

TEORI BAHASA DAN OTOMATA

“Tugas Kelompok 2: Parsing Kalimat Baku Kalimat Bahasa Indonesia”



Dosen Pengampu:

Anak Agung Istri Ngurah Eka Karyawati, S.Si., M.Eng.

Kelompok D1:

1. I Wayan Gede Gemuh Raharja R.L (2208561004)
2. I Made Prenawa Sida Nanda (2208561017)
3. I Gede Yogananda Adi Baskara (2208561061)
4. Putu Chandra Mayoni (2208561111)
5. I Gusti Bagus Sutha Arianata Putra (2208561141)

PRODI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

2023

DAFTAR ISI

DAFTAR ISI.....	i
BAB I PENDAHULUAN.....	1
1.1 Perumusan Masalah.....	2
1.2 Tujuan.....	3
1.3 Asumsi.....	3
1.4 Batasan.....	3
BAB II METODE CONTEXT-FREE GRAMMAR (CFG).....	4
2.1 Konsep Dasar CFG.....	4
2.2 Four-Tuple CFG.....	5
2.3 Konversi CFG Rules menjadi CNF (Chomsky Normal Form).....	8
2.4 Algoritma CYK.....	14
BAB III ANALISIS DAN DESAIN.....	17
3.1 Desain Eksperimen.....	17
3.1.1 Skenario Eksperimen.....	17
3.2 Desain Aplikasi.....	18
3.2.1 Flowchart/Arsitektur Sistem.....	18
3.2.2 Mockup.....	19
3.2.3 Use Case Diagram.....	21
3.2.4 Activity Diagram.....	22
3.3 Implementasi.....	23
3.3.1 Source Code.....	23
3.3.2 Penjelasan Source Code.....	23
BAB IV HASIL DAN PEMBAHASAN.....	26
4.1 Hasil Eksperimen.....	26
4.1.1 Skenario Eksperimen dan Hasil.....	26
4.2 Pembahasan dan Analisis Hasil.....	36
BAB V KESIMPULAN.....	37
5.1. Kesimpulan.....	37
5.2. Saran.....	38
DAFTAR PUSTAKA.....	39
LAMPIRAN.....	40

BAB I

PENDAHULUAN

1.1 Perumusan Masalah

Suatu tata bahasa harus memenuhi kriteria ilmiah atau empiris artinya tata bahasa harus bisa dibuktikan secara ilmiah oleh setiap orang, di setiap tempat dan pada setiap waktu. Pada kenyataannya bahasa Indonesia memiliki struktur gramatikal yang relatif fleksibel dibandingkan dengan bahasa lain seperti bahasa Inggris. Posisi subjek, predikat, objek, dan keterangan/pelengkap dalam kalimat sangat bervariasi, yang membuat bahasa Indonesia menjadi lebih kompleks sehingga dengan menggunakan parser sintaksis dalam pemrosesan bahasa alami (NLP) dapat membantu meningkatkan akurasi dan efektivitas sistem (NLP).

Parsing merupakan proses menganalisis string masukkan bahasa untuk memahami struktur gramatikalnya. Parsing membantu dalam mengidentifikasi subjek, objek, predikat, dan elemen lain dalam kalimat adapun syntactic parsing mengacu pada aturan yang mengatur susunan kata dalam kalimat untuk membentuk frasa yang gramatikal dalam suatu bahasa. Syntactic ini merupakan fondasi dari proses parsing karena aturan-aturan yang digunakan untuk menentukan bagaimana kalimat harus diurai dalam proses analisis ini digunakan context-free grammar (CFG) untuk mendeskripsikan bagaimana kata dapat digabungkan dalam membentuk kalimat dengan memakai algoritma Cocke-Younger-Kasami (CYK).

Algoritma Cocke-Younger-Kasami (CYK) adalah metode parsing yang khusus dirancang bekerja dengan CFG dalam bentuk CNF. Algoritma ini berbasis tabel yang menggunakan pendekatan pemrograman dinamis untuk menentukan apakah string tertentu dapat dihasilkan oleh CFG. Algoritma ini sangat efektif dalam menganalisis struktur kalimat kompleks dan digunakan secara luas dalam aplikasi NLP. Oleh karena itu, laporan ini dibuat untuk memaparkan aplikasi parsing Bahasa Indonesia yang telah dibuat oleh kelompok kami.

1.2 Tujuan

Tujuan dari pembuatan aplikasi parsing kalimat bahasa indonesia ini adalah untuk menerapkan penggunaan syntactic parsing, algoritma CFG, CNF dan CYK serta memberikan wawasan baru tentang struktur dan kompleksitas gramatikal Bahasa Indonesia

1.3 Asumsi

Adapun asumsi - asumsi yang terdapat pada aplikasi parsing kalimat bahasa indonesia adalah

1. Kalimat yang diinputkan ke dalam aplikasi parsing harus memakai kalimat berbahasa indonesia
2. Kalimat yang dapat dimasukkan adalah kalimat sederhana dengan pola S P, S P O, S P Pel, S P Ket, S P O Pel, S P O Ket, S P O Pel Ket.
3. Aplikasi memakai kosakata yang sudah ditentukan berdasarkan tugas sebelumnya

1.4 Batasan

Adapun asumsi - asumsi yang terdapat pada aplikasi parsing kalimat bahasa indonesia adalah

1. Hanya bisa memasukkan kalimat sederhana dengan pola S P, S P O, S P Pel, S P Ket, S P O Pel, S P O Ket, S P O Pel Ket.
2. Terdapat batas kalimat yang dapat diproses
3. Program mungkin tidak dapat menguraikan kalimat yang terlalu ambiguitas
4. Aplikasi tidak memiliki kemampuan untuk memperbarui data secara otomatis berdasarkan input.

BAB II

METODE CONTEXT-FREE GRAMMAR (CFG)

2.1 Konsep Dasar CFG

Context Free Grammar atau CFG merupakan suatu tata bahasa formal yang digunakan untuk menjelaskan struktur sintaks suatu bahasa. Dalam CFG, setiap aturan produksinya memiliki format $A \rightarrow B$, yang dimana A berperan sebagai pemproduksi dan B berperan sebagai hasil produksi. Adapun batasannya yaitu ruas kiri hanya dapat berupa simbol variabel dan ruas kanan yang dapat berupa simbol, variabel, terminal ataupun ϵ .

CFG bertujuan seperti tata bahasa regular, yaitu memberikan petunjuk tentang cara menghasilkan urutan-urutan dalam suatu bahasa.

Adapun contoh dari aturan produksi yang tersebut pada CFG yang melibatkan pola-pola seperti berikut:

$$X \rightarrow bY \mid Za$$

$$Y \rightarrow aY \mid b$$

$$Z \rightarrow bZ \mid \epsilon$$

Pada Context-Free Grammar atau CFG diperlukan penyederhanaan untuk menerapkan pembatasan, sehingga pohon penurunan yang dihasilkan tidak mengandung kompleksitas/kesulitan atau aturan produksi yang tidak memiliki makna. Berikut merupakan langkah-langkah dalam penyederhanaan CFG :

1. Eliminasi ϵ -production

Pada eliminasi ϵ -production, langkah pertama yaitu menghilangkan aturan produksi yang menghasilkan string kosong (ϵ). Hal itu dilakukan untuk mencegah terjadinya ambiguitas pada bahasa yang dihasilkan. Selanjutnya untuk menghilangkan ϵ -production, maka perlu mengganti setiap aturan produksi yang mengandung ϵ dengan aturan produksi baru yang tidak mengandung string kosong (ϵ).

2. Eliminasi unit production

Pada langkah ini, apabila terdapat hasil produksi yang lebih dari satu variabel, maka hasil dari produksi tersebut akan disubstitusi dengan hasil produksi grammar.

3. Eliminasi useless symbol

Dalam melakukan eliminasi useless symbol terdapat beberapa langkah yang harus dilakukan, yaitu untuk langkah pertama kenali simbol-simbol yang tidak dapat dipahami dari simbol awal kemudian hapus simbol yang tidak dapat dikenali tersebut dan juga aturan produksinya.

2.2 Four-Tuple CFG

Sebuah Tata Bahasa Bebas Konteks (CFG) G dijelaskan dalam bentuk 4 tuple menggunakan rumus: $G = (V, T, P, S)$, yang dapat diartikan sebagai berikut:

1. V (Set of variables):

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

2. T (Set of terminals):

$T = \{ \text{adik} \mid \text{ibu} \mid \text{nasi} \mid \text{hubungan} \mid \text{pertemanan} \mid \text{kota} \mid \text{keluarga} \mid \text{guru} \mid \text{prestasi} \mid \text{muridnya} \mid \text{teman} \mid \text{saat} \mid \text{kue} \mid \text{para} \mid \text{siswa} \mid \text{hari} \mid \text{keputusan} \mid \text{gadis} \mid \text{pemandangan} \mid \text{desa} \mid \text{suara} \mid \text{penyanyi} \mid \text{alunan} \mid \text{musik} \mid \text{rasa} \mid \text{sup} \mid \text{sayuran} \mid \text{es} \mid \text{teh} \mid \text{lemon} \mid \text{aroma} \mid \text{parfum} \mid \text{baju} \mid \text{rumah} \mid \text{lebam} \mid \text{gitar} \mid \text{kucing} \mid \text{mobil} \mid \text{bapak} \mid \text{kota} \mid \text{gedung} \mid \text{lapangan} \mid \text{bola} \mid \text{harga} \mid \text{sepatu} \mid \text{anak} \mid \text{kabar} \mid \text{ayah} \mid \text{pintu} \mid \text{dapur} \mid \text{warna} \mid \text{biru} \mid \text{ikan} \mid \text{mawar} \mid \text{merah} \mid \text{anjing} \mid \text{kaki} \mid \text{materi} \mid \text{konsep} \mid \text{pegawai} \mid \text{toko} \mid \text{pelanggan} \mid \text{matahari} \mid \text{pertunjukan} \mid \text{malam} \mid \text{semangat} \mid \text{tim} \mid \text{sepakbola} \mid \text{suasana} \mid \text{kerja} \mid \text{jalan} \mid \text{kantor} \mid \text{sedih} \mid \text{orang} \mid \text{dunia} \mid \text{kampung} \mid \text{udara} \mid \text{pagi} \mid \text{buku} \mid \text{kenanga} \mid \text{mobil} \mid \text{bunga} \mid \text{taman} \mid \text{cuaca} \mid \text{ruang} \mid \text{tamu} \mid \text{gedung-gedung} \mid \text{kaos} \mid \text{makanan} \mid \text{orang} \mid \text{bahan} \mid \text{baju} \mid \text{kucing} \mid \text{bocah} \mid \text{sekolah} \mid \text{kopi} \mid \text{harimau} \mid \text{halaman} \mid \text{kampung} \mid \text{anak} \mid \text{matematika} \mid \text{literasi} \mid \text{suhu} \mid \text{dasar} \mid \text{pipi} \mid \text{buah} \mid \text{tingkat} \mid \text{jeruk} \mid \text{oranye} \mid \text{langit} \mid \text{hitam} \mid \text{kacamata} \mid \text{pria} \mid \text{bentuk} \mid \text{wajah} \mid \text{wanita} \mid \text{laju} \mid \text{pesawat} \mid \text{tugas} \mid \text{waktu} \mid \text{sayur} \mid \text{memasak} \mid \text{tersebut} \mid \text{goreng} \mid \text{kuliah} \mid \text{diberikan} \mid \text{menyambut} \mid \text{libur} \mid \text{mengambil} \mid \text{bersepeda} \mid \text{membuat} \mid \text{mengecat} \mid \text{dirancang} \mid \text{membelikan} \mid \text{menganggap} \mid \text{tugas} \mid \text{bermain} \mid \text{menggonggong} \mid \text{berjalan} \mid \text{menari} \mid \text{dipukuli} \mid \text{menggambar} \mid \text{melangkah} \mid \text{menjelaskan} \mid \text{terbenam} \mid \text{menawan} \mid \text{menarik} \mid \text{baca} \mid$

belajar | tidur | berlari | diminum | berburu | ditempati | dibaca | dipukul | digunakan | berolahraga | terbang | menyelesaikan | menunggu | cepat | dekat | jauh | bahagia | bangga | gembira | ragu-ragu | kecil | cantik | indah | merdu | klasik | lembut | lezat | segar | harum | merah | bagus | sederhana | indah | tua | mahal | putih | jinak | lebam | antik | populer | baru | megah | luas | kecil | muda | susah | keras | lambat | gemulai | baik | sulit | sabar | ramah | seni | tinggi | nyaman | pasti | selamat | kuat | becek | sejuk | tebal | menarik | panas | dramatis | megah | besar | rendah | dingin | lebam | sedih | tua | kikir | lembut | lincah | gemuk | gesit | panas | ganas | rindu | pintar | lapang | rapat | sempit | lebam | longgar | pekat | bulat | lonjong | singkat | lama | dari | setelah | dengan | dalam | di | untuk | sangat | selalu | sekali | cukup | akan | sedang | ingin | masih | terlalu | dengan | akan | bersama | sudah | setiap | saya | itu | kami | dia | mereka | ini | aku | atlantis | andi | buwit | tabanan | afgan | randi | roy | dodu | amba | doni }

