

**TUGAS FINAL PROJECT
TEORI BAHASA DAN OTOMATA**



Oleh :
Kelompok 5

Yasinta Anita Kewa Nilan	(2108561034)
Nyoman Krisna Ari Sudarsana	(2108561072)
I Gede Arisudana Samanjaya	(2108561098)
Rafly Shaquille Subhan	(2108561118)

**KELAS C
PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA
2022**

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam Teori Otomata terdapat pendekatan untuk memetakan struktur suatu bahasa menjadi tata bahasa yang lebih formal. Sejak tahun 1950-an para ahli bahasa telah mempelajari struktur bahasa natural manusia seperti bahasa Inggris dan memetakan tata bahasanya secara formal. Setelah itu pada tahun 1960-an dimana teknik komputasi dan komputer mulai banyak dikembangkan, muncullah bahasa pemrograman untuk menjembatani antara programmer dengan mesin komputer agar mempermudah programmer membuat sebuah program yang memiliki fungsionalitas tertentu. Ilmuwan komputer yang mengembangkan bahasa pemrograman, turut serta memberikan tata bahasa (pemrograman) secara formal. Tata bahasa ini diciptakan secara bebas-konteks dan disebut CFG.

CFG adalah tata bahasa yang mempunyai tujuan sama seperti halnya tata bahasa reguler yaitu merupakan suatu cara untuk menunjukkan bagaimana menghasilkan suatu untai-untai dalam sebuah bahasa, dengan cara menggunakan algoritma CYK. Algoritma CYK adalah algoritma untuk menentukan apakah suatu untai dapat diterima oleh suatu tata bahasa bebas-konteks (CFG) yang sudah dinormalkan dalam bentuk Chomsky Normal Form (CNF). Pengecekan dilakukan dengan mengelompokkan setiap kata sesuai dengan KBBI (Kamus Besar Bahasa Indonesia).

1.2 Problem Komputasi

Adapun problem/permasalahan komputasi pada laporan ini adalah:

Mengembangkan aplikasi parsing sintaksis kalimat bahasa Indonesia menggunakan Algoritma CYK yang memiliki:

Fitur utama: Parsing kalimat

1. Masukkan Kalimat Bahasa Indonesia
2. Menampilkan apakah kalimat itu valid atau tidak valid berdasarkan Tata Bahasa Indonesia.

Fitur opsional: Pohon Parse atau Struktur Kalimat

Jika kalimat itu sah maka:

1. Tampilkan pohon parse dari kalimat, atau
2. Menampilkan struktur kalimat.

1.3 Tujuan

Adapun tujuan dari laporan ini antara lain, yaitu:

1. Untuk mengetahui bagaimana aplikasi CFG dalam penyelesaian masalah komputasi yang ada.
2. Untuk mengetahui bagaimana penggunaan algoritma CYK dalam menyelesaikan masalah komputasi yang ada.
3. Membuat aplikasi yang dapat menyelesaikan masalah komputasi yang ada menggunakan bahasa pemrograman python.

1.4 Manfaat

Adapun manfaat yang didapat dari laporan ini antara lain, yaitu:

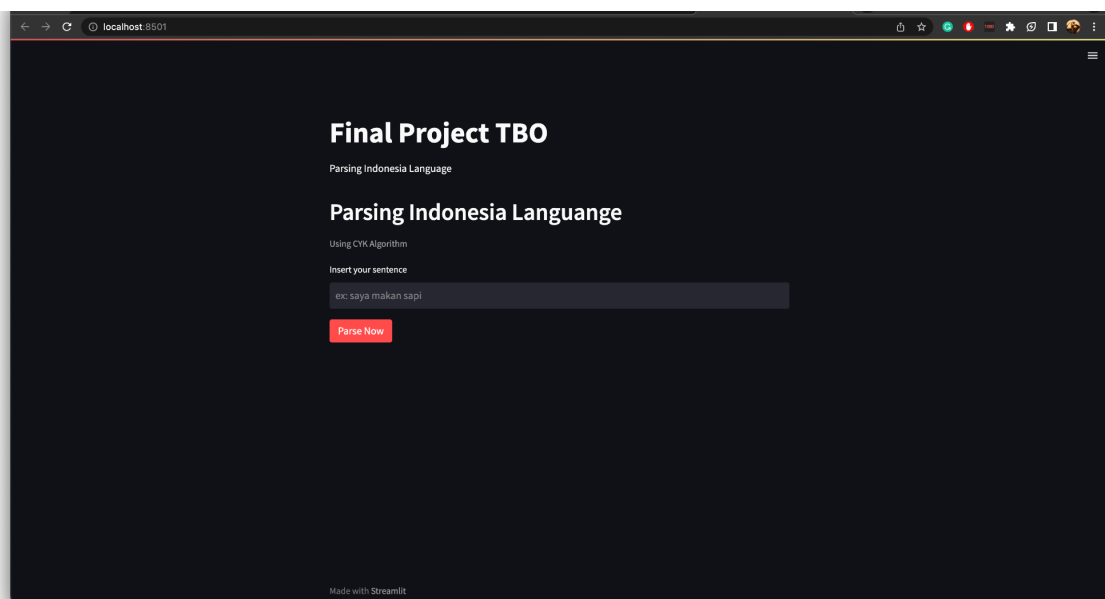
1. Mengetahui bagaimana aplikasi CFG dalam menyelesaikan masalah komputasi yang ada.
2. Mengetahui bagaimana penggunaan algoritma CYK dalam menyelesaikan masalah komputasi yang ada.
3. Terbentuk aplikasi yang dapat menyelesaikan masalah komputasi yang ada menggunakan bahasa pemrograman python.

BAB II

MANUAL APLIKASI

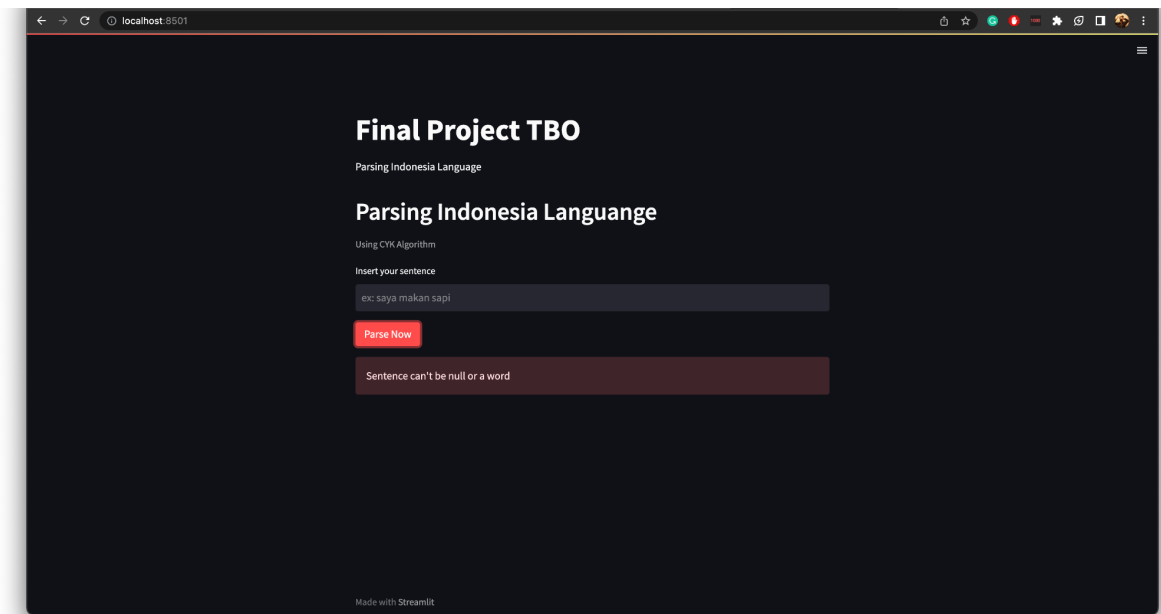
2.1 Fitur Sistem dan Sistem Antarmuka

Pada aplikasi yang kami buat terdapat satu fitur utama yaitu menerima inputan user yang berupa kalimat kemudian mengecek apakah kalimat tersebut baku sesuai dengan set of rule grammar yang telah di buat sebelumnya. Pada aplikasi yang kami buat kami menggunakan frame work **streamlit** untuk membuat tampilan antar muka dari aplikasi kami. Adapun tampilan dari aplikasi kami ketika menerima beberapa inputan sebagai berikut.



