

TUGAS KELOMPOK

FINAL PROJECT

MATA KULIAH TEORI BAHASA DAN OTOMATA

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



Oleh:

KELOMPOK 3 / KELAS E

I Ngurah Komang Agus Suryadiyatmika. S	(2208561031)
I Kadek Revan Aditya Prawira	(2208561050)
Devon Vivian Gunawan	(2208561081)
I Putu Aditya Pradana	(2208561124)

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

2023

KATA PENGANTAR

Puji syukur kami panjatkan ke Tuhan Yang Maha Esa karena dengan rahmat dan karunia-Nya, kami panjatkan puja dan puji syukur atas kehadiran-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga kami dapat menyelesaikan *final project* mata kuliah Teori Bahasa dan Otomata ini dengan baik.

Tugas *final project* ini telah kami susun dengan maksimal walau pada mengalami berbagai kendala yang harus kami hadapi, sehingga tidak jarang kami mendapatkan bantuan dari berbagai pihak dengan harapan dapat memperlancar proses pengerjaan *final project* ini. Perancangan penugasan ini terkait dengan penerapan algoritma CYK dalam melakukan analisis sintaksis kalimat berbahasa Indonesia apakah sesuai dengan tata bahasa Indonesia atau tidak. Harapannya, sistem ini dapat membantu untuk mempelajari pola kata bahasa Indonesia.

Terlepas dari semua itu, kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar kami dapat memperbaiki makalah ilmiah ini. Akhir kata, kami berharap semoga tugas kelompok ini dapat memberikan manfaat maupun inspirasi terhadap pembaca.

Jimbaran, 26 Desember 2023

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	2
----------------------------	----------

DAFTAR ISI.....	3
------------------------	----------

BAB I

PENDAHULUAN.....	5
-------------------------	----------

1.1 Latar Belakang.....	5
-------------------------	---

1.2 Tujuan Penulisan.....	6
---------------------------	---

1.3 Asumsi.....	6
-----------------	---

1.4 Batasan.....	6
------------------	---

BAB II

METODE CONTEXT-FREE GRAMMAR.....	8
---	----------

2.1 Context Free Grammar (CFG).....	8
-------------------------------------	---

2.2 Konversi CFG Rules Menjadi CNF.....	10
---	----

2.3 Algoritma CYK.....	19
------------------------	----

2.3.1 Contoh Penerapan Algoritma CYK secara manual.....	20
---	----

BAB III

ANALISIS DAN DESAIN.....	25
---------------------------------	-----------

3.1 Desain Eksperimen.....	25
----------------------------	----

3.2.1 Skenario Eksperimen.....	25
--------------------------------	----

3.2.2 Ukuran Evaluasi.....	25
----------------------------	----

3.2 Desain Aplikasi.....	26
--------------------------	----

3.2.1 Flowchart.....	26
----------------------	----

3.2.2 Mockup.....	27
-------------------	----

3.2.3 Use Case Diagram.....	28
-----------------------------	----

3.2.4 Activity Diagram.....	28
-----------------------------	----

3.2.5 Source Code.....	29
------------------------	----

BAB IV

HASIL DAN PEMBAHASAN.....	33
----------------------------------	-----------

4.1 Hasil Eksperimen.....	33
---------------------------	----

4.2 Pembahasan dan Analisis Hasil.....	38
--	----

BAB V

PENUTUP.....	41
5.1 Kesimpulan.....	41
5.2 Daftar Pustaka.....	41

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kehidupan sehari-hari, tidak terlepas akan kegiatan berkomunikasi. Komunikasi merupakan suatu hal yang esensial dalam interaksi antarindividu maupun kelompok, memungkinkan pertukaran ide, informasi, dan ekspresi. Bahasa menjadi sarana utama dalam proses komunikasi, dan dalam era modern ini, pemrosesan bahasa alami (Natural Language Processing/NLP) menjadi bidang yang semakin berkembang untuk memahami dan memanfaatkan bahasa manusia secara efisien.

Salah satu aspek penting dalam pemrosesan bahasa adalah analisis sintaksis, yang bertujuan untuk memahami struktur gramatikal suatu kalimat. Pada konteks bahasa Indonesia, analisis sintaksis menjadi tantangan tersendiri mengingat kompleksitas struktur dan variasi makna yang terkandung dalam setiap kalimat. Oleh karena itu, diperlukan metode atau algoritma yang mampu menguraikan struktur sintaksis dengan akurasi tinggi.

Salah satu algoritma yang telah diterapkan dalam analisis sintaksis adalah Cocke-Younger-Kasami (CYK) *algorithm*. Algoritma ini menjadi subjek penelitian yang menarik karena kemampuannya dalam melakukan parsing secara efisien, terutama dalam konteks bahasa natural seperti bahasa Indonesia. Misalkan, ketika seorang pengguna ingin memberikan input kalimat bahasa Indonesia yang sesuai dengan pola kalimat tertentu, seperti kalimat tanya atau kalimat perintah, CYK dapat digunakan untuk memvalidasi struktur sintaktis kalimat tersebut.

Sebagai contoh, pertimbangkan pola kalimat tanya sederhana, seperti "Siapa yang melakukan tindakan tertentu?" Pengguna dapat memberikan input kalimat seperti "Ani makan nasi" dan algoritma CYK dapat digunakan untuk memeriksa apakah struktur sintaktis kalimat tersebut sesuai dengan pola kalimat tanya yang telah ditetapkan. CYK akan melakukan parsing kalimat, menghasilkan pohon sintaksis, dan memeriksa apakah struktur pohon tersebut memenuhi kriteria kalimat tanya yang diinginkan.

Selain itu, CYK juga dapat diterapkan dalam kasus di mana pengguna hendak memberikan input dengan variasi struktur kalimat yang kompleks. Misalnya, dalam kalimat pasif seperti "Buku itu ditulis oleh Maya," algoritma CYK dapat membantu dalam memvalidasi struktur sintaktis yang sesuai dengan pola kalimat pasif. Keunggulan utama dari penerapan CYK dalam konteks ini adalah kemampuannya untuk mengatasi struktur kalimat yang kompleks dengan efisien. Algoritma ini bekerja dengan prinsip pemrograman dinamis

dan tabel parsing, memungkinkan pengguna untuk mengonfirmasi keabsahan struktur sintaksis kalimat bahasa Indonesia. Menelusuri lebih jauh lagi, pembahasan penugasan kali ini akan ditujukan untuk menjelaskan dan mengaplikasikan bagaimana algoritma CYK dalam proses analisis sintaksis kalimat bahasa Indonesia.

1.2 Tujuan Penulisan

Berdasarkan perancangan kasus yang telah dipaparkan pada bagian latar belakang, adapun tujuan dari pembuatan laporan ini yaitu

- a. Untuk mengetahui implementasi *Context free grammar* dalam melakukan *parsing* bahasa baku bahasa Indonesia,
- b. Untuk mengetahui bagaimana implementasi dari algoritma Cocke-Younger-Kasami (CYK) dengan *rule* yang telah dirubah dalam bentuk *chomsky normal form* melalui *table filling* dalam bahasa pemrograman Python untuk melakukan *parsing* bahasa baku bahasa Indonesia, serta
- c. Untuk mengembangkan sebuah sistem aplikasi yang dapat digunakan oleh khalayak umum.

1.3 Asumsi

Adapun asumsi yang harus diketahui dalam perancangan penulisan dari topik laporan ini yaitu sebagai berikut :

1. Dataset terkait dengan kata-kata yang ada dalam perancangan sistem mencukupi jumlah dari kata umum yang tersedia dalam kehidupan sehari-hari,
2. *Rules* yang digunakan berasal dari aturan ketetapan bahasa baku Indonesia sehingga dapat sesuai dengan struktur kalimat umum yang telah dikenal masyarakat luas,
3. Asumsi terhadap pengguna yaitu memberikan input sesuai dengan ketentuan atau Pedoman Umum Ejaan Bahasa Indonesia (PUEBI)

1.4 Batasan

Adapun batasan yang harus diketahui dalam perancangan sistem yang dirancang oleh penulis yaitu :

1. Kalimat yang diinputkan adalah kalimat tunggal. Kalimat majemuk tidak mampu dianalisis oleh sistem sehingga akan memunculkan hasil yang tidak sesuai dengan harapan pengguna.

2. Kata-kata yang tersusun adalah kata baku bahasa Indonesia. Apabila inputnya menggunakan bahasa tidak baku ataupun bahasa daerah atau kata yang memiliki kesalahan penulisan (*typo*), maka sistem tidak mampu untuk menganalisis.
3. Ketersediaan dataset kata yang terbatas, sesuai dengan contoh yang terlampir pada laman pembelajaran. Apabila muncul kata di luar dari dataset tersebut namun sudah terdaftar dalam PUEBI, maka sistem tidak memberikan jawaban yang sesuai.
4. Kalimat yang diinputkan ke sistem untuk saat ini bersifat *case sensitive* dan hanya menerima kalimat yang bersifat *lowercase*

BAB II

METODE CONTEXT-FREE GRAMMAR

2.1 Context Free Grammar (CFG)

Context-Free Grammar (CFG) adalah suatu bentuk notasi formal yang digunakan untuk mendefinisikan tata bahasa dalam linguistik dan ilmu komputer. Aturan produksinya biasanya mengikuti format $A \rightarrow B$, di mana A adalah pemproduksi dan B adalah hasil produksi. CFG memberikan panduan dalam menyusun urutan-urutan dalam bahasa, mirip dengan tata bahasa regular.

Contoh aturan produksi dalam CFG dapat mencakup pola seperti berikut:

- $X \rightarrow bY \mid Za$
- $Y \rightarrow aY \mid b$
- $Z \rightarrow bZ \mid \epsilon$

Namun, penerapan CFG memerlukan penyederhanaan untuk mencegah kompleksitas yang tidak diinginkan atau aturan produksi yang tidak bermakna. Proses penyederhanaan melibatkan beberapa langkah kunci:

1. Eliminasi ϵ -production

- Menghilangkan aturan produksi yang menghasilkan string kosong (ϵ) untuk menghindari ambiguitas.
- Mengganti aturan produksi yang mengandung ϵ dengan aturan produksi baru tanpa string kosong.

