TUGAS AKHIR - KS141501

ANALISIS TOPIK PADA SUARA KONSUMEN PT ANGKASA PURA 1 (PERSERO) CABANG JUANDA DENGAN MENGGUNAKAN PEMODELAN GAUSSIAN LATENT DIRICHLET ALLOCATION (GLDA)

TOPIC ANALYSIS ON CUSTOMER VOICE PT ANGKASA PURA 1 (PERSERO) JUANDA USING GAUSSIAN LATENT DIRICHLET ALLOCATION (GLDA) MODELS

BAIQ ZUYYINA HILYATUR ROZALIYA
NRP 0521141 000 7002

Dosen Pembimbing:
Renny Pradina Kusumawardani, S.T., M.T., SCJP

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

Halaman Sengaja Dikosongkan

TUGAS AKHIR - KS141501

ANALISIS TOPIK PADA SUARA KONSUMEN PT ANGKASA PURA 1 (PERSERO) CABANG JUANDA DENGAN MENGGUNAKAN PEMODELAN GAUSSIAN LATENT DIRICHLET ALLOCATION (GLDA)

BAIQ ZUYYINA HILYATUR ROZALIYA
NRP 0521444 000 7002

Dosen Pembimbing:
Renny Pradina Kusumawardani, S.T., M.T., SCJP

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komuniasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

Halaman Sengaja Dikosongkan

FINAL PROJECT - KS 141501

TOPIC ANALYSIS ON CUSTOMER VOICE PT ANGKASA PURA 1 (PERSERO) JUANDA USING GAUSSIAN LATENT DIRICHLET ALLOCATION (GLDA) MODELS

BAIQ ZUYYINA HILYATUR ROZALIYA
NRP 0521144 000 7002

Supervisor:
Renny Pradina Kusumawardani, S.T., M.T., SCJP

DEPARTMENT OF INFORMATION SYSTEMS
Faculty of Information Technology and Communication
Institut Teknologi Sepuluh Nopember
Surabaya 2018

Halaman Sengaja Dikosongkan

LEMBAR PENGESAHAN

ANALISIS TOPIK PADA SUARA KONSUMEN PT ANGKASA PURA 1 (PERSERO) CABANG JUANDA DENGAN MENGGUNAKAN PEMODELAN GAUSSIAN LATENT DIRICHLET ALLOCATION (GLDA)

TUGAS AKHIR
Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

BAIQ ZUYYINA HILYATUR ROZALIYA
NRP. 05211440007002

Surabaya, Juli 2018

KEPALA
DEPARTEMEN SISTEM INFORMASI

Dr. Ir. Aris Tjahyanto. M.Kom
NIP. 19650310 199102 1 001

Halaman Sengaja Dikosongkan

LEMBAR PERSETUJUAN

ANALISIS TOPIK PADA SUARA KONSUMEN PT ANGKASA PURA 1 (PERSERO) CABANG JUANDA DENGAN MENGGUNAKAN PEMODELAN GAUSSIAN LATENT DIRICHLET ALLOCATION (GLDA)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Oleh :

BAIQ ZUYYINA HILYATUR ROZALIYA
NRP. 05211440007002

Disetujui Tim Penguji : Tanggal Ujian : 9 Juli 2018
Periode Wisuda : September 2018

Renny Pradina K., S.T, M.T., SCJP **(Pembimbing I)**

Nur Aini R., S.Kom., M.Sc.Eng., Ph.D **(Penguji I)**

Radityo Prasetianto W., S.Kom., M.Kom. **(Penguji II)**

Halaman Sengaja Dikosongkan

ANALISIS TOPIK PADA SUARA KONSUMEN PT ANGKASA PURA 1 (PERSERO) CABANG JUANDA DENGAN MENGGUNAKAN PEMODELAN GAUSSIAN LATENT DIRICHLET ALLOCATION (GLDA)

Nama Mahasiswa	**:**	**Baiq Zuyyina H Rozaliya**
NRP	**:**	**0521144 000 7002**
Departemen	**:**	**Sistem Informasi**
Pembimbing 1	**:**	**Renny Pradina K., S.T., M.T., SCJP**

ABSTRAK

PT. Angkasa Pura 1 (Persero) Cabang Juanda merupakan perusahaan yang bergerak di bidang penyediaan layanan penyelenggaraan penerbangan yang bertujuan untuk mengutamakan keselamatan, keamanan dan kenyamanan pelanggan. Untuk mewujudkan tujuan tersebut, Angkasa Pura Juanda mengimplementasikan website suara konsumen untuk menampung semua kritik dan saran yang diberikan pelanggan atas pelayanan yang disediakan. Selama pengimplementasian website suara konsumen, cukup banyak masukan yang didapatkan dari pelanggan. Namun, sejauh ini pihak perusahaan belum melakukan analisis lebih lanjut pada data masukan pelanggan untuk mengetahui trend kritik maupun saran yang diberikan. Maka dari itu, penelitian ini melakukan analisis topic modelling pada data kritikan maupun saran dari pelanggan untuk mengetahui topik-topik apa saja yang disampaikan oleh pelanggan serta trend dari topik tersebut, sehingga dapat mempermudah perusahaan dalam menentukan strategi yang dapat dilakukan untuk meningkatkan kualitas layanan dan kepuasan pelanggan.

Analisis topic yang dilakukan menggunakan pemodelan Gaussian Latent Dirichlet Allocation (Gaussian LDA). Topik-

topik yang dihasilkan dari pemodelan Gaussian LDA selanjutnya dianalisis dengan uji Perplexity dan Pointwise Mutual Information (PMI) untuk menganalisis kuantitas dan dibandingkan dengan hasil pemodelan LDA untuk menganalisis kualitas dari topik-topik yang dihasilkan oleh model.

Berdasarkan hasil eksperimen pemodelan topik yang dilakukan, dapat disimpulkan bahwa skenario data frasa batasan 1 dengan stemming pada metode pemodelan Gaussian LDA menghasilkan model terbaik. Dari hasil pengujian didapatkan bahwa model pada skenario dengan jumlah topik sebanyak 13 dapat menghasilkkan nilai PMI sebesar 3.473 dan dari analisis hasil, model ini dapat melakukan pemodelan dengan akurasi sebesar 73% dan tingkat error sebesar 27%.

Kata Kunci: Topic Modelling, Gaussian Latent Dirichlet Allocation, Suara Konsumen Juanda, Pointwise Mutual Information.

TOPIC ANALYSIS ON CUSTOMER VOICE PT ANGKASA PURA 1 (PERSERO) JUANDA USING GAUSSIAN LATENT DIRICHLET ALLOCATION MODELS

Nama Mahasiswa	**:**	**Baiq Zuyyina H Rozaliya**
NRP	**:**	**0521144 000 7002**
Departemen	**:**	**Sistem Informasi**
Pembimbing 1	**:**	**Renny Pradina K., S.T., M.T., SCJP**

ABSTRACT

PT. Angkasa Pura 1 (Persero) Juanda is a company that providing services to prioritize safety and comfort. To realize that goal, Angkasa Pura Juanda implements a consumer voice website to accommodate all the criticisms and suggestions provided by customers for the services. During the implementation of the consumer voice website, a considerable amount of input is gained from customers. However, so far the company has not done further analysis on the data that can be used to know the trend of criticism and advice given by customers. Therefore, this research does topic modeling on customer criticism and information to find out what topics are delivered by customers and the trends of the topic, enabling you to know the strategies and performance that can be done to improve service quality and customer satisfaction.

Topic analysis was performed using Gaussian Latent Dirichlet Allocation (Gaussian LDA) modeling. The topics resulting from the LDA Gaussian modeling are then analyzed by the Perplexity and Pointwise Mutual Information (PMI) tests to analyze and compare with LDA modeling results to analyze the quality of the topics generated by the model.

Based on the experimental results, it can be concluded that skenario data with limits 1 using stemming on modeling Gaussian LDA method gives the best model. From the test results obtained that the model in the scenario with the number of topics as much as 13 can generate the value of PMI of 3.473 and from the analysis result, this model can do modeling with an accuracy 73% and error rate 27%.

Keyword: Topic Modelling, Gaussian Latent Dirichlet Allocation, Customer Voice Juanda, Pointwise Mutual Information.

KATA PENGANTAR

Puji dan syukur penulis tuturkan ke hadirat Allah SWT, Tuhan Semesta Alam yang telah memberikan kekuatan dan hidayah-Nya kepada penulis sehingga penulis mendapatkan kelancaran menyelesaikan tugas akhir yang merupakan salah satu syarat kelulusan pada Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materiil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang sudah meluangkan waktu, tenaga dan pikirannya. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sebanyak-banyaknya kepada:

1. Allah SWT, yang telah memberikan kesehatan, kemudahan, kelancaran dan kesempatan untuk penulis sehingga dapat menyelesaikan tugas akhir ini.
2. Bapak Lalu Sungkul dan Ibu Neyim selaku kedua orang tua serta Baiq Annisa Mulya Kartini selaku adik kandung dari penulis yang tiada henti memberikan dukungan dan semangat secara lahir dan batin.
3. Keluarga besar penulis di Desa Rembitan dan Sengkol yang tiada henti mendukung dan mendoakan penulis selama masa perkuliahan.
4. Bapak Dr. Ir. Aris Tjahyanto, M.Kom selaku Ketua Departemen Sistem Informasi ITS yang telah menyediakan fasilitas terbaik untuk kebutuhan penelitian Mahasiswa.
5. Bapak Agus Zainal Arifin, S.Kom, M.Kom., dan Bapak Darmaji, S.T., M.Kom., selaku Pembina CSSMoRA ITS yang telah membantu penulis selama masa perkuliahan baik lahir maupun batin.

6. Kementrian Agama Republik Indonesia yang telah memberikan dukungan materiil bagi penulis selama masa perkuliahan.
7. Ibu Renny Pradina Kusumawardani., S.T., M.T., SCJP, selaku dosen pembimbing dan sebagai narasumber yang senantiasa meluangkan waktu dan tenaga, memberikan ilmu dan petunjuk yang sangat berarti bagi penulis, serta memotivasi untuk kelancaran pengerjaan tugas akhir.
8. Bapak Tony Dwi Susanto, S.T, M.T, Ph.D., selaku dosen wali yang selalu membantu penulis dalam masa perkuliahan sejak mahasiswa baru hingga tugas akhir.
9. Ibu Nur Aini R., S.Kom., M.Sc.Eng., Ph.D., dan Ibu Irmasari Hafidz, S.Kom., M.Sc., selaku dosen penguji yang telah memberikan saran dan kritik untuk perbaikan tugas akhir.
10. Seluruh dosen Departemen Sistem Informasi ITS yang telah memberikan ilmu yang bermanfaat kepada penulis.
11. Diana Musabbihah, Isnaini Nur R, Umdah Ardillah dan Zuli Maulidati teman- teman kontrakan manja yang telah mendukung dan menemani penulis selama pengerjaan tugas akhir ini.
12. Mbak Nani Latifatun Nada dan Mbak Novi Azizah Pahlawati (Almh) sebagai kakak pendamping (KP) yang telah membimbing, menasehati dan mendukung penulis selama masa perkuliahan.
13. Nelly Safitri, Dini Amalina, Dzakiroh, Dinah Istiqomah dan Mira Ardiningsih yang telah mendukung dan menyemangati penulis dalam mengerjakan tugas akhir.
14. Sahabat-sahabat FATCAS, Alumni PPKh KMMI Putri 2014, yang tidak bisa disebutkan namanya satu per satu yang selalu mendukung, menyemangati dan mendoakan penulis selama masa perkuliahan dan pengerjaan tugas akhir.
15. Mbak Oryza dan Anugrah D Putra yang telah memberikan pencerahan serta praktek dan teori terkait dengan materi pengerjaan tugas akhir.

16. Alden, Adrian, Arif, Dewangga, Hans, Pras, Putra yang telah mendukung dan berjuang bersama penulis dalam pengerjaan tugas akhir.
17. Teman teman di *Teman Masa Gitu,* Nadya, Egin, Zuli, Noptrina, Devita, Nani, Erma, Rima selaku teman yang menemani penulis selama masa perkuliahan
18. Sahabat-sahabat angkatan 2014 CSSMoRA ITS yang selalu mendukung dan memberikan semangat positif, canda tawa bagi penulis selama menjalani masa perkuliahan.
19. Keluagra besar CSSMoRA ITS yang selalu memberikan kenyamanan dan dukungan untuk penulis selama masa perkuliahan.
20. Teman-teman OSIRIS, Sistem Informasi 2014, yang telah menemani dan berjuang bersama penulis selama masa perkuliahan.
21. Mbak Poppy, Mbak Ira, Mas Kevin, Mas Joshua, Mr. Yoon selaku teman les penulis yang selalu memberikan semangat positif bagi penulis.
22. Kak Nana, Kak Nik, Kak Iqbal, Kak Eka, Kak Dure, Keke, Nesa, Kak iin, Puji, Tari, Sinta, Vera selaku kerabat penulis yang selalu mendukung dan mendoakan penulis selama pengerjaan tugas akhir.

Penyusunan laporan ini masih jauh dari kata sempurna sehingga penulis menerima kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, Juni 2018
Penulis,

Baiq Zuyyina Hilyatur
Rozaliya

Halaman Sengaja Dikosongkan

DAFTAR ISI

Halaman Sengaja Dikosongkan

DAFTAR GAMBAR

Halaman Sengaja Dikosongkan

DAFTAR KODE

DAFTAR TABEL

Halaman Sengaja Dikosongkan

BAB I
PENDAHULUAN

Pada bab ini akan diuraikan tentang pengidentifikasian masalah dalam penelitian meliputi latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat dan relevansi dari tugas akhir sehingga gambaran umum dan pemecahan masalah dapat dipahami.

1.1 Latar Belakang Masalah

PT. Angkasa Pura (Persero) adalah perusahaan Badan Usaha Milik Negara di bawah Departemen Perhubungan yang memberikan pelayanan lalu lintas udara dan bisnis bandar udara di Indonesia yang menitikberatkan pada kawasan Indonesia bagian tengah dan kawasan Indonesia bagian timur. PT. Angkasa Pura yang kemudian dikenal dengan Angkasa Pura Airports berusaha mewujudkan perusahaan profesional berkelas dunia. PT. Angkasa Pura bertekad memberikan keselamatan, kenyamanan, dan keamanan berstandar internasional dengan menyediakan pelayanan yang terbaik [1].

Sebagai perusahaan penyedia layanan dalam penyelenggaraan penerbangan yang mengutamakan kenyamanan, keselamatan dan keamanan pelanggan, Angkasa Pura khususnya cabang Juanda mencoba untuk menampung kritik dan saran dari para pelanggan melalui website khusus suara konsumen juanda. Website suara konsumen juanda yang kemudian dikenal dengan suarajuanda.com dimanfaatkan oleh PT. Angkasa Pura Cabang Juanda untuk meningkatkan kualitas layanan dan kepuasan pelanggan [2]. Website ini memiliki fitur Suara Konsumen, dimana fitur inilah yang berguna untuk menampung saran dan kritik untuk perbaikan layanan yang diberikan perusahaan. Selain dapat memberikan kritik dan saran melalui website, pelanggan juga dapat memberikan kritik dan saran mereka

dengan menghubungi Angkasa Pura melalui e-mail, twitter, telepon dan facebook perusahaan.

Pengimplementasian website suara konsumen untuk Bandara Juanda mulai diterapkan pada tahun 2015 sampai sekarang di bawah Departemen ICT. Selama masa pengimplementasian, jumlah data masukan dari pelanggan yang dapat diarsipkan yaitu sebanyak 733 data pada 17 Januari 2018. Semua kritik dan saran yang diberikan oleh pelanggan melalui website suara konsumen Bandara Juanda sejauh ini ditindaklanjuti dengan cara membalas semua pesan yang masuk dari pelanggan kemudian semua pesan yang masuk ditampilkan kembali pada fitur tinjauan dalam bentuk tabel dengan tujuan agar pelanggan lain dapat membaca pesan-pesan tersebut. Namun, sejauh ini website suarajuanda.com belum memiliki fitur yang dapat menampilkan topik-topik apa saja yang disampaikan oleh pelanggan sehingga perusahaan tidak dapat mengetahui trend kritikan maupun saran pelanggan. Oleh karena itu, analisis lebih lanjut oleh perusahaan pada data masukan pelanggan perlu dilakukan. Dengan mengetahui trend tersebut dapat membantu perusahaan dalam pengambilan keputusan untuk menanggapi keluhan ataupun saran dari pelanggan sehingga tujuan utama pengimplementasian website dapat terpenuhi dengan baik.

Berdasarkan uraian di atas, penelitian ini bertujuan menawarkan solusi berupa topic modelling untuk melakukan analisis pada data suara konsumen PT Angkasa Pura Cabang Juanda. Topic modelling merupakan salah satu metode dalam text mining yang dapat digunakan untuk mengetahui topik apa saja yang disampaikan oleh pelanggan [3] sehingga dapat mempermudah perusahaan dalam menentukan strategi yang dapat dilakukan untuk meningkatkan kualitas layanan dan kepuasan pelanggan. Dalam penelitian ini, topic modelling yang dilakukan akan menggunakan pemodelan Gaussian Latent Dirichlet Allocation yang kemudian dikenal dengan Gaussian LDA. Gaussian LDA merupakan pemodelan yang digunakan untuk mengetahui suatu pola tertentu dalam dokumen yang merepresentasikan struktur tematik dari kumpulan dokumen

[4]. Dengan Gaussian LDA, akan dilakukan pencarian probabilitas suatu topik dalam suatu dokumen dan penentuan probabilitas suatu topik akan dihasilkan dari vector kata atau distribusi normal multivariat dalam embedding space [4]. Topik yang dihasilkan dari analisis dengan Gaussian LDA kemudian akan diuji dengan menggunakan pengujian koherensi untuk menguji kemudahan topik tersebut dipahami atau diinterpretasikan oleh manusia [5].

Diharapkan dengan hasil dari analisis topic modelling pada website suarajuanda.com dapat membantu perusahaan dalam mencapai tujuan pengimplementasian website.

1.2 Perumusan Masalah

Berikut rumusan masalah yang akan difokuskan dan diselesaikan dalam tugas akhir ini berdasarkan pada pemaparan latar belakang di atas

1. Bagaimana melakukan *topic modeling* untuk mengetahui topik yang disampaikan oleh pelanggan dari suara konsumen Juanda dengan menggunakan LDA dan *Gaussian* LDA?
2. Bagaimana cara mengukur akurasi dari topik yang dihasilkan dari *topic modeling*?
3. Bagaimana cara membandingkan model yang dihasilkan dengan pemodelan LDA dan *Gaussian* LDA

1.3 Batasan Masalah

Berikut batasan masalah dalam pengerjaan tugas akhir ini berdasarkan pada penguraian rumusan masalah di atas

1. Studi kasus yang diangkat hanya berfokus pada PT Angkasa Pura 1 (Persero) Cabang Juanda
2. Data yang digunakan adalah data yang terdapat pada website suara konsumen juanda dari tahun 2015 sampai 2018

3. Jenis data yang akan dianalisis menggunakan model Gausssian LDA yaitu berupa data teks komentar pelanggan dalam Bahasa Indoensia.

1.4 Tujuan Penelitian

Berikut tujuan dari tugas akhir ini berdasarkan pada pemaparan latar belakang dan rumusan masalah di atas

1. Melakukan analisis topic modelling pada suara konsumen Juanda untuk mengetahui topik-topik apa saja yang disampaikan oleh pelanggan dengan menggunakan model Gaussian LDA.
2. Melakukan validasi ouput berupa topik yang dihasilkan dari topic modelling untuk mengukur kinerja dari model Gaussian LDA..

1.5 Manfaat Penelitian

Berikut manfaat yang diharapkan akan diperoleh dari hasil pengerjaan tugas akhir ini

Bagi Penulis:

1. Memahami topic modelling dengan menggunakan model Gaussian LDA
2. Memahami cara kerja Gaussian LDA untuk melakukan *topic modeling*
3. Mengetahui bagaimana cara mengukur atau melakukan validasi hasil dari model untuk mengukur kinerja dari model Gaussian LDA.

1.6 Relevansi

Tugas akhir ini berhubungan dengan penerapan mata kuliah Sistem Cerdas, Sistem Pendukung Keputusan, Kecerdasan Bisnis dan Penggalian Data dan Analitika Bisnis yang merupakan mata kuliah bidang keilmuan Laboratorium Akuisisi Data dan Diseminasi Informasi.

BAB II
TINJAUAN PUSTAKA

Bab ini akan membahas penelitian sebelumnya yang berhubungan dengan tugas akhir dan teori - teori yang berkaitan dengan permasalahan tugas akhir ini.

2.1 Studi Sebelumnya

Tabel 1 menampilkan daftar penelitian sebelumnya yang mendasari tugas akhir ini.

Tabel 2.1 Peneilian Sebelumnya

Judul Penelitian	Metode	Penulis	Hasil yang Didapatkan
A Biterm Topic Model for Short Texts	*Biterm Model*	Xiaohui Yan, Jiafeng Guo, Yanyan Lan dan Xueqi Cheng	Penelitian ini berfokus pada analisis topik dalam dokumen dengan ukuran kecil atau pendek seperti data *twitter* dengan jumlah kata dalam dokumen yang tidak banyak. Analisis topik dilakukan dengan memanfaatkan pola *co-occurance* kata dalam data dalam dokumen. Hal ini dilakukan untuk memperkaya dictionary kata pembentuk topik

Judul Penelitian	Metode	Penulis	Hasil yang Didapatkan
			dalam dokumen yang pendek. Dari hasil analisis yang dilakukan didapatkan nilai rata-rata *coherence score* adalah -990.2 ± 3.8 untuk 20 kata dalam topik dan nilai dari *H score* adalah 0.474 ± 0.005 [6].
Latent Dirichlet Allocation	*Latent Dirichlet Allocation*	David M. Blei, Andrew Y. Ng dan Michael I. Jordan	Pada penelitian ini dilakukan analisis topik pada data berupa teks. Penelitian ini menyajikan sebuah *framework* baru untuk melakukan pemodelan struktur dari pada topik dalam dokumen dan data diskrit lainnya [7].
Gaussian LDA for Topic Models with Word Embedding	*Gaussian LDA*	Rajarshi Das, Manzil Zaheer dan Chris Dyer	Pada penelitian ini dilakukan analisis pada topik dalam dokumen dengan teknik analisis *topic modelling* yang baru dimana dalam penelitian ini

Judul Penelitian	Metode	Penulis	Hasil yang Didapatkan
			dokumen dianggap sebagai kumpulan dari *word embedding* dan topik sebagai distribusi normal multivariatif dalam *embedding space.* Dari hasil analisis dengan menggunakan 15 topik didapatkan hasil *Point Mutual Information (PMI)* dengan rata-rata 275% yang menandakan bahwa topik memiliki koherensi yang baik [4].
Optimizing Semantic Coherence in Topic Models	*Latent Dirichlet Allocation*	David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders dan Andrew McCallum	Pada penelitian ini dilakukan analisis untuk mengetahui bagaimana suatu topik dikatakan cacat. Dari penelitian ini dapatkan hasil analisis bahwa ukuran dan probabilitas dari suatu topik memiliki hubungan yang erat, semakin

Judul Penelitian	Metode	Penulis	Hasil yang Didapatkan
			meningkat jumlah topik, maka jumlah kata yang ada dalam topik menurun, hal ini menunjukkan kurangnya kualitas dari hasil analisis [5].

2.2 Dasar Teori

2.2.1 *Topic Modeling*

Topic modelling merupakan metode yang digunakan untuk menemukan pola topik yang tersembunyi dalam kumpulan dokumen yang merepresentasikan informasi di dalamnya [8]. Ide dasar dari adanya *topic modelling* adalah suatu dokumen merupkan kumpulan dari topik-topik, dimana topik yang terdapat di dalam dokumen merupakan suatu distribusi probabilitas dari kata. *Topic modelling* menyajikan metode untuk mengorganisir, memahami dan menyimpulkan informasi dari jumlah data teks yang besar. *Topic modelling* dapat membantu dalam [3]:

1. Menemukan pola topik yang tersembunyi yang ada di dalam kumpulan dokumen
2. Menganotasi dokumen berdasarkan topik yang ditemukan
3. Menggunakan anotasi ini untuk mengorganisir, mencari dan menyimpulkan teks

Algoritma dari *topic modelling* tidak melakukan anotasi atau pelabelan pada dokumen terlebih dahulu, karena topik dianggap terdiri dari kelompok kata yang secara frekuensi terjadi atau muncul bersamaan yang dianalisis langsung dari dokumen teks Tanpa Frasa. Dengan menggunakan petunjuk kontekstual, *topic*

modelling dapat menghubungkan kata-kata dengan arti atau makna yang mirip dan membedakan antara penggunaan kata yang memiliki banyak makna [9].

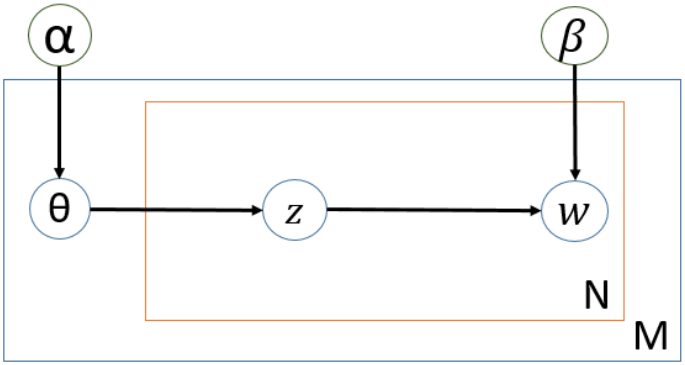
Menurut Blei *topic modelling* terdiri dari 3 jenis entitas yaitu kata, dokumen dan topik [10]. Dokumen merupakan kumpulan dari N kata-kata, dimana kumpulan dari dokumen dalam penelitian yang dilakukan Blei disebut sebagai corpus. Topik merupakan susunan yang didapatkan dari distribusi probabilitas kata yang ada dalam dokumen. Sedangkan kata merupakan unit terkecil dalam dokumen.

2.2.2 *Latent Dirichlet Allocation (LDA)*

Latent Dirichlet Allocation (LDA) merupakan teknik bayesan dalam topic modelling yang sering digunakan dalam melakukanan analisis terhadap struktur dan menentukan probabilitas kemunculan suatu topik yang terdapat dalam dokumen dari corpora [11]. Dalam LDA dokumen diterjemahkan sebagai kumpulan dari banyak topik dan topik sebagai kumpulan dari distribusi kata dalam dokumen [12]. LDA menghasilkan daftar distribusi kelompok kata yang diberikan bobot sesuai probabilitas kemunculan kata tersebut dalam dokumen, dari kelompok distribusi kata ini kemudian dibentuklah topik-topik yang menyusun suatu dokumen dalam corpora [5]. Blei menjelaskan dalam papernya, dokumen dimodelkan melalui variable dirichlet random yang tersembunyi yang memberikan spesifikasi terkait dengan probabilitas distribusi suatu topik. Hasil dar dirichlet digunakan untuk menentukan kata-kata yang terdapat dalam dokumen untuk penentuan topik yang berbeda [7]. Pada gambar 2.1 ditunjukkan visualisasi konsep LDA dari hasil penelitian Blei.

Pada gambar di atas divisualisasikan bahwa terdapat 3 tingkat Bayesian. Tingkatan pertama yaitu pada kotak M, menjelaskan distribusi atau kumpulan dari topik yang terdapat dalam kumpulan dokumen atau corpus, dimana Bayesian ini disimbolkan dengan parameter α, semakin besar nilai dari α dalam dokumen maka semakin besar pula kemungkinana

terdapat banyak campuran topik dalam dokumen tersebut. Dalam kotak M juga terdapat varibael Θ yang merepresentasikan distribusi topik dalam dokumen tertentu, dimana semakin tinggu nilai variable Θ maka semakin banyak topik yang terdapat dalam dokumen tersebut. Kemudian pada tingkatan ke dua terdapat dalam kotak N, variabel z merepresentasikan topik yang terdapat dalam dokumen tertentu, kemudian dihubungkan dengan tingkatan Bayesian ke tiga dengan p(w|z) yang merepresentasikan probabilitas dari suatu kata terdapat dalam suatu topik, distribusi atau kumpulan kata yang terdapat di dalam topik z dalam penelitian oleh Blei direpresentasikan dengan parameter β, dimana semakin besar nilai β maka semakin besar pula probabilitas kata-kata yang terdapat dalam topik.



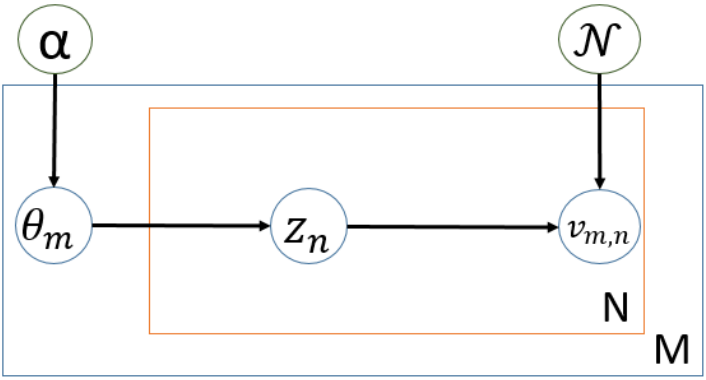
Gambar 2.1 Visualisasi Konsep LDA oleh Blei

Kemudian pada tingkat Bayesian ke tiga, w merepresentasiakan kumpulan kata-kata yang terdapat di dalam dokumen. Secara keseluruhan, dilihat dari visualisasi di atas, ide dasar LDA adalah untuk menentukan probabilitas kemunculan dari setiap kata yang terdapat di dalam corpus, atau kumpulan dokumen.

2.2.3 *Gaussian Latent Dirichlet Allocation (GLDA)*

Pemodelan dengan menggunakan LDA merupakan pemodelan dengan kosa kata yang tetap sehingga tidak dapat mengatasi *out of vocabulary* (OOV). Untuk mengatasi masalah ini, Rajarshi Das dkk membuat pemodelan yang merupakan variasi lain dari LDA yaitu *Gaussian LDA.* Ide dasar dari *Gaussian LDA* adalah dengan mengasumsikan dokumen menjadi kumpulan dari kata-kata yang saling terhubung *(word embedding)* dan topik yang ada di dalam dokumen menjadi *multivariate Gaussian distribution* di dalam *embedding space* [4]. Proses generalisasi dengan *Gaussian LDA* diilustrasikan sebagai berikut

a. Untuk setiap topik *k* dari 1 sampai *K*
   i. Gambar kovarian topik $\sum_k \sim \mathrm{w}^{-1}(\psi, v)$
   ii. Gamabr *mean* topik $\mu_k \sim \mathcal{N}(\mu, \frac{1}{k}\sum_k)$
b. Untuk setiap dokumen *m* di dalam corpus M
   i. Gambar distribusi topik $\theta_m \sim Dir\ (\alpha)$
   ii. Untuk setiap kata indeks ke *n* dari 1 sampai N_m
      - Gambar sebuah topik z_n
      - Gambar $v_{m,n} \sim \mathcal{N}(\mu_{z_n}, \sum_{z_n})$



Gambar 2.2 Visualisai Konsep *Gaussian LDA* oleh Rajarshd

Dengan *Gaussian LDA* akan dilakukan eksploitasi terhadap hubungan dari kata yang memiliki kesamaan secara semantik di dalam *embedding space* dan dapat memberikan probabilitas

yang tinggi pada kata yang memiliki kemiripan dengan kata-kata yang ada dalam topik meskipun kata tersebut tidak pernah dilihat sebelumnya. Pada gambar 2.2 akan ditunjukkan visualisasi dari konsep *Gaussian LDA*

2.2.4 Evaluasi Model Berdasarkan Nilai *Perplexity*

Uji perplexity merupakan pengujian yang digunakan untuk menilai sebarapa baik model yang telah dihasilkan dengan pemodelan *Gaussian* LDA. Pengujian ini dilakukan untuk mengetahui seberapa baik model dapat melakukan generalisasi teks dalam dokumen [8]. Suatu model dikatakan baik jika memiliki nilai *perplexity* yang kecil, semakin kecil nilai *perplexity* yang dihasilkan maka semakin bervariasi dan berbeda pula topik yang ditemukan oleh model [7]. Ide dasar dari uji perplexity adalah dengan mengambil n sampel dari N populasi data. Sejumlah n sampel yang diambil kemudian dianalisis apakah sampel tersebut memiliki kesamaan topik dengan topik-topik yang ada dalam N populasi.

Nilai *perplexity* dari sebuah kelompok kata-kata uji didefinisikan sesuai persamaan (1).

$$perplexity(w_d|D_d) = exp\left[-\frac{\ln p(w_d|D_d)}{N_d}\right] \quad (1)$$

2.2.5 Evaluasi Model Berdasarkan *Topic Coherence*

Uji koherensi topik merupakan pengujian untuk mengetahui seberapa mudah topik yang dihasilkan oleh model dapat dipahami atau diinterpretasikan oleh manuasia [13]. Koherensi topik bertujuan untuk mengukur topik berdasarkan tingkat kesamaan semantik dari kata-kata yang menyusun topik tersebut. Hasil yang diharapkan dari pengujian ini adalah mengetahui topik-topik mana saja yang dapat diinterpretasikan secara semantik dan topik yang memiliki keterkaitan secara statistik.

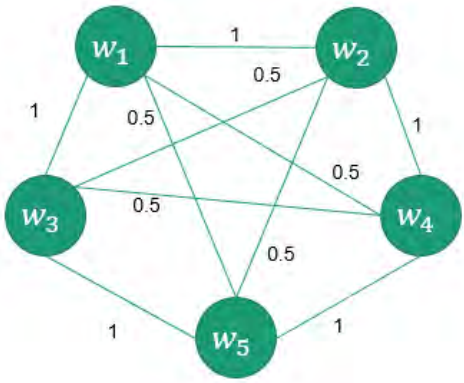
Nilai *topic coherence* dari kelompok kata-kata dalam topik didefinisikan sesuai persamaan (2).

$$Score_{UMass}(w_i, w_j) = log \frac{D(w_i, w_j) + 1}{D(w_i)} \quad (2)$$

2.2.6 Perhitungan Nilai *Pointwise Mutual Information*

Pointwise Mutual Information merupakan pengukuran yang digunakan untuk mengetahui hubungan antar *single event* yang banyak digunakan dalam teori informasi dan statistik. Ide dasar dari penggunaan PMI dalam penlitian ini adalah untuk mengetahui hubungan antar kata yang ada di dalam topik yang akan diketahui dengan menghitung jumlah kemunculan kata serta jumlah *co-occurrence* dari kata-kata yang ada di dalam topik [14]. Berikut rumus yang digunakan untuk menghitung PMI

$$PMI\ (X = x, Y = y) = \log(\frac{p(X = x, Y = y)}{p(X = x)p(Y = y)}) \quad (1)$$



Gambar 2.3 Visualisasi Konsep *Pointwise Mutual Information*

Dimana, $p(X = x)$ merupakan jumlah kemunculan kata x dalam dokumen dan $p(Y = y)$jumlah kemunculan kata y dalam

dokumen serta $p(X = x, Y = y)$ merupakan jumlah kemunculan bersama kata x dan y dalam dokumen.

Dalam penelitian ini, topik akan diurtukan berdasarkan pada PMI score yang dihasilkan dari rata-rata nilai PMI yang dimiliki oleh 10 kata teratas dari topik [14]. PMI Score akan dihitung sebegai berikut:

$$PMI\ Score = mean(PMI(x_i, y_j), ij\ \epsilon\ 1 \dots 10, i \neq j) \quad (2)$$

Pada gambar 2.3 akan diilustrasika PMI antar kata dalam topik.

2.2.7 PT Angkasa Pura 1 (Persero) Cabang Juanda

2.2.7.1 Profil Perusahaan

PT. Angkasa Pura (Persero) adalah perusahaan Badan Usaha Milik Negara di bawah Departemen Perhubungan yang memberikan pelayanan lalu lintas udara dan bisnis bandar udara di Indonesia yang menitikberatkan pada kawasan Indonesia bagian tengah dan kawasan Indonesia bagian timur. PT. Angkasa Pura yang kemudian dikenal dengan Angkasa Pura *Airports* berusaha mewujudkan perusahaan profesional berkelas dunia. PT. Angkasa Pura bertekad memberikan keselamatan, kenyamanan, dan keamanan berstandar Internasional dengan menyediakan pelayanan yang terbaik.

Angkasa Pura Airports didirikan atas insisatif presiden Soekarno kepada Menteri Perhubungan dan Menteri Pekerjaan Umum setelah melakukan kunjungan kenegaraan ke Amerika Serikat dan bertemu dengan John F. Kennedy, beliau menginginkan agar lapangan terbang di Indonesia dapat setara dengan lapangan terbang di negara maju. Sejak saat itu Angkasa Pura sebagai pelopor pengusahaan kebandarudaraan secara komersial di Indonesia dan terbitlah Peraturan Pemerintah (PP) Nonor 33 Tahun 1962 tentang Pendirian Perusahaan Negara (PN) Angkasa Pura Kemayoran Didirikan pada tanggal 20 Februari 1962 berdasarkan Peraturan Pemerintah nomor 33 tahun 1962 dengan nama Perusahaan Negara Angkasa Pura

Kemayoran yang mempunyai tugas pokok sebagai pengelola dan pengusahaan bandar udara Internasional Kemayoran Jakarta, dan pada tanggal tersebut ditetapkan sebagai hari jadi Angkasa Pura Airport.

Pemerintah mengubah nama Perusahaan Negara Angkasa Pura Kemayoran menjadi Perusahaan Negara Angkasa Pura pada tanggal 17 Mei 1965 berdasarkan Peraturan Pemerintah nomor 21 tahun 1965 dengan maksud untuk lebih membuka kemungkinan mengelola bandar udara lain di wilayah Indonesia. Berdasarkan PP Nonor 5 Tahun 1992 bentuk Perusahan Umum Angkasa Pura 1 diubah menjadi Perusahan Angkasa Pura 1 (Persero).

2.2.7.2 Visi dan Misi Perusahaan

Visi PT. Angkasa Pura 1 (Persero)

Visi PT. Angkasa Pura 1 (Persero) adalah: “Menjadi salah satu dari sepuluh perusahaan pengelola bandar udara terbaik di Asia [1].”

Misi PT. Angkasa Pura 1 (Persero)

Adapula beberapa misi yang diangkat oleh PT. Angkasa Pura 1 (Persero) yaitu sebagai berikut:

1. Meningkatkan nilai pemangku kepentingan.
2. Menjadi mitra pemerintah dan pendorong pertumbuhan ekonomi.
3. Mengusahakan jasa kebandarudaraan melalui pelayanan prima yang memenuhi standar keamanan, keselamatan, dan kenyamanan.
4. Meningkatkan daya saing perusahaan melalui kreativitas dan inovasi.
5. Memberikan kontribusi positif terhadap lingkungan hidup.

2.2.7.3 Suara Konsumen Juanda

Website suara konsumen junada, yang dapat diakses melalui suarajuanda.com, merupakan salah satu layanan yang disediakan oleh PT angkasa Pura 1 (Persero) Cabang Juanda dengan tujuan meningkatkan kepuasan layanan terhadap *customer* agar dapat memberikan nilali tambah kepada pelanggan. Website ini berguna untuk menampung kritik maupun saran untuk perbaikan layanan penerbangan yang disediakan oleh perusahaan [2]. Website Suara Konsumen Juanda di kembangkan oleh *Incheon Internasional Airport Consulting Team*, sebuah perusahaan konsultan penerbangan dalam semua sektor dalam memberikan layanan kualitas terbaik untuk penumpang internasional dan jumlah kargo pada setiap waktu yang merupakan perusahaan di Korea Selatan dan memiliki jaringan aviasi di Asia Selatan.

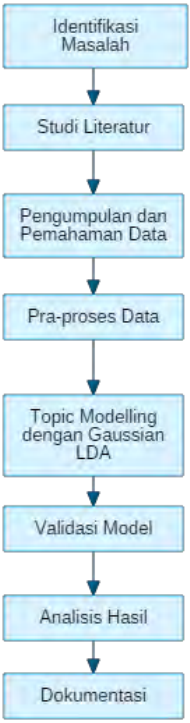
Website tersebut memiliki dua sub menu diantaranya *input* dan *view*. Fitur *input* adalah laman dimana penumpang dapat memberikan masukan saran dan kritik dengan menginputkan terlebih dahulu informasi pelanggan, informasi kejadian/*Event* dan *form* masukan saran dan kritik dari pelanggan dengan memasukkan judul keluhan. Kemudian fitur yang kedua adalah *view*, fitur tersebut diperuntukkan bagi pelanggan dan divisi *Customer Service Relation Section* dalam menanggapi dan memperbaiki layanan yang dikeluhkan pelanggan. Fitur tersebut menampilkan semua masukan dari fitur *input* seperti nama pelanggan, saran dan kritik, tanggal pengiriman dan ditampilkan dengan list berdasarkan tanggal submit dari keluhan tersebut serta keterangan "*finish*" jika keluhan tersebut telah selesai ditangani oleh divisi yang bersangkutan.

BAB III
METODOLOGI

Bab ini menjelaskan tentang metodologi yang akan digunakan dalam penyusunan tugas akhir. Metodologi akan digunakan sebagai panduan dalam penyusunan tugas akhir agar terarah dan sistematis.

3.1 Tahapan Pelaksanaan Tugas Akhir

Alur tahapan pelaksanaan yang dilakukan dalam mengerjakan tugas akhir ini sesuai dengan alur pada gambar 3.1.



Gambar 3.1 Alur Metodologi Pengerjaan Tugas Akhir

3.1.1 Identifikasi Masalah

Pada tahap ini dilakukan pengidentifikasian masalah pada perusahaan yang dijadikan sebagai studi kasus, yaitu PT Angkasa Pura 1 (Persero) Cabang Juanda. Pengidentifikasian masalah mencakup pemahaman proses bisnis dari PT Angkasa Pura 1 (Persero) Cabang Juanda. Pengidentifikasian masalah dilakukan dengan cara observasi pada website suara konsumen junada. Dari hasil observasi didapatkan bahwa belum adanya analisis lebih lanjut pada data yang disampaikan pelanggan melalui website sehingga pengimplemenntasian website untuk menampung suara konsumen sebagai perbaikan untuk perusahaan masih belum optimal. Berdasarkan pada permasalahan yang didapatkan maka dirasa perlu untuk melakukan analisis lebih lanjut untuk mengetahui topik apa saja yang disampaikan pelanggan.

3.1.2 Studi Literatur

Pada tahap ini dilakuakan studi literatur untuk memahami konsep, metode dan teknologi yang terkiat dan sesuai dengan permasalahan serta solusi yang ditawarkan. Studi literature juga dilakukan untuk menggali informasi melalui literatur-literatur penelitian terkait dengan permasalahan dan solusi yang ditawarkan. Dari studi literatur, solusi berupa analisis lebih lanjut untuk mengetahui topik yang disampaikan oleh pelanggan akan mengacu pada penelitian yang telah dilakukan oleh David Rajarshi Das dkk yaitu menggunakan *Gaussian Latent Dirichlet Allocation* (*Gaussian LDA*) [4] dan pengujian pada hasil analisis akan mengacu pada penelitian yang telah dilakukan oleh David Newman dkk [14] mengenai *Poinwise Mutual Infromation* dengan judul *Evaluating Topic Models for Digital Libraries*.

3.1.3 Proses Pengumpulan Data

Pada tahap ini dilakukan pengumpulan dan pemahaman data yang akan dianalisis. Data didapatkan dari arsip kritik dan saran pelanggan melalui website suara konsumen. Data yang didapatkan dalam bentuk file Microsoft Excel dengan format

.xlsx dan merupakan data periodik dari tahun 2015 sampai dengan 2018. Data terdiri dari atrbiut: Nama pelanggan, tanggal, waktu, lokasi, jenis penerbangan, komentar, maskapai, dan kebangsaan pelanggan. Berikut merupakan contoh dari data yang didapatkan dari website suarajuanda.com.

Tabel 3.1 Contoh Data Suara Pelanggan

Nama	Tanggal	Lok	Jenis penerbangan	Komen	Maskapai	Kebangsaan
Adam	04/10/2016	T2	Domestik	Bandara INTERNATIONAL Juanda Surabaya Terminal 2 mati lampu. Gak ada switch ke genset. Gak ada penjelasan dari petugas. Mati lampu sampai 10 menit baru akhirnya genset menyala.	Garuda Indonesia	WNI

Setelah dilakukan konversi *file* dari bentuk *.xlsx* menjadi *.txt*, dilakukan pembersihan data dengan hanya mengambil atribut komentar. Hal ini dilakukan karena atribut yang dianggap relevan untuk digunakan sebagai data permodelan hanya atribut komentar. Teks di luar itu hanya akan membiaskan data yang digunakan dan kurang.

```
tolong informasi jujur penerbangan delay
perbedaan penerbangan terminal terminal ya
berangkat batik air terminal kendaraan diparkir terminal kedatangan garuda terminal
terminal
cs nya tiang bandara hahahaha
kamar mandinya bersih wangi nyaman bagus berkualitas internasional
layanan public peningkatan layanan kenyamanan bandara international juanda layanan
kenyamanan hilangkan -bandar udara international juanda bandara khusus golongan
orang masuk area terminal pemeriksaan
kurir trolli ya bandara bayar enak digratisin aja trolli
kursi ruang kedatangan pengambilan barang min diliat baris aja
mohon informasinya parkir inap sepeda motor terminal alurnya parkir inap sepeda
motor parkir tinggal melapor biaya parkir inap sepeda motor terminal aman parkir
inap sepeda motor terima kasih
jalur drop tolong kasi atap hujan jalur crowded memperlambat penumpang berangkat
mepet nya menjumpai mobil sengaja parkir jalur terima kasih
```

Gambar 3.2 Contoh data *file paper* yang telah dikonversi menjadi *.txt*

Contoh *file .txt* yang telah dibersihkan dapat dilihat pada Gambar 3.2.

3.1.4 Pra-proses Data

Pada tahap ini akan dilakukan langkah-langkah yang diperlukan sebelum analisis data dengan *Gaussian Latent Dirichlet Allocation* (*Gaussian LDA*) agar data menjadi data yang sesuai dan siap diolah. Berikut langkah-langkah yang akan dilakukan dalam tahap ini:

3.1.4.1 *Data Cleaning*

Pada tahap *data cleaning*, dilakukan pembersihan data dari karakter numerik dan simbol-simbol. Selain itu, dilakukan juga penghapusan satu digit karakter. Hal ini dilakukan untuk menghindari tingginya frekuensi kemunculan karakter numerik, simbol, dan satu digit karakter yang luput dari proses pembersihan data untuk mengambil isi abstrak hingga kesimpulan yang telah dilakukan sebelum pra-proses data. Contoh tahapan *data cleaning* dapat dilihat pada Tabel 3.2.

Tabel 3.2 *Data cleaning*

Sebelum *data cleaning*	Setelah *data cleaning*
Bandara INTERNATIONAL Juanda Surabaya Terminal 2 mati lampu. Tidak ada switch ke genset. Tidak ada penjelasan dari petugas. Mati lampu sampai 10 menit baru akhirnya genset menyala	Bandara INTERNATIONAL Juanda Surabaya Terminal mati lampu Tidak ada switch ke genset Tidak ada penjelasan dari petugas Mati lampu sampai menit baru akhirnya genset menyala

3.1.4.2 *Case Folding*

Tabel 3.3 *Case Folding*

Sebelum Case Folding:	Setelah Case Folding:
Bandara INTERNATIONAL Juanda Surabaya Terminal 2 mati lampu. Tidak ada switch ke genset. Tidak ada penjelasan dari petugas. Mati lampu sampai 10	bandara international juanda surabaya terminal mati lampu tidak ada switch ke genset tidak ada penjelasan dari petugas mati lampu sampai

menit baru akhirnya genset menyala	menit baru akhirnya genset menyala

Dalam langkah ini semua data berupa teks akan diubah menjadi huruf kecil. Tujuan dari *case folding* adalah untuk mencegah kata yang sama namun memiliki penulisan huruf capital yang berbeda diproses menjadi dua kata yang berbeda. Berikut merupakan contoh data sebelum dan setelah dilakukan *case folding.*

3.1.4.3 Data *Formalizer*

Dalam tahap ini, data berupa teks akan dinormalisasi dengan tujuan untuk menjadikan semua kata yang ada di dalam data menjadi kata baku sesuai dengan kaidah standar Bahasa Indonesia.

Tabel 3.4 *Formalizer*

Sebelum Normalisasi:	Setelah Normalisasi:
bandara international juanda surabaya terminal 2 mati lampu. gak ada switch ke genset. gak ada penjelasan dari petugas. mati lampu sampai 10 menit baru akhirnya genset menyala.	bandara international juanda surabaya terminal mati lampu tidak ada switch ke genset tidak ada penjelasan dari petugas mati lampu sampai menit baru akhirnya genset menyala

Dalam penelitian ini, normalisasi data akan dilakukan dengan menggunakan *repository* yang disusun oleh Purwarianti [15].

