ISSN 1693-1629 vol.2 no.1/Mei, 2004

Jurnal Ilmiah

ilmu komputer

Publikasi Hasil Penelitian Departemen Ilmu Komputer
Institut Pertanian Bogor

1
Analisis Kinerja Algoritma Pelacakan String Dengan Toleransi Kesalahan
(Performance Analysis Of Text Searching Allowing Errors Algorithms)
Sukmasari, Julio Adisantoso, Meuthia Rachmaniah

10
Analisis Dan Penerapan Algoritma *RIPEMD-160* Sebagai Fungsi Penyandi Dalam Proses Autentikasi Password
Hardiyan Eka P, Sugi Guritman, Agus Buono

20
Implementasi Dan Analisis Algoritma *Linear Discriminant* Dan *Local Linear Discriminant* Dalam Mengklasifikasi Gender Dengan Praproses *Principal Component Analysis*
Meuthia Rachmaniah, Mochamad Tito Julianto, Dedi Muhammad Imron

30
Penerapan Algoritme Genetik Pada *Travelling Salesman Problem*
Naviansony Tandriarto, Agus Buono , Utari Wijayanti

42
Penyempurnaan Dan Implementasi *Software* Kompresi Multi Tahap Menggunakan *Huffman Coding*
(Improvement And Implementation Of Multi-Stage Compression Using Huffman Coding)
Marimin, Kudang Boro Seminar, Layungsari

Departemen Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Pertanian Bogor

Sekapur Sirih

Alhamdulillah, Jurnal Ilmiah **Ilmu Komputer** yang kedua akhirnya dapat juga kami terbitkan. Untuk menghasilkan Jurnal berikutnya setelah terbitan perdana ternyata tidak jauh lebih mudah. Banyak kendala yang harus dihadapi tetapi dengan semangat dan dorongan dari berbagai pihak, akhirnya segala hambatan bisa diatasi.

Terbitan edisi perdana cukup banyak mengundang kritik dan saran perbaikan dari berbagai pihak. Ini sangat bermanfaat untuk upaya perbaikan di masa datang. Selain itu, ada hal lain yang membesarkan hati, yaitu kesediaan beberapa institusi/Perguruan Tinggi di luar IPB untuk berlangganan. Untuk itu kami sangat berterima kasih. Kesemuanya ini lebih memacu kami untuk bekerja lebih baik lagi.

Lima tulisan yang kami sajikan pada edisi kali ini kesemuanya membahas tentang analisis kinerja dan penerapan algoritma, Materi yang dibahas cukup beragam sehingga kami berharap akan menambah khasanah atau referensi bagi para pembaca.

Kami berharap kritik dan saran yang membangun dilayangkan lewat e-mail kami di: jurnal@ilkom.fmipa.ipb.ac.id.

Semoga bermanfaat.

Kampus Baranangsiang , Mei 2004

Salam,

Redaksi

Jurnal Ilmiah **ilmu komputer**

Diterbitkan oleh: Departemen Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam - Institut Pertanian Bogor

Vol. 2. No. 1 . Mei 2004
ISSN : 1693-1629. Tanggal 4 April 2003

Susunan Redaksi

Sekretariat Jurnal Ilmiah Ilmu Komputer :

Departemen Ilmu Komputer FMIPA IPB
Jln. Raya Pajajaran, Kampus Baranangsiang Bogor 16144
Telp/Fax : 0251-356653. E-mail : jurnal@ilkom.fmipa.ipb.ac.id
Rekening : Tabungan Taplus BNI Pajajaran Bogor.
No: 061.000322402.911 A.n : Annisa/Jurnal Ilkom

Jurnal Ilmiah Ilmu Komputer diterbitkan dua kali setahun, memuat tulisan ilmiah yang berhubungan dengan bidang Ilmu Komputer dan merupakan Media Publikasi Ilmiah di lingkungan Departemen Ilmu Komputer FMIPA-IPB.
Tulisan Ilmiah dapat berupa hasil penelitian, bahasan tentang metodologi, tulisan populer, dan tinjauan buku.

Pihak perorangan / alumni yang telah memperoleh Jurnal Ilmu Komputer mohon mengganti biaya cetak Rp50.000,- / exemplar. ditransfer melalui Tabungan Taplus BNI Pajajaran . No.Rek : 061.000322402.911. A.n : Annisa / Jurnal Ilkom.

Daftar Isi

Sekapur Sirih i

Daftar Isi iii

Analisis Kinerja Algoritma Pelacakan String Dengan Toleransi Kesalahan
(Performance Analysis Of Text Searching Allowing Errors Algorithms)
Sukmasari, Julio Adisantoso, Meuthia Rachmaniah 1

Analisis Dan Penerapan *Algoritma RIPEMD-160* Sebagai Fungsi Penyandi Dalam Proses Autentikasi *Password*
Hardyan Eka P, Sugi Guritman, Agus Buono 10

Implementasi Dan Analisis Algoritma *Linear Discriminant* Dan *Local Linear Discriminant* Dalam Mengklasifikasi Gender Dengan Praproses *Principal Component Analysis*
Meuthia Rachmaniah, Mochamad Tito Julianto, Dedi Muhammad Imron 20

Penerapan Algoritme Genetik Pada *Travelling Salesman Problem*
Naviansony Tandriarto, Agus Buono , Utari Wijayanti 30

Penyempurnaan Dan Implementasi *Software* Kompresi Multi Tahap Menggunakan *Huffman Coding*
(Improvement And Implementation Of Multi-Stage Compression Using Huffman Coding)
Marimin, Kudang Boro Seminar, Layungsari 42

Analisis Kinerja Algoritma Pelacakan String Dengan Toleransi Kesalahan

(Performance Analysis Of Text Searching Allowing Errors Algorithms)

