# **BAB II**

**Kajian Pustaka**

## **Kajian Terdahulu**

Pada bagian ini akan dipaparkan mengenai kajian penelitian terdahulu yang akan dijadikan sebagai acuan dasar dalam penelitian. Selain kajian terdahulu, akan dijelaskan tentang kajian teori yang mendukung penelitian ini. Berikut beberapa kajian terdahulu yang menjadi acuan penelitian:

Nurul Justina Mahardianing Verdaningroem dan Aries Saifudin. (2018) memaparkan penelitian terkait tentang penerapan kamus dasar pada algoritma Porter untuk mengurangi kesalahan stemming Bahasa Indonesia. Penelitian ini bertujuan untuk membandingkan ketepatan algoritma Nazief & Adriani dengan kemampuan Algoritma Porter, proses pembandingan Algoritma porter dan Nazief & Adriani dilakukan dengan membuat program sederhana yang memproses dokumen teks Bahasa Indonesia sehingga diketahui stem, waktu proses, presisi dari hasil *stemming* dokumen tersebut. Pengujian Algoritma dilakukan pada 30 dokumen teks Bahasa Indonesia dengan ukuran dokumen yang bervariasi. Penelitian ini melakukan eksperimen menggunakan komputer untuk melakukan proses *stemming*. Sedangkan perangkat lunak yang digunakan untuk menyimpan *database* kamus adalah MySQL.

**Gambar 2.1** Hasil Pengujian 30 Kata Yang Dipilih Secara Acak



Dari Gambar 2.1 hasil pengujian di atas menyatakan bahwa ada beberapa kata imbuhan yang di *stemming* dengan kamus menghasilkan kata dasar yang rata-rata masih memiliki makna, namun pada hasil kata dasar yang di stemming tanpa kamus menghasilkan beberapa kata dasar yang ambigu atau membingungkan seperti “gum”, ”rogram”, “gala”, “njara”, “iksa”, ”masu” dan “ail”. Sementara untuk perbandingan waktu menunjukkan bahwa waktu yang diperlukan dalam proses *stemming* dengan kamus membutuhkan waktu beberapa detik lebih lama dari pada *stemming* tanpa menggunakan kamus. Dapat dilihat pada kolom waktu *stemming*, pada *stemming* tanpa kamus menunjukkan bahwa dari 30 kata berimbuhan yang diproses stemming rata-rata menunjukkan waktu 0,00030 detik, sementara untuk *stemming* dengan menggunakan kamus membutuhkan lebih dari 0,001 detik pada Gambar 2.1 tersebut menunjukkan bahwa rata-rata waktu *stemming* dengan kamus adalah 0,50767 detik. Percobaan waktu ini dicoba pada proses *stemming* setiap 1 kata berimbuhan. Hasil dari proses *stemming* dengan memasukkan 30 kata secara langsung dan memprosesnya secara bersamaan menghasilkan waktu 0,00813 detik untuk proses *stemming* tanpa kamus dan 15,50625 detik untuk proses *stemming* dengan menggunakan kamus. Selisih waktu yang dihasilkan adalah 15,49812 detik. Hasil dari persentase kata dasar yang di *stemming* dengan kamus dapat dilihat pada perhitungan berikut: ((30 - 5)) / 30 X 100% = 83.333%. Sementara hasil kata dasar yang di *stemming* tanpa kamus dapat dilihat pada perhitungan berikut: ((30 - 9)) / 30 X 100% = 70%. Dari hasil presentase menunjukkan bahwa proses *stemming* dengan ditambahkan kamus dasar memiliki persentase keakuratan kata yang benar lebih besar 13.333% dari pada persentase keakuratan *stemming* yang hanya menggunakan aturan yang sudah ada.

