Topic Modeling in Conversational Dialogs for Naming Intent Labels Using LDA

Pemodelan Topik pada Dialog Percakapan untuk Penamaan Label Intent Menggunakan LDA

Laksma Wiramurti Narendra[1]
[1]Teknologi Informasi, Fakultas Sains dan Informatika, Institut Sains Terapan dan Teknologi Surabaya
E-mail: *[1]laksma80@gmail.com

Abstract – *Chatbot research has grown in recent years in line with the development of Artificial Intelligent (AI) and Neural Network (NN) technologies. Natural Language Processing (NLP) as part of NN is used by chatbots, especially in Natural Language Understanding (NLU) tasks. Chatbots make use of intent classifiers to understand the intent of user-sent messages. To make the chatbot function properly according to its domain, mapping intent on model training data becomes a separate problem for researchers. This is because datasets with intent labeled for training chatbot models in Indonesian are still rarely available. In this study, the naming of the intent for the chatbot training data can be made using Latent Dirichlet Allocation (LDA) method, the question datasets are taken from the complaint log of one of the credit distributors in Indonesia totaling 143,520 messages from 2015 to 2019. From the results of topic modeling using LDA, it is able to map 8 topics that can be used in naming the intent for making chatbot models.*

Keywords *— chatbot, intent labelling, lda, topic modelling*

Abstrak – Penelitian chatbot semakin berkembang dalam beberapa tahun ini seiring dengan perkembangan teknologi *Machine Learning* (ML) dan *Artificial Intelligent* (AI). *Natural Language Processing* (NLP) sebagai bagian dari ML digunakan oleh *chatbot* terutama pada tugas *Natural Language Understanding* (NLU). *Chatbot* memanfaatkan pengklasifikasian *intent* untuk memahami maksud pada pesan yang dikirim pengguna. Untuk menjadikan *chatbot* berfungsi dengan baik sesuai dengan domainnya maka pemetaan *intent* pada data pelatihan model menjadi permasalahan tersendiri bagi para peneliti. Hal ini disebabkan *dataset* berlabel *intent* untuk pelatihan model *chatbot* dalam bahasa Indonesia masih jarang tersedia. Pada penelitian ini, penamaan intent untuk data pelatihan *chatbot* dapat dibuat dengan menggunakan metode *Latent Dirichlet Allocation* (LDA), *dataset* pertanyaan diambil dari *log* komplain salah satu distributor pulsa di Indonesia sejumlah 143.520 pesan sejak 2015 hingga 2019. Dari hasil pemodelan topik menggunakan LDA mampu memetakan 8 topik yang kemudian dapat digunakan dalam penamaan *intent* pada pelatihan model *chatbot*.

Kata Kunci — *intent*, lda, pemodelan topik, penamaan label

1. PENDAHULUAN

Chatbot adalah program komputer yang dapat melakukan tugas otomatis dengan pengguna menggunakan pesan atau ucapan dalam bentuk percakapan [1]. Kemampuan *chatbot* yang semakin cerdas tidak terlepas dari teknologi NLP, *chatbot* dapat belajar dan berkembang dengan mempelajari data contoh *question-answer* (QA) sehingga *chatbot* dapat berkomunikasi secara alami [2]. Versi awal *chatbot* yang memanfaatkan teknologi NLP adalah EIZA, dikembangkan oleh *Josep Weizenbaum* pada pertengahan tahun 1960 [3]. *Chatbot* tersebut menggunakan metode "*pattern-matching*" dan '*substitution*" untuk mempresentasikan responnya kepada pengguna.

Dengan berkembangnya teknologi NLP, konsep *chatbot* yang pada mulanya berorientasi pada tugas, kini mulai berubah ke konsep percakapan berbasiskan data yang menggunakan *machine learning* dan *artificial intelligence* [4]. Beberapa *framework chatbot* popular berbasiskan *machine learning* dikembangkan seperti *Cortana* dari *Microsoft*, *Dialogflow* milik *Google*, *IBM Watson*, *Amazon Lex* dan *Rasa Open Source*. *Framework* tersebut bekerja dengan tugas-tugas NLP serta NLU. Salah satu tahap dalam pembuatan *chatbot* tersebut adalah membuat data yang berlabel *intent* untuk pelatihan model NLU [5]–[7].

