

## PERANCANGAN SISTEM CERDAS CHATBOT ADMIN TOKO ONLINE DENGAN ALGORITMA LEVESHTEIN DISTANCE

### Mulyana Ibrahim

Program Studi S1 Teknik Informatika, Fakultas Teknologi Informasi, Universitas Hasyim Asy'ari  
Email: [mulyanaibrahim00@gmail.com](mailto:mulyanaibrahim00@gmail.com)

### Achmad Imam Agung

Fakultas Teknologi Informasi, Universitas Hasyim Asy'ari  
Email: [imamagung@unhasy.ac.id](mailto:imamagung@unhasy.ac.id)

### Hadi Sucipto

Program Studi D3 Sistem Informasi, Fakultas Teknologi Informasi, Universitas Hasyim Asy'ari  
Email: [hadisucipto@unhasy.ac.id](mailto:hadisucipto@unhasy.ac.id)

### Abstrak

Perkembangan teknologi yang semakin pesat membuat banyak perubahan dalam berbagai aspek kehidupan manusia. Salah satu aspek perubahan ini yaitu bagaimana proses jual beli terjadi. Proses jual beli kini mulai menjadi konsep jual beli online dimana penjual menjual barangnya di internet dan konsumen membeli barangnya di internet. Fenomena ini lah yang melahirkan konsep awal dari penelitian ini, dimana fenomena ini melahirkan sebuah masalah baru yaitu kebutuhan akan fungsi admin dalam melayani pelanggan. Dari masalah tersebut penelitian ini dilakukan untuk menciptakan sebuah rekayasa admin toko *online* dengan menggunakan teknologi *chatbot*. Metode yang digunakan dalam penelitian ini adalah pencocokan kata (*String Matching*) dengan menggunakan algoritma *Levenshtein Distance*. Proses dari pencocokan kata ini kemudian nanti akan dimanipulasi dengan rekayasa *Natural Language Processing (NLP)*, Sehingga pelanggan yang berkomunikasi dengan sistem ini akan merasa berkomunikasi dengan manusia. Hasil dari penelitian ini yaitu sebuah sistem yang dapat melakukan percakapan melalui chat dengan menggunakan bahasa yang lebih natural atau bahasa manusia sehari-hari. Dari hasil pengujian test sistem, menunjukkan bahwa sistem mampu menangani beberapa pengguna dalam waktu yang bersamaan, selain itu dari hasil pengujian juga didapatkan bahwa sistem mampu mengenali dengan baik setiap kata yang dimasukkan ke dalam sistem.

**Kata kunci:** *Chatbot, Levenshtein Distane(LD), Natural Language Processing(NLP)*.

### Abstract

The progress of technology has become faster every year, this progress has impact so many aspect of humans life. One of the aspect that change so much in this days is online shopping where the seller sell their product on the internet and the customer buy it from the internet. The phenomenon make the need of customer service become more crucial than before. Because of this phenomenon this research try to make the system that can change customer service with chatbot. The method that used in this research is a string matching with *Levenshtein Distance* algorithm. After the string matching has a result, the result that will be manipulate with *natural language processing* so the customer who use the system will feel talk with human rather than with the machine. The result of this research is a system that can make a conversation with human language. From the test indicating the system can handle a couples user in the same time, and from the test give a result that system can understand the word that being inputed.

**Keyword:** *Chatbot, Levenshtein Distane(LD), Natural Language Processing(NLP)*.

## PENDAHULUAN

Perkembangan bisnis online yang semakin meningkat dan tidak lagi terikat oleh ruang dan waktu, membuat banyak individu yang semakin banyak melakukan aktivitas belanja online. Hampir setiap aktivitas belanja online di setiap situs yang ada saat ini menggunakan cara pembayaran dengan kartu kredit. Namun sulitnya proses pembuatan kartu kredit di bank melahirkan sebuah produk yang bernama *Virtual Credit Card (VCC)*. Banyak permintaan VCC membuat banyak beberapa pelaku bisnis online menjual produk VCC ini dengan beragam harga salah satunya yaitu *vccmurah.net*. Pemilik *vccmurah.net* adalah pelaku bisnis online independen atau mandiri dengan proses bisnis yang tidak seratus persen otomatis mengharuskan admin harus selalu siap dalam melayani berbagai pertanyaan dari pelanggan. Kebutuhan akan admin yang harus selalu siaga dalam merespon pelangganpun menjadi sangat dibutuhkan website *vccmurah.net*.