2. Eliminasi unit production:

- Mengganti hasil produksi yang lebih dari satu variabel dengan hasil produksi grammar yang sesuai.
- Memastikan bahwa hasil produksi yang melibatkan lebih dari satu variabel untuk meningkatkan ekspresivitas grammar.

3. Eliminasi useless symbol:

- Identifikasi simbol-simbol yang tidak dapat dipahami dari simbol awal.
- Hapus simbol-simbol dan aturan produksinya yang tidak dapat dikenali.

Dalam definisinya, *Context-Free Grammar* (CFG) memiliki 4 komponen penting dalam analisis deskripsi *grammar* dari sebuah bahasa, atau yang sering dikenal dengan *four-tuple*, yaitu $G = (V, \Sigma, P, S)$ dengan :

- V yaitu kumpulan dari variabel
- Σ atau T yaitu kumpulan set dari simbol terminal

- P yaitu kumpulan dari aturan atau *rule*
- S yaitu simbol awal atau *start symbol*.

Perancangan sistem ini juga menggunakan *tuple* yang bersesuaian dengan *context-free grammar*, berikut adalah set lebih rincinya :

- V = kumpulan variabel = {K, S, P, O, Pel, Ket, NP, VP, PP, AdjP, PropNoun, Pronoun, Noun, NumP, Noun, AdvP, Prep, Adj, Adv, Num}
- Σ atau T = terminal = kata-kata yang termasuk dalam sistem = { banyak, pucat, ceria, istimewa, gemilang, dinamis, penuh, harmonis, lancar, intens, anggun, luar biasa, matang, fleksibel, cukup, riang, penting, menang, becek, seru, efisien, pahit, lelah, keras, abstrak, salah, rapuh, curam, unik, enak, mahal, berat, megah, selamat, biru lebam, pekat, baik, gemulai, lambat, susah, biru muda, bangga, prestasi, bahagia, antik, populer, kecil, cantik, jinak, merdu, klasik, parfum, sangat, agak, jarang, sudah, akan, sedang, terlalu, sederhana, harus, tidak, berat, dengan, cukup, banyak, sungguh, sekerasnya, selahapnya, se jelasnya, ... }

- P = kumpulan dari *set of rules* = {

$K \rightarrow S P O$

$K \rightarrow S P Pel$

$K \rightarrow S P Ket$

$K \rightarrow S P O Pel$

$K \rightarrow S P O Ket$

$K \rightarrow S P Pel Ket$

$K \rightarrow S P O Pel Ket$

$S \rightarrow NP$

$P \rightarrow PP \mid VP \mid AdjP$

$O \rightarrow NP$

$Pel \rightarrow VP NP \mid NP \mid VP \mid AdjP$

$Ket \rightarrow PP \mid NumP \mid AdjP$

$NP \rightarrow PropNoun \mid Pronoun \mid Noun \mid$

$NP AdjP \mid NumP NP \mid NP Noun \mid$

$NP PropNoun \mid NP Pronoun$

$VP \rightarrow Verb \mid AdvP VP$

$PP \rightarrow PP NP \mid Prep AdjP \mid Prep$

$AdjP \rightarrow Adj \mid Adv AdjP$

$AdvP \rightarrow Adv \mid Adv AdvP$

$NumP \rightarrow Num \mid NumP NP \mid NumP$

VP

}

dimana variabel yang dimaksud di *set of rules* tersebut, yaitu :

- K: Kalimat
- S: Subjek
- P: Predikat
- O: Objek
- Pel: Pelengkap
- Ket: Keterangan
- NP: Noun Phrase
- VP: Verb Phrase
- AdjP: Adjectival Phrase
- PP: Prepositional Phrase
- NumP: Numeral Phrase
- Verb: kata kerja (verba)

- Noun: kata benda (nomina)
- Adj: kata sifat (Adjectival)
- Adv: kata keterangan (Adverb: sangat, agak, jarang, sudah, akan, sedang, ...)
- Num: numeralia
- Prep: Preposisi (pada, dengan, di, ke, untuk, ...)
- PropNoun: Proper Noun (nama-nama: Agung, Kali Unda, Denpasar, New York,...)
- Pronoun: kata ganti (ini, itu, saya, dia,..)

- S = simbol awal = {K}

2.2 Konversi CFG Rules Menjadi CNF

Chomsky Normal Form (CNF) merupakan salah satu bentuk normal yang sangat berguna untuk Context Free Grammar (CFG) . Bentuk normal Chomsky dapat dibuat dari sebuah tata bahasa bebas konteks yang telah mengalami penyederhanaan yaitu penghilangan produksi useless, unit, dan ϵ .

Proses konversi ke CNF dapat dilakukan dengan beberapa cara, salah satunya adalah menggunakan algoritma Cocke-Younger-Kasami (CYK). Algoritma CYK bekerja dengan cara membagi string input menjadi beberapa bagian, kemudian menggunakan aturan produksi CFG untuk membangun pohon sintaksis dari bawah ke atas. Alasan mengapa CFG perlu dikonversi ke CNF adalah karena CNF memiliki beberapa sifat yang menguntungkan, seperti yang telah disebutkan di atas. Selain itu, konversi ke CNF juga dapat membantu untuk menyederhanakan tata bahasa, sehingga lebih mudah untuk dipahami dan digunakan.

Berikut adalah beberapa keuntungan dari konversi ke CNF:

1. Analisis yang Lebih Mudah: CNF menyederhanakan struktur grammar, membuat analisis sintaktis dan struktural bahasa menjadi lebih mudah.
2. Efisiensi Algoritma: Banyak algoritma analisis sintaks dirancang untuk bekerja lebih efisien dengan CFG dalam bentuk CNF.
3. Keterbacaan dan Keseragaman: CNF menciptakan format standar, meningkatkan keterbacaan dan keseragaman grammar.
4. Pencarian Struktur: Struktur pohon derivasi menjadi lebih teratur dalam CNF, membantu pencarian struktur sintaktis dengan lebih efisien.
5. Keamanan Analisis: CNF memastikan grammar tidak mengandung elemen yang kompleks atau ambigu, meningkatkan keamanan dalam analisis sintaktis

Berikut adalah hasil konversi dari *original grammar CFG* menjadi *chomsky normal form* :

Original Grammar	Chomsky Normal Form
$K \rightarrow S P$ $K \rightarrow S P O$ $K \rightarrow S P Pel$ $K \rightarrow S P Ket$ $K \rightarrow S P O Pel$ $K \rightarrow S P O Ket$ $K \rightarrow S P Pel Ket$ $K \rightarrow S P O Pel Ket$	$K \rightarrow S P$ $K \rightarrow P1 O$ $P1 \rightarrow S P$ $K \rightarrow P1 Pel$ $K \rightarrow P1 Ket$ $K \rightarrow P2 Pel$ $P2 \rightarrow P1 O$ $K \rightarrow P2 Ket$ $K \rightarrow P3 Ket$ $P3 \rightarrow P1 Pel$ $K \rightarrow P4 Ket$ $P4 \rightarrow P2 Pel$
$S \rightarrow NP$	$S \rightarrow NP AdjP$ $S \rightarrow NumP NP$ $S \rightarrow NP Pronoun$ $S \rightarrow NP PropNoun$ $S \rightarrow NP Noun$ $S \rightarrow Budi \mid Maria \mid Rahayu \mid Ronaldo \mid Agus \mid Andi \mid Rina \mid Kiki \mid$ $Sinta \mid Adi \mid Wayan \mid Kilimanjaro \mid Tanzania \mid Bandung \mid Imlek \mid$ $Indonesia \mid Dinka \mid Jimbaran \mid Udayana \mid Bali \mid Jakarta \mid David \mid$ $Sarah \mid London \mid Paris \mid Berlin \mid Juan \mid Sofia \mid Mexico City \mid$ $Cancun \mid Guadalajara \mid Alessio \mid Isabella \mid Rome \mid Florence \mid Milan$ $\mid Chen \mid Mei \mid Beijing \mid Shanghai \mid Guangzhou \mid Pablo \mid Isabella \mid$ $Buenos Aires \mid Cordoba \mid Rosario \mid Ali \mid Fatima \mid Istanbul \mid Ankara$

	Izmir Alex Emma Sydney Melbourne Brisbane Diego Lucia Lima Arequipa Cusco Ivan Natasha Moscow Saint Petersburg Kazan Raj Priya Mumbai Delhi Bangalore Saya Kamu Dia Kita Ini Itu Aku Engkau Ia Mereka Sendiri Dirimu Dirinya tersebut beliau Anda kelas Ketua Ibu Dapur dengan sekolah juara kejurnas Bapak rapat gol mainan anak Matematika kelereng taman kursi susu ekor ayam gunung baju pesta suku tradisi adik toko bank pulau air makanan sejajar meja lembaga negara lainnya baju bapak ibu ayam pintu guru olahraga adik meja pak tukang sepeda mobil paman ayah guru bayi seorang penculik santan daging kelapa makanan sungai sampah jamu rempah kopi hitam gelas keramik tanaman kakak alat musik pohon babi rambut tepung beras tas jaket bulu domba gitar awetan kulit buaya kayu pertanyaan mobil sayur rumah tugas lukisan orang bubur perbaikan penagihan kota polisi utang posisi tanah kedatangan pencuri tahun pilihan sungai
P →	P → Prep NP Prep AdjP NP AdjP AdvP VP Adv AdjP NumP VP NP PropNoun AdvP PP P → mencari berjumpa dilakukan mengucapkan menjadi mempersilakan masuk mengurung tutup dipukul diliburkan dimainkan bernyanyi belajar disebabkan menyiram tinggal berjalan ditemukan menabung menarik memikat pergi memiliki berlatih menonton mengunjungi pulang bermain menendang keluar bekerja berterimakasih membahas memberikan dilatih pergi memasak mencatat datang bekerja merokok ditemukan menyentuh dibuat berangkat menangis tersedu-sedu paham mencari kerja berkunjung berlari berada bersikap kirim berbicara meletakkan

