JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika

[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]

IMPLEMENTASI *SPELLING CORRECTOR* UNTUK MENGATASI *TYPOGRAPHICAL ERROR* PADA FITUR PENCARIAN APLIKASI KAMUS ISTILAH INFORMATIKA

Dian Markuci [a,1], Cahyo Prianto [b,2*], Syafrial Fachri Pane [c,3]
[a,b,c] Universitas Logistik Bisnis Internasional, Jalan Sariasih No. 54 Bandung
[1] dianmarkucii@gmail.com; *[2] cahyo@ulbi.ac.id; [3] syafrial.fachri@ulbi.ac.id
* corresponding author

ARTICLE INFO

Keywords

Search
Spelling Corrector
Flask
Probability
Dictionary

ABSTRACT

Information needs can arise because of a knowledge gap in a person with the necessary information needs, one of which is knowledge in the field of computers and informatics, especially related to terms in the computer field. Therefore we need a system that makes it easy for users to meet the information needs needed by building a digital dictionary application related to computer terms and informatics by utilizing the search engine features in it. Search activities are carried out daily to meet information needs. However, an error that is often unavoidable in performing a search is a typing error in the query. As a result, the information sought is not as expected. Based on this, we need a system that can identify typographical errors in the search text. So in this research, a website-based dictionary of computer and informatics terms will be developed by applying Peter Norvig's spelling corrector using the Python language with the flask framework. The implementation results show that Peter Norvig's spelling corrector method can be applied to computer and informatics term dictionary applications. This can be seen at the level of accuracy reaching 89% in correcting 180 word variations that contain typographical errors based on the highest probability of each possible word in the corpus. However, there is a lack of this spelling corrector method, it is still difficult to overcome typos in spelling abbreviations and typographical errors that exceed 1 letter

1. Pendahuluan

Pencarian adalah suatu proses yang dilakukan seseorang untuk memenuhi kebutuhan informasi. Adanya kesenjangan pengetahuan yang ada dalam diri seseorang memunculkan berbagai kebutuhan informasi, salah satunya pengetahuan di bidang komputer dan informatika khususnya kebutuhan informasi terkait istilah pada bidang komputer dan informatika. Sampai saat ini masih banyak kamus istilah yang berbentuk buku dan dokumen sehingga dalam menemukan istilah, pencarian dilakukan dengan membuka lembaran buku dan melihat urutan berdasarkan abjad[1]. Maka, diperlukan suatu mekanisme terkomputerisasi yang memudahkan pengguna dalam memenuhi kebutuhan informasi yang diperlukan yaitu dengan membangun aplikasi kamus digital terkait istilah informatika dengan memanfaatkan fitur *search engine* didalamnya[2]. Terdapat kendala yang seringkali tidak dapat dihindari saat melakukan proses pencarian yaitu kesalahan pengetikan *query* atau *typographical error*, *Typographical error* adalah kesalahan pada penulisan kata yang mungkin terjadi ketika pengguna sedang menggunakan *keyboard* komputer ataupun *smartphone* [3], akibatnya informasi yang dicari tidak sesuai yang diharapkan. Berdasarkan hal tersebut, diperlukan suatu sistem yang dapat

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]

mengidentifikasi *typographical error* pada teks pencarian dan dapat melakukan proses koreksi terhadap *typographical error*.