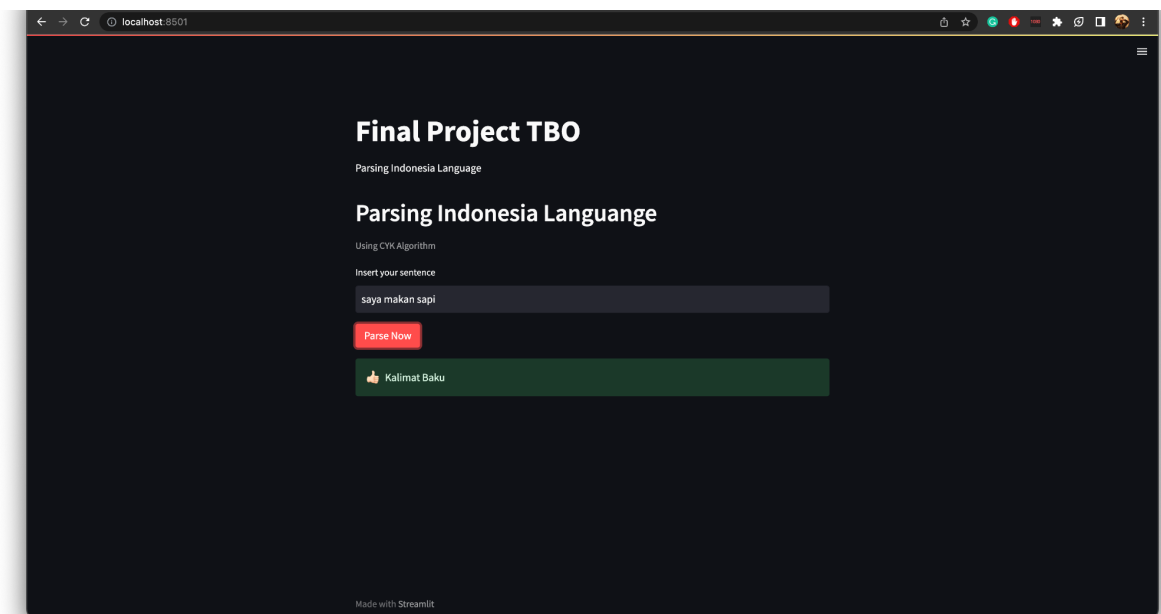
Gambar 2. Tampilan ketika pertama kali aplikasi dijalankan

Gambar diatas merupakan tampilan awal ketika menjalankan program kami, dimana terdapat judul, textbox dan tombol Parse Now untuk mengecek inputan kalimat dari user.

