

Aplikasi Chatbot Informasi Kampus Polban Menggunakan Aplikasi *LINE Messenger*

Said Al-Fakhri¹, Haya Utami Lutfi², Wira Kusuma Wardana³,
Ghifari Munawar, S.Kom., M.T. ⁴, Bambang Wisnuadhi, S.Si., M.T.⁵

^{1,2,3}Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Bandung 40012

¹E-mail : said.al.tif16@polban.ac.id

²E-mail : haya.utami.tif16@polban.ac.id

³E-mail : wira.kusuma.tif16@polban.ac.id

⁴E-mail : ghifari.munawar@jtk.polban.ac.id

⁵E-mail : bwisnu@jtk.polban.ac.id

ABSTRAK

Pengembangan aplikasi dengan judul “Aplikasi Chatbot Informasi Kampus Polban Menggunakan Aplikasi *LINE Messenger*” untuk membuat sebuah aplikasi chatbot dalam platform *LINE* yang diharapkan dapat membantu menjawab pertanyaan yang diajukan oleh pengguna yang ingin mengetahui lebih lanjut mengenai informasi pada kampus Politeknik Negeri Bandung, sehingga informasi yang didapat pengguna merupakan informasi yang benar dan jelas. Aplikasi chatbot ini diharapkan dapat digunakan secara aktif sehingga pengguna mendapat tanggapan atau jawaban melalui chatbot ini. Metode yang digunakan dalam pembuatan chatbot ini adalah metode *Sentence Similarity Measurement* dan Bigram. Metode *Sentence Similarity Measurement* merupakan metode perhitungan persamaan suatu kalimat dengan pemecahan kata yang dicari kata terdekatnya menggunakan metode bigram. Pembuatan chatbot dengan bahasa pemrograman Java. Chatbot ini diimplementasikan dalam platform *LINE*, sehingga pembuatannya menggunakan *SDK Line* yang diimplementasikan dengan *Messaging LINE API*. *API* dari *LINE* ini yang bertugas untuk menghubungkan data dari pesan yang disampaikan oleh user dari server dengan platform *LINE*.

Kata Kunci

Chatbot, LINE, Informasi Kampus, Sentence Similarity Measurement

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat berdampak pada perilaku informasi. Kebutuhan informasi yang lebih mudah, murah dan mempunyai relevansi merupakan mutu dalam syarat informasi yang baik [1]. Hal ini disampaikan oleh Budi Sutedjo Dharma Oetomo. Sehingga para pemberi informasi dituntut untuk memiliki sebuah media agar informasi yang disajikan sesuai dengan mutu yang didapatkan oleh pengguna.

Media pemberi informasi yang banyak digunakan saat ini rata-rata bersifat statis dalam bentuk penyajian web dan tidak semua informasi biasanya dapat tertangkap oleh masyarakat umum, serta ada beberapa informasi yang bersifat khusus berdasarkan kepentingan tertentu [2]. Pemberi informasi dalam mengatasi hal tersebut menggunakan media tambahan seperti layanan tanya jawab

atau *contact me* yang ditangani oleh customer service. Layanan tanya jawab yang diangkat pada studi kasus tugas akhir ini yaitu informasi kampus pada Politeknik Negeri Bandung yang ditangani oleh Humas.

Layanan tanya jawab biasanya tidak langsung mendapatkan, dikarenakan harus menunggu customer service untuk menjawabnya dalam jangka waktu yang tidak ditentukan. Oleh karena itu, diperlukan media pengganti yang lebih mudah dan tersedia kapan saja yang diharapkan dapat bekerja 24 jam dan memberikan informasi yang cepat. Media pengganti ini adalah chatbot. Chatbot adalah layanan obrolan robot dengan kecerdasan buatan atau Artificial Intelligent yang menirukan percakapan manusia melalui pesan suara, obrolan teks atau keduanya. Chatbot memanfaatkan teknologi *Natural Language Processing* sebagai salah satu bidang teknologi ilmu Artificial Intelligence yang mempelajari komunikasi antara manusia dengan komputer

melalui bahasa alami yang biasa digunakan sehari-hari.

Pada dasarnya chatbot bekerja dengan cara melihat kata kunci dalam data yang masuk dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip. Sedangkan untuk melihat kata kunci dalam data yang masuk sangat beragam jenis struktur kata, ada yang baku dan ada yang tidak baku. Sehingga perlu penanganan ekstra dalam tahap pre-processing atau tahap persiapan. Diperlukan disiplin ilmu atau metode agar chatbot mengerti pola kata atau kata kunci yang paling cocok, salah satunya adalah Sentence Similarity Measurement atau SSM sebagai perhitungan tingkat kemiripan antara dua kalimat.

Saat ini, dari sekian banyak media sosial, pengguna *Line Messenger* salah satu sarana komunikasi yang sering dipakai oleh kalangan anak muda di era milenial ini. Sejak pertama kali hadir di Indonesia pada tahun 2013, tak kurang dari 90 juta netizen Indonesia merupakan pengguna layanan chatting asal Jepang tersebut. Dari angka tersebut, sebanyak 80 persen di antaranya (72 juta) merupakan pengguna aktif bulanan alias Monthly Active Users (MAU). Sementara itu, dari 72 juta pengguna aktif tersebut, 55 persennya merupakan pengguna wanita dan 45 persen merupakan pengguna pria. Kemudian dari segi usia, statistik menunjukkan pengguna LINE didominasi oleh usia 18 hingga 22 tahun dengan persentase sebesar 41 persen. Disusul usia 23-32 tahun dengan persentase sebesar 21 persen [3].

