

PENGGUNAAN METODE *EXACT MATCH* UNTUK MENENTUKAN KEMIRIPAN NASKAH DOKUMEN TEKS

Heriyanto

Jurusan Teknik Informatika UPN "Veteran" Yogyakarta
Jl. Babarsari 2 Tambakbayan 55281 Telp(0274) 485323
email : mr_heriyanto_skom@yahoo.com

Abstract

Comparison of resemblance document text to know how precisely from its source resemblance or do not with efficient and effective algorithm. Made system and designed to do examination both of source of text 1 and source of text 2 (goals) with prediksi if there is resemblance from text 1 and text 2 more near or not.

In generally system consist of three process that is process early (praprocessing with index), seeking process (process and searching with index) calculation of presisi with sign and match exact. Analysis in the form of checking : the first equality of word with word location differ, second equality of word with is same word location, and third equality of word equivalent. Equality of word equivalent combine also with equality of word with word location differ, and combine equality of word with is same word location. Resemblance used analys with Exact Match method and similarity function. Function Similarity represent process with Exact Match checking of word if precisely is equal to value 1 (true) and not equal to value 0 (false). Resemblance of source and goals base analysis match exact without used steam although there is suffix and prefix that is difference word, so added with suffix and prefix will mean word which different. Because writer assume needn't be steam because by the prefix dand suffix most word always have meaning differ.

The Finally, Analysis come near number 1 is similarity and near number 0 didn't similary. If the number 0.5 or a half that mean half similarity.

Keywords : indexing, text, similarity, exact, match, searching, source, goal, document

Perbandingan kemiripan untuk menguji beberapa dokumen yang diambil untuk mengetahui seberapa persis dari naskah tersebut mirip ataukah tidak dengan algoritma yang efektif dan efisiensi. Sistem yang dibuat dan dirancang untuk melakukan pengujian kedua naskah baik naskah sumber teks 1 dan naskah sumber teks 2 (target) dalam bentuk penyimpanan *databases* yang telah diindeks, dengan prediksi apabila ada kemiripan dari teks 1 dan teks 2 maka mendekati sama atau tidak sama. Secara garis besar sistem terdiri dari tiga proses yaitu proses awal (*praprocessing with index*), proses pencarian (*searching*) dan proses perhitungan presisi dengan penandaan dan *exact match*. Analisa yang dilakukan berupa pengecekan : pertama kesamaan kata dan letak berbeda, kedua kesamaan kata dan letak sama, dan ketiga kesamaan kata berdasarkan padanan kata. Kesamaan berdasarkan padanan kata akan di padukan pula dengan kesamaan kata dan letak berbeda, dipadukan juga dengan kesamaan kata dan letak sama. Kemiripan naskah dilakukan dengan metode *Exact Match* dan fungsi *similarity*. Fungsi *Similarity* merupakan proses penanda yang dilakukan dengan *Exact Match* pengecekan kata apabila ketemu sama persis maka dikatakan tepat sama dengan nilai 1 (*true*) dan tidak sama maka 0 (*false*). Kemiripan naskah sumber dan target berdasarkan analisa *exact match* tidak memperdulikan *stemming* sehingga apabila ada awalan dan akhiran tidak termasuk sama, sehingga kata dasar ditambah dengan awalan dan akhiran akan berarti kata yang berbeda-beda sehingga penulis menganggap tidak perlu dilakukan *steam* karena pada akhir dan awalan memang kebanyakan kata selalu mempunyai arti berbeda. Analisa yang dilakukan semakin mendekati angka 1 maka naskah target dinyatakan semakin mirip tetap sebaliknya jika mendekati angka 0 maka semakin tidak mirip. Jika terletak ditengah-tengah 0.5 setengah mirip setengah tidak.

Kata kunci : indek, teks, similarity, exact, match, searching, sumber, target, dokumen

1. PENDAHULUAN

Dalam kehidupan sehari-hari data teks atau naskah banyak dijumpai di perkantoran maupun non perkantoran dan tidak asing lagi dalam kehidupan kita sehari-hari. Dokumen teks tidak terlepas juga dengan multimedia yang terdiri dari kumpulan data teks, suara maupun gambar atau citra yang sekarang ini banyak digunakan baik melalui komputer, internet, bahkan media *handphone*.

Pada prinsipnya selama pengutipan naskah dicantumkan sumbernya menjadi tidak masalah, namun menjadi masalah manakala pencuplikan naskah atau dokumen dari kutipan tidak menyertakan nama sumbernya. Bertolak dari pengutipan yang tidak menyebutkan nama sumber tersebut maka perbandingan kemiripan naskah teks asli dengan yang akan diuji dapat dibandingkan sejauh mana suatu naskah itu mirip atautkah tidak sebagai langkah awal penelitian tentang plagiat.

