***CONTEXT-FREE GRAMMAR* BAHASA INDONESIA DENGAN BAHASA PEMROGRAMAN *PYTHON***



**LAPORAN**  
**Disusun untuk Memenuhi Tugas Individu**  
**Mata Kuliah Pemrosesan Bahasa Alami Semester VIII**  
**yang diampu oleh Sukmawati Nur Endah, S.Si, M.Kom.**

**DISUSUN OLEH:**  
**LINGGAR MARETVA CENDANI (24060117120031)**

**PROGRAM STUDI STRATA 1 INFORMATIKA**  
**DEPARTEMEN ILMU KOMPUTER/INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**  
**SEMARANG**  
**2021**

DAFTAR ISI

[DAFTAR ISI 2](#_Toc52999350)

[BAB I 3](#_Toc52999351)

[PENDAHULUAN 3](#_Toc52999352)

[1.1 Permasalahan 3](#_Toc52999353)

[1.2 Tujuan 3](#_Toc52999354)

[BAB II 4](#_Toc52999355)

[PEMBAHASAN 4](#_Toc52999356)

[2.1 CFG (](#_Toc52999357)*[Context-Free Grammar](#_Toc52999357)*[) Bahasa Indonesia Yang Dibuat 4](#_Toc52999357)

[2.2 Source Code Program](#_Toc52999358) 6

[a. Import Library yang akan Digunakan 6](#_Toc52999357)

[b. Membuat CFG (](#_Toc52999357)*[Context-Free Grammar](#_Toc52999357)*[) Bahasa Indonesia 7](#_Toc52999357)

[c. Melakukan Generate Kalimat Sebanyak 50 Kalimat 7](#_Toc52999357)

[d. Menampilkan](#_Toc52999357) *[Derivation](#_Toc52999357)* [dari Dua Contoh Kalimat 8](#_Toc52999357)

[e. Menyimpan Semua Kemungkinan Hasil Generate Kalimat Dari Grammar Yang Telah Dibuat 8](#_Toc52999357)

[2.3 Output Program 9](#_Toc52999358)

[a. Hasil Grammar Yang Dibuat 9](#_Toc52999357)

[a. Hasil Generate 50 Kalimat 10](#_Toc52999357)

[a. Hasil](#_Toc52999357) *[Derivation](#_Toc52999357)* [dari 2 Contoh Kalimat 11](#_Toc52999357)

[2.4 Analisa Hasil 1](#_Toc52999358)1

[BAB III 1](#_Toc52999359)3

[PENUTUP 1](#_Toc52999360)3

[3.1 Kesimpulan 1](#_Toc52999361)3

BAB I

PENDAHULUAN

1. Permasalahan
2. Bagaimana CFG (*Context-Free Grammar*) Bahasa Indonesia Yang Dibuat?
3. Bagaimana source code program CFG (*Context-Free Grammar*) Bahasa Indonesia Yang Dibuat?
4. Bagaimana output dari program CFG (*Context-Free Grammar*) Bahasa Indonesia Yang Dibuat?
5. Bagaimana analisa hasil dari program CFG (*Context-Free Grammar*) Bahasa Indonesia Yang Dibuat?
6. Tujuan
7. Mengetahui bagaimana CFG (*Context-Free Grammar*) Bahasa Indonesia Yang Dibuat?
8. Mengetahui source code program CFG (*Context-Free Grammar*) Bahasa Indonesia Yang Dibuat?
9. Mengetahui output dari program CFG (*Context-Free Grammar*) Bahasa Indonesia Yang Dibuat?
10. Mengetahui analisa hasil dari program CFG (*Context-Free Grammar*) Bahasa Indonesia Yang Dibuat?

BAB II

PEMBAHASAN

* 1. CFG (Context-Free Grammar) Bahasa Indonesia Yang Dibuat

CFG atau ***Context Free Grammar*** adalah tata bahasa formal di mana setiap aturan produksi adalah dalam bentuk A → B di mana A adalah pemproduksi, dan B adalah hasil produksi. Batasannya adalah ruas kiri (A) adalah non-terminal. Dan pada ruas kanan (B) bisa berupa terminal maupun non-terminal.

CFG yang dibuat pada program ini didapatkan dari Jurnal yang berjudul “*Penerapan Algoritma Lalr Parser Dan Context-Free Grammar Untuk Struktur Kalimat Bahasa Indonesia*” (Pratama et al, 2017). serta Sebuah skripsi berjudul “*Penerapan Algoritma Lalr Parser Dan Context-Free Grammar Untuk Struktur Kalimat Bahasa Indonesia*” (Hayati, 2014).