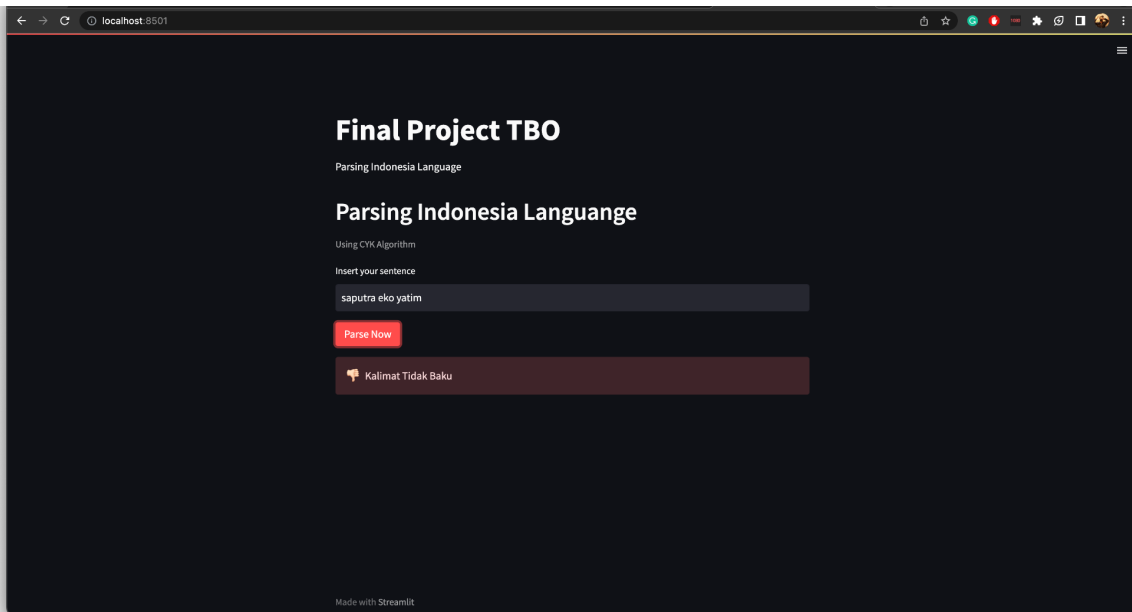


Gambar 3. Gambar tampilan ketika user tidak menginput string atau null

Apabila user tidak menginputkan string atau user menginputkan inputan berupa NULL maka akan muncul pemberitahuan “Sentence can’t be null or a word”



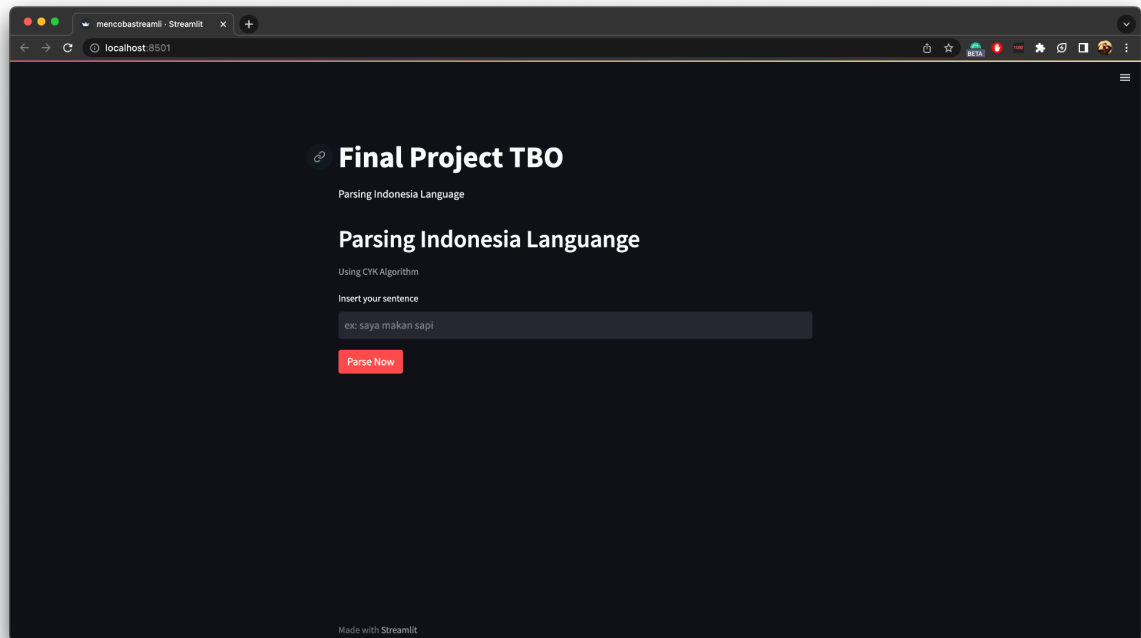
Gambar 4. Gambar tampilan ketika kalimat yang diinput user merupakan kalimat baku



Gambar 5. Gambar tampilan ketika kalimat yang dinput user merupakan kalimat tidak baku

BAB III DESIGN APLIKASI

3.1 Mockup



Pada tampilan aplikasi kami terdapat judul dan beberapa komponen lainnya. Terdapat textbox yang berisi contoh tampilan kalimat yang diterima yaitu “saya makan sapi” kemudian terdapat button/tombol “Parse Now” untuk menerima inputan user dan menjalankan program.

