

Penerapan Metode Levenshtein Distance Dalam Mendeteksi Hasil Jawaban Peserta Kelas Daring Dengan Teknik Web Scraping Manual

Guntur Syahputra*, Ahmad Fitri Boy**, Nurcahyo Budi Nugroho**

* Program Studi Teknologi Rekayasa Multimedia, Politeknik Negeri Lhokseumawe

** Program Studi Sistem Informasi, STMIK Triguna Dharma

Article Info

Article history:

Received Jul 12th, 2020

Revised Aug 20th, 2020

Accepted Oct 26th, 2020

Keyword:

First keyword

Second keyword

Third keyword

Fourth keyword

Fifth keyword

ABSTRACT

Pandemi COVID 19 saat ini menyebabkan kegiatan belajar mengajar harus merubah sistem tatap muka untuk beralih kepada sistem daring, hal ini bukan menjadi alasan bagi pengampu atau penanggung jawab kelas menurunkan kualitas pengajaran. Evaluasi tetap harus dilakukan melalui pertanyaan dan tugas, namun karena dilakukan secara daring banyak peserta didik yang menjawab dengan tidak serius, menyalin dari laman web ataupun menyalin jawaban teman tanpa merubah data sedikitpun yang berasal sumber salinan. Levenshtein Distance adalah metode yang digunakan untuk mendeteksi kemiripan data berbasis karakter (String Metric). Metode ini dapat membantu dan memudahkan pekerjaan pengajar dalam kelas daring untuk deteksi jawaban peserta, kemiripan jawaban yang mencapai 95% dapat dideteksi dan diasumsikan sebagai jawaban hasil salinan. Sehingga dapat diambil tindakan terhadap peserta yang melakukan hal tersebut.

Copyright © 2021 STMIK Triguna Dharma.
All rights reserved.

Corresponding Author: *First Author

Nama :

Program Studi

Sistem Informasi STMIK Triguna Dharma

Email: guntur_capt@yahoo.co.id

1. PENDAHULUAN

Pembelajaran Daring merupakan suatu tantangan yang harus dilaksanakan bagi tenaga pendidik di masa pandemi Covid 19. Hal ini merupakan sudah ditetapkan oleh peraturan KEMDIKBUD DIKTI sesuai Surat Edarannya Nomor 4 tahun 2020 tentang pelaksanaan pendidikan dalam masa darurat Covid 19 serta Surat Edaran Nomor 15 tahun 2020.

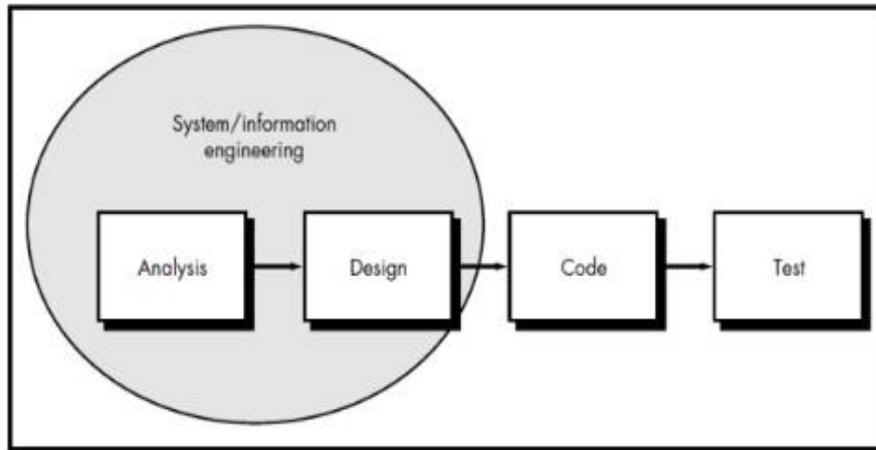
Meskipun dalam proses belajar dilakukan secara Daring para Pendidik dituntut tetap menjaga kualitas pendidikan sebagai mana mestinya sehingga Pendidik harus kreatif dan inovatif dengan menyediakan konten-konten yang menarik dan mempergunakan media yang mudah untuk di jangkau Peserta didik. Evaluasi merupakan salah satu cara untuk meningkatkan kualitas pendidikan [1], namun karena dilakukan secara daring evaluasi sering tidak efektif karena banyak peserta didik yang menjawab dengan tidak serius, menyalin dari laman web ataupun menyalin jawaban teman tanpa merubah data sedikitpun yang berasal sumber salinan. Sementara bagi pendidik untuk memeriksa dan membandingkan satu persatu jawaban perserta didik merupakan hal yang rumit membutuhkan waktu yang cukup lama. Untuk itu perlu adanya metode yang tepat sehingga siswa yang memiliki jawaban yang sama dapat dideteksi tingkat kemiripannya dan diketahui serupa dengan jawaban Peserta lain yang menjadi rujukan jawaban.