Dari kedua jurnal dan skripsi di atas, diberikan beberapa pola kalimat seperti SP, SPO, SPK, SPOK, dimana S = Subjek, P = Predikat, O = Objek, dan K = Keterangan.

Dari beberapa contoh kalimat yang terdapat pada jurnal, dilakukan pembuatan rules, dan didapatkan hasil sebagai berikut :

Dengan **G = (N, T ,P, S)**

1. **N :** {S, Sub, P, O, K, KB, KG, KTun, KK, KD, KW, KWS, H, KT}
2. **T :** {anjing, kucing, pohon , bola, susu, emas, cincin, aku, kamu, dia, ayah, ibu, adik, kakak, paman, bibi, kakek, nenek, itu, ini, berenang, menangis, bermain, memanjat, meminum, menjaga, di, dari, ke, pada, siang, pagi, sore, hari, warung, kolam, lapangan, kantor, kampus}
3. **S :** Simbol Permulaan
4. **P :**

|  |
| --- |
| <S> -> <Sub> <P> | <Sub <P> <O> | <Sub> <P> <K> | <Sub> <P> <O> <K>  <Sub> -> <KB> | <KG> | <KB> <KG> | <KB> <KTun>  <P> -> <KK>  <O> -> <KB> | <KG> | <KB> <KG>  <K> -> <KD> <KW> | <KD> <KT>  <KB> -> {anjing, kucing, pohon , bola, susu, emas, cincin}  <KG> -> {aku, kamu, dia, ayah, ibu, adik, kakak, paman, bibi, kakek, nenek}  <KTun> -> {itu, ini}  <KK> -> {berenang, menangis, bermain, memanjat, meminum, menjaga}  <KD> -> {di, dari, ke, pada}  <KW> -> <KWS> <H>  <KWS> -> {siang, pagi, sore}  <H> -> {hari}  <KT> -> {warung, kolam, lapangan, kantor, kampus} |

Dengan keterangan tiap non-terminal sebagai berikut :

* S = awal kalimat
* Sub = subjek
* P = predikat
* O = objek
* K = keterangan
* KB = kata benda
* KG = kata ganti
* KTun = kata tunjuk
* KK = kata kerja
* KD = kata depan
* KW = keterangan waktu
* KWS = keterangan waktu spesifik
* H = kata 'hari'
* KT = keterangan tempat

Dari awal kalimat (S), terdapat beberapa pilihan pola kalimat seperti SP, SPO, SPK, SPOK. Subjek dapat terbuat dari kata benda, kata ganti, atau kombinasi kata benda dan kata ganti, contohnya seperti “kucing kakak”, atau bisa juga terbuat dari kombinasi kata benda dan kata tunjuk, seperti “kucing itu”.

Predikat disini dispesifikan hanya terdiri dari kata kerja. Objek hampir sama dengan subjek, namun tidak bisa terdiri dari gabungan kata benda dan kata tunjuk.

Keterangan terbuat dari gabungan kata depan diikuti keterangan waktu maupun keterangan tempat (salah satu, tidak bisa keduanya). Keterangan waktu sendiri terdiri dari gabungan KWS (keterangan waktu spesifik) dan H (kata ‘hari’), sehingga ketika generate kalimat, contoh hasilnya adalah seperti “... di sore hari”, bukan “... di sore”, karena terdengan kurang lengkap, sehingga ditambah kata ‘hari’ di belakang keterangan waktu (pagi, siang, sore).

Sedangkan untuk keterangan tempat hanya dapat diawali dengan kata depan, sehingga contoh hasilnya adalah “... di kantor”.

* 1. **Source Code Program**

1. **Import Library yang akan Digunakan**

|  |
| --- |
| #Import Library yang akan Digunakan  import nltk  from nltk import Tree  import re  from collections import Counter  from nltk.grammar import CFG, Nonterminal  from nltk.parse.chart import BottomUpChartParser  from nltk.parse.generate import generate, demo\_grammar |