Spelling Corrector adalah fitur koreksi otomatis untuk melakukan proses deteksi kesalahan kata dan pemberian saran kata yang mengalami kesalahan eja pada suatu teks [4][5]. Penelitian terkait sebelumnya berjudul Analisis Perbandingan Metode *Spelling corrector* Peter Norvig dan Spelling Checker BK-Trees pada Kata Berbahasa Indonesia membahas perbandingan antara dua metode *spelling corrector* yang diterapkan pada kata bahasa indonesia berdasarkan data *corpus* berita *online*. Hasil dari penelitian menunjukkan bahwa metode *spelling corrector* Peter Norvig mampu menghasilkan 52,8% tingkat ketepatan yang lebih unggul dibandingkan metode BK-Trees dengan hasil 9%. Namun, metode BK-Trees lebih baik pada tingkat keberhasilan 100% menghasilkan sugesti kata secara cepat dibandingkan metode Peter Norvig [6]. Selanjutnya penelitian berjudul Aplikasi Pendeteksi Kesalahan Ejaan Bahasa Indonesia pada Karya Ilmiah Bidang Ilmu Komputer Menggunakan KEBI 1.0 Checker mengimplementasikan metode *spelling corrector* Peter Norvig untuk mendeteksi dan mengoreksi dokumen yang mengandung kata tidak baku dan kesalahan pengetikan kata berdasarkan data KBBI. Hasil penelitian tersebut menunjukkan aplikasi sudah memberi kinerja akurasi terbaik dalam mengoreksi kesalahan mengetik dan kata tidak baku masing-masing 49,10% dan 100% [7]. Penelitian berjudul *Improving Topical Social Media Sentiment Analysis by Correcting Unknown Words Automatically* meneliti efektivitas menerapkan algoritma koreksi ejaan algoritma levenshtein dan algoritma Peter Norvig, hasil persentase polaritas yang cocok adalah 81,60% dan 82,00% masing-masing untuk Levenshtein dan Algoritma Peter Norvig. hasil yang didapat, ada sedikit peningkatan dalam hal persentase polaritas yang cocok, di mana Peningkatan 1,6% dengan menggunakan algoritma Levenshtein distance dan 2,0% perbaikan dengan menggunakan algoritma Peter Norvig[8]

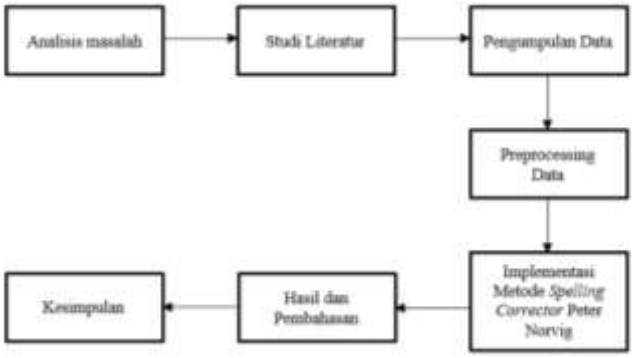
Dari penelitian-penelitian terkait sebelumnya dapat diketahui bahwa algoritma *spelling corrector* Peter Norvig memiliki tingkat ketepatan yang baik dalam mencari saran atau sugesti kata dalam koreksi ejaan otomatis sehingga pada penelitian ini peneliti memilih algoritma *spelling corrector* Peter Norvig untuk di implementasikan pada pencarian istilah informatika dan menganalisis cara kerja metode dalam melakukan proses identifikasi kesalahan ketik pada pencarian istilah. Maka dari permasalahan-permasalahan yang ada dapat di ambil tujuan penelitian ini adalah membuat aplikasi kamus istilah informatika berbasis *website* dengan penerapan *spelling corrector* pada fitur pencariannya diharapkan dapat mempermudah dan meningkatkan kinerja layanan yang lebih baik dan mempermudah pengguna dalam mengakses informasi.

2. Metodologi Penelitian

Berikut tahap-tahap proses yang dilakukan dalam penelitian ini di gambarkan pada diagram alir yang ditunjukkan gambar 1.1 dibawah ini:

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

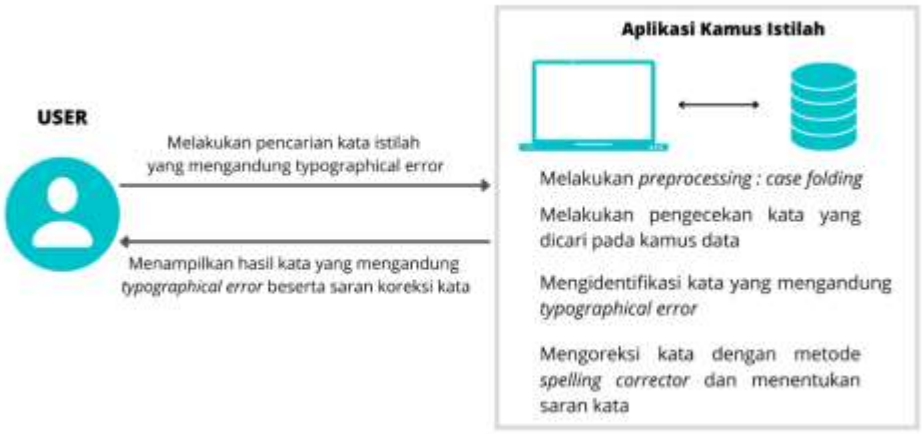
JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]



