JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 4
No. 1
Juni 2018

Spelling Corrector Bahasa Indonesia dengan Kombinasi Metode Peter Norvig dan *N*-Gram

Maya Salinka Simanjuntak[#1], Herry Sujaini[#2], Novi Safriadi[#3]
[#]*Program Studi Informatika Universitas Tanjungpura*
Jalan Prof. Dr. H. Hadari Nawawi, Pontianak, Kalimantan Barat
[1]salinkamaya@gmail.com
[2]hs@untan.ac.id
[3]safriadi@informatics.untan.ac.id

***Abstrak*-Kesalahan pengetikan dalam suatu dokumen merupakan *human error* yang sulit dihindari, akibatnya pesan yang ingin disampaikan tidak maksimal. Menggunakan fitur *Spelling Corrector* menjadi salah satu cara untuk mengecek kesalahan-kesalahan pengetikan. Metode-metode yang digunakan mampu memberikan saran-saran kata yang benar, tapi tidak mampu memperbaiki kata yang salah secara langsung. Pengguna harus memilih satu kata yang diinginkan dari saran-saran kata yang dihasilkan oleh fitur. Dibutuhkan fitur *Spelling Corrector* yang mampu memberikan hanya satu saran kata dan langsung memperbaikinya. Melihat cara berbagai macam metode memberikan saran kata, kombinasi metode Peter Norvig dan *N*-Gram mampu menghasilkan satu saran kata. Kedua metode mencari saran kata menggunakan nilai probabilitas kata yang paling sering muncul di dalam kamus. Perbedaan dari kedua metode tersebut adalah Peter Norvig menggunakan algoritma yang mengkombinasikan proses menghapus, menambah, memisahkan, mengganti, dan memindahkan huruf pada kata yang salah. Sedangkan, *N*-Gram menggunakan algoritma yang memperhatikan kata-kata sebelum dan sesudahnya berdasarkan kalimat di dalam kamus. Kamus yang digunakan adalah dokumen ARPAyang merupakan hasil dari proses membangun model bahasa menggunakan SRILM. Kombinasi metode ini diuji dalam 9 skenario kesalahan penulisan dengan 160 kalimat yang masing-masing memiliki satu kata yang salah. Hasil pengujian menyatakan bahwa kombinasi kedua metode memberikan tingkat ketepatan 65,926% dan tingkat keberhasilan 78,07% untuk menghasilkan satu saran kata yang benar dari satu kata yang salah dalam sebuah kalimat. Kombinasi kedua metode ini dapat digunakan dalam memperbaiki kesalahan pengetikan, walaupun tidak dapat memperbaiki kata dengan tingkat kesalahan dua huruf atau lebih. Hal ini dikarenakan, Peter Norvig tidak mampu memperbaiki kata dengan tingkat kesalahan dua huruf dan membutuhkan korpus yang baik.**

***Kata kunci*: Kombinasi, *Spelling Corrector*, Peter Norvig, *N*-Gram, ARPA *file*.**

I. PENDAHULUAN

Spelling correction adalah proses mendeteksi dan memberikan saran untuk kata-kata yang salah eja di dalam suatu teks [1]. Sedangkan *spelling corrector* merupakan fitur atau aplikasi yang akan melakukan proses tersebut. Fitur ini mencari kata-kata yang salah berdasarkan data korpus yang digunakan aplikasi.Sedangkan saran kata diberikan dengan perhitungan algoritma yang juga digunakan oleh aplikasi.

Pemeriksa ejaan terbagi menjadi dua jenis, yaitu : pemeriksa kesalahan yang bukan kata dan pemeriksa kesalahan kata yang sebenarnya. Pemeriksa kesalahan yang bukan kata berfokus pada penanganan kata yang salah eja yang disebabkan oleh kesalahan tipografi. Sedangkan pemeriksa kesalahan kata yang sebenarnya ditekankan pada penanganan kesalahan penempatan kata dalam sebuah kalimat [2].

Beberapa penelitian tentang pemeriksa ejaan lebih banyak membahas kesalahan kata yang disebabkan oleh kesalahan tipografi. Referensi [3] memperbaiki kesalahan-kesalahan kata di dalam dokumen menggunakan metode pendekatan *Dictionary Lookup*, *N*—Gram dengan perhitungan *Cosine Similarity*, dan *Levensthein Distance*. Data yang digunakan sebagai data penentu adalah kamus Bahasa Indonesia, akibatnya metode-metode yang digunakan hanya memperhatikan huruf-huruf di dalam kata yang dianggap salah.

Pada penelitian [4], perbaikan ejaan kata di dalam dokumen menggunakan metode *N*-Gram dan *Cosine Similarity*. Metode *N*-Gram yang digunakan memperhatikan tiga kata atau disebut trigram. Kata-kata yang salah ditentukan dari kamus Bahasa Indonesia dan data latih. Data latih berupa kalimat-kalimat yang kemudian dipecah menjadi perkata, disertai kata sebelum dan kata setelahnya. *Cosine Similarity* menghitung nilai kesamaan antara kata yang salah, disertai kata sekitarnya, dengan kalimat target yang ada di data latih. Kata yang memiliki nilai kesamaan tertinggi menjadi saran kata.Hasil persentase perbaikan sangat minim karena kuantitas data latih yang sangat kurang untuk menggunakan metode *N*-Gram dengan memperhatikan tiga kata.