Komunikasi yang digunakan admin vccmurah.net yang menjual VCC ini menggunakan banyak cara. Cara-cara tersebut yaitu dengan telepon, email, dan chat. Cara-cara manual seperti ini tentunya memiliki banyak kendala dimana telepon harus memiliki banyak admin yang menangani pelanggan yang menelepon pada pihak pengusaha, email memiliki waktu yang cukup lama untuk dibalas dan chat manual yang butuh waktu untuk membalas setiap pertanyaan ataupun keluhan pelanggan.

Thomas N T (2016: 2740-2742) menyatakan Otomatisasi online chat bisa menjadi salah satu solusi untuk masalah ini. Chat robot (*chatbot*) diharapkan dapat menangani segala macam permintaan dan keluhan dari pelanggan.

Sistem *chatbot* sendiri yaitu sebuah perangkat lunak yang dapat melakukan interaksi dengan manusia menggunakan bahasa alami (Divya Madhu dkk., 2017). Banyak penelitian sudah dilakukan dalam pengembangan *chatbot*, seperti pengembangan mesin percakapan dan robot chat. Tujuannya sendiri dari *chatbot* ini yaitu adalah mensimulasikan percakapan antara manusia dan komputer dengan menggunakan bahasa alami manusia.

Penelitian yang dilakukan Setiaji dkk.,(2016:72-77), membahas pengembangan *chatbot* dengan menggunakan basis pengetahuan di database. Hasil sistem *chatbot* yang dihasilkan oleh mereka yaitu sebuah *chatbot* yang mampu membalas percakapan dasar. Pengembangan yang dilakukan oleh mereka menggunakan bahasa pemrograman khusus pembuatan kecerdasan buatan *chatbot* yaitu *Artificial Intelegent Markpup Language (A.I.M.L)*. Basisdata pada sistem ini yaitu sebagai otak *chatbot* dimana template kalimat disimpan di dalamnya.

Dalam penelitian ini penulis mencoba untuk merancang *chatbot* yang mampu untuk menjawab pertanyaan yang berhubungan dengan produk vcc. Rancangan yang akan digunakan adalah dengan membuat sebuah basis pengetahuan yang memiliki pengetahuan kata dasar dan kata yang akan dipelajari oleh sistem cerdas ini, sehingga apabila ada *input* kata yang tidak sesuai dengan kata dasar maka kata akan diperbaiki dan disimpan dalam sistem dan juga membuat sebuah template kalimat dan jawaban yang akan menjawab pertanyaan yang masuk ke dalam sistem.

## METODE

Sistem Cerdas Chatbot ini akan dibangun untuk kebutuhan khusus yaitu sistem ini akan berperan sebagai pengganti admin toko online yang melayani proses tanya jawab antara pelanggan dengan admin toko online. Adapun metode perancangan sistem ini, yaitu sebagai berikut :

### 1. Pengumpulan Data

Peneliti dalam masalah ini ini menggunakan beberapa metode pengumpulan data. Pengumpulan data yang dilakukan yaitu pertama wawancara dengan pemilik usaha berbasis web sebagai objek penelitian. Selain menggunakan metode wawancara peneliti juga menggunakan teknik observasi dengan mengamati secara langsung berbagai macam model pertanyaan yang diajukan oleh pelanggan melalui kolom komentar yang ada pada *posting* website.

