

TUGAS TEORI BAHASA DAN OTOMATA

(Dosen Pengampu: **Dr. Anak Agung Istri Ngurah Eka Karyawati, S.Si., M.Eng.**)



Oleh :

Gusti Agung Diah Sri Ari Ningsih	(2208561006)
I Komang Bisma Bendesa Jaya	(2208561024)
Ida Bagus Made Wiguna Tedja Sukmana	(2208561056)
Ida Bagus Satria Masyana Citarsa	(2208561088)
Ni Made Ayu Pranasanthi Dewi	(2208561130)

PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA
2023

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan kemajuan teknologi informasi, pemrosesan bahasa alami menjadi elemen krusial dalam pengembangan sistem komputer untuk memperbaiki interaksi antara manusia dan komputer. Analisis sintaksis, sebagai bagian integral dari pemrosesan bahasa alami, memungkinkan komputer untuk memahami struktur dan tata bahasa dari kalimat yang dimasukkan. Dalam konteks ini, penerapan Context-Free Grammar (CFG) dan algoritma parsing seperti *Cocke-Younger-Kasami* (CYK) menjadi pendekatan yang efektif untuk menguraikan struktur sintaksis bahasa natural. Untuk mendukung perkembangan ini, khususnya dalam bahasa Indonesia, diperlukan alat yang dapat memvalidasi keberadaan kalimat berdasarkan aturan tata bahasa Indonesia.

Aplikasi syntactic parsing yang menggunakan CFG dan algoritma CYK menawarkan solusi untuk menghadapi tantangan dalam memahami struktur sintaksis bahasa Indonesia. Dengan adanya fitur utama untuk melakukan analisis terhadap kalimat, aplikasi ini dapat memberikan umpan balik cepat terkait kebenaran struktur kalimat sesuai dengan tata bahasa. Keberadaan algoritma CYK menjadi kunci dalam proses ini, membantu mempercepat dan meningkatkan akurasi dalam penentuan validitas suatu kalimat. Sebagai hasilnya, pengguna dapat dengan mudah memahami dan memastikan kesesuaian kalimat mereka dengan norma bahasa Indonesia.

1.2 Tujuan

Tujuan yang ingin dicapai

1. Menyajikan konsep analisis sintaksis bahasa Indonesia, termasuk penerapan CFG dan algoritma CYK
2. Memberikan umpan balik cepat terkait kebenaran struktur kalimat sesuai dengan tata bahasa

1.3 Asumsi

Adapun asumsinya yaitu tidak terdapat typo pada input kalimat, semua kata pada kalimat input terdaftar dalam dictionary.

1.4 Batasan

Adapun batasannya yaitu hanya dapat menerima kalimat sederhana atau tunggal, tidak dapat menerima kalimat majemuk, dan tidak dapat menerima tanda baca seperti titik.

BAB II

Metode Context-Free Grammar

2.1 Context-Free Grammar

Dari kasus tersebut kami merancang sebuah Context-Free Grammar yang dapat ditulis dengan $G(V, \Sigma, R, K)$ dengan keterangan:

$V = \{K, S, P, O, Pel, Ket, Noun, Verb, Num, Pronoun, PropNoun, Prep, Adv, Adj, NP, VP, NumP, AdjP, PP, saya, lelah, \dots\}$

$\Sigma = \{K, S, P, O, Pel, Ket, Noun, Verb, Num, Pronoun, PropNoun, Prep, Adv, Adj, NP, VP, NumP, AdjP, PP\}$

$R = \{$

1. $K \rightarrow S P$
2. $K \rightarrow S Ket$
3. $K \rightarrow S P O$
4. $K \rightarrow S P Pel$
5. $K \rightarrow S P Ket$
6. $K \rightarrow S P Pel Ket$
7. $K \rightarrow S P O Pel$
8. $K \rightarrow S P O Ket$
9. $K \rightarrow S P O Pel Ket$
10. $S \rightarrow NP$
11. $S \rightarrow Pronoun$
12. $S \rightarrow PropNoun$
13. $P \rightarrow NP$
14. $P \rightarrow VP$
15. $P \rightarrow PP$
16. $P \rightarrow AdjP$
17. $O \rightarrow NP$
18. $O \rightarrow NumP$
19. $Pel \rightarrow NP$
20. $Pel \rightarrow AdjP$
21. $Pel \rightarrow NumP$
22. $Ket \rightarrow PP$
23. $NP \rightarrow Num NP$
24. $NP \rightarrow NP Adj$
25. $NP \rightarrow NP Noun$
26. $NP \rightarrow NP Pronoun$
27. $NP \rightarrow NP PropNoun$
28. $NP \rightarrow NP Adv$
29. $NP \rightarrow Adv NP$
30. $NP \rightarrow Noun$
31. $NP \rightarrow PropNoun$
32. $NP \rightarrow Pronoun$
33. $VP \rightarrow Adv VP$
34. $VP \rightarrow Verb$