Penelitian [5] membandingkan metode Peter Norvig dan BK-Trees yang dibantu metode *Levensthein Distance*

dalam memberikan saran-saran kata untuk kata-kata yang di-*input*-kan melalui aplikasi.Kedua metode memproses satu per satu kata yang salah, namun menentukan saran kata tertinggi dengan cara yang berbeda. Aplikasi yang dibuat menggunakan korpus yang dipecah menjadi kamus. Hasil penelitian menyatakan bahwa nilai *precision*Peter Norvig lebih tinggi daripada BK-Trees.Sebaliknya, nilai *recall* Peter Norvig lebih rendah daripada BK-Trees.

Penelitian ini menggabungkan dua metode Peter Norvig dan *N*-Gram. Kedua metode diimplementasikan ke dalam sebuah aplikasi untuk mencari saran kata. Aplikasi tersebut menguji kesalahan satu kata di dalam kalimat*input*. Data yang digunakan adalah dokumen ARPA (*Advanced Research Project* Agency) yang berisi nilai-nilai probabilitas dari serangkaian *n* kata. Dokumen ARPA didapatkan dari hasil proses membangun *language model* dari suatu dokumen .txt atau disebut korpus menggunakan *script* SRILM.

Korpus didefinisikan sebagai koleksi atau sekumpulan contoh teks tulis atau lisan dalam bentuk data yang dapat dibaca dengan menggunakan seperangkat mesin dan dapat diberi catatan berupa berbagai bentuk informasi linguistik [6]. Korpus dimodifikasi terlebih dahulu untuk dapat membangun *language model* korpus tersebut.

Language model menetapkan probabilitas P($w1$, n) ke serangkaian n kata dengan means sebuah distribusi probabilitas. Rangkaian-rangkaian tersebut bisa berupa frase-frase atau kalimat-kalimat dan probabilitasnya dapat diperkirakan dari korpus dokumen-dokumen yang besar. Salah satu contoh pendekatan *language model* adalah n-gram model. Model bahasa n-gram merupakan jenis probalilistik *language model* untuk memprediksi item berikutnya dalam urutan tersebut dalam bentuk (n-1) [7].

Pengujian dilakukan dengan menghitung nilai *precision* dan *recall*. *Precision* dapat diartikan sebagai ketepatan atau kecocokan antara permintaan informasi dengan jawaban terhadap permintaan itu, sedangkan *recall* adalah kemampuan menemukan kembali informasi yang sudah tersimpan [8]. Hasil pengujian dapat dijadikan acuan sebagai pilihan membuat aplikasi yang membutuhkan fitur pengoreksian kata berbahasa Indonesia.

II. METODOLOGI PENELITIAN

A. *Alat Penelitian*

Alat penelitian yang digunakan untuk menunjukkan alur kerja metode adalah *Flowchart* atau Bagan Alir. *Flowchart* adalah penggambaran secara grafik dari langkah-langkah pemecahan masalah yang harus diikuti oleh pemroses [9].

B. *Perangkat Penelitian*

Perangkat penelitian yang digunakan dalam penelitian ini terdiri dari perangkat keras dan perangkat lunak.

1)Perangkat Keras

Satu unit laptop HP Intel® Core™ i3-3110M CPU @ 2,40 GHz

2)Perangkat Lunak

- Windows 7 Ultimate 64 bit SP1
- Linux Ubuntu 14.04 LTS 64 Bit
- SRILM untuk pemodelan bahasa
- Notepad++ untuk mengedit korpus
- Notepad IDE 8.2 untuk membuat aplikasi

C. *Data Penelitian*

Data penelitian adalah korpus bahasa Indonesia yang dikumpulkan dari artikel berita *online* dan penelitian-penelitian sebelumnya yang menggunakan korpus bahasa Indonesia. Korpus yang siap digunakan adalah korpus yang satu barisnya berisi satu kalimat. Korpus yang digunakan pada penelitian ini berjumlah 337.271 baris kalimat atau 170.420 kata unik.

D. *Pemodelan Bahasa*

Pemodelan bahasa adalah proses untuk menghasilkan ARPA *file*. Sebelum membangun model bahasa dengan menggunakan SRILM, korpus dimodifikasi terlebih dahulu untuk disesuaikan dengan format yang diperlukan dalam proses pembuatan model.

```
1. sudo /home/Ubuntu/moses/mosesdecoder/scripts/
   training/clean-corpus-n.perl txt korpus txt.clean 1 80
2. perl clean.plx txt.clean.korpus txt.tokenized.korpus
3. sudo /home/Ubuntu/moses/mosesdecoder/scripts/
   tokenizer/lowercase.perl< txt.tokenized.korpus >
   txt.lowercased.korpus
```

Gambar 1 Perintah Modifikasi Korpus

Selanjutnya, perintah untuk membangun model bahasa dapat dilihat pada Gambar 2.

```
sudo /home/ubuntu/moses/srilm/bin/i686/ngram-count
–order 3 –interpolate –unk –text txt.lowercased.korpus
–lm korpus.lm
```

Gambar 2 Perintah Membangun Model Bahasa

Berikut ini adalah contoh dokumen korpus.lm yang dihasilkan.