Melalui pengolah data naskah atau teks maka bagaimana suatu naskah yang ada di dalam dokumen atau majalah atau naskah teks di dalam dokumen dapat diolah dengan *software* komputer dapat mengetahui perbandingan kemiripan naskah sumber satu dengan naskah sumber lain/target. Pemecahan masalah dengan membandingkan :

- a. Bagaimana mengidentifikasi bahwa suatu dokumen merupakan mirip antara dokumen satu dengan yang lain.
- b. Bagaimana cara/ metode membuat perbandingan kemiripan dengan kemiripan dokumen teks sumber dengan dokumen teks sumber yang lain.

Pada pembahasan saat ini penulis hanya membatasi :

1. *Input* sistem berupa
Dokumen teks berbahasa Indonesia berupa ASCII (*American Standard Code for Information Interchange*), tanpa simbol-simbol dan tanpa rumus-rumus.
2. Proses sistem berupa
melalui proses indek dan pencarian data suatu kata terdapat pada indeks dan ditemukan identifikasi kata tersebut ada di berbagai kata dan letak kata kemunculannya serta kesamaan kata-kata atau didalam naskah sumber dan tujuan dengan metode *Exact Match*. Pencarian kata tidak menggunakan **stemming**.
3. *Output* sistem berupa :
Hasil indeks diolah lagi menjadi *flag* akan dibandingkan secara langsung seberapa besar kemiripan dengan naskah satu dengan naskah lainnya.

Adapun beberapa manfaat yang didapat yaitu :

- a. Mendapatkan hasil perbandingan kemiripan naskah tersebut dengan mengetahui seberapa kemiripan naskah dengan sumbernya.
- b. Memberikan kontribusi perkembangan teknologi *search engine*
- c. Dengan hasil tersebut dapat mempunyai naskah berbagai sumber sebagai referensi yang dapat digunakan sebagai acuan terhadap naskah yang akan diuji sebagai naskah target yang dicurigai ada kemiripan.

2. TINJAUAN PUSTAKA

Adapun beberapa penelitian yang sudah pernah dilakukan diantaranya :

A. Penerapan Algoritma Boyer-Moore untuk Pengecekan Plagiatisme Source Code (Redya Febriyanto) 2008

Penelitian pengecekan plagiatisme *source code* (Redya Febriyanto) Metode pencocokan *string* dapat diterapkan dalam berbagai macam hal. Salah satu contoh penerapannya adalah untuk membandingkan dua *source code* untuk mengetahui apakah terdapat kesamaan diantara kedua *source code* tersebut. Namun berbeda dari teks biasa, *source code* memiliki sifat-sifat khusus. Tingkat kesamaan antar *source code* tidak dilihat dari karakter per karakter pada teks tersebut, namun lebih ke kesamaan alur program. Oleh karena itu strategi untuk menentukan tingkat kecocokannya agak berbeda dibandingkan pencocokan *string* biasa. Penerapan metode pencocokan *string* (Boyer-Moore) untuk mencari kata kunci dalam sebuah *source* dan kemudian melakukan penghitungan sederhana untuk menentukan tingkat kesamaan dua buah *source*. Satu hal yang menarik pada algoritma Boyer-Moore ini adalah perbandingan karakter dalam sebuah *string* yang dilakukan dari belakang ke depan.

Proses pengisian tabel tersebut dapat dijelaskan sebagai berikut. Karakter pertama dalam string "MAKALAH" adalah 'M' maka dalam tabel pada karakter M diisi dengan nilai 1 (indeks nya dari awal *string*) demikian seterusnya hingga karakter terakhir. Proses

pengisian tabel yang terjadi adalah proses penyimpanan sehingga hanya nilai terbesar yang pada akhirnya tersimpan dalam tabel. Berikut ini algoritmanya dalam *pseudo-code*.

```
{init: last table}
{last adalah tabel, pat=string}
i <- 1
while i <= m do
last[pat[i]] = i
i := i + 1
end
```

Setelah tabel selesai diisi maka langkah selanjutnya adalah membandingkan karakter demi karakter dalam teks yang ada. Perbandingan dimulai pada karakter ke-N dimana N adalah panjang string yang dicari. Ketika perbandingan menemukan kecocokan maka ia akan melanjutkan ke karakter sebelumnya. Demikian seterusnya hingga ditemukan bahwa karakter pertama dalam string cocok (yang berarti string ditemukan). Ketika perbandingan menemukan ketidakcocokan, maka proses akan dilanjutkan dengan meloncat ke beberapa karakter sesudahnya berdasarkan tabel yang ada.

B. TESSY (*Test of Texts Similarity*) 2008

Berdasarkan referensi dari aplikasi anticontek (Didi Achyari) 2008 mengembangkan TESSY. Namanya *Test of Texts Similarity*, atau disingkat TESSY. *Test of Texts Similarity* dibuat untuk mendeteksi kemiripan karya tulis mahasiswa. Fitur *Test of Texts Similarity* dikembangkan sebagai program desktop, bukan aplikasi berbasis web.

