



JTIM : Jurnal Teknologi Informasi dan Multimedia

p-ISSN : 2715-2529

e-ISSN : 2684-9151

<https://journal.sekawan-org.id/index.php/jtim/>



Pemeriksaan Pola Kalimat Otomatis Pada Sebuah Karangan Menggunakan POS Tagging Bahasa Indonesia Dan LALR Parser

Sukmandaru Hari Wijayanto ¹, Feddy Setio Pribadi²

Sitasi: Pemeriksaan Pola Kalimat Otomatis Pada Sebuah Karangan Menggunakan POS Tagging Bahasa Indonesia Dan LALR Parser. JTIM: Jurnal Teknologi Informasi Dan Multimedia, 4(3), hlm. 149-167 <https://doi.org/10.35746/jtim.v4i3.263>

¹ Pendidikan Teknik Informatika dan Komputer, Universitas Negeri Semarang; sukmandaruhari@students.unnes.ac.id

² Pendidikan Teknik Informatika dan Komputer, Universitas Negeri Semarang; feddy.setio@mail.unnes.ac.id

* Korespondensi: sukmandaruhari@students.unnes.ac.id



Copyright: © 2022 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

Abstract: In this era of rapid technological development, language has an important role in everyday life such as to communicate with others orally or in writing. Communication will take place well if the language used can be understood so that the message can be conveyed. In written communication, writing skills are needed to produce writing that can convey messages well. One form of the result of writing skills is an essay. Essay writing must pay attention to the rules of language usage, namely phonology, morphology, and syntax. The importance of these rules, especially the syntax or structure and pattern of sentences can express ideas that can be conveyed well and are easy to understand through writing. This study aims to assist in checking sentence patterns in an essay automatically. In this examination using NLTK library for preprocessing process that is the process of break up paragraph and cleaning text, Flair NLP library for the process of part of speech tagging or labeling Indonesian word types and the use of LALR Parser table for the examination of sentence patterns and methods the method is implemented with python programming language on jupyter notebook. The sentence patterns used in this examination are subject-predicate, subject-predicate-object, subject-predicate-adverb, subject-predicate-object-adverb, subject-predicate-complement-adverb, and subject-predicate-object-complement-adverb. The results of this study is an examination of automatic sentence patterns in a simple essay with the limitation of using single sentences and active sentences. This automatic sentence pattern check can help in checking a simple essay automatically. In this examination can check correctly 335 of 423 the total number of existing sentences with a success rate of 0.792 or success (%) of 79.2%, and has an accuracy rate of 0.851 or accuracy (%) of 85.1%. Factors that influence the results of this study is the accuracy of Flair NLP library in labeling the type of word, the limitation in the variation of the tag component of the pattern, and also the limitation of sentence patterns used in this study using a single active sentence only.

Keywords: Sentence pattern; Flair NLP; LALR Parser

Abstrak: Dalam era perkembangan teknologi yang pesat ini, berbahasa mempunyai peran penting dalam kehidupan sehari-hari seperti untuk berkomunikasi dengan sesama secara lisan maupun tulisan. Komunikasi akan berlangsung dengan baik jika bahasa yang digunakan dapat dipahami sehingga pesan dapat tersampaikan. Dalam komunikasi tulisan, keterampilan menulis diperlukan untuk menghasilkan tulisan yang dapat menyampaikan pesan dengan baik. Salah satu bentuk hasil dari keterampilan menulis adalah sebuah karangan. Penulisan karangan harus memperhatikan kaidah pemakaian bahasa yaitu fonologi, morfologi, dan sintaksis. Pentingnya kaidah tersebut khususnya sintaksis atau struktur dan pola kalimat dapat mengungkapkan ide yang dapat

tersampaikan dengan baik dan mudah untuk dipahami melalui karangan. Penelitian ini bertujuan untuk membantu dalam memeriksa pola kalimat pada sebuah karangan secara otomatis. Dalam pemeriksaan ini menggunakan library nltk untuk proses preprocessing yaitu proses break up paragraph dan cleaning text, library Flair NLP untuk proses part of speech tagging atau pelabelan jenis kata bahasa Indonesia dan penggunaan tabel LALR Parser untuk pemeriksaan pola kalimat dan metode metode tersebut diimplementasikan dengan bahasa pemrograman python pada jupyter notebook. Pola kalimat yang digunakan pada pemeriksaan ini adalah Subjek-Predikat, Subjek-Predikat-Objek, Subjek-Predikat-Keterangan, Subjek-Predikat-Objek-Keterangan, Subjek-Predikat-Pelengkap-Keterangan, dan Subjek-Predikat-Objek-Pelengkap-Keterangan. Hasil dari penelitian ini adalah berupa pemeriksaan pola kalimat otomatis pada sebuah karangan sederhana dengan batasan menggunakan kalimat tunggal dan kalimat aktif. Pemeriksaan pola kalimat otomatis ini dapat membantu dalam melakukan pemeriksaan terhadap suatu karangan sederhana secara otomatis. Pada pemeriksaan ini dapat memeriksa dengan benar 335 dari 423 jumlah total kalimat yang ada dengan nilai keberhasilan 0,792 atau keberhasilan (%) sebesar 79,2%, dan memiliki tingkat akurasi sebesar 0,851 atau akurasi (%) sebesar 85,1%. Faktor yang mempengaruhi dari hasil penelitian ini adalah ketepatan library Flair NLP dalam pemberian label jenis kata, batasan dalam variasi komponen tag penyusun pola, dan juga batasan pola kalimat yang digunakan dalam penelitian ini menggunakan kalimat tunggal aktif saja.

Kata kunci: Pola Kalimat; Flair NLP; LALR Parser.

1. Pendahuluan

Dalam era perkembangan teknologi yang pesat ini, berbahasa adalah hal yang berperan penting bagi setiap orang. Terbukti dari sisi penggunaannya dalam kehidupan sehari-hari seperti penggunaan bahasa dalam berkomunikasi antar sesama. Komunikasi saat ini tidak hanya terpaku pada lisan saja, tetapi dapat juga berbentuk tulisan. Sebuah komunikasi akan berlangsung lancar jika bahasa yang digunakan dapat dipahami sehingga pesan dapat tersampaikan dan dalam hal komunikasi tulisan dibutuhkan kemampuan menulis yang baik untuk dapat membuat tulisan yang baik juga. Menulis merupakan suatu keterampilan berbahasa yang dipergunakan untuk berkomunikasi secara tidak langsung, tidak secara tatap muka dengan orang lain[1]. Aktivitas menulis merupakan suatu bentuk manifestasi kemampuan dan keterampilan berbahasa yang paling akhir dikuasai oleh pembelajaran bahasa setelah kemampuan menyimak, berbicara, dan membaca[2]. Salah satu bentuk hasil dari keterampilan menulis tersebut adalah sebuah karangan. Karangan ialah hasil wujud gagasan seseorang dalam bahasa tulis yang dapat dibaca dan dimengerti oleh pembaca[3]. Karangan adalah bahasa tulis yang merupakan rangkaian kata demi kata sehingga menjadi sebuah kalimat, paragraf, dan akhirnya menjadi sebuah wacana yang dibaca dan dipahami[4]. Dalam sebuah penulisan karangan, penulis harus mengetahui aturan atau kaidah pemakaian bahasa yang meliputi fonologi, morfologi, dan sintaksis untuk mencapai kesepakatan antar sesama pengguna bahasa.

Dalam penelitian yang dilakukan[5], yang berjudul "Ketaksaan Bahasa Dalam Karangan Siswa Kelas V SD Negeri 2 Abean Kecamatan Mirit Kabupaten Kebumen Tahun Pembelajaran 2013/2014" menghasilkan kesimpulan yang meliputi (1) ketaksaan fonologi yang terdiri atas kesalahan penulisan kata 16,31%, kesalahan tanda baca 6,46%, kesalahan penulisan huruf kapital 8,31%, kesalahan penulisan preposisi *di* dan *ke* 18,46%, dan kesalahan penyingkatan kata kata tertentu 4,55%; (2) ketaksaan morfologis terdiri atas kesalahan afiksasi 3,10%, kesalahan duplikasi 3,10%, kesalahan pemajemukan 4,55%, kesalahan diksi 3,10%, dan kesalahan pleonasm 3,10%; (3) ketaksaan sintaksis 30,17%.

Dalam penelitian yang dilakukan[6], yang berjudul "Analisis Isi Karangan Mata Pelajaran Bahasa Indonesia Pada Siswa Kelas V MIN Sumberjati Kademangan Kabupaten Blitar" menunjukkan bahwa : (1) Kompetensi siswa dalam menulis isi karangan sudah bagus, (2) Siswa memiliki hambatan dalam pengembangan dan penyusunan kata-kata

dalam menulis isi karangan, (3) Pengupayaan guru dalam meningkatkan kemampuan siswa menulis isi karangan dengan pengkajian RPP dan kegiatan pembiasaan menulis pada siswa, (4) Dari 26 siswa yang membuat karangan, 12 siswa membuat karangan narasi, 8 siswa membuat karangan deskripsi, dan 6 siswa membuat karangan bebas atau masih belum jelas jenisnya.

