

SISTEM DETEKSI TOPIK POLITIK PADA TWITTER MENGUNAKAN ALGORITMA *LATENT DIRICHLET* *ALLOCATION*

Naskah Publikasi Jurnal



Diajukan oleh:

KHAIRUL HUDHA NASUTION
5235154528

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER
FAKULTAS TEKNIK
UNIVERSITAS NEGERI JAKARTA
2019**

NASKAH PUBLIKASI JURNAL

**SISTEM DETEKSI TOPIK POLITIK PADA TWITTER
MENGUNAKAN ALGORITMA *LATENT DIRICHLET*
*ALLOCATION***

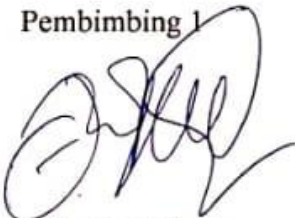
yang diajukan oleh :

KHAIRUL HUDHA NASUTION

5235154528

Telah disetujui oleh :

Pembimbing 1



Widodo, M. Kom.
NIP. 197203252005011002

Tanggal 20 - 8 - 2019

Pembimbing 2



Bambang Prasetya Adhi, S.Pd., M.Kom
NIP. 198302252014041001

Tanggal 20 - 8 - 2019

SISTEM DETEKSI TOPIK POLITIK PADA TWITTER MENGUNAKAN ALGORITMA LATENT DIRICHLET ALLOCATION

Khairul Hudha Nasution¹, Widodo², Bambang Prasetya Adhi³

¹ Mahasiswa Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

^{2,3} Dosen Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

¹ hudanasution@gmail.com, ² widodo@unj.ac.id, ³ bambangpadhi@unj.ac.id

Abstrak

Tahun 2019 Indonesia melaksanakan tahun politik, banyak terjadi peristiwa politik yang membuat masyarakat Indonesia menyikapi dengan berbagai macam tanggapan dari berbagai banyak tanggapan tersebut beberapa dituliskan dalam media sosial Twitter dan data tersebut dapat diolah untuk menggambarkan bagaimana pendapat masyarakat akan suatu kejadian politik. Penelitian ini bertujuan untuk mendapatkan analisis dari implementasi algoritma LDA untuk menentukan topik politik pada Twitter. Metode yang digunakan adalah dengan algoritma LDA yang digunakan untuk menghitung kemungkinan topik yang ada untuk setiap tweet-nya, LDA merupakan model probabilistik yang dapat menggambarkan topik tanpa perlu melakukan proses klasifikasi sebelumnya, sistem akan otomatis mendeteksi topik-topik yang ada. Hasil penelitian dengan pengujian 3 kali dengan jumlah data masing-masing 100, 1000 dan 6000 dengan menggunakan setingan LDA bawaan dari library Genism dan jumlah topik 10 menghasilkan rata-rata nilai kebenaran 90%. Sehingga dapat disimpulkan bahwa LDA dapat digunakan dan memiliki nilai kebenaran yang tinggi dalam mendeteksi topik politik pada Twitter.

Kata kunci: Latent Dirichlet Allocation, Natural Language Processing, Twitter

1. Pendahuluan

Informasi merupakan data yang telah diberi makna melalui konteks. Informasi sendiri dapat disampaikan dengan berbagai macam cara dan media, mulai dari bentuk gambar, suara ataupun tulisan, tergantung dari tujuan dan keinginan penyampai atau pembuat untuk menyampaikan informasi tersebut, dan informasi dalam bentuk tulisan adalah model penyampaian informasi yang paling banyak digunakan karena kemudahan dalam membuat, menyebarkan dan banyaknya informasi yang disampaikan dalam satu tulisan.

Ada 2 macam teks, yaitu teks panjang dan teks pendek. Teks panjang biasa kita temui pada paragraf, dimana paragraf memiliki banyak kata dan teks pendek yang biasa kita temui pada kalimat atau judul, dimana kalimat tidak memiliki banyak kata di dalamnya, sehingga memungkinkan penyampain informasi yang tidak sesuai. Dalam menentukan isi kandungan dari suatu berita atau paragraf kita bisa melakukan dengan menghitung kata unik yang paling banyak keluar, sedangkan pada teks pendek hal tersebut cenderung lebih sulit dilakukan karena keterbatasan kata yang ada dalam teks pendek.