Maksud atau tujuan dari pesan pengguna dikenal sebagai *intent*. Dengan kata lain, intent mencerminkan apa yang coba dikatakan atau dicapai pengguna, ini akan menentukan tindakan selanjutnya untuk mendapatkan hasil yang sesuai tujuan [2]. Untuk ucapan "Saya ingin memesan es krim vanilla", *intent* dapat berupa "*order_ice_cream*", contoh lain ucapan sederhana seperti "Tidak" dapat dikategorikan ke dalam *intent* "*deny*" atau ucapan "Selamat pagi" ke dalam kategori *intent* "*greeting*". Sistem dalam *chatbot* akan mendeteksi *intent* tersebut dan selanjutnya akan memberikan respon jawaban atau tindakan yang akan dipilih berdasarkan *intent* yang dituju [8].

Deteksi *intent* atau pengklasifikasian intent membutuhkan data pelatihan dalam bentuk frasa atau kalimat [9]. Data pelatihan tersebut dapat dibuat secara manual seperti *question-answer* (QA) namun teknik ini akan membutuhkan banyak waktu dan biaya. Pendeteksian topik pada sekumpulan dokumen pertanyaan atau percakapan dapat menjadi solusi untuk permasalahan ini [10].

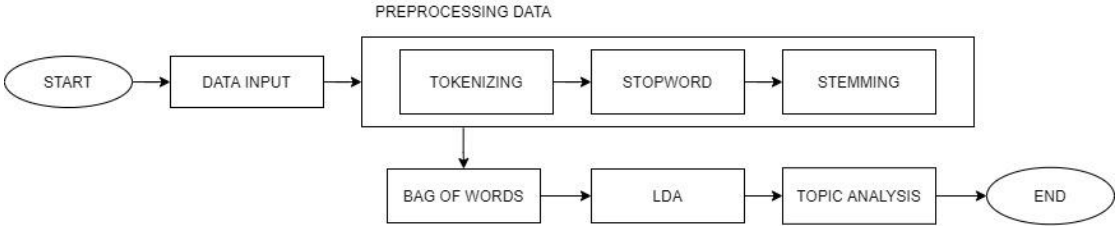
Pemodelan topik merupakan salah satu pendekatan pada bidang *text mining* untuk menemukan data-data teks tersembunyi dan menemukan hubungan antara teks yang satu dengan lainnya [11]. Secara sederhana metode ini mengelompokkan data teks berdasarkan suatu topik tertentu. Pemodelan topik bekerja seperti metode *clustering* yakni mengelompokkan dokumen berdasarkan kemiripannya.

Metode pada pemodelan topik banyak jenisnya yang dapat digunakan. Penelitian [12] menggunakan *Word2Vec* sebagai *word embedding* dan *clustering intent* dengan *K-Means*, jumlah *cluster* ditentukan dengan metode *elbow* dan algoritma *greedy* digunakan untuk menggabungkan *cluster* yang representasi kata-katanya berdekatan. Namun, metode tradisional ini tidak efektif dalam data dimensi tinggi karena keterbatasan ruang fitur dan pilihan metrik jarak antar kata [13]. Metode lain yang digunakan oleh [13] memanfaatkan *transfer learning* dengan jumlah data berlabel yang terbatas untuk menemukan *intent* baru. Metode ini masih membutuhkan data contoh berlabel yang harus disediakan. Selain itu, pemodelan topik dapat juga menggunakan metode seperti *Latent Semantic Analysis* (LSA), *Probabilistic Latent Semantic Analysis* (PLSA), dan yang saat ini cukup popular adalah *Latent Dirichlet Allocation* (LDA). Menurut [14], LDA merupakan peningkatan model campuran yang menangkap pertukaran kata-kata dan dokumen dari model PLSA dan LSA.

Pada penelitian ini akan digunakan metode LDA untuk mengekstrak topik pada data *log* komplain berbahasa Indonesia, sehingga diharapkan dapat memperoleh analogi kata yang akan digunakan untuk pengelompokkan *intent* yang selanjutnya data tersebut dapat digunakan dalam pembuatan data pelatihan *chatbot* yang telah berlabel *intent*.