Dalam penelitian yang dilakukan[7], yang berjudul "Analisis Pola Kalimat Bahasa Indonesia Dalam Karangan Siswa Kelas VI SD Negeri 02 Nglegok" bertujuan untuk mengetahui dan mendeskripsikan penguasaan dan penerapan pola kalimat Indonesia dalam karangan serta perbaikan kualitas isi dalam karangan. Bentuk Penelitian ini adalah Teoritis dengan menggunakan teknik pengumpulan data metode tes dan teknik analisis data menggunakan metode deskriptif. Hasil penelitian ini dapat disimpulkan penguasaan pola kalimat bahasa Indonesia siswa sudah cukup memadai. Demikian halnya dengan kualitas isi karangan siswa sudah cukup untuk tingkat sekolah lanjutan tingkat pertama.

Dalam penelitian yang dilakukan[8], yang berjudul "Sistem Pemeriksaan Pola Kalimat Bahasa Indonesia berbasis Algoritme Left-Corner Parsing dengan Stemming" telah berhasil memeriksa pola kalimat dari sebuah kalimat sederhana. Hasil pengujian pemeriksaan pola kalimat SP, SPO, SPK, dan SPOK diperoleh tingkat keberhasilan masing-masing 96,2% dari 53 kalimat dan 162 kata, 100% dari 22 kalimat dan 78 kata, 85% dari 20 kalimat dan 85 kata, dan 100% dari sejumlah 11 kalimat dan 61 kata. Akan tetapi, perlu diperhatikan bahwa jumlah total data uji kalimat dan kata yang digunakan relatif kecil, yaitu 106 kalimat dan 188 kata.

Dalam penelitian yang dilakukan[9], yang berjudul "Pemeriksaan Struktur Kalimat Bahasa Indonesia Menggunakan Part-Of-Speech Tagging dan LALR Parser" telah melakukan pengujian terhadap kalimat tunggal yang diperoleh dari media cetak Kompas dengan jumlah total 150 kalimat. Diperoleh 130 kalimat dengan nilai persentasi 86.7 % yang berhasil diuraikan. 15 kalimat dengan nilai persentasi 10% tidak dapat diuraikan karena ada kata yang tidak dapat dikenali, dan 5 kalimat dengan persentasi 3.3% tidak dapat diuraikan karena struktur salah.

Dalam penelitian yang dilakukan[10], yang berjudul "Penerapan Algoritma LALR Parser Dan Context-Free Grammar Untuk Struktur Kalimat Bahasa Indonesia" yang diimplementasikan dalam bahasa pemrograman Java berhasil melakukan pemeriksaan pola kalimat pada sebuah kalimat. Tahapan pada sistem ini adalah perancangan Lexicon untuk penentuan jenis kata, perancangan CFG untuk membagi jenis kalimat, penerapan LALR Parser untuk pendeteksian struktur kalimat. Hasil dari sistem ini menampilkan nilai dengan tingkat akurasi diatas 70% dengan rincian antara lain akurasi kalimat tunggal sebesar 90.30%, akurasi kalimat majemuk setara sebesar 76.66%, dan akurasi kalimat majemuk bertingkat sebesar 78.57%.

Dari beberapa penelitian yang sudah dilakukan di atas, dapat dikatakan bahwa salah satu kesalahan pada karangan yang dibuat siswa terjadi di bidang sintaksis yang mempunyai beberapa aspek pembahasan, salah satunya adalah struktur dan pola kalimat. Selain itu, dari beberapa penelitian di atas belum tersedia pemeriksaan untuk sebuah karangan secara langsung atau yang terdiri dari beberapa kalimat secara langsung dapat diperiksa dan diketahui pola kalimatnya. Penguasaan struktur dan pola kalimat akan menjadi hal yang sangat penting dalam proses penulisan sebuah karangan. Seperti yang diungkapkan oleh David Wilkins yang menyatakan bahwa "tanpa tata bahasa, sangat sedikit yang bisa disampaikan, dan tanpa kosakata, tidak ada yang bisa disampaikan"[11]. Pendapat tersebut menegaskan pentingnya penguasaan struktur dan pola kalimat untuk dapat mengungkapkan ide lewat bahasa tulis, sehingga ide tersebut dapat tersampaikan dengan baik dan mudah untuk dipahami.

Berdasarkan latar belakang tersebut, maka tujuan penelitian ini untuk melakukan sebuah pemeriksaan pola kalimat otomatis pada sebuah karangan menggunakan pos tagging Bahasa Indonesia dan LALR Parser yang dapat membantu dalam melakukan pemeriksaan pola kalimat otomatis pada sebuah karangan. Hasil dari pemeriksaan ini

akan dianalisis dengan uji tingkat keberhasilan proses pemeriksaan pola kalimat dan uji tingkat keakuratan dari hasil pemeriksaan secara manual.

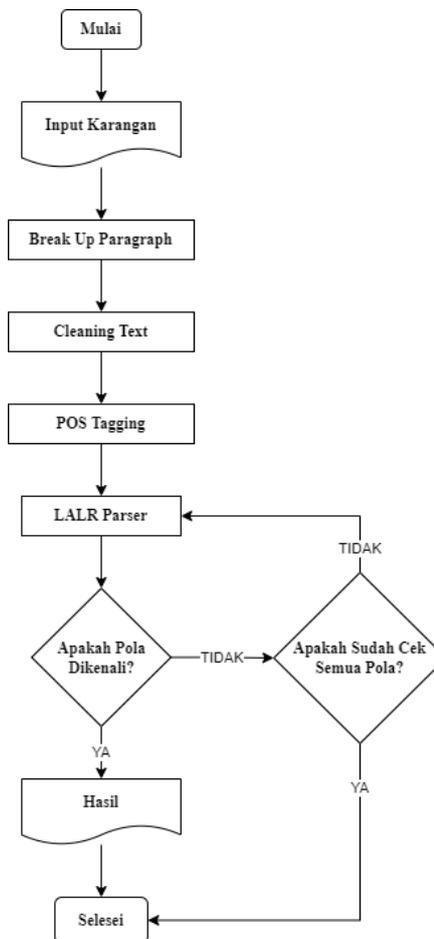
2. Dataset dan Metode

2.1. Dataset

Dalam pemeriksaan ini menggunakan dataset karangan deskripsi sederhana Bahasa Indonesia dengan batasan berisikan kalimat tunggal dan kalimat aktif. Dataset yang digunakan berupa tugas membuat karangan sederhana bahasa Indonesia siswa kelas 4 dan 5 SD Negeri 2 Sidamulih Kecamatan Rawalo Kabupaten Banyumas berjumlah 34 karangan yang terdiri dari 14 karangan siswa kelas 4 dan 20 karangan siswa kelas 5.

2.2. Metode

Implementasi dalam pemeriksaan pola kalimat otomatis pada sebuah karangan menggunakan pos tagging Bahasa Indonesia dan LALR Parser ini menggunakan bahasa pemrograman python versi 3.8.0 pada aplikasi web jupyter notebook. Diagram alir dari pemeriksaan ini dapat dilihat pada Gambar 1. di bawah ini.



Gambar 1. Diagram Alir Pemeriksaan

2.2.1. Input Karangan

Proses pertama dalam pemeriksaan ini adalah dengan menginputkan atau menambahkan data yang akan diperiksa. Data karangan tersebut berformat file text document (.txt).

2.2.2. Break Up Paragraph

Break Up Paragraph adalah proses pemisahan kalimat dari satu karangan utuh menjadi beberapa beberapa kalimat terpisah. Proses ini diawali pemisahan setiap paragraf dari satu karangan utuh dan selanjutnya pemisahan setiap kalimat dari paragraf yang sudah dipisahkan.

2.2.3. Cleaning Text

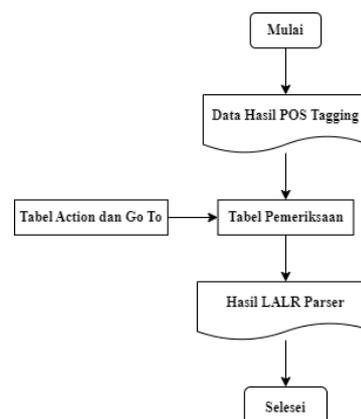
Cleaning text adalah proses penghilangan tanda baca yang ada pada text. Data hasil dari proses break up paragraph akan diproses untuk penghilangan semua tanda baca yang ada.

2.2.4. POS Tagging Menggunakan Flair NLP

POS Tagging adalah suatu proses memberikan label kata secara otomatis pada suatu kata dalam kalimat[12]. Masing-masing kata akan diperiksa dengan korpus library yang digunakan. Pada pemeriksaan ini, library yang digunakan untuk proses pos tagging adalah Flair Natural Language Processing (NLP). Flair NLP ini mempunyai korpus untuk kata berbahasa Inggris dan Indonesia. Label tersebut akan dituliskan disebelah kanan dari kata yang diperiksa.

