

# Rancang Bangun LINE Chatbot Informasi dan Edukasi Kesehatan Mental Menggunakan Algoritma Jaro–Winkler

Leilly Indahsari<sup>1</sup>, Kusnadi<sup>2</sup>, Tiara Eka Putri<sup>3</sup>

Fakultas Teknologi Informasi - Program Studi Teknik Informatika  
Universitas Catur Insan Cendekia  
Kota Cirebon, Indonesia

e-mail: <sup>1</sup>leillyindahsari@gmail.com, <sup>2</sup>kusnadi@cic.ac.id, <sup>3</sup>tiaaraekaputri@cic.ac.id

Diajukan: 25 Agustus 2020; Direvisi: 26 November 2020; Diterima: 10 Desember 2020

## Abstrak

Tingkat pemahaman masyarakat di Kota Cirebon mengenai mental yang sehat dan cara penanganannya masih rendah karena minimnya sosialisasi sehingga banyak mitos, stigma negatif, anggapan sepele, dan mispersepsi. Pengetahuan penanganan kesehatan mental merupakan faktor yang penting untuk dipahami karena menurut data kementerian kesehatan Indonesia masyarakat modern di perkotaan dan usia remaja lebih rentan mengalami risiko gangguan mental. Salah satu solusi untuk mengatasi masalah tersebut adalah dengan membangun chatbot sebagai media informasi edukasi pada platform LINE yang bertujuan untuk mensosialisasikan kesehatan mental pada masyarakat dan membantu masyarakat dalam mendapatkan informasi layanan konsultasi offline (tatap muka) yang berlokasi di Cirebon dan konsultasi online yang merupakan konsultasi dengan konselor secara online melalui WhatsApp, LINE, E-mail, atau website di luar Cirebon. Metode yang diterapkan pada chatbot adalah algoritma Jaro–Winkler sebagai pemrosesan string pencocokan kata antara keyword dengan input user. Metode yang digunakan untuk mengembangkan perangkat lunak adalah SDLC model Waterfall. Metode perancangan yang digunakan adalah analisis berorientasi objek menggunakan Unified Modeling Language (UML). Bahasa pemrograman yang digunakan adalah PHP dengan MySQL sebagai database yang perlu di-hosting. Hasil dari penelitian ini adalah sebuah chatbot informasi dan edukasi kesehatan mental yang dapat memberikan sosialisasi dengan mudah, cepat, terkini, dan praktis kepada masyarakat untuk meningkatkan pemahaman, kewaspadaan, dan kesadaran masyarakat terhadap hal yang berkaitan dengan kesehatan mental.

**Kata kunci:** Chatbot, Edukasi, Informasi, Kesehatan Mental, LINE.

## Abstract

The level of public understanding in Cirebon about being mentally healthy and how to deal with it is still low due to the lack of socialization so there are many myths, negative stigma, trivial assumptions, and misperceptions. In fact, knowledge of mental health management is an important factor to apply because according to data from the Indonesian Ministry of Health, adolescents and urban areas are more susceptible to mental disorders. One solution to overcoming the above problems is to build a chatbot using the LINE platform which aims to increase public understanding by providing education and proper mental health information. The method used is the Jaro–Winkler algorithm as a smart string between keywords and user input, with the SDLC Waterfall model as a software development method. The design method used is object-oriented analysis using Unified Modeling Language (UML). The programming language used is PHP with MySQL as a database that needs to be hosted. The result of this research is mental health information and education that can provide easy, fast, up-to-date and practical socialization to the public so as to eliminate stigma, increase awareness and understanding of mental care and treatment.

**Keywords:** Chatbot, Education, Information, Mental Health, LINE.

## 1. Pendahuluan

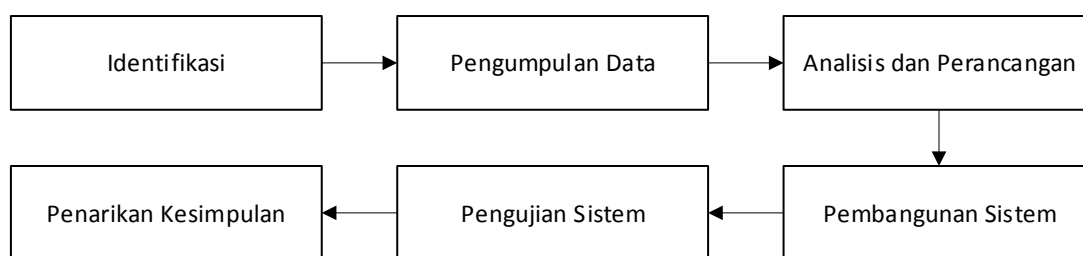
Data kombinasi Riset Kesehatan Dasar (Riskesdas) dan Pusat Data dan Informasi (Pusdatin) di tahun 2013 menunjukkan angka prevalensi gangguan jiwa berat di Indonesia 1,7 per 1000 (permil) yang artinya ada sekitar 1.7 kasus gangguan jiwa berat di antara 1000 orang penduduk Indonesia [1]. Tingkat pemahaman masyarakat di Kota Cirebon mengenai mental yang sehat dan bagaimana menanganinya masih

rendah karena minimnya sosialisasi. Selain itu, informasi mengenai lokasi tempat layanan konsultasi baik *online* maupun yang berada di Kota Cirebon juga masih minim. Masalah kesehatan mental tidak bisa diremehkan begitu saja karena jumlahnya penderitanya terus meningkat di kalangan remaja maupun dewasa, World Federation of Mental Health (WFMH) sebagai bagian dari WHO menyatakan bahwa masalah kasus kesehatan mental tidak lagi dilihat secara individual, namun harus diintervensi dalam skala makro atau sistem. Oleh karena itu, pengetahuan praktis mengenai kesehatan mental selayaknya juga dipahami oleh masyarakat [1]. Pengetahuan praktis tersebut bisa diolah dengan memanfaatkan *bot* atau *chatbot*. Berdasarkan hasil survei Shawar, Atwell, dan Roberts (dikutip dalam [2]) *chatbot* berhasil dikembangkan dan diimplementasikan dalam bidang edukasi, pencarian informasi, bisnis, dan *e-commerce*.

