

## Penerapan Algoritma Jaro-Winkler Untuk Autocorrect Dan Spelling Suggestion Pada Aplikasi Speech Recognition Cms Berbasis Website

I Gede Adi Wijaya<sup>1)</sup> Tri Sagirani<sup>2)</sup> Norma Ningsih<sup>3)</sup>

Program Studi/Jurusan Sistem Informasi  
Universitas Dinamika

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email : 1)[adyw19@gmail.com](mailto:adyw19@gmail.com), 2)[tris@dinamika.ac.id](mailto:tris@dinamika.ac.id), 3)[norma@dinamika.ac.id](mailto:norma@dinamika.ac.id)

**Abstract:** *PT. Bukaloka Teknologi Indonesia is an online buying and selling site that distributes products and services equipped with an online store & instant web generation platform where this service is a platform that integrates the marketplace with Online Company Profile creation services for micro, small, and medium enterprises (UMKM) who joined. Currently, the process of changing UMKM Company Profile information still has to use a standard Content Management System (CMS) which requires changes to be made in the CMS dashboard itself and data entry is done by typing in the data form. This study discusses the implementation of Speech Recognition which is supported by the Jaro-Winkler Algorithm which can simplify and speed up the process of entering words compared to using CMS without the need for the use of a keyboard. The Jaro-Winkler algorithm itself is intended to improve words and the process of word accuracy on potential word errors generated by Speech Recognition. 6 prospective users were surveyed to determine the feasibility of the application in the community and get the results of the survey calculations with the number of responses saying "Yes" as many as 87.20%, while "No" as many as 12.8%. With a total of 87.20%, it can be concluded that the result is "Successful".*

**Keywords:** *Speech Recognition, Jaro-Winkler Algorithm*

PT. Bukaloka Teknologi Indonesia merupakan sebuah *startup* yang bertempat di pulau Bali dengan Putu Aditya Santana sebagai Founder & CEO PT. Bukaloka Teknologi Indonesia dan berdiri Oktober 2016 di Art Center, Denpasar, Bali. PT. Bukaloka Teknologi Indonesia merupakan situs jual beli online yang mendistribusikan produk dan jasa yang dilengkapi dengan *platform generate toko online & web instant* yang dimana layanan ini merupakan sebuah *platform* yang mengintegrasikan *marketplace* dengan layanan pembuatan *Company Profile Online* untuk usaha mikro, kecil, dan menengah (UMKM) yang ikut bergabung.

Saat ini proses yang masih berjalan di PT. Bukaloka Teknologi Indonesia untuk saat ini dalam pembuatan situs toko online UMKM yang akan bergabung dengan PT. Bukaloka masih menggunakan *Content Management System (CMS)* yang biasa kita jumpai pada saat membuat blog pada [www.blogger.com](http://www.blogger.com), Metode *entry data* yang akan ditampilkan ke *Company Profile* pada saat ini masih harus menuju ke halaman CMS dan harus menggunakan perangkat *input* seperti keyboard, yang kebanyakan orang yang masih

awam dalam mengetik untuk menghasilkan beberapa kata-kata yang typo selain itu proses pengetikan untuk data yang banyak membutuhkan waktu yang lama, berdasarkan data observasi dari 30 Owner UMKM yang bergabung mempunyai kecepatan rata-rata 19-25 kata per menitnya, kecepatannya masih tergolong sangat rendah. *Speech Recognition* sangat cocok untuk menjawab permasalahan tersebut. dimana *speech recognition* sangat berguna untuk mempercepat proses pembuatan konten, karena proses input data dengan berbicara jauh lebih mudah digunakan oleh orang yang awam dalam mengetik menggunakan *keyboard*, maka peneliti mengusulkan untuk pengimplementasian *Speech Recognition* kedalam *Content Management System (CMS)* menjadi pengganti perangkat input seperti *keyboard*.

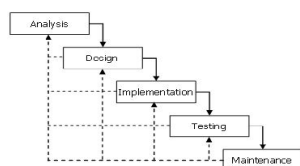
*Speech Recognition* adalah salah satu bidang kecerdasan mesin yang sedang berkembang pesat, hal itu ditandai oleh hampir semua *device* teknologi dilengkapi oleh *voice command*. Hal ini telah menarik bagi para peneliti untuk menjadikan *Speech Recognition* sebagai disiplin ilmu yang penting untuk menciptakan dampak teknologi pada masyarakat dan

diharapkan akan berkembang lebih jauh di bidang interaksi mesin dengan manusia. *Speech Recognition* sendiri adalah proses menangkap kata yang diucapkan melalui mikropon atau telepon dan mengubahnya ke dalam kata-kata yang tersimpan secara digital (Jelinek, 1997). Untuk membatasi kesalahan kata yang dihasilkan oleh *Speech Recognition* ini maka akan digunakan algoritma *Jaro-Winkler*. Algoritma *Jaro-Winkler* merupakan varian dari *Jaro Distance* metrik yaitu sebuah algoritma untuk mengukur kesamaan antara dua buah string, biasanya algoritma ini digunakan di dalam pendeteksian duplikat dan dapat dialih fungsikan menjadi fitur *Autocorrect* dan *Spelling Suggestion*.