3.1.4.4 *Stemming*

Dalam langkah ini data teks akan diubah menjadi kata dasar dengan menghilangkan semua imbuhan yang ada dalam kata baik awal, akhiran, sisipan serta kombinasi dari awaln dan akhiran [16]. Tujuan dari langkah ini adalah untuk mencegah kata yang sama namun berbeda dalam penulisan dikarenakan adanya imbuhan pada kata yang lain diproses menjadi kata yang berbeda. Dalam tugas akhir ini, *stemming* dilakukan dengan menggunakan library sastrawi yang merupakan stammer untuk

Bahasa Indonesia. Berikut contoh data sebelum dan setelah proses *stemming*.

Tabel 3.5 *Stemming*

Sebelum *Stemming*:	Setelah *Stemming:*
bandara international juanda surabaya terminal mati lampu switch genset penjelasan petugas. mati lampu menit akhirnya genset menyala	bandara international juanda surabaya terminal mati lampu switch genset jelas petugas mati lampu menit akhir genset nyala

3.1.4.5 *Stopwords removal*

Dalam langkah ini dilakukan penghapusan *stopword* yang ada dalam data. *Stopword* merupakan kata umum yang pada umumnya muncul dalam jumlah yang cukup besar dalam suatu kalimat dan dalam pemrosesan data berupa teks dianggap tidak memiliki makna. Berikut merupakan contoh *stopword* berdasarkan kelas kata

Tabel 3.6 Contoh *Stopword*

Kelas Kata	Contoh
Kata Hubung	dan, atau, sehingga, sedangkan, serta, bila, jika, sebagai, sebelum, sesudah, dengan dll
Kata Ganti	aku, saya, ku, kau, kamu, dia, kita, kalian, anda, engkau, ia, beliau, mereka dll
Kata Depan	pada, dari, di, oleh, pada, kepada, ke dll

Menghapus *stopword* dari data teks bertujuan untuk mencegah kemunginan kata-kata dalam *stopword* memiliki frekuensi kemunculan dalam topik yang lebih tinggi dibandingkan kata yang memiliki makna sehingga menyebabkan topik sulit untuk diinterpretasi dengan baik. Pada penelitian ini daftra *stopword* yang digunakan diambil dari *repository* yang disusun oleh Purwarianti [15]. Berikut contoh data sebelum dan setelah penghapusan *stopword.*

Tabel 3.7 Penghapusan *Stopword*

Sebelum *stopword-removal* :	Setelah *stopword-removal:*

bandara international juanda surabaya terminal mati lampu tidak ada switch ke genset tidak ada penjelasan dari petugas mati lampu sampai menit baru akhirnya genset menyala	bandara international juanda surabaya terminal mati lampu switch genset penjelasan petugas. mati lampu menit akhirnya genset menyala

3.1.4.6 Tokenisasi

Dalam langkah ini data teks dalam paragraf akan dipisahkan menjadi potongan kata tunggal dengan spasi sebagai pemishanya. Berikut merupakan contoh kata sebelum dan setelah tekonisasi

Tabel 3.8 *Tokenization*

Sebelum Tokenisasi: bandara international juanda surabaya terminal mati lampu switch genset jelas petugas mati lampu menit akhir genset nyala	Setelah Tokenisasi: [`bandara`, `international`, `juanda`, `surabaya`, `terminal`, `mati`, `lampu`, `switch`, `genset`, `jelas`, `petugas`, `mati`, `lampu`, `menit`, `akhir`, `genset`, `nyala`]

3.1.5 Mencari TF-IDF Data

Pada tahap ini dilakukan pencarian nilai tf-idf dari masing-masing kata yang ada di dalam data. Tujuan dari pencarian tf-idf ini adalah untuk menemukan kata-kata yang tergolong dalam kata umum dalam corpus untuk meminimalisir kemungkinan hasil yang ambigu dari model.

3.1.6 Membentuk *Phrase* Data

Pada tahap ini dilakukan pembentukan frase dari kata-kata yang ada dalam data. Pembentukan frase ini dilakukan dengan mencari *co-occurance* antar kata dalam data, kata-kata yang memiliki frekuensi *co-occurance* yang tinggi akan dibentuk menjadi frase. Tujuan dari pembentukan frase ini adalah untuk mencegah keambiguan kata yang dihasilkan di dalam topik. Pembentukan frase dilakukan dengan menggunakan library genism.

3.1.7 *Topic Modeling* dengan *Gaussian Latent Dirichlet Allocation*

Pada tahap ini akan dilakukan pembentukan model topik dengan menggunakan metode *Gaussian Latent Dirichlet Allocation*. Model yang dibuat akan dipilih melalui tahap validasi model untuk ditentukan model yang paling tepat dengan hasil luaran yang baik. Untuk menghasilkan model dengan luaran yang baik dan tepat, pembentukan model akan melalui eksperimen dengan menentukan nilai dari iterasi yang akan digunakan serta nilai input dari parameter *Gaussian LDA*. Parameter *Gaussian LDA* yang dimaksud adalah jumlah topik yang ada dalam dokumen.

3.1.8 Evaluasi Model

Pada tahap ini dilakukan validasi pada model topik yang dihasilkan dari tahap sebelumnya, validasi model bertujuan untuk memastikan apakah model topik yang dihasilkan dari tahap sebelumnya sudah benar, baik topik dan kata dalam topik yang muncul. Dalam tugas akhir ini, validasi akan dilakukan dengan menggunakan uji *Perplexity* dan *Pointwise Mutual Information*.

Pada tahap juga ini dilakukan analisis kualitas dari model dengan membandingkan hasil pemodelan menggunakan metode LDA dan *Gaussian LDA*. Perbandingan ini akan menguji seberapa baik topik yang dihasilkan model dapat merepresentasikan dokumen dilihat dari hasil uji *perplexity* dan *pointwise mutual information* dari masing-masing pemodelan. Selain itu pada tahap ini juga akan dilakukan pemetaan topik-topik yang dihasilkan dari model ke dalam label yang telah ditentukan.

3.1.9 Analisis Hasil

Pada tahap ini akan dilakukan pengambilan keputusan hasil dari tahap topic modelling dengan Gaussian LDA dan validasi model. Tahap ini akan menghasilkan 2 kesimpulan yaitu topik apa saja yang dibahas dalam dokumen serta seberapa baik topik

tersebut menginterpretasikan dokumen dan kata-kata penyusun setiap topik serta seberapa baik kata-kata tersebut dapat merepresentasikan topik.

3.1.10 Dokumentasi

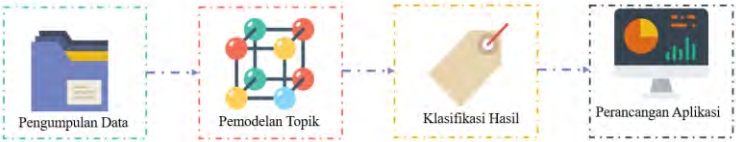
Pada tahap ini dilakukan penyusunan dokumentasi dari proses selama pengerjaan tugas akhir yang akan menghasilkan buku tugas akhir. Buku tugas akhir yang dihasilkan diharakan dapat dimanfaatkan sebagai referensi untuk penelitian selanjutnya.

Halaman Sengaja Dikosongkan

BAB IV
PERANCANGAN

Bab ini menjelaskan bagaimana rancangan dari penelitian tugas akhir yang meliputi subyek dan obyek dari penelitian, pemilihan subyek dan obyek penelitian dan bagaimana penelitian ini akan dilakukan.

4.1 Cakupan *Topic Modeling*



Gambar 4.1 Cakupan *Topic Modeling*

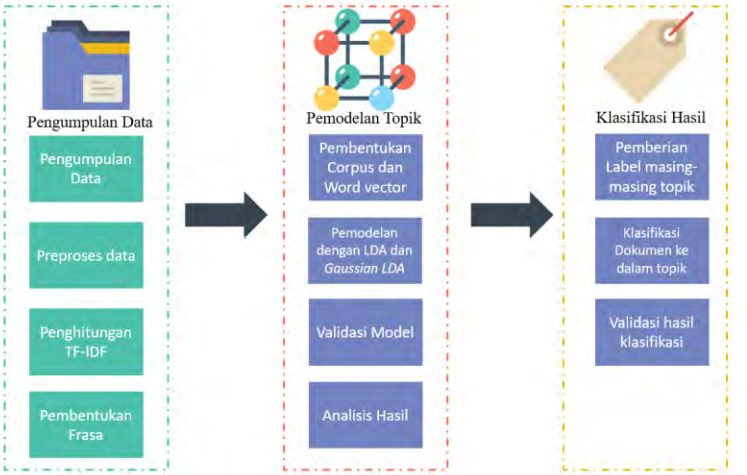
Tahap-tahap yang dilakukan dalam *topic modeling* secara umum di kelompokkan menjadi 4 bagian yaitu

1. Pengumpulan data yang akan digunakan untuk pemodelan, data bisa didapatkan dari berbagai sumber sesuai dengan kebutuhan dan objek pemodelan
2. Pemodelan topik yang merupakan inti dari proses *topic modeling,* pemodelan dapat dilakukan dengan menggunakan metode pemodelan topik yang ada seperti LDA, LSI, BTM, CTM, *Gaussian LDA* ataupun melakukan ekperimen baru dengan menggunakan metode baru sesuai dengan objek yang akan diteliti. Pada tahap ini pula dilakukan validasi dan analisis pada hasil dari pemodelan topik untuk mendapatkan hasil terbaik. Validasi dan analisis dilakaukan dengan metode tertentu sesuai dengan objek yang akan divalidasi dan analisis.
3. Tahap ke tiga yaitu melakukan klasifikasi data berdasarkan pada hasil pemodelan, pada tahap ini dilakukan pemberian label untuk masing-masing topik yang didapatkan kemudian dilakukan pelabelan pada masing-masing

dokumen dalam data untuk ditentukan dokumen tersebut masuk ke dalam topik apa.

4. Tahap ke empat yaitu tahap pengembangan aplikasi yang dapat digunakan untuk melakukan *topik modeling* secara otomatis. Input dari aplikasi ini yaitu berupa model terbaik yang dihasilkan dari ekperimen pemodelan topik dan label dari masing-masing topik dalam model. Pada tahap ini sejauh mana aplikasi dikembangkan akan disesuikan dengan kebutuhan dari *user* mengenai hasil dari *topic modeling.*

Berikut gambaran umum proses pemodelan yang dilakukan dalam tugas akhir ini:



Gambar 4.2 Rangkaian proses pemodelan topik dalam tugas akhir

Dalam tugas akhir ini, tahap *topic modeling* dilakukan tahap 1 sampai dengan tahap 3 yaitu melakukan pengumpulan data, *topic modeling* dan klasifikasi hasil pemodelan. Tahap pemodelan topik dilakukan dengan menggunakan 2 metode yaitu *LDA* dan *Gaussian LDA* yang kemudian akan divalidasi dengan nilai *perplexity* dan *topic coherence* untuk menentukan jumlah topik terbaik yang akan digunakan serta dianalisis dengan menggunakan *pointwise mutual information* untuk

mengetahui metode mana yang menghasilkan model terbaik yang dapat mengelompokkan kata ke dalam topik berdasarkan pada kesamaan makna atau semantik dari setiap kata.

4.2 Pengambilan Data

Data merupakan objek analisis utama yang dibutuhkan dalam melakukan analisis pada suara pelanggan PT Angkasa Pura 1 (Persero) cabang Juanda. Data yang dibutuhkan untuk melakukan anlisis adalah data dalam bentuk teks yang didapatkan dari website suara pelaggan Juanda.

Data didapatkan dari website suara pelanggan juanda yang telah diekstraksi oleh *Customer Service* Juanda dari tahun 2015 sampai dengan Maret 2018. Atribut dari data yang dihasilkan yaitu Nama, Tanggal, Jenis Penerbangan, Lokasi, Komentar, Maskapai, dan Kebangsaan. Tabel di bawah merupakan penjelasan mengenai keterangan dari atribut beserta tipe data dari database suara pelanggan Juanda.

Tabel 4.1 Keterangan Atribut Database

Nama Atribut	Tipe Data	Keterangan
Nama	Varchar(50)	Atribut ini berisi nama dari pelanggan yang ingin menyampaikan masukan
Tanggal	Date	Atribut ini berisi tanggal pengiriman masukan oleh pelanggan
Jenis Penerbangan	Varchar (25)	Atribut ini berisi jenis penerbangan

		atau perjalanan dari pelanggan
Lokasi	Varchar (25)	Atribut ini berisi tentang Lokasi dari pelanggan saat memberikan masukan
Komentar	Text	Atribut ini berisi tentang masukan atau keluhan yang ingin disampaikan oleh pelanggan kepada pihak perusahaan
Maskapai	Varchar (25)	Atribut ini berisi tentang maskpai yang digunakan oleh pelanggan
Kebangsaan	Varchar (3)	Atribut ini berisi tentang kebangsaan dari pelanggan

Data yang didapatkan dari website suara pelanggan Juanda adalah sejumlah 722 data.

4.3 Seleksi Atribut

Setelah pengambilan data, langkah yang dilakukan selanjutnya yaitu melakukan selesi dari atribut data. Penyeleksian atribut ini bertujuan untuk memilih atribut apa saja yang akan dianalisis selama penelitian berlangsung. Tabel di bawah menampilkan ketrangan penyeleksian atribut dari data.

Tabel 4.2 Seleksi Atribut

Nama Atribut	Seleksi	Keterangan
Nama	X	Tidak digunakan
Tanggal	X	Tidak digunakan
Jenis Penerbangan	X	Tidak digunakan
Lokasi	X	Tidak digunakan
Komentar	V	Digunakan
Maskapai	X	Tidak digunakan
Kebangsaan	X	Tidak digunakan

Dari hasil penyeleksian atribut didapatkan bahwa atribut yang akan digunakan untuk penelitian ini adalah atribut komentar. Atribut komentar dipilih karena mengandung teks yang panjang serta berisi konten yang sesuai dengan objek dari penelitian.

4.4 Metodologi Implementasi Penelitian

Metodologi implementasi penelitian menjelaskan tentang metodologi yang digunakan dalam mengimplementasikan penelitian untuk mencapai tujuan penelitian. Tahapan implementasi ini dilakukan dengan penyesuaian pada komputasi secara otomatis menggunakan bahasa pemrograman *Python*. Terdapat lima tahapan dalam melakukan implementasi penelitian, yaitu *load* data, pra-proses data, pemrosesan data atau *topic modeling* dengan menggunakan *Latent Dirichlet Allocation* dan *Gaussian Latent Dirichlet Topic*, validasi model menggunakan nilai *perlexity, topic coherence* dan analisis topik model.

4.3.1 *Load* Data

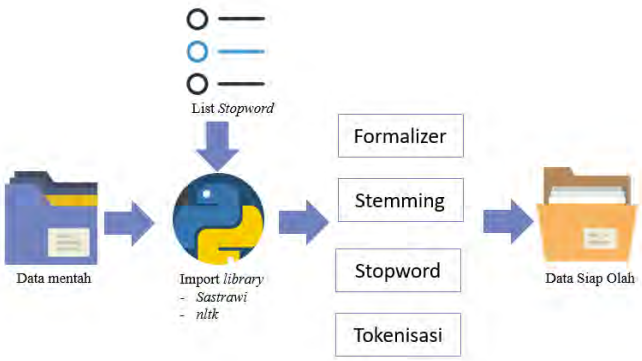
Tahap load data merupakan tahapan untuk mengolah data yang sudah dimiliki menjadi bentuk struktur data yang dibutuhkan sebelum melakukan analisis. Data yang dimiliki sebelumnya masih dalam format xls. Data ini kemudian akan diubah menjadi format *txt*. Alur tahap persiapan data sesuai dengan gambar 4.3.

Gambar 4.3 Alur Persiapan Data

Data mentah dalam format *.xlxs* akan dikonversi menjadi file dengan format *.txt* dengan menggunakan jupyter notebook. Data yang dimuat dalam proses ini adalah data keseluruhan suara pelanggan Juanda dari tahun 2015 sampai dengan 2018.

4.3.2 Pra-Proses Data

Pra-proses data merupakan proses yang mencakup beberapa langkah utama. Tujuan dari pra-proses daya adalah untuk menyeragamkan data dan membuang kata-kata yang tidak memiliki makna signifikan dalam data. Langkah-langkah yang dilakukan adalah menjadikan semua data menjadi huruf kecil, menormalisasi semua kata yang ada dalam data sehingga menjadi kata yang baku, *stemming* atau mengubah semua kata menjadi kata dasar, *stopword removal* atau menghapus kata-kata yang tidak memberikan informasi yang signifikan untuk data dan tokenisasi atau pemecahan kata menjadi token yang selanjutnya disimpan di dalam *array*. Alur pra-proses data dapat dilihat pada gambar 4.4.



Gambar 4.4 Alur Pra-proses Data

Pada tahap pengubahan data menjadi huruf kecil, tokenisasi, penghapusan karakter angka, dan penghapusan *stopwords* digunakan *library nltk.* Namun pendefinisian daftar *stopwords* yang digunakan pada tugas akhir ini mengacu pada susunan *stopword* yang telah disusun oleh Purwarianti [15] dan dalam penelitian ini dilakukan pendefinisian stopword dari data keluhan pelanggan yang dianggap masih merupakan *common word* dalam data, pendefinisan stopword ini akan dilakukan dengan melihat nilai TF-IDF dari masing-masing kata. Selain menghitung nilai TF-IDF dari kata dalam data, dilakukan pembentukan frase dengan mencari *co-occurance* antar kata dalam data untuk meminimalisir kata ambigu yang akan dihasilkan pada model, pembentukan frase dalam data dilakukan dengan menggunakan library *gensim.* Sementara itu, untuk proses *stemming*, digunakan *library Sastrawi* di bawah lisensi MIT.

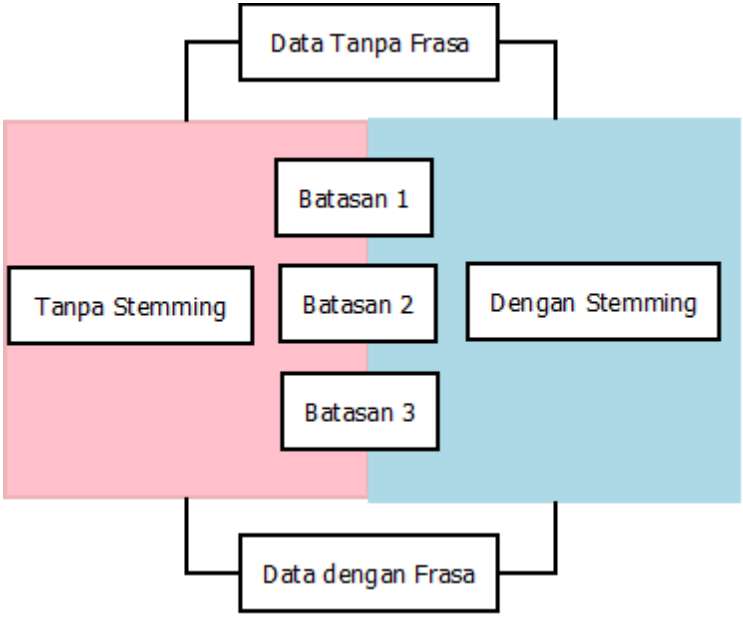
Setelah melalui pra-proses di atas, data dianggap sudah bersih dan siap untuk diproses pada tahap proses data.

4.3.3 Penetuan Skenario dalam Pemodelan Topik

Pada tahap ini dilakukan pembentukan skenario dari pemodelan topik yang akan dilakukan baik menggunakan *Gaussian LDA* maupun *LDA.* Skenario yang dibuat berdasarkan pada jenis

*input*an data yang akan dimasukkan pada saat melakukan pemodelan. Dalam penelitian ini data yang dimasukkan terbagi menjadi 2 yaitu data tanpa frasa dan data dengan frasa. Kemudian kedua jenis data dibagi lagi berdasarkan pada batasan bawah, batas minimum kemunculan kata dalam data, yang dihasilkan dari penrhitungan TFIDF per jenis data. Setelah skenario ditentukan data selanjutnya diteruskan ke proses pemodelan.

Berikut Skenario yang akan digunakan dalam pemodelan topik:



Gambar 4.5 Skema Skenario dalam Pemodelan Topik Berdasarkan Data *Input*

- Data Tanpa Frasa dengan batasan bawah 1 tanpa *stemming*
- Data Tanpa Frasa dengan batasan bawah 1 dengan *stemming*
- Data Tanpa Frasa dengan batasan bawah 2 tanpa *stemming*

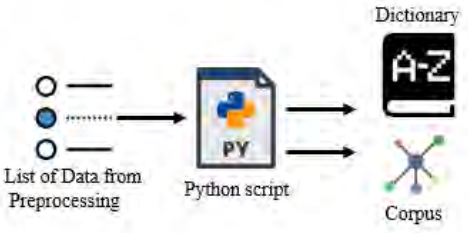
- Data Tanpa Frasa dengan batasan bawah 2 dengan *stemming*
- Data Tanpa Frasa dengan batasan bawah 3 tanpa *stemming*
- Data Tanpa Frasa dengan batasan bawah 3 dengan *stemming*
- Data frasa dengan batasan bawah 1 tanpa *stemming*
- Data frasa dengan batasan bawah 1 dengan *stemming*
- Data frasa dengan batasan bawah 2 tanpa *stemming*
- Data frasa dengan batasan bawah 2 dengan *stemming*
- Data frasa dengan batasan bawah 3 tanpa *stemming*
- Data frasa dengan batasan bawah 3 dengan *stemming*

4.3.4 *Proses Data dengan Latent Diriclet Allocation (LDA)*

Pada tahap ini mulai dilakukan pemrosesan data dengan menggunakan metode *Latent Dirichlet Allocation* dari pembentukkan *corpus* hingga penggunaan metode *Topic Models* untuk melakukan klasterisasi pada dokumen.

4.3.4.1 Pembentukan Dictionary dan Corpus

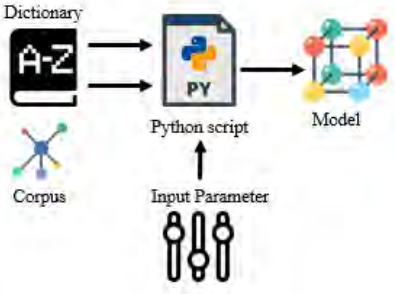
Tahap pertama yang dilakukan untuk melakukan topik modeling dengan *Latent Dirichlet Allocation* yaitu dengan membentuk *dictionary* dan *corpus* dengan input berupa dat yang telah diberishkan sebelumnya.. *Dictionary* merupakan format data yang mengandung himpunan kata unik yang diberi indeks, untuk memudahkan proses menampilkan kata dalam model. Sedangkan *corpus* merupakan format data yang berbentuk dokumen *term-matrix*, untuk digunakan dalam pembentukan model. Alur pembuatan *corpus* dan *dictionary* dat dilihat pada gambar 4.6.



Gambar 4.6 Alur Pembuatan *Corpus dan Dictionary*

4.3.4.2 Topic Modeling dengan Latent Diriclet Allocation (LDA)

Topic modeling dengan menggunakan *Latent Dirichlet Allocation* dilakukan dengan pembentukan model menggunakan *library genism,* model yang dihasilkan kemudian dievaluasi dengan evaluasi *perplexity* dan *topic coherence* untuk menentukan model terbaik. Alur proses *topic modeling* dapat dilihat dari gambar 4.7.



Gambar 4.7 Alur Proses *Topic Modeling* dengan LDA

Dalam membentuk model, dilakukan eksperimen pada *input parameter*. Eksperimen ini dilakukan untuk menghitung perplexity dari model untuk menentukan *passes* proses sehingga mencapai konvergen dan jumlah topik yang menjadi kelompok klasterisasi kata. Model dengan nilai *perplexity* terkecil dan tidak berubah-ubah akan dipilih menjadi model yang digunakan dan dianggap yang paling mendekati akurat.

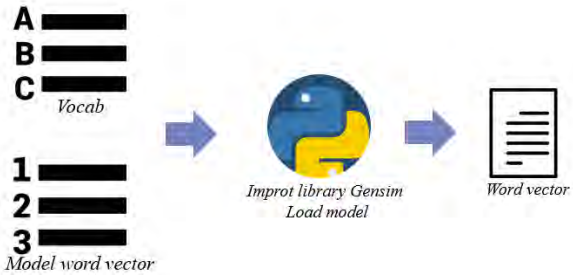
4.3.5 *Proses Data dengan Gaussian Latent Diriclet Allocation (GLDA)*

Pada tahap ini mulai dilakukan pemrosesan data dengan menggunakan metode *Gaussian Latent Dirichlet Allocation* dari pemrosesan *corpus* dan pemrosesan kata menjadi vektor hingga penggunaan metode *Topic Models* untuk melakukan klasterisasi pada dokumen.

4.3.5.1 Pemrosesan Corpus

Pada tahap ini dilakukan pemrosesan pada data yang diinputkan yaitu data bersih dari hasil pra-proses data. Data yang menjadi input akan diproses menjadi *corpus* yang merupakan data map berisikan dokumen serta kata dan topik yang ada di dalam setiap dokumen tersebut.

4.3.5.2 Pemrosesan Vektor Kata



Gambar 4.8 Alur Pemrosesan Kata Menjadi Vektor

Pada tahap ini dilakukan pemrosesan seluruh kata yang ada di dalam data menjadi bentuk vektor. Referensi vector kata yang digunakan untuk mengubah kata menjadi vektor akan pada penelitian ini adalah model word2vec yang telah disusun oleh Purwarianti [15]. Alur pemrosesan vektor kata dapat dilihat dari gambar 4.8.

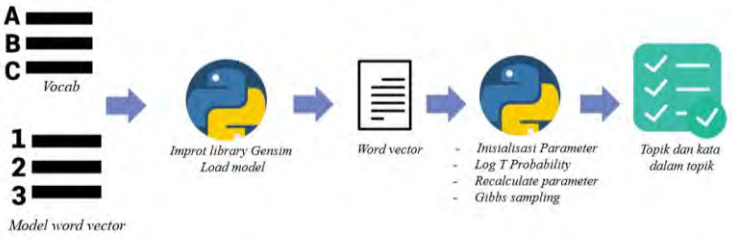
4.3.5.3 Inisialisasi Parameter

Pada tahap ini dilakukan inisiasi untuk semua parameter yang digunakan dalam *topic modeling*. Parameter-parameter yang

akan diinisiasi adalah *topic mean, topic covariance, sample mean dan sample covariance.*

4.3.5.4 *Topic Modeling* dengan *Gussian Latent Diriclet Allocation (GLDA)*

Pada tahap ini dilakukan pembentukan model topik dari data yang diinputkan. Tahapan pembentukan model topik dilakukan dengan penyesuaian pada komputasi secara otomatis menggunakan bahasa pemrograman *Python*. model yang dihasilkan kemudian dievaluasi dengan evaluasi *perplexity* dan *topic coherence* untuk menentukan model terbaik. Alur proses *topic modeling* dapat dilihat dari gambar 4.9.



Gambar 4.9 Alur Pemodelan Topik dengan *Gaussian LDA*

Dalam membentuk model, dilakukan eksperimen pada jumlah topik yang akan digunakan dan data yang dijadikan *input* dalam pemrosesan data.

4.3.6 Validasi Model

Tahapan ini bertujuan untuk memastikan model yang dibentuk dari proses *topic modeling* pada dokumen menghasilkan nilai probabilitas tertinggi, baik luaran berupa topik maupun kata-kata yang menyusun setiap topik. Beberapa hal yang diperhatikan dalam tahap validasi model adalah;

1. Jumlah *passes* yang tepat untuk membentuk model.
2. Jumlah topik yang sesuai berdasarkan nilai *perplexity* tiap model.
3. Jumlah topik yang sesuai berdasarkan nilai *topic coherence* tiap model.

4.4 Perancangan Pengujian Model

Perancangan pengujian model yang telah didapatkan dari hasil validasi topik bertujuan untuk mengetahui kemampuan masing-masing model dalam melakukan prediksi kemiripan antara kata yang terdapat dalam setiap topik. Terdapat 3 tahapan pada proses pengujian ini, yaitu perhitungan jumlah kemunculan kata dalam *corpus*, kemudian mencari nilai dari *p(x), p(y)* dan *p(x,y)* dan menghitung kemiripan kata dalam masing-masing topik menggunakan *Pointwise Mutual Information*.

4.4.1 Jumlah Kemunculan Kata dalam *Corpus*

Pada tahap ini dilakukan perhitungan jumlah kemunculan kata dalam *corpus.* Perhitungan ini akan menghasilkan jumlah kemunculan masing-masing kata dari 5 kata dengan probabilitas tertinggi dalam topik yang selanjutnya dapat digunakan untuk menghitung PMI. *Corpus* yang digunakan merupakan hasil dari *scrapping post* dari 23 halaman di *facebook.*

4.4.2 Perhitungan *p(x), p(y), p(x,y)* dalamPMI

Setelah mengetahui jumlah kemunculan seluruh kata pada 5 kata dengan probabilitas tertinggi dalam topik, selanjutnya dihitung *p(x)* yang merupakan nilai probabilitas dari kata *x* yang akan dicari nilai PMInya, kata-kata dengan probabilitas tertinggi dalam setiap topik dan *p(y)* yang merupakan probabilitas dari kata *y,* kata lain selain *x* yang memiliki probabilitas tertinggi. Selanjutnya yaitu menghitung *p(x,y)* yaitu probabilitas dari *co-occurance* kata *x* dan *y* di dalam *corpus*.

4.4.3 Perhitungan Kemiripan Kata-Kata dalam Topik dengan *Pointwise Mutual Information*.

Setelah mengetahui jumlah kemunculan setiap kata dalam *corpus* dan nilai *p(x), p(y), p(x,y)* selanjutnya dilakukan perhitungan kemiripan kata dalam topik. Perhitungan kemiripan kata dalam topik ini dilakukan menggunakan

Pointwise Mutual Information. *Pointwise Mutual Information* digunakan untuk mengetahui hubungan antar kata yang ada di dalam topik dengan menghitung jumlah kemunculan kata serta probabilitas kata dalam *corpus* dan probabilitas *co-occurrence* dari kata-kata yang ada di dalam topik.

Hasil perhitungan kemiripan kata-kata dalam topik akan dianalisa dengan analisis kuantitas dan kualitas yang kemudian dibandingkan antara hasil dari metode *Gaussian LDA* dengan *LDA* dengan tujuan untuk mengetahui metode mana yang menghasilkan model dengan kata-kata dalam topik yang lebih mirip.

4.5 Analisis Hasil

Setelah mengetahui model terbaik dari hasil pemodelan topik, kemudian dilakukan penentuan atau pelabelan untuk setiap topik yang dihasilkan dilihat dari kata-kata penyusun topik tersebut. Kemudian setelah setiap topik ditentukan labelnya, akan dilakukan pengujian dengan 100 data *testing* untuk mengetahui seberapa baik model dapat melakukan pemodelan terhadap topik.

Pengujian dilakukan dengan menghitung probabilitas setiap topik terhadap dokumen dan mengambil 5 topik tertinggi dari jumlah topik yang dimiliki model. Kemudian 5 topik ini akan dibandingkan dengan label yang diberikan untuk setiap dokumen.

BAB V
IMPLEMENTASI

Bab ini menjelaskan implementasi dari perancangan yang telah dilakukan sesuai dengan metode pengembangan yang dibuat. Bagian implementasi ini akan menjelaskan mengenai lingkungan implementasi, pembuatan fitur-fitur aplikasi dalam bentuk kode, serta pengujian aplikasi.

5.1 Lingkungan Implementasi

Pengembangan aplikasi ini menggunakan komputer dengan spesifikasi sesuai Tabel 5.1.

Tabel 5.1 Spesifikasi Komputer

Windows Based Operating Systems	
Prosesor	Intel® Core™ i3-5005U CPU @2.00GHz 2.00GHz
Memory	4 GB RAM
Graphic Card	AMD Radeon R7 M370 2GB
Sistem Operasi	*Windows 10 Home*
Arsitektur Sistem	*64-bit Operating System, x64-based processor*

Model dikembangkan dengan menggunakan beberapa teknologi seperti editor, bahasa pemrograman, dan *library* yang disajikan dalam Tabel 5.2.

Tabel 5.2 Teknologi yang digunakan untuk mengembangkan model

Bahasa Pemrograman	• Python 3.6

***Editor* (IDE)**	• *Sublime Text* • *Command Prompt* • *Jupyter Notebook*
Software/Tools	Microsoft Excel 2013
Library	• Anaconda 4.4 • Gensim 3.1 • Sastrawi 1.0.1

5.2 *Load* Data

Sebelum memulai tahapan *topic modeling*, dilakukan persiapan data untuk diolah sehingga siap digunakan dalam pemrosesan. Tahapan yang akan dilalui dalam mempersiapkan data, yaitu memuat data ke dalam *Python environment*.

Tahapan memuat data dilakukan dengan menggunakan Jupyter Notebook. Data yang dimuat memiliki format *.csv*. Agar format ini dapat dibaca maka digunakan modul CSV seperti dalam kode 5.2.

```
import logging
logging.basicConfig(format='%(asctime)s :
%(levelname)s : %(message)s', level=logging.INFO)

import os, re, csv
from pprint import pprint

with open('dir_data/pmi.csv', encoding='utf8') as
myFile:
        reader = csv.reader(myFile)
```

Kode 1 ***Import Library logging, os, re & csv***

Setelah data dimuat data kemudian dilanjutkan ke tahap pra-proses data.

5.3 Pra-proses Data

Tahap pra-proses data merupakan tahap yang dilakukan agar data yang digunakan sesuai dengan masukkan yang dibutuhkan untuk diproses dengan model. Dalam pra-proses data terdapat

enam tahapan yang dilakukan, yaitu pendefinisian *stopword* dan kata baku, menjalankan *case folding, formalizer, stemming*, tokenisasi, dan penghapusan stopword, perhitungan TF-IDF, pembuatan frasa berdasarkan kata-kata pada dokumen.

5.3.1 Pendefinisian *stopword,* kata baku dan *formalizer*

Pendefinisian *stopword* dilakukan dengan menyimpan daftar *stopword* pada *file* dengan format *.csv* dengan satu kata per baris. Daftar *stopword* ini dimuat ke dalam variabel list dengan nama *'list_stopword'*. Karena *stopword* disimpan dalam *file .csv*, dibutuhkan *library csv* untuk digunakan memuat daftar *stopword.* Kode yang digunakan untuk mendefinisikan *stopword* terdapat pada Kode 5.8.

```
import csv

def initiation_dictionary():
with open('dir_data/stopwords.txt') as dictionary3:
       for line3 in dictionary3:
              list_stopwords.append(line3.strip())
```

Kode 2 Pendefinisian *Stopword*

Pendefinisian kata baku juga dilakukan dengan menyimpan dafar kata baku dalam *file* dengan format *.csv* dengan satu kata per baris. Daftar kata baku ini kemudian dimuat ke dalam *file .csv* dan membutuhkan *library csv* yang digunakan untuk memuat daftar kata baku ini. Kode yang digunakan untuk mendefinisikan kata baku terdapat pada Kode 3.

```
import csv

# Stopword List
def initiation_dictionary():
with open('dir_data/katabaku.csv') as dictionary1:
   for line1 in dictionary1:
        katabaku_dict.append(line1.split('\n')[0])
```

Kode 3 Pendefinisian Kata Baku

Pendefinisian *formalizer* dilakukan dengan menyimpan daftar kata-kata yang tidak formal dan kata formal dalam *file* berformat *.csv* dengan dua kata per baris, dimana kata pertama merupakan kata tidak formal dan kata kedua merupakan kata

dalam bentuk formal. Daftar kata baku ini kemudian di muat ke dalam *file .csv* dengan menggunakan *library csv* yang disediakan dalam *python.* Kode yang digunakan untuk mendefiniskan *formalizer* tedapat dalam Kode 4.

```
import csv

def initiation_dictionary():
with open('dir_data/dt.csv') as dictionary2:
        for line2 in dictionary2:
                formalizer_dict.append(line2.replace("\
        n", "").split(','))
```

Kode 4 Pendefinisian *Formalizer*

5.3.2 Data *Cleaning*

Dalam data *cleaning*, data yang akan dibersihkan merupakan *list* data yang telah dimuat pada tahap sebelumnya. Pembersihan data akan dilakukan dengan membersihkan karakter numberik dan symbol. Setelah setiap *list* dalam data selesai dibersihkan maka akan dikembalikan dalam variable *sentences* untuk diproses ke tahap berikutnya. Kode yang digunakan untuk melakukan pembersihan data terdapat dalam Kode 5.

```
def removesymbol(word):
# ubah pakai .strip()
symbol = "`1234567890=~!@#$%^&*()_\"'+[];:,./<>?"
        newword = ""
        for char in word:
                newchar = char
                for char2 in symbol:
                        if(char2 == char):
                                newchar = " "

                newword = newword + newchar
return newword
```

Kode 5 Data Cleaning

5.3.3 *Case folding, formalizer, stemming*, tokenisasi, dan penghapusan *stopword*

Setelah *data cleaning* dan pendefinisian *stopword*, kata baku dan *formalizer*. *Stopword* akan digunakan untuk menghapus *stopword* atau kata-kata umum yang dalam data. Daftar kata baku akan digunakan untuk melakukan *checking* pada kata-kata dalam data yang tidak baku untuk diproses ke dalam *formalizer.* Serta daftar *formalizer* yang telah didefiniskan akan digunakan untuk mengubah kata-kata yang tidak formal ke dalam bentuk formal.

Tahap pertama yang dilakukan adalah tahap *case folding.* Dokumen yang telah diberishkan dan tersimpan dalam variable *sentences* kemudian diproses dalam tahap *case folding* untuk diubah menjadi *lowercase* dengan tipe data *string.* Dokumen yang dihasilkan dari tahap *case folding* kemudian dilanjutkan ke tahap *formalizer.* Pada tahap *formalizer,* kata-kata yang ada di dalam dokumen akan diperiksa satu persatu, apakah kata tersebut sudah baku atau tidak. Kemudian kata-kata yang tidak formal atau baku akan diganti dengan kata baku.

```
def formalize(sentence):
    newsentence = ''

    # Mengubah menjadi huruf kecil
    sentence = sentence.lower()
    sentence = removesymbol(sentence)
    sentence = sentence.replace('\n'," ")
    isfirst = 0
    for word in sentence.split(' '):
        if(status != 1):
           for line2 in formalizer_dict:
             if(word == line2[0]):
                word = line2[1]
                status = 1
                break
```

Kode 6 Melakukan *Formalizer* pada Data

Dalam *formalizer,* pertama data berupa *list* kata dipisah dan disimpan di dalam variable '*word'*, sebelum masuk ke *formalizer* kata-kata dalam dokumen disaring, kata yang terdiri kurang dari 2 huruf akan dihapus. Kemudian semua kata dalam variable '*word'* akan di-*check* satu persatu dengan kata yang

ada di dalam variable *formalizer_dict* untuk dicocokkan dengan kata-kata yang ada di dalam daftar *formalizer*. Jika kata dalam variable '*word'* cocok dengan kata pertama dalam dafar *formalizer* maka akan diganti dengan kata ke dua dalam dafatr *formalizer* yang berarti kata tersebut termasuk dalam kata tidak formal atau baku. Daftar kata yang telah melalui tahap *formalizer* kemudian disimpan di dalam *newsentence* untuk diproses pada tahap selanjutnya. Kode yang digunakan untuk melakukan *formalizer* ditunjukkan dalam Kode 6.

Pada tahap *formalizer* dilakukan *checking* kata pada komentar yang berbahasa inggris. *Checking* dilakukan dengan menggunakan library *enchant* untuk mendeteksi kata dalam Bahasa inggris. *Treatment* yang dilakukan dengan menggunakan library ini adalah menghapus kata-kata yang dideteksi menggunakan Bahasa inggris. Kode yang digunakan untuk melakukan pendekteksian Bahasa inggirs ditunjukkan dalam kode 7.

```
if(status != 1):
        d = enchant.Dict("en_US")
        # Dihapus jika bahasa inggris
        if(d.check(word)):
                word = ""
                status = 1
```

Kode 7 Pendeteksian Kata Berbahasa Inggris

Tahap selanjutnya yaitu *stemming*. Dalam tahap *stemming*, pertama dilakukan pendefinisian variable *stemmer* seperti dalam Kode 8.

```
def steammer(sentence):
        fixsentence = ""
        factory = StemmerFactory()
        stemmer = factory.create stemmer()
```

Kode 8 Pendefinsian *stemming*

Proses *stemming* akan dilakukan dengan menggunakan *Sastrawi.Stemmer* dengan modul *StemmerFactory* untuk proses *stemming* data dalam variable *sentence*. Kode yang digunakan untuk melakukan *stemming* ditunjukkan dalam Kode 9.

```
sentence = stemmer.stem(sentence)
```

Kode 9 Melakukan *Stemming*

Setelah melalui tahap *stemming* data yang disimpan dalam variable *sentence* kemudian dilanjutkan ke tahap tokenisasi. Tokenisasi dilakukan dengan menggunakan *nltk.tokenize* dengan modul *RegexpTokenizer*. Dalam tahap ini kata-kata di dalam dokumen dipecah dan disimpan dalam bentuk token per kata pada variable *'tokens'*. Kode yang digunakan untuk melakukan tonkenisasi ditunjukkan dalam Kode 10.

```
def tokenize(text):
    return [tok.strip().lower() for tok in REGEX.split(text)]

def stopwords(sentence):
        tokens = tokenize(sentence)
```

Kode 10 Pendefinisian Tokenisasi

Tahap selanjutnya adalah penghapusan *stopword*. Dalam tahap ini *list* kata dalam data akan di-*check* dengan menggunakan daftar *stopword* yang telah didefiniskan untuk dicocokkan. Selanjutnya, kata-kata dalam dokumen yang sesuai dengan kata-kata pada daftar s*topword* akan dihapus dari dokumen dan disimpan di dalam *'stopped_tokens'*. Kode yang digunakan untuk melakukan penghapusan *stopword* ditunjukkan dalam Kode 10.

```
stopped_tokens = [sentence for sentence in tokens if not
sentence in list_stopwords]
return stopped_tokens
```

Kode 11 Melakukan Tokenisasi

Data dalam variable *'stopped_tokens'* kemudian disimpan dalam variable *sentence* untuk digunakan pada tahap selanjutnya.

5.3.4 Perhitungan TF-IDF

Pada tahap ini dilakukan perhitungan TF-IDF dari kata-kata yang ada di dalam dokumen. Perhitungan TF-IDF dilakukan karena dalam data masih terdapat kata-kata yang bersifat umum dan tidak memberikan informasi yang spesifik jika masuk ke

dalam kata dalam topik. Misalnya kata *'bandara'* yang muncul di dalam 5 topik dengan probabilitas tinggi sedangkan kata bandara tidak memberikan informasi yang cukup spesifik untuk mendeskripsikan topik-topik tersebut. Data yang akan menjadi input dalam tahap ini adalah data hasil dari pra-proses data.

Dalam tahap ini pertama dilakukan *load data* yang akan diproses untuk dicari TF-IDFnya seperti yang ditunjukkan dalam kode 12.

```
train_set = [line.strip() for line in
open('input\phrase_withstem_batasan2.txt', 'r')]
```

Kode 12 Memuat Data yang Diproses

Selanjutnya dilakukan pendefinisian daftar *vocab* atau kata unik yang ada di dalam data dengan Kode 13.

```
from sklearn.feature_extraction.text import CountVectorizer

def tokenize(text):
    return [tok.strip().lower() for tok in REGEX.split(text)]

count_vectorizer = CountVectorizer(tokenizer=tokenize)
data = count_vectorizer.fit_transform(train_set).toarray()
vocab = count_vectorizer.get_feature_names()
```

Kode 13 Mendefinisikan *Vocab*

Langkah selanjutnya yaitu perhitungan TF dari semua kata yang ada di dalam dokumen. Kode yang digunakan untuk menghitung TF kata dalam dokumen ditunjukkan dalam Kode 14.

```
xls="Term-Frequency,"
for vc in vocab:
        xls+=vc+","

xls += "\n"

i = 0
for dt in data:
        xls += "DOKUMEN "+str(i)+","
        for dta in dt:
                xls += str(dta)+","
                # print(len(train_set[i].split()))
        xls += "\n"
        i += 1
```

Kode 14 Melakukan Perhitungan *Term Frequency*

Selanjutnya dilakukan perhitungan DF yaitu jumlah dokumen tempat kemunculan kata dengan menggunakan kode 15.

```
xls += "Document-Frequency,"
for vc in vocab:
        xls+=vc+","
dfs = []
for dt in data:
        index2 = 0
        xls += ","
        for dta in dt:
                count = 0
                if dta>0:count += 1
                dfs.append(count)
                xls += str(count)+","
                index2 += 1
        break"\n"
        i += 1
```

Kode 15 Melakukan Perhitungan *Document Frequency*

Selanjutnya dilakukan perhitungan IDF dari setiap kata dengan menggunakan kode 16.

```
xls += "IDF,"
for vc in vocab:
        xls+=vc+","
i=0
IDF=[]
for dt in data:
        xls += "DOKUMEN "+str(i)+","
        j=0
        data_idf=[]
        for dta in dt:
                idf = math.log10(722/dfs[j])
                data_idf.append(idf)
                xls += str(idf)+","
                j += 1

        IDF.append(data_idf)
        i += 1
```

Kode 16 Melakukan Perhitungan *Invers Document Frequency*

5.3.5 Pembuatan frasa

Pada tahapan ini dilakukan penyusunan frasa kata yang paling sering muncul dalam seluruh dokumen. Untuk menyusun frasa, digunakan *library gensim.models* modul *Phrases*. Kemudian pembentukan *bigram* atau frasa yang terdiri dari dua kata dilakukan dengan memilih frasa. Data yang digunakan merupakan data hasil pra-proses data yang sebelumnya sudah

disimpan di dalam bentuk file *output_without_stem.txt*. Setiap frasa yang terbentuk dari kata-kata yang ada pada dokumen, akan ditambahkan ke dalam masing-masing dokumen yang ada pada file *output_pharse.txt*. Kode implementasi penyusunan *bigram* tertera pada Kode 17.

```
from collections import Counter
import nltk
from gensim.models import Phrases

sentences = []
bigram = Phrases()
with open("input/output_without_stem.txt", "r") as
sentencesfile:
    reader = sentencesfile.read().splitlines()
    sentencesfile.close()
    for row in reader:
        sentence = [row]
        sentences.append(sentence)
        bigram.add_vocab([sentence])

bigram_counter =Counter()
for key in bigram.vocab.keys():
    if range(len(key.split("_"))) > 1:
        bigram_counter[key] += bigram.vocab[key]

for key, counts in bigram_counter.most_common(20):
    print ('{0: <20} {1}'.format(key, counts))
```

Kode 17 Membuat Frasa Kata

5.4 Proses Data dengan *Latent Dirichlect Allocation* (LDA)

Tahap proses data dengan metode *Latent Dirichlect Allocation (LDA)* merupakan tahapan membentuk model dan melakukan eksperimen model untuk mendapatkan model yang paling baik.

5.4.1 Pembentukan *Dictionary* dan *Corpus*

Tahap awal yang dilakukan sebelum melakukan pemodelan topik dengan menggunakan metode *Latent Dirichlet Allocation* yaitu membentuk *Dictionary* dan *Corpus*. Tujuan dari pembentukan *Dictionary* adalah untuk menyimpan data token ke dalam bentuk *dictionary* yang diberikan indeks yang

berfungsi untuk mengidentifikasi kata-kata tersebut. Pembentukan *dictionary* dilakukan dengan menggunakan *library genism* dengan modul *corpora.* Kode yang digunakan untuk melakukan pembentukan *dictionary* ditunjukkan dalam kode 18.

```
from gensim import corpora

f = 'input/output_with_stem.txt'
with open(f, 'r') as fi:
    docs = fi.read() #  These are all cleaned out
    fi.close()

dictionary_dataAgregat_stem =
corpora.Dictionary([docs.split()])
dictionary_dataAgregat_stem.save('output/dictionary_da
taAgregatwithstem2.dict')
```

Kode 18 Membentuk *Dictionary*

Setelah *dictionary* terbentuk, tahap selanjutnya yaitu membentuk *corpus* yang didapatkan dari hasil konversi *dictionary* ke dalam bentuk matriks dokumen. Pembentukan *corpus* dilakukan dengan menggunakan *library genism* dengan modul *corpora.* Kode yang digunakan untuk membuat *corpus* ditunjukkan dalam Kode 19.

```
from gensim import corpora

corpus = [dictionary_dataAgregat.doc2bow(i) for i in
tokenlist_dataAgregat]
print("corpus")
print(corpus)
corpora.MmCorpus.serialize('output/corpora_stem.mm',
corpus)
```

Kode 19 Membentuk *Corpus*

5.4.2 Pemodelan Topik dengan *Latent Diriclet Allocation*

Setelah *dictionary* dan *corpus* dibentuk kemudian tahap selanjutnya yaitu pemodelan topik dengan menggunakan metode *Latent Dirichlet Allocation*. Untuk membentuk model topik, digunakan *library* dari *gensim.models* modul *ldamodel*. Dalam peneilitan ini beberapa *input parameter* yang harus diperhatikan dalam membentuk model topik yaitu input data

yang akan digunakan yang telah dibentuk menjadi frase dan melalui TF-IDF, batasan atas dan bawah dari data yang akan digunakan, jumlah topik dan jumlah *passes*. Dari ke empat nilai *input* ini nantinya dilakukan eksperimen untuk mendapatkan model yang terbaik. Implementasi pembentukan model sesuai dengan Kode 20.

```
from gensim.models import ldamodel

jumlahTopik = 4
jumlahPasses = 20

dictionary.filter_extremes(no_below=3, no_above=0.1)

lda = ldamodel.LdaModel(corpus,
num_topics=jumlahTopik, id2word=dictionary,
passes=jumlahPasses, iterations=100)
```

Kode 20 Pemodelan dengan Menggunakan *LDA*

5.4.3 Pendokumentasian *Logging*

Logging sangat diperlukan dalam melakukan ekperimen untuk mencatat setiap kejadian yang terjadi selama proses pembentukan model topik. Catatan hasil yang penting dan dibutuhkan untuk tahap selanjutnya yaitu catatan nilai *perplexity* yang telah dikalkulasi secara otomatis oleh modul genism. *Logging* dilakukan dengan menggunakan *library logging*. Hasil *logging* kemudian disimpan dalam file dengan format *.txt* dengan penamaan sesuai dengan kebutuhan. Kode yang digunakan untuk melakukan *logging* ditunjukkan dalam Kode 21.

```
Import logging

info = logging.basicConfig(filename='LDA-
Koheren/log/topik_log_without_' + str(jumlahTopik) +
'topics_' + str(jumlahPasses) +'passes_' +
str(no_uji_model) +'_Fr-below3'+'.txt', filemode='w',
format='%(message)s', level=logging.INFO)
```

Kode 21 Melakukan *Logging untuk* Semua Ekperimen

5.4.4 Eksperimen pemodelan topik dengan *Latent Dirichlet Allocation*

Dalam tahap ekperimen, dilakukan pemodelan topik dengan metode *Latent Dirichlet Allocation* yang bertujuan untuk membentuk model terbaik. Ekperimen dilakukan pada

1. Input data yang akan diproses yaitu data yang telah dibentuk frasa dan data tanpa frasa
2. Jumlah batas atas dan bawah, jumlah minimal dan maksimal kemunculan, dari kata yang akan diguakan setelah melalui penghitungan TF-IDF
3. *Input parameter*, yaitu *passes* dan jumlah topik.