3. P (Set of productions) :

P = {

- K -> S P | S P O | S P Pel | S P Ket | S P O Pel | S P O Ket | S P O Pel Ket
- S -> NP
- P -> VP | PP | AdjP
- O -> NP
- Pel -> AdjP | VP | NP | PP
- Ket -> PP
- NP -> Noun | Pronoun | PropNoun | NP Pronoun | NP Noun | NP PropNoun | Num NP | NP AdjP | NP VP | PP NP
- VP -> Verb | VP Verb | Adv VP | AdjP VP | VP PP
- AdjP -> Adj | AdjP Adj | Adv AdjP | AdjP Adv
- PP -> Prep NP | Prep AdjP | Prep VP
- Noun -> adik | ibu | nasi | hubungan | pertemanan | kota | keluarga | guru | prestasi | muridnya | teman | saat | kue | para | siswa | hari |

keputusan | gadis | pemandangan | desa | suara | penyanyi | alunan | musik | rasa | sup | sayuran | es | teh | lemon | aroma | parfum | baju | rumah | lebam | gitar | kucing | mobil | bapak | kota | gedung | lapangan | bola | harga | sepatu | anak | kabar | ayah | pintu | dapur | warna | biru | ikan | mawar | merah | anjing | kaki | materi | konsep | pegawai | toko | pelanggan | matahari | pertunjukan | malam | semangat | tim | sepakbola | suasana | kerja | jalan | kantor | sedih | orang | dunia | kampung | udara | pagi | buku | kenanga | mobil | bunga | taman | cuaca | ruang | tamu | gedung-gedung | kaos | makanan | orang | bahan | baju | kucing | bocah | sekolah | kopi | harimau | halaman | kampung | anak | matematika | literasi | suhu | dasar | pipi | buah | tingkat | jeruk | oranye | langit | hitam | kacamata | pria | bentuk | wajah | wanita | laju | pesawat | tugas | waktu | sayur

- Verb -> tersebut | memasak | goreng | kuliah | diberikan | menyambut | libur | mengambil | bersepeda | membuat | mengecat | dirancang | membelikan | menganggap | tugas | bermain | menggonggong | berjalan | menari | dipukuli | menggambar | melangkahkan | menjelaskan | terbenam | menawan | menarik | baca | belajar | tidur | berlari | diminum | berburu | ditempati | dibaca | dipukul | digunakan | berolahraga | terbang | menyelesaikan | menunggu

- Adj -> cepat | dekat | jauh | bahagia | bangga | gembira | ragu-ragu | kecil | cantik | indah | merdu | klasik | lembut | lezat | segar | harum | merah | bagus | sederhana | indah | tua | mahal | putih | jinak | lebam | antik | populer | baru | megah | luas | kecil | muda | susah | keras | lambat | gemulai | baik | sulit | sabar | ramah | seni | tinggi | nyaman | pasti | selamat | kuat | becek | sejuk | tebal | menarik | panas | dramatis | megah | besar | rendah | dingin | lebam | sedih | tua | kikir | lembut | lincah | gemuk | gesit | panas | ganas | rindu | pintar |

lapang | rapat | sempit | lebam | longgar | pekat | bulat | lonjong |
singkat | lama

- Prep -> dari | setelah | dengan | dalam | di | untuk
- Adv -> sangat | selalu | sekali | cukup | akan | sedang | ingin | masih
| terlalu | dengan | akan | bersama | sudah
- Num -> setiap
- Pronoun -> saya | itu | kami | dia | mereka | ini | aku
- PropNoun -> atlantis | andi | buwit | tabanan | afgan | randi | roy |
dodi | amba | doni

4. S (Start symbol) : $S = \{K\}$

2.3 Konversi CFG Rules menjadi CNF (Chomsky Normal Form)

- $K \rightarrow S P \mid K1 O \mid K1 Pel \mid K1 Ket \mid K2 Pel \mid K2 Ket \mid K3 Ket \mid K4 Ket$
- $K1 \rightarrow S P$
- $K2 \rightarrow K1 O$
- $K3 \rightarrow K1 Pel$
- $K4 \rightarrow K2 Pel$
- $S \rightarrow NP Pronoun \mid NP Noun \mid NP PropNoun \mid Num NP \mid NP AdjP \mid NP VP$
| $PP NP \mid adik \mid ibu \mid nasi \mid aku \mid hubungan \mid pertemanan \mid kota \mid keluarga$
| $guru \mid prestasi \mid muridnya \mid teman \mid saat \mid kue \mid para \mid siswa \mid hari \mid keputusan$
| $gadis \mid pemandangan \mid desa \mid suara \mid penyanyi \mid alunan \mid musik \mid rasa \mid sup$
| $sayuran \mid es \mid teh \mid lemon \mid aroma \mid parfum \mid baju \mid rumah \mid gitar \mid kucing$
| $mobil \mid bapak \mid gedung \mid lapangan \mid bola \mid harga \mid sepatu \mid anak \mid kabar \mid ayah$
| $pintu \mid dapur \mid warna \mid biru \mid ikan \mid mawar \mid merah \mid anjing \mid kaki \mid materi$
| $konsep \mid pegawai \mid toko \mid pelanggan \mid matahari \mid pertunjukan \mid malam$
| $semangat \mid tim \mid sepakbola \mid suasana \mid kerja \mid jalan \mid kantor \mid sedih \mid orang$

dunia | kampung | udara | pagi | buku | kenanga | bunga | taman | cuaca |
 ruang | tamu | gedung-gedung | kaos | makanan | bahan | bocah | sekolah |
 kopi | harimau | halaman | matematika | literasi | suhu | dasar | dirancang |
 pipi | buah | tingkat | jeruk | oranye | langit | hitam | kacamata | pria | bentuk
 | wajah | wanita | laju | pesawat | tugas | waktu | saya | itu | kami | dia |
 mereka | ini | atlantis | andi | buwit | tabanan | afgan | randi | roy | dodi |
 amba | doni | sayur

- P -> VP Verb | Adv VP | AdjP VP | AdjP Adj | Adv AdjP | AdjP Adv | VP
 PP | tersebut | memasak | goreng | kuliah | diberikan | menyambut | libur |
 dipukuli | mengambil | bersepeda | membuat | mengecat | membelikan |
 menganggap | tugas | bermain | menggonggong | berjalan | menari |
 menggambar | melangkah | menjelaskan | terbenam | menawan |
 menarik | baca | belajar | tidur | berlari | diminum | berburu | ditempati |
 dibaca | dipukul | digunakan | berolahraga | terbang | menyelesaikan |
 menunggu | cepat | dekat | dirancang | jauh | bahagia | bangga | gembira |
 ragu-ragu | kecil | cantik | indah | merdu | klasik | lembut | lezat | segar |
 harum | merah | bagus | sederhana | tua | mahal | putih | jinak | antik |
 populer | baru | megah | luas | muda | susah | keras | lambat | gemulai | baik
 | sulit | sabar | ramah | seni | tinggi | nyaman | pasti | selamat | kuat | becek |
 sejuk | tebal | panas | dramatis | besar | rendah | dingin | lebam | sedih | kikir
 | lincah | gemuk | gesit | ganas | rindu | pintar | lapang | rapat | sempit |
 longgar | pekat | bulat | lonjong | singkat | lama

- O -> NP Pronoun | NP Noun | NP PropNoun | Num NP | NP AdjP | NP VP
 | PP NP | ibu | adik | nasi | hubungan | pertemanan | kota | keluarga | guru |
 prestasi | muridnya | teman | saat | kue | para | siswa | hari | keputusan |
 gadis | pemandangan | desa | suara | penyanyi | alunan | musik | rasa | sup |
 sayuran | es | teh | lemon | aroma | parfum | baju | aku | rumah | gitar |
 kucing | mobil | bapak | dirancang | gedung | lapangan | bola | harga | sepatu
 | anak | kabar | ayah | pintu | dapur | warna | biru | ikan | mawar | merah |
 anjing | kaki | materi | konsep | pegawai | toko | pelanggan | matahari |

pertunjukan | malam | semangat | tim | sepakbola | suasana | kerja | jalan | kantor | sedih | orang | dunia | kampung | udara | pagi | buku | kenanga | bunga | taman | cuaca | ruang | tamu | gedung-gedung | kaos | makanan | bahan | bocah | sekolah | kopi | harimau | halaman | matematika | literasi | suhu | dasar | pipi | buah | tingkat | jeruk | oranye | langit | hitam | kaca mata | pria | bentuk | wajah | wanita | laju | pesawat | tugas | waktu | saya | itu | kami | dia | mereka | ini | atlantis | andi | buwit | tabanan | afgan | randi | roy | dodi | amba | doni | sayur

- Pel -> AdjP Adj | Adv AdjP | AdjP Adv | VP Verb | Adv VP | AdjP VP | NP Pronoun | NP Noun | NP PropNoun | Num NP | NP AdjP | NP VP | Prep NP | Prep AdjP | Prep VP | PP NP | cepat | dekat | jauh | bahagia | bangga | gembira | ragu-ragu | kecil | cantik | indah | merdu | klasik | lembut | lezat | segar | harum | merah | bagus | sederhana | tua | mahal | putih | jinak | antik | populer | baru | megah | luas | muda | susah | keras | lambat | gemulai | baik | sulit | sabar | ramah | seni | tinggi | nyaman | pasti | selamat | kuat | becek | sejuk | tebal | menarik | panas | dramatis | besar | rendah | dingin | lebam | sedih | kikir | lincah | gemuk | gesit | ganas | rindu | pintar | lapang | rapat | sempit | longgar | pekat | bulat | lonjong | singkat | lama | memasak | tersebut | goreng | kuliah | diberikan | menyambut | dirancang | libur | mengambil | bersepeda | membuat | mengecat | membelikan | menganggap | tugas | bermain | menggonggong | berjalan | menari | menggambar | melangkah | menjelaskan | terbenam | menawan | baca | belajar | tidur | berlari | diminum | berburu | ditempati | dibaca | dipukul | digunakan | berolahraga | terbang | menyelesaikan | menunggu | ibu | adik | nasi | hubungan | pertemanan | kota | keluarga | guru | prestasi | muridnya | teman | saat | kue | para | siswa | hari | keputusan | gadis | pemandangan | desa | suara | penyanyi | alunan | musik | rasa | sup | sayuran | es | teh | lemon | aroma | parfum | baju | rumah | gitar | kucing | mobil | bapak | gedung | lapangan | bola | harga | sepatu | anak | kabar | ayah | pintu | dapur | warna | biru | lebam | ikan | mawar | anjing | kaki | materi | konsep | pegawai | toko | pelanggan | matahari | pertunjukan | malam | semangat |

tim | sepakbola | suasana | kerja | jalan | kantor | orang | dunia | kampung | udara | pagi | buku | kenanga | bunga | taman | cuaca | ruang | tamu | gedung-gedung | kaos | makanan | bahan | bocah | sekolah | kopi | harimau | halaman | matematika | literasi | suhu | dasar | pipi | buah | tingkat | jeruk | oranye | langit | hitam | kaca mata | pria | bentuk | wajah | wanita | laju | pesawat | waktu | saya | aku | itu | kami | dia | mereka | ini | atlantis | andi | buwit | tabanan | afgan | randi | roy | dodi | amba | doni | sayur