Pada tahun ini (2019) Indonesia menjalani pesta politik yaitu pemilihan wakil rakyat dan presiden dan wakilnya yang dilaksanakan 5 tahun sekali, reaksi masyarakatpun bermacam-macam

dalam mengomentari terhadap tindakan dan peristiwa politik yang ada. Salah satu komentar-komentar yang disampaikan tertulis di media sosial dimana masyarakat dapat dengan bebas menyampaikan pendapatnya atas kejadian politik yang terjadi. Dari media sosial pula dapat menjadi tolak ukur tanggapan masyarakat atas suatu putusan atau suatu kejadian tertentu baik buruknya suatu kejadian di pandangan masyarakat umum, bahkan dapat menggambarkan secara tidak langsung menang kalahnya pasangan calon berdasarkan jumlah dan sentimen publik pada beberapa tokoh.

Salah satu media sosial yang paling populer adalah Twitter. Menurut (Yudha Pratomo, 2019) yang dikutip dari tekno.kompas.com pada kuartal empat 2018 tercatat rata-rata user aktif di Indonesia perharinya kurang lebih 126 juta user. Pengguna Twitter dapat menulis (tweet) tulisan hingga batas maksimal 280 karakter.

Untuk mengetahui tanggapan orang pada media sosial perlu dikumpulkannya data dan melakukan penggambaran berdasarkan tanggapan masyarakat, proses ini biasanya memakan banyak waktu karena memproses banyak data, dan proses yang dilakukan adalah proses yang repetitif yang artinya dapat digantikan dengan komputer. Dalam menyelesaikan masalah tersebut ilmu teknologi sekarang ini memiliki dua solusi utama yaitu menggunakan machine learning atau artificial

intelligence. Artificial Intelligence adalah sistem yang bertujuan untuk melakukan apa yang dilakukan manusia dengan lebih baik sedangkan machine learning bertujuan untuk belajar dan bertindak berdasarkan apa yang dipelajari, pada kesempatan kali ini peneliti memilih menggunakan machine learning karena dirasa sudah cukup dan paling baik digunakan pada kasus ini.

Dalam machine learning terdapat natural language processing yang intinya bekerja dengan mencoba memahami tulisan manusia, untuk memahami tulisan manusia ada teknik yang namanya topic modeling atau permodelan topik yang bekerja untuk mencari inti topik dari kumpulan-kumpulan kata atau dokumen. Didalam permodelan topik ada dua cara populer yaitu Latent Dirichlet Allocation dan Latent Semantic Analysis atau Probabilistic Latent Semantic Analysis.

LDA (Latent Dirichlet Allocation) merupakan model probabilistik generatif dari koleksi data diskrit, dimana LDA akan memberikan hasil perhitungan antara document, topics dan words untuk menentukan tingkat kesesuaian antar document (pada kasus ini tweet). LDA sendiri dipilih sebagai algoritma yang digunakan karena dirasa paling cocok dan paling tinggi tingkat keakuratannya untuk menggambarkan topic dari dokumen dibanding model lainnya dan sebagai pengujian seberapa bagus implementasinya pada teks pendek (Twitter).

Kita sebagai manusia terkadang mencari sesuatu berdasarkan topik yang kita senangi, tak terkecuali pada tulisan yang kita baca pada media sosial Twitter, kita sebagai manusia mungkin mengetahui dan bisa memilih mana yang sesuai dengan yang kita cari dan butuhkan mana yang tidak, namun komputer tidak memahami apa yang ditulis manusia. Oleh karena itu diharapkan dengan menerapkan implementasi LDA untuk menentukan topik dapat membuat komputer dapat menggambarkan topik pada tweet khususnya pada kasus politik dan umumnya pada teks pendek.

2. Dasar Teori

2.1. Kerangka Teoritik

2.1.1. Deteksi Topik

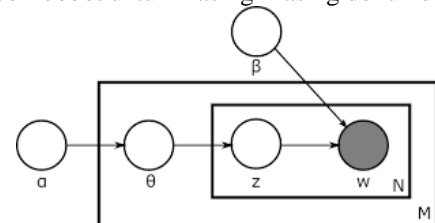
Menurut (James Allan, 2002) tugas deteksi topik adalah dapat menentukan topik dari suatu novel yang sebelumnya belum diketahui. Dalam pengerjaannya topik ditentukan dari cerita yang dibahasnya, namun sistem tidak diberi tahu topiknya. Sehingga sistem harus memiliki kemampuan untuk mengetahui dan memahami topik dari cerita, dan kinerja sistem haruslah independent.