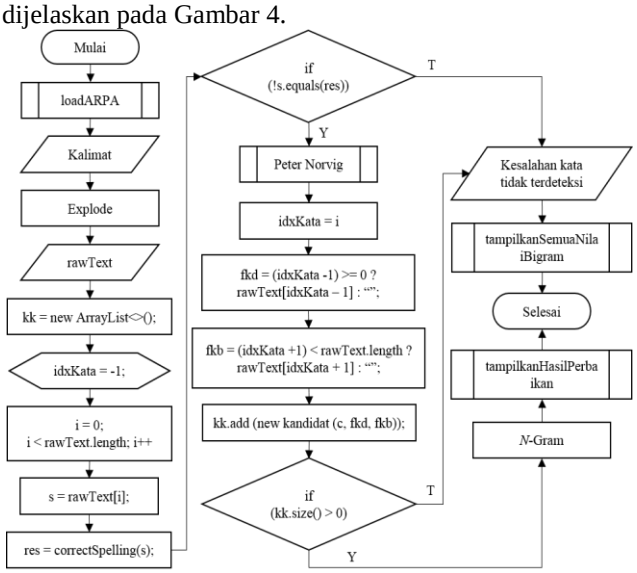
```
\data\
ngram 1=170420
ngram 2=1807115
ngram 3=766361

\1-grams:
-6.785976   --dari  -0.2381614
-6.484947   -amerika    -0.3452322
-6.785976   -amino--hidroksibutanoat    -0.213
(...)

\2-grams:
-2.860265   - akad
-2.860265   - akan
-2.860265   - akses
-2.860265   - aksesori
-2.171454   - al-farabi -0.1999283
(...)

\3-grams:
-0.9112481  adalah a </s>
-0.1961723  agama a menggunakan
```

Gambar 3 Contoh Dokumen ARPA

E. *Perancangan* Sistem

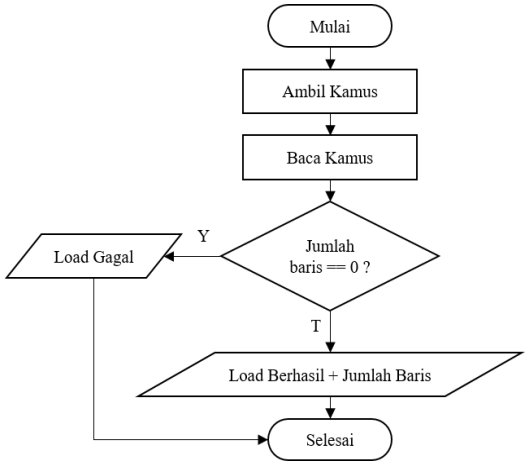
Perancangan sistem dilakukan dengan menggunakan *Flowchart.* Kemudian dilanjutkan dengan perancangan

antarmuka sistem. *Flowchart* keseluruhan aplikasi dijelaskan pada Gambar 4.



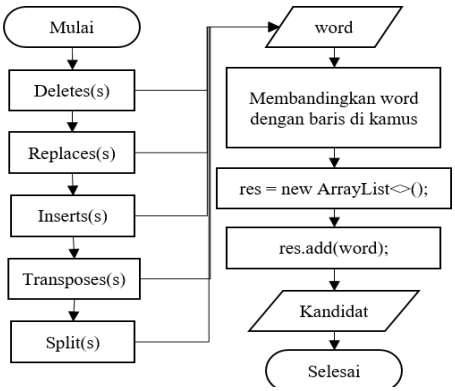
Gambar 4 *Flowchart* Aplikasi

Proses sistem mengambil dan menyimpan *file* kamus dijelaskan dalam Gambar 5 *Flowchart* loadARPA .



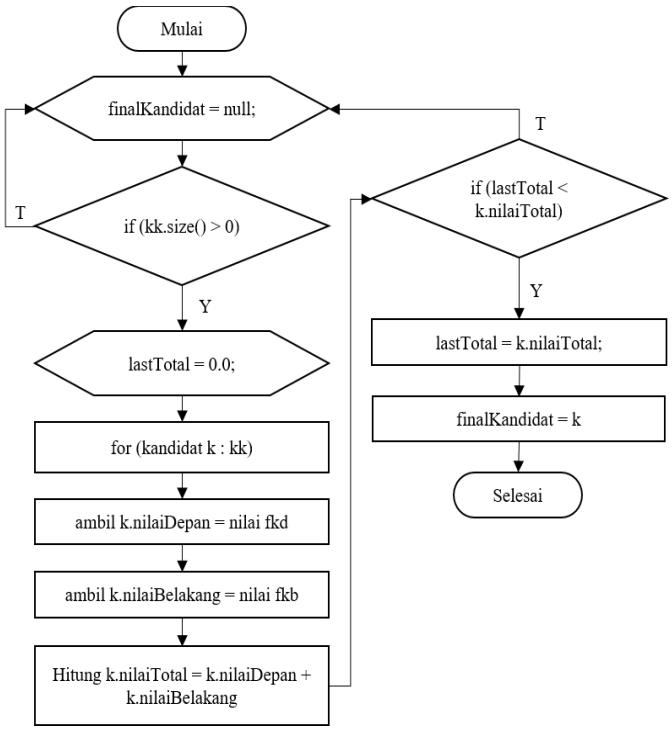
Gambar 5 *Flowchart* loadARPA

Langkah-langkah yang dikerjakan metode Peter Norvig dijelaskan dalam Gambar 6 *Flowchart* Peter Norvig.