Berdasarkan hal ini, eksperimen akan dibedakan ke dalam dua tahap, yaitu penentuan jumlah *passes* dan penentuan jumlah topik.

5.4.4.1 Penentuan Jumlah *Passes*

Penentuan jumlah *passes* yang merupakan istilah untuk menyebut iterasi atau perulangan dalam model untuk belajar sangatlah penting. Penentuan jumlah *passes* ini dilakukan untuk mengetahui dan mendapatkan jumlah perulangan yang tepat sehingga dapat menghasilkan model yang baik. Jumlah *passes* berlebihan akan menghasilkan model yang *overfitting* dan jumlah *passes* yang kecil akan menghasilkan model yang tidak konvergen untuk mendistirbusikan seluruh kata-kata yang mirip dalam satu topik.

Penentuan jumlah *passes* agar dapat menghasilkan jumlah yang tepat dilakukan dengan melalui beberapa eksperimen, dalam penelitian ini untuk menentukan jumlah *passes* dilakukan tujuh kali ekperimen dengan jumlah topik yang berbeda untuk setiap ekperimennya. Jumlah topik yang digunakan untuk menentukan jumlah *passes* adalah 2, 5, 8, 10, 11, 15, dan 20 topik. Berdasarkan ekperimen ini, semua nilai *perplexity* yang muncul dari setiap topik dan skenario akan dicatat untuk selanjutnya dianalisa tren nilai *perplexity* untuk setiap *passes.*

```
from gensim.models import ldamodel

jumlahTopik = 4
jumlahPasses = 100

dictionary.filter_extremes(no_below=3, no_above=0.1)

lda = ldamodel.LdaModel(corpus,
num_topics=jumlahTopik, id2word=dictionary,
passes=jumlahPasses, iterations=100)
```

Kode 22 Penentuan Jumlah *Passes*

Nilai *perplexity* yang dihasilkan untuk setiap *passes* kemudian akan divisualisasikan menggunakan *line chart* di *Microsoft Excel.* Dari hasil visualisasi akan dilakukan analisis tren nilai *perplexity* akan mulai stabil pada *passes* ke berapa. Jumlah *Passes* yang akan diambil adalah *passes* yang menunjukkan nilai awal dari stabilnya tren *perplexity.* Pada penelitian untuk setiap ekperimen digunakan jumlah *passes* sebanyak 100 *passes.*

5.4.4.2 Penentuan Jumlah Topik

Eksperimen untuk menentukan jumlah topik selanjutnya dilakukan setelah mendapatkan jumlah *passes* yang sesuai. Untuk melakukan ekperimen penentuan jumlah topik dilakukan percobaan dengan beberapa jumlah topik yaitu dimulai dari 4, 7, 10, 11, 13 dan 15 topik. Penggunaan beberapa jumlah topik dilakukan untuk mendapatkan nilai *perplexity* yang paling rendah. Nilai *perplexity* yang semakin rendah menunjukkan kemampuan probabilitas model yang semakin baik. Eksperimen untuk masing-masing jumlah topik dilakukan sebanyak 10 kali, kemudian dihitung nilai rata-rata dan nilai standar deviasi dari kesepuluh eksperimen per topik.

Nilai rata-rata dan standar deviasi *perplexity* yang telah didapatkan kemudian divisualisasikan menggunakan *bar chart* di dalam *Microsoft Excel*. Nilai rata-rata dan standar deviasi paling rendah menunjukkan model yang lebih baik. Kode yang digunakan untuk mengimplementasikan tahap penentuan jumlah topik tertera pada Kode 23.

```
from gensim.models import ldamodel
import logging

jumlahTopik = 4
jumlahPasses = 35

dictionary.filter_extremes(no_below=3, no_above=0.1)

info = logging.basicConfig(filename='LDA-
Koheren/log/topik_log_without_' + str(jumlahTopik) +
'topics_' + str(jumlahPasses) +'passes_' +
str(no_uji_model) +'_Fr-below3'+'.txt', filemode='w',
format='%(message)s', level=logging.INFO)

lda = ldamodel.LdaModel(corpus,
num_topics=jumlahTopik, id2word=dictionary,
passes=jumlahPasses)
```

Kode 23 Penentuan Jumlah Topik

5.4.4.3 Menyimpan Model

Setelah mendapatkan model terbaik berdasarkan jumlah *passes* yang sesuai serta rata-rata dan standar deviasi dari nilai *perplexity* setiap model, kemudian model disimpan dalam format *.model*. Kode yang digunakan untuk menyimpan model sesuai dengan Kode 24.

```
# Save the model
lda.save('LDA-Koheren/model/hasil_lda_withoutstem' +
str(jumlahTopik) + 'topics_' + str(jumlahPasses)
+'passes_' +  str(no_uji_model) +'_Fr-below3'+
'.model')
```

Kode 24 Menyimpan Model

5.5 Proses Data dengan Gaussian *Latent Dirichlect Allocation* (GLDA)

Tahap proses data dengan metode *Gaussian Latent Dirichlect Allocation (GLDA)* merupakan tahapan membentuk model dan melakukan eksperimen model untuk mendapatkan model yang paling baik dengan kata-kata yang mirip untuk setiap topik di dalam modelnya.

5.5.1 Pemrosesan *Corpus* dan *Dictionary*

Tahap awal yang dilakukan untuk memodelkan topik menggunakan metode *Gaussian LDA* adalah mendefinisikan *corpus* dan *dictionary* yang akan diproses dalam pemodelan kedepannya. Data yang dimasukkan sebagai *input* untuk memproses *corpus* dan *dictionary* adalah data dari hasil pra-proses data yaitu semua data baik yang telah ditentukan TF-IDF dan dibentuk frasanya.

Pertama yang dilakukan adalah memasukkan *file* data hasil pra-proses data seperti yang ditunjukkan dalam Kode 25. Pada kode 25 data *input* dimasukkan ke dalam variable *'docs'* yang kemudian akan digunakan untuk proses selanjutnya.

```
if __name__ == "__main__":

    f = 'input\phrase_withstem_batasan2.txt'
    with open(f, 'r', encoding="utf8") as fi:
        docs = fi.read().splitlines() #  These are all
cleaned out
        fi.close()
```

Kode 25 Memuat *File* Data yang Diproses

Selanjutnya, dilakukan inisasi variable untuk menyimpan *corpus* dan *dictionary* seperti yang ditunjukkan pada kode 26. Inisasi variable *preprocess* dilakukan untuk *checking* apakah pra-proses data telah dilakukan atau tidak. Inisasi *preprocess, corpus* dan *vocab* dilakukan pada fungsi *_init_* dan selanjutnya akan digunakan pada proses-proses yang lain. Kemudian, inisiasi variable *temp_corpus* dilakukan pada fungsi *process_corpus* yang akan digunakan untuk menyimpan data *corpus* sementara selama pemrosesan *corpus.*

```
self.preprocess = False
self.vocab = set([])
self.corpus = None

#inisiasi_corpus
temp_corpus = defaultdict(dict)
```

Kode 26 Inisiasi variable *corpus, vocab dan preprocess*

Selanjutnya data *input* diproses menjadi *corpus* dan *dictionary*. Pemrosesan ini dilakukan dalam fungsi yang membutuhkan parameter berupa data *input* yang telah disimpan di dalam variable *'docs'*. Selanjutnya setiap kata yang ada di dalam data di-*shuffle* untuk dikelompokkan secara acak, hal ini dikarenakan *corpus* yang akan diproses berbentuk *map* dimana setiap *keys* berupa nomor dokumen dan nilai berupa kata dan topikknya. Kode yang digunakan untuk pemrosesan *corpus* dan *dictionary* ditunjukkan dalam kode 27.

```
def process_corpus(self, documents):
        """
        Tokenizes documents into dict of lists of tokens
        :param documents: expects list of strings
        :return: dict{document ID: list of tokens
        """
        if not self.preprocess:
            temp_corpus = defaultdict(dict)
            random.shuffle(documents)  # making sure
topics are formed semi-randomly
            for index, doc in enumerate(documents):
                words = doc.split()
                temp_corpus[index]['words'] = words
                temp_corpus[index]['topics'] =
np.empty(len(words))  # Random topic assign
                for word in words:
                    self.vocab.add(word)
            self.corpus = temp_corpus
            print ("Done processing corpus with {}
documents".format(len(documents)))

        else: # Docs are tokenized and such, just add it
into class
            temp_corpus = defaultdict(dict)
            for idx, doc in enumerate(documents):
                temp_corpus[idx]["words"] = doc
                temp_corpus[idx]["topics"] =
np.empty(len(doc))
                for word in words:
                    self.vocab.add(word)
            self.corpus = temp_corpus
```

Kode 27 Pemrosesan *Corpus* dan *Dirctionary*

Hasil pemrosesan *corpus* disimpan di dalam variable *self.corpus* dan hasil pemrosesan *dictionary* disimpan di dalam variable *self.vocab* yang kemudian akan digunakan pada proses-proses selanjutnya.

5.5.2 Pemrosesan Vektor Kata

Tahap yang akan dilakukan setelah *corpus* terbentuk adalah melakukan pemrosesan vektor kata. Dalam tahap ini akan dilakukan *load* daftar vektor kata dan mengubah kata-kata yang ada di dalam *corpus* menjadi vektor dengan daftar vektor kata yang dimiliki. Dalam penelitian ini vektor kata yang digunakan adalah hasil *training word embedding* yang dikembangkan oleh Purwarianti [15]. Dalam proses ini dibutuhkan 2 *file* yaitu *file* yang menyimpan model vektor kata dan *file* data atau *corpus* yang akan diproses.

Pertama, dilakukan *load file* penyimpanan *model* vektor kata. Seperti pada kode 28. *Model* vektor kata selanjutnya disimpan di dalam variable *wordvec_fileapth*.

```
if __name__ == "__main__":

wordvec_fileapth = "D:\KULIAH\8th SEMESTER\Tugas
Akhir\Dokumen\Code\GLDA\dir_data\w2vec.txt"
```

Kode 28 Memuat *File Model* Vektor Kata

```
from gensim.models import KeyedVectors

#inisiasi wordvektor
self.word_vecs = {}
self.word_vec_size = None

#load model word vektor
self.wvmodel =
KeyedVectors.load_word2vec_format(fname=filepath,
limit=200000)

#inisiasi jumlah kata
useable_vocab = 0
unusable_vocab = 0
```

Kode 29 Inisiasi variable word_vecs, word_vec_size, useable_vocab dan unusebale_vocab

Selanjutnya yaitu melakukan pemrosesan kata menjadi vektor kata, proses ini dilakukan dalam fungsi *process_wordvectors* dengan parameter yang dibutuhkan yaitu keberadaan *directory* penyimpanan model vektor kata. Dalam proses ini dilakukan *load* model vektor kata dengan *library genism.models* pada

modul *KeyedVectors.* Setelah itu dilakukan inisiasi variable yang menyimpan vektor dari semua kata yang ada di dalam *corpus,* besar dimensi dari vektor kata, jumlah kata yang dapat dikonversi ke dalam vektor dan jumlah kata yang tidak dapat dikonversi menjadi vektor. Kode yang digunakan seperti yang ditunjukkan dalam kode 29.

```
from gensim.models import KeyedVectors

def process_wordvectors(self, filepath=None):
        if filepath:
            print ("Processing word-vectors, this takes a
moment")
            self.wvmodel =
KeyedVectors.load_word2vec_format(fname=filepath,
limit=200000)
            useable_vocab = 0
            unusable_vocab = 0
            self.word_vec_size = self.wvmodel.vector_size

            for word in self.vocab:
                try:
                    self.word_vecs[word] = self.wvmodel[word]
#match between word on corpus and word embedding
                    useable_vocab += 1
                except KeyError:
                    self.word_vecs[word] = self.wvmodel['unk']
                    unusable_vocab += 1

            print ("There are {0} words that could be
converted to word vectors in your corpus \n" \
                   "There are {1} words that could NOT be
converted to word vectors".format(useable_vocab,

unusable_vocab))
        else:
            useable_vocab = 0
            unusable_vocab = 0
            self.word_vec_size = self.wvmodel.vector_size

            for word in self.vocab:
                try:
                    self.word_vecs[word] = self.wvmodel[word]
                    useable_vocab += 1
                except KeyError:
                    self.word_vecs[word] = self.wvmodel['unk']
                    unusable_vocab += 1

            print ("There are {0} words that could be
converted to word vectors in your corpus \n" \
                   "There are {1} words that could NOT be
converted to word vectors".format(useable_vocab,

unusable_vocab))
```

Kode 30 Proses Konversi Kata ke Vektor

Selanjutnya yaitu melakukan konversi kata-kata di dalam *corpus* yang disimpan di dalam *self.vocab* menjadi vektor yang didapatkan dari *self.wvmodel*. Semua kata yang tidak terdapat di dalam *model* akan dihitung dan dimasukkan ke dalam variable *unusable_vocab* dan diubah menjadi vektor kata dari *'unk'* yaitu vektor untuk kata-kata yang tidak diketahui atau *unknown*. Proses pengonversian kata menjadi vektor ditunjukkan dalam kode 30.

Selanjutnya vektor kata disimpan di dalam variable *self.word_vec* yang akan digunakan pada proses-proses berikutnya.

5.5.3 Inisalisasi Parameter

Setelah pemrosesan vektor kata selanjutnya dilakukan inisiasi untuk semua parameter yang dibutuhkan untuk melakukan pemodelan dengan *Gaussian LDA*. Parameter yang dibutuhkan yaitu: *prior mean, prior covariance, sample mean, sample covariance, posterior mean dan posterior covariance*. *Prior mean* dan *prior covariance* merupakan parameter yang menyimpan nilai *mean* dan *covariance* dari pemodelan sebelum dilakukan *gibbs sampling, sample mean* dan *covariance* merupakan variable yang digunakan untuk menyimpan nilai *mean* dan *covariance* dari pemodelan pada saat melakukan *sampling, posterior mean* dan *covariance* merupakan parameter yang digunakan untuk menyimpan nilai akhir dari *mean* dan *covariance* setelah melalui *gibbs sampling*. Inisiasi parameter dilakukan dalam fungsi *init()*.

```
self.process_corpus(self.corpus)
self.process_wordvectors(self.wordvecFP)
self.priors = Wishart(self.word_vecs)
self.doc_topic_CT = np.zeros((len(self.corpus.keys()),
self.numtopics))

#shuffle topik kata
self.word_topics = {word:
random.choice(range(self.numtopics)) for word in
self.vocab}
```

Kode 31 Melakukan *Shuffle* Kata ke dalam Topik

Pertama dilakukan inisiasi untuk nilai paremeter *prior mean* dan *covariance.* Inisiasi ini dimulai dengan melakukan *shuffle* pada topik kata yang ada di dalam *self.vocab.*

Kemudian dilakukan inisiasi variable yang akan menyimpan nilai *prior mean* untuk sementara, kemudian dilakukan penjumlahan pada semua vektor kata yang ada di dalam *corpus* yang disimpan di dalam variable *mu_0,* nilai dari *mu_0* dibagi dengan jumlah seluruh kata yang ada di dalam *corpus* yang disimpan di dalam variable *count.* Hasil bagi dari *mu_0* dan *count* kemudian disimpan di dalam variable *self.prior.mu* yang merupakan variable untuk menyimpan nilai *prior mean.*

```
mu_0 = np.zeros(self.word_vec_size)
count = 0
for docID in self.corpus.keys():
      for i, word in
enumerate(self.corpus[docID]['words']):
        self.corpus[docID]['topics'][i] =
        self.word_topics[word]
        mu_0 += self.word_vecs[word]
        count += 1
self.priors.mu = mu_0 / float(count)
```

Kode 32 Inisiasi Nilai untuk Parameter *prior mean*

Selanjutnya melakukan inisasi parameter *prior covariance.* Untuk melakukan inisiasi *prior covariance* dibutuhkan variable ukuran dari vektor kata yang telah disimpan dalam *self. word_vec_size.* Hasil dari inisiasi *prior covariance* kemudian disimpan di dalam variable *self.prior.psi*. Inisiasi dilakukan seperti ditunjukkan dalam kode 33.

```
self.priors.psi = .01 * np.identity(self.word_vec_size)
```

Kode 33 Inisiasi Nilai untuk Paremeter *prior covariance*

Selanjutnya dilakukan inisiasi *sample mean* dan *covariance,* pertama dilakukan inisiasi untuk variable yang akan menyimpan nilai *covariance* dan *mean sample* dimana untuk setiap jumlah topik yang ditentukan masing-masing akan memiliki nilai *sample mean* dan *covariance.*

```
for k in range(self.numtopics):
  self.topic_params[k]["Topic Sum"] =
np.zeros(self.word_vec_size)
  self.topic_params[k]["Topic Mean"] = mu_0[k]
  self.topic_params[k]["Topic Cov"] =
np.zeros((self.word_vec_size, self.word_vec_size))
```

Kode 34 Inisiasi Parameter untuk Masing-masing Topik

Kemudian nilai *sample mean* dari masing-masing topik diinisasi dengan menggunakan nilai *prior mean*nya dan disimpan di dalam variable *simple_mu.* Kemudian *sample covariance* diinisasi dengan meng-*assign* seluruh nilai dalam matriks menjadi 0, lalu dihitung dengan menggunakan *np.outer* dengan nilai matriks sama dengan nilai vektor kata dikurangi dengan *sample_mu.*

```
co_variances = [np.zeros((self.word_vec_size,
self.word_vec_size)) for _ in range(self.numtopics)]
for docID in self.corpus.keys():
   for topic, word in zip(self.corpus[docID]['topics'],
   self.corpus[docID]['words']):
     topic = int(topic)
     wv = self.word_vecs[word]
     sample_mu = self.topic_params[topic]["Topic Mean"]
     self.doc_topic_CT[docID, topic] += 1.
     self.topic_params[topic]['Topic Sum'] += wv
     co_variances[topic] += np.outer(wv - sample_mu, wv -
sample_mu)
```

Kode 35 Inisiasi Parameter *sample mean* dan *covariance*

5.5.4 Topic Modeling dengan *Gaussian LDA*

Setelah dilakukan pemrosesan *corpus, dictionary* dan vektor kata serta melakukan inisiasi untuk semua parameter yang dibutuhkan, selanjutnya dilakukan pemodelan topik dengan menggunakan metode *Gaussian LDA.* Dalam peneilitan ini beberapa *input* yang harus diperhatikan dalam membentuk model topik yaitu input data yang akan digunakan yang telah dibentuk menjadi frasa dan melalui TF-IDF, batasan atas dan bawah dari data yang akan digunakan, dan jumlah topik. Dari ke tiga nilai *input* ini nantinya dilakukan eksperimen untuk mendapatkan model yang terbaik.

Pada tahap pemodelan topik dengan *Gaussian LDA* terdapat beberapa langkah yang akan dilewati yaitu, melakukan *sampling* dengan menggunakan *gibbs sampling,* menghitung log probabilitas *student T distribution* untuk masing-masing kata, menghitung ulang kembali nilai dari masing-masing parameter yang dibutuhkan untuk *Gaussian LDA*.

5.5.4.1 *Gibbs Sampling*

Pada tahap ini dilakukan *sampling* untuk meng-*assign* kata-kata yang ada di dalam *corpus* ke dalam topik-topik yang baru. Metode *sampling* yang digunakan dalam metode *Gaussian LDA* adalah *gibbs sampling.* Dalam melakukan *sampling* ini pertama yang dilakukan adalah melakukan inisiasi untuk variable yang akan menyimpan kata dan topik dari kata tersebut.

```
for docID in self.corpus.keys():
  for idx in range(len(self.corpus[docID]['words'])):
     word = self.corpus[docID]['words'][idx]
     current_topic = self.corpus[docID]['topics'][idx]
```

Kode 36 Inisiasi Variable *word* dan *current_topic*

Kemudian dilakukan penghitungan ulang nilai dari parameter untuk *Gaussian LDA* yang akan dijelaskan pada subbab berikutnya. Kemudian setelah melakukan perhitungan ulang nilai parameter dilakukan inisiasi untuk variable yang akan menyimpan nilai log probabilitas dari masing-masing kata. Setelah itu dilakukan perhitungan log probabilitas untuk semua kata dalam *corpus* yang akan dijelaskan pada subbab berikutnya.

Setelah perhitungan log probability dilakukan *gibbs sampling* dengan menggunakan nilai *log probability* sebagai input dari *gibbs sampling*. *Gibbs sampling* dilakukan untuk semua kata dalam topik yang telah ditentukkan jumlahnya. Hasil dari *gibbs sampling* disimpan dalam *log_posterior* yang berbentuk *array* dengan nilai berupa hasil log kata untuk masing-masing topik. Kode yang digunakan untuk melakukan *gibbs sampling* ditunjukkan dalam kode 37.

```
self.recalculate_topic_params(word, int(current_topic),
int(docID), "-")
log_posterior = np.zeros(self.numtopics)
for k in range(self.numtopics):
  log_pdf = self.draw_new_wt_assgns(word, k)
  Nkd = self.doc_topic_CT[docID, k]
#Gibbs Sampling
  log_posterior[k] = np.real(np.log(Nkd + self.alpha) +
log_pdf)
```

Kode 37 Melakukan *Gibbs Sampling*

Kemudian, kata-kata yang ada di dalam *corpus* yang telah melalui *gibbs sampling* akan diproses untuk mengetahui topik mana yang memiliki nilai *log_posterior* yang paling tinggi, topik dengan *log_posterior* yang tinggi akan di-*assgin* ke dalam kata tersebut. Kode yang digunakan untuk mengambil nilai tertinggi ditunjukkan dalam kode 38.

```
max_log_posterior = np.max(log_posterior)
log_posterior -= max_log_posterior
normalized_post = np.exp(log_posterior -
np.log(np.sum(np.exp(log_posterior))))
```

Kode 38 Mengambil Nilai Tertinggi untuk *log_posterior*

Jika topik awal dari kata sama dengan topik yang memiliki nilai *log_posterior* tertinggi maka tidak dilakukan *reassignment* untuk kata tersebut, dan jika topik awal kata tidak sama dengan topik yang memiliki *log_posterior* teringgi maka akan dilakukan pergantian pada topik dari kata. Kode yang digunakan untuk meng-*assign* topik ke dalam kata ditunjukkan pada kode 39.

```
if MULTINOMIAL_TOPIC_SELECTION:
   new_topic = np.argmax(np.random.multinomial(1,
pvals=normalized_post))
else:
   new_topic = np.argmax(normalized_post)

if not ASSIGN_NEW_TOPICS:
   new_topic = current_topic
```

Kode 39 Memasukkan Kata ke dalam Topik

Kemudian setelah *gibbs sampling* selesai untuk semua kata, dilakukan update kata dalam topik dan perhitungan ulang parameter yang dibutuhkan untuk *Gaussian LDA*.

```
self.corpus[docID]['topics'][idx] = new_topic
self.recalculate_topic_params(word, new_topic, docID, "+")
```

Kode 40 Melakukan *Update* Nilai Parameter dan Jumlah Kata dalam Topik

5.5.4.2 *Recalculate Paremeter*

Pada tahap ini dilakukan perhitungan ulang untuk nilai dari masing-masing parameter yang dibutuhkan untuk *Gaussian LDA* sebelum dan setelah dilakukannya *gibbs sampling.* Alasan dari perhitungan ulang nilai parameter ini adalah dikarenakan, jika terdapat update dari jumlah kata untuk masing-masing topik maka akan mempengaruhi nilai *mean* dan *covariance* dari semua topik.

```
UPDATE_COUNT = True
if UPDATE_COUNT:
  self.update_document_topic_counts(word, topic, docID,
operation)

#fungsi update_document_topic_counts
def update_document_topic_counts(self, word, topic, docID,
operation):

   if operation == "-":
        self.topic_params[topic]["Topic Sum"] -=
        self.word_vecs[word] #jumlah vektor kata dalam
        topik minus(-) current_kata
        self.doc_topic_CT[int(docID), int(topic)]-= 1.

   if operation == "+":
        self.topic_params[topic]["Topic Sum"] +=
        self.word_vecs[word]
        self.doc_topic_CT[docID, topic] += 1.
```

Kode 41 Melakukan *Update* Jumlah Kata di dalam Topik

Dalam perhitungan ulang nilai paremeter terdapat 2 tahap yang dilalui yaitu: pengurangan jumlah kata yang terdapat di dalam topik dan penambahan jumlah kata yang terdapat dalam topik. Pertama dilakukan update untuk jumlah kata yang ada di dalam

topik. Update kata dalam topik dilakukan dalam fungsi *update_document_topic_counts* dengan parameter berupa kata, topik, ID dari dokumen dan operasi yang digunakan, apakah penambahan atau pengurangan. Kode yang digunakan seperti ditunjukkan dalam kode 41.

Kemudian, setelah dilakukan *update* kata dalam topik selanjutnya dilakukan update pada nilai parameter. Jika dilakukan pengurangan pada kata di dalam topik maka topik *covariance* akan diubah menjadi hasil kurang dari pengurangan vektor kata dengan topic *mean* sebelumnya dan perkalian dari *scale recursive downdate.* Kode yang digunakan ditunjukkan pada kode 42.

```
if operation == "-":
   if UPDATE_DISTS:
      wv = self.word_vecs[word]
      mu = self.topic_params[topic]["Topic Mean"]

      centered = wv - mu
      centered *= np.sqrt((kappa_k + 1.) / kappa_k)
      self.topic_params[topic]["Topic Cov"] -=
np.outer(centered, centered)

       sample_mean_K = self.topic_sample_mean(topic,
topic_count)
       topic_mean = ((self.priors.kappa * self.priors.mu)
+ (topic_count * sample_mean_K)) / kappa_k
```

Kode 42 *Update* Nilai Parameter untuk Pengurangan Kata

```
else:  # operation == "+":
   sample_mean_K = self.topic_sample_mean(topic,
topic_count)
   topic_mean = ((self.priors.kappa * self.priors.mu) +
(topic_count * sample_mean_K)) / kappa_k  # Mu_k

   if UPDATE_DISTS:
      centered = (self.word_vecs[word] - topic_mean)
      centered *= np.sqrt(kappa_k / (kappa_k - 1.))
      self.topic_params[topic]["Topic Cov"] +=
np.outer(centered, centered)
```

Kode 43 *Update* Nilai Parameter untuk Penambahan Jumlah Kata

Jika dilakukan penambahan pada kata di dalam topik maka topik *covariance* akan diubah menjadi hasil tambah dari pengurangan *topik mean* yang baru dengan vektor kata dan

scale recursive update. Kode yang digunakan ditunjukkan pada kode 43.

```
def topic_sample_mean(self, topic, topic_count):
        scaled_topic_mean =
        self.topic_params[topic]["Topic Sum"] / \
                              float(topic_count)
        if topic_count > 0
        else np.zeros(self.word_vec_size)

        return scaled_topic_mean
```

Kode 44 *Update* Nilai Parameter *sample mean*

Untuk *topic mean* baik melalui pengurangan atau penambahan kata dalam topik ubah dengan menggunakan hasil bagi dari jumlah seluruh vektor kata dibagi dengan jumlah kata di dalam topik. Kode yang digunakan untuk menjalankan fungsi ini ditunjukkan pada kode 44.

5.5.4.3 Perhitungan *Log Probability Density*

```
cov_det = self.topic_params[topic]["Chol Det"] # covariance
determinan
Nk = self.topic_params[topic]["Topic Count"] # count of topic

centered = self.word_vecs[word] -
self.topic_params[topic]["Topic Mean"]
d = self.word_vec_size  # dimensionality of word vector
kappa_k = self.topic_params[topic]["Topic Kappa"]

scaleT = np.sqrt((kappa_k + 1.) / kappa_k * (self.priors.nu -
d + 1.))
nu = self.priors.nu + Nk - d + 1.

cov_inv = self.topic_params[topic]["Topic Inv"]
cov_inv = centered.T.dot(cov_inv).dot(centered)
cov_inv *= scaleT

cov_det = self.topic_params[topic]["Topic Det"]

a = gammaln((nu + d) / 2.)
b =  (gammaln(nu / 2.) + (d / 2.) * (np.log(nu) + np.log(pi))
+ (0.5 * cov_det) + ((nu + d) / 2.) * np.log(1. + cov_inv/nu))

# Log Multivariate T – PDF
return a-b
```

Kode 45 Menghitung *log probability T Distribution*

Dalam tahap ini dilakukan perhitungan *log probability density* untuk *student T distribution*. *Log probability density* digunakan

untuk menghitung *gibbs sampling*. Untuk menghitung log probability dilakukan dengan menggunakan fungsi *draw_new_wt_assgns* yang membutuhkan parameter *input* berupa kata, topik, *new_document* dan *model word vector*. Kode yang digunakan untuk melakukan perhitungan *log probability* ditunjukkan dalam kode 45.

5.5.5 Pendokumentasian Ekperimen

Pendokumentasian eksperimen sangat diperlukan dalam untuk mencatat setiap kejadian yang terjadi selama proses pembentukan model topik. Catatan yang dibutuhkan yaitu nilai dari topik *mean* dan *covariance* serta selutuh variable yang ada dalam *self*. Kode yang digunakan untuk melakukan *logging* ditunjukkan dalam Kode 46.

```
def fit(self, iterations=1, init=True):
  if init:
  self.init()
  init = False
  print ("Starting fit")
  for i in range(iterations):
     self.sample()
     print ("{0} iterations complete".format(i))
     for k in range(self.numtopics):
         for param in ("Topic Mean", "Topic Cov"):
         results_file =\"GLDA
TFIDF/hasil_tester/hasil_fasttext_pujangga/p_w/{3}iter{0}topic
{1}_{2}_{4}.txt".format(i, k, param, self.run_name,
self.numtopics)
open(results_file, 'w')
np.savetxt(results_file, self.topic_params[k][param])
```

Kode 46 Melakukan *Logging*

Contoh kode yang digunakan untuk menyimpan variable *self.doc_topic_CT* dalam ditunjukkan dalam kode 47.

```
import pickle

filehandler = open('GLDA-
TFIDF/simpanan/p_w/{0}_doc_topic_CT_{1}'.format(self.run_name,
self.numtopics), 'wb')
pickle.dump(self.doc_topic_CT, filehandler)
```

Kode 47 Menyimpan Model

```
file = open('GLDA-
TFIDF/topic_word/p_w/tfidf_with_{0}_{1}.csv'.format(self.run_n
ame, self.numtopics),'a')
                file.write('{0}, {1},
{2},{3}\n'.format(new_topic,word, max_log_posterior, docID))
```

Kode 48 Menyimpan Kata, Probabilitas Kata, Topik dan Dokumen

Selain itu, data yang disimpan dalam tahap ini adalah data daftar kata, probabilitas, topik dan dokumen dari masing-masing kata dalam topik yang dihasilkan dari setiap eksperimen. Kode yang dihgunakan untuk menyimpan data ini ditunjukkan dalam kode 48. Pengkodean ini dilakukan dalam fungsi *sample()* setelah meng-*assign* kata ke dalam topik.

5.5.6 Ekperimen Pemodelan Topik dengan *Gaussian LDA*

Dalam tahap ekperimen, dilakukan pemodelan topik dengan metode *Gaussian Latent Dirichlet Allocation* yang bertujuan untuk membentuk model terbaik. Ekperimen dilakukan pada

1. Input data yang akan diproses yaitu data yang telah dibentuk frasa dan data tanpa frasa
2. Jumlah batas atas dan bawah, jumlah minimal dan maksimal kemunculan, dari kata yang akan diguakan setelah melalui penghitungan TF-IDF
3. *Input parameter*, yaitu jumlah iterasi dan topik.

Berdasarkan hal ini, eksperimen akan dilakukan eksperimen untuk penentuan jumlah topik.

5.5.6.1 Penentuan Jumlah Iterasi

Penentuan jumlah iterasi agar dapat menghasilkan jumlah yang tepat dilakukan dengan melalui beberapa eksperimen, dalam penelitian ini untuk menentukan jumlah iterasi dilakukan 30 kali iterasi dengan jumlah topik sebesar 13 topik. Berdasarkan ekperimen ini, semua nilai *perplexity* yang muncul dari setiap topik dan skenario akan dicatat untuk selanjutnya dianalisa tren

nilai *perplexity* untuk setiap iterasi. Kode yang digunakan untuk menginisiasi jumlah iterasi ditunjukkan pada kode 49.

```
g = Gauss_LDA(3, docs, wordvec_fileapth,
run_name='below_1')
g.fit(10)
```

Kode 49 Inisiai Jumlah Iterasi dalam Ekperimen

Semua hasil ekperimen didokumentasikan dengan menyimpan seluruh nilai dari *variable self* pada setiap iterasi untuk selanjutnya menghitung nilai *perplexity* pada setiap iterasi yang dilakukan. Kode yang digunakan untuk menghitung nilai perplexity dari masing-masing iterasi akan dijelaskan pada subbab berikutnya.

5.5.6.2 Penentuan Jumlah Topik

Untuk melakukan ekperimen penentuan jumlah topik dilakukan percobaan dengan beberapa jumlah topik yaitu dimulai dari 4, 7, 10, 11, 13, 15, 25 dan 35 topik. Penggunaan beberapa jumlah topik dilakukan untuk mendapatkan nilai *perplexity* yang paling rendah. Nilai *perplexity* yang semakin rendah menunjukkan kemampuan probabilitas model yang semakin baik.

```
final_prob_w = dict()
readdoc = 1
for doc in docs:
  token_word_topic_probability =
  extract_topics_new_doc(doc,wv_model,jlhtopic,run_name)
  dict_z_d = dict()
  sum_z_d = dict()
  sum_topic = 0
  for word,prob,num_topic in token_word_topic_probability:
        if num_topic not in sum_z_d:
           sum_z_d[num_topic] = 1
        else:
           sum_z_d[num_topic] += 1
           sum_topic += 1

  prob_w = dict()
  for word,prob,num_topic in token_word_topic_probability:
        prob_w[word] = np.exp(prob) *
        (sum_z_d[num_topic]/sum_topic) * 1/722
        if word not in final_prob_w:
           final_prob_w[word] = prob_w[word]
        else:
           final_prob_w[word] += prob_w[word]

   print('Calculate Probability Word in document : ' +
str(readdoc))
   readdoc += 1
with open('GLDA-Phrase/topic_word/p_wo_' + jlhtopic + '_'
+ run_name + '.csv', 'w') as final_prob:
    for word ,prob_w in final_prob_w.items():
        print("Probability word " + word + " : " +
str(prob_w))
        final_prob.write(word + ',' + str(prob_w) + '\n')
```

Kode 50 Menghitung Probabilitas Kata terhadap Dokumen

Untuk menghitung nilai *perplexity* dilakukan perhitungan probabilitas kata terhadap *corpus* yang dihasilkan dari perhitungan log probabilitas kata terhadap topik dikalikan dengan probabilitas topik terhadap dokumen. Setelah mendapatkan nilai probabilitas kata terhadap dokumen kemudian dilakukan perhitungan nilai *perplexity.* Untuk menghitung *perplexity* dibutuhkan *load model* yang dihasilkan dari pendokumentasian setiap eksperimen pada tahap pemodelan topik.

Nilai rata-rata *perplexity* yang telah didapatkan kemudian divisualisasikan menggunakan *bar chart* di dalam *Microsoft*

Excel. Nilai rata-rata paling rendah menunjukkan model yang lebih baik. Kode yang digunakan untuk mengimplementasikan tahap penentuan jumlah topik tertera pada Kode 50.

Dalam kode 51, dilakuakan *load file* yang menyimpan jumlah probabilitas masing-masing kata terhadap dokumen. Kemudian variable *b* merupakan variable yang menyimpan nilai jumlah *term-frequency* untuk semua kata dan variable *a* digunakan untuk menyimpan jumlah nilai probabilitas kata yang telah di log, variable *c* menyimpan nilai perplexity sebelum dijadikan *exponensial* dan variable *d* menyimpan nilai akhir dari *perplexity.*

```
jlhtopic = 4
run_name = 'below_1_'
f = 'topic_word/p_w_below_1/p_w_' + '_' + str(jlhtopic) +
'_' + run_name + '.csv'
with open(f, 'r', encoding="utf8") as fi:
    word_prob = fi.read().splitlines()
    fi.close()

a = 0
b = 9299

z = 0
for line in word_prob:
    a += np.log10(float(line.split(',')[1]))
c = -(a/b)
d = np.exp(c)
d
```

Kode 51 Menghitung Nilai *Perplexity*

5.6 Validasi Model Topik *LDA*

Setelah melakukan perhitungan dan analisa nilai *perplexity* dan mendapatkan model yang dianggap baik berdasarkan nilai *perplexity*, selanjutnya dilakukan validasi model topik dengan menggunakan nilai *topic coherence*.

```
# Load the model
file_dir =
"model\hasil_lda_withstem4topics_20passes_1_biasa.model"
model = models.ldamodel.LdaModel.load(file_dir)
```

Kode 52 Melakukan *load model LDA*

Sebelum melakukan validasi topik dengan perhitungan rata-rata *coherence score*, dilakukan tahap *load* model. Model yang di *load* merupakan model yang telah disimpan pada tahap sebelumnya. Kode yang digunakan untuk *load* model ditunjukkan dalam kode 52.

Selain *load* model, juga dilakukan *load file* data yang digunakan untuk membuat model seperti yang ditunjukkan dalam kode 53.

```
# Load the file
keluhan = []
corpus = []
file = open('input/output_with_stem.txt', 'r')
lines = file.read().split('\n')
for line in lines:
    words = line.split(',')
    datasentence = []
    for word in words:
        datasentence.append(word)
    keluhan.append(datasentence)
```

Kode 53 Melakukan *load File*

Kemudian setelah load data, dilakukan penentuan jumlah batasan atas dan bawah sesuai dengan batasan yang dimiliki model seperti yang ditunjukkan dalam kode 53.

```
dictionary = Dictionary(keluhan)
dictionary.filter_extremes(no_below=10, no_above=0.3)
for curhat in curhatan:
    corpus.append(dictionary.doc2bow(curhat))
```

Kode 54 Melakukan Penentuan Jumlah Batasa Atas dan Bawah

5.6.1 Rata-rata *Coherence Score*

Setelah melakukan *load model, data* dan menentukan batasan selanjutnya dilakukan perhitungan rata-rata *coherence score* dari setiap model. Hal ini dilakukan dengan menjumlahkan seluruh nilai *coherence score* masing-masing topik dan dibagi dengan jumlah topik yang ada. Kode yang digunakan untuk melakukan perhitungan rata-rata *coherence score* ditunjukkan dalam kode 55.

```
cm = CoherenceModel(model=model, corpus=corpus,
coherence='u_mass')
print(cm.get_coherence())
```

Kode 55 Menghitung Rata-Rata *Coherence Score*

5.7 Validasi Model Topik *Gaussian LDA*

Setelah melakukan perhitungan dan analisa nilai *perplexity* dan mendapatkan model yang dianggap baik berdasarkan nilai *perplexity*, selanjutnya dilakukan validasi model topik dengan menggunakan nilai *topic coherence*. Validasi dengan *topic coherence* akan dilakukan dengan menggunakan *pointwise mutual information*.

Untuk melakukan validasi dengan menggunakan metode *pointwise mutual information.* Dibutuhkan data berupa kata yang akan dijadikan sebagai *bank* kata. Dalam penelitian ini, daftar kata yang akan digunakan untuk melakukan validasi diambil dari hasil *scrapping* data *post* atau status pada 23 halaman di *facebook.* Data yang digunakan sebanyak 54264 *post*.

Selanjutnya, tahap yang dilakukan adalah dengan memilih 5 kata dengan probabilitas tertinggi pada 5 topik dengan probabilitas tertinggi dari model terbaik yang dihasilkan pada tahap sebelumnya. Pemilihan kata dan topik ini dilakukan dengan menggunakan *Microsoft excel* dari data hasil kata dan probabilitas kata yang telah disimpan pada tahap sebelumnya. Daftar 5 kata dengan probabilitas tertinggi terdapat dalam lampiran D-1.

```
word_list =
['menit','pagi','wib','malam','pkl','salat','jamaah','ibad
ah','sejahtera','umroh','barat','bangkal','kota','terap','
seberang','rokok','makan','minum','ikan','enak','airport',
'airlines','maskapai','cargo','jetstar']

with open('pmi.txt', errors='ignore', encoding='utf-8') as
openfile:
     docs = openfile.read().splitlines()
     openfile.close()
```

Kode 56 *Load Data* untuk *Pointwise Mutual Information*

Sebelum melakukan perhitungan dengan *pointwise mutual information* pertama dilakukan *load* data yang akan digunakan untuk melakukan perhitungan, data yang dimasukkan yaitu data *corpus* dan *list* 5 kata dengan probabilitas tertinggi untuk setiap topik. Kode yang digunakan untuk melakukan *load* data ditunjukkan pada kode 56.

5.7.1 Menghitung Jumlah Kemunculan Kata dalam *Corpus*

Untuk melakukan validasi dengan menggunakan *pointwise mutual information* pertama dilakukan penghitung jumlah kemunculan seluruh kata dalam *list* 5 kata dengan probabilitas tertinggi dalam topik. Kata-kata ini dibagi menjadi 2 yaitu kata *x* dan kata *y,* kata *x* merupakan kata yang akan dicari nilai PMInya terhadap kata *y* yang merupakan 4 kata lain yang ada di dalam topik. Pertama dilakukan inisiasi *variable* yang akan menyimpan nilai kata *x* dan *y,* total *co-occurance* kata *x* dan *y,* total kata, dan total dokumen.

```
tf_word1 = 0
tf_word2 = 0
total_coocurence = 0
total_doc = 0
total_word = 0
```

Kode 57 Inisiasi *Variable*

Selanjutnya, dilakukan perhitungan jumlah kemunculan kata dalam *corpus*. Kode yang digunakan untuk melakukan perhitungan jumlah kata ditunjukkan dalam Kode 58.

```
for num_topic1 in range(0,len(word_list)):
                print(word_list[num_topic1])
   for num_topic2 in range((num_topic1+1),len(word_list)):
         word1 = word_list[num_topic1]
         word2 = word_list[num_topic2]
         docs = [w.replace(',', ' ') for w in docs]
         for doc in docs:
             find_word1 = 0
             find_word2 = 0

             token = doc.split(' ')
             # print(token)

             if word1 in token:
                find_word1 = 1
                tf_word1 += 1
             if word2 in token:
                find_word2 = 1
                tf_word2 += 1
             if(find_word1 == 1 and find_word2 == 1):
                total_coocurence += 1
             total_doc += 1
             total_word += len(token)
```

Kode 58 Menghitung Jumlah Kata dalam Data

5.7.2 Menghitung *p(x), p(y), p(x, y)* dalam PMI

Setelah melakukan perhitungan jumlah kemunculan kata dalam *corpus,* selanjutnya dilakukan perhitungan untuk masing-masing nilai *p(x)* yang merupakan nilai probabilitas dari kata *x* yang akan dicari nilai PMInya dan *p(y)* yang merupakan probabilitas dari kata *y,* kata lain yang ada di dalam topik. Perhitungan ini dilakukan dengan menggunakan kode 58. Probabilitas kata *x* dan *y* didapatkan dari hasil bagi jumlah kemunculan kata *x* dan *y* dengan jumlah seluruh kata dalam *corpus.* Sedangkan, probabilitas *co-occurance* kata *x* dan *y* didapatkan dari hasil bagi total *co-occurance* dengan jumlah seluruh dokumen dalam *corpus.*

```
px = tf_word1/total_word
py = tf_word2/total_word
pxy = total_coocurence/total_doc
```

Kode 59 Perhitungan untuk Mencari Nilai *p(x), p(y) dan p(x,y)*

5.7.3 Menghitung Kemiripan Kata dengan *Pointwise Mutual Information*

Setelah melakukan perhitungan pada niai *p(x), p(y)* dan *p(x,y)* selanjutnya dilakukan perhitungan untuk mendapatkan nilai PMI dari kata *x.*

```
pmi = np.log10(pxy / (px * py))
print(pmi)
```

Kode 60 Menghitung *Pointwise Mutual Information*

5.8 Pengujian model dengan menggunakan nilai *pointwise mutual information*

Pada tahap ini model terbaik dari ke dua metode, *Gaussian LDA* dan *LDA*, akan diuji dengan menggunakan metode *pointwise mutual information.* Pengujian ini bertujuan untuk mengetahui sejauh mana kemiripan antar kata dalam topik yang dimiliki oleh masing-masing model. Hasil dari perhitungan PMI model dari ke dua metode tersebut dibandingkan untuk mengetahui metode mana yang menghasilkan model dengan kata-kata dalam topik yang lebih mirip.

Seperti dalam tahap validasi *Gaussian LDA* dengan PMI, untuk melakukan pengujian model dengan menggunakan metode *pointwise mutual information* juga dibutuhkan data berupa kata yang akan dijadikan sebagai *bank* kata. Dalam tahap ini, data yang digunakan sama dengan data yang digunakan dalam validasi *Gaussian LDA* yaitu data *scrapping post* dari 23 halaman di *facebook.*

Untuk melakukan uji model, dilakukan pemilihan 5 kata dengan probabilitas tertinggi dari model terbaik *Gaussian LDA* mapun *LDA*. Pemilihan kata ini dilakukan dengan menggunakan *Microsoft excel* dari data hasil kata dan probabilitas kata yang telah disimpan pada tahap sebelumnya. Selanjutnya dilakukan analisis kuantitas berdasarkan hasil PMI dan kualitas berdasarkan kata-kata yang dihasilkan.

Tahapan yang dilalui untuk melakukan perhitungan PMI pada masing-masing kata yang terpilih sebagai 5 kata dengan

probabilitas tertinggi dalam tiap topik sama dengan tahap yang dilakukan dalam validasi *Gaussian LDA*.

5.9 Analisis Hasil

```
def extract_topics_new_doc(self, doc, wv_model):
        assert wv_model.vector_size == self.word_vec_size,
        "word-vector dimensionality does not match trained
        topic" \"distribution
        dimensions({0})".format(self.word_vec_size)
        filtered_doc = []
        nkd = defaultdict(float)
        for word in doc.split():
            try:
                wv_model[word]
                filtered_doc.append(word
                nkd[self.word_topics[word]] += 1.
            except KeyError:
                continue
        print(filtered_doc)
        print ("{} words removed from
doc".format(len(filtered_doc) - len(doc.split())))
        word_topics = []
        c = Counter(self.word_topics.values())
        for word in filtered_doc:
            posterior = []
            for k in range(self.numtopics):
                # print nkd[k]
                prob = self.draw_new_wt_assgns(word, k,
wvmodel=wv_model, new_doc=True) + log(self.alpha + c[k])
                print ("probablity of {0} for word {1}
assigned to topic {2}".format(prob, word, k))
                posterior.append(prob)
            posterior /= np.sum(posterior)

            word_topics.append((word, np.argmax(posterior)))
        return word_topics
```

Kode 61 Menganailis Topik dari Dokumen

Pada tahap ini dilakukan pemetaan dokumen ke dalam topik yang dihasilkan dari model dan melakukan analisis seberapa baik model dapat memodelkan topik. Analisis ini dilakukan dengan menggunakan 100 dokumen testing yang telah diberi label. Kemudian setiap dokumen *testing* diuji dengan menggunakan model terbaik yang dihasilkan kemudian diambil 4 topik tertinggi pada masing-masing dokumen dari hasil *testing* untuk dibandingkan dengan label dari dokumen. Kode yang digunakan untuk melakukan *testing* ditunjukkan dalam kode 61.