- Ket -> Prep NP | Prep AdjP | Prep VP
- NP -> NP Pronoun | NP Noun | NP PropNoun | Num NP | NP AdjP | NP VP | PP NP | ibu | adik | nasi | hubungan | pertemanan | kota | keluarga | guru | prestasi | muridnya | teman | saat | kue | para | siswa | hari | keputusan | gadis | pemandangan | desa | suara | penyanyi | alunan | musik | rasa | sup | dirancang | sayuran | es | teh | lemon | aroma | parfum | baju | rumah | gitar | kucing | mobil | bapak | gedung | lapangan | bola | lebam | harga | sepatu | anak | kabar | ayah | pintu | dapur | warna | biru | ikan | mawar | merah | anjing | kaki | materi | konsep | pegawai | toko | pelanggan | matahari | pertunjukan | malam | semangat | tim | sepakbola | suasana | kerja | jalan | kantor | sedih | orang | dunia | kampung | udara | pagi | buku | kenanga | bunga | taman | cuaca | ruang | tamu | gedung-gedung | kaos | makanan | bahan | bocah | sekolah | kopi | harimau | halaman | matematika | literasi | suhu | dasar | pipi | buah | tingkat | jeruk | oranye | langit | hitam | kaca mata | pria | bentuk | wajah | wanita | laju | pesawat | tugas | waktu | saya | itu | aku | kami | dia | mereka | ini | atlantis | andi | buwit | tabanan | afgan | randi | roy | dodi | amba | doni | sayur
- VP -> VP Verb | Adv VP | AdjP VP | VP PP | memasak | tersebut | goreng | kuliah | diberikan | menyambut | libur | mengambil | bersepeda | membuat | mengecat | membelikan | menganggap | tugas | bermain | menggonggong | berjalan | menari | menggambar | melangkah | menjelaskan | terbenam | menawan | menarik | baca | belajar | tidur | dirancang | berlari | diminum |

berburu | ditempati | dibaca | dipukul | digunakan | digunakan | berolahraga
| terbang | menyelesaikan | menunggu

- AdjP -> AdjP Adj | Adv AdjP | AdjP Adv | cepat | dekat | jauh | bahagia | bangga | gembira | ragu-ragu | kecil | cantik | indah | merdu | klasik | lembut | lezat | segar | harum | merah | bagus | sederhana | tua | mahal | putih | jinak | antik | populer | baru | megah | luas | muda | susah | keras | lambat | gemulai | baik | sulit | sabar | ramah | seni | tinggi | nyaman | pasti | selamat | kuat | becek | sejuk | tebal | menarik | panas | dramatis | besar | rendah | dingin | lebam | sedih | kikir | lincah | gemuk | gesit | ganas | rindu | pintar | lapang | rapat | sempit | longgar | pekat | bulat | lonjong | singkat | lama
- PP -> Prep NP | Prep AdjP | Prep VP
- Noun -> ibu | adik | nasi | hubungan | pertemanan | kota | keluarga | guru | prestasi | muridnya | teman | saat | kue | para | siswa | hari | keputusan | gadis | pemandangan | desa | suara | penyanyi | alunan | musik | rasa | sup | sayuran | es | teh | lemon | aroma | parfum | baju | rumah | lebam | gitar | kucing | mobil | bapak | kota | gedung | lapangan | bola | harga | sepatu | anak | kabar | ayah | pintu | dapur | warna | biru | ikan | mawar | merah | anjing | kaki | materi | konsep | pegawai | toko | pelanggan | matahari | pertunjukan | malam | semangat | tim | sepakbola | suasana | kerja | jalan | kantor | sedih | orang | dunia | kampung | udara | pagi | buku | kenanga | mobil | bunga | taman | cuaca | ruang | tamu | gedung-gedung | kaos | makanan | orang | bahan | baju | kucing | bocah | sekolah | kopi | harimau | halaman | kampung | anak | matematika | literasi | suhu | dasar | pipi | buah | tingkat | jeruk | oranye | langit | hitam | kacamata | pria | bentuk | wajah | wanita | laju | pesawat | tugas | waktu | sayur
- Verb -> memasak | tersebut | goreng | kuliah | diberikan | menyambut | libur | mengambil | bersepeda | membuat | mengecat | dirancang | membelikan | menganggap | tugas | bermain | menggonggong | berjalan |

menari | dipukuli | menggambar | melangkah | menjelaskan | terbenam |
menawan | menarik | baca | belajar | tidur | berlari | diminum | berburu |
ditempati | dibaca | dipukul | digunakan | berolahraga | terbang |
menyelesaikan | menunggu

- Adj -> cepat | dekat | jauh | bahagia | bangga | gembira | ragu-ragu | kecil |
cantik | indah | merdu | klasik | lembut | lezat | segar | harum | merah | bagus
| sederhana | indah | tua | mahal | putih | jinak | lebam | antik | populer | baru
| megah | luas | kecil | muda | susah | keras | lambat | gemulai | baik | sulit |
sabar | ramah | seni | tinggi | nyaman | pasti | selamat | kuat | becek | sejuk |
tebal | menarik | panas | dramatis | megah | besar | rendah | dingin | lebam |
sedih | tua | kikir | lembut | lincah | gemuk | gesit | panas | ganas | rindu |
pintar | lapang | rapat | sempit | lebam | longgar | pekat | bulat | lonjong |
singkat | lama
- Prep -> dari | setelah | dengan | dalam | di | untuk
- Adv -> sangat | selalu | sekali | cukup | akan | sedang | ingin | masih | terlalu
| dengan | akan | bersama | sudah
- Num -> setiap
- Pronoun -> saya | itu | kami | dia | mereka | ini | aku
- PropNoun -> atlantis | andi | buwit | tabanan | afgan | randi | roy | dodi |
amba | doni

2.4 Algoritma CYK

Algoritma CYK merupakan algoritma yang digunakan untuk menunjukkan apakah sebuah kata w di-generate oleh grammar context free atau tidak. Algoritma CYK ini memanfaatkan fakta bahwa kalimat lengkap dibuat dengan cara menggabungkan sub-pohon. Sebuah pohon akan valid apabila

sub-pohonnya juga valid. Proses algoritma CYK melibatkan pembentukan tabel segitiga, di mana ukuran tabel ditentukan oleh panjang string yang diberikan. Setiap baris dalam tabel mencerminkan panjang substring dari kata yang sedang diproses. Dimulai dari bagian bawah tabel yang berisi string dengan panjang 1, baris-baris di atasnya merepresentasikan string dengan panjang yang semakin besar.

Adapun langkah-langkah dalam algoritma CYK, yaitu sebagai berikut:

1. Membuat tabel terlebih dahulu dengan panjang simbol yang akan diperiksa. Setiap sel pada tabel akan berisi kumpulan simbol non-terminal yang potensial menghasilkan sub-rangkaian yang sesuai dengan posisi sel tersebut.
2. Inisialisasi isi sel-sel pada baris pertama tabel dengan kumpulan simbol non-terminal yang dapat menghasilkan simbol terminal pada posisi yang sesuai di rangkaian simbol. Proses ini dilakukan dengan memeriksa aturan produksi pada grammar context free dan mencocokkan simbol-simbol terminal dengan simbol-simbol non-terminal yang menghasilkannya.
3. Melakukan pengisian tabel secara berurutan dari baris kedua hingga baris terakhir. Untuk setiap sel pada baris tersebut, mencari tau semua kemungkinan pemisahan sub-rangkaian yang sesuai dengan posisi sel tersebut. Gabungkan kumpulan simbol non-terminal yang dapat menghasilkan sub-rangkaian tersebut dengan memanfaatkan aturan produksi pada grammar context free . Isi sel tersebut dengan kumpulan simbol non-terminal yang telah dihasilkan.
4. Hasil apabila simbol awal (start symbol) terdapat pada sel terakhir tabel, maka rangkaian simbol tersebut diterima oleh tata bahasa bebas konteks. Namun, jika tidak maka rangkaian simbol tersebut tidak dapat diterima oleh grammar context free.

Berikut merupakan contoh grammar pada algoritma CYK :

1. Kalimat : “Harimau itu sangat ganas saat berburu”

	0	1	2	3	4	5
0	{K, S, O, Pel, NP, K1, K2, K3, K4}	<NA>	<NA>	<NA>	<NA>	<NA>
1	{K, S, O, Pel,	{K, S, O,	<NA>	<NA>	<NA>	<NA>

	NP, K2, K3}	Pel, NP, K1, K2, K3, K4}				
2	{K, S, O, Pel, NP, K1}	{K, S, Pel, NP, K2, K3}	∅	<NA>	<NA>	<NA>
3	∅	{K, S, O, Pel, NP, K1}	∅	∅	<NA>	<NA>
4	{S, O, Pel, NP}	∅	{P, Pel, AdjP}	∅	{K, S, O, Pel, NP, K1}	<NA>
5	{S, O, Pel, NP, Noun}	{S, O, Pel, NP, Pronoun}	{Adv}	{P, Pel, AdjP, Adj}	{S, O, Pel, NP, Noun}	{P, Pel, VP, Verb}
	Harimau	itu	sangat	ganas	saat	berburu

2. Kalimat : Buku ini terlalu tebal untuk dibaca

	0	1	2	3	4	5
0	{K, K1, K3}	<NA>	<NA>	<NA>	<NA>	<NA>
1	∅	{K, K1, K3}	<NA>	<NA>	<NA>	<NA>
2	{K, S, O, Pel, NP, K1}	∅	∅	<NA>	<NA>	<NA>
3	∅	{K, S, O, Pel, NP, K1}	∅	∅	<NA>	<NA>
4	{S, O, Pel, NP}	∅	∅	∅	{P, Pel, K et, PP}	<NA>
5	{S, O, Pel, NP, Noun}	{S, O, Pel, NP, Pronoun}	{Adv}	{P, Pel, AdjP, Adj}	{Prep}	{P, Pel, VP, Verb}
	Buku	ini	terlalu	tebal	untuk	dibaca

Kelebihan dan keterbatasan dari algoritma CYK, yaitu sebagai berikut :

Kelebihan :

1. Penerapan Algoritma CYK memungkinkan pengecekan apakah suatu urutan masukan sesuai dengan aturan tata bahasa bebas konteks tertentu.

2. Algoritma ini memiliki kompleksitas waktu yang efisien, yaitu $O(n^3)$, di mana n adalah panjang urutan yang sedang diperiksa.

Keterbatasan:

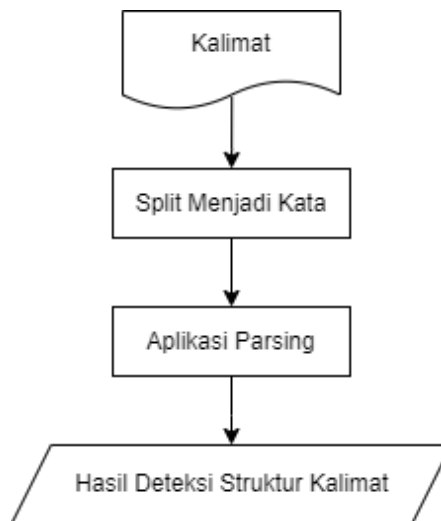
1. Algoritma CYK hanya relevan untuk tata bahasa bebas konteks tertentu. Pada tata bahasa dengan struktur yang lebih kompleks, algoritma ini mungkin kurang efektif.
2. Implementasi dari Algoritma CYK ini melibatkan pembuatan tabel segitiga atas yang memerlukan alokasi ruang memori yang signifikan, terutama pada urutan masukan yang panjang.