Karena setiap lembaga pendidikan umumnya punya standar kemiripan yang berbeda untuk dimasukkan dalam kategori penjiplakan, tingkat kemiripan yang ingin dideteksi lewat *Test of Texts Similarity* sengaja dibuat agar dapat diatur secara bebas. Aplikasi ini hanya mengecek kata-kata. Jadi, bisa membandingkan hasil karya tulis dengan dokumen yang berasal dari mana dan tahun kapanpun. Selain itu, *Test of Texts Similarity* juga dilengkapi dengan fitur untuk mencetak laporan formal.

Dua metoda pengujian yang diterapkan dalam aplikasi ini, yaitu uji kemiripan teks dan uji kemiripan frase. Untuk uji kemiripan teks, *Test of Texts Similarity* menggunakan algoritma perbandingan huruf yang sudah baku, yaitu *difference algorithm*. Algoritma ini antara lain digunakan oleh Adobe dan Microsoft dan yang dikembangkan *Test of Texts Similarity* adalah untuk membandingkan kata. Uji kemiripan teks ini dipakai untuk menghitung prosentase kemiripan dokumen yang diuji dengan dokumen yang sudah ada. Nilai prosentase tinggi menunjukkan tingkat kemiripan yang sangat tinggi. "Hasilnya cukup valid, namun *Test of Texts Similarity* belum mengecek sampai level kutipan, dasar teori, dan sejenisnya. *Test of Texts Similarity* murni hanya membandingkan kata. Dengan demikian, hasil kemiripan belum bisa dijadikan keputusan plagiarisme. Perlu ahli untuk mengecek dokumen yang dinyatakan suspect plagiat." Sementara, untuk metode kesamaan frase, *Test of Texts Similarity* akan mencari dan menghitung kemunculan frase dan kombinasinya dalam dokumen.

3. METODE PENELITIAN

1. Studi Literatur

Penulis mencari sumber pustaka atau dokumen untuk mempelajari permasalahan seperti teori *Information Retrieval*, sehingga penulis memahami konsep IR, indeks dan ilmu pendukung lainnya.

2. Analisa dan perancangan

Melakukan pendekatan *exact match* (and) dan melakukan penghitungan *similarity*. Hasil dari kesamaan indeks dilakukan pencocokan apabila tepat sama maka *exatch match 1 apabila tidak 0* memberi tanda/ *flag*, dikarenakan banyaknya pengecekan maka dilakukan dengan model pengelompokkan kata yang *exact match* pada *query*. Sistem mampu menganalisa metode *exact match boolean* (and) dengan :

- a. Kesamaan tanpa padanan kata dengan
 - Kesamaan kata dan letak berbeda
 - Kesamaan kata dan letak sama
- b. Kesamaan padanan kata dengan
 - Kesamaan kata dan letak berbeda
 - Kesamaan kata dan letak sama

4. HASIL DAN PEMBAHASAN

Sistem yang dibuat dan dirancang untuk melakukan pengujian kedua naskah baik naskah sumber teks 1 dan naskah sumber teks 2 sebagai target dengan prediksi apabila ada kemiripan dari teks 1 dan teks 2 maka mendekati sama atau tidak sama. Secara garis besar sistem terdiri dari tiga proses yaitu proses awal (*praprocessing*), proses pencarian (*searching*) dan proses perhitungan presisi dengan *flag exact match*.

a. Tahapan Metode Exact Match, Flag dan Similary

Exact Match merupakan tahapan dilakukan untuk mengecek ketepatan kata jika ketemu sama maka kata tersebut tepat dan jika tidak ketemu sama maka kata tersebut *Not Match*, sedangkan *Flag* merupakan penanda apabila benar diberi *flag* tanda 1 (*true*) *exact Match* jika tidak benar maka diberi *flag* /tanda 0 (*not match*) di dalam pembuatannya *flag* 0 tidak disertakan karena akan menambah banyak penyimpanan. Proses selanjutnya adalah hitungan *similary* menjumlahkan semua *flag* yang bernilai 1 (*true*) yang berasal dari *flag* 1 dari indeks 1 dan juga menjumlahkan *flag* 2 dari indeks 2. Hasil penjumlahan tersebut diambil nilai yang terkecil.

Flag 1 (Indeks 1)	Flag2 (Indeks 2)	hasil
1	0	0
1	1	1
0	1	0

nilai 0 tidak di lakukan penyimpanan karena akan menambah banyak penyimpanan maka misalkan *Flag* 1 (indeks 1) 1 1 1 1 1 kemudian *Flag* 2 (indeks 2) 1 1 1 maka akan diambil nilai yang terkecil yaitu jumlahnya menjadi 3 dapat diilustrasikan

indeks 1	1 1 1 1 1		1 1 1 1 1		1 1 1 1 1
indeks 2	1 1 1	→ digeser kekanan	1 1 1	→ digeser kekanan	1 1 1
hasil	1 1 1		1 1 1		1 1 1

(hasil tersebut akan selalu sama sehingga diambil yang terkecil)