Untuk melakukan analisis dibutuhkan parameter berupa *doc* yang merupakan dokumen yang akan dimodelkan topiknya serta model *word embedding* yang akan digunakan. Dalam kode ini dilakukan pemanggilan *method* yang digunakan untuk menghitung nilai log probabilitas per kata dalam dokumen.

Halaman Sengaja Dikosongkan

BAB VI
HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan proses pengujian dan analisis terhadap hasil pengujian yang diperoleh dari proses implementasi yang telah dibahas pada bab sebelumnya.

6.1 *Load* Data

Data sura pelanggan bandar udara Juanda yang didapatkan dari repositori *suarajuanda.com* sejak tahun 2015 hingga 2018 adalah sebanyak 722 dokumen.

Tabel 6.1 Jumlah data pelatihan

Jumlah Dokumen	Jumlah Kata
722	18741

6.2 Pra-Proses Data

Tahap ini dilakukan pada dua jenis skenario data yaitu data TF-IDF dengan *stemming* dan tanpa *stemming* serta data frasa dengan *stemming* dan tanpa *stemming*. *Case folding,* tokenisasi, dan penghapusan *stopword* dilakukan pada seluruh skenario. Pembeda dari skenario tersebut adalah skenario data tanpa *stemming*: tidak dilakukan *stemming* pada kata-kata di dalam data, skenario data dengan *stemming*: dilakukan *stemming* pada kata-kata di dalam data. Perubahan jumlah kata pada dua jenis skenario data setelah dilakukan tahap pra-proses dapat dilihat pada Table 6.3.

Tabel 6.2 Perubahan Jumlah Kata setelah Pra-Proses Data

Tahapan	Tanpa Frasa		Frasa	
	Tanpa *Stemming*	*Stemming*	Tanpa *Stemming*	*Stemming*
Jumlah Awal	18741	18741	18741	18471
Pembersihan data	11031	11332	10921	10633

Pada tahap pra-proses ini, jumlah kata sebelum dan setelah melalui tahap *stemming* adalah sama karena proses *stemming* tidak melakukan penghapusan kata maupun token. Namun terdapat beberapa kata yang setelah melalui tahap *stemming* berubah menjadi kata yang termasuk *stopword*, sehingga terdapat selisih jumlah kata antara yang sudah dan belum melalui tahap *stemming*.

Penghitungan TF-IDF dilakukan pada ke dua jenis scenario data, yaitu data tanpa frasa dan dengan frasa. Dari hasil perhitungan TF-IDF didapatkan jumlah kemunculan kata tertinggi ditunjukkan dalam tabel 6.3.

Tabel 6.3 Kata dengan DF 10 Tertinggi pada Data dengan Frasa

Kata	Persentase DF
bandara	29.92%
juanda	17.59%
kasih	17.59%
terima	16.90%
tumpang	14.82%
mohon	13.71%
tunggu	12.05%
masuk	11.91%
pesawat	10.80%
terbang	10.80%
terminal	10.39%
tugas	10.39%

Dari tabel di atas diambil kesimpulan untuk batas atas, atau maksimal jumlah kemunculan kata dalam *corpus* harus kurang dari 10% atau 0.1. Untuk hasil perhitungan TF-IDF untuk data tanpa frasa ditunjukkan dalam tabel 6.4.

Tabel 6.4 Kata dengan DF 10 Tertinggi pada Data tanpa Frasa

Term-Frequency	Persentase DF
bandara	29.9%

Term-Frequency	Persentase DF
kasih	17.6%
juanda	17.6%
terima	16.9%
tumpang	14.8%
mohon	13.7%
tunggu	12.0%
masuk	11.9%
terbang	10.8%
pesawat	10.8%

Dari tabel di atas diambil kesimpulan untuk batas atas, atau maksimal jumlah kemunculan kata dalam *corpus* harus kurang dari 10% atau 0.1.

6.3 Pembuatan *Dictionary* dari Dokumen

Pada pembuatan *dictionary*, seluruh token diambil secara unik sehingga menghasilkan *unique tokens* yang mewakili setiap kata yang ada pada keseluruhan token. Setiap *unique token* ini memiliki *key* pada *dictionary* untuk memudahkan pembuatan *corpus.*

Setelah dibentuk *dictionary*, terdapat jumlah *unique tokens* pada masing-masing skenario data sesuai dengan Tabel 6.5

Tabel 6.5 Jumlah *Unique Tokens*

Tahapan	Tanpa Frasa		Frasa	
	Tanpa *Stemming*	*Stemming*	Tanpa *Stemming*	*Stemming*
Seluruh *unique tokens*	3479	3305	3299	2508
Min = 1 Max = 0.1	3238	2442	3291	2498
Min = 2 Max = 0.1	1189	980	1191	991
Min = 3 Max = 0.1	691	620	668	594

Jumlah akhir *unique tokens* inilah yang kemudian digunakan sebagai *dictionary* pada proses pemodelan dengan *Latent Dirichlet Allocation* dan *Gaussian Latent Dirichlet Allocation* di tahap selanjutnya.

6.4 Pemodelan dengan *Latent Dirichlet Allocation*

Pemodelan dengan *Latent Dirichlet Allocation* dilakukan dengan menggunakan 2 jenis skenario eksperimen yang akan berdasarkan pada 2 jenis data seperti yang telah dijelaskan sebelumnya. Untuk manghasilkan model yang terbaik dari ke dua jenis skenario, terlebih dahulu dilakukan penentuan jumlah *passes* dan penentuan jumlah topik yang akan membentuk model.

6.4.1 Penentuan Jumlah *Passes*

Gambar 6.1 Nilai *Perplexity* Penentuan Jumlah *Passes*

Jumlah *passes* ditentukan dengan melakukan analisa berdasarkan pada kestabilan nilai *perplexity* pada setiap *passes*. Percobaan dilakukan dengan mula-mula menggunakan jumlah *passes* yaitu 100 *passes* untuk kemudian dianalisa tren nilai *perplexity*. Percobaan ini dilakukan sebanyak enam kali dengan jumlah topik yaitu 2, 5, 8, 10, 11, 15 dan 20 untuk memastikan kestabilan tren nilai *passes* yang dihasilkan. Hasil nilai

perplexity yang muncul dari eksperimen kemudian dicatat dan ditampilkan dalam bentuk *line chart.*

Pada tahap ini akan dilakukan penentuan jumlah *passes* untuk semua skenario yang telah ditentukan

Pertama, dilakukan eksperimen penentuan jumlah *passes* dengan data yang digunakan sebagai *input* merupakan data hasil pra-proses. Hasil eksperimen nilai *perplexity* untuk penentuan jumlah *passes* pada skenario data Tanpa Frasa dapat dilihat pada Lampiran A-1.

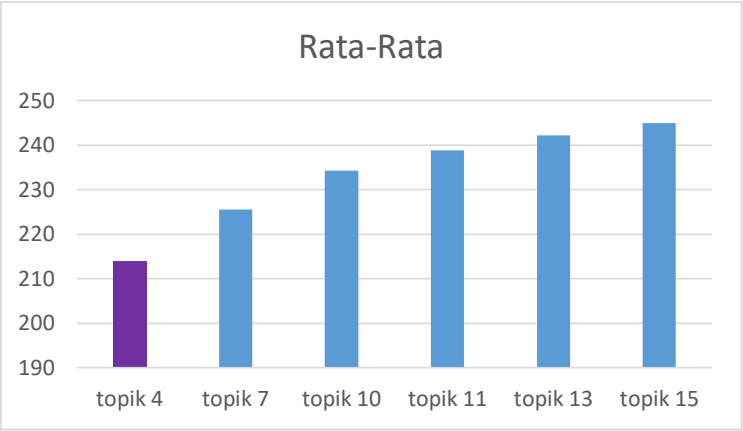
Dari jumlah *passes* awal yang ditentukan yaitu 100 *passes,* didapatkan dari hasil ekperimen bahwa nilai *perplexity* mulai stabil pada *passes* ke-13 pada seluruh jumlah topik yang digunakan. Selanjutnya disimpulkan bahwa nilai *passes* yang digunakan pada tahap selanjutnya yaitu 15 *passes.*

6.4.2 Penentuan Jumlah Topik

Penentuan jumlah topik dilakukan juga dengan menganalisa nilai *perplexity.* Namun, dalam penentuan jumlah topik, nilai *perplexity* bukan dilihat dari kestabilannya, tetapi untuk menentukan jumlah topik akan dilihat nilai *perplexity* terakhir dari masing-masing *passes* yang akan digunakan. Dalam penentuan topik dilakukan juga eksperimen dengan menggunakan beberapa topik yaitu 4, 7, 10, 11, 13, dan 15 topik. Ekperiman untuk menentukan jumlah topik dilakukan untuk ke dua jenis skenario yang telah ditentukan dengan 12 skenario pada masing-masing skenario data.

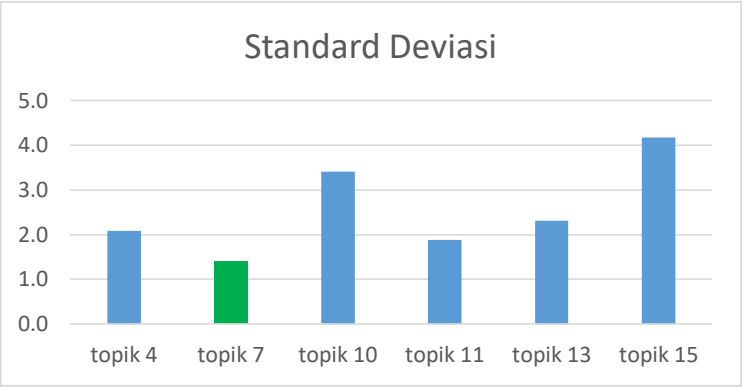
6.4.2.1 Data dengan Batasan Bawah 1 tanpa *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data batasan bawah 1 tanpa *stemming* dapat dilihat pada Lampiran A-2.



Gambar 6.2 Rata-Rata Nilai *Perplexity* dalam 10 Pecobaan

Berdasarkan gambar 6.2, pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat dan skenario data Tanpa Frasa batasan 1 tanpa *stemming* dengan jumlah topik 4 topik memiliki nilai rata-rata *perplexity* yang paling rendah, yaitu dengan nilai 213.86. Semakin rendah nilai *perplexity* yang dihasilkan menunjukkan kemampuan model untuk menghasilkan topik-topik yang bervariasi dan dengan kata yang tidak mirip antar topik sehingga berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 1 tanpa *stemming*.



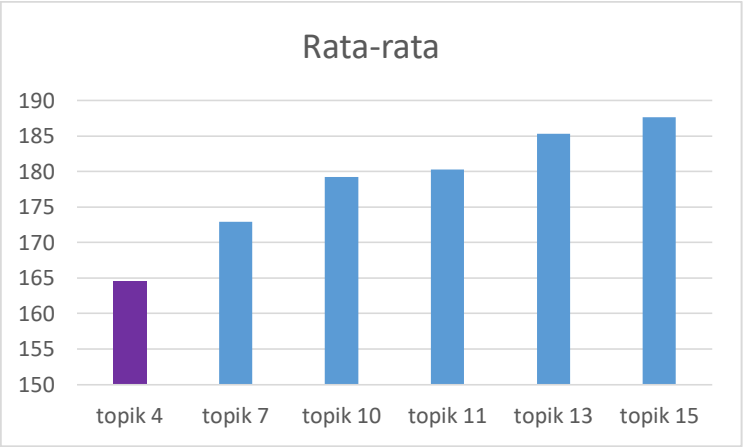
Gambar 6.3 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.3, diketahui bahwa standar deviasi nilai *perplexity* terendah adalah model dengan jumlah topik 7 yaitu 1.4 sedangkan, nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 2.5. Jika keduanya dibandingkan maka, nilai 1.4 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 7 topik yang didapatkan dari model ini terlampir pada Lampiran B-3.

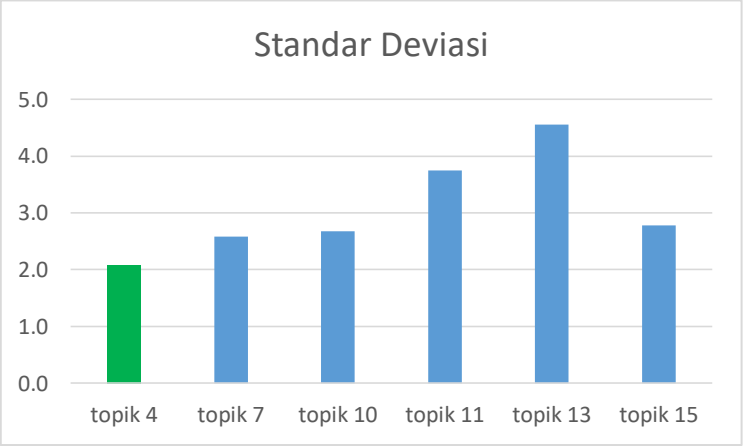
6.4.2.2 Data dengan Batasan Bawah 1 dengan *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario dapat dilihat pada Lampiran A-3.

Berdasarkan gambar 6.4, nilai rata-rata *perplexity* yang paling rendah pada skenario data Tanpa Frasa batasan 1 dengan *stemming* didapatkan dari model dengan jumlah topik 4, yaitu dengan nilai 164.59 serta pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 1 dengan *stemming*.



Gambar 6.4 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan



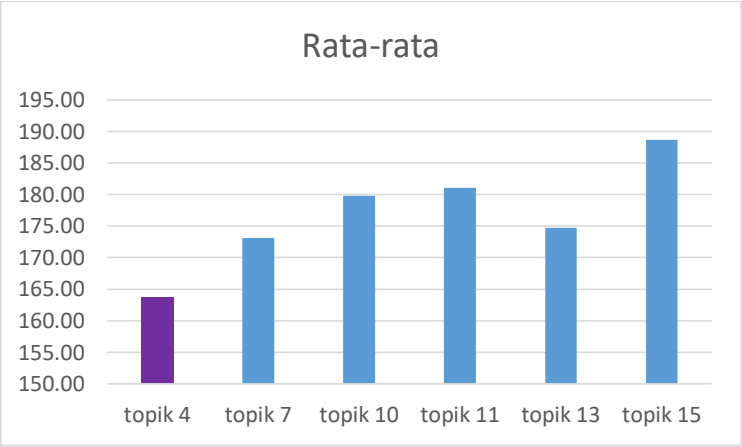
Gambar 6.5 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.5, diketahui bahwa standar deviasi nilai *perplexity* terendah adalah model dengan jumlah topik 4 yaitu 2.1 sedangkan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 3.1. Jika keduanya dibandingkan maka, nilai 2.1 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 4 topik yang didapatkan dari model ini terlampir pada Lampiran B-1.

6.4.2.3 Data dengan Batasan Bawah 2 dengan *Stemming*

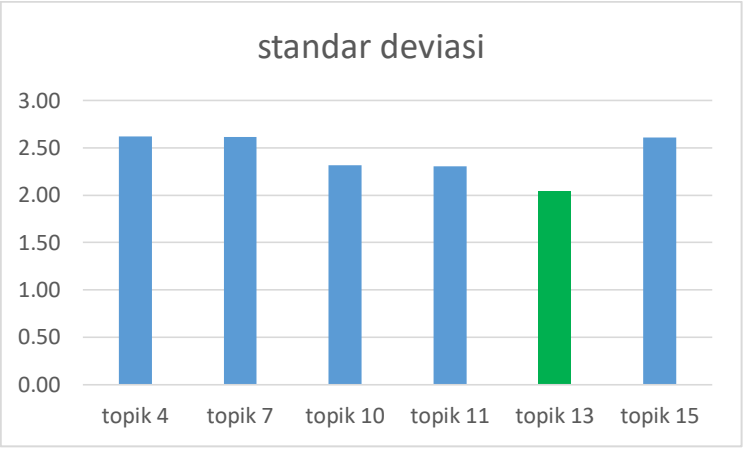
Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data batasan bawah 2 dengan *stemming* dapat dilihat pada Lampiran A-4.

Berdasarkan gambar 6.6, nilai rata-rata *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 2 dengan *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 164.59. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat, namun pada jumlah topik 13 mengalami penurunan dan kembali naik pada topik ke 15. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 2 dengan *stemming*.



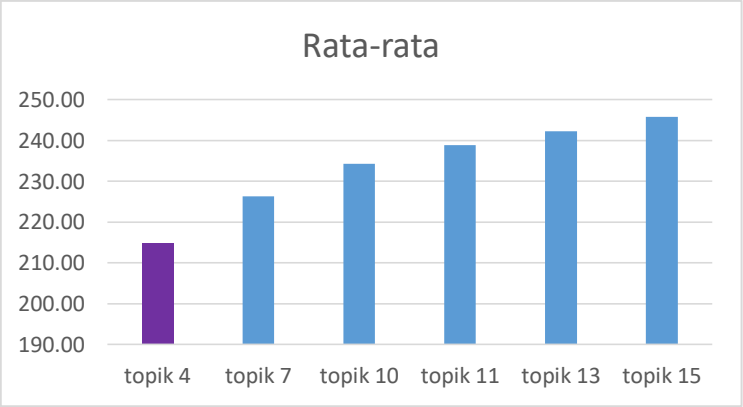
Gambar 6.6 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.18, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 13 adalah 2.0, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 2.4. Jika keduanya dibandingkan maka, nilai 2.0 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 13 topik yang didapatkan dari model ini terlampir pada Lampiran B-5.



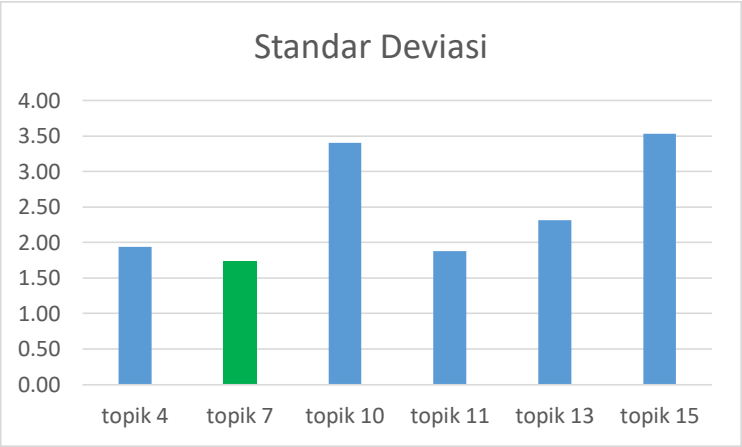
Gambar 6.7 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

6.4.2.4 Data dengan Batasan Bawah 2 tanpa *Stemming*



Gambar 6.8 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data batasan bawah 2 tanpa *stemming* dapat dilihat pada Lampiran A-5.



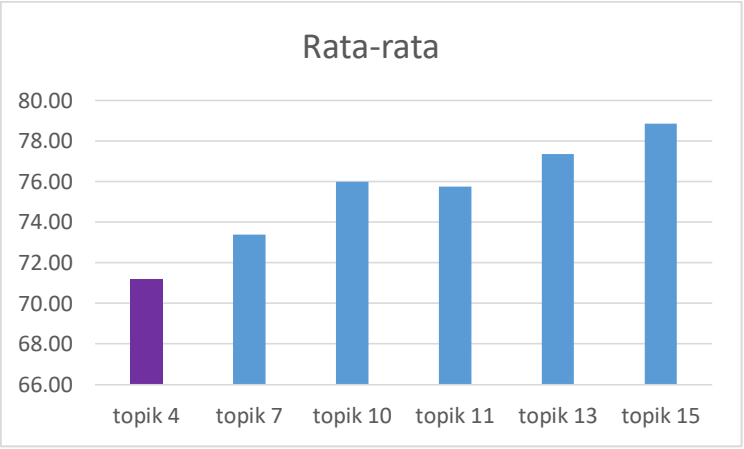
Gambar 6.9 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Berdasarkan gambar 6.8, nilai rata-rata *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 2 tanpa *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 214.84. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 2 tanpa *stemming*.

Sesuai dengan gambar 6.9, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 7 adalah 1.7, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 2.4. Jika keduanya dibandingkan maka, nilai 1.7 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 7 topik yang didapatkan dari model ini terlampir pada Lampiran B-3.

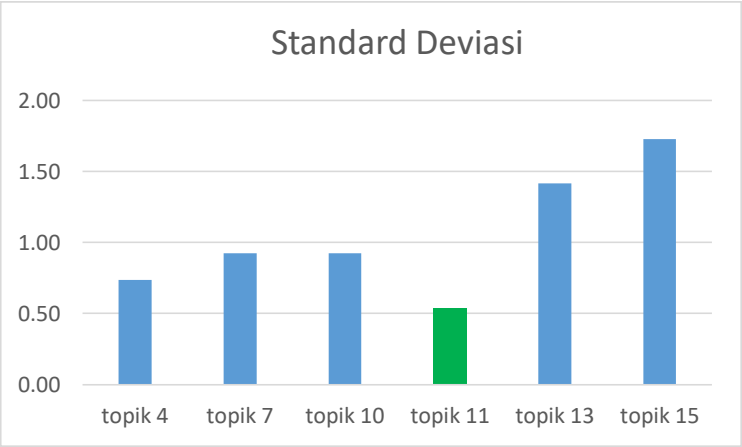
6.4.2.5 Data dengan Batasan Bawah 3 dengan *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data batasan bawah 3 dengan *stemming* dapat dilihat pada Lampiran A-6.



Gambar 6.10 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan

Berdasarkan gambar 6.10, nilai rata-rata *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 2 tanpa *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 71.19. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat, namun mengalami penuruann yang tidak begitu drastic pada topik ke 11 dan kembali naik pada topik ke 13. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 3 dengan *stemming*.



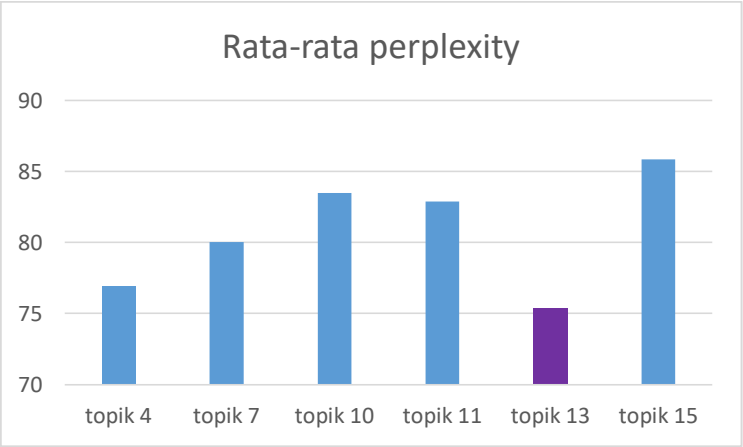
Gambar 6.11 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.11, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 11 adalah 0.5, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 1.0. Jika keduanya dibandingkan maka, nilai 0.5 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 11 topik yang didapatkan dari model ini terlampir pada Lampiran B-4.

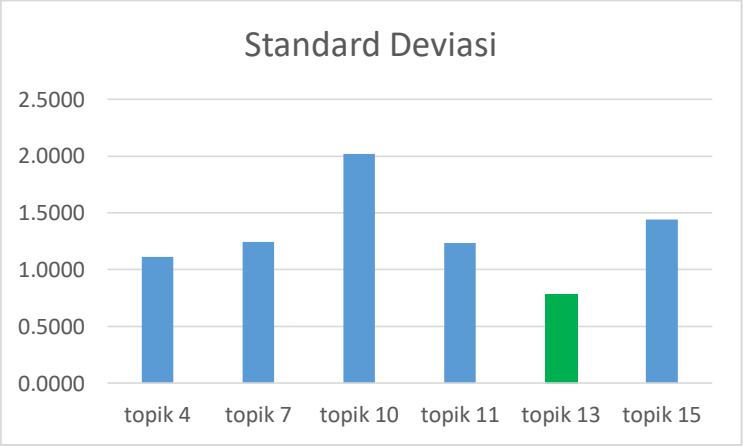
6.4.2.6 Data dengan Batasan Bawah 3 tanpa *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data batasan bawah 3 tanpa *stemming* dapat dilihat pada Lampiran A-7.

Berdasarkan gambar 6.12, nilai rata-rata *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 2 tanpa *stemming* didapatkan dari model dengan jumlah topik 13 topik, yaitu dengan nilai 75.38. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat, namun mengalami penuruann yang drastic pada topik ke 13 dan kembali naik pada topik ke 13. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 13 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 3 tanpa *stemming*



Gambar 6.12 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan

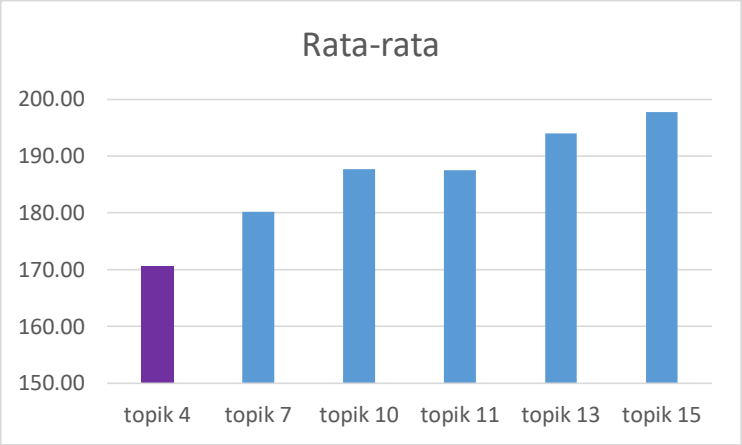


Gambar 6.13 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.13, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 13 adalah 0.8, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 1.3. Jika keduanya dibandingkan maka, nilai 0.8 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 13 topik yang didapatkan dari model ini terlampir pada Lampiran B-6.

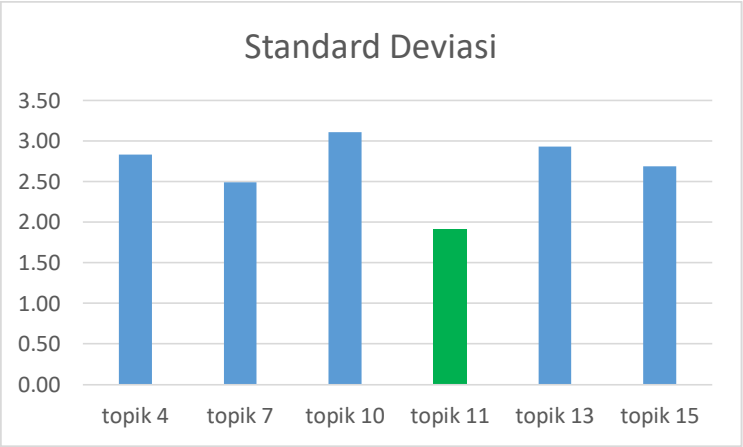
6.4.2.7 Data Frasa dengan Batasan Bawah 1 dengan *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data frasa batasan bawah 1 dengan *stemming* dapat dilihat pada Lampiran A-8.



Gambar 6.14 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan

Berdasarkan gambar 6.14, nilai rata-rata *perplexity* yang paling rendah pada skenario data frasa batasan bawah 1 dengan *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 170.61. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat, namun mengalami penuruann yang tidak begitu drastic pada topik ke 11 dan kembali naik pada topik ke 13 dan 15. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data frasa batasan bawah 1 dengan *stemming*.



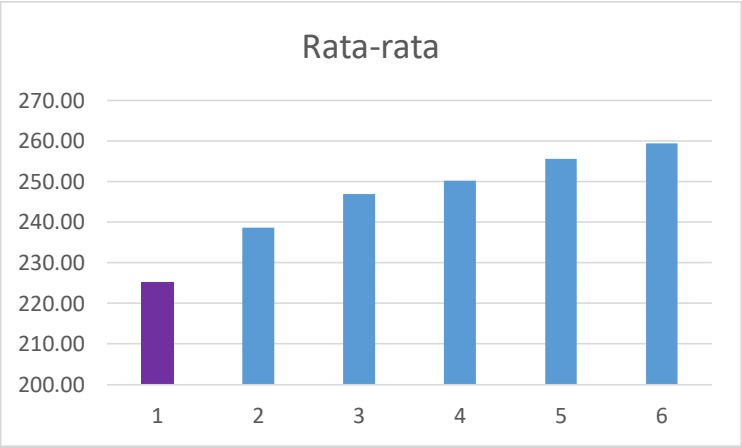
Gambar 6.15 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.15, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 11 adalah 1.9, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 2.6. Jika keduanya dibandingkan maka, nilai 1.9 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 11 topik yang didapatkan dari model ini terlampir pada Lampiran B-11.

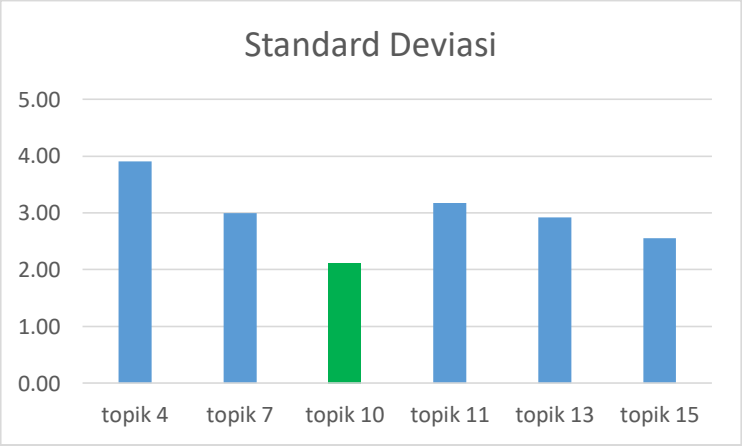
6.4.2.8 Data Frasa dengan Batasan Bawah 1 tanpa *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data frasa batasan bawah 1 tanpa *stemming* dapat dilihat pada Lampiran A-9.

Berdasarkan gambar 6.16, nilai rata-rata *perplexity* yang paling rendah pada skenario data frasa batasan bawah 1 tanpa *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 225.34. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data frasa batasan bawah 1 tanpa *stemming*.



Gambar 6.16 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan



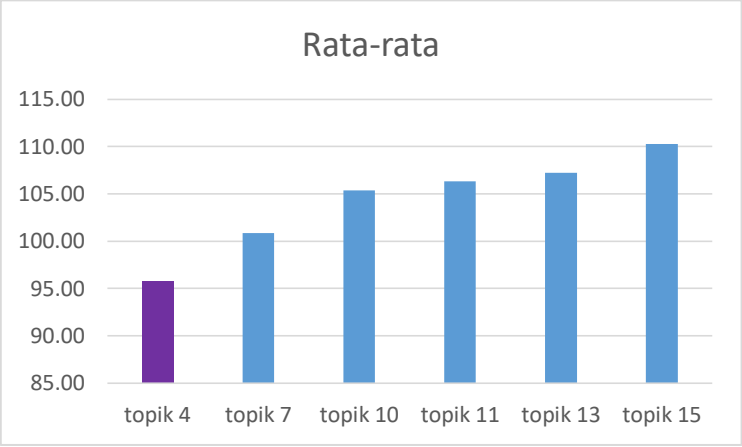
Gambar 6.17 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.17, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 10 adalah 2.1, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 2.9. Jika keduanya dibandingkan maka, nilai 2.1 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 10 topik yang didapatkan dari model ini terlampir pada Lampiran B-10.

6.4.2.9 Data Frasa dengan Batasan Bawah 2 dengan *Stemming*

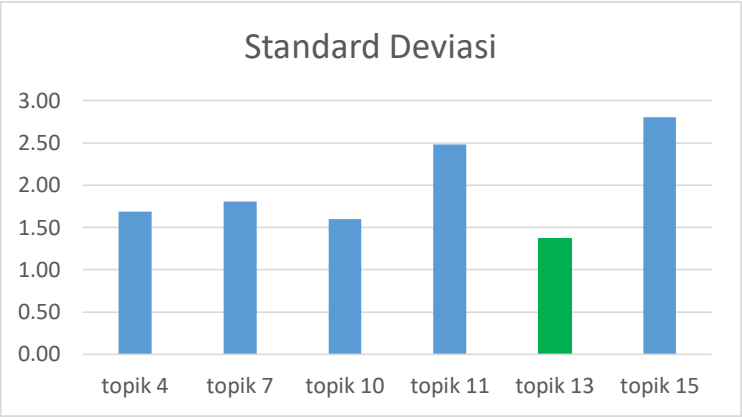
Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data frasa batasan bawah 2 dengan *stemming* dapat dilihat pada Lampiran A-10.

Berdasarkan gambar 6.18, nilai rata-rata *perplexity* yang paling rendah pada skenario data frasa batasan bawah 2 dengan *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 95.79. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data frasa batasan bawah 2 dengan *stemming*.



Gambar 6.18 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan

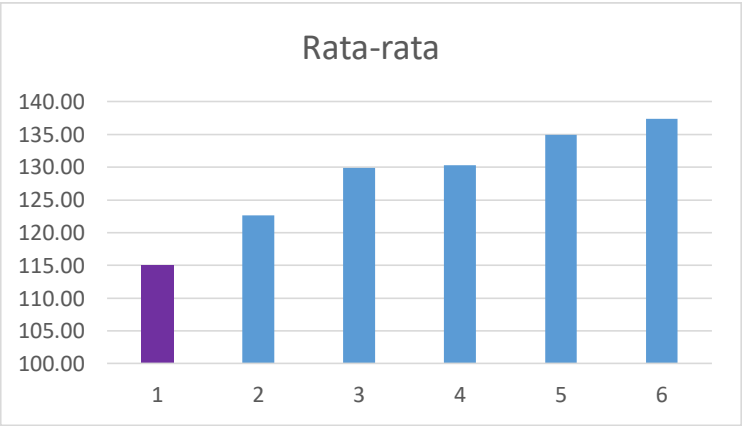
Sesuai dengan gambar 6.19, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 13 adalah 1.3, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 1.9. Jika keduanya dibandingkan maka, nilai 1.3 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 13 topik yang didapatkan dari model ini terlampir pada Lampiran B-12.



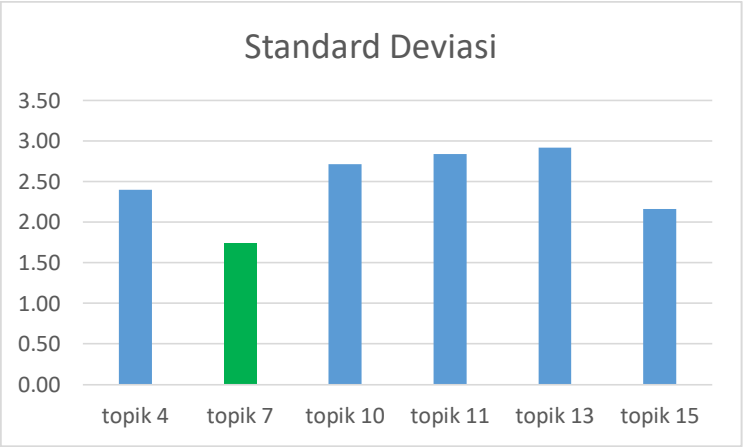
Gambar 6.19 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

6.4.2.10 Data Frasa dengan Batasan Bawah 2 tanpa *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data frasa batasan bawah 2 tanpa *stemming* dapat dilihat pada Lampiran A-11.



Gambar 6.20 Rata-Rata Nilai *Perplexity* dalam 10 Perocobaan



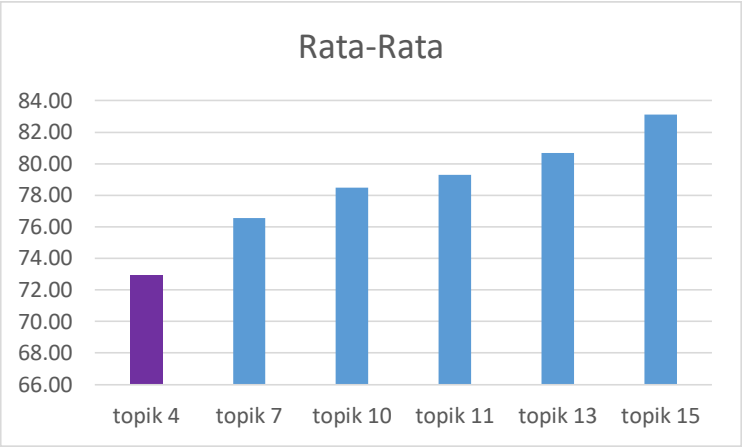
Gambar 6.21 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Berdasarkan gambar 6.20, nilai rata-rata *perplexity* yang paling rendah pada skenario data frasa batasan bawah 2 tanpa *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 115.09. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data frasa batasan bawah 2 tanpa *stemming*.

Sesuai dengan gambar 6.21, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 7 adalah 1.7, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 2.4. Jika keduanya dibandingkan maka, nilai 1.7 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 7 topik yang didapatkan dari model ini terlampir pada Lampiran B-9.

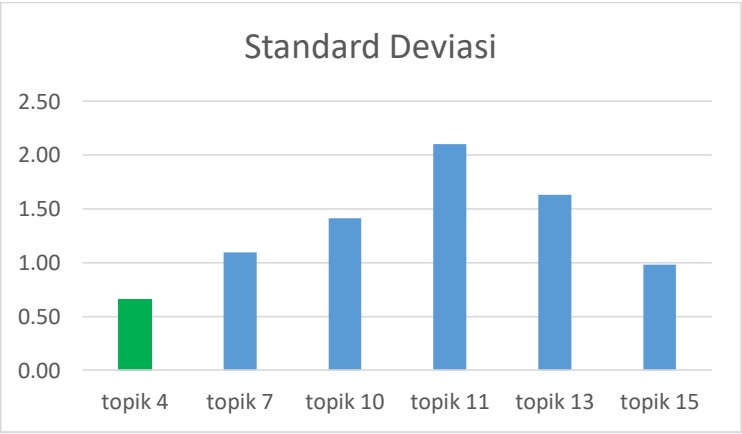
6.4.2.11 Data Frasa dengan Batasan Bawah 3 dengan *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data frasa batasan bawah 3 dengan *stemming* dapat dilihat pada Lampiran A-12.



Gambar 6.22 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan

Berdasarkan gambar 6.22, nilai rata-rata *perplexity* yang paling rendah pada skenario data frasa batasan bawah 3 dengan *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 72.97. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data frasa batasan bawah 3 dengan *stemming*.



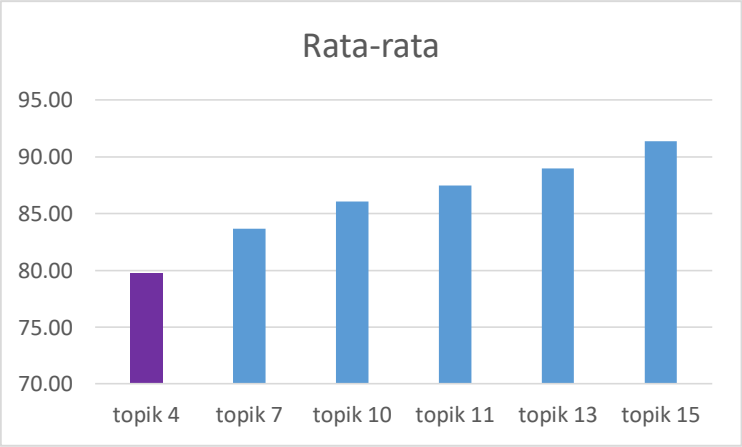
Gambar 6.23 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.23, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 4 adalah 0.6, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 1.3. Jika keduanya dibandingkan maka, nilai 0.6 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 4 topik yang didapatkan dari model ini terlampir pada Lampiran B-7.

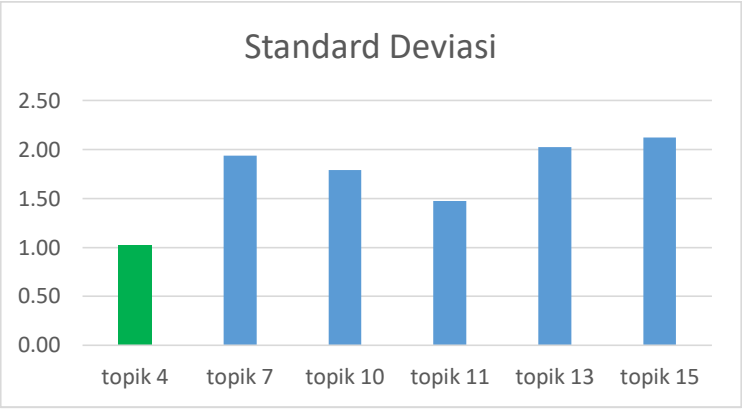
6.4.2.12 Data Frasa dengan Batasan Bawah 3 tanpa *Stemming*

Hasil eksperimen nilai *perplexity* untuk penentuan jumlah topik pada skenario data frasa batasan bawah 3 tanpa *stemming* dapat dilihat pada Lampiran A-13.

Berdasarkan gambar 6.24, nilai rata-rata *perplexity* yang paling rendah pada skenario data frasa batasan bawah 3 tanpa *stemming* didapatkan dari model dengan jumlah topik 4 topik, yaitu dengan nilai 79.78. Pada jumlah topik yang tinggi, tren rata-rata nilai *perplexity* semakin meningkat. Berdasarkan rata-rata nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 4 topik merupakan model terbaik dalam skenario data frasa batasan bawah 3 tanpa *stemming*.



Gambar 6.24 Rata-Rata Nilai *Perplexity* dalam 10 Percobaan



Gambar 6.25 Nilai Standar Deviasi *Perplexity* dalam 10 Percobaan

Sesuai dengan gambar 6.25, diketahui bahwa standar deviasi nilai *perplexity* model dengan jumlah topik 4 adalah 1.0, dan nilai rata-rata dari standar deviasi nilai *perplexity* keseluruhan adalah 1.3. Jika keduanya dibandingkan maka, nilai 1.0 dapat dikatakan cukup stabil karena lebih rendah dari nilai rata-rata standar deviasi keseluruhan. Daftar 4 topik yang didapatkan dari model ini terlampir pada Lampiran B-8.

Berdasarkan hasil analisis nilai *perplexity* berikut jumlah topik yang akan digunakan untuk masing-masing skenario. Jumlah topik yang diambil berdasarkan nilai rata-rata *perplexity* berada pada kolom dengan *heading* '*jumlah topik mean*' dan jumlah topik yang diambil berdasatkan pada nilai standar deviasi berada pada kolom dengan *heading 'jumlah topik std'*.

Tabel 6.6 Jumlah Topik Per Skenario

Tahapan	Tanpa Frasa				Frasa			
	Tanpa *Stemming*		*Stemming*		Tanpa *Stemming*		*Stemming*	
	Jumlah topik (mean)	Jumlah topik (std)	Jumlah topik (mean)	Jumlah topik (std)	Jumlah topik (mean)	Jumlah topik (std)	Jumlah topik (mean)	Jumlah topik (std)
Min = 1 Max = 0.1	4	7	4	4	4	10	4	11
Min = 2	4	7	4	13	4	7	4	13

Max = 0.1								
Min = 3 Max = 0.1	13	13	4	11	4	4	4	4

6.5 Pemodelan dengan *Gaussian Latent Dirichlet Allocation*

Pemodelan dengan *Gaussian Latent Dirichlet Allocation* dilakukan dengan menggunakan 2 jenis skenario eksperimen yang akan berdasarkan pada 2 jenis data seperti yang telah dijelaskan sebelumnya. Untuk manghasilkan model yang terbaik dari ke dua jenis skenario, terlebih dahulu dilakukan pembentukan *corpus,* vektor kata dan penentuan jumlah topik yang akan membentuk model.

6.5.1 Pembentukan *Corpus*

Dalam pembentukan *corpus* seluruh kata yang ada di dalam data pertama-tama di-*shuffle* kemudian dikelompokkan menjadi beberapa kelompok sesuai dengan jumlah topik yang ditentukan. Hal ini dilakukan karena *corpus* yang dibentuk memiliki tipe *map* dengan *key* berupa nomor dokumen dari kelompok kata dengan topiknya dengan *value* berupa kata dan topik.

Tabel 6.7 Jumlah Dokumen dalam *Corpus* pada masing-masing Skenario Data

Tahapan	Tanpa Frasa		Frasa	
	Tanpa *Stemming*	*Stemming*	Tanpa *Stemming*	*Stemming*
Seluruh *unique tokens*	722	722	722	722
Min = 1 Max = 0.1	710	710	710	710
Min = 2 Max = 0.1	686	688	686	688
Min = 3 Max = 0.1	676	680	676	680

Setelah dibentuk *corpus*, terdapat jumlah dokumen yang berbeda pada masing-masing skenario data sesuai dengan Tabel 6.7. Hasil dari *corpus* yang telah dibentuk selanjutnya akan digunakan untuk tahap selanjutnya.

6.5.2 Pembentukan Vektor Kata

Dalam Pembentukan vektor kata, dibutuhkan 2 data sebagai *input,* yang pertama yaitu data berupa *word embedding* model yang berisikan vektor-vektor kata dan yang kedua yaitu data *vocab* yang berisikan kata-kata dalam dokumen yang akan dikonversi menjadi vektor.

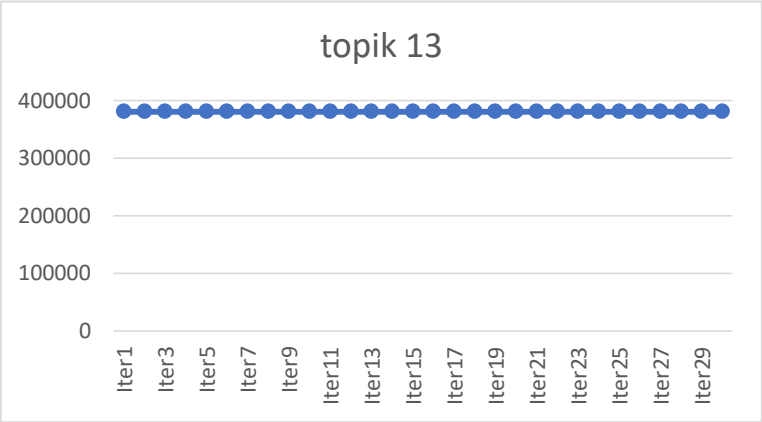
Seluruh kata dalam dokumen akan dikonversi ke dalam bentuk vektor, kata-kata yang tidak terdapat di dalam model akan dihitung sebagai kata yang *unusable_vocab* dan dikonversi menjadi vektor kata *'unk'* yaitu vektor yang disediakan untuk kata-kata diluar kata yang ada di dalam model (*unknown)*.

Setelah dibentuk *corpus*, terdapat jumlah dokumen yang berbeda pada masing-masing skenario data sesuai dengan Tabel 6.8

Tabel 6.8 Jumlah Kata yang Dapat Dikonversi dan Tidak Dapat Dikonversi ke Bentuk Vektor

Tahapan	Tanpa Frasa				Frasa			
	Tanpa *Stemming*		*Stemming*		Tanpa *Stemming*		*Stemming*	
	Yes	*No*	*Yes*	*No*	*Yes*	*No*	*Yes*	*No*
Seluruh *unique tokens*	3044	435	2786	519	2912	567	2025	483
Min = 1 Max = 0.1	2765	473	2041	401	2717	574	2016	482
Min = 2 Max = 0.1	1148	41	942	38	1115	76	924	67
Min = 3 Max = 0.1	678	13	604	16	635	33	590	34

6.5.3 Penentuan Jumlah Iterasi



Gambar 6.26 Nilai *Perplexity* Penentuan Jumlah Iterasi

Jumlah *passes* ditentukan dengan melakukan analisa berdasarkan pada kestabilan nilai *perplexity* pada setiap *passes*. Percobaan dilakukan dengan mula-mula menggunakan jumlah iterasi yaitu 30 iterasi untuk kemudian dianalisa tren nilai *perplexity*. Percobaan ini dilakukan dengan menggunakan jumlah topik yaitu 13 topik untuk memastikan kestabilan tren nilai iterasi yang dihasilkan. Hasil nilai *perplexity* yang muncul dari eksperimen kemudian dicatat dan ditampilkan dalam bentuk *line chart.*

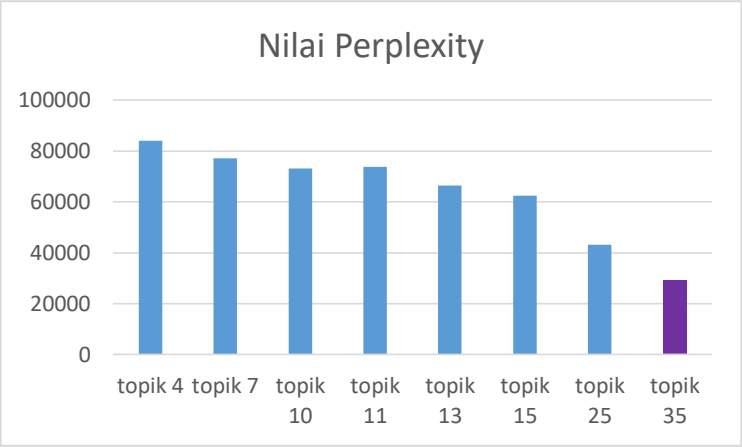
Gambar 6.26 merupakan visualisasi hasil dari perhitungan nilai *perplexity* dari 30 iterasi yang dijalankan pada jumlah topik 13. Dari gambar dapat dilihat bahwa nilai *perplexity* yang dihasilkan memiliki nilai yang sama dan tidak memiliki perubahan yang begitu signifikan di setiap iterasi, sehingga dapat disimpulkan bahwa jumlah iterasi yang dapat digunakan untuk menentukan nilai topik pada proses selanjutnya yaitu dapat 1 atau jumlah lain, karena setiap iterasi memiliki nilai yang sama. Pada penelitian ini, jumlah iterasi yang digunakan adalah 1 iterasi.