Berdasarkan kondisi tersebut, timbul gagasan untuk membuat sebuah aplikasi chatbot yang dapat digunakan sebagai media layanan tanya jawab yang aktif dalam menangani informasi terkait kampus Polban. Aplikasi chatbot tersebut berjalan pada media sosial yang banyak digunakan yaitu *LINE Messenger*. Dengan menggunakan teknologi NLP sehingga pengguna bisa menggunakan bahasa sehari-hari untuk berinteraksi dengan aplikasi chatbot dengan bantuan metode SSM aplikasi *chatbot* bisa menjawab pertanyaan dari pengguna yang lebih akurat.

1.2 Identifikasi Masalah

Identifikasi masalah untuk Chatbot Informasi Kampus adalah sebagai berikut:

1. Belum tersedianya media resmi yang interaktif berkomunikasi dengan pengguna diluar dan didalam jam kerja untuk bertanya seputar informasi kampus Polban, sehingga perlu dibuatkannya aplikasi chatbot sebagai sarana *customer* untuk berkomunikasi.
2. Informasi kampus Politeknik Negeri Bandung sewaktu-waktu dapat berubah atau dinamis, sehingga perlu dibuatnya aplikasi panel untuk mengelola informasi kampus Polban.
3. Belum tersedianya informasi tambahan yang tercantum pada media resmi, sehingga perlu dilakukan dengan fitur *broadcast* oleh aplikasi.

1.3 Metode yang Digunakan

Metode yang digunakan untuk mengembangkan aplikasi chatbot yaitu proses perhitungan pencocokan pola percakapan dan proses normalisasi.

1. Metode Normalisasi Pertanyaan

Proses normalisasi adalah proses untuk menormalkan input atau penyesuaian kalimat sebelum proses pencocokan pola. Proses ini di implementasikan dalam fungsi *normalizer()*, didalamnya terdapat dua sub proses yaitu:

- a. *Mark removal*, yaitu penghilangan tanda baca seperti tanda titik (.), tanda koma (,), tanda tanya(?), tanda seru (!), tanda titik dua (:), tanda titik koma (;), tanda hubung (-), tanda elipsis (...), tanda petik ("..."), tanda petik tunggal ('...'), tanda kurung ((...)), tanda kurung siku ([...]), dan tanda garis miring (/).

Tabel 1. Contoh Hasil Proses Mark Removal

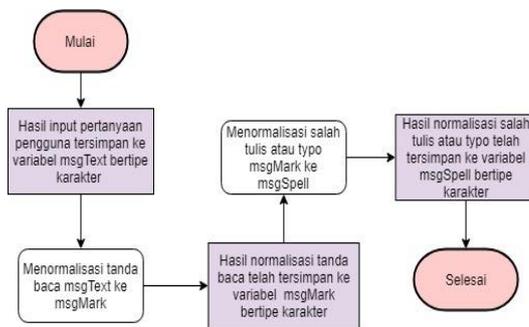
No	Input	Hasil
1.	Hai, siapa namamu?	Hai siapa namamu
2.	Stop!	Stop
3.	Merah; Kuning; Hijau	Merah Kuning Hijau
4.	Sebagai berikut :	Sebagai berikut

- b. *Spell Correction*, yaitu membenarkan kata atau ejaan yang salah tulis dan

menggantikan dengan kata yang benar.

Tabel 2. Contoh Hasil Proses *Spell Correction*

No	Input	Hasil
1.	Kpn PMDK Polban dibka	Kapan PMDK Polban dibuka
2.	Ada brp jrsan di Polban	Ada berapa jurusan di Polban
3.	Apa yg dimksd PMDK	Apa yang dimaksud PMDK
4.	jalur msk dfttr Polban	jalur msk daftar Polban



Gambar 1. Flowchart Proses Metode Normalisasi Pertanyaan

2. Metode Perhitungan Pencocokan Pola Percakapan

Proses perhitungan pencocokan pola percakapan atau *pattern-matching* sebagai fungsi utama yang digunakan untuk pengembangan aplikasi chatbot. Proses perhitungan skor tertinggi menggunakan *sentence-similarity measurement* atau SSM antara input dari pengguna dengan *pattern* yang tersimpan di dalam *database*. Rumus perhitungan skor SSM menggunakan fungsi *n-gram* akan dijelaskan dibawah ini. [4]

$$\frac{\text{count}(b1 \in b2) + \text{count}(b2 \in b1)}{\text{count}(b1) + \text{count}(b2)} \quad (1)$$