	<p> meloncat tinggal mengintip masuk duduk bersiap-siap masuk mati mengerjakan berjalan terpilih pindah ikut bertemu memiliki memutuskan pindah bekerja sampai pulang membawa berada merupakan berharap berakhir terletak menginap pergi bermain terjebak dibersihkan mencegah bangun terasa memilih meningkatkan mengatasi sengaja diletakkan digunakan menghemat diperlukan membuat senantiasa dijaga menghindari bekerja sama tercapai ditutup mempercepat sakit terluka jatuh pura-pura pergi dimarahi pulang masuk melanggar mengurangi bepergian mencegah mengikuti diperbaiki menjaga mendapatkan tumbuh dirawat dilakukan berlari berteriak bernyanyi memainkan bekerja menikmati berolahraga mencuri bersama tertarik minum digedornya berteriak berjalan berangkat dituntut sengaja dipesan bernyanyi bunuh berbahasa bepergian tinggal bermain berkhotbah datang berdiskusi dipersilahkan memasuki terbuat terkenang menjadi berjumpa sengaja bertemu serupa terjadi datang berangkat bergantung keluar menyesal berdamai mengeluh berlawanan tergolong bercerita berdiskusi terbuat berhadapan bertemu siap bisa terdiri ditentukan pergi pulang duduk disusun disetrika dimasak dibuka belajar dipasang mengendarai diajarkan dimarahi dibuang dibersihkan disiram ditarik guling merupakan adalah ditanam berdebat bercerita menjawab mengetahui mempertanyakan didorong mogok membangkang bergurau dilukis berdiskusi menjelaskan menyangkul berasal berbelanja pergi terkena tidur terbuat dibuat menghadiri mempersiapkan teringat mempunyai makan pasti serius dini sepi baik keras takut enak menarik ramah indah lengkap penting keras mandiri kecil baru hormat sendiri mudah terkenal asing sulit tinggi malas panik macet subur baik lincah lantang semangat cerdik seru kotor </p>
--	--

	terbersih rendah hemat sakit lapar ramai tenang malas adiktif terdahulu pahit sedikit senang gembira sesuai dulu malang cantik akustik ringan sembelit senang suka
O → NP	<p>O → NP AdjP O → NP Noun O → NP Pronoun O → NP PropNoun O → NumP NP</p> <p>O → Budi Maria Rahayu Ronaldo Agus Andi Rina Kiki Sinta Adi Wayan Kilimanjaro Tanzania Bandung Imlek Indonesia Dinka Jimbaran Udayana Bali Jakarta David Sarah London Paris Berlin Juan Sofia Mexico City Cancun Guadalajara Alessio Isabella Rome Florence Milan Chen Mei Beijing Shanghai Guangzhou Pablo Isabella Buenos Aires Cordoba Rosario Ali Fatima Istanbul Ankara Izmir Alex Emma Sydney Melbourne Brisbane Diego Lucia Lima Arequipa Cusco Ivan Natasha Moscow Saint Petersburg Kazan Raj Priya Mumbai Delhi Bangalore Saya Kamu Dia Kita Ini Itu Aku Engkau Ia Mereka Sendiri Dirimu Dirinya tersebut beliau Anda kelas Ketua Ibu Dapur dengan sekolah juara kejurnas Bapak rapat gol mainan anak Matematika kelereng taman kursi susu ekor ayam gunung baju pesta suku tradisi adik toko bank pulau air makanan sejajar meja lembaga negara lainnya baju bapak ibu ayam pintu guru olahraga adik meja pak tukang sepeda mobil paman ayah guru bayi seorang penculik santan daging kelapa makanan sungai sampah jamu rempah kopi hitam gelas keramik tanaman kakak alat musik pohon babi rambut tepung beras tas jaket bulu domba gitar awetan kulit buaya kayu pertanyaan mobil sayur rumah tugas lukisan orang bubur perbaikan </p>

	penagihan kota polisi utang posisi tanah kedatangan pencuri tahun pilihan sungai
Pel → NP Pel → VP Pel → AdjP	Pel → NP AdjP Pel → NP Noun Pel → NP Pronoun Pel → NP PropNoun Pel → NumP NP Pel → VP NP Pel → PP NP Pel → AdvP VP Pel → AdvP NP Pel → Adv AdjP Pel → Budi Maria Rahayu Ronaldo Agus Andi Rina Kiki Sinta Adi Wayan Kilimanjaro Tanzania Bandung Imlek Indonesia Dinka Jimbaran Udayana Bali Jakarta David Sarah London Paris Berlin Juan Sofia Mexico City Cancun Guadalajara Alessio Isabella Rome Florence Milan Chen Mei Beijing Shanghai Guangzhou Pablo Isabella Buenos Aires Cordoba Rosario Ali Fatima Istanbul Ankara Izmir Alex Emma Sydney Melbourne Brisbane Diego Lucia Lima Arequipa Cusco Ivan Natasha Moscow Saint Petersburg Kazan Raj Priya Mumbai Delhi Bangalore Saya Kamu Dia Kita Ini Itu Aku Engkau Ia Mereka Sendiri Dirimu Dirinya tersebut beliau Anda kelas Ketua Ibu Dapur dengan sekolah juara kejurmas Bapak rapat gol mainan anak Matematika kelereng taman kursi susu ekor ayam gunung baju pesta suku tradisi adik toko bank pulau air makanan sejajar meja lembaga negara lainnya baju bapak ibu ayam pintu guru olahraga adik meja pak

	<p> tukang sepeda mobil paman ayah guru bayi seorang penculik santan daging kelapa makanan sungai sampah jamu rempah kopi hitam gelas keramik tanaman kakak alat musik pohon babi rambut tepung beras tas jaket bulu domba gitar awetan kulit buaya kayu pertanyaan mobil sayur rumah tugas lukisan orang bubur perbaikan penagihan kota polisi utang posisi tanah kedatangan pencuri tahun pilihan sungai ...</p>
<p>Ket → PP</p> <p>Ket → NumP</p> <p>Ket → AdjP</p>	<p>Ket → PP NP</p> <p>Ket → Prep AdjP</p> <p>Ket → NumP NP</p> <p>Ket → Adv AdjP</p> <p>Ket → suatu semua pertama semua ribuan banyak setiap semua satu lima belas beberapa banyak segelas berulang kali empat dua Num NumP pasti serius dini sepi baik keras takut enak menarik ramah indah lengkap penting keras mandiri kecil baru hormat sendiri mudah terkenal asing sulit tinggi malas panik macet subur baik lincah lantang semangat cerdik seru kotor terbersih rendah hemat sakit lapar ramai tenang malas adiktif terdahulu pahit sedikit senang gembira sesuai dulu malang cantik akustik ringan sembelit senang</p>
<p>NP → NP AdjP</p> <p>NP → NP Noun</p> <p>NP → NP Pronoun</p> <p>NP → NP PropNoun</p> <p>NP → NumP NP</p> <p>NP → Pronoun</p> <p>NP → PropNoun</p>	<p>NP → NP AdjP</p> <p>NP → NP Noun</p> <p>NP → NP Pronoun</p> <p>NP → NP PropNoun</p> <p>NP → NumP NP</p> <p>NP → NP VP</p> <p>NP → Dia saya aku</p> <p>NP → Budi Maria Budi Rahayu Ronaldo Agus Andi Rina</p>

<p>NP → Noun</p>	<p> Kiki Sinta Adi Wayan Kilimanjaro Tanzania Bandung Imlek Indonesia Dinka</p> <p>NP → kelas Ketua Ibu Dapur dengan ayahnya sekolah juara kejurnas Bapak rapat gol mainan anak Matematika kelereng taman kursi segelas susu ekor ayam gunung baju pesta suku tradisi adik</p>
<p>VP → PP</p> <p>VP → AdjP</p> <p>VP → Verb</p>	<p>VP → Prep NP</p> <p>VP → Prep AdjP</p> <p>VP → AdvP VP</p> <p>VP → Adv AdjP</p> <p>VP → mencari berjumpa dilakukan mengucapkan menjadi mempersilakan masuk mengurung tutup dipukul diliburkan dimainkan bernyanyi belajar disebabkan menyiram tinggal berjalan ditemukan menabung menarik memikat pergi memiliki berlatih menonton mengunjungi pulang bermain menendang keluar bekerja berterimakasih membahas memberikan dilatih pergi memasak mencatat datang bekerja merokok ditemukan menyentuh dibuat berangkat menangis tersedu-sedu paham mencari kerja berkunjung berlari berada bersikap kirim berbicara meletakkan meloncat tinggal mengintip masuk duduk bersiap-siap masuk mati mengerjakan berjalan terpilih pindah ikut bertemu memiliki memutuskan pindah bekerja sampai pulang membawa berada merupakan berharap berakhir terletak menginap pergi bermain terjebak dibersihkan mencegah bangun terasa memilih meningkatkan mengatasi sengaja diletakkan digunakan menghemat diperlukan membuat senantiasa dijaga menghindari bekerja sama tercapai ditutup mempercepat sakit terluka jatuh pura-pura </p>