Beberapa penelitian implementasi algoritma Jaro–Winkler yang sudah pernah dilakukan, dijadikan rujukan pada penelitian ini. Penelitian yang dilakukan Yuninda [3] menghasilkan pengukuran kesamaan pada teks dokumen membutuhkan waktu 90.58989 detik, dengan rata-rata waktu 6.039 detik, dipengaruhi ukuran, *type*, dan kandungan dokumen. Penelitian yang dilakukan Ahmad [4] menguji plagiarisme 100 data abstrak jurnal skripsi berbahasa Indonesia dengan Jaro–Winkler menghasilkan nilai deteksi 30.58%. Penelitian yang dilakukan Mudawil [5] menunjukkan tingkat akurasi pengujian 85% atau 51 kata berhasil di-*stemming* dalam mengubah kata imbuhan. Penelitian yang dilakukan Dicky [6] menerapkan algoritma *sentence similarity* pada *chatbot* LINE untuk mengelompokkan kata lalu mencari kesamaannya sehingga *chatbot* memiliki keakuratan dalam merespons. Penelitian Herwin [7] menerapkan *cosine similarity* untuk menghitung kemiripan teks antara *input* dan *keyword* yang tersimpan untuk diterapkan pada *chatbot*, bila hasil melebihi 70% maka akan memberi respons sesuai *keyword*, jika tidak sesuai maka *user* diharuskan meng-*input* ulang kata. Berdasarkan uraian di atas, peneliti memilih menggunakan algoritma Jaro–Winkler untuk mengidentifikasi kecocokan antara kata atau kalimat dari *input user* dengan kata (*keyword*) yang tersimpan di *knowledge-base* kemudian hasilnya terintegrasi dengan *chatbot* berbasis LINE untuk menampilkan respons.

## 2. Metode Penelitian

Penelitian memiliki prosedur penelitian sebagai berikut:



Gambar 1. Prosedur penelitian.

### 2.1. Identifikasi Masalah

Tahapan penelitian untuk mengidentifikasi masalah yang terjadi mengenai bagaimana mensosialisasikan kesehatan mental di Kota Cirebon dan bagaimana hal tersebut dapat diwujudkan dengan membangun LINE *chatbot*.

### 2.2. Pengumpulan Data

Pengumpulan data dilakukan dengan mencari literatur berupa jurnal penelitian, karya ilmiah, *e-book*, jelajah artikel di internet, membuat kuesioner untuk mengetahui minat dan wawasan masyarakat tentang kesehatan mental dan mengetahui kepuasan pengguna terhadap *bot* kesehatan mental.

Kuesioner dibuat dengan Google Form dan disebarakan kepada 41 responden. Hasil survei minat dan wawasan masyarakat terkait kesehatan mental disajikan pada Tabel 1:

Tabel 1. Hasil kuesioner dengan total 41 responden.

No	Pertanyaan	Pilihan	Persentase
1.	Menurut Anda, apakah sosialisasi mengenai kesehatan mental itu penting?	Ya	85,4%
		Mungkin	14,6%
		Tidak	-
2.	Apakah Anda pernah mendapat sosialisasi kesehatan mental?	Ya	39%
		Tidak	61%
3.	Apakah Anda ingin mengetahui lebih dalam mengenai kesehatan mental?	Ya	95,1%
		Tidak	4,9%
4.	Menurut Anda, apakah sosialisasi tentang kesehatan mental saat ini sudah cukup terdistribusi?	Ya	9,8%
		Mungkin	43,9%
		Tidak	46,3%
5.	Jika ada aplikasi yang terintegrasi dengan <i>chatbot</i> LINE yang memungkinkan memberi informasi kesehatan mental, <i>screening</i> mental, kuis edukasi, daftar layanan konsultasi <i>online</i> , daftar puskesmas atau klinik, rumah sakit penyedia layanan konsultasi kesehatan mental dengan sumber khusus dari kota Cirebon, apakah Anda ingin menggunakannya?	Ya	68,3%
		Mungkin	31,7%
		Tidak	-

### 2.3. Analisis dan Perancangan Sistem

Terdiri dari beberapa bagian yaitu, analisis kebutuhan non-fungsional yang merupakan batasan pada sistem, analisis kebutuhan fungsional yang merupakan perancangan fungsi yang dapat dilakukan sistem dan analisis implementasi algoritma Jaro–Winkler. Sedangkan perancangan sistem menggunakan *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

#### 1. Analisis Fungsional

Berikut ini proses yang akan dibangun pada bagian *back-end* (web) *chatbot* informasi dan edukasi kesehatan mental untuk mengatur konten *bot*:

- a. Sistem mampu memproses *login*.
- b. Sistem mampu memproses *logout*.
- c. Sistem mampu menampilkan menu informasi dengan sub menu (menu) berupa *input*, *edit*, *delete* data menu informasi kesehatan mental.
- d. Sistem mampu mengelola sub menu informasi (menu) berupa *input*, *edit*, *delete* data menu informasi kesehatan mental.
- e. Sistem mampu mengelola sub menu informasi (jenis kesehatan mental) berupa *input*, *edit*, *delete* data jenis kesehatan mental.
- f. Sistem mampu mengelola sub menu informasi (fakta) berupa *input*, *edit*, *delete* data fakta informasi kesehatan mental.
- g. Sistem mampu mengelola sub menu informasi (mitos) berupa *input*, *edit*, *delete* data mitos informasi kesehatan mental.
- h. Sistem mampu mengelola sub menu informasi (*story*) berupa *delete* data *story* dari *user chatbot*.
- i. Sistem mampu mengelola sub menu informasi (saran) berupa *delete* data saran dari *user chatbot*.
- j. Sistem mampu mengelola menu konsultasi *online* berupa *input*, *edit*, *delete* data daftar layanan konsultasi *online*.
- k. Sistem mampu mengelola menu konsultasi daftar konsultasi di Cirebon berupa *input*, *edit*, *delete* data daftar layanan konsultasi yang berada di Kota Cirebon.
- l. Sistem mampu mengelola menu *screening* SRQ-20 [8] berupa *input*, *edit*, *delete* data soal dan jawaban *screening*.
- m. Sistem mampu mengelola menu *quiz* berupa *input*, *edit*, *delete* data soal dan jawaban *quiz* kesehatan mental.
- n. Sistem mampu mengelola menu *event* berupa *input*, *edit*, *delete* data *event* bertema seputar kesehatan mental.

Berikut ini proses yang akan dibangun pada bagian *front-end* LINE *chatbot* Informasi dan Edukasi Kesehatan Mental:

- a. Sistem *bot* mampu menampilkan isi konten yang dikelola pengelola informasi berupa: informasi kesehatan mental, konsultasi *online*, konsultasi Cirebon, *screening*, *quiz*, dan *event*
- b. Sistem *bot* mampu merespons masukan (*input*) pesan yang dikirim *user*.

#### 2. Analisis Non-Fungsional

- a. *Bot* yang dibangun hanya bisa dijalankan melalui aplikasi LINE.

- b. Bot yang dibangun hanya membahas seputar kesehatan mental.
- c. Bot yang dibangun menggunakan algoritma Jaro–Winkler.