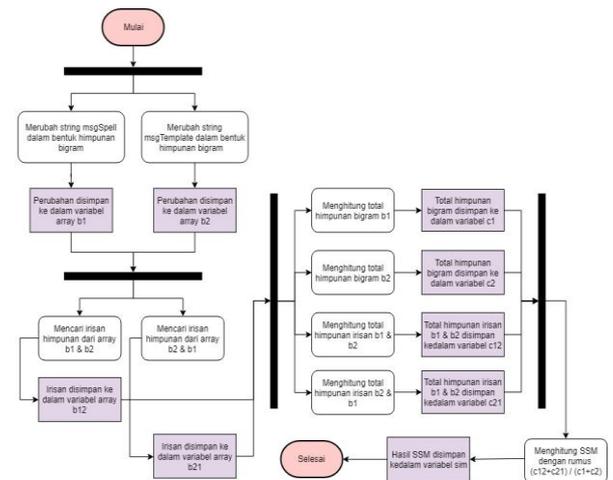
Berikut penjelasan dari rumus diatas sebagai berikut :

- $b1$ adalah himpunan bigram yang dihasilkan dari string pertama, dan $b2$ adalah himpunan bigram yang dihasilkan dari string kedua.
- $\text{count}(b1 \in b2)$ adalah sebagai operasi himpunan untuk jumlah dari "setiap bigram di $b1$ merupakan anggota

bigram di $b2$ ". Demikian sebaliknya untuk $\text{count}(b2 \in b1)$

- $\text{count}(b1)$ adalah jumlah bigram yang ada di string pertama. Demikian sebaliknya untuk $\text{count}(b2)$

Berikut merupakan alur dari perhitungan pencocokan pola dengan metode SSM, dimulai dari setelah pengguna menginputkan pertanyaan melalui aplikasi *LINE Messenger* yang sudah dilakukan proses normalisasi sesuai dari penjelasan sebelumnya.



Gambar 2. Flowchart Metode SSM

Berikut contoh implementasi dari perhitungan skor SSM menggunakan fungsi *n-gram* :

string 1 : "pengumuman pmdk polban dibuka tanggal berapa"

string 2 : "kapan tanggal pengumuman pmdk dibuka"

$b1 = \{ "pe", "en", "ng", "gu", "um", "mu", "um", "ma", "an", "n", "p", "pm", "md", "dk", "k", "p", "po", "ol", "lb", "ba", "an", "n", "d", "di", "ib", "bu", "uk", "ka", "a", "t", "ta", "an", "ng", "gg", "ga", "al", "l", "b", "be", "er", "ra", "ap", "pa" \}$

$b2 = \{ "ka", "ap", "pa", "an", "n", "t", "ta", "an", "ng", "gg", "ga", "al", "l", "p", "pe", "en", "ng", "gu", "um", "mu", "um", "ma", "an", "n", "p", "pm", "md", "dk", "k", "d", "di", "ib", "bu", "uk", "ka" \}$

$b1 \in b2 = \{ "pe", "en", "ng", "gu", "um", "mu", "um", "ma", "an", "n", "p", "pm", "md", "dk", "k", "p", "an", "n", "d", "di" \}$

“ib”, “bu”, “uk”, “ka”, “t”, “ta”, “an”, “ng”,
 “gg”, “ga”, “al”, “l”, “ap”, “pa”}

$b2 \in b1 = \{“ka”, “ap”, “pa”, “an”, “n”, “t”, “ta”, “an”, “ng”, “g”, “ga”, “al”, “l”, “p”, “pe”, “en”, “ng”, “gu”, “um”, “mu”, “um”, “ma”, “an”, “n”, “p”, “pm”, “md”, “dk”, “k”, “d”, “di”, “ib”, “bu”, “uk”, “ka”\}$

Maka diperoleh nilai *string* 1 dan *string* 2:

$$34 + 35 / 43 + 35 = 69 / 78 = 0,88$$

Tabel 3. Contoh Hasil Proses Perhitungan Pencocokan Pola Percakapan

No	Input 1	Input 2	Skor
1.	bagaimana kabar hari ini	bagaimana kabar hari ini	1
2.	bagaimana kabar hari ini	gimana kabar hari ini	0.9069
3.	bagaimana kabar hari ini	apa kabar hari ini	0.775
4.	bagaimana kabar hari ini	hari ini kabarnya gimana	0.7826
5.	bagaimana kabar hari ini	hari ini gimana kabar	0.8372
6.	bagaimana kabar hari ini	halo	0.0769

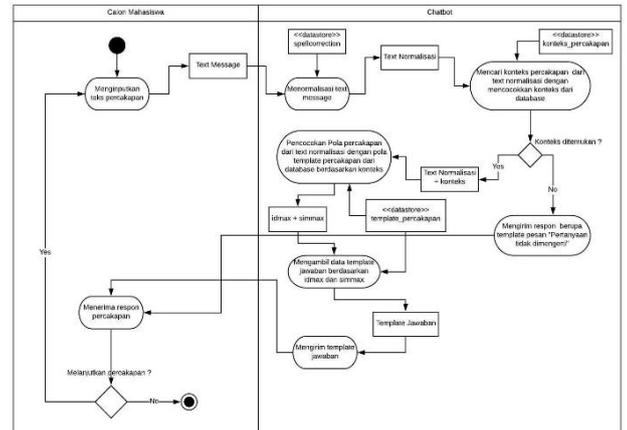
Tabel 3 menunjukkan input 1 berupa string yang sama sedangkan input 2 adalah variasi string. Percobaan satu menghasilkan skor 1 karena input 1 dan 2 sama persis. Percobaan enam menghasilkan skor 0 karena kedua input berbeda sama sekali.