3.2 Source Code

Source code dari program parsing bahasa baku bahasa Indonesia dapat dilihat pada link GitHub berikut :

source code CNF

```

1. RESULT = {} # ini adalah dictionary yang berisi hasil dari proses CNF
2.
3. def remove_unit_production(keyList): # fungsi untuk menghapus produksi unit
4.     global RESULT # mengakses variabel global
5.     for key, value in RESULT.items(): # perulangan untuk setiap key dan value
        pada dictionary
6.         if key in keyList: # jika key ada pada list keyList

```

```

7.         tempList = [] # buat list kosong temporary
8.         for prod in value: # perulangan untuk setiap produksi pada value
9.             if len(prod.split(" ")) == 2: # jika panjang produksi adalah 2
10.                 tempList.append(prod) # tambahkan produksi ke list
                temporary
11.             else: # jika panjang produksi bukan 2
12.                 for i in RESULT[prod]: # perulangan untuk setiap produksi
                pada value dari key
13.                     if i not in tempList: # jika produksi belum ada di
                list temporary
14.                         tempList.append(i) # tambahkan produksi ke list
                temporary
15.             RESULT[key] = tempList # set value dari key dengan list temporary
16.
17. def get_set_of_production(): # fungsi untuk mengambil set produksi
18.     global RESULT # mengakses variabel global
19.     RESULT.clear() # menghapus semua isi dictionary
20.     f = open("/Users/arisamanjaya/Documents/KULIAH BROW/SEMESTER 3 YUK
        SERIUS/Teori Bahasa dan Otomata/TBO Final Project/modules/raw.txt", "r",
        encoding="utf-8") # membuka file variable_list.txt
21.     for lines in f: # perulangan untuk setiap baris pada file
22.         line = lines.splitlines() # memisahkan baris dengan newline
23.         line = line[0].split(" -> ") # memisahkan baris dengan ->
24.         lhs = line[0] # mengambil bagian kiri
25.         rhs = line[1].split(" | ") # mengambil bagian kanan dan memisahkan
                dengan |
26.         if lhs in RESULT.keys(): # jika key sudah ada di dictionary
27.             RESULT[lhs].extend(rhs) # tambahkan value ke list value
28.         else: # jika key belum ada di dictionary
29.             RESULT[lhs] = rhs # buat key baru dengan value baru
30.     f.close() # menutup file
31.     for key, value in RESULT.items(): # perulangan untuk setiap key dan value
        pada dictionary
32.         if key == "PropNoun": # jika key adalah PropNoun
33.             tempList = [] # buat list kosong temporary

```



```

34.         for val in value: # perulangan untuk setiap produksi pada value
35.             if val not in tempList: # jika produksi belum ada di list
                temporary
36.                 tempList.append(val.lower()) # tambahkan produksi ke list
                temporary dengan huruf kecil
37.         RESULT[key] = tempList # set value dari key dengan list temporary
38.     phrases = ["NumP", "AdvP", "AdjP", "PP", "NP", "VP"] # list key yang akan
        dihapus produksi unitnya
39.     remove_unit_production(phrases) # panggil fungsi untuk menghapus produksi
        unit
40.     patterns = ["S", "P", "O", "Pel", "Ket"] # list key yang akan dihapus
        produksi unitnya
41.     remove_unit_production(patterns) # panggil fungsi untuk menghapus produksi
        unit
42.     tempList = [] # buat list kosong temporary
43.     tempDict = {} # buat dictionary kosong temporary
44.     counter = 1 # buat counter untuk key baru
45.     for key, value in RESULT.items(): # perulangan untuk setiap key dan value
        pada dictionary
46.         if key == "K": # jika key adalah K
47.             for val in value: # perulangan untuk setiap produksi pada value
48.                 if len(val.split(" ")) > 2: # jika panjang produksi lebih dari
                    2
49.                     temp = val.split(" ") # pisahkan produksi dengan spasi
50.                     while len(temp) > 2: # perulangan selama panjang produksi
                        lebih dari 2
51.                         checkStr = temp[0] + " " + temp[1] # buat string untuk
                            dicek
52.                         isFound = False # buat boolean untuk mengecek apakah
                            string sudah ada di dictionary
53.                         for k, v in tempDict.items(): # perulangan untuk
                            setiap key dan value pada dictionary temporary
54.                             if checkStr == v: # jika string sudah ada di
                                dictionary
55.                                 isFound = True # set boolean menjadi True
56.                                 temp.pop(0) # hapus produksi yang sudah dicek
57.                                 temp.pop(0) # hapus produksi yang sudah dicek