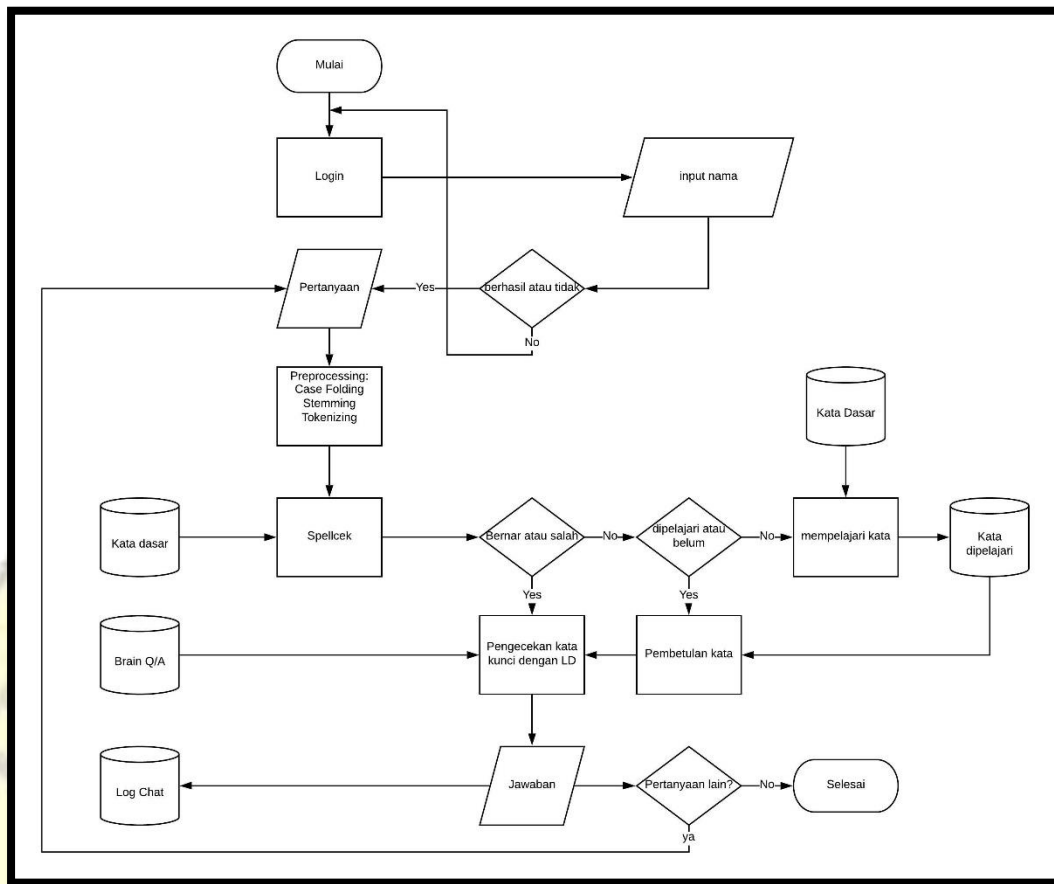
### 2. Analisis Data

Data yang digunakan dalam perancangan sistem cerdas chatbot ini yaitu sebagai berikut:

- a. Data Produk
- b. Data Pertanyaan
- c. Data Pesanan
- d. Data Bukti Pembayaran
- e. Data VCC

### 3. Flowchart Sistem

Sistem Cerdas Chatbot ini memiliki beberapa tahapan dalam menyelesaikan setiap proses percakapan yang dilakukan. Proses pertama adalah pengguna sistem melakukan *login* ke dalam sistem dengan memasukkan nama pengguna. Setelah pengguna berhasil masuk ke dalam sistem kemudian pengguna memasukkan pertanyaan yang ingin diajukan kepada sistem terkait dengan produk vcc. Pertanyaan yang masuk kemudian akan dilakukan proses *preprocessing* yaitu proses dimana kata yang tidak diperlukan akan dibuang. Setelah proses tersebut selesai maka berikutnya sistem akan melakukan pengecekan ejaan kata. Jika semua proses tersebut telah selesai maka berikutnya sistem akan mulai menyusun kata kunci dari pertanyaan yang masuk dan memulai pencarian jawaban yang sesuai dengan kata kunci yang ada. Alur sistem ini ditampilkan pada gambar 1 di bawah ini.



Gambar 1. Flowchart Sistem Cerdas *Chatbot* Admin Toko Online

#### 4. Algoritma *Levenshtein Distance*

Algoritma *Levenshtein Distance* (LD) merupakan salah satu dari algoritma *string matching*. Algoritma ini diciptakan oleh Vladimir Levenshtein pada tahun 1965 (Rahmat Ramadhan dkk, 2016). Algoritma ini bekerja dengan mencari perbedaan yang terdapat dalam dua string yang sama atau hampir sama yang kemudian mentransformasi string tersebut dengan menggunakan matriks string. Dalam algoritma ini dilakukan perbandingan antara string sumber(s) dan string target (t). Jarak (*Distance*) adalah nilai minimum perbedaan yang akan dihitung dari 3 operasi yang akan dilakukan dalam algoritma ini yaitu:

- 1) Operasi Penyisipan Karakter (*Insertion*)  
Operasi penyisipan karakter yaitu operasi yang melakukan penyisipan karakter pada sebuah string. Contohnya string 'kerta' menjadi 'kertas', dilakukan penyisipan karakter 's' pada akhir string. Penyisipan karakter ini tidak terbatas oleh tempat, sehingga penyisipan dapat dilakukan di awal, di tengah, maupun di akhir.
- 2) Operasi Penukaran Karakter (*Substitution*)  
Operasi penukaran yaitu operasi yang menukarkan salah satu karakter dengan karakter lain. Contohnya 'puku' menjadi 'buku'. Dalam contoh ini dilakukan penukaran karakter 'p' menjadi karakter 'b'.
- 3) Operasi Penghapusan Karakter (*Deletion*)  
Operasi penghapusan yaitu operasi yang melakukan penghapusan salah satu karakter dari sebuah string. Contohnya 'kuang' menjadi 'uang', dilakukan penghapusan pada karakter 'k' pada string.