1.4 PERANCANGAN

Hasil dari perancangan pada tahap ini sebagai acuan pada tahap implementasi. Perancangan aktivitas aplikasi chatbot menggambarkan proses aktivitas aplikasi chatbot bagian pengguna dan bagian humas Polban atau admin yang terjadi dari awal hingga akhir.

1. Perancangan Proses Aplikasi Chatbot

keluaran dari proses aplikasi *chatbot* adalah respon jawaban yang sesuai berdasarkan hasil pencocokan pola dari pertanyaan yang diinputkan oleh pengguna. Gambar 3. adalah perancangan proses aplikasi *chatbot*.



Gambar 3. Activity Diagram Aplikasi Chatbot

Penjelasan detail dari perancangan aktivitas chatbot berdasarkan Gambar 3:

a. Proses normalisasi *text message*

Berdasarkan analisis metode normalisasi, setelah pengguna menginputkan sebuah pertanyaan yang tersimpan kedalam *textMessage* dilakukan proses normalisasi hasil *Output* dari proses ini adalah *text* hasil normalisasi. Berikut ini adalah *pseudocode* normalisasi yang diimplementasikan kedalam *class normalizer()* terbagi menjadi dua *method* yaitu fungsi *markRemoval()* dan *spellCorrection()*.

```

Function markRemoval(textMessage: String)

Kamus Data
msgMark : String

Algoritma:
BEGIN

    REPLACE “?” IN textMessage WITH “”

    REPLACE “!” IN textMessage WITH “”
  
```

```

REPLACE “,” IN textMessage
WITH “”

REPLACE “:” IN textMessage
WITH “”

REPLACE “.” IN textMessage
WITH “”

REPLACE “,” IN textMessage
WITH “”

REPLACE “/” IN textMessage
WITH “”

REPLACE “\” IN textMessage
WITH “”

REPLACE “-” IN textMessage
WITH “”

REPLACE “(” IN textMessage
WITH “”

REPLACE “)” IN textMessage
WITH “”

REPLACE “{” IN textMessage
WITH “”

REPLACE “}” IN textMessage
WITH “”

msgMark ← textMessage

RETURN msgMark

END

```

```

ng IN
msgSpell
WITH
listSpell.
correction

END IF

i ← i + 1

END WHILE

RETURN msgSpell

END

```

b. Proses mencari kategori percakapan

Pesan yang diinputkan oleh pengguna setelah melewati proses normalisasi, selanjutnya yaitu mencari kategori percakapan atau inti dari pertanyaan yang diinputkan oleh pengguna. kategori percakapan yang sudah ditemukan digunakan untuk proses pencocokan pola supaya proses yang dilakukan bisa lebih cepat dan tidak melakukan proses pencocokan pola pada keseluruhan data dari data *template* percakapan. Berikut ini adalah *pseudocode* mencari kategori percakapan yang diimplementasikan kedalam *function getCategory()*.

```

Function spellCorrection(msgMark:
String)

Kamus Data
listSpell: Array
msgSpell: String
i: Integer

Initial State
ListSpell menyimpan data array of
object yang berisi data Spell
Correction yang telah diambil dari
tabel database spellcorrection

Algoritma:
BEGIN

msgSpell ← msgMark
i ← 0

WHILE (i < listSpell) DO
    IF (msgSpell REGEX
listSpell.misspelling) TRUE THEN
        REPLACE
listSpell.
misspelli

```

```

Function getCategory(msgSpell:
String)

Kamus Data
listCategory: Array of object
idCategory: String
i: integer

Initial State
listCategory menyimpan data array
of object yang berisi data Kategori
Percakapan yang telah diambil dari
tabel database kategori_percakapan

Algoritma:
BEGIN

i ← 0

WHILE(i <
count(listCategory)) DO
    IF (msgSpell REGEX
listCategory.kateg
ori_isi,
listCategory.unsur
_isi) TRUE
    THEN

```

```

IdCategory
←
listCategory.konteks
Id

ELSE

i ← i + 1

END WHILE

RETURN IdCategory

END

```

- c. Proses kirim respon berupa template pesan “pertanyaan tidak dimengerti”.

Proses ini dilakukan jika kategori percakapan tidak ditemukan dengan mengirim respon berupa template pesan “pertanyaan tidak dimengerti” ke pengguna. Berikut ini pseudocode kirim respon berupa template pesan yang akan diimplementasikan kedalam class LineBotController().

```

IF (idCategory[] = NULL) THEN

msgText = “Pertanyaan tidak dimengerti”

END IF

```

- d. Proses pencocokan pola

Proses ini dilakukan setelah proses normalisasi dan proses mencari kategori percakapan. Proses pencocokan pola dilakukan dengan menggunakan perhitungan SSM dengan bantuan Bigram.