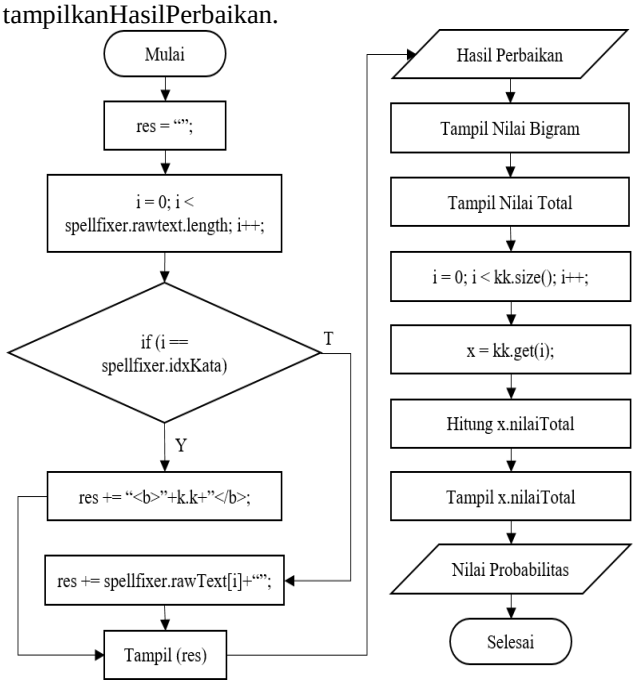
Gambar 6 *Flowchart* Peter Norvig

Langkah-langkah yang dikerjakan metode *N*-Gram dijelaskan dalam Gambar 7 *Flowchart N*-Gram.



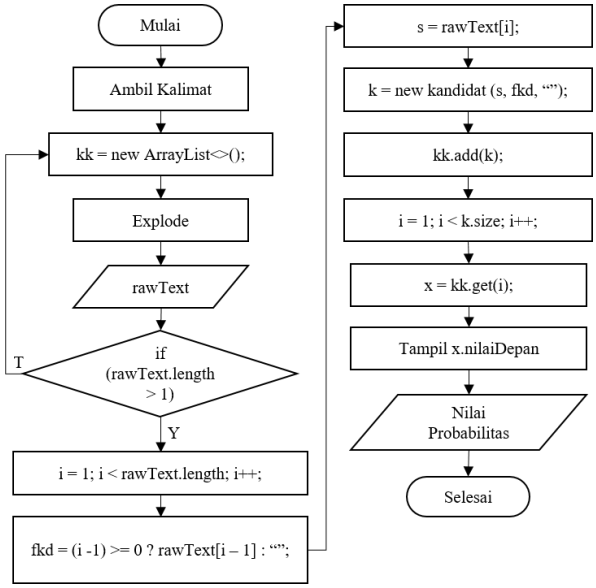
Gambar 7 *Flowchart N-Gram*

Langkah-langkah untuk menampilkan hasil perbaikan dijelaskan dalam Gambar 8 *Flowchart* tampilkanHasilPerbaikan.



Gambar 8 *Flowchart* Tampilkan Hasil Perbaikan

Langkah-langkah untuk menampilkan semua nilai bigram dijelaskan dalam Gambar 9 *Flowchart* tampilkan Semua Nilai Bigram.



Gambar 9 *Flowchart* Tampilkan Semua Nilai Bigram

F. Pembuatan Sistem

Aplikasi diprogram menggunakan bahasa pemrograman Java. Pembuatan sistem dilakukan berdasarkan perancangan yang telah dikerjakan pada tahapan sebelumnya. Pada awalnya, sistem harus membaca kamus untuk merekam jumlah baris kamus yang disediakan. Kalimat uji yang telah di-*input*-kan diproses dahulu oleh sistem untuk menemukan kata yang salah dengan membandingkan antara setiap kata uji dan setiap baris kata yang ada di kamus. Apabila kata uji salah, maka diproses oleh metode Peter Norvig.

Dalam prosesnya, sebuah kata dengan jumlah n huruf akan ada n penghapusan, n – 1 pemisahan, 26 (n+1) penambahan, n – 1 pemindahan, 26n penggantian, dengan total 54n + 25 proses [10]. Proses-proses ini diterapkan untuk semua huruf pada kata tersebut secara bergantian. Tiap satu langkah dalam masing-masing proses tersebut menghasilkan satu kata yang berbeda dari kata asal, kemudian kata tersebut diidentifikasi kembali. Kata-kata dari setiap langkah yang terdapat di dalam kamus disimpan sebagai kandidat-kandidat kata.