Langkah-langkah dari algoritma *Levenshtein distance* ini yaitu:

Misalkan S= String Awal, dan T= String Target

Langkah 1 : Inisialisasi

- 1) Hitung panjang S dan T, misalkan m dan n
- 2) Buat matriks berukuran 0...m baris dan 0...n kolom
- 3) Inisialisasi baris pertama dengan 0...n
- 4) Inisialisasi kolom pertama dengan 0...m

Langkah 2 : Proses

- 1) Periksa  $S[i]$  untuk  $1 < i < n$
- 2) Periksa  $T[j]$  untuk  $1 < j < m$
- 3) Jika  $S[i] = S[j]$ , maka entrinya adalah nilai yang terletak pada diagonal atas sebelah kiri, yaitu  $d[i, j] = d[i - 1, j - 1]$
- 4) Jika  $S[i] \neq T[j]$ , maka entrinya adalah  $d[i, j]$  minimum dari:
  - Nilai yang terletak tepat di atasnya ditambah satu, yaitu:  $d[i, j - 1] + 1$ .....(1)
  - Nilai yang terletak tepat dikirinya ditambah satu yaitu:  $d[i - 1, j] + 1$ .....(2)
  - Nilai yang terletak tepat pada diagonal atas sebelah kirinya ditambah satu yaitu:  $d[i - 1, j - 1] + 1$ .....(3)

Langkah 3 : Hasil entri matriks pada baris ke-i dan kolom ke j, yaitu  $d[i, j]$

Langkah 2 diulang hingga entri  $d[m, n]$  ditemukan.

Setelah nilai *distance* dari string ditemukan maka berikutnya dihitung bobot *similarity* dengan menggunakan persamaan:

$$Bobot\ Similarity = \left(1 - \frac{d[m,n]}{MAX(S,T)}\right) * 100\% \dots\dots\dots(4)$$

$d[m, n]$  adalah nilai *distance* yang didapatkan dari langkah-langkah sebelumnya yang terletak pada bari ke m dan kolom ke n, S adalah panjang string awal, T adalah panjang string target, dan  $Max(S, T)$  adalah panjang string terbesar antara string awal dan string target.

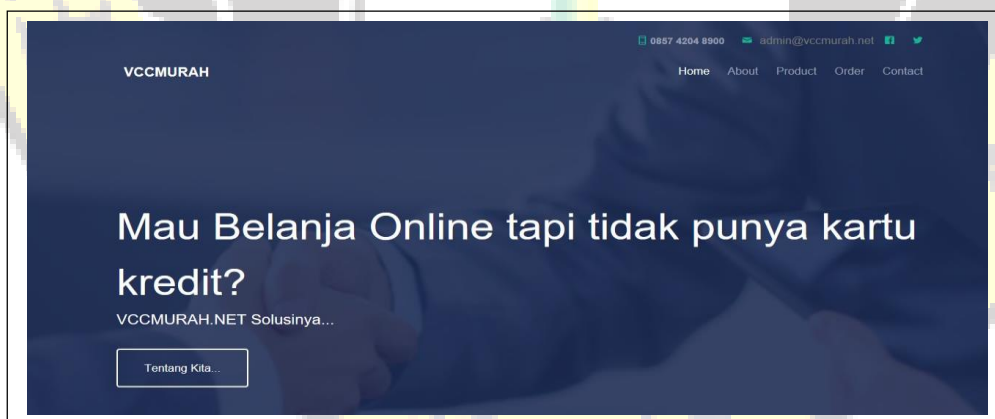
Bobot *similarity* diasumsikan dengan rentang 0 (nol) sampai 100 (seratus) persen, yang memiliki arti bahwa nilai 100 adalah nilai maksimum yang menunjukkan bahwa kedua string yang diproses memiliki kemiripan yang identik.