- d.1 Proses mengambil template respon berdasarkan idmax dan simmax

Sub proses ini dilakukan diakhir proses pencocokan pola, aka akan didapatkan skor tertinggi dari proses pencocokan pola. Dari skor tertinggi didapatkan juga index dari template pertanyaan yang disimpan kedalam variabel idmax. Berikut ini adalah *pseudocode* proses pencocokan pola yang diimplementasikan kedalam *function handlechat()*.

```

Function handlechat(msgSpell:
String)

```

Kamus Data

```

listTemplate: Array of Object
msgCategory: String
idmax: String
simmax: Float
sim: Float
jawaban: String
i: Integer

INITIAL STATE

listTemplate menyimpan data array
of object yang berisi data
template_percakapan yang telah
diambil dari tabel database
template_percakapan

Algoritma:

BEGIN

msgCategory ←
getCategory(msgSpell)

idmax ← “ ”

simmax ← 0

i ← 0

WHILE(i <
count(listTemplate)) DO

sim ←
similar(msgSpell,
listTemplate.template_
te_pertanyaan[i])

IF sim > simmax
THEN

simmax ←
sim

idmax ←
listTempla
te.prcId[i
]

END IF

i ← i + 1

END WHILE

jawaban← SELECT
template_jawaban FROM
templatepercakapan WHERE
templatepercakapan.prcId =
idmax

RETURN jawaban

END

```

Dibawah ini adalah *pseudocode* bagaimana mendapatkan skor dengan pendekatan SSM yang akan diimplementasikan kedalam *function similar()*.

```
Function similar(msgSpell:
String, templatePercakapan)
```

KAMUS DATA

```
msgTemplate : String
b1, b2, b12, b21 : ArrayList
c1, c2, c12, c21 : Float
```

Algoritma:

BEGIN

```
msgTemplate ←
templatePercakapan
b1 ← bigram(msgSpell)
b2 ← bigram(msgTemplate)
b12 ← b1.retainAll(b2)
//Irisan b1 & b2
b21 ← b2.retainAll(b1)
//Irisan b2 & b1
c1 ← array_count(b1)
c2 ← array_count(b2)
c12 ← array_count(b12)
c21 ← array_count(b21)
similar ← (c12+c21) /
(c1+c2)
RETURN similar
```

END

Algoritma:

BEGIN

```
textMessage ← messageText
replyMessage ←
textMessage, replyToken
lineMessagingClient.reply
(replyMessage).get()
```

END

2. Perancangan Proses Aplikasi Chatbot Bagian Admin Humas.

a. Perancangan proses melihat riwayat percakapan.

Proses untuk melihat riwayat percakapan menghasilkan keluaran data riwayat percakapan. Proses melihat riwayat percakapan digambarkan dengan Gambar 4. dibawah ini.

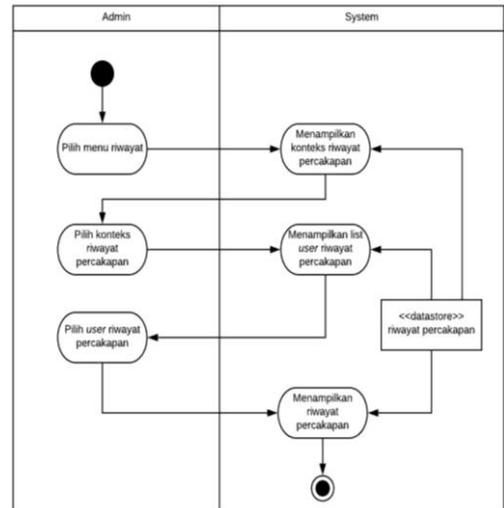
e. Kirim template jawaban kepada pengguna.

Proses ini dilakukan jika proses pencocokan pola telah selesai dan telah mendapatkan template jawaban yang sesuai dengan hasil skor tertinggi dari pencocokan pola. proses ini memerlukan fungsi dari *library* yang dimiliki oleh *Line messenger* yaitu *reply message*. Berikut ini adalah *pseudocode* yang diimplementasikan kedalam prosedur *reply()*.

```
Procedure reply(replyToken:
String, msgText: String)
```

Kamus Data

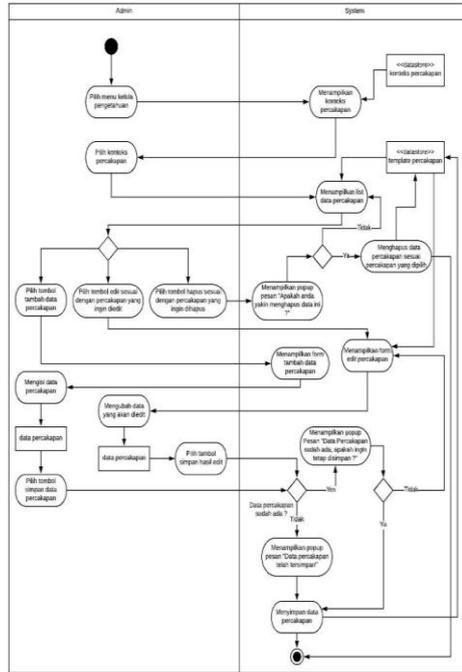
```
textMessage : TextMessage
replyMessage : ReplyMessage
lineMessagingClient :
LineMessagingClient
```