	pergi dimarahi pulang masuk melanggar mengurangi bepergian mencegah mengikuti diperbaiki menjaga mendapatkan tumbuh dirawat dilakukan berlari berteriak bernyanyi memainkan bekerja menikmati berolahraga mencuri bersama tertarik minum digedornya berteriak berjalan berangkat dituntut sengaja dipesan bernyanyi bunuh berbahasa bepergian tinggal bermain berkhotbah datang berdiskusi dipersilahkan memasuki terbuat terkenang menjadi berjumpa sengaja bertemu serupa terjadi datang berangkat bergantung keluar menyesal berdamai mengeluh berlawanan tergolong bercerita berdiskusi terbuat berhadapan bertemu siap bisa terdiri ditentukan pergi pulang duduk disusun disetrika dimasak dibuka belajar dipasang mengendarai diajarkan dimarahi dibuang dibersihkan disiram ditarik guling merupakan adalah ditanam berdebat bercerita menjawab mengetahui mempertanyakan didorong mogok membangkang bergurau dilukis berdiskusi menjelaskan menyangkul berasal berbelanja pergi terkena tidur terbuat dibuat menghadiri mempersiapkan teringat mempunyai makan
NumP → NumP NP NumP → Num	NumP → NumP NP NumP → 5
PP → Prep NP PP → Prep	PP → Prep NP PP → Prep AdjP di ke dari dalam sejak menuju dengan tanpa pada sejak antara
AdjP → Adv AdjP AdjP → Adj	AdjP → Adv AdjP AdjP → pasti serius dini sepi baik keras takut enak

	menarik ramah indah lengkap penting keras mandiri kecil baru hormat sendiri mudah terkenal asing sulit tinggi malas panik macet subur baik lincah lantang semangat cerdik seru kotor terbersih rendah hemat sakit lapar ramai tenang malas adiktif terdahulu pahit sedikit senang gembira sesuai dulu malang cantik akustik ringan sembelit senang suka
AdvP → Adv AdvP AdvP → Adv	AdvP → Adv AdvP AdvP → masih akan sangat sudah pernah sungguh-sungguh belum sedang harus jarang akan sangat sedang baru saja sudah tidak boleh belum sekali tentu harus sering hampir selalu juga sedang sudah baru saja selalu masih lebih sebelah secara bukan dengan harus sering larut tidak jarang sambil terus bersama bisa sangat sengaja sering tidak pertamakalinya pasti sudah sangat belum jarang masih beberapa kali harus akan selalu sedang telah ingin tidak sedang baru saja sudah selalu sering telah
NumP → Num NumP NumP → Num	NumP → Num NumP NumP → suatu semua pertama semua ribuan banyak setiap semua satu lima belas beberapa banyak segelas berulang kali empat dua

2.3 Algoritma CYK

CYK (Cocke-Younger-Kasami) adalah sebuah algoritma yang digunakan untuk melakukan parsing atau analisis sintaksis terhadap kalimat berdasarkan suatu Context-Free Grammar (CFG). CYK termasuk dalam kategori algoritma parsing yang menggunakan pendekatan bottom-up, yang berarti dimulai dari kata-kata atau token terkecil untuk membangun struktur sintaksis secara bertahap.

Proses kerja CYK melibatkan tabel dinamis berukuran $n \times n$, di mana n adalah panjang kalimat atau jumlah kata dalam kalimat. Setiap sel dalam tabel ini berisi himpunan non-terminal yang mungkin pada posisi tertentu dalam kalimat. Aturan-aturan produksi dari

CFG digunakan untuk mengisi tabel dan membangun struktur sintaksis secara berurutan. Jika simbol start (biasanya "S" untuk Sentence) terdapat di sel paling atas dan paling kanan tabel, maka kalimat tersebut dapat dihasilkan oleh CFG.

2.3.1 Contoh Penerapan Algoritma CYK secara manual

Contoh kalimat : “Saya sedang bermain di taman”

Mengisi baris pertama sesuai dengan set of product yang telah dibuat.

{S, NP, Noun}	{Adv}	{P, Pel, VP, Verb}	{Ket, PP, Prep}	{O, Ket, NP, Noun, Pel}
Saya	sedang	bermain	di	taman

Mengisi baris kedua dengan menggunakan rumus : $(X_{i,i} , X_{i+1, j})$, sehingga menjadi seperti berikut:

$$\begin{aligned}
 X_{1,2} &= (X_{1,1} , X_{2,2}) \\
 &= \{S, NP, Noun\} \{Adv\} \\
 &= \{S Adv, NP Adv, Noun Adv\} \\
 &= \emptyset
 \end{aligned}$$

$$\begin{aligned}
 X_{2,3} &= (X_{2,2} , X_{3,3}) \\
 &= \{Adv\} \{P, Pel, VP, Verb\} \\
 &= \{Adv P, Adv Pel, Adv VP, Adv Verb\} \\
 &= \{P, VP\}
 \end{aligned}$$

$$X_{3,4} = (X_{3,3} , X_{4,4})$$

$$\begin{aligned}
&= (X_{3,3}, X_{4,4}) \\
&= \{P, Pel, VP, Verb\} \{Ket, PP, Prep\} \\
&= \{P Ket, P PP, P Prep, Pel Ket, Pel PP, Pel Prep, VP Ket, VP PP, VP Prep, Verb Ket, \\
&\quad Verb PP, Verb Prep\} \\
&= \emptyset
\end{aligned}$$

$$\begin{aligned}
X_{4,5} &= (X_{i,i}, X_{i+1,j}) \\
&= (X_{4,4}, X_{5,5}) \\
&= \{Ket, PP, Prep\} \{O, Ket, NP, Noun, Pel\} \\
&= \{Ket O, Ket Ket, Ket NP, Ket NP, Ket Noun, Ket Pel, PP O, PP Ket, PP NP, PP \\
&\quad Noun, PP Pel, Prep O, Prep Ket, Prep NP, Prep Noun, Prep Pel\} \\
&= \{P, Ket, VP, PP\}
\end{aligned}$$

\emptyset	{P, VP}	\emptyset	{P, Ket, VP, PP}	
{S, NP, Noun}	{Adv}	{P, Pel, VP, Verb}	{Ket, PP, Prep}	{O, Ket, NP, Noun, Pel}
Saya	sedang	bermain	di	taman

Mengisi baris ketiga dengan menggunakan rumus : (X_i, i, X_{i+1}, j) (X_{i+1}, X_{i+2}, j) , sehingga menjadi seperti berikut.

$$\begin{aligned}
X_{1,3} &= (X_{i,i}, X_{i+1,j}) (X_{i+1}, X_{i+2}, j) \\
&= (X_{1,1}, X_{2,3}), (X_{1,2}, X_{3,3}) \\
&= \{S, NP, Noun\} \{P, VP\} \cup \emptyset \{P, Pel, VP, Verb\} \\
&= \{S P, S VP, NP P, NP VP, Noun P, Noun VP\} \\
&= \{P1\}
\end{aligned}$$

$$\begin{aligned}
X_{2,4} &= (X_{i,i}, X_{i+1,j}) (X_{i,i+1}, X_{i+2,j}) \\
&= (X_{2,2}, X_{3,4}), (X_{2,3}, X_{4,4}) \\
&= \{\text{Adv}\} \emptyset \cup \{P, VP\} \{Ket, PP, Prep\} \\
&= \{P Ket, P PP, P Prep, VP Ket, VP PP, VP Prep\} \\
&= \emptyset
\end{aligned}$$

$$\begin{aligned}
X_{3,5} &= (X_{i,i}, X_{i+1,j}) (X_{i,i+1}, X_{i+2,j}) \\
&= (X_{3,3}, X_{4,5}), (X_{3,4}, X_{5,5}) \\
&= \{P, Pel, VP, Verb\} \{P, Ket, VP, PP\} \cup \emptyset \{O, Ket, NP, Noun, Pel\} \\
&= \{P P, P Ket, P VP, P PP, Pel P, Pel Ket, Pel VP, Pel PP, VP P, VP Ket, VP VP, VP PP, Verb P, Verb Ket, Verb VP, Verb PP\} \\
&= \emptyset
\end{aligned}$$

{K, P1}	∅	∅		
∅	{P, VP}	∅	{P, Ket, VP, PP}	
{S, NP, Noun}	{Adv}	{P, Pel, VP, Verb}	{Ket, PP, Prep}	{O, Ket, NP, Noun, Pel}
Saya	sedang	bermain	di	taman

Mengisi baris keempat dengan menggunakan rumus : $(X_{i,i}, X_{i+1,j}) (X_{i,i+1}, X_{i+2,j}) (X_{i,i+2}, X_{i+3,j})$, sehingga menjadi seperti berikut

$$\begin{aligned}
X_{1,4} &= (X_{i,i}, X_{i+1,j}) (X_{i,i+1}, X_{i+2,j}) (X_{i,i+2}, X_{i+3,j}) \\
&= (X_{1,1}, X_{2,4}) (X_{1,2}, X_{3,4}) (X_{1,3}, X_{4,4}) \\
&= \{S, NP, Noun\} \emptyset \cup \emptyset \emptyset \cup \{K, P1\} \{Ket, PP, Prep\} \\
&= \{K Ket, K PP, K Prep, P1 Ket, P1 PP, P1 Prep\}
\end{aligned}$$

$$= K$$

$$\begin{aligned}
X_{2,5} &= (X_{i,i}, X_{i+1,j}) (X_{i,i+1}, X_{i+2,j}) (X_{i,i+2}, X_{i+3,j}) \\
&= (X_{2,2}, X_{3,5}) (X_{2,3}, X_{4,5}) (X_{2,4}, X_{5,5}) \\
&= \{Adv\} \emptyset \cup \{P, VP\} \{P, Ket, VP, PP\} \cup \emptyset \{O, Ket, NP, Noun, Pel\} \\
&= \{P P, P Ket, P VP, P PP, VP P, VP Ket, VP VP, VP PP\} \\
&= \emptyset
\end{aligned}$$

K	\emptyset			
{K, P1}	\emptyset	\emptyset		
\emptyset	{P, VP}	\emptyset	{P, Ket, VP, PP}	
{S, NP, Noun}	{Adv}	{P, Pel, VP, Verb}	{Ket, PP, Prep}	{O, Ket, NP, Noun, Pel}
Saya	sedang	bermain	di	taman

Mengisi baris keempat dengan menggunakan rumus : $(X_{i,i}, X_{i+1,j}) (X_{i,i+1}, X_{i+2,j}) (X_{i,i+2}, X_{i+3,j}) (X_{i,i+3}, X_{i+4,j})$, sehingga menjadi seperti berikut.