Julio Adisantoso[1], Meuthia Rachmaniah[1], Sukmasari[2]

[1] *Staf Departemen Ilmu Komputer FMIPA IPB*
[2] *Mahasiswi Departemen Ilmu Komputer FMIPA IPB*

Abstrak

Pelacakan string banyak diperlukan pada pemrosesan string, misalnya untuk pengeditan teks. Terkadang pelacakan string tidak berhasil dilakukan; salah satu penyebabnya adalah adanya kesalahan ejaan. Pelacakan string yang tidak eksak atau pelacakan dengan toleransi kesalahan diperlukan untuk mengatasi masalah ini. Pada penelitian ini dilakukan perbandingan dua jenis algoritma pelacakan string dengan toleransi kesalahan, yaitu algoritma Wu-Manber (WM) dan algoritma Pemrograman Dinamis (DP). Tujuannya adalah untuk melihat efisiensi kedua jenis algoritma ditinjau dari keefektifannya dalam menampilkan perkiraan kata dan waktu proses yang diperlukan dalam menampilkan perkiraan kata. Hasil dari penelitian ini diperoleh bahwa algoritma WM dan DP keduanya memberikan kinerja yang sama bagusnya, yaitu 100% terkoreksi untuk semua jenis kesalahan dan waktu proses algoritma WM kurang lebih 3 kali lebih cepat dibandingkan waktu proses algoritma DP.

Kata Kunci: *Pelacakan string, pelacakan string tidak eksak atau pelacakan string dengan toleransi kesalahan, jarak edit, algoritma Wu-Manber, algoritma pemrograman dinamis, struktur data trie.*

Abstract

String matching is needed in many string processings, for example in a text editor. Sometimes searching for the pattern in a text is not successful. One of the problems is there is a misspelling in the text. It is important to do an approximate string matching or string matching with k errors to solve this problem. This research presents the comparison of two approximate string matching algorithms, namely Wu-Manber algorithm (WM) and Dynamic Program algorithm (DP) using edit distance. The purpose are to discover the efficiency of both algorthms and the effectiveness in showing the approximate words and the processing time of the searching until finding the approximate words. The results of this research are: WM algorithm gives the effectiveness in showing the approximate word as good as DP algorithm, which is 100% correct for all kinds of mistakes and the processing time of WM algorithm is 3 times as fast as that of DP algorithm.

Key Words: *String matching, approximate string matching or string matching with k errors, edit distance, Wu-Manber algorithm, Dynamic Programming Algorithm, Trie Data Structure.*

PENDAHULUAN

Latar Belakang

Pelacakan string adalah salah satu komponen yang sangat penting dalam pemrosesan string, termasuk pengeditan teks, temu kembali bibliografi, dan manipulasi simbol. Pelacakan string dilakukan dengan cara menyesuaikan suatu pola dengan teks *(Baeza Yates & Gonnet, 1992).*

Untuk beberapa kasus tertentu, seringkali pelacakan string tidak berhasil dilakukan karena adanya kesalahan pengetikan string yang akan dicari ataupun yang terdapat dalam file teks, sebagai akibat dari pengguna tidak mengingat secara pasti ejaan string yang akan dicari, atau kesalahan pengejaan pada file teks itu sendiri. Untuk mengatasi hal ini, perlu adanya suatu cara untuk melakukan pelacakan string dengan mempertimbangkan toleransi kesalahan sehingga memungkinkan untuk melakukan pelacakan yang tidak eksak *(Wu & Manber, 1992).*

Tujuan

Mempelajari dan membandingkan dua jenis algoritma pelacakan string dengan toleransi kesalahan, yaitu algoritma Wu–Manber *(WM)* dan Pemrograman Dinamis *(DP)*, dalam hal keefektifan kedua jenis algoritma dalam menemukan perkiraan kata serta dalam hal waktu proses untuk menampilkan perkiraan kata.

Ruang Lingkup

Penelitian yang dilakukan dibatasi sampai dengan pembuatan program dan menganalisis perbandingan antara algoritma WM dan algoritma DP dalam hal keefektifan kedua algoritma untuk menemukan perkiraan kata yang cukup baik serta dalam hal waktu proses untuk menampilkan perkiraan kata dengan menggunakan jarak edit k = 1dan 2.

Manfaat Penelitian

Output dari penelitian ini adalah algoritma pelacakan string dengan toleransi kesalahan yang efektif dalam menemukan perkiraan kata dan waktu proses yang cepat.

Penelitian ini diharapkan dapat dijadikan acuan lebih lanjut untuk pengembangan sistem yang memerlukan pelacakan string seperti pengoreksian ejaan, temu kembali informasi untuk algoritma filtering, temu kembali bibliografi untuk sistem katalog buku, pencarian DNA yang mirip pada serangkaian DNA, atau pencarian suatu not yang mirip pada serangkaian not.

TINJAUAN PUSTAKA

Pelacakan String

Pelacakan string disebut juga sebagai pencarian string. Operasi yang terjadi pada pelacakan string dapat dijelaskan sebagai berikut : Misalkan suatu pola *(string yang akan dipadankan dengan teks)* dengan panjang m (P = $p_1\ p_2\ p_3 ..... p_m$) direpresentasikan sebagai suatu array P[1.....m] dan teks dengan panjang n (T = $t_1\ t_2\ t_3\ .....t_n$) sebagai suatu array T [1...n]. Masalah pada pelacakan string adalah apakah P terjadi pada T atau apakah P merupakan substring dari T.