Untuk memeriksa jawaban peserta didik pada kelas daring perlu menentukan teknik web scraping/web scanning yang tepat agar proses dapat berlangsung secara efektif dan efisien, dalam penelitian ini digunakan web scraping manual dengan melihat satu persatu jawaban dan memindahkan jawaban tersebut kedalam aplikasi spreadsheet untuk selanjutnya akan diperiksa tingkat kemiripan antara satu dengan jawaban yang lain dengan metode Levenshtein Distance. Teknik web scraping manual dinilai efektif karena memiliki tingkat keakuratan tinggi dibandingkan dengan teknik web scraping lain.

Levenshtein Distance adalah metode yang digunakan untuk mendeteksi kemiripan data berbasis karakter (*String Metric*) dengan cara mengukur perbedaan jarak (*Distance*) antara 2 *String* [2]. Ditemukan oleh Vladmird Levensthein pada saat meneliti tentang kemampuan Koreksi kode biner dengan teknik *Deletions*, *Insertions* dan *Revesals* pada tahun 1965 [3]. Metode ini juga sudah banyak digunakan seperti pada penelitian yang mendeteksi identitas peserta ujian *TryOut Bimbel Expert* [4] dan banyak lagi penelitian yang berkaitan dengan deteksi kemiripan data.

2. METODE PENELITIAN

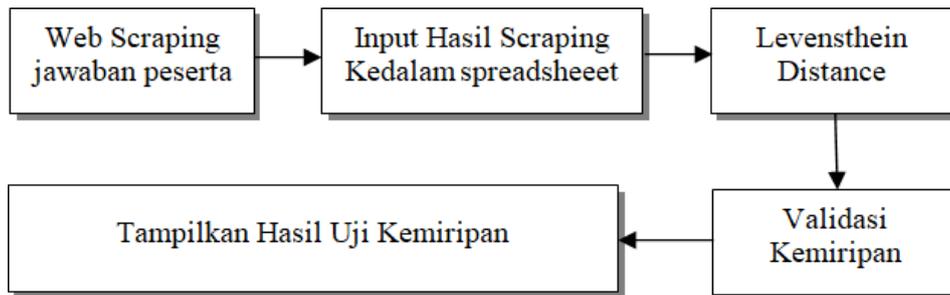
Penelitian ini dilakukan pada awal bulan mei 2020 sampai Juli 2020 (Sekitar 3 bulan) dengan melakukan *obeservasi* pada situs *google classroom* sebagai salah satu sarana kelas daring. *Linear Sequential Model* mejadi pilihan untuk metode pengembangan sistem dengan fase-fase seperti pada gambar dibawah.