Selanjutnya dalam penelitian Wahyudi dkk. (2017) memaparkan tentang implementasi dan analisis Algoritma *stemming* Nazief & Adriani dan Porter pada dokumen berbahasa Indonesia. Penelitian ini bertujuan untuk membandingkan Algoritma Nazief & Adriani dan Algoritma Porter untuk mengetahui proses *stemming* pada teks berbahasa Indonesia. Pada penelitian ini menggunakan data berupa kamus kata dasar yang di unduh dari internet dengan jumlah kata 28.532 kata. Kata-kata tersebut kemudian disimpan dalam *database* yang akan digunakan untuk proses *stemming*. Pada penelitian ini menggunakan 26 dokumen yang memiliki 8.168 kata yang diujikan, kata yang mucul lebih dari satu kali (*double word*) adalah 1.312, kata yang termasuk dalam *stopword* ada 4.724 dan kata yang benar-benar diujikan pada kedua *stemmer* yang diterapkan pada penelitian ini adalah 2.132. Dari total 2.132 kata yang diuji, Algortima Nazief & Adriani menghasilkan *root word* yang benar sebanyak 2.031 kata, sedangkan Algoritma Porter menghasilkan *root word* yang benar sebanyak 1.687 kata. Tingkat kesalahan *stemmer* Nazief & Adriani hanya 5%, sangat jauh jika dibandingkan dengan Porter yaitu 21%. Tetapi *stemmer* Nazief &Adriani membutuhkan waktu proses hampir 2 kali lebih lama jika dibanding Algoritma Porter. Dari hasil tersebut Algoritma Nazief & Adriani memberikan hasil yang lebih baik untuk *stemming* pada dokumen berbahasa Indonesia. Namun untuk efisiensi waktu proses, Algoritma Porter lebih baik dibanding Agoritma Nazief & Adriani.

Penelitian lainnya tentang *stemming* artikel berbahasa Indonesia dengan pendekatan *confix-stripping*. Tujuan penelitian ini adalah mencari kata dasar dari kata imbuhan teks bebahasa Indonesia dalam sebuah dokumen. Tahap *stemming* dilakukan dengan cara: pertama, *parsing* dokumen dilakukan untuk memecah sebuah dokumen menjadi kata-kata atau disebut token. Tahap kedua, *stopword* yang merupakan proses penghilangan kata yang tidak penting dalam dokumen. Dan tahap ketiga, proses *confix-stripping*. Dimana *Confix-Stripping* merupakan Algoritma *stemming* untuk pemenggalan atau pengupasan kata imbuhan awalan, akhiran, dan gabungan antara awalan-akhiran menjadi bentuk dasar. Berdasarkan hasil pengujian yang dilakukan dengan menggunakan pendekatan *Confix-Stripping*, maka hasil rata-rata nilai akurasi yang diperoleh sebesar 94.85% dari 20 dokumen teks berbahasa Indonesia yang diuji (Hapsari dan Santoso, 2015).

Berdasarkan kajian terdahulu yang telah dikaji, maka dilakukan penelitian tentang aplikasi pencarian kata dasar dan kata baku pada dokumen berbahasa Indonesia menggunakan metode Nazief & Adriani. Masalah dari penelitian ini adalah masih seringnya terjadi kesalahan-kesalahan penulisan kata baku pada sebuah dokumen berbahasa Indonesia. Tujuannya adalah untuk menerapkan stemming pada aplikasi pencarian kata dasar dan kata baku pada dokumen berbahasa Indonesia menggunakan Algoritma Nazief & Adriani dan mengukur tingkat presisi pada aplikasi pencarian kata dasar dan kata baku pada dokumen berbahasa Indonesia menggunakan Algoritma Nazief & Adriani.. Adapun hasil dari penelitian ini adalah menghasilkan suatu sistem yang dapat membantu mahasiswa dalam perncarian kata dasar pada dokumen dan dapat menjadi referensi dalam pencarian kata baku pada sebuah dokumen.