Pelacakan string dapat dibedakan menjadi dua, yaitu pelacakan eksak dan pelacakan tidak eksak. Disebut sebagai pelacakan eksak karena string hasil pelacakan yang diperoleh dari teks sama persis dengan pola. Algoritma pelacakan eksak misalnya algoritma Brute Force *(BF)*, Knutt-Morris-Pratt *(KMP)*, Boyer Moore *(BM)*, Shift Or *(SO)* dan masih banyak lagi.

Pada penelitian ini, digunakan algoritma pelacakan string tidak eksak yang memperbolehkan adanya ketidakpadanan antara pola dengan substring dari teks dengan toleransi kesalahan tertentu. Pelacakan string tidak eksak selanjutnya juga disebut sebagai pelacakan string dengan toleransi kesalahan.

Pelacakan String Tidak Eksak

Ada beberapa kasus dimana pelacakan string tidak dapat dilakukan secara eksak. Sebagai contoh pencarian nama seorang nasabah pada suatu bank dengan memberikan nama depan dari nasabah yang dicari. Permasalahan akan muncul jika tidak diketahui dengan pasti bagaimana ejaan atau cara penulisan nama depan nasabah tersebut. Misalnya, pengguna tidak mengetahui apakah nama yang dicari adalah **Ahmad**, **Achmad** atau **Akhmad**.Untuk memecahkan masalah tersebut, maka dilakukan pencarian pada teks yang mendekati pola dengan jumlah ketidakpadanan karakter yang sudah ditentukan.

Ketidakpadanan antara dua string dapat dikelompokkan menjadi dua, yaitu *Hamming distance* dan *Levenshtein distance*. *Hamming distance* adalah ketidakpadanan antara dua string yang memiliki panjang string yang harus sama. Oleh karena itu, ketidakpadanan ini diterapkan pada kasus pelacakan string dengan k ketidakcocokan. Sedangkan *Levenshtein distance* adalah ketidakpadanan antara dua string yang boleh memiliki panjang string yang berbeda. Ketidakpadanan ini diterapkan pada kasus pelacakan string dengan k perbedaan *(Crochemore & Lecroq, 1996)*.

Jarak Edit

Jarak edit (*edit distance*) disebut juga sebagai *Levenshtein distance*. Jarak edit ini adalah jumlah minimal perbedaan antara dua string. String A dikatakan berjarak edit k dengan string B jika string A dapat ditransformasikan menjadi sama dengan string B dengan melakukan k kali operasi perubahan. Operasi perubahan yang dilakukan adalah :

1. Operasi penggantian satu karakter dengan karakter lain, misalnya kata "bantu" berjarak edit satu dengan "pantu".
2. Operasi penghapusan satu karakter, misalnya kata "pangku" berjarak edit satu dengan "panku".
3. Operasi penyisipan satu karakter, misalnya kata "ceria" berjarak edit satu dengan "cerria".

Kombinasi dari ketiga operasi perubahan ini akan menghasilkan ketidakpadanan yang lebih kompleks dengan jarak edit lebih besar dari 1. Jumlah perbedaan karakter antara dua string untuk masalah jarak edit k ditentukan oleh banyaknya operasi penggantian, penghapusan dan penyisipan yang dilakukan *(French et al, 1997)*.

Algoritma Wu-Manber (WM)

Algoritma Wu-Manber *(WM)* adalah algoritma yang dikembangkan oleh Sun Wu dan Udi Manber *(Wu & Manber, 1992)*. Algoritma ini adalah algoritma pelacakan string dengan k toleransi kesalahan. Jika k = 0 maka algoritma ini menjadi algoritma pelacakan eksak. Algoritma WM dapat dilihat pada **Gambar 1**.

```
Pemadanan_Tidak_Eksak_WuManber(P,T)
{P [1...m] ; T[1....n];
Praproses untuk semua karakter pada pola (S);
For All j (1 ≤ j ≤ n)
  R_{j+1} = Right Shift R_j AND S
  For All d (1 ≤ d ≤ k)
    R^d_0 = 11..100...000 ( d buah 1)
    R^d_{j+1} = RightShift [ R^d_j ] AND S OR
                RightShift [ R^{d-1}_j ] OR
                RightShift [ R^{d-1}_{j+1} ] OR
                R^{d-1}_j

If R^k[m] = 1 "Ditemukan kesepadanan "}
```

Gambar 1. Algoritma Tidak Eksak Wu-Manber dengan Manipulasi Bit.

Algoritma Pemrograman Dinamis (DP)

Algoritma Pemrograman Dinamis/ *Dynamic Programming (DP)* dapat menyelesaikan masalah pelacakan string dengan menggunakan prinsip perhitungan jarak edit dalam menentukan kesamaan antara dua buah string. Algoritma DP dapat dirangkum seperti pada **Gambar 2**.

```
Pemadanan_Tidak_Eksak_DP(P,m,T,n)
{P[1...m]; T[1...n] ;
For All i (0<=i<=m) D(i,0) = i;
For All j (0<=j<=n) D(0,j) = j;
For All j ( 1<=j<=n)
    For All i (1<=i<=m)
      If (P[i] = T[j] )
        D(i,j) = Min ( D(i-1,j) + 1,
                       D(i-1,j-1),
                       D(i,j-1) + 1)
      Else
        D(i,j) = Min ( D(i-1,j) + 1,
                       D(i-1,j-1) + 1,
                       D(i,j-1) + 1)
    If D(m,n)<=k  "Ditemukan kesepadanan"}
```

Gambar 2. Algoritma Tidak Eksak DP.