2.2.5. Pemeriksaan Pola Kalimat Menggunakan LALR Parser

LR Parser merupakan parser yang bekerja dengan membaca input dari kiri ke kanan. LALR Parser merupakan salah satu varian dari LR Parser yang mampu menangani grammar[13]. LALR Parser atau Look-ahead LR Parser adalah versi sederhana dari Canonical LR Parser, dan dalam prosesnya, LALR Parser tidak melakukan backtracking (pengulangan pencarian)[14]. Pada proses ini adalah proses pengecekan data hasil pos tagging yang berisikan kumpulan label jenis kata kedalam variasi jenis kata pola kalimat yang tersedia. Setiap kata yang sudah melalui tahap POS Tagging akan berubah yang awalnya terdiri dari kata seperti yang di inputkan akan berubah menjadi label dari setiap jenis katanya. Label label tersebut yang nantinya akan diperiksa melalui proses LALR Parser. Dalam proses LALR Parser ini adalah dengan memeiksa inputan yang berupa label jenis kata dalam satu kalimat menggunakan action dan go to table dari setiap grammar yang tersedia. Action table akan memberikan aksi berupa shift dan reduce pada setiap inputan. Kemudian go to table akan menentukan reduce inputan tersebut sebagai subjek, predikat, objek, pelengkap, keterangan. Setelah itu, akan diperoleh output atau hasil parser yang berupa pola kalimat dari inputan. Setiap inputan akan diperiksa dengan parser grammar yang tersedia. Jika inputan tidak menemukan hasil atau menemui eror pada parser grammar pertama, maka akan diperiksa dengan menggunakan parser grammar kedua dan begitu seterusnya sampai dengan parser grammar terakhir. Proses LALR Parser ini dapat dilihat seperti pada gambar 2. di bawah ini.



Gambar 2. Diagram Alir LALR Parser

Pemeriksaan ini menggunakan 6 pola kalimat antara lain S-P, S-P-O, S-P-K, S-P-O-K, S-P-Pel-K, dan S-P-O-Pel-K. Masing-masing pola kalimat ini memiliki variasi komponen penyusun yang berbeda. Komponen penyusun pola kalimat ini berisikan kumpulan jenis kata sebagai berikut:

- Subjek (S)

Tabel 1. Komponen Tag Subjek

| 4 Tag | 3 Tag | 2 Tag | 1 Tag |
|--------------------|----------------|-----------|-------|
| NOUN NOUN NOUN DET | NOUN NOUN DET | DET PROP | PRON |
| NOUN NUM NOUN DET | NOUN NOUN NOUN | NOUN ADJ | NOUN |
| | NOUN NUM PRON | NOUN DET | PROP |
| | NUM NOUN DET | NOUN NOUN | |
| | NUM NOUN NOUN | NOUN NUM | |
| | | NOUN PRON | |
| | | NOUN PROP | |
| | | NUM NOUN | |
| | | PRON DET | |
| | | PRON NUM | |

- Predikat (P)

Tabel 2. Komponen Tag Predikat

| 3 Tag | 2 Tag | 1 Tag |
|----------------|-----------|-------|
| ADV PART VERB | ADV VERB | VERB |
| ADV VERB VERB | PART VERB | |
| PART ADV VERB | VERB VERB | |
| PART VERB VERB | | |
| VERB VERB VERB | | |

- Objek (O)

Tabel 3. Komponen Tag Objek

| 4 Tag | 3 Tag | 2 Tag | 1 Tag |
|--------------------|----------------|-----------|-------|
| NOUN NOUN NOUN DET | NOUN NOUN DET | DET NOUN | NOUN |
| | NOUN NOUN NOUN | NOUN ADJ | PRON |
| | NUM NOUN DET | NOUN DET | PROP |
| | NUM NOUN NOUN | NOUN NOUN | |
| | | NOUN PRON | |
| | | NOUN PROP | |
| | | NUM NOUN | |
| | | PRON DET | |

- Pelengkap (Pel)

Tabel 4. Komponen Tag Pelengkap

| 3 Tag | 2 Tag | 1 Tag |
|-----------------|------------|-------|
| CCONJ NOUN NOUN | ADV ADJ | NUM |
| SCONJ ADV ADJ | CCONJ ADJ | |
| SCONJ NOUN ADJ | CCONJ NOUN | |
| SCONJ NOUN NOUN | CCONJ VERB | |
| SCONJ PART NOUN | NUM NOUN | |
| SCONJ VERB NOUN | SCONJ ADJ | |
| | SCONJ ADV | |
| | SCONJ NOUN | |
| | SCONJ NUM | |
| | SCONJ VERB | |

- Keterangan (K)

Tabel 5. Komponen Tag Keterangan

| 4 Tag | 3 Tag | 2 Tag |
|-------------------|--------------|----------|
| ADP NOUN ADP ADJ | ADP ADJ NOUN | ADP ADJ |
| ADP NOUN ADP PROP | ADP DET NOUN | ADP NOUN |

| 4 Tag | 3 Tag | 2 Tag |
|---------------------|----------------|------------|
| ADP NOUN ADV NOUN | ADP NOUN ADJ | ADP NUM |
| ADP NOUN NOUN DET | ADP NOUN ADV | ADP PRON |
| ADP NOUN NOUN NOUN | ADP NOUN DET | ADP PROPN |
| ADP NOUN NUM PROPN | ADP NOUN NOUN | CCONJ ADJ |
| ADP NOUN PROPN VERB | ADP NOUN PROPN | CCONJ VERB |
| ADP NUM NOUN ADJ | | SCONJ ADJ |
| ADP NUM NUM NOUN | | SCONJ VERB |
| ADP PROPN ADP NOUN | | |
| ADP VERB ADP ADJ | | |

Keterangan dari variasi komponen penyusun pola dapat dilihat pada tabel 6. di bawah ini.

Tabel 6. Keterangan Komponen Penyusun

| No | Label POS | Nama POS | Contoh |
|----|-----------|---------------------------|--------------------|
| 1 | ADJ | Adjective | Baik, Jahat |
| 2 | ADP | Adposition | Di, Ke, Dari |
| 3 | ADV | Adverb | Bisa, Akan, Nanti |
| 4 | CCONJ | Coordinating Conjunction | Dan, Atau, Tetapi |
| 5 | DET | Determiner | Para, Ini, Itu |
| 6 | INTJ | Interjection | Wah, Aduh, Oi |
| 7 | NOUN | Noun | Buku, Komputer |
| 8 | NUM | Numeral | Satu, Dua, Pertama |
| 9 | PART | Particle | Bukan, Tidak |
| 10 | PRON | Pronoun | Saya, Kamu, Engkau |
| 11 | PROP | Proper Noun | Indonesia, Medan |
| 12 | PUNCT | Punctuation | Pun, Kah, |
| 13 | SCONJ | Subordinating Conjunction | Jika, Ketika |
| 14 | VERB | Verb | Pergi, Belajar |

2.2.6. Penyimpanan Hasil Pemeriksaan

Pada proses ini, proses penyimpanan dari hasil pemeriksaan akan disimpan pada file awal yang diinputkan. Hasil pemeriksaan akan tertulis pada file berada di bawah karangan awal.

2.3. Pengujian Hasil

Analisis pengujian hasil dilakukan dengan dua cara yaitu menghitung tingkat keberhasilan pemeriksaan ini untuk dapat memeriksa pola kalimat pada sebuah karangan dan menghitung tingkat keakuratan dengan membandingkan hasil pemeriksaan otomatis dengan hasil pemeriksaan secara manual.

Pengujian pertama dilakukan dengan menguji tingkat keberhasilan dalam memeriksa pola kalimat pada sebuah karangan dengan menghitung jumlah kalimat yang diperiksa dan menghasilkan output sebuah pola kalimat dengan jumlah keseluruhan kalimat yang diperiksa. Keberhasilan dalam pemeriksaan ini dapat dilihat dari suatu kalimat memiliki output sebuah pola kalimat tertentu sesuai dengan pola yang tersedia.

$$\text{Tingkat Keberhasilan} = \frac{\sum \text{Jumlah Kalimat Berpola}}{\sum \text{Jumlah Kalimat Keseluruhan}} \quad (1)$$

$$\text{Keberhasilan (\%)} = \text{Tingkat Keberhasilan} \times 100\% \quad (2)$$

Pengujian kedua dilakukan dengan menggunakan perhitungan akurasi pada Confusion Matrix. Confusion Matrix merupakan matrix yang memberikan informasi perbandingan hasil pemeriksaan (prediksi) dengan hasil yang sesungguhnya (aktual). Untuk menghitung akurasi pada confusion matriks menggunakan persamaan (3)[15].

$$\text{Tingkat Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3)$$

$$\text{Akurasi (\%)} = \text{Tingkat Akurasi} \times 100\% \quad (4)$$

Keterangan :

1. TP (True Positif) adalah istilah untuk kondisi dimana hasil prediksi sama dengan hasil aktual yaitu sama benar atau hasil dari sistem sama dengan hasil dari pengujian sebenarnya.
2. FP (False Positive) adalah istilah untuk kondisi dimana hasil prediksi tidak sama dengan hasil aktual yaitu hasil prediksi benar dan hasil aktual salah atau hasil dari sistem benar namun hasil dari pengujian salah.
3. FN (False Negative) adalah istilah untuk kondisi dimana hasil prediksi tidak sama dengan hasil aktual yaitu hasil prediksi salah dan hasil aktual benar atau hasil dari sistem salah namun hasil dari pengujian benar.
4. TN (True Negative) adalah istilah untuk kondisi dimana hasil prediksi sama dengan hasil aktual yaitu sama salah atau hasil dari sistem sama dengan hasil dari pengujian sebenarnya.