Penggunaan *Speech Recognition* sendiri akan dapat mempermudah dan mempercepat, proses pemasukan kata yang akan menjadi konten dari *website* dibandingkan dari penggunaan *Content Management System* (CMS) data tanpa perlu adanya penggunaan *keyboard* yang akan meminimalkan potensi kesalahan kata serta proses pengetikan yang lama. Algoritma *Jaro-Winkler* dimaksudkan membenarkan kata dan proses pengakuratan kata pada potensi kesalahan kata yang dihasilkan oleh *Speech Recognition*. Berdasarkan uraian di atas diperlukan penerapan suatu metode yang mengimplementasikan Teknologi *Speech Recognition* pada CMS berbasis *website* dengan menggunakan Algoritma *Jaro-Winkler* guna membantu proses *Autocorrect* dan *Spelling Suggestion*, yang memudahkan para UMKM pada proses input data dan meminimalisir potensi kesalahan kata dalam mengembangkan *Company Profile*.

## METODE

Metode pengembangan yang digunakan adalah model *Software Development Life Cycle* (SDLC). Dari beberapa varian Model SDLC, *Waterfall Model* (Bassil,2017). Akan digunakan dalam penelitian ini, yang Gambaran tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Metode *Waterfall* (Bassil, 2017)

### 1. Analysis

*analysis* merupakan tahapan paling awal dimana diperlukannya komunikasi dari narasumber yang bertujuan untuk mengidentifikasi dan memperkirakan kebutuhan software yang mungkin dibutuhkan pengguna serta batasan dari software tersebut. Informasi seperti ini dapat diperoleh melalui proses wawancara, observasi lapangan, menganalisa proses bisnis, dan diskusi.

#### a. Wawancara

Wawancara dilakukan dengan Staff bagian IT PT. Bukaloka Teknologi Indonesia yang dilakukan 2 May 2020 pukul 11.40 WIB ~ 12.25 WIB via Whatapps Call, dikarenakan tidak mendukungnya kondisi dan permasalahan jarak kantor PT. Bukaloka Teknologi Indonesia.

#### b. Observasi

Dilanjutkan dengan proses observasi beberapa kebutuhan data, untuk mencari berbagai kebutuhan data yang akan diperlukan pada saat pembuatan aplikasi dimulai seperti, pencarian nilai standar minimal dan maksimal pada proses *autocorrect & spelling suggestion*.

#### c. Analisis Proses Bisnis

Dimulai dengan ketika adanya UMKM yang ingin bergabung dengan *marketplace* PT. Bukaloka, dengan mendaftar melalui form yang dapat diakses *online* atau datang ke kantor secara langsung. yang dimana setelah pendaftaran selesai akan mendapatkan layanan pembuatan *Company Profile Online* untuk usaha UMKM tersebut. Proses pembuatan *Company Profile*-nya sendiri menggunakan *Content Management System* (CMS) yang sudah tersedia.

#### d. Identifikasi Masalah

Berdasarkan proses bisnis yang ada pada PT. Bukaloka Teknologi Indonesia tersebut, maka dapat dilakukan identifikasi permasalahan. Identifikasi permasalahan mencakup permasalahan yang ada, dampak dari permasalahan tersebut, dan solusi yang diusulkan. Berikut adalah hasil dari identifikasi masalah dan alternatif yang terdapat pada tabel 1.

Tabel 1. Identifikasi Masalah

Masalah	Solusi
Pada proses <i>input</i> data adanya kata-kata yang <i>typo</i>	Membuat sistem yang mempunyai fitur <i>autocorrect</i> dan <i>spelling suggestion</i> untuk mengurangi potensi kesalan kata
Proses pengetikan untuk data yang	Membuat sistem yang dapat memasukan

banyak membutuhkan waktu yang lama	data menggunakan perangkat input seperti <i>keyboard</i>	tanpa
------------------------------------	--	-------

e. Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna dilakukan dengan tujuan untuk mengetahui data dan informasi yang digunakan dan/atau dibutuhkan oleh pengguna sistem (perangkat lunak) yang akan dibuat. Berdasarkan hasil analisis proses bisnis, pengguna dari sistem (perangkat lunak) berjumlah 3 pengguna yaitu: Staff IT UMKM, Owner UMKM dan Staff IT Bukaloka dengan kebutuhan data seperti: data produk, data about\_us, data event, data testimoni, dan data perusahaan.