Struktur Data Trie

Pada kasus pelacakan string diperlukan suatu kamus referensi yang dijadikan sebagai acuan dalam memeriksa setiap kata sehingga jika terjadi kesalahan ejaan, maka dapat diberikan perkiraan kata-katanya dari kata yang terdapat pada kamus referensi tersebut. Struktur data *trie* memberikan kinerja yang cukup baik untuk merepresentasikan kamus referensi. Hal ini dikarenakan struktur data *trie* memiliki kompleksitas yang rendah yaitu *O(1)* pada kondisi terbaik dan *O(m)* pada kondisi terburuk dengan m adalah panjang kata yang akan dicari.

Untuk lebih mempercepat pencarian, maka digunakan prefix *trie* dimana setiap simpul dalam *trie* boleh berisikan ≥ 1 karakter dari sebuah kata. Jumlah karakter dari suatu kata pada sebuah simpul tergantung pada keberadaan kata-kata lain di dalam *trie*. Namun pada kasus terburuk suatu simpul dapat hanya berisikan satu karakter dari sebuah kata *(Dundas, 1991)*.

Tipe data *trie* yang digunakan dalam penelitian ini yaitu :

1. pow (*part of word*) adalah suatu peubah bertipe string yang digunakan untuk menampung satu kata atau sebagian kata dalam struktur data *trie*.
2. *is_word* adalah nilai dari setiap simpul. Jika nilai simpul 1 maka mulai dari simpul tersebut sampai rootnya adalah sebuah kata, jika nilai simpul adalah 0 maka simpul tersebut adalah bagian dari sebuah kata.
3. *parent* adalah penunjuk simpul sebelumnya dalam satu kata.
4. *next* adalah penunjuk kata berikutnya.
5. *child* adalah penunjuk simpul berikutnya dalam satu kata.

METODE PENELITIAN

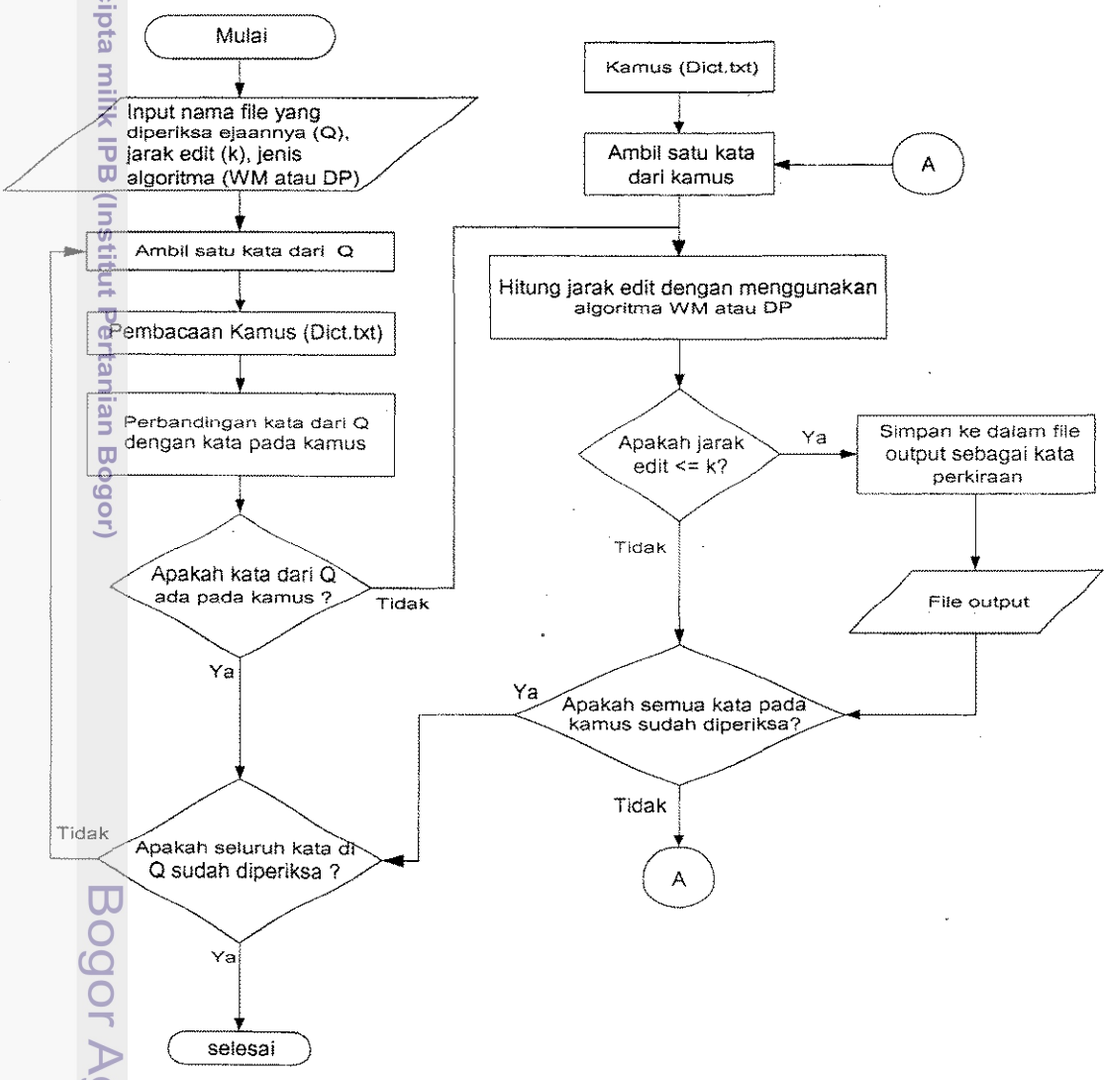
Pengumpulan Data

Data penelitian yang digunakan adalah kata yang diambil dari Kamus Besar Bahasa Indonesia *(KBBI)* Edisi ke tiga tahun 2001. Jumlah kata yang dimasukkan sebagai kamus referensi sekitar 3720 kata.