**HASIL DAN PEMBAHASAN**

Hasil dari perancangan sistem cerdas chatbot admin toko online ini yaitu sebagai berikut:

1. Tampilan *Home Website*

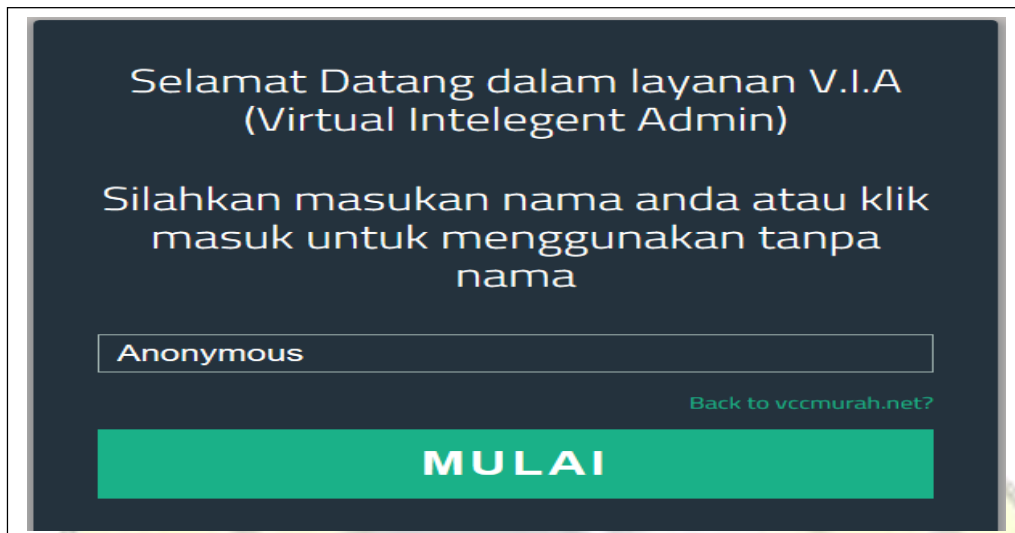
Halaman *Home* webiste ini adalah hakaman utama yang akan dilihat pengguna saat pertama kali masuk ke dalam website *vccmurah.net*. model dari tampilan website ini yaitu menggunakan model protfolio dimana pengguna dapat hampir seluruh informasi yang ada pada website ini pada halaman utamanya. Pada halaman ini pengguna memiliki akses untuk menuju hamam lain dengan menu yang ada, menu yang tersedia yaitu *Home, About, Product, Order, dan Contact* seperti yang ditampilkan dalam gambar 2.



Gambar 2. Tampilan halaman *Home* webiste

2. Tampilan Halaman *Login Chatbot*

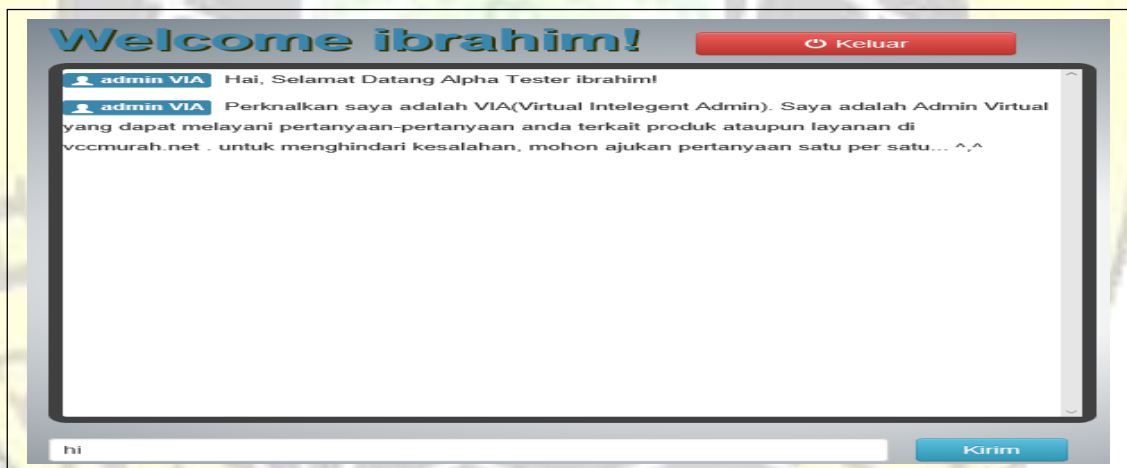
Halaman *Login Chatbot* adalah halaman dimana pengguna akan memasukan nama sebelum masuk ke dalam sistem utama yaitu halaman dinding percakapan yang ditampilkan pada gambar 3. Pada halaman ini pengguna dapat memulai percakapan dengan mengklik mulai atau dapat kembali ke halaman awal dengan mengklik link *Back*.