35. NumP \rightarrow Num NP
 36. NumP \rightarrow Num
 37. AdjP \rightarrow NP Adj
 38. AdjP \rightarrow Adv Adj
 39. AdjP \rightarrow Adj
 40. PP \rightarrow Prep NP
 41. Prep \rightarrow {di, ke, dari, oleh, pada, dengan, tanpa, untuk, dari, ke, sampai, sejak, hingga}
 42. Num \rightarrow {1, 2, 3, 4, 5, 6, 7, 8, 9, 0, ...}
 43. Adj \rightarrow {Manis, pahit, besar, baru, biru, merah, panas, dingin,...}
 44. Adv \rightarrow {sangat, agak, jarang, sudah, akan, sedang, ...}
 45. Noun \rightarrow {Putra, Saya, Burung, Tebu, Rasa, Pohon, Hutan, Tanaman, Kebun, bunga, Sepeda, Garasi, Komputer, Meja, Buku, Rak, Lampu, meja, ...}
 46. Verb \rightarrow {bermain, memasak, memberi, belajar, membantu, menjaga, melukis, memadamkan, menulis, bekerja, ...}
 47. Pronoun \rightarrow {ini, itu, saya, dia,...}
 48. PropNoun \rightarrow {Agung, Kali Unda, Denpasar, New York,...}
- }

Dengan Context-Free Grammar tersebut kita dapat menderivasi sebuah kalimat tunggal, contohnya:

$K \Rightarrow S P O \Rightarrow NP VP NP \Rightarrow \text{Noun Adv VP Noun} \Rightarrow \text{Noun Adv Verb Noun} \Rightarrow \text{saya sedang makan lampu}$

2.2 Chomsky Normal Form

Setelah mendapatkan CFG, sekarang kita mengonversikan aturan CFG tersebut menjadi bentuk yang lebih efisien yaitu Chomsky Normal Form (CNF). Sebuah aturan produksi dalam bentuk CNF memiliki aturan hanya boleh jika:

1. Hanya ada 2 non-terminal di sisi kanan, contohnya $A \rightarrow AB$
2. Hanya ada 1 terminal di sisi kanan, contohnya $A \rightarrow z$
3. Menjadi null string, contohnya $A \rightarrow \epsilon$

Adapun hasil konversi kami :

1. $K \rightarrow S P$	$K \rightarrow S P \mid S \text{ Ket} \mid L1 O \mid L1 \text{ Pel} \mid L1 \text{ Ket} \mid L2 \text{ Pel} \mid L2 \text{ Ket} \mid L3 \text{ Ket} \mid$
2. $K \rightarrow S \text{ Ket}$	$L4 \text{ Ket} \mid L5$
3. $K \rightarrow S P O$	$L1 \rightarrow S P$
4. $K \rightarrow S P \text{ Pel}$	$L2 \rightarrow L1 O$
5. $K \rightarrow S P \text{ Ket}$	$L3 \rightarrow L1 \text{ Pel}$
6. $K \rightarrow S P \text{ Pel Ket}$	$L4 \rightarrow L1 L3$
7. $K \rightarrow S P O \text{ Pel}$	$L5 \rightarrow L4 \text{ Ket}$
8. $K \rightarrow S P O \text{ Ket}$	
9. $K \rightarrow S P O \text{ Pel Ket}$	