**Tabel 2.1** Kajian Penelitian Terdahulu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kriteria | Kajian 1 | Kajian 2 | Kajian 3 | Penelitian yang akan dikembangkan |
| Peneliti | Nurul Justina Mahardianing Verdaningroem, Aries Saifudin | Dwi Wahyudi, Teguh Susyanto, Didik Nugroho | 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* menggunkan 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, proses pembandingan Algoritma Porter dan Nazief & Adriani dilakukan dengan membuat program sederhana yang memproses dokumen teks Bahasa Indonesia sehingga diketahui stem, waktu proses, presisi dari hasil *stemming* dokumen tersebut. | Membandingkan Algoritma Nazief & Adriani dan Algoritma Porter untuk mengetahui proses *stemming* pada teks berbahasa Indonesia, sehingga dari hasil penelitian akan diketahui manakah Algoritma yang memiliki akurasi lebih tinggi dan manakah Algoritma yang lebih cepat atau yang lebih banyak melakukan kesalahan *stemming.* | Mencari kata dasar dari kata imbuhan teks beBahasa  Indonesia dalam sebuah dokumen. | 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 |  |  |  |  |
| Hasil Penelitian | Hasil persentase menunjukkan bahwa proses stemming dengan ditambahkan kamus dasar memiliki persentase keakuratan kata yang benar lebih besar 13.333% dari pada persentase keakuratan stemming yang hanya menggunakan aturan yang sudah ada. | Algoritma Nazief & Adriani memberikan hasil yang lebih baik untuk stemming pada dokumen berbahasa Indonesia. Namun untuk efisiensi waktu proses, algoritma Porter lebih baik dibanding algoritma Nazief & Adriani. | Hasil pengujian yang dilakukan stemming terhadap beberapa kata dalam dokumen dengan menggunakan pendekatan Confix-Stripping, dihasilkan rata-rata nilai akurasi yang diperoleh sebesar 94.85% dari 20 dokumen teks berbahasa Indonesia yang diuji. | Menghasilkan suatu sistem yang dapat membantu mahasiswa dalam perncarian kata dasar pada dokumen dan dapat menjadi referensi dalam pencarian kata baku pada sebuah dokumen. |

## **Landasan Teori**

* + 1. **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).

Poole dan Mackworth. (2010) menjelaskan bahwa ada 3 (tiga) aspek utama pada teori pemahaman mengenai *Natural* *Language*:

* + - 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 ketiga istilah tersebut ada juga beberapa istilah lainnya yang terkait dengan NLP, yaitu berdasarkan:

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

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

* + 1. ***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 string input berdasarkan tiap kata yang menyusunnya. Hasil pemrosesan akan berupa kata yang disebut dengan token/term. Term ini nantinya akan disimpan ke dalam database untuk dilakukan indexing saat melakukan pencarian (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* merupakan suatu proses untuk menemukan kata dasar dari sebuah kata dengan cara menghilangkan semua imbuhan (*affix*) baik yang terdiri dari awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*) dan kombinasi dari awalan dan akhiran (*confix*) pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi Bahasa Indonesia yang baik dan benar (Dini Nopiyanti dan Kemal Ade Sekarwati, 2014).

Menurut Singh & Gupta. (2017) *stemming* adalah prosedur dimana berbagai varian kata morfologis dicocokkan dengan *root word*. Misalnya, kata-kata seperti dalam Bahasa Inggis berikut: *maintaining*, *maintained*, *maintenance* dicocokan pada *root word* untuk *maintain* melalui *stemming*. Program yang melakukan *stemming* disebut *stemmers*. *Stemming* adalah jenis pemrosesan bahasa paling sederhana yang digunakan dalam sistem IR (*Information Retrieval*) dan ditemukan lebih bermanfaat dalam bahasa dengan morfologi kompleks di mana satu kata memiliki sejumlah besar varian.

