TOPIC MODELLING SKRIPSI MENGGUNAKAN METODE *LATENT DIRICHLET ALLOCATION*

SKRIPSI

Disusun Oleh:

ALIF IFFAN ALFANZAR

H76215030

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN AMPEL

SURABAYA

2019

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama	: Alif Iffan Alfanzar
NIM	: H76215030
Program Studi	: Sistem Informasi
Angkatan	: 2015

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul: "*TOPIC MODELLING* SKRIPSI MENGGUNAKAN METODE *LATENT DIRICHLET ALLOCATION*". Apabila susatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya akan menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 26 Desember 2019

(Alif Iffan Alfanzar)

NIM. H76215030

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh:

NAMA : ALIF IFFAN ALFANZAR

NIM : H76215030

JUDUL : *TOPIC MODELLING* SKRIPSI MENGGUNAKAN METODE *LATENT DIRICHLET ALLOCATION.*

Telah di periksa dan disetujui untuk diajukan.

Surabaya, 29 April 2019

Dosen Pembimbing 1	Dosen Pembimbing 2
(Khalid, M. Kom)	(Indri Sudanawati Rozas, M. Kom)
NIP. 197906092014031002	NIP. 198207212014032001

Ketua Program Studi,
Sistem Informasi

(Muhammad Andik Izzudin, M.T)
NIP.19840307201431001

LEMBAR PENGESAHAN TIM PENGUJI

Skripsi Alif Iffan Alfanzar ini telah dipertahankan

di depan tim penguji skripsi

di Surabaya,

Mengesahkan,

Dewan Penguji

Dosen Penguji 1

(Khalid, M. Kom)
NIP. 197906092014031002

Dosen Penguji 2

(Indri Sudanawati Rozas, M. Kom)
NIP. 198207212014032001

Dosen Penguji 3

(Noor Wahyudi, M.Kom)
NIP. 198403232014031002

Dosen Penguji 4

(Mujib Ridwan, S.Kom., M.T)
NIP. 198604272014031004

Mengetahui,

Dekan Fakultas Sains dan Teknologi

UIN Sunan Ampel Surabaya

Dr. Evi Purwati, M.Ag
NIP. 196512211990022001

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN
Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : ALIF IFFAN ALFANZAR
NIM : H76215030
Fakultas/Jurusan : SAINS DAN TEKNOLOGI/SISTEM INFORMASI
E-mail address : alfanzar27@gmail.com

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :
■ Sekripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)
yang berjudul :

TOPIC MODELLING SKRIPSI MENGGUNAKAN METODE LATENT DIRICHLET ALLOCATION

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara ***fulltext*** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 5 Januari 2020

Penulis

(Alif Iffan Alfanzar)

ABSTRAK

TOPIC MODELLING SKRIPSI MENGGUNAKAN METODE

LATENT DIRICHLET ALLOCATION

Oleh:

Alif Iffan Alfanzar

Program Studi Sastra Inggris Universitas Islam Negeri Sunan Ampel Surabaya (UINSA) merupakan salah satu program studi yang skripsinya ditulis secara penuh menggunakan bahasa inggris. Permasalahan yang terjadi pada Program Studi Sastra Inggris UINSA adalah belum pernah dilakukan *clustering* pada topik skripsi yang telah diambil mahasiswa. Sedangkan *clustering* diperlukan untuk melihat tren dan kesesuaian konsentrasi pada Program Studi Sastra Inggris UINSA. *Latent Dirichlet Allocation* (LDA) merupakan salah satu metode dari *topic modelling* yang paling populer saat ini. Selain dapat meringkas, mengklusterkan, menghubungkan, LDA memiliki kelebihan utama yaitu mampu memproses data yang sangat besar. Untuk itu penelitian ini menggunakan metode LDA. Penelitian ini menggunakan dataset berupa 584 *abstract* skripsi pada Program Studi Sastra Inggris UINSA. Penggunaan dataset *abstract* Program Studi Sastra Inggris UINSA ini dikarenakan untuk *pre-processing*, data *Stopword* serta data pendukung proses *Lemmatization* dan *Stemming* yang tersedia standarnya baru untuk bahasa inggris. Dataset setelah melewati proses tersebut dijadikan sebagai *document term matriks* menggunakan metode *bag of word*. Metode LDA melakukan *clustering* dengan menggunakan *bag of word* sebagai kata yang diolah, kemudian menentukan jumlah *cluster* atau disebut dengan jumlah topik dan menentukan jumlah iterasi. Metode LDA menandai setiap kata pada topik yang di tentukan secara semi random distribution dan dihitung probabilitas topik pada dokumen dan probabilitas kata pada topik setiap iterasinya. Pada penelitian ini dilakukan percobaan sebanyak 5 uji iterasi dengan iterasi berbeda yakni: 100, 500, 1000, dan 5000. Sedangkan terhadap setiap uji iterasi dimasukkan jumlah topik yang berbeda yaitu: 2, 3, 4, 5, dan 7. Berdasarkan percobaan tersebut diperoleh hasil analisis bahwa 3 adalah jumlah topik yang paling fit. Hasil tersebut yang telah diuji secara kualitatif kepada stakeholder Program Studi Sastra Inggris, dan dinyatakan sesuai dengan tren serta konsentrasi yang ada pada Program Studi Sastra Inggris.

Kata Kunci: *Clustering*, Iterasi, *LDA*, Probabilitas, *Topic Modelling*.

ABSTRACT

TOPIC MODELLING OF SCIPSE AT ENGLISH DEPARTMENT

UIN SUNAN AMPEL SURABAYA

By:

Alif Iffan Alfanzar

English Literature Study Program Sunan Ampel State Islamic University Surabaya (UINSA) is one of the study programs whose thesis is written in full in English. The problem that occurs in UINSA's English Literature Study Program is that clustering has never been done on the thesis topic that students have taken. While clustering is needed to see trends and suitability of concentration in the UINSA English Literature Study Program. Latent Dirichlet Allocation (LDA) is one of the methods of the most popular topic modeling today. Besides being able to summarize, cluster, connect, LDA has the main advantage of being able to process very large data. For this reason, this research uses the LDA method. This research uses a dataset in the form of 584 thesis abstracts at the UINSA English Literature Study Program. The use of UINSA's English Literature Study abstract dataset is due to the pre-processing, Stopword data and supporting data for the Lemmatization and Stemming process, which are available for new English standards. The dataset after passing through the process is used as a matrix document term using the bag of word method. The LDA method does clustering by using a bag of words as a processed word, then determines the number of clusters or called the number of topics and determines the number of iterations. The LDA method marks each word on the topic in a semi random distribution and calculates the probability of the topic in the document and the probability of the word on the topic for each iteration. In this study, an experiment of 5 iterations with different iterations: 100, 500, 1000, and 5000. While for each iteration test, the number of different topics was entered: 2, 3, 4, 5, and 7. Based on these experiments, it was obtained the analysis results that 3 is the number of topics that are most fit. These results have been tested qualitatively to stakeholders of the English Literature Study Program, and are stated in accordance with the trends and concentrations that exist in the English Literature Study Program.

Key Word: Clustering, Iteration,* LDA*, Probability, Topic Modeling.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Saat ini perusahaan seringkali menggunakan sistem data mining secara efektif untuk menunjukkan segmentasi market dengan menggunakan data mining (Albert Verasius Dian Sano, 2019). *Data Mining* merupakan suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan pada *database*. *Data mining* adalah proses dengan menggunakan teknik statistik, matematika, kecerdasan buatan dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan terakit dari berbagai *database* besar (Turban, Aronson and Liang, 2004). Penerapan data mining bukan hanya dalam menganalisa perusahaan saja. Ada banyak bidang yang menjadikan data mining sebagai solusi untuk memecahkan masalah yang ada. Menurut (Putra, 2019) beberapa contoh penerapan *data mining* diantaranya adalah pada bidang telekomunikasi untuk melihat pola pada jutaan transaksi yang masuk, pada bidang keuangan untuk melihat pola pada transaksi keuangan yang mencurigakan, dan dalam internet *web surf-aid* untuk melihat pola perilaku dan minat *customer*. *Data mining* memiliki bidang khusus yang hampir sama, yakni *text mining* (Ryan Diaz, 2013). *Text mining* menurut (Ronen and Sanger, 2007) merupakan bagian dari *data mining* yang berupaya menemukan pola yang menarik dari sekumpulan data tekstual dalam jumlah yang besar. pencarian pola dalam *text* merupakan tujuan dari *text mining* melalui proses analisis *text* guna mencarikan informasi yang bermanfaat untuk tujuan tertentu.

Salah satu fungsi *data mining* maupun *text mining* adalah *clustering*. *Clustering* merupakan salah satu metode *data mining* atau *text mining* yang bersifat tanpa arahan (*unsupervised*). Tanpa arahan yang dimaksud dalam metode ini diterapkan tanpa adanya data latihan (*training*) dan tanpa ada guru serta tidak memerlukan target keluaran (*output*). *Clustering* dibagi menjadi dua jenis metode *cluster* yang digunakan dalam pengelompokan data, yaitu *hierarchical clustering* dan *non-hierarchical clustering*. *Hierarchical clustering* adalah suatu metode pengelompokan data yang dimulai dengan mengelompokkan dua atau lebih objek

yang memiliki kesamaan paling dekat. Kemudian proses diteruskan ke objek lain yang memiliki kedekatan kedua. Demikian seterusnya sehingga *cluster* akan membentuk semacam pohon 2 dimana ada hierarki (tingkatan) yang jelas antar objek, dari yang paling mirip sampai yang paling tidak mirip. Berbeda dengan metode *non-hierarchical clustering*, metode *non-hierarchical clustering* justru dimulai dengan menentukan terlebih dahulu jumlah *cluster* yang diinginkan (dua *cluster*, tiga *cluster*, atau lain sebagainya). Setelah jumlah *cluster* diketahui, baru proses *cluster* dilakukan tanpa mengikuti proses hierarki (Agusta, 2007).

Permasalahan yang terjadi saat ini pada Program Studi Sastra Inggris UIN Sunan Ampel Surabaya (UINSA) adalah belum mengetahui jumlah cluster topik penelitian untuk skripsi Program Studi Sastra Inggris UIN Sunan Ampel Surabaya (UINSA). Teknik pengklusteran pada topik penelitian dibutuhkan untuk melihat tren topik yang ada. Teknik mengklusterkan topik bisa dilakukan manual oleh manusia, akan tetapi dapat menghabiskan banyak waktu. Permasalahan waktu tersebut dapat diselesaikan dengan menggunakan bantuan *computer* dengan metode *topic modelling*. *Topic modelling* merupakan metode *non-hierarchical clustering* yang secara otomatis mengklusterkan kedalam topik yang muncul dari pemodelan sehingga didapatkan topik *cluster* yang sesuai. Solusi ini dapat mengatasi pemasalahan pada Program Studi Sastra Inggris UIN Sunan Ampel Surabaya saat ini.

Topic modelling mempunyai banyak metode yang dapat digunakan seperti *Latent Semantic Analysis* (LSA), *Probabilistic Latent Semantic Analysis* (PLSA), *Latent Dirichlet Allocation* (LDA). Menurut (Alghamdi, 2015) LSA merupakan sebuah metode pada bidang *Natural Language Processing* (NLP) yang mempunyai tujuan untuk menciptakan vektor berdasarkan representasi dari teks untuk membuat *semantic content*. PLSA merupakan sebuah pendekatan yang telah dirilis oleh Jan Puzicha dan Thomas Hofmann untuk memperbaiki metode LSA yang memiliki banyak kerugian. PLSA menurut (Hofmann, 2001) merupakan metode yang dapat mengotomatiskan pengindeksan dokumen yang didasarkan pada model kelas laten statistika untuk analisis faktor jumlah data, dan juga metode ini mencoba untuk meningkatkan metode LSA dalam arti probalistik dengan menggunakan *generatif model*. LDA Menurut (David M. Blei, Andrew Y. Ng, 2003) merupakan

peningkatan cara model campuran yang menangkap pertukaran dari kata-kata dan dokumen dari cara lama oleh PLSA dan LSA.

Metode LDA pada penelitian (Zulhanif, 2016) digunakan untuk pengidentifikasian dan pengklasteran topik dengan mengambil sampel sebanyak 1500 tweet. Metode pengklasteran LDA dengan mengelompokan data tweet tersebut menghasilkan 24 topik, pengklasteran ini diimplementasikan pada pemrograman R yang masih terbatas dalam hal jumlah dimensi. Penelitian yang dilakukan oleh (Fajriyanto, 2018) bertujuan untuk mengetahui berita apa yang dominan dibahas dalam masyarakat menggunakan metode LDA untuk pemodelan topik. Penelitian tersebut mengolah data tweet masyarakat yang sedang membicarakan berita yang diterbitakan oleh @kompascom. Implementasi metode LDA menggunakan bahasa pemrograman R didapatkan 10 topik. 10 topik ini diandingkan dengan berita terkait dari @kompascom dan mendapatkan topik ke 9 yang paling mendominasi dilihat dari nilai probabilitas topik dengan nilai 0.010057. Penelitian yang dilakukan (Putra and Renny Pradina Kusumawardani, 2017) menggunakan metode LDA untuk *topic modelling*. Dalam penelitian tersebut, (Putra and Renny Pradina Kusumawardani, 2017) mengambil *dataset facebook fanpage* dan *twitter*. Penelitian tersebut melakukan eksperimen penentuan jumlah topik menggunakan stem dengan penentuan jumlah topik tanpa menggunakan *stem*. *Topic modelling* dengan *stem* dalam Penelitian Tersebut menghasilkan 4 topik terbaik dalam membentuk *topic modelling*. Untuk itu metode LDA dipilih untuk penelitian ini dikarenakan metode LDA merupakan salah satu metode dari *topic modelling* yang paling populer saat ini. Selain dapat meringkas, mengklusterkan, menghubungkan, LDA memiliki kelebihan utama yaitu mampu memproses data yang sangat besar. Sehingga penelitian ini menggunakan metode LDA pada Program Studi Sastra Inggris UIN Sunan Ampel Surabaya (UINSA).

1.2 Rumusan Masalah

Berdasarkan pada latar belakang di atas, maka masalah pada penelitian ini adalah bagaimana proses implementasi *topic modelling* menggunakan metode *Latent Dirichlet Allocation* (LDA) pada data *abstract* skripsi Program Studi Sastra Inggris UIN Sunan Ampel Surabaya?

1.3 Batasan Masalah

Batasan Masalah dalam penelitian ini adalah:

1. Data yang diperoleh berasal dari *digital library* (digilib) UIN Sunan Ampel Surabaya (UINSA).
2. Studi kasus yang diteliti yaitu *abstract* penelitian pada Program Studi Sastra Inggris UIN Sunan Ampel Surabaya (UINSA) tahun 2014-2019 yang berjumlah 580 *abstract*.
3. Tools pada penelitian ini menggunakan *web scrapper*, *jupytherlab.*
4. Bahasa pemograman untuk *topic modelling* menggunakan bahasa pemrograman python.
5. Metode yang digunakan dalam *topic modelling* menggunakan metode *Latent Dirichlet Allocation*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini untuk mengetahui bagaimana proses implementasi *topic modelling* menggunakan metode Latent Dirichlet Allocation (LDA) pada data *abstract* skripsi Program Studi Sastra Inggris UIN Sunan Ampel Surabaya.

1.5 Manfaat Penelitian

Manfaat yang di dapat dari penelitian ini yaitu:

1. Manfaat akademis:
   a. Menambah wawasan bagaimana mengimplementasikan metode LDA dalam mengclusterkan topik penelitian Sastra Inggris UINSA.
   b. Mendapatkan banyak ilmu dan pengetahuan terkait *data mining*, *text mining*, dan metode LDA.
2. Manfaat praktis:
   a. Membantu pihak Program Studi Sastra Inggris UINSA dalam menentukan jumlah cluster topik penelitian.
   b. Pemodelan yang dilakukan dalam penelitian ini dapat menghemat waktu, tenaga dan biaya.

BAB II
TINJAUAN PUSTAKA

Pada bab ini berisi tentang tinjauan penelitian sebelumnya yang berhubungan dengan penelitian selanjutnya, dan juga teori dasar sebagai pendukung penelitian.

2.1 Tinjauan Penelitian Terdahalu

Tinjauan penelitian terdahulu terkait dengan penelitian yang relavan dengan penelitian yang telah dilakukan. Penelitian ini berkaitan dengan permasalahan dalam beberapa penelitian yang memiliki kata kunci yaitu *topic modeling* seperti terlihat pada Tabel 2.1 berikut:

Tabel 2.1 Tinjauan Pustaka Terdahulu

Judul	Tahun	Metode	Studi Kasus	Sumber data
Penerapan Metode Bayersian Dalam *Model Latent Dirichlet Allocation* (LDA) Di Media Sosial	2018	Bayesian *Latent Dirichlet Allocation* (LDA)	Media Sosial	@kompascom Twitter
Analisis Topik Informasi Publik Media Sosial di Surabaya Menggunakan Pemodelan *Latent Dirichlet Allocation* (LDA)	2017	*Latent Dirichlet Allocation* (LDA)	Radio Suara Surabaya	@e100ss Twitter
Analisis Topik Data Media Sosial Twitter Menggunakan Model Topik	2017	*Latent Dirichlet Allocation* (LDA)	Twitter	Sumber data tidak disertakan
Pemodelan Topik Dengan *Latent Dirichlet Allocation* (LDA)	2016	*Latent Dirichlet Allocation* (LDA)	Data Twitter	#bandung

Dari penjelasan yang telah dijelaskan terkait dengan penelitian sejenis tersebut, terdapat relevansi dan perbedaan dengan penelitian penulis. Pada peneitian yang dilakukan oleh penulis dijelaskan sebagai berikut:

1) Dalam penelitian (Fajriyanto, 2018) penelitian ini bertujuan untuk mengetahui berita apa yang dominan dibahas dimasyarakat dengan metode

Latent Dirichlet Allocation (LDA) dengan studi kasus @kompascom pada pemrograman R.

2) Dalam penelitian (Putra and Renny Pradina Kusumawardani, 2017) pemodelan topik yang mampu secara otomatis mengklasifikasikan pesan media sosial dengan menggunakan metode *Latent Dirichlet Allocation* (LDA) studi kasus Radio Suara Surabaya.
3) Dalam Penelitian (Utami, 2017) Menganalisis Topik Data Media Sosial Twitter dengan menggunakan metode *Latent Dirichlet Allocation* (LDA) menggunakan bahasa pemrograman R.
4) Dalam penelitian (Zulhanif, 2016) metode yang digunakan untuk topik modeling menggunakan metode *Latent Dirichlet Allocation* (LDA) pada studi kasus sample sebanyak 1500 tweet dengan kata kunci #bandung menggunakan software R.

2.2 Teori Dasar yang Digunakan

2.2.1 *Data Mining*

Pengertian Data mining menurut (Turban, Aronson and Liang, 2005) ialah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam database. Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar.

Menurut Gartner Group, data mining adalah proses menemukan hubungan baru yang mempunyai arti, pola dan kebiasaan dengan memilah-milah sebagian besardata yang disimpan dalam media penyimpanan dengan menggunakan teknologi pengenalan pola seperti teknik statistik dan matematika. Data mining merupakan gabungan dari beberapa disiplin ilmu yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, database, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari database yang besar (Daniel, 2005)

2.2.2 *Text Mining*

Text mining dapat diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, secara otomatis mengekstrak informasi dari sumber-sumber yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber (Marti Hearst, 2003).

Menurut (Marti Hearst, 2003) *Text mining* merupakan sebuah variasi dari bidang *Data mining* yang mencoba menemukan pola-pola menarik dari *database*. Contoh pada umumnya dalam *data mining* yaitu mengetahui pola produk apa saja yang dibeli oleh konsumen sehingga produk tersebut diletakkan berdekatan dalam sebuah rak.

Pada *Text Mining*, teknik *clustering* digunakan untuk mengelompokkan data tekstual berdasarkan kesamaan konten yang dimiliki ke dalam beberapa klaster, sehingga didalam setiap klaster akan berisi data tekstual dengan konten semirip mungkin (Ronen and Sanger, 2007). Menurut (Yahir Even dan Zohar, 2002) *text mining* memiliki tiga tahap utama, proses-proses tersebut diantaranya adalah proses awal terhadap teks (*text preprocessing*), transformasi text (*text transformation*), dan penemuan pola (*pattern discovery*).