```

```

58.             temp.insert(0, k) # tambahkan key baru ke
           produksi
59.             break # hentikan perulangan
60.             if not isFound: # jika string belum ada di dictionary
61.                 tempDict["K" + str(counter)] = checkStr #
           tambahkan key baru dengan value baru ke dictionary temporary
62.                 temp.pop(0) # hapus produksi yang sudah dicek
63.                 temp.pop(0) # hapus produksi yang sudah dicek
64.                 temp.insert(0, "K" + str(counter)) # tambahkan key
           baru ke produksi
65.                 counter += 1 # tambahkan counter
66.                 tempList.append(" ".join(temp)) # tambahkan produksi ke
           list temporary
67.             else: # jika panjang produksi tidak lebih dari 2
68.                 tempList.append(val) # tambahkan produksi ke list
           temporary
69.             RESULT[key] = tempList # set value dari key dengan list temporary
70.             for key, value in tempDict.items(): # perulangan untuk setiap key dan
           value pada dictionary temporary
71.                 RESULT[key] = [value] # set value dari key dengan value baru
72.             return RESULT # kembalikan dictionary

```

Pada source code diatas berfungsi untuk membaca file raw.txt yang merupakan set of rule yang telah dibuat kemudian akan dikonversi menjadi format cnf sehingga bisa dibaca oleh sistem nantinya.

Source code CYK

```

1. from modules.cnf import get_set_of_production
2. # ini adalah library yang digunakan untuk mengimport fungsi
   get_set_of_production
3.
4. TRIANGULAR_TABLE = {} # ini adalah tabel yang digunakan untuk menyimpan hasil
   perhitungan CYK
5. g = None # ini adalah variabel global yang digunakan untuk menyimpan graf
   yang dibangun dari inputan user
6. previousNode = None # ini adalah variabel global yang digunakan untuk
   menyimpan node sebelumnya

```

```

7.
8. def is_accepted(inputString): # ini adalah fungsi yang digunakan untuk
    mengecek apakah inputan user diterima atau tidak
9.     global TRIANGULAR_TABLE # ini adalah variabel global yang digunakan untuk
    menyimpan tabel yang digunakan untuk menyimpan hasil perhitungan CYK
10.    TRIANGULAR_TABLE.clear() # ini adalah fungsi yang digunakan untuk
    menghapus isi dari tabel yang digunakan untuk menyimpan hasil perhitungan CYK
11.    prodRules = get_set_of_production() # ini adalah variabel yang digunakan
    untuk menyimpan hasil dari fungsi get_set_of_production
12.    temp = inputString.lower().split(" ") # ini adalah variabel yang digunakan
    untuk menyimpan hasil dari fungsi get_set_of_production
13.    inputString = temp # ini adalah variabel yang digunakan untuk menyimpan
    hasil dari fungsi get_set_of_production
14.    for i in range(1, len(inputString)+1): # ini adalah perulangan yang
    digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
    perhitungan CYK
15.        for j in range(i, len(inputString)+1): # ini adalah perulangan yang
    digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
    perhitungan CYK
16.            TRIANGULAR_TABLE[(i,j)] = [] # ini adalah perulangan yang
    digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
    perhitungan CYK
17.    for i in reversed(range(1, len(inputString)+1)): # ini adalah perulangan
    yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
    perhitungan CYK
18.        for j in range(1, i+1): # ini adalah perulangan yang digunakan untuk
    mengisi tabel yang digunakan untuk menyimpan hasil perhitungan CYK
19.            if (j == j + len(inputString) - i): # ini adalah perulangan yang
    digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
    perhitungan CYK
20.                tempList = [] # ini adalah perulangan yang digunakan untuk
    mengisi tabel yang digunakan untuk menyimpan hasil perhitungan CYK
21.                for key, value in prodRules.items(): # ini adalah perulangan
    yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
    perhitungan CYK
22.                    for val in value: # ini adalah perulangan yang digunakan
    untuk mengisi tabel yang digunakan untuk menyimpan hasil perhitungan CYK