2. Design

Tahap design adalah proses yang mengubah kebutuhan-kebutuhan yang sudah disebutkan diatas menjadi bentuk *software blueprint*, sebelum proses pengkodean dapat dimulai.

a. System Flowchart

*System flowchart* memberikan gambaran alur proses dalam Penerapan Algoritma *Jaro-Winkler* Untuk *Autocorrect* Dan *Spelling Suggestion* Pada Aplikasi *Speech Recognition* CMS Berbasis *Website*.

b. Context Diagram

*Context diagram* dari Penerapan Algoritma *Jaro-Winkler* untuk *Autocorrect* dan *Spelling Suggestion* pada Aplikasi *Speech Recognition* CMS berbasis *Website* ini mempunyai tiga entitas pelaku diantaranya adalah Staff IT UMKM, Owner UMKM, dan Staff IT PT. Bukaloka.

c. Hierarchy Input Process Output (HIPO)

Diagram HIPO memberikan gambaran proses dan sub-proses dari penelitian ini. Pada aplikasi ini terdapat 4 proses utama yaitu *Speech Recognition*, Algoritma *Jaro-Winkler*, *Autocorrect & Spelling Suggestion*, dan Validasi Data.

d. Data Flow Diagram

*Data Flow Diagram* (DFD) merupakan *context diagram* yang dikembangkan lagi secara rinci. DFD penelitian ini memiliki lima proses utama yaitu posting konten menggunakan *speech recognition*, perhitungan algoritma *jaro-winkler* (dj), perhitungan algoritma *jaro-winkler* (dw), *autocorrect & spelling suggestion*, dan validasi. Memiliki total tiga entitas yaitu Staff IT UMKM, Owner, dan Staff IT PT. Bukaloka.

e. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) merupakan sebuah model yang digunakan untuk menjelaskan hubungan antar data yang berdasarkan objek-objek yang mempunyai relasi. ERD dapat disajikan dalam dua bentuk yaitu *Conceptual Data Model* (CDM) dan hasil *generate* dari CDM yaitu *Physical Data Model* (PDM).

• *Conceptual Data Model* (CDM)

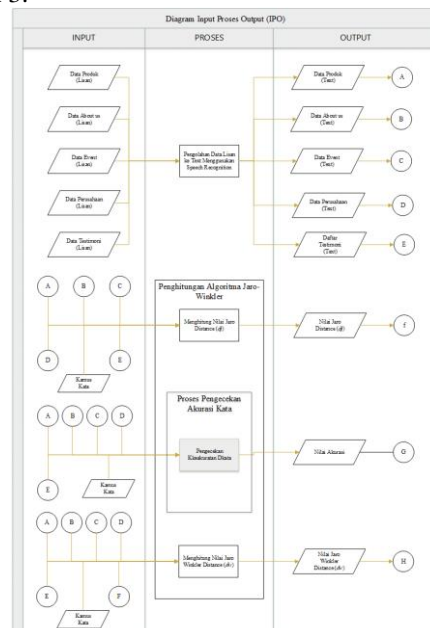
Desain dari Penerapan Algoritma *Jaro-Winkler* untuk *Autocorrect* dan *Spelling Suggestion* pada Aplikasi *Speech Recognition* CMS berbasis *Website* ini dapat dirancang dalam bentuk model logika yang dapat digambarkan dengan CDM, hal ini berfungsi untuk mengidentifikasi atribut, entitas serta hubungan antar entitas.

• *Physical Data Model* (PDM)

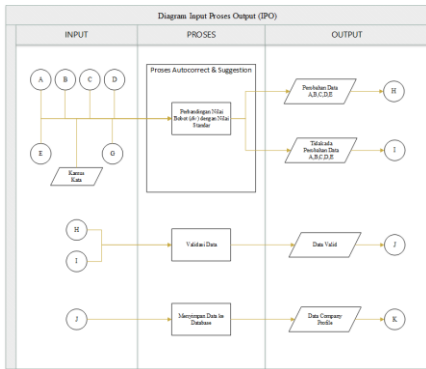
*Physical Data Model* (PDM) merupakan hasil *generate* dari CDM yang bertujuan untuk menggambarkan struktur table yang mempunyai hubungan yang dimana akan diterapkan di *Database Management System* (DBMS).

f. Diagram Input Process Output (IPO)

Selanjutnya akan dilakukan perancangan sistem yang menggunakan pemodelan sistem IPO Diagram untuk menggambarkan kebutuhan input, proses, dan output dari setiap modul. Gambaran tahapan penelitian dapat dilihat pada Gambar 2 dan 3.