3. Analisis Algoritma Jaro–Winkler

Algoritma ini diterapkan pada *file* bot.php untuk mengidentifikasi kalimat atau kata saat terjadi kesalahan pengejaan yang dilakukan *user* saat mengetik teks pada *chatbot*. Ditetapkan fungsi pembatas kesalahan ketik sebesar 0.85 dan minimal nilai *m* sebesar 2 [9]. Pada algoritma Jaro–Winkler terdapat rumus untuk menghitung jarak (*dj*) antara dua *string* yaitu *s1* dan *s2* yaitu:

$$dj = \frac{1}{3} \times \left( \frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \right) \tag{1}$$

Simbol *m* adalah jumlah karakter yang sama persis (dari *s1* dan *s2* yang di simpan dalam *server* dan di-*input*-kan *user*). *|s1|* adalah panjang *string* 1 (*string* atau *keyword* yang disimpan pada *server*). *|s2|* adalah panjang *string* 2 (*string* atau *keyword* yang di-*input*-kan *user* pada *chatbot*). Simbol *t* adalah jumlah transposisi (posisi karakter yang berbeda dari nilai *s1* dan *s2*, contoh: *schizophrenia* dan *scizhophrenia*)

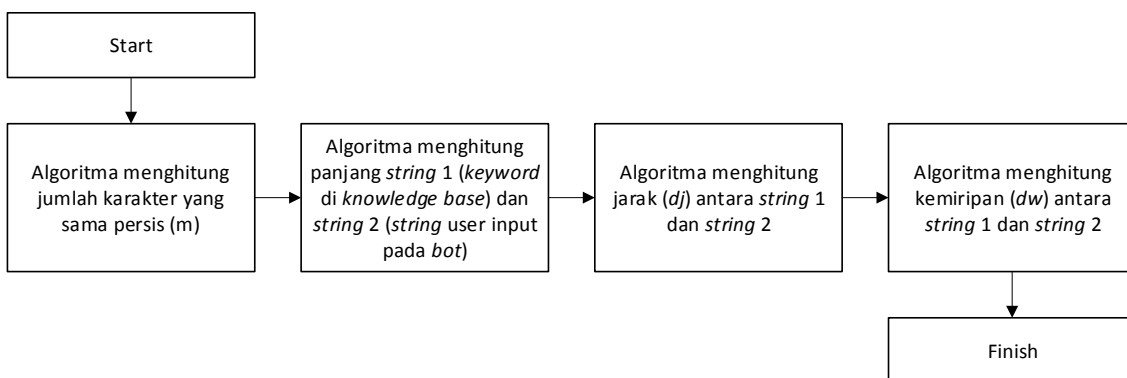
Keberadaan *s1* ini dijadikan acuan dalam mencari jumlah transposisi. Transposisi merupakan posisi kesamaan karakter *s1* dan *s2* yang tertukar saat dibandingkan. Sebagai contoh, jika *s1* adalah MANULA dan *s2* adalah MANUAL, keduanya memiliki karakter yang sama tetapi telah terjadi transposisi (urutan posisi *string* yang berbeda) antara A dan L, maka transposisi antara *s1* dan *s2* berjumlah 1. Dalam contoh lain, jika *s1* ANXIETY dan *s2* AMXIETY maka dalam perbandingan keduanya tidak terjadi transposisi, karena tidak ada pertukaran posisi dari karakter yang sama. Algoritma Jaro–Winkler menghasilkan nilai jarak maksimal sebesar 1, yang berarti kedua *string* yang dicocokkan bernilai 100% sama.

Jika *string* 1 (*s1*) dan *string* 2 (*s2*) diperbandingkan, maka Jaro–Winkler *distance* (*dw*) yaitu:

$$dw = dj + (lp(1 - dj)) \tag{2}$$

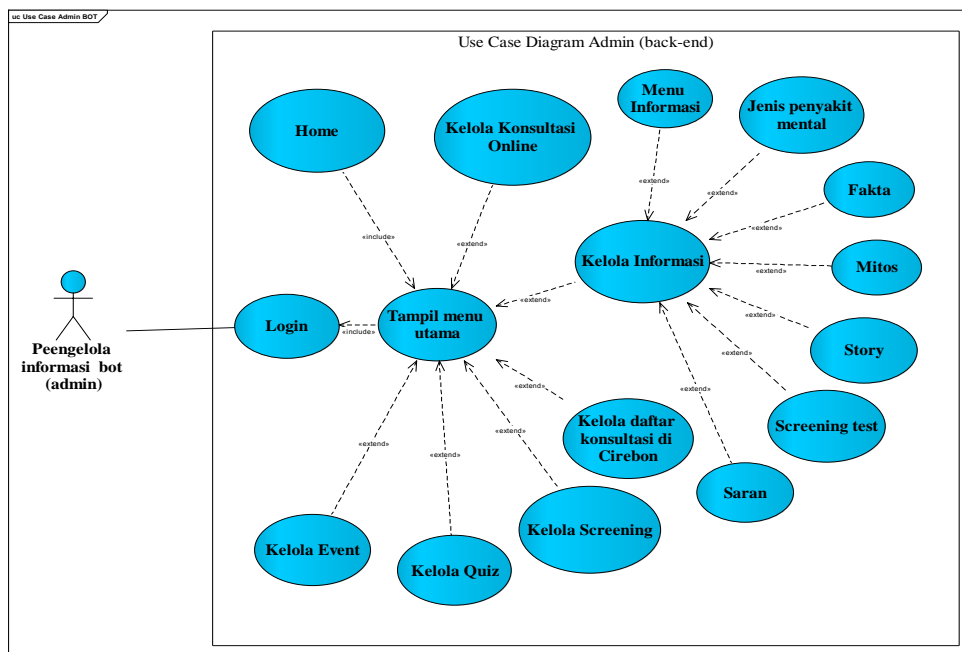
Simbol *dj* menunjukkan Jaro *distance* untuk *string* *s1* dan *s2*, *l* adalah panjang *prefiks* umum di awal *string*, yang nilai maksimalnya 4 karakter (panjang karakter yang sama sebelum ditemukan ketidaksesuaian *max* 4). Simbol *p* menunjukkan konstanta *scaling factor*. Nilai standar untuk konstanta ini menurut Winkler adalah *p* = 0,1.