3. Hasil

3.1. Hasil Pemeriksaan Pola Kalimat

3.1.1. Input Karangan

Proses input karangan menggunakan fungsi open(). Implementasi input karangan dapat dilihat seperti pada Gambar 3. di bawah ini.

```
1 text_file = open("karangan.txt", "r")
2 text = text_file.read()
3 print(text)
4 text_file.close()
```

Aku mempunyai seekor kucing. Kucing itu bernama Kitty. Kitty adalah kucing persia. Kitty memiliki bulu sangat lembut ketika disentuh. Ia memiliki ekor yang sangat panjang dan halus. Kitty adalah kucing yang lincah. Kitty senang bermain. Aku melempar bola ke Kitty. Ia berlari mengejar bola itu. Ia senang menjilat kakinya. Aku rajin memberi makan Kitty. Ia senang makan makanan kaleng. Aku juga menjaga kebersihan kandangnya. Kitty tidak suka tempat kotor. Ia tidak pernah buang kotoran sembarangan. Aku sayang Kitty.

Gambar 3. Implementasi Input Karangan

3.1.2. Break Up Paragraph

Proses break up paragraph ini dibagi menjadi 2 bagian. Bagian pertama adalah pemisahan paragraph dari suatu karangan. Pemisahan tersebut menggunakan fungsi split(). Bagian kedua dari proses break up paragraph adalah pemisahan setiap kalimat dari paragraf hasil dari bagian pertama. Pemisahan ini menggunakan sent_tokenize pada library Natural Language Toolkit (NLTK). Implementasi break up paragraph dapat dilihat seperti pada Gambar 4. di bawah ini.

```
1 import string
2 from nltk.tokenize import sent_tokenize
3
4 #Pemisahan Setiap Paragraf
5 paragraf = text.split('\n')
6 #Pemisahan Setiap Kalimat
7 kalimat = [sent_tokenize(paragraf) for paragraf in paragraf]
8 print(kalimat)
```

```
[['Aku mempunyai seekor kucing.', 'Kucing itu bernama Kitty.', 'Kitty adalah kucing persia.', 'Kitty memiliki bulu sangat lembut ketika disentuh.', 'Ia memiliki ekor yang sangat panjang dan halus.'], ['Kitty adalah kucing yang lincah.', 'Kitty senang bermain.', 'Aku melempar bola ke Kitty.', 'Ia berlari mengejar bola itu.', 'Ia senang menjilat kakinya.', 'Aku rajin memberi makan Kitty.', 'Ia senang makan makanan kaleng.', 'Aku juga menjaga kebersihan kandangnya.', 'Kitty tidak suka tempat kotor.', 'Ia tidak pernah buang kotoran sembarangan.', 'Aku sayang Kitty.']]
```

Gambar 4. Implementasi Break Up Paragraph

3.1.3. Cleaning Text

Proses cleaning text menggunakan library Natural Language Toolkit (NLTK). Implementasi cleaning text dapat dilihat seperti pada Gambar 5. di bawah ini.

```

1 import copy
2 from nltk.tokenize import word_tokenize
3 kalimat_no_punct = copy.deepcopy(kalimat)
4 for i in range(len(kalimat)):
5     for j in range(len(kalimat[i])):
6         kalimat_no_punct [i][j]=(kalimat[i][j].translate(str.maketrans('', '', string.punctuation)))
7 print(kalimat_no_punct)

```

[['Aku mempunyai seekor kucing', 'Kucing itu bernama Kitty', 'Kitty adalah kucing persia', 'Kitty memiliki bulu sangat lembut ketika disentuh', 'Ia memiliki ekor yang sangat panjang dan halus', ['Kitty adalah kucing yang lincah', 'Kitty senang bermain', 'Aku melempar bola ke kitty', 'Ia berlari mengejar bola itu', 'Ia senang menjilati kakinya', 'Aku rajin memberi makan Kitty', 'Ia senang makan makanan kaleng', 'Aku juga menjaga kebersihan kandangnya', 'Kitty tidak suka tempat kotor', 'Ia tidak pernah buang kotoran sembarangan', 'Aku sayang Kitty']]

Gambar 5. Implementasi Cleaning Text

3.1.4. POS Tagging Menggunakan Flair NLP

Proses pos tagging ini menggunakan library Flair NLP. Implementasi pos tagging dapat dilihat seperti pada Gambar 6. di bawah ini.

```

1 from flair.data import Sentence
2 import copy
3
4 sentence = copy.deepcopy(kalimat_no_punct)
5 for i in range(len(kalimat_no_punct)):
6     for j in range(len(kalimat_no_punct[i])):
7         sentence[i][j] = Sentence(kalimat_no_punct[i][j])
8         tag_pos = SequenceTagger.load('resources/taggers/example-universal-pos/best-model.pt')
9         tag_pos.predict(sentence[i][j])
10
11 hasil_flair = copy.deepcopy(kalimat_no_punct)
12 for i in range(len(sentence)):
13     for j in range(len(sentence[i])):
14         temp = (sentence[i][j].to_tagged_string().split())
15         hasil_flair[i][j] = " ".join(temp)
16 print(hasil_flair)

```

2022-09-21 01:36:23,472 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 01:39:13,990 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 01:45:18,450 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 01:51:02,091 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 01:52:01,775 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 02:20:35,710 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 02:31:54,502 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 02:32:53,002 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 02:49:41,471 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 02:53:29,867 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 03:06:23,576 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 03:14:58,342 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 03:19:20,640 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 03:22:12,975 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 03:24:29,566 loading file resources/taggers/example-universal-pos/best-model.pt
2022-09-21 03:35:41,636 loading file resources/taggers/example-universal-pos/best-model.pt
[['Aku <PRON> mempunyai <VERB> seekor <NOUN> kucing <NOUN>', 'Kucing <NOUN> itu <DET> bernama <VERB> Kitty <PROPN>', 'Kitty <PROPN> adalah <AUX> kucing <NOUN> persia <NOUN>', 'Kitty <PROPN> memiliki <VERB> bulu <NOUN> sangat <ADV> lembut <ADJ> ketika <SCONJ> disentuh <VERB>', 'Ia <PRON> memiliki <VERB> ekor <NOUN> yang <PRON> sangat <ADV> panjang <ADJ> dan <CCONJ> halus <ADJ>'], ['Kitty <PROPN> adalah <AUX> kucing <NOUN> yang <PRON> lincah <VERB>', 'Kitty <PROPN> senang <ADV> bermain <VERB>', 'Aku <PRON> melempar <VERB> bola <NOUN> ke <ADP> kitty <NOUN>', 'Ia <PRON> berlari <VERB> mengejar <VERB> bola <NOUN> itu <DET>', 'Ia <PRON> senang <VERB> menjilati <VERB> kakinya <NOUN>', 'Aku <PRON> rajin <VERB> memberi <VERB> makan <NOUN> Kitty <PROPN>', 'Ia <PRON> senang <VERB> makan <NOUN> makanan <NOUN> kaleng <NOUN>', 'Aku <PRON> juga <ADV> menjaga <VERB> kebersihan <NOUN> kandangnya <NOUN>', 'Kitty <PROPN> tidak <PART> suka <VERB> tempat <NOUN> kotor <ADJ>', 'Ia <PRON> tidak <PART> pernah <ADV> buang <VERB> kotoran <NOUN> sembarangan <NOUN>', 'Aku <PRON> sayang <VERB> Kitty <NOUN>']]

Gambar 6. Implementasi POS Tagging Menggunakan Flair NLP

Setelah setiap kata mendapat jenis kata, kata asli akan dihilangkan dan diganti dengan label jenis katanya saja. Implementasi penghilangan kata asli dapat dilihat seperti pada Gambar 7. di bawah ini.

```

1 hasil_spok = copy.deepcopy(kalimat_no_punct)
2 for i in range(len(hasil_flair)):
3     for j in range(len(hasil_flair[i])):
4         temp = (hasil_flair[i][j].split())
5         n = len(temp)
6         for k in range(n):
7             if((k%2) != 0):
8                 temp.pop(n-1-k)
9         hasil_spok[i][j] = " ".join(temp)
10 print(hasil_spok)