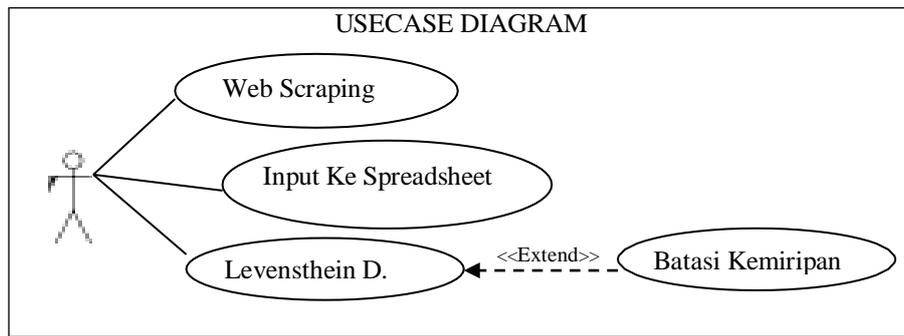
Gambar 1. Fase Metode *Linear Sequential Search*

Analisis

Berdasarkan *observasi* yang telah dilakukan maka dapat digambarkan proses yang terjadi untuk mendeteksi kemiripan jawaban peserta didik pada kelas daring adalah: 1) Lakukan Web Scraping seluruh jawaban peserta, 2) Masukkan hasil scraping kedalam *spreadsheet* yang telah ditentukan formatnya, 3) Bandingkan setiap jawaban dengan masing-masing jawaban lain dengan metode Levensthein Distance, 4) Validasi persentase tingkat kemiripan yang telah disepakati dalam penelitian ini 95%, 5) Tampilkan jawaban yang mirip.



Gambar 2. Proses Deteksi Kemiripan



Gambar 3. Usecase Diagram

Design

Untuk memudahkan proses *web scraping* maka data yang diambil diletakan terlebih dahulu kedalam spreadsheet dengan format yang telah ditentukan dan untuk selanjutnya sekaligus di upload kedalam sistem pendeteksi kemiripan jawaban, berikut format *spreadsheet*:

	A	B	C
1	NO	PESERTA	JAWABAN
2			
3			
4			
5			
6			
7			

Gambar 4. Format Spreadsheet

Setelah dibentuk format *spreadsheet* maka untuk merancang aplikasi akan dimulai dengan membentuk database yang diperlukan, berikut rancangan database yang dipakai dalam aplikasi deteksi kemiripan jawaban peserta kelas daring.

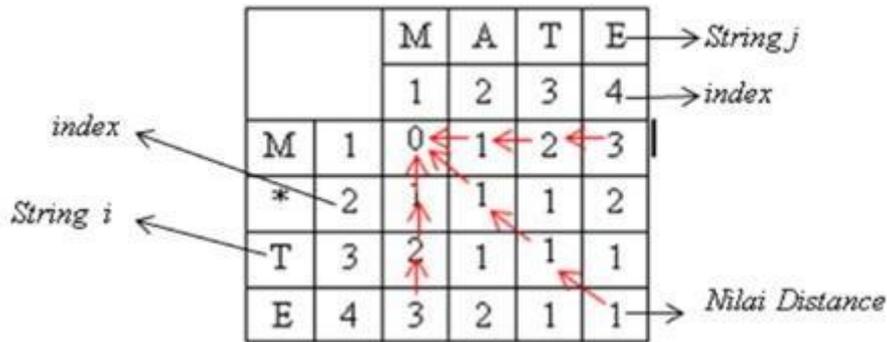
Tabel 1. Data Jawaban

No	Nama Field	Type Data	Keterangan
1	id	Int	Primary key data peserta
2	peserta	Varchar	Nama peserta
3	jawaban	Memo	Jawaban peserta
4	kemiripan	Varchar	Persentasi kemiripan dengan peserta lain

Coding

Metode *levenshtein distance* mendekteksi kemiripan berdasarkan operasi-operasi perubahan sebuah string untuk menjadi sebuah string lain dalam sebuah *string metric*. Operasi perubahan yang dipakai terdiri atas *insertion*, *deletion* dan *substitution* [4]. Untuk setiap operasi yang digunakan maka akan menambahkan nilai jarak (*Distance*) perbedaan antara string yang dibandingkan, contoh:

String 1 = M A T E
String 2 = M * T E
Subtitution = A
Distance = 1



Gambar 5. Menghitung Distance

Setelah jarak *distance* ditemukan maka tingkat kemiripan (*bobot similarity*) akan dinyatakan dengan persentase, rumus yang digunakan adalah sebagai berikut:

$$Bobot\ Similarity = \left(1 - \frac{d[m,n]}{Max(S,T)}\right) * 100\% \quad (1)$$