BAB III

ANALISIS DAN DESAIN

3.1 Desain Eksperimen

3.1.1 Skenario Eksperimen



Gambar 3.1. Bagan Pengujian Aplikasi Parsing

Pengujian pada aplikasi dilakukan dengan melakukan uji coba berbagai macam kalimat Berbahasa Indonesia dengan pola kalimat sederhana. Dari semua data uji yang dimasukkan, sistem

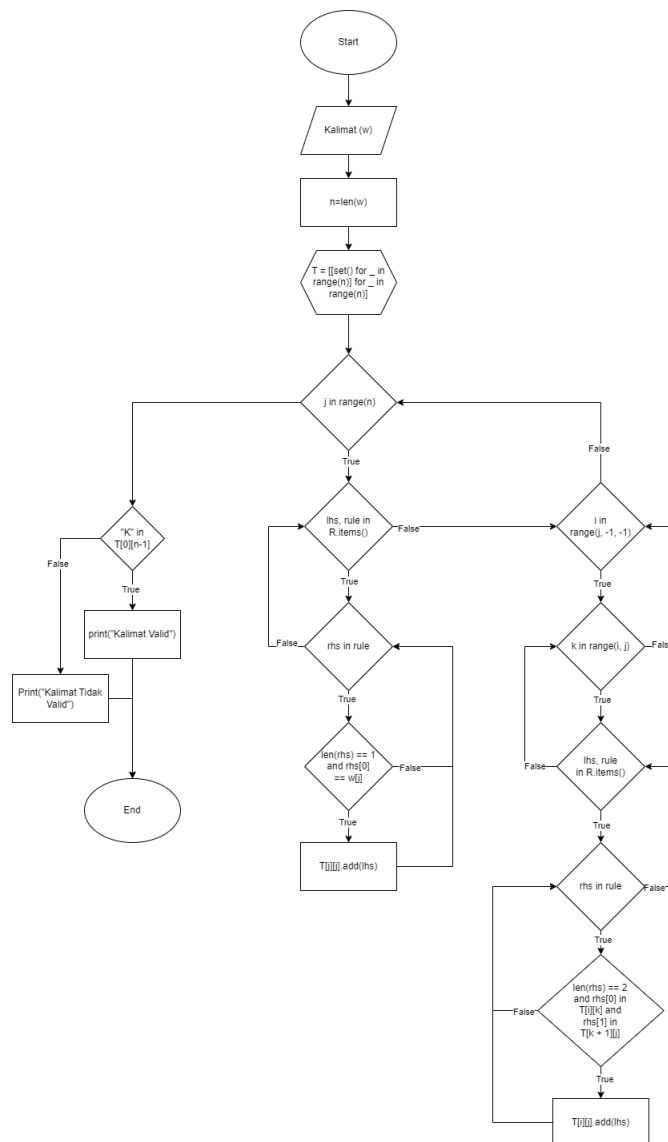
melakukan parsing kalimat tersebut. Keluarannya adalah berupa status valid atau tidak valid suatu kalimat dan pohon parsingnya. Analisisnya dilakukan dengan membandingkan kesesuaian antara data yang berasal dari sumber file Indonesian Sentences yang disediakan ibu dosen dan tugas-tugas sebelumnya dengan hasil pada sistem deteksi yang menggunakan algoritma CYK Parser. Kalimat yang diujikan adalah kalimat sederhana dengan pola kalimat S P, S P O, S P Pel, S P O Ket, S P Pel Ket, Kalimat juga tidak boleh ambigu dan daftar kata yang digunakan berdasarkan file Indonesian Sentences yang disediakan oleh ibu dosen dan tugas-tugas sebelumnya.

3.1.2 Ukuran Evaluasi

Ukuran yang kami gunakan dalam evaluasi sistem aplikasi parsing ini berupa persentase ketepatan parsing yang dilakukan oleh aplikasi parsing kami. Semakin mendekati seratus persen semakin sempurna aplikasi parsing yang dibuat. Selain itu terdapat batasan dalam pengujian dimana kalimat yang dapat diuji adalah kalimat Bahasa Indonesia sederhana dan tidak ambigu.

3.2 Desain Aplikasi

3.2.1 Flowchart/Arsitektur Sistem



Gambar 3.2. Flowchart Algoritma CYK

Pada gambar 3.2, menggambarkan alur algoritma CYK. Pada gambar diatas dijelaskan bahwa sistem akan menerima sebuah kalimat yang disimbolkan oleh variabel w. Kemudian sistem akan mendeklarasikan variabel n yang merupakan panjang dari kalimat, sistem juga akan mendeklarasikan tabel filing untuk proses pemeriksaan CYK. Setelah itu akan dilakukan proses perulangan

untuk tabel filing dan pengecekan rule CFN. Jika pada baris paling atas dan kolom pertama pada tabel filing maka sistem akan mengembalikan pesan “Kalimat Valid” dan apabila tidak ditemukan K pada tabel filing maka sistem akan mengembalikan pesan “Kalimat Tidak Valid”

3.2.2 Mockup

The mockup shows a web browser window with the title "Aplikasi Parser Kalimat Bahasa Indonesia". The address bar shows "http://localhost:8000". The main heading is "Aplikasi Parsing Kalimat Baku Berbahasa Indonesia". Below this, a description states: "Aplikasi ini merupakan aplikasi yang digunakan untuk melakukan parsing kalimat berbahasa Indonesia dengan pola kalimat sederhana sesuai dengan Pedoman Umum Ejaan Bahasa Indonesia (PUEBI) dan Tata Bahasa Baku Bahasa Indonesia Edisi Keempat." A text input field contains "Saya pergi ke pasar". A green "Parsing" button is next to it. Below the button, a green box displays the result: "Berdasarkan hasil pemeriksaan, kalimat **saya pergi ke pasar** **VALID** Sesuai dengan pola dasar bahasa Indonesia".

Gambar 3.3. Desain *Mockup* jika kalimat yang dimasukan valid

The mockup shows the same web browser window. Below the heading "Aplikasi Parsing Kalimat Baku Berbahasa Indonesia", there is a section titled "Table Filling". It contains a table with 4 columns and 4 rows. The columns are labeled "Saya", "Pergi", "Ke", and "Pasar". The rows contain the following patterns:

{K}	{}		
{K, K1}	{}	{P, Ket, PP}	
{S, O, Pel, Ket, NP, Pronoun}	{P, Pel, VP, Verb}	{Prep}	{S, O, Pel, Ket, NP, Noun}
Saya	Pergi	Ke	Pasar

Gambar 3.4. Desain *Mockup* jika kalimat yang dimasukan valid

Pada gambar 3.3 dan 3.4, Desain *Mockup* jika kalimat yang dimasukan valid. Pada gambar diatas dijelaskan bahwa jika aplikasi menerima kalimat bahasa Indonesia yang sesuai PUEBI dan TBBBI. Jika kalimat yang dimasukan sudah sesuai maka aplikasi akan mengembalikan pesan Valid dan Tabel filling dari kalimat yang dimasukan.

The screenshot shows a web browser window with the title 'Aplikasi Parser Kalimat Bahasa Indonesia'. The address bar shows 'http://localhost:8000'. The main heading is 'Aplikasi Parsing Kalimat Baku Berbahasa Indonesia'. Below it, a description states: 'Aplikasi ini merupakan aplikasi yang digunakan untuk melakukan parsing kalimat berbahasa Indonesia dengan pola kalimat sederhana sesuai dengan Pedoman Umum Ejaan Bahasa Indonesia (PUEBI) dan Tata Bahasa Baku Bahasa Indonesia Edisi Keempat.' A label 'Masukan kalimat untuk diparsing:' is followed by a text input field containing 'Setiap orang itu'. A green 'Parsing' button is below the input. A pink feedback box contains the text: 'Berdasarkan hasil pemeriksaan, kalimat **Setiap orang itu** **TIDAK VALID** Sesuai dengan pola dasar bahasa Indonesia'.

Gambar 3.5. *Mockup* jika kalimat yang dimasukan tidak valid

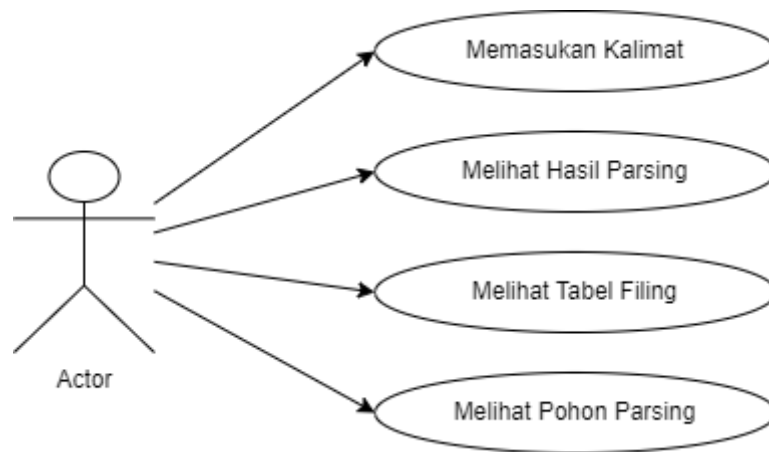
The screenshot shows the same web application interface. Below the pink feedback box, there is a section titled 'Table Filling' on an orange background. It contains a table with three columns labeled 'Setiap', 'Orang', and 'Itu' at the bottom. The table cells contain syntactic tree notations: the first column has two cells with '[S, O, Pel, Ket, NP]' and '[S, O, Pel, Ket, NP]'; the second column has one cell with '[S, O, Pel, Ket, NP]' and another with '[S, O, Pel, Ket, NP, Noun]'; the third column has one cell with '[S, O, Pel, Ket, NP, Pronoun]'. The first cell of the first column also contains '[Ket, NumP, Num]'.

[S, O, Pel, Ket, NP]	[S, O, Pel, Ket, NP]	[S, O, Pel, Ket, NP, Pronoun]
[S, O, Pel, Ket, NP]	[S, O, Pel, Ket, NP, Noun]	
[Ket, NumP, Num]		
Setiap	Orang	Itu

Gambar 3.6. Desain *Mockup* jika kalimat yang dimasukan tidak valid

Pada gambar 3.5 dan 3.6, Desain *Mockup* jika kalimat yang dimasukan valid. Pada gambar diatas dijelaskan bahwa jika aplikasi menerima kalimat bahasa Indonesia yang sesuai PUEBI dan TBBBI. Jika kalimat yang dimasukan tidak sesuai maka aplikasi akan mengembalikan pesan Tidak Valid dan Tabel filling dari kalimat yang dimasukan.

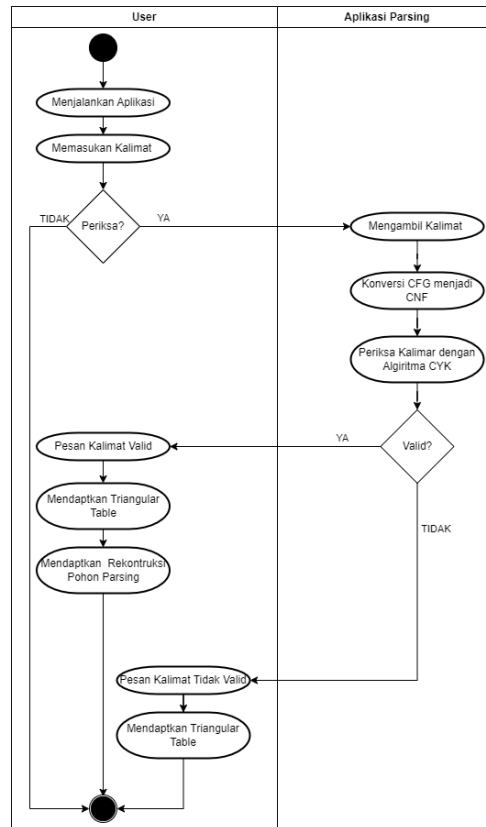
3.2.3 Use Case Diagram



Gambar 3.7. *Use Case Diagram* Aplikasi Parsing

Pada gambar 3.7, menggambarkan apa yang dapat dilakukan oleh pengguna (user) sebagai aktor terhadap aplikasi parsing. Pengguna dapat memasukkan kalimat ke dalam aplikasi parsing, pengguna dapat melihat hasil parsing, jika kalimat valid maka user juga dapat melihat tabel filing dan pohon parsing.

