 11 pola kalimat tidak dikenali karena tertukarnya kata "belajar" dengan kata "kakak". Perbaikan pola kalimat dilakukan dengan langkah awal membuat data kamus untuk perbandingan susunan kalimatnya.

Perbaikan pola kalimat sama seperti perbaikan kata dengan algoritma DLD. Berikut ini pola kalimat yang diperbaiki menggunakan algoritma DLD pada Tabel 12.

Tabel 12. Perbandingan pola kalimat

			Belajar	Kakak	di	Kamar
			c	a	f	a
		0	1	2	3	4
Kakak	a	1	1	1	2	3
Belajar	c	2	1	1	2	3
di	f	3	2	2	1	3
Kamar	a	4	3	2	2	1

Proses perbandingan pola kalimat dilakukan dengan seluruh data kamus, data kamus yang memiliki nilai terendah akan menjadi rekomendasi untuk perbaikan kalimat.

3. HASIL DAN PEMBAHASAN

Penerapan algoritma CYK dalam memeriksa kalimat sederhana maupun memeriksa kalimat luas menunjukkan hasil yang baik, dalam memeriksa kalimat sederhana hasil pengujian menunjukkan seluruh kalimat dapat diperiksa, namun pada kalimat luas terdapat dua kalimat yang tidak berhasil diperiksa atau hasil pengujian menunjukkan bahwa kalimat tersebut salah, kesalahan disebabkan karena lengkapnya aturan produksi yang dibuat, Penelitian ini menggunakan aturan produksi dari beberapa *sample* kalimat sehingga tidak dapat memeriksa seluruh bentuk atau pola dari kalimat.

Algoritma DLD sebagai algoritma yang digunakan untuk memberi rekomendasi perbaikan baik kata maupun kalimat pada pengujian menunjukkan hasil yang baik dalam memeriksa kata, kegagalan dalam pemeriksaan disebabkan oleh banyaknya perbedaan pada kata yang dimasukkan dan kata yang seharusnya menjadi rekomendasi, banyaknya perbedaan pada susunan kata dapat diartikan bahwa kata sama sekali tidak mirip dengan kata manapun, kondisi tersebut pada sistem telah ditanggulangi dengan menampilkan pemberitahuan kepada pengguna bahwa kata yang dimasukkan tidak terdapat dalam kamus. Penerapan lain algoritma DLD yaitu merekomendasikan kalimat sederhana, namun hasil rekomendasi untuk pemeriksaan tidak terlalu baik, dari sepuluh kalimat sederhana yang dimasukkan hanya 6 (enam) kalimat yang berhasil memberikan rekomendasi sesuai dengan kalimat yang diharapkan sedangkan pada pemeriksaan kalimat luas hanya 5 (lima) kalimat. Hal ini disebabkan karena algoritma DLD mencari minimum *edit distance* yang dihasilkan menunjuk bentuk kalimat yang mendekati kalimat

masuk, sedangkan kalimat yang diharapkan belum tentu kalimat yang mendekati dengan kalimat salah yang dimasukkan. Kondisi ini mirip dengan rekomendasi kata saat menemukan susunan kata yang bentuk katanya berbeda jauh.

4. KESIMPULAN

Berdasarkan hasil pengujian terhadap algoritma *CYK* dan algoritma *Damerau Levenshtein distance* disimpulkan bahwa algoritma *CYK* dapat mengenali kalimat dengan baik sesuai dengan aturan produksi yang dibuat, semakin banyak aturan produksi yang dibuat maka semakin banyak pola yang dapat dikenali. Algoritma *DLD* juga baik dalam merekomendasikan kata, namun untuk rekomendasi kalimat masih kurang baik akibat nilai *edit distance* yang terendah.

DAFTAR PUSTAKA

- Erlin, A. 2017. Pengertian Kata, Hakikat Kata dan Pengklasifikaian Kata. <https://argadiaerlin97.wordpress.com/2017/06/22/pengertian-kata-hakikat-kata-dan-%09pengklasifikasian-kata>, diakses 17 Maret 2018.
- Hayati, V.N. 2014. Aplikasi Segmentasi Teks Bahasa Indonesia Menggunakan Metode Context Free Grammar Untuk Pencocokan Kalimat Skripsi. Skripsi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Maghfira, T., Cholissodin, I. & Widodo, A. 2017. Deteksi Kesalahan Ejaan dan Penentuan Rekomendasi Koreksi Kata yang Tepat Pada Dokumen Jurnal JTIK Menggunakan Dictionary Lookup dan Damerau-Levenshtein Distance. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1(6), 498–506.
- Malik, A.K. 2011. Unsur-Unsur Pembentuk Kalimat. <https://akmalik.wordpress.com/materi-%09kuliah/bahasa-indonesia/modul-2-unsur-unsur-pembentuk-kalimat/>. Diakses 17 Maret 2018
- Pahdi, A. 2016. Koreksi Ejaan Istilah Komputer Berbasis Kombinasi Algoritma Damerau-Levenshtein dan Algoritma Soundex. *Journal Speed–Sentra Penelitian Engineering Dan Edukasi*–8(2),1–8.
- Pyriana, D.R., Suprpto., & Suharsono. 2012. Program Aplikasi Editor Kata Bahasa Indonesia Menggunakan Metode Approximate String Matching Dengan Algoritma Levenshtein Distance Berbasis Java . Program Studi Teknik Informatika Un. Program Studi Teknik Informatika Universitas Brawijaya, 1–10.
- Rachmawati, I. 2016. Pengertian Klausa dan Jenis – Jenis Klausa. <https://portal-ilmu.com/pengertian-dan-jenis-jenis-klausa/>. Diakses 17 Maret 2018
- Setyawan, C.A. 2016. Program Dinamis dan Pencocokan String untuk Penyelesaian Soal Pemrograman Kompetitif. Makalah IF2211 Strategi Algoritma, Semester II Tahun 2015/2016. ITB, Bandung, Indonesia
- Susanti, S. 2016. Analisis Perbandingan Algoritma Lcp(Left-Corner-Parsing) Dan Algoritma Cyk (Cocke-Younger-Kasami) Untuk Memeriksa Pola Kalimat Baku Bahasa Indonesia. Universitas Komputer Indonesia.
- Wahyudipraja, M.F. 2015. Implementasi Algoritma Cocke -Younger-Kasami (CYK) Dan Levenshtein Untuk Merekomendasikan Perbaikan Struktur Kalimat Dan Kesalahan Pengetikkan Bahasa Indonesia. Skripsi Universitas Komputer Indonesia.
- Warsito, A. 2017. 6 Jenis-jenis Klausa Verbal dan Contohnya dalam Kalimat. <https://dosenbahasa.com/jenis-jenis-klausa-verbal>. diakses 17 Maret 2018