Gambar 1. 1 Metodologi Penelitian

2.1. Analisis Masalah

Tahapan analisis dimulai dengan menguraikan sebab dan akibat dari permasalahan yang ada. Pada penelitian yang berjudul " Implementasi *Spelling Corrector* untuk Mengatasi *Typographical Error* pada Fitur Pencarian Aplikasi Kamus Informatika" memiliki permasalahan pada kesalahan pengejaan dalam mencari istilah informatika, sehingga diperlukan suatu fitur yang dapat mengatasi permasalahan tersebut. Setelah melakukan analisis, dibuat perancangan struktur pogram dan detail konten yang disiapkan untuk pengembangan lebih lanjut. Hasil arsitektur diagram aplikasi kamus istilah komputer dan informatika dapat dilihat pada gambar 1.2 :



Gambar 1. 2 Arsitektur Diagram Kamus Istilah [5]

Pengembangan kamus istilah komputer dan informatika akan dibangun menggunakan bahasa pemrograman python dan *framework flask. Flask* merupakan *framework* aplikasi web WSGI yang ringan dan dirancang untuk membangun *project* secara mudah dan cepat. Flask tidak menerapkan dependensi. *Library* dan alat dipilih sesuai kebutuhan. Selain itu terdapat banyak ekstensi yang disediakan oleh komunitas yang memudahkan penambahan fungsionalitas baru [5][9][10]. Maka penulis memilih *framework flask* untuk membangun visualisasi kamus istilah komputer dan informatika.

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]

2.2 Studi Literatur

Tahapan studi literatur menambah referensi penelitian dari berbagai sumber terpercaya seperti buku referensi, jurnal, konferensi paper yang memiliki keterkaitan topik penelitian yang akan atau sedang dikerjakan.Tahapan ini menambah atau memperkaya referensi penelitian dari berbagai sumber terpercaya, seperti buku referensi, jurnal, dan makalah konferensi yang memiliki keterkaitan dengan topik penelitian yang dilakukan. [11]

2.3 Pengumpulan Data

Data sangat diperlukan dalam melakukan suatu analisis [12]. Untuk mendapatkan analisis yang baik dibutuhkan teori konsep dasar dan alat bantu yang memadai . Data yang digunakan dalam penelitian ini berasal dari karya Edward Erik yang berjudul "Kamus Populer Istilah Komputer dan Informatika"

2.4 *Preprocessing* Data

Preprocessing data dilakukan untuk mempersiapkan data mentah atau data yang belum bisa siap dianalisis hingga siap untuk dianalisis sesuai dengan metode yang digunakan [13]. Preprocesing pada penelitian ini menggunakan teknik *case folding*. *Case folding* merupakan teknik *preprocessing* data untuk mengubah huruf besar ke huruf kecil secara keseluruhan pada dataset atau bisa disebut *lowercase*[3]. Dalam bahasa pemrograman python sudah tersedia *library* dari python yaitu *lower*[14].

2.5 Implementasi Metode *Spelling Corrector* Peter Norvig

Metode Peter Norvig adalah metode *spelling corrector* sederhana menggunakan pendekatan *teorema bayes*[15]. *spelling corrector* Peter Norvig adalah suatu metode yang dapat mengubah kata yang memiliki *typographical error* atau kesalahan eja, kemudian dilakukan pencarian kandidat-kandidat kata yang dianggap benar dengan beberapa operasi pengubahan bentuk kata dari suatu bentuk satu ke bentuk yang lain. Cara kerja metode *spelling corrector* Peter Norvig menggunakan beberapa teori probabilitas. Dengan *teorema bayes* maka secara matematis setara dengan pernyataan berikut [16][17]:

$$\text{argmax}_{\text{c kandidat}}\ P(c)P(w|c)/P(w) \quad (1)$$

Selanjutnya karena P(w) sama untuk setiap kandidat (c), maka dapat difaktorkan kembali menjadi seperti berikut :

$$\text{argmax}_{\text{c kandidat}}\ P(c)P(w|c) \quad (2)$$

Penjelasan dari persamaan diatas :

- Argmaxc merupakan mekanisme kontrol yang dilakukan untuk perhitungan nilai layak c dan selanjutnya menentukan pilihan salah satu yang memberi nilai probabilitas tertinggi.
- P (c), P merupakan probabilitas bahwa koreksi yang diusulkan (c) disebut *language model*
- P (w | c), P merupakan probabilitas, w merupakan teks yang akan dituliskan atau diketik. Sedangkan c disini merupakan model *error.*