```

```

[['<PRON> <VERB> <NOUN> <NOUN>', '<NOUN> <DET> <VERB> <PROPN>', '<PROPN> <AUX> <NOUN> <NOUN>', '<PROPN> <VERB> <NOUN> <ADV> <ADJ> <SCONJ> <VERB>', '<PRON> <VERB> <NOUN> <PRON> <ADV> <ADJ> <CCONJ> <ADJ>'], ['<PROPN> <AUX> <NOUN> <PRON> <VERB>', '<PROPN> <ADV> <VERB>', '<PRON> <VERB> <NOUN> <ADP> <NOUN>'], '<PRON> <VERB> <VERB> <NOUN> <DET>', '<PRON> <VERB> <VERB> <NOUN>', '<PRON> <VERB> <VERB> <NOUN> <PROPN>', '<PRON> <VERB> <NOUN> <NOUN> <NOUN>', '<PRON> <ADV> <VERB> <NOUN> <NOUN>', '<PROPN> <PART> <VERB> <NOUN> <ADJ>', '<PRON> <PART> <ADV> <VERB> <NOUN> <NOUN>', '<PRON> <VERB> <NOUN>']]]

```

Gambar 7. Implementasi Penghilangan Kata Asli

3.1.5. Pemeriksaan Pola Kalimat Menggunakan LALR Parser

Sebelum melakukan pemeriksaan pola kalimat, diperlukan tabel action dan go to untuk digunakan dalam pengecekan sebuah kalimat. Tabel action dan go to digunakan untuk mencari hasil perbandingan yang dilakukan pada proses pemeriksaan. Pada tabel action dan go to berisikan kombinasi huruf (S atau R) dan angka, hanya angka saja dan juga ACC. Kombinasi S diikuti angka memberikan aksi Shift atau melanjutkan pemeriksaan pada variasi tersebut. Kombinasi R diikuti angka memberikan aksi Replace terhadap variasi yang sudah diperiksa dengan data replace yang sesuai dengan angka yang dimunculkan. Isian angka saja memberikan aksi menambahkan angka tersebut ke dalam kolom parsing stack. Dan untuk isian ACC memberikan aksi proses selesai dengan input yang diperiksa termasuk variasi dari pola kalimat yang sedang memeriksa. Sedangkan isian kosong menandakan input yang sedang di periksa tidak termasuk dalam variasi yang sedang memeriksa dan akan dilakukan looping untuk pemeriksaan variasi dengan pola kalimat selanjutnya. Jika pemeriksaan tetap tidak ditemukan pada pola kalimat terakhir maka proses akan berhenti dan memunculkan kalimat "Kalimat tidak sesuai dengan variasi pola yang ada". Tabel 7. di bawah ini merupakan contoh tabel action dan go to dari pola kalimat "S-P".

Tabel 7. Action dan Go To "S-P"

| i | ACTION | | | | | | | | | GOTO | | | |
|----|--------|------|-------|-----|-----|-----|------|-----|-------|------|---|---|---|
| | NOUN | PRON | PROPN | DET | NUM | ADJ | VERB | ADV | PAR T | \$ | G | S | P |
| 0 | S3 | S4 | S5 | S6 | S7 | | | | | | | 1 | 2 |
| 1 | | | | | | | | | | ACC | | | |
| 2 | | | | | | | S9 | S10 | S11 | | | | 8 |
| 3 | S14 | S16 | S17 | S13 | S15 | S12 | R2 | R2 | R2 | | | | |
| 4 | | | | S18 | S19 | | R3 | R3 | R3 | | | | |
| 5 | | | | | | | R4 | R4 | R4 | | | | |
| 6 | S20 | | | | | | | | | | | | |
| 7 | S21 | | | | | | | | | | | | |
| 8 | | | | | | | | | | R1 | | | |
| 9 | | | | | | | S22 | | | R22 | | | |
| 10 | | | | | | | S23 | | S24 | | | | |
| 11 | | | | | | | S25 | S26 | | | | | |
| 12 | | | | | | | R6 | R6 | R6 | | | | |
| 13 | | | | | | | R7 | R7 | R7 | | | | |
| 14 | S28 | | | S27 | | | R8 | R8 | R8 | | | | |
| 15 | S30 | S29 | | | | | R9 | R9 | R9 | | | | |
| 16 | | | | | | | R10 | R10 | R10 | | | | |
| 17 | | | | | | | R11 | R11 | R11 | | | | |

| i | ACTION | | | | | | | | | | GO TO | | | |
|----|--------|------|-------|-----|-----|-----|------|-----|----------|----|-------|---|---|--|
| | NOUN | PRON | PROPN | DET | NUM | ADJ | VERB | ADV | PAR T | \$ | G | S | P | |
| 18 | | | | | | | R13 | R13 | R13 | | | | | |
| 19 | | | | | | | R14 | R14 | R14 | | | | | |
| 20 | | | | | | | R5 | R5 | R5 | | | | | |
| 21 | S32 | | | S31 | | | R12 | R12 | R12 | | | | | |
| 22 | | | | | | | S33 | | | | R25 | | | |
| 23 | | | | | | | S34 | | | | R23 | | | |
| 24 | | | | | | | S35 | | | | | | | |
| 25 | | | | | | | S36 | | | | R24 | | | |
| 26 | | | | | | | S37 | | | | | | | |
| 27 | | | | | | | R15 | R15 | R15 | | | | | |
| 28 | | | | S38 | | | R16 | R16 | R16 | | | | | |
| 29 | | | | | | | R17 | R17 | R17 | | | | | |
| 30 | | | | S39 | | | | | | | | | | |
| 31 | | | | | | | R18 | R18 | R18 | | | | | |
| 32 | | | | | | | R19 | R19 | R19 | | | | | |
| 33 | | | | | | | | | | | R30 | | | |
| 34 | | | | | | | | | | | R27 | | | |
| 35 | | | | | | | | | | | R26 | | | |
| 36 | | | | | | | | | | | R29 | | | |
| 37 | | | | | | | | | | | R28 | | | |
| 38 | | | | | | | R20 | R20 | R20 | | | | | |
| 39 | | | | | | | R21 | R21 | R21 | | | | | |

Untuk implementasi persiapan tabel dalam proses pemeriksaan dapat dilihat seperti pada Gambar 8. di bawah ini.

```

1 import pandas as pd
2 pola = ['SP', 'SPO', 'SPK', 'SPOK', 'SPPELK', 'SPOPELK']
3 hasil_akhir = copy.deepcopy(hasil_spok)
4 #Mempersiapkan Tabel Action dan Go To
5 for z in range(len(pola)):
6     db_pola = pd.read_excel('data.xlsx', pola[z], engine='openpyxl')
7     db_pola = db_pola.iloc[:, 0:len(db_pola.iloc[0,:].values)].values #[baris][kolom]
8     for i in range(len(db_pola)):
9         temp = len(db_pola[i])-1
10        for j in range(len(db_pola[i])):
11            if (isinstance(db_pola[i][temp-j], float)):
12                db_pola[i][temp-j] = 'error'
13        #Mempersiapkan Data Replace
14        data_replace = pd.read_excel('data_replace.xlsx', pola[z], engine='openpyxl')
15        data_replace = data_replace.iloc[:, 0:len(data_replace.iloc[0,:].values)].values
16        data_replace = data_replace.tolist()
17        for i in range(len(data_replace)):
18            temp = len(data_replace[i])-1
19            for j in range(len(data_replace[i])):
20                if (isinstance(data_replace[i][temp-j], float)):
21                    data_replace[i].pop(temp-j)
22        #Mempersiapkan Tabel Pemeriksaan
23        for x in range(len(hasil_spok)):
24            for y in range(len(hasil_spok[x])):
25                kalimat = hasil_spok[x][y]
26                kalimat = kalimat.replace("<", "")
27                kalimat = kalimat.replace(">", "")
28                var_input = kalimat.split()
29                var_parsingstack = ['$ ', 0]
30                var_input.append('$ ')
31                var_action = ''

```

Gambar 8. Implementasi Persiapan Tabel

Untuk proses pada tabel pemeriksaan akan membandingkan antara kolom parsing stack yang berisi state dari tabel action dan go to dan kolom input yang berisi data hasil dari proses pos tagging menggunakan Flair NLP. Pemeriksaan pola kalimat dilakukan dengan membandingkan isi paling kanan pada kolom parsing stack yang berupa angka dengan isi paling kiri pada kolom input berupa komponen pola kalimat. Ada pula kondisi dimana isi paling kanan pada kolom parsing stack adalah bukan berupa angka, maka dilakukan perbandingan antara angka yang muncul terakhir pada kolom parsing stack

dengan komponen pola kalimat paling kanan pada kolom parsing stack. Proses pemeriksaan dapat dilihat pada Tabel 8. di bawah ini.

Tabel 8. Pemeriksaan Pola Kalimat

| PARSING STACK | INPUT | ACTION |
|-------------------------|-------------------|--------|
| \$ 0 | PROPN ADV VERB \$ | S5 |
| \$ 0 PROPON 5 | ADV VERB \$ | R4 |
| \$ 0 S | ADV VERB \$ | 2 |
| \$ 0 S 2 | ADV VERB \$ | S10 |
| \$ 0 S 2 ADV 10 | VERB \$ | S23 |
| \$ 0 S 2 ADV 10 VERB 23 | | \$ R23 |
| \$ 0 S 2 P | | \$ 8 |
| \$ 0 S 2 P 8 | | \$ R1 |
| \$ 0 G | | \$ 1 |
| \$ 0 G 1 | | \$ ACC |

Proses Pemeriksaan diatas adalah memeriksa kalimat "Kitty senang bermain" pada pemeriksaan pola kalimat "S-P".