```

```

23.         if (val == inputString[j-1] and key not in tempList):
            # ini adalah perulangan yang digunakan untuk mengisi tabel yang digunakan
            untuk menyimpan hasil perhitungan CYK
24.         tempList.append(key) # ini adalah perulangan yang
            digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
            perhitungan CYK
25.         TRIANGULAR_TABLE[(j, j + len(inputString) - i)] = tempList #
            ini adalah perulangan yang digunakan untuk mengisi tabel yang digunakan untuk
            menyimpan hasil perhitungan CYK
26.         else: # ini adalah perulangan yang digunakan untuk mengisi tabel
            yang digunakan untuk menyimpan hasil perhitungan CYK
27.         tempList = [] # ini adalah perulangan yang digunakan untuk
            mengisi tabel yang digunakan untuk menyimpan hasil perhitungan CYK
28.         resultList = [] # ini adalah perulangan yang digunakan untuk
            mengisi tabel yang digunakan untuk menyimpan hasil perhitungan CYK
29.         for k in range(len(inputString) - i): # ini adalah perulangan
            yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
            perhitungan CYK
30.         first = TRIANGULAR_TABLE[(j,j+k)] # ini adalah perulangan
            yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
            perhitungan CYK
31.         second = TRIANGULAR_TABLE[(j+k+1,j+len(inputString) - i)]
            # ini adalah perulangan yang digunakan untuk mengisi tabel yang digunakan
            untuk menyimpan hasil perhitungan CYK
32.         for fi in first: # ini adalah perulangan yang digunakan
            untuk mengisi tabel yang digunakan untuk menyimpan hasil perhitungan CYK
33.         for se in second: # ini adalah perulangan yang
            digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
            perhitungan CYK
34.         if (fi + " " + se not in tempList): # ini adalah
            perulangan yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan
            hasil perhitungan CYK
35.         tempList.append(fi + " " + se) # ini adalah
            perulangan yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan
            hasil perhitungan CYK
36.         for key, value in prodRules.items(): # ini adalah perulangan
            yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
            perhitungan CYK
37.         for val in value: # ini adalah perulangan yang digunakan
            untuk mengisi tabel yang digunakan untuk menyimpan hasil perhitungan CYK

```

```

38.         if (val in tempList and key not in resultList): # ini
           adalah perulangan yang digunakan untuk mengisi tabel yang digunakan untuk
           menyimpan hasil perhitungan CYK
39.         resultList.append(key) # ini adalah perulangan
           yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
           perhitungan CYK
40.         TRIANGULAR_TABLE[(j,j+len(inputString) - i)] = resultList #
           ini adalah perulangan yang digunakan untuk mengisi tabel yang digunakan untuk
           menyimpan hasil perhitungan CYK
41.     if "K" in TRIANGULAR_TABLE[(1, len(inputString))]: # ini adalah perulangan
           yang digunakan untuk mengisi tabel yang digunakan untuk menyimpan hasil
           perhitungan CYK
42.         return True # ini adalah perulangan yang digunakan untuk mengisi tabel
           yang digunakan untuk menyimpan hasil perhitungan CYK
43.     else: # ini adalah perulangan yang digunakan untuk mengisi tabel yang
           digunakan untuk menyimpan hasil perhitungan CYK
44.         return False # ini adalah perulangan yang digunakan untuk mengisi
           tabel yang digunakan untuk menyimpan hasil perhitungan CYK

```

Source code diatas berfungsi untuk melakukan parsing dan table filing dari hasil konversi program file CNF.py

BAB IV

PENUTUP

4.1 Kesimpulan

Tata Bahasa Bebas Konteks (Context Free Grammar atau CFG) adalah tata bahasa yang mempunyai tujuan sama seperti halnya tata bahasa reguler yaitu merupakan suatu cara untuk menunjukkan bagaimana menghasilkan suatu untai-untai dalam sebuah bahasa, dengan cara menggunakan algoritma CYK. Algoritma CYK adalah algoritma untuk menentukan apakah suatu untai dapat diterima oleh suatu tata bahasa bebas-konteks (CFG) yang sudah dinormalkan dalam bentuk Chomsky Normal Form (CNF). Pengecekan dilakukan dengan mengelompokkan setiap kata sesuai dengan KBBI (Kamus Besar Bahasa Indonesia).

DAFTAR PUSTAKA

- [1] Inas. Luthfi, “informatika.stei.itb.ac.id”, 2007. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2007-2008/Makalah2008/MakalahI F2251-2008-079.pdf> [18 Desember 2022]
- [2] Alvina. Aulia, “BINUS UNIVERSITY School of Computer Science”, 20 Desember 2018. [Online]. Available: <https://socs.binus.ac.id/2018/12/20/penyederhanaan-context-free-grammar/> [18 Desember 2022]