Dalam menetukan kandidat koreksi kata terbaik dipilih berdasarkan nilai probabilitas tertinggi, untuk menghitung probabilitas dapat dilihat rumus berikut :

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]

$$P(A) = \frac{n(A)}{n(S)} \quad (3)$$

Keterangan :
P(A) = Probabilitas terjadinya kejadian A
n(A) = Banyaknya kejadian A
n(S) = Jumlah total kejadian sample

2.6 Hasil dan Pembahasan

Tahap ini merupakan hasil implementasi dan pengujian sistem yang sudah dibangun. Pengujian dilakukan dengan perhitungan akurasi metode yang diterapkan pada sistem dengan tujuan mengetahuin keakuratan nilai yang direkomendasikan oleh sistem. Perhitungan akurasi dapat menggunakan persamaan/rumus dibawah ini[18]:

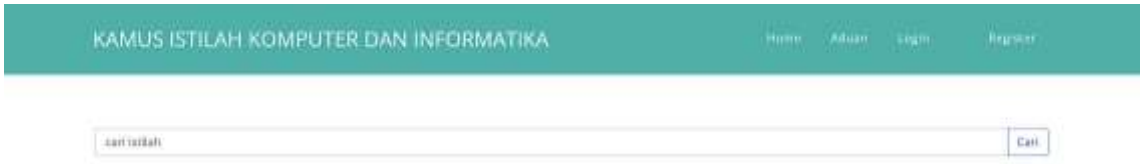
$$\text{Akurasi} = \frac{\text{Jumlah Identifikasi Benar}}{\text{Jumlah Data}} \times 100\% \quad (4)$$

3. Hasil dan Pembahasan

Sistem yang dirancang pada penelitian ini merupakan kamus digital istilah yang di kembangkan menggunakan bahasa pemrograman *python* dan *framework flask*. *Spelling corrector* di implementasikan pada fitur pencarian aplikasi kamus. Sistem yang dirancang pada penelitian ini terdiri dari 4 tahap diantaranya [19]

1. Memasukkan kata istilah informatika yang akan dicari
2. Mengidentifikasi *typographical* atau kesalahan ketik
3. Memberikan rekomendasi atau sugesti kata yang sudah diperbaiki berdasarkan probabilitas tertinggi yang ada pada kamus data *corpus*
4. Menampilkan pengertian atau informasi dari kata yang dicari

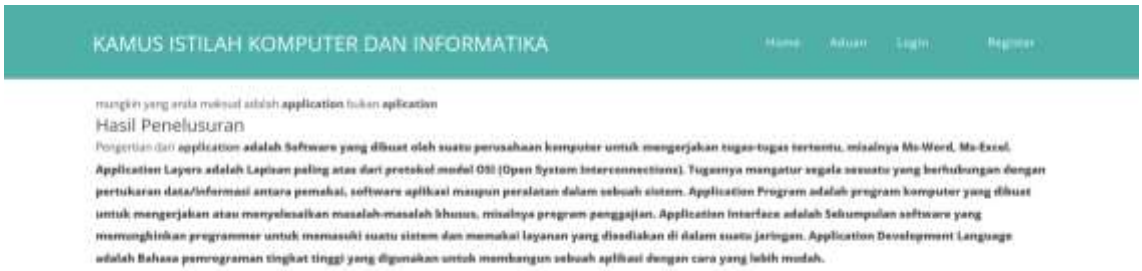
Berikut gambar 1.3 merupakan hasil tampilan antarmuka *website* kamus istilah komputer dan informatika.



Gambar 1. 3 Antarmuka Kamus Istilah

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]



Gambar 1. 4 Antarmuka Hasil Pencarian dengan *Spelling Corrector*

Penerapan *spelling corrector* dapat dilihat pada gambar 1.4 dimana terdapat hasil pencarian istilah beserta saran kata, pengguna bisa melakukan pencarian istilah di *form* pencarian seperti yang dapat dilihat pada gambar 1.3 kemudian hasilnya akan ditampilkan dibawah *form input* pencarian. hasil penelusuran akan menampilkan kemungkinan *typographical error* pengguna dan sugesti atau saran kata terdekat yang dicari pengguna.