Perancangan Program

Program yang dibuat adalah program yang mengimplementasikan algoritma WM dan algoritma DP. Input program adalah nama dokumen file teks yang akan diperiksa ejaannya, jarak edit dan jenis algoritma *(WM atau DP)* yang digunakan. Pertama kali program dijalankan, sistem akan membentuk struktur data *trie*

berdasarkan kamus referensi yang disimpan dalam bentuk file Dict.txt. Pemeriksaan dilakukan satu persatu pada setiap kata yang terdapat pada file teks dengan membandingkan terhadap kata pada kamus referensi struktur data *trie*. Dengan menggunakan struktur data *trie* ini maka proses perbandingan dilakukan dengan cepat sebab sistem tidak perlu membandingkan setiap kata yang terdapat pada kamus referensi. Jika kata terdapat pada kamus referensi, maka kata tersebut mempunyai ejaan yang benar. Jika tidak, dilakukan pencarian kata perkiraan pada kamus referensi dengan melakukan perhitungan jarak edit antara kata yang diperiksa terhadap kata yang terdapat pada kamus referensi menggunakan algoritma WM atau algoritma DP. Kata pada kamus referensi yang memenuhi jarak edit yang telah ditentukan akan disimpan ke dalam file kata perkiraan. Output dari sistem adalah file txt yang berisi kata dan panjang kata yang salah ejaannya, kata dan panjang kata perkiraan, jarak edit dan waktu pencarian kata. Bagan alir program yang dibuat dapat dilihat pada **Gambar 3**.

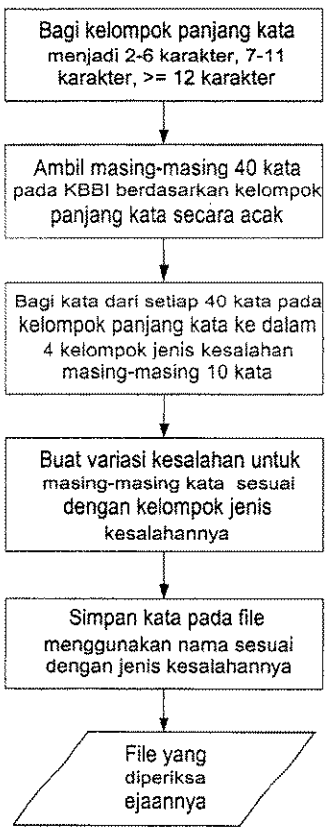


Gambar 3. *Bagan Alir Program Pelacakan String dengan Toleransi Kesalahan.*

Percobaan

Pada tahap percobaan, kata-kata pada Kamus Besar Bahasa Indonesia *(KBBI)* dibagi ke dalam 3 kelompok berdasarkan panjang kata, yaitu kelompok panjang kata 2-6 karakter, kelompok panjang kata 7-11 karakter dan kelompok panjang kata ≥12 karakter. Selanjutnya didefinisikan kelompok jenis kesalahan, yaitu : penggantian satu karakter, penghapusan satu karakter, penyisipan satu karakter, dan kombinasi dari ketiganya.

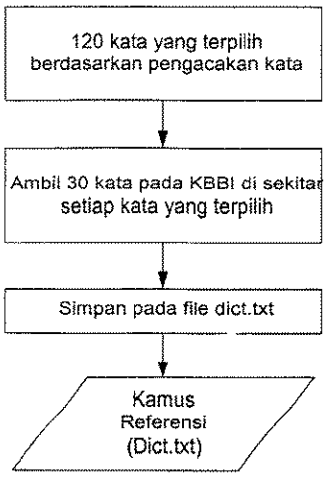
Pada tahap percobaan, hal yang dilakukan adalah menyiapkan file yang akan diperiksa ejaannya yang merupakan input sistem dan pembentukan kamus referensi. Langkah-langkah dalam pembentukan file yang akan diperiksa ejaannya dapat dilihat pada **Gambar 4**.



Gambar 4. Pembentukan File yang Diperiksa Ejaannya.

Kamus referensi diperlukan sebagai acuan dalam memeriksa setiap kata. Jika ada kata salah, maka dapat diberikan kata perkiraan berdasarkan kata pada kamus referensi yang memenuhi jarak edit yang telah ditentukan.

Langkah-langkah pembentukan kamus referensi pada percobaan ini dapat dilihat pada **Gambar 5**.



Gambar 5. Pembentukan Kamus Referensi.

Analisis Kinerja Algoritma Wu-Manber dan Algoritma Pemrograman Dinamis (DP)

Kinerja algoritma WM dan algoritma DP dinilai berdasarkan keefektifannya dalam menemukan kata-kata perkiraan yang tepat dan waktu proses yang cepat.

HASIL DAN PEMBAHASAN

Input dan Output Sistem

Input dari sistem ini adalah nama file yang diperiksa ejaannya, jarak edit yang digunakan dan jenis algoritma WM atau DP. Pada saat percobaan penelitian ini banyaknya file yang diperiksa ejaannya yang digunakan adalah 4 file yaitu :

1. Penyisipan_1_huruf.txt. Berisi 30 kata yang telah dibuatkan variasi kesalahan penyisipan 1 huruf dengan posisi yang dibuat secara acak.
2. Penghapusan_1_huruf.txt. Berisi 30 kata yang telah dibuatkan variasi kesalahan penghapusan 1 huruf dengan posisi yang dibuat secara acak.

3. Penggantian_1_huruf.txt. Berisi 30 kata yang telah dibuatkan variasi kesalahan penggantian 1 huruf dengan posisi yang dibuat secara acak.
4. Gabungan.txt. Berisi 30 kata yang telah dibuatkan variasi kesalahan penyisipan, penghapusan atau penggantian 1 huruf dengan posisi yang dibuat secara acak.