2.2.3 *Pre-Processing*

Dalam penelitian (Jaka, 2015) *Pre-processing* adalah proses pengubahan bentuk data yang belum terstruktur menjadi data yang terstruktur sesuai dengan kebutuhan, yang dilakukan untuk proses mining yang lebih lanjut. Tahap-tahap pada *text pre-processing* secara umum adalah *tokenizing*, penghapusan *Stopwords*, *lemmatization*, dan *stemming*.

A. Tokenizing

Dalam penelitian (Utami, 2017) *Tokenizing* adalah proses memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi token atau potongan kata tunggal atau *termmed word* yang berdiri sendiri. Di dalam *tokenizing* karakter dan *symbol* selain a-z dihilangkan, pemecahan kalimat dan kata dilakukan berdasarkan pada spasi di dalam kalimat tersebut. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (*lower case*).

B. Penghapusan *Stopwords*

Menurut (Kaur and Buttar, 2018) *stopword* yang dikenal sebagai *stoplist* merupakan sebuah metode otomatis yang dikembangkan untuk mengidentifikasi sebuah data dengan menghapus data dari dataset sebelumnya. Hal ini dilakukan untuk mengambil informasi yang mempunyai proporsi besar yang berguna bagi yang melakukan penelitian. *Stopwords* pertama kali diperkenalkan pada tahun 1958 oleh H.P. Luhn. *Stopwords* adalah kata-kata yang paling sering muncul dalam sebuah dokumen dan berisi sedikit informasi yang biasanya tidak diperlukan dalam penelitian. Misalnya, dalam bahasa inggris ada beberapa kata sperti di atas, setelah, lagi, terhadap, semua, pagi, sebuah, dan, menjadi, selama, dan lain-lain.

C. Lemmatization

Mengemukakan (Ingason *et al.*, 2008) bahwa *lemmatization* adalah sebuah proses untuk menemukan bentuk dasar dari sebuah kata. mendukung teori ini dengan kalimatnya yang mengatakan bahwa *lemmatization* merupakan proses yang bertujuan untuk melakukan normalisasi pada teks/kata dengan berdasarkan pada bentuk dasar yang merupakan bentuk lemma-nya. Normalisasi disini adalah dalam artian mengidentifikasikan dan menghapus prefiks serta suffiks dari sebuah kata. *Lemma* adalah bentuk dasar dari sebuah kata yang memiliki arti tertentu berdasar pada kamus.

D. Stemming

Stemming adalah proses pencarian bentuk dasar suatu kalimat dengan cara menghilangkan imbuhannya. *Stemming* merupakan suatu proses yang terdapat dalam sistem IR yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu. *Stemming* sangat penting dalam mendukung efektivitas pencarian informasi dalam bahasa Indonesia, penerjemahan dokumen, dan pencarian dokumen teks. Imbuhan bahasa Indonesia lebih kompleks dari pada bahasa inggris karena di dalam bahasa Indonesia terdapat awalan (*prefiks*), sisispan (*infiks*), akhiran (*sufiks*), *konfiks* (gabungan *prefiks* dan *sufiks*). Sehingga stemming bahasa Indonesia harus mampu menemukan akar kata sesuai dengan aturan baku bahasa Indonesia.

2.2.4 *Bag of Word*

Dalam model ini, sebuah teks yang berupa kalimat ataupun dokumen diwakili sebagai kantung (*bag*) multiset dari kata-kata yang terkandung di dalamnya, tanpa memandang urutan kata dan tata bahasa namun tetap mempertahankan keberagamannya. Definisi lain untuk *bag of word* adalah sebuah model yang mempelajari sebuah kosakata dari seluruh dokumen, lalu memodelkan tiap dokumen dengan menghitung jumlah kemunculan setiap kata (S, Raj and S.Rajaraajeswari, 2016).

2.2.5 *Topic Modelling*

Topic modelling menurut (David M. Blei, Andrew Y. Ng, 2003) terdiri dari entitas-entitas yaitu "kata", "dokumen", dan "corpora". "Kata" dianggap sebagai unit dasar dari data diskrit dalam dokumen, yang didefinisikan sebagai item dari kosakata yang diberi indeks untuk setiap kata unik yang ada dalam dokumen. "Dokumen" merupakan susunan N kata-kata. Sebuah *corpus* adalah kumpulan M dokumen dan corpora merupakan bentuk jamak dari *corpus*. "Topic" adalah distribusi dari beberapa kosakata yang bersifat tetap. Secara sederhana, setiap dokumen dalam *corpus* mengandung proporsi yang berbeda dari topik-topik yang dibahas sesuai dengan kata-kata yang ada di dalamnya.

Ide dasar dari *topic modeling* ialah sebuah topik terdiri dari kata-kata tertentu yang menyusun topik tersebut, dan dalam satu dokumen memiliki kemungkinan terdiri dari beberapa topik dengan probabilitas masing-masing. Namun manusia memahami dokumen-dokumen merupakan sebuah objek yang dapat diamati, sedangkan topik, distribusi topik per-dokumen, dan penggolongan setiap kata pada topik perdokumen merupakan bagian yang tersembunyi, maka dari itu *topic modelling* bertujuan untuk menemukan topik dan kata yang yang tersembunyi pada topik tersebut.

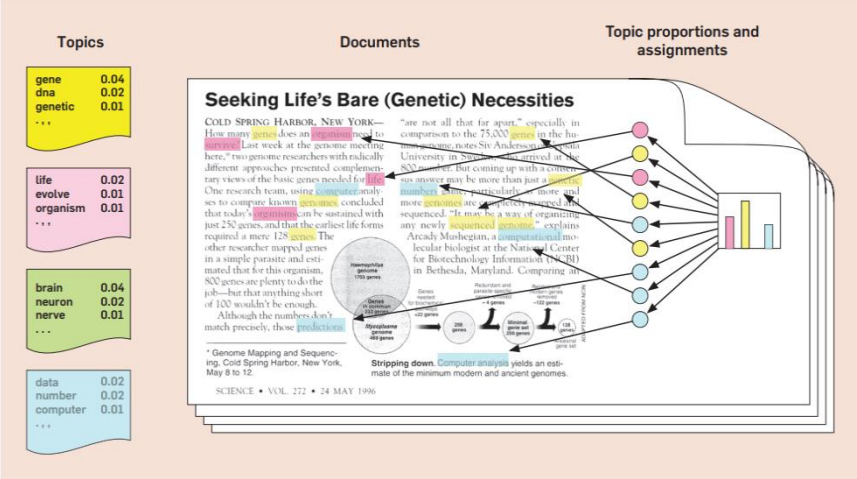
Kumpulan dokumen memiliki distribusi probabilitas topik, setiap kata dianggap diambil dari salah satu topik tersebut. Dengan distribusi probabilitas topik di setiap dokumen, dapat diketahui seberapa banyak masing-masing topik terlibat dalam sebuah dokumen. Hal ini dapat mengetahui topik mana yang terutama dibicarakan suatu dokumen.

2.2.6 *Latent Dirichlet Allocation* (LDA)

Latent Dirichlet Allocation (LDA) merupakan metode *topic modelling* dan topik analisis yang paling populer saat ini. LDA muncul sebagai salah satu metode yang dipilih dalam melakukan analisis pada dokumen yang berukuran sangat besar. LDA dapat digunakan untuk meringkas, melakukan klasterisasi, menghubungkan maupun memproses data yang sangat besar karena LDA menghasilkan daftar topik yang diberi bobot untuk masing-masing dokumen (Campbell, Hindle and Stroulia, 2014).

Model topik LDA merupakan model probabilistik generatif yang memungkinkan data teramati dapat dijelaskan oleh data tersembunyi yang menjelaskan mengapa beberapa bagian data dapat serupa. Intuisi dibalik LDA adalah dokumen menunjukkan beberapa topik. Setiap dokumen dianggap sebagai campuran topik korpus ukuran besar. Sebuah topik adalah distribusi kata dari korpus. Kumpulan topik dihasilkan dari kumpulan dokumen. Sebagai contoh, topik olahraga memiliki kata "basket", "berenang" dengan probabilitas tinggi. Topik komputer memiliki kata "data", "internet" dengan probabilitas tinggi.

Menurut (David M. Blei, Andrew Y. Ng, 2003) *Latent Dirichlet Allocation* (LDA) merupakan *topic modelling* yang paling sederhana. Sebagai contoh digunakan artikel berjudul *Seeking Life's Bare* (*Genetic*) *Necessities* untuk melihat intuisi dibalik LDA seperti pada Gambar 2.1. Dengan menyesuaikan Gambar 2.1, berasumsi bahwa terdapat sejumlah topik dengan distribusi kata untuk keseluruhan dokumen (bagian kiri gambar). Setiap dokumen diasumsikan untuk dihasilkan sebagai berikut. Pilih distribusi topik (histogram pada bagian kanan gambar). Kemudian untuk setiap kata, pilih penugasan topik (koin yang berwarna pada gambar) dan pilih kata dari topik yang sesuai. Distribusi yang digunakan untuk menarik distribusi topik per-dokumen (histogram bagian kanan gambar) disebut dengan distribusi *dirichlet*. Pada proses generatif LDA, hasil dari distribusi *dirichlet* digunakan untuk mengalokasikan kata dalam dokumen ke topik berbeda.



Gambar 2.1 *Topic Modelling*

Terdapat karakteristik LDA yang mana semua dokumen berbagi kumpulan topik yang sama. Tetapi setiap dokumen menunjukkan topik-topik tersebut dalam proporsi yang berbeda. Seperti tujuan pemodelan topik yaitu secara otomatis menemukan topik dari kumpulan dokumen. Kumpulan dokumen adalah data teramati, sedangkan stuktur topik yang berupa kumpulan topik, distribusi topik perdokumen, dan penugasan topik per-kata per-dokumen adalah struktur tersembunyi. LDA dan pemodelan topik lainnya adalah bagian dari pemodelan probabilistik.

Dalam pemodelan probabilistik generatif, data yang muncul dari proses generatif mencakup variabel tersembunyi. Proses generatif mendefinisikan distribusi probabilitas bersama untuk kedua variabel teramati dan variabel acak tersembunyi. Analisis data menggunakan distribusi bersama untuk menghitung distribusi bersyarat dari variabel tersembunyi (struktur topik) dengan variabel teramati yang diketahui. Distribusi bersyarat disebut juga dengan distribusi *posterior*.

Mengasumsikan proses generatif berikut untuk setiap dokumen $\boldsymbol{w}$ dalam sebuah corpus $\boldsymbol{D}$ adalah sebagai berikut:

1. Pilih $N \sim$ Pissson(ξ)
2. Pilih $\boldsymbol{\Theta} \sim$ Dir(α)
3. Untuk setiap N kata : W_n
   a. Pilih Topik $z_n \sim$ *Multinomial* ($\boldsymbol{\Theta}$)
   b. Pilih sebuah kata dari w_n dari $p(w_n \mid z_n, \beta)$

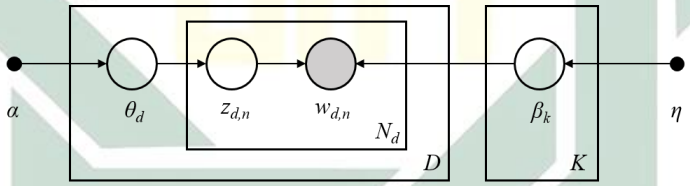
Beberapa asumsi penyederhanaan yang dibuat dalam model dasar LDA. Pertama, distribusi dari topik (*latent*) diketahui mengikuti *k* distribusi *Dirichlet*. Kedua, probabilitas kata adalah matriks **β** berukuran *k x V* yang mana $\beta_{ij} = p(w^j = 1 \mid z^i = 1)$. Sedangkan *k* distribusi *Dirichlet* memiliki fungsi densitas sbb:

$$p(\theta \mid \alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1} \tag{2.1}$$

Bentuk distribusi bersama dari Topik mixture Θ dari *N* topik z dan *N* kata *w* bersyarat α dan β :

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \tag{2.2}$$

Representasi model LDA jika digambarkan dalam sebuah diagram dapat digambarkan sebagai berikut:



Gambar 2.2 Representasi Model LDA

Berdasarkan gambar 2.2 didapat distribusi bersama yang dari parameter pada model LDA sbb:

$$p(w, z, \theta, \phi \mid \alpha, \beta) = p(\theta \mid \alpha) p(z \mid \theta) p(\phi \mid \beta) p(w \mid z, \phi) \tag{2.3}$$

Pada Persamaan (2.3) diasumsikan bahwa topik setiap dokumen mengikuti distribusi *Dirichlet* sbb:

$$p(\theta \mid \alpha) = \frac{\Gamma(\alpha)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \tag{2.4}$$

Sedangkan distribusi peluang dari *z* untuk semua dokumen dan topik dalam terms dinotasikan $n_{d,\,k}$ yang merupakan berapa banyak topik *k* dikelompokan dengan kata dalam dokumen *d*:

$$p(z \mid \theta) = \prod_{d=1}^{D} \prod_{k=1}^{K} \theta_{d,k}^{n_{d,k}} \tag{2.5}$$

Distribusi peluang bersyarat untuk semua corpus $\emptyset_k$ juga mengikuti distribusi Dirichlet dengan parameter $\beta\ \emptyset_{k,v}$. yang diformulasikan pada Persamaan.

$$p(\phi \mid \beta) = \prod_{k=1}^{K} \frac{\Gamma(\beta_{k,\cdot})}{\prod_{v=1}^{V} \Gamma(\beta_{k,v})} \prod_{v=1}^{V} \phi_{k,v}^{\beta_{k,v} - 1} \tag{2.6}$$

Lda bekerja dengan menggunakan gibbs sampling. Gibbs sampling adalah salah satu algoritma keluarga dari Markov Chain Monte Carlo (MCMC). Kurang lebih intinya adalah kita bisa menghitung joint probability distribution dengan cara melakukan sampling satu per satu terhadap setiap variabel dengan berdasarkan nilai variabel lainnya. Dalam melakukan pengelompokan topik ada dua bentuk distribusi probabilitas yang harus dicari yaitu:

A. Distribusi Probabilitas dokumen pada suatu dokumen.

Probabilitas topik pada dokumen merupakan nilai probabilitas tiap topik pada suatu dokumen. misal pada dokumen I mempunyai probabilitas topik A senilai X, mempunyai probabilitas topik B senilai Y dan seterusnya sesuai dengan jumlah topik. nilai tersebut dapat kita temukan melalui rumus pada berikut ini.

$$\text{Probabilitas topik pada suatu dokumen } (\mathrm{k}, \mathrm{d}, \text{alpha} = 0.1)$$

$$\frac{\text{Jumlah topik } \mathbf{k} \text{ pada dokumen } \mathbf{d}+\text{alpha}}{\text{Panjang dokumen } \mathbf{d}+\text{jumlah topik}*\text{alpha}} \quad (2.7)$$

B. Distribusi Probabilitas kata pada suatu topik.

Probabilitas kata pada topikmerupakan nilai probabilitas tiap kata pada suatu topik. misal pada kata I mempunyai probabilitas topik A senilai X, mempunyai probabilitas topik B senilai Y dan seterusnya sesuai dengan jumlah topik. nilai tersebut dapat kita temukan melalui rumus pada berikut ini.

$$\text{Probabilitas kata pada suatu topik } (\mathrm{t}, \mathrm{k}, \mathrm{beta} = 0.1)$$

$$\frac{\text{Jumlah kata } \mathbf{t} \text{ pada topik } \mathbf{k} + \text{beta}}{\text{total kata pada topik } \mathbf{k} + \text{jumlah distinct word} * \text{beta}} \quad (2.8)$$

2.2.7 Integrasi Keilmuan

Integrasi keilmuan memaparkan bagaimana konsep hunian dari sudut pandang islam. Untuk mengetahui konsep integrasi keilmuan tersebut dilakukan wawancara kepada salah satu tokoh agama bernama Ustad Ahmad Baidowi. Konsep *topic modelling* dalam perspektif islam sama halnya dengan seruan untuk menggali informasi atau untuk menimba ilmu sedalam dalamnya. Dalam (Kahfi, 2006) Informasi yang disampaikan bertujuan untuk mencapai efektivitas pengaruh informasi yang tidak merugikan kedua belah pihak, al Qur'an dan al Hadits telah memberikan beberapa aturan yang perlu ditaati oleh setiap individu yang mengaku sebagai seorang Muslim seperti yang ada dalam uraian berikut ini:

1. Qashash/Naba al Haq, yaitu informasi yang disampaikan harus menggambarkan kisah, berita, dan informasi yang benar, terutama yang berhubungan dengan isi informasi yang disampaikan. Hal ini sejalan dengan pola al-Quran dalam menceritakan kisah yang terjadi pada para Rasul Allah dan berita tentang sekelompok atau individu manusia yang terjadi pada kehidupan masa lalu. Hal ini dapat dilihat dalam QS. Hud ayat 120, QS. Yusuf ayat 3, dan Al-Kahfi ayat 13.

وَكُلًّا نَقُصُّ عَلَيْكَ مِنْ أَنْبَاءِ الرُّسُلِ مَا نُثَبِّتُ بِهِ فُؤَادَكَ ۚ وَجَاءَكَ فِي هَٰذِهِ الْحَقُّ وَمَوْعِظَةٌ وَذِكْرَىٰ لِلْمُؤْمِنِينَ

Dan semua kisah dari rasul-rasul Kami ceritakan kepadamu, ialah kisah-kisah yang dengannya Kami teguhkan hatimu; dan dalam surat ini telah datang kepadamu kebenaran serta pengajaran dan peringatan bagi orang-orang yang beriman (QS. 11:120).

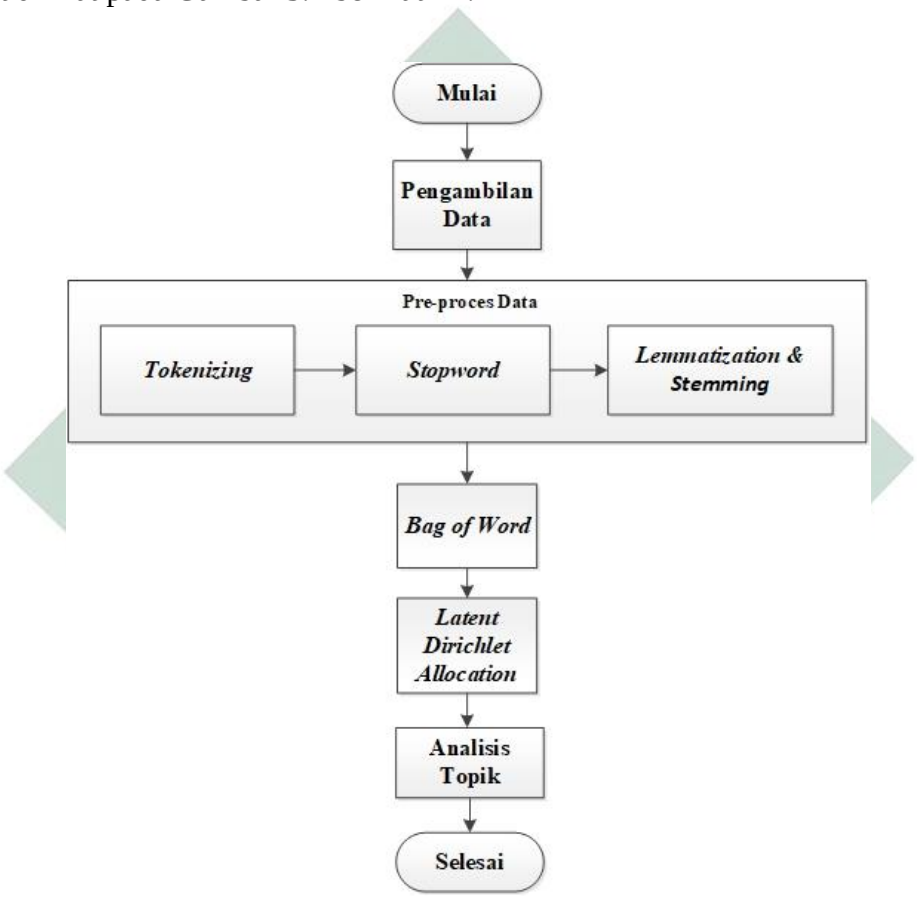
2. Tabayyun, yaitu informasi yang disampaikan telah melalui upaya klarifikasi. Artinya, menyampaikan informasi setelah dicari kejelasan dari sumber utama, bahkan beberapa sumber yang dianggap bisa memberikan kejelasan informasi QS. Al-Hujurat ayat 6, sehingga informasi yang disampaikan dapat bersifat adil (tidak berpihak).