1. **Membuat CFG (Context-Free Grammar) Bahasa Indonesia**

|  |
| --- |
| #Membuat CFG (Context-Free Grammar) Bahasa Indonesia  cfg\_grammar = """    S -> Sub P | Sub P O | Sub P K | Sub P O K    Sub -> KB | KG | KB KG | KB KTun    P -> KK    O -> KB | KG | KB KG    K -> KD KW | KD KT    KB -> 'anjing' | 'kucing' | 'pohon'  | 'bola' | 'susu' | 'emas' | 'cincin'    KG -> 'aku' | 'kamu' | 'dia' | 'ayah' | 'ibu' | 'adik' | 'kakak' | 'paman' | 'bibi' | 'kakek' | 'nenek'    KTun -> 'itu' | 'ini'    KK -> 'berenang' | 'menangis' | 'bermain' | 'memanjat' | 'meminum' | 'menjaga'    KD -> 'di' | 'dari' | 'ke' | 'pada'    KW -> KWS H    KWS -> 'siang' | 'pagi' | 'sore'    H -> 'hari'    KT -> 'warung' | 'kolam' | 'lapangan' | 'kantor' | 'kampus'  """  grammar = CFG.fromstring(cfg\_grammar)  print(grammar) |

1. **Melakukan Generate Kalimat Sebanyak 50 Kalimat**

|  |
| --- |
| #Melakukan Generate Kalimat Sebanyak 50 Kalimat  N = 50  print("Generate %d kalimat dari grammar:" % (N,))  for n, sent in enumerate(generate(grammar, n=N), 1):      print("%3d. %s" % (n, " ".join(sent))) |

1. **Menampilkan *Derivation* dari Dua Contoh Kalimat**

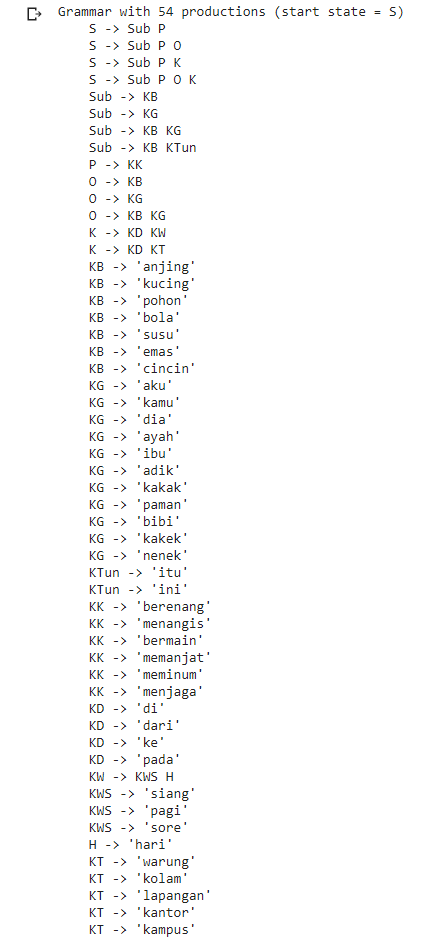
|  |
| --- |
| #Menampilkan Derivation dari Dua Contoh Kalimat  #Kalimat 1 = “anjing itu berenang di kolam”  #Kalimat 2 = “kakak bermain bola pada pagi hari”  kalimat = "anjing itu berenang di kolam"  parser = BottomUpChartParser(grammar)  kalimat\_split = kalimat.split()  kalimat\_parsed = list(parser.parse(kalimat\_split))  print(kalimat\_parsed[0].pretty\_print())  kalimat = "kakak bermain bola pada pagi hari"  parser = BottomUpChartParser(grammar)  kalimat\_split = kalimat.split()  kalimat\_parsed = list(parser.parse(kalimat\_split))  print(kalimat\_parsed[0].pretty\_print()) |

1. **Menyimpan Semua Kemungkinan Hasil Generate Kalimat Dari Grammar Yang Telah Dibuat**

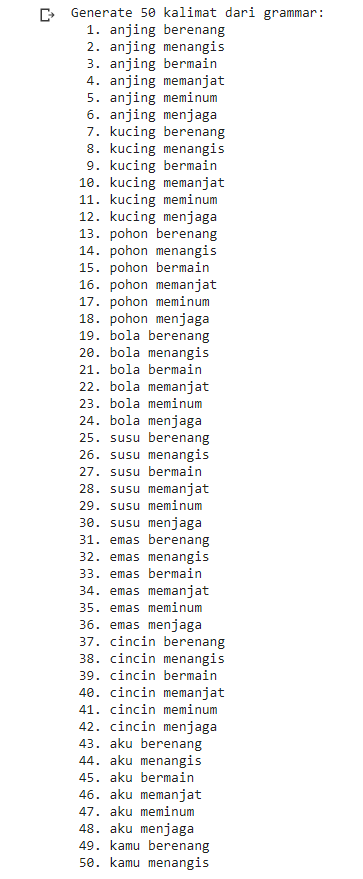
|  |
| --- |
| from google.colab import drive  drive.mount('/content/drive')  import os  os.chdir("/content/drive/")  N = 10000000  output = open('/content/drive/My Drive/Upload/Akademik/Tugas 3 PBA/output.txt','w')  for n, sent in enumerate(generate(grammar, n=N), 1):      output.write("%3d. %s\n" % (n, " ".join(sent)))  output.close() |