Pada baris ke-1 adalah membandingkan isi paling kanan pada kolom parsing stack yaitu 0 dengan isi paling kiri kolom input yaitu PROPON. Perbandingan tersebut menghasilkan action S5. Hasil perbandingan S5 memberikan aksi memindahkan PROPON ke dalam tabel parsing stack diikuti angka yang muncul pada kolom action yaitu 5 untuk baris selanjutnya pada kolom parsing stack.

Pada baris ke-2, membandingkan isi paling kanan pada parsing stack yaitu 5 dengan isi paling kiri pada kolom input yaitu ADV. Perbandingan ini menghasilkan R4 pada kolom action. Hasil perbandingan R4 memberikan aksi mengganti PROPON 5 menjadi S untuk baris selanjutnya pada kolom parsing stack.

Pada baris ke-3, isi paling kanan pada parsing stack tidak berupa angka maka akan dilakukan perbandingan antara angka yang muncul terakhir pada kolom parsing stack yaitu 0 dengan komponen pola kalimat paling kanan pada kolom parsing stack yaitu S. Perbandingan ini menghasilkan angka 2 pada kolom action. Hasil Perbandingan angka 2 memberikan aksi memindahkan angka 2 tersebut ke dalam kolom parsing stack dengan posisi paling kanan untuk baris selanjutnya pada kolom parsing stack.

Pada baris ke-4, membandingkan isi paling kanan pada parsing stack yaitu 2 dengan isi paling kiri pada kolom input yaitu ADV. Perbandingan ini menghasilkan S10 pada kolom action. Hasil perbandingan S10 memberikan aksi memindahkan ADV ke dalam tabel parsing stack diikuti angka yang muncul pada kolom action yaitu 10 untuk baris selanjutnya pada parsing stack.

Pada baris ke-5, membandingkan isi paling kanan pada parsing stack yaitu 10 dengan isi paling kiri pada kolom input yaitu VERB. Perbandingan ini menghasilkan S23 pada kolom action. Hasil perbandingan S10 memberikan aksi memindahkan VERB ke dalam tabel parsing stack diikuti angka yang muncul pada kolom action yaitu 23 untuk baris selanjutnya pada parsing stack.

Pada baris ke-6, membandingkan isi paling kanan pada parsing stack yaitu 23 dengan isi paling kiri pada kolom input yaitu simbol \$. Perbandingan ini menghasilkan R23 pada kolom action. Hasil perbandingan R23 memberikan aksi mengganti ADV 10 VERB 23 menjadi P untuk baris selanjutnya pada kolom parsing stack.

Pada baris ke-7, isi paling kanan pada parsing stack tidak berupa angka maka akan dilakukan perbandingan antara angka yang muncul terakhir pada kolom parsing stack yaitu 2 dengan komponen pola kalimat paling kanan pada kolom parsing stack yaitu P. Perbandingan ini menghasilkan angka 8 pada kolom action. Hasil Perbandingan angka 8 memberikan aksi memindahkan angka 8 tersebut ke dalam kolom parsing stack dengan posisi paling kanan untuk baris selanjutnya pada kolom parsing stack

Pada baris ke-8, membandingkan isi paling kanan pada parsing stack yaitu 8 dengan isi paling kiri pada kolom input yaitu simbol \$. Perbandingan ini menghasilkan R1 pada

kolom action. Hasil perbandingan R1 memberikan aksi mengganti S 2 P 8 menjadi G untuk baris selanjutnya pada kolom parsing stack.

Pada baris ke-9, isi paling kanan pada parsing stack tidak berupa angka maka akan dilakukan perbandingan antara angka yang muncul terakhir pada kolom parsing stack yaitu 0 dengan komponen pola kalimat paling kanan pada kolom parsing stack yaitu G. Perbandingan ini menghasilkan angka 1 pada kolom action. Hasil Perbandingan angka 1 memberikan aksi memindahkan angka 1 tersebut ke dalam kolom parsing stack dengan posisi paling kanan untuk baris selanjutnya pada kolom parsing stack

Pada baris ke-10, membandingkan isi paling kanan pada parsing stack yaitu 1 dengan isi paling kiri pada kolom input yaitu simbol \$. Perbandingan ini menghasilkan ACC pada kolom action. Hasil perbandingan ACC memberikan aksi bahwa kalimat yang diperiksa yaitu "Kitty senang bermain" merupakan bagian dari variasi komponen dari pola kalimat yaitu pola kalimat "S-P" dan proses pemeriksaan berakhir.

Untuk implementasi proses pemeriksaan pola kalimat dapat dilihat seperti pada Gambar 9. di bawah ini.

```

33 if (z==0):
34     hasil_akhir[x][y] = ''
35     looping = 0
36     batas = 0
37     while var_action != 'acc' and var_action != 'error':
38         if (z > 0 and hasil_akhir[x][y] != 'error'):
39             if (looping == 0):
40                 break
41             looping = looping + 1
42         if (z > 0 and hasil_akhir[x][y] == 'error'):
43             hasil_akhir[x][y] = ''
44         sentences = sentence[x][y].to_tagged_string().split()
45         for a in range(int(len(sentences)/2)):
46             sentences[a] = sentences[a]+' '+sentences[a+1]
47             sentences.pop(a+1)
48         #Melakukan Pemeriksaan
49         if (isinstance(var_parsingstack[-1], int)):
50             for i in range(1,len(db_pola[0])):
51                 if (db_pola[0][i] == var_input[0]):
52                     var_action = db_pola[var_parsingstack[-1]+1][i]
53                     if (var_action == 'acc' or var_action == 'error'):
54                         break
55                     act = ''.join(filter(str.isalpha, var_action))
56                     num = int(''.join(filter(str.isdigit, var_action)))
57                     if (act == 'S'):
58                         var_parsingstack.append(var_input[0])
59                         var_parsingstack.append(num)
60                         var_input.pop(0)
61                     elif (act == 'R'):
62                         if (data_replace[num-1][1] != 'G'):
63                             hasil_akhir[x][y] = hasil_akhir[x][y] + ' ' + data_replace[num-1][1] + ' : '
64                             for a in range((len(data_replace[num-1])-3)):
65                                 hasil_akhir[x][y] = hasil_akhir[x][y] + ' ' + sentences[batas]
66                                 batas = batas+1
67                             hasil_akhir[x][y] = hasil_akhir[x][y].replace(' S ', 'S ')
68                             for p in range((len(data_replace[num-1])-3)*2):
69                                 var_parsingstack.pop(-1)
70                                 var_parsingstack.append(data_replace[num-1][1])
71                             break
72                         elif ((i == len(db_pola[0])-1) and (db_pola[0][i] != var_input[0])):
73                             var_action = 'error'
74                     else:
75                         for i in range(1,len(db_pola[0])):
76                             if (db_pola[0][i] == var_parsingstack[-1]):
77                                 var_action = db_pola[((var_parsingstack[-2])+1)][i]
78                                 var_parsingstack.append(var_action)
79                                 break
80
81                 if (var_action == 'error'):
82                     hasil_akhir[x][y] = 'error'
83             if (var_action == 'error' and z == len(pola)-1):
84                 hasil_akhir[x][y] = 'Kalimat tidak sesuai dengan variasi pola yang ada'

```

1 print(hasil_akhir)

[['S : Aku <PRON> P : mempunyai <VERB> O : seekor <NOUN> kucing <NOUN>', 'S : Kucing <NOUN> itu <DET> P : bernama <VERB> O : Kitty <PROPN>', 'Kalimat tidak sesuai dengan variasi pola yang ada', 'S : Kitt y <PROPN> P : memiliki <VERB> O : bulu <NOUN> PEL : sangat <ADV> lembut <ADJ> K : ketika <SCONJ> dise ntuh <VERB>', 'S : Ia <PRON> P : memiliki <VERB> O : ekor <NOUN> yang <PRON> PEL : sangat <ADV> panja ng <ADJ> K : dan <CONJ> halus <ADJ>', ['Kalimat tidak sesuai dengan variasi pola yang ada', 'S : Ki tty <PROPN> P : senang <ADV> bermain <VERB>', 'S : Aku <PRON> P : melempar <VERB> O : bola <NOUN> K : ke <ADP> kitty <NOUN>', 'S : Ia <PRON> P : berlari <VERB> mengejar <VERB> O : bola <NOUN> itu <DET>', 'S : Ia <PRON> P : senang <VERB> menjilati <VERB> O : kakinya <NOUN>', 'S : Aku <PRON> P : rajin <VER B> memberi <VERB> O : makan <NOUN> Kitty <PROPN>', 'S : Ia <PRON> P : senang <VERB> O : makan <NOUN> makanan <NOUN> kaleng <NOUN>', 'S : Aku <PRON> P : juga <ADV> menjaga <VERB> O : kebersihan <NOUN> ka ndangnya <NOUN>', 'S : Kitty <PROPN> P : tidak <PART> suka <VERB> O : tempat <NOUN> kotor <ADJ>', 'S : Ia <PRON> P : tidak <PART> pernah <ADV> buang <VERB> O : kotoran <NOUN> sembarangan <NOUN>', 'S : A ku <PRON> P : sayang <VERB> O : Kitty <NOUN>']]

Gambar 9. Implementasi Proses Pemeriksaan

Hasil pemeriksaan secara keseluruhan untuk karangan yang sudah diperiksa dapat dilihat seperti pada Tabel 9. di bawah ini.