Kandidat-kandidat kata diproses oleh metode *N*-Gram. Metodeini mengambil potongan-potongan karakter kata sejumlah *n* dalam sebuah kalimat. *N*-Gramdibedakan berdasarkan jumlah pemenggalan, nilai *n* = 1 adalah unigrams atau monogram, *n* =2 adalah bigrams, *n* = 3 adalah trigrams, dan seterusnya.Persamaan umum *N*-Gramuntuk memperkirakan probabilitas urutan kata berikutnya adalah [11] :

$$P(w_n|w_1^{n-1}) \approx P((w_n|w_{n-N+1}^{n-1})$$

Contoh proses *N*-Gram dengan contoh kata ‘ini contoh kata’ sebagai berikut:

Unigrams : ini, contoh, kata
Bigrams : ini contoh, contoh kata
Trigrams : ini contoh kata

Penelitian ini menggunakan *n* = 2 atau bigram. Pada penelitian [12] menyatakan bahwa performansi sistem pengoreksian menggunakan implementasi metode bigram lebih tinggi dibandingkan menggunakan trigram. Penelitian tersebut mengoreksi ejaan kata dengan menggunaan metode bigram dan trigram. Penelitian [13] yang menggunakan *N*-Gram mendeteksi suatu kata merupakan bahasa Indonesia atau bukan bahasa Indonesia, juga menyatakan bahwa hasil yang cukup akurat dengan waktu identifikasi tidak terlalu lama adalah bigram.

Dari contoh yang diberikan, apabila “contoh” merupakan kandidat yang dihasilkan Peter Norvig, maka sistem mencari nilai probabilitas dari “ini contoh” dan “contoh kata” di dalam kamus. Selanjutnya, mencari nilai probabilitas dari kandidat-kandidat lainnya. Total nilai probabilitas yang tertinggi menjadi kata yang benar untuk kata uji.

G. Pengujian

Pengujian yang dimaksud adalah menguji kalimat-kalimat uji yang masing-masing terdapat satu kata yang salah. Pengujian dilakukan dengan 160 kata salah dalam 9 skenario kesalahan berikut ini:

1. Skenario pengujian kesalahan 1 huruf adalah mengganti 1 huruf di kata sebenarnya dengan huruf lainnya.
2. Skenario pengujian kekurangan 1 huruf adalah mengurangi 1 huruf sembarang tanpa mengubah huruf lainnya.
3. Skenario pengujian kelebihan 1 huruf adalah penambahan 1 huruf di sembarang tempat tanpa mengubah huruf lainnya.
4. Skenario pengujian kesalahan letak 2 huruf adalah memindahkan 2 huruf tanpa mengubah huruf dan jumlahnya.
5. Skenario pengujian kesalahan 2 huruf adalah mengganti 2 huruf dengan 2 huruf lainnya.
6. Skenario pengujian kekurangan 2 huruf adalah mengurangi 2 huruf sembarang tanpa mengubah huruf lainnya.
7. Skenario pengujian kelebihan 2 huruf adalah menambahkan 2 huruf di sembarang tempat tanpa mengubah huruf lainnya.
8. Skenario pengujian kesalahan 1 huruf dan kekurangan 1 huruf adalah mengganti 1 huruf dengan huruf lainnya dan mengurangi 1 huruf lainnya di sembarang tempat.
9. Skenario pengujian kesalahan 1 huruf dan kelebihan 1 huruf adalah mengganti 1 huruf dengan huruf lainnya dan menambahkan 1 huruf lainnya di sembarang tempat

Setiap hasil pengujian dikategorikan terlebih dahulu berdasarkan simbol-simbol yang digunakan oleh rumus *precision* dan *recall*, yaitu TP, TN, dan FN. Dalam pemeriksa ejaan, *precision* dapat digunakan untuk mengukur persentase metode memberikan saran kata yang relevan dari jumlah saran kata yang diberikan. Sedangkan *recall* mengukur persentase metode memberikan saran kata yang relevan dari jumlah kata relevan yang

sebenarnya. Secara umum, *precision* dan *recall* dirumuskan sebagai berikut:

$$Precision = \frac{TP}{TP+TN}\ x\ 100\%$$

$$Recall = \frac{TP}{TP+FN}\ x\ 100\%$$

TP = Jumlah data yang benar yang dihasilkan sistem
TN = Jumlah data yang tidak relevan yang dihasilkan
FN = Jumlah data yang relevan yang tidak dihasilkan
FP = Jumlah data yang tidak relevan dan tidak dihasilkan

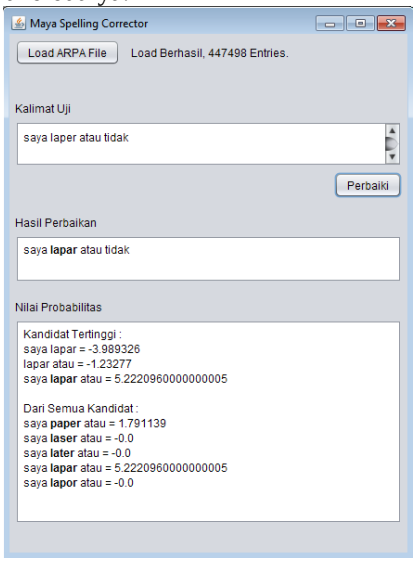
Apabila hasil yang diberikan adalah benar, maka dikategorikan sebagai TP. Apabila salah, maka dikategorikan sebagai TN. Dan apabila tidak dapat menghasilkan satu pun saran kata, maka dikategorikan sebagai FN.