* 1. **Output Program**

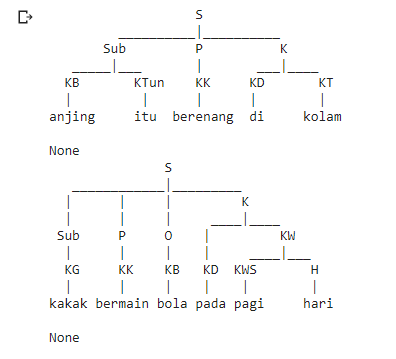
1. **Hasil Grammar Yang Dibuat**



1. **Hasil Generate 50 Kalimat**

****

1. **Hasil *Derivation* dari 2 Contoh Kalimat**

****

* 1. Analisa Hasil

Hasil grammar yang dibuat dari kumpulan rules menghasilkan **54 production**, dimana sudah termasuk production sampai ke tingkat terminal.

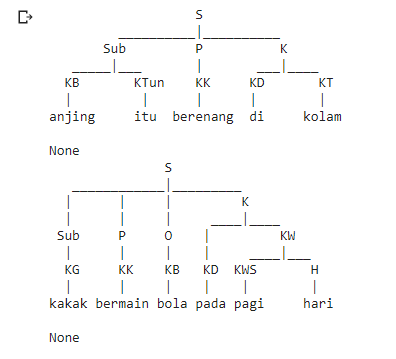
Dilakukan generate 50 kalimat, dan didapatkan 50 kalimat yang dihasilkan dari grammar yang telah didefinisikan. Dari 50 kalimat pertama yang di-generate, semua hanya terdiri dari 2 kata, karena generate dilakukan terurut mengikuti semua kemungkinan yang memungkinkan dari rules paling atas, sehingga aturan yang didapat pada semua 50 kalimat yang digenerate adalah aturan SP (Subjek Predikat).

Namun dari 50 kalimat yang digenerate, dapat kita lihat bahwa kebanyakan kalimat merupakan **kalimat yang masuk akal**, seperti "*anjing berenang*", "*anjing menjaga*", "*aku bermain*", "*kamu menangis*", dll. Namun terdapat juga beberapa **kalimat yang aneh** atau tidak masuk akal seperti "*susu menangis*", "*emas meminum*", "*cincin bermain*", dll.

Untuk bisa mendapatkan semua hasil kemungkinan kalimat yang terbentuk, dilakukan output ke dalam bentuk file. Dan dari hasil output semua kemungkinan kalimat tersebut, didapatkan **2071872 buah kalimat**.

Sama seperti hasil generate 50 kalimat di atas, kebanyakan membentuk kalimat yang masuk akal dan dapat dipahami, namun terdapat kalimat yang aneh atau tidak masuk akal. Contoh **kalimat yang dapat dipahami** adalah "*anjing itu berenang di kolam*", "*kakak bermain bola pada pagi hari*", "*anjing bibi memanjat pohon ayah di sore hari*". Sedangkan contoh **kalimat yang aneh** dan tidak masuk akal adalah "*anjing ini berenang pohon adik ke siang hari*", “*pohon aku meminum kucing ayah di warung*”.

Dari hasil kalimat "*anjing itu berenang di kolam*", "*kakak bermain bola pada pagi hari*", dilakukan derivation dan didapatkan hasil yang memenuhi rules, sebagai berikut :

****

BAB III

PENUTUP

3.1 Kesimpulan

CFG atau *Context Free Grammar* adalah tata bahasa formal yang terdiri dari G = (N, T ,P, S), dengan simbol non-terminal dan non-terminal yang dapat digunakan untuk membentuk kalimat.

Dari CFG yang dibentuk dengan referensi dari dua jurnal dan skripsi, dihasilkan total kemungkinan sebanyak 2071872 kalimat.

Kebanyakan kalimat yang terbentuk adalah kalimat yang masuk akal dan dapat dipahami seperti "*anjing itu berenang di kolam*", "*kakak bermain bola pada pagi hari*", "*anjing bibi memanjat pohon ayah di sore hari*", namun terdapat juga kalimat yang aneh dan tidak masuk akal seperti "*anjing ini berenang pohon adik ke siang hari*", “*pohon aku meminum kucing ayah di warung*”.