Langkah Alternatif 0 Linier pengecekan yang dilakukan secara *sequensial* pengecekan satu persatu secara *linier* membutuhkan waktu yang sangat lama. Data yang A tentu tidak sama dengan B masih saja dilakukan pengecekan. Langkah Alternatif 1. Dengan mengurutkan Abjad A akan di cek dengan Abjad yang hanya A saja Abjad B akan

dicek dengan abjad yang hanya B saja metode ini masih terlalu lama sehingga langkah alternatif 2. Langkah Alternatif 2 dengan melakukan *query* abjad Ab akan dicek dengan abjad yang hanya Ab saja sehingga tidak Ab tidak akan dilakukan pengecekan data keseluruhan tetapi ini tentunya tidak efektif dan efisien karena masih terlalu lama. Sehingga pada langkah alternatif 0, 1, 2 masih terlalu banyak pengecekan sehingga dengan melakukan alternatif 3 dengan *query* kesamaan kata *exact match* melakukan pengelompokkan kata dalam *query* sehingga kata yang tepat dan sama saja yang dilakukan sehingga data yang lain yang tidak tepat akan diabaikan.

Contoh kalimat 1 : “saya makan nasi” kalimat 2 : “saya makan saya nasi”

Kalimat 1 dibuat Indeks 1 menjadi :

Nom1	Nomkata1	Kodedokumen1	Kata1	Letak1
20930	1	Tt	Saya	1
20931	2	Tt	Makan	2
20932	3	Tt	Nasi	3

Keterangan

- Nom1 : Nomor Indek 1 sumber dalam Auto Number
- NomKata1: : Nomor urut kata sumber
- KodeDokumen1 : Kode Dokumen1 diambil dari master dokumen
- Kata1 : Kata pada sumber
- Letak1 : Letak pada sumber

Kalimat 2 dibuat Indeks 2 menjadi :

Nom	Kata2	Letak2
1	Saya	1
2	Makan	2
3	Saya	3
4	Nasi	4

- Keterangan : Nom : Nomor urut kata
- Kata2 : kata pada target
- Letak2 : letak pada target

Dilakukan pengecekan kesamaan kata dan beda letak maka *query* : "saya"

Indeks 1		Indeks 2		Query
Kata1	Letak1	Kata2	Letak2	
saya	1	saya	1	Saya
		saya	3	

Akan di tampilkan kata yang sama dengan kata "saya" sesuai dengan *query* sementara yang tidak sama tidak perlu dicek dan diabaikan saja. Letak tidak diperdulikan.

Perintah dalam *query* dibuat *procedure*

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  SQL.Append('select * from '+tabel+' where '+field1+' ='+key1+' and '+field2+' ='+key2+'
  ORDER BY indeks 1.kata1');
  ExecSQL();
end;
```

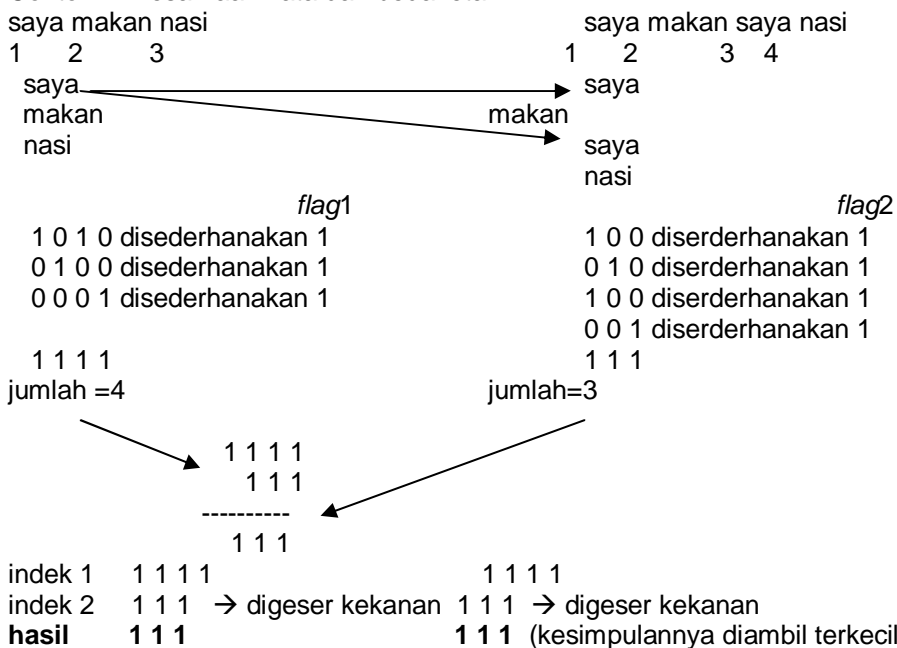
begin