Berikut langkah analisis dalam algoritma Jaro–Winkler:

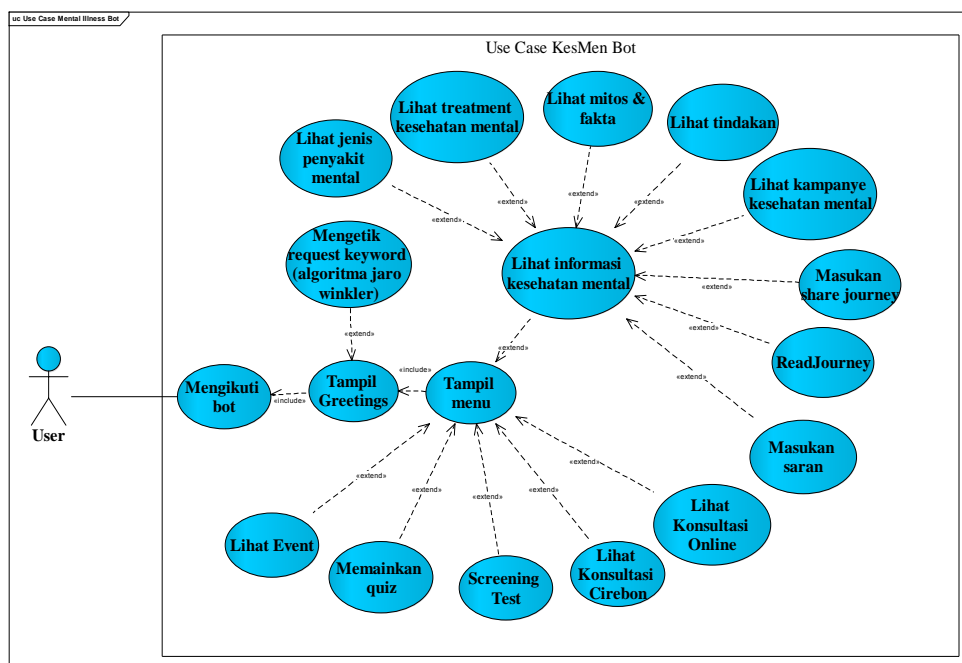


Gambar 2. Alur Analisis Algoritma Jaro–Winkler

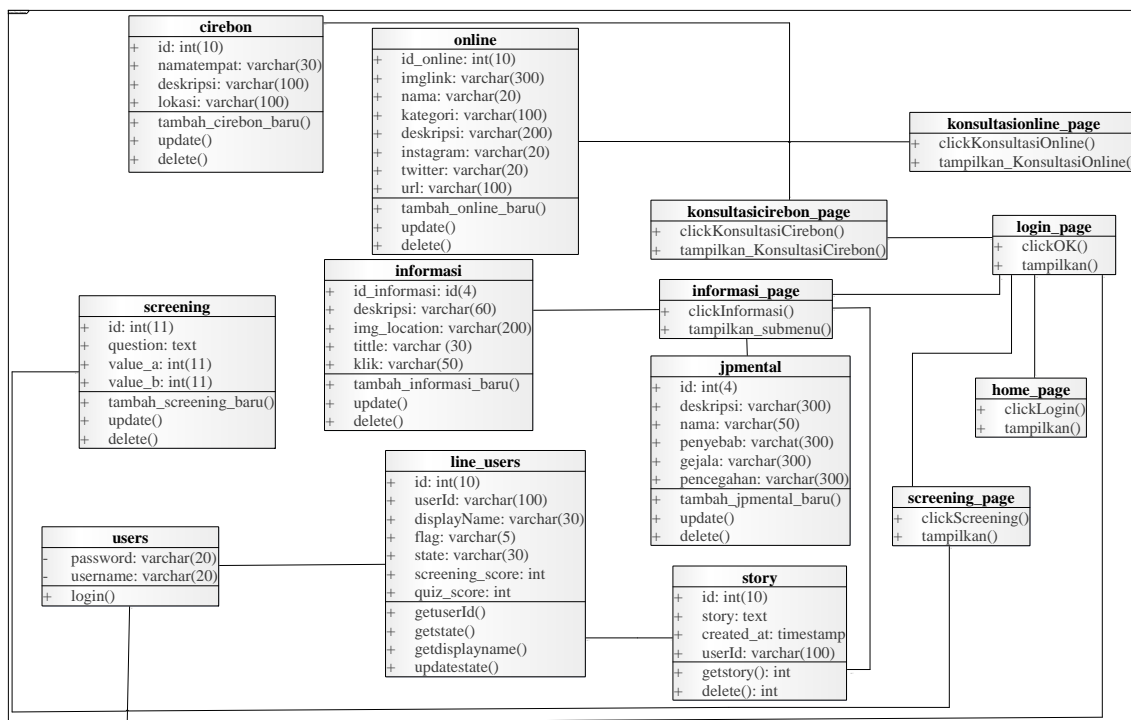
4. Perancangan Sistem



Gambar 3. Use case diagram pengelola informasi bot.



Gambar 4. Use case diagram user chatbot.



Gambar 5. Class diagram.

## 2.4. Pembangunan Sistem (Metode Pengembangan Perangkat Lunak)

Metode *Software Development Life Cycle* (SDLC) model *waterfall* digunakan sebagai proses pembuatan pola dalam mengembangkan sistem. Adapun tahapan dari model *waterfall* adalah sebagai berikut:

- Analisis kebutuhan perangkat lunak (*Software Requirement Analysis*), tahap membangun dan menentukan alur aplikasi LINE Chatbot. Keluaran dari tahap ini adalah dokumentasi kebutuhan.
- Desain (*System Design*), tahap membuat gambaran tahapan pengerjaan arsitektur sistem keseluruhan pembangunan aplikasi.
- Implementasi (*Implementation*), tahap melakukan pemrograman berdasarkan arsitektur yang dibuat di tahap sebelumnya hingga aplikasi dapat dijalankan dan digunakan.
- Pengujian (*Testing*), untuk mengukur keberhasilan sistem maka dilakukan pengujian menggunakan *blackbox testing* dan *user testing*.
- Pemeliharaan (*Maintenance*), tahap melakukan pemeliharaan atau pembaruan konten pada aplikasi yang telah selesai dibangun.

## 2.5. Pengujian Sistem

Tahap pengujian dilakukan dengan *blackbox testing* dan *user testing* guna menemukan, mengatasi, dan mengetahui apakah sistem yang telah dibangun telah sesuai dengan tujuan awal penelitian.

## 2.6. Penarikan Kesimpulan

Tahapan penelitian di mana peneliti menarik sebuah kesimpulan dari sistem yang berhasil dibangun.

## 3. Hasil dan Pembahasan

### 3.1. Implementasi Jaro–Winkler pada Sistem

Contoh 1. Perhitungan  $s1$  dan  $s2$   $dw > 0.85$  maka  $s1$  dan  $s2$  dinyatakan “Mirip”.