III. PENGUJIAN DAN ANALSIS

A. *Hasil Perancangan*

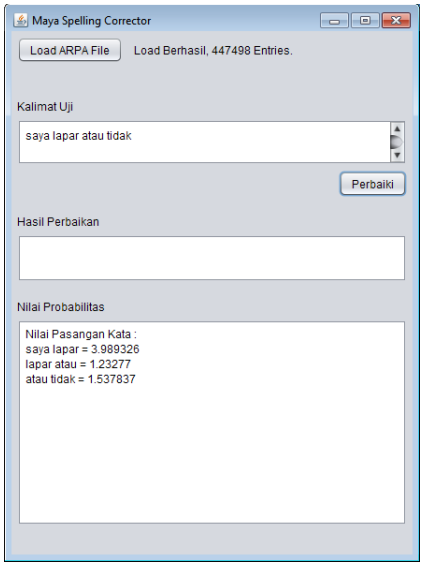
Pada aplikasi ini terdapat dua tombol, yaitu Load ARPA File dan Perbaiki. Tombol Load ARPA File berfungsi untuk mengambil file ARPA dan menyimpannya sebagai kamus. Sedangkan, tombol Perbaiki berfungsi untuk mengecek dan memperbaiki kalimat uji yang sudah dimasukkan di kolom yang sudah disediakan. Berikut ini adalah antarmuka aplikasi Spelling Corrector Peter Norvig dan N-Gram, serta contoh hasil aplikasi memperbaiki kalimat.

Pada Gambar 10 terdapat satu kata yang salah dalam sebuah kalimat. Kolom hasil perbaikan menampilkan kalimat yang sudah diperbaiki. Kolom nilai probabilitas menampilkan nilai bigram dari kata yang diperbaiki dan kandidat-kandidatnya.



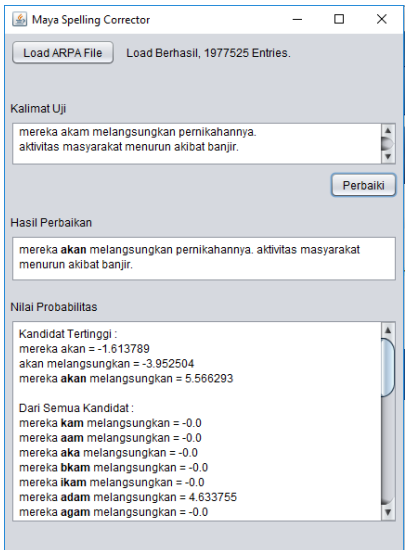
Gambar 10 Contoh Perbaiki Kata (1)

Namun, pada Gambar 11 menjelaskan bahwa, apabila tidak terdapat kata yang salah dalam kalimat uji, maka kolom hasil perbaikan tidak perlu menampilkan kalimat uji. Sedangkan, kolom nilai probabilitas menampilkan semua nilai bigram dari kata-kata yang terdapat di dalam kalimat uji.



Gambar 11 Contoh Perbaiki Kata (2)

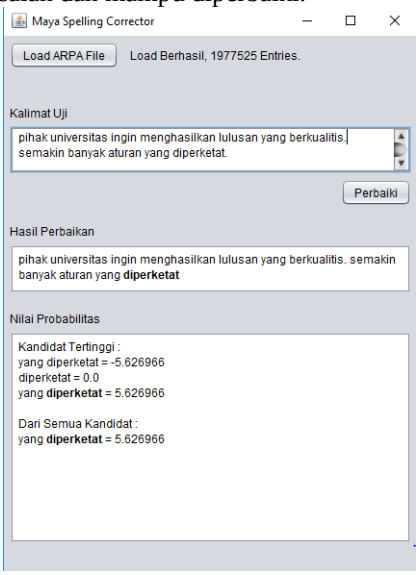
Pada Gambar 12, sistem dicoba untuk menguji satu kata yang salah dalam dua kalimat. Masing-masing kalimat dipisah dengan titik dan baris baru. Dan posisi kata yang salah berada ditengah kalimat. Sistem dapat mendeteksi kata yang salah dan memperbaikinya dengan benar. Namun, kalimat uji tidak ditampilkan dalam dua baris kalimat di kolom perbaikan. Hal ini dikarenakan, sistem dapat membaca titik, koma, baris baru, dan karakter-karakter sejenisnya, namun karakter tersebut dibaca sebagai satu kesatuan kata dari kata sebelum dan setelahnya.