Setiap file yang diperiksa ejaannya terdiri dari 10 kata dari kelompok panjang kata 2-6 karakter, 10 kata dari kelompok panjang kata 7-11 karakter dan 10 kata dari kelompok panjang kata ≥12 karakter.

Jarak edit yang diinputkan pada sistem adalah 1 dan 2. Jarak edit ini dibatasi dengan tujuan membatasi kata perkiraan yang dapat terambil. Untuk setiap jenis kesalahan penggantian 1 huruf, penghapusan 1 huruf, penyisipan 1 huruf diujicobakan dengan menggunakan jarak edit k=1 dan 2. Khusus untuk jenis kesalahan gabungan hanya diujicobakan dengan menggunakan jarak edit k=2, sebab pada file yang diperiksa ejaannya untuk file Gabungan.txt berisi variasi kata yang semuanya mempunyai jarak edit k=2 yang merupakan gabungan variasi kesalahan dari penyisipan, penggantian atau penghapusan. Dengan menggunakan 2 jenis algoritma, yaitu algoritma WM dan algoritma DP, didapatkan variasi 14 unit percobaan, yaitu untuk masing-masing algoritma file penyisipan 1 huruf dipasangkan dengan jarak edit k=1 dan 2, penggantian 1 huruf dipasangkan dengan jarak edit k=1 dan 2, penghapusan 1 huruf dipasangkan dengan jarak edit k=1 dan 2, sedangkan Gabungan.txt dipasangkan hanya dengan jarak edit k= 2. Hal ini dilakukan dengan tujuan untuk mengamati keefektifannya dan waktu yang diperlukan dalam hal menampilkan kata perkiraan dilihat dari jarak editnya, kelompok panjang katanya, jenis kesalahannya dan perbandingan antara hasil yang ditampilkan untuk algoritma WM dan DP.

Proses yang dilakukan oleh sistem selanjutnya membangun struktur data *trie* berdasarkan kamus referensi yang telah disimpan pada file Dict.txt. Sistem akan mengambil kata satu per satu pada file yang diperiksa ejaannya dan mencocokkannya pada kamus referensi struktur data *trie* dengan tujuan untuk menentukan apakah kata yang diperiksa sudah mempunyai ejaan yang benar atau tidak. Dengan menggunakan struktur data *trie*, sistem tidak perlu mencocokkan semua kata yang terdapat pada kamus referensi sebab struktur data *trie* menggunakan setiap karakter yang terdapat di dalam kata untuk membedakan kata yang satu dengan kata lainnya. Jika kata ditemukan pada struktur data *trie*, maka kata yang diperiksa telah mempunyai ejaan yang benar. Jika tidak, maka dilakukan pencarian dengan menggunakan jarak edit dari algoritma WM atau DP yang telah dibahas. Proses selanjutnya adalah membandingkan setiap kata pada kamus referensi struktur data *trie*. Kata yang memenuhi jarak edit yang telah ditentukan disimpan pada file output.

File output dari sistem berisi informasi kata yang diperiksa ejaannya dan panjang katanya *(m)*, kata perkiraan dan panjang katanya *(n)*, jarak edit *(k)* dan waktu yang diperlukan untuk menampilkan kata perkiraan tersebut. Setiap unit percobaan diulang sebanyak 5 kali untuk mendapatkan waktu rataan dari waktu pencarian. Hal ini dilakukan sebab terkadang sistem tidak memberikan waktu pencarian yang sama setiap kali sistem dijalankan.

Analisis Keefektifan Algoritma Wu Manber (WM) dan Algoritma Pemrograman Dinamis (DP)

Output dari sistem ini berupa kata perkiraan yang bukan didasarkan pada kedekatan makna dari kata yang salah, melainkan kata perkiraan yang susunan hurufnya mendekati kata yang salah berdasarkan jarak edit yang telah ditentukan. Jumlah kata perkiraan yang diperoleh tergantung pada jumlah kata yang terdapat pada kamus referensi yang mempunyai nilai jarak edit sesuai dengan jarak edit yang telah diberikan *(Dalam percobaan ini menggunakan jarak edit k = 1 dan 2).*

Analisis Berdasarkan Jarak Edit

Untuk jarak edit k=1, kata perkiraan yang diberikan lebih sedikit dan mendekati kata yang salah sehingga hasil yang diberikan lebih akurat, tetapi sangat tergantung dari ketersediaan kata pada kamus referensi yang memiliki jarak edit k=1 dari kata yang salah. Ada kemungkinan suatu kata yang salah tidak diberikan kata perkiraan atau kata perkiraan yang diberikan bukan kata pengoreksian yang benar dari kata yang salah.

Jika menggunakan jarak edit k=2, lebih banyak alternatif kata dalam menentukan pengoreksian kata yang salah tetapi dengan semakin banyaknya kata perkiraan yang diberikan akan membuat pengguna bingung menentukan pengoreksian kata.

Analisis Berdasarkan Kelompok Panjang Kata

Kelompok panjang kata 2-6 karakter memberikan lebih banyak kata perkiraan dibandingkan kelompok panjang kata 7-11 dan ≥ 12, terutama pada kata perkiraan yang menggunakan jarak edit k = 2. Dapat dikatakan bahwa semakin panjang kata yang salah akan memberikan kata perkiraan yang semakin sedikit dan semakin mendekati pada kata perkiraan yang diharapkan. Sebaliknya semakin pendek kata yang salah maka semakin banyak jumlah kata perkiraan yang diberikan.