Dalam KBBI sendiri, deteksi artinya usaha menemukan dan menentukan keberadaan, anggapan, atau kenyataan. Topik artinya pokok pembicaraan dalam diskusi, ceramah, karangan, dan sebagainya; bahan diskusi. Berdasarkan penjelasan diatas dapat ditarik kesimpulan bahwa deteksi topik adalah

kegiatan untuk menentukan pokok pembicaraan dalam suatu diskusi atau tulisan.

2.1.2. Latent Dirichlet Allocation

Menurut (Blei, 2003) Latent Dirichlet Allocation (LDA) merupakan model probabilistik generative dari kumpulan tulisan yang disebut corpus. Ide dasarnya adalah setiap dokumen adalah hasil presentasi berbagai macam topik acak, yang mana setiap topik memiliki karakter yang ditentukan berdasarkan distribusi kata-kata yang terdapat di dalamnya. Menurut (Campbell, 2014) LDA dapat digunakan untuk meringkas, melakukan klusterisasi, menghubungkan maupun memproses data yang sangat besar karena LDA menghasilkan daftar topik yang diberi bobot untuk masing-masing dokumen.



Gambar 2.1. Visualisasi LDA

Blei memvisualisasikan LDA seperti gambar 2.1, dimana terdapat tiga tingkatan pada representasi LDA. Variable α dan β sebagai variable tingkat corpus, variable θ sebagai variable tingkat dokumen (M) dan variabel Z dan W sebagai variable tingkat kata(N). Dengan variable α sebagai banyak distribusi topik pada dokumen, semakin besar nilainya menandakan campuran topik yang dibahas di dalam dokumen semakin banyak. Variable β sebagai banyak distribusi kata dalam topik, semakin tinggi nilainya maka semakin banyak kata di dalam topik. Variable θ sebagai distribusi topik untuk dokumen tertentu, semakin tinggi nilainya maka semakin banyak topik dalam satu dokumen. Variable Z mempersentasikan topik dari kata tertentu pada sebuah dokumen dan variable W mempersentasikan kata yang berkaitan dengan topik tertentu yang terdapat di dalam dokumen. Bentuk lingkaran mempersentasikan individual kata, lingkaran abu-abu yang diteliti dan yang kosong yang tidak secara langsung diteliti, jika kita mengambil kemungkinan marginal dari satu dokumen, maka dapat rumusan dari LDA dapat didefinisi sebagai berikut:

$$p(D|\alpha, \beta) = \prod_{d=1}^M \int p(\theta_d | \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} | \theta_d) p(w_{dn} | z_{dn}, \beta) \right) d\theta_d$$

Menurut (Campbell, 2014) secara umum LDA bekerja dengan memberi input beberapa parameter yang menghasilkan keluaran berupa model yang memiliki bobot sehingga model tersebut dapat dinormalisasi sesuai probabilitas. Probabilitas mengacu pada dua jenis yakni (a) probabilitas bahwa suatu dokumen spesifik akan menghasilkan topik yang spesifik pula dan (b) probabilitas bahwa suatu topik spesifik menghasilkan kata-kata spesifik tertentu. Probabilitas jenis (a), dokumen yang sudah diberi label dengan daftar topik seringkali

dilanjutkan hingga menghasilkan probabilitas jenis (b), yang menghasilkan kata-kata spesifik tertentu.

2.1.3. Politik

Menurut (Miriam Budiardjo, 2007) dalam bukunya mengatakan politik dalam suatu negara (state) berkaitan dengan masalah kekuasaan (power) pengambilan keputusan (decision making), kebijakan publik (public policy), dan alokasi atau distribusi (allocation or distribution). Dewasa ini definisi mengenai politik yang sangat normatif itu telah terdesak oleh definisi-definisi lain yang lebih menekankan pada upaya (means) untuk mencapai masyarakat yang baik, seperti kekuasaan, pembuatan keputusan, kebijakan, alokasi nilai, dan sebagainya.

Pada umumnya dapat dikatakan bahwa politik (politics) adalah usaha untuk menentukan peraturan-peraturan yang dapat diterima baik oleh sebagian besar warga, untuk membawa masyarakat ke arah kehidupan bersama yang harmonis. Usaha menggapai the good life ini menyangkut bermacam-macam kegiatan yang antara lain menyangkut proses penentuan tujuan dari sistem, serta cara-cara melaksanakan tujuan itu. Masyarakat mengambil keputusan mengenai apakah yang menjadi tujuan dari sistem politik itu dan hal ini menyangkut pilihan antara beberapa alternatif serta urutan prioritas dari tujuan-tujuan yang telah ditentukan itu.