Gambar 12 Contoh Perbaiki Kata (3)

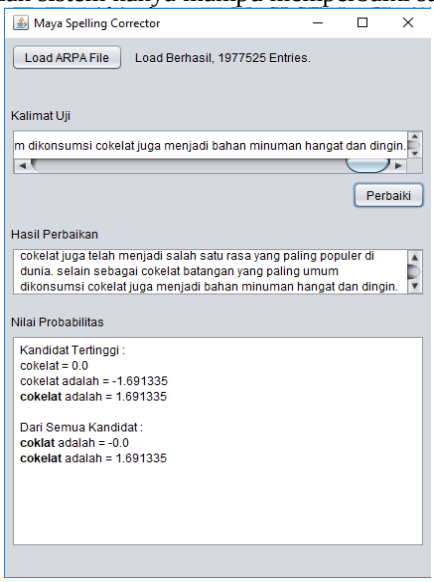
Pada Gambar 13 sistem dicoba memperbaiki kalimat uji dengan satu kata yang salah dalam dua kalimat, namun kata yang salah berada di akhir kalimat. Sistem tidak dapat memperbaiki kata yang sengaja disalahkan,

dikarenakan karakter titik dan baris baru dibaca sebagai satu kesatuan kata. Serta keterbatasan metode dalam memperbaiki kata dengan tingkat kesalahan di atas satu huruf. Akibatnya, sistem mendeteksi kata lain yang dianggap salah dan mampu diperbaiki.



Gambar 13 Contoh Perbaiki Kata (4)

Pada Gambar 14 sistem dicoba memperbaiki kalimat uji dalam bentuk satu paragraf dengan satu kata yang sengaja disalahkan. Sistem mampu mendeteksi kata yang sengaja disalahkan dan memperbaikinya dengan benar. Padahal kata yang terdapat titik juga dianggap salah oleh sistem, namun tidak diperbaiki. Hal ini dikarenakan, kata yang sengaja disalahkan berada sebelum kata salah yang lainnya, dan sistem hanya mampu memperbaiki satu kata.



Gambar 14 Contoh Perbaiki Kata (5)

B. *Analisis*

Berikut ini nilai pengujian *Precision* dan *Recall* dari masing-masing skenario yang dijelaskan dalam Tabel I.

TABEL I
Nilai Pengujian

No. SP	KU	TP	TN	FN	Precision	Recall
SP 1	40	29	9	2	76,316	93,548
SP 2	40	25	9	6	73,529	80,645
SP 3	40	29	6	5	82,857	85,294
SP 4	13	6	4	3	60,0	66,667
SP 5	3	0	2	1	0	0
SP 6	11	0	6	5	0	0
SP 7	3	0	3	0	0	0
SP 8	3	0	2	1	0	0
SP 9	7	0	5	2	0	0
TOTAL	**160**	**89**	**46**	**25**	**65,926**	**78,07**

Dari hasil pengujian dapat dilihat metode tidak mampu menghasilkan kata yang relevan di beberapa kasus kesalahan. Baik itu kata yang tidak tepat, maupun tidak menghasilkan satu pun kandidat. Bahkan metode tidak menghasilkan satu pun kata yang relevan di sebagian besar skenario pengujian dengan tingkat kesalahan 2 huruf. Hal ini bisa disebabkan oleh beberapa hal, yaitu :

1) Kata salah lainnya: Sistem mendeteksi kesalahan kata yang seharusnya tidak salah, dan posisi kata tersebut berada di sebelum kata yang sengaja disalahkan. Karena sistem sudah menemukan kata yang menurutnya salah, dan sistem hanya mampu memproses satu kata yang salah dalam satu kalimat uji, akibatnya kata yang sengaja disalahkan tidak terbaca. Contohnya : 'pantai lumukutan kini sudah menjadi obyek wisata berkat mapala teknik'. 'obyek' adalah kata yang salah, namun 'lumukutan' juga merupakan kata yang tidak terdapat didalam kamus. Sistem telah menemukan kata 'lumukutan' sebagai kata salah dan tidak mendeteksi kata setelahnya.

2) Kualitas korpus: Kata-kata dalam korpus masih ada yang keliru. Akibatnya, apabila kesalahan kata dalam kata uji terdapat juga di dalam kamus, sistem tidak akan merekam ada kata yang salah. sistem hanya mengeluarkan nilai bigram dari semua kata dalam kalimat uji.Contohnya : 'kesetaraan jender merupakan permainan untuk merusak tatanan kehidupan'. Kata 'jender' tidak diperbaiki karena ada di dalam kamus, walaupun kata tersebut salah.

3) Peter Norvig: Algoritma Peter Norvig adalah mengubah satu per satu huruf dalam sebuah kata. Saat satu langkah dijalankan, hasilnya akan diperiksa di kamus. Apabila tidak ada, langkah tersebut diulang kembali namun dengan huruf yang lain. Hasil dari langkah sebelumnya tidak ada disimpan atau digunakan. Akibatnya, metode Peter Norvig tidak mampu memperbaiki kata yang salah dengan tingkat kesalahan dua huruf. Contohnya : 'kita dibuat terus merangsek dalam menjalani hidup'. Kata 'merangsek' merupakan kata dengan kesalahan dua huruf. Sistem tidak dapat mengganggap kata tersebut benar dan juga tidak dapat memperbaikinya.

4) N-Gram: Algoritma ¬N-Gram mengambil nilai bigram dari semua kandidat kata yang dihasilkan Peter

Norvig. Nilai bigram yang lebih tinggi akan menjadi final kandidat. Namun, apabila tidak ada satu pun nilai bigram yang lebih dari nol, akibatnya ¬¬N-Gram tidak menghasilkan final kandidat dan tidak menampilkan kandidat-kandidat yang telah dihasilkan Peter Norvig.Contohnya : 'dia kehilangan kuci kotak harta karunnya'. Kata 'kuci' adalah kata salah yang tidak dapat diperbaiki, padahal kata tersebut hanya kekurangan 1 huruf. Setelah diperiksa di dalam korpus secara manual,tidak ada kalimat 'kehilangan kunci' atau 'kunci kotak' didalamnya.