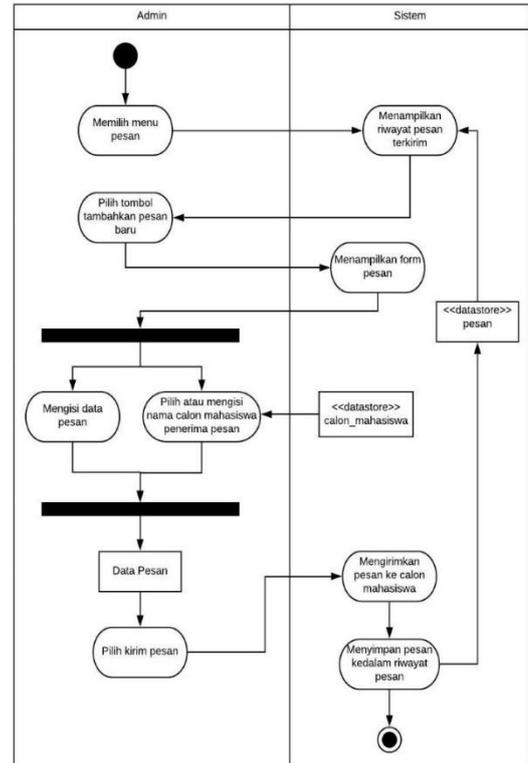
Gambar 4. Activity Diagram melihat riwayat percakapan

b. Perancangan proses mengelola pengetahuan informasi percakapan

Gambar 5. dibawah ini adalah proses untuk mengelola pengetahuan informasi percakapan yang menghasilkan keluaran proses data yang telah ditambahkan atau diedit bisa tersimpan kedalam *database* dan data yang telah dihapus bisa terhapus dari *database*.



Gambar 5. Activity Diagram mengelola pengetahuan informasi percakapan



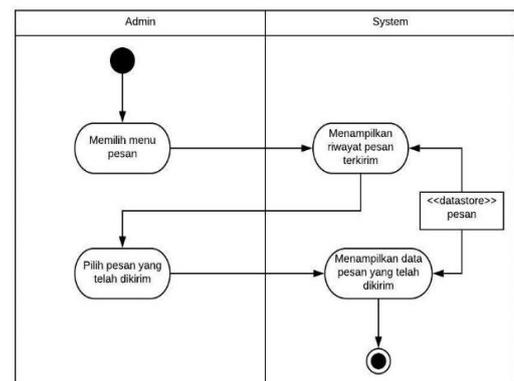
Gambar 6. Activity Diagram mengirim pesan

- c. Perancangan proses mengirim pesan kepada pengguna

Gambar 6 dibawah ini adalah proses untuk mengirim pesan kepada pengguna yang menghasilkan keluaran akhir dari proses adalah menyimpan pesan kedalam riwayat pesan yang terkirim.

- d. Perancangan proses melihat riwayat pesan terkirim

Gambar 7 dibawah ini adalah proses untuk melihat riwayat pesan terkirim yang menghasilkan keluaran akhir dari proses adalah menampilkan data riwayat pesan yang terkirim.



Gambar 7. Activity Diagram melihat riwayat pesan terkirim

1.5 PEMBAHASAN DAN IMPLEMENTASI

Setelah penjelasan terkait metode normalisasi dan metode pencocokan pola, dibawah ini adalah contoh tahapan untuk penerapan metode tersebut, tahapannya sebagai berikut :

- a. **Tahap 1:** Pengguna menginputkan sebuah pesan pertanyaan ke aplikasi chatbot informasi kampus Polban. Isi dari pesan yang diinputkan tersebut adalah “kpn tgl pengumuman PMDK dibuka ???”. Pesan tersebut disimpan kedalam variabel msgText.
- b. **Tahap 2:** variabel msgText yang berisi pesan akan dilakukan proses normalisasi dengan melalui tiga tahap, yaitu :

1. Penghilangan tanda baca atau mark removal.

Tabel 4. Contoh Penerapan *Mark Removal*

Sebelum diproses	Hasil Proses
kpn tgl pengumuman PMDK dibuka ???	kpn tgl pengumuman PMDK dibuka

Dalam proses tersebut tanda baca yang dihilangkan adalah tanda tanya (?). Pesan yang telah diproses dari msgText akan disimpan kedalam variabel msgMark.

2. Membenarkan kata ejaan yang salah.

Tabel 5. Contoh Penerapan *Spell Correction*

Sebelum diproses	Hasil Proses
kpn tgl pengumuman PMDK dibuka	kapan tanggal pengumuman PMDK dibuka

Dalam proses tersebut membenarkan kata ejaan yang salah adalah kata “kpn” menjadi “kapan”. Pesan yang telah diproses dari msgSpace akan disimpan kedalam variabel msgSpell.

- c. **Tahap 3 :** variabel msgSpell yang berisi pesan yang telah diproses normalisasi akan dilakukan proses mencari konteks percakapan dengan cara menyeleksi dari setiap kata dibandingkan apakah terdapat konteks atau tidak. Konteks percakapan dibagi menjadi dua konteks yaitu kategori dan kata tanya.