يَا أَيُّهَا الَّذِينَ آمَنُوا إِنْ جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَنْ تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْبِحُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ

Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu (QS. 49:6).

BAB III
METODOLOGI PENELITIAN

Metodologi penelitian merupakan cara atau prosedur beserta langkah-langkah yang tersusun secara sistematis untuk menyelesaikan suatu permasalahan yang sedang diteliti dengan landasan ilmiah tertentu. Kerangka metodologi penelitian dapat dilihat pada Gambar 3.1 berikut ini:



Gambar 3.1 Kerangka Metodologi Penelitian.

Dalam penelitian *topic modeling* untuk skripsi dilakukan dengan langkah-langkah sebagai berikut:

3.1 Tempat dan Waktu Penelitian

Penelitian ini bertempat pada Universitas Islam Negeri Sunan Ampel Surabaya yang bertempat di Jl. Jend. A. Yani 117 Surabaya. Waktu penelitian ini dilaksanakan pada tanggal 16 Oktober 2019 – 12 Desember 2019.

3.2 Data Peneltian

Data yang digunakan dalam penelitian ini berupa *abstract* penelitian Program Studi Sastra Inggris UINSA yang diperoleh pada website http://digilib.uinsby.ac.id menggunakan ekstensi *Web Scrapper* yang ada pada *Google Chrome*. Data yang diambil pada penelitian ini adalah data *abstract* penelitian Program Studi Sastra Inggris UINSA dari tahun 2014 – 2019 sejumlah 584 *rows abstract*.

3.3 Langkah – Langkah Penelitian

3.3.1 Pengambilan Data

Tahap pertama penelitian adalah pengambilan data dengan mengambil data dari *website* digilib uinsa. Proses pengambilan data menggunakan *tool* dari *Google Chrome* yang bernama *web scrapper*. Penelitian ini mempunyai batasan masalah yaitu mengambil data *abstract* pada penelitian yang dilakukan oleh program studi sastra inggris UINSA.

3.3.2 *Pre-processing data*

Pada tahap *Pre-processing* merepresentasikan koleksi dokumen kedalam bentuk tertentu untuk memudahkan dan mempercepat proses pencarian dan penemuan kembali dokumen yang relevan. Pembangunan *index* dari koleksi dokumen merupakan tugas pokok pada tahapan *pre-processing*. *Index* akan membedakan suatu dokumen dari dokumen lain yang berada di dalam koleksi. Pembuatan *inverted index* harus melibatkan konsep *linguistic processing* yang bertujuan mengekstrak term-term penting dari dokumen yang direpresentasikan sebagai *bag of words*. Konsep *linguistic processing* terdiri dari *tokenizing*, *stopwords*, *lemmatization* dan *stemming*. Berikut merupakan empat tahap *pre-process*.

a. *Tokenizing*

Pada tahap pertama dari *pre-processing* data adalah *tokenizing.* Penggunaan tokenisasi ini dengan menggunakan *library* nltk dengan mengimport word_tokenize. kemudian mengimport *RegexpTokenizer* dari nltk.tokenize. Proses *tokenizing* memecah kalimat menjadi kata yang berdiri sendiri berdasarkan pada spasi di dalam kalimat tersebut. Tahapan ini juga

menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil. Hasil dari proses *tokenizing* berbetuk array yang berisi kata-kata atau term yang sudah terpisah atau.

b. *Stopwords*

Pada tahap ini dilakukan penghapusan *stopwords* berdasarkan kata yang telah ditentukan atau *stoplist*. Kata-kata tersebut akan secara otomatis di hapus apabila ada kata yang sama dalam term terhadap *stoplist*.

c. *Lemmatization* dan *stemming*

Pada tahap Lemmatization ini data term atau *text* diproses untuk menemukan sebuah bentukan dasar dengan menganalisis setiap kata dalam kata dasar yang terdapat pada bahasa yang digunakan.Tahap *stemming* juga bertujuan untuk mendapatkan kata dasar dengan memotong awalan dan akhiran setiap kata yang berimbuhan.

3.3.3 *Bag of words*

Setelah proses pre-processing dilakukan, matriks yang berisi kata-kata tersebut dimodelkan dengan model *bag of words*. *Bag of words* digunakan untuk memodelkan setiap dokumen dengan menghitung jumlah kemunculan setiap katanya. Setiap dokumen direpresentasikan dengan model *bag of words* yang mengabaikan urutan dari kata-kata di dalam dokumen, struktur sintaktis dari dokumen dan kalimat. Nilai perhitungan jumlah kemunculan setiap kata tersebut digunakan dalam *topic modelling*.

3.3.4 *Latent Dirichlet Allocation*

Proses pemodelan topik bertujuan untuk memperoleh distribusi kata yang membentuk suatu topik dan dokumen dengan topik tertentu. Pemodelan topik memiliki dua tahapan yang dilakukan. Tahap pertama adalah melakukan pemodelan topik berdasarkan penambahan dan pengurangan jumlah topik. Tahap kedua adalah melakukan pemodelan topik berdasarkan banyaknya iterasi. Hasil dari kedua pemodelan topik kemudian dilakukan analisa dengan cara membandingkan kata kata setiap klasternya dalam topik dan melihat visualisasi dari pemodelan LDA tersebut. Proses pemodelan topik dapat berulang selama rentang kandidat jumlah topik dan jumlah iterasi yang di tentukan.

Latent Dirichlet Allocation (LDA) merupakan salah satu model dari pemodelan topik. Model topik LDA merupakan *unsupervised machine learning.* Model tersebut berguna dalam mengidentifikasi informasi tersembunyi dalam kumpulan dokumen yang berukuran besar. Metode ini dapat diselesaikan menggunakan *python,* dengan terlebih dahulu mengaktifkan *package* "LdaModel" dalam *library* gensim. *Package* "LdaModel" untuk memodelkan probabilitas kemunculan kata dalam dokumen. Menghasilkan data keluaran berupa grafik yang menunjukan topik pada data yang diteliti.

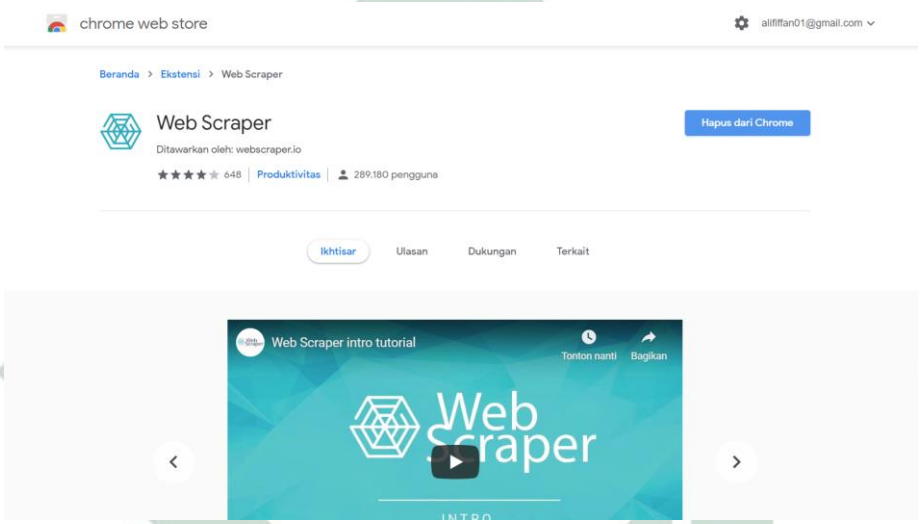
3.3.5 Analisis Topik

Analisis topik dilakukan berdasarkan pada data keluaran dari tahap sebelumnya. Pada tahap sebelumnya diperoleh grafik data keluaran dari kumpulan penelitian dengan topik tertentu. Analisis topik dilakukan secara subjektif dengan melihat data keluaran. Data keluaran berupa kumpulan kata yang membentuk topik, kemudian setiap dokumen tersebut disesuaikan dengan data keluaran yang memuat dokumen dengan topik. Proses ini menghasilkan deskripsi topik yang bersifat informatif mengenai hal yang dapat mewakili isi dari masing-masing topik tersebut.

BAB IV
HASIL DAN PEMBAHASAN

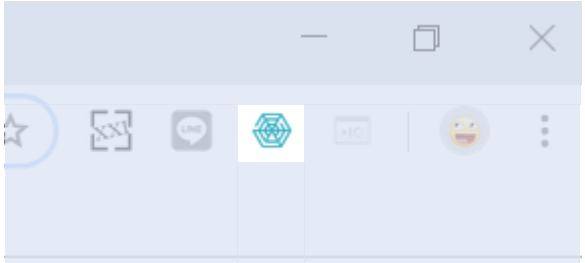
4.1 Pengambilan Data

Tahap pertama yang dilakukan dalam penelitian ini adalah pengambilan data. Pada proses pengambilan data digunakan metode *scrapping* dengan menggunakan *tools extension google chrome* yang bernama *web scraper*. *web scraper* tersebut diunduh pada *chrome web store* seperti pada Gambar 4.1.



Gambar 4.1 Halaman Download Web Scraper

Setelah *web scraper* sudah terunduh maka muncul ikon berbentuk jaring laba-laba pada pojok kanan *browser google chrome*. Munculnya ikon sebagaimana Gambar 4.2 di bawah ini menandakan bahwa *Web Scraper* telah siap digunakan.



Gambar 4.2 Ikon Web Scraper

Selanjutnya mengambil data yang dibutuhkan dengan menggunakan *web scraper* dengan terlebih dahulu membuka *web* yang akan diambil datanya. Dimana

pada penelitian ini mengambil data pada digilib uinsby http://digilib.uinsby.ac.id/. Data-data yang diambil merupakan data *abstract* pada penelitian program studi sastra inggris. Data *abstract* yang diambil berdasarkan rentang waktu lima tahun kebelakang, yakni mulai tahun 2014-2019.

Berikut merupakan langkah-langkah pengambilan data abstract menggunakan *web scraper*:

1. Membuka *link* data yang diambil.
2. Mencari letak data.
3. Membuat *sitemap*.
4. Menambahkan *selector* baru.
5. Mengecek pengurutan data.
6. Mengambil data secara berurut dan otomatis.
7. Menyimpan data.

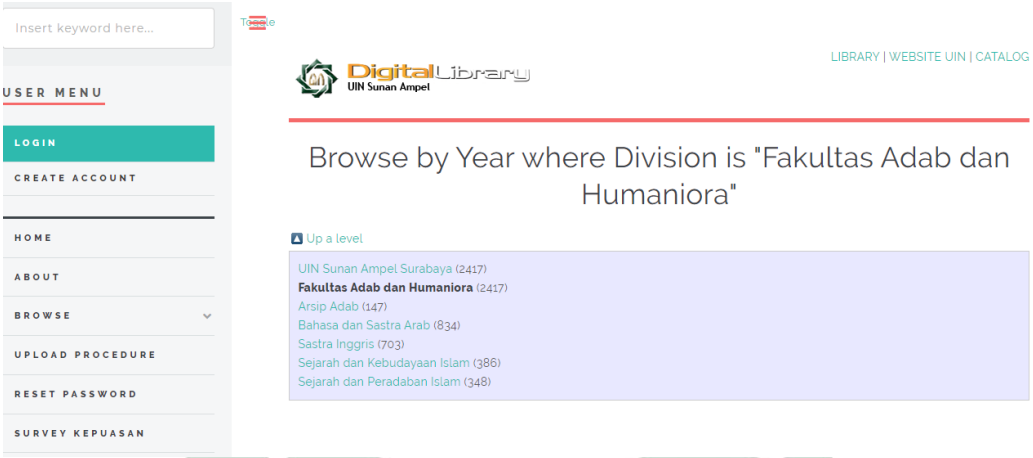
Perincian langkah-langkah pengambilan data *abstract* menggunakan *web scraper* pada penelitian ini dijelaskan sebagai berikut:

Tahap pertama yang dilakukan adalah membuka link data yang diambil. Data yang diambil merupakan *abstract* sastra inggris yang berasal dari digilib uinsa, dengan cara membuka link http://digilib.uinsby.ac.id/. Selanjutnya muncul halaman utama digilib uinsa seperti Gambar 4.3.

Gambar 4.3 Halaman Utama Digilib Uinsa

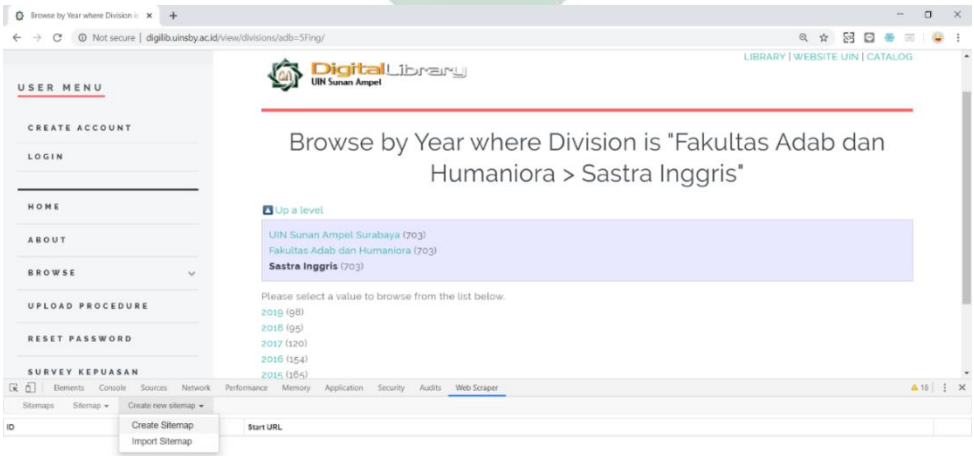
Tahap selanjutnya adalah mencari letak data. Pencarian dilakukan dengan memilih menu *sidebar browse* yang ada pada halaman utama digilib uinsa.

Kemudian memilih *browse by divisions* untuk memilih prodi sastra inggris dan masuk ke dalam halaman *Browse by Year where Division is* "Fakultas Adab dan Humaniora > Sastra Inggris" sebagaimana dapat dilihat pada Gambar 4.4.



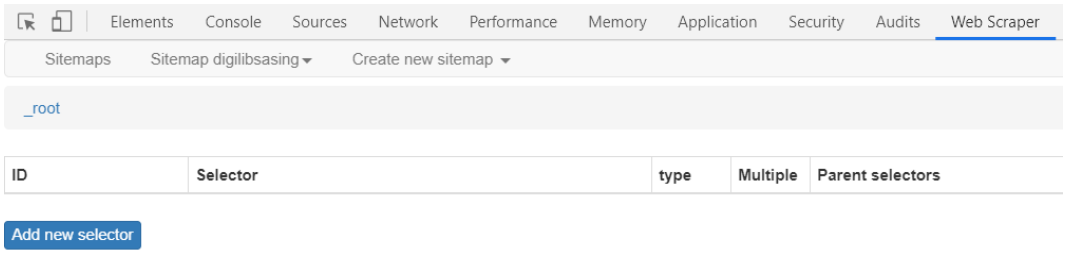
Gambar 4.4 Letak Data yang akan Diambil

Langkah ketiga adalah membuat *sitemap*. Membuat *sitemap* dilakukan dengan mengklik menu *create new sitemap* dan memilih *create sitemap*. Setelah itu mengisi *sitemap name* dan *start* URL. *Sitemap* yang digunakan dalam penelitian ini adalah digilib sasing dan http://digilib.uinsby.ac.id/view/divisions/adb=5Fing/ merupakan URL yang berasal dari mulai bagian pertama data mulai diambil. Kemudian klik tombol *create sitemap* yang ada dibawah start URL sebagaimana Gambar 4.5.



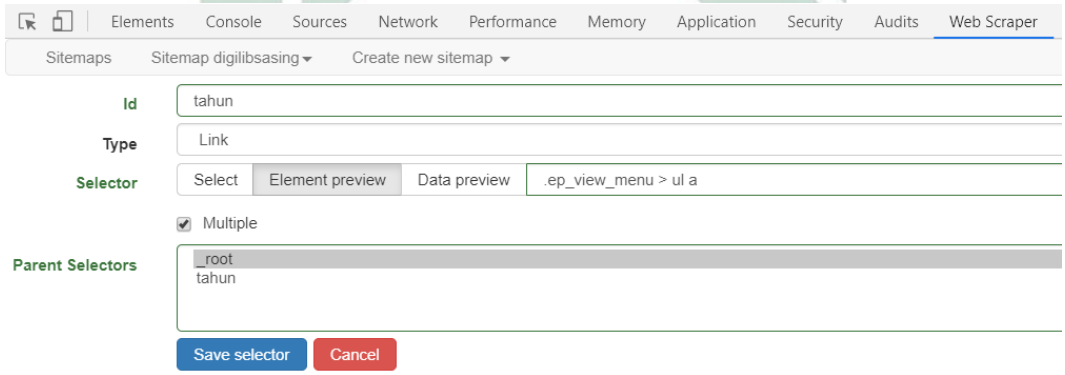
Gambar 4.5 Membuat Sitemap

Langkah selanjutnya adalah menambahkan *selector* baru yang berfungsi untuk memilih data yang akan diambil dengan detail. Proses dilakukan dengan cara tombol *create sitemap* diklik, sehingga muncul tampilan baru dengan tombol *add new selector* seperti Gambar 4.6



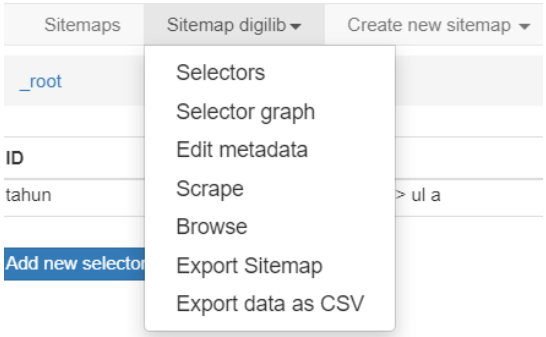
Gambar 4.6 Tambah *Selector* Baru

Detail pengambilan data berupa rincian id, *type* data, selector, dan *parent Selectors*. Kemudian menekan tombol *save selector* seperti Gambar 4.7



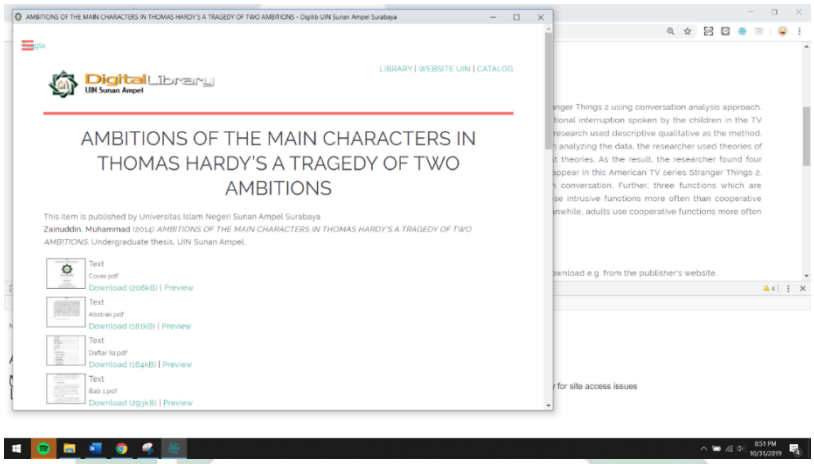
Gambar 4.7 Detail Pengambilan Data

Langkah kelima adalah mengecek pengurutan data. Perlu kita pastikan bahwasanya runtutan pengambilan data diambil secara urut. Runtutan tersebut dapat dilihat pada menu sitemap kemudian pilih *selector graph* untuk memastikan runtutan yang diambil seperti yang ada pada gambar 4.8.