2.1.4. Twitter

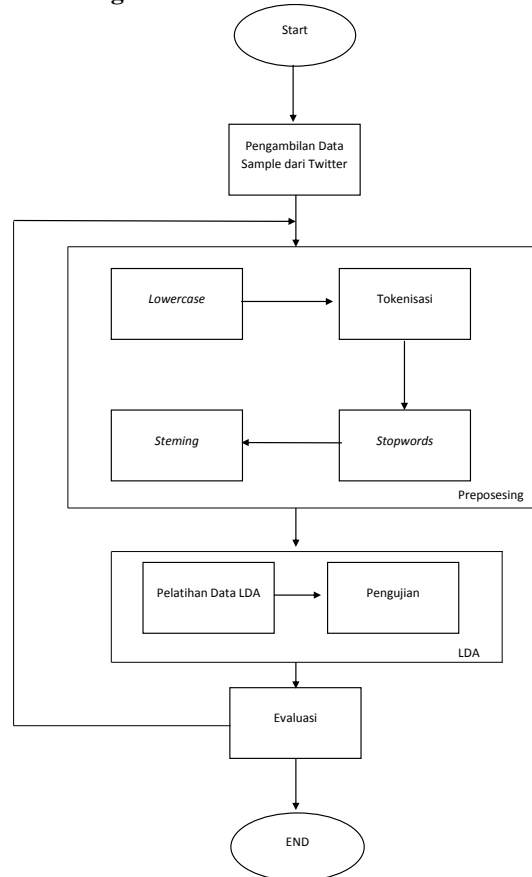
Twitter adalah berita online dan jejaring sosial dimana orang berkomunikasi dengan teks pendek yang disebut tweet. Tweet yang di publish akan dikirimkan kepada orang-orang yang mengikutinya di Twitter. Twitter juga dapat dideskripsikan sebagai microblogging (Paul Gil, 2019).

Twitter bekerja dengan cara mengirimkan dan menerima pesan pendek dari dan kepada yang mengikuti dan diikuti di Twitter. Twitter juga dapat memfollow dan unfollow user, retweet dan hastag. Twitter dapat digunakan sebagai sosial media, layanan berita, alat bantu jualan, forum, layanan pesan dan lain sebagainya.

Pada bulan September dilakukan uji coba perubahan batasan karakter dari yang awalnya 140 karakter per tweet menjadi 280 karakter, kecuali pada bahasa Jepang, Korea dan China karena perbedaan bahasa. Uji coba tersebut diterapkan secara global pada 7 November 2017 (Aliza Rosen, 2017).

3. Metodologi

3.1 Diagram Alir Penelitian



Gambar Error! No text of specified style in document..1. Diagram Alir Penelitian

3.2 Pengambilan Data

Untuk proses pengambilan data ini akan digunakan bantuan dari library Tweepy, yang bekerja dengan cara men-streaming Twitter, api ini bekerja dengan menyalin data dari Twitter. Pada proses ini peneliti membuat parameter untuk data-data yang ingin diambil dan tidak lupa menyaring data-data yang diambil sesuai dengan apa yang dibutuhkan. Setelah data di copy peneliti memilih untuk meletakkan data tersebut di dalam format teks (.txt) untuk nantinya diolah lagi kedepannya. Untuk indikator-indikator yang diambil adalah:

1. User atau id dari penulis tweet,
 2. Waktu penulisan tweet atau waktu publish,
 3. Tweet yang berisi maximal 280 karakter,
- akan dispesifikkan menjadi seperti berikut:
1. Mengambil data dengan bantuan hastag (fitur dalam Twitter) untuk pengelompokan tweet dari hastag #pemilu2019, #pilpres2019 dan #pileg2019 untuk jumlah masing masing data diacak.
 2. Jumlah total keseluruhan data yang diambil berkisar antara 100 (seratus) hingga 6000 (enam ribu) tweet.
 3. Proses pengambilan pada tanggal 10 Agustus 2019.

3.3 PreProcessing

Berikut tahapan preprocessing yang dilakukan pada penelitian ini:

1. Lowercase

Pada tahapan ini data yang diperoleh (tweet) disamaratakan menjadi huruf kecil semua per hurufnya tanpa mengubah struktur katanya, karna jika tidak dilakukan maka contoh kata “makan” akan diartikan beda dengan “Makan”, “MAKAN” dan lain sebagainya (case sensitive).

2. Tokenisasi

Setelah semua huruf diubah menjadi huruf kecil semua kata-kata dibuat menjadi potongan kecil yang disebut token atau potongan kata tunggal. Tahapan ini akan menghilangkan tanda baca, angka dan karakter lainnya yang tidak ada di dalam alfabet sehingga kata akan berdiri sendiri-sendiri.