6.5.4 Penentuan Jumlah Topik

Penentuan jumlah topik dalam pemodelan *Gaussian LDA* juga dilakukan dengan menganalisa nilai *perplexity*. Dalam penentuan topik dilakukan eksperimen dengan menggunakan beberapa topik yaitu 4, 7, 10, 11, 13, 15, 25 dan 35 topik. Ekperiman untuk menentukan jumlah topik dilakukan untuk ke dua jenis skenario yang telah ditentukan.

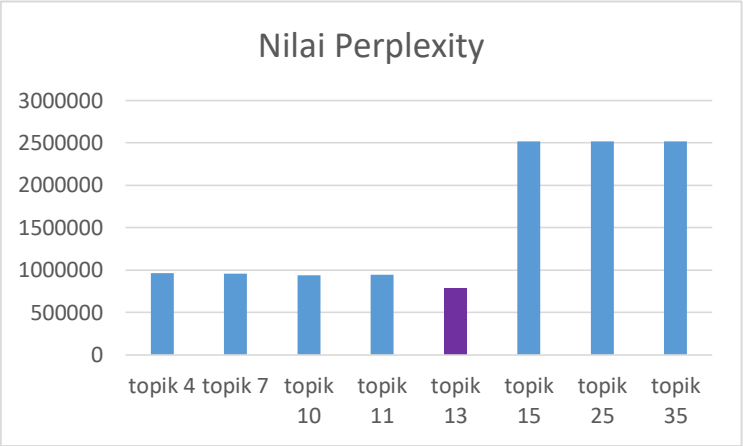
6.5.4.1 Data dengan Batasan Bawah 1 dengan *Stemming*

Berdasarkan gambar 6.27, nilai *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 1 dengan *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 29068. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 1 dengan *stemming*.



Gambar 6.27 Nilai *Perplexity* Data

6.5.4.2 Data dengan Batasan Bawah 1 tanpa *Stemming*

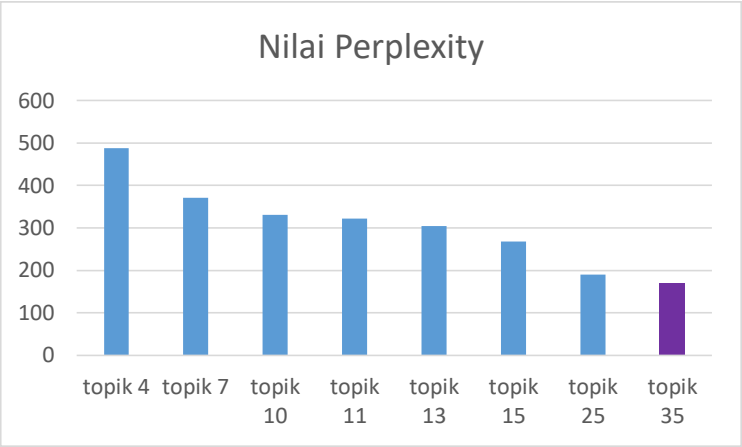


Gambar 6.28 Nilai *Perplexity* Data

Berdasarkan gambar 6.28, nilai *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 1 tanpa *stemming* didapatkan dari model dengan jumlah topik 13 topik, yaitu dengan nilai 789834. Pada jumlah topik yang rendah, tren nilai *perplexity* stabil dari topik 4 sampai dengan topik 11, namun pada topik 13 mengalami penuruanan dan kembali naik dengan sangat drastic pada topik 15, 25 dan 35. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 13 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 1 tanpa *stemming*.

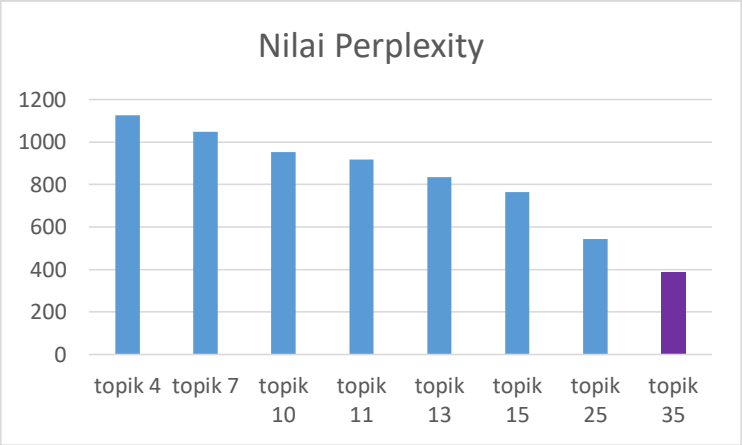
6.5.4.3 Data dengan Batasan Bawah 2 dengan *Stemming*

Berdasarkan gambar 6.29, nilai *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 2 dengan *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 170. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 2 dengan *stemming*.



Gambar 6.29 Nilai *Perplexity* Data

6.5.4.4 Data dengan Batasan Bawah 2 tanpa *Stemming*

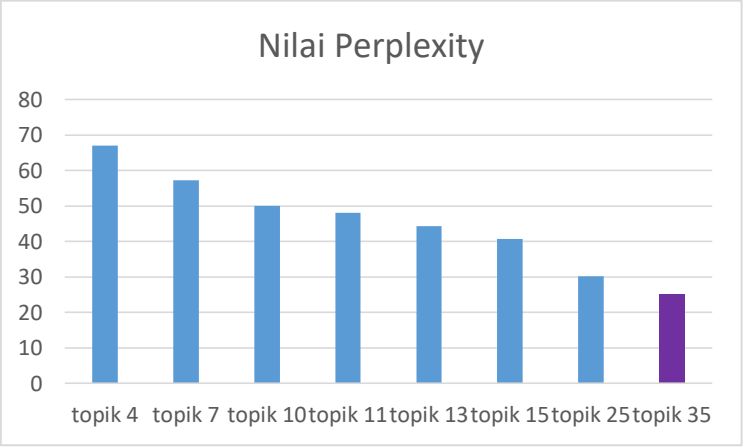


Gambar 6.30 Nilai *Perplexity* Data

Berdasarkan gambar 6.30, nilai *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 2 tanpa *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 386. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik

merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 2 tanpa *stemming*.

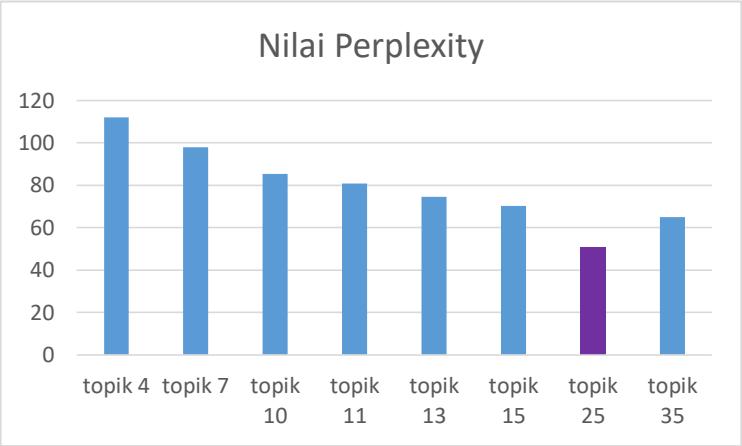
6.5.4.5 Data dengan Batasan Bawah 3 dengan *Stemming*



Gambar 6.31 Nilai *Perplexity* Data

Berdasarkan gambar 6.31, nilai *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 3 dengan *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 25. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 3 dengan *stemming*.

6.5.4.6 Data dengan Batasan Bawah 3 tanpa *Stemming*

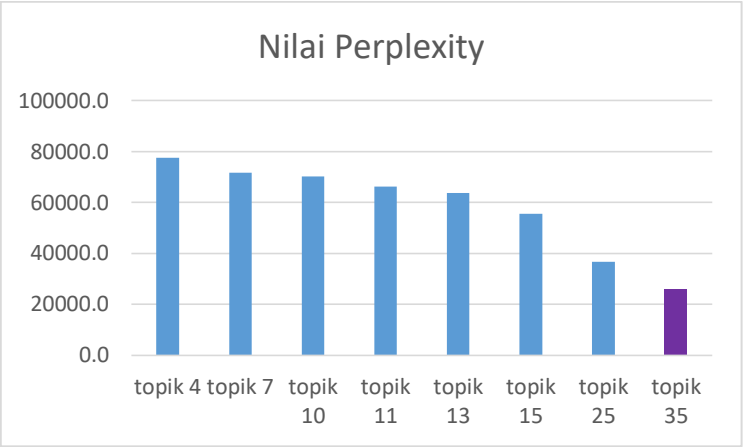


Gambar 6.32 Nilai *Perplexity* Data

Berdasarkan gambar 6.32, nilai *perplexity* yang paling rendah pada skenario data Tanpa Frasa dengan batasan bawah 3 tanpa *stemming* didapatkan dari model dengan jumlah topik 25 topik, yaitu dengan nilai 51. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun sampai dengan topik 25 dan mengalami kenaikan pada topik 35. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 25 topik merupakan model terbaik dalam skenario data data Tanpa Frasa dengan batasan bawah 3 tanpa *stemming*.

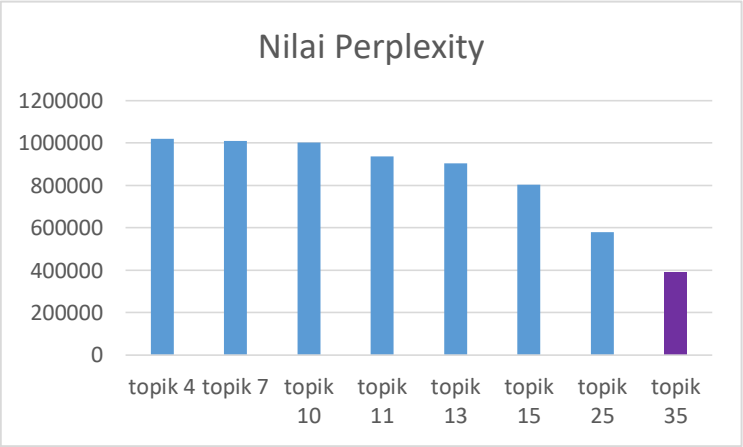
6.5.4.7 Data Frasa dengan Batasan Bawah 1 dengan *Stemming*

Berdasarkan gambar 6.33, nilai *perplexity* yang paling rendah pada skenario data frasa dengan batasan bawah 1 dengan *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 25692. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data frasa dengan batasan bawah 1 dengan *stemming*.



Gambar 6.33 Nilai *Perplexity* Data

6.5.4.8 Data Frasa dengan Batasan Bawah 1 tanpa *Stemming*

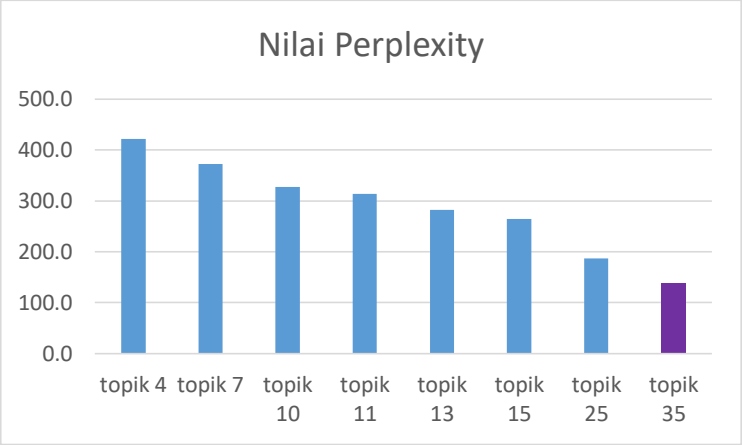


Gambar 6.34 Nilai *Perplexity* Data

Berdasarkan gambar 6.34, nilai *perplexity* yang paling rendah pada skenario data frasa dengan batasan bawah 1 tanpa *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 390396. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity*

ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data frasa dengan batasan bawah 1 tanpa *stemming*.

6.5.4.9 Data Frasa dengan Batasan Bawah 2 dengan *Stemming*



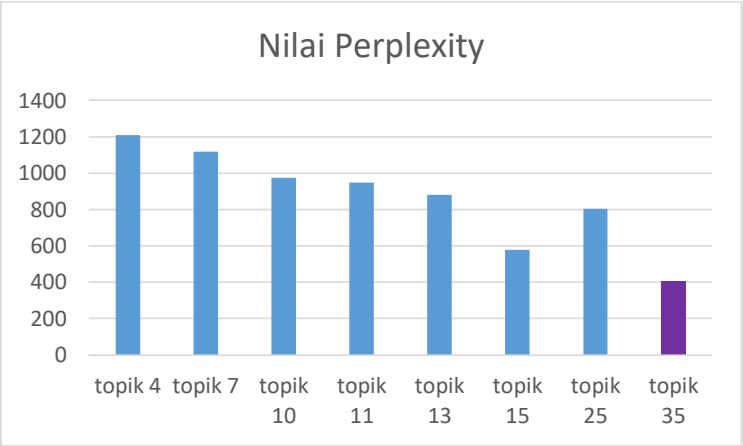
Gambar 6.35 Nilai *Perplexity* Data

Berdasarkan gambar 6.35, nilai *perplexity* yang paling rendah pada skenario data frasa dengan batasan bawah 2 dengan *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 137. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data frasa dengan batasan bawah 2 dengan *stemming*.

6.5.4.10 Data Frasa dengan Batasan Bawah 2 tanpa *Stemming*

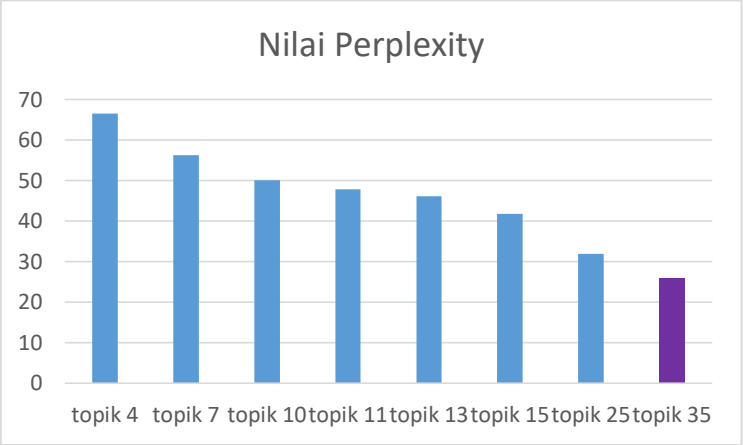
Berdasarkan gambar 6.36, nilai *perplexity* yang paling rendah pada skenario data frasa dengan batasan bawah 2 tanpa *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 401.6. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun sampai dengan topik 15 dan mengalami peningkatan pada topik 25 serta kembali menurun

pada topik 35. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data frasa dengan batasan bawah 2 tanpa *stemming*.



Gambar 6.36 Nilai *Perplexity* Data

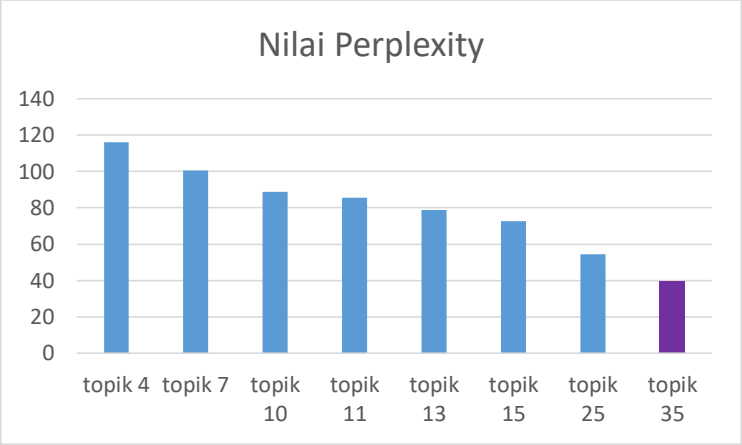
6.5.4.11 Data Frasa dengan Batasan Bawah 3 dengan *Stemming*



Gambar 6.37 Nilai *Perplexity* Data

Berdasarkan gambar 6.37, nilai *perplexity* yang paling rendah pada skenario data frasa dengan batasan bawah 3 dengan *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 25.7. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data frasa dengan batasan bawah 3 dengan *stemming*.

6.5.4.12 Data Frasa dengan Batasan Bawah 3 tanpa *Stemming*



Gambar 6.38 Nilai *Perplexity* Data

Berdasarkan gambar 6.38, nilai *perplexity* yang paling rendah pada skenario data frasa dengan batasan bawah 3 tanpa *stemming* didapatkan dari model dengan jumlah topik 35 topik, yaitu dengan nilai 39.7. Pada jumlah topik yang tinggi, tren nilai *perplexity* semakin menurun. Berdasarkan nilai *perplexity* ini, dapat disimpulkan bahwa model dengan jumlah 35 topik merupakan model terbaik dalam skenario data data frasa dengan batasan bawah 3 tanpa *stemming*.

Berdasarkan hasil analisis nilai *perplexity* berikut jumlah topik yang akan digunakan untuk masing-masing skenario.

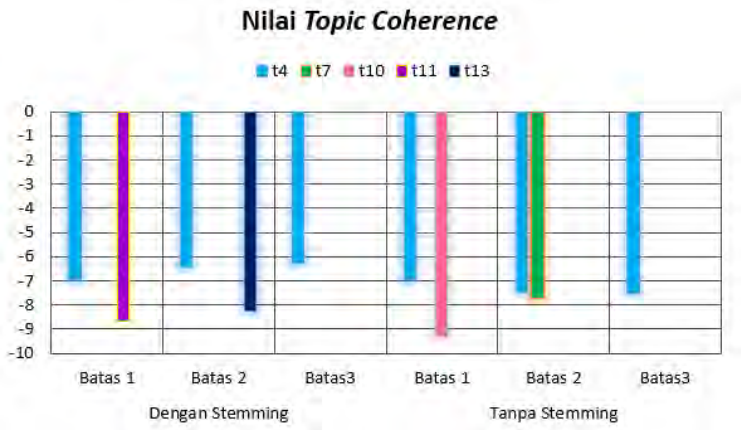
Tabel 6.9 Jumlah Topik untuk masing-masing Skenario

Tahapan	Tanpa Frasa		Frasa	
	Tanpa *Stemming*	*Stemming*	Tanpa *Stemming*	*Stemming*
Min = 1 Max = 0.1	13	35	35	35
Min = 2 Max = 0.1	35	35	35	35
Min = 3 Max = 0.1	25	35	35	35

6.6 Validasi Model Topik *Latent Dirichlet Allocation*

Validasi Model dilakukan dengan menghitung topik terbaik yang dihasilkan dari pemodelan masing-masing skenario. Model yang akan digunakan utuk divalidasi adalah model yang paling baik dari hasil perhitungan rata-rata niai *perplexity* dan standar deviasi. Perhitungan nilai *topic coherence* akan dilakukan untuk masing-masing model yang kemudian dibandingkan antara satu dengan yang lain.

6.6.1 Nilai Rata-Rata *Topic Coherence*



Gambar 6.39 Rata-Rata Nilai *Topic Coherence*

Hasil *topic coherence* yang dianalisis didapatkan dari pemanggilan fungsi *get_coherence* untuk setiap model.

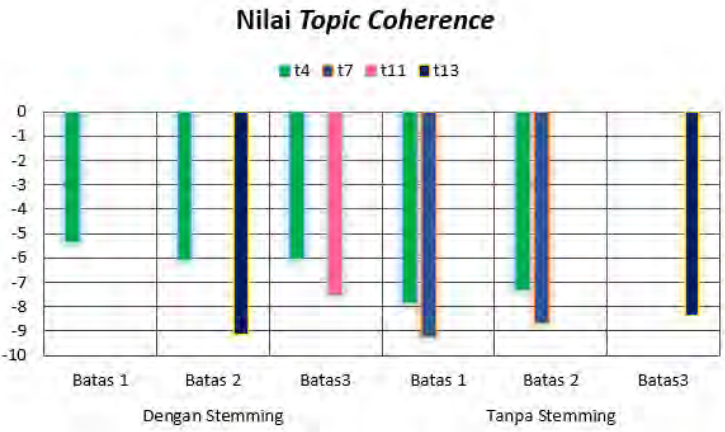
Berdasarkan pemanggilan fungsi *get_coherence*, dapat dihitung rata-rata nilai *topic coherence* yang dimiliki oleh setiap model. Semakin tinggi nilai menunjukkan hasil yang semakin baik. Rata-rata nilai *topic coherence* dari setiap model pada skenario data Tanpa Frasa dapat dilihat pada gambar 6.39.

Dari gambar 6.38, dapat diketahui bahwa model pada skenario data Tanpa Frasa batasan 1 dengan *stemming* memiliki nilai rata-rata nilai *topic coherence* yaitu -5.326 untuk model dengan jumlah topik 4, model pada skenario data Tanpa Frasa batasan 1 tanpa *stemming* memiliki nilai *rata-rata coherence* yaitu -7.853 untuk jumlah topik 4 dan -9.219 untuk jumlah topik 7.

Model pada skenario data Tanpa Frasa batasan 2 dengan *stemming* memiliki nilai *rata-rata coherence* yaitu -6.046 untuk jumlah topik 4 dan -9.122 untuk jumlah topik 13. Model pada skenario data Tanpa Frasa batasan 2 tanpa *stemming* memiliki nilai *rata-rata coherence* yaitu -7.312 untuk jumlah topik 4 dan -8.653 untuk jumlah topik 7.

Model pada skenario data Tanpa Frasa batasan 3 dengan *stemming* memiliki nilai *rata-rata coherence* yaitu -5.992 untuk jumlah topik 4 dan -7.514 untuk jumlah topik 11. Model pada skenario data Tanpa Frasa batasan 3 tanpa *stemming* memiliki nilai *rata-rata coherence* yaitu -8.333 untuk jumlah topik 13.

Dari gambar 6.40, dapat diketahui bahwa model pada skenario data frasa batasan 1 dengan *stemming* memiliki nilai rata-rata nilai *topic coherence* yaitu -6.922 untuk model dengan jumlah topik 4 dan -8.659 dengan jumlah topik 11, model pada skenario data frasa batasan 1 tanpa *stemming* memiliki nilai *rata-rata coherence* yaitu -6.946 untuk jumlah topik 4 dan -9.314 untuk jumlah topik 10.



Gambar 6.40 Rata-Rata Nilai *Topic Coherence*

Model pada skenario data frasa batasan 2 dengan *stemming* memiliki nilai *rata-rata coherence* yaitu -6.441 untuk jumlah topik 4 dan -8.274 untuk jumlah topik 13. Model pada skenario data frasa batasan 2 tanpa *stemming* memiliki nilai *rata-rata coherence* yaitu -7.469 untuk jumlah topik 4 dan -7.709 untuk jumlah topik 7.

Model pada skenario data frasa batasan 3 dengan *stemming* memiliki nilai *rata-rata coherence* yaitu -6.272 untuk jumlah topik 4. Model pada skenario data frasa batasan 3 tanpa *stemming* memiliki nilai *rata-rata coherence* yaitu -7.534 untuk jumlah topik 4

Berdasarkan pada ke dua gambar yang telah dijelaskan di atas, untuk skenario data Tanpa Frasa dapat disimpulkan bahwa model dengan skenario data Tanpa Frasa batasan 1 dengan *stemming* memiliki nilai *topic coherence* yang lebih baik dibandingkan dengan model pada 5 skenario lainnya. Kemudian pada gambar 6.49 juga diketahui bahwa pada skenario data Tanpa Frasa batasan 1 dengan *stemming* memiliki model dengan nilai *topic coherence* tertinggi dengan jumlah topik 4 dengan nilai -5.326.

Selain itu, untuk skenario data frasa dapat disimpulkan bahwa model dengan skenario data frasa batasan 3 dengan *stemming*

memiliki nilai *topic coherence* yang lebih baik dibandingkan dengan model pada 5 skenario lainnya. Kemudian pada gambar 6.50 juga diketahui bahwa pada skenario data frasa batasan 3 dengan *stemming* memiliki model dengan nilai *topic coherence* tertinggi dengan jumlah topik 4 dengan nilai -6.272.

Maka dari itu, dapat disimpulkan bahwa model adalah dengan *topic coherence* terbaik dari ke dua jenis skenario data model dengan skenario data Tanpa Frasa batasan 1 dengan *stemming*.

Model dengan jumlah topik 4 pada skenario data Tanpa Frasa batasan 1 dengan *stemming* memiliki rata-rata nilai *perplexity* 164.59 dan merupakan nilai *perplexity* paling rendah dari model.

Dari tahap validasi model berdasarkan nilai *perplexity* yang menjelaskan tingkat ketidakmiripan antar topik dan nilai *topic coherence* yang menjelaskan kualitas topik berdasarkan interpretasi manusia, didapatkan model dengan skenario data Tanpa Frasa batasan 1 dengan *stemming* untuk jumlah topik 4 dengan nilai *perplexity*, yaitu itu 164.59 dan nilai *topic coherence* tertinggi yaitu -5.326. Kedua nilai ini menjelaskan bahwa model *stemming* dengan jumlah topik 4 memiliki topik-topik yang lebih tidak mirip dibandingkan dengan model lain dan lebih mudah untuk diinterpretasikan oleh manusia.

6.7 Validasi Model Topik *Gaussian Latent Dirichlet Allocation*

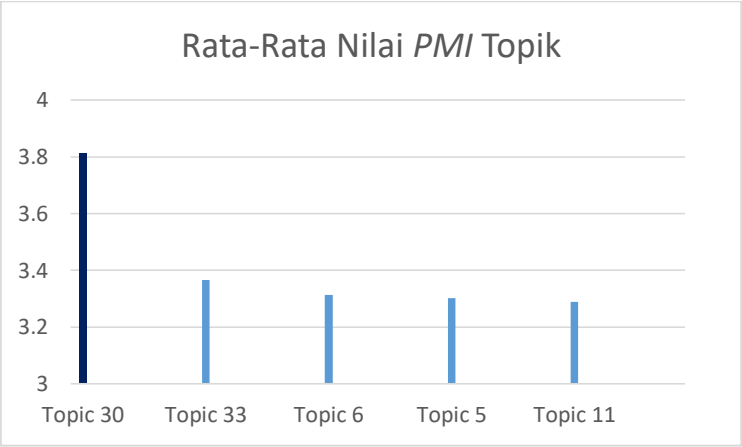
Validasi Model dilakukan dengan menghitung topik terbaik yang dihasilkan dari pemodelan masing-masing skenario. Model yang akan digunakan utuk divalidasi adalah model yang paling baik dari hasil perhitungan rata-rata niai *perplexity*. Perhitungan nilai *topic coherence* akan dilakukan untuk masing-masing model yang kemudian dibandingkan antara satu dengan yang lain.

Hasil *topic coherence* yang dianalisis didapatkan dari perhitungan dengan menggunakan metode *pointwise mutual information* untuk setiap topik yang ada di dalam model.

Berdasarkan perhitungan PMI, dapat dihitung rata-rata nilai *topic coherence* yang dimiliki oleh setiap model. Semakin tinggi nilai menunjukkan hasil yang semakin baik.

6.7.1 Data Frasa Batasan 1 dengan *Stemming*

Eksperimen pertama dalam mencari nilai PMI untuk mengetahui kemiripan kata dalam topik, dilakukan pada skenario data frasa dengan batasan 1 dengan *stemming*. Pada eksperimen, dilakukan pencarian 5 topik dengan probabilitas yang paling tinggi pada model skenario, kemudian dicari 5 kata dengan probabilitas paling tinggi pada setiap topik. Dari hasil eksperimen dengan *perplexity* model terbaik yang dihasilkan memiliki jumlah topik sebanyak 35 topik.

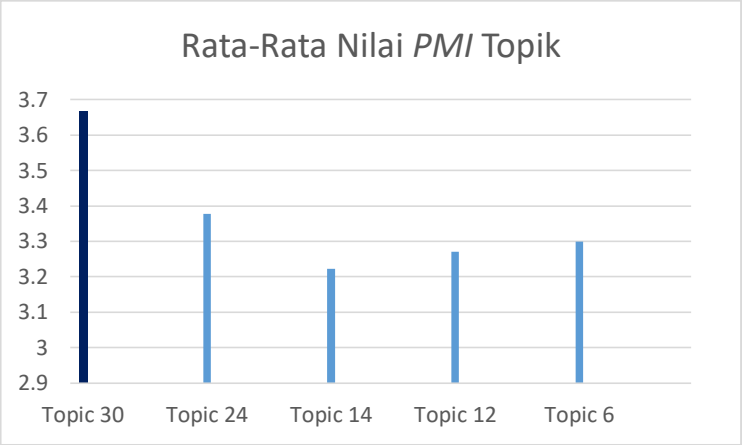


Gambar 6.41 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Frasa Batasan 1 – *Stemming*

Dari hasil eksperimen didapatkan bahwa nilai PMI mengalami penurunan yang cukup derastis dari topik 30 ke 33 dan nilai PMI stabil pada saat topik di bawah 15 topik dimana nilainya sedikt lebih rendah dibandingkan dengan topik 33. Dari gambar 6.41 didapatkan skenario model data frasa batasan 1 dengan *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model dengan jumlah topik topik 30 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.814 yang

menandakan topik ke-30 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.

6.7.2 Data Frasa Batasan 1 dengan Tanpa *Stemming*



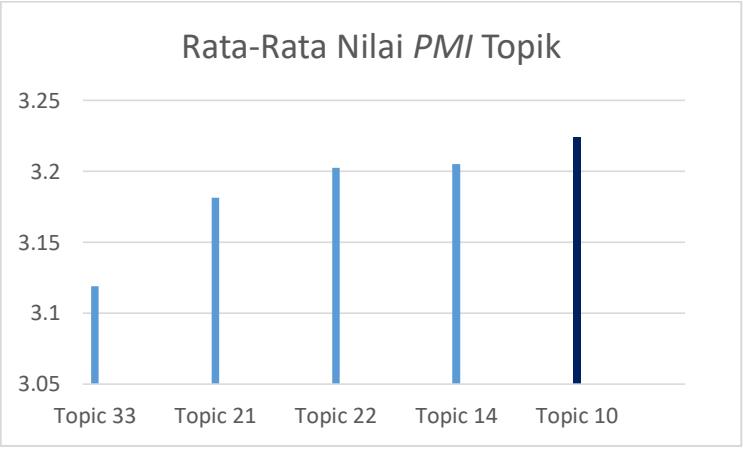
Gambar 6.42 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Frasa Batasan 1 – Tanpa *Stemming*

Selanjutnya, dilakukan eksperimen pada skenario data frasa batasan 1 tanpa *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data frasa batasan 1 tanpa *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model rata-rata nilai PMI mengalami penurunan dari topik 6 sampai topik 12 dan kembali naik untuk topik 24 dan mengalami kenaikan yang cukup derastis dari topik 24 ke topik 30 sehingga didapatkan topik 30 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.666 yang menandakan topik 30 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.

6.7.3 Data Frasa Batasan 2 dengan *Stemming*

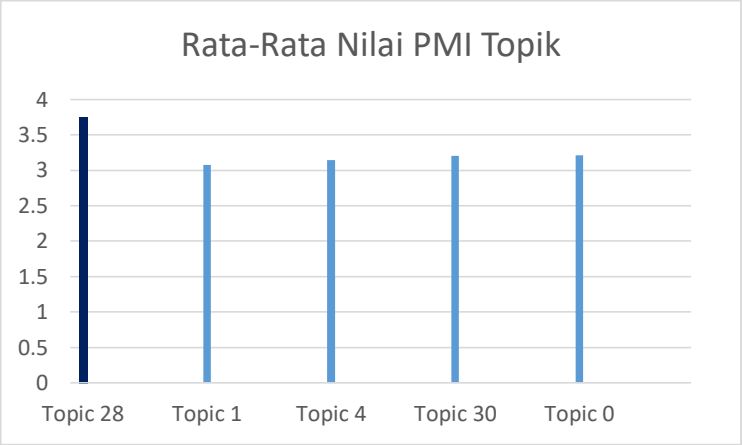
Selanjutnya, dilakukan eksperimen pada skenario data frasa batasan 2 dengan *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa

pada skenario model data frasa batasan 2 dengan *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model rata-rata nilai PMI mengalami penurunan dari topik 10 sampai topik 33, penurunan nilai PMI yang sangat derastis bila dibandingakan nilai PMI topik 10 dengan 33 sehingga didapatkan topik 10 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.119 yang menandakan topik 10 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.



Gambar 6.43 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Frasa Batasan 2 – Stemming

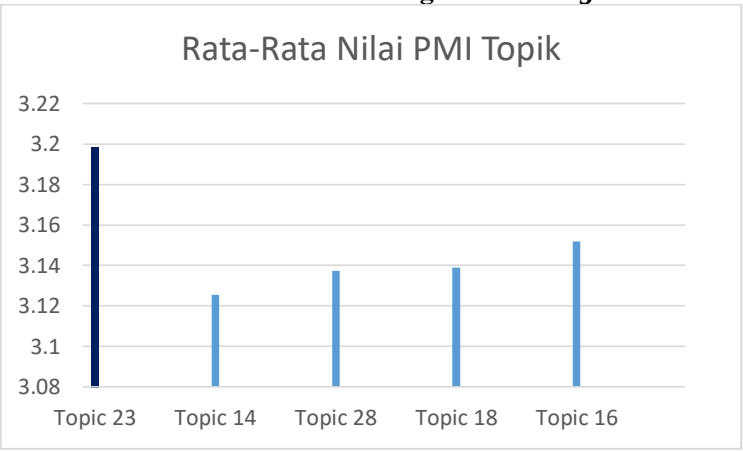
6.7.4 Data Frasa Batasan 2 dengan Tanpa *Stemming*



Gambar 6.44 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Frasa Batasan 2 – Tanpa *Stemming*

Selanjutnya, dilakukan eksperimen pada skenario data frasa batasan 2 tanpa *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data frasa batasan 2 tanpa *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model rata-rata nilai PMI stabil pada topik 0, 1 dan 4, nilai ini mengalami kenaikan yang derastis pada topik 28 dan mengalami penurunan yang derastis pada topik 30 sehingga didapatkan topik 28 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.752 yang menandakan topik 28 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.

6.7.5 Data Frasa Batasan 3 dengan *Stemming*



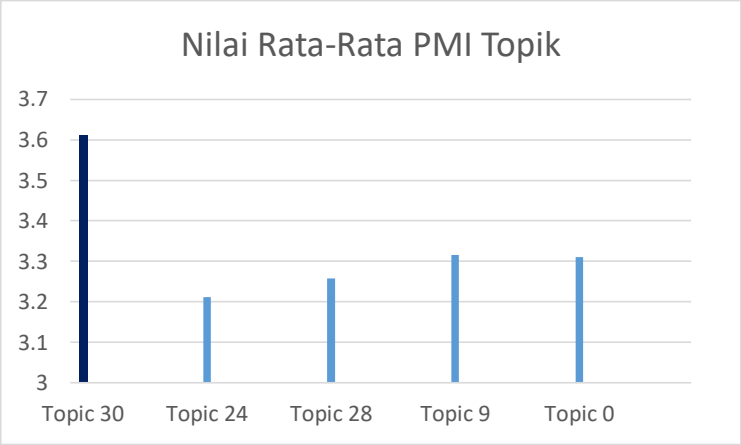
Gambar 6.45 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Frasa Batasan 3 – Stemming

Selanjutnya, dilakukan eksperimen pada skenario data frasa batasan 3 dengan *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data frasa batasan 3 dengan *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model rata-rata nilai PMI mengalami kenaikan dari topik 14 ke 16 dan mengalami penurunan dari topik 16 ke topik 18, dari topik 18 ke 23 nilai PMI kenaikan yang sangkat derastis dan dari topik 23 ke topik 28 nilai PMI mengalami penurunan yang cukup derastis sehingga didapatkan topik 23 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.198 yang menandakan topik 23 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.

6.7.6 Data Frasa Batasan 3 dengan Tanpa *Stemming*

Selanjutnya, dilakukan eksperimen pada skenario data frasa batasan 3 tanpa *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data frasa batasan 3 tanpa *stemming,* dari

kelima topik yang memiliki probabilitas tinggi dalam model nilai rata-rata PMI pada topik 0 dan 9 memiliki nilai yang stabil kemudian mengalami penurunan yang tidak begitu segnifikan pada topik 24, dari topik 24 ke topik 28 nilai PMI mengalami kenaikan yang tidak begitu besar dan dari topik 28 ke topik 30 nilai PMI mengalami peningkatan yang sangat drastis sehingga didapatkan topik 30 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.611 yang menandakan topik 30 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.

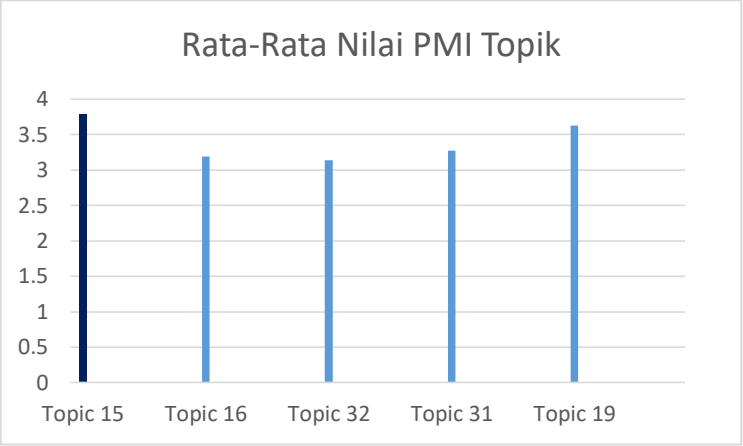


Gambar 6.46 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Frasa Batasan 3 – Tanpa *Stemming*

6.7.7 Data Tanpa Frasa Batasan 1 dengan *Stemming*

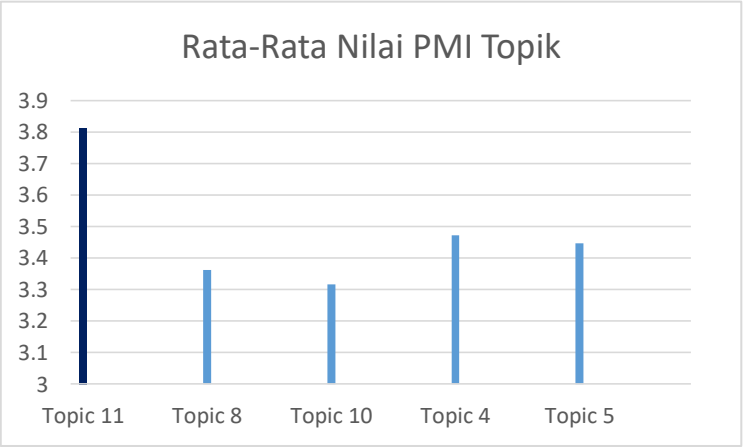
Selanjutnya, dilakukan eksperimen pada skenario data Tanpa Frasa batasan 1 dengan *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data Tanpa Frasa batasan 1 dengan *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model nilai rata-rata PMI mengalami penurunan dari topik 15 ke 16 dan kemudian mengalami kenaikan dari topik 15 ke topik 19 kemudian mengakami penurunan dari topik 19 ke - topik 31 dan 32 namun tidak begitu signifikan sehingga didapatkan topik 15 merupakan topik dengan nilai rata-rata PMI

tertinggi yaitu 3.790 yang menandakan topik ke-15 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.



Gambar 6.47 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Tanpa Frasa Batasan 1 – *Stemming*

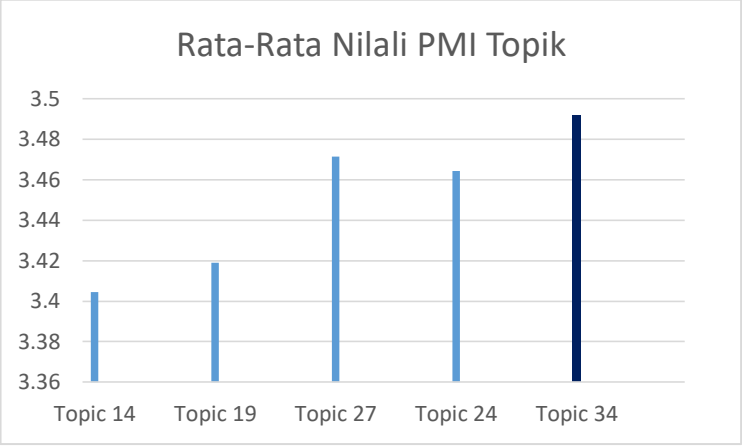
6.7.8 Data Tanpa Frasa Batasan 1 dengan Tanpa *Stemming*



Gambar 6.48 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Tanpa Frasa Batasan 1 – Tanpa *Stemming*

Selanjutnya, dilakukan eksperimen pada skenario data Tanpa Frasa batasan 1 tanpa *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 13 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data Tanpa Frasa batasan 1 tanpa *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model rata-rata nilai PMI mengalami kenaikan penurunan pada topik 4 ke topik 5, 8 dan 10 namun tidak signifikan, dari topik 10 ke topik 11 nilai PMI mengalami kenaikan yang cukup derastis sehingga didapatkan topik ke-10 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 4.751 yang menandakan topik ke-0 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.

6.7.9 Data Tanpa Frasa Batasan 2 dengan *Stemming*



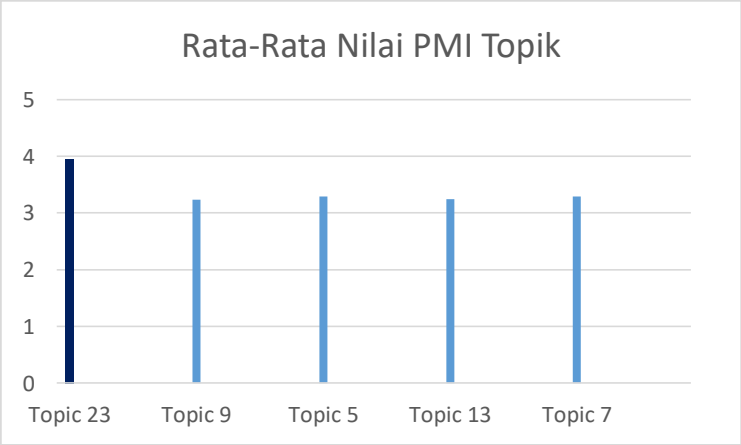
Gambar 6.49 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Tanpa Frasa Batasan 2 – *Stemming*

Selanjutnya, dilakukan eksperimen pada skenario data Tanpa Frasa batasan 2 dengan *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data Tanpa Frasa batasan 2 dengan *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model rata-rata nilai PMI mengalami kenaikan seiring

dengan bertambahnya topik yaitu dari topik 14 sampai dengan topik 34 sehingga didapatkan topik ke-34 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.491 yang menandakan topik ke-34 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.

6.7.10 Data Tanpa Frasa Batasan 2 Tanpa *Stemming*

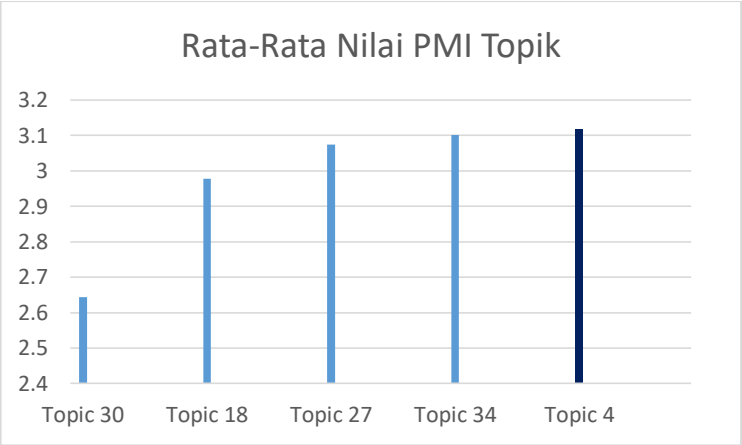
Selanjutnya, dilakukan eksperimen pada skenario data Tanpa Frasa batasan 2 tanpa *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data Tanpa Frasa batasan 2 tanpa *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model nilali rata-rata PMI stabil pada topik 5, 7, 9 dan 13 kemudian mengalami kenaikan pada topik 23 sehingga didapatkan topik ke-23 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.950 yang menandakan topik ke-23 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.



Gambar 6.50 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Tanpa Frasa Batasan 2 – Tanpa *Stemming*

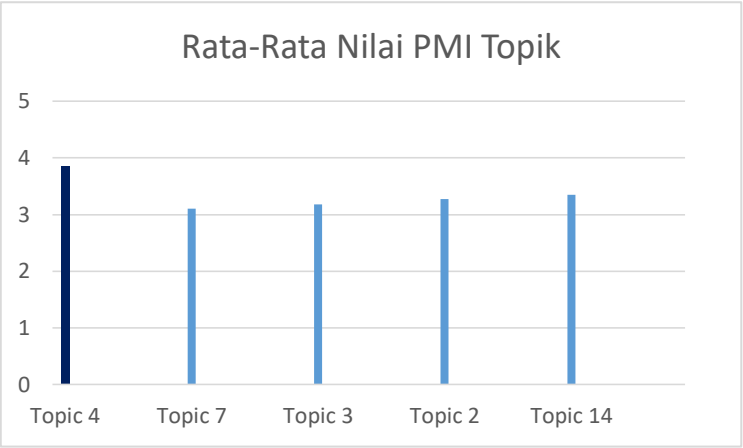
6.7.11 Data Tanpa Frasa Batasan 3 dengan *Stemming*

Selanjutnya, dilakukan eksperimen pada skenario data Tanpa Frasa batasan 3 dengan *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 35 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data Tanpa Frasa batasan 3 dengan *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model nilai rata-rata PMI mengalami penurunan dari topik 4 ke topik 18 keudian mengalami kenaikan yang tidak begitu signifikan ke topik 27 namun mengalami penurunan yang sangat derastis pada topik 30 sehingga didapatkan topik ke-4 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.118 yang menandakan topik ke-4 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.



Gambar 6.51 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Tanpa Frasa Batasan 3 – *Stemming*

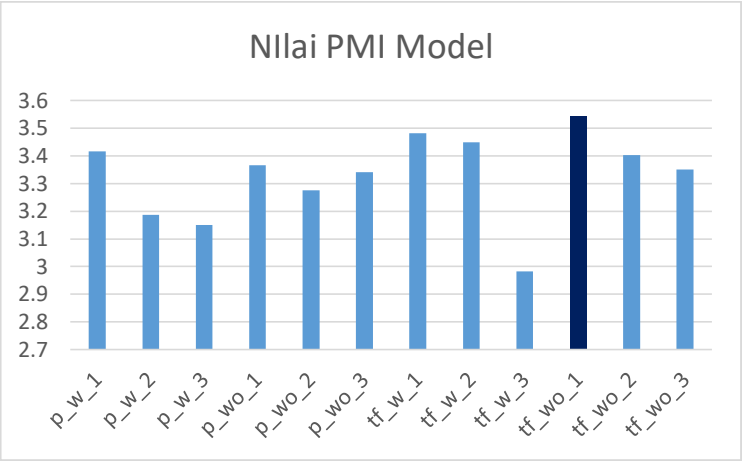
6.7.12 Data Tanpa Frasa Batasan 3 dengan Tanpa *Stemming*



Gambar 6.52 Hasil Rata-Rata Nilai PMI masing-masing Topik pada Data Tanpa Frasa Batasan 3 – Tanpa *Stemming*

Selanjutnya, dilakukan eksperimen pada skenario data Tanpa Frasa batasan 3 tanpa *stemming,* dimana model terbaik yang dihasilkan dari perhitungan *perplexity* memiliki jumlah topik sebanyak 25 topik. Dari hasil eksperimen didapatkan bahwa pada skenario model data Tanpa Frasa batasan 3 tanpa *stemming,* dari kelima topik yang memiliki probabilitas tinggi dalam model nilai rata-rata PMI mengalami kenaikan dari topik 3 ke topik 4 kemudian menurun dari topik 4 ke topik 7, nilai rata-rata untuk topik 2, 3, 7 dan 14 stabil sehingga didapatkan topik ke-4 merupakan topik dengan nilai rata-rata PMI tertinggi yaitu 3.853 yang menandakan topik ke-4 merupakan topik yang memiliki kata-kata yang paling mirip di dalamnya.

Dari seluruh hasil analisis nilai PMI per topik pada setiap skenario model, selanjutnya dilakukan analisis secara keseluruhan pada hasil perhitungan nilai PMI per model untuk menentukan model mana yang memiliki topik dengan kata-kata yang paling mirip dan dapat dimengerti oleh manusia.



Gambar 6.53 Hasil Rata-Rata Nilai PMI masing-masing Model

Analisis ini dilakuakan dengan menghitung nilai rata-rata dari seluruh nilai PMI topik yang ada di dalam masing-masing model, kemudian nilai PMI model ini dibandingkan antara model yang satu dengan yang lainnya. Hasil analisis nilai PMI per model ditunjukkan pada gambar 6.53.