Gambar 4.8 Pengecekan Urutan Data

Setelah memastikan urutan, selanjunya melakukan pengambilan data secara berurut dan otomatis. Langkahnya adalah dengan cara mengklik *scrape* di *menu sitemap* seperti Gambar 4.8. Kemudian klik *start scrapping* untuk memulai pengambilan data secara beruntut dan otomatis seperti pada Gambar 4.9.



Gambar 4.9 Proses Pengambilan Data Secara Urut dan Otomatis

Langkah terakhir yang dilakukan pada tahap pengambilan data adalah menyimpan data yang sudah terambil dalam bentuk csv berupa tampilan seperti pada Gambar 4.10 sebagai berikut.

564	This analysis focuses on the expression of the words "love" and "thought" of the poem "Bond and Free" by Robert Frost. The theory and method of approach of this analysis
565	The basic purpose of this research is to examine the frequent use of grammatical cohesion in the film script along with its functions. This research was used in the script of th
566	In linguistics, cohesion is the use of forms of language to show the semantic relations between the elements in the discourse. Cohesion is how sentences and sentence parts
567	Most of the writers and poets use majas to make literary works more colorful, meaning that it has an indirect meaning. As stated by Abrams in his book "A Glossary of Literar
568	This study examines the politeness strategy in rejection used by child characters in the James W. Ellison novel, Akeelah and The Bee. The data is classified using the Courtesy
569	There are many ways to communicate so that the listener receives an information from the speaker. In formal situations such as discoures, the listener cannot provide a dire
570	In communication, we cannot avoid disagreement when we have different ideas or opinions from our interlocutors. By doing disagreement, we may offend our interlocutors
571	This study is aimed at finding the illocutionary act of the words of the prophet Joseph in the English translation of the verses of Joseph in the Qur'an. The author analyzes the
572	Alay language is a new language variation in Indonesia that is not appropriate with Indonesian grammar. For this case, the Indonesian young people prefer to use Alay langua
573	This thesis tries to analyze a novel from Nicholas Sparks entitled A Walk to Remember. This novel tells about Landon when he was a 17-year-old teenager who reached his m
574	This thesis discusses the phenomenon of persuasive techniques used in Colors magazine. This discussion is divided into two problem formulations. First, the type of persuasi
575	This research focuses on the associative meaning in the song lyrics of Maher Zain. Specifically, the song lyrics used in Maher Zain's song lyrics are the songs titled This Worldly
576	Addressing is the way how someone to call or refer for others which is as the interlocutor or as the person mentioned in their conversation by their names or some terms. It
577	This thesis is conducted to analyze the novel written by Bram Stoker entitled Dracula. This novel tells about Jonathan Harker who struggle to prevent Dracula killing people ir
578	Film is one of the entertainment media that mimics and reflects real life, especially in terms of conversation. This study discusses the implicature of conversations through ve
579	In this study analyzes the moral development of the Sydney Carton at the time of sacrifice to help others in the novel "A Tale of Two Cities" by Charless Dickens. This study c
580	Communication is a part of social life. Without communication, humans cannot live properly. In general, communicating with humans will be helped in establishing relations
581	This research focuses on the characterization and factors that influence Brida to become a witch in Pulo Coelho's Brida. The problem of this research is about the characteriz
582	This paper titled The Shifts in English and Indonesian Noun Phrase: A Case in the translation of Stephani Meyer's novel The Twilight Saga: New Moon and Dua Cinta by Dwi Cl
583	Cohesion is a relation where an element is depend on another element in the next. Cohesion device is divided into two types which all of its types are used to related word,
584	Toni Morrison is an African American writer, Morrison still has a theme African Americans in his works. In this novel, Morrison tells about black people are treated unfairly be

Gambar 4.10 Tampilan Data yang Tersimpan

Jumlah dari keseluruhan total *abstract* yang diambil sebanyak 584 *row* data *abstract*. 585 *row* data yang telah tersimpan tersebut nantinya akan digunakan pada proses selanjutnya, yaitu tahap *pre-processing* data dengan menggunakan Jupytherlab.

Pada tahap pra proses ini, ditemukan beberapa kendala diantaranya:

1. Ada beberapa data *abstract* pada digilib sastra inggris yang menggunakan Bahasa Indonesia,
2. Terdapat format *text abstract* yang seharusnya tidak boleh ada format sehingga *pre-processing* data berjalan kurang baik.

Sehingga pengambilan data *abstract* ini dilakukan secara manual untuk mendapatkan *abstract* berbahasa inggris dengan format yang tidak mengganggu *pre-processing* data.

4.2 *Pre-Procesing* Data

Tahap yang bertujuan untuk mempersiapkan data sebelum dianalisis menggunakan LDA. *Pre-processing* dilakukan dengan menggunakan aplikasi Jupytherlab. Langkah pada proses ini dimulai dengan mengunduh dan mengaktifkan beberapa *library* yang dibutuhkan. Tahap pada *Pre-processing* data dibagi menjadi lima tahap. Tahapan tersebut diantaranya *tokenizing, stopword, lemmatization, stemming,* dan pembuatan document *term matrix* (DTM).

4.2.1 *Tokenizing*

Langkah pertama dari *Pre-processing* data adalah *tokenizing*. Tujuan dari proses ini adalah untuk memisahkan setiap kata ke dalam unit-unit kecil dalam suatu *array* atau *term*. *Tokenizing* memisahkan setiap katanya oleh karakter spasi, sehingga pada proses ini mengandalkan karakter spasi pada dokumen untuk melakukan pemisahan. Pada proses ini bertujuan juga untuk menghilangkan *mention*, url, dan tanda baca yang ada pada teks. *Tokenizing* juga merupah setiap huruf dengan karakter *uppercase* menjadi karakter huruf *lowercase* menggunakan fungsi *lower*. Perbedaan sebelum *tokenizing* dan sesudah *tokenizing* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Tabel Perbedaan Sebelum dan Sesudah *Tokenizing*

Sebelum *Tokenizing*	Sesudah *Tokenizing*
This thesis tries to analyze the drama script from Samuel Beckett entitled Endgame. This drama tells of an isolated Hamm in his own home with a saturating activity with his Clov aides and his parents Nagg and Nell, so they are trying to bring the side of	["", 'this', 'thesis', 'tries', 'to', 'analyze', 'the', 'drama', 'script', 'from', 'samuel', 'beckett', 'entitled', 'endgame', '.', 'this', 'drama', 'tells', 'of', 'an', 'isolated', 'hamm', 'in', 'his', 'own', 'home', 'with', 'a', 'saturating', 'activity', 'with', 'his', 'clov', 'aides', 'and', 'his', 'parents', 'nagg', 'and', 'nell', ',', 'so', 'they', 'are', 'trying', 'to', 'bring', 'the', 'side',

the reality of life to be peeled. This thesis focuses on the analysis of Hamm's consciousness of all past and present events. The purpose of this thesis is to describe the absurdity of life of the main character and to reveal the meaning behind it all in the absurdity drama script.	'of', 'the', 'reality', 'of', 'life', 'to', 'be', 'peeled', '.' , 'this', 'thesis', 'focuses', 'on', 'the', 'analysis', 'o f', 'hamm', "'s", 'consciousness', 'of', 'all', 'past', 'and', 'present', 'events', '.', 'the', 'purpose', 'of', ' this', 'thesis', 'is', 'to', 'describe', 'the', 'absurdity ', 'of', 'life', 'of', 'the', 'main', 'character', 'and', 't o', 'reveal', 'the', 'meaning', 'behind', 'it', 'all', 'in ', 'the', 'absurdity', 'drama', 'script', '.',

4.2.2 *Stopword*

Setelah melewati tahap *tokenizing*, selanjutnya term dokumen diolah pada proses *stopword*. Proses *stopword* dilakukan untuk menghapus kata-kata yang tidak mempunyai informasi atau dengan kata lain hanya mengambil kata yang penting saja. Dalam penelitian ini proses *stopword* terbagi menjadi 3 tahapan. Tahap pertama menggunakan *library* yang sudah tersedia di dalam *python* yaitu nltk.download('stopword'). Kemudian pada *stopword* ditambahkan nltk.corpus.stopword.words('english') di dalam fungsi set() untuk mendownload kata-kata *stopwords* bahasa inggris yang sudah ditetapkan. Penulisan *source code stopword* seperti berikut ini:

```
nltk.download('stopwords')
en_stop = set(nltk.corpus.stopwords.words\('english'))
```

Kata-kata *stopword* yang didownload tersebut berfungsi sebagai acuan dalam penghapusan kata. Apabila jika ada kata yang sama pada sebuah row dokumen tersebut, maka kata tersebut akan dihapus secara otomatis. Acuan kata dari *Stopword* inggris yang didownload berisi sebagai berikut.

```
{'at', 'with', 'no', 'more', 'hadn', 'and', 'be', 'once', 'doing
', "isn't", 'than', "needn't", 'they', "doesn't", "shouldn't", '
he', 'myself', 'ours', 'under', 'into', 'will', "wasn't", 'she',
'aren', 'them', 'so', 'by', 'it', 'itself', 'the', 'down', 'were
', 'then', "didn't", "wouldn't", "it's", "don't", 't', 'couldn',
'shouldn', 'until', 'these', "couldn't", 'don', 'few', 'wasn', '
how', 'does', 'between', 'our', 'had', "you'll", 'shan', 'during
', 'each', 'we', 'should', 'my', 'have', 'as', "mustn't", 'where
', "mightn't", "won't", 'his', "she's", 'both', 'yours', 'not',
"hadn't", 'on', 'same', 'hers', 'why', "should've", 'off', 'when
', 'are', 'you', 'that', 'being', 'your', 'an', 'out', 'weren',
'hasn', 'a', 'has', 'can', "haven't", 're', "shan't", 'y', 'ma',
```

```
'such', "weren't", "that'll", 'any', 'wouldn', 'o', 'll', 'might
n', 'ain', 'whom', 'its', 'who', "you're", 'their', 'was', 'some
', 'm', 'him', 'all', 'won', 'needn', 'again', 'themselves', 'wh
at', 'been', 'before', 'herself', 'below', 'isn', 'nor', 'while'
, 'about', 'mustn', "hasn't", 'through', 'above', 'very', 'but',
'or', 'to', 'd', 'own', 'theirs', "you've", 'against', 'up', 'yo
urself', 'from', 'didn', 'himself', 'having', 'here', 'which', '
those', 'of', 'after', 'in', "aren't", "you'd", 'this', 'for', '
just', 'haven', 'ourselves', 'do', 'most', 'further', 'i', 'am',
'is', 'only', 'doesn', 'there', 'if', 'too', 've', 's', 'now', '
other', 'her', 'did', 'yourselves', 'me', 'because', 'over'}
```

Dari hasil *stopword* tahap pertama terlihat pengurangan kata seperti kata "this", "a", "the", "to"," his" dan lain lain. Perbedaan sebelum dan sesudah proses *stopword* tahap pertama dapat dilihat Tabel 4.2.

Tabel 4.2 Tabel Perbedaan Sebelum dan Sesudah *Stopword* Tahap Pertama

Proses	Sebelum	Sesudah
Stopword	["", 'this', 'thesis', 'tries', 'to', 'analyze', 'the', 'drama', 'script', 'from', 'samuel', 'beckett', 'entitled', 'endgame', '.', 'this', 'drama', 'tells', 'of', 'an', 'isolated', 'hamm', 'in', 'his', 'own', 'home', 'with', 'a', 'saturating', 'activity', 'with', 'his', 'clov', 'aides', 'and', 'his', 'parents', 'nagg', 'and', 'nell', ',', 'so', 'they', 'are', 'trying', 'to', 'bring', 'the', 'side', 'of', 'the', 'reality', 'of', 'life', 'to', 'be', 'peeled', '.', 'this', 'thesis', 'focuses', 'on', 'the', 'analysis', 'of', 'hamm', "'s", 'consciousness', 'of', 'all', 'past', 'and', 'present', 'events', '.', 'the', 'purpose', 'of', 'this', 'thesis', 'is', 'to', 'describe', 'the', 'absurdity', 'of', 'life', 'of', 'the', 'main', 'character', 'and', 'to', 'reveal', 'the', 'meaning', 'behind', 'it', 'all', 'in', 'the', 'absurdity', 'drama', 'script', '.',	["", 'thesis', 'tries', 'analyze', 'drama', 'script', 'samuel', 'beckett', 'entitled', 'endgame', '.', 'drama', 'tells', 'isolated', 'hamm', 'home', 'saturating', 'activity', 'clov', 'aides', 'parents', 'nagg', 'nell', ',', 'trying', 'bring', 'side', 'reality', 'life', 'peeled', '.', 'thesis', 'focuses', 'analysis', 'hamm', "'s", 'consciousness', 'past', 'present', 'events', '.', 'purpose', 'thesis', 'describe', 'absurdity', 'life', 'main', 'character', 'reveal', 'meaning', 'behind', 'absurdity', 'drama', 'script', '.', 'faced', 'focus', 'study', ',', 'study', 'using', 'theory', 'existentialism', 'basic', 'data', 'bring', 'analysis', 'uses', 'psychoanalysis', 'theory', 'illustrate', 'subject', 'condition', 'main', 'character', ',', 'also', 'supported', 'theory', 'society', 'individual', '.', 'latter', ',', 'main', 'point', 'study', '.', 'directed', 'theory', 'personality', 'social', 'crises', 'reveal', 'everyone', 'shortcomings', 'interdependent', 'course', 'always', 'influenced', 'environment', '.', 'furthermore', ',', 'thesis', 'tries', 'reflect', 'points', 'value', 'humanity', 'personal', '’s', 'awareness', 'identity', '.', 'exactly', ',', 'could', 'forming', 'personal', '’s', 'capacity', 'building', 'become', 'better', 'person', 'integrity', '.', 'important', 'value', 'self', '-', 'discovery', 'interpreted', 'needed', 'human', 'effort', 'get', 'understanding', 'meaningfully', 'life', '.', ""]

Berdasarkan tahap pertama *stopwords* yang telah dilakukan, hasilnya masih terdapat kata-kata yang masih bersifat tidak informatif. Kata-kata yang tidak

informatif ini sering kali muncul dalam sebuah *abstract* penelitian. Untuk menghilangkan kata-kata yang tidak informatif, perlu dilakukan *stopword* tahap kedua. Pada Tahap kedua *stopword* dilakukan dengan cara menambahkan kata-kata secara manual sesuai dengan kebutuhan. Sebelumnya perlu ada proses pengecekan kepada pihak sastra inggris. Pengecekan dilakukan untuk memastikan apakah kata-kata yang ditambahkan benar tidak memiliki informasi. Berikut merupakan daftar kata-kata yang ditambahkan dalam penelitian ini:

1. *Researcher*
2. *Analysis*
3. *Thesis*
4. *Study*
5. *Research*
6. *Article*
7. *Analyze*
8. *Meaning*
9. *Language*
10. *Writer*
11. *Analyzes*
12. *Method*

Setelah dilakukan wawancara kepada pihak sastra inggris yang diwakilkan oleh Ketua Prodi Sastra Inggris, terdapat beberapa kata yang ditambahkan. Berikut merupakan daftar kata-kata yang ditambahkan dalam penelitian ini:

1. *Researcher*
2. *Analysis*
3. *Thesis*
4. *Study*
5. *Research*
6. *Article*
7. *Analyze*
8. *Meaning*
9. *Language*
10. *Writer*

11. *Analyzes*
12. *Method*
13. *Found*
14. *Theory*
15. *Result*
16. *Describe*
17. *Function*

Dengan begitu penambahan *source code* dalam *stopwords* pun ditambah seperti pada berikut ini:

```
nltk.download('stopwords')
en_stop = set(nltk.corpus.stopwords.words('english'))
more_stop =
['researcher','analysis','thesis','study','research','article','
analyze','meaning','language','writer','analyzes',’method’,’foun
d’,‘theory’,‘result’,’describe’,’function’]
```

Perbedaan dokumen sebelum dan sesudah nya proses *stopword* tahap kedua ada pada Tabel 4.3

Tabel 4.3 Tabel Perbedaan Sebelum dan Sesudah *Stopword* Tahap Kedua

Proses	Sebelum	Sesudah
Stopwords tambahan	["", 'thesis', 'tries', 'analyze', 'drama', 'script', 'samuel', 'beckett', 'entitled', ' endgame', '.', 'drama', 'tells', 'isolated ', 'hamm', 'home', 'saturating', 'activit y', 'clov', 'aides', 'parents', 'nagg', 'nel l', ',', 'trying', 'bring', 'side', 'reality', 'l ife', 'peeled', '.', 'thesis', 'focuses', 'an alysis', 'hamm', "'s", 'consciousness', 'past', 'present', 'events', '.', 'purpose', 'thesis', 'describe', 'absurdity', 'life', ' main', 'character', 'reveal', 'meaning', 'behind', 'absurdity', 'drama', 'script', '.', 'faced', 'focus', 'study', ',', 'study', ' using', 'theory', 'existentialism', 'basi c', 'data', 'bring', 'analysis', 'uses', 'ps ychoanalysis', 'theory', 'illustrate', 's ubject', 'condition', 'main', 'character' , ',', 'also', 'supported', 'theory', 'socie ty', 'individual', '.', 'latter', ',', 'main', ' point', 'study', '.', 'directed', 'theory', '	["", 'tries', 'drama', 'script', 'samuel' , 'beckett', 'entitled', 'endgame', '.', ' drama', 'tells', 'isolated', 'hamm', 'h ome', 'saturating', 'activity', 'clov', ' aides', 'parents', 'nagg', 'nell', ',', 'try ing', 'bring', 'side', 'reality', 'life', 'pe eled', '.', 'focuses', 'hamm', "'s", 'con sciousness', 'past', 'present', 'events ', '.', 'purpose', 'absurdity', 'life', 'ma in', 'reveal', 'behind', 'absurdity', 'dr ama', 'script', '.', 'faced', 'focus', ',', 'u sing', 'existentialism', 'basic', 'data', 'bring', 'uses', 'psychoanalysis', 'illu strate', 'subject', 'condition', 'main', ',', 'also', 'supported', 'society', 'indi vidual', '.', 'latter', ',', 'main', 'point', '.', 'directed', 'personality', 'social', ' crises', 'reveal', 'everyone', 'shortco mings', 'interdependent', 'course', '

	personality', 'social', 'crises', 'reveal', 'everyone', 'shortcomings', 'interdependent', 'course', 'always', 'influenced', 'environment', '.', 'furthermore', ',', 'thesis', 'tries', 'reflect', 'points', 'value', 'humanity', 'personal', ''s', 'awareness', 'identity', '.', 'exactly', ',', 'could', 'forming', 'personal', ''s', 'capacity', 'building', 'become', 'better', 'person', 'integrity', '.', 'important', 'value', 'self', '-', 'discovery', 'interpreted', 'needed', 'human', 'effort', 'get', 'understanding', 'meaningfully', 'life', '.', ""]	always', 'influenced', 'environment', '.', 'furthermore', ',', 'tries', 'reflect', 'points', 'value', 'humanity', 'personal', ''s', 'awareness', 'identity', '.', 'exactly', ',', 'could', 'forming', 'personal', ''s', 'capacity', 'building', 'become', 'better', 'person', 'integrity', '.', 'important', 'value', 'self', '-', 'discovery', 'interpreted', 'needed', 'human', 'effort', 'get', 'understanding', 'meaningfully', 'life', '.', ""]

Hasil yang didapatkan berdasarkan tahap kedua *stopword* masih terdapat tanda baca yang terpisahkan dari proses *tokenize*. Tanda baca tersebut perlu dihilangkan. Untuk menghilangkan tanda baca, perlu dilakukan tahap ketiga *stopword*. Tahap ketiga *stopword* ini dengan cara menambahkan fungsi len(). Fungsi tersebut digunakan untuk mengembalikan panjang (jumlah anggota) dari suatu objek. Hasil dari tahap ketiga *stopword* ini dapat dilihat pada Tabel 4.4.