3.2.4 Activity Diagram



Gambar 3.8. Activity Diagram Aplikasi Parsing

Pada gambar 3.8, ditampilkan sebuah UML berupa activity diagram. Activity diagram memperlihatkan urutan aktifitas proses secara jelas sehingga lebih mudah untuk dikomunikasikan ke berbagai pihak. Pada gambar 3.4 activity diagram menunjukkan bagaimana diagram aktivitas dari sebuah aplikasi parsing dan penggunaanya berinteraksi. Pada UML terdapat 2 kolom, dengan deskripsi user dan aplikasi parsing. Pada kolom user menjelaskan kegiatan yang dilakukan oleh pengguna kepada sebuah sistem dan kolom aplikasi parsing menjelaskan kegiatan yang dilakukan pada aplikasi parsing yang akan disusun.

3.3 Implementasi

3.3.1 Source Code

App.py

```
import graphviz
```

```

import streamlit as st
from graphviz import Source

TRIANGULAR_TABLE = {}
PARSE_TREE = None
PREV_NODE = None

RESULT = {}

def remove_unit_production(keyList):
    global RESULT
    for key, value in RESULT.items():
        if key in keyList:
            tempList = []
            for prod in value:
                if len(prod.split(" ")) == 2:
                    tempList.append(prod)
                else:
                    for i in RESULT[prod]:
                        if i not in tempList:
                            tempList.append(i)
            RESULT[key] = tempList

def get_set_of_production():
    global RESULT
    RESULT.clear()
    f = open("./set_of_production.txt", "r", encoding="utf-8")
    for lines in f:
        line = lines.splitlines()
        line = line[0].split(" -> ")
        lhs = line[0]
        rhs = line[1].split(" | ")
        if lhs in RESULT.keys():
            RESULT[lhs].extend(rhs)
        else:
            RESULT[lhs] = rhs
    f.close()
    for key, value in RESULT.items():
        if key == "PropNoun":
            tempList = []
            for val in value:
                if val not in tempList:
                    tempList.append(val.lower())
            RESULT[key] = tempList
    phrases = ["NumP", "AdvP", "AdjP", "PP", "NP", "VP"]
    remove_unit_production(phrases)
    patterns = ["S", "P", "O", "PeI", "Ket"]
    remove_unit_production(patterns)
    tempList = []
    tempDict = {}
    counter = 1
    for key, value in RESULT.items():

```



```

if key == "K":
    for val in value:
        if len(val.split(" ")) > 2:
            temp = val.split(" ")
            while len(temp) > 2:
                checkStr = temp[0] + " " + temp[1]
                isFound = False
                for k, v in tempDict.items():
                    if checkStr == v:
                        isFound = True
                        temp.pop(0)
                        temp.pop(0)
                        temp.insert(0, k)
                        break
                if not isFound:
                    tempDict["K" + str(counter)] = checkStr
                    temp.pop(0)
                    temp.pop(0)
                    temp.insert(0, "K" + str(counter))
                    counter += 1
            tempList.append(" ".join(temp))
        else:
            tempList.append(val)
    RESULT[key] = tempList
for key, value in tempDict.items():
    RESULT[key] = [value]
return RESULT

def get_raw_set_of_production():
    global RESULT
    RESULT.clear()
    f = open("./set_of_production.txt", "r", encoding="utf-8")
    for lines in f:
        line = lines.splitlines()
        line = line[0].split(" -> ")
        lhs = line[0]
        rhs = line[1].split(" | ")
        if lhs in RESULT.keys():
            RESULT[lhs].extend(rhs)
        else:
            RESULT[lhs] = rhs
    f.close()
    for key, value in RESULT.items():
        if key == "PropNoun":
            tempList = []
            for val in value:
                if val not in tempList:
                    tempList.append(val.lower())
            RESULT[key] = tempList
    tempList = []
    tempDict = {}
    counter = 1

```

```

for key, value in RESULT.items():
    if key == "K":
        for val in value:
            if len(val.split(" ")) > 2:
                temp = val.split(" ")
                while len(temp) > 2:
                    checkStr = temp[0] + " " + temp[1]
                    isFound = False
                    for k, v in tempDict.items():
                        if checkStr == v:
                            isFound = True
                            temp.pop(0)
                            temp.pop(0)
                            temp.insert(0, k)
                            break
                    if not isFound:
                        tempDict["K" + str(counter)] = checkStr
                        temp.pop(0)
                        temp.pop(0)
                        temp.insert(0, "K" + str(counter))
                        counter += 1
                tempList.append(" ".join(temp))
            else:
                tempList.append(val)
        RESULT[key] = tempList
for key, value in tempDict.items():
    RESULT[key] = [value]
return RESULT

def is_accepted(inputString):
    global TRIANGULAR_TABLE
    TRIANGULAR_TABLE.clear()
    prodRules = get_set_of_production()
    inputString = inputString.lower().split(" ")
    for i in range(1, len(inputString)+1):
        for j in range(i, len(inputString)+1):
            TRIANGULAR_TABLE[(i,j)] = []
    for i in reversed(range(1, len(inputString)+1)):
        for j in range(1, i+1):
            if (j == j + len(inputString) - i):
                tempList = []
                for key, value in prodRules.items():
                    for val in value:
                        if (val == inputString[j-1] and key not in tempList):
                            tempList.append(key)
                TRIANGULAR_TABLE[(j, j + len(inputString) - i)] = tempList
            else:
                tempList = []
                resultList = []
                for k in range(len(inputString) - i):
                    first = TRIANGULAR_TABLE[(j,j+k)]
                    second = TRIANGULAR_TABLE[(j+k+1,j+len(inputString) - i)]

```

```

        for fi in first:
            for se in second:
                if (fi + " " + se not in tempList):
                    tempList.append(fi + " " + se)
            for key, value in prodRules.items():
                for val in value:
                    if (val in tempList and key not in resultList):
                        resultList.append(key)
            TRIANGULAR_TABLE[(j,j+len(inputString) - i)] = resultList
    if "K" in TRIANGULAR_TABLE[(1, len(inputString))]:
        return True
    else:
        return False

def is_parent(posX, posY, limit, check, prodRules):
    global TRIANGULAR_TABLE
    x = posX
    y = posY
    while posX > 1 and posY <= limit:
        posX -= 1
        if TRIANGULAR_TABLE[(posX, posY)] != []:
            backVar = TRIANGULAR_TABLE[(posX, posY)][-1]
            for i in prodRules[backVar]:
                if check in i.split(" "):
                    return [True, posX, posY]
            return [False, None, None]
    posX = x
    posY = y
    while posX >= 1 and posY < limit:
        posY += 1
        if TRIANGULAR_TABLE[(posX, posY)] != []:
            backVar = TRIANGULAR_TABLE[(posX, posY)][-1]
            for i in prodRules[backVar]:
                if check in i.split(" "):
                    return [True, posX, posY]
            return [False, None, None]
    return [False, None, None]

def search_left(listVar, checkPos, curPost, posX, posY, limit, prodRules):
    global PARSE_TREE
    global PREV_NODE

    structureTier = ["S", "P", "O", "PeI", "Ket"]

    if len(listVar) == 1:
        if (listVar[0] == "K"):
            PARSE_TREE.edge("K", PREV_NODE)
            return
        else:
            res, x, y = is_parent(posX, posY, limit, listVar[curPost], prodRules)
            if res == True:
                temp = TRIANGULAR_TABLE[(x, y)][-1]

```

```

        parentNode = str(temp + "(" + str(x) + "," + str(y) + ")")
        PARSE_TREE.edge(parentNode, PREV_NODE)
        PREV_NODE = parentNode
        search_left(TRIANGULAR_TABLE[(x, y)],
len(TRIANGULAR_TABLE[(x, y))]-2, len(TRIANGULAR_TABLE[(x, y))]-1, x, y,
limit, prodRules)
    else:
        return
    else:
        if listVar[checkPos] in structureTier and checkPos == 0:
            res, x, y = is_parent(posX, posY, limit, listVar[checkPos], prodRules)
            if res == True:
                parentNode = str(listVar[checkPos] + "(" + str(posX) + "," + str(posY)
+ ")")
                PARSE_TREE.edge(parentNode, PREV_NODE)
                PREV_NODE = parentNode
                PARSE_TREE.edge("K", parentNode)
                return
            else:
                res2, x2, y2 = is_parent(posX, posY, limit, listVar[curPost],
prodRules)
                if res2 == True:
                    temp = TRIANGULAR_TABLE[(x2, y2)][-1]
                    parentNode = str(temp + "(" + str(x2) + "," + str(y2) + ")")
                    PARSE_TREE.edge(parentNode, PREV_NODE)
                    PREV_NODE = parentNode
                    search_left(TRIANGULAR_TABLE[(x2, y2)],
len(TRIANGULAR_TABLE[(x2, y2))]-2, len(TRIANGULAR_TABLE[(x2, y2))]-1,
x2, y2, limit, prodRules)
                else:
                    return

        elif listVar[checkPos] in structureTier and checkPos > 0:
            res, x, y = is_parent(posX, posY, limit, listVar[checkPos], prodRules)
            if res == True:
                parentNode = str(listVar[checkPos] + "(" + str(posX) + "," + str(posY)
+ ")")
                PARSE_TREE.edge(parentNode, PREV_NODE)
                PREV_NODE = parentNode
                search_left(TRIANGULAR_TABLE[(x, y)],
len(TRIANGULAR_TABLE[(x, y))]-2, len(TRIANGULAR_TABLE[(x, y))]-1, x, y,
limit, prodRules)
            else:
                search_left(listVar, checkPos-1, curPost, posX, posY, limit,
prodRules)

        elif listVar[checkPos] not in structureTier and checkPos == 0:
            parentNode = str(listVar[checkPos] + "(" + str(posX) + "," + str(posY) +
)")")
            PARSE_TREE.edge(parentNode, PREV_NODE)
            PREV_NODE = parentNode
            res, x, y = is_parent(posX, posY, limit, listVar[checkPos], prodRules)

```

```

    if res:
        temp = TRIANGULAR_TABLE[(x, y)][-1]
        parentNode = str(temp + " (" + str(x) + "," + str(y) + ")")
        PARSE_TREE.edge(parentNode, PREV_NODE)
        PREV_NODE = parentNode
        search_left(TRIANGULAR_TABLE[(x, y)],
len(TRIANGULAR_TABLE[(x, y)]-2, len(TRIANGULAR_TABLE[(x, y)]-1, x, y,
limit, prodRules)

    elif listVar[checkPos] not in structureTier and checkPos > 0:
        isFound = False
        for i in prodRules[listVar[checkPos]]:
            if listVar[checkPos] in i.split(" "):
                parentNode = str(listVar[checkPos] + " (" + str(posX) + "," +
str(posY) + ")")
                PARSE_TREE.edge(parentNode, PREV_NODE)
                PREV_NODE = parentNode
                isFound = True
                break
        if isFound:
            search_left(listVar, checkPos-1, checkPos, posX, posY, limit,
prodRules)

def get_parse_tree(inputString):
    if is_accepted(inputString):
        global TRIANGULAR_TABLE
        global PARSE_TREE
        global PREV_NODE

        PARSE_TREE = graphviz.Graph("G", strict=True)
        PARSE_TREE.attr("node", shape="circle")
        PARSE_TREE.node("K")

        prodRules = get_raw_set_of_production()
        inputString = inputString.lower().split(" ")

        for i in range(1, len(inputString)+1):
            baseList = TRIANGULAR_TABLE[(i, i)]
            childNode = str(inputString[i-1] + " (" + str(i) + "," + str(i) + ")")
            parentNode = str(baseList[-1] + " (" + str(i) + "," + str(i) + ")")
            PARSE_TREE.edge(parentNode, childNode)
            PREV_NODE = parentNode
            if (len(baseList) == 1):
                search_left(baseList, len(baseList)-1, len(baseList)-1, i, i,
len(inputString), prodRules)
            else:
                search_left(baseList, len(baseList)-2, len(baseList)-1, i, i,
len(inputString), prodRules)
            return PARSE_TREE
        else:
            return None