3.1 Implementasi Metode *Spelling Corrector* Peter Norvig

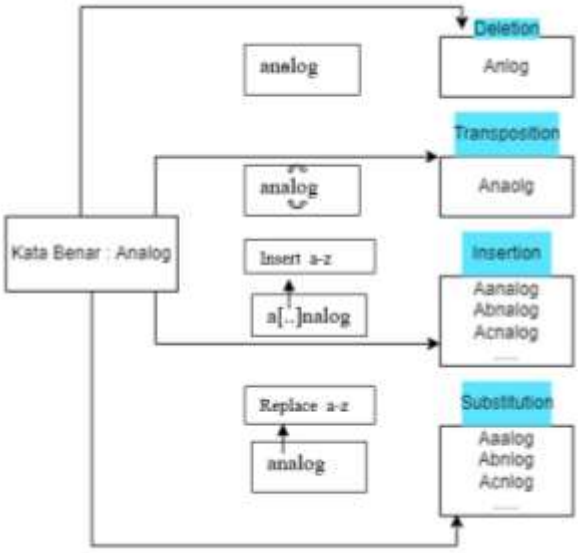
Spelling corrector di implementasikan di fitur pencarian kata istilah komputer dan informatika. Cara kerja metode *spelling corrector* Peter Norvig dapat di jelaskan menjadi 4 tahapan seperti yang sudah dijelaskan di metodologi penelitian bahwa *spelling corrector* Peter Norvig menggunakan pendekatan teorema bayesian, 4 tahapan tersebut diantaranya :

1. Selection mechanism (argmax)
   Dalam bahasa pemrograman *python* Argmax merupakan argument of maxima, dalam memilih kandidat akan dipilih, tahap ini akan memilih kandidat dengan probabilitas gabungan tertinggi

2. Candidate Model (c kandidat)
   Tahapan untuk mempertimbangkan kandidat c. Metode *Spelling corrector* Dalam mencari *candidates* kata menggunakan empat operasi yaitu *deletion, replacement, transpose,* dan *insertion*.

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]



Gambar 1. 5 Operasi *Edit Distance* untuk Pencarian Kandidat Kata

- Fungsi hapus atau *delete* yang mengambil *string* sebagai *input* dan mengembalikan daftar *string* dengan satu karakter dihapus. Misalnya, jika memanggil hapus('word') pada input 'word', fungsi akan mengembalikan ['ord', 'wrd', 'wod', 'wor', 'wr']
- Fungsi tukar yang mengambil *string* sebagai input dan mengembalikan daftar *string* dengan dua karakter yang berdekatan ditukar. Misalnya, jika kita memanggil tukar('word') pada input 'word', fungsi akan mengembalikan ['wodr', 'word', 'wordd', 'worw', 'wor']
- Fungsi ganti atau *replace* yang mengambil string sebagai input dan mengembalikan daftar *string* dengan satu karakter diganti dengan yang lain. Misalnya, jika memanggil ganti('word') pada *input* 'word', fungsi akan mengembalikan ['oord', 'wrod', 'wodr', 'worf', 'word']
- Fungsi tambah atau *insert* yang mengambil *string* sebagai *input* dan mengembalikan daftar *string* dengan satu karakter ditambahkan. Misalnya, jika memanggil tambah('word') pada *input* 'word', fungsi akan mengembalikan ['woad', 'woed', 'woid', 'woud', 'worq', 'worda', 'wordb ', 'wordc', 'wordd', 'worde', 'wordf', 'wordg', 'wordh', 'wordi', 'wordj', 'wordk', 'wordl', 'wordm', 'wordn', 'wordo', 'wordp', 'wordq', 'wordr', 'words', 'wordt', 'wordu', 'wordv', 'wordw', 'wordx', 'wordy', 'wordz']
- Dari 4 operasi *edit distance* dibuat fungsi edit_1 yang mengambil *string* sebagai *input* dan mengembalikan satu set semua *string* yang berjarak satu edit dari *string input*. Misalnya, jika kita memanggil edit_1('word') pada input 'word', fungsi akan mengembalikan set {'ord', 'wrd', 'wod', 'wor', 'wr', 'oad', 'oed', 'oid', 'oud', 'orq', 'orda', 'ordb', 'ordc', 'ordd', 'orde', 'ordf', 'ordg', 'orh', 'ordi ', 'ordj', 'ordk', 'ordl', 'ordm', 'ordn', 'ordo', 'ordp', 'ordq', 'ord', 'ordt', 'ordu', 'ordv', 'ordw', 'ordx', 'ordy', 'ordz', 'woad', 'woed', 'woid', 'woud', 'worq', 'worda', 'wordb
- Selain itu membuat fungsi edit_2 yang memanggil untuk fungsi pembantu lainnya.