Tabel 9. Hasil Pemeriksaan Keseluruhan

| Nama file | Jumlah kalimat dalam karangan | Hasil pemeriksaan otomatis | | Hasil pemeriksaan manual | | | |
|------------|-------------------------------|----------------------------|-------|--------------------------|----|----|----|
| | | Berhasil | Gagal | TP | TN | FP | FN |
| Karangan 1 | 15 | 12 | 3 | 12 | 2 | 0 | 1 |
| Karangan 2 | 13 | 10 | 3 | 9 | 1 | 0 | 3 |
| Karangan 3 | 15 | 11 | 4 | 9 | 1 | 0 | 5 |

| Nama file | Jumlah kalimat dalam karangan | Hasil pemeriksaan otomatis | | Hasil pemeriksaan manual | | | |
|-------------|-------------------------------|----------------------------|-------|--------------------------|----|----|----|
| | | Berhasil | Gagal | TP | TN | FP | FN |
| Karangan 4 | 10 | 8 | 2 | 6 | 0 | 0 | 4 |
| Karangan 5 | 12 | 9 | 3 | 8 | 1 | 0 | 3 |
| Karangan 6 | 15 | 13 | 2 | 12 | 1 | 0 | 2 |
| Karangan 7 | 18 | 14 | 4 | 14 | 3 | 0 | 1 |
| Karangan 8 | 13 | 10 | 3 | 10 | 1 | 0 | 2 |
| Karangan 9 | 12 | 10 | 2 | 9 | 2 | 0 | 1 |
| Karangan 10 | 13 | 11 | 2 | 11 | 1 | 0 | 1 |
| Karangan 11 | 11 | 9 | 2 | 9 | 1 | 0 | 1 |
| Karangan 12 | 11 | 8 | 3 | 7 | 3 | 0 | 1 |
| Karangan 13 | 16 | 14 | 2 | 12 | 0 | 0 | 4 |
| Karangan 14 | 12 | 9 | 3 | 9 | 2 | 0 | 1 |
| Karangan 15 | 12 | 9 | 3 | 9 | 1 | 0 | 2 |
| Karangan 16 | 10 | 8 | 2 | 8 | 2 | 0 | 0 |
| Karangan 17 | 11 | 9 | 2 | 9 | 1 | 0 | 1 |
| Karangan 18 | 11 | 8 | 3 | 8 | 2 | 0 | 1 |
| Karangan 19 | 11 | 9 | 2 | 8 | 2 | 0 | 1 |
| Karangan 20 | 11 | 8 | 3 | 8 | 2 | 0 | 1 |
| Karangan 21 | 11 | 9 | 2 | 9 | 0 | 0 | 2 |
| Karangan 22 | 13 | 10 | 3 | 9 | 2 | 0 | 2 |
| Karangan 23 | 13 | 12 | 1 | 10 | 1 | 0 | 2 |
| Karangan 24 | 11 | 8 | 3 | 7 | 1 | 0 | 3 |
| Karangan 25 | 12 | 10 | 2 | 10 | 2 | 0 | 0 |
| Karangan 26 | 16 | 12 | 4 | 11 | 2 | 0 | 3 |
| Karangan 27 | 13 | 10 | 3 | 10 | 2 | 0 | 1 |
| Karangan 28 | 10 | 8 | 2 | 7 | 2 | 0 | 1 |
| Karangan 29 | 10 | 9 | 1 | 9 | 0 | 0 | 1 |
| Karangan 30 | 12 | 10 | 2 | 10 | 0 | 0 | 2 |
| Karangan 31 | 13 | 10 | 3 | 9 | 2 | 0 | 2 |
| Karangan 32 | 14 | 11 | 3 | 11 | 1 | 0 | 2 |
| Karangan 33 | 11 | 8 | 3 | 8 | 0 | 0 | 3 |
| Karangan 34 | 12 | 9 | 3 | 9 | 1 | 0 | 2 |
| Jumlah | 423 | 335 | 88 | 316 | 45 | 0 | 62 |

3.1.6. Penyimpanan Hasil

Proses penyimpanan hasil pemeriksaan akan dituliskan di bawah karangan asli pada file yang diinputkan pada proses input karangan. Proses ini menggunakan fungsi `text_file.write()`. Implementasi menyimpan hasil dapat dilihat seperti pada Gambar 10. di bawah ini.

```

1 save_hasil = text + '\n'
2 for i in range(len(hasil_akhir)):
3     save_hasil = save_hasil + '\n'
4     for j in range(len(hasil_akhir[i])):
5         save_hasil = save_hasil + hasil_akhir[i][j] + ' '
6
7 text_file = open("karangan.txt", "w")
8 text_file.write(save_hasil)
9 text_file.close()

```

Gambar 10. Implementasi Menyimpan Hasil

3.2. Pengujian Hasil

3.2.1. Uji Keberhasilan

Hasil pengujian keberhasilan secara keseluruhan karangan yang sudah diperiksa dapat dilihat seperti pada Tabel 10. di bawah ini.

Tabel 10. Hasil Pengujian Keberhasilan

| Nama file | Tingkat keberhasilan | Keberhasilan (%) |
|-------------|----------------------|------------------|
| Karangan 1 | 0,800 | 80,0% |
| Karangan 2 | 0,769 | 76,9% |
| Karangan 3 | 0,733 | 73,3% |
| Karangan 4 | 0,800 | 80,0% |
| Karangan 5 | 0,750 | 75,0% |
| Karangan 6 | 0,867 | 86,7% |
| Karangan 7 | 0,778 | 77,8% |
| Karangan 8 | 0,769 | 76,9% |
| Karangan 9 | 0,833 | 83,3% |
| Karangan 10 | 0,846 | 84,6% |
| Karangan 11 | 0,818 | 81,8% |
| Karangan 12 | 0,727 | 72,7% |
| Karangan 13 | 0,875 | 87,5% |
| Karangan 14 | 0,750 | 75,0% |
| Karangan 15 | 0,750 | 75,0% |
| Karangan 16 | 0,800 | 80,0% |
| Karangan 17 | 0,818 | 81,8% |
| Karangan 18 | 0,750 | 75,0% |
| Karangan 19 | 0,818 | 81,8% |
| Karangan 20 | 0,727 | 72,7% |
| Karangan 21 | 0,818 | 81,8% |
| Karangan 22 | 0,769 | 76,9% |
| Karangan 23 | 0,923 | 92,3% |
| Karangan 24 | 0,727 | 72,7% |
| Karangan 25 | 0,833 | 83,3% |
| Karangan 26 | 0,750 | 75,0% |
| Karangan 27 | 0,769 | 76,9% |
| Karangan 28 | 0,800 | 80,0% |
| Karangan 29 | 0,900 | 90,0% |
| Karangan 30 | 0,833 | 83,3% |
| Karangan 31 | 0,769 | 76,9% |
| Karangan 32 | 0,786 | 78,6% |
| Karangan 33 | 0,727 | 72,7% |
| Karangan 34 | 0,750 | 75,0% |
| Rata-rata | 0,792 | 79,2% |

3.2.2. Uji Keakuratan

Hasil pengujian keakuratan secara keseluruhan karangan yang sudah diperiksa dapat dilihat seperti pada Tabel 11. di bawah ini.

Tabel 11. Hasil Pengujian Keakuratan

| Nama file | Tingkat akurasi | Akrurasi (%) |
|-------------|-----------------|--------------|
| Karangan 1 | 0,933 | 93,3% |
| Karangan 2 | 0,769 | 76,9% |
| Karangan 3 | 0,667 | 66,7% |
| Karangan 4 | 0,600 | 60,0% |
| Karangan 5 | 0,750 | 75,0% |
| Karangan 6 | 0,867 | 86,7% |
| Karangan 7 | 0,944 | 94,4% |
| Karangan 8 | 0,846 | 84,6% |
| Karangan 9 | 0,917 | 91,7% |
| Karangan 10 | 0,923 | 92,3% |
| Karangan 11 | 0,909 | 90,9% |
| Karangan 12 | 0,909 | 90,9% |
| Karangan 13 | 0,750 | 75,0% |
| Karangan 14 | 0,917 | 91,7% |

| Nama file | Tingkat akurasi | Akrurasi (%) |
|-------------|-----------------|--------------|
| Karangan 15 | 0,833 | 83,3% |
| Karangan 16 | 1,000 | 100,0% |
| Karangan 17 | 0,909 | 90,9% |
| Karangan 18 | 0,909 | 90,9% |
| Karangan 19 | 0,909 | 90,9% |
| Karangan 20 | 0,909 | 90,9% |
| Karangan 21 | 0,818 | 81,8% |
| Karangan 22 | 0,846 | 84,6% |
| Karangan 23 | 0,846 | 84,6% |
| Karangan 24 | 0,636 | 63,6% |
| Karangan 25 | 1,000 | 100,0% |
| Karangan 26 | 0,813 | 81,3% |
| Karangan 27 | 0,923 | 92,3% |
| Karangan 28 | 0,900 | 90,0% |
| Karangan 29 | 0,900 | 90,0% |
| Karangan 30 | 0,833 | 83,3% |
| Karangan 31 | 0,846 | 84,6% |
| Karangan 32 | 0,857 | 85,7% |
| Karangan 33 | 0,727 | 72,7% |
| Karangan 34 | 0,83 | 83,0% |
| Rata-rata | 0,851 | 85,1% |

4. Pembahasan

Pemeriksaan pola kalimat otomatis ini memeriksa 34 karangan dengan jumlah kalimat 423 kalimat dan dapat memeriksa dengan benar 335 dari 423 kalimat yang ada pada karangan dengan 0,792 atau keberhasilan (%) sebesar 79,2%, dan memiliki tingkat akurasi sebesar 0,851 atau akurasi (%) sebesar 85,1%.