$$\begin{aligned}
X_{1,5} &= (X_{i,i}, X_{i+1,j}) (X_{i,i+1}, X_{i+2,j}) (X_{i,i+2}, X_{i+3,j}) (X_{i,i+3}, X_{i+4,j}) \\
&= (X_{1,1}, X_{2,5}) (X_{1,2}, X_{3,5}) (X_{1,3}, X_{4,5}) (X_{1,4}, X_{5,5}) \\
&= \{S, NP, Noun\} \emptyset \cup \emptyset \emptyset \cup \{P1\} \{P, Ket, VP, PP\} \cup \{K\} \{O, Ket, NP, Noun, Pel\} \\
&= \{P1 P, P1 Ket, P1 VP, P1 PP\} \{K O, K Ket, K NP, K Noun, K Pel\} \\
&= K
\end{aligned}$$

K	
K	\emptyset

{K, P1}	Ø	Ø		
Ø	{P, VP}	Ø	{P, Ket, VP, PP}	
{S, NP, Noun}	{Adv}	{P, Pel, VP, Verb}	{Ket, PP, Prep}	{O, Ket, NP, Noun, Pel}
Saya	sedang	bermain	di	taman

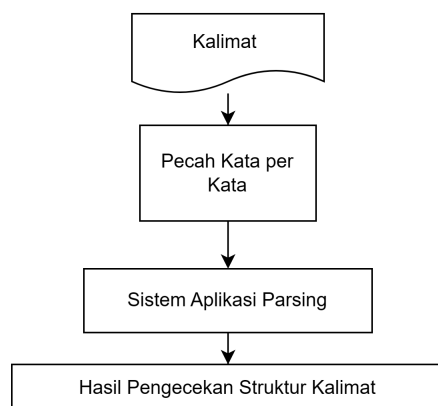
Karena terdapat K yang merupakan start symbol pada $X_{1,5}$ maka kalimat “Saya Sedang bermain di taman” adalah “Valid” atau “Diterima”.

BAB III

ANALISIS DAN DESAIN

3.1 Desain Eksperimen

3.2.1 Skenario Eksperimen



Gambar 3.1 Bagan Pengujian Aplikasi Parsing

Pengujian pada aplikasi dilakukan dengan menginputkan berbagai macam kalimat bahasa Indonesia dengan pola kalimat sederhana. Kalimat sederhana yang dimaksud yaitu kalimat tunggal dengan struktur pola kalimat, seperti S P, S P O, S P Pel, S P O Ket, S P O Pel, S P Pel Ket, dan S P O Pel Ket. Kalimat di luar dari pola kalimat berikut akan tidak dapat diterima oleh sistem (lebih detailnya ada pada pembahasan). Melalui eksperimen ini juga dilakukan apakah pernyataan tersebut benar atau tidak. Ketika kalimat sudah diinputkan, nantinya, sistem akan melakukan pengecekan apakah kalimat tersebut dapat diterima atau valid, atau tidak dapat diterima/ invalid.

Eksperimen juga dilakukan dengan kata-kata yang berbagai macam jenisnya. Apakah memang benar, terbatas pada dataset yang telah diberikan sebelumnya pada laman pembelajaran penulis, atau tidak. Begitu halnya dengan kebakuan dari bahasa atau kata yang diinputkan oleh pengguna, eksperimen tersebut terkait dengan tata baku bahasa indonesia yang diterapkan dalam sistem. Berbeda dengan sebelumnya yang terpaku pada rules dari CFG yang telah dikonversi menjadi CNF, di luar rules tersebut apakah masih memungkinkan untuk menerima kalimat yang diinputkan pengguna atau tidak.

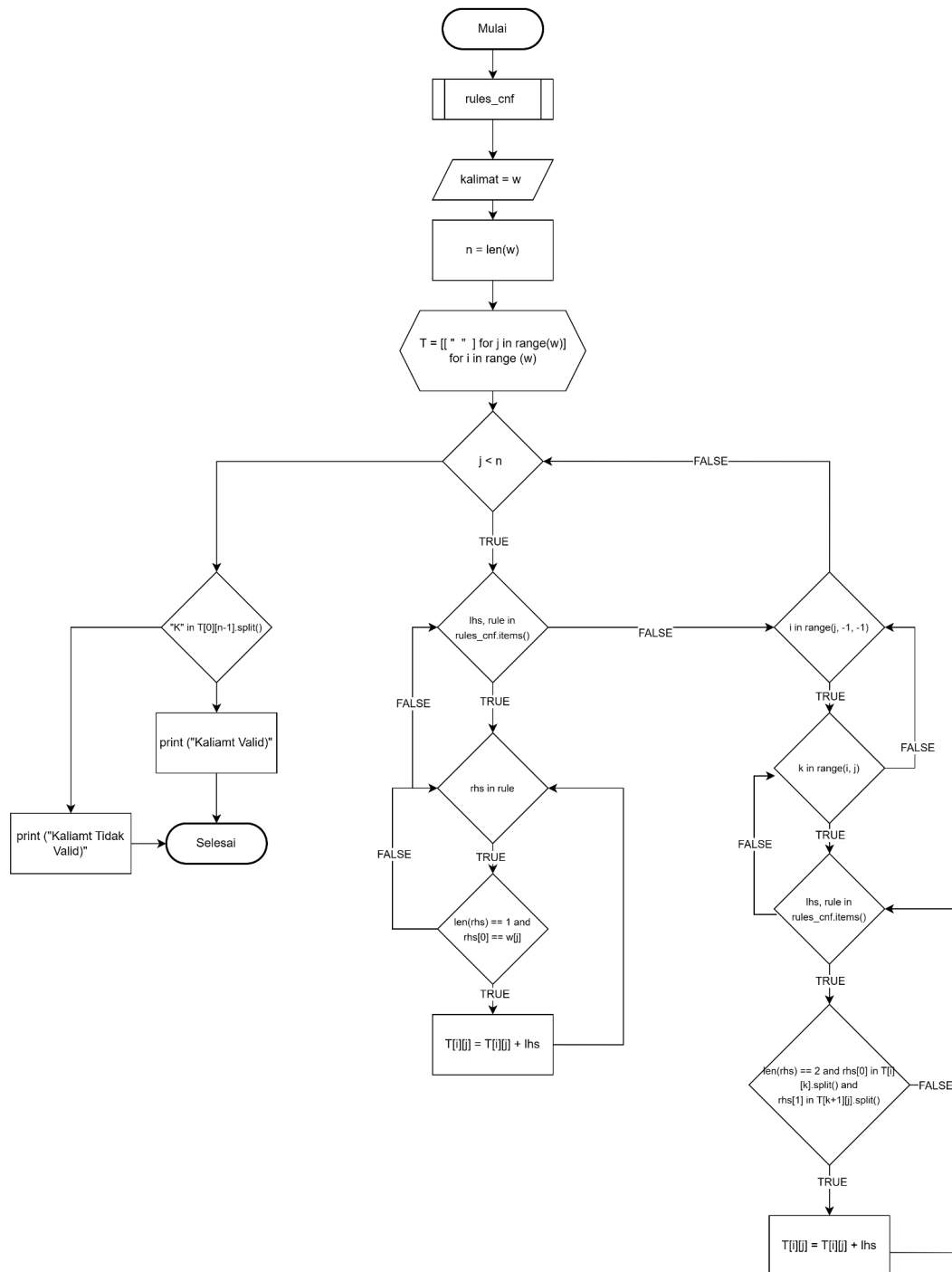
3.2.2 Ukuran Evaluasi

Sebagai upaya indikator yang mesti dipenuhi dari sistem telah dirancang, penulis menetapkan ukuran keberhasilan dari sebuah pengecekan kalimat yang

dilakukan oleh sistem. Dimulai dari kalimat yang berbentuk kalimat tunggal, hingga pada ketepatan setiap kata yang telah dipecah telah sesuai pada pola kalimat yang telah ditetapkan. Sehingga apabila terpenuhi dengan baik, maka sistem dapat berjalan sesuai dengan fungsinya.

3.2 Desain Aplikasi

3.2.1 Flowchart



Gambar 3.2 Flowchart algoritma CYK

Gambar berikut menjelaskan terkait dengan diagram alir dari sebuah algoritma CYK. Algoritma ini dimulai dari inisialisasi sebuah rules yang diperoleh dari rules CFG yang sudah dikonversi menjadi CNF, yaitu hanya memiliki dua buah variabel atau non-terminals atau hanya memiliki satu buah terminal. Kemudian dilanjutkan dengan inputan sebuah kalimat yang nantinya, kalimat tersebut akan dipecah sesuai dengan jumlah kata yang terkandung. Barulah akan dijalankan algoritma CYK dengan melakukan pengecekan tiap-tiap kata yang ada pada kalimat tersebut. Apabila, pada pengecekan paling terakhir menemukan sebuah value bernilai “K” yang artinya start symbol, maka sistem akan mencetak bahwa kalimat tersebut adalah kalimat valid. Apabila tidak maka sistem akan mencetak pesan “kalimat tidak valid”.

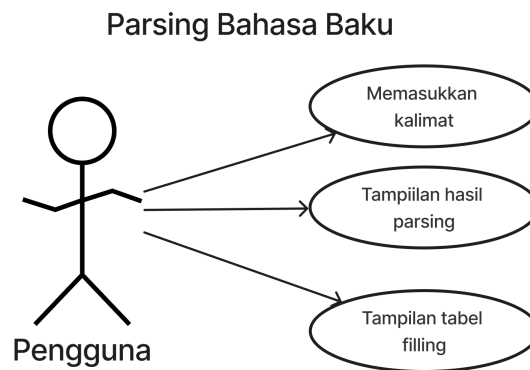
3.2.2 Mockup

The mockup shows a simple application interface within a rectangular frame. At the top, there are three horizontal lines of varying lengths, likely representing a title bar or decorative elements. Below these, there is a large rectangular input field labeled "Input kalimat". Underneath the input field is a smaller rectangular button labeled "Check Kalimat". At the bottom of the interface is a wide rectangular output field labeled "Valid / tidaknya kaliaamt".