Tabel 4.4 Tabel Perbedaan Sebelum dan Sesudah Len()

Proses	Sebelum	Sesudah
Len()	['"', 'tries', 'drama', 'script', 'samuel', 'beckett', 'entitled', 'endgame', '.', 'drama', 'tells', 'isolated', 'hamm', 'home', 'saturating', 'activity', 'clov', 'aides', 'parents', 'nagg', 'nell', ',', 'trying', 'bring', 'side', 'reality', 'life', 'peeled', '.', 'focuses', 'hamm', "'s", 'consciousness', 'past', 'present', 'events', '.', 'purpose', 'absurdity', 'life', 'main', 'reveal', 'behind', 'absurdity', 'drama', 'script', '.', 'faced', 'focus', ',', 'using', 'existentialism', 'basic', 'data', 'bring', 'uses', 'psychoanalysis', 'illustrate', 'subject', 'condition', 'main', ',', 'also', 'supported', 'society', 'individual', '.', 'latter', ',', 'main', 'point', '.', 'directed', 'personality', 'social', 'crises', 'reveal', 'everyone', 'shortcomings', 'interdependent', 'course', 'always', 'influenced', 'environment', '.', 'furthermore', ',', 'tries', 'reflect', 'points', 'value', 'humanity', 'personal', ''s', 'awareness', 'identity', '.', 'exactly', ',', 'could', 'forming', 'personal', ''s', 'capacity', 'building', 'become', 'better', 'person', 'int	['tries', 'drama', 'script', 'samuel', 'beckett', 'entitled', 'endgame', 'drama', 'tells', 'isolated', 'saturating', 'activity', 'aides', 'parents', 'trying', 'bring', 'reality', 'peeled', 'focuses', 'consciousness', 'present', 'events', 'purpose', 'absurdity', 'reveal', 'behind', 'absurdity', 'drama', 'script', 'faced', 'focus', 'using', 'existentialism', 'basic', 'bring', 'psychoanalysis', 'illustrate', 'subject', 'condition', 'supported', 'society', 'individual', 'latter', 'point', 'directed', 'personality', 'social', 'crises', 'reveal', 'everyone', 'shortcomings', 'interdependent', 'course', 'always', 'influenced', 'environment', 'furthermore', 'tries', 'reflect', 'points', 'value', 'humanity', 'personal', 'awareness', 'identity', 'exactly', 'could', 'forming', 'personal', 'capacity', 'building', 'become', 'better', 'person', 'integrity', 'important', 'value', 'discovery', 'interpreted', 'needed', 'human', 'effort', 'understanding', 'meaningfully']

	egrity', '.', 'important', 'value', 'self', '-', 'discovery', 'interpreted', 'needed', 'human', 'effort', 'get', 'understanding', 'meaningfully', 'life', '.', ""]	

4.2.3 *Lemmatization* & *Stemming*

Hasil dari *stopwords* dilanjut ke proses *Pre-processing* teks mencakup proses *stemming* dan *lemmatization*. *Lemmatization* adalah sebuah proses pengelompokan kata yang berbeda dengan melalui tahap analisis sebagai satu kata yang sama. Berbeda dengan *stemming*, *stemming* merupakan sebuah proses menemukan sebuah kata dasar dari kata yang mempunyai awalan dan akhiran tanpa menganalisis apakah kata itu mempunyai arti yang sama dengan kata yang lain.

Oleh karena itu dalam proses penelitian ini digabungkan *lemmatization* dan *stemming* untuk mendapatkan hasil yang lebih baik dari pada menggunakan hanya dengan satu metode saja. Berikut merupakan *source code* untuk mengimplementasikan *lemmatization* dan *stemming*.

```
import nltk
nltk.download('wordnet')
from nltk.corpus import wordnet as wn
def get_lemma(word):
    lemma = wn.morphy(word)
    if lemma is None:
        return word
    else:
        return lemma

from nltk.stem.wordnet import WordNetLemmatizer
def get_lemma2(word):
    return WordNetLemmatizer().lemmatize(word)
```

Hasil yang didapatkan berdasarkan proses *lemmatization dan stemming* seperti pada Tabel 4.5.

Tabel 4.5 Perbandingan Sebelum dan Sesudah *Lemmatization* dan *Stemming*

Sebelum *Lemmatization* dan *stemming*	Sesudah *Lemmatization* dan *stemming*
['tries', 'drama', 'script', 'samuel', 'beckett', 'entitled', 'endgame', 'drama', 'tells', 'isolated', 'saturating', 'activity', 'aides', 'parents', 't	['try', 'drama', 'script', 'samuel', 'beckett', 'entitle', 'endgame', 'drama', 'tell', 'isolate', 'saturate', 'activity', 'aides', 'parent', 'try', 'brin

rying', 'bring', 'reality', 'peeled', 'focuses', 'c onsciousness', 'present', 'events', 'purpose', 'absurdity', 'reveal', 'behind', 'absurdity', 'dr ama', 'script', 'faced', 'focus', 'using', 'existen tialism', 'basic', 'bring', 'psychoanalysis', 'ill ustrate', 'subject', 'condition', 'supported', 's ociety', 'individual', 'latter', 'point', 'directed ', 'personality', 'social', 'crises', 'reveal', 'eve ryone', 'shortcomings', 'interdependent', 'co urse', 'always', 'influenced', 'environment', ' furthermore', 'tries', 'reflect', 'points', 'value ', 'humanity', 'personal', 'awareness', 'identi ty', 'exactly', 'could', 'forming', 'personal', 'c apacity', 'building', 'become', 'better', 'perso n', 'integrity', 'important', 'value', 'discovery ', 'interpreted', 'needed', 'human', 'effort', 'u nderstanding', 'meaningfully']	g', 'reality', 'peel', 'focus', 'consciousness', 'p resent', 'event', 'purpose', 'absurdity', 'revea l', 'behind', 'absurdity', 'drama', 'script', 'fac e', 'focus', 'using', 'existentialism', 'basic', 'br ing', 'psychoanalysis', 'illustrate', 'subject', ' condition', 'support', 'society', 'individual', 'l atter', 'point', 'direct', 'personality', 'social', ' crisis', 'reveal', 'everyone', 'shortcoming', 'in terdependent', 'course', 'always', 'influence', 'environment', 'furthermore', 'try', 'reflect', ' point', 'value', 'humanity', 'personal', 'aware ness', 'identity', 'exactly', 'could', 'form', 'per sonal', 'capacity', 'building', 'become', 'bette r', 'person', 'integrity', 'important', 'value', ' discovery', 'interpret', 'need', 'human', 'effor t', 'understanding', 'meaningfully']

4.3 *Bag of Words*

Hasil dari pre-processing data berupa term atau matriks yang berisi kata-kata yang muncul berulang-ulang. Model *Bag of words* digunakan menghitung jumlah kemunculan setiap kata pada term atau matriks kata-kata tersebut. Hasil perhitungan jumlah setiap kata tersebut digunakan dalam dalam perhitungan distribusi yang ada pada LDA. Berikut merupakan source code untuk menghitung jumlah setiap kata menggunakan model *bag of word*.

```
from gensim import corpora
dictionary = corpora.Dictionary(text_data)
print(dictionary)
corpus = [dictionary.doc2bow(text) for text in text_data]
print(corpus)
import pickle
pickle.dump(corpus, open('corpus.pkl', 'wb'))
dictionary.save('dictionary.gensim')
```

Langkah pertama pada proses ini yaitu menyimpan kata yang berbeda atau unique yang ada dalam term atau matriks kata-kata. Dalam proses tersebut dihasilkan 5993 unique kata yang berbeda.

```
Dictionary(5993 unique tokens: ['absurdity', 'activity', 'aides', 'always', 'awa
reness']...)
```

Langkah kedua dalam proses ini yaitu menghitung kata yang muncul. Dalam menghitung kata tersebut, *Bag of Words* mengindekskan setiap kata yang ada beserta menghitung kemunculan kata. *Bag of words* menghitung berdasarkan

unique kata yang telah ditentukan pada proses sebelumnya. Hasil dari proses *bag of words* ditunjukan pada Gambar 4.11.

Gambar 4.11 Hasil Proses pada *Bag of Words*

4.4 Pemodelan Topik Menggunakan LDA

Setelah serangkaian *pre-processing* yang telah dilakukan dan dimasukkan ke dalam *bag of words* tahap selanjunya adalah memodelkan topik menggunakan LDA. Sebelumnya pada tahap *bag of words* telah muncul token yang berasal dari banyaknya kata yang muncul dalam suatu dokumen. Token berfungsi sebagai ukuran dalam LDA agar dapat dimodelkan. Dalam pemodelan LDA perlu diadakan beberapa kali percobaan, supaya dapat menentukan jumlah topik yang sesuai. Percobaan yang dilakukan dengan mengubah jumlah topik dan percobaan dalam jumlah iterasi. Percobaan dilakukan dengan mengisi NUM_TOPIK untuk jumlah topik yang dicoba. Sedangkan untuk iterasi menggunakan *code passes*, untuk selanjutnya diisi dengan berapa kali iterasi akan dilakukan.

Percobaan pertama yang dilakukan adalah percobaan dengan jumlah topik 2 pada iterasi ke-100 dengan *code* seperti dibawah ini.

```
import genism
NUM_TOPICS = 2
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=100)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 2 dan iterasi 100, diperoleh hasi topik seperti berikut:

Topik 1: *novel, strategy, character, maxim, style, focus, analyze, politeness, qualitative, utterance.*

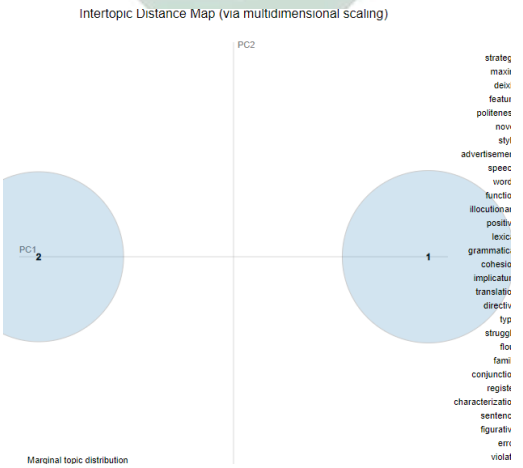
Topik 2: *type, speech, words, qualitative, deixis, movie, feature, descriptive, function, analyse.*

Bobot setiap kata yang dihasilkan pada percobaan yang telah dilakukan seperti pada uraian dibawah. Dimana letak kata yang paling atas menandakan kata tersebut muncul berulang-ulang kali. Dan setiap topiknya dianggap memiliki kesamaan rumpun. Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.6.

Tabel 4.6 List Kata dan Bobot dengan Jumlah Topik 2 pada Iterasi ke-100

Topik 1	Topik 2
0.016*"novel"	0.011*"type"
0.012*"strategy"	0.009*"speech"
0.008*"character"	0.008*"words"
0.008*"maxim"	0.007*"qualitative"
0.007*"style"	0.007*"deixis"
0.007*"focus"	0.007*"movie"
0.006*"analyze"	0.007*"feature"
0.006*"politeness"	0.006*"descriptive"
0.006*"qualitative"	0.006*"function"
0.006*"utterance"	0.006*"analyze"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 2 topik dan 100 iterasi seperti Gambar 4.12 dibawah ini.



Gambar 4.12 Visualisasi LDA 2 Topik dan 100 Iterasi

Percobaan kedua yang dilakukan adalah percobaan dengan jumlah topik 2 pada iterasi ke-500 dengan *code* seperti dibawah ini.

```
import genism
NUM_TOPICS = 2
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=500)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 2 dan iterasi 100, diperoleh hasi topik seperti berikut:

Topik 1: *strategy, type, speech, movie, words, style, character, deixis, qualitative, feature.*

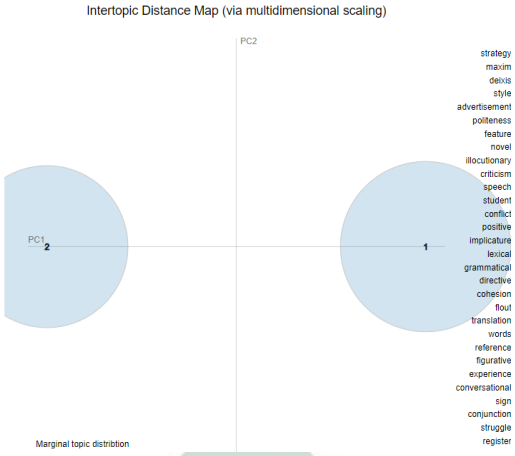
Topik 2: *novel, maxim, focus, qualitative, analyze, descriptive, advertisement, utterance, type, problem.*

Bobot setiap kata yang dihasilkan pada percobaan yang telah dilakukan seperti pada uraian dibawah. Dimana letak kata yang paling atas menandakan kata tersebut muncul berulang-ulang kali. Dan setiap topiknya dianggap memiliki kesamaan rumpun. Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.7.

Tabel 4.7 List Kata dan Bobot dengan Jumlah Topik 2 pada Iterasi ke-500

Topik 1	Topik 2
0.012*"strategy"	0.015*"novel"
0.009*"type"	0.008*"maxim"
0.008*"speech"	0.007*"focus"
0.008*"movie"	0.007*"qualitative"
0.007*"words"	0.006*"analyze"
0.007*"style"	0.006*"descriptive"
0.007*"character"	0.006*"advertisement"
0.007*"deixis"	0.005*"utterance"
0.007*"qualitative"	0.005*"type"
0.006*"feature"	0.005*"problem"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 2 topik dan 500 iterasi seperti Gambar 4.13 dibawah ini.



Gambar 4.13 Visualisasi LDA 2 Topik dan 500 Iterasi

Percobaan ketiga yang dilakukan adalah percobaan dengan jumlah topik 2 pada iterasi ke-1000 dengan *code* seperti dibawah ini.

```
import genism
NUM_TOPICS = 2
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=1000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 2 dan iterasi 1000, diperoleh hasi topik seperti berikut:

Topik 1: *novel, focus, story, qualitative, criticism, analyze, woman, characterization, descriptive, problem*.

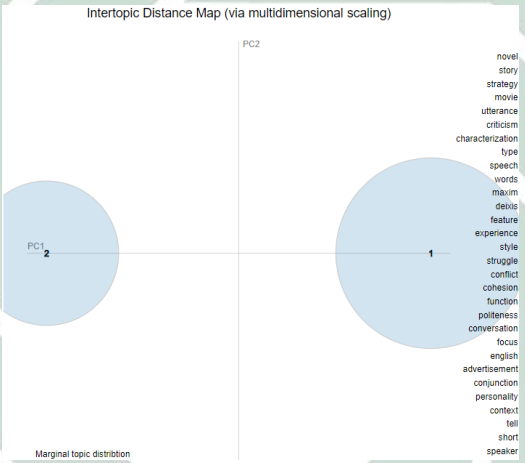
Topik 2: *type, strategy, movie, utterance, speech, words, qualitative, character, descriptive, analyze*.

Bobot setiap kata yang dihasilkan pada percobaan yang telah dilakukan seperti pada uraian dibawah. Dimana letak kata yang paling atas menandakan kata tersebut muncul berulang-ulang kali. Dan setiap topiknya dianggap memiliki kesamaan rumpun. Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.8.

Tabel 4.8 List Kata dan Bobot dengan Jumlah Topik 2 pada Iterasi ke-1000

Topik 1	Topik 2
0.025*"novel"	0.011*"type"
0.009*"focus"	0.010*"strategy"
0.009*"story"	0.010*"movie"
0.006*"qualitative"	0.009*"utterance"
0.006*"criticism"	0.008*"speech"
0.006*"analyze"	0.007*"words"
0.006*"woman"	0.007*"qualitative"
0.006*"characterization"	0.007*"character"
0.006*"descriptive"	0.006*"descriptive"
0.005*"problem"	0.006*"analyze"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 2 topik dan 1000 iterasi seperti Gambar 4.14 dibawah ini.



Gambar 4.14 Visualisasi LDA 2 Topik dan 1000 Iterasi

Percobaan keempat yang dilakukan adalah percobaan dengan jumlah topik 2 pada iterasi ke-5000 dengan *code* seperti dibawah ini.

```
import genism
NUM_TOPICS = 2
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=5000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 2 dan iterasi 5000, diperoleh hasi topik seperti berikut:

topik 1: *novel, maxim, focus, story, qualitative, criticism, analyze, characterization, descriptive, problem*.

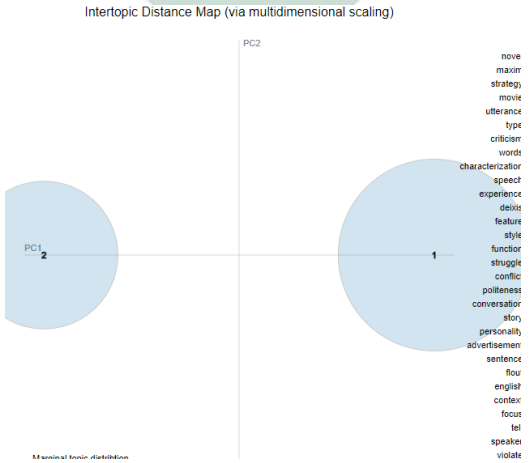
topik 2: *type, strategy, movie, utterance, speech, words, qualitative, character, analyze, style*.

Bobot setiap kata yang dihasilkan pada percobaan yang telah dilakukan seperti pada uraian dibawah. Dimana letak kata yang paling atas menandakan kata tersebut muncul berulang-ulang kali. Dan setiap topiknya dianggap memiliki kesamaan rumpun. Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.9.

Tabel 4.9 List Kata dan Bobot dengan Jumlah Topik 2 pada Iterasi ke-5000

Topik 1	Topik 2
0.022*"novel"	0.012*"type"
0.010*"maxim"	0.010*"strategy"
0.009*"focus"	0.010*"movie"
0.007*"story"	0.009*"utterance"
0.006*"qualitative"	0.008*"speech"
0.006*"criticism"	0.007*"words"
0.006*"analyze"	0.007*"qualitative"
0.006*"characterization"	0.006*"character"
0.006*"descriptive"	0.006*"analyze"
0.006*"problem"	0.006*"style"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 2 topik dan 5000 iterasi seperti Gambar 4.15 dibawah ini.