Gambar 2. Input Process Output part 1



Gambar 3. Input Process Output part 2

### 3. Implementation

Pada tahap *implementation*, akan dilakukan proses membangun aplikasi berdasarkan *system flowchart*, *Context diagram*, DFD, dan IPO diagram yang telah dibuat. Proses akan terbagi menjadi 3 bagian dimana pertama-tama dilakukannya pembuatan database lalu pembuatan aplikasi secara plain lalu akan dilanjutkan dengan penerapan algoritma yang digunakan.

### 4. Testing

Setelah proses implementasi selesai, proses pengujian pun dapat dilakukan. Aplikasi akan diuji dengan metode *black box* untuk mengetahui tingkat keberhasilan dari fitur yang ada dalam sistem. Serta adanya *Usability Test* yang dilakukan kepada 6 calon pengguna aplikasi yang digunakan untuk mengukur apakah aplikasi dapat diterima di masyarakat.

### 5. Maintenance

Proses *maintenance* dilakukan peneliti untuk melakukan beberapa perbaikan terhadap bagian yang terdapat *error*, proses ini akan dilakukan secara berkala. Sehingga akan memudahkan peneliti tanpa harus mengulangi semua tahap kembali.

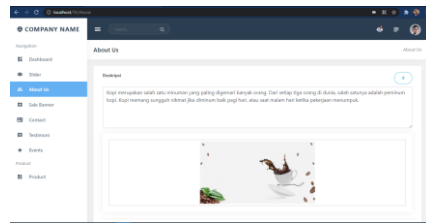
## HASIL DAN PEMBAHASAN

### 1. Hasil Implementasi

Berikut merupakan hasil implementasi dari Penerapan Algoritma *Jaro-Winkler* Untuk *Autocorrect* dan *Spelling Suggestion* Pada Aplikasi *Speech Recognition* CMS Berbasis *Website*.

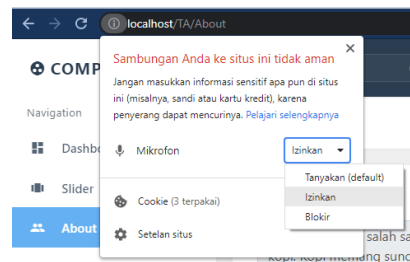
- a. Penerapan *Speech Recognition* di *Dashboard* CMS

Penerapan algoritma pada *Speech Recognition* di *Dashboard* CMS ini berlokasi pada seluruh halaman *dashboard* yang biasa digunakan oleh admin untuk mengubah informasi *company profile*. Pada tahap ini sama seperti proses *input/edit* data pada CMS tradisional pengguna diharuskan untuk *login* agar dapat memasuki halaman *dashboard*, lalu pengguna dapat memilih menu yang akan di *input/edit* datanya. Tampilan *Speech Recognition* pada *Dashboard* CMS dapat dilihat pada Gambar 4.



Gambar 4. *Speech Recognition* pada *Dashboard* CMS

Fitur *speech recognition* ini tidak bersifat *default* melainkan *optional*, yang berarti fitur ini dapat diaktifkan jika ingin menggunakan fitur dan dapat dinonaktifkan jika tidak ingin menggunakan fitur ini. Perlu diperhatikan untuk memastikan bahwa *browser* yang digunakan harus sudah memberikan izin pada aplikasi untuk dapat mengakses *microphone* sebelum fitur digunakan, sehingga *speech recognition* dapat menangkap suara. Langkah paling mudah untuk mendapatkan izin pada *browser Google Chrome* adalah dengan menekan *icon* yang terdapat pada samping kiri URL aplikasi, lalu ubah izin mikrofon dari pengaturan “Tanyakan(*default*) / Ask(*default*)” menjadi “Izinkan / Allow”, tampilan dapat dilihat pada Gambar 5.



Gambar 5. Pemberian akses mikrofon

Dalam proses pengolahan data suara menjadi teks setelah data ditangkap oleh *Speech Recognition* diperlukan beberapa tahapan seperti *Natural Language Processing* dan Algoritma

*Jaro-Winkler Distance*. Berikut merupakan contoh beserta tahapan prosesnya.

