**Tabel 2.1** Kajian Penelitian Terdahulu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kriteria | Kajian 1 | Kajian 2 | Kajian 3 | Penelitian yang akan dikembangkan |
| Peneliti | Nurul Justina Mahardianing Verdaningroem, Aries Saifudin | Wahyudi, D., Susyanto, T., & Nugroho, D. | Rinci Kembang Hapsari dan Yunus Juli Santoso | Oko Carono |
| Judul | Penerapan Kamus Dasar Pada Algoritma Porter untuk Mengurangi Kesalahan Stemming Bahasa Indonesia | Implementasi dan analisis Algoritma stemming Nazief & Adriani dan Porter pada dokumen berbahasa Indonesia | *Stemming* Artikel Berbahasa Indonesia Dengan Pendekatan *Confix-Stripping.* | Aplikasi pencarian kata dasar dan kata baku pada dokumen berbahasa Indonesia menggunakan metode Nazief & Adriani |
| Nama Jurnal | Jurnal Teknologi 10 | Jurnal Ilmiah SINUS 15 | Prosiding Seminar Nasional Manajemen Teknologi XXII |  |
| Sumber | Sinta 2 | Sinta 2 | Sinta 2 |  |
| Volume, Nomor | Volume : 10, Nomor : 2 | - | - |  |
| Bulan, tahun Terbit | Juli, 2018 | Juli, 2017 | 24 Januari, 2015 |  |
| Teknologi | Php dan msql | Php dan msql | Php dan msql |  |
| Masalah Peneitian | Seberapa akurat proses *stemming* menggunakan kamus dan tidak menggunakan kamus. | Algoritma manakah yang lebih cepat, lebih akurat atau yang lebih banyak melakukan kesalahan stemming antara algoritma Nazief & Adriani dan algoritma Porter. | Untuk mencari nilai akurasi menggunakan pengujian stemming pada artikel berbahasa Indonesia menggunakan pendekatan *Confix-Stripping.* | Kesalahan penulisan kata baku pada sebuah dokumen berbahasa Indonesia |
| Tujuan Penelitian | Membandingkan ketepatan algoritma Nazief & Adriani dengan kemampuan Algoritma Porter dengan cara membuat program sederhana untuk mengetahui mana yang lebih baik dalam melakukan *stemming* pada dokumen berbahasa Indonesia. | Untuk mengetahui Algoritma manakah yang akurasinya lebih tinggi dan manakah Algoritma yang lebih cepat atau yang memiliki kesalahan *stemming* yang banyak. | Untuk mencari kata dasar dari sebuah kata berimbuhan dalam sebuah dokumen berbahasa Indonesia. | Menerapkan *stemming* pada aplikasi pencarian kata dasar dan kata baku pada dokumen berbahasa Indonesia menggunakan Algoritma Nazief & Adriani dan mengukur ketepatan presisi Algoritma Nazief & Adriani untuk proses *stemming* pada aplikasi pencarian kata dasar dan kata baku pada dokumen berbahasa Indonesia. |
| Metode/Teori | Menggunakan Algortima Porter. | Menggunakan Algoritma Nazief & Adriani dan Algortima Porter. | Menggunakan pendekatan *Confix-Stripping.* | Menggunakan Algoritma Nazief & Adriani. |
| Objek dan Variable | Dokumen berbahasa Indonesia | Dokumen berbahasa Indonesia | Dokumen berbahasa Indonesia | Dokumen berbahasa Indonesia |
| Metode Pengujian | Menggunakan 30 dokumen teks Bahasa Indonesia dan hanya 30 kata yang diujikan. | Menggunakan 26 dokumen yang berisi 8.168 kata, *double word* ada 1.312, kata stopword ada 4.724, dan kata yang benar digunakan untuk diuji ada 2.132 kata. | Menggunakan 20 dokumen teks Bahasa Indonesia. | Menggunakan 30 dokumen teks Bahasa Indonesia. |
| Hasil Penelitian | Hasil persentase menunjukkan bahwa proses *stemming* dengan ditambahkan kamus dasar memiliki persentase keakuratan kata yang benar lebih besar 13.333%. | Algoritma Nazief & Adriani mendapatkan akurasi 95.26% dan waktu proses 12.3822753429. Sedangkan Algoritma Porter mendapatkan akurasi 79.13% dan waktu proses 22.1668348312. | Hasil pengujian *stemming* yang dilakukan menggunakan pendekatan *Confix-Stripping*, mendapatkan nilai rata-rata akurasi 94.85%. | ketepatan pada pemrosesan *stemming* menggunakan kata yang ada dalam dokumen. |

## **Landasan Teori**

### **Pemrosesan Bahasa Alami**

Pemrosesan bahasa alami biasa di singkat dengan PBA atau NLP (*Natural* *Language* *Processing*) adalah salah satu bidang dari kecerdasan buatan, ilmu computer, dan bahasa (*linguistik*) yang bertujuan untuk mengkaji interaksi antara kompter dengan bahasa alami manusia dengan segala aturan gramatikal dan semantiknya serta mengubah bahasa tersebut menjadi repsentasi formal yang dapat diperoses oleh komputer (Charniak, 1984).