```

```

def get_table_element(inputString):
    global TRIANGULAR_TABLE
    result = []
    n = len(inputString.split(" "))
    for i in range(1, n+1):
        temp = []
        for j in range(i):
            res = TRIANGULAR_TABLE[(j+1, n-i+j+1)]
            if len(res) == 0:
                temp.append("\u2205")
            else:
                temp.append("{ " + ", ".join(res) + "}")
        result.append(temp)
    result.append(inputString.split(" "))
    return result

def main():
    st.title("Aplikasi Parsing Kalimat Baku Berbahasa Indonesia")
    st.markdown("""
    <div style="text-align: justify;">
        Aplikasi ini merupakan aplikasi yang digunakan untuk melakukan parsing
        kalimat berbahasa indonesia dengan pola kalimat sederhana sesuai dengan
        pedoman Umum Ejaan Bahasa Indonesia (PUEBI) dan Tata Bahasa Baku
        Bahasa Indonesia Edisi Keempat.
    </div> <br>
    """, unsafe_allow_html=True)

    input_string = st.text_input("Masukan kalimat untuk diparsing:")

    if st.button("Parse"):
        parse_tree = get_parse_tree(input_string)
        triangular_table = get_table_element(input_string)

        if parse_tree is not None:
            #st.subheader("Parse Tree:")
            #st.graphviz_chart(parse_tree)

            st.markdown(
                f"""<div style="background-color: #8AFF8A; padding: 10px;
                border-radius: 10px; text-align: center;">
                    Berdasarkan hasil pemeriksaan, <span style="font-weight:
                bold;">{input_string}</span><br>
                    <span style="font-weight: bold; color: #07AC17;">VALID</span><br>
                    Sesuai dengan pola dasar bahasa Indonesia
                </div>""",
                unsafe_allow_html=True
            )

            st.subheader("Triangular Table:")
            st.table(triangular_table[:-1]) # Display the table without the last row

        else:

```

```

st.markdown(
f"""<div style="background-color: #FF8A8A; padding: 10px;
border-radius: 10px; text-align: center;">
    Berdasarkan hasil pemeriksaan, <span style="font-weight:
bold;">{input_string}</span><br>
    <span style="font-weight: bold; color: #D81717;">TIDAK
VALID</span><br>
    Tidak sesuai dengan pola dasar bahasa Indonesia
</div>""",
unsafe_allow_html=True
)

if __name__ == "__main__":
    main()

```

set_of_production.txt

```

K -> S P
K -> S P O
K -> S P Pel
K -> S P Ket
K -> S P O Pel
K -> S P O Ket
K -> S P Pel Ket
K -> S P O Pel Ket
S -> NP
P -> VP | PP | AdjP
O -> NP
Pel -> AdjP | VP | NP | PP
Ket -> PP
NP -> Noun | Pronoun | PropNoun | NP Pronoun | NP Noun | NP PropNoun |
Num NP | NP AdjP | NP VP | PP NP
VP -> Verb | VP Verb | Adv VP | AdjP VP | VP PP
AdjP -> Adj | AdjP Adj | Adv AdjP | AdjP Adv
PP -> Prep NP | Prep AdjP | Prep VP
Noun -> ibu | adik | nasi | hubungan | pertemanan | kota | keluarga | guru |
prestasi | muridnya | teman | saat | kue | para | siswa | hari | keputusan | gadis |
pemandangan | desa | suara | penyanyi | alunan | musik | rasa | sup | sayuran |
es | teh | lemon | aroma | parfum | baju | rumah | lebam | gitar | kucing | mobil |
bapak | kota | gedung | lapangan | bola | harga | sepatu | anak | kabar | ayah |
pintu | dapur | warna | biru | ikan | mawar | merah | anjing | kaki | materi |
konsep | pegawai | toko | pelanggan | matahari | pertunjukan | malam |
semangat | tim | sepakbola | suasana | kerja | jalan | kantor | sedih | orang |
dunia | kampung | udara | pagi | buku | kenanga | mobil | bunga | taman | cuaca |
ruang | tamu | gedung-gedung | kaos | makanan | orang | bahan | baju |
kucing | bocah | sekolah | kopi | harimau | halaman | kampung | anak |
matematika | literasi | suhu | dasar | pipi | buah | tingkat | jeruk | oranye | langit |
hitam | kacamata | pria | bentuk | wajah | wanita | laju | pesawat | tugas | waktu

```

| sayur
 Verb -> memasak | tersebut | goreng | kuliah | diberikan | menyambut | libur |
 mengambil | bersepeda | membuat | mengecat | dirancang | membelikan |
 menganggap | tugas | bermain | menggonggong | berjalan | menari | dipukuli |
 menggambar | melangkah | menjelaskan | terbenam | menawan | menarik |
 baca | belajar | tidur | berlari | diminum | berburu | ditempati | dibaca | dipukul |
 digunakan | berolahraga | terbang | menyelesaikan | menunggu
 Adj -> cepat | dekat | jauh | bahagia | bangga | gembira | ragu-ragu | kecil |
 cantik | indah | merdu | klasik | lembut | lezat | segar | harum | merah | bagus |
 sederhana | indah | tua | mahal | putih | jinak | lebam | antik | populer | baru |
 megah | luas | kecil | muda | susah | keras | lambat | gemulai | baik | sulit |
 sabar | ramah | seni | tinggi | nyaman | pasti | selamat | kuat | becek | sejuk |
 tebal | menarik | panas | dramatis | megah | besar | rendah | dingin | lebam |
 sedih | tua | kikir | lembut | lincah | gemuk | gesit | panas | ganas | rindu | pintar |
 lapang | rapat | sempit | lebam | longgar | pekat | bulat | lonjong | singkat |
 lama
 Prep -> dari | setelah | dengan | dalam | di | untuk
 Adv -> sangat | selalu | sekali | cukup | akan | sedang | ingin | masih | terlalu |
 dengan | akan | bersama | sudah
 Num -> setiap
 Pronoun -> saya | itu | kami | dia | mereka | ini | aku
 PropNoun -> atlantis | andi | buwit | tabanan | afgan | randi | roy | dodi | amba |
 doni

3.3.2 Penjelasan Source Code

Program ini merupakan implementasi algoritma CYK (Cocke-Younger-Kasami) untuk melakukan syntactic parsing berdasarkan Context-Free Grammar (CFG). Algoritma CYK digunakan untuk membentuk chart, suatu struktur data tabel, yang merepresentasikan kemungkinan struktur sintaksis dari suatu kalimat. Program ini menggunakan Streamlit sebagai antarmuka pengguna web untuk memudahkan penggunaan dan visualisasi.

1. remove_unit_production(keyList):

Deskripsi: Fungsi ini menghapus produksi unit dari tata bahasa.

Parameter:

keyList: Sebuah daftar kunci yang mewakili non-terminal dengan produksi unit.

Perilaku: Memodifikasi variabel global RESULT dengan menghilangkan produksi unit.

2. `get_set_of_production()`:
Deskripsi: Membaca file tata bahasa bebas konteks dan mengembalikan kamus yang mewakili himpunan produksi.
Perilaku: Membaca aturan tata bahasa dari sebuah file (`./rules_cfg.txt`) dan memprosesnya untuk membuat kamus (RESULT) non-terminal yang dipetakan ke aturan produksinya.
3. `get_raw_set_of_production()`:
Deskripsi: Serupa dengan `get_set_of_production`, tetapi tanpa menghapus beberapa produksi unit.
Perilaku: Membaca aturan tata bahasa dari sebuah file (`./rules_cfg.txt`) dan memprosesnya untuk membuat kamus (RESULT) non-terminal yang dipetakan ke aturan produksinya tanpa menghapus beberapa produksi unit.
4. `is_accepted(inputString)`:
Deskripsi: Memeriksa apakah sebuah string input diterima oleh tata bahasa bebas konteks.
Parameter:
`inputString`: String input yang akan diperiksa untuk diterima atau tidak.
Perilaku: Membangun tabel parsing segitiga dan memeriksa apakah string input diterima dengan memeriksa entri tertentu di dalam tabel.
5. `is_parent(posX, posY, limit, check, prodRules)`:
Deskripsi: Mengecek apakah sebuah simbol adalah induk dari simbol lainnya dalam tabel parsing segitiga.
Parameter:
`posX`: Posisi X dalam tabel parsing segitiga.
`posY`: Posisi Y dalam tabel parsing segitiga.
`limit`: Batas atas untuk posisi Y.
`check`: Simbol yang akan diperiksa apakah merupakan induk.
`prodRules`: Aturan produksi dari tata bahasa.
Perilaku: Memeriksa apakah simbol `check` adalah induk dari simbol lain dalam tabel parsing segitiga.

6. `search_left(listVar, checkPos, curPost, posX, posY, limit, prodRules):`
Deskripsi: Membangun pohon parse dari kiri ke kanan berdasarkan tabel parsing segitiga.
Parameter:
listVar: Daftar simbol dalam tabel parsing segitiga.
checkPos: Posisi saat ini yang sedang diperiksa.
curPost: Posisi saat ini dalam daftar simbol.
posX: Posisi X saat ini dalam tabel parsing segitiga.
posY: Posisi Y saat ini dalam tabel parsing segitiga.
limit: Batas atas untuk posisi Y.
prodRules: Aturan produksi dari tata bahasa.
Perilaku: Membangun pohon parse dari kiri ke kanan berdasarkan tabel parsing segitiga.
7. `get_parse_tree(inputString):`
Deskripsi: Memeriksa apakah string input diterima dan mengembalikan pohon parse jika iya.
Parameter:
inputString: String input yang akan diperiksa.
Perilaku: Memeriksa apakah string input diterima oleh tata bahasa bebas konteks dan mengembalikan pohon parse jika iya.
8. `get_table_element(inputString):`
Deskripsi: Menghasilkan elemen tabel parsing segitiga untuk string input.
Parameter:
inputString: String input yang akan dianalisis.
Perilaku: Menghasilkan elemen tabel parsing segitiga berdasarkan aturan produksi dan string input.
9. `main():`

Deskripsi: Fungsi utama untuk menangani antarmuka Streamlit dan mengeksekusi analisis string input.

Perilaku: Menggunakan antarmuka Streamlit untuk meminta input dari pengguna, menjalankan analisis string, dan menampilkan hasilnya.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Eksperimen

4.1.1 Skenario Eksperimen dan Hasil

Pengujian pada aplikasi dilakukan dengan melakukan uji coba berbagai macam kalimat Berbahasa Indonesia dengan pola kalimat sederhana. Dari semua data uji yang dimasukkan, sistem melakukan parsing kalimat tersebut. Keluarannya adalah berupa status valid atau tidak valid suatu kalimat dan pohon parsingnya. Analisisnya dilakukan dengan membandingkan kesesuaian antara data yang berasal dari sumber file Indonesian Sentences yang disediakan ibu dosen dan tugas-tugas sebelumnya dengan hasil pada sistem deteksi yang menggunakan algoritma CYK Parser. Kalimat yang diujikan adalah kalimat sederhana dengan pola kalimat S P, S P O, S P Pel, S P Ket, S P O Pel, S P O Ket, S P O Pel Ket, Kalimat juga tidak boleh ambigu dan daftar kata yang digunakan berdasarkan file Indonesian Sentences yang disediakan oleh ibu dosen dan tugas-tugas sebelumnya.

Kalimat Baku

Menurut Kosasih dan Hermawan (2012:83), kalimat baku adalah kalimat yang cara pengucapan ataupun penulisannya sesuai dengan kaidah-kaidah yang disesuaikan dengan ejaan yang disempurnakan. Kaidah yang dimaksud yaitu pedoman ejaan berupa tata bahasa baku dan kamus (Keputusan Presiden No.57 tahun 1972). Ejaan yang disempurnakan (EYD) dalam Pedoman Umum Ejaan Bahasa Indonesia yang disempurnakan termasuk edisi terbaru berdasarkan Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 50 tahun 2015 tanggal 26 November 2016 dan berdasarkan Tata Bahasa Baku Bahasa Indonesia Edisi Keempat yang dikeluarkan oleh Badan Pengembangan Dan Pembinaan

Bahasa Kementerian Pendidikan Dan Kebudayaan Republik Indonesia pada tahun 2017.