2. METODE PENELITIAN

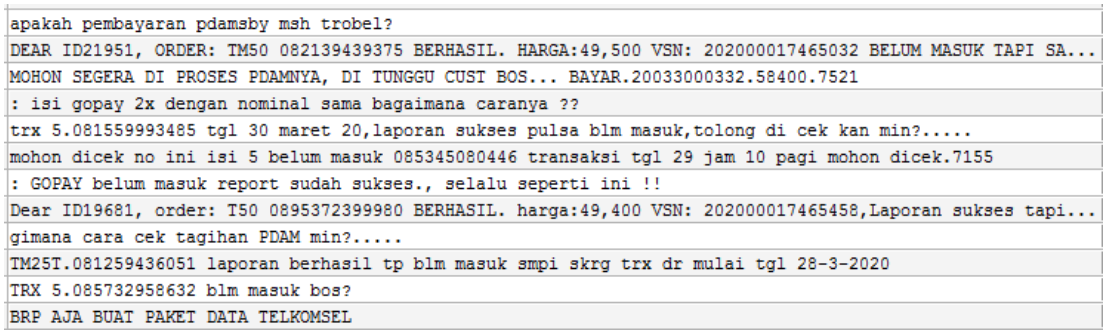
Penelitian ini akan menggunakan data yang berasal dari *server* salah satu perusahaan yang bergerak di bidang distributor pulsa di Indonesia. Perusahaan tersebut menggunakan aplikasi *chat* seperti Whatsapp, Telegram dan LINE untuk bertransaksi maupun berkomunikasi dengan agen-agennya. Penelitian ini secara sistematis dilakukan berdasarkan Gambar 1.



Gambar 1. Alur Metode Penelitian

2.1. Pengambilan Data

Dataset diambil dari data *log server* pada tabel *chat* komplain sejak tahun 2015 hingga tahun 2020 dengan jumlah data sebesar 143.520 baris. Data tersebut berisikan pertanyaan dan komplain pelanggan dan disimpan kembali dalam bentuk *file excel*. Cuplikan data tersebut ditunjukkan pada Gambar 2.



Gambar 2. Cuplikaan isi info pertanyaan dan komplain

2.2. Preprocessing Data

Tahap *preprocessing* data bertujuan untuk merubah bentuk dokumen ke dalam bentuk yang lebih mudah dipahami oleh komputer serta mempercepat proses komputasi terutama pencarian dokumen yang relevan. Tugas utama tahap ini adalah membangun indeks pada koleksi dokumen yang akan membedakan antara dokumen yang satu dengan dokumen lainnya. Pembuatan indeks akan melibatkan proses linguistik yang bertujuan ekstraksi kata-kata penting pada setiap dokumen dan direpresentasikan sebagai *Bag of ords*. Proses linguistik yang digunakan terdiri dari tokenisasi, *slangwords*, *stopwords* dan *stemming*. Karena domain data berbasis bahasa Indonesia maka *stopwords* dan *stemming* yang digunakan menggunakan *library* Sastrawi.

2.3. Bag of Words

Bag of words merupakan representasi teks yang menggambarkan kemunculan kata-kata pada setiap dokumen. Model *Bag of words* melibatkan dua hal yakni kosakata dari kata-kata yang dikenal serta ukuran keberadaan kata-kata tersebut, dalam model ini akan mengabaikan urutan atau struktur kata-kata dalam dokumen. Model hanya memperhatikan apakah kata-kata tersebut muncul dalam dokumen, bukan posisi dimana kata-kata tersebut berada dalam dokumen. Nilai perhitungan kemunculan kata-kata tersebut yang akan digunakan pada proses pemodelan topik.

2.4. Latent Dirichlet Allocation (LDA)

LDA merupakan model probabilistik generatif yang dapat digunakan dalam pendeteksian topik pada suatu kumpulan data yang berukuran besar. Model LDA sendiri merupakan mesin pembelajaran yang bertipe *unsupervised*, cara kerjanya dengan mengasumsikan bahwa setiap dokumen memiliki topik yang dibentuk dari kata-kata yang berhubungan, sehingga suatu dokumen dapat dikatakan sebagai representasi topik-topik tersembunyi pada sekumpulan dokumen tersebut.

Menurut [15] pemodelan topik dilakukan dalam dua tahapan, yang pertama adalah melakukan pemodelan topik dengan mengatur penambahan dan pengurangan jumlah topik, dan selanjutnya tahap kedua pemodelan topik dilakukan berdasarkan jumlah iterasi. Hasil dari kedua tahap tersebut kemudian dilakukan analisa topik dengan membandingkan kata-kata yang ada pada pengelompokkan topik yang terbentuk. Analisa ini juga dapat dilakukan dengan melihat hasil visualisasi pada pemodelan LDA. Pada proses pemodelan topik dapat dilakukan secara berulang berdasarkan inisialisasi yang ditentukan pada jumlah iterasi dan jumlah topik.