Tabel 6. Contoh Penerapan Pencarian Konteks

Sebelum diproses	Hasil Proses Pencarian	
	Kategori	Kata Tanya
kapan tanggal pengumuman PMDK dibuka	tanggal	kapan

Hasil proses tersebut ditemukan konteks percakapan yaitu kategori “tanggal”, kata tanya “kapan” dan disimpan kedalam variabel msgContext. Sehingga tabel dibawah ini adalah tabel konteks yang diambil dari database dengan query sebagai lingkup untuk melakukan proses pencocokan pola pada tahap selanjutnya. Hasil dari *query* disimpan kedalam variabel msgTemplate untuk mengambil data pertanyaan berdasarkan indeks.

Tabel 7. Contoh Tabel Konteks dan Template

Kategori	Kata Tanya	Indeks	Pertanyaan
Tanggal	Kapan	1	kapan tanggal penyebaran undangan PMDK Polban
		2	kapan tanggal pengumuman PMDK Polban
		3	Kapan tanggal pengiriman berkas
		4	kapan tanggal pendaftaran PMDK Polban ditutup

- d. **Tahap 4 :** mengisialisasi idmax dengan null atau belum terdefinisi dan simmax dengan nilai nol.

Idmax = null,
simmax = 0;

Melakukan pengulangan jika indeks dalam msgTemplate belum mencapai nilai 4 sesuai dengan total tabel pertanyaan. Dibawah ini sebagai tahapan pengulangan dan perhitungan pencocokan pola SSM dan bigram antara msgSpell dengan msgTemplate, sebagai berikut :

1. Perhitungan dari indeks satu.

MsgSpell : “kapan tanggal pengumuman PMDK dibuka”

MsgTemplate : “kapan tanggal penyebaran undangan PMDK Polban”

Bigram1 msgSpell : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n g, g u, u m, m u, u m, m a, a n, n , p, p m, m d, d k, k , d, d i, i b, b u, u k, k a}

Bigram2 msgTemplate : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n y, y e, e b, b a, a r, r a, a n, n , u, u n, n d, d a, a n, n g, g a, a n, n , p, p m, m d, d k, k , p, p o, o l, l b, b a, a n}

Bigram1 ∈ Bigram2 : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, a n, n , a n, n g, g a, a n, n , p, p m, m d, d k, k , p, a n}

Bigram2 ∈ Bigram1 : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n g, a n, n , p, p m, m d, d k, k , k a}

Diperoleh hasil nya sebagai berikut :

Count Bigram1 = 35

Count Bigram2 = 44

Count Bigram1 ∈ Bigram2 = 30

Count Bigram2 ∈ Bigram1 = 25

$30 + 25 / 35 + 44 = 55 / 79 = 0,69$

Hasil perhitungan dari indeks satu disimpan kedalam variabel sim. Setelah itu dilakukan perbandingan jika nilai sim lebih besar dari nilai simmax bernilai *true* maka nilai sim diperbarui dengan nilai simmax dan idmax diperbarui dengan indeks satu, jika bernilai *false* maka nilai sim dan nilai idmax tidak diperbarui.

2. Perhitungan dari indeks dua

MsgSpell : “kapan tanggal pengumuman PMDK dibuka”

MsgTemplate : “kapan tanggal pengumuman PMDK Polban”

Bigram1 msgSpell : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n g, g u, u m, m u, u m, m a, a n, n , p, p m, m d, d k, k , d, d i, i b, b u, u k, k a}

Bigram2 msgTemplate : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n g, g u, u m, m u, u m, m a, a n, n , p, p m, m d, d k, k , p, p o, o l, l b, b a, a n}

Bigram1 ∈ Bigram2 : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n g, g u, u m, m u, u m, m a, a n, n , p, p m, m d, d k, k , p, a n}

Bigram2 ∈ Bigram1 : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n g, g u, u m, m u, u m, m a, a n, n , p, p m, m d, d k, k , k a}

Diperoleh hasil nya sebagai berikut :

Count Bigram1 = 35

Count Bigram2 = 35

Count Bigram1 ∈ Bigram2 = 31

Count Bigram2 ∈ Bigram1 = 30

$31 + 30 / 35 + 35 = 61 / 70 = 0,87$

Hasil perhitungan dari indeks dua disimpan kedalam variabel sim untuk diperbarui. Setelah itu dilakukan perbandingan seperti perhitungan sebelumnya dan algoritma yang sama. Sehingga dikarenakan nilai sim 0,87 lebih besar dari nilai simmax 0,69 maka nilai simmax diperbarui dengan nilai 0,87 dan nilai idmax adalah 2.

3. Perhitungan dari indeks empat

MsgSpell : “kapan tanggal pengumuman PMDK dibuka”

MsgTemplate : “kapan tanggal pendaftaran PMDK Polban ditutup”

Bigram1 msgSpell : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n g, g u, u m, m u, u m, m a, a n, n , p, p m, m d, d k, k , d, d i, i b, b u, u k, k a}

Bigram2 msgTemplate : {k a, a p, p a, a n, n , t, t a, a n, n g, g g, g a, a l, l , p, p e, e n, n d, d a, a f, f t, t a, a r, r a, a n, n , p, p m, m d, d k, k , p, p o, o l, l b, b a, a n, n , d, d i, i t, t u, u t, t u, u p}

Bigram1 ∈ Bigram2 : {k a, a p, p a, a n, n, t, t a, a n, n g, g g, g a, a l, l, p, p e, e n, t a, a n, n, p, p m, m d, d k, k, p, a n, n, d, d i}