$s1$  ANXIETY,  $s2$  XIETY, maka  $m = 5, s1 = 7, s2 = 5, t = 0$ . Nilai Jaro distance adalah:

$$dj = \frac{1}{3} \times \left( \frac{5}{|7|} + \frac{5}{|5|} + \frac{5-0}{5} \right) = 0.905$$

Dari  $s1$  dan  $s2$  diketahui nilai  $l = 0$ , dan dengan nilai konstan  $p = 0.1$ . Maka nilai Jaro–Winkler

*distance* adalah:

$$dw = 0,905 + (0 \times 0.1(1 - 0,905)) = 0,905$$

Contoh cara kerja algoritma Jaro–Winkler yang diimplementasikan pada *chatbot* saat *user* mengetikkan pertanyaan pada *bot* (bukan akses melalui menu) digambarkan sebagai berikut:

1. Pengguna mengirim kata “Apa arti xiety?”
2. Algoritma menghitung jumlah karakter yang sama (*m*), jika *input user* lebih dari 1 kata seperti “apa”, “arti”, dan “xiety”, setiap kata akan diproses satu persatu untuk dicocokkan dengan *keyword* yang tersimpan yaitu “anxiety”.
3. Algoritma menghitung panjang *string* dari *s1* dan *s2*.
4. Algoritma menghitung transposisi yang terjadi.
5. Algoritma menghitung jarak (*dj*) antara *s1* dan *s2*.
6. Algoritma menghitung kemiripan (*dw*) *s1* dan *s2*, “apa” = 0, “arti” = 0, dan “xiety” = 0.905.
7. Hasil perhitungan *dw* antara *keyword* tersimpan “anxiety” dengan *input user* “apa” “arti” “xiety” menghasilkan nilai *dw* sebesar 0.914 dari kata “xiety”. Maka respons *bot* akan menampilkan informasi pengertian, pengobatan, gejala, dan penyebab anxiety karena nilai *dw* > 0.85.

Contoh 2. Perhitungan *s1* dan *s2*  $dw < 0.85$  maka *s1* dan *s2* dinyatakan “Tidak Mirip”. *s1* ANXIETY, *s2* ANSIETI, maka *m* = 5, *s1* = 7, *s2* = 7, *t* = 0. Nilai Jaro *distance* adalah:

$$dj = \frac{1}{3} \times \left( \frac{5}{|7|} + \frac{5}{|7|} + \frac{5 - 0}{5} \right) = 0.810$$

Dapat diketahui dari *s1* dan *s2* nilai *l* = 2, dan dengan nilai konstan *p* = 0.1. Maka nilai Jaro–Winkler *distance* adalah:

$$dw = 0,810 + (2 \times 0.1(1 - 0,810)) = 0,848$$

Contoh cara kerja algoritma Jaro–Winkler yang diimplementasikan pada *chatbot* saat *user* mengetikkan pertanyaan pada *bot* (bukan akses melalui menu) digambarkan sebagai berikut:

1. Pengguna mengirim kata “Apa arti ansieti?”
2. Algoritma menghitung jumlah karakter yang sama (*m*), jika *input user* lebih dari 1 kata maka akan dilakukan pemecahan per kata, seperti “Apa”, “Arti”, “Ansieti”, masing-masing kata akan diproses satu persatu untuk dicocokkan pada *keyword* yang tersimpan yaitu “anxiety”
3. Algoritma menghitung panjang *string* dari *s1* dan *s2*.
4. Algoritma menghitung transposisi yang terjadi.
5. Algoritma menghitung jarak (*dj*) antara *s1* dan *s2*.
6. Algoritma menghitung kemiripan (*dw*) antara *s1* dan *s2*, “Apa”=0, “Arti”=0, “Ansieti”=0,848.
7. Hasil perhitungan *dw* antara *keyword* tersimpan “anxiety” dengan *input user* “apa” “arti” “ansieti” menghasilkan nilai *dw* sebesar 0.848 dari kata “ansieti”. Maka respons *bot* yang muncul adalah “duh maaf aku belum paham maksud kakak nih : (” karena nilai *dw* < 0.85.

Berikut ini contoh lebih lanjut pencocokan *string* 1 dan *string* 2 dengan 0.85 sebagai batas maksimum kesalahan ketik:

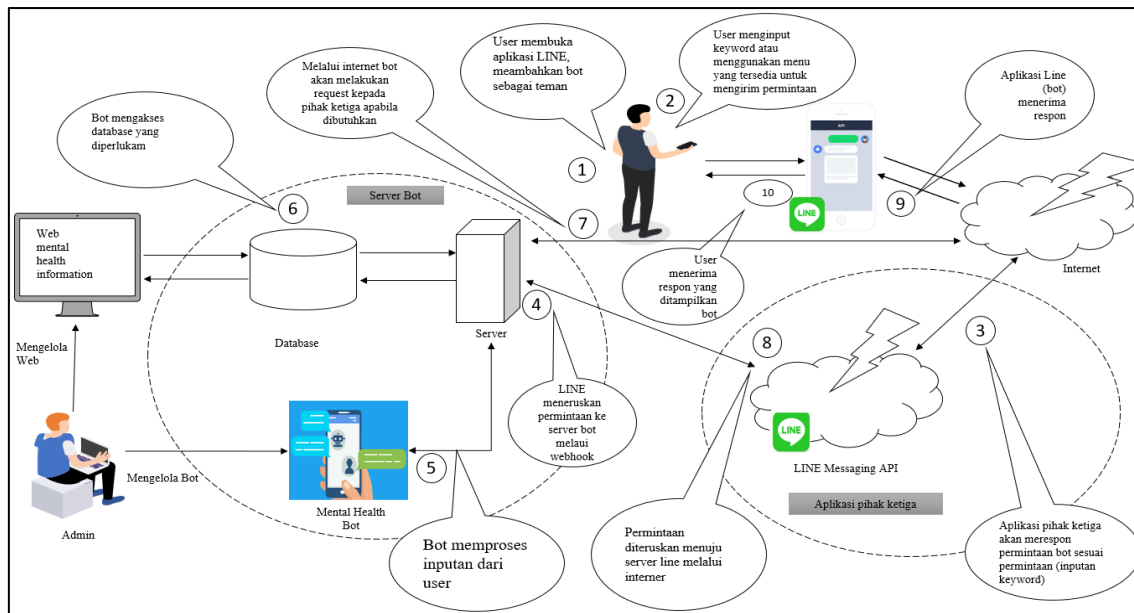
Tabel 2. Contoh perhitungan *s1* dan *s2*.

s1	s2	m	s1	s2	t	l	P	HASIL		KET	Hasil Bot
								dj	dw		
anxiety	xiety	5	7	5	0	0	0,1	0,905	0,905	Mirip	Muncul
	nanxiety	7	7	9	1	0	0,1	0,878	0,878	Mirip	Muncul
	ansieti	5	7	7	0	2	0,1	0,810	0,848	Tidak mirip	Tidak Muncul
	amsiety	5	7	7	0	1	0,1	0,810	0,829	Tidak mirip	Tidak Muncul
	amxiety	6	7	7	0	1	0,1	0,905	0,914	Mirip	Muncul