Pada penelitian ini akan memanfaatkan *library Gensim* pada *python* dengan mengaktifkan terlebih dahulu *package* "*LdaModel*", *package* ini berfungsi untuk memodelkan probabilitas

 doi: https://doi.org/10.32503/jtecs.v2i1.1820

kemunculan kata-kata dalam dokumen dan menghasilkan *output* berupa data visual grafik yang menunjukkan *cluster* topik dengan pendistribusian kata-kata pada *cluster* tersebut.

2.5. Analisa Topik

Analisa topik yang dilakukan dengan memperhatikan data *output* berupa visual grafik yang membentuk *cluster* topik. Setiap topik yang terbentuk terdiri dari sekumpulan kata-kata berdasarkan *cluster*-nya, pengamatan secara subyektif dilakukan untuk menentukan kata-kata yang terkandung pada *cluster* tersebut akan mengarah pada topik tertentu dan akan diambil kesimpulan yang mendekati untuk suatu tema tertentu. Hasil dari penentuan topik tersebut perlu divalidasi kebenarannya, untuk itu dibutuhkan bantuan dari pihak perwakilan perusahaan yang lebih profesional dalam bidangnya.

3. HASIL DAN PEMBAHASAN

3.1. Pengambilan Data

Tahap pengambilan data dari *database server* dieksport kedalam bentuk file *csv* dan selanjutnya dilakukan proses filter terlebih dahulu secara manual berdasarkan grup pertanyaan saja dan bersifat unik, sehingga pertanyaan yang sama atau berulang akan dihapus dari *dataset*. Hasil data setelah difilter menjadi 65.536 baris pertanyaan. Data tersebut nantinya akan digunakan pada tahap berikutnya yaitu tahap *preprocessing* data.

3.2. Preprocessing Data

Preprocessing dilakukan dengan menggunakan *python* pada *Anaconda3*. Langkah pada proses ini dimulai dengan men-*download* dan mengaktifkan beberapa *library* yang dibutuhkan. Tahap *preprocessing* data dibagi menjadi empat tahap, yakni *tokenizing*, *replace slangword*, *stopword*, dan selanjutnya *stemming*.

3.2.1. Tokenizing

Langkah pertama *preprocessing* data adalah *tokenizing*. Proses ini akan memisahkan setiap kata menjadi unit-unit kecil ke dalam suatu *array*. Proses *tokenizing* yang dilakukan dengan memanfaatkan karakter spasi pada dokumen sebagai proses pemisah kata-kata tersebut. Tahap ini juga dilakukan proses *cleaning words*, langkah yang digunakan pada *cleaning words* terdiri dari menghilangkan teks kalimat yang terdiri dari satu kata saja, menghilangkan *mention*, *url*, tanda baca dan angka yang ada pada teks. Normalisasi juga dilakukan dengan merubah setiap huruf dengan karakter *uppercase* menjadi karakter *lowercase*.

3.2.2. Replace Slangword

Replace slang word adalah proses subtitusi kata-kata singkat atau kata gaul dengan ejaan bahasa Indonesia yang benar. Metode yang digunakan yakni dengan mengganti *slang word* yang ada dalam data *token* berdasarkan kamus data *slangword* yang telah disediakan. Tujuan proses ini supaya makna kata pada *term* dokumen dapat distandarkan. Skrip ditunjukkan pada Gambar 3.

```
#ambil data kata singkatan atau tidak baku
df=open('slang.txt',"r",encoding="utf-8", errors='replace')
slangS = df.readlines(); df.close()
#pisah key dan value data slang
slangS = [t.split(":") for t in slangS]
slangS = [[k.strip(), v.strip()] for k,v in slangS]
slangS = {k:v for k,v in slangS}
#substitute from slang
def slangword(str):
    T = word_tokenize(str)
    for i,t in enumerate(T):
        if t in slangS.keys():
            T[i] = slangS[t]
    return ' '.join(T)
```