Bigram2 ∈ Bigram1 : {k a, a p, p a, a n, n, t, t a, a n, n g, g g, g a, a l, l, p, p e, e n, n g, a n, n, p, p m, m d, d k, k, d, d i, k a}

Diperoleh hasil nya sebagai berikut :

Count Bigram1 = 35

Count Bigram2 = 44

Count Bigram1 ∈ Bigram2 = 29

Count Bigram2 ∈ Bigram1 = 27

$29 + 27 / 35 + 44 = 56 / 79 = 0,70$

Hasil perhitungan dari indeks empat disimpan kedalam variabel sim untuk diperbarui. Setelah itu dilakukan perbandingan seperti perhitungan sebelumnya dan algoritma yang sama. Sehingga dikarenakan nilai sim 0,70 tidak lebih besar dari nilai simmax 0,87 maka nilai simmax tidak diperbarui. Maka hasilnya tetap dengan nilai 0,87 dan nilai idmax adalah 2.

Setelah dilakukan tahapan pengulangan Tabel 8 dibawah ini sebagai hasil rekapitulasi perhitungan pencocokan pola SSM antara variabel msgSpell dengan variabel msgTemplate diurutkan berdasarkan nilai tertinggi.

Tabel 8. Hasil Rekapitulasi Perhitungan Pencocokan Pola

Indeks	Pertanyaan	Hasil Perhitungan
2	kapan tanggal penyebaran undangan PMDK Polban	0,87
4	kapan tanggal pengumuman PMDK Polban	0,70
1	Kapan tanggal pengiriman berkas	0,69
3	kapan tanggal pendaftaran PMDK Polban ditutup	0,66

f. **Tahap 5** : Hasil perhitungan tertinggi adalah 0,87. Selanjutnya mengambil data jawaban berdasarkan indeks dan pertanyaan hasil perhitungan tertinggi untuk dikirimkan kepada pengguna.

Dibawah ini ada Tabel 9 untuk mengetahui data jawaban dari tabel template.

Tabel 8. Tabel Template Berdasarkan Konteks

Indeks	Pertanyaan	Jawaban
1	kapan tanggal penyebaran undangan PMDK Polban	Penyebaran undangan dilakukan dari tanggal 4 Februari sampai dengan 29 Maret 2019
2	kapan tanggal pengumuman PMDK Polban	Pengumuman PMDK-PN pada tanggal 16 April 2019
3	Kapan tanggal pengiriman berkas	29 Maret 2019 (dengan cap pos) dan 3 April 2019 diterima panitia (mandiri)
4	kapan tanggal pendaftaran PMDK Polban ditutup	Pendaftaran ditutup pada tanggal 29 Maret 2019.

Berikut implementasi tampilan dari chatbot pada aplikasi *LINE Messenger* dan aplikasi untuk admin.



Gambar 6. Tampilan LINE Chatbot saat pengguna pertama kali menambahkan chatbot sebagai teman



Gambar 7. Tampilan LINE Chatbot saat pengguna memilih pilihan seperti jurusan yang ada di kampus Polban.



Gambar 8. Tampilan untuk admin dapat mengolah data informasi yang digunakan untuk informasi pada chatbot

pertanyaan pengguna dengan data template percakapan yang ada didalam database.

DAFTAR PUSTAKA

- [1] B. S. D. Oetomo, "Perencanaan dan Pembangunan Sistem Informasi," *Yogyakarta: Penerbit Andi*, 2002.
- [2] I. Ma'ruf, "Perkembangan Media Komunikasi di Indonesia," 12 March 2012. [Online]. Available: <http://nanangsuryadi.lecture.ub.ac.id/2012/03/perkembangan-media-komunikasi-di-indonesia/>. [Diakses 19 February 2019].
- [3] A. T. Mahany, "Pengguna LINE di Indonesia naik hingga 200 persen," *Brilio.net*, 3 September 2016. [Online]. Available: <https://www.brilio.net/gadget/pengguna-line-di-indonesia-naik-hingga-200-persen-1609031.html>. [Diakses 19 Februari 2019].
- [4] Y. Li, Z. Bandar, D. McLean dan J. O'Shea, "A Method for Measuring Sentence Similarity and its Application to Conversational Agents," 2004.
- [5] Z. Pratama, E. Utami dan R. M. Arief, "Analisa Perbandingan Jenis N-GRAM Dalam Penentuan Similarity Pada Deteksi Plagiat," *Citec Journal*, vol. 4, pp. 256-257, 260-261, 2017.

1.6 KESIMPULAN

Dari hasil perumusan masalah sampai pada tahap implementasi dapat disimpulkan bahwa, penggunaan chatbot merupakan suatu solusi untuk menyelesaikan masalah yang terjadi pada sistem yang sedang berjalan saat ini. Hal ini karena adanya NLP yang memungkinkan komputer dapat berinteraksi dengan manusia dengan menggunakan salah satu metode pada NLP seperti *Similarity Sentences Measurement* yang digunakan untuk mengembangkan sebuah aplikasi chatbot. Adanya proses Text Preprocessing demi mengurangi kesalahan dalam proses pencocokan pola dari hasil masukan