Novitasari. (2016) menyatakan bahwa *stemming* juga bisa disebut sebagai proses untuk menghilangkan imbuhan dari suatu kata di dokumen atau *query* menjadi *root* *word*. Algoritma *Stemming* adalah prosedur komputasi yang digunakan untuk mencari asal kata dari suatu kalimat dengan cara memisahkan masing-masing kata dari kata dasar dan imbuhannya. Algoritma *stemming* mempunyai efektifitas yang dipengaruhi oleh beberapa factor, yaitu:

1. *Overstemming*: yaitu pemenggalan imbuhan yang melebihi dari yang seharusnya. Contoh: kata “masalah” menjadi “masa”. Kesalahan ini dapat timbul karena bentuk kata dasar yang menyerupai imbuhan.
2. *Understemming*: yaitu pemenggalan imbuhan yang terlalu sedikit dari yang seharusnya. Contoh: kata “belajar” menjadi lajar. Kesalahan ini dapat timbul karenakekurangan pada aturan pola imbuhan yang didefinisikan.
3. *Unchange*: yaitu kasus khusus dari understemming, dimana tidak terjadi pemenggalan imbuhan samasekali. Contoh: kata “telapak”, setelah pemenggalan kata dasar yang didapat tetap “telapak”. Kesalahan ini dapat ditimbulkan karena kekurangan pada aturan pola imbuhan yang didefinisikan.
4. *Spelling exception*: yaitu huruf pertama kata dasar yang didapat tidak benar yang diakibatkan dari penggalan awalan. Contoh: kata “memukul” menjadi “ukul”. Kesalahan ini dapat ditimbulkan karena ada beberapa imbuhan yang berubah bentuk ketika ditempelkan pada suatu kata dasar. Misalnya awalan beR-, meN-, teR-, peR-, akan bergantung pada huruf pertama kata dasar dimana imbuhan tersebut ditempelkan (Contoh: ber- + ajar = belajar, pen- + lihat = penglihatan, pen- + sakit = penyakit). Atau sebaliknya ada imbuhan y ang mengakibatkan huruf pertama kata dasar yang ditempelinya menjadi luluh. Misalnya meng- / peng- meluluhkan huruf „k‟ ( Contoh: mengarang dari meng- dan karang) atau men- / pen- meluluhkan huruf „p‟ (Contoh: menuai dari men- dan tuai).
   * 1. **Algoritma Nazief & Adriani**

Algoritma ini dibuat oleh Boby Nazief dan Mirna Andriani dari Fakultas Ilmu Komputer Universitas Indonesia tahun 1996, algoritma ini mengacu pada aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan, yaitu imbuhan yang diperbolehkan atau imbuhan yang tidak diperbolehkan. Algoritma Nazief & Adriani merupakan algoritma untuk mengubah kata yang memiliki *sufiks*, *prefiks* dan *konfiks* menjadi bentuk kata dasar Wibowo. (2016). Algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yaitu penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih. Kamus kata dasar tersebut dibutuhkan untuk memeriksa apakah kata dasar yang melalui proses *stemming* benar dan ditemukan pada kamus saat proses stemming dilakukan. Algoritma Nazief & Adriani memiliki beberapa tahapan sebagai berikut (Wahyudi dkk.*,* 2017).:

Kata yang belum di *stemming* dicari pada kamus, jika ditemukan, kata tersebut dianggap sebagai kata dasar yang benar dan algoritma dihentikan.

Hilangkan *Inflectional suffixes*, yaitu dengan menghilangkan partikel (“-lah”, ”-kah”, “-tah” atau “-pun”), kemudian hilangkan *inflectional* *possessive pronoun suffixes* (“-ku”, “-mu” atau ”-nya”). Cek kata di dalam kamus kata dasar, jika ditemukan, algortima dihentikan, jika tidak lanjut ke langkah 3.

Hapus Derivational Suffix (“-i” atau ”-an”). Jika kata ditemukan dalam kamus kata dasar, maka algoritma berhenti. Jika tidak, maka lanjut ke langkah 3a:

Jika akhiran “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.

Akhiran yang dihapus (“-i”, “- an” atau “-kan”) dikembalikan, lanjut ke langkah 4.