Dari gambar dapat disimpulkan bahwa untuk semua scenario data nilai PMI akan mengalami penurunan seiring dengan bertambahnya nilai dari batasan bawah yang telah ditentukan di awal dan model dengan jumlah topik 13 pada skenario data Tanpa Frasa batasan 1 tanpa *stemming* merupakan model dengan nilai PMI tertinggi sehingga dapat dikatakan bahawa model memiliki topik dengan kata-kata yang paling mirip. Dan jika.

6.8 Pengujian Model dengan *Pointwise Mutual Information*

Penggujian model dengan menggunakan metode *pointwise mutual information* dilakukan dengan membandingkan nilai PMI dari model terbaik yang dihasilkan oleh ke dua metode yaitu *Gaussian LDA* dan *LDA*. Dimana model terbaik dari metode *LDA* adalah model dengan jumlah topik 4 dan model

terbaik dari *Gaussian LDA* adalah model dengan jumlah topik 13.

Dalam pengujian ini terdapat 4 skenario yang digunakan yaitu: *Gaussian LDA* dengan 4 dan 13 topik serta *LDA* dengan 4 dan 13 topik. Pembentukan 4 skenario ini berdasarkan pada jumlah topik yang dihasilkan dari model terbaik masing-masing metode, sehingga nanti akan dibandingkan hasil perhitungan nilai PMI dari masing-masing skenario untuk mendapatkan model terbaik.

6.8.1 Analisis Kuantitatif

Untuk melakukan perhitungan PMI, masing-masing topik dalam model dipilih 5 kata dengan probabiltas tertinggi untuk menghitung nilai PMI.

Tabel 6.10 Nilai PMI per Topik *Gaussian LDA* dan *LDA* dengan Jumlah Topik 4

Topik 0	Topik 1	Topik 2	Topik 3
Gaussian LDA			
taxi	tolong	petugas	support
compliment	membantu	penerbangan	security
cleanliness	biasakan	berjalan	international
polite	kecewa	ruangan	airport
avsec	marah	aktifitas	understand
3.462	3.483	3.350	3.473
LDA			
gate	taksi	parkir	internet
in	taxi	jam	sedia
check	toilet	ruang	surabaya
layan	harga	berangkat	jalan
jam	layan	informasi	airport
3.453	3.513	3.380	3.318

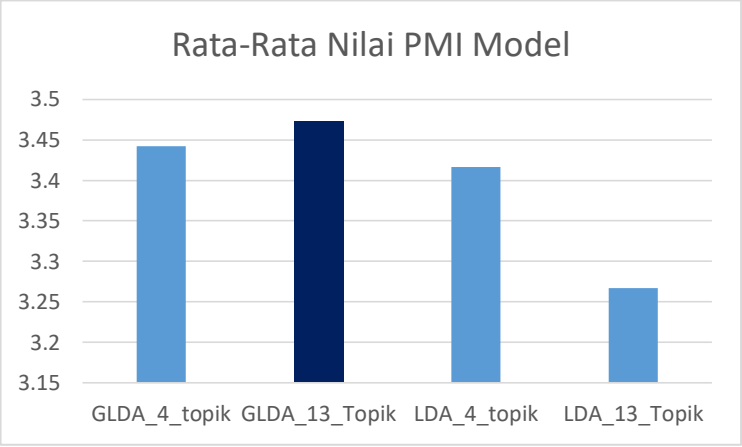
Pada tabel 6.7, ditampilkan 5 kata dengan probabilitas tertinggi dari skenario *Gaussian LDA* dan *LDA* dengan jumlah topik 4. Jika dilihat pada nilai PMI yang dimiliki oleh masing-masing topik pada ke dua metode, dapat disimpulkan bahwa metode *Gaussian LDA* memiliki nilai PMI per topik yang lebih tinggi dibandingkan dengan nilai PMI per topik pada metode LDA. Hal ini menunjukkan bahwa model dari metode *Gaussian LDA* dengan jumlah topik 4 memiliki *topic coherence* atau kemiripan antar kata dalam topik yang lebih baik dibandingkan dengan model dari metode *LDA*.

Pada tabel 6.8, ditampilkan 5 kata dengan probabilitas tertinggi pada masing-masing topik dari skenario *Gaussian LDA* dan *LDA* dengan jumlah topik 13. Dalam tabel ditampilkan juga nilai dari perhitungan PMI dari masing-masing topik dari kedua model. Dari hasil perhitungan topik didapatkan bahwa nilai PMI topik pada model denngan metode *Gaussian LDA* memiliki nilai yang lebih tinggi dibandingkan dengan nilai PMI topik dari model *LDA*. Hal ini menunjukkan bahwa model dengan metode *Gaussian LDA* dengan jumlah topik 13 memiliki kemiripan antar kata yang lebih baik dibandingkan dengan model dari metode *LDA*.

Tabel 6.11 Nilai PMI per Topik *Gaussian LDA* dan *LDA* dengan Jumlah Topik 13

Topik 0	Topik 1	Topik 2	Topik 3	Topik 4	Topik 5	Topik 6	Topik 7	Topik 8	Topik 9	Topik 10	Topik 11	Topik 12
Gaussian LDA												
tindak	polite	bayi	samsung	pelayanan	kecewa	rokok	petugasnya	ahmad	shalat	pemberitahuan	penerbangan	parkir
narkotika	boardingpass	sakit	phone	sesuai	marah	makan	bantuan	fata	bersuci	status	airlines	motor
kantor	paperless	inap	mobile	pengelola	tertawa	minum	imigrasi	khoirul	muslim	memeriksa	airways	sepeda
prosedur	security	hamil	laptop	kerja	terkejut	ikan	keamanan	abdul	jamaah	diperbolehkan	jet	inap
operasional	trolli	cacat	iphone	kualitas	bodoh	enak	penjaga	fatah	musholla	konfirmasi	qantas	tiket
3.569	3.449	3.490	3.488	3.477	3.476	3.481	3.468	3.461	3.464	3.449	3.445	3.435
LDA												
fasilitas	delay	laku	orang	bersih	check	parkir	jam	gate	air	fasilitas	in	taksi
bagus	informasi	in	rokok	nyaman	in	motor	layan	ruang	lounge	ramah	check	kerja
ruang	air	antri	area	room	tolong	barang	ruang	check	informasi	saran	nama	tawar
kursi	maskapai	berangkat	jemput	prekitiu	security	mobil	berangkat	in	wifi	duduk	boarding	ya
senang	kompensasi	layan	bawa	layan	layan	area	internet	jam	gate	air	airport	air
3.505	3.383	3.316	3.323	3.261	3.264	3.256	3.229	3.184	3.186	3.187	3.189	3.190

Selain melakukan perhitungan nilai PMI per topik, dilakukan juga perhitungan nilai PMI model dari masing-masing skenario untuk mengetahui model mana yang memiliki nilai PMI lebih tinggi. Visualisasi hasil perhitungan nilai PMI model ini ditunjukkan dalam gambar 6.64.



Gambar 6.54 Nilai PMI Model dengan Metode *Gaussian LDA* dan *LDA*

Dari gambar 6.54 didapatkan nilai PMI model skenario *Gaussian LDA* dengan jumlah topik 4 yaitu sebesar 3.442 dan dengan jumlah topik 13 sebesar 3.473. Untuk skenario *LDA* dengan jumlah topik 4 yitu sebesar 3.416 dan dengan jumlah topik 14 sebesar 3.267. Dari gambar dapat disimpulkan bahwa model yang memiliki nilai PMI tertinggi adalah model dengan metode *Gaussian LDA* jumlah topik 13 yaitu sebesar 3.473.

6.8.2 Analisis Kualitatif

Berdasarkan dari tabel 6.7 dan tabel 6.8, dapat dilihat 5 kata-kata dengan probabilitas tertinggi pada masing-masing topik dari ke dua model. Didapatkan bahwa model dengan menggunakan metode *Gaussian LDA* dapat mengelompokkan kata-kata dengan kemiripan semantic dengan baik, misalkan saja pada model dengan jumlah topik 13 pada tabel 6.8, pada topik ke 8 model dapat mengelompokkan kata-kata yang dapat

dianggap sebagai kelompok nama-nama orang, kemudian pada topik ke 5 model dapat mengelompokkan yang dapat dianggap sebagai kelompok kata perasaan. Dibandingkan dengan model yang didapakan dari metode *LDA* meskipun kata terkelompokkan dengan baik namun masih ada kata yang masuk ke dalam lebih dari 1 topik kemudian pada topik ke 6 tabel 6.8 kelompok kata yang dapat dikatakan sebagai topik kendaraan namun karena adanya kata *barang* maka membuat topik tersebut menjadi ambigu.

Namun, secara keseluruhan baik model yang dihasilkan dengan menggunakan metode *Gaussian LDA* dan *LDA* menghasilkan model yang cukup baik dalam mengelompokkan kata ke dalam topik. Kemampuan *Gaussian LDA* mengelompokkan topik berdasarkan kemiripan semantik dari kata-kata juga memiliki sisi negative yaitu adanya topik yang tidak infromatif jika digunakan oleh perusahaan untuk dianalisis seperti topik dengan kelompok kata nama orang.

6.8.3 Analisis Hasil

Setelah mendapatkan model terbaik dari pengujian model, selanjutnya dilakukan klasifikasi data ke dalam topik. Berdasarkan keterkaitannya dengan data Angkasa Pura. Berikut Topik-topik hasil analisis yang ada di data Angkasa Pura:

1. Fasilitas
    - Kursi
    - Kamar Mandi
    - Monitor
    - Trolli
    - X-Ray
    - Fasilitas *Disable*
    - Lampu
    - Pengeras Suara
    - *Waiting Room*
    - *Musholla*
    - *Internet Corner*

2. Layanan
    - Check-in
    - Petunjuk Arah
    - CS
    - Wifi
3. Petugas
    - Keamanan
    - Pemeriksaan
    - Kenyamanan
    - Peraturan
    - Kesopanan
4. Transportasi
    - Taksi
    - Bus
    - Parkiran

Untuk melakukan pemetaan topik-topik pada model terbaik yang dihasilkan dari pengujian model akan dilakukan pencarian *similarity word* dari masing-masing topik yang ada dalam model untuk memperkaya kata dalam topik , hasil dari pencarian *similarity word* ditunjukkan dalam tabel 22.

Tabel 6.12 *Similarity Word* per Topik

TOPIC 0: ('kumdil', 'lpdb-kumkm', 'penasihatan', 'pelaksanaan', 'kewenangan', 'bakamla', 'wewenang', 'bppspam', 'perundang-undangan')
TOPIC 1: ('singapore', 'centre', 'education', 'international', 'organization', 'schools', 'boarding', 'initiative', 'trade')
TOPIC 2: ('inap', 'rawat', 'poliklinik', 'klinik', 'perawatan', 'gawat', 'bersalin', 'ugd', 'sakit')
TOPIC 3: ('smartphone', 'ponsel', 'handphone', 'ipad', 'kitkat', 'iphone', 'miui', 'laptop', 'konektifitas')
TOPIC 4: ('ramah', 'bersih', 'nyaman', 'menyenangkan', 'hubungan', 'penting', 'memperhatikan', 'kualitas', 'rapi')
TOPIC 5: ('tentunya', 'panas', 'sampah', 'menyebalkan', 'sangatlah', 'kotor', 'memperhatikan', 'cukup', 'terjaga')

TOPIC 6: ('minuman', 'makanan', 'makan', 'masak', 'kudapan', 'kue-kue', 'lezat', 'sarapan', 'sosis')

TOPIC 7: ('petugas', 'polisi', 'melapor', 'aparat', 'narapidana', 'orpo', 'penggeledahan', 'penahanan', 'eksekusi')

TOPIC 8: ('rialdy', 'darmawan', 'soenardi', 'soediro', 'widjanarko', 'hidayati', 'setyowati', 'didu', 'evy')

TOPIC 9: ('shalat', 'salat', 'sholat', 'tarawih', 'berjama', 'dhuha', 'dhuhur', 'dzikir', 'tahlil')

TOPIC 10: ('memeriksa', 'semiperlindungan', 'pemberitahuan', 'pengaju', 'pengajuan', 'bersangkutan', 'prosedur', 'amandemen', 'klarifikasi')

TOPIC 11: ('penerbangan', 'maskapai', 'airlines', 'airways', 'berjadwal', 'airblue', 'atlasjet', 'sewaan', 'cargolux')

TOPIC 12: ('bus', 'keberangkatan', 'penumpang', 'taksi', 'stasiunnya', 'bongkar-muat', 'pulang-pergi', 'loket', 'diparkir')

- Untuk topik 0 : dilihat dari kata-kata yang menyusun topik ke – 0 maka dapat diasumsikan topik 0 masuk ke dalam label Petugas – Peraturan
- Untuk topik 1 : dilihat dari kata-kata yang menyusun topik ke – 1 yang membahas prilaku, petugas dan tujuan penerbangan maka topik 1 diasumsikan masuk ke dalam label Petugas - Kesopanan.
- Untuk topik 2 : dilihat dari kata-kata yang menyusun topik ke – 2 yang membahas tentang bayi, hamil, perawatan, penginapan dan cacat maka dapat diasumsikan topik 2 masuk ke dalam label fasilitas – fasilitas *disable.*
- Untuk topik 3 : dilihat dari kata-kaya yang menyusun topik ke – 3 yang membahas tentang perangkat keras dan *merk handphone* maka diasumsikan topik ke – 3 tidak masuk ke dalam ke tiga label yang ada.
- Untuk topik 4 : dilihat dari kata-kata yang menyusun topik ke – 4 yang membahas tentang pelayanan yang positif maka dapat diasumsikan masuk ke label petugas – kenyamanan.
- Untuk topik 5 : dilihat dari kata-kata yang menyusun topik ke – 5 yang membahas tentang perasaan yang negative maka dapat diasumsikan tidak masuk ke dalam ke tiga label yang ada.

- Untuk topik 6 : dilihat dari kata-kata yang menyususn topik ke – 6 yang membahas tentang makanan maka dapat diasumsikan topik 6 tidak masuk ke label manapun.
- Untuk topik 7 : dilihat dari kata-kata yang menyusun topik ke – 7 yang membahas tentang pelayanan oleh petugas maka dapat diasumsikan topik 7 masuk ke dalam label petugas – keamanan
- Untuk topik 8 : dilihat dari kata-kata yang menyusun topik ke – 8 yang membahas tentang nama-nama orang maka diasumsikan topik 8 tidak masuk ke dalam label manapun.
- Untuk topik 9 : dilihat dari kata-kata yang menyusun topik ke – 9 yang membahas tentang *sholat, wudhu dan musholla* maka dapat diasumsikan topik 9 masuk ke dalam label fasilitas – musholla.
- Untuk topik 10 : dilihat dari kata-kata yang menyusun topik ke – 10 yang membahas tentang pemeriksaan maka diasumsikan topik 10 masuk ke dalam label petugas – pemeriksaan.
- Untuk topik 11 : dilihat dari kata-kata yang menyusun topik ke – 11 yang membahas tentang penerbangan dan maskapai maka dapat diasumsikan topik ke 11 tidak masuk ke dalam label manapun.
- Untuk topik 12 : dilihat dari kata-kata yang menyusun topik ke – 12 yang membahas tentang parkiran, motor dan mobil maka dapat diasumsikan topik 12 masuk ke dalam label transportasi – parkiran.

Dari hasil pemetaan topik ke dalam hasil analisis topik yang ada di dalam data Angkasa Pura maka disimpulkan bahwa terdapat topik yang memiliki kata yang sangat spesifik sehingga tidak dapat dimasukkan ke dalam label hasil analisis dan topik-topik tersebut tidak informatif jika akan digunakan oleh perusahaan.

Dari hasil pengujian dengan menggunakan data testing didapatkan 73 data dari 100 data *testing* yang memiliki label tepat. Data yang memiliki perbedaan label adalah sebanyak 27 data. Sehingga, jika diukur dengan menggunakan persentase, akurasi dari data adalah

$$Akurasi = \frac{Jumlah\ data\ tepat\ berlabel}{total\ seluruh\ data} \times 100\ \% \quad (1)$$

Dari rumus 1 didapatkan akurasi 73%. Dan untuk tingkat *error* dari data adalah 27%. Hal ini dipengaruhi oleh beberapa faktor. Faktor utama yang mempengaruhi nilai akurasi adalah distribusi topik dalam setiap dokumen. Sebuah dokumen memiliki kemungkinan untuk memiliki beberapa jumlah topik. Hal ini memiliki pengaruh pada nilai probabilitas. Faktor kedua adalah pelabelan pada topik berdasarkan pada kata-kata penyusun topik, semakin baik dan cocok pemeberian label pada topik berdasarkan kata-kata dalam topik maka akan semakin baik pula nilai yang dimiliki. Hasil Dari pengujian ini ditunjukkan dalam Lampiran E-1.

BAB VII
KESIMPULAN DAN SARAN

Pada bab ini dibahas mengenai kesimpulan dari semua proses yang telah dilakukan dan saran yang dapat diberikan untuk pengembangan yang lebih baik.

7.1 Kesimpulan

Kesimpulan yang didapatkan dari proses pengerjaan tugas akhir yang telah dilakukan antara lain:

1. Dari eksperimen penentuan jumlah *passes* pada LDA didapatkan jumlah *passes* sebanyak 18 *passes* untuk semua skenario data.
2. Dari ekspermen jumlah topik didapatkan model terbaik dari *LDA* dengan jumlah topik sebesar 4 pada skenario data Tanpa Frasa dengan *stemming* batasan 1 dan memiliki nilai *topic coherence* sebesar -5.326.
3. Dari eksperimen penentuan iterasi dan topik pada *Gaussian* didapatkan iterasi dapat berjumlah 1 dst, dan jumlah topik sebanyak 13 untuk skenario data Tanpa Frasa tanpa *stem* dengan batasan 1, 25 topik untuk skenario data Tanpa Frasa tanpa *stem* dengan batasan 3 dan 35 topik untuk semua skenario yang lain.
4. Dari hasil validasi model *Gaussian LDA* dengan menggunakan PMI didapatkan skenario data Tanpa Frasa tanpa *stem* dengan batasan satu yang memiliki jumlah topik 13 merupakan model terbaik dari *Gaussian LDA* dengan nilai PMI paling tinggi yaitu 3.482.
5. Dari hasil pengujian model didapatkan bahwa model dengan metode *Gaussian LDA* pada skenario data Tanpa Frasa batasan 1 tanpa *stemming* dengan jumlah topik sebanyak 13 topik merupakan model yang memiliki nilai *PMI* model yang paling tinggi yaitu 3.473 diantara semua

model yang diuji. Maka dapat disimpulkan bahwa model *Gaussian LDA* memiliki kata-kata dalam topik dengan kemiripan yang lebih baik atau lebih mudah diinterpretasikan dibandingkan dengan model *LDA*.

6. Dari analisis hasil didapatkan bahwa akurasi model terbaik yang didapatkan dari pemodelan dengan *Gaussian LDA* dalam melakukan pemodelan topik yaitu sebesar 73% dan dengan tingkat *error* sebesar 27% dari 100 data *testing* yang digunakan.

7.2 Saran dan Penelitian Selanjutnya

Dari pengerjaan tugas akhir ini, adapun beberapa saran untuk pengembangan penelitian ke depan.

1. *Formalizer* yang dilakukan pada pra proses data menggunakan *repository* yang disususn oleh Purwarianti [15] masih tidak dapat melakukan formalisasi pada data secaa optimal sehingga dibutuhkan perbendaharaan kata yang lebih banyak lagi agar dapat mencakup seluruh kata yang ada di dalam data.
2. *Checking* dan pendeteksian kata berbahasa inggris dalam data yang dilakukan dengan menggunakan library *enchant* masih tidak bisa melakukan pendeteksian kata berbahasa inggris secara optimal sehingga dibutuhkan pendefinisan *list* atau daftar kata berbahasa inggris yang lebih banyak agar dapat mencakup seluruh kata yang ada di dalam data.
3. *Stemming* yang dilakukan pada tahap pra proses data menggunakan library sastrawi masih tidak dapat melakukan *stemming* secara optimal sehingga dibutuhkan perbendaharaan kata untuk anti*stem* yaitu kata-kata yang tidak berhasil di*stemming* oleh sastrawi.
4. Model *word embedding* yang digunakan untuk menjalankan pemodelan dengan *Gaussian LDA* merupakan model yang disusun oleh Purwarianti [15] dengan data set dari Wikipedia Indonesia, sehingga terdapat banyak kata yang tidak tercakup di dalam model, dan penulis model

word embedding yang akan digunakan pada pemodelan dengan *Gaussian LDA* sebaiknya ditraining sendiri untuk memperkaya model.

5. Penentuan jumlah topik pada metode *Gaussian LDA* masih menggunakan cara yang sama dengan metode *LDA* yaitu dengan menghitung nilai *perplexity* dari model, namun cara ini menurut penulis masih tidak dapat merepresentasikan jumlah topik yang paling baik untuk metode *Gaussian LDA* sehingga diperlukan pencarian metode dengan studi literature untuk dapat menentukan jumlah topik terbaik dari pemodelan dengan menggunakan *Gaussian LDA*.
6. Perhitungan nilai PMI yang dilakukan pada penelitian ini menggunakan data hasil *scrapping* dari 23 halaman di *facebook.* Data ini menurut penulis masih kurang banyak untuk dapat mencakup seluruh kata-kata dalam topik sehingga dibututhkan proses *scrapping* data yang lebih banyak lagi.
7. Pemberian label untuk analisis topik dilakukan oleh penulis sendiri sehingga masih kurang baik jika dijadikan acuan label untuk setiap topik pada data *testing,* sehingga selanjutnya diharapkan pelabelan dapat dilakukan oleh ahli dibidangnya.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

[1] "PT Angkasa Pura I (Persero)." [Online]. Available: https://www.ap1.co.id/id/about/visi-misi. [Accessed: 04-Dec-2017].

[2] S. H. Aminah and Z. Muhibah, *MIGRASI APLIKASI VOICE OF CUSTOMER BANDAR UDARA INTERNASIONAL JUANDA SURABAYA PADA PT. ANGKASA PURA 1 (PERSERO)*, vol. 1. 2015.

[3] "Text Mining 101: Topic Modeling." [Online]. Available: https://www.kdnuggets.com/2016/07/text-mining-101-topic-modeling.html. [Accessed: 04-Feb-2018].

[4] R. Das, M. Zaheer, and C. Dyer, "Gaussian LDA for Topic Models with Word Embeddings," pp. 795–804, 2015.

[5] D. Mimno, H. M. Wallach, E. Talley, and M. Leenders, "Optimizing Semantic Coherence in Topic Models," no. 2, pp. 262–272, 2011.

[6] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM : Topic Modeling over Short Texts," vol. 26, no. 12, pp. 2928–2941, 2014.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "US," 2003.

[8] M. Steyvers and T. Griffiths, "Probabilistic Topic Models," 2007.

[9] "Topic Modeling." [Online]. Available: http://mallet.cs.umass.edu/topics.php. [Accessed: 06-Feb-2018].

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," vol. 3, pp. 993–1022, 2003.

[11] D. M. Blei and J. D. Lafferty, "A correlated topic model of Science," *Ann. Appl. Stat.*, vol. 1, no. 1, pp. 17–35, 2007.

[12] J. Stolee, "An Evaluation of Topic Modelling Techniques for Twitter," pp. 1–11, 2016.

[13] J. Chang *et al.*, “Reading Tea Leaves : How Humans Interpret Topic Models,” 2009.

[14] D. Newman, Y. Noh, E. Talley, S. Karimi, and T. Baldwin, “Evaluating topic models for digital libraries,” *Proc. 10th ACM/IEEE-CS Jt. Conf. Digit. Libr.*, pp. 215–224, 2010.

[15] A. Purwarianti, A. Andhika, A. F. Wicaksono, I. Afif, and F. Ferdian, “InaNLP : Indonesia Natural Language Processing Toolkit Case study : Complaint Tweet Classification,” pp. 5–9, 2016.

[16] V. Srividhya and R. Anitha, “Evaluating preprocessing techniques in text categorization,” *Int. J. Comput. Sci. Appl.*, no. 2010, pp. 49–51, 2010.

BIODATA PENULIS

Penulis lahir di Praya pada tanggal 17 Maret 1996. Penulis merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh beberapa pendidikan formal yaitu, SDN 4 Sengkol, MTs Dakwah Ismaliyah Putri Kediri, dan MA Dakwah Ismaliyah Putri Kediri dan pendidikan non-formal di Pondok Pesantren Nurul Hakim Kediri Lombok Barat.

Pada tahun 2014 setelah kelulusan SMA, penulis melanjutkan pendidikan dengan jalur PBSB (Program Beasiswa Santri Berprestasi) di Jurusan Sistem Informasi FTIK – Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan terdaftar sebagai mahasiswa dengan NRP 05211440007002. Selama menjadi mahasiswa, penulis aktif berorganisasi di Himpunan Mahasiswa Sistem Informasi, CSSMoRA ITS, Kajian Sistem Infromasi, dan Jamaah Masjid Manarul Ilmi.

Pada tahun keempat, karena penulis memiliki ketertarikan di bidang *Natural Language Processing*, maka penulis mengambil bidang minat Akuisisi Data dan Diseminasi Informasi (ADDI). Penulis dapat dihubungi melalui *email* di zuyyinahilya56@gmail.com.

Halaman ini sengaja dikosongkan

LAMPIRAN A

A-1. Hasil Eksperimen Penentuan Jumlah *Passes* berdasarkan Nilai *Perplexity* – Data Tanpa Frasa Batasan 1 – *Stemming*

Percobaan	Jumlah Topik					
	2	5	8	10	15	20
iterasi 1	339.5	662.9	1526.3	2895.3	1572	339.5
iterasi 2	165.3	190.5	198.4	209.4	258.7	165.3
iterasi 3	161	182.1	186.2	197.8	234.4	161
iterasi 4	159.1	178.4	180.8	191.7	222.5	159.1
iterasi 5	158	176.3	177.9	188	215.4	158
iterasi 6	157.3	174.9	176.1	185.7	210.9	157.3
iterasi 7	156.6	174	174.8	184.1	207.8	156.6
iterasi 8	156.2	173.4	174	183.1	205.6	156.2
iterasi 9	155.9	172.9	173.3	182.3	204.1	155.9
iterasi 10	155.7	172.6	172.7	181.7	202.9	155.7
iterasi 11	155.5	172.3	172.3	181.2	201.9	155.5
iterasi 12	155.3	172	172	180.7	201	155.3
iterasi 13	155.2	171.8	171.7	180.3	200.5	155.2
iterasi 14	155.1	171.6	171.4	180	199.9	155.1
iterasi 15	154.9	171.4	171.2	179.8	199.4	154.9
iterasi 16	154.8	171.3	171	179.5	199	154.8
iterasi 17	154.7	171.2	170.9	179.3	198.7	154.7
iterasi 18	154.6	171	170.7	179.1	198.4	154.6
iterasi 19	154.5	170.9	170.6	178.9	198.1	154.5

Percobaan	Jumlah Topik					
	2	5	8	10	15	20
iterasi 20	154.4	170.8	170.5	178.7	197.9	154.4
iterasi 21	154.3	170.8	170.4	178.5	197.7	154.3
iterasi 22	154.2	170.7	170.3	178.3	197.5	154.2
iterasi 23	154	170.6	170.2	178.1	197.3	154
iterasi 24	153.9	170.5	170.1	178	197.1	153.9
iterasi 25	153.8	170.5	170	177.9	196.9	153.8
iterasi 26	153.8	170.4	169.9	177.8	196.9	153.8
iterasi 27	153.7	170.3	169.9	177.7	196.7	153.7
iterasi 28	153.7	170.3	169.8	177.5	196.6	153.7
iterasi 29	153.6	170.2	169.7	177.4	196.6	153.6
iterasi 30	153.6	170.2	169.7	177.3	196.5	153.6
iterasi 31	153.5	170.1	169.7	177.3	196.4	153.5
iterasi 32	153.5	170.1	169.6	177.1	196.3	153.5
iterasi 33	153.4	170	169.6	177	196.2	153.4
iterasi 34	153.4	169.9	169.5	177	196.1	153.4
iterasi 35	153.3	169.9	169.5	176.8	196	153.3
iterasi 36	153.3	169.9	169.4	176.8	195.9	153.3
iterasi 37	153.2	169.8	169.4	176.7	195.8	153.2
iterasi 38	153.2	169.8	169.4	176.6	195.7	153.2
iterasi 39	153.1	169.7	169.3	176.5	195.7	153.1
iterasi 40	153.1	169.7	169.3	176.4	195.6	153.1
iterasi 41	153.1	169.7	169.2	176.4	195.6	153.1
iterasi 42	153	169.7	169.2	176.3	195.5	153
iterasi 43	153	169.6	169.2	176.2	195.4	153
iterasi 44	153	169.6	169.2	176.2	195.4	153
iterasi 45	152.9	169.6	169.1	176.1	195.3	152.9

Percobaan	Jumlah Topik					
	2	5	8	10	15	20
iterasi 46	152.9	169.6	169.1	176	195.3	152.9
iterasi 47	152.9	169.5	169.1	176	195.2	152.9
iterasi 48	152.9	169.5	169.1	175.9	195.1	152.9
iterasi 49	152.8	169.5	169.1	175.9	195.1	152.8
iterasi 50	152.8	169.5	169	175.8	195.1	152.8
iterasi 51	152.8	169.4	169	175.8	195	152.8
iterasi 52	152.7	169.4	169	175.8	195	152.7
iterasi 53	152.7	169.4	169	175.7	195	152.7
iterasi 54	152.7	169.4	169	175.7	194.9	152.7
iterasi 55	152.6	169.4	169	175.7	194.9	152.6
iterasi 56	152.6	169.3	169	175.6	194.9	152.6
iterasi 57	152.6	169.3	169	175.6	194.9	152.6
iterasi 58	152.6	169.3	168.9	175.5	194.9	152.6
iterasi 59	152.5	169.3	168.9	175.5	194.8	152.5
iterasi 60	152.5	169.3	168.9	175.4	194.7	152.5
iterasi 61	152.5	169.2	168.9	175.4	194.7	152.5
iterasi 62	152.5	169.2	168.9	175.3	194.7	152.5
iterasi 63	152.4	169.2	168.9	175.3	194.6	152.4
iterasi 64	152.4	169.2	168.8	175.3	194.6	152.4
iterasi 65	152.4	169.2	168.8	175.2	194.5	152.4
iterasi 66	152.4	169.2	168.8	175.2	194.5	152.4
iterasi 67	152.4	169.1	168.8	175.2	194.5	152.4
iterasi 68	152.3	169.1	168.8	175.1	194.4	152.3
iterasi 69	152.3	169.1	168.8	175.1	194.4	152.3
iterasi 70	152.3	169.1	168.8	175.1	194.3	152.3
iterasi 71	152.3	169.1	168.8	175.1	194.3	152.3

Percobaan	Jumlah Topik					
	2	5	8	10	15	20
iterasi 72	152.3	169.1	168.8	175	194.3	152.3
iterasi 73	152.3	169.1	168.8	175	194.2	152.3
iterasi 74	152.2	169.1	168.7	175	194.2	152.2
iterasi 75	152.2	169.1	168.7	175	194.1	152.2
iterasi 76	152.2	169.1	168.7	175	194.1	152.2
iterasi 77	152.2	169.1	168.7	174.9	194.1	152.2
iterasi 78	152.2	169	168.7	174.9	194	152.2
iterasi 79	152.2	169	168.7	174.9	194	152.2
iterasi 80	152.2	169	168.6	174.8	194	152.2
iterasi 81	152.2	169	168.6	174.8	194	152.2
iterasi 82	152.1	169	168.6	174.8	194	152.1
iterasi 83	152.1	169	168.6	174.8	193.9	152.1
iterasi 84	152.1	169	168.6	174.8	193.9	152.1
iterasi 85	152.1	169	168.6	174.8	193.9	152.1
iterasi 86	152.1	169	168.6	174.7	193.9	152.1
iterasi 87	152.1	168.9	168.6	174.7	193.8	152.1
iterasi 88	152.1	168.9	168.6	174.7	193.8	152.1
iterasi 89	152.1	168.9	168.5	174.6	193.8	152.1
iterasi 90	152.1	168.9	168.5	174.6	193.8	152.1
iterasi 91	152.1	168.9	168.5	174.6	193.7	152.1
iterasi 92	152	168.9	168.5	174.6	193.7	152
iterasi 93	152	168.9	168.5	174.6	193.7	152
iterasi 94	152	168.9	168.5	174.5	193.7	152
iterasi 95	152	168.9	168.5	174.5	193.7	152
iterasi 96	152	168.9	168.5	174.5	193.7	152
iterasi 97	152	168.9	168.5	174.5	193.6	152

Percobaan	Jumlah Topik					
	2	5	8	10	15	20
iterasi 98	152	168.9	168.5	174.5	193.6	152
iterasi 99	152	168.9	168.5	174.5	193.6	152
iterasi 100	152	168.8	168.5	174.5	193.6	152

A-2. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Tanpa Frasa Batasan 1 – *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	165.3	173.2	180.7	170.5	184.1	185.6
iterasi 2	165.5	173.7	177.1	181.2	186.7	191.1
iterasi 3	167.2	170.5	180.2	181.3	184.1	191.7
iterasi 4	160.6	177.8	173.9	179.7	192.5	184.7
iterasi 5	162	172.8	182.6	184.1	174.8	188.7
iterasi 6	163.4	173.2	180.2	180.4	184.3	182.9
iterasi 7	166.2	169.3	177.2	181.4	188.1	188.6
iterasi 8	164.1	170.5	178.9	183.8	184.1	186.2
iterasi 9	166.5	176.1	182.7	180.1	188.1	188.8
iterasi 10	165.1	172.4	179	180.7	186.5	188.1

A-3. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Tanpa Frasa Batasan 1 – Tanpa *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	216.1	228.8	238.6	236.1	246.1	247.7
iterasi 2	211.1	224.9	232.4	240.2	243.5	243.9
iterasi 3	212.9	226.7	235.8	235	239.8	238.6
iterasi 4	211.3	224	236.6	240.2	240.5	243
iterasi 5	217.4	226.3	236.6	240.1	242.6	248.1
iterasi 6	211.9	224.9	234.9	238.8	242.7	238.3
iterasi 7	214.4	224.6	236.2	239.6	245.5	246.3
iterasi 8	213.8	225.2	229.9	240.6	241	247.7
iterasi 9	215.3	225.7	234.5	239.5	239.2	244.8
iterasi 10	214.4	224.6	227.5	238.6	241.2	251.3

A-4. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Tanpa Frasa Batasan 2 – *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	164.8	173	178.1	179	178	190.3
iterasi 2	159.5	174.5	181.7	184.3	176.3	190.2
iterasi 3	165.1	171.4	176	182.5	174.1	185.2
iterasi 4	159.7	177.6	179	184.3	173.9	192.6
iterasi 5	163.9	167.9	182	179	173.2	190.2
iterasi 6	162.5	175.5	183.3	181.4	172.4	186.8

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 7	166.7	172.3	181.1	179.3	174	189.6
iterasi 8	166.8	171.8	180.8	183.1	172.4	188.2
iterasi 9	165.9	174.3	177.5	178.7	177.8	184.1
iterasi 10	163.1	173.2	178.5	179.2	175	189.7

A-5. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Tanpa Frasa Batasan 2 – Tanpa *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	216.9	229.7	238.6	236.1	246.1	247.7
iterasi 2	212.4	226.5	232.4	240.2	243.5	243.9
iterasi 3	213.5	228.6	235.8	235	239.8	243
iterasi 4	212.6	224.9	236.6	240.2	240.5	248.1
iterasi 5	218.2	227.4	236.6	240.1	242.6	246.6
iterasi 6	213.1	226.1	234.9	238.8	242.7	238.3
iterasi 7	215.1	224.6	236.2	239.6	245.5	246.3
iterasi 8	215	225.2	229.9	240.6	241	247.7
iterasi 9	216.3	225.7	234.5	239.5	239.2	244.8
iterasi 10	215.3	224.6	227.5	238.6	241.2	251.3

A-6. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Tanpa Frasa Batasan 3 – *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	70.6	74.1	76	76.1	76.5	79.7
iterasi 2	71.5	72.7	77.3	75	77.8	81.5
iterasi 3	70.6	74.3	77.1	76	77.4	77.9
iterasi 4	70.6	71.8	76	75.3	76.5	82.1
iterasi 5	72.1	74.5	75.1	76.8	75.1	77.3
iterasi 6	71	72.3	74.6	75.8	77.2	77.8
iterasi 7	70.8	73.8	76	75.4	78.5	78.8
iterasi 8	71.8	73.4	77	75.6	77.1	77.8
iterasi 9	72.5	72.9	76	76.2	77.1	77.1
iterasi 10	70.4	74.1	75	75.3	80.5	78.6

A-7. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Tanpa Frasa Batasan 3 – Tanpa *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	78	79.9	82.1	82.1	77.1	86.7
iterasi 2	77.7	79.4	83.5	81.3	75.6	86.6
iterasi 3	76.6	79.3	85.9	83.8	75	84.5
iterasi 4	77.7	78.6	83.4	82.8	75.7	85.3
iterasi 5	77.8	79.6	81.8	84.4	76.2	88
iterasi 6	74.4	81.5	85.3	82.3	74.8	85.7

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 7	77.2	79.6	86.9	84	75	85
iterasi 8	76.5	82.8	82.2	81	74.6	87.3
iterasi 9	77.7	79.3	83.6	83	74.8	83.1
iterasi 10	76.1	80.5	80.4	84.4	75	86.4

A-8. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Frasa Batasan 1 – *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	169.4	180.2	190.9	185.7	189.1	198.8
iterasi 2	166.3	179	185.9	187.1	196.6	197.6
iterasi 3	165.8	175.6	184	186.7	193.7	200.8
iterasi 4	173.8	177.8	190.7	185.8	196.1	197.9
iterasi 5	171.9	184.4	183.8	187.5	197.2	195.9
iterasi 6	170.6	179.6	185.6	186.3	190.8	195.5
iterasi 7	172.6	183.1	187.8	187.6	195.5	192.6
iterasi 8	171.6	180.9	186.6	186.5	190.7	198.7
iterasi 9	174.1	180.4	193.1	190.9	196.8	197.8
iterasi 10	170	181	188.6	191	193.2	202.1

A-9. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Frasa Batasan 1 – Tanpa *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	229.1	241.6	245.7	251	252.9	260.3
iterasi 2	230.7	237.4	246.9	251.8	252.8	261.8
iterasi 3	221.4	241.2	248.2	246.6	252.7	258.2
iterasi 4	222.1	236.9	249.4	250	257.9	260.8
iterasi 5	225.8	235	246.1	256.1	259	257.3
iterasi 6	224.6	238.1	248.6	251.4	254.1	260.9
iterasi 7	225.4	243.4	245	252.1	257.1	256.8
iterasi 8	218.1	241.4	246.2	247	254.1	263.2
iterasi 9	228.1	237.7	250.3	251.4	254.9	260.6
iterasi 10	228.1	234.6	243.5	245.4	261	255

A-10. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Frasa Batasan 2 – *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	97.4	97.1	106.7	111	110.5	114.3
iterasi 2	93.9	102.4	104.9	105.8	105.9	108.3
iterasi 3	95.5	99.8	104.2	107.1	108.2	108.8
iterasi 4	95.6	100.3	104	110.2	107.4	113.5
iterasi 5	98.9	100.9	108	106.7	105.8	111.2
iterasi 6	95.5	99.9	103.5	103.7	107.1	106.4
iterasi 7	97.6	101.5	105.1	104.4	106.3	113.1
iterasi 8	93.8	103.1	107.9	104.7	107.7	111.4

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 9	95.4	100.7	104.5	105.2	106.6	107.2
iterasi 10	94.3	103.1	104.9	104.6	106.8	108.4

A-11. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Frasa Batasan 2 – Tanpa *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	114.5	122.4	128.1	133	131.5	138.7
iterasi 2	117.5	122.9	127.8	128.8	131	133.4
iterasi 3	116.9	120.1	134.6	134.2	132	140.4
iterasi 4	118.9	126.4	130.4	129.2	136.8	138.8
iterasi 5	115.1	123.8	128.7	128.9	135.4	135.9
iterasi 6	111.4	123.1	125.6	129.1	137.5	137.1
iterasi 7	114.4	122.2	131	127.7	137.1	139.7
iterasi 8	115.4	121.4	130.9	128.2	132.3	136.2
iterasi 9	111.4	120.9	133.5	128.6	138.9	138.3
iterasi 10	115.4	123.2	128.7	135.7	136.7	135.5

A-12. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Frasa Batasan 3 – *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	73.3	77.1	78.1	78	80.1	84.5

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 2	73.5	75.9	76.7	81.8	80.5	83.2
iterasi 3	73.9	76.2	77	81	81.8	81.8
iterasi 4	73.2	75.1	80.8	80.2	79.1	82.9
iterasi 5	72.8	77.2	79.1	76.9	82.3	83.9
iterasi 6	72.4	77.2	77.8	77.5	82.9	83.4
iterasi 7	73.5	78.6	77.8	78.1	79.2	82.1
iterasi 8	72.9	75.5	80.8	82.3	77.8	82.3
iterasi 9	72.6	77.3	78	76.6	81.9	82.5
iterasi 10	71.6	75.5	78.8	80.5	81.2	84.6

A-13. Hasil Eksperimen Penentuan Jumlah Topik berdasarkan Nilai *Perplexity* – Data Frasa Batasan 3 – Tanpa *Stemming*

Percobaan	Jumlah Topik					
	4	7	10	11	13	15
iterasi 1	79.4	83.7	86	87.5	88.8	90.6
iterasi 2	79	85.5	86.7	87.4	89.7	92.2
iterasi 3	78.9	80.1	83	88.4	87.6	90
iterasi 4	81.2	85.5	84.1	84.8	90.7	90.1
iterasi 5	79	85.5	87.5	87.7	90.8	95.2
iterasi 6	79.4	81.9	86.3	89.8	90.4	88.2
iterasi 7	79.5	81.9	86.1	88.2	87.3	91.5
iterasi 8	80.7	85.7	84.8	86.1	84.6	94.1
iterasi 9	79	82.9	86.6	88.8	89.1	89.7
iterasi 10	81.7	83.9	89.4	86.1	91	92.3

LAMPIRAN B

B-1. Daftar 4 Topik – Data Tanpa Frasa – *Stemming*

topic number: 0	topic number: 1	topic number: 2	topic number: 3
in , 0.00918322	tunggu , 0.012773	parkir , 0.0143211	the , 0.00777872
surabaya , 0.00861584	ruang , 0.0106198	petugas , 0.0141088	taksi , 0.00719175
wifi , 0.0082885	gate , 0.00990948	penerbangan , 0.00933832	and , 0.00713339
jam , 0.00827839	barang , 0.00946906	in , 0.00906447	taxi , 0.00702927
orang , 0.00803515	jam , 0.00896219	check , 0.00801028	airport , 0.0067851
ya , 0.00790744	tolong , 0.00831775	gate , 0.00780221	delay , 0.00612421
penerbangan , 0.00757428	air , 0.00821051	motor , 0.00711582	to , 0.00573241
nya , 0.00717607	petugas , 0.0069727	area , 0.00699578	in , 0.00508261
pelayanan , 0.00717132	ya , 0.00622255	taksi , 0.00620402	petugas , 0.00507826
min , 0.00710691	penerbangan , 0.00620084	keberangkatan , 0.00611337	internet , 0.0048797

B-2. Daftar 4 Topik – Data Tanpa Frasa – Tanpa *Stemming*

topic number: 0	topic number: 1	topic number: 2	topic number: 3
parkir , 0.027928	in , 0.0191051	jam , 0.0124808	taksi , 0.0183405
gate , 0.0143201	check , 0.0161089	layan , 0.0116151	delay , 0.0106779

motor , 0.0140082	tiket , 0.00943767	tolong , 0.0101511	taxi , 0.0102968
berangkat , 0.0129588	layan , 0.0090392	ruang , 0.0100202	ya , 0.010172
ya , 0.00978069	ruang , 0.00897861	bagus , 0.00971493	berangkat , 0.00893768
air , 0.00828976	the , 0.00895119	bersih , 0.00931966	nyaman , 0.00790827
area , 0.00708227	orang , 0.00744853	nya , 0.00843957	informasi , 0.00780106
min , 0.0068281	gate , 0.00688384	nyaman , 0.00839691	jadwal , 0.00774952
jalan , 0.00681116	laku , 0.00619424	air , 0.00835189	jam , 0.00729461
pagi , 0.00660417	surabaya , 0.00615616	surabaya , 0.00685071	fasilitas , 0.00670704

B-3. Daftar 7 Topik – Data Tanpa Frasa – Tanpa *Stemming*

topic number: 0	topic number: 1	topic number: 2	topic number: 3
parkir , 0.0150022	the , 0.021049	layan , 0.0146077	berangkat , 0.0229072
orang , 0.0130398	taksi , 0.0186731	taxi , 0.0126702	wifi , 0.0164547
tiket , 0.0117216	and , 0.0145282	barang , 0.0101364	jadwal , 0.0134037
ya , 0.0104796	to , 0.0145056	jam , 0.00969156	jam , 0.0126715
langsung , 0.00978536	airport , 0.00942018	harga , 0.00924026	delay , 0.0122823
tahan , 0.00974211	internet , 0.00856425	nya , 0.00910021	ya , 0.0107204
bersih , 0.00970933	on , 0.00841885	tuju , 0.00824169	informasi , 0.0105114

toilet , 0.00879826	is , 0.00795493	surabaya , 0.00728165	ruang , 0.0104887
tolong , 0.00858273	very , 0.00790431	in , 0.00644272	orang , 0.00873206
in , 0.00823735	prekitiu , 0.0069209	bawa , 0.00619015	gate , 0.00737961

topic number: 4	topic number: 5	topic number: 6
fasilitas , 0.0189558	parkir , 0.0206048	air , 0.0176139
sedia , 0.0170013	check , 0.0201526	delay , 0.0159892
nyaman , 0.0123764	in , 0.0198567	berangkat , 0.0138929
bagus , 0.0121749	min , 0.0155523	jam , 0.0120821
layan , 0.0113869	ruang , 0.0153532	lion , 0.0115283
ruang , 0.0104921	motor , 0.0142361	gate , 0.00912219
kursi , 0.00929442	bantu , 0.0117322	in , 0.00823859
bersih , 0.0086467	ya , 0.0111355	mandi , 0.00799671
gate , 0.00840627	gate , 0.0100199	kamar , 0.00799664
surabaya , 0.00801445	laku , 0.00945458	asia , 0.00693891

B-4. Daftar 11 Topik – Data Tanpa Frasa – *Stemming*

topic number: 6	topic number: 1	topic number: 3	topic number: 5	topic number: 8
motor , 0.00802691	the , 0.0225374	wifi , 0.0227095	jam , 0.0167392	taksi , 0.0187965
barang , 0.00779182	and , 0.0109308	orang , 0.0157857	jalur , 0.0122703	ruang , 0.0157707
petugas , 0.00708565	of , 0.0104106	penerbangan , 0.0134333	penerbangan , 0.0103643	penerbangan , 0.0116245
airport , 0.0070357	to , 0.00948845	in , 0.0129117	delay , 0.0101885	calo , 0.0112271
orang , 0.0070351	taksi , 0.00936198	air , 0.0125022	nya , 0.00895385	nyaman , 0.00766819
sepeda , 0.00703483	on , 0.00772855	check , 0.0113557	in , 0.00860193	petugas , 0.00678079
nya , 0.00686442	prekitiu , 0.00765858	handphone , 0.00851112	to , 0.00714479	tunggu , 0.00677802
parkir , 0.00618039	that , 0.00692167	ya , 0.00830213	petugas , 0.00633652	menunggu , 0.00622688
to , 0.00604362	gate , 0.00680747	maskapai , 0.00770776	very , 0.00610021	kali , 0.00579459
jalan , 0.00604201	is , 0.0060948	security , 0.00770605	tiket , 0.0060009	nya , 0.00568712
topic number: 0	topic number: 9	topic number: 4	topic number: 7	topic number: 10
petugas , 0.0147466	lounge , 0.0154738	min , 0.0247747	pelayanan , 0.0175372	ruang , 0.0245496
in , 0.0110229	dimana , 0.0133923	ya , 0.0189088	petugas , 0.0174076	tunggu , 0.022964
air , 0.0102818	tiket , 0.0113431	jam , 0.0167595	tunggu , 0.0127559	kursi , 0.0115694
taxi , 0.0085472	parkir , 0.0109529	banget , 0.0130405	gate , 0.0124231	jam , 0.0106369
tolong , 0.00843937	petugas , 0.0105985	tolong , 0.0121721	bagus , 0.0118382	gate , 0.0102976

harga , 0.00699846	gate , 0.00872566	parkir , 0.0116989	boarding , 0.0110342	fasilitas , 0.0101181
lion , 0.00684409	ya , 0.00862316	check , 0.0116157	in , 0.0104192	mandi , 0.00902372
check , 0.00684015	area , 0.00823499	internet , 0.0114449	jam , 0.0101379	kamar , 0.00902366
bagasi , 0.00630324	pengumuman , 0.00823301	in , 0.0100993	surabaya , 0.00968672	penerbangan , 0.00773629
barang , 0.00618067	lion , 0.00802989	motor , 0.00959742	pintu , 0.00960439	taksi , 0.0068867