Ada 3 (tiga) aspek utama pada teori pemahaman mengenai *Natural* *Language* (Poole dan Mackworth, 2010):

1. *Syntax*: menjelaskan bentuk dari bahasa, *syntax* biasa dispesifikasikan oleh sebuah grammar.
2. *Semantics*: menjelaskan arti dari kalimat dalam satu bahasa.
3. *Pragmatics*: menjelaskan bagaimana pernyataan yang ada berhubungan dengan dunia.

Selain dari beberapa istilah tersebut ada juga beberapa istilah lainnya yang terkait dengan NLP, yaitu berdasarkan:

Morfologi merupakan sebuah pengetahuan tentang kata dan bentuknya sehingga bisa dibedakan antara yang satu dengan yang lainnya. Contoh : membangunkan –> bangun (kata dasar), mem- (prefix), -kan (suffix).

Fonetik merupakan semua hal yang berhubungan dengan suara dan menghasilkan kata yang dapat dikenali. Fonetik digunakan dalam pengembangan NLP khususnya bidang *speech based system.*

Pemrosesan bahasa alami atau NLP (*Natural* *Language* *Processing*) mempelajari tentang bagaimana memahami  bahasa manusia alami dengan cara mengubah gambaran bahasa manusia (seperti dokumen teks) menjadi penyajian yang lebih formal (dalam bentuk data *numeric* dan simbolik) yang lebih mudah untuk dimanipulasi oleh program komputer. NLP berupaya memecahkan masalah untuk memahami bahasa alami manusia dengan segala aturan gramatikal dan semantiknya dan mengubah bahasa tersebut menjadi representasi formal yang dapat diproses oleh komputer. NLP merupakan komponen yang penting dari *text mining* dan merupakan sub-bidang dari *artificial intelligence* dan komputasional linguistik. *Text mining* menggunakan *natural language processing* untuk memasukkan stuktur kedalam kumpulan teks (Sulhan & Kurniawan, 2014). *Text mining* menggunakan *text preprocessing* untuk pencarian, pengambilan intisari, dan pengkategorian kata pada sebuah dokumen.

### ***Text Preprocessing***

Herwijayanti dkk. (2018) menyatakan bahwa *processing* bertujuan untuk mendapatkan dataset yang dapat diolah dengan cepat dan menghasilkan kesimpulan yang tepat, salah satu proses *processing* data yang dapat dilakukan adalah pemilihan fitur (*feature* *selection*). Ada beberapa tahapan dalam pemilihan fitur, antara lain:

* + - * 1. *Tokenizing*

*Tokenizing* adalah tahap pemotongan kata atau kalimat berdasarkan tiap kata. Kata yang diproses disebut dengan token/*term* dan yang nantinya disimpan di dalam *database* adalah *term* lalu untuk melakukan pencarian maka dilakukan indexing. Contoh : Aku makan bakso urat, hasil dari tokenizing dari kalimat tersebut ada empat token, yaitu : “Aku”, “makan”, “bakso”, “urat”. (Oeyliawan & Gunawan, 2017).

* + - * 1. *Filtering*

*Filtering* adalah tahap mengambil kata-kata yang penting dari hasil *tokenizing*. Proses *filtering* dapat menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata yang penting). *Stoplist* / *stopword* adalah kata-kata yang tidak *deskriptif* yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopword* adalah “yang”, “dan”, “di”, “dari” dan lain – lain (Juang, 2016).

* + - * 1. *Stemming*

*Stemming* adalah salah satu cara yang digunakan untuk meningkatkan performa IR (*Information Retrieval*) dengan cara mentransformasi kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya. *Stemming* adalah proses yang digunakan untuk mencari kata dasar dari kata berimbuhan dengan cara menghapus semua imbuhan (*affix*) yang ada diawalan (*prefix*), disisipan (*infix*), diakhiran (*suffix*) dan imbuhan yang dikombinasikan menggunakan awalan dan akhiran (*confix*). (Dini Nopiyanti dan Kemal Ade Sekarwati, 2014).