Pada kelompok panjang kata 2-6 karakter, terlebih pada kata yang memiliki panjang kata=2, kata perkiraan yang ditampilkan sangat berbeda dengan usulan kata yang sedang diperiksa. Dapat disimpulkan untuk kata salah yang memiliki panjang kata yang lebih kecil atau sama dengan jarak editnya maka kata perkiraan yang ditampilkan akan sangat berbeda dari kata salahnya.

Analisis Berdasarkan Jenis Kesalahan

Terdapat pola yang sama antara jenis kesalahan penggantian 1 karakter, penyisipan 1 karakter dan kesalahan gabungan. Kata perkiraan yang dihasilkan dari ketiga jenis kesalahan ini diamati pada kelompok panjang kata 2-6 memberikan jumlah kata perkiraan yang banyak, sedangkan kelompok panjang kata 7-11 dan ≥ 12 cenderung memberikan kata perkiraan yang jumlahnya sama dan jauh lebih sedikit dari jumlah kelompok panjang kata 2-6 karakter.

Pada jenis kesalahan penghapusan 1 karakter, selain memberikan lebih banyak kata perkiraan pada kelompok panjang kata 2-6 karakter, juga memberikan lebih banyak kata perkiraan pada kelompok panjang kata 7-11 karakter meskipun jumlah kata perkiraannya lebih sedikit dari jumlah kata perkiraan pada kelompok panjang kata 2-6 karakter dan lebih banyak dari kelompok panjang kata ≥ 12. Dapat disimpulkan bahwa kata yang mengalami kesalahan penghapusan akan memberikan kata perkiraan yang jumlahnya lebih banyak daripada jenis kesalahan penggantian 1 karakter, penyisipan 1 karakter atau kesalahan gabungan.

Analisis Perbandingan Hasil Output Kata Perkiraan Algoritma WM dan DP

Algoritma WM dan DP keduanya dapat menampilkan semua kata pengoreksian yang benar untuk kata yang sengaja dibuatkan kata salahnya sehingga keduanya memberikan kinerja yang sama bagusnya, yaitu 100% terkoreksi untuk semua jenis kesalahan.

Pada beberapa kata yang salah yang diujicobakan, jumlah kata perkiraan yang ditampilkan oleh algoritma WM ≥ jumlah kata perkiraan yang ditampilkan oleh algoritma DP.

Analisis Waktu Proses Algoritma Wu-Manber (WM) dan Algoritma Pemrograman Dinamis (DP)

Analisis Waktu Pencarian Berdasarkan Panjang Kata

Pada **Tabel 1** dapat diamati untuk kedua algoritma WM dan DP waktu pencarian yang

Tabel 1. Waktu Proses Algoritma WM dan DP pada setiap kelompok panjang kata, jenis kesalahan dengan jarak edit k = 1 dan 2

NO	Jenis Kesalahan	Algoritma DP				Algoritma WM			
		2-6	7-11	≥12	Rataan	2-6	7-11	≥12	Rataan
1	Penggantian k=1	1,9	5,82	13,06	6,93	1,93	2,24	2,8	2,32
2	Penghapusan k=1	1,92	4,37	10,66	5,65	1,95	2,13	2,6	2,23
3	Penyisipan k=1	2,68	6,2	13,22	7,37	2,04	2,42	2,7	2,39
4	Penggantian k=2	1,86	5,47	12,85	6,73	2,06	2,57	3,2	2,61
5	Penghapusan k=2	1,74	4,2	10,5	5,48	2,06	2,37	3	2,48
6	Penyisipan k=2	2,71	6,45	13,14	7,43	2,24	2,61	3,14	2,66
7	Gabungan	2,18	5,39	12,7	6,76	2,14	2,52	3,06	2,57
	Rataan	2,14	5,41	12,30	6,62	2,06	2,41	2,93	2,47

diberikan oleh kelompok panjang kata 2-6 karakter < waktu pencarian kelompok panjang kata 7-11 karakter < waktu pencarian kelompok panjang kata ≥12 karakter. Dapat disimpulkan bahwa semakin panjang kata salah, maka semakin lama waktu yang diperlukan untuk menampilkan kata perkiraan dan sebaliknya, semakin pendek kata salahnya, maka waktu yang diperlukan untuk menampilkan kata perkiraan akan semakin cepat. Hal ini disebabkan karena pada kedua algoritma untuk menampilkan kata perkiraan diperlukan penelusuran perbandingan terhadap panjang kata salah (m).

Analisis Waktu Pencarian Berdasarkan Jenis Kesalahan

Berdasarkan **Tabel 1**, pada algoritma WM waktu proses yang paling cepat dilihat dari jenis kesalahan yaitu Penghapusan 1 huruf pada jarak edit k=1 dengan waktu 2.23 mikro detik. sedangkan waktu proses terlama adalah Penyisipan 1 huruf pada jarak edit k=1 dengan waktu 2.66 mikro detik. Pada algoritma DP waktu proses yang paling cepat berdasarkan jenis kesalahan adalah jenis kesalahan penghapusan 1 huruf dengan jarak edit k= 2 dengan waktu 5.48 mikro detik sedangkan waktu proses terlama adalah Penyisipan 1 karakter pada jarak edit k=1 dengan waktu 7.43 mikro detik. Dapat disimpulkan bahwa berdasarkan jenis kesalahannya, waktu pencarian tercepat diberikan oleh jenis kesalahan penghapusan dan waktu pencarian terlama diberikan oleh jenis kesalahan penyisipan. Hal ini disebabkan secara rata-rata jenis kesalahan penghapusan memiliki panjang karakter kata salah yang lebih kecil daripada jenis kesalahan lainnya sebab pada saat dibuatkan variasi kesalahannya setiap kata pada kelompok kata penghapusan sengaja dihilangkan satu karakter. Sebaliknya pada jenis kesalahan penyisipan 1 karakter memiliki panjang karakter kata salah yang lebih panjang daripada jenis kesalahan lainnya sebab pada saat dibuatkan variasi kesalahannya setiap kata pada kelompok kata penyisipan sengaja disisipkan satu karakter.