3. Stopwords

Setelah kata dipisah-pisah kata akan masuk ketahapan ini yaitu penghilangan kata yang tidak memiliki makna yang dianggap akan banyak muncul dalam hasil sehingga akan merusak hasil.

4. Stemming

Pada tahapan ini kata-kata akan diolah kembali menjadi bentuk aslinya dengan cara menghilangkan imbuhan, yang bertujuan mengurangi jumlah kata yang masuk kedalam data penelitian dengan harapan meningkatkan akurasi.

3.4 LDA

Pelatihan menggunakan LDA Proses pelatihan menggunakan LDA bertujuan untuk mendapatkan nilai topic proportion dan probabilitas kata topik. Pelatihan dilakukan dengan menggunakan aproksimasi inferensi Gibbs Sampling. Persamaan matematis Gibbs Sampling dapat dilihat pada Persamaan (1) (Kusumaningrum, 2014).

$$p(z_i = k | \vec{z}_{-i}, \vec{w}, a, b) = \frac{n_{k, \sim i}^{(t)} + b_t}{\sum_{t=1}^V n_{k, \sim i}^{(t)} + b_t} * n_{d, \sim i}^{(k)} + a_k$$

Dari Persamaan berikut dicari nilai topic proportion dengan menggunakan Persamaan (1) dan probabilitas kata topiknya menggunakan Persamaan

$$r_{k,t} = p(w = t | z = k) = \frac{n_{t,k} + b_t}{\sum_{t=1}^V n_{t,k} + b_t}$$
$$q_{d,k} = p(z = k | d) = \frac{n_{d,k} + a_k}{\sum_{k=1}^V n_{d,k} + a_k}$$

Hasil dari pelatihan data ini akan menghasilkan kumpulan nilai untuk setiap corpus yang nantinya akan dijadikan panduan untuk menentukan kata apa yang akan menjadi indikasi dari suatu topik atau bukan.

Setelah didapat indikator suatu topik melalui proses pelatihan dilakukan pengujian menggunakan LDA untuk data-data sisanya proses pengujian bertujuan untuk mengukur validitas dan menghitung nilai topic proportion dokumen uji berdasarkan nilai probabilitas kata topik untuk setiap kata dalam dokumen uji. Pengujian dengan LDA dilakukan menggunakan Persamaan (4) (Pang dkk., 2002).

$$q_{d,k} = p(z_k) * \sum_{i=1}^{N_d} r_{k,t}^{(i)}$$

Setelah dilakukan pengujian menggunakan LDA dan didapatkan nilai untuk data uji, proses selanjutnya adalah melakukan perhitungan untuk mencari pola kemiripan distribusi probabilitas menggunakan Kullback Leiber Divergence (KLD). Rumus KLD dapat dituliskan dalam bentuk matematis seperti dalam Persamaan berikut.

$$KLD = \frac{DPQ + DQP}{2}$$

3.5 Visualisasi Data

Setelah data diproses akan menghasilkan nilai-nilai matematis, dan nilai matematis tersebut yang nantinya akan digunakan untuk mengukur jarak antara satu corpus dan lainnya, digunakan sebagai dasar untuk penentuan topik. Untuk mempermudah analisis dibuatlah pemvisualisasian menggunakan bantuan library matplotlib, matplotlib membantu memvisualisasikan angka tersebut dalam bentuk vektor 2 atau 3 dimensi seperti pada gambar ilustrasi

3.6 Evaluasi

Evaluasi dilakukan setelah proses pelatihan dan pengujian dengan LDA selesai dilakukan. Proses evaluasi bertujuan untuk mendapatkan hasil mengenai seberapa baik model LDA yang dibuat, dengan menghitung akurasi berdasarkan jumlah topik benar dibagi total. Perhitungan akurasi dilakukan dengan menggunakan Persamaan berikut.

$$\text{Accuracy} = \text{NB} / (\text{NB} + \text{NS})$$

Dimana :

NB : Total nilai benar

NS : Total nilai salah

Setelah melakukan evaluasi akan didapat persentase keberhasilan pengujian, berdasarkan rumus akurasi. peneliti menargetkan angka diatas 80% dari total jumlah pengujian dengan harapan dapat menggambarkan kinerja LDA pada topik teks pendek dalam penentuan topik politik.