3. *Language Model*

Membuat *language model* dari *corpus* yang berisi kumpulan kata kata terkait istilah informatika. Data *corpus* sementara terdapat 6530 kata. Namun sebagai catatan data *corpus* masih dapat terus bertambah tergantung dari penambahan kata istilah baru yang terdapat pada kamus istilah. Berikut merupakan contoh perhitungan probabilitas kata, dimana diketahui

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]

banyak kata *file* yang muncul pada kamus data *corpus* adalah 59 kata dari keseluruhan kemunculan kosakata yaitu 6530 kata, maka untuk menghitung probabilitas kata *file* adalah sebagai berikut :

P('file', N = 6530) = 59 / 6530
P('file', N = 6530) = 0.00903522205206738

Keterangan :
P = Probabilitas
N = Jumlah Kata

4. *Model Error*
Probabilitas dari koreksi kata baru. Memeriksa apakah ada kata-kata pada fungsi edit_1 atau edit_2 dalam kosakata. Jika ada, maka sistem mengembalikan tersebut. Jika tidak ada kata dari keduanya, maka akan mengembalikan daftar dengan kata asli sebelumnya.

3.2 Mekanisme *Spelling Corrector*

```
kata = baca_corpus("dataistilah.txt")
checker = KoreksiOtomatis("dataistilah.txt")
hitung_kata = Counter(kata)
kosakata = set(kata)
total_hitung_kata = float(sum(hitung_kata.values()))
prob = {kata: hitung_kata[kata] / total_hitung_kata for kata in hitung_kata.keys()}
if request.method == 'POST':
    kalimat = request.form['text']
    word = LIST(kalimat)
    if len(word)==1:
        kata = word[0].lower()
        koreksi = koreksi_kata(kata, kosakata, prob)
        if koreksi:
            probs = np.array([c[1] for c in koreksi])
            best_ix = np.argmax(probs)
            kor_kata = koreksi[best_ix][0]
            penjelasan = get_penjelasan(df, kor_kata)
            return render_template('result.html',salah=kata.lower(),revisi=kor_kata.lower(),penjelasan=penjelasan)
```

Gambar 1. 6 *Pseudocode* Mekanisme *Spelling Corrector*

Mekanisme koreksi kata dimulai membaca *corpus* dan menyimpan kata-kata dalam daftar. Kemudian, membuat satu set kata-kata unik dari *corpus*. Selanjutnya, menghitung berapa kali kemunculan kata di *corpus*. Kemudian, menghitung probabilitas setiap kata dalam *corpus*. Terakhir, memeriksa apakah pengguna telah memasukkan kata. Jika ya, ia akan memanggil fungsi koreksi_kata() untuk mendapatkan kata yang benar. Berikut ini gambar 1.7 merupakan fungsi koreksi kata :

```
def koreksi_kata(kata, kosakata, prob):
    if kata in kosakata:
        kor_kata = kata
        penjelasan = get_penjelasan(df, kor_kata)

    saran = edit_1(kata) or edit_2(kata) or [kata]
    kemungkinan = [w for w in saran if w in kosakata]
    return [(w, prob[w]) for w in kemungkinan]
```

Gambar 1. 7 *Pseudocode* Koreksi Kata

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 17 No 1]
[Januari 2023]

Gambar 1.7 menjelaskan Pertama-tama, memeriksa apakah kata yang dicari ada dalam kamus. Jika ya, maka sistem akan mengembalikan kata apa adanya. Jika kata tersebut tidak terdapat dalam kamus, maka dilakukan pencarian kata terdekat dalam kamus menggunakan operasi *edit distance*. Jika tidak ada kata dalam kamus yang mendekati kata yang diberikan, maka akan mengembalikan kata aslinya. Jika ada kata dalam kamus yang dekat dengan kata yang diberikan, maka akan mengembalikan kata saran yang paling mungkin.

3.3 Pengujian

Pengujian metode *spelling corrector* dilakukan dengan inputan kata pencarian dari 180 variasi kata yang mengandung *typographical error*. Berikut tabel hasil pengujian metode *spelling corrector* untuk mengatasi *typographical error*

Tabel 1. 1 Pengujian *Spelling Corrector*

Kata Typo	**Kata Benar**	**Hasil**
Abo	Abi	Inaccurate
Abned	Abend	Accurate
Abiwor	Abiword	Accurate
Abortt	Abort	Accurate
Acces	Access	Accurate
AND	ADC	Inaccurate
Aplication	Application	Accurate
Aray	Array	Accurate
DSL	ADSL	Inaccurate
Accessibillity	Accessibility	Accurate
Accessoris	Accessories	Accurate
Aknowledge	Acknowledge	Accurate
……	…….	…….