```
SQL.Append('select * from '+tabel+' where '+field1+' ='+key1+' and '+field2+' ='+key2+'
  ORDER BY indeks 1.kata1');
  ExecSQL();
```

ExecSQL();

End;

Contoh 1. Kesamaan kata dan beda letak

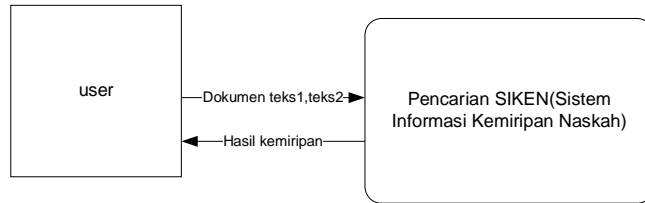


dilakukan penyederhanaan dikarenakan apabila 0 akan ditulis dan menambah banyak penyimpanan sehingga nilai 1 saja yang sama akan disimpan. Data yang berjumlah 3 X 4 akan sepanjang 12 penyimpanan data sehingga diperlukan penyederhanaan hanya yang bernilai 1 saja yang tersimpan dan 1 yang lebih dari satupun akan dihitung 1. Hasil kemiripan $\frac{3}{4} = 0.75$, diambil 3 jumlah terkecil dan di bagi 4 diambil dari banyaknya n target.

Pada *table flag 2* kode dokumen diambil dari *table target*, kolom2 untuk penyimpanan kolom2 digunakan untuk urut pada *table flag 2* sedangkan *flag 2* digunakan untuk penyimpanan kesamaan maka 1. Dari hasil *flag table flag1* dan *flag 2* dilakukan analisa yang terakhir yaitu dengan mencari kesamaan diantara kata yang sama maupun baris yang sama dan letak kemunculan kata yang sama.

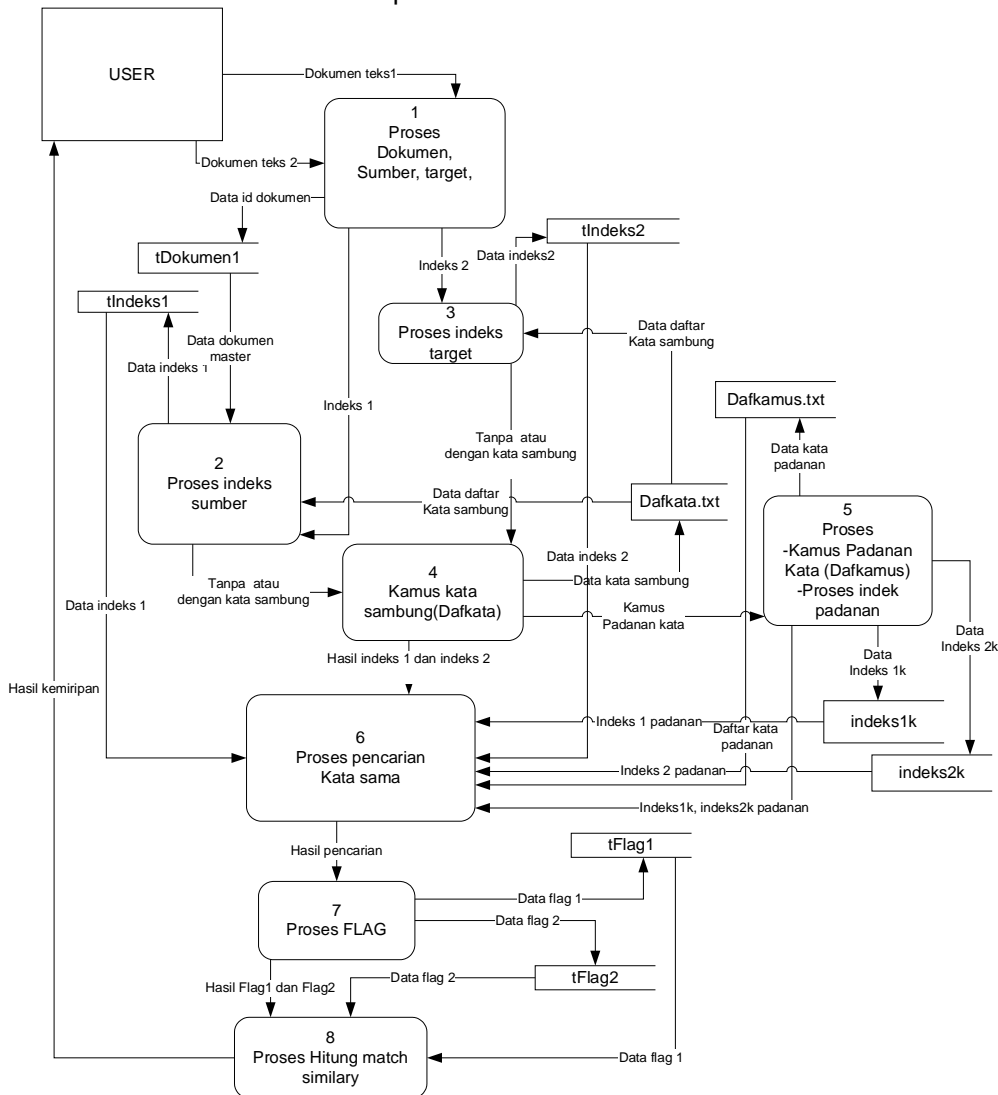
Hasil penghitungan *flag1* dan *flag2* dilakukan proses penghitungan sebagai berikut : dalam fungsi *similarly* menjumlah *flag1* dan *flag2* kemudian dicari nilai terkecil dari keduanya. Selanjutnya apabila dilakukan pengecekan pada kata sesuai dengan padanan kata dilakukan penyimpana *file* khusus untuk kamus padanan kata

Gambaran kerja sistem dijelaskan secara detail melalui *Data Flow Diagram* (DFD) Gambaran kerja sistem secara global dapat dibuat sebagai berikut :



Gambar 1. DAD Level Konteks

Penggunaan naskah teks 1 dan teks 2 diambil dari *file* dengan *extensi* *.txt atau ASCII, Data dokumen teks 1 dan teks 2 lalu akan diproses oleh sistem SIKEN (Sistem Informasi Kemiripan Naskah). Hasil perhitungan dan tampilan kemiripan naskah akan ditampilkan ke dalam hasil kemiripan kepada user. *Data Flow Diagram* Level 1 Proses DAD Level 1 untuk melihat proses awal untuk mencari indeks dari hasil input teks 1 dan teks 2.



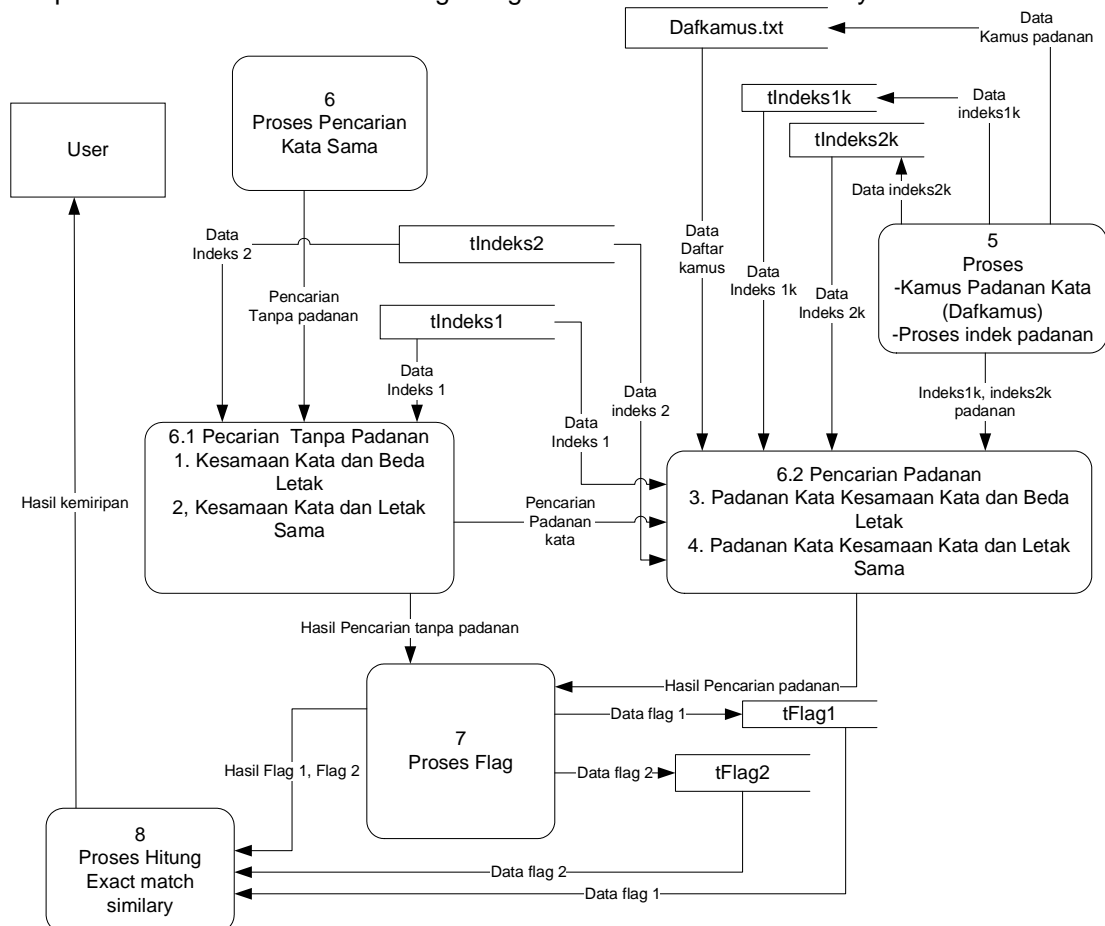
Gambar 2. DAD Level 1

Pembacaan dari dokumen sumber dan dokumen target dokumen sumber di dalam teks 1 dan dokumen target di dalam teks 2. Pada tahapan input data di proses dengan melalui tahapan awal yaitu *praprocessing* yang merupakan langkah awal pengindekan sumber menjadi indek1 dan target menjadi indek2, dimana pada tahapan awal rangkaian kata atau paragraph akan dipotong perkata dengan mengecek pada pemisahan spasi berarti sebagai *flag* untuk memecah kata-kata.