Gambar 3. Tampilan halaman peta pondok pesantren

### 3. Tampilan Halaman Dinding Percakapan

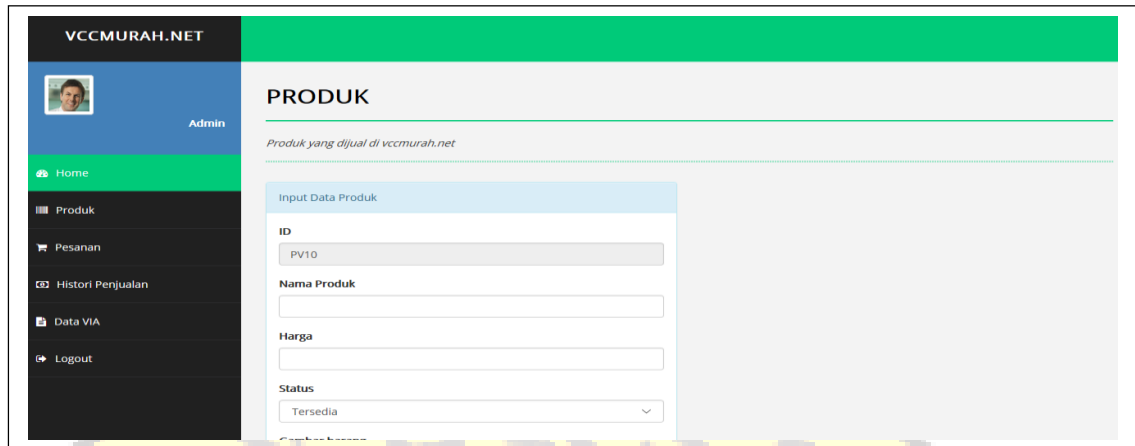
Tampilan Halaman dinding percakapan adalah tampilan utama dalam penelitian ini seperti yang ditampilkan pada gambar 4 di bawah. Dalam halaman ini pengguna dapat memulai percakapan dengan sistem yang kemudian akan dibalas secara otomatis oleh sistem cerdas chatbot dengan menggunakan teknik pencocokan kata dengan algoritma *levenshtein distance*. Pada halaman yang tampilan pada gambar 4 ini juga ditampilkan tombol merah yang berfungsi sebagai tombol keluar dari halaman dinding percakapan.



Gambar 4. Tampilan halaman dinding percakapan

### 4. Tampilan Halaman Admin

Halaman admin yang ditampilkan pada gambar 5 memiliki fungsi sebagai tempat utama admin melakukan setiap aktivitas utamanya seperti menerima pesanan, mengkonfirmasi pesanan, dan juga pada halaman ini admin dapat mengatur setiap kata kunci yang dimiliki oleh sistem cerdas ini, seperti menambah kata kunci, menambah pengetahuan kata pada sistem, dan juga menambah jawaban terhadap kata kunci yang ada pada menu data via.



Gambar 5. Tampilan halaman admin

5. Hasil Uji Coba

Uji coba yang penulis lakukan pada penelitian ini yaitu dengan memasukan 3 kalimat pertanyaan yang memiliki karakteristik berbeda. Kalimat pertama memiliki kemiripan seratus persen dengan kata aslinya, kalimat kedua memiliki bentuk kata yang tidak sempurna dan kalimat ketiga tidak memiliki kata kunci dalam basis pengetahuan. Uji coba pertama yaitu uji *case folding*. Hasil dari uji coba ini ditampilkan pada tabel 1 di bawah ini:

Tabel 1. Hasil *Case folding*

NO	Kalimat	Hasil
1	Bagaimana Cara Pesan VCC?	bagaimana cara pesan vcc?
2	mau Tanya Min, Bgmn cra Pemesanan Vccnya?	mau tanya min, bgmn cra pemesanan vccnya?
3	mau Tanya min...	mau tanya min...

Tabel 1 memperlihatkan 3 kalimat uji coba yang sudah dilakukan proses *case folding*. Setelah proses ini selesai maka berikutnya kalimat akan melalalui proses uji coba stemming yaitu proses pembuangan imbuhan dan tanda baca seperti yang ditampilkan pada tabel 2 dan setekah proses *stemming* kalimat akan melalui proses *tokenizing* yaitu memecah kalimat menjadi kata per kata seperti yang ditampilkan pada tabel 3.