Gambar 3. *Replace Slangword*

3.2.3. Stopword

Langkah selanjutnya adalah proses *stopword*, proses ini dilakukan untuk menghapus kata-kata yang tidak berarti sehingga kata yang penting saja atau bermakna yang akan menentukan pemodelan topik yang akan digunakan. Pada penelitian ini tahap *stopword* tidak menggunakan *library* yang disediakan *nltk* karena *stopword* dalam bahasa Indonesia tidak didukung oleh *library* tersebut. Untuk itu daftar *stopword* bahasa Indonesia dipersiapkan terlebih dahulu [16] kemudian *diupdate* sesuai kebutuhan, data *stopword* pada penelitian ini mencapai 817 kata *stopword* dalam bahasa Indonesia. Skrip ditunjukkan pada Gambar 4.

```
#ambil data stopwords bahasa Indonesia
df=open('stopwords_id.txt',"r",encoding="utf-8", errors='replace')
id_stop = df.readlines()
df.close()
id_stop = [t.strip().lower() for t in id_stop]

def removeStopword(str):
    stop_words  = set(id_stop)
    word_tokens = word_tokenize(str)
    filtered_sentence = [w for w in word_tokens if not w in stop_words]
    return ' '.join(filtered_sentence)
```

Gambar 4. *Stopword* Bahasa Indonesia

3.2.4. Stemming

Proses *stemming* merupakan proses untuk menemukan kata dasar dari sebuah kata, proses ini akan menghapus kata yang mempunyai awalan dan akhiran tanpa menganalisa apakah kata tersebut mempunyai kesamaan arti dengan kata yang lain. Dalam penelitian ini, proses *stemming* berbahasa Indonesia menggunakan *library* sastrawi untuk *python*. Skrip ditunjukkan pada Gambar 5.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

#Stemming Indonesian
def stemmingIndo(str):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return stemmer.stem(str)
```

Gambar 5. *Stemming* Dengan *Library* Sastrawi

3.3. Bag of Words

Hasil dari tahap *preprocessing* data berupa matriks yang berisi kata-kata yang muncul secara berulang, matriks tersebut selanjutnya akan dikonversikan ke model *bag of words* yang memperhitungkan jumlah kemunculan setiap kata pada matriks tersebut. Hasil perhitungan jumlah setiap kata tersebut selanjutnya akan digunakan pada perhitungan probabilitasnya pada LDA.

Untuk membuat *bag of words* dapat dilakukan dengan *library Gensim*, langkah awalnya adalah identifikasi frasa pada dokumen dengan menggabungkan semua kata dalam bentuk bigram dan trigram dengan syarat minimal 5. Kata bigram dan trigram yang berjumlah minimal 5 akan dianggap sebagai satu frasa kemudian frasa tersebut dijadikan ke dalam satu kalimat. Selanjutnya pembuatan kamus dilakukan dengan memanggil modul *dictionary*. Pembuatan *bag of words* menggunakan fungsi *doc2bow* pada *Gensim*, ini akan merubah dokumen menjadi format matriks dengan menghitung jumlah kemunculan setiap kata yang unik. Setelah *bag of words* berhasil dibuat, selanjutnya adalah mempersiapkan pembuatan model dengan memperhitungkan *text frequency-indexed document frequency* atau *tf-idf* dan membuat *corpus* menggunakan *tf-idf* tersebut..

 doi: https://doi.org/10.32503/jtecs.v2i1.1820

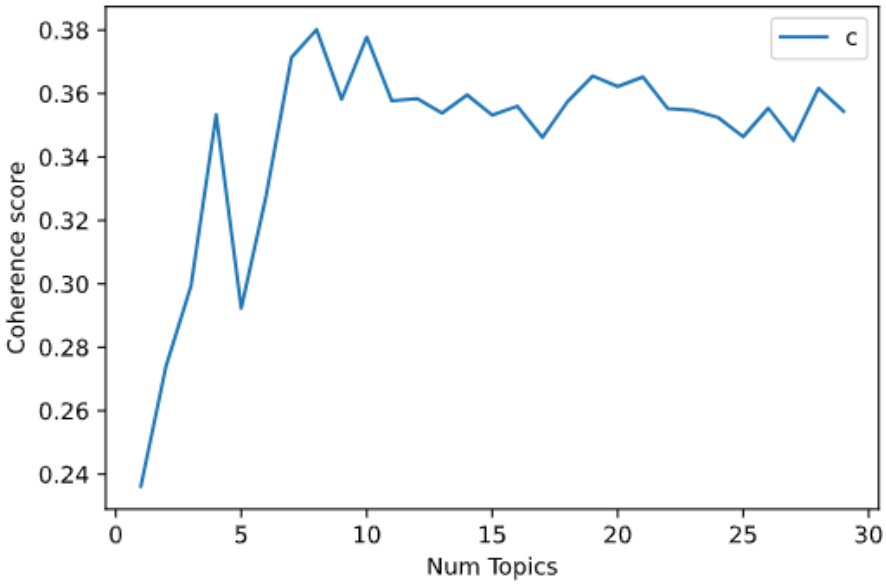
3.4. LDA

Tahapan berikutnya adalah pembuatan model LDA serta melakukan perhitungan *topic coherence*. Dua *input* utama untuk model LDA adalah kamus dan *corpus* yang didapatkan pada tahap sebelumnya. Penentuan jumlah topik pada model LDA dilakukan dengan cara melihat secara visual pada grafik *coherence score*. *Coherence score* merupakan ukuran yang digunakan untuk mengevaluasi pemodelan topik, model yang baik akan menghasilkan topik dengan *coherence score* yang tinggi. Untuk mendapatkan jumlah topik semaksimal mungkin maka langkah pengujian dilakukan dengan melakukan perulangan sebanyak n topik. Dalam penelitian ini jumlah awal n di-*setting* mulai dari 1 hingga 50 dengan *step* senilai 1 seperti *script* yang ditunjukkan pada Gambar 6. Grafik *coherence score* yang dihasilkan ditunjukkan pada Gambar 7.