4. Hasil dan Analisis

4.1 Deskripsi Hasil Penelitian

Penelitian dilakukan dengan mengambil sumber data dari tweet masyarakat yang dapat diakses secara mudah melalui situs twitter.com, data tersebut akan diolah sehingga menghasilkan nilai untuk nantinya digunakan untuk menentukan topik dalam kasus yang sama, dalam hal ini teks pendek. Penelitian ini akan menghasilkan analisis dan prototipe mengenai kinerja dari algoritma LDA dalam deteksi topik pada Twitter

4.2 Hasil Penelitian dan Analisis

4.2.1 Pengambilan Data

Dalam pengambilan data peneliti tidak mengambil data (tweet) secara manual melainkan menggunakan bantuan library untuk mempermudah dan mempercepat kerja, library yang gunakan disini adalah Tweepy. Pertama-tama sebelum menggunakan library dibutuhkan akses token yang berupa kode yang harus kita miliki, kode tersebut didapat dari Twitter langsung dengan cara mendaftar

di halaman developer Twitter, developer.twitter.com, di halaman itu user harus mengisi data yang diminta, proses pembuatan berlangsung 1 jam hingga 2 hari. Setelah terdaftar kita dapat meng-generate token yang nantinya akan digunakan untuk menggunakan tweepy. Pengambilan data dilakukan pada tanggal 10 Agustus 2019. Untuk menggunakan Tweepy kita cukup melakukan query dan filtering sesuai kebutuhan, karna pada saat penarikan data API mengirimkan semua data (tweet) biasanya dalam format “json”, data tersebut tidak semua dipakai sesuai dengan yang dibutuhkan saja, berikut contoh hasil pemanggilan data tweet menggunakan library Tweepy sebelum dan sesudah filtering Gambar 4.1.

Sebelum filtering	sesudah filtering
<pre>created_at: 'Wed Aug 07 22:32:45 +0000 2019', 'id': 1159230927471439874, 'id_str': '1159230927471439874', 'text': '@kompascom https://t.co/MR36NeGspT', 'truncated': False, 'entities': {'hashtags': [], 'symbols': [], 'user_mentions': [{'screen_name': 'kompascom', 'name': 'Kompas.com', 'id': 23343960, 'id_str': '23343960', 'indices': [0, 10]}, {'url': [], 'media': [{'id': 1159230918885699584, 'id_str': '1159230918885699584', 'indices': [11, 34], 'media_url': https://pbs.twimg.com/media/EBZqYPzU8AAYMv6.jpg, 'media_url_https': https://pbs.twimg.com/media/EBZqYPzU8AAYMv6.jpg, url': https://t.co/MR36NeGspT, 'display_url': pic.twitter.com/MR36NeGspT, 'expanded_url': https://twitter.com/ibnuasyary1975/status/1159230927471439 874/photo/1, 'type': 'photo', 'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'}, 'small': {'w': 680, 'h': 510, 'resize': 'fit'}, 'large': {'w': 2048, 'h': 1536, 'resize': 'fit'}, 'medium': {'w': 1200, 'h': 900, 'resize': 'fit'}}, 'extended_entities': {'media':</pre>	<pre>2018-03-13 09:29:38 ibnuasyary1975 @kompascom https://t.co/MR36NeGspT</pre>

Gambar 4.1 Data Sebelum dan Sesudah Filtering

Dalam melakukan filtering dari sekian banyak data yang ada peneliti memilih hanya untuk mengambil waktu penulisan tweet, user dan tweetnya hanya tiga data yaitu yang dipertahankan untuk data yang lainnya tidak digunakan. Untuk query yang digunakan untuk melakukan pemanggilan data adalah sebagai berikut:

1. Data diambil dengan query/kata kunci pilpres
2. Lokasi tweet dibuat adalah di Indonesia
3. Tweet yang diambil termasuk retweet

Hasil dari semua query tersebut menghasilkan data yang di letakkan di file dengan format .txt (raw.txt) dan sheet excel dengan nama RawData, file .txt digunakan untuk proses selanjutnya dan file excel digunakan untuk visualisasi data agar lebih mudah dipahami dan lebih rapih

4.2.2 PreProcesing

a) Lowercase

Proses lowercase (pengubahan ke bentuk non-kapital) hanya dengan menggunakan fungsi bawaan python, contoh kode dan contoh kasus ada pada Gambar 4.2. Untuk file yang digunakan adalah raw.txt dengan keluaran sheet Prelowercase.