1. $S \rightarrow NP$ 2. $S \rightarrow \text{Pronoun}$ 3. $S \rightarrow \text{PropNoun}$	$S \rightarrow NP \text{ Noun} \mid NP \text{ Adj} \mid NP \text{ PropNoun} \mid \text{Num NP} \mid NP \text{ Pronoun} \mid NP \text{ Adv} \mid \text{Adv NP} \mid \text{saya} \mid \text{Putra} \mid \text{batu} \mid \text{hari} \mid \text{tiga} \mid \text{itu}$
1. $P \rightarrow NP$ 2. $P \rightarrow VP$ 3. $P \rightarrow \text{AdjP}$	$P \rightarrow NP \text{ Noun} \mid NP \text{ Adj} \mid NP \text{ PropNoun} \mid \text{Num NP} \mid NP \text{ Pronoun} \mid NP \text{ Adv} \mid \text{Adv NP} \mid \text{Adv VP} \mid \text{Adv NP} \mid \text{Prep NP} \mid VP \text{ Adj} \mid NP \text{ Adj} \mid \text{guru} \mid \text{makan} \mid \text{tidur} \mid \text{marah}$
1. $O \rightarrow NP$ 2. $O \rightarrow \text{NumP}$	$O \rightarrow NP \text{ Noun} \mid NP \text{ Adj} \mid NP \text{ PropNoun} \mid \text{Num NP} \mid NP \text{ Pronoun} \mid NP \text{ Adv} \mid \text{Adv NP} \mid \text{saya} \mid \text{Putra} \mid \text{batu} \mid \text{tiga}$
1. $Pel \rightarrow NP$ 2. $Pel \rightarrow \text{AdjP}$ 3. $Pel \rightarrow \text{NumP}$	$Pel \rightarrow NP \text{ Noun} \mid NP \text{ Adj} \mid \text{Adv Adj} \mid NP \text{ PropNoun} \mid \text{Num NP} \mid NP \text{ Pronoun} \mid NP \text{ Adv} \mid \text{Adv NP} \mid \text{saya} \mid \text{Putra} \mid \text{batu} \mid \text{hari} \mid \text{tiga} \mid \text{sedih}$
1. $Ket \rightarrow PP$	$Ket \rightarrow \text{Prep NP} \mid$
1. $NP \rightarrow \text{Num NP}$ 2. $NP \rightarrow NP \text{ Adj}$ 3. $NP \rightarrow NP \text{ Noun}$ 4. $NP \rightarrow NP \text{ Pronoun}$ 5. $NP \rightarrow NP \text{ PropNoun}$ 6. $NP \rightarrow NP \text{ Adv}$ 7. $NP \rightarrow \text{Adv NP}$ 8. $NP \rightarrow \text{Noun}$ 9. $NP \rightarrow \text{PropNoun}$ 10. $NP \rightarrow \text{Pronoun}$	$NP \rightarrow NP \text{ Noun} \mid NP \text{ Adj} \mid NP \text{ PropNoun} \mid \text{Num NP} \mid NP \text{ Pronoun} \mid NP \text{ Adv} \mid \text{Adv NP} \mid \text{saya} \mid \text{Putra} \mid \text{batu} \mid \text{hari} \mid \text{tiga} \mid \text{itu} \mid$
1. $VP \rightarrow \text{Adv VP}$ 2. $VP \rightarrow \text{Verb}$	$VP \rightarrow \text{Adv VP} \mid \text{kerja} \mid \text{pergi} \mid \text{terbang} \mid \text{loncat} \mid \text{makan}$
1. $\text{NumP} \rightarrow \text{Num NP}$ 2. $\text{NumP} \rightarrow \text{Num}$	$\text{NumP} \rightarrow \text{Num NP} \mid 1 \mid 2 \mid 3 \mid \text{satu} \mid$
1. $\text{AdjP} \rightarrow NP \text{ Adj}$ 2. $\text{AdjP} \rightarrow \text{Adv Adj}$ 3. $\text{AdjP} \rightarrow \text{Adj}$	$\text{AdjP} \rightarrow NP \text{ Adj} \mid \text{Adv Adj} \mid \text{marah} \mid \text{sedih} \mid \text{besar} \mid \text{berat}$
1. $PP \rightarrow \text{Prep NP}$	$PP \rightarrow \text{Prep NP}$

2.3 Algoritma CYK

Algoritma CYK (Cocke-Younger-Kasami) yang diimplementasikan dalam code tersebut berfungsi untuk melakukan parsing string berdasarkan aturan-aturan grammar yang telah didefinisikan dalam bentuk CNF (Chomsky Normal Form).

Kami memiliki langkah-langkah secara sistematis yang kami gunakan secara spesifik untuk membangun aplikasi parsing ini. Berikut adalah langkah-langkah dari algoritma kami:

1. Memisahkan input string kata per kata
Input string yang diberikan dipisahkan kata per kata lalu masing-masing kata dimasukkan ke dalam array.
2. Memeriksa kata
Memeriksa kata kata yang dipisahkan tadi apakah ada di dalam aturan tata bahasa
3. Mengisi tabel
Mengisi tabel dengan memeriksa produksi yang memiliki satu kata di sisi kanan dan menambahkan terminal ke sel yang sesuai di tabel. Setelah itu memeriksa produksi yang memiliki dua kata di sisi kanan dan mencoba membangun kemungkinan kombinasi non-terminal yang dapat menghasilkan sub-string dari input.
4. Memeriksa String input
Memeriksa string input apakah dapat diterima berdasarkan isi tabel. Jika simbol awal dari CFG ada di sel teratas kiri tabel dan semua kata dalam string ada di tata bahasa, maka string dapat diterima