* 1. Hapus Derivational Prefix (“be-”, ”di-”, ”ke-”, ”me-”, ”pe-“, ”se-” dan “te-“). Jika kata yang didapat ditemukan didalam database kata dasar, maka proses dihentikan. Jika tidak, maka lakukan recoding. Tahapan ini dihentikan jika memenuhi beberapa kondisi berikut:
     + - 1. Terdapat kombinasi awalan dan akhiran yang tidak diijinkan
         2. Awalan yang dideteksi sama dengan awalan yang dihilangkan sebelumnya.
         3. Tiga awalan telah dihilangkan.
  2. Jika semua langkah telah dilakukan tetapi kata dasar tersebut tidak ditemukan pada kamus, maka algoritma ini mengembalikan kata yang asli sebelum dilakukan stemming.

Tipe awalan ditentukan melalui langkah-langkah berikut (Pramudita, 2014):

* + - 1. Jika awalannya adalah: “di-”, “ke-”, atau “se-” maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
      2. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-” maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
      3. Jika dua karakter pertama bukan “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-” maka berhenti.
      4. Jika tipe awalan adalah “none” maka berhenti.

Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada.

**Tabel 2.2** Kombinasi Awalan Akhiran Yang Tidak Diijinkan

|  |  |
| --- | --- |
| **Awalan** | **Akhiran yang tidak diijinkan** |
| be- | -i |
| di- | -an |
| ke- | -i, -kan |
| me- | -an |
| se- | -i, -kan |

**Tabel 2.3** Cara Menentukan Tipe Awalan Untuk Kata Yang Diawali Dengan “Te-”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Following Characters*** | | | | **Tipe Awalan** |
| **Set 1** | **Set 2** | **Set 3** | **Set4** |
| “-r-” | “-r-” | - | - | none |
| “-r-” | vowel | - | - | ter-luluh |
| “-r-” | not (vowel or “-r-“) | “-er-“ | vowel | ter |
| “-r-” | not (vowel or “-r-“) | “-er-“ | not vowel | ter- |
| “-r-” | not (vowel or “-r-“) | not “-er-“ | - | ter |
| not (vowel or “-r-“) | “-er-“ | vowel | - | none |
| not (vowel or “-r-“) | “-er-“ | not vowel | - | te |

**Tabel 2.4** Jenis Awalan Berdasarkan Tipe Awalannya

|  |  |
| --- | --- |
| Tipe Awalan | Awalan yang harus dihapus |
| di- | di- |
| ke- | ke- |
| se- | se- |
| te- | te- |
| ter- | ter- |
| ter-luluh | ter |

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan di bawah (Pramudita, 2014):

Aturan untuk reduplikasi.

Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word* adalah bentuk tunggalnya, contoh: “buku-buku” *root word*-nya adalah “buku”.

Kata lain, misalnya “bolak-balik”, “berbalasbalasan,dan ”seolah-olah”. Untuk mendapatkan *root word*-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama maka diubah menjadi bentuk tunggal, contoh: kata “berbalas-balasan”, “berbalas” dan “balasan” memiliki root word yang sama yaitu “balas”, maka *root word* “berbalas-balasan” adalah “balas”. Sebaliknya, pada kata “bolak-balik”, “bolak” dan “balik” memiliki *root word* yang berbeda, maka root word-nya adalah “bolak-balik”.

Tambahan bentuk awalan dan akhiran serta aturannya. Untuk tipe awalan “mem-“, kata yang diawali dengan awalan “memp-” memiliki tipe awalan “mem-”. Tipe awalan “meng-“, kata yang diawali dengan awalan “mengk-” memiliki tipe awalan “meng-”.

* + 1. **Kata Baku Bahasa Indonesia**

Dalam KBBI Edisi Keempat disebutkan pengertian baku adalah pokok, utama; tolok ukur yang berlaku untuk kuantitas dan kualitas yang ditetapkan berdasarkan kesepakatan; standar (Setiawati, 2016).