Pada DAD Level 1 tahap *praprocessing* sebelum indek1 dan indek2 dilakukan maka langkah proses kamus kata sambung untuk data kata-kata yang tidak terpakai atau ingin di

buang dan kamus padanan kata digunakan untuk mengecek kata yang sama atau sepadan. Dalam program memilih dapat menggunakan padanan kata atau tidak. Proses 1 sebagai awal untuk indeks/*praprocessor* dan proses 2 dan 3 merupakan proses untuk menentukan apakah menggunakan kata sambung atau tidak pada proses 2 dan proses 3 menentukan apakah menggunakan padanan kata atau tidak. Lalu proses ke 4 pada DAD Level 1 melakukan pengecekan kata dengan pencarian (*searching*) merupakan pengecekan indeks1 dengan indeks2 yang digunakan untuk *flag* jika tepat maka *flag* 1 dan jika tidak tepat maka 0.

Pada DAD level 2 dilakukan *flag* pada indeks 1 dan indeks 2, proses pengecekan *flag* apakah sama atau tidak dan menghitung hasil akumulasi kesamaannya.



Gambar 3. DAD Level 2

pada saat *User* memasukkan dokumen teks 1 dan dokumen teks 2, Sistem akan pemroses input teks dari user menjadi indeks 1 dan indeks 2. Lalu dari indeks 1 dan indeks 2 di cek kesamaan naskah apabila sama akan dibuat *flag* 1 dan apabila tidak sama akan dibuat *flag* 0 dengan metode *exact match*. Setelah itu melakukan pengecekan hasil *flag* 1 dan *flag* 2 dengan menampilkan hasil prediksi hitungan kemiripan naskah teks 1 dan naskah teks 2.

b. Tahapan Perbandingan *Flag*

Hasil dari indeks 1 dan indeks 2 proses diverifikasi apakah sama atau tidak. Apabila kata teks 1 dan kata teks 2 sama maka akan diberi *flag* nilai 1 sama dan 0 tidak sama. Kemudian hasil akhir akan diproses dengan metode *flag* dan di lakukan proses perhitungan. Sebelum dilakukan hasil *flag* 1 dan *flag* 2 dilakukan pencarian kesamaan kata dan beda letak, kesamaan kata dan kesamaan letak, khusus padanan kata

c. Proses Analisa *Flag* dan *Matching similiry*

Proses *flag* dan *Exact Match* dilakukan pengecekan kata apabila ketemu sama persis maka dikatakan tepat sama dengan nilai 1(*true*) dan tidak sama maka 0(*false*) dengan *variabel* sebagai berikut :

T adalah *variabel* sebagai Target

S adalah variabel sebagai Sumber

Kemiripan :

Fungsi *Similarity* :

$$Similarity (T,S) = \frac{\sum_{i=1}^n f(T_i, S_i)}{n} \quad (4.1)$$

Fungsi $f(T_i, S_i)$ didefenisikan sebagai berikut :