Jika *dw* antara *s1* dan *s2* lebih dari atau sama dengan 0.85 maka *chatbot* akan merespons jawaban pengertian, gejala, penyebab, dan pengobatan *anxiety*. Sedangkan jika *dw* antara *s1* dan *s2* kurang dari 0.85 maka itu menandakan bahwa *s1* dan *s2* tidak memiliki kemiripan namun *chatbot* tetap akan membalas

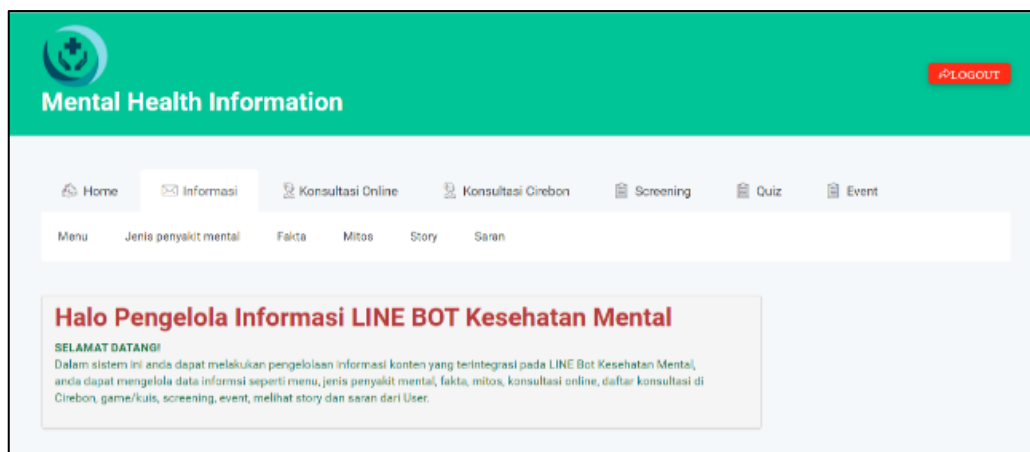
dengan respons “duh maaf aku belum paham maksud kakak nih : (“. Nilai 0.85 diambil sebagai nilai batas maksimum agar ketika terjadi kesalahan eja atau kesalahan ketik respons *bot* tetap optimal tidak berbeda jauh dengan apa yang *user* maksudkan.

### 3.2. Implementasi Sistem



Gambar 6. Deskripsi sistem.

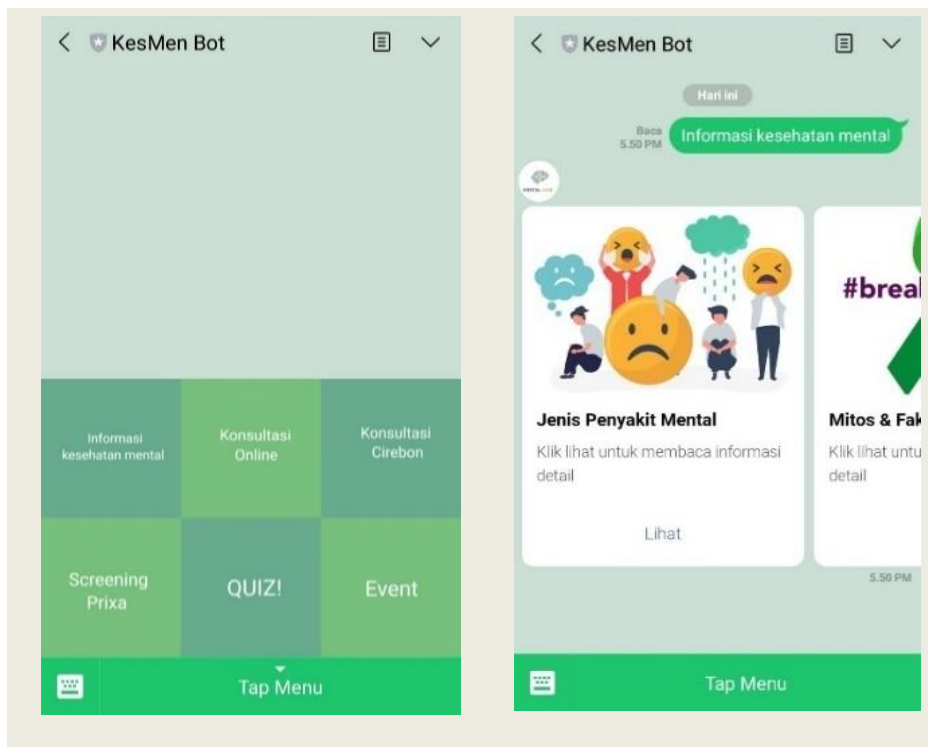
Gambar 6 menunjukkan bahwa *user* membuka aplikasi LINE dan menambahkan KesMen *bot* (id: @060aiznu) sebagai teman, *user* meng-input *keyword* atau menggunakan menu yang tersedia untuk mengirim *request*, aplikasi pihak ketiga akan merespons permintaan *bot* sesuai *request* (input dari *keyword*), LINE meneruskan permintaan ke server *bot* melalui *webhook*, *bot* memproses input dari *user*, *bot* mengakses *database* yang diperlukan, melalui internet *bot* akan melakukan *request* kepada pihak ketiga apabila dibutuhkan, permintaan diteruskan menuju server LINE melalui internet, aplikasi LINE *bot* menerima respons, *user* menerima respons.



Gambar 7. Halaman menu web *mental health information*.