Sementara menurut Kosasih dan Hermawan (2012:83) kata baku adalah kata yang cara pengucapan ataupun penulisannya sesuai dengan kaidah- kaidah yang dibakukan. Kaidah standar yang dimaksud dapat berupa pedoman ejaan (EYD), tata bahasa baku, dan kamus. Kata baku digunakan dalam konteks ragam baku, baik lisan maupun tulisan. Sementara kata tidak baku digunakan dalam ragam tidak baku.Ragam bahasa baku dapat dibatasi dengan beberapa sudut pandang, di antaranya: (1) sudut pandang kebakuan bahasa yang digunakan, (2) sudut pandang informasi, dan (3) sudut pandang pengguna bahasa.

Berdasarkan sudut pandang kebakuan bahasa, bahasa baku adalah bahasa yang baik tata tulis, kosakata, maupun tata bahasanya sesuai dengan hasil pembakuan bahasa. Dari sudut pandang informasi, bahasa baku adalah ragam bahasa yang digunakan dalam berkomunikasi tentang ilmu pengetahuan. Lalu berdasarkan sudut pandang pengguna bahasa, ragam bahasa baku dapat dibatasi dengan ragam bahasa yang lazim digunakan oleh penutur yang paling berpengaruh, seperti ilmuan, pemerintah, tokoh masyarakat, dan kaum jurnalis atau wartawan. Bahasa merekalah yang dianggap ragam bahasa baku (Mulyono dalam Chaer, 2011:5).

Berdasarkan definisi-definisi di atas dapat disimpulkan bahwa kata baku adalah kata-kata yang lazim digunakan dalam situasi formal atau resmi yang penulisannya sesuai dengan kaidah-kaidah yang dibakukan. Baku tidaknya sebuah kata dapat dilihat dari segi lafal, ejaan, gramatika, dan kenasionalan-nya (Chaer, 2011:131).

Menurut Moeliono dalam Chaer (2011:131) lafal baku bahasa Indonesia adalah lafal yang tidak menampakkan lagi ciri-ciri bahasa daerah atau bahasa asing seperti contoh berikut:

|  |  |
| --- | --- |
| **Tidak Baku** | **Baku** |
| rapet | rapat |
| cuman | cuma |
| dudu’ | duduk |
| gubug | gubuk |

Baku dari sudut pandang ejaan berarti semua kata yang tidak ditulis menurut kaidah yang diatur dalam EYD adalah kata yang tidak baku. Sementara yang ditulis sesuai dengan aturan EYD adalah kata yang baku. Contoh:

|  |  |
| --- | --- |
| **Tidak Baku** | **Baku** |
| *Ekrpes* | *ekspres* |
| *Komplek sistem* | *Kompleks sistim* |
| *do’a* | *doa* |
| *Jum’at* | *jumat* |

Secara gramatikal kata-kata baku ini harus dibentuk menurut kaidah- kaidah gramatika. Perhatikan kata- kata ngontrak, sekolah, tinjau, kedudukk an, dan bikin bersih pada kalimat- kalimat berikut!

1. Beliau ngontrak rumah di Cilandak
2. Seorang ibu sedang urus KJP anaknya
3. Rumah ini akan dikontrakan
4. Tolong bikin rapi kelas ini.

Bentuk baku kata ngontrak adalah mengontrak. Bentuk baku kata urus adalah kata mengurus; sebuah awalan me- harus digunakan secara konsisten. Bentuk baku kata dikontrakan adalah dikontrakkan (dua k). Lalu, bentuk baku kata bikin rapi adalah rapikan.

Berikut adalah beberapa contoh kosakata baku dalam KBBI yang dapat disosialisasikan:

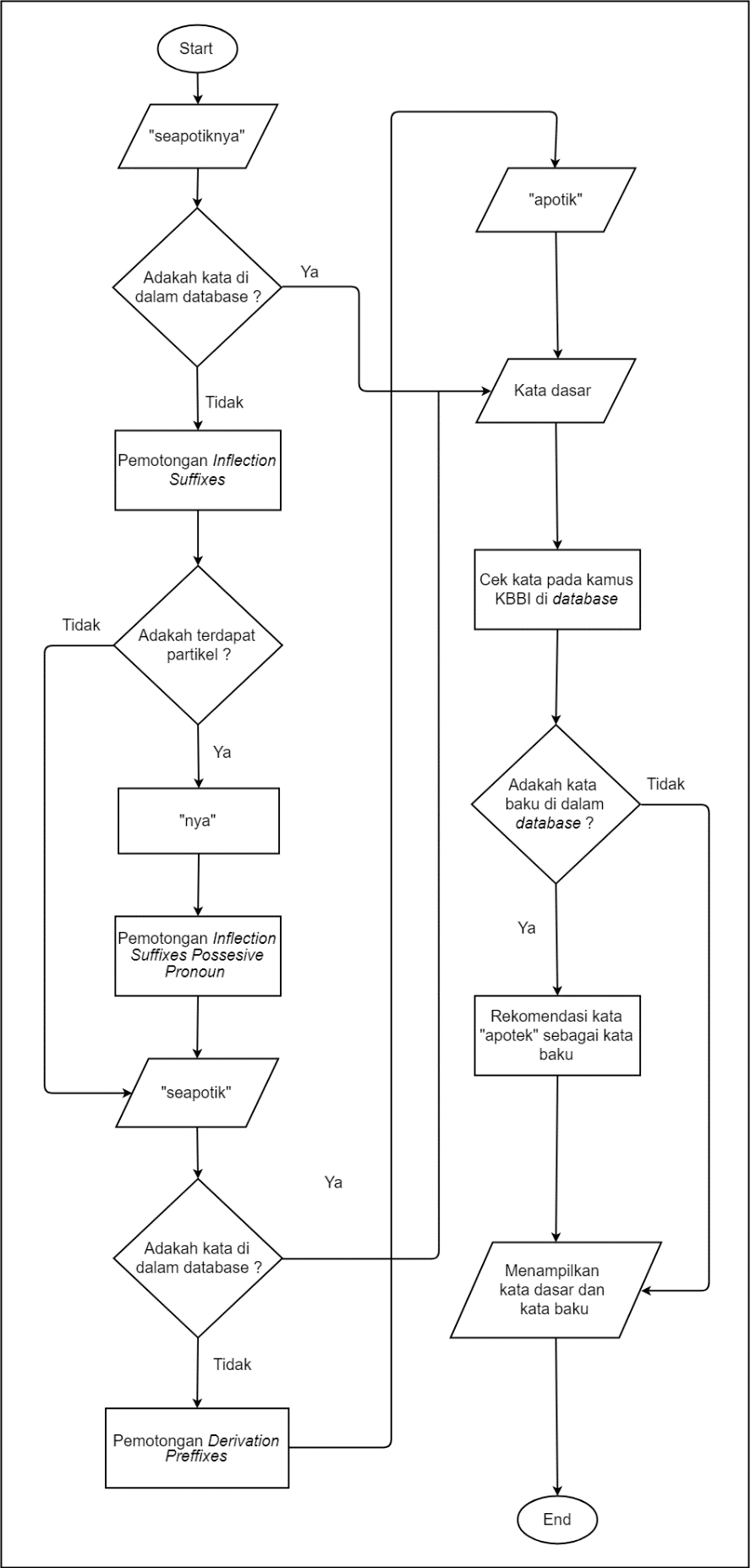
**Tabel 2.5** Contoh Kosakata dalam KKBI