Sebelum Lowercase	Sesudah Lowercase
RT @KPU_ID: Bergugurannya permohonan Pemohon pada sidang pembacaan putusan PHPU Pileg 2019 hari kedua Rabu (7/8) akibat tidak jelas atau ka...	rt @kpu_id: bergugurannya permohonan pemohon pada sidang pembacaan putusan phpu pileg 2019 hari kedua rabu (7/8) akibat tidak jelas atau ka...
code	
tweet=tweet.lower()	

Gambar 4.2. Proses Lowercase

b) Tokenisasi

Tokenisasi dilakukan dengan fungsi bawaan python dan memisahkan tiap kata menjadi berdiri

sendiri, file yang dihasilkan dari proses ini adalah sheet PreTokenization. Untuk contoh dan kode pemrograman ada pada Gambar 4.3.

Sebelum Tokenisasi	Sesudah Tokenisasi
rt @kpu_id: bergugurannya permohonan pemohon pada sidang pembacaan putusan phpu pileg 2019 hari kedua rabu (7/8) akibat tidak jelas atau ka...	['rt', '@kpu_id', ':', 'bergugurannya', 'permohonan', 'pemohon', 'pada', 'sidang', 'pembacaan', 'putusan', 'phpu', 'pileg', '2019', 'hari', 'kedua', 'rabu', '(', '7/8', ')', 'akibat', 'tidak', 'jelas', 'atau', 'ka', ', ...']
code	
tweet =tkz.tokenize(tweet)	

Gambar 4.3. Proses Tokenisasi

c) Stopwords

Stopword dilakukan dengan bantuan corpus dari library Nltk dan tambahan stopword dari peneliti. Dilakukan juga penghilangan beberapa hal yang dianggap akan mengganggu nantinya seperti hastag, link dan sebagainya, file yang dihasilkan dari proses ini adalah sheet PreStopwords. Untuk contoh dan kode pemrograman ada pada Gambar 4.4.

Sebelum Tokenisasi	Sesudah Tokenisasi
['rt', '@kpu_id', ':', 'bergugurannya', 'permohonan', 'pemohon', 'pada', 'sidang', 'pembacaan', 'putusan', 'phpu', 'pileg', '2019', 'hari', 'kedua', 'rabu', '(', '7/8', ')', 'akibat', 'tidak', 'jelas', 'atau', 'ka', ', ...']	['bergugurannya', 'permohonan', 'pemohon', 'sidang', 'pembacaan', 'putusan', 'phpu', 'pileg', 'rabu', 'akibat']
code	
stopwords = nltk.corpus.stopwords.words('indonesian')	
token = tokenfiltering(token)	
if re.search('[a-zA-Z]', token) and (token not in stopwords):	
dataoutput.append(token)	

Gambar 4.4. Proses Stopword

d) Stemming

Stemming dilakukan seluruhnya oleh library bernama Sastrawi, file yang dihasilkan dari proses ini adalah sheet PreStemming. Untuk contoh dan kode pemrograman ada pada Gambar 4.5.

Sebelum Tokenisasi	Sesudah Tokenisasi
['bergugurannya', 'permohonan', 'pemohon', 'sidang', 'pembacaan', 'putusan', 'phpu', 'pileg', 'rabu', 'akibat']	['gugur', 'mohon', 'mohon', 'sidang', 'baca', 'putus', 'phpu', 'pileg', 'rabu', 'akibat']
code	
factory = StemmerFactory()	
stemmer = factory.create_stemmer()	
hasil=stemmer.stem(data)	

Gambar 4.5. Proses Stemming

Berikut adalah waktu yang dibutuhkan untuk menyelesaikan setiap proses sesuai dengan jumlah dokumen dalam Tabel 4.1.

Tabel 4.1. Perbandingan Waktu Preprocessing per Jumlah Data

Jumlah Dokumen	Proses	Waktu yang Dibutuhkan (detik)
100	Lowercase	0.22
	Tokenisasi	0.22
	Stopword	0.24
	Stemming	24.17
1000	Lowercase	0.72
	Tokenisasi	0.82
	Stopword	1.21
	Stemming	138.08
6000	Lowercase	3.04
	Tokenisasi	3.63
	Stopword	5.43
	Stemming	454.53

4.2.3 Latent Dirichlet Allocation

Proses pengerjaan LDA sebagian besar dikerjakan oleh Gensim library proses pembuatan model LDA dimulai dengan :

a) Pembuatan kamus dan corpus

File yang dihasilkan dari preprocessing (PreStemming) berupa list kata-kata bersih, kata-kata ini akan diolah sebagai bahan utama, untuk pembuatan kamus dilakukan filterasi lagi dengan ketentuan pengujian untuk 100 data menghilangkan kata yang muncul kurang dari 3 dokumen, untuk 1000 data menghilangkan kata yang muncul kurang dari 20 dokumen dan untuk 6000 data menghilangkan kata yang muncul kurang dari 120 dokumen.