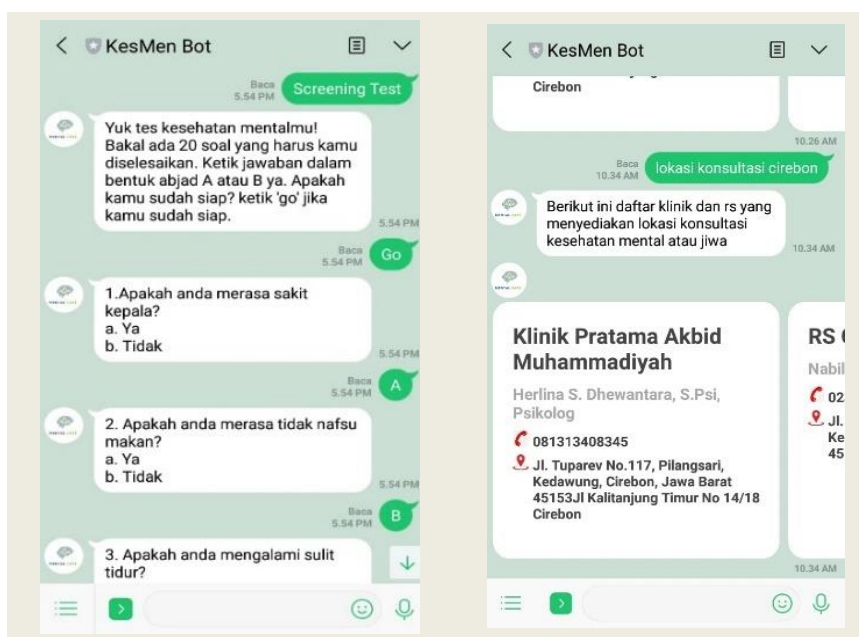
Gambar 7 menunjukkan halaman yang digunakan oleh pengelola informasi *bot* (*back-end*) untuk mengatur konten materi pada KesMen LINE *Bot*.





Gambar 8. Interface menu utama & carousel.

Pada Gambar 8, gambar sisi kiri menunjukkan menu utama yang tersedia pada bot, gambar pada sisi kanan terjadi ketika pengguna melakukan klik pada menu utama “informasi kesehatan mental” maka akan menampilkan carousel jenis penyakit mental, mitos fakta, treatment, tindakan, kampanye, share journey, read journey, dan saran. Selain akses melalui menu, gambar pada sisi kanan juga akan muncul ketika pengguna melakukan input kalimat yang mengandung kata “kesehatan mental” sebagai contoh “tampilin info kesehatan mental dong!”, hal itu terjadi karena bot telah diimplementasikan dengan algoritma Jaro–Winkler.



Gambar 9. Interface screening dan daftar konsultasi Cirebon.

Screening test terdiri dari 20 pertanyaan dengan model SRQ-20 untuk mendeteksi gejala awal *distress* atau gangguan emosional. daftar konsultasi terdiri dari 2 macam yaitu konsultasi *online* dan daftar konsultasi yang datanya berasal dari Kota Cirebon.

### 3.3. Blackbox Testing

Pengujian *blackbox* adalah metode pengujian yang memfokuskan pada fungsional perangkat lunak. Metode ini digunakan untuk menemukan kesalahan dalam kategori seperti fungsi yang tidak benar atau hilang, kesalahan antarmuka, kesalahan dalam struktur data, dan kesalahan kinerja.

Tabel 3. Perhitungan algoritma Jaro–Winkler dan *blackbox testing*.

Jenis Uji	Pertanyaan	Perhitungan Jaro–Winkler	Hasil yang Bot Diharapkan	Hasil Uji
Keyword 'Anxiety'	Mengetikkan 'apa arti anxiety?'	Apa = 0 Arti = 0 Anxiety = 1	Menampilkan menu pengertian, penyebab, gejala, pengobatan.	Sesuai
Keyword 'Anxiety'	Mengetikkan 'maksud xiety?'	Maksud = 0 Xiety = 0.905 (m=5, s1=7, s2=5, t=0, l=0, p=0,1. Maka dj=0,905, dw=0,905)	Menampilkan menu pengertian, penyebab, gejala, pengobatan.	Sesuai
Keyword 'Anxiety'	Mengetikkan 'nanxiety itu apa?'	Nanxiety=0,878 Itu = 0 Apa = 0	Menampilkan menu pengertian, penyebab, gejala, pengobatan.	Sesuai
Keyword 'Anxiety'	Mengetikkan 'apa maksud amxiety?'	Apa = 0 Maksud = 0 Amxiety = 0.914	Menampilkan menu pengertian, penyebab, gejala, pengobatan.	Sesuai
Keyword 'Anxiety'	Mengetikkan 'apa arti ansieti?'	Apa = 0 Arti = 0 Ansieti = 0.848	Muncul respon "Duh maaf aku belum paham maksud kakak nih:("	Sesuai
Keyword 'Anxiety'	Mengetikkan 'apa maksud amsiety?'	Apa = 0 Maksud = 0 Amsiety = 0.829	Muncul respon "Duh maaf aku belum paham maksud kakak nih:("	Sesuai

Dari hasil pengujian Tabel 3 disimpulkan bahwa "xiety", "nanxiety", dan "amxiety" dapat teridentifikasi sebagai bentuk kesalahan pengejaan dari *anxiety* di mana *dw* berada di atas nilai batas 0.85 akan menampilkan pengertian, penyebab gejala, dan pengobatan. Sedangkan "ansieti" dan "amsiety" tidak dapat teridentifikasi sebagai suatu kesalahan eja dari *anxiety* karena nilai *dw* berada di bawah nilai batas 0,85 akan menampilkan "duh maaf aku belum paham maksud kakak nih: (" Berikut ini hasil tampilan respons pada KesMen Bot.



Gambar 10. Uji keyword "nanxiety".



Gambar 11. Uji keyword "ansieti" & "amsiety".



Gambar 12. Uji keyword "xiety".



Gambar 13. Uji keyword "amxiety".

### 3.4. User Testing

Pengujian ini menggunakan metode kuesioner yang terdiri dari 5 pertanyaan diproses dengan Skala Likert. Kuesioner yang diberikan kepada responden adalah sebagai berikut:

Tabel 4. Daftar kuesioner kepuasan pengguna terhadap KesMen Bot.

No	Pertanyaan Kuesioner
1.	Saya merasa KesMen bot mudah untuk dipahami dan mudah digunakan.
2.	Saya merasa teredukasi dengan adanya KesMen bot ini.
3.	Saya merasa KesMen bot ini layak untuk digunakan.

Skala Likert mempunyai 5 level persetujuan, memiliki skor dimulai dari 1 sampai 5. Kriteria penafsiran responden menurut [10] sebagai berikut:

Tabel 5. Rentang Skala Likert.

Bobot	Rentang	Level
1.	1,00 – 1,80	Sangat Tidak Setuju
2.	1,81 – 2,61	Tidak Setuju
3.	2,61 – 3,41	Netral
4.	3,41 – 4,21	Setuju
5.	4,20 – 5,00	Sangat Setuju

Untuk mendapatkan penafsiran atau interpretasi akan menggunakan rumus Bakrie Siregar (1981: 20) (dalam [11]) yaitu analisis *Weight Means Score*, dengan rumus sebagai berikut:

$$M = \frac{\sum f(x)}{n} \tag{3}$$

Keterangan: M = Perolehan angka penafsiran, f = frekuensi, x = pembobotan skala nilai (skor), Σ = Penjumlahan, n = Jumlah responden. Perhitungan kuesioner Skala Likert dalam penelitian ini berjumlah 21 responden, berikut penjelasannya:

1. Tanggapan responden terhadap pertanyaan pertama, diketahui n = 21.

Tabel 6. Uji pertanyaan pertama.