Dalam bahasa program PHP nilai *distance* dapat dikerjakan dengan sangat mudah, karena sudah disediakan function untuk pengujian maximal 255 karakter dalam satu perintah yaitu *levenshtein(string1, string2)*. Berikut cuplikan pemrograman hitung *distance* dari metode *levenshtein*.

```

22 for ($a=0; $a<=$n; $a++)
23 { $no=$a+1;
24   echo "<tr><td>". $no. "</td><td>". $nama[$a]. "</td><td>". $jawaban[$a].
25     "</td><td>";
26   for ($b=0; $b<=$n; $b++)
27   { if ($a!=$b)
28     {
29       $distance=levenshtein($jawaban[$a], $jawaban[$b]);
30       $similarity=(1- ($distance/max(strlen($jawaban[$a]),
31         strlen($jawaban[$b]))) *100;
32       if ($similarity>=95)
33       {
34         echo $nama[$b]. "(" .round($similarity). "%<br>";
35       }
36     }
37 }

```

Gambar 6. Bagian Program

Testing

Untuk menguji hasil program maka dipilih sebuah media kelas daring yaitu *Google Classroom* dan diambil sebuah latihan seperti yang ditunjukkan dalam gambar 7.



Gambar 7. Soal latihan pada Google Classroom

Dari contoh soal latihan diatas maka dapat dilakukan *web scraping* dan menyimpan dalam *spreadsheet* dalam penelitian digunakan Microsoft Excel, seperti pada gambar 8.

NO	PESERTA	JAWABAN
1	Cris Samuel	<pre>#include <iostream> using namespace std; int main() { string jabatan[3]='(Karyawan','Staff','Manager)'; int bonus[3]{500,600,700}; cout<<jabatan[1]<<endl;//tampil Staff cout<<bonus[2];//tampil 700 return 0; }</pre>
2	Cindy Emeliasna	<pre>#include <iostream> using namespace std; int main(){ int umur [7]; umur[0] = 15; umur[1] = 18; umur[2] = 21; umur[3] = 24; umur[4] = 27; cout<<"Nilai pada array umur elemen ke '<3<' adalah '<<umur[1]<<endl; cout<<"Nilai pada array umur elemen ke '<3<' adalah '<<umur[3]<<endl; return 0; }</pre>
3	fransisco Sitepu	<pre>#include <iostream> using namespace std; int main() { string pangkat[3]='(Kapolsek','Kapolres','Kapolda)'; int remon[3]{200,600,950}; cout<<pangkat[1]<<endl;//tampil Kapolres cout<<remon[2];//tampil 950 return 0; }</pre>
4	Immanuel Tarigan	<pre>#include <iostream> using namespace std; int main() { string jabatan[3]='(Karyawan','Staff','Manager)'; int bonus[3]{500,600,700}; cout<<jabatan[1]<<endl;//tampil Staff cout<<bonus[2];//tampil 700 return 0; }</pre>
5	Meygel Christin	<pre>#include <iostream> using namespace std; int main() { int umur[5]={25,23,22,20,18}; cout<<"umur: '<< umur[4]<<endl; }</pre>

Gambar 8. Hasil Web Scraping

Setelah Web Scraping di lakukan maka selanjut pengujian kemiripan dilakukan melalui aplikasi yang sudah dibuat untuk menerapkan Algoritma Levenshtein Distance, dan hasilnya dapat dilihat dibawah ini.