Tabel 2. Hasil *Stemming*

NO	Kalimat	Hasil
1	bagaimana cara pesan vcc?	bagaimana cara pesan vcc
2	mau tanya min, bgmn cra pemesanan vccnya?	mau tanya min bgmn cra mesan vccnya
3	mau tanya min...	mau tanya min

Tabel 3. Hasil *Tokenizing*

NO	Kalimat	Hasil
1	bagaimana cara pesan vcc	array(4) { [0]=> string(9) "bagaimana" [1]=> string(4) "cara" [2]=> string(5) "pesan" [3]=> string(3) "vcc" }
2	mau tanya min bgmn cra mesan vccnya	array(7) { [0]=> string(3) "mau" [1]=> string(5) "tanya" [2]=> string(3) "min" [3]=> string(4) "bgmn" [4]=> string(3) "cra" [5]=> string(5) "mesan" [6]=> string(6) "vccnya" }
3	mau tanya min	array(3) { [0]=> string(3) "mau" [1]=> string(5) "tanya" [2]=> string(3) "min" }

Setelah semua proses *preprocessing* di atas selesai kata kemudian masuk proses pengecekan ejaan. Pada proses ini kata akan dicek kata per kata dengan kata yang ada pada basis pengetahuan. Hasil dari proses ini yaitu kata yang akan diperbaiki ejaannya jika ada kesalahan ejaan dan kata yang tidak ada pada basis pengetahuan akan dibuang seperti yang disajikan pada tabel 4. Kata yang sudah selesai proses ini kemudian akan menjadi kata kunci untuk mencari jawaban yang sesuai dengan kata kunci. Hasil dari kata kunci ini ditampilkan pada tabel 5. Kata kunci yang didapatkan ini kemudian dicari ke dalam basis pengetahuan yang menyimpan data kata kunci dan jawaban. Hasil dari pencarian ini ditampilkan pada tabel 6 di bawah ini.

Tabel 4. Hasil pencocokan kata

No	Kata Masuk	Kata Target	Hasil Cek	Koreksi
1	Bagaimana	Bagaimana	100%	-
2	Cara	Cara	100%	-
3	Pesan	Pesan	100%	-
4	vcc	vcc	100%	-
5	Bgmn	Bagaimana	44,4%	Bagaimana
6	Cra	Cara	75%	Cara
7	mesan	Pesan	80%	Pesan
8	vccnya	vcc	50%	vcc
9	mau	-	-	-
10	tanya	-	-	-
11	min	-	-	-

Tabel 5. Hasil pengecekan ejaan kata dan *filtering*

NO	Kalimat	Hasil
1	array(4) { [0]=> string(9) "bagaimana" [1]=> string(4) "cara" [2]=> string(5) "pesan" [3]=> string(3) "vcc" }	array(4) { [0]=> string(9) "bagaimana" [1]=> string(4) "cara" [2]=> string(5) "pesan" [3]=> string(3) "vcc" }
2	array(7) { [0]=> string(3) "mau" [1]=> string(5) "tanya" [2]=> string(3) "min" [3]=> string(4) "bgmn" [4]=> string(3) "cra" [5]=> string(5) "mesan" [6]=> string(6) "vccnya" }	array(4) { [0]=> string(9) "bagaimana" [1]=> string(4) "cara" [2]=> string(5) "pesan" [3]=> string(3) "vcc" }
3	array(3) { [0]=> string(3) "mau" [1]=> string(5) "tanya" [2]=> string(3) "min" }	array(0) { }

Tabel 6. Hasil Pencarian Jawaban

NO	Kalimat	Hasil
1	array(4) { [0]=> string(9) "bagaimana" [1]=> string(4) "cara" [2]=> string(5) "pesan" [3]=> string(3) "vcc" }	untuk melakukan pemesanan, silahkan pilih vcc yang mau dipesan, kemudian masukan data pesanan ke
2	array(4) { [0]=> string(9) "bagaimana" [1]=> string(4) "cara" [2]=> string(5) "pesan" [3]=> string(3) "vcc" }	untuk melakukan pemesanan, silahkan pilih vcc yang mau dipesan, kemudian masukan data pesanan ke
3	array(0) { }	Maaf pertanyaan yang anda ajukan tidak dapat saya pahami, silahkan masukan pertanyaan lainnya...