Gambar 3.3 Rancangan Aplikasi

Gambar berikut merupakan rancangan tampilan yang dibuat oleh penulis. Tampilannya cukup sederhana, yaitu dimulai dari judul, kemudian form untuk input kalimat, tombol untuk pengecekan dan sebuah info ketika kalimat yang diinputkan sudah valid atau belum.

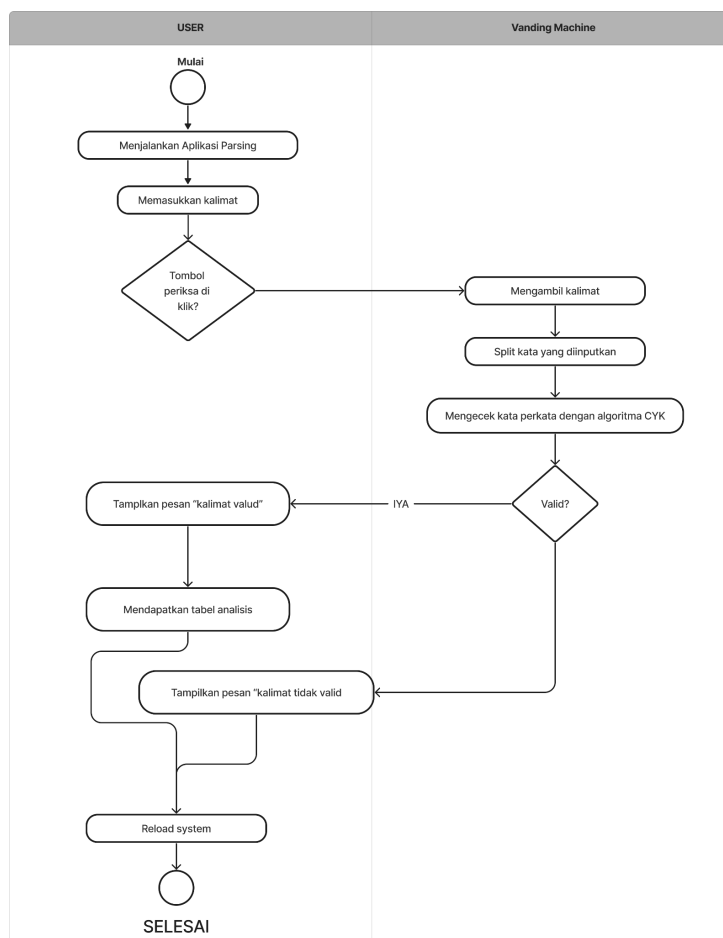
3.2.3 Use Case Diagram



Gambar 3.4 *use case diagram* aplikasi parsing

Pada gambar 3.4, 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 filling dan pohon parsing.

3.2.4 Activity Diagram




Gambar 3.5 *activity diagram* aplikasi parsing

Pada gambar berikut, *activity diagram* atau diagram aktivitas merupakan sebuah runtutan atau urutan aktivitas proses secara jelas sehingga lebih mudah untuk dikomunikasikan ke berbagai pihak. Hal ini terlihat dapat dipahami lebih mudah dan jelas ketika dikomunikasikan dengan pihak luar.

3.2.5 Source Code

Dalam implementasinya, perancang menyusun kodingan dengan membagi ke dalam beberapa file yang berbeda, yaitu `web.py`, `main.py`, `parsing_cyk.py`, dan `rules.py`. Tiap-tiap file memiliki fungsi dan peranannya tersendiri, dimulai dari fungsi `main`, yaitu sebagai berikut :



```
# import the web view
from web import tampilan_web

# run run_streamlit() function for the first time use
if __name__ == "__main__":
    tampilan_web()
```

Gambar 3.6 Source code dari `main.py`

Fungsi berikut yaitu untuk menjalankan tampilan web dari sistem yang dirancang oleh penulis. Fungsi ini dipanggil dari file `web` dan mengambil fungsi `tampilan_web`. Nantinya fungsi yang tertulis di `tampilan_web` akan muncul disini. Lalu isi dari file `tampilan_web` adalah sebagai berikut. Fungsi `tampilan_web` akan menjalankan baris kode berikut dengan bantuan kerangka kerja dari Python yaitu `streamlit`. Fungsi ini bertujuan untuk menampilkan judul, nama, tampilan form untuk input kalimat, dan juga tampilan hasil parsing yang dilakukan dari sistem. Nantinya, di dalam fungsi ini akan memanggil fungsi lainnya dari file `parsing_cyk` dengan fungsi yaitu `algoritma_cyk`. Jadi istilahnya ada yang berfokus pada tampilan saja, kemudian ada yang berfokus pada algoritmanya saja. Berikut adalah isi dari file `parsing_cyk`.

```

import streamlit as st

from parsing_cyk import algortima_cyk

def tampilan_web():
    st.set_page_config(layout='centered', page_title="Parsing Kalimat Baku Bahasa Indonesia",
    menu_items={
        'About': f"""
            ## "Final Project Teori Bahasa dan Otomata"
            ##### Parsing kalimat baku bahasa Indonesia menggunakan algoritma CYK
            Oleh Kelompok 3 Kelas E :\\n
            I Ngurah Komang Agus Suryadiyatmika. S      (2208561031)\\n
            I Kadek Revan Aditya Prawira                (2208561050)\\n
            Devon Vivian Gunawan                        (2208561081)\\n
            I Putu Aditya Pradana                        (2208561124)\\n
            GitHub:
            """
    })

    st.write(f"<h1 style='text-align:center;'>Parsing Kalimat Baku Bahasa Indonesia</h1>",
    unsafe_allow_html=True)
    st.write(f"<h5 style='text-align:center;'>Menggunakan Algoritma CYK | Mata Kuliah Teori Bahasa dan  
Otmata</h5>", unsafe_allow_html=True)
    st.write(f"<p style='text-align:center;'>---- Kelompok 3 Kelas E -----</p>",
    unsafe_allow_html=True)

    input_kalimat = st.text_input('Masukkan Kalimat Anda :')
    list_string = input_kalimat.split(' ')

    button_click = st.button('Check', type='primary')

    if button_click:
        if len(list_string) <= 1: #list_string sudah berupa array, dengan index masing" adalah kata
            dari kalimat yang diinput
                st.error("Kalimat Tidak Boleh Kosong atau Hanya 1 kata!")
        elif input_kalimat != '':
            st.write('<h3>Hasil Parsing: </h3>', unsafe_allow_html=True)
            algortima_cyk(input_kalimat.split())

```

Gambar 3.7 Source code web.py

```

import streamlit as st
from rules import RULES_CFG

def algortima_cyk(kalimat):
    empty = '\u2205'
    banyak_kata = len(kalimat)
    T = [[' ' for j in range(banyak_kata)] for i in range(banyak_kata)]
    # filling_table = [[' ' for j in range(banyak_kata)] for i in range(banyak_kata)]

    # print(T);

    for j in range(0, banyak_kata):
        for LHS, rule in RULES_CFG.items():
            for RHS in rule:
                if len(RHS) == 1 and RHS[0] == kalimat[j]:
                    T[j][j] += LHS + " "

    for j in range(0, banyak_kata):
        for i in range(j, -1, -1):
            for k in range(i, j):
                for LHS, rule in RULES_CFG.items():
                    for RHS in rule:
                        if len(RHS) == 2 and RHS[0] in T[i][k].split() and RHS[1] in T[k+1]
[j].split():
                            T[i][j] += LHS + " "
                    # print(T)

    if "K" in T[0][banyak_kata-1].split():
        print("True")
        st.success("Berdasarkan hasil pemeriksaan, kalimat yang diinputkan adalah kalimat VALID")
        st.write('<h3>Filling Table:</h3>', unsafe_allow_html=True)
        st.table(T)

    else:
        print("False")
        st.error("Berdasarkan hasil pemeriksaan, kalimat yang diinputkan adalah kalimat TIDAK VALID")

# algortima_cyk("Kiki tertidur di kampus".split())

```

Gambar 3.8 Source code parsing_cyk.py

Di file berikut algoritma CYK akan berperan penring dalam melakukan parsing kalimat yang diinput oleh user. Disini kalimat yang diinput akan diolah, dimulai dari menentukan banyak kata yang terkandung di kalimat tersebut dengan melalui fungsi split terlebih dahulu. Kemudian menyiapkan sebuah array dua dimensi dengan panjang banyak_kata yang sudah didapat sebelumnya. Array tersebut berperan seperti tabel untuk melakukan pengecekan kalimat melalui metode *filling table* untuk dapat memastikan apakah kalimat yang diinputkan sudah valid atau belum

Pengisian value dari masing-masing index terjadi secara dua tahap (dua perulangan). Loop pertama (`for j in range(0, banyak_kata)`) mengulangi setiap kata dalam kalimat. Kode tersebut memeriksa aturan dengan simbol terminal tunggal (kata) pada RHS yang cocok dengan kata saat ini. Jika ditemukan, kode tersebut menambahkan LHS yang sesuai ke $T[j][j]$, menunjukkan parse potensial untuk kata tunggal yang dimaksud.

Kemudian untuk perulangan yang kedua yaitu di bagian (`for j in range(0, banyak_kata)`) mengulangi panjang substring, dimulai dari satu kata dan memperluas ke frasa yang lebih besar. Loop bersarang (`for i in range(j, -1, -1)` dan `for k in range(i, j)`) akan tereksekusi dengan kemungkinan cara menggabungkan substring yang diparse yang lebih kecil menjadi yang lebih besar. Kode tersebut memeriksa apakah dua substring yang diparse ($T[i][k]$ dan $T[k+1][j]$) dapat digabungkan menggunakan aturan dengan dua non-terminal pada RHS. Jika ditemukan, kode tersebut menambahkan LHS dari aturan tersebut ke $T[i][j]$, mewakili parse potensial untuk substring yang lebih besar.