Sampel kata yang akan digunakan dalam uji coba adalah **“kopi merupakan minuman yang paling banyak digemari orang”**

- Tahapan *Natural Language Processing*

Sebelum dapat mengeluarkan output data berupa teks, sistem akan memproses terlebih dahulu data suara yang ditangkap oleh *speech recognition* lalu sistem mengekstrak gelombang suara lebih jauh dan merepresentasi data tersebut dari suatu teks bebas. Untuk dapat mengubah data suara menjadi data teks, dapat menggunakan kode seperti pada kode berikut ini.

```
const SpeechRecognition = window.SpeechRecognition ||
window.webkitSpeechRecognition;
const recognition = new SpeechRecognition();

recognition.continuous = true;

recognition.onstart = function(params) {
  console.log('Voice Recognition is activated');
};

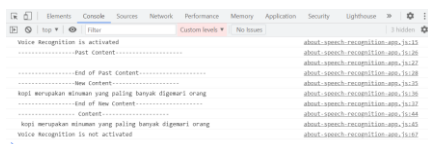
recognition.onresult = function(event) {
  var past_content = document.getElementById('content').value;

  const current = event.resultIndex;
  const transcript = event.results[current][0].transcript;

  content = past_content+" ";
  content += transcript;

  $('#content').val(content);
};
```

Setelah data diproses oleh sistem data teks akan terbentuk yang dimana merupakan hasil dari *Natural Language Processing* yang ditangkap oleh *Speech Recognition*. Hasil pemrosesan data dapat dilihat pada Gambar 5.



Gambar 5. Hasil pemrosesan suara menjadi teks

- Tahapan Algoritma *Jaro-Winkler Distance*

Pada tahap ini data teks yang sudah didapatkan akan melalui pengecekan kata menggunakan Algoritma *Jaro-Winkler Distance*. Dalam tahap ini terdapat 2 proses yaitu pencarian nilai *“dj”* (*jaro-distance*). Pencarian nilai *“dj”* dapat menggunakan kode seperti berikut ini.

```
function Jaro( $s1, $s2 ){
  $s_lth1 = strlen( $s1 );
  $s_lth2 = strlen( $s2 );
```

```
$jarak = (int) floor(min( $ s_lth1, $ s_lth2 ) / 2.0);
// get common characters
$kesamaan1 = getCommonCharacters( $ s1, $ s2, $jarak);
$kesamaan2 = getCommonCharacters( $ s2, $ s1, $jarak);

if( ($kesamaan1_len = strlen( $kesamaan1 )) == 0) return 0;
if( ($kesamaan2_len = strlen( $kesamaan2 )) == 0) return 0;

// menghitung transposisi
$transposisi = 0;
$awalan = min( $kesamaan1_len, $kesamaan2_len );
for( $i = 0; $i < $awalan; $i++){
  if( $kesamaan1[$i] != $kesamaan2[$i] ) $transposisi++;
}
$transposisi/= 2.0;

// mengembalikan data ke jaro distance
return ( $kesamaan1_len/($s_lth1) + $ kesamaan2_len/($s_lth2) +
($kesamaan1_len - $transposisi)/($kesamaan1_len) ) / 3.0;
}
```

Lalu dilanjutkan dengan pencarian nilai *“dw”* (*jaro-winkler distance*) yang dimana nilai *“dj”* yang didapatkan sebelumnya akan kembali dihitung kembali dengan nominasi kata dasar untuk menghasilkan nilai *“dw”*, semakin dekat nilai *“dw”* dengan nilai 1 maka dapat dikatakan kedua kata adalah serupa. Nilai *“dw”* ini juga berfungsi sebagai nilai bobot dari kata yang di cek. Pencarian nilai *“dw”* dapat menggunakan kode seperti berikut ini.

```
function JaroWinkler($string1, $string2, $PREFIXSCALE = 0.1 ){
  $JaroDistance = Jaro( $string1, $string2 );
  $prefixLength = panjang_prefix ( $string1, $string2 );
  return $JaroDistance + $prefixLength * $PREFIXSCALE * (1.0 -
  $JaroDistance);
}

function panjang_prefix( $s1, $s2, $pjm_minimal = 4 ){
  $n = min( array( $pjm_minimal, strlen($s1), strlen($s2) ) );
  for($i = 0; $i < $n; $i++){
    if( $s1[$i] != $s2[$i] ){
      return $i;
    }
  }
  return $n;
}
```

Dengan selesainya seluruh proses penghitungan maka didapatlah hasil dari penghitungan *Jaro-Winkler Distance*. Hasil dari perhitungan *Jaro-Winkler Distance* menghasilkan data seperti Tabel 2.