Dapat dilihat pada tabel 1.1 merupakan tabel pengujian metode *spelling corrector* Peter Norvig pada sistem menggunakan 180 variasi kesalahan kata istilah komputer. Batasan untuk *spelling corrector* Peter Norvig hanya untuk *typographical error* tidak melebihi 1 huruf maka variasi kesalahan kata istilah yang ada pada tabel tidak melebihi batasan yang ada. Nilai akurasi yang didapatkan 89% dalam mengoreksi kata pada pencarian di aplikasi kamus istilah informatika. dimana terdapat 161 merupakan kata yang dikoreksi dengan benar dan 19 kesalahan kata yang tidak dapat dikoreksi dengan benar, dari 19 kata tersebut banyak merupakan istilah komputer yang berupa singkatan seperti ABI(*Application Binary Interface*), ADSL(*Asymmetric Digital Subscriber Line*), API (*Application Programming Interface)*. Sehingga dapat dihitung nilai akurasi yang diperoleh dari pengujian 180 variasi kata *typographical error* menggunakan metode *spelling corrector* Peter Norvig.

4. Kesimpulan

Setelah melakukan implementasi dan pengujian pada Aplikasi kamus istilah informatika yang dibangun dengan bahasa *python* dan *framework flask*. Dapat diambil kesimpulan bahwa *framework flask* merupakan *framework* yang tepat dalam membangun visualisasi *website* kamus istilah informatika karena penerapannya yang tidak rumit. *Spelling corrector* yang diterapkan pada fitur pencarian aplikasi mampu mendeteksi dan mengoreksi kata yang mengandung *typographical error* menggunakan metode *spelling corrector* Peter Norvig. Hal ini di lihat pada tingkat ketepatannya dalam mengoreksi kata dan memberi sugesti yang akurat dengan akurasi mencapai 89% mengoreksi kata berdasarkan probabilitas tertinggi dari setiap kemungkinan kata yang ada pada *corpus*. Namun terdapat kekurangan metode

https://doi.org/10.47111/JTI
Available online at https://e-journal.upr.ac.id/index.php/JTI

spelling corrector yang ditemukan dalam penelitian ini yaitu *spelling corrector* Peter Norvig masih kesulitan mengidentifikasi *typographical error* melebihi 1 huruf dan metode ini masih kesulitan dalam menangani *typographical error* pada kata singkatan istilah komputer seperti ADSL, API, ABI dan singkatan lainnya. Saran untuk penelitian selanjutnya untuk mengoptimalkan pengembangan kamus digital istilah komputer dan informatika meliputi meningkatkan fitur-fitur baru agar tampilan sistem menjadi lebih menarik dan pengembangan kamus istilah informatika dapat dimanfaatkan secara luas. Saran selanjutnya akan lebih baik penelitian selanjutnya mengkombinasikan fitur *spelling corrector* yang mampu mengidentifikasi kesalahan dan memberikan sugesti yang tepat pada pencarian *typographical error* yang melebihi 1 huruf dan disarankan meningkatkan *corpus* yang lebih baik untuk menangani kasus *typographical erro*r yang lebih kompleks.

Daftar Pustaka

[1] R. Rismayani, N. Sambo Layuk, S. Wahyuni, H. Wali, and N. K. Marselina, "Pencarian Kata Pada Aplikasi Kamus Istilah Komputer dan Informatika Menggunakan Algoritma Brute Force Berbasis Android," *Komputika J. Sist. Komput.*, vol. 10, no. 1, pp. 43–52, 2021, doi: 10.34010/komputika.v10i1.3644.

[2] N. K. Tran, Q. Z. Sheng, M. Ali Babar, L. Yao, W. E. Zhang, and S. Dustdar, "Internet of things search engine," *Commun. ACM*, vol. 62, no. 7, pp. 66–73, 2019, doi: 10.1145/3284763.

[3] A. I. Fahma, I. Cholissodin, and R. S. Perdana, "Identifikasi Kesalahan Penulisan Kata (Typographical Error) Pada Dokumen Berbahasa Indonesia Menggunakan Metode N-Gram Dan Levenshtein Distance," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 53–62, 2018.