$$f(T_i, S_i) = \begin{cases} 1 ; T_i = S_i \\ 0 ; T_i \neq S_i \end{cases} \quad (4.2)$$

Nilai *flag* T(target) = S(sumber) maka *flag* 1

Nilai *flag* T(target) tidak sama S(sumber) maka *flag* 0

Perhitungan metode *Match* dengan aturan sebagai berikut :

S(sumber) = 1 dan T(target) = 1 maka hasil *flag* 1

S(sumber) = 0 dan T(target) = 1 maka hasil *flag* 0

S(sumber) = 0 dan T(target) = 0 maka hasil *flag* 0

Pengujian dilakukan dengan mengambil beberapa contoh naskah suatu kalimat di dalam paragraph dengan data sumber dan target dengan analisa yang berbeda-beda. Analisa dikelompokkan pada dua yaitu kesamaan tanpa padanan kata dan kesamaan dengan padanan kata.

A. Kesamaan tanpa padanan kata :

1. Kesamaan kata tetapi letak berbeda
2. Kesamaan kata tetapi letak sama

B. Kesamaan Padanan Kata :

1. Kesamaan kata tetapi letak berbeda
2. Kesamaan kata tetapi letak sama

Beberapa contoh dilakukan untuk menguji dengan hasil-hasil yang diperoleh.

1. Pengujian Sumber dan Target dengan contoh 1

Naskah Sumber (S):

sepengetahuan aku kebanyakan orang mencari rizki yang halal. kekuatan untuk mendapatkan nafkah kebanyakan orang untuk menuntut ilmu. Ilmu yang bermanfaat bagi masyarakat kepada ayah ibu bangsa dan negara dua fungsi.

Naskah Target(T) :

kebanyakan orang mencari rizki yang barokah itu pendapat aku. kekuatan untuk mendapatkan nafkah kebanyakan orang untuk menuntut ilmu dan berusaha. Ilmu yang bermanfaat bagi masyarakat bagi ibu dan bapak bangsa dan negara serta dunia mempunyai satu fungsi.

Tabel 1. Contoh 1 Pengujian

Naskah	A		B	
	1	2	1	2
S-T	0,689	0,379	0,724	0,413

Keterangan : A : Kesamaan tanpa padanan kata :

- A1 : K = L <>
A2 : K = L =

B : Kesamaan Padanan Kata :

- B1 : K = L <>
B2 : K = L =

The screenshot shows the 'FormDataTarget' application window. It has a menu bar with 'Naskah Sumber', 'Naskah Target', 'Analisa Kemiripan', and 'Setting Padanan Kata'. The main area is divided into several sections:

- INDEX 1 (SUMBER):** A table with columns 'kodebuku1', 'nomkata', 'kata1', and 'letak1'. It lists words from 'sepengatahuan' to 'halal' with their respective positions.
- INDEX 1 Padanan Kata:** A table with columns 'kodebuku1k', 'nomkata', 'kata1k', and 'letak1k'. It shows matches for the source index, such as 'saya' at position 2.
- INDEX 2 (TARGET):** A table with columns 'kodebuku2', 'nom', 'kata2', and 'letak2'. It lists words from 'kebanyakan' to 'aku'.