Tabel 2. Hasil perhitungan *Jaro-Winkler Distance*

No	Data		<i>Jaro Winkler</i>	
	Input	Kamus	<i>dj</i>	<i>dw</i>
1	kopi	kopi	1	1

2	merupakan	merupakan	1	1
3	minuman	minuman	1	1
4	yang	yang	1	1
5	paling	paling	1	1
6	banyak	banyak	1	1
7	digemari	gemar	0.87	0.87
8	orang	orang	1	1

b. Penerapan *Speech Recognition* di *Company Profile*

Penerapan algoritma pada *Speech Recognition* di *Company Profile* ini berlokasi pada seluruh halaman *company profile* yang biasa dilihat pada saat pertama kali membuka website. Penambahan fitur *speech recognition* yang disertai algoritma bertujuan untuk mempermudah pengeditan *company profile* tanpa harus pergi ke halaman *dashboard* CMS serta memberikan inovasi yang dimana pengguna akan bisa mengubah informasi didalamnya hanya dengan perintah suara. Penggunaan *speech recognition* di *company profile* ini hanya dapat diakses melalui “live demo” pada *dashboard* yang mempunyai icon mata. Tampilan *Speech Recognition* pada *Company profile* dapat dilihat pada Gambar 6.



Gambar 5. *Speech Recognition* pada *Company Profile*

Proses penggunaan *speech recognition* pada *company profile* ini sedikit berbeda dengan penggunaan pada halaman *dashboard* CMS, penggunaan *speech recognition* pada halaman ini memiliki perintah kata yang harus digunakan untuk mengubah informasi sesuai dengan *component* yang ingin diubah. Untuk mengetahui perintah kata pengguna dapat berkata “bantuan” sebagai bantuan untuk mengetahui urutan perintah kata yang tepat dan juga sistem akan memberikan bantuan ketika pengguna menyebutkan perintah kata yang salah.

Proses pengolahan data yang terdapat di *speech recognition* pada *company profile* ini kurang lebih sama seperti pengolahan data yang terdapat di *speech recognition* pada *dashboard* CMS. Serta terdapatnya sebuah AI (*Artificial*

*Intelligence*) *Top-down* sederhana yang bertugas sebagai asisten untuk memberikan bantuan untuk menginformasikan perintah kata yang harus digunakan untuk mengubah informasi pada *company profile*. Untuk dapat berinteraksi dengan pengguna AI *Top-down* sederhana menggunakan kode seperti berikut ini.

```
function Slider_readOutLoud(message) {
  const speech = new SpeechSynthesisUtterance();
  speech.volume = 1;
  speech.rate = 1;
  speech.text = "Aku gak ngerti, bisa diulangi?";

  if (message.includes('ubah') || message.includes('edit') ||
  message.includes('Ubah')) {
    if (message.includes('komponen')) {
      if (message.includes('slogan')) {
        const slogan = document.querySelector('.slider-speech-
        slogan-1');
        var newmessage = message.substring(21, message.length);
        slogan.textContent = newmessage;
        speech.text = "Slogan siap";
      } else if (message.includes('nama produk')) {
        const produk = document.querySelector('.slider-speech-
        produk-1');
        var newmessage = message.substring(27, message.length);
        produk.textContent = newmessage;
        speech.text = "nama produk siap";
      } else if (message.includes('deskripsi')) {
        const deskripsi = document.querySelector('.slider-speech-
        deskripsi-1');
        var newmessage = message.substring(24, message.length);
        deskripsi.textContent = newmessage;
        speech.text = "deskripsi produk siap";
      } else {
        speech.text = "Perintah kurang lengkap // contoh :
        'ubah'+ 'komponen'+ 'slogan. atau nama produk. atau deskripsi.'+ 'kata
        slogan baru";
      }
    } else {
      speech.text = "Perintah kurang lengkap // contoh :
      'ubah'+ 'komponen'+ 'slogan'+ 'kata slogan baru";
    }
  } else if (message.includes('bantuan')) {
    speech.text = "Perintah kurang lengkap // contoh :
    'ubah'+ 'komponen'+ 'slogan'+ 'kata slogan baru";
  }
  // content.textContent = speech.text;
  console.log("-----Jawaban-----");
  console.log(speech.text);
  window.speechSynthesis.speak(speech);
}
```

2. Testing

Berikut merupakan hasil testing Penerapan Algoritma *Jaro-Winkler* Untuk *Autocorrect* dan *Spelling Suggestion* Pada Aplikasi *Speech Recognition* CMS Berbasis *Website*.

a. Testing Penerapan *Speech Recognition* di *Dashboard* CMS

Membantu pengguna memasukan banyak data yang digunakan untuk informasi yang akan dimunculkan pada *company profile* hanya dengan menggunakan suara tanpa harus mengetik menggunakan *keyboard*. Berikut adalah tahapan proses testing pada Tabel 3.