No.	Alternatif Jawaban	f	x	f(x)	M = Σ f(x)/ n
1.	Sangat Setuju	9	5	45	93/21
2.	Setuju	12	4	48	
3.	Netral		3	0	
4.	Tidak Setuju		2	0	
5.	Sangat Tidak Setuju		1	0	
Jumlah		21		93	4.42

Berdasarkan hasil perhitungan pada Tabel 6, jumlah M = 4.42 berada pada rentang 4,20 – 5,00 yang termasuk pada level sangat setuju. Hasil tersebut menunjukkan bahwa 21 Responden rata-rata merasa sangat setuju KesMen Bot mudah digunakan dan dipahami.

2. Tanggapan responden terhadap pertanyaan kedua, diketahui n = 21.

Tabel 7. Uji pertanyaan kedua.

No.	Alternatif Jawaban	f	x	f(x)	M = Σ f(x)/ n
1	Sangat Setuju	7	5	35	90/21
2	Setuju	13	4	52	
3	Netral	1	3	3	
4	Tidak Setuju		2	0	
5	Sangat Tidak Setuju		1	0	
Jumlah		21		90	4.28

Berdasarkan hasil perhitungan pada Tabel 7, jumlah M = 4.28 berada pada rentang 4,20 – 5,00 yang termasuk pada level sangat setuju. Hasil tersebut menunjukkan bahwa rata rata 21 Responden merasa sangat setuju KesMen Bot dapat membuat penggunanya lebih teredukasi.

3. Tanggapan responden terhadap pertanyaan ketiga, diketahui n = 21.

Tabel 8. Uji pertanyaan ketiga.

No.	Alternatif Jawaban	f	x	f(x)	$M = \Sigma f(x) / n$
1	Sangat Setuju	11	5	55	94/21
2	Setuju	9	4	36	
3	Netral	1	3	3	
4	Tidak Setuju		2	0	
5	Sangat Tidak Setuju		1	0	
Jumlah		21		94	4.47

Berdasarkan hasil perhitungan pada Tabel 8, jumlah  $M = 4.47$  berada pada rentang 4,20 – 5,00 yang termasuk pada level sangat setuju. Hasil tersebut menunjukkan bahwa rata-rata 21 Responden merasa sangat setuju KesMen *Bot* layak digunakan.

#### 4. Kesimpulan

Dari hasil implementasi dan pengujian yang telah diuraikan, maka dapat disimpulkan beberapa hal yaitu, dengan adanya LINE *chatbot* kesehatan mental (KesMen *bot*) ini informasi dan edukasi seputar kesehatan mental dapat terpublikasi dan tersaji dengan baik berdasarkan *chatbot* yang berhasil digunakan masyarakat sehingga tujuan mensosialisasikan kesehatan mental dapat tercapai dan berdasarkan hasil survei *user testing chatbot* yang dilakukan pada 21 responden. Dengan demikian informasi dapat diakses oleh masyarakat umum untuk meningkatkan kesadaran kesehatan mental diri sendiri dan mensosialisasikan informasi yang ada tanpa terbatas jarak dan waktu. Adanya fitur *screening* telah memberikan kebaruan untuk mempermudah *screening* awal pemeriksaan kesehatan mental, selain itu KesMen *bot* ini dapat mempermudah menemukan informasi penyedia layanan konsultasi kesehatan mental *online* dan konsultasi kesehatan mental yang berada di Cirebon. Algoritma Jaro–Winkler menjadi solusi pemrosesan teks yang dapat diterapkan pada *chatbot* ini. *Chatbot* kesehatan mental ini dapat digunakan dengan menambahkan teman pada platform LINE dengan ID *bot @060aiznu*.

#### Daftar Pustaka

- [1] Y. Kurniawan and I. Sulistyarini, “Komunitas sehat (sehat jiwa dan hati) sebagai intervensi kesehatan mental berbasis masyarakat,” *Insa. J. Psikol. dan Kesehat. Ment.*, vol. 1, no. 2, p. 112, Jan. 2017, doi: 10.20473/jpkm.v1i22016.112-124.
- [2] D. Domarco and N. M. S. Iswari, “Rancang bangun aplikasi chatbot sebagai media pencarian informasi anime menggunakan regular expression pattern matching,” *J. Ultim.*, vol. 9, no. 1, pp. 19–24, 2017, doi: 10.31937/ti.v9i1.559.
- [3] Y. Faranika, N. Nikentari, and H. Kurniawan, “Sistem Pengukuran Dokumen Menggunakan Algoritma Jaro-Winkler Distance,” pp. 1–8.
- [4] A. Kornain, F. Yansen, and T. Tinaliah, “Penerapan Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Teks Berbahasa Indonesia,” *Progr. Stud. Tek. Inform. STMIK GI MDP*, pp. 1–10, 2014.
- [5] M. Qulub, E. Utami, and A. Sunyoto, “Stemming Kata Berimbuhan Tidak Baku Bahasa Indonesia Menggunakan Algoritma Jaro-Winkler Distance,” *Creat. Inf. Technol. J.*, vol. 5, no. 4, p. 254, 2020, doi: 10.24076/citec.2018v5i4.218.
- [6] D. A. Rizaldi, G. A. K. Rosyad, and A. D. Hartanto, “Implementasi Algoritma Sentence Similarity Terhadap Chatbot Seputar Amikom,” *METHOMIKA J. Manaj. Inform. dan Komputerisasi Akunt.*, vol. 4, no. 1, pp. 10–14, 2020.
- [7] H. H. Herwin, “Super Agent Chatbot ‘3S’ Sebagai Media Informasi Menggunakan Metoda Natural Language Processing(NLP),” *J. Teknol. Dan Open Source*, vol. 2, no. 1, pp. 53–64, 2019, doi: 10.36378/jtos.v2i1.144.
- [8] S. Idaiani and A. Y. Kristanto, “Analisis Gejala Gangguan Mental Emosional Penduduk Indonesia,” *Major. Kedokt. Indones.*, vol. 59, pp. 473–479, 2007.
- [9] C. Wibisono, “Pemanfaatan Approximate String Matching Menggunakan Jaro-Winkler Distance untuk Mengidentifikasi Typographical Error pada,” 2018.
- [10] Sugiono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta, 2013.
- [11] T. Helmi, R. A. Munjin, and I. Purnamasari, “Kualitas Pelayanan Publik dalam Pembuatan Izin Trayek oleh DLLAJ Kabupaten Bogor,” *J. Governansi*, vol. 2, no. April, pp. 47–59, 2016.