Hasil yang didapatkan pada tabel 6 membuktikan bahwa sistem berhasil mencari kata kunci yang sesuai dengan kata kunci yang didapatkan dari semua proses yang berlangsung dalam sistem. Dua kalimat pertama dan kedua memiliki arti makna yang sama dengan perbedaan ejaan sehingga di dapat hasil jawaban yang sama, sedangkan kalimat ketiga tidak

menghasilkan kata kunci sama sekali karena dalam basis pengetahuan tidak terdapat kata yang dicari sehingga hasil jawaban yang keluar adalah jawaban *default* apabila kata kunci tidak ditemukan.

## PENUTUP

### Simpulan

Berdasarkan hasil penelitian dan perancangan yang dilakukan peneliti, maka peneliti dapat menarik beberapa kesimpulan, yaitu:

1. Algoritma *Levenshtein Distance* dapat digunakan untuk merancang *chatbot* dengan melakukan pencocokan *string* dan juga dapat digunakan untuk melakukan pengkoreksian kata dari pertanyaan yang diajukan pengguna dengan tingkat akurasi mencapai 100% dengan minimal kecocokan kata sebesar 40% dari uji coba yang dilakukan dalam penelitian ini. Dari kesimpulan ini membuktikan bahwa asumsi awal yaitu algoritma *Levenshtein Distance* efektif digunakan untuk *chatbot* terbukti.
2. Sistem mampu menangani beberapa pengguna dalam waktu bersamaan dengan rata-rata waktu proses sebesar 4,1s dalam penggunaan input kalimat secara bersamaan. Dari hasil uji coba terhadap waktu penggunaan *chatbot* membuktikan bahwa asumsi ke dua pada awal penelitian ini terbukti, yaitu *chatbot* lebih efektif dari segi waktu maupun tenaga dalam melayani pelanggan.

### Saran

Penelitian lebih lanjut disarankan untuk meningkatkan dan menambahkan jumlah template jawaban yang lebih banyak dan bervariasi. Selain itu penggunaan dan pengolahan tabel kosa kata perlu untuk dibedakan berdasarkan jenis katanya sehingga perkoreksian kata menjadi lebih akurat.

## DAFTAR PUSTAKA

- Abdul-Kader, Sameera A., Woods, John., 2016. Survey on *Chatbot* Design Technique in Speech Conversation Systems. *International Journal of Advanced Computer Science and Applications*, Vol. 6, 72-80.
- Al-Khamaiseh, Koloud., Al-Shagarin, Shadi., 2014. A Survey of String Matching Algorithms. *Int. Journal of Engineering Research and Applications*, Vol. 4, 144-156.
- Ariyani, Na'firul Hasna. Sutardi. Ramadhan, Rahmat., 2016. Aplikasi Pendeteksi Kemiripan isi teks dokumen menggunakan metode *Levenshtein Distance*. *semantik*, Vol. 2, 279-286.
- Jaka H, Aris Tri. 2015. Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti Dalam Proses Text Mining. *Jurnal Informatika UPGRIS*. Vol-1. 1-9.
- Janani, R., Vijayarani, S., 2016. An Efficient Text Pattern Matching Algorithm for Retrieving Information from Desktop. *Indian Journal of Science and Technology*, Vol. 9, 1-10.
- Janani, R., Vijayarani, S., 2016. Pattern Matching Algorithms for Retrieving Information from Web Documents. *International Journal of Informative & Futuristic Research*, Vol. 4, 5343-5351.
- Madhu, Divya. J, Neeraj Jain C. Sebastian, Elmy. Shaji, Shinoy. Ajayakumar, Anandhu. 2017. A Novel Approach for Medical Assistance Using Trained *Chatbot*. *International Conference on Inventive Communication and Computational Technologies*. IEEE. 978-1-5090-5297-4/17.
- N T, Thomas. 2016. An E-business *Chatbot* using *AIML* and *LSA*. *ICACCI*, 2740-2742.
- Nugroho, Eko. 2011. *Perancangan sistem deteksi plagiarisme dokumen teks dengan algoritma Rabin-Karp*. Malang: Universitas Brawijaya.
- Pratama, B.P., dan Pamungkas, S.A. 2016. Analisis Kinerja Algoritma *Levenshtein Distance* Dalam Mendeteksi Kemiripan Dokumen Teks. *Jurnal Logika*. Jilid 6. 131-143.
- Setiaji, Bayu dan Wibowo, Ferry Wahyu. 2016. *Chatbot* Using A Knowledge in Database. *International Conference on Intelligent Systems, Modelling, and Simulation, IEEE*, 2166-0670/16, 72-77.