Tabel 4.1. Hasil pengujian kalimat pada sistem

NO	KALIMAT	JENIS KALIMAT		KESESUAIAN
		FILE PENGUJIAN	SISTEM	
1	Orang tua itu sangat kikir	Baku (VALID)	VALID	SESUAI
2	Bahan baju tidur ini sangat lembut sekali	Baku (VALID)	VALID	SESUAI
3	Kucing kecil itu lincah sekali	Baku (VALID)	VALID	SESUAI
4	Bocah gemuk itu cukup gesit berlari di lapangan sekolah	Baku (VALID)	VALID	SESUAI
5	Kopi itu terlalu panas untuk diminum	Baku (VALID)	VALID	SESUAI
6	Harimau itu sangat ganas saat berburu	Baku (VALID)	VALID	SESUAI
7	Aku sangat rindu kampung halaman	Baku (VALID)	VALID	SESUAI
8	Anak itu sangat pintar dalam matematika	Baku (VALID)	VALID	SESUAI
9	Rumah mereka terlalu besar untuk ditempati	Baku (VALID)	VALID	SESUAI
10	Ruang tamu ini sangat lapang untuk rapat keluarga	Baku (VALID)	VALID	SESUAI
11	Buku ini terlalu tebal untuk dibaca	Baku (VALID)	VALID	SESUAI
12	Gedung-gedung baru itu sangat tinggi	Baku (VALID)	VALID	SESUAI
13	Tingkat literasi para siswa itu masih sangat rendah	Baku (VALID)	VALID	SESUAI
14	Kaos kaki adik saya sudah sangat longgar untuk digunakan	Baku (VALID)	VALID	SESUAI

15	Suhu udara ini sangat dingin untuk berolahraga	Baku (VALID)	VALID	SESUAI
16	Ruang belajar ini sangat sempit saat digunakan bersama	Baku (VALID)	Tidak VALID	TIDAK SESUAI
17	Harga makanan itu terlalu mahal untuk anak sekolah dasar	Baku (VALID)	valid	SESUAI
18	Pipi Doni itu biru lebam setelah dipukul Dodi	Baku (VALID)	VALID	SESUAI
19	Warna buah jeruk itu oranye tua	Baku (VALID)	VALID	SESUAI
20	Langit malam hari ini hitam pekat	Baku (VALID)	VALID	SESUAI
21	Kacamata pria itu bulat	Baku (VALID)	VALID	SESUAI
22	Bentuk wajah wanita ini terlalu lonjong	Baku (VALID)	VALID	SESUAI
23	Laju pesawat terbang itu sangat cepat	Baku (VALID)	VALID	SESUAI
24	Dia menyelesaikan tugas dalam waktu singkat	Baku (VALID)	VALID	SESUAI
25	Teman saya itu sudah sangat lama menunggu	Baku (VALID)	VALID	SESUAI
26	Ibu adik saya itu sangat cepat memasak nasi goreng	Baku (VALID)	VALID	SESUAI
27	Hubungan pertemanan kami sangat dekat	Baku (VALID)	VALID	SESUAI
28	Kota Atlantis itu sangat jauh dari kota kami	Baku (VALID)	VALID	SESUAI
29	Keluarga saya sangat bahagia setelah adik kuliah	Baku (VALID)	VALID	SESUAI
30	Ibu guru kami selalu bangga dengan prestasi muridnya	Baku (VALID)	VALID	SESUAI
31	Teman Andi itu sangat bahagia saat diberikan kue	Baku (VALID)	VALID	SESUAI
32	Para siswa itu sangat gembira menyambut hari libur	Baku (VALID)	VALID	SESUAI

33	Dia sangat ragu-ragu dalam mengambil keputusan	Baku (VALID)	VALID	SESUAI
34	Gadis kecil itu sangat cantik	Baku (VALID)	VALID	SESUAI
35	Pemandangan Desa Buwit Tabanan itu sangat indah	Baku (VALID)	VALID	SESUAI
36	Suara penyanyi Afgan itu sangat merdu sekali	Baku (VALID)	VALID	SESUAI
37	Suara alunan musik klasik itu sangat lembut	Baku (VALID)	VALID	SESUAI
38	Rasa sup sayur itu sangat lezat	Baku (VALID)	VALID	SESUAI
39	Es teh lemon itu sangat segar	Baku (VALID)	VALID	SESUAI
40	Aroma parfum ibu guru itu sangat harum	Baku (VALID)	VALID	SESUAI
41	Baju merah itu sangat bagus	Baku (VALID)	VALID	SESUAI
42	Rumah sederhana itu cukup indah	Baku (VALID)	VALID	SESUAI
43	Gitar tua itu sangat mahal	Baku (VALID)	VALID	SESUAI
44	Kucing putih itu sangat jinak	Baku (VALID)	VALID	SESUAI
45	Mobil antik bapak itu sangat populer di kota kami	Baku (VALID)	VALID	SESUAI
46	Gedung baru itu sangat megah	Baku (VALID)	VALID	SESUAI
47	Lapangan bola itu cukup luas untuk bersepeda	Baku (VALID)	VALID	SESUAI
48	Harga gedung baru itu cukup mahal	Baku (VALID)	VALID	SESUAI
49	Sepatu anak itu sangat kecil	Baku (VALID)	VALID	SESUAI
50	Kabar itu membuat mereka sangat gembira	Baku (VALID)	VALID	SESUAI
51	Ayah mengecat pintu dapur dengan warna biru muda	Baku (VALID)	VALID	SESUAI
52	Ibu selalu membelikan adik	Baku (VALID)	VALID	SESUAI

	saya ikan segar			
53	Kabar itu membuat mereka sangat bahagia	Baku (VALID)	VALID	SESUAI
54	Roy membelikan saya mawar merah	Baku (VALID)	VALID	SESUAI
55	Dia menganggap tugas itu susah	Baku (VALID)	VALID	SESUAI
56	Anak itu bermain dengan gembira	Baku (VALID)	VALID	SESUAI
57	Anjing itu menggonggong dengan sangat keras	Baku (VALID)	VALID	SESUAI
58	Mobil itu berjalan sangat lambat	Baku (VALID)	VALID	SESUAI
59	Gadis kecil itu menari dengan sangat gemulai	Baku (VALID)	VALID	SESUAI
60	Anak itu menggambar dengan cepat	Baku (VALID)	VALID	SESUAI
61	Amba melangkahkan kaki dengan sangat cepat	Baku (VALID)	VALID	SESUAI
62	Bapak guru itu menjelaskan materi dengan sangat baik	Baku (VALID)	VALID	SESUAI
63	Randi menjelaskan konsep sulit itu dengan sabar	Baku (VALID)	VALID	SESUAI
64	Pegawai toko menyambut setiap pelanggan dengan ramah	Baku (VALID)	VALID	SESUAI
65	Pemandangan matahari terbenam itu akan sangat menawan	Baku (VALID)	VALID	SESUAI
66	Pertunjukan seni malam ini	Baku (VALID)	VALID	SESUAI

	akan sangat indah			
67	Semangat tim sepakbola itu sudah sangat tinggi	Baku (VALID)	VALID	SESUAI
68	Suasana kerja kantor itu akan sangat nyaman	Baku (VALID)	VALID	SESUAI
69	Anak kecil itu sedang sedih	Baku (VALID)	VALID	SESUAI
70	Setiap orang itu pasti ingin selamat di dunia ini	Baku (VALID)	VALID	SESUAI
71	Anak itu sangat kuat sekali	Baku (VALID)	VALID	SESUAI
72	Jalan kampung itu becek sekali	Baku (VALID)	VALID	SESUAI
73	Udara pagi hari ini cukup sejuk	Baku (VALID)	VALID	SESUAI
74	Buku tebal itu sangat menarik untuk dibaca	Baku (VALID)	VALID	SESUAI
75	Bunga kenanga itu sangat harum	Baku (VALID)	VALID	SESUAI
76	Mobil cepat itu sangat mahal	Baku (VALID)	VALID	SESUAI
77	Bunga itu masih segar di taman	Baku (VALID)	VALID	SESUAI
78	Cuaca hari ini terlalu panas	Baku (VALID)	VALID	SESUAI
79	Pertunjukkan tersebut sangat dramatis	Baku (VALID)	VALID	SESUAI
80	Gedung itu dirancang dengan sangat megah	Baku (VALID)	VALID	SESUAI
81	Terlalu besar untuk ditempati	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
82	Ruang tamu ini	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
83	Terlalu tebal untuk dibaca	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
84	Gedung-gedung baru itu	Tidak Baku (Tidak VALID)	VALID	TIDAK SESUAI
85	Masih sangat rendah	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
86	Kaos kaki adik saya	Tidak Baku	TIDAK	SESUAI

		(Tidak VALID)	VALID	
87	Dingin untuk berolahraga	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
88	Saat digunakan bersama	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
89	Harga makanan itu	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
90	Setelah dipukul Dodi biru lebab	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
91	Menari dengan sangat gemulai	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
92	Menggambar dengan cepat	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
93	Melangkahkan kaki dengan sangat cepat	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
94	Materi dengan sangat baik	Tidak Baku (Tidak VALID)	VALID	TIDAK SESUAI
95	Konsep sulit itu dengan sabar	Tidak Baku (Tidak VALID)	VALID	TIDAK SESUAI
96	Pegawai toko sedang	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
97	Pemandangan matahari terbenam itu	Tidak Baku (Tidak VALID)	VALID	TIDAK SESUAI
98	Pertunjukan seni malam ini	Tidak Baku (Tidak VALID)	VALID	TIDAK SESUAI
99	Semangat tim sepakbola itu	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
100	Suasana kerja di kantor itu akan sangat nyaman	Tidak Baku (Tidak VALID)	VALID	TIDAK SESUAI
101	Sedang sedih	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
102	Setiap orang itu	Tidak Baku (Tidak VALID)	TIDAK VALID	SESUAI
Nilai Akurasi				93,14%

Pada eksperimen yang telah dilakukan, kami mencoba untuk menguji sebanyak 102 kalimat, yang dimana terdiri dari 80 kalimat baku dan 22 kalimat tidak baku. Dimana pada 80 kalimat baku yang telah diuji, 79 kalimat valid sehingga hasilnya sesuai dan 1 kalimat yang tidak valid sehingga hal tersebut tidak sesuai. Kemudian pada pengujian 22 kalimat yang tidak baku terdiri dari 6 kalimat yang pada saat dimasukan pada program yaitu valid sehingga hasilnya tidak sesuai. Namun untuk 16 kalimat yang tidak baku pada saat diuji pada program hasilnya tidak valid sehingga hal tersebut sesuai. Dari pengujian yang telah dilakukan hasil akurasi yaitu 93,14%.

Aplikasi Parsing Kalimat Baku Berbahasa Indonesia

Aplikasi ini merupakan aplikasi yang digunakan untuk melakukan parsing kalimat berbahasa indonesia dengan pola kalimat sederhana sesuai dengan pedoman Umum Ejaan Bahasa Indonesia (PUEBI) dan Tata Bahasa Baku Bahasa Indonesia Edisi Keempat.

Masukan kalimat untuk diparsing:

Ruang belajar ini sangat sempit saat digunakan bersama

Parse

Berdasarkan hasil pemeriksaan, **Ruang belajar ini sangat sempit saat digunakan bersama**

TIDAK VALID

Tidak sesuai dengan pola dasar bahasa Indonesia

Gambar 4.1. Hasil Pengujian kalimat tidak baku

Pada gambar 4.1 merupakan tampilan dari hasil pengujian kalimat tidak baku. Ketika kita menginput kalimat yang tidak baku pada program contohnya “Ruang belajar ini sangat sempit saat digunakan bersama” maka sistem akan memberikan pemberitahuan bahwa kalimat tersebut tidak valid dan akan ditampilkan dengan nuansa sistem yang berwarna merah yang menandakan ketidaksesuaian kalimat dengan pola dasar bahasa Indonesia.