```
start=1
limit=50
step=1
model_list, coherence_values = compute_coherence_values(dictionary,
corpus=corpus_tfidf,texts=text_list, start=start, limit=limit, step=step)

#show graphs
import matplotlib.pyplot as plt
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()
```

Gambar 6. Skrip Perhitungan *Coherence Score*

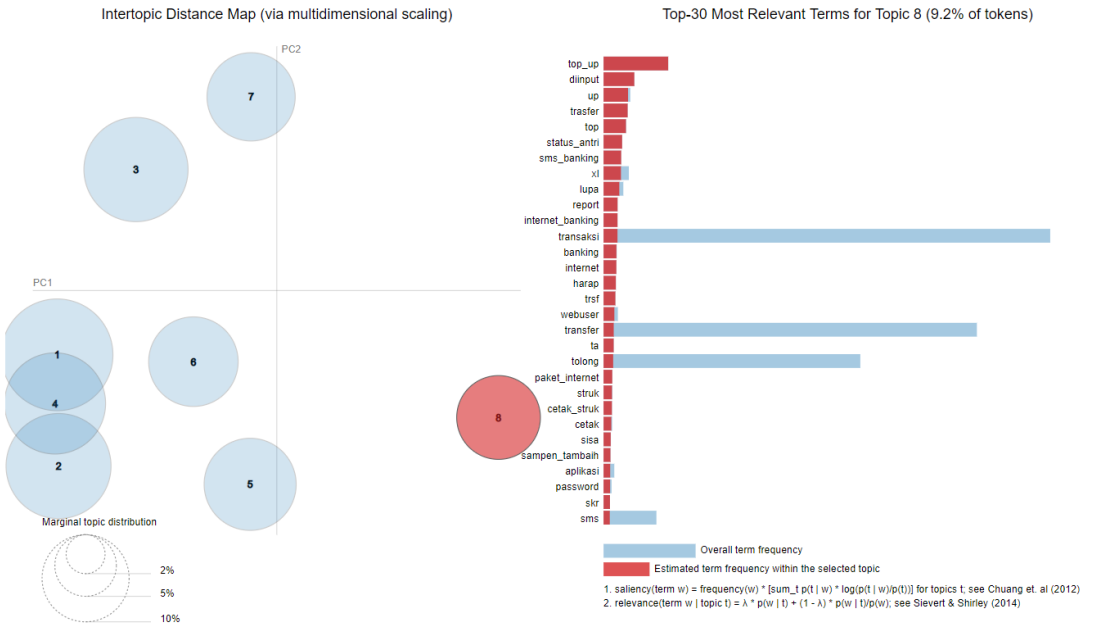


Gambar 7. Grafik *Coherence Score*

3.5. Analisa Data

Pada Gambar 7 menunjukkan nilai *coherence score* dengan nilai tertinggi 0,38019 untuk 8 topik. Semakin besar *coherence score*, maka hasil interpretasi pemodelan topik akan semakin baik. Hasil visual pemodelan topik ini menggunakan *library PyLDAvis*. *PyLDAvis* digunakan untuk membantu dalam penafsiran topik yang sesuai dengan dokumen. *Library* ini mengekstraksi informasi dari model LDA yang dibuat dengan memberikan informasi secara visual berupa tampilan *web* yang

interaktif. Tampilan visual ini dapat digunakan pada *jupyter notebook* serta dapat disimpan dalam bentuk *file html* sehingga mudah untuk dibagikan dan digunakan secara bersama-sama..



Gambar 7. Visualisasi Pemodelan Topik dengan *PyLDAvis*

Dari hasil visualisasi pada gambar 8 kemudian dilakukan evaluasi untuk mencari kesimpulan yang sesuai dengan data. Dalam menganalisa hasil pemodelan topik, penelitian ini membutuhkan bantuan profesional pada bidang keagenan pulsa tersebut sehingga didapatkan hasil akhir nama topik, kata-kata yang muncul dan penamaan intent yang ditunjukkan pada Tabel 1.

Tabel 1. Hasil Analisa Pemodelan Topik

Topik ke	Tema Topik	*Term*	Nama *Intent*
1	Bantuan Transaksi	transaksi, tunggu, tolong, proses	need_help
2	Informasi Produk	data, telkomsel, paket_data, pulsa	product_info
3	Chat Terima Kasih	terima_kasih, kasih, terima, mohon	thank_you
4	Komplain Deposit	topup, diinput, transfer, top, sms_banking	deposit_confirm
5	Kirim Uang	transaksi, uang, kirim, pending	send_money
6	Jalur Transaksi	telegram, tuju, referensi, transaksi	transaction_line
7	Komplain Pulsa	transaksi, pulsa, lapor, sukses, nomor	pulsa_complaint
8	Downline	transfer, downline, nama, tambah	downline

Pembuatan nama *intent* ditentukan berdasarkan tema topik dan menggunakan huruf kecil tanpa spasi, format ini mengikuti *platform chatbot* yang akan digunakan. Dalam penelitian ini *platform* yang akan digunakan selanjutnya adalah *framework Rasa Open Source* [8], pembahasan pada penelitian ini dititikberatkan pada pembuatan nama *intent* menggunakan LDA sehingga penggunaan *intent* pada pembuatan *chatbot* tidak akan dibahas pada penelitian ini.

4. KESIMPULAN

Dari hasil penelitian dapat diambil beberapa kesimpulan sebagai berikut:

a. Metode LDA terbukti dapat melakukan pemodelan topik dengan menampilkan kata-kata yang sering dibahas pada data layanan komplain antara pengguna dan sistem yang ditangani oleh

petugas. Dalam penentuan jumlah topik terbaik dilakukan perhitungan pada *coherence score* pada jumlah n topik.

b. Dalam proses pembuatan model LDA dibutuhkan dua inputan yakni *dictionary* dan *corpus* yang dapat diperoleh dari pembuatan *Bag of Words* berdasarkan dataset kata-kata atau sekumpulan dokumen. Perhitungan jarak antar kata pada metrik kata-kata tersebut mengindikasikan seberapa jauh kemiripan antar kata secara simantik dan makna, dengan demikian metode LDA melakukan pengelompokkan kata-kata dalam suatu *cluster* yang berdekatan.

c. Penggunaan model topik dapat diterapkan untuk penentuan nama *intent* yang dibutuhkan dalam pembuatan *chatbot*, metode ini lebih efektif dalam proses pembuatan *chatbot* yang telah memiliki data percakapan dibandingkan membuat data pelatihan percakapan dari awal.

5. SARAN

Untuk efisiensi waktu, maka performa komputasi perlu ditingkatkan untuk menangani proses seperti stemming dan pembentukan *bag of words* dengan *dataset* yang berjumlah besar. Selain itu, hasil *processing* data masih banyak ditemukan kata-kata yang tidak baku, ini akan mempengaruhi performa pemodelan topik. Dengan demikian daftar kata *slangword* berbahasa Indonesia perlu dilengkapi kedepannya.

Untuk membuat *chatbot* menjadi alami selayaknya manusia, maka pembuatan data pelatihan model *chatbot* memerlukan data berlabel *intent* yang lengkap dan mampu membedakan maksud antar frasa atau antar kalimat, metode LDA pada penelitian ini merupakan salah satu metode yang dapat digunakan dalam proses tersebut, namun disarankan pada penelitian selanjutnya untuk mencari metode yang lebih baik dalam pembentukan data berlabel *intent*.

DAFTAR PUSTAKA

[1] N. Akma, M. Hafiz, A. Zainal, M. Fairuz, and Z. Adnan, "Review of Chatbots Design Techniques," Int. *J. Comput. Appl*., vol. 181, no. 8, pp. 7–10, Aug. 2018, doi: 10.5120/ijca2018917606.

[2] S. Alias, M. S. Sainin, T. S. Fun, and N. Daut, "Intent Pattern Discovery for Academic Chatbot - A Comparison between N-gram model and Frequent Pattern-Growth method," in *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, Kuala Lumpur, Malaysia, Dec. 2019, pp. 1–5. doi: 10.1109/ICETAS48360.2019.9117315.

[3] J. Agassi and J. Wiezenbaum, "Computer Power and Human Reason: From Judgment to Calculation," *Technol. Cult*., vol. 17, no. 4, p. 813, Oct. 1976, doi: 10.2307/3103715.

[4] N. T. M. Trang and M. Shcherbakov, "Enhancing Rasa NLU model for Vietnamese chatbot," vol. 9, p. 7, 2021.