- INDEX 2 Padanan Kata:** A table with columns 'nom', 'kata2k', and 'letak2k'. It shows matches for the target index, such as 'saya' at position 9.
- BOBOT:** Two tables showing word weights for 'kolom1' and 'kolom2'. The first table has columns 'kodebuku', 'kolom1', and 'bobot1'. The second has 'kodebuku', 'kolom2', and 'bobot2'. Both show weights for 'c001' across different positions.
- Analisa:** A section with radio buttons for 'Kesamaan Kata dan Letak Berbeda', 'Kesamaan Kata dan Letak Sama', 'Tanpa Padanan Kata', and 'Padanan Kata'. A 'Hasil' field shows '0,413793'. There is also a 'Kembali' button.
- LAPORAN:** A section with 'Cetak Layar' and 'Cetak Kertas' buttons.
- HASIL:** A section showing results for 'Tanpa Padanan Kata' and 'Padanan Kata'. It displays 'Kesamaan Kata dan Letak Berbeda' and 'Kesamaan Kata dan Letak Sama' scores for both analysis types.

Gambar 4. Pengujian Contoh 1

5. KESIMPULAN

Secara garis besar sistem yang dibangun untuk mengetahui naskah sumber dan naskah target terdapat kemiripan atau tidak, sistem terdiri dari tiga proses yaitu proses awal (*praprocessing*) dilakukan untuk pembuatan indeks, proses pencarian (*searching exact match*) yang melalui proses *flag* dan proses perhitungan presisi dengan *flag similarity*. Program sistem SIKEN dilakukan seminimal mungkin penyimpanan indeks dengan pembacaan kelompok kata yang dilakukan *query* sehingga dengan kata yang sama saja yang dimunculkan *flag 1 (true)*.

Program Sistem Informasi Kemiripan Naskah (SIKEN) menganalisa dengan 4 analisa kesamaan yaitu :

1. Kesamaan kata tetapi letak berbeda pada kalimat-kalimat
2. Kesamaan kata tetapi letak sama pada kalimat-kalimat
3. Kesamaan padanan kata dan memadukan nomor 1
4. Kesamaan padanan kata dan memadukan nomor 2.

6. DAFTAR PUSTAKA

Achyari, Didi, 2008, *TESSY (Test of Texts Similarity)*

Bouras, 2006, *The Importance Of The Difference In Text Types To Keyword Extraction: Evaluating A Mechanism.* URL
http://ru6.cti.gr/ru6/publications/76991COMP2006_bouras.pdf

Febriyanto, R, 2008, *Penerapan Algoritma Boyer-Moore untuk Pengecekan Plagiatisme Source Code*

Lu, G, 1999, *Multimedia Database Manajemen Systems*, Artech House, Inc

Novia, W, *Kamus Lengkap Bahasa Indonesia*, Penerbit Kashiko Surabaya

Reality, Tim, *Kamus Terbaru Bahasa Indonesia*, Penerbit Reality Publisher.

Rhodes, B J., 2000., *Just-In-Time Information Retrieval, Ph.D. Thesis*, Massachusetts Institute of Technology, Cambridge, URL: <http://www.bradleyrhodes.com/papers/rhodes-phd-JITIR.pdf>