Kode CYK:

```
def cyk(dictionary_cnf, input_text):  
    n = len(input_text.split())  
    table_filling = [[set([]) for j in range(n)] for i in range(n + 1)]  
    get_all_right_hand(dictionary_cnf, n)  
    flag = string_exist(input_text, n)  
    filling_bottom(dictionary_cnf, input_text, table_filling, n)  
    filling_remaining(dictionary_cnf, table_filling, n)  
    # tabulate_table = tabulate.tabulate(table_filling)  
    # print(tabulate_table)  
    return is_accepted(dictionary_cnf, table_filling, flag, n)
```

BAB III

Analisis dan Desain

3.1 Desain Eksperimen

Eksperimen dilakukan dengan cara memasukkan 100 kalimat yang telah diberikan. 80 diantaranya adalah kalimat yang baku mengikuti tata bahasa indonesia, sedangkan 20 kalimat lainnya adalah kalimat tidak baku. Menghitung Metrik Keakuratan dengan cara sebagai berikut.

$$Keakuratan = \left(\frac{\text{Jumlah Kalimat yang Diprediksi dengan Benar}}{\text{Total Kalimat yang Diuji}} \times 100\% \right)$$

3.2 Desain Aplikasi

A. Flowchart



B. Mockup

Tampilan Awal :



The initial mockup shows a dark-themed interface. At the top, the title "Parsing Kalimat Bahasa Indonesia" is displayed in white. Below it, a subtitle reads "Sistem kami akan memeriksa apakah input kalimat anda valid atau tidak". The "Input Kalimat" section features a placeholder text "Masukan kalimat anda..." inside a dark input field. A red "Periksa" button is positioned below the input field.

Tampilan Kalimat Valid:



This mockup shows the application's response to a valid input. At the top, a green message states "Kalimat yang anda berikan sudah valid ✓". The title "Parsing Kalimat Bahasa Indonesia" remains. The subtitle is the same. The "Input Kalimat" section now contains the text "Nenek mengambil sedikit daun ubi di kebun" in the input field. The red "Periksa" button is still present below the input field.

Kalimat Tidak Valid:

Kalimat yang anda berikan sudah **tidak valid** ❌

Parsing Kalimat Bahasa Indonesia

Sistem kami akan memeriksa apakah input kalimat anda valid atau tidak

Input Kalimat

Lima belas pasang sepatu terpajang di rak sepatu

Periksa

Kalimat tidak terdaftar:

Kalimat yang anda berikan sudah **tidak didaftarkan** ⚠️

Parsing Kalimat Bahasa Indonesia

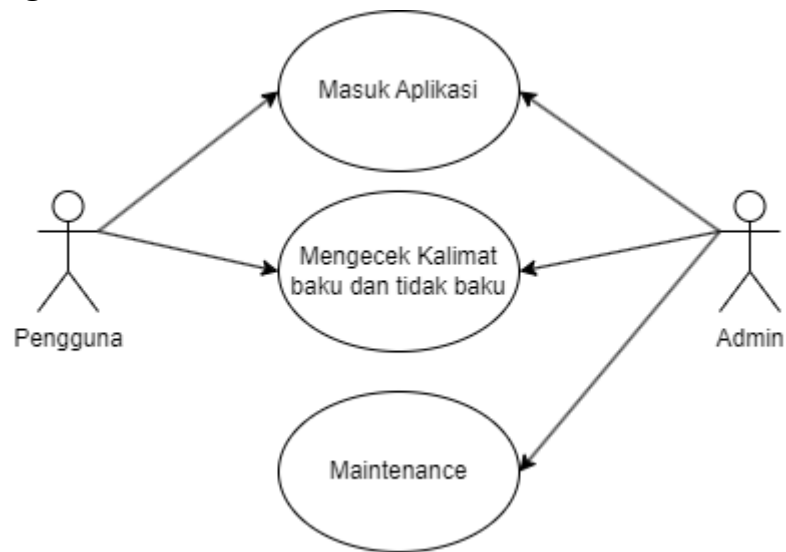
Sistem kami akan memeriksa apakah input kalimat anda valid atau tidak

Input Kalimat

uang haram

Periksa

C. Use Case Diagram



Pada use case diatas terdapat 2 aktor yakni pengguna dan Admin dimana keduanya memiliki deskripsi sebagai berikut:

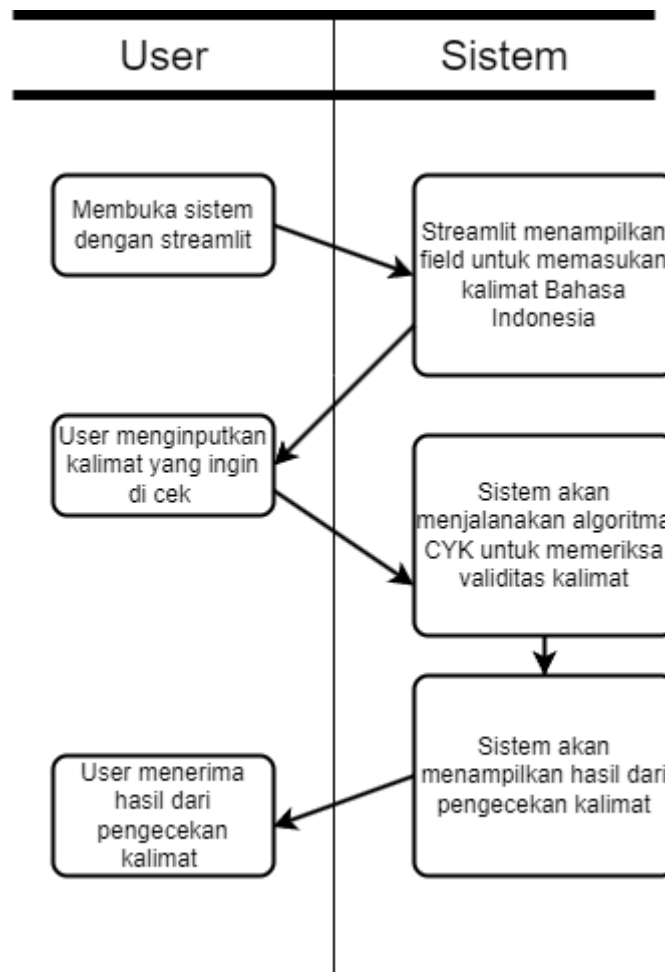
a. Pengguna

Pengguna dapat masuk kedalam aplikasi dan mengecek kalimat yang diberikan apakah baku atau tidak baku.

b. Admin

Admin dapat masuk kedalam aplikasi dan mengecek kalimat yang diberikan apakah baku atau tidak baku, selain itu juga admin dapat melakukan pemeliharaan terhadap aplikasi dan juga menambah rule yang baru pada aplikasi.

D. Activity Diagram



E. Implementasi Desain Eksperimen

Desain eksperimen diimplementasikan dengan cara mengumpulkan kalimat baku dalam satu dokumen “txt.” begitu juga dengan kumpulan kalimat tidak baku. File tersebut kemudian akan di scan dan langsung menjadi input dalam algoritma CYK. Kemudian kami menghitung berapa jumlah prediksi dari algoritma yang benar sesuai dengan kalimat yang dibaca. Hasil hitungan dimasukkan ke dalam rumus keakuratan yang telah kami buat.

F. Implementasi Source Code

```
import tabulate
get_right_hand = []

def get_all_right_hand(dictionary_cnf, n):
    for i in range(0, n):
        for left_hand, rule in dictionary_cnf.items():
            for right_hand in rule:
```

```

        get_right_hand.append(right_hand)

def string_exist(string, n):
    for i in range(0, n):
        if string.split()[i] not in get_right_hand:
            return -1
    return 1

def filling_bottom(dictionary_cnf, input_text, table_filling, n):
    string_split = input_text.split()
    for i in range(0, n):
        for left_hand, rule in dictionary_cnf.items():
            for right_hand in rule:
                if len(right_hand.split()) == 1 and right_hand ==
string_split[i]:
                    table_filling[i][i].add(left_hand)

def filling_remaining(dictionary_cnf, table_filling, n):
    for i in range(0, n):
        for j in range(i, -1, -1):
            for k in range(j, i + 1):
                for left_hand, rule in dictionary_cnf.items():
                    for right_hand in rule:
                        if len(right_hand.split()) == 2 and
right_hand.split()[0] in table_filling[j][k] and
right_hand.split()[1] in table_filling[k + 1][i]:
                            table_filling[j][i].add(left_hand)

def is_accepted(dictionary_cnf, table_filling, flag, n):
    if list(dictionary_cnf.keys())[0] in table_filling[0][n - 1] and
flag == 1:
        return ["valid", table_filling]
    elif flag == -1:
        return ["not_registered", table_filling]
    else:
        return ["invalid", table_filling]

```

```
def cyk(dictionary_cnf, input_text):
    n = len(input_text.split())
    table_filling = [[set([]) for j in range(n)] for i in range(n + 1)]

    get_all_right_hand(dictionary_cnf, n)
    flag = string_exist(input_text, n)
    filling_bottom(dictionary_cnf, input_text, table_filling, n)
    filling_remaining(dictionary_cnf, table_filling, n)
    # tabulate_table = tabulate.tabulate(table_filling)
    # print(tabulate_table)
    return is_accepted(dictionary_cnf, table_filling, flag, n)
```

Fungsi `get_all_right_hand`:

- Fungsi ini mengumpulkan semua sisi kanan dari aturan produksi dalam tata bahasa bebas konteks yang diberikan.
- Menerima kamus (`dictionary_cnf`) yang mewakili tata bahasa bebas konteks dan jumlah kata dalam teks masukan (`n`).
- Sisi kanan dikumpulkan dalam daftar `get_right_hand`.