[5] S. Sahay, S. H. Kumar, E. Okur, H. Syed, and L. Nachman, "Modeling Intent, Dialog Policies and Response Adaptation for Goal-Oriented Interactions," *ArXiv191210130 Cs*, Dec. 2019, Accessed: Jul. 05, 2021. [Online]. Available: http://arxiv.org/abs/1912.10130

[6] A. Jiao, "An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network," *J. Phys. Conf. Ser*., vol. 1487, p. 012014, Mar. 2020, doi: 10.1088/1742-6596/1487/1/012014.

[7] D. Theosaksomo and D. H. Widyantoro, "Conversational Recommender System Chatbot Based on Functional Requirement," in *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, Bali, Indonesia, Oct. 2019, pp. 154–159. doi: 10.1109/TSSA48701.2019.8985467.

[8] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management," *ArXiv171205181 Cs*, Dec. 2017, Accessed: Jul. 05, 2021. [Online]. Available: http://arxiv.org/abs/1712.05181

[9] J.-K. Kim, G. Tur, A. Celikyilmaz, B. Cao, and Y.-Y. Wang, "Intent detection using semantically enriched word embeddings," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, Dec. 2016, pp. 414–419. doi: 10.1109/SLT.2016.7846297.

[10] M. Maryamah, A. Z. Arifin, R. Sarno, and R. W. Sholikah, "Enhanced Topic Modelling using Dictionary For Questions and Answers Problem," *in 2019 12th International Conference on Information & Communication Technology and System (ICTS)*, Surabaya, Indonesia, Jul. 2019, pp. 219–223. doi: 10.1109/ICTS.2019.8850986.

[11] H. Jelodar et al., "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimed. Tools Appl.*, vol. 78, no. 11, pp. 15169–15211, Jun. 2019, doi: 10.1007/s11042-018-6894-4.

[12] A. P. Sam, B. Singh, and A. S. Das, "A Robust Methodology for Building an Artificial Intelligent (AI) Virtual Assistant for Payment Processing," in *2019 IEEE Technology & Engineering Management Conference (TEMSCON)*, Atlanta, GA, USA, Jun. 2019, pp. 1–6. doi: 10.1109/TEMSCON.2019.8813584.

[13] T.-E. Lin, H. Xu, and H. Zhang, "Discovering New Intents via Constrained Deep Adaptive Clustering with Cluster Refinement," *ArXiv191108891 Cs*, Nov. 2019, Accessed: Jul. 06, 2021. [Online]. Available: http://arxiv.org/abs/1911.08891

[14] D. M. Blei, "Latent Dirichlet Allocation," p. 30.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *ArXiv13013781 Cs*, Sep. 2013, Accessed: Jul. 06, 2021. [Online]. Available: http://arxiv.org/abs/1301.3781

[16] Putra Pandu Adikara. 2012. Kamus Kata Dasar dan Stopword List Bahasa Indonesia. http://hikaruyuuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-indonesia diakses pada 11 Nopember 2020

 doi: https://doi.org/10.32503/jtecs.v2i1.1820