Gambar 4.15 Visualisasi LDA 2 Topik dan 5000 Iterasi

Selanjutnya dilakukan percobaan dengan menambah jumlah topik sebelumnya. percobaan pertama yang dilakukan adalah percobaan dengan jumlah topik 3 pada iterasi ke-100 dengan *code* seperti dibawah ini.

```
import genism
NUM_TOPICS = 3
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=100)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 3 dan iterasi 100, diperoleh hasi topik seperti berikut:

Topik 1: *type, speech, movie, utterance, maxim, words, deixis, qualitative, function, descriptive*.

Topik 2: *novel, focus, story, criticism, characterization, analyze, problem, character, woman, qualitative*.

Topik 3: *strategy, style, politeness, advertisement, english, student, positive, factor, feature, qualitative.*

Bobot setiap kata yang dihasilkan pada percobaan yang telah dilakukan seperti pada uraian dibawah. Dimana letak kata yang paling kiri menandakan kata tersebut muncul berulang-ulang kali. Dan setiap topiknya dianggap memiliki kesamaan rumpun. Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.10.

Tabel 4.10 List Kata dan Bobot dengan Jumlah Topik 3 pada Iterasi ke-100

Topik 1	Topik 2	Topik 3
0.015*"type"	0.027*"novel"	0.020*"strategy"
0.013*"speech"	0.010*"focus"	0.013*"style"
0.012*"movie"	0.007*"story"	0.010*"politeness"
0.012*"utterance"	0.007*"criticism"	0.009*"advertisement"
0.010*"maxim"	0.007*"characterization"	0.009*"english"
0.010*"words"	0.006*"analyze"	0.008*"student"
0.009*"deixis"	0.006*"problem"	0.007*"positive"
0.008*"qualitative"	0.006*"character"	0.006*"factor"
0.008*"function"	0.006*"woman"	0.006*"feature"
0.007*"descriptive"	0.006*"qualitative"	0.006*"qualitative"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 3 topik dan 100 iterasi seperti Gambar 4.16 dibawah ini.

Gambar 4.16 Visualisasi LDA 3 Topik dan 100 Iterasi

Percobaan kedua dilakukan dengan jumlah topik 3, namun berbeda iteriasi yang dilakukan sebanyak 500 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 3
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=500)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 3 dan iterasi 500, diperoleh hasi topik seperti berikut:

Topik 1: *movie, type, maxim, speech, words, utterance, feature, qualitative, english, descriptive.*

Topik 2: *strategy, style, deixis, type, politeness, utterance, advertisement, character, qualitative, positive.*

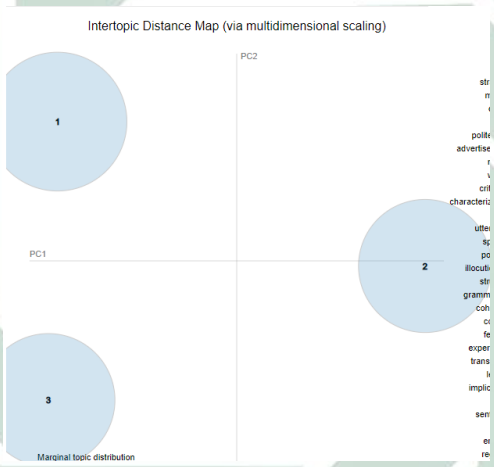
Topik 3: *novel, focus, story, criticism, characterization, qualitative, woman, analyze, problem, descriptive.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.11.

Tabel 4.11 List Kata dan Bobot dengan Jumlah Topik 3 pada Iterasi ke-500

Topik 1	Topik 2	Topik 3
0.012*"movie"	0.017*"strategy"	0.026*"novel"
0.012*"type"	0.010*"style"	0.010*"focus"
0.012*"maxim"	0.010*"deixis"	0.007*"story"
0.011*"speech"	0.009*"type"	0.007*"criticism"
0.011*"words"	0.009*"politeness"	0.007*"characterization"
0.008*"utterance"	0.008*"utterance"	0.007*"qualitative"
0.007*"feature"	0.008*"advertisement"	0.006*"woman"
0.007*"qualitative"	0.007*"character"	0.006*"analyze"
0.007*"english"	0.007*"qualitative"	0.006*"problem"
0.006*"descriptive"	0.006*"positive"	0.006*"descriptive"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 3 topik dan 500 iterasi seperti Gambar 4.17 dibawah ini.



Gambar 4.17 Visualisasi LDA 3 Topik dan 500 Iterasi

Percobaan ketiga dilakukan dengan jumlah topik 3, namun berbeda iteriasi yang dilakukan sebanyak 1000 kali. *Code* yang digunakan sebagai berikut:

```
import gensim
NUM_TOPICS = 3
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=1000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 3 dan iterasi 1000, diperoleh hasi topik seperti berikut:

Topik 1: *type, movie, utterance, speech, style, maxim, deixis, character, qualitative, advertisement.*

Topik 2: *english, words, student, process, qualitative, using, translation, feature, analyze, people.*

Topik 3: *novel, focus, politeness, strategy, analyze, criticism, qualitative, characterization, problem, descriptive.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.12.

Tabel 4.12 List Kata dan Bobot dengan Jumlah Topik 3 pada Iterasi ke-1000.

Topik 1	Topik 2	Topik 3
0.015*"type"	0.013*"english"	0.026*"novel"
0.013*"movie"	0.009*"words"	0.009*"focus"
0.013*"utterance"	0.009*"student"	0.009*"politeness"
0.012*"speech"	0.007*"process"	0.008*"strategy"
0.010*"style"	0.006*"qualitative"	0.007*"analyze"
0.009*"maxim"	0.006*"using"	0.007*"criticism"
0.009*"deixis"	0.006*"translation"	0.006*"qualitative"
0.008*"character"	0.006*"feature"	0.006*"characterization"
0.008*"qualitative"	0.005*"analyze"	0.006*"problem"
0.007*"advertisement"	0.005*"people"	0.006*"descriptive"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 3 topik dan 1000 iterasi seperti Gambar 4.18 dibawah ini.

Gambar 4.18 Visualisasi LDA 3 Topik dan 1000 Iterasi

Percobaan keempat dilakukan dengan jumlah topik 3, namun berbeda iteriasi yang dilakukan sebanyak 5000 kali. *Code* yang digunakan sebagai berikut:

```
import gensim
NUM_TOPICS = 3
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=5000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 3 dan iterasi 5000, diperoleh hasi topik seperti berikut:

topik 1: *novel, focus, story, criticism, characterization, analyze, woman, qualitative, problem, descriptive.*

topik 2: *type, movie, words, style, deixis, english, speech, qualitative, descriptive, utterance.*

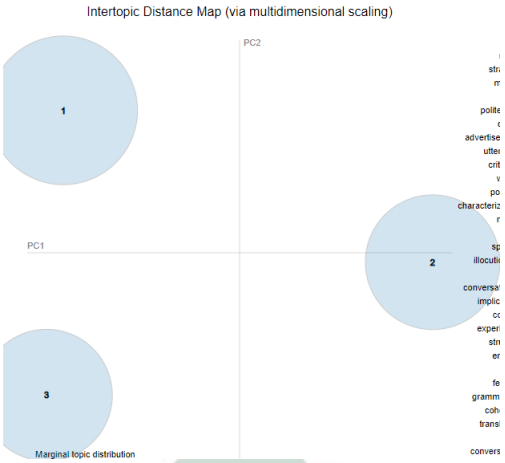
topik 3: *strategy, maxim, utterance, politeness, advertisement, speech, type, character, positive, movie.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.13.

Tabel 4.13 List Kata dan Bobot dengan Jumlah Topik 3 pada Iterasi ke-5000

Topik 1	Topik 2	Topik 3
0.028*"novel"	0.012*"type"	0.020*"strategy"
0.010*"focus"	0.010*"movie"	0.012*"maxim"
0.008*"story"	0.010*"words"	0.011*"utterance"
0.007*"criticism"	0.010*"style"	0.010*"politeness"
0.007*"characterization"	0.009*"deixis"	0.009*"advertisement"
0.006*"analyze"	0.008*"english"	0.008*"speech"
0.006*"woman"	0.007*"speech"	0.008*"type"
0.006*"qualitative"	0.007*"qualitative"	0.007*"character"
0.006*"problem"	0.006*"descriptive"	0.007*"positive"
0.006*"descriptive"	0.006*"utterance"	0.007*"movie"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 3 topik dan 5000 iterasi seperti Gambar 4.19 dibawah ini.



Gambar 4.19 Visualisasi LDA 3 Topik dan 5000 Iterasi

Setelah dua percobaan yang telah dilakukan yakni pada jumlah topik 2 dan 3, kemudian dilakukan percobaan pertama dengan jumlah topik 4. Percobaan dengan jumlah topik 4 diawali degan iteriasi sebanyak 100 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 4
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=100)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 4 dan iterasi 100, diperoleh hasi topik seperti berikut:

topik 1: *style, speech, advertisement, type, qualitative, descriptive, novel, movie, cohesion, analyze.*

topik 2: *utterance, maxim, type, movie, feature, function, conversation, illocutionary, character, analyze.*

topik 3: *novel, deixis, focus, person, problem, story, woman, qualitative, criticism, character.*

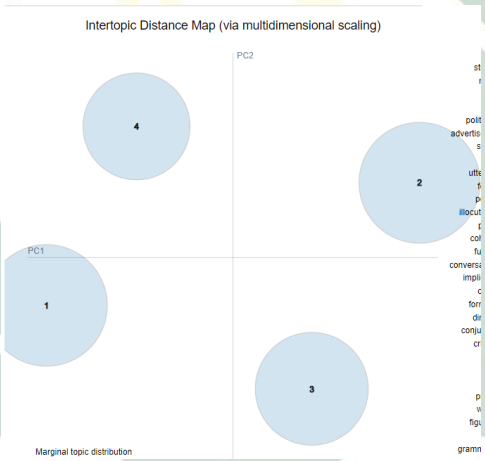
topik 4: *strategy, novel, politeness, words, positive, qualitative, process, analyze, descriptive, formation.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.14.

Tabel 4.14 List Kata dan Bobot dengan Jumlah Topik 4 pada Iterasi ke-100

Topik 1	Topik 2	Topik 3	Topik 4
0.015*"style"	0.016*"utterance"	0.017*"novel"	0.022*"strategy"
0.015*"speech"	0.014*"maxim"	0.013*"deixis"	0.015*"novel"
0.012*"advertisement"	0.013*"type"	0.011*"focus"	0.014*"politeness"
0.010*"type"	0.013*"movie"	0.009*"person"	0.009*"words"
0.007*"qualitative"	0.010*"feature"	0.007*"problem"	0.009*"positive"
0.007*"descriptive"	0.009*"function"	0.007*"story"	0.007*"qualitative"
0.007*"novel"	0.007*"conversation"	0.007*"woman"	0.007*"process"
0.006*"movie"	0.007*"illocutionary"	0.007*"qualitative"	0.006*"analyze"
0.006*"cohesion"	0.007*"character"	0.007*"criticism"	0.006*"descriptive"
0.006*"analyze"	0.007*"analyze"	0.006*"character"	0.006*"formation"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 4 topik dan 100 iterasi seperti Gambar 4.20 dibawah ini.



Gambar 4.20 Visualisasi LDA 4 Topik dan 100 Iterasi

Percobaan kedua dilakukan dengan jumlah topik 4, namun berbeda iteriasi yang dilakukan sebanyak 500 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 4
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=500)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 4 dan iterasi 500 diperoleh hasi topik seperti berikut:

Topik 1: *utterance, type, style, movie, maxim, deixis, function, speech, character, qualitative.*

Topik 2: *strategy, politeness, character, positive, speech, qualitative, advertisement, factor, influence, using.*

Topik 3: *novel, story, focus, criticism, characterization, analyze, qualitative, woman, descriptive, problem.*

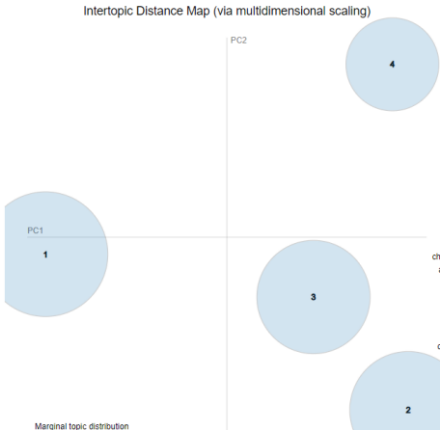
Topik 4: *feature, english, student, translation, people, error, lexical, woman, words, shift.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.15.

Tabel 4.15 List Kata dan Bobot dengan Jumlah Topik 4 pada Iterasi ke-500

Topik 1	Topik 2	Topik 3	Topik 4
0.016*"utterance"	0.024*"strategy"	0.026*"novel"	0.015*"feature"
0.015*"type"	0.012*"politeness"	0.010*"story"	0.014*"english"
0.014*"style"	0.008*"character"	0.010*"focus"	0.011*"student"
0.014*"movie"	0.008*"positive"	0.007*"criticism"	0.006*"translation"
0.014*"maxim"	0.008*"speech"	0.007*"characterization"	0.006*"people"
0.013*"deixis"	0.008*"qualitative"	0.007*"analyze"	0.006*"error"
0.010*"function"	0.007*"advertisement"	0.007*"qualitative"	0.006*"lexical"
0.010*"speech"	0.007*"factor"	0.006*"woman"	0.006*"woman"
0.009*"character"	0.007*"influence"	0.006*"descriptive"	0.006*"words"
0.008*"qualitative"	0.007*"using"	0.006*"problem"	0.006*"shift"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 4 topik dan 500 iterasi seperti Gambar 4.21 dibawah ini.



Gambar 4.21 Visualisasi LDA 4 Topik dan 500 Iterasi

Percobaan ketiga dilakukan dengan jumlah topik 4, namun berbeda iteriasi yang dilakukan sebanyak 1000 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 4
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=1000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 4 dan iterasi 1000 diperoleh hasi topik seperti berikut:

Topik 1: *movie, strategy, utterance, speech, character, type, maxim, feature, politeness, conversation.*

Topik 2: *novel, deixis, person, focus, story, type, qualitative, analyze, problem, cohesion.*

Topik 3: *style, novel, conflict, implicature, focus, shift, qualitative, descriptive, student, problem.*

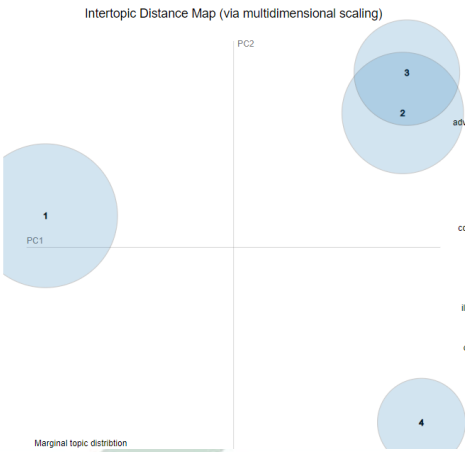
Topik 4: *advertisement, novel, process, words, descriptive, qualitative, analyze, focus, type, formation.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.16.

Tabel 4.16 List Kata dan Bobot dengan Jumlah Topik 4 pada Iterasi ke-1000

Topik 1	Topik 2	Topik 3	Topik 4
0.016*"movie"	0.018*"novel"	0.018*"style"	0.013*"advertisement"
0.016*"strategy"	0.013*"deixis"	0.010*"novel"	0.010*"novel"
0.015*"utterance"	0.008*"person"	0.010*"conflict"	0.010*"process"
0.014*"speech"	0.008*"focus"	0.010*"implicature"	0.009*"words"
0.011*"character"	0.007*"story"	0.007*"focus"	0.007*"descriptive"
0.011*"type"	0.007*"type"	0.007*"shift"	0.007*"qualitative"
0.010*"maxim"	0.007*"qualitative"	0.006*"qualitative"	0.007*"analyze"
0.009*"feature"	0.006*"analyze"	0.006*"descriptive"	0.006*"focus"
0.008*"politeness"	0.006*"problem"	0.005*"student"	0.006*"type"
0.008*"conversation"	0.005*"cohesion"	0.005*"problem"	0.006*"formation"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 4 topik dan 1000 iterasi seperti Gambar 4.22 dibawah ini.



Gambar 4.22 Visualisasi LDA 4 Topik dan 1000 Iterasi

Percobaan keempat dilakukan dengan jumlah topik 4, namun berbeda iteriasi yang dilakukan sebanyak 5000 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 4
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=5000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 4 dan iterasi 5000 diperoleh hasi topik seperti berikut:

Topik 1: *movie, style, maxim, utterance, type, speech, character, illocutionary, qualitative, base.*

Topik 2: *novel, focus, story, criticism, characterization, analyze, problem, qualitative, descriptive, experience.*

Topik 3: *words, advertisement, type, qualitative, sentence, cohesion, analyze, function, descriptive, grammatical.*

Topik 4: *strategy, deixis, feature, politeness, positive, utterance, english, type, woman, movie.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.17.

Tabel 4.17 List Kata dan Bobot dengan Jumlah Topik 4 pada Iterasi ke-5000

Topik 1	Topik 2	Topik 3	Topik 4
0.019*"movie"	0.032*"novel"	0.013*"words"	0.023*"strategy"
0.018*"style"	0.011*"focus"	0.012*"advertisement"	0.013*"deixis"
0.017*"maxim"	0.008*"story"	0.010*"type"	0.013*"feature"
0.016*"utterance"	0.008*"criticism"	0.007*"qualitative"	0.012*"politeness"
0.013*"type"	0.007*"characterization"	0.007*"sentence"	0.008*"positive"
0.013*"speech"	0.007*"analyze"	0.006*"cohesion"	0.008*"utterance"
0.011*"character"	0.007*"problem"	0.006*"analyze"	0.008*"english"
0.009*"illocutionary"	0.006*"qualitative"	0.006*"function"	0.008*"type"
0.008*"qualitative"	0.006*"descriptive"	0.006*"descriptive"	0.007*"woman"
0.008*"base"	0.006*"experience"	0.005*"grammatical"	0.007*"movie"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 4 topik dan 5000 iterasi seperti Gambar 4.23 dibawah ini.

Gambar 4.23 Visualisasi LDA 4 Topik dan 5000 Iterasi

Setelah dua percobaan yang telah dilakukan yakni pada jumlah topik 2, 3, dan 4 kemudian dilakukan percobaan pertama dengan jumlah topik 5. Percobaan pertama dilakukan dengan jumlah topik 5, namun berbeda iteriasi yang dilakukan sebanyak 100 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 5
```

```
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=100)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 5 dan iterasi 100, diperoleh hasi topik seperti berikut:

Topik 1*: maxim, utterance, type, movie, illocutionary, advertisement, implicature, deixis, qualitative, directive*.

Topik 2: *type, words, process, cohesion, movie, grammatical, register, conjunction, qualitative, formation*.

Topik 3: *novel, focus, story, criticism, characterization, analyze, woman, character, problem, qualitative*.

Topik 4: *speech, style, feature, deixis, movie, english, woman, qualitative, using, type*.

Topik 5: *strategy, politeness, positive, factor, influence, type, character, using, analyze, sentence*.

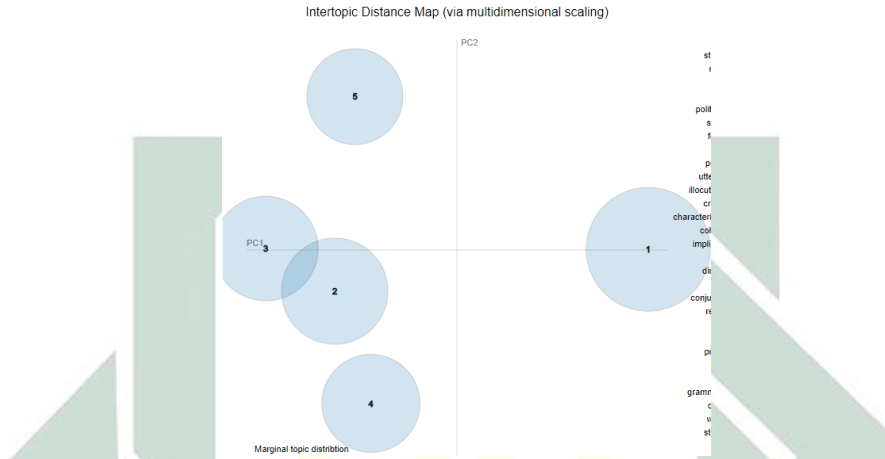
Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.18.

Tabel 4.18 List Kata dan Bobot dengan Jumlah Topik 5 pada Iterasi ke-100

Topik 1	Topik 2	Topik 3	Topik 4	Topik 5
0.019*"maxim"	0.013*"type"	0.030*"novel"	0.019*"speech"	0.034*"strategy"
0.017*"utterance"	0.011*"words"	0.011*"focus"	0.018*"style"	0.018*"politeness"
0.013*"type"	0.011*"process"	0.010*"story"	0.013*"feature"	0.011*"positive"
0.011*"movie"	0.009*"cohesion"	0.008*"criticism"	0.011*"deixis"	0.009*"factor"
0.010*"illocutionary"	0.009*"movie"	0.008*"characterization"	0.009*"movie"	0.007*"influence"
0.009*"advertisement"	0.007*"grammatical"	0.008*"analyze"	0.008*"english"	0.007*"type"
0.008*"implicature"	0.007*"register"	0.007*"woman"	0.007*"woman"	0.007*"character"
0.007*"deixis"	0.007*"conjunction"	0.007*"character"	0.007*"qualitative"	0.006*"using"

0.007*"qualitative"	0.007*"qualitative"	0.007*"problem"	0.007*"using"	0.006*"analyze"
0.007*"directive"	0.006*"formation"	0.007*"qualitative"	0.006*"type"	0.006*"sentence"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 5 topik dan 100 iterasi seperti Gambar 4.24 dibawah ini.



Gambar 4.24 Visualisasi LDA 5 Topik dan 100 Iterasi

Percobaan kedua dilakukan dengan jumlah topik 5, namun berbeda iteriasi yang dilakukan sebanyak 500 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 5
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=500)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 5 dan iterasi 500, diperoleh hasi topik seperti berikut:

Topik 1: *novel, words, type, grammatical, qualitative, cohesion, english, process, reference, movie*.

Topik 2: *feature, speech, woman, movie, register, using, figurative, type, sign, linguistic.*

Topik 3: *strategy, style, politeness, positive, student, influence, character, factor, movie, type.*

Topik 4: *novel, deixis, focus, story, person, criticis, "analyze, people, qualitative, problem.*

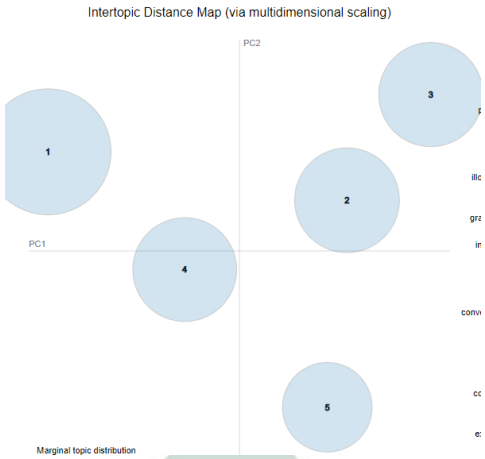
Topik 5: *utterance, maxim, type, speech, movie, function, illocutionary, conversation, implicature, qualitative.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.19.

Tabel 4.19 List Kata dan Bobot dengan Jumlah Topik 5 pada Iterasi ke-500

Topik 1	Topik 2	Topik 3	Topik 4	Topik 5
0.015*"novel"	0.020*"feature"	0.031*"strategy"	0.024*"novel"	0.020*"utterance"
0.013*"words"	0.014*"speech"	0.018*"style"	0.013*"deixis"	0.020*"maxim"
0.010*"type"	0.011*"woman"	0.016*"politeness"	0.010*"focus"	0.013*"type"
0.008*"grammatical"	0.008*"movie"	0.010*"positive"	0.009*"story"	0.012*"speech"
0.008*"qualitative"	0.008*"register"	0.010*"student"	0.009*"person"	0.010*"movie"
0.008*"cohesion"	0.008*"using"	0.009*"influence"	0.008*"criticism"	0.010*"function"
0.007*"english"	0.007*"figurative"	0.009*"character"	0.007*"analyze"	0.010*"illocutionary"
0.007*"process"	0.007*"type"	0.009*"factor"	0.007*"people"	0.009*"conversation"
0.007*"reference"	0.007*"sign"	0.008*"movie"	0.007*"qualitative"	0.008*"implicature"
0.006*"movie"	0.006*"linguistic"	0.008*"type"	0.006*"problem"	0.007*"qualitative"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 5 topik dan 500 iterasi seperti Gambar 4.25 dibawah ini.



Gambar 4.25 Visualisasi LDA 5 Topik dan 500 Iterasi

Percobaan ketiga dilakukan dengan jumlah topik 5, namun berbeda iteriasi yang dilakukan sebanyak 1000 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 5
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=1000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 5 dan iterasi 1000, diperoleh hasi topik seperti berikut:

Topik 1: *utterance, maxim, type, movie, speech, feature, function, words, illocutionary, character*.

Topik 2: *novel, focus, english, woman, problem, figurative, analyze, shift, cause, translation*.

Topik 3: *strategy, politeness, novel, character, positive, influence, factor, student, analyze, descriptive*.

Topik 4: *style, deixis, advertisement, type, speech, words, qualitative, person, movie, descriptive*.

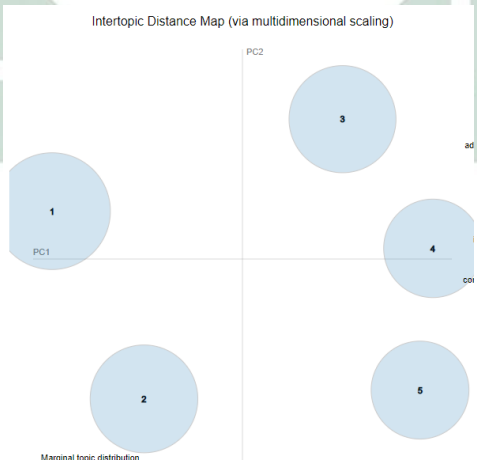
Topik 5: *novel, story, focus, cohesion, conflict, qualitative, short, conjunction, people, grammatical*.

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.20.

Tabel 4.20 List Kata dan Bobot dengan Jumlah Topik 5 pada Iterasi ke-1000.

Topik 1	Topik 2	Topik 3	Topik 4	Topik 5
0.018*"utterance"	0.019*"novel"	0.029*"strategy"	0.018*"style"	0.019*"novel"
0.016*"maxim"	0.009*"focus"	0.015*"politeness"	0.017*"deixis"	0.014*"story"
0.014*"type"	0.008*"english"	0.014*"novel"	0.013*"advertisement"	0.009*"focus"
0.014*"movie"	0.008*"woman"	0.011*"character"	0.009*"type"	0.008*"cohesion"
0.014*"speech"	0.006*"problem"	0.010*"positive"	0.009*"speech"	0.008*"conflict"
0.013*"feature"	0.006*"figurative"	0.010*"influence"	0.008*"words"	0.007*"qualitative"
0.009*"function"	0.006*"analyze"	0.007*"factor"	0.007*"qualitative"	0.007*"short"
0.008*"words"	0.006*"shift"	0.007*"student"	0.007*"person"	0.007*"conjunction"
0.008*"illocutionary"	0.006*"cause"	0.006*"analyze"	0.006*"movie"	0.006*"people"
0.007*"character"	0.006*"translation"	0.006*"descriptive"	0.006*"descriptive"	0.006*"grammatical"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 5 topik dan 1000 iterasi seperti Gambar 4.26 dibawah ini.



Gambar 4.26 Visualisasi LDA 5 Topik dan 1000 Iterasi

Percobaan keempat dilakukan dengan jumlah topik 5, namun berbeda iteriasi yang dilakukan sebanyak 5000 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 5
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=5000)
```

```
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 5 dan iterasi 5000, diperoleh hasi topik seperti berikut:

Topik 1: *advertisement, function, type, cohesion, qualitative, register, conjunction, grammatical, feature, lexical.*

Topik 2: *novel, focus, woman, characterization, criticism, analyze, experience, become, struggle, problem.*

Topik 3: *novel, character, conflict, shift, focus, story, qualitative, descriptive, disorder, presupposition.*

Topik 4: *strategy, maxim, politeness, movie, utterance, words, character, type, positive, analyze.*

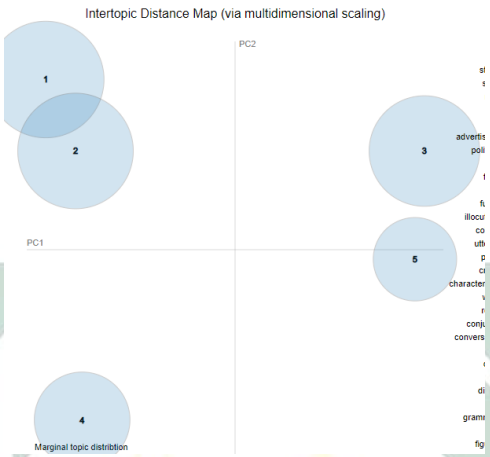
Topik 5: *speech, style, deixis, type, movie, utterance, feature, illocutionary, character, qualitative.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.21.

Tabel 4.21 List Kata dan Bobot dengan Jumlah Topik 5 pada Iterasi ke-5000

Topik 1	Topik 2	Topik 3	Topik 4	Topik 5
0.016*"advertisement"	0.027*"novel"	0.024*"novel"	0.025*"strategy"	0.020*"speech"
0.012*"function"	0.010*"focus"	0.009*"character"	0.016*"maxim"	0.016*"style"
0.010*"type"	0.009*"woman"	0.008*"conflict"	0.013*"politeness"	0.015*"deixis"
0.009*"cohesion"	0.008*"characterization"	0.008*"shift"	0.011*"movie"	0.014*"type"
0.008*"qualitative"	0.008*"criticism"	0.007*"focus"	0.011*"utterance"	0.012*"movie"
0.007*"register"	0.007*"analyze"	0.007*"story"	0.010*"words"	0.010*"utterance"
0.007*"conjunction"	0.007*"experience"	0.006*"qualitative"	0.009*"character"	0.009*"feature"
0.007*"grammatical"	0.006*"become"	0.006*"descriptive"	0.009*"type"	0.008*"illocutionary"
0.007*"feature"	0.006*"struggle"	0.005*"disorder"	0.008*"positive"	0.007*"character"
0.007*"lexical"	0.006*"problem"	0.005*"presupposition"	0.008*"analyze"	0.007*"qualitative"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 5 topik dan 5000 iterasi seperti Gambar 4.27 dibawah ini.



Gambar 4.27 Visualisasi LDA 5 Topik dan 5000 Iterasi

Percobaan selanjutnya dilakukan dengan jumlah topik 7, namun berbeda iteriasi yang dilakukan sebanyak 100 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 7
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=100)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 7 dan iterasi 100, diperoleh hasi topik seperti berikut:

Topik 1: *deixis, type, advertisement, story, person, function, cohesion, sentence, translation, reference*.

Topik 2: *novel, focus, process, problem, conflict, criticism, characterization, qualitative, words, struggle*.

Topik 3: *style, speech, figure, descriptive, context, qualitative, intimate, movie, type, consultative*.

Topik 4: *speech, english, error, novel, feature, student, movie, school, words, american.*

Topik 5: *strategy, politeness, factor, novel, positive, influence, analyze, using, qualitative, people*.

Topik 6: *utterance, maxim, movie, character, type, strategy, conversation, speaker, conversational, illocutionary*.

Topik 7: *woman, feature, character, novel, speech, interruption, female, movie, analyze, hedge*.

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.22.

Tabel 4.22 List Kata dan Bobot dengan Jumlah Topik 7 pada Iterasi ke-100

Topik 1	Topik 2	Topik 3	Topik 4	Topik 5	Topik 6	Topik 7
0.021*"deixis"	0.027*"novel"	0.038*"style"	0.014*"speech"	0.022*"strategy"	0.022*"utterance"	0.031*"woman"
0.013*"type"	0.011*"focus"	0.017*"speech"	0.012*"english"	0.016*"politeness"	0.020*"maxim"	0.014*"feature"
0.012*"advertisement"	0.010*"process"	0.008*"figure"	0.011*"error"	0.011*"factor"	0.018*"movie"	0.009*"character"
0.011*"story"	0.009*"problem"	0.008*"descriptive"	0.011*"novel"	0.010*"novel"	0.015*"character"	0.009*"novel"
0.009*"person"	0.008*"conflict"	0.007*"context"	0.011*"feature"	0.009*"positive"	0.014*"type"	0.007*"speech"
0.009*"function"	0.008*"criticism"	0.007*"qualitative"	0.010*"student"	0.009*"influence"	0.011*"strategy"	0.007*"interruption"
0.009*"cohesion"	0.008*"characterization"	0.006*"intimate"	0.007*"movie"	0.007*"analyze"	0.009*"conversation"	0.007*"female"
0.008*"sentence"	0.007*"qualitative"	0.006*"movie"	0.006*"school"	0.006*"using"	0.008*"speaker"	0.007*"movie"
0.008*"translation"	0.006*"words"	0.006*"type"	0.006*"words"	0.006*"qualitative"	0.008*"conversational"	0.007*"analyze"
0.007*"reference"	0.006*"struggle"	0.006*"consultative"	0.006*"american"	0.006*"people"	0.008*"illocutionary"	0.007*"hedge"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 7 topik dan 100 iterasi seperti Gambar 4.28 dibawah ini.

Gambar 4.28 Visualisasi LDA 7 Topik dan 100 Iterasi

Percobaan kedua dilakukan dengan jumlah topik 7, namun berbeda iteriasi yang dilakukan sebanyak 500 kali. *Code* yang digunakan sebagai berikut:

```
import genism
NUM_TOPICS = 7
ldamodel = genism.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=500)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 7 dan iterasi 500, diperoleh hasi topik seperti berikut:

Topik 1: *advertisement, type, cohesion, grammatical, conjunction, figurative, product, story, short, lexical*.

Topik 2: *woman, feature, speech, movie, conversation, american, type, utterance, using, character*.

Topik 3: *strategy, politeness, character, type, utterance, positive, illocutionary, function, movie, descriptive*.

Topik 4: *words, student, english, process, formation, slang, feature, affix, qualitative, analyze*.

Topik 5: *maxim, deixis, speech, person, utterance, type, flout, qualitative, context, violate*.

Topik 6: *novel, focus, criticism, story, characterization, experience, problem, conflict, struggle, analyze.*

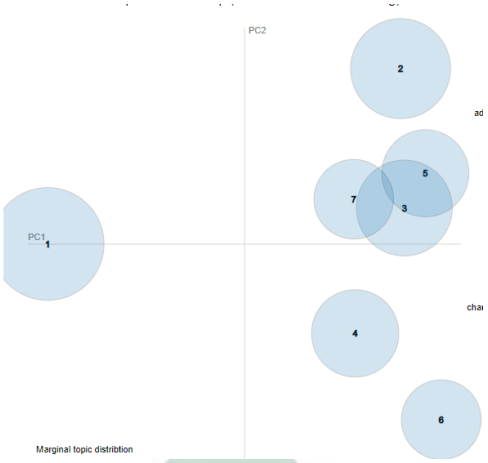
Topik 7: *style, movie, register, factor, shift, character, disagreement, utterance, social, situation.*

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.23.

Tabel 4.23 List Kata dan Bobot dengan Jumlah Topik 7 pada Iterasi ke-500

Topik 1	Topik 2	Topik 3	Topik 4	Topik 5	Topik 6	Topik 7
0.021*"advertisement"	0.016*"woman"	0.036*"strategy"	0.020*"words"	0.025*"maxim"	0.040*"novel"	0.035*"style"
0.012*"type"	0.014*"feature"	0.019*"politeness"	0.018*"student"	0.023*"deixis"	0.012*"focus"	0.022*"movie"
0.011*"cohesion"	0.012*"speech"	0.014*"character"	0.016*"english"	0.017*"speech"	0.011*"criticism"	0.011*"register"
0.009*"grammatical"	0.011*"movie"	0.013*"type"	0.012*"process"	0.011*"person"	0.010*"story"	0.010*"factor"
0.009*"conjunction"	0.008*"conversation"	0.013*"utterance"	0.010*"formation"	0.011*"utterance"	0.010*"characterization"	0.010*"shift"
0.008*"figurative"	0.008*"american"	0.012*"positive"	0.008*"slang"	0.010*"type"	0.008*"experience"	0.008*"character"
0.008*"product"	0.008*"type"	0.011*"illocutionary"	0.007*"feature"	0.008*"flout"	0.008*"problem"	0.008*"disagreement"
0.008*"story"	0.007*"utterance"	0.010*"function"	0.007*"affix"	0.008*"qualitative"	0.007*"conflict"	0.007*"utterance"
0.007*"short"	0.007*"using"	0.008*"movie"	0.007*"qualitative"	0.007*"context"	0.007*"struggle"	0.007*"social"
0.007*"lexical"	0.007*"character"	0.008*"descriptive"	0.007*"analyze"	0.007*"violate"	0.007*"analyze"	0.007*"situation"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 7 topik dan 500 iterasi seperti Gambar 4.29 dibawah ini.



Gambar 4.29 Visualisasi LDA 7 Topik dan 500 Iterasi

Percobaan ketiga dilakukan dengan jumlah topik 7, namun berbeda iteriasi yang dilakukan sebanyak 1000 kali. *Code* yang digunakan sebagai berikut:

```
import gensim
NUM_TOPICS = 7
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=1000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 7 dan iterasi 1000, diperoleh hasi topik seperti berikut:

Topik 1: *story, cohesion, grammatical, type, sentence, implicature, short, conjunction, movie, qualitative.*

Topik 2: *maxim, feature, woman, movie, utterance, process, words, flout, character, qualitative.*

Topik 3: *style, speech, movie, utterance, type, advertisement, illocutionary, character, function, qualitative.*

Topik 4: *speech, advertisement, words, affix, identity, error, toward, qualitative, three, feature.*

Topik 5: *strategy, deixis, politeness, type, positive, person, character, utterance, conversation, social.*

Topik 6: *student, english, sign, school, vocabulary, trump, language, donald, surabaya, learning*.

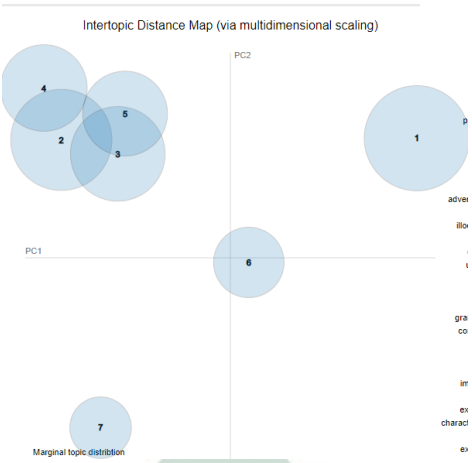
Topik 7: *novel, focus, criticism, problem, characterization, experience, analyze, conflict, struggle, become*.

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.24.

Tabel 4.24 List Kata dan Bobot dengan Jumlah Topik 7 pada Iterasi ke-1000.

Topik 1	Topik 2	Topik 3	Topik 4	Topik 5	Topik 6	Topik 7
0.012*"story"	0.023*"maxim"	0.020*"style"	0.012*"speech"	0.042*"strategy"	0.026*"student"	0.037*"novel"
0.011*"cohesion"	0.017*"feature"	0.016*"speech"	0.009*"advertisement"	0.025*"deixis"	0.020*"english"	0.013*"focus"
0.010*"grammatical"	0.011*"woman"	0.013*"movie"	0.008*"words"	0.022*"politeness"	0.014*"sign"	0.011*"criticism"
0.010*"type"	0.011*"movie"	0.012*"utterance"	0.008*"affix"	0.015*"type"	0.012*"school"	0.009*"problem"
0.010*"sentence"	0.011*"utterance"	0.012*"type"	0.006*"identity"	0.014*"positive"	0.009*"vocabulary"	0.008*"characterization"
0.009*"implicature"	0.009*"process"	0.010*"advertisement"	0.006*"error"	0.012*"person"	0.009*"trump"	0.008*"experience"
0.009*"short"	0.009*"words"	0.010*"illocutionary"	0.006*"toward"	0.010*"character"	0.009*"language"	0.008*"analyze"
0.009*"conjunction"	0.008*"flout"	0.009*"character"	0.006*"qualitative"	0.010*"utterance"	0.009*"donald"	0.007*"conflict"
0.007*"movie"	0.007*"character"	0.008*"function"	0.006*"three"	0.008*"conversation"	0.007*"surabaya"	0.007*"struggle"
0.007*"qualitative"	0.007*"qualitative"	0.008*"qualitative"	0.006*"feature"	0.008*"social"	0.007*"learning"	0.007*"become"

Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 7 topik dan 1000 iterasi seperti Gambar 4.30 dibawah ini.



Gambar 4.30 Visualisasi LDA 7 Topik dan 1000 Iterasi

Percobaan keempat dilakukan dengan jumlah topik 7, namun berbeda iteriasi yang dilakukan sebanyak 5000 kali. *Code* yang digunakan sebagai berikut:

```
import gensim
NUM_TOPICS = 7
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics =
NUM_TOPICS, id2word=dictionary, passes=5000)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=10)
for topic in topics:
    print(topic)
```

Berdasarkan percobaan yang telah dilakukan pada jumlah topik 7 dan iterasi 5000, diperoleh hasi topik seperti berikut:

Topik 1: *student, english, register, people, school, error, base, vocabulary, author, dialect*.

Topik 2: *feature, type, cohesion, woman, grammatical, lexical, conjunction, words, qualitative, sentence*.

Topik 3: *novel, conflict, human, mechanism, defense, anxiety, focus, descriptive, analyze, qualitative*.

Topik 4: *strategy, deixis, politeness, speech, type, utterance, positive, illocutionary, person, words*.

Topik 5: *novel, focus, story, criticism, characterization, problem, struggle, become, analyze, character*.

Topik 6: *style, advertisement, movie, implicature, type, conversational, figurative, qualitative, sentence, speech*.

Topik 7: *maxim, utterance, function, type, movie, flout" + 0.009*"words, violation, violate, character*.

Isi kata pada setiap topik berserta bobot kemunculan tersebut ditunjukan pada Tabel 4.25.

Tabel 4.25 List Kata dan Bobot dengan Jumlah Topik 7 pada Iterasi ke-5000.

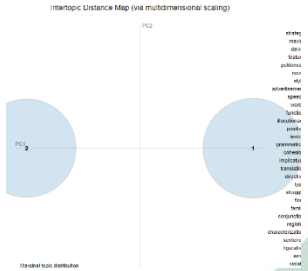
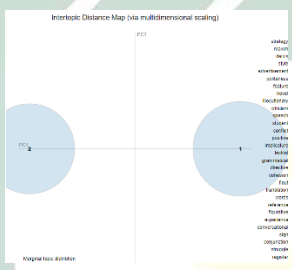
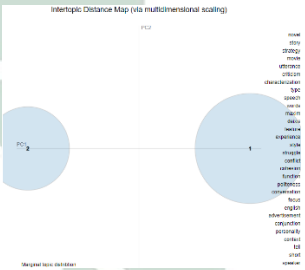
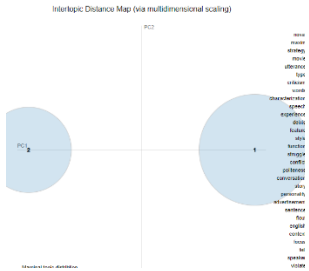
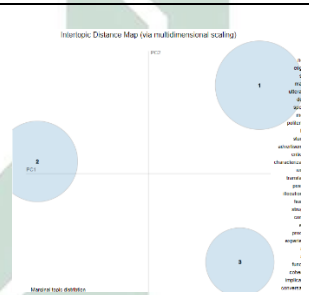
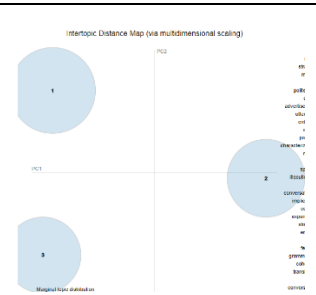
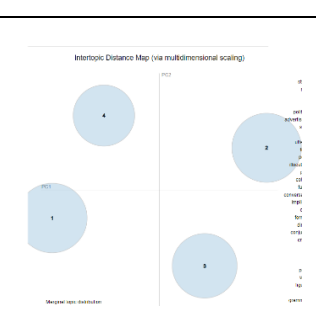
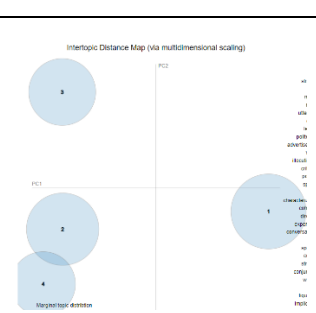
Topik 1	Topik 2	Topik 3	Topik 4	Topik 5	Topik 6	Topik 7
0.018*"student"	0.019*"feature"	0.023*"novel"	0.030*"strategy"	0.030*"novel"	0.029*"style"	0.033*"maxim"
0.016*"english"	0.011*"type"	0.014*"conflict"	0.018*"deixis"	0.012*"focus"	0.021*"advertisement"	0.016*"utterance"
0.011*"register"	0.010*"cohesion"	0.008*"human"	0.016*"politeness"	0.009*"story"	0.013*"movie"	0.016*"function"
0.008*"people"	0.010*"woman"	0.008*"mechanism"	0.015*"speech"	0.008*"criticism"	0.011*"implicature"	0.012*"type"
0.007*"school"	0.010*"grammatical"	0.007*"defense"	0.014*"type"	0.008*"characterization"	0.010*"type"	0.011*"movie"
0.007*"error"	0.009*"lexical"	0.007*"anxiety"	0.011*"utterance"	0.008*"problem"	0.009*"conversational"	0.011*"flout"
0.007*"base"	0.008*"conjunction"	0.007*"focus"	0.010*"positive"	0.007*"struggle"	0.008*"figurative"	0.009*"words"
0.007*"vocabulary"	0.008*"words"	0.006*"descriptive"	0.010*"illocutionary"	0.007*"become"	0.008*"qualitative"	0.009*"violation"
0.007*"author"	0.007*"qualitative"	0.006*"analyze"	0.009*"person"	0.007*"analyze"	0.008*"sentence"	0.009*"violate"
0.007*"dialect"	0.007*"sentence"	0.006*"qualitative"	0.009*"words"	0.007*"character"	0.008*"speech"	0.009*"character"

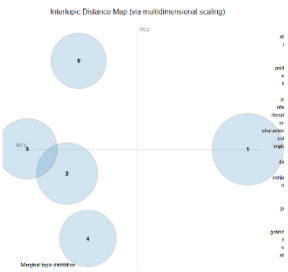
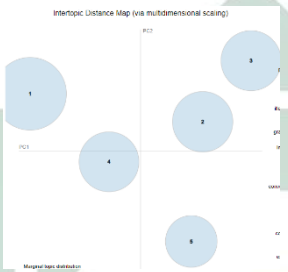
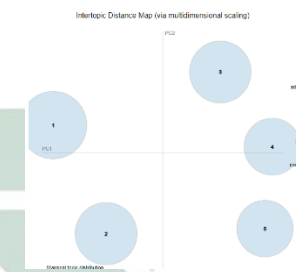
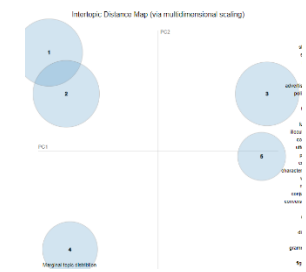
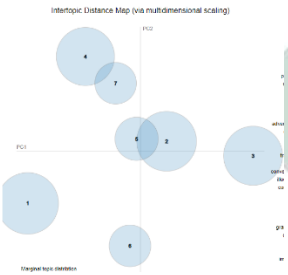
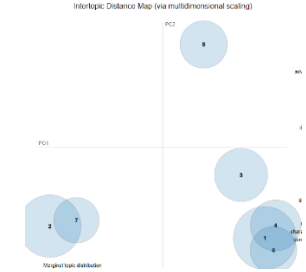
Bentuk visualisasi pemodelan LDA dari pembagian topik berdasarkan 7 topik dan 5000 iterasi seperti Gambar 4.31 dibawah ini.

Gambar 4.31 Visualisasi LDA 7 Topik dan 5000

4.5 Analisis Topik

Tabel 4.1 Rangkuman Visualisasi Pemodelan LDA

Jumlah Topik / Iterasi	100	500	1000	5000
2				
3				
4				

Jumlah Topik / Iterasi	100	500	1000	5000
5				
7				

Berdasarkan pemodelan yang sudah dilakukan sebelumnya, maka dapat dilihat hasil visualisasi seperti yang ada pada Tabel 4.6. Untuk menganalisis keseluruhan model LDA dari berbagai percobaan jumlah topik dan jumlah iterasi, maka dipermudah dengan menggunakan tabel. Dapat dilihat pada tabel dalam baris pertama dengan jumlah topik 2 yang dilakukan pada iterasi 100, 500, 1000, dan 5000. Pada iterasi ke-100 menghasilkan 2 topik yang berbeda. Hasil yang sama juga didapat pada iterasi ke-500 dengan mengahasilkan 2 topik yang berbeda. Begitupun pada iterasi ke-1000 dan 5000 dimana masing-masing menghasilkan 2 topik yang berbeda. Berdasarkan pada percobaan dengan perbedaan iterasi yang digunakan, kedua topik mempunyai jarak yang berjauhan. Dari 100, 500, 1000, dan 5000 iterasi menunjukan bahwasanya topik yang terbentuk berjumlah 2 topik dari 2 jumlah topik. 2 topik yang terbentuk memang sudah dapat dikategorikan sebagai 2 cluster yang berbeda. Akan tetapi untuk mengukur apakah 2 topik tersebut benar-benar 2 cluster yang berbeda, maka dilakukan pemodelan dengan jumlah topik 3 dan iterasi yang sama.

Dapat dilihat pada tabel dalam baris kedua dengan jumlah topik 3 yang dilakukan dari 100, 500, 1000, dan 5000 iterasi. Pada iterasi ke-100 menghasilkan 3 topik yang berbeda. Hasil yang sama didapat pada iterasi ke-500 dengan mengahasilkan 3 topik yang berbeda. Begitupun pada iterasi ke-1000 dan 5000 dimana masing-masing menghasilkan 3 topik yang berbeda. Berdasarkan pada percobaan dengan perbedaan iterasi yang digunakan, ketiga topik mempunyai jarak yang berjauhan. Dari 100, 500, 1000, dan 5000 iterasi menunjukan bahwasanya topik yang terbentuk berjumlah 3 topik dari 3 jumlah topik. 3 topik yang terbentuk memang sudah dapat dikategorikan sebagai 3 cluster yang berbeda. Akan tetapi untuk mengukur apakah 3 topik tersebut benar-benar 3 cluster yang berbeda, maka dilakukan pemodelan dengan jumlah topik 4 dan iterasi yang sama.

Untuk melihat apakah 4 jumlah topik merupakan topik yang cocok, maka penelitian ini mencoba melakukan pemodelan dengan jumlah topik berjumlah 4 yang ada pada baris ketiga. Iterasi yang dilakukan pun sama dengan pemodelan sebelumnya. Hasil pemodelan untuk jumlah topik 4 untuk iterasi ke-100 menghasilkan 4 topik yang berbeda. Pada percobaan tersebut tidak ada topik yang bergabung. Begitu juga pada iterasi ke-500 dihasilkan 4 topik berbeda, namun ada

2 topik yang berdekatan tetapi tidak beririsan. Selanjutnya dilakukan pemodelan pada iterasi ke-1000 dan didapatkan hasil 4 topik, dimana ada 2 topik yang beririsan. Sehingga dapat dikatakan 2 topik yang beririsan tersebut merupakan satu cluster. Sama halnya pada iterasi ke-5000 yang menghasilkan 4 topik yang berbeda dan 2 topik diantaranya beririsan. Dan untuk mengukur apakah 4 topik tersebut benar-benar 3 cluster yang berbeda, maka dilakukan pemodelan dengan jumlah topik 5 dan iterasi yang sama.

Pada baris keempat dilakukan pemodelan dengan jumlah topik 5 dan jumlah iterasi yang sama. Hasil pemodelan pada iterasi ke-100 menunjukkan 5 topik berbeda dengan 2 topik beririsan, namun ada 1 topik yang cenderung berdekatan terhadap topik yang beririsan. 3 topik tersebut mempunyai kemungkinan sebagai satu cluster yang sama. Untuk iterasi ke-500 dihasilkan 5 topik yang terpisah dengan 2 topik yang cenderung berdekatan. 2 topik yang cenderung berdekatan ini mempunyai kemungkinan berada dalam cluster yang sama. Untuk iterasi ke-1000 dihasilkan 5 topik yang berbeda. Dan untuk iterasi ke-5000 dihasilkan 5 topik yang berbeda. dimana terdapat 2 topik yang saling beririsan. 2 topik tersebut dapat dikatakan sebagai satu cluster yang sama. Untuk 2 topik yang berdekatan mempunyai kemungkinan berada dalam satu cluster yang sama.

Kemudian pemodelan dengan jumlah topik 7 ditunjukkan pada tabel baris kelima dengan jumlah iterasi 100, 500, 1000, dan 5000. Pada iterasi ke-100 dihasilkan 7 topik berbeda, dimana ada 4 bagian topik yang beririsan dan saling berdekatan. 4 topik tersebut berdekatan dengan 1 topik lainnya yang dapat dikatakan merupakan 1 cluster yang sama. Pada iterasi ke-500 terdapat 3 topik yang saling beririsan dan berdekatan dengan 1 topik lainya. Dan ada 2 topik yang cenderung berdekatan mempunyai kemungkinan berada dalam 1 cluster yang sama. Pada iterasi ke-1000 terdapat 4 topik yang saling beririsan. Keempat topik tersebut terdapat berada didalam 1 cluster yang sama. Dan untuk 3 topik lainnya berada terpisah jauh yang mempunyai kemungkinan merupakan cluseter yang berbeda. Dan terkahir pada iterasi ke-5000 mengasilkan 7 topik berbeda dengan 4 topik yang saling beririsan yang dapat dikatakan masuk kedalam cluster yang sama. Dan 2 topik yang beririsan yang terpisah dari topik lainnya dapat dikatakan cluster yang sama. Kesimpulan sementara yang dihasilkan pada pemodelan dengan jumlah topik

2, 3, 4, 5, dan 7 dengan iterasi 100, 500, 1000, dan 5000 mengerucut bahwa jumlah topik 3 merupakan pemodelan topik yang fit. Namun untuk melihat apakah jumlah topik 3 merupakan jumlah yang fit, penelitian ini melakukan verifikasi terhadap stakeholder program studi sastra inggris uinsa. Hasil verifikasi dari stakeholder mengatakan bahwasanya jumlah topik 3 belum sesuai dengan pembagian topik sebenarnya. Pihak stakeholder mengatakan bahwa seharusnya ada 2 pembagian topik pada program studi uinsa. Berdasarkan hasil verifikasi pertama, dilakukan penambahan pemodelan dengan jumlah topik 2. Secara visual dapat dilihat bahwa isi dari topik 2 dan 3 memiliki pola yang dapat dilihat pada Gambar 4.32.

Gambar 4.1 Hasil analisis output kata-kata antara jumlah topik 2 dan 3

Dari hasil analisis tersebut dapat dilihat bahwa dalam pemodelan dengan jumlah topik 3, terdapat 2 topik yang merupakan pecahan bagian dari salah satu topik tersebut. Kemudian untuk membuktikan apakah benar ada 2 topik yang merupakan hasil pecahan dari satu toik, penelitian ini melakukan verifikasi tahap dua. Hasil verifikasi tersebut, pihak stakeholder mengatakan bahwa memang benar 2 topik tersebut adalah pecahan dari suatu topik. Jadi pemodelan dengan jumlah topik 3 merupakan pemodelan cluster yang terbaik diantara pemodelan dengan jumlah topik lainya. Hal tersebut dikarenakan tidak adanya topik yang beririsan dan saling berjauhan.

BAB V
PENUTUP

5.1 Kesimpulan

Proses implementasi *topic modelling* menggunakan metode *Latent Dirichlet Allocation* (LDA) pada data *abstract* skripsi Program Studi Sastra Inggris Universitas Islam Negeri Sunan Ampel Surabaya (UINSA) dimulai dari tahap pengambilan data. Data yang diperoleh berjumlah 584 *abstract* skripsi, data tersebut dipersiapkan melalui tahap *pre-processing* untuk mempermudah dalam *topic modelling*. Hasil dari pre-processing kemudian dihitung jumlah kemunculan setiap kata dengan model *bag of words*. Jumlah kemunculan setiap kata tersebut menjadi ukuran dalam metode Latent Dirichlet Allocation (LDA) untuk dimodelkan. Dalam metode LDA jumlah topik *cluster* dan jumlah iterasi ditentukan diawal. Percobaan yang dilakukan dengan mengubah jumlah topik dan jumlah iterasi. Hasil dari pemodelan topik tersebut kemudian dilakukan analisa untuk melihat seberapa lazim kata tersebut dalam suatu topik. Pada penelitian ini dilakukan percobaan sebanyak 5 uji iterasi dengan iterasi berbeda yakni: 100, 500, 1000, dan 5000. Sedangkan terhadap setiap uji iterasi dimasukkan jumlah topik yang berbeda yaitu: 2, 3, 4, 5, dan 7. Hasil cluster topik terbaik didapat pada jumlah topik 3. Hasil *cluster* tersebut telah diverifikasi oleh pihak *stakeholder* Program Studi Sastra Inggris (UINSA) bahwa kata-kata yang ada pada topik *cluster* sesuai dengan pembagian topik menurut konsentrasi pada Program Studi Sastra Inggris (UINSA).

5.2 Saran

Saran pertama yang dapat dilakukan pada penelitian selanjutnya adalah menggunakan metode LDA dengan studi kasus data yang berbasis Bahasa Indonesia. Hal tersebut dikarenakan rata-rata penelitian menggunakan abstrak berbahasa Indonesia.

Saran kedua yang dapat dilakukan pada penelitian selanjutnya adalah nama *cluster* dapat ditentukan secara langsung dengan melihat *output* Latent Dirichlet Allocation berisi kata-kata dalam topik *cluster* tersebut.

DAFTAR PUSTAKA

Agusta, Y. (2007) 'K-Means – Penerapan, Permasalahan dan Metode Terkait', 3(Pebruari), pp. 47–60.

Albert Verasius Dian Sano (2019) *CARA KERJA DATA MINING – SERI DATA MINING FOR BUSINESS INTELLIGENCE (3)*. Available at: https://binus.ac.id/malang/2019/01/cara-kerja-data-mining-seri-data-mining-for-business-intelligence-3/.

Alghamdi, R. (2015) 'A Survey of Topic Modeling in Text Mining', 6(1), pp. 147–153.

Campbell, J. C., Hindle, A. and Stroulia, E. (2014) 'No Title', *Latent Dirichlet Allocation: Extracting Topics*.

Daniel, T. (2005) *An Introduction to Data Mining*.

David M. Blei, Andrew Y. Ng, M. I. J. (2003) 'Machine Learning Research 3', *Latent Dirichlet Allocation*, pp. 993–1022.

Fajriyanto, M. (2018) 'Penerapan metode bayesian dalam model latent dirichlet allocation di media sosial application of bayesian methods in latent dirichlet allocation model in social media', pp. 1–6.

Hofmann, T. (2001) 'Unsupervised Learning by Probabilistic Latent Semantic Analysis', pp. 177–196.

Ingason, A. K. *et al.* (2008) 'A Mixed Method Lemmatization Algorithm Using a Hierarchy of Linguistic Identities (HOLI)', *Lecture Notes In Artificial Intelligence*, pp. 205–216.

Jaka, A. T. (2015) 'Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining', *Jurnal informatika UPGRIS*.

Kahfi, A. S. (2006) 'Informasi dalam Perspektif Islam', *Ejournal Unisba*, 7.

Kaur, J. and Buttar, P. K. (2018) 'A Systematic Review on Stopword Removal Algorithms', *International Journal on Future Revolution in Computer Science & Communication Engineering*, 4(4).

Marti Hearst (2003) 'What Is Text Mining'.

McTear, M., Callejas, Z. and Barres, D. G. (2016) *The Conversational Interface*.

Putra, Fi. R. (2019) *Data Mining dan Contoh Implementasi di Berbagai Bidang*. Available at: https://www.kompasiana.com/figorahput/5c927a740b531c34651a0062/data-mining-dan-contoh-implementasi-di-berbagai-bidang.

Putra, M. K. B. and Renny Pradina Kusumawardani (2017) 'Analisis Topik Informasi Publik Media Sosial Di Surabaya Menggunakan Pemodelan Latent Dirichlet Allocation (Lda) Topic Analysis of Public Information in Social

Media in Surabaya Based on Latent Dirichlet Allocation (Lda) Topic Modelling', *Jurnal Teknik Its*, 6(2), pp. 2–7.

Ronen, F. and Sanger, J. (2007) *The Text Mining Handbook: Advance Approaches in Analyzing Unstructured Data*. United States of America: Cambridge University Press.

Ryan Diaz (2013) *Pengertian Data Mining,Teks Mining,dan Web Mining*. Available at: http://yosephoriolryandiaz.blogspot.com/2013/03/pengertian-data-miningteks-miningdan.html.

S, D., Raj, P. and S.Rajaraajeswari (2016) 'A Framework for Text Analytics using the Bag of Words (BoW) Model for Prediction', *International Journal of Advanced Networking & Applications (IJANA)*, pp. 320–323.

Turban, E., Aronson, J. E. and Liang, T.-P. (2004) *Decision Support Systems and Intelligent Systems (7th Edition)*.

Turban, E., Aronson, J. E. and Liang, T.-P. (2005) *Decision support systems and intelligent systems*. Andi Offset.

Utami, K. P. (2017) 'Analisis topik data media sosial twitter menggunakan model topik latent dirichlet allocation keke putri utami'.

Yahir Even dan Zohar (2002) *introduction to text mining. Automeated Learning Group National Center For Supercomputing Aplications*.

Zulhanif (2016) 'Pemodelan Topik Dengan Latent Dirichlet Allocation', *Seminar Nasional Pendidikan Matematika*, pp. 1–8.