B-5. Daftar 13 Topik – Data Tanpa Frasa – *Stemming*

topic number: 4	topic number: 7	topic number: 1	topic number: 10	topic number: 11
jam , 0.0216481	parkir , 0.0248584	taksi , 0.027096	in , 0.0217467	lounge , 0.0168565
parkir , 0.0133564	ya , 0.0189298	ya , 0.0125987	check , 0.0200434	jam , 0.0142854
check , 0.012866	tolong , 0.0124954	in , 0.0116394	orang , 0.0141586	jadwal , 0.0115795
delay , 0.0127168	calo , 0.0102145	delay , 0.00983744	petugas , 0.013263	gate , 0.0113754
penerbangan , 0.0106183	motor , 0.00995946	berangkat , 0.00914075	barang , 0.0117102	petugas , 0.0112073
orang , 0.0105664	min , 0.00931776	check , 0.00898618	penerbangan , 0.00984424	nya , 0.010545
pintu , 0.00891846	penerbangan , 0.00853522	wingsair , 0.00792926	pintu , 0.00813537	board , 0.00846781
in , 0.00787702	mengalami , 0.00835789	terbang , 0.00792625	security , 0.00798221	call , 0.00821153
motor , 0.0076459	koper , 0.00832676	alasan , 0.00662483	kalo , 0.00756685	jalur , 0.00741991
tiket , 0.00758987	menunggu , 0.00830915	anak , 0.00662483	gate , 0.00709516	wib , 0.00704837

topic number: 12	topic number: 9	topic number: 8	topic number: 6	topic number: 2
air , 0.0221071	the , 0.0202539	tunggu , 0.0548831	gate , 0.0102193	toilet , 0.0133562
petugas , 0.0197301	and , 0.0154436	ruang , 0.052409	airport , 0.0101937	gate , 0.0121279
in , 0.0124933	to , 0.01468	keberangkatan , 0.0178188	boarding , 0.00990703	prekitiu , 0.0114439
lion , 0.00967058	in , 0.01266	kursi , 0.0171234	surabaya , 0.00954422	penerbangan , 0.0104451
arah , 0.00891612	petugas , 0.0107583	jam , 0.0148738	time , 0.00906907	jam , 0.010411
pelayanan , 0.00891226	is , 0.00892227	disediakan , 0.0122312	bagus , 0.00811983	disediakan , 0.00971011
barang , 0.00875561	on , 0.00772589	duduk , 0.0112675	request , 0.00796659	ditambah , 0.00913016
check , 0.00841197	tolong , 0.00712355	gate , 0.0109702	special , 0.00796638	petugas , 0.00842815
surabaya , 0.00772954	mandi , 0.00702902	bagus , 0.00962429	jember , 0.00795893	jadwal , 0.0073401
penerbangan , 0.00752239	kamar , 0.0069305	penerbangan , 0.00888822	jam , 0.0076929	selamat , 0.00731103

B-6. Daftar 13 Topik – Data Tanpa Frasa – Tanpa *Stemming*

topic number: 4	topic number: 2	topic number: 11	topic number: 9	topic number: 8
layan , 0.0267368	taksi , 0.0407807	, 0.0183438	jam , 0.0375635	parkir , 0.0280369
air , 0.0231494	pintu , 0.0141235	sedia , 0.015927	informasi , 0.0158091	mobil , 0.0234405
lion , 0.0167505	calo , 0.0121492	room , 0.0138041	pagi , 0.0107122	jalur , 0.0175033
taxi , 0.0164594	min , 0.01088	fasilitas , 0.0137044	bus , 0.0104337	area , 0.0170972

bagasi , 0.0141973	tolong , 0.0100421	kursi , 0.0136999	nya , 0.010353	drop , 0.015198
harga , 0.00881329	nyaman , 0.00907164	khusus , 0.0134951	delay , 0.0101587	jalan , 0.0147973
kerja , 0.00823071	ruang , 0.00803714	informasi , 0.0134583	jadwal , 0.00994248	off , 0.0127194
surabaya , 0.00812397	tawar , 0.00797985	laptop , 0.0115326	toilet , 0.00932821	orang , 0.0117168
tolong , 0.00808573	jam , 0.00752834	smoking , 0.0115325	tolong , 0.00832441	macet , 0.0117121
jalan , 0.00762709	alat , 0.0065309	liat , 0.0113406	arah , 0.0080981	berangkat , 0.00900051
topic number: 1	topic number: 3	topic number: 0	topic number: 7	topic number: 6
ya , 0.0276456	wifi , 0.0305022	the , 0.0365185	parkir , 0.047876	internet , 0.0204298
tiket , 0.0202626	layan , 0.0285843	to , 0.0282468	motor , 0.032165	in , 0.0199638
barang , 0.0194035	nyaman , 0.0218789	and , 0.0257677	gate , 0.0157131	check , 0.0191418
kalo , 0.0112902	bersih , 0.0217784	of , 0.0162095	bantu , 0.0143521	ruang , 0.0187413
lihat , 0.0109212	bagus , 0.0195362	in , 0.0144369	lounge , 0.0123534	orang , 0.0177268
tolong , 0.0107815	nya , 0.0150646	that , 0.0119406	sepeda , 0.0120136	corner , 0.0142072
nyaman , 0.0104444	ramah , 0.0127624	is , 0.0111151	jam , 0.0113913	bantu , 0.0121963
cetak , 0.00979436	tahan , 0.011834	airport , 0.0104639	inap , 0.011351	security , 0.0100068
min , 0.00943115	tingkat , 0.00790507	my , 0.00853616	tolong , 0.0103756	handphone , 0.0100029
jalur , 0.00922645	fasilitas , 0.00787391	very , 0.00844902	tarif , 0.010319	fasilitas , 0.00903406

B-7. Daftar 4 Topik – Data Frasa – *Stemming*

topic number: 0	topic number: 1	topic number: 2	topic number: 3
parkir , 0.0137909	berangkat , 0.0166618	check_in , 0.0127123	layan , 0.0145925
barang , 0.0126768	jam , 0.0134113	taksi , 0.0116693	sedia , 0.0106581
area , 0.00821603	delay , 0.0129277	masuk , 0.01127	masuk , 0.0104441
bawa , 0.00809013	informasi , 0.0123293	orang , 0.00899288	bersih , 0.00950299
layan , 0.00754481	tunggu , 0.011671	taxi , 0.0087292	tiket , 0.0093267
nyaman , 0.00701424	parkir , 0.0115322	laku , 0.00803321	ya , 0.0079449
aman , 0.00614936	jadwal , 0.0112531	jam , 0.00765711	gate , 0.00714978
motor , 0.0060691	ya , 0.010402	gate , 0.00697592	ruang_tunggu , 0.00705423
and , 0.00566153	gate , 0.00955054	tolong , 0.00645383	surabaya , 0.00685947
air , 0.00560793	min , 0.00935153	boarding , 0.00618127	bagus , 0.00634532

B-8. Daftar 4 Topik – Data Frasa – Tanpa *Stemming*

topic number: 0	topic number: 1	topic number: 2	topic number: 3
taksi , 0.0113459	masuk , 0.0140651	wifi , 0.0117223	jam , 0.0133191
the , 0.0102783	petugas , 0.0105638	penerbangan , 0.0102402	gate , 0.011857
to , 0.00929091	pelayanan , 0.00730741	ya , 0.01002	penerbangan , 0.0118252

and , 0.00751994	parkir , 0.00634089	air , 0.00796138	petugas , 0.0109514
fasilitas , 0.00691179	taxi , 0.00620235	barang , 0.0066333	check_in , 0.00911373
parkir , 0.00581918	jalur , 0.00596507	nya , 0.00652355	delay , 0.00874484
petugas , 0.00559974	bagus , 0.00594713	petugas , 0.00644414	surabaya , 0.00669982
of , 0.00510596	gate , 0.00523911	min , 0.00611581	boarding , 0.00667285
bersih , 0.00475369	orang , 0.00521749	airport , 0.00554673	tolong , 0.00642141
harga , 0.00466632	tolong , 0.00512876	dimana , 0.00526511	masuk , 0.00621194

B-9. Daftar 7 Topik – Data Frasa – Tanpa *Stemming*

topic number: 0	topic number: 2	topic number: 4	topic number: 6
nya , 0.0160185	wifi , 0.0179698	masuk , 0.0139722	gate , 0.00924534
bagus , 0.0157027	jam , 0.0127624	ruang_tunggu , 0.0119676	air , 0.00882846
petugas , 0.0127622	surabaya , 0.0117715	petugas , 0.011176	parkir , 0.00853824
barang , 0.0112261	ya , 0.00932506	taksi , 0.011157	jam , 0.00844676
air , 0.00947792	lounge , 0.0091785	tolong , 0.0101324	penerbangan , 0.00835498
pelayanan , 0.00881588	boarding , 0.00777208	jam , 0.0084845	check_in , 0.00697645
taksi , 0.00846961	tiket , 0.0075444	gate , 0.0080595	kejadian , 0.00692498
imigrasi , 0.00691676	masuk , 0.00739537	penerbangan , 0.00765245	jalur , 0.00616451

ramah , 0.00642757	gate , 0.00695478	ditambah , 0.00758205	koper , 0.00616414
banget , 0.00626104	garuda , 0.00688988	duduk , 0.00735898	, 0.00616367
topic number: 1	topic number: 3	topic number: 5	
taxi , 0.0184762	penerbangan , 0.0123812	the , 0.0231073	
parkir , 0.0170211	check_in , 0.0111139	and , 0.0198075	
masuk , 0.0128034	petugas , 0.010919	to , 0.0189335	
tolong , 0.00917682	fasilitas , 0.0106049	is , 0.0108507	
ya , 0.00825048	delay , 0.00798985	on , 0.00840555	
barang , 0.00652637	prekitiu , 0.00752429	that , 0.00788996	
tas , 0.00596066	pelayanan , 0.00735395	my , 0.00670792	
terbang , 0.00596048	surabaya , 0.00687622	ticket , 0.00666072	
argo , 0.00596022	keberangkatan , 0.00653039	petugas , 0.00622032	
jalur , 0.00595904	security , 0.00616801	of , 0.0059354	

B-10. Daftar 10 Topik – Data Frasa – Tanpa *Stemming*

topic number: 0	topic number: 2	topic number: 4	topic number: 6	topic number: 8
taxi , 0.0101182	jam , 0.0153169	petugas , 0.0165172	delay , 0.0194894	petugas , 0.0197626
mahal , 0.00895824	parkir , 0.0149514	check_in , 0.00976289	area , 0.011678	dimana , 0.0103411

tolong , 0.00886975	nya , 0.0133975	orang , 0.00960627	tolong , 0.00907771	prekitiu , 0.00952583
taksi , 0.00844842	boarding , 0.0110094	penerbangan , 0.00956771	penerbangan , 0.0090114	ya , 0.00942628
penerbangan , 0.00842827	petugas , 0.0104707	mobil , 0.00715048	surabaya , 0.00885275	check_in , 0.00882191
ya , 0.00772097	penerbangan , 0.0104324	ya , 0.00693631	parkir , 0.00882247	gate , 0.00734967
lion_air , 0.00678254	masuk , 0.00940483	menunggu , 0.00655297	kali , 0.00834218	tiket , 0.00714897
keluhan , 0.00567091	surabaya , 0.00925682	security , 0.00650519	motor , 0.00812579	, 0.00695042
min , 0.00559427	check_in , 0.00780981	handphone , 0.0065037	ya , 0.00757556	tau , 0.00695003
pemberitahuan , 0.00544761	gate , 0.00698454	banget , 0.00630236	bersih , 0.00712634	parkir , 0.0064648
topic number: 1	topic number: 3	topic number: 5	topic number: 7	topic number: 9
masuk , 0.0206106	taksi , 0.0168082	tolong , 0.0116599	gate , 0.0129003	the , 0.0215514
antrian , 0.0102258	kursi , 0.0147956	penerbangan , 0.0114958	min , 0.0118925	jam , 0.0145443
ruang_tunggu , 0.00942236	duduk , 0.0123381	informasi , 0.00791387	petugas , 0.0106094	to , 0.013027
orang , 0.00931931	disediakan , 0.0121133	check_in , 0.0074434	lounge , 0.010495	and , 0.0125935
keberangkatan , 0.00893124	masuk , 0.0120398	hilang , 0.00693664	ya , 0.00998062	in , 0.00967749
pelayanan , 0.00726296	ruang_tunggu , 0.0110987	internet_corner , 0.00666291	jalur , 0.00988629	airport , 0.00833591
informasi , 0.00683244	orang , 0.0107194	nyaman , 0.00596032	kedatangan , 0.00931289	on , 0.0078608
tiket , 0.00656369	pagi , 0.0100347	air , 0.00595987	banget , 0.00874823	masuk , 0.00762697

petugas , 0.00626205	bagus , 0.00990043	tas , 0.00595957	menunggu , 0.00726079	of , 0.00730954
konsumen , 0.00626079	fasilitas , 0.00902307	hp , 0.00595948	fasilitas , 0.0069141	petugas , 0.00716607

B-11. Daftar 11 Topik – Data Frasa – *Stemming*

topic number: 7	topic number: 9	topic number: 0	topic number: 1	topic number: 10
petugas , 0.0131975	wifi , 0.0275198	keberangkatan , 0.0100825	duduk , 0.0145199	parkir , 0.0285868
gate , 0.0100343	surabaya , 0.00965468	taxi , 0.00908309	parkir , 0.0108834	penerbangan , 0.0149183
taxi , 0.00988537	nya , 0.00962374	pelayanan , 0.00908305	masuk , 0.0108279	mobil , 0.0115395
masuk , 0.00689927	pelayanan , 0.00960666	jam , 0.00808463	jam , 0.00995538	ya , 0.011439
boarding , 0.00661426	bagus , 0.00919364	wifi , 0.00771466	tolong , 0.0081996	petugas , 0.0110855
tas , 0.00660118	tolong , 0.00903463	tiket , 0.00768767	delay , 0.00674168	kursi , 0.00997859
orang , 0.00646898	penerbangan , 0.00847615	dimana , 0.00708952	nyaman , 0.00622535	ruang_tunggu , 0.00904279
powerbank , 0.00579701	nih , 0.00794988	surabaya , 0.00708497	orang , 0.00619441	jam , 0.00781088
jam , 0.00515282	boarding , 0.00772702	air , 0.00652954	menunggu , 0.00594013	sepeda_motor , 0.00771688
tolong , 0.00498028	request , 0.00704728	motor , 0.00646604	tunggu , 0.00585217	inap , 0.00767976
topic number: 2	topic number: 4	topic number: 8	topic number: 5	topic number: 6
delay , 0.0180899	the , 0.017864	masuk , 0.0233435	gate , 0.027073	taksi , 0.0172671
petugas , 0.0126661	and , 0.0131765	banget , 0.011922	check_in , 0.0111187	lounge , 0.0138755

taxi , 0.00956329	to , 0.0118967	min , 0.0109837	the , 0.00911249	petugas , 0.0123944
barang , 0.00744394	penerbangan , 0.0117144	ruang_tunggu , 0.0104892	of , 0.00911188	penerbangan , 0.0107057
tolong , 0.00640276	barang , 0.00804874	petugas , 0.00969519	toilet , 0.0071092	ya , 0.0101183
air , 0.00628316	prekitiu , 0.0073289	pintu_masuk , 0.00957896	service , 0.0071077	gate , 0.00740216
imigrasi , 0.00588654	in , 0.00724984	kamar_mandi , 0.00914792	angkasa , 0.0067846	layar , 0.00690067
konter , 0.00586849	is , 0.00714626	surabaya , 0.00880799	petugas , 0.00656309	pengumuman , 0.00689286
room , 0.00525184	on , 0.00649998	ya , 0.008602	tolong , 0.00625887	tiket , 0.00671699
smoking , 0.00525183	jam , 0.00541652	area , 0.00807941	customer , 0.00622685	informasi , 0.00639128

B-12. Daftar 13 Topik – Data Frasa – *Stemming*

topic number: 1	topic number: 2	topic number: 0	topic number: 12	topic number: 3
wifi , 0.0337135	berangkat , 0.0138003	masuk , 0.0322039	gate , 0.0225896	barang , 0.0323929
delay , 0.0199003	ruang_tunggu , 0.0137918	jam , 0.0159997	boarding , 0.0119042	bawa , 0.0246189
ya , 0.0173031	surabaya , 0.0135766	orang , 0.0141274	jemput , 0.0112147	tas , 0.0153466
nih , 0.0114588	check_in , 0.012349	taxi , 0.011672	toilet , 0.00949308	bayar , 0.0143373
trolli , 0.0100966	taksi , 0.0116505	laku , 0.0116474	tolong , 0.00905507	bagasi , 0.0134589
nyaman , 0.0100106	langsung , 0.0102163	antri , 0.0103984	nama , 0.00838822	masuk , 0.0091776
kali , 0.00928788	masuk , 0.00967299	ruang_tunggu , 0.0090676	ruang , 0.00786539	jalur , 0.00832036

sedia , 0.00815044	calo , 0.00959802	powerbank , 0.007456	lihat , 0.00765388	sedia , 0.00761769
air , 0.00791352	tinggal , 0.0085929	jemput , 0.00744525	area , 0.00746579	orang , 0.007289
wingsair , 0.00775867	tolong , 0.00850477	bawa , 0.00709612	ya , 0.00722011	saudara , 0.00728605
topic number: 5	topic number: 4	topic number: 7	topic number: 10	topic number: 11
layan , 0.0233088	parkir , 0.0695157	bersih , 0.0328963	check_in , 0.0206941	arah , 0.014428
tunggu , 0.0197405	motor , 0.0273285	bagus , 0.0266162	delay , 0.017979	wifi , 0.0137185
berangkat , 0.0162524	inap , 0.0185005	layan , 0.0257252	biar , 0.0136673	taksi , 0.0133677
gate , 0.015659	sepeda_motor , 0.0160641	nyaman , 0.0246066	air , 0.011452	internet_corner , 0.0117736
tiket , 0.0155097	ya , 0.0154945	fasilitas , 0.0234272	laku , 0.0105251	laku , 0.0117223
jam , 0.0124567	tarif , 0.0149201	kamar_mandi , 0.0167759	tau , 0.00979419	tunggu , 0.0090267
ya , 0.0114933	informasi , 0.0133533	sedia , 0.0154726	jam , 0.00964669	nya , 0.00872695
lounge , 0.0111161	bantu , 0.0124805	kursi , 0.0137119	min , 0.00932038	fasilitas , 0.00859584
kerja , 0.0111013	mobil , 0.0121203	tahan , 0.0125962	bantu , 0.00882314	jadwal , 0.00804153
jalan , 0.0104888	telpon , 0.00963053	banget , 0.0123662	bus , 0.00812789	air , 0.00769598

LAMPIRAN C

C-1. *Word Similarity* pada Skenario Data Tanpa Frasa batasan 1 – *Stemming*

TOPIC 0: ('unk', 'yukihiro', 'amott', 'youbi', 'dobro', 'siouxsie', 'tambourine', 'cherone', 'bennie')
TOPIC 1: ('kumdil', 'lpdb-kumkm', 'penasihatan', 'pelaksanaan', 'kewenangan', 'bakamla', 'wewenang', 'bppspam', 'perundang-undangan')
TOPIC 2: ('kakak', 'jae-yeol', 'ayah', 'roosminah', 'adik', 'teman', 'yoon-ha', 'il-ri', 'ibunya')
TOPIC 3: ('in', 'and', 'underdevelopment', 'peacebuilding', 'statelessness', 'of', 'with', 'lessons', 'between')
TOPIC 4: ('zapped', 'eye', 'gate', 'bulletproof', 'shakedown', 'climbers', 'breakers', 'five', 'goonies')
TOPIC 5: ('air', 'airnya', 'vegetasi', 'tawar', 'basah', 'subsiden', 'bebatuan', 'terendam', 'hujan')
TOPIC 6: ('polariskop', 'halfton', 'giroskop', 'accelerometers', 'berputar', 'statis', 'tachometer', 'penyetelan', 'geophone')
TOPIC 7: ('seseorang', 'sikap', 'hal-hal', 'menurutnya', 'akhlaknya', 'tegas', 'kenyataan', 'keinginan', 'sesuatu')
TOPIC 8: ('toilet', 'kantin', 'musholla', 'parkir', 'mushola', 'uks', 'ruang', 'aula', 'hotspot')
TOPIC 9: ('but', 'them', 'that', 'are', 'when', 'would', 'have', 'there', 'become')
TOPIC 10: ('linknya', 'sarankan', 'rapihkan', 'penghapusannya', 'informasikan', 'wiki-en', 'memvandal', 'tanyakan', 'bersediakah')
TOPIC 11: ('smartphone', 'ponsel', 'miui', 'voip', 'kitkat', 'handphone', 'mobomarket', 'viber', 'seluler')
TOPIC 12: ('mengkilap', 'bergaris-garis', 'roknya', 'tungging', 'kehijauan', 'berwarna', 'bercoret', 'pucat', 'menggarpu')
TOPIC 13: ('jam', 'pukul', 'sabtu', 'senin', 'rabu', 'jumat', 'senin-jumat', 'pagi', 'wib')
TOPIC 14: ('motor', 'mobil', 'mesinnya', 'chasis', '-silinder', 'mesin', 'berpenggerak', 'sasis', 'truk')
TOPIC 15: ('inap', 'rawat', 'pasien', 'poliklinik', 'klinik', 'perawatan', 'penderita', 'persalinan', 'tht')
TOPIC 16: ('manaqib', 'al-baihaqi', 'at-taqrib', 'utsaimin', 'mukarramah', 'asy-syaikh', 'al-bukhari', 'syaikh', 'anhuma')
TOPIC 17: ('kue-kue', 'minuman', 'kudapan', 'kaleng', 'botol', 'bungkusan', 'samgyeopsal', 'pemanggang', 'permen')
TOPIC 18: ('seluruh', 'pertama', 'tersebut', 'sementara', 'khusus', 'dipilih', 'kedua', 'diadakan', 'lainnya')
TOPIC 19: ('jalur', 'kereta', 'stasiun', 'bus', 'komuter', 'angkutan', 'citayam-nambo', 'cikampek-padalarang', 'gyeongbu')
TOPIC 20: ('jq', 'aj', 'uj', 'wt', 'wl', 'tq', 'wg', 'jf', 'vo')
TOPIC 21: ('rupanya', 'malah', 'teringat', 'takut', 'utty', 'irij', 'sindokht', 'rudabeh', 'ternyata')
TOPIC 22: ('rialdy', 'darmawan', 'soenardi', 'soediro', 'widjanarko', 'hidayati', 'setyowati', 'didu', 'evy')
TOPIC 23: ('layan', 'ruyung', 'kameloh', 'sanggan', 'kelekar', 'tuwung', 'hurun', 'duhung', 'niur')
TOPIC 24: ('harga', 'pembayaran', 'pembelian', 'jual', 'sewa', 'barang', 'cicilan', 'ongkos', 'rekening')
TOPIC 25: ('jalan-jalan', 'loket', 'dijalan', 'memarkir', 'lorong', 'parkiran', 'pintu', 'eskalator', 'ruangannya')

TOPIC 26: ('just', 'like', 'everytime', 'good', 'come', 'something', 'take', 'shout', 'feelings')
TOPIC 27: ('airlines', 'airways', 'maskapai', 'skymark', 'airblue', 'transaero', 'aircalin', 'qantaslink', 'atlasjet')
TOPIC 28: ('clbk', 'ketarik', 'ngojek', 'unyu', 'antri', 'pujaan', 'kepentok', 'jodoh', 'kupeluk')
TOPIC 29: ('kalo', 'sih', 'gimana', 'begini', 'udah', 'deh', 'biar', 'nyambung', 'iya')
TOPIC 30: ('memfasilitasi', 'kualitas', 'kebutuhan', 'spipise', 'frpba', 'lpdb-kumkm', 'solusi', 'penyediaan', 'efektivitas')
TOPIC 31: ('supadio', 'danskadron', 'danwing', 'kodikau', 'danlanud', 'wingdikterbang', 'pangkosekhanudnas', 'papok', 'sopsau')
TOPIC 32: ('surabaya', 'malang', 'semarang', 'salatiga', 'balikpapan', 'yogyakarta', 'bandung', 'surakarta', 'hangtuah')
TOPIC 33: ('redesign', 'roll-out', 'improvements', 'datavalues', 'access', 'deployed', 'multiple', 'capable', 'capabilities')
TOPIC 34: ('check', 'maxerrors', 'add', 'onprotectfailure', 'deleteprocessapi', 'metaargs', 'onmovefailure', 'errorcode', 'disambiguator')

Document-Topic Counts:, [566 136 247 396 398 290 257 376 377 268 431 134 126 283 180 37 45 156
531 96 180 359 68 221 357 559 248 127 342 493 316 79 181 357 161]

C-2. *Word Similarity* pada Skenario Data Tanpa Frasa batasan 1 - Tanpa *Stemming*

TOPIC 0: ('kumdil', 'lpdb-kumkm', 'penasihatan', 'pelaksanaan', 'kewenangan', 'bakamla', 'wewenang', 'bppspam', 'perundang-undangan')
TOPIC 1: ('singapore', 'centre', 'education', 'international', 'organization', 'schools', 'boarding', 'initiative', 'trade')
TOPIC 2: ('inap', 'rawat', 'poliklinik', 'klinik', 'perawatan', 'gawat', 'bersalin', 'ugd', 'sakit')
TOPIC 3: ('smartphone', 'ponsel', 'handphone', 'ipad', 'kitkat', 'iphone', 'miui', 'laptop', 'konektifitas')
TOPIC 4: ('ramah', 'bersih', 'nyaman', 'menyenangkan', 'hubungan', 'penting', 'memperhatikan', 'kualitas', 'rapi')
TOPIC 5: ('tentunya', 'panas', 'sampah', 'menyebalkan', 'sangatlah', 'kotor', 'memperhatikan', 'cukup', 'terjaga')
TOPIC 6: ('minuman', 'makanan', 'makan', 'masak', 'kudapan', 'kue-kue', 'lezat', 'sarapan', 'sosis')
TOPIC 7: ('petugas', 'polisi', 'melapor', 'aparat', 'narapidana', 'orpo', 'penggeledahan', 'penahanan', 'eksekusi')
TOPIC 8: ('rialdy', 'darmawan', 'soenardi', 'soediro', 'widjanarko', 'hidayati', 'setyowati', 'didu', 'evy')
TOPIC 9: ('shalat', 'salat', 'sholat', 'tarawih', 'berjama', 'dhuha', 'dhuhur', 'dzikir', 'tahlil')
TOPIC 10: ('memeriksa', 'semiperlindungan', 'pemberitahuan', 'pengaju', 'pengajuan', 'bersangkutan', 'prosedur', 'amandemen', 'klarifikasi')
TOPIC 11: ('penerbangan', 'maskapai', 'airlines', 'airways', 'berjadwal', 'airblue', 'atlasjet', 'sewaan', 'cargolux')
TOPIC 12: ('bus', 'keberangkatan', 'penumpang', 'taksi', 'stasiunnya', 'bongkar-muat', 'pulang-pergi', 'loket', 'diparkir')

Document-Topic Counts:, [811 648 784 789 782 822 697 860 697 769 1001 677 758]

C-3. *Word Similarity* pada Skenario Data Tanpa Frasa batasan 2 – *Stemming*

TOPIC 0: ('rokok', 'minuman', 'makan', 'makanan', 'berjualan', 'minum', 'menjajakan', 'makanan-makanan', 'kudapan')
TOPIC 1: ('angkring', 'tunggu', 'mujirun', 'tongsis', 'bisa', 'jemput', 'menunggu', 'sediakan', 'tawarkan')
TOPIC 2: ('unk', 'amott', 'dobro', 'yukihiro', 'cherone', 'bennie', 'siouxsie', 'tambourine', 'youbi')
TOPIC 3: ('bikin', 'maumu', 'ketemu', 'bilang', 'gak', 'cari', 'siapa-siapa', 'lupa', 'enggak')
TOPIC 4: ('cikampek-padalarang', 'jalur', 'bus', 'busway', 'rute', 'citayam-nambo', 'stasiun', 'jalan', 'cikampek-cirebon')
TOPIC 5: ('check', 'add', 'maxerrors', 'needed', 'date-holding', 'metaargs', 'onprotectfailure', 'deleteprocessapi', 'exists')
TOPIC 6: ('surabaya', 'malang', 'semarang', 'balikpapan', 'makassar', 'manado', 'jakarta', 'salatiga', 'denpasar')
TOPIC 7: ('when', 'but', 'them', 'that', 'would', 'still', 'have', 'there', 'are')
TOPIC 8: ('wi-fi', 'wifi', 'laptop', 'handphone', 'koneksi', 'smartphone', 'terkoneksi', 'wireless', 'konektifitas')
TOPIC 9: ('in', 'and', 'of', 'underdevelopment', 'peacebuilding', 'cosmopolitanism', 'between', 'lessons', 'statelessness')
TOPIC 10: ('menyenangkan', 'tentunya', 'memperhatikan', 'jujur', 'sifatnya', 'hal-hal', 'menurutnya', 'sikap', 'seseorang')
TOPIC 11: ('jadwal', 'pertama', 'kali', 'dimulai', 'bulan', 'minggu', 'berikutnya', 'setiap', 'waktu')
TOPIC 12: ('pintu', 'lokasinya', 'gerbang', 'tempat', 'menjorok', 'lokasi', 'letak', 'diatas', 'ungapan')
TOPIC 13: ('lpdb-kumkm', 'pengawasan', 'pelaksanaan', 'kumdil', 'kerja', 'bppspam', 'frpba', 'instansi', 'koordinasi')
TOPIC 14: ('angkasa', 'suborbital', 'berawak', 'soyuz', 'gsat-', 'irs-p', 'pesawat', 'shuttle', 'spaceplane')
TOPIC 15: ('mengarsipkan', 'masukan', 'letterater', 'kuesioner', 'data-data', 'konten-konten', 'penggunanya', 'pengunggahan', 'backlink')
TOPIC 16: ('delay', 'continuous', 'built-in', 'contactless', 'interoperability', 'direct', 'boost', 'high-level', 'frequency')
TOPIC 17: ('layan', 'perapat', 'ruyung', 'niur', 'suaq', 'tuwung', 'lapai', 'merawa', 'kelekar')
TOPIC 18: ('jam', 'pagi', 'pukul', 'sabtu', 'sore', 'senin', 'siang', 'jumat', 'malam')
TOPIC 19: ('pembayaran', 'rekening', 'pembelian', 'harga', 'tagihan', 'barang', 'transaksi', 'sewa', 'cicilan')
TOPIC 20: ('jt', 'jq', 'jf', 'qk', 'jz', 'jn', 'xw', 'qe', 'qg')
TOPIC 21: ('gimana', 'sih', 'tolong', 'kalo', 'perbaiki', 'begini', 'udah', 'iya', 'ngusulin')
TOPIC 22: ('dijahit', 'berbentuk', 'kotak-kotak', 'coretan-coretan', 'direntangkan', 'helai-helai', 'selutut', 'menggantung', 'pengganjal')
TOPIC 23: ('to', 'some', 'many', 'articleplaceholder', 'through', 'also', 'was', 'on', 'with')
TOPIC 24: ('irij', 'sindokht', 'rudabeh', 'hendak', 'rupanya', 'sangmaima', 'minuchihr', 'mammu', 'zohak')
TOPIC 25: ('airport', 'shahjalal', 'hub', 'benina', 'mehrabad', 'zvartnots', 'airports', 'ghaydah', 'suvarnabhumi')
TOPIC 26: ('stand', 'up', 'comedy', 'kompas', 'combreak', 'indo', 'komunitas', 'finalis', 'vyna')
TOPIC 27: ('air', 'airnya', 'subsiden', 'basah', 'salinisasi', 'limpasan', 'tawar', 'salinitas', 'penguapan')

TOPIC 28: ('boarding', 'ratmalana', 'suvarnabhumi', 'internasional', 'cardig', 'stn', 'shahjalal', 'kualalumpur', 'changi')
TOPIC 29: ('mobil', 'truk', 'motor', 'pengemudi', 'taksi', 'bagasi', 'limusin', 'pengendara', 'mesinnya')
TOPIC 30: ('parkir', 'toilet', 'mushola', 'musholla', 'kantin', 'ruang', 'aula', 'uks', 'parkiran')
TOPIC 31: ('html', 'min', 'programmainlist', 'cineseoul', 'imgmovie', 'nhn', 'plvijynmnlvnogphlvtixduqbjum', 'http', 'details')
TOPIC 32: ('sementara', 'orang', 'erwiana', 'bablo', 'semuanya', 'orangtua', 'sendiri', 'dibully', 'satu-satunya')
TOPIC 33: ('zapped', 'perfect', 'night', 'heaven', 'eye', 'train', 'posterjpeg', 'climbers', 'goonies')
TOPIC 34: ('maskapai', 'airlines', 'airways', 'cargolux', 'penerbangan', 'berjadwal', 'atlasjet', 'cargo', 'skymark')

Document-Topic Counts:, [40 4002 70 207 99 76 83 145 71 221 181 140 93 103 33
201 84 38 90 146 14 192 107 97 185 29 9 110 66 72
174 58 267 208 32]

C-4. *Word Similarity* pada Skenario Data Tanpa Frasa batasan 2 – Tanpa *Stemming*

TOPIC 0: ('dial-up', 'konektifitas', 'speakerphone', 'modem', 'bandwidth', 'set-top', 'hardisk', 'multisentuh', 'swype')
TOPIC 1: ('menemui', 'hendak', 'terkejut', 'menunggu', 'tiba-tiba', 'mammu', 'terpaksa', 'encup', 'mubids')
TOPIC 2: ('them', 'are', 'that', 'which', 'some', 'but', 'more', 'have', 'this')
TOPIC 3: ('pengawasan', 'pelayanan', 'hhi', 'penegakan', 'lpdb-kumkm', 'badan-badan', 'pelaksanaan', 'bppspam', 'publik')
TOPIC 4: ('penerbangan', 'maskapai', 'airlines', 'berjadwal', 'airways', 'sewaan', 'airblue', 'atlasjet', 'bertarif')
TOPIC 5: ('boarding', 'buwitashakti', 'pbkl', 'pahoa', 'haskam', 'ppsp', 'pskd', 'sbi', 'prosia')
TOPIC 6: ('tolong', 'perbaiki', 'bersediakah', 'harap', 'birukan', 'terimakasih', 'semoga', 'sepertinya', 'gimana')
TOPIC 7: ('anak', 'kakak', 'adik', 'ibunya', 'suami', 'ayah', 'perempuannya', 'orangtua', 'perempuan')
TOPIC 8: ('diletakkan', 'hiasan', 'jendela', 'berjajar', 'berbentuk', 'menggantung', 'lampu', 'dijahit', 'dihiasi')
TOPIC 9: ('cowok', 'cewek', 'kepentok', 'clbk', 'ngojek', 'jodoh', 'ketarik', 'gara-gara', 'cinta')
TOPIC 10: ('memeriksa', 'pemberitahuan', 'perlu', 'persyaratan', 'keluhan', 'bersangkutan', 'masukan', 'mengajukan', 'prosedur')
TOPIC 11: ('zapped', 'eye', 'gate', 'bulletproof', 'five', 'train', 'heaven', 'climbers', 'goonies')
TOPIC 12: ('in', 'and', 'underdevelopment', 'statelessness', 'peacebuilding', 'between', 'with', 'of', 'including')
TOPIC 13: ('penjual', 'masak', 'restoran', 'bakmi', 'penjaja', 'telor', 'cendol', 'bakso', 'gado-gado')
TOPIC 14: ('surabaya', 'malang', 'semarang', 'balikpapan', 'makassar', 'hangtuah', 'salatiga', 'denpasar', 'surakarta')
TOPIC 15: ('apalagi', 'terkesan', 'susah', 'memang', 'tentunya', 'bersemangat', 'tentu', 'nyaman', 'jelek')
TOPIC 16: ('sementara', 'penuh', 'hanya', 'juga', 'dimana', 'nya', 'dipilih', 'menjadi', 'membuat')

TOPIC 17: ('kalo', 'sih', 'biar', 'gak', 'deh', 'udah', 'gimana', 'begini', 'engga')
TOPIC 18: ('harga', 'barang', 'pembelian', 'pembayaran', 'sewa', 'jual', 'transaksi', 'nasabah', 'biaya')
TOPIC 19: ('tunggu', 'informasikan', 'sarankan', 'numpang', 'dicek', 'tsb', 'wiki-en', 'kopdar', 'makanya')
TOPIC 20: ('check', 'setwatchlist', 'default', 'setwatchlistfrompreferences', 'mtext', 'demospace', 'getfriendlypref', 'prefs', 'actiontype')
TOPIC 21: ('kebutuhan', 'dibutuhkan', 'kemudahan', 'kenyamanan', 'mempermudah', 'diharapkan', 'menambah', 'memanfaatkan', 'ketersediaan')
TOPIC 22: ('satellitecoverage', 'programmainlist', 'vp', 'wt', 'techinasia', 'scontent-sin-', 'tw', 'plvijynmnlvnogphlvtixduqbjum', 'esrc')
TOPIC 23: ('salat', 'shalat', 'sholat', 'dhuhur', 'tarawih', 'isya', 'dhuha', 'zuhur', 'berjama')
TOPIC 24: ('unk', 'amott', 'yukihiro', 'dobro', 'youbi', 'siouxsie', 'tambourine', 'cherone', 'bennie')
TOPIC 25: ('terhadap', 'pembegalan', 'lalich', 'perlakukan', 'erwiana', 'orang-orang', 'menganggap', 'menurutnya', 'seseorang')
TOPIC 26: ('mobil', 'motor', 'mesinnya', 'truk', 'chasis', 'berpenggerak', 'sasis', '-silinder', 'mobil-mobil')
TOPIC 27: ('jam', 'pukul', 'sabtu', 'senin', 'rabu', 'jumat', 'senin-jumat', 'pagi', 'wib')
TOPIC 28: ('just', 'like', 'leave', 'something', 'bring', 'take', 'come', 'nothing', 'really')
TOPIC 29: ('bus', 'jalur', 'angkutan', 'komuter', 'kereta', 'rute', 'trem', 'stasiun', 'gyeongbu')
TOPIC 30: ('kawasan', 'terletak', 'sebelah', 'timur', 'jalan', 'barat', 'kota', 'dongdo', 'terbentang')
TOPIC 31: ('kernet', 'petugas', 'penjemput', 'menaiki', 'memarkir', 'pramuniaga', 'taksi', 'mengantar', 'menyewa')
TOPIC 32: ('subsiden', 'membeku', 'penguapan', 'air', 'menguap', 'salinitas', 'meleleh', 'mencair', 'disemprot')
TOPIC 33: ('toilet', 'parkir', 'mushola', 'musholla', 'kantin', 'uks', 'ruang', 'aula', 'hotspot')
TOPIC 34: ('pesawat', 'terbang', 'kargo', 'ulang-alik', 'menerbangkan', 'berawak', 'diterbangkan', 'suborbital', 'helikopter')

Document-Topic Counts:, [237 332 219 189 205 92 181 103 268 85 261 324 367 100 181 284 474 261
 354 237 214 305 159 20 126 274 151 259 151 153 162 426 215 367 107]

C-5. *Word Similarity* pada Skenario Data Tanpa Frasa batasan 3 – *Stemming*

TOPIC 0: ('angkasa', 'berawak', 'gslv', 'propulsi', 'peluncur', 'roket', 'dnepropetrovsk', 'suborbital', 'yuzhnoye')
TOPIC 1: ('jujur', 'sopan', 'menyenangkan', 'bersemangat', 'terbiasa', 'selalu', 'apalagi', 'mementingkan', 'memperhatikan')
TOPIC 2: ('interaktivitas', 'terramodel', 'telekonferensi', 'pemindaian', 'kemudahan', 'bandwidth', 'peer-to-peer', 'mengakses', 'msan')
TOPIC 3: ('should', 'when', 'but', 'non-nil', 'possible', 'make', 'them', 'only', 'this')
TOPIC 4: ('shvetsov', 'mikulin', 'maksimum', 'liquid-cooled', 'ash-', 'beratnya', 'kg', 'dorong', 'hp')
TOPIC 5: ('larang', 'pribawa', 'pajang', 'peguron', 'joyonegoro', 'adikoro', 'keraton', 'kacirebonan', 'singacala')
TOPIC 6: ('bikin', 'bilang', 'ketemu', 'gak', 'maumu', 'siapa-siapa', 'gemes', 'cari', 'lupa')

TOPIC 7: ('berangkat', 'pulang', 'pergi', 'tiba', 'datang', 'hendak', 'setibanya', 'menemui', 'singgah')
TOPIC 8: ('parkiran', 'jemput', 'parkir', 'gudang', 'konter', 'eskalator', 'loker', 'kamar', 'mushala')
TOPIC 9: ('linknya', 'sarankan', 'informasikan', 'wiki-en', 'tanyakan', 'rapihkan', 'sandinya', 'mengecek', 'membetulkan')
TOPIC 10: ('development', 'organization', 'initiative', 'security', 'health', 'education', 'resources', 'board', 'conference')
TOPIC 11: ('three', 'two', 'years', 'four', 'behind', 'five', 'inside', 'after', 'winds')
TOPIC 12: ('internet', 'smartphone', 'ponsel', 'miui', 'voip', 'viber', 'mengakses', 'mobomarket', 'handphone')
TOPIC 13: ('khusus', 'juga', 'tersebut', 'langsung', 'penuh', 'pertama', 'acffest', 'dipilih', 'beberapa')
TOPIC 14: ('dijahit', 'pengganjal', 'menggantung', 'berjajar', 'diletakkan', 'helai-helai', 'hiasan', 'berbentuk', 'kotak-kotak')
TOPIC 15: ('jt', 'jq', 'jn', 'jz', 'jx', 'jf', 'jy', 'mq', 'rq')
TOPIC 16: ('airlines', 'airways', 'transaero', 'airblue', 'qantaslink', 'maskapai', 'atlasjet', 'skyways', 'cargo')
TOPIC 17: ('area', 'hektare', 'luas', 'seluas', 'kawasan', 'mangrove', 'lahan', 'hektar', 'areal')
TOPIC 18: ('surabaya', 'malang', 'semarang', 'balikpapan', 'jakarta', 'makassar', 'hangtuah', 'salatiga', 'yogyakarta')
TOPIC 19: ('kalang', 'santuai', 'santuei', 'parenggean', 'antang', 'seranau', 'tualan', 'cempaga', 'mujam')
TOPIC 20: ('air', 'airnya', 'subsiden', 'basah', 'limpasan', 'penguapan', 'salinitas', 'salinisasi', 'comberan')
TOPIC 21: ('jalur', 'stasiun', 'bus', 'rute', 'jalan', 'cikampek-padalarang', 'angkutan', 'gyeongbu', 'kereta')
TOPIC 22: ('lpdb-kumkm', 'pelaksanaan', 'pengawasan', 'perundang-undangan', 'lpdp', 'penyusunan', 'instansi', 'kerja', 'kumdil')
TOPIC 23: ('min', 'html', 'sec', 'tw', 'lang', 'fullargs', 'gi', 'ok', 'main')
TOPIC 24: ('bicara', 'pembicaraan', 'pengguna', 'utc', 'badio', 'ariefz', 'halo', 'tremonist', 'ardfeb')
TOPIC 25: ('motor', 'sepeda', 'mobil', 'balap', 'sasis', 'honda', 'kategorisepeda', 'pabrikan', 'cbrxx')
TOPIC 26: ('unk', 'vusi', 'bennie', 'gatica', 'tamio', 'cherone', 'amott', 'siouxsie', 'dobro')
TOPIC 27: ('harga', 'barang', 'jual', 'pembelian', 'pembayaran', 'pembeli', 'transaksi', 'ongkos', 'sewa')
TOPIC 28: ('toilet', 'kantin', 'parkir', 'musholla', 'ruang', 'mushola', 'uks', 'hotspot', 'aula')
TOPIC 29: ('udara', 'bandar', 'sastranegara', 'adisutjipto', 'supadio', 'hub', 'bandara', 'juwata', 'adisumarmo')
TOPIC 30: ('ternyata', 'selfy', 'tiba-tiba', 'panik', 'terkejut', 'kaget', 'teringat', 'rupanya', 'kecopetan')
TOPIC 31: ('sih', 'kalo', 'tolong', 'gimana', 'begini', 'perbaiki', 'udah', 'iya', 'deh')
TOPIC 32: ('orang', 'keluarga', 'sementara', 'perempuan', 'anak', 'erwiana', 'dianiaya', 'orang-orang', 'simpatisan')
TOPIC 33: ('jam', 'pukul', 'sabtu', 'pagi', 'senin-jumat', 'senin', 'jumat', 'wib', 'rabu')
TOPIC 34: ('minimal', 'diajukan', 'selambat-lambatnya', 'sahnya', 'pencabutan', 'permohonan', 'jumlah', 'pemungutan', 'kuorum')

Document-Topic Counts:, [40 325 216 282 83 21 359 116 540 377 112 627 103 451 278 21 119 86

228 3 291 119 184 97 5 82 142 340 253 16 204 324 239 230 5]

C-6. *Word Similarity* pada Skenario Data Tanpa Frasa batasan 3 - Tanpa *Stemming*

TOPIC 0: ('should', 'when', 'but', 'possible', 'non-nil', 'this', 'them', 'created', 'started')
TOPIC 1: ('mobomarket', 'gratis', 'bimatri', 'phonemarket', 'mengakses', 'pelanggan', 'layanan', 'aplikasi', 't-money')
TOPIC 2: ('jalur', 'bus', 'angkutan', 'rute', 'komuter', 'kereta', 'trem', 'stasiun', 'gyeongbu')
TOPIC 3: ('informasi', 'pemberitahuan', 'konten', 'pengunggahan', 'pupns', 'formulir', 'backlink', 'penggunanya', 'konten-konten')
TOPIC 4: ('minuman', 'masak', 'makanan', 'lezat', 'kudapan', 'sosis', 'resep', 'dimasak', 'kue-kue')
TOPIC 5: ('perlu', 'dipertimbangkan', 'sebaiknya', 'seharusnya', 'flooder', 'penghapusannya', 'layak', 'diamandemen', 'usulannya')
TOPIC 6: ('bersih', 'airnya', 'disemprot', 'meleleh', 'basah', 'subsiden', 'mencair', 'menguap', 'mengapung')
TOPIC 7: ('kursi', 'gerindra', 'parlemen', 'pdi-p', 'suara', 'hanura', 'demokrat', 'pemilu', 'fraksi')
TOPIC 8: ('petugas', 'erwiana', 'pasien-pasiennya', 'polisi', 'melapor', 'aparat', 'narapidana', 'korban', 'disandera')
TOPIC 9: ('penerbangan', 'maskapai', 'kargo', 'airlines', 'berjadwal', 'airways', 'airblue', 'atlasjet', 'menerbangi')
TOPIC 10: ('toilet', 'parkir', 'musholla', 'mushola', 'kantin', 'uks', 'ruang', 'aula', 'hotspot')
TOPIC 11: ('menyenangkan', 'bersemangat', 'nyaman', 'terbiasa', 'apalagi', 'terkesan', 'tentunya', 'selalu', 'sopan')
TOPIC 12: ('surabaya', 'malang', 'semarang', 'balikpapan', 'denpasar', 'salatiga', 'surakarta', 'hangtuah', 'yogyakarta')
TOPIC 13: ('nimah', 'ketarik', 'kepentok', 'keciduk', 'supir', 'cewek', 'jodoh', 'beningnya', 'lovepedia')
TOPIC 14: ('pelayanan', 'pengawasan', 'publik', 'lembaga', 'lpdb-kumkm', 'hhi', 'lpdp', 'instansi', 'badan-badan')
TOPIC 15: ('pengemudi', 'bagasi', 'diparkir', 'mobil', 'truk', 'memarkirkan', 'roda', 'pengendara', 'terparkir')
TOPIC 16: ('sih', 'kalo', 'gimana', 'tolong', 'begini', 'biar', 'udah', 'sewot', 'iya')
TOPIC 17: ('barang', 'pembelian', 'tiket', 'harga', 'tarif', 'pembayaran', 'membayar', 'ongkos', 'sewa')
TOPIC 18: ('pertama', 'kedua', 'terakhir', 'langsung', 'setiap', 'dimulai', 'berikutnya', 'sebelum', 'sehari')
TOPIC 19: ('min', 'jung-ho', 'suh', 'jq', 'tae-kyun', 'jj', 'do-hoon', 'sung-mo', 'younha')
TOPIC 20: ('librarians', 'underdevelopment', 'with', 'statelessness', 'lessons', 'arguably', 'decreased', 'peacebuilding', 'btr-pb')
TOPIC 21: ('jam', 'pagi', 'pukul', 'sabtu', 'senin', 'sore', 'senin-jumat', 'wib', 'siang')
TOPIC 22: ('menemui', 'hendak', 'pulang', 'menunggu', 'pergi', 'datang', 'tiba-tiba', 'mammu', 'rasminah')
TOPIC 23: ('just', 'good', 'like', 'take', 'everytime', 'away', 'shout', 'turn', 'looks')
TOPIC 24: ('memerlukan', 'kurangnya', 'mengatasi', 'mengurangi', 'membutuhkan', 'risiko', 'dibutuhkan', 'mengganggu', 'p-book')
Document-Topic Counts:, [279 297 154 89 50 202 413 72 316 296 337 188 175 198 171 405 581 219
424 181 654 175 369 195 285]