[4] D. Hládek, J. Staš, and M. Pleva, "Survey of automatic spelling correction," *Electron.*, vol. 9, no. 10, pp. 1–29, 2020, doi: 10.3390/electronics9101670.

[5] R. Martin, D. S. Naga, and V. C. Mawardi, "Penggunaan Spelling Correction Dengan Metode Peter Norvig Dan N-Gram," *J. Ilmu Komput. dan Sist. Inf.*, vol. 9, no. 1, p. 175, 2021, doi: 10.24912/jiksi.v9i1.11591.

[6] Mutammimah, H. Sujaini, and R. D. Nyoto, "Peter Norvig dan Spelling Checker BK-Trees pada Kata Berbahasa Indonesia," *J. Sist. dan Teknol. Inf.*, vol. 5, no. 1, pp. 1–5, 2016.

[7] T. M. Fahrudin, I. Sa'diyah, L. Latipah, I. Z. Atha Illah, C. C. Bey Lirna, and B. S. Acarya, "Aplikasi Pendeteksi Kesalahan Ejaan Bahasa Indonesia Pada Karya Ilmiah Bidang Ilmu Komputer Menggunakan Kebi 1.0 Checker," *Pros. Semin. Nas. Inform. Bela Negara*, vol. 2, pp. 64–72, 2021, doi: 10.33005/santika.v2i0.104.

[8] R. Alfred and R. W. Teoh, *Improving topical social media sentiment analysis by correcting unknown words automatically*, vol. 937. Springer Singapore, 2019.

[9] R. Irsyad, "Penggunaan Python Web Framework Flask Untuk Pemula," *Lab. Telemat. Sekol. Tek. Elektro Inform.*, pp. 1–4, 2018.

[10] D. Ghimire, "Comparative study on Python web frameworks: Flask and Django," no. May, pp. 1–40, 2020, [Online]. Available: http://www.theseus.fi/handle/10024/339796.

[11] M. Dr. Umar Sidiq, M.Ag Dr. Moh. Miftachul Choiri, *Metode Penelitian Kualitatif di Bidang Pendidikan*, vol. 53, no. 9. 2019.

[12] A. Rijali, "Analisis Data Kualitatif," *Alhadharah J. Ilmu Dakwah*, vol. 17, no. 33, p. 81, 2019, doi: 10.18592/alhadharah.v17i33.2374.

[13] N. M. A. J. Astari, Dewa Gede Hendra Divayana, and Gede Indrawan, "Analisis Sentimen Dokumen Twitter Mengenai Dampak Virus Corona Menggunakan Metode Naive Bayes Classifier," *J. Sist. dan Inform.*, vol. 15, no. 1, pp. 27–29, 2020, doi: 10.30864/jsi.v15i1.332.

[14] U. Chuzaimah Zulkifli, "Pengembangan Modul PreprocessingTeks untuk Kasus Formalisasi dan Pengecekan Ejaan Bahasa Indonesia pada Aplikasi Web Mining Simple Solution (WMSS)," *J. Mat. Stat. dan Komputasi*, vol. 15, no. 2, p. 95, 2018, doi:

10.20956/jmsk.v15i2.5718.
[15] T. M. Fahrudin, I. Sa'diyah, Latipah, I. Z. Atha Illah, C. C. Beylirna, and B. S. Acarya, "Analysis and Development of KEBI 1.0 Checker Framework as an Application of Indonesian Spelling Error Detection," *Int. J. Data Sci. Eng. Anaylitics*, vol. 1, no. 2, pp. 12–26, 2021, doi: 10.33005/ijdasea.v1i2.9.
[16] P. Norvig, "How to Write a Spelling Corrector," 2007. https://norvig.com/spell-correct.html.
[17] T. Shooting, D. Metode, and S. Pakar, "JURNAL CAHAYA BAGASKARA Vol. 6. No. 1 - Februari 2021," vol. 6, no. 1, 2021.
[18] M. N. Cholis, E. Yudaningtyas, and M. Aswin, "Pengaruh Penggunaan Synonym Recognition dan Spelling Correction pada Hasil Aplikasi Penilaian Esai dengan Metode Longest Common Subsequence dan Cosine Similarity," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 3, no. 2, pp. 138–142, 2019, doi: 10.30743/infotekjar.v3i2.1061.
[19] P. Norvig, "technical papers, essays, reports, software, and other materials by Peter Norvig." https://norvig.com/.