Analisis Perbandingan Waktu Pencarian Algoritma WM dan DP

Pada **Tabel 1** juga dapat dibandingkan waktu yang diperoleh masing-masing algoritma untuk jenis kesalahan, kelompok panjang kata dan jarak edit yang sama. Secara umum algoritma WM rata-rata memiliki waktu proses kurang lebih 3 kali lebih cepat dibandingkan algoritma DP. Hal ini disebabkan waktu pada saat pencarian untuk algoritma WM dipengaruhi oleh jarak edit k dan panjang kata pada kamus referensi n. Sedangkan pada algoritma DP waktu pada saat pencariannya dipengaruhi oleh panjang kata salah m dan panjang kata pada kamus referensi n. Hal yang mempengaruhi kedua algoritma ini berbeda pada k dan m. Nilai k hanya berkisar antara 1-2 *(karena jarak edit k yang digunakan hanya dibatasi pada k=1 dan 2)* sedangkan pada algoritma DP yang dipengaruhi oleh m dimana m mempunyai kisaran nilai dari 2-16 karakter.

Analisis Regresi Algoritma DP

Dari hasil analisis regresi untuk algoritma DP dapat disimpulkan bahwa panjang kata salah dan panjang kata pada kamus referensi berpengaruh nyata terhadap waktu pencarian sedangkan jarak edit tidak berpengaruh nyata terhadap waktu pencarian. Secara simultan, variabel-variabel bebas berpengaruh nyata terhadap variabel terikat.

Analisis Regresi Algoritma WM

Dari hasil analisis regresi untuk algoritma WM dapat disimpulkan bahwa panjang kata salah dan panjang kata pada kamus referensi berpengaruh nyata terhadap waktu pencarian sedangkan jarak edit seharusnya berpengaruh nyata terhadap waktu pencarian tetapi menurut pemodelan jarak edit tidak berpengaruh nyata terhadap waktu pencarian. Untuk menjelaskan keadaan ini perlu ditinjau kembali nilai k yang digunakan pada percobaan dimana jarak edit k yang diinputkan sangat kecil *(k=1,2)* sehingga tidak bisa digunakan untuk melihat pengaruhnya terhadap waktu percobaan. Secara simultan, variabel-variabel bebas berpengaruh nyata terhadap variabel terikat.

KESIMPULAN DAN SARAN

Kesimpulan

Penentuan jarak edit sangat mempengaruhi banyaknya kata perkiraan. Semakin besar jarak editnya, maka semakin banyak kata perkiraan yang diberikan.

Panjang kata yang salah ejaannya mempengaruhi keakuratan kata perkiraan yang diberikan dan waktu pencarian. Semakin panjang kata yang salah akan memberikan kata perkiraan yang semakin akurat tetapi waktu pencariannya

akan semakin lama; sebaliknya, semakin pendek kata yang salah ejaannya, maka kata perkiraan yang diberikan semakin tidak akurat tetapi memiliki waktu pencarian yang semakin cepat.

Jenis kesalahan penghapusan memberikan kata perkiraan yang jumlahnya lebih banyak daripada jenis kesalahan penggantian 1 karakter, penyisipan 1 karakter atau kesalahan gabungan. Jenis kesalahan penghapusan juga memberikan waktu pencarian tercepat sedangkan jenis kesalahan penyisipan memberikan waktu pencarian terlama untuk kedua algoritma.

Algoritma WM dan DP memberikan kinerja yang sama bagusnya, yaitu 100% terkoreksi untuk semua jenis kesalahan.

Algoritma WM memiliki waktu proses 3 kali lebih cepat dibandingkan dengan algoritma DP. Berdasarkan analisis regresi, panjang kata yang salah ejaannya dan panjang kata pada kamus referensi mempengaruhi waktu proses pencarian pada algoritma WM dan DP.

Saran

Faktor yang sangat penting dalam pelacakan string adalah keefektifan menampilkan kata perkiraan dan waktu proses. Algoritma WM memiliki kinerja yang baik dalam hal keefektifan dan waktu proses, sedangkan algoritma DP memiliki kinerja yang baik untuk keefektifan dalam memberikan kata perkiraan tetapi waktu prosesnya 3 kali lebih lama dibandingkan algoritma WM. Untuk penelitian selanjutnya, disarankan untuk mengembangkan algoritma WM agar diaplikasikan pada pengembangan sistem yang memerlukan pelacakan string seperti pengoreksian ejaan pada teks Bahasa Indonesia, temu kembali informasi untuk algoritma filtering, temu kembali bibliografi untuk sistem katalog buku, pencarian DNA yang mirip pada serangkaian DNA, atau pencarian suatu not yang mirip pada serangkaian not.

DAFTAR PUSTAKA

Bacza Yates, R. & G. H. Gonnet. 1992. *A New Approach to Text Searching*. Communication of the ACM. 35*(10)*: 74-82.

Crochemore, M & T. Lecroq. 1996. *Pattern Matching and Text Compression Algorithms*. ACM Comput. Surveys.

Dundas III, J.A. 1991. *Implementing DynamicMinimal-Prefix Tries*. Software-Practice And Experience. 22(10) : 1027-1040.

French, J. C., A. L. Powell & E. Schulman. 1997. *Application of Approximate Word Matching in Information Retrieval*. Department of Computer Science University of Virginia, Charlottesville, Virginia.

Wu, S. & U. Manber. 1992. *Fast Text Searching Allowing Error*. Communication of the. ACM. 35 *(10): 83-91.*