C-7. *Word Similarity* pada Skenario Data Frasa batasan 1 – *Stemming*

TOPIC 0: ('harga', 'pembayaran', 'pembelian', 'cicilan', 'jual', 'ongkos', 'biaya', 'rekening', 'membayar')
TOPIC 1: ('-kecepatan', 'vanos', 'matic', 'four-wheel', 'gearbox', 'continuously', 'charger', 'timing', 'vnt')
TOPIC 2: ('sementara', 'dimana', 'memasuki', 'datang', 'kembali', 'ke', 'kemudian', 'membawa', 'meninggalkan')
TOPIC 3: ('unk', 'yukihiro', 'amott', 'youbi', 'dobro', 'siouxsie', 'tambourine', 'cherone', 'tamio')
TOPIC 4: ('them', 'that', 'are', 'but', 'have', 'which', 'would', 'when', 'should')
TOPIC 5: ('priyono', 'tranggono', 'agus', 'setyowati', 'kurniadi', 'riawan', 'purnomo', 'herwanto', 'handoyo')
TOPIC 6: ('shalat', 'salat', 'sholat', 'dhuha', 'berjama', 'dzikir', 'dhuhur', 'tarawih', 'tahlil')
TOPIC 7: ('clbk', 'unyu', 'ngojek', 'ketarik', 'membawamu', 'cowok', 'ketemu', 'adikku', 'kupeluk')
TOPIC 8: ('airnya', 'basah', 'subsiden', 'comberan', 'bersih', 'salinisasi', 'disemprot', 'penguapan', 'menguap')
TOPIC 9: ('singapura', 'boarding', 'dubai', 'bangkok', 'aerowisata', 'asiaworld', 'konter', 'suntec', 'catering')
TOPIC 10: ('lpdb-kumkm', 'pengawasan', 'lpdp', 'ulp', 'perizinan', 'bppspam', 'pelaksanaan', 'instansi', 'pengelolaan')
TOPIC 11: ('samsung', 'nokia', 'htc', 'kitkat', 'asus', 'cyber-shot', 'iphone', 'symbian', 'blackberry')
TOPIC 12: ('bus', 'angkutan', 'jalur', 'komuter', 'rute', 'trem', 'busway', 'stasiun', 'cikampek-padalarang')
TOPIC 13: ('gimana', 'sih', 'kalo', 'begini', 'udah', 'tolong', 'perbaiki', 'sewot', 'nyambung')
TOPIC 14: ('smartphone', 'berkamera', 'rtsp', 'konektifitas', 'mobomarket', 'olahpesan', 'laptop', 'handphone', 'ponsel')
TOPIC 15: ('surabaya', 'malang', 'semarang', 'balikpapan', 'salatiga', 'hangtuah', 'makassar', 'surakarta', 'yogyakarta')
TOPIC 16: ('parkir', 'toilet', 'musholla', 'mushola', 'kantin', 'uks', 'hotspot', 'ruang', 'aula')
TOPIC 17: ('with', 'through', 'to', 'their', 'after', 'into', 'on', 'between', 'years')
TOPIC 18: ('jq', 'min', 'je', 'vo', 'aj', 'jung-ho', 'dong-seok', 'na', 'ko')
TOPIC 19: ('motivasi', 'memahami', 'memperhatikan', 'sikap', 'hal-hal', 'perilaku', 'menurutnya', 'pentingnya', 'persuasif')
TOPIC 20: ('actiontype', 'default', 'xfdtarget', 'cfds', 'pcmodify', 'message-id', 'demospace', 'optout', 'tagtype')
TOPIC 21: ('zapped', 'eye', 'goonies', 'bulletproof', 'gate', 'heaven', 'naked', 'shakedown', 'posterjpeg')
TOPIC 22: ('jam', 'pukul', 'sabtu', 'senin', 'rabu', 'jumat', 'wib', 'senin-jumat', 'minggu')
TOPIC 23: ('berjajar', 'jendela', 'diletakan', 'diletakkan', 'dilantai', 'menyangga', 'sisinya', 'berlubang', 'lampu')
TOPIC 24: ('kawasan', 'pesisir', 'sekitarnya', 'barat', 'pieh', 'dongdo', 'pantai', 'wilayah', 'terluas')
TOPIC 25: ('accelerometers', 'memudahkan', 'memerlukan', 'pengunaan', 'menghemat', 'algortima', 'terkomputerisasi', 'memproses', 'statis')
TOPIC 26: ('cargolux', 'airblue', 'transaero', 'penerbangan', 'atlasjet', 'airways', 'airlines', 'cargo', 'qantaslink')

TOPIC 27: ('just', 'like', 'leave', 'come', 'bring', 'really', 'again', 'something', 'going')
TOPIC 28: ('minuman', 'kudapan', 'kue-kue', 'lezat', 'samgyeopsal', 'masak', 'makanan', 'disantap', 'sosis')
TOPIC 29: ('melupakan', 'utty', 'merasa', 'rupanya', 'malah', 'lilu', 'yumeta', 'rudabeh', 'apalagi')
TOPIC 30: ('komandan', 'danskadron', 'kodikau', 'khussus', 'danwing', 'komando', 'kopatdara', 'darat', 'tni')
TOPIC 31: ('layan', 'ruyung', 'kameloh', 'sanggan', 'tumbu', 'kelekar', 'hurun', 'perapat', 'rurah')
TOPIC 32: ('bagi-bagi', 'sore-sore', 'angkring', 'kos-kosan', 'tongsis', 'buka', 'jemput', 'jaga', 'nonton')
TOPIC 33: ('berwarna', 'krem', 'hitam', 'kehijauan', 'kemerahan', 'pucat', 'kecoklatan', 'kebiruan', 'bergaris-garis')
TOPIC 34: ('development', 'initiative', 'board', 'resource', 'organization', 'health', 'ttip', 'security', 'resources')

Document-Topic Counts:, [306 229 912 907 215 91 50 319 272 168 306 25 193 743 134 185 308 364
 200 311 144 302 247 334 114 239 191 146 112 540 19 197 212 62 202]

C-8. *Word Similarity* pada Skenario Data Frasa batasan 1 - Tanpa *Stemming*

TOPIC 0: ('sops', 'dangrup', 'pamen', 'asops', 'opslat', 'pabandya', 'wadan', 'kasdivif', 'sespri')
TOPIC 1: ('perlu', 'kriteria-kriteria', 'tidaknya', 'masukan', 'disesuaikan', 'seharusnya', 'diperhatikan', 'bersangkutan', 'flooder')
TOPIC 2: ('unk', 'yukihiro', 'amott', 'youbi', 'dobro', 'siouxsie', 'tambourine', 'cherone', 'tamio')
TOPIC 3: ('them', 'that', 'are', 'but', 'have', 'which', 'some', 'would', 'when')
TOPIC 4: ('ketemu', 'bikin', 'bilang', 'gak', 'unyu', 'cowok', 'itukan', 'siapa-siapa', 'banget')
TOPIC 5: ('jam', 'pukul', 'sabtu', 'senin-jumat', 'pagi', 'senin', 'jumat', 'sore', 'rabu')
TOPIC 6: ('diperbuatnya', 'firman-ku', 'hukum-nya', 'dursila', 'berkat-nya', 'perkataannya', 'diberikan-nya', 'kefasikan', 'hukuman-nya')
TOPIC 7: ('chasis', 'mesinnya', 'pistonnya', 'mesin', 'dieselnya', 'rodanya', 'bersilinder', 'roda', '-silinder')
TOPIC 8: ('zapped', 'eye', 'bulletproof', 'goonies', 'climbers', 'train', 'five', 'one-man', 'shakedown')
TOPIC 9: ('pesawat', 'kargo', 'helikopter', 'terbang', 'yc-', 'diterbangkan', 'menerbangkan', 'ulang-alik', 'uav')
TOPIC 10: ('parkir', 'toilet', 'mushola', 'musholla', 'kantin', 'uks', 'hotspot', 'aula', 'ber-ac')
TOPIC 11: ('membohonginya', 'kecopetan', 'malah', 'terkejut', 'menolongnya', 'yumeta', 'menamparnya', 'utty', 'memaki')
TOPIC 12: ('nur', 'rivan', 'jejen', 'suganda', 'yanuar', 'zulfa', 'hasnah', 'rialdy', 'didu')
TOPIC 13: ('pertama', 'dimana', 'kedua', 'kalinya', 'sementara', 'berikutnya', 'ketiga', 'tersebut', 'sebelum')
TOPIC 14: ('minuman', 'kudapan', 'makanan', 'kue-kue', 'samgyeopsal', 'lezat', 'dimasak', 'gorengan', 'rebusan')
TOPIC 15: ('tentunya', 'nyaman', 'baik', 'menyenangkan', 'sangatlah', 'ramah', 'memperhatikan', 'cukup', 'terjaga')
TOPIC 16: ('just', 'like', 'leave', 'come', 'bring', 'something', 'too', 'really', 'again')

TOPIC 17: ('bus', 'angkutan', 'komuter', 'keberangkatan', 'jalur', 'trem', 'rute', 'kereta', 'gyeongbu')
TOPIC 18: ('scribunto', 'input', 'disabled', 'actiontype', 'message-id', 'multiple', 'uservalue', 'errorcode', 'formatter')
TOPIC 19: ('surabaya', 'malang', 'semarang', 'balikpapan', 'salatiga', 'hangtuah', 'surakarta', 'palembang', 'yogyakarta')
TOPIC 20: ('smartphone', 'ponsel', 'konektifitas', 'miui', 'handphone', 'multisentuh', 'mobomarket', 'berkamera', 'kitkat')
TOPIC 21: ('statelessness', 'underdevelopment', 'peacebuilding', 'including', 'working', 'advocating', 'trans-pacific', 'lessons', 'macroeconomic')
TOPIC 22: ('tindakan', 'tuntutan', 'menanggapi', 'sanksi', 'daesh', 'mempertimbangkan', 'membenarkan', 'penolakan', 'pembatasan')
TOPIC 23: ('penerbangan', 'maskapai', 'airlines', 'airways', 'berjadwal', 'airblue', 'cargolux', 'atlasjet', 'sewaan')
TOPIC 24: ('lokasinya', 'clungup', 'danau', 'ungapan', 'pinggir', 'pantai', 'bukit', 'air', 'dongdo')
TOPIC 25: ('jf', 'jq', 'rq', 'jt', 'jz', 'qk', 'wx', 'qy', 'mq')
TOPIC 26: ('min', 'na', 'jung-ho', 'dong-seok', 'suh', 'lim', 'jin-goo', 'in-kwon', 'seung-min')
TOPIC 27: ('tolong', 'gimana', 'perbaiki', 'sih', 'kalo', 'begini', 'bersediakah', 'iya', 'udah')
TOPIC 28: ('kondisi', 'terpapar', 'menyebabkan', 'berbahaya', 'amenorea', 'berlebihan', 'terganggunya', 'kelangkaan', 'pengkonsumsian')
TOPIC 29: ('menyediakan', 'spipise', 'konsultasi', 'layanan', 'mengelola', 'penyedia', 'penyediaan', 'memfasilitasi', 'kemudahan')
TOPIC 30: ('shalat', 'salat', 'sholat', 'tarawih', 'berjama', 'dhuha', 'dhuhur', 'dzikir', 'tahlil')
TOPIC 31: ('dijahit', 'hiasan', 'kotak-kotak', 'selutut', 'berbentuk', 'roknya', 'coretan-coretan', 'dihiasi', 'menggantung')
TOPIC 32: ('pengawasan', 'pelaksanaan', 'bppspam', 'kewenangan', 'kumdil', 'lpdb-kumkm', 'bnptki', 'kepegawaian', 'keimigrasian')
TOPIC 33: ('harga', 'pembelian', 'tiket', 'biaya', 'sewa', 'barang', 'ongkos', 'membayar', 'pembayaran')
TOPIC 34: ('menemui', 'mammu', 'ogdenville', 'sedang', 'hendak', 'membawa', 'mo-yeon', 'mengikutinya', 'erwiana')

Document-Topic Counts:, [12 372 971 255 251 233 118 245 398 121 301 236 69 599 119 398 199 230
 230 223 139 427 233 211 245 45 212 595 260 337 50 229 147 325 810]

C-9. *Word Similarity* pada Skenario Data Frasa batasan 2 – *Stemming*

TOPIC 0: ('konter', 'jemput', 'lounge', 'boarding', 'loker', 'eskalator', 'catering', 'mezzanine', 'antar-jemput')
TOPIC 1: ('linknya', 'sarankan', 'mengecek', 'informasikan', 'wiki-en', 'membetulkan', 'tunggu', 'penghapusannya', 'sandinya')
TOPIC 2: ('in', 'and', 'with', 'underdevelopment', 'between', 'of', 'their', 'peacebuilding', 'the')
TOPIC 3: ('unk', 'yukihiro', 'amott', 'youbi', 'dobro', 'siouxsie', 'tambourine', 'cherone', 'bennie')
TOPIC 4: ('tentunya', 'terkesan', 'nyaman', 'apalagi', 'memang', 'tentu', 'susah', 'cukup', 'bersemangat')
TOPIC 5: ('antri', 'ketarik', 'clbk', 'kepentok', 'kepepet', 'ngojek', 'pembantuku', 'cewek', 'kesandung')
TOPIC 6: ('dibutuhkan', 'menambah', 'banyaknya', 'memerlukan', 'bersih', 'kondisi', 'khalayaknya', 'memanfaatkan', 'diharapkan')

TOPIC 7: ('mesin', '-silinder', 'ditenagai', 'mesinnya', 'yuneec', 'chasis', 'bertenaga', 'kendaraan', 'piston')
TOPIC 8: ('berangkat', 'pulang', 'pergi', 'tiba', 'datang', 'hendak', 'singgah', 'menemui', 'sesampainya')
TOPIC 9: ('lpdb-kumkm', 'pengawasan', 'lpdp', 'bppspam', 'pelaksanaan', 'instansi', 'ulp', 'frpba', 'pengelolaan')
TOPIC 10: ('maskapai', 'airlines', 'airways', 'airblue', 'berjadwal', 'aircalin', 'atlasjet', 'penerbangan', 'skymark')
TOPIC 11: ('default', 'getfriendlypref', 'setwatchlist', 'radioorcheckbox', 'templatename', 'getlink', 'getcontentlanguage', 'engscale', 'getutcmonthname')
TOPIC 12: ('air', 'airnya', 'tawar', 'basah', 'bebatuan', 'salinitas', 'subsiden', 'salinisasi', 'penguapan')
TOPIC 13: ('kalo', 'sih', 'gimana', 'begini', 'udah', 'biar', 'deh', 'tolong', 'nyambung')
TOPIC 14: ('minuman', 'makanan', 'makan', 'masak', 'kudapan', 'kue-kue', 'lezat', 'sarapan', 'sosis')
TOPIC 15: ('harga', 'pembelian', 'jual', 'barang', 'pembayaran', 'biaya', 'ongkos', 'sewa', 'tarif')
TOPIC 16: ('are', 'them', 'that', 'which', 'but', 'some', 'this', 'more', 'when')
TOPIC 17: ('zapped', 'eye', 'heaven', 'perfect', 'posterjpeg', 'five', 'goonies', 'woman', 'naked')
TOPIC 18: ('just', 'like', 'when', 'makes', 'still', 'leave', 'too', 'need', 'better')
TOPIC 19: ('wff', 'redesign', 'capabilities', 'access', 'control', 'boards', 'mitigation', 'offers', 'reporting')
TOPIC 20: ('pintu', 'atap', 'lantai', 'menara', 'lantainya', 'beratap', 'selasar', 'teras', 'pelataran')
TOPIC 21: ('sholat', 'salat', 'shalat', 'ibadah', 'dhuhur', 'dhuha', 'dzikir', 'tarawih', 'puasa')
TOPIC 22: ('layan', 'kategoribukit', 'perapat', 'sukabangun', 'jelmu', 'rurah', 'kategorisimpang', 'lapai', 'sedulang')
TOPIC 23: ('jt', 'jq', 'jn', 'jz', 'jf', 'jx', 'mq', 'qk', 'wx')
TOPIC 24: ('ketakutan', 'sadar', 'buruk', 'justru', 'khawatir', 'lilu', 'benar-benar', 'mammu', 'deevs')
TOPIC 25: ('dijahit', 'helai-helai', 'kotak-kotak', 'selutut', 'digunting', 'celemek', 'hiasan', 'berbentuk', 'pengganjal')
TOPIC 26: ('satu-satunya', 'shettar', 'fadnavis', 'geetu', 'chiranjeevi', 'sementara', 'sembilan', 'palaszczuk', 'savithri')
TOPIC 27: ('parkir', 'toilet', 'musholla', 'mushola', 'kantin', 'uks', 'hotspot', 'ruang', 'loker')
TOPIC 28: ('surabaya', 'malang', 'semarang', 'balikpapan', 'makassar', 'hangtuah', 'jakarta', 'salatiga', 'surakarta')
TOPIC 29: ('bus', 'angkutan', 'terminal', 'rute', 'jalur', 'komuter', 'busway', 'trem', 'depo')
TOPIC 30: ('smartphone', 'ponsel', 'handphone', 'konektifitas', 'miui', 'laptop', 'mobomarket', 'berkamera', 'multisentuh')
TOPIC 31: ('masuk', 'sebelum', 'pertama', 'berikutnya', 'kali', 'kedua', 'ke', 'langsung', 'kalinya')
TOPIC 32: ('bicara', 'pembicaraan', 'pengguna', 'utc', 'badio', 'ariefz', 'halo', 'tremonist', 'ardfeb')
TOPIC 33: ('jam', 'pukul', 'sabtu', 'senin', 'pagi', 'wib', 'senin-jumat', 'jumat', 'rabu')
TOPIC 34: ('teringat', 'menangis', 'rupanya', 'pura-pura', 'ameera', 'bahagianya', 'sangmaima', 'encun', 'ternyata')

Document-Topic Counts:, [133 404 201 451 291 145 369 159 136 337 90 141 156 490 73 307 207 286

132 270 112 32 144 30 327 218 409 272 176 176 104 289 5 177 371]

C-10. *Word Similarity* pada Skenario Data Frasa batasan 2 - Tanpa *Stemming*

TOPIC 1: ('kering', 'basah', 'airnya', 'blenyik', 'segar', 'berair', 'air', 'merendam', 'diminum')
TOPIC 2: ('help', 'please', 'changes', 'affect', 'tech-newsletter-software-news', 'these', 'special', 'users', 'recent')
TOPIC 3: ('parkir', 'toilet', 'mushola', 'musholla', 'kantin', 'uks', 'hotspot', 'aula', 'ber-ac')
TOPIC 4: ('web', 'blog', 'facebook', 'bilna', 'twitter', 'resmi', 'situs', 'website', 'kidalang')
TOPIC 5: ('sulit', 'cukup', 'membutuhkan', 'mengganggu', 'tentunya', 'memerlukan', 'p-book', 'cocok', 'sifatnya')
TOPIC 6: ('berjajar', 'lengkungan', 'bersambungan', 'pintu', 'ventilasi', 'diatas', 'menjorok', 'sempit', 'dinding')
TOPIC 7: ('memeriksa', 'keluhan', 'memblokir', 'komplain', 'memverifikasi', 'pengaju', 'klarifikasi', 'akunnya', 'pembegalan')
TOPIC 8: ('min', 'jq', 'aj', 'wt', 'tq', 'wg', 'wl', 'jn', 'jy')
TOPIC 9: ('good', 'just', 'glee-ver', 'like', 'take', 'again', 'away', 'everytime', 'come')
TOPIC 10: ('underdevelopment', 'peacebuilding', 'of', 'statelessness', 'in', 'librarians', 'and', 'investigators', 'lessons')
TOPIC 11: ('unk', 'amott', 'yukihiro', 'dobro', 'youbi', 'siouxsie', 'cherone', 'tambourine', 'bennie')
TOPIC 12: ('sepertinya', 'bersediakah', 'layak', 'dipertimbangkan', 'memang', 'makanya', 'tentu', 'memvandal', 'tsb')
TOPIC 13: ('but', 'that', 'them', 'when', 'would', 'have', 'are', 'there', 'is')
TOPIC 14: ('bus', 'keberangkatan', 'komuter', 'angkutan', 'penumpang', 'taksi', 'trem', 'gerbong', 'kereta')
TOPIC 15: ('bikin', 'gak', 'bilang', 'nggak', 'tuh', 'banget', 'biar', 'enggak', 'emang')
TOPIC 16: ('sementara', 'tetap', 'dimana', 'hanya', 'membuat', 'membuatnya', 'penuh', 'mendapat', 'menjadi')
TOPIC 17: ('layanan', 'penyedia', 'pelanggan', 'menyediakan', 'mengakses', 't-money', 'gratis', 'phonemarket', 'bimatri')
TOPIC 18: ('jam', 'pukul', 'pagi', 'sabtu', 'senin', 'wib', 'jumat', 'senin-jumat', 'sore')
TOPIC 19: ('baju', 'dijahit', 'sepatu', 'rajutan', 'kalkir', 'kotak-kotak', 'celemek', 'topi', 'celana')
TOPIC 20: ('pesawat', '-penumpang', 'dua-seater', 'kendaraan', 'bertenaga', 'uav', 'vtvl', 'kategorimesin', 'ditenagai')
TOPIC 21: ('terkejut', 'tiba-tiba', 'mammu', 'rupanya', 'meninggalkannya', 'menemui', 'menyangka', 'encup', 'ternyata')
TOPIC 22: ('pelayanan', 'pengawasan', 'instansi', 'bppspam', 'lpdb-kumkm', 'konsultasi', 'pengadaan', 'pelaksanaan', 'perizinan')
TOPIC 23: ('datavalues', 'multiple', 'redesign', 'scribunto', 'access', 'generate', 'sitelinks', 'fixed', 'bugfixes')
TOPIC 24: ('surabaya', 'malang', 'semarang', 'balikpapan', 'makassar', 'hangtuah', 'salatiga', 'surakarta', 'jakarta')
TOPIC 25: ('kebutuhan', 'meningkatkan', 'memfasilitasi', 'efisien', 'efektivitas', 'prioritas', 'kualitas', 'kesiapan', 'diharapkan')
TOPIC 26: ('penerbangan', 'maskapai', 'airlines', 'berjadwal', 'airways', 'sewaan', 'airblue', 'atlasjet', 'cargolux')
TOPIC 27: ('singapura', 'ratmalana', 'sematan', 'dhiu', 'bandar-bandar', 'muskat', 'boarding', 'kolombo', 'gbia')

TOPIC 28: ('award', 'awards', 'best', 'bafta', 'iifa', 'terbaik', 'academy', 'newcomer', 'nominasi')
TOPIC 29: ('diantar', 'duduk-duduk', 'mengantar', 'hendak', 'beristirahat', 'kernet', 'dompetnya', 'memarkir', 'menunggu')
TOPIC 30: ('smartphone', 'ponsel', 'handphone', 'ipad', 'kitkat', 'iphone', 'miui', 'laptop', 'konektifitas')
TOPIC 31: ('jadwal', 'minggu', 'sehari', 'bulan', 'dimulai', 'pengumuman', 'ditunda', 'diumumkan', 'kali')
TOPIC 32: ('petugas', 'menyamar', 'pria', 'orang', 'pelayan', 'mengaku', 'perawat', 'diculik', 'wanitanya')
TOPIC 33: ('tolong', 'gimana', 'perbaiki', 'sih', 'begini', 'kalo', 'iya', 'bersediakah', 'tlg')
TOPIC 34: ('harga', 'pembelian', 'barang', 'tiket', 'ongkos', 'tarif', 'pembayaran', 'membayar', 'biaya')

Document-Topic Counts:, [79 159 26 282 39 372 132 162 111 217 462 431 263 201 235 224 507 279
 158 152 155 301 144 236 182 163 183 214 10 389 41 178 306 342 207]

C-11. *Word Similarity* pada Skenario Data Frasa batasan 3 – *Stemming*

TOPIC 0: ('minimal', 'diajukan', 'selambat-lambatnya', 'sahnya', 'pencabutan', 'permohonan', 'jumlah', 'pemungutan', 'kuorum')
TOPIC 1: ('min', 'na', 'ye-seul', 'myung-hoon', 'mi-yeon', 'chae-ah', 'ji-seok', 'seo-yeon', 'je-moon')
TOPIC 2: ('but', 'that', 'them', 'when', 'are', 'to', 'would', 'have', 'become')
TOPIC 3: ('mempermudah', 'kuesioner', 'memudahkan', 'data-data', 'masukan', 'spipise', 'informasi', 'khalayaknya', 'panduan')
TOPIC 4: ('biar', 'gak', 'kalo', 'sih', 'tuh', 'deh', 'nyari', 'udah', 'nyambung')
TOPIC 5: ('memilih', 'langsung', 'sementara', 'penuh', 'akan', 'hanya', 'mendapatkan', 'masuk', 'berikutnya')
TOPIC 6: ('clbk', 'ngojek', 'unyu', 'antri', 'membawamu', 'ketarik', 'adikku', 'ketemu', 'kupeluk')
TOPIC 7: ('mobil', '-silinder', 'mesinnya', 'motor', 'berpenggerak', 'az-fe', 'dieselnya', 'chasis', 'uz-fe')
TOPIC 8: ('smartphone', 'handphone', 'ponsel', 'mobomarket', 'laptop', 'picmix', 'konektifitas', 'appstore', 'viber')
TOPIC 9: ('unk', 'yukihiro', 'amott', 'youbi', 'dobro', 'siouxsie', 'tambourine', 'cherone', 'bennie')
TOPIC 10: ('lpdb-kumkm', 'pengawasan', 'kumdil', 'kerja', 'pelaksanaan', 'lpdp', 'instansi', 'bppspam', 'bpws')
TOPIC 11: ('tolong', 'perbaiki', 'harap', 'bersediakah', 'gimana', 'sih', 'sepertinya', 'birukan', 'terimakasih')
TOPIC 12: ('harga', 'barang', 'jual', 'konsumen', 'impor', 'konsumennya', 'modal', 'beli', 'murah')
TOPIC 13: ('maskapai', 'airlines', 'airways', 'berjadwal', 'penerbangan', 'sewaan', 'cargolux', 'atlasjet', 'bertarif')
TOPIC 14: ('surabaya', 'malang', 'semarang', 'balikpapan', 'makassar', 'hangtuah', 'jakarta', 'denpasar', 'salatiga')
TOPIC 15: ('kirim', 'novirion', 'iwan', 'udah', 'koq', 'bonaditya', 'blokir', 'prasetyono', 'keliatan')
TOPIC 16: ('inap', 'rawat', 'poliklinik', 'klinik', 'perawatan', 'gawat', 'bersalin', 'ugd', 'sakit')
TOPIC 17: ('info', 'pagetype', 'message-id', 'imageusage', 'backlinks', 'isni', 'formatlink', 'dvd-notes', 'makewikitexterror')

TOPIC 18: ('jam', 'pukul', 'sabtu', 'pagi', 'senin', 'wib', 'jumat', 'senin-jumat', 'rabu')
TOPIC 19: ('ubah', 'reason', 'tokobanten', 'kontribusi', 'bintangpesbuks', 'bpwildan', 'homeremediesforacne', 'rajapremi', 'hmmunpam')
TOPIC 20: ('berat', 'mikulin', 'kg', 'beratnya', 'landas', 'shvetsov', 'kotor', 'ash-', 'liquid-cooled')
TOPIC 21: ('kasi', 'pabandya', 'wadan', 'waasintel', 'paban', 'danrem', 'dik', 'danbrigif', 'denma')
TOPIC 22: ('facilities', 'statelessness', 'affiliated', 'building', 'trade', 'government', 'wff', 'areas', 'policing')
TOPIC 23: ('hendak', 'pulang', 'datang', 'berangkat', 'menemui', 'diantar', 'tiba', 'pergi', 'mendatangi')
TOPIC 24: ('airnya', 'subsiden', 'vegetasi', 'bebatuan', 'basah', 'air', 'bersih', 'limpasan', 'berair')
TOPIC 25: ('fasilitas', 'lantai', 'ruangan', 'parkir', 'berlantai', 'mushola', 'parkiran', 'gedung', 'aula')
TOPIC 26: ('jalan', 'pintu', 'persimpangan', 'gerbang', 'pinggir', 'pertigaan', 'disebelah', 'menuju', 'perempatan')
TOPIC 27: ('undang', 'gnt', 'undangan', 'tertuang', 'uu', 'pokok-pokok', 'permendiknas', 'peraturan', 'perundang')
TOPIC 28: ('apalagi', 'selalu', 'jujur', 'senang', 'susah', 'tentu', 'punya', 'mengerti', 'bersemangat')
TOPIC 29: ('zapped', 'time', 'hundred-year-old', 'goonies', 'females', 'posterjpeg', 'half', 'five', 'inside')
TOPIC 30: ('parkir', 'toilet', 'kantin', 'musholla', 'mushola', 'uks', 'hotspot', 'ruang', 'loker')
TOPIC 31: ('pengganjal', 'pikulan', 'tikar', 'baskom', 'teng-teng', 'taplak', 'sprei', 'tas', 'seprai')
TOPIC 32: ('angkasa', 'suborbital', 'irs-p', 'gsat-', 'shuttle', 'soyuz', 'berawak', 'spaceflight', 'iss')
TOPIC 33: ('bus', 'angkutan', 'komuter', 'taksi', 'trem', 'kereta', 'jalur', 'gyeongbu', 'antarkota')
TOPIC 34: ('dibayarkan', 'tiket', 'cicilan', 'rugi', 'rp', 'membayar', 'tunai', 'tarif', 'rekening')

Document-Topic Counts:, [5 228 238 279 397 630 215 122 100 376 198 199 189 51 173 4 30 43
155 8 48 3 343 347 277 80 127 4 533 401 205 406 47 159 160]

C-12. *Word Similarity* pada Skenario Data Frasa batasan 3 - Tanpa *Stemming*

TOPIC 0: ('petugas', 'polisi', 'melapor', 'aparat', 'narapidana', 'orpo', 'penggeledahan', 'penahanan', 'eksekusi')
TOPIC 1: ('unk', 'amott', 'yukihiro', 'dobro', 'youbi', 'siouxsie', 'tambourine', 'cherone', 'bennie')
TOPIC 2: ('min', 'programmainlist', 'html', 'je-dong', 'imgmovie', 'hyo-ri', 'shinvi', 'reply', 'cineseoul')
TOPIC 3: ('bus', 'angkutan', 'keberangkatan', 'komuter', 'jalur', 'trem', 'melayani', 'rute', 'kereta')
TOPIC 4: ('keciduk', 'ketarik', 'nimah', 'kepentok', 'jodoh', 'cewek', 'lovepedia', 'kepincut', 'selfie')
TOPIC 5: ('tolong', 'perbaiki', 'bersediakah', 'harap', 'maaf', 'birukan', 'semoga', 'terimakasih', 'sepertinya')
TOPIC 6: ('penerbangan', 'maskapai', 'airlines', 'berjadwal', 'airways', 'sewaan', 'airblue', 'atlasjet', 'cargolux')
TOPIC 7: ('just', 'come', 'like', 'leave', 'good', 'really', 'again', 'believe', 'wish')

TOPIC 8: ('multiple', 'datavalues', 'output', 'generate', 'redesign', 'scribunto', 'improvements', 'fixed', 'deployed')
TOPIC 9: ('pura', 'bali', 'sriwijaya', 'mandala', 'denpasar', 'akti', 'besakih', 'pamecutan', 'kencana')
TOPIC 10: ('biar', 'kalo', 'sih', 'gak', 'deh', 'begini', 'udah', 'gimana', 'engga')
TOPIC 11: ('sementara', 'dimana', 'masuk', 'satu-satunya', 'orang', 'dua', 'namun', 'sembilan', 'kedua')
TOPIC 12: ('subsiden', 'kondisi', 'membeku', 'terendam', 'hujan', 'tergenang', 'airnya', 'mencair', 'air')
TOPIC 13: ('harga', 'barang', 'biaya', 'ongkos', 'jual', 'pembelian', 'sewa', 'tiket', 'membayar')
TOPIC 14: ('kargo', 'terbang', 'penerbangan', 'pesawat', 'beregistrasi', 'menerbangi', 'pesawatnya', 'didarati', 'menerbangkan')
TOPIC 15: ('menemui', 'hendak', 'pergi', 'pulang', 'menunggu', 'tiba-tiba', 'datang', 'mammu', 'rasminah')
TOPIC 16: ('years', 'three', 'bodies', 'arguably', 'two', 'with', 'four', 'in', 'the')
TOPIC 17: ('kebutuhan', 'pelayanan', 'lpdb-kumkm', 'kelancaran', 'kesiapan', 'prioritas', 'efektivitas', 'penyediaan', 'frpba')
TOPIC 18: ('ukuran', 'ketebalan', 'bervariasi', 'permukaannya', 'tipis', 'silindris', 'halus', 'tebalnya', 'bongkah')
TOPIC 19: ('jendela', 'ruangan', 'loteng', 'lantai', 'didepan', 'memasang', 'diletakan', 'dilantai', 'atap')
TOPIC 20: ('memeriksa', 'semiperlindungan', 'pemberitahuan', 'pengaju', 'pengajuan', 'bersangkutan', 'prosedur', 'amandemen', 'klarifikasi')
TOPIC 21: ('menyenangkan', 'bersemangat', 'nyaman', 'ramah', 'tentunya', 'baik', 'rajin', 'memperhatikan', 'berfikir')
TOPIC 22: ('merugikan', 'pembegalan', 'masalah', 'keluhan', 'tindakan', 'hal-hal', 'mengganggu', 'wajar', 'disengaja')
TOPIC 23: ('penuh', 'hanya', 'membuat', 'nya', 'shamedbyyou', 'langsung', 'memasukkan', 'setelahnya', 'juga')
TOPIC 24: ('makanya', 'buat', 'dibikin', 'sepertinya', 'bersediakah', 'tsb', 'cuma', 'soalnya', 'disini')
TOPIC 25: ('angkasa', 'berawak', 'dnepropetrovsk', 'observasi', 'antariksa', 'probe', 'soyuz', 'suborbital', 'kosmos')
TOPIC 26: ('parkir', 'toilet', 'musholla', 'mushola', 'kantin', 'uks', 'hotspot', 'ber-ac', 'loker')
TOPIC 27: ('layanan', 'gratis', 'mengakses', 'penyedia', 'phonemarket', 'pelanggan', 'menyediakan', 'mobomarket', 'bimatri')
TOPIC 28: ('mobil', 'mesinnya', 'truk', 'chasis', 'berpenggerak', 'motor', '-silinder', 'roda', 'pengemudi')
TOPIC 29: ('mobile', 'kitkat', 'server', 'icloud', 'rtsp', 'kakaotalk', 'appstore', 'phone', 'smartphone')
TOPIC 30: ('singapore', 'centre', 'education', 'international', 'organization', 'schools', 'boarding', 'initiative', 'trade')
TOPIC 31: ('xt', 'xr', 'xq', 'xk', 'xw', 'xn', 'xj', 'xg', 'xh')
TOPIC 32: ('that', 'but', 'them', 'would', 'are', 'when', 'have', 'which', 'to')
TOPIC 33: ('jam', 'pukul', 'sabtu', 'senin', 'rabu', 'jumat', 'wib', 'senin-jumat', 'pagi')
TOPIC 34: ('surabaya', 'malang', 'semarang', 'jakarta', 'balikpapan', 'yogyakarta', 'salatiga', 'bandung', 'surakarta')
Document-Topic Counts:, [205 330 175 192 197 174 174 76 199 45 247 351 193 284 76 351 349 197
  96 193 100 158 126 349 233 19 239 282 107 26 105   4 164 221 168]

Halaman Sengaja Dikosongkan

LAMPIRAN D

D-1. Daftar 5 Kata dengan Probabilitas Tertinggi dalam Topik

Daftar 5 Kata – Data Frasa – Batasan 1 – *Stemming*

Topik 30	Topik 33	Topik 6	Topik 5	Topik 11	Topik 28	Topik 24	Topik 15	Topik 9	Topik 14
polri	warna	shalat	richi	samsung	rokok	kota	surabaya	boarding	internet
tni	hitam	jamaah	ahmad	phone	minum	pusat	semarang	lounge	laptop
udara	terang	shubuh	khoirul	ericsson	pipis	batas	jember	mdc	koneksi
kasi	baju	wudhu	andreas	xperia	bungkus	timur	yogyakarta	asia	google
darat	putih	ummat	abdul	sony	botol	daerah	pekanbaru	konter	handphone

Daftar 5 Kata – Data Frasa – Batasan 1 – Tanpa *Stemming*

Topik 30	Topik 24	Topik 12	Topik 14	Topik 6	Topik 9	Topik 7	Topik 20	Topik 27	Topik 16
shalat	arah	ahmad	makan	kejadian	pesawatnya	mobil	internet	tunggu	nyaman
bersuci	barat	khoirul	kasir	sopan	jet	bagasi	laptop	mepet	ramah
muslim	air	abdul	minum	jujur	bermesin	ukuran	drive	tolong	aman
jamaah	kota	fatah	minuman	sungguh	terbang	kendaraan	google	semoga	kesan
shubuh	kinabalu	richi	botol	bodoh	flight	troli	koneksi	diperbaiki	nada

Daftar 5 Kata – Data Frasa – Batasan 2 – *Stemming*

Topik 33	Topik 21	Topik 22	Topik 14	Topik 10	Topik 20	Topik 30	Topik 34	Topik 3	Topik 12
menit	salat	barat	rokok	airport	duduk	handphone	sedia	unjung	air
pagi	jamaah	bangkal	makan	airlines	lantai	koneksi	bawa	check_in	tawar
wib	ibadah	kota	minum	maskapai	rumah	samsung	keluh	ruang_tunggu	alami
malam	sejahtera	terap	ikan	cargo	emper	galaxy	senang	boardingpass	tahan
pkl	umroh	seberang	enak	jetstar	bangku	internet	koper	waiting_room	dingin

Daftar 5 Kata – Data Frasa – Batasan 2 – Tanpa *Stemming*

Topik 28	Topik 1	Topik 4	Topik 30	Topik 0	Topik 7	Topik 22	Topik 19	Topik 32	Topik 34
calo	air	web	samsung	aturan	memeriksa	pelayanan	tas	orang	uang
pengunjung	makan	resmi	phone	ditindaklanjuti	menghubungi	pengelola	buah	bernama	membayar
telpon	enak	google	mobile	hak	menginformasikan	publik	koper	wanita	tiket
bepergian	masak	situs	laptop	status	menyarankan	pemerintah	hitam	bom	pajak
perjalanan	rokok	youtube	iphone	konfirmasi	bantuan	kerja	batik	tinggal	rupiah

Daftar 5 Kata – Data Frasa – Batasan 3 – *Stemming*

Topik 23	Topik 14	Topik 28	Topik 18	Topik 16	Topik 17	Topik 20	Topik 34	Topik 13	Topik 12
berangkat	surabaya	laku	pagi	bayi	display	berat	shuttle	airlines	masjid
orang	garuda	kecewa	wib	sakit	info	angkat	landing	citilink	emper
jaga	semarang	nyata	malam	inap	cs	kg	flight	cargo	lantai
turun	jember	sulit	jam	hamil	note	jarak	angkasa	maskapai	fasilitas
pergi	barat	sopan	minggu	cacat	print	kosong	pura	airbus	mancur

Daftar 5 Kata – Data Frasa – Batasan 3 – Tanpa *Stemming*

Topik 30	Topik 24	Topik 28	Topik 9	Topik 0	Topik 14	Topik 18	Topik 20	Topik 33	Topik 11
security	tunggu	motor	sriwijaya	petugasnya	terbang	arah	pemberitahuan	jadwal	masuk
singapore	mepet	hp	bali	bantuan	pesawatnya	ukuran	status	pagi	orang
international	lumayan	diparkir	ngurah	imigrasi	mendarat	hitam	memeriksa	menit	salah
management	bagus	bagasi	rai	keamanan	buru	warna	diperbolehkan	tanggal	asia
boarding	enak	kendaraan	pura	penjaga	landing	kantong	konfirmasi	wib	saudara

Daftar 5 Kata – Data Tanpa Frasa – Batasan 1 – *Stemming*

Topik 15	Topik 16	Topik 22	Topik 31	Topik 19	Topik 12	Topik 27	Topik 11	Topik 1	Topik 17
diabetes	zam	zein	upg	jalur	tutup	maskapai	laptop	tindak	tas
dokter	wasalam	yuwono	udara	bus	hitam	airlines	koneksi	narkotika	buah
insulin	umrah	wahyu	tni	argo	warna	airways	drive	kantor	merk
cacat	ummat	untung	sriwijaya	tol	terang	qantas	google	prosedur	minum
sakit	surah	tri	polri	transportasi	baju	malindo	internet	operasional	kasir

Daftar 5 Kata – Data Tanpa Frasa – Batasan 1 – Tanpa *Stemming*

Topik 0	Topik 1	Topik 4	Topik 6	Topik 7	Topik 8	Topik 9	Topik 10	Topik 11	Topik 12
bebenah	ahmad	tarif	petugas	surabaya	air	pintu	tolong	internet	penerbangan
berlama	fata	berjalan	keluhan	jakarta	rafia	boarding	semoga	laptop	maskapai
boardingpass	wr	damri	aturan	semarang	kunci	lantai	cari	koneksi	airlines
diginikan	khoirul	bungurasih	jasa	barat	masak	jaga	biasakan	drive	airways
dikedepannya	abdul	bis	menjalankan	yogyakarta	rusak	mengantri	bilang	computer	jet

Daftar 5 Kata – Data Tanpa Frasa – Batasan 2 – *Stemming*

Topik 24	Topik 19	Topik 27	Topik 34	Topik 14	Topik 17	Topik 0	Topik 32	Topik 28	Topik 8
kecewa	barang	air	airlines	angkasa	timbang	rokok	orang	boarding	wifi
turun	bayar	tawar	citilink	flight	bangkal	minum	nama	garuda	laptop
pergi	cukai	bersih	maskapai	jet	wenang	kasir	aman	domestik	koneksi
tertawa	pajak	tahan	cargo	shuttle	layan	makan	saran	lumpur	handphone
marah	bea	kondisi	airline	landing	kalang	restoran	tinggal	kuala	internet

Daftar 5 Kata – Data Tanpa Frasa – Batasan 2 – Tanpa *Stemming*

Topik 23	Topik 9	Topik 5	Topik 13	Topik 7	Topik 34	Topik 24	Topik 27	Topik 26	Topik 29
ibadah	ketawa	tata	calo	teman	terbang	powerbank	jam	motor	jalur
shalat	cowok	sejahtera	penjual	anak	pesawatnya	kejelasannya	pagi	sepeda	argo
jamaah	bawa	pusat	kedai	perempuan	mendarat	boardingpass	menit	mobil	damri
salat	supir	pimpinan	makanan	tua	landasan	dibandara	diumumkan	kendaraan	tol
abu	kaget	pensiunan	bis	bernama	jet	avsec	jadwal	diparkir	angkutan

Daftar 5 Kata – Data Tanpa Frasa – Batasan 3 – *Stemming*

Topik 30	Topik 18	Topik 27	Topik 34	Topik 4	Topik 14	Topik 0	Topik 26	Topik 2	Topik 17
selamat	surabaya	sepeda	pagi	hp	tas	mesin	ray	delay	luas
macet	garuda	mobil	malam	hubung	buah	komputer	trolly	informasi	lindung
teman	semarang	motor	jadwal	berat	arah	kelas	avsec	sistem	area
marah	jember	seat	menit	jarak	koper	angkasa	shutle	aman	lokasi
hilang	pusat	transport	wib	maksimal	alas	komputer	penuruanan	deteksi	letak

Daftar 5 Kata – Data Tanpa Frasa – Batasan 3 – Tanpa *Stemming*

Topik 4	Topik 7	Topik 3	Topik 2	Topik 14	Topik 12	Topik 21	Topik 18	Topik 11	Topik 24
salut	suara	pemberitahuan	jalur	pelayanan	yogyakarta	pagi	langsung	nyaman	mengalami
makanan	lumpur	status	argo	pengelola	surabaya	malam	menit	ramah	aman
bau	kuala	penjelasan	tol	kerja	sriwijaya	wib	tanggal	nada	menginformasikan
minuman	malaysia	konfirmasi	jalan	publik	semarang	jam	salah	kesan	sistem
rokok	kursi	email	dibuka	pemerintah	malang	malam	pengumuman	berbicara	apresiasi

LAMPIRAN E

E-1 . Hasil Analisis Topik

Dokumen	Label	Top Topic				Kesesuaian
Dok1	12	12	5	7	6	Ya
Dok2	11	3	2	6	10	Tidak
Dok3	0	1	0	11	6	Ya
Dok4	4	9	2	11	4	Ya
Dok5	2	3	9	2	1	Tidak
Dok6	12	2	12	11	3	Ya
Dok7	0	12	10	1	9	Tidak
Dok8	1	0	7	2	1	Ya
Dok9	12	12	0	3	7	Ya
Dok10	11	9	1	0	3	Tidak
Dok11	4	7	10	4	1	Ya
Dok12	12	1	6	7	12	Ya
Dok13	9	1	10	7	9	Ya
Dok14	10	10	1	7	6	Ya
Dok15	12	12	0	5	3	Ya
Dok16	11	10	3	9	7	Tidak
Dok17	11	6	11	12	9	Ya
Dok18	4	10	7	1	4	Ya
Dok19	10	7	3	6	5	Tidak
Dok20	4, 7	5	7	10	1	Ya
Dok21	2	10	6	1	2	Ya
Dok22	9	3	5	0	9	Ya
Dok23	0, 7	7	10	6	1	Ya
Dok24	12	6	1	7	12	Ya
Dok25	4	4	3	9	10	Ya

Dokumen	Label	Top Topic				Kesesuaian
Dok26	12	9	5	6	12	Ya
Dok27	11	7	8	10	11	Ya
Dok28	11	5	1	4	3	Tidak
Dok29	0	10	0	3	1	Ya
Dok30	11	7	5	6	10	Tidak
Dok31	1, 4	10	6	1	2	Ya
Dok32	1, 4	9	4	11	3	Ya
Dok33	12	2	10	3	1	Tidak
Dok34	11	1	10	6	11	Ya
Dok35	0	6	8	0	12	Ya
Dok36	9	9	3	1	7	Ya
Dok37	11	5	12	10	6	Tidak
Dok38	9	6	4	2	1	Tidak
Dok39	4	9	3	4	2	Ya
Dok40	11	4	5	6	11	Ya
Dok41	4	3	7	1	0	Tidak
Dok42	4	10	6	8	1	Tidak
Dok43	9	0	4	10	3	Tidak
Dok44	4,1	1	3	7	10	Ya
Dok45	9, 4	4	6	7	2	Ya
Dok46	11	10	3	11	12	Ya
Dok47	11	3	5	6	11	Ya
Dok48	12	10	12	1	7	Ya
Dok49	11	9	7	11	4	Ya
Dok50	9	9	10	6	7	Ya
Dok51	7	1	0	2	0	Tidak
Dok52	12	2	8	12	1	Ya
Dok53	4	3	0	9	4	Ya
Dok54	0	2	1	7	0	Ya
Dok55	9	0	2	3	9	Ya

Dokumen	Label	Top Topic				Kesesuaian
Dok56	9	8	1	3	5	Tidak
Dok57	4	9	3	2	1	Tidak
Dok58	4	4	5	1	9	Ya
Dok59	4	0	1	4	5	Ya
Dok60	4, 0	9	2	1	0	Ya
Dok61	0	1	4	5	2	Tidak
Dok62	0	0	7	2	1	Ya
Dok63	1	4	1	6	7	Ya
Dok64	4	0	2	1	3	Tidak
Dok65	11	9	3	1	0	Ya
Dok66	1	1	4	3	9	Tidak
Dok67	1,4	8	1	9	7	Ya
Dok68	11	10	11	3	22	Ya
Dok69	0	0	2	6	1	Ya
Dok70	9	0	7	2	1	Tidak
Dok71	11	9	2	12	1	Tidak
Dok72	0,4,7	7	2	1	9	Ya
Dok73	11	9	11	2	9	Ya
Dok74	0, 4	3	4	1	2	Ya
Dok75	0, 4	0	3	1	7	Ya
Dok76	4	8	9	0	1	Tidak
Dok77	4	4	3	5	9	Ya
Dok78	4	1	4	2	11	Ya
Dok79	11	12	10	2	11	Ya
Dok80	4,0	12	11	0	3	Ya
Dok81	9	9	12	11	4	Ya
Dok82	4	4	5	3	1	Ya
Dok83	11	10	12	11	1	Ya
Dok84	4	7	3	2	1	Tidak
Dok85	4	3	4	7	9	Ya

Dokumen	Label	Top Topic				Kesesuaian
Dok86	9, 0	0	3	6	10	Ya
Dok87	11	10	11	12	0	Ya
Dok88	4	5	4	10	1	Ya
Dok89	10	11	10	3	4	Ya
Dok90	2	2	10	11	4	Ya
Dok91	11	3	6	10	11	Ya
Dok92	9	10	1	0	9	Ya
Dok93	0	10	11	12	6	Tidak
Dok94	11	12	11	10	4	Ya
Dok95	0,4,8	8	1	4	5	Ya
Dok96	4	9	3	7	6	Tidak
Dok97	11	11	12	10	3	Ya
Dok98	11	12	11	9	10	Ya
Dok99	11	3	4	9	11	Ya
Dok100	4	7	10	6	1	Tidak