Aplikasi Parsing Kalimat Baku Berbahasa Indonesia

Aplikasi ini merupakan aplikasi yang digunakan untuk melakukan parsing kalimat berbahasa Indonesia dengan pola kalimat sederhana sesuai dengan pedoman Umum Ejaan Bahasa Indonesia (PUEBI) dan Tata Bahasa Baku Bahasa Indonesia Edisi Keempat.

Masukan kalimat untuk diparsing:

Bocah gemuk itu cukup gesit berlari di lapangan sekolah

Parse

Berdasarkan hasil pemeriksaan, **Bocah gemuk itu cukup gesit berlari di lapangan sekolah**

VALID

Sesuai dengan pola dasar bahasa Indonesia

Triangular Table:

	0	1	2	3	4	5	6	7	8
0	{K, S, O, Pel, NP, K1, K2, K3, K4}	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
1	{K, S, O, Pel, NP, K1, K2, K3, K4}	Ø	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
2	Ø	Ø	{K, S, O, Pel, NP, K1, K2, K3}	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
3	{K, S, O, Pel, NP, K1, K2, K3, K4}	Ø	{K, S, O, Pel, NP, K1, K3}	{P, Pel, VP}	<NA>	<NA>	<NA>	<NA>	<NA>
4	{K, S, O, Pel, NP, K1, K2, K3, K4}	Ø	Ø	{P, Pel, VP}	{P, Pel, VP}	<NA>	<NA>	<NA>	<NA>
5	Ø	Ø	{K, S, O, Pel, NP, K1, K3}	Ø	{P, Pel, VP}	{P, Pel, VP}	<NA>	<NA>	<NA>
6	{K, S, O, Pel, NP, K2, K3}	Ø	{K, S, O, Pel, NP, K1}	{P, Pel, VP}	Ø	{P, Pel, VP}	{S, P, O, Pel, Ket, NP, PP}	<NA>	<NA>
7	{K, S, O, Pel, NP, K1}	Ø	Ø	{P, Pel, AdjP}	{P, Pel, VP}	Ø	{P, Pel, Ket, PP}	{S, O, Pel, NP}	<NA>
8	{S, O, Pel, NP, Noun}	{P, Pel, AdjP, Adj}	{S, O, Pel, NP, Pronoun}	{Adv}	{P, Pel, AdjP, Adj}	{P, Pel, VP, Verb}	{Prep}	{S, O, Pel, NP, Noun}	{S, O, Pel, NP, Noun}

Gambar 4.2. Hasil Pengujian kalimat baku

Pada gambar 4.2 merupakan tampilan dari hasil pengujian kalimat baku. Seperti contohnya, kita menginput kalimat “Bocah gemuk itu cukup gesit berlari di lapangan sekolah” maka sistem akan memberikan pemberitahuan bahwa kalimat tersebut valid karena telah sesuai dengan pola dasar bahasa Indonesia dan sistem akan menampilkan warna hijau pada hasil kalimat.

4.2 Pembahasan dan Analisis Hasil

Dengan menerapkan Algoritma CYK, parsing kalimat dalam bahasa Indonesia dapat dilakukan secara efisien dengan tingkat akurasi di atas 93,14%. Meskipun demikian, terdapat beberapa kegagalan dalam pengujian, yang dapat diatributkan pada tiga faktor utama:

1. Ketidaksesuaian struktur kalimat dengan kaidah bahasa Indonesia,
2. Ketidaktersediaan unsur pembentuk kalimat dalam CFG.

Semua pola kalimat dalam Context-Free Grammar (CFG) dapat dideteksi dengan baik oleh Algoritma CYK. Oleh karena itu, Algoritma CYK dapat diandalkan untuk melakukan parsing pada CFG. Dalam mengembangkan penelitian ini, beberapa aspek perlu dipertimbangkan. Pertama, perlu memperluas grammar atau tata bahasa dalam CFG agar dapat mendeteksi pola kalimat yang lebih kompleks. Kegagalan dalam pengujian terkadang disebabkan oleh ketidaktersediaan unsur pembentuk kalimat di CFG, meskipun kalimat tersebut sebenarnya benar.

BAB V

KESIMPULAN

5.1. Kesimpulan

Proyek ini bertujuan untuk mengembangkan aplikasi parsing untuk Bahasa Indonesia. Fokus utamanya adalah menerapkan teknik syntactic parsing menggunakan algoritma Context-Free Grammar (CFG), Chomsky Normal Form (CNF), dan Cocke-Younger-Kasami (CYK). Tujuannya adalah untuk memberikan pemahaman yang lebih dalam tentang struktur gramatikal Bahasa Indonesia dan meningkatkan efektivitas serta akurasi dalam pemrosesan bahasa alami (Natural Language Processing atau NLP). Parsing sintaksis memainkan peran penting dalam memahami struktur gramatikal Bahasa Indonesia, yang terkenal dengan fleksibilitasnya dibandingkan dengan bahasa lain seperti Bahasa Inggris. Penggunaan syntactic parsing membantu dalam mengidentifikasi komponen-komponen penting dalam kalimat seperti subjek, objek, dan predikat selain itu algoritma Cocke-Younger-Kasami (CYK) dipilih karena efektivitasnya dalam bekerja dengan CFG dalam bentuk CNF. Algoritma ini, yang berbasis tabel dan menggunakan pendekatan pemrograman dinamis, sangat berguna dalam menganalisis kalimat-kalimat kompleks, membuatnya ideal untuk aplikasi dalam NLP.

Aplikasi ini beroperasi dengan beberapa asumsi dan batasan, seperti hanya menerima kalimat sederhana yang mengikuti struktur gramatikal tertentu dalam Bahasa Indonesia. Selain itu, aplikasi ini menggunakan kosakata yang telah ditentukan dan memiliki batasan dalam jumlah kalimat serta kompleksitas kalimat yang dapat diproses. Aplikasi juga tidak dilengkapi dengan fitur pembaruan data secara otomatis. Aplikasi ini memiliki potensi signifikan dalam bidang NLP, terutama dalam memahami dan memproses Bahasa Indonesia. Meskipun memiliki beberapa batasan, aplikasi ini bisa menjadi alat penting untuk penelitian lebih lanjut dan pengembangan dalam pemrosesan bahasa alami Bahasa Indonesia.

5.2. Saran

Proyek pengembangan aplikasi parsing Bahasa Indonesia dengan teknik syntactic parsing menggunakan CFG, CNF, dan CYK memiliki tujuan yang relevan dalam pengembangan NLP untuk Bahasa Indonesia. Meskipun ada batasan dalam hal jenis kalimat yang dapat diproses dan pembaruan data, potensi aplikasi ini dalam meningkatkan pemahaman struktur gramatikal Bahasa Indonesia sangat menjanjikan. Dengan perbaikan dan pengembangan lebih lanjut, aplikasi ini dapat menjadi alat yang berharga dalam pengolahan bahasa alami Bahasa Indonesia.

DAFTAR PUSTAKA

- Alvina Aulia, Aulia, A., & Subject Content Coordinator Software Development | School of Computer Science. (2022). *Penyederhanaan Context Free Grammar*. School of Computer Science. [Online] Available at: <https://socs.binus.ac.id/2018/12/20/penyederhanaan-context-free-grammar/> [28 December 2023].
- Cahyani, D. E., Gumilar, L., & Pangestu, A. (2020). Indonesian parsing using probabilistic context-free grammar (PCFG) and Viterbi-Cocke younger Kasami (Viterbi-CYK). *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. [Online] Available at: <https://www.semanticscholar.org/paper/Indonesian-Parsing-using-Probabilistic-Context-Free-Cahyani-Gumilar/09f9f7ee25b79d2a85c3bc3fcb8048de1d40493c> [28 December 2023].
- Kadar, A. (2021). *Parsing I tata bahasa bebas konteks dan algoritma CYK*. Borealis. [Online] Available at: <https://www.borealisai-com.translate.goog/research-blogs/tutorial-15-parsing-i-context-free-grammars-and-cyk-algorithm/> [28 December 2023].
- Prabowo, B., Rustamadji, H. C., & Fauziah, Y. (2020). Algoritma Cocke younger kasami untuk deteksi struktur kalimat Dan merekomendasikanya menggunakan algoritma damerau levenshtein distance. *Telematika*, 17(2), 101. [Online] Available at: <https://www.researchgate.net/publication/https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2017-2018/Makalah-2017/Makalah-Matdis-2017-108.pdf> [28 December 2023].
- Pratama, M. R., Kusumadewi, S., & Hidayat, T. (2017). Penerapan algoritma lalr parser Dan context-free grammar untuk struktur kalimat Bahasa Indonesia. *Jurnal Teknologi Elektro*, 8(1), 1. [Online] Available at: <https://publikasi.mercubuana.ac.id/index.php/jte/article/view/1364>. [28 December 2023].

- Lampiran
 - Indonesian Sentences for System Testing :
https://drive.google.com/file/d/1otF5kwGAQjr_UkWF6bfX5EoYc3YUwKsd/view?usp=sharing

LAMPIRAN

- File Source Code:
<https://drive.google.com/file/d/1hQR5ux6bNeQLqzkW6UdokLv4AxAHVKL8/view?usp=sharing>
- Screenshot Run Program/Aplikasi

Aplikasi Parsing Kalimat Baku Berbahasa Indonesia

Aplikasi ini merupakan aplikasi yang digunakan untuk melakukan parsing kalimat berbahasa indonesia dengan pola kalimat sederhana sesuai dengan pedoman Umum Ejaan Bahasa Indonesia (PUEBI) dan Tata Bahasa Baku Bahasa Indonesia Edisi Keempat.

Masukan kalimat untuk diparsing:

Ruang belajar ini sangat sempit saat digunakan bersama

Parse

Berdasarkan hasil pemeriksaan, **Ruang belajar ini sangat sempit saat digunakan bersama**

TIDAK VALID

Tidak sesuai dengan pola dasar bahasa Indonesia

Aplikasi Parsing Kalimat Baku Berbahasa Indonesia

Aplikasi ini merupakan aplikasi yang digunakan untuk melakukan parsing kalimat berbahasa Indonesia dengan pola kalimat sederhana sesuai dengan pedoman Umum Ejaan Bahasa Indonesia (PUEBI) dan Tata Bahasa Baku Bahasa Indonesia Edisi Keempat.

Masukan kalimat untuk diparsing:

Bocah gemuk itu cukup gesit berlari di lapangan sekolah

Parse

Berdasarkan hasil pemeriksaan, **Bocah gemuk itu cukup gesit berlari di lapangan sekolah**

VALID

Sesuai dengan pola dasar bahasa Indonesia

Triangular Table:

	0	1	2	3	4	5	6	7	8
0	{K, S, O, Pel, NP, K1, K2, K3, K4}	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
1	{K, S, O, Pel, NP, K1, K2, K3, K4}	Ø	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
2	Ø	Ø	{K, S, O, Pel, NP, K1, K2, K3}	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
3	{K, S, O, Pel, NP, K1, K2, K3, K4}	Ø	{K, S, O, Pel, NP, K1, K3}	{P, Pel, VP}	<NA>	<NA>	<NA>	<NA>	<NA>
4	{K, S, O, Pel, NP, K1, K2, K3, K4}	Ø	Ø	{P, Pel, VP}	{P, Pel, VP}	<NA>	<NA>	<NA>	<NA>
5	Ø	Ø	{K, S, O, Pel, NP, K1, K3}	Ø	{P, Pel, VP}	{P, Pel, VP}	<NA>	<NA>	<NA>
6	{K, S, O, Pel, NP, K2, K3}	Ø	{K, S, O, Pel, NP, K1}	{P, Pel, VP}	Ø	{P, Pel, VP}	{S, P, O, Pel, Ket, NP, PP}	<NA>	<NA>
7	{K, S, O, Pel, NP, K1}	Ø	Ø	{P, Pel, AdjP}	{P, Pel, VP}	Ø	{P, Pel, Ket, PP}	{S, O, Pel, NP}	<NA>
8	{S, O, Pel, NP, Noun}	{P, Pel, AdjP, Adj}	{S, O, Pel, NP, Pronoun}	{Adv}	{P, Pel, AdjP, Adj}	{P, Pel, VP, Verb}	{Prep}	{S, O, Pel, NP, Noun}	{S, O, Pel, NP, Noun}