Setelah seluruh perulangan berakhir, untuk mengecek apakah memang benar kalimat yang diinputkan valid atau tidak yaitu dilakukan pengecekan pada index $[0][\text{banyak_kata} - 1]$ apakah terdapat simbol non-terminal “K”. Mengingat “K” merupakan start symbol sehingga apabila terdapat K di saat perulangan terakhir, maka kalimat tersebut merupakan kalimat yang valid. Lalu, aturan tersebut bisa didapat di file `rules.py` yang berisikan seluruh rules baik itu non terminal ataupun terminal dari pola kalimat yang telah ditentukan. Berikut adalah tampilan sebagian dari isi file tersebut.

```

RULES_CFG = {
    "K": [{"S", "P"}, {"P1", "O"}, {"P1", "Pel"}, {"P1", "Ket"}, {"P2", "Pel"}, {"P2", "Ket"},
    {"P3", "Ket"}, {"P4", "Ket"}],
    "P1": [{"S", "P"}],
    "P2": [{"P1", "O"}],
    "P3": [{"P1", "Pel"}],
    "P4": [{"P2", "Pel"}],
    "S": [{"NP", "AdjP"}, {"NumP", "NP"}, {"NP", "Noun"}, {"NP", "PropNoun"}, {"NP", "Pronoun"},
    {"Budi"}, {"Maria"}, {"Rahayu"}, {"Ronaldo"}, {"Agus"}, {"Andi"}, {"Rina"}, {"Kiki"}, {"Sinta"},
    {"Adi"}, {"Wayan"}, {"Kilimanjaro"}, {"Tanzania"}, {"Bandung"}, {"Imlek"}, {"Indonesia"}, {"Dinka"},
    {"Jimbaran"}, {"Udayana"}, {"Bali"}, {"Jakarta"}, {"David"}, {"Sarah"}, {"London"}, {"Paris"},
    {"Berlin"}, {"Juan"}, {"Sofia"}, {"Mexico City"}, {"Cancun"}, {"Guadalajara"}, {"Alessio"},
    {"Isabella"}, {"Rome"}, {"Florence"}, {"Milan"}, {"Chen"}, {"Mei"}, {"Beijing"}, {"Shanghai"},
    {"Guangzhou"}, {"Pablo"}, {"Isabella"}, {"Buenos Aires"}, {"Cordoba"}, {"Rosario"}, {"Ali"},
    {"Fatima"}, {"Istanbul"}, {"Ankara"}, {"Izmir"}, {"Alex"}, {"Emma"}, {"Sydney"}, {"Melbourne"},
    {"Brisbane"}, {"Diego"}, {"Lucia"}, {"Lima"}, {"Arequipa"}, {"Cusco"}, {"Ivan"}, {"Natasha"},
    {"Moscow"}, {"Saint Petersburg"}, {"Kazan"}, {"Raj"}, {"Priya"}, {"Mumbai"}, {"Delhi"}, {"Bangalore"},
    {"Saya"}, {"Kamu"}, {"Dia"}, {"Kita"}, {"Ini"}, {"Itu"}, {"Aku"}, {"Engkau"}, {"Ia"}, {"Mereka"},
    {"Sendiri"}, {"Dirimu"}, {"Dirinya"}, {"tersebut"}, {"beliau"}, {"Anda"}, {"kelas"}, {"Ketua"},
    {"Ibu"}, {"Dapur"}, {"dengan"}, {"sekolah"}, {"juara"}, {"kejujurnas"}, {"Bapak"}, {"rapat"}, {"gol"},
    {"mainan"}, {"anak"}, {"Matematika"}, {"kelereng"}, {"taman"}, {"kursi"}, {"susu"}, {"ekor"}, {"ayam"},
    {"gunung"}, {"baju"}, {"pesta"}, {"suku"}, {"tradisi"}, {"adik"}, {"toko"}, {"bank"}, {"pulau"},
    {"air"}, {"makanan"}, {"sejajar"}, {"meja"}, {"lembaga"}, {"negara"}, {"lainnya"}, {"baju"}, {"bapak"},
    {"ibu"}, {"ayam"}, {"pintu"}, {"guru"}, {"olahraga"}, {"adik"}, {"meja"}, {"pak"}, {"tukang"},
    {"sepeda"}, {"mobil"}, {"paman"}, {"ayah"}, {"guru"}, {"bayi"}, {"seorang"}, {"penculik"}, {"santan"},
    {"daging"}, {"kelapa"}, {"makanan"}, {"sungai"}, {"sampah"}, {"jamu"}, {"rempah"}, {"kopi"}, {"hitam"},
    {"gelas"}, {"keramik"}, {"tanaman"}, {"kakak"}, {"alat"}, {"musik"}, {"pohon"}, {"babi"}, {"rambut"},
    {"tepung"}, {"beras"}, {"tas"}, {"jaket"}, {"bulu"}, {"domba"}, {"gitar"}, {"awetan"}, {"kulit"},
    {"buaya"}, {"kayu"}, {"pertanyaan"}, {"mobil"}, {"sayur"}, {"rumah"}, {"tugas"}, {"lukisan"},
    {"orang"}, {"bubur"}, {"perbaikan"}, {"penagihan"}, {"kota"}, {"polisi"}, {"utang"}, {"posisi"},
    {"tanah"}, {"kedatangan"}, {"pencuri"}, {"tahun"}, {"pilihan"}, {"sungai"}, {"Nenek"}, {"Banyak"}],

```

Gambar 3.9 Source Code dari rules.py

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Eksperimen

Untuk menguji apakah sistem yang dirancang oleh penulis sudah sesuai dengan pola kata bahasa yang berlaku, pengajar menyediakan 100 jenis kalimat yang berbeda dengan pola kata yang berbeda pula, dengan 80 kalimat diantaranya tergolong kalimat baku dan 20 kalimat sisanya adalah kalimat tidak baku. Berikut adalah tabel pengujiannya.

Kalimat	Pola Kalimat	Baku	Tidak Baku	Pengujian Sistem	
				Sesuai	Tidak Sesuai
Nenek mengambil sedikit daun ubi di kebun.	S P O Ket	✓		✓	
Banyak orang menghadiri acara itu.	S P O	✓		✓	
Sekitar lima mahasiswa mengikuti kegiatan pengabdian tersebut	S P O	✓		✓	
Orang dewasa saja diperbolehkan menaiki wahana itu	S P Pel	✓		✓	
Pemadam kebakaran itu hanya mempunyai sedikit waktu untuk menyelamatkan penduduk desa tersebut.	S P O Pel	✓		✓	
Anak laki-laki itu jarang bermain di taman kota	S P Ket	✓		✓	
Ibu memasak banyak makanan lezat	S P O	✓		✓	
Para siswa sering belajar di perpustakaan kampus	S P Ket	✓		✓	
Polisi selalu menjaga ketertiban di jalan.	S P O Ket	✓		✓	
Burung merpati itu hanya terbang di siang hari	S P Ket	✓		✓	
Ikan koi banyak berenang di kolam taman	S P Ket	✓		✓	
Kuda pacuan itu selalu berlari dengan	S P Pel Ket	✓		✓	

cepat di lintasan pacu					
Monyet kecil itu sering bergelantungan di ranting pohon	S P Ket	✓		✓	
Kelinci putih itu melompat-lompat di kebun.	S P Ket	✓		✓	
Lumba-lumba itu bermain di laut lepas	S P Ket	✓		✓	
Kupu-kupu sering hinggap di bunga	S P Ket	✓		✓	
Sapi selalu mengunyah rumput di padang rumput.	S P O Ket	✓		✓	
Lebah itu mengumpulkan banyak nektar dari bunga	S P O Pel	✓		✓	
Seekor buaya sedang berjemur di tepi danau	S P Ket	✓		✓	
Banyak mahasiswa hadir pada acara seminar itu	S P Pel	✓		✓	
Dr. Siti Rahayu memberikan kuliah di universitas.	S P O Ket	✓		✓	
Ani sedang berlibur di pantai	S P Pel	✓		✓	
Adi Santoso merupakan pemimpin proyek tersebut	S P Pel	✓		✓	
Bu Ratna adalah pemilik warung kopi itu	S P Pel	✓		✓	
Prof. Dr. Hadi Prayitno adalah pakar bidang ekologi	S P Pel	✓		✓	
Kiki menjadi juara pada lomba menyanyi	S P Pel	✓		✓	
Ibu Sinta akan menjadi pembicara dalam seminar itu	S P Pel Ket	✓		✓	
Bapak Haryono selalu membantu tetangga	S P O	✓		✓	
Laut Mediterania sangat terkenal dengan keindahan pantainya	S P Pel	✓		✓	
Bukit Tinggi memiliki pemandangan alam indah	S P O	✓		✓	

Danau Baikal adalah danau terdalam di dunia	S P Pel Ket	✓		✓	
Pantai Copacabana menjadi ikon kota Rio de Janeiro	S P Pel	✓		✓	
Bali selalu terkenal dengan keindahan pantainya	S P Pel	✓		✓	
Kota Bandung itu terkenal dengan kulinernya	S P Pel	✓		✓	
Bogor dikenal dengan kota hujan	S P Pel	✓		✓	
Monas terletak di tengah Jakarta	S P Ket	✓		✓	
Bali memiliki banyak budaya indah	S P Pel	✓		✓	
Denpasar sangat ramai pada malam hari	S P Ket	✓		✓	
Surabaya merupakan kota terbesar di Jawa Timur	S P Pel Ket	✓		✓	
Pentas ogoh-ogoh dilakukan menyambut Tahun Baru Saka	S P Pel	✓		✓	
Arab Saudi merupakan negara penghasil minyak terbesar di dunia	S P Pel Ket	✓		✓	
Indonesia adalah negara kepulauan terbesar di dunia	S P Pel Ket	✓		✓	
Lima belas pasang sepatu terpajang di rak sepatu	S P Ket	✓		✓	
Delapan buah rumah sedang dalam tahap konstruksi	S P	✓		✓	
Dua ekor kucing selalu bermain di taman belakang rumah	S P Ket	✓		✓	
Sepuluh orang siswa sedang belajar di perpustakaan	S P Ket	✓		✓	
Tujuh puluh lima orang pengunjung hadir dalam acara tersebut	S P Pel	✓		✓	
Kamar tidur ini sangat nyaman	S P	✓		✓	
Pertunjukan teater ini sangat indah	S P	✓		✓	
Mobil tersebut sudah lama digunakan	S P Pel	✓		✓	