Fungsi `string_exist`:

- Fungsi ini memeriksa apakah setiap kata dalam teks masukan adalah simbol terminal yang valid dalam tata bahasa.
- Menerima string (`string`) yang mewakili teks masukan dan jumlah kata dalam teks masukan (`n`).
- Mengembalikan 1 jika semua kata dalam teks masukan adalah simbol terminal yang valid; sebaliknya, mengembalikan -1.

Fungsi `filling_bottom`:

- Fungsi ini mengisi baris paling bawah dari tabel parsing berdasarkan simbol terminal dalam teks masukan.
- Menerima kamus tata bahasa bebas konteks (`dictionary_cnf`), teks masukan (`input_text`), tabel parsing (`table_filling`), dan jumlah kata dalam teks masukan (`n`).
- Mengisi tabel berdasarkan aturan produksi untuk kata tunggal.

Fungsi `filling_remaining`:

- Fungsi ini mengisi sel-sel yang tersisa dari tabel parsing menggunakan aturan produksi tata bahasa.
- Menerima kamus tata bahasa bebas konteks (`dictionary_cnf`), tabel parsing (`table_filling`), dan jumlah kata dalam teks masukan (`n`).
- Mengisi tabel berdasarkan aturan produksi untuk dua kata.

Fungsi `is_accepted`:

- Fungsi ini memeriksa apakah teks masukan diterima oleh tata bahasa berdasarkan keberadaan simbol awal di sel kanan atas tabel parsing.
- Menerima kamus tata bahasa bebas konteks (`dictionary_cnf`), tabel parsing (`table_filling`), bendera yang menunjukkan validitas teks masukan (`flag`), dan jumlah kata dalam teks masukan (`n`).
- Mengembalikan daftar yang menunjukkan status validitas dan tabel parsing.

Fungsi `cyk`:

- Fungsi utama yang mengorganisir algoritma CYK.
- Menginisialisasi tabel parsing, mengumpulkan sisi kanan, mengisi tabel, dan memeriksa apakah diterima.
- Mengembalikan daftar yang berisi status validitas dan tabel parsing.

BAB IV

Hasil dan Pembahasan

4.1. Hasil Eksperimen

A. Kalimat baku:

Nenek mengambil sedikit daun ubi di kebun. (S P O Ket)	Valid
Banyak orang menghadiri acara itu. (S P O)	Valid
Sekitar lima mahasiswa mengikuti kegiatan pengabdian tersebut. (S P O)	Valid
Orang dewasa saja diperbolehkan menaiki wahana itu. (S P Pel)	Valid
Pemadam kebakaran itu hanya mempunyai sedikit waktu untuk menyelamatkan penduduk desa tersebut. (S P O Pel)	Invalid
Anak laki-laki itu jarang bermain di taman kota. (S P Ket)	Valid
Ibu memasak banyak makanan lezat. (S P O)	Valid
Para siswa sering belajar di perpustakaan kampus. (S P Ket)	Valid
Polisi selalu menjaga ketertiban di jalan. (S P O Ket)	Valid
Burung merpati itu hanya terbang di siang hari. (S P Ket)	Valid
Ikan koi banyak berenang di kolam taman. (S P Ket)	Valid
Kuda pacuan itu selalu berlari dengan cepat di lintasan pacu. (S P Pel Ket)	Valid
Monyet kecil itu sering bergelantungan di ranting pohon. (S P Ket)	Valid
Kelinci putih itu melompat-lompat di kebun. (S P Ket)	Valid
Lumba-lumba itu bermain di laut lepas. (S P Ket)	Valid
Kupu-kupu sering hinggap di bunga (S P Ket).	Valid
Sapi selalu mengunyah rumput di padang rumput. (S P O Ket)	Valid
Lebah itu mengumpulkan banyak nektar dari bunga. (S P O Pel)	Valid
Seekor buaya sedang berjemur di tepi danau (S P Ket)	Invalid
Banyak mahasiswa hadir pada acara seminar itu. (S P Pel)	Valid
Dr. Siti Rahayu memberikan kuliah di universitas. (S P O Ket)	Valid
Ani sedang berlibur di pantai. (S P Pel)	Valid
Adi Santoso merupakan pemimpin proyek tersebut. (S P Pel)	Valid