5) Kuantitas korpus: Nilai probabilitas dari kandidat yang relevan lebih rendah dari kandidat lain. Akibatnya, metode tidak menghasilkan kata yang relevan.Contohnya : 'orang tua yang kerab membiarkan anaknya menonton tv sendirian'. Kata 'kerab' merupakan kesalahan kata dengan 1 huruf. Sistem tidak memberikan kata yang relevan, yaitu 'kerap'. Nilai probabilitas 'yang kerap membiarkan' lebih rendah dari 'yang keras membiarkan'.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan yang dapat diambil dalam penelitian ini antara lain :

1) Hasil kombinasi metode Spelling Corrector Peter Norvig dan N-Gram pada kalimat Bahasa Indonesia memberikan 65,926% tingkat ketepatan menemukan satu saran kata.

2) Hasil kombinasi metode Spelling Corrector Peter Norvig dan N-Gram pada kalimat Bahasa Indonesia memberikan 78,07 tingkat keberhasilan menemukan satu saran kata.

3) Kombinasi kedua metode ini dapat digunakan dalam memperbaiki kesalahan pengetikan, walaupun tidak dapat memperbaiki kata dengan tingkat kesalahan dua huruf atau lebih.

4) Kombinasi kedua metode ini sangat bergantung pada korpus yang digunakan. Semakin baik korpus yang digunakan, maka semakin baik pula sistem menemukan kata yang relevan.

B. Saran

Saran untuk pengembangan penelitian ini antara lain :

1) Kombinasi metode ini dikembangkan untuk memperbaiki dua atau lebih kata yang salah.

2) Kombinasi metode ini dikembangkan untuk dapat menggunakan nilai trigram dari kalimat uji, sehingga mampu memperbaiki kesalahan kata dengan memperhatikan frasa-frasa kalimat.

3) Aplikasi ini sangat bergantung pada kualitas dan kuantitas korpus sehingga diperlukan aplikasi yang dapat menghasilkan korpus dengan kualitas yang baik dan kuantitas yang tinggi.

REFERENSI

[1] Mishra, Ritika dan Navjot Kaur. 2013. *A Survey of Spelling Error Detection and Correction Techniques*. International Journal of Computer Trends and Technology Vol. 4, Issue 3.

[2] Ratnasari, C. Indah. 2017. *A Non-Word Error Spell Checker for Patient Complaints in Bahasa Indonesia*. International Journal of Information Technology, Computer Science and Open Source Vol . 1, No. 1.

[3] Fahma, A. Indana. 2018. *Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-Gram dan Levensthein Distance*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 1.

[4] Fachrurrozi, Muhammad. 2015. *Perbaikan Ejaan Kata pada Dokumen Bahasa Indonesia dengan Metode Cosine Similarity*. Jurnal. Palembang: Universitas Sriwijaya.

[5] Mutammimah. 2017. *Analisis Perbandingan Metode Spelling Corrector Peter Norvig dan Spelling Checker BK-Trees pada Kata Berbahasa Indonesia*. Jurnal Edukasi dan Penelitian Informatika (JEPIN) Vol 5, No. 1.

[6] Mandira, Soni. 2016. *Perbaikan Probabilitas Lexical Model untuk Meningkatkan Akurasi Mesin Penerjemah Statistik*. Jurnal Edukasi dan Penelitian Informatika (JEPIN) Vol 2, No. 1.

[7] Hadi, Ibnu. 2014. *Uji Akurasi Mesin Penerjemah Statistik Bahasa Indonesia ke Bahasa Melayu Sambas dan Mesin Penerjemahan Statistik Bahasa Melayu Sambas ke Bahasa Indonesia*. Jurnal Sistem dan Teknologi Informasi (JUSTIN) Vol 2, No. 3.

[8] Suprapto, Kadarisman Tejo Yuwono, Totok Sukardiyono, dan Adi Dewanto. 2008. *Buku Bahasa Pemrograman Untuk SMK*. Departemen Pendidikan Nasional : Direktorat Pembinaan Sekolah Menengah Kejuruan

[9] Pendit, P. L. 2008. *Perpustakaan Digital dari A sampai Z.* Jakarta : Cita Karyakarsa Mandiri

[10] Norvig, Peter. 2007. How to Write a Spelling Corrector. [Online] Available: http://www.norvig.com/spell-correct.html

[11] Jurafsky, D. Saul. dan James H. Martin. 1999. *Speech and Language Processing*. USA : Library of Congress Cataloging in Publication Data

[12] Wardhana, W. Satya. 2011. *Pengoreksian Ejaan Kata Menggunakan Metode N-Gram (Studi Kasus: Dokumen Teks Berbahasa Indonesia)*. Jurnal. Bandung: Universitas Telkom

[13] Hamzah, Amir. 2010. *Deteksi Bahasa untuk Dokumen Teks Berbahasa Indonesia*. Yogyakarta: Seminar Nasional Informatika (semnasIF) 2010.