Tabel 3. Testing *Speech Recognition* di *Dashboard CMS*

No	Deskripsi	Tahap	Input	Output
1	Masuk ke halaman <i>dashboard CMS</i>	Menuju ke halaman login	Masukka n <i>email</i> dan <i>password</i>	Halaman <i>dashboard</i> tampil
2	Memilih bagian <i>Company Profile</i> yang akan diubah	Menuju ke halaman menu yang dituju	Klik menu tujuan.	Halaman menu tampil
3	Menggunakan <i>speech recognition</i> untuk input data	Memilih komponen yang akan diubah informasinya	Klik button dengan <i>icon</i> mikrofon lalu bersuara	Sistem menghasilkan data text

b. Testing Penerapan *Speech Recognition* di *Company Profile*

Membantu pengguna memasukan data yang digunakan untuk informasi yang akan dimunculkan pada *company profile* hanya dengan menggunakan suara tanpa harus mengetik menggunakan *keyboard* dan tanpa perlu pergi ke halaman *dashboard*. Berikut adalah desain testing untuk Penerapan Algoritma pada *Speech Recognition* di *Company Profile*. Berikut adalah tahapan proses testing pada Tabel 4.

Tabel 4. Testing *Speech Recognition* di *Company Profile*

No	Deskripsi	Tahap	Input	Output
1	Masuk ke halaman <i>Live Demo</i>	Menuju ke <i>dashbo</i>	Klik button dengan <i>icon</i> mata	Halaman <i>Live Demo</i> tampil
2	Menggunakan <i>speech recognition</i> untuk input data	Memilih komponen yang akan diubah informasinya	Klik button dengan <i>icon</i> mikrofon lalu bersuara menggunakan urutan perintah suara.	Sistem menghasilkan data text

3. Hasil Uji Coba Lapangan

Berikut merupakan hasil uji coba dengan calon pengguna aplikasi “Penerapan Algoritma *Jaro-Winkler* Untuk *Autocorrect* dan *Spelling Suggestion* Pada Aplikasi *Speech Recognition CMS* Berbasis *Website*”. Uji Coba yang akan ditanyakan kepada pengguna menggunakan

*usability test script* (Steven Krug, 2020). dengan total 6 pengguna.

a. Menyusun Tugas *Usability test*

Pembuatan tugas *usability testing* yang akan ditanyakan kepada pengguna yang telah dipilih bersarkan *learnability, efficiency, memorability, errors* dan *satisfaction*.

b. Pengujian Aplikasi

Proses pengujian aplikasi dilakukan kepada 6 *Owner* *UMKM* yang akan menjadi calon pengguna aplikasi untuk mencoba menggunakan aplikasi beserta fitur yang tersedia didalamnya.

c. Survei Dengan Kuisisioner

Survei dengan kuisisioner merupakan tahap dimana dilakukannya penyebaran kuisisioner kepada para pengguna untuk mendapatkan pendapat pengguna. Survei diberikan kepada total 6 *Owner* *UMKM* yang akan memberikan pendapatnya.

d. Analisa Data Hasil *Usability Test* dan Survei

Proses analisa dilakukan dengan menghitung prosentase pada Task di “Formulir Uji Coba Ketergunaan”. Perhitungan ini dilakukan dengan menggunakan bentuk angka dari setiap jawaban dengan aturan angka 1 (satu) apabila jawaban “Ya” atau tanda centang (✓) dan angka 0 (nol) diberikan kepada jawaban “Tidak”. Dari setiap point Task dihitung prosentasenya, berapa persen yang menjawab “Ya” dan berapa persen yang menjawab “Tidak”. Berdasarkan analisa dilakukan mendapatkan hasil yang dapat dilihat pada Tabel 5.

Tabel 5. Hasil Perhitungan

No	Kriteria	Respon	
		Ya (%)	Tidak (%)
1	Learnability	100	0
2	Memorability	77.74	22.21
3	Efficiency	74.99	25.01
4	Error	83.31	16.66
5	Satisfaction	100	0
Total Keseluruhan		87.20	12.8

Seperti pada tabel 5 hasil perhitungan survei mendapatkan hasil dengan jumlah respon yang mengatakan “Ya” sebanyak 87.20%, sedangkan “Tidak” sebanyak 12.8%. Agar dapat disimpulkan aplikasi layak maka diperlukan pengelompokan kuantitatif yang mengidentifikasi apakah aplikasi sudah layak. Tabel Kuantitatif dapat dilihat pada Tabel 6.