Bu Ratna adalah pemilik warung kopi itu.(S P Pel)	Valid
Prof. Dr. Hadi Prayitno adalah pakar bidang ekologi. (S P Pel)	Valid
Kiki menjadi juara pada lomba menyanyi.	Valid
Ibu Sinta akan menjadi pembicara dalam seminar itu. (S P Pel Ket)	Valid
Bapak Haryono selalu membantu tetangga. (S P O)	Valid
Laut Mediterania sangat terkenal dengan keindahan pantainya (S P Pel).	Valid
Bukit Tinggi memiliki pemandangan alam indah (S P O).	Valid
Danau Baikal adalah danau terdalam di dunia. (S P Pel Ket)	Valid
Pantai Copacabana menjadi ikon kota Rio de Janeiro (S P Pel).	Valid
Bali selalu terkenal dengan keindahan pantainya. (S P Pel)	Valid
Kota Bandung itu terkenal dengan kulinernya. (S P Pel)	Valid
Bogor dikenal dengan kota hujan. (S P Pel)	Valid
Monas terletak di tengah Jakarta. (S P Ket)	Valid
Bali memiliki banyak budaya indah. (S P Pel)	Valid
Denpasar sangat ramai pada malam hari. (S P Ket)	Valid
Surabaya merupakan kota terbesar di Jawa Timur. (S P Pel Ket)	Valid
Pentas ogoh-ogoh dilakukan menyambut Tahun Baru Saka. (S P Pel)	Valid
Arab Saudi merupakan negara penghasil minyak terbesar di dunia. (S P Pel Ket)	Valid
Indonesia adalah negara kepulauan terbesar di dunia. (S P Pel Ket)	Valid
Lima belas pasang sepatu terpajang di rak sepatu (S P Ket)	Invalid
Delapan buah rumah sedang dalam tahap konstruksi. (S P)	Invalid
Dua ekor kucing selalu bermain di taman belakang rumah. (S P Ket)	Invalid
Sepuluh orang siswa sedang belajar di perpustakaan. (S P Ket)	Invalid
Tujuh puluh lima orang pengunjung hadir dalam acara tersebut. (S P Pel)	Invalid
Kamar tidur ini sangat nyaman. (S P)	Valid
Pertunjukan teater ini sangat indah. (S P)	Valid
Mobil tersebut sudah lama digunakan. (S P Pel)	Valid

Bunga itu harus disiram setiap pagi. (S pel Ket)	Valid
Kotak besar itu berisi banyak mainan anak-anak. (S P Pel)	Valid
Kuda hitam itu sedang berlari. (S P)	Valid
Dia membeli tiga buah buku. (S P O)	Valid
Bapak selalu membeli tiga bungkus rokok. (S P O)	Valid
Saya sedang meneliti sejarah kebudayaan Bali. (S P O)	Valid
Teman saya sedang kursus bahasa Jepang. (S P Pel)	Valid
Asrama putri itu sangat luas. (S P)	Valid
Laki-laki itu adalah pemain sepak bola. (S P Pel)	Valid
Bapak guru saya itu orang Medan. (S P)	Valid
Kakek saya sangat menyukai kursi rotan. (S P O)	Valid
Pelatih tari itu dua orang. (S P)	Valid
Rencana perjalanan kita selalu disusun dengan sangat baik (S P Pel)	Valid
Pengembangan teknologi berlangsung dengan pesat (S P Pel).	Valid
Gadis kecil itu sangat senang dengan boneka barunya (S P Pel)	Valid
Air terjun pegunungan itu sangat indah. (S P)	Valid
Pertunjukan seni malam ini sangat menghibur (S P).	Valid
Rumah tua itu memiliki sejarah panjang. (S P O)	Valid
Bunga yang mekar itu adalah kebanggaan taman kita (S P Pel).	Valid
Kucing hitam itu menjadi teman setia anak-anak (S P Pel)	Valid
Udara sejuk memberikan kesegaran pikiran (S P O)	Valid
Prestasi anak-anak itu adalah kebanggaan orang tua (S P Pel).	Valid
Rasa coklat menjadi kelemahannya. (S P Pel)	Valid
Pohon rindang itu memberikan kesejukan. (S P O)	Valid
Air hujan membuat suasana romantis. (S P O)	Valid
Cita-cita tinggi adalah pendorong kesuksesan. (S P Pel)	Valid
Kucing itu hewan peliharaan manis. (S P)	Valid

Pameran itu menampilkan karya seni kontemporer. (S P O)	Valid
Budaya lokal menjadi daya tarik wisata daerah itu. (S P Pel)	Valid
Hewan peliharaan ini adalah teman setia keluarga kami. (S P Pel)	Valid