Hasil pengujian tersebut dipengaruhi oleh beberapa faktor. Faktor pertama adalah hasil pos tagging menggunakan Flair NLP. Hasil dari pos tagging tersebut dapat mengalami perubahan tergantung dengan posisi kata dalam kalimat tersebut. Semisal kata "makan" dalam kalimat "Rona senang makan ikan" akan menghasilkan label jenis kata <NOUN> atau kata benda. Sedangkan kata "makan" dalam kalimat "Kami makan di warung makan" menghasilkan label jenis kata <VERB> atau kata kerja. Hal ini dapat membuat kesalahan dalam proses selanjutnya.

Faktor yang kedua adalah variasi komponen tag penyusun pola dari masing masing pola kalimat yang terbatas. Dalam penelitian ini, peneliti menggunakan variasi komponen tag penyusun pola yang sangat terbatas. Selain dari terbatasnya variasi komponen tag penyusun pola, penggunaan pola kalimatnya juga masih tergolong sedikit. Peneliti hanya menggunakan 6 pola kalimat dalam penelitian ini, yaitu pola kalimat "S-P", "S-P-O", "S-P-K", "S-P-O-K", "S-P-Pel-K", dan "S-P-O-Pel-K". Pola-pola kalimat tersebut tergolong dalam kalimat tunggal aktif. Belum terdapat pola kalimat untuk memeriksa kalimat tunggal pasif, kalimat majemuk dan jenis kalimat lainnya. Artinya untuk kalimat pada karangan yang berjenis kalimat tersebut mengalami ketidaksesuaian dengan variasi komponen tag penyusun pola yang tersedia walaupun dalam pemeriksaan secara manual menghasilkan suatu pola kalimat yang benar.

Pada Tabel 10. dapat dilihat tingkat keberhasilan dan keberhasilan (%) dari masing masing karangan yang di inputkan. Pengujian ini menghitung jumlah kalimat yang dapat diperiksa dan menghasilkan sebuah pola kalimat dibagi dengan jumlah kalimat keseluruhan. Kalimat yang berhasil diperiksa dan menghasilkan sebuah pola kalimat otomatis sebanyak 335 kalimat dari 423 kalimat total yang diinputkan. Hasil ini menunjukkan bahwa penelitian ini memiliki tingkat keberhasilan rata-rata sebesar 0,792 dan keberhasilan (%) rata-rata sebesar 79,2%.

Pada Tabel 11. dapat dilihat bahwa hasil pada penelitian ini memiliki nilai tingkat keakurasian antara hasil pemeriksaan secara otomatis dan hasil pemeriksaan manual.

Pengujian ini membandingkan hasil dari pemeriksaan otomatis dengan hasil dari pemeriksaan manual. Terdapat 316 kalimat tergolong TP (True Positive), 45 kalimat tergolong TN (True Negative), 0 kalimat tergolong FP (False Positive), dan 62 kalimat tergolong FN (False Negative). Hasil ini menunjukkan bahwa penelitian ini memiliki tingkat keakurasian rata-rata sebesar 0,851 dan akurasi (%) rata-rata sebesar 85,1%.

5. Kesimpulan

Pemeriksaan pola kalimat otomatis ini dapat membantu dalam melakukan pemeriksaan terhadap suatu karangan sederhana secara otomatis. Pada pemeriksaan ini dapat memeriksa dengan benar 335 dari 423 jumlah total kalimat yang ada dengan nilai keberhasilan 0,792 atau keberhasilan (%) sebesar 79,2%, dan memiliki tingkat akurasi sebesar 0,851 atau akurasi (%) sebesar 85,1%. Adapun faktor yang mempengaruhi dari hasil penelitian ini adalah ketepatan library Flair NLP dalam pemberian label jenis kata, batasan dalam variasi komponen tag penyusun pola, dan juga batasan pola kalimat yang digunakan dalam penelitian ini menggunakan kalimat tunggal aktif saja.

Saran untuk penelitian berikutnya adalah (1) Penggunaan library nlp yang memiliki bank kata atau korpus yang lebih lengkap dan lebih akurat dalam penentuan label jenis kata dari library Flair NLP yang digunakan peneliti dalam penelitian ini. (2) Perlu ditambahkan untuk variasi komponen tag penyusun pola yang lebih lengkap untuk memeriksa lebih banyak lagi variasi kalimat. (3) Perlu ditambahkan untuk pola kalimat yang digunakan. Tidak hanya untuk pola kalimat dalam kalimat tunggal aktif saja, akan tetapi untuk pola kalimat dari jenis kalimat lainnya seperti kalimat tunggal pasif, kalimat majemuk dan jenis kalimat lainnya untuk dapat diperiksa dan diidentifikasi pola kalimatnya.

Referensi

- [1] H. G. Tarigan, *Menulis Sebagai Suatu Keterampilan Berbahasa*. Bandung: Angkasa, 2008.
- [2] Iskandarwassid and D. Sunendar, *Strategi Pembelajaran Bahasa*. Bandung: PT. Remaja Rosdakarya, 2015.
- [3] P. Susi, "Analisis Jenis-Jenis Kalimat Dalam Rangka Karangan Narasi Siswa Kelas VII SMP Pangudi Luhur Yogyakarta Tahun Ajaran 2017/2018," Universitas Sanata Dharma, 2019.
- [4] R. Aljatila, "Meningkatkan Keterampilan Menulis Karangan Deskriptif Melalui Model Kooperatif Tipe Round Table pada siswa kelas X-1 SMAN Kulisusu Barat," *Jurnal Humanika*, vol. 3, no. 15, pp. 1–14, 2015.
- [5] B. Wijiono, "Ketaksaan Bahasa Dalam Karangan Siswa Kelas V SD Negeri 2 Abean Kecamatan Mirit Kabupaten Kebumen Tahun Pembelajaran 2013/2014," Universitas Muhammadiyah Purworejo, 2014.
- [6] M. Izzudin, "Analisis Isi Karangan Mata Pelajaran Bahasa Indonesia Pada Siswa Kelas V MIN Sumberjati Kademangan Kabupaten Blitar," Universitas Islam Negeri Maulana Malik Ibrahim Malang, 2017.
- [7] S. Suwanto, "Analisis Pola Kalimat Bahasa Indonesia Dalam Karangan Siswa Kelas VI SD Negeri 02 Nglegok," pp. 1–15, 2016.
- [8] H. K. Wardana, I. Swanita, and B. W. Yohanes, "Sistem Pemeriksa Pola Kalimat Bahasa Indonesia berbasis Algoritme Left-Corner Parsing dengan Stemming," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 8, no. 3, p. 211, 2019, doi: 10.22146/jnteti.v8i3.515.
- [9] H. P. Siregar, "Pemeriksaan Struktur Kalimat Bahasa Indonesia Dengan Menggunakan Part-Of-Speech Tagging dan LALR Parser," Universitas Sumatera Utara, 2019.
- [10] M. R. Pratama, S. Kusumadewi, and T. Hidayat, "Penerapan Algoritma LALR Parser Dan Context-Free Grammar Untuk Struktur Kalimat Bahasa Indonesia," *Jurnal Teknologi Elektro, Universitas Mercu Buana*, vol. 1, pp. 1–8, 2017.
- [11] D. A. Wilkins, *Linguistics in Language Teaching*, vol. 68, no. 3. London: MIT Press Written Production, 1973. doi: 10.2307/3725001.
- [12] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. New Jersey: Prentice Hall PTR, 2000.
- [13] S. Gupta and S. Yadav, "Extraction of Link Context using Tag Tree and LALR Parsing," *Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013)*, pp. 253–257, 2013.

-
- [14] A. Aho and J. Ullman, *Principals of Compiler Design*. Narosa Publishing House (25th reprint 2003), 1977.
- [15] M. Y. H. Setyawan, R. M. Awangga, and S. R. Efendi, "Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot," *Proceedings of the 2018 International Conference on Applied Engineering, ICAE 2018*, p. 8579372, 2018, doi: 10.1109/INCAE.2018.8579372.