Tabel 6. Tabel Kuantitatif

Skor	Kualifikasi	Hasil
85 – 100 %	Sangat Baik (SB)	Berhasil
65 – 84 %	Baik (B)	Berhasil
55 – 64 %	Cukup (C)	Tidak Berhasil
0 – 54 %	Kurang (K)	Tidak Berhasil

Dari perhitungan persentase jawaban responden pada *usability testing* untuk mengukur penggunaan pada Penerapan Algoritma *Jaro-Winkler* Untuk *Autocorrect* dan *Spelling Suggestion* Pada Aplikasi *Speech Recognition* CMS Berbasis Website ini dapat disimpulkan bahwa sebanyak 87.20 % pertanyaan mampu dilakukan oleh pengguna (menjawab “Ya”). 12,8 % pertanyaan tidak dapat dilakukan oleh pengguna (jawaban “Tidak”). Dengan total keseluruhan yang didapat yaitu 87.20%, berdasarkan tabel kuantitatif pada Tabel 4.5 maka dapat ditarik kesimpulan bahwa aplikasi “Penerapan Algoritma *Jaro-Winkler* Untuk *Autocorrect* dan *Spelling Suggestion* Pada Aplikasi *Speech Recognition* CMS Berbasis Website” mendapatkan hasil “Berhasil”.

## KESIMPULAN

Berdasarkan dari hasil implementasi dan testing Penerapan Algoritma *Jaro-Winkler* Untuk *Autocorrect* dan *Spelling Suggestion* Pada Aplikasi *Speech Recognition* CMS Berbasis Website, maka dapat dibuat kesimpulan sebagai berikut :

1. Penerapan Algoritma *Jaro-Winkler* pada *Speech Recognition* ini akan mempercepat pada saat memasukan data dengan jumlah yang banyak karena tidak perlu mengetik.
2. Penerapan *Speech Recognition* pada *company profile* dapat mempercepat proses perubahan konten tanpa perlu kembali ke halaman *dashboard*.
3. Berdasarkan hasil testing dan survei yang dilakukan, respon yang mengatakan “Ya” sebanyak 87.20%, sedangkan “Tidak” sebanyak 12.8%, dengan total keseluruhan yang didapat yaitu 87.20% maka aplikasi dapat dinyatakan berhasil atau layak dipakai.

## SARAN

Dalam Penerapan Algoritma *Jaro-Winkler* Untuk *Autocorrect* dan *Spelling Suggestion* Pada Aplikasi *Speech Recognition* CMS Berbasis Website ini tentunya masih jauh dari kata sempurna dan memiliki beberapa kekurangan.

Oleh sebab itu, untuk pengembangan aplikasi ini agar dapat menjadi lebih baik kedepannya, maka diberikan saran sebagai berikut : Penerapan Algoritma *Jaro-Winkler* pada *Speech Recognition* ini akan mempercepat pada saat memasukan data dengan jumlah yang banyak karena tidak perlu mengetik.

1. Penerapan *Speech Recognition* pada *company profile* dapat mempercepat proses perubahan konten tanpa perlu kembali ke halaman *dashboard*.
2. Berdasarkan hasil testing dan survei yang dilakukan, aplikasi dapat dinyatakan berhasil atau layak dipakai

## RUJUKAN

- Bassil, Y. (2017). *A Simulation Model for the Waterfall Software Development Life Cycle. International Journal of Engineering & Technology (iJET)*, Vol 2, No 5.
- Gueddah, H. Y. (2015). The Filtered Combination of The Weighted Edit. Institute of Electrical and Electronics Engineers.
- Hasugian, J. (2016). Penelusuran Informasi Ilmiah Secara Online: Perlakuan Terhadap Seorang Pencari Informasi Sebagai *Real User* . *Jurnal Studi Perpustakaan dan Informasi* .
- James Pustejovsky, A. S. (2012). *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media, Inc.
- Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. United States: Massachusetts.
- Krug, S. (2020). *Don't Make Me Think! A Common Sense Approach to Web Usability*. California.
- Raharjo, B. (2018). *Belajar Otodidak Framework Codeigniter Edisi Revisi*. Bandung : Informatika.
- Yulianingsih. (2017). Implementasi Algoritma *Jaro-Winkler* dan *Levenstein* . *Jurnal String*.