B. Kalimat Tidak Baku

Sangat menyukai kursi rotan.	Invalid
Pelatih tari itu.	Invalid
Rencana perjalanan kita selalu.	Invalid
Pengembangan teknologi.	Invalid
Gadis kecil itu.	Invalid
Air terjun pegunungan itu.	Valid
Sangat menghibur.	Invalid
Memiliki sejarah panjang.	Invalid
Kebanggaan taman kita.	Invalid
Menjadi teman setia anak-anak.	Invalid
Memberikan kesegaran pikiran.	Invalid
Kebanggaan orang tua.	Invalid
Rasa coklat menjadi.	Valid
Pohon rindang itu memberikan.	Valid
Air hujan membuat.	Valid
Cita-cita tinggi adalah pendorong kesuksesan.	Valid
Hewan peliharaan manis.	Valid
Karya seni kontemporer.	Valid
Menjadi daya tarik wisata daerah itu.	Invalid
Teman setia keluarga kami.	Invalid

Dari eksperimen yang kami lakukan, kami mendapatkan hasil yaitu 7 prediksi yang salah pada kalimat baku dan 7 prediksi yang salah pada kalimat tidak baku. Maka dari itu, kami mendapatkan keakuratan:

$$Keakuratan = \left(\frac{86}{100} \times 100\% \right) = 86\%$$

Sebuah kesalahan prediksi terjadi pada kalimat dengan struktur S P O Pel, yaitu “Pemadam kebakaran itu hanya mempunyai sedikit waktu untuk menyelamatkan penduduk desa tersebut”. Kesalahan juga terjadi pada kalimat berstruktur S P Ket yaitu, “Seekor buaya sedang berjemur di tepi danau”. Lima kesalahan lainnya terdapat pada kalimat yang memiliki awalan kata numeralia, seperti “Tujuh puluh lima orang pengunjung hadir dalam acara tersebut”.

Pada kalimat tidak baku, terjadi kesalahan prediksi pada kalimat yang menggunakan kata kerja transitif seperti “air hujan membuat”.

4.2 Pembahasan Hasil

Keakuratan model sebesar 86% bisa dianggap baik, tetapi masih memungkinkan untuk ditingkatkan. Perbaikan dapat dilakukan pada model dengan cara membuat aturan produksi yang lebih spesifik. Hal ini dicerminkan pada hasil kesalahan prediksi kalimat tidak baku yang memakai kata kerja transitif. Kata kerja transitif tidak kami pisahkan jenisnya pada dictionary aturan yang kami buat. Maka dari itu perbaikan dapat dilakukan dengan cara menggolongkan kata kerja yang transitif dan intransitif.

Kesalahan yang terjadi pada kalimat dengan awalan kata numeralia menunjukkan adanya potensi masalah dalam memproses konteks khusus seperti ini. Mungkin diperlukan penyesuaian aturan produksi agar mereka dapat diterima oleh program.

BAB V

Kesimpulan

5.1 Simpulan

Dalam pengembangan sistem pemrosesan bahasa alami, analisis sintaksis menjadi elemen krusial untuk meningkatkan interaksi antara manusia dan komputer. Penerapan Context-Free Grammar (CFG) dan algoritma parsing seperti Cocke-Younger-Kasami (CYK) membuktikan menjadi pendekatan efektif dalam memahami struktur dan tata bahasa dari kalimat bahasa natural.

Eksperimen yang dilakukan menghasilkan keakuratan model sebesar 86%, yang dapat dianggap baik namun masih memungkinkan untuk ditingkatkan. Kesalahan pada kalimat tidak baku, terutama yang menggunakan kata kerja transitif, mengindikasikan perlunya membuat aturan produksi yang lebih spesifik. Memisahkan jenis kata kerja transitif dan intransitif dalam aturan produksi dapat menjadi solusi perbaikan yang signifikan.

Selain itu, kesalahan pada kalimat dengan awalan kata numeralia menunjukkan potensi masalah dalam memproses konteks khusus. Penyesuaian aturan produksi untuk mengakomodasi struktur kalimat dengan awalan kata numeralia dapat meningkatkan kemampuan model dalam mengatasi kasus-kasus semacam ini.

Melalui penggunaan aplikasi syntactic parsing yang memanfaatkan CFG dan algoritma CYK, pengguna dapat dengan cepat memvalidasi keberadaan kalimat berdasarkan aturan tata bahasa Indonesia. Namun, perlu diingat bahwa aplikasi ini memiliki batasan, seperti kemampuan menerima hanya kalimat sederhana tanpa tanda baca atau kalimat majemuk.

Dengan memahami dan mengatasi kesalahan-kesalahan spesifik yang diidentifikasi dalam eksperimen, perbaikan model dapat dilakukan untuk meningkatkan keakuratan dan kemampuan aplikasi dalam menguraikan struktur sintaksis bahasa Indonesia.

DAFTAR PUSTAKA

Zia code. 2021. Penggunaan Algoritma CYK untuk Membuktikan Inputan CFG dengan Mudah. <https://ziakode.com/contoh-algoritma-cyk-inputan-cfg/>. Diakses pada [30 Desember 2023]