Hasil Uji Kemiripan Jawaban Peserta			
NO	PESERTA	JAWABAN	KEMIRIPAN
1	Cris Samuel	<pre>{ string jabatan[3]='(Karyawan','Staff','Manager)'; int bonus[3]{500,600,700}; cout<</pre>	Immanuel Tarigan(100%)
2	Cindy Emeliasna	<pre>{ int umur [7]; umur[0] = 15; umur[1] = 18; umur[2] = 21; umur[3] = 24; umur[4] = 27; cout<<"Nilai pada array u</pre>	
3	fransisco Sitepu	<pre>{ string pangkat[3]='(Kapolsek','Kapolres','Kapolda)'; int remon[3]{200,600,950}; cout<</pre>	
4	Immanuel Tarigan	<pre>{ string jabatan[3]='(Karyawan','Staff','Manager)'; int bonus[3]{500,600,700}; cout<</pre>	Cris Samuel(100%)
5	Meygel Christin	<pre>{ int umur[5]={25,23,22,20,18}; cout<<"umur: '<< umur[4]<</pre>	

Gambar 9. Hasil Deteksi Kemiripan Jawaban

Pada pengujian diatas kemiripan hanya dimunculkan apabila bobot *similarity*-nya lebih besar dari 95% sehingga dapat disimpulkan jawaban tersebut memang merupakan jawaban yang dikerjakan dengan cara menyalin bukan dikerjakan secara mandiri, dan hasilnya dari 5 orang yang dijadikan sampel ditemukan 2 orang memiliki jawaban yang sangat mirip mencapai 100%. Setelah diperiksa pada jawaban siswa pada *classroom* ternyata jawabannya sama persis dapat dilihat pada gambar dibawah ini.

<pre>chris samuel 19 Jun #include <iostream> using namespace std; int main() { string jabatan[3]="Karyawan","Staff","Manager"; int bonus[3]={500,600,700}; cout<<jabatan[1]<<endl;//tampil Staff cout<<bonus[2];//tampil 700 return 0; }</pre>	<pre>Immanuel Tarigan 19 Jun #include <iostream> using namespace std; int main() { string jabatan[3]="Karyawan","Staff","Manager"; int bonus[3]{500,600,700}; cout<<jabatan[1]<<endl;//tampil Staff cout<<bonus[2];//tampil 700 return 0; }</pre>
--	---

Gambar 10. Jawaban Peserta

3. KESIMPULAN

Penerapan Algoritma *Levenshtein Distance* sudah teruji untuk mendeteksi kemiripan data berupa string yang dapat dilihat pada penelitian-penelitian sebelumnya dan dalam penelitian ini algoritma *Levenshtein Distance* juga sudah berhasil mendeteksi kemiripan jawaban peserta kelas daring. Harapan untuk perkembangan kedepannya agar deteksi jawaban bisa berlangsung dengan cepat, dengan menggunakan metode *web scraping* otomatis yang juga memperhatikan keakuratan hasil.

4. UCAPAN TERIMA KASIH

Penulis menyadari tidak akan berhasil bila tidak didukung oleh orang-orang yang terlibat didalamnya untuk itu ucapan terima kasih yang sebesar-besarnya disampaikan kepada peneliti-peneliti sebelumnya yang menjadi dasar teoritis yang mendukung penelitian ini, peserta kelas daring sebagai sampel penelitian kelas 2mial Tahun Ajaran Genap 2019/2020 di Kampus STMIK Trigna Dharma, serta seluruh yang terlibat langsung maupun tidak langsung.

REFERENSI

- [1] B. Mahirah, "Evaluasi Belajar Peserta Didik (Siswa)," *Idaarah J. Manaj. Pendidik.*, vol. 1, no. 2, pp. 257–267, 2017, doi: 10.24252/idaarah.v1i2.4269.
- [2] B. P. Pratama and S. A. Pamungkas, "Analisis Kinerja Algoritma Levenshtein Distance dalam Mendeteksi Kemiripan Dokumen Teks," *J. "LOG!K@,"* 2016.
- [3] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Sov. Phys. Dokl.*, 1966.
- [4] J. R. S. Sagala Guntur, "Deteksi Tingkat Kemiripan Data Siswa Peserta Ujian Berbasis Komputer Pada Bimbingan Belajar Expert Dengan Menggunakan Metode Levenshtein Distance," *Algoritm. J. Ilmu Komput. Dan Inform.*, 2017.