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

Prestasi anak-anak itu adalah kebanggaan orang tua	S P Pel	✓		✓	
Rasa coklat menjadi kelemahannya	S P Pel	✓		✓	
Pohon rindang itu memberikan kesejukan	S P O	✓		✓	
Air hujan membuat suasana romantis	S P O	✓		✓	
Cita-cita tinggi adalah pendorong kesuksesan	S P Pel	✓		✓	
Kucing itu hewan peliharaan manis	S P	✓		✓	
Pameran itu menampilkan karya seni kontemporer	S P O	✓		✓	
Budaya lokal menjadi daya tarik wisata daerah itu	S P Pel	✓		✓	
Hewan peliharaan ini adalah teman setia keluarga kami	S P Pel	✓		✓	
Sangat menyukai kursi rotan			✓		✓
Pelatih tari itu			✓		✓
Rencana perjalanan kita selalu			✓		✓
Pengembangan teknologi			✓		✓
Gadis kecil itu			✓		✓
Air terjun pegunungan itu			✓		✓
Sangat menghibur			✓		✓
Memiliki sejarah panjang			✓		✓
Kebanggaan taman kita			✓		✓
Menjadi teman setia anak-anak			✓		✓
Memberikan kesegaran pikiran			✓		✓
Kebanggaan orang tua			✓		✓
Rasa coklat menjadi			✓		✓
Air hujan membuat			✓		✓
Pohon rindang itu memberikan			✓		✓

Cita-cita tinggi adalah pendorong kesuksesan			✓	✓	
Hewan peliharaan manis			✓		✓
Karya seni kontemporer			✓		✓
Menjadi daya tarik wisata daerah itu			✓		✓
Teman setia keluarga kami			✓		✓

4.2 Pembahasan dan Analisis Hasil

Contoh kalimat : “Ibu memasak makanan lezat”

Mengisi baris pertama sesuai dengan set of product yang telah dibuat.

{S, NP, Noun}	{P, VP, Verb}	{O, NP, Noun}	{O, NP, AdjP, Adj}
Ibu	memasak	makanan	lezat

Mengisi baris kedua dengan menggunakan rumus : $(X_{i,i} , X_{i+1, j})$, sehingga menjadi seperti berikut:

$$\begin{aligned}
 X_{1,2} &= (X_{i,i} , X_{i+1, j}) \\
 &= (X_{1,1} , X_{2,2}) \\
 &= \{S, NP, Noun\} \{P, VP, Verb\} \\
 &= \{S P, S VP, S Verb, NP P, NP VP, NP Verb, Noun P, Noun VP, Noun Verb\} \\
 &= \{K, P1\}
 \end{aligned}$$

$$\begin{aligned}
 X_{2,3} &= (X_{i,i} , X_{i+1, j}) \\
 &= (X_{2,2} , X_{3,3}) \\
 &= \{P, VP, Verb\} \{O, NP, Noun\} \\
 &= \{P O, P NP, P Noun, VP O, VP NP, VP Noun, Verb O, Verb NP, Verb Noun\}
 \end{aligned}$$

$$= \emptyset$$

$$\begin{aligned}
X_{3,4} &= (X_{i,i}, X_{i+1}, j) \\
&= (X_{3,3}, X_{4,4}) \\
&= \{O, NP, Noun\} \{O, NP, AdjP, Adj\} \\
&= \{O O, O NP, O AdjP, O Adj, NP O, NP NP, NP AdjP, NP Adj, Noun O, Noun NP, \\
&\text{Noun AdjP, Noun Adj}\} \\
&= \{S, O, P, NP\}
\end{aligned}$$

{K, P1}	\emptyset	{S, O, P, NP}	
{S, NP, Noun}	{P, VP, Verb}	{O, NP, Noun}	{O, NP, AdjP, Adj}
Ibu	memasak	makanan	lezat

Mengisi baris ketiga dengan menggunakan rumus : $(X_i, i, X_{i+1}, j) (X_i, i+1, X_{i+2}, j)$, sehingga menjadi seperti berikut.

$$\begin{aligned}
X_{1,3} &= (X_{i,i}, X_{i+1}, j) (X_{i,i+1}, X_{i+2}, j) \\
&= (X_{1,1}, X_{2,3}), (X_{1,2}, X_{3,3}) \\
&= \{S, NP, Noun\} \emptyset \cup \{K, P1\} \{O, NP, Noun\} \\
&= \{K O, K NP, K Noun, P1 O, P1 NP, P1 Noun\} \\
&= \{K, P2\}
\end{aligned}$$

$$\begin{aligned}
X_{2,4} &= (X_{i,i}, X_{i+1}, j) (X_{i,i+1}, X_{i+2}, j) \\
&= (X_{2,2}, X_{3,4}), (X_{2,3}, X_{4,4}) \\
&= \{P, VP, Verb\} \{S, O, P, NP\} \cup \emptyset \{O, NP, AdjP, Adj\} \\
&= \{P S, P O, P P, P NP, VP S, VP O, VP P, VP NP, Verb S, Verb O, Verb P, Verb NP\} \\
&=
\end{aligned}$$

{K, P2}	Ø		
{K, P1}	Ø	{S, O, P, NP}	
{S, NP, Noun}	{P, VP, Verb}	{O, NP, Noun}	{O, NP, AdjP, Adj}
Ibu	memasak	makanan	lezat

$$\begin{aligned}
X_{1,4} &= (X_{i,i}, X_{i+1,j}) (X_{i,i+1}, X_{i+2,j}) (X_{i,i+2}, X_{i+3,j}) \\
&= (X_{1,1}, X_{2,4}) (X_{1,2}, X_{3,4}) (X_{1,3}, X_{4,4}) \\
&= \{S, NP, Noun\} \emptyset \cup \{K, P1\} \{S, O, P, NP\} \cup \{K, P2\} \{O, NP, AdjP, Adj\} \\
&= \{K S, K O, K P, K NP, P1 S, P1 O, P1 P, P1 NP, K O, K NP, K AdjP, K Adj, P2 O, \\
&P2 NP, P2 AdjP, P2 Adj\} \\
&= \{K, P2\}
\end{aligned}$$

{K, P2}			
{K, P2}	Ø		
{K, P1}	Ø	{S, O, P, NP}	
{S, NP, Noun}	{P, VP, Verb}	{O, NP, Noun}	{O, NP, AdjP, Adj}
Ibu	memasak	makanan	lezat

Karena terdapat K yang merupakan start symbol pada $X_{1,4}$ maka kalimat “Ibu Memasak makanan lezat” adalah “Valid” atau “Diterima”.

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan dari hasil pembahasan laporan ini adalah sebagai berikut :

- Penerapan CFG dalam parsing kalimat baku bahasa Indonesia memerlukan aturan-aturan yang akan digunakan dalam CFG tersebut. Aturan dapat dibuat sendiri atau menggunakan aturan-aturan yang telah dikembangkan sebelumnya sesuai dengan sintaksis yang diinginkan. Setelah aturan-aturan ditentukan, kita dapat mem parsing CFG untuk menganalisis sebuah kalimat dan menentukan struktur sintaksisnya. Parsing CFG dapat dilakukan dengan membaca kalimat dari kiri ke kanan atau dari kanan ke kiri, lalu mencocokkan setiap kata dengan aturan yang telah ditentukan, dan menghasilkan sebuah pohon sintaksis yang menggambarkan struktur kalimat tersebut. Namun sebelum melakukan parsing, kita perlu menyediakan sebuah input yang terdiri dari kalimat baku bahasa Indonesia yang ingin dianalisis, serta aturan-aturan yang telah ditentukan dalam CFG.
- Algoritma CYK dalam mengimplementasikan CFG dengan menggunakan salah satu metode yaitu *table filling* yang dibuat dengan membangun tabel segitiga bawah atau segitiga atas. Kemudian tabel tersebut digunakan dalam membantu pengecekan apakah suatu CFG itu valid atau tidak dengan cara mengisi tabel dari bawah ke atas. Dimana nanti pada final tabel segitiga tersebut jika menghasilkan himpunan kosong atau nilai start symbol tidak ada maka kalimat tersebut tidak valid atau ditolak, namun jika terdapat himpunan yang anggotanya terdapat start symbol maka kalimat tersebut diterima atau valid. Dengan kata lain apabila kalimat yang akan diinputkan sudah mengikuti aturan tata bahasa baku bahasa Indonesia dengan benar maka kalimat tersebut akan diterima oleh mesin atau valid.

5.2 Daftar Pustaka

- [1] Sulianto, T.T. and Herawati, R. (2021) 'The implementation of CYK algorithm to separate sentence-forming elements in Indonesian', *Proxies : Jurnal Informatika*, 2(2), p. 52. doi:10.24167/proxies.v2i2.3210.
- [2] M, S.G., Haryoko, A. and Nurlifa, A. (2021) *PENENTUAN POLA KALIMAT BAHASA INDONESIA PADA KALIMAT AKTIF DAN KALIMAT PASIF DENGAN MENGGUNAKAN METODE CONTEXT FREE GRAMMER (CFG)*, 2.

[3] Alwi, Hasan, dkk 2003. Tata bahasa Buku Bahasa Indonesia (Edisi Ketiga) Jakarta:Bali Pustaka

[4] Anggara P. A., “Translasi Context-Free Grammar Menjadi Parsing Tree Berbasis Algoritma Cocke-Younger-Kasami”, Ilmu Komputer, Universitas Dian Nuswantoro Semarang, 2015

[5] Eka, A.K. (no date) ‘Parsing - Application of Context-Free Diagram’. Universitas Udayana .