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Contoh Kosakata dalam KBBI** | | **Keterangan** |
| **Tidak Baku** | **Baku** |
| 1 | Apotik | Apotek | Toko tempat meramu dan menjual obat berdasarkan resep dokter serta memperdagangk an barang medis; rumah obat |
| 2 | Astana | Istana | Rumah kediaman resmi raja  (kepala negara, presiden) dan keluarganya |
| 3 | Beludru | Beledu | Kain dengan permukaan yang tebal, berbulu halus pada  bagian depan dan rata pada bagian belakang, lembut, berkilat, sering dibuat kopiah atau baju kebesaran |
| 4 | Coklat | Cokelat | Bubuk dari biji cokelat; gula- gula yang dibuat dari bibit cokelat; merah kehitam-hitaman seperti warna sawo matang |
| 5 | Hutang | Utang | Uang yang dipinjam dari orang  lain; kewajiban membayar kembali apa telah diterima |
| 6 | Kate | Katai | Kecil atau kerdil pertumbuhannya |
| 7 | Kawatir | Khawatir | Takut (gelisah, cemas) terhadap suatu hal yang belum diketahui dengan pasti |
| 8 | Kwartet | Kuarted | Kelompok, kumpulan, dan sebagainya yang terdiri  Atas empat; komposisi musik yang terdiri atas empat instrument |
| 9 | Lahat | Lahad | Ceruk atau relung di lubang kubur tempat meletakkan mayat |
| 10 | Materai | Meterai | Cap tanda berupa gambar yang tercantum pada kertas atau terukir pada kayu, besi, dsb; cap; tera;.segel |
| 11 | Meleset | Peleset | Tidak mengenai sasaran; tidak mengenai yang dituju |
| 12 | Milyar | Miliar | Seribu juta |
| 13 | Nampak | Tampak | Dapat dilihat, kelihatan |
| 14 | Nakoda | Nakhoda | Juragan (pemimpin) perahu (kapal); perwira laut yang memegang komando tertinggi di atas kapal niaga; kapten kapal |
| 15 | Pamungkas | Pemungkas | Yang terakhir, yang menimbulkan kematian |
| 16 | Rubah | Ubah | Tukar, gant |
| 17 | Salih | Saleh | Taatdan sungguh- sungguh menjalankan ibadah |
| 18 | Selebritis | Selebritas | Pesohor |
| 19 | Stop | Stop | Berhenti; terhenti |
| 20 | Senen | Senin | Hari ke-2 dalam jangka waktu satu minggu |

* + 1. **Contoh Kasus Stemming Bahasa Indonesia Untuk Mencari Kata Dasar Dan Kata Baku Menggunakan Algoritma Nazief & Adriani**

Pada saat *stemming* kamus Bahasa Indonesia yang digunakan sangat mempengaruhi hasil dari *stemming* yang dilakukan, kerena semakin lengkap kamus yang digunakan maka semakin akurat pula hasil *stemming* yang di dapat. Contoh kata yang akan di *stemming* adalah kata “seapotiknya”. Untuk mencari kata baku dari kata “seapotiknya” maka harus dilakukan pencarian kata dasarnya terlebih dahulu, berikut langkah-langkah pencarian kata dasar menggunakan algoritma Nazief & Adriani:

**Gambar 2.2** Contoh Kasus *Stemming* “seapotiknya”



Penjelasan dari Gambar 2.2contoh kasus s*temming* “seapotiknya” adalah sebagai berikut:

1. Cek kata “seapotiknya” di dalam kamus kata dasar, kata “seapotiknya” tidak ditemukan di dalam kamus kata dasar.
2. Hapus *Inflection Suffixes*. Pada proses hapus *Inflection Suffixes* tidak terdapat partikel sehingga langsung dilakukan pengecekan hapus *Inflection Suffixes* *Possesive Pronoun* (“nya”) sehingga kata menjadi “seapotik”. Cek kata “seapotik” di dalam kamus kata dasar, kata “seapotik” tidak ditemukan di dalam kamus kata dasar.
3. Hapus *Derivation Preffixes* (“se”). Cek kata “apotik” di dalam kamus kata dasar, kata “apotik” tidak ditemukan pada kamus kata dasar.
4. Proses *stemming* berhenti dan menghasilkan kata dasar “apotik”.
5. Cek kata “apotik” di kamus besar Bahasa Indonesia (KBBI) yang tersimpan di dalam *database* untuk mencari kata baku dari “apotik”, kata “apotek” ditemukan di *database* dan di rekomendasikan sebagai kata baku dari kata “apotik”.

Jadi, kata dasar dari “seapotiknya” adalah “apotik” dan kata bakunya adalah “apotek”.