Kamus merupakan kumpulan kata-kata yang dirasa unik oleh Gensim, untuk contoh dan kode pemrograman pada Gambar 4.6.

Corpus disini adalah pemberian angka random untuk setiap katanya oleh Gensim, setiap kata memiliki nomernya masing-masing, nomer ini yang nantinya akan digunakan untuk proses perhitungan lanjutan oleh LDA. Untuk contoh dan kode pemrograman pada Gambar 4.7.

Contoh Kamus
Dictionary(112 unique tokens: ['akibat', 'baca', 'gugur', 'mohon', 'phpu']...)
Kode Program
id2word = gensim.corpora.Dictionary(datafordic)
id2word.filter_extremes(no_below=2,no_above=0.9)

Gambar 4.6. Kamus yang Digunakan untuk LDA

Contoh Corpus
[[0, 1), (1, 1), (2, 1), (3, 2), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1)], [(9, 1), (10, 1)], [(5, 1), (11, 1), (12, 1), (13, 1), (14, 1)], [(5, 1), (11, 1), (14, 1), (15, 1), (16, 1)], [(17, 1), (18, 1), (19, 1)], [(7, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1)], [(0, 1), (1, 1), (2, 1), (3, 2), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1)], [(0, 1), (1, 1), (2, 1), (3, 2), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1)], [(0, 1), (1, 1), (2, 1), (3, 4), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (29, 1), (30, 1)], [(16, 1), (31, 1), (32, 1)], [(18, 1), (33, 1), (34, 1), (35, 1), (36, 1), (37, 1)], [(32, 1), (38, 1), (39, 1), (40, 1), (41, 1), (42, 1), (43, 1)], [(24, 1), (38, 1), (44, 1), (45, 1), (46, 1), (47, 1)], [(5, 1), (38, 1), (48, 1), (49, 1), (50, 1), (51, 1), (52, 1)], [(38, 1), (53, 1)], [(5, 1), (11, 1), (54, 1)], [(23, 1), (24, 1), (38, 1), (55, 1), (56, 1)], [(15, 1), (19, 1), (38, 1), (43, 1), (57, 1)], [(48, 1), (51, 1), (58, 1), (59, 1), (60, 1), (61, 1), (62, 1)], [(48, 1), (51, 1), (58, 1), (59, 1), (60, 1), (61, 1), (62, 1)], [(63, 1), (64, 1), (65, 1)], [(19, 1), (38, 1), (48, 1), (66, 1), (67, 1), (68, 1)], [(18, 1), (19, 1), (69, 1), (70, 1)], [(53, 1), (71, 1)], [(6, 1), (24, 1), (51, 1), (54, 1), (65, 1), (71, 2)], [(7, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1)], [(9, 1), (24, 1), (38, 1), (72, 1)], [(32, 1), (39, 1), (40, 1), (41, 1), (42, 1), (43, 1), (73, 1)], [(2, 1), (11, 1), (14, 1), (24, 1), (38, 1)], [(33, 1), (34, 1), (35, 1), (36, 1), (37, 1), (60, 1), (74, 1)], [(32, 1)], [(43, 1), (60, 1), (75, 1), (76, 1), (77, 1)], [(18, 1), (26, 1), (78, 1)], [(9, 1), (18, 1), (72, 1), (79, 1)], [(24, 1), (60, 1), (70, 1), (80, 1)], [(55, 1), (81, 1), (82, 1), (83, 1)], [(16, 1), (49, 1), (84, 1)], [(53, 1)], [(7, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1)], [(45, 1), (46, 1), (85, 1), (86, 1)], [(7, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1)], [(19, 1), (73, 1)], [(24, 1), (38, 1), (46, 1), (86, 1)], [(15, 1), (19, 1), (57, 1), (60, 1)], [(7, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1)], [(19, 1), (73, 1), (87, 1), ...]
Kode Program
corpus = [id2word.doc2bow(text) for text in corpustext]

Gambar 4.7. Corpus yang Digunakan untuk LDA

b) Pemodelan LDA

Pada tahapan ini LDA mengolah corpus, jumlah dokumen, jumlah topik seperti yang di gambarkan dalam visualisasi LDA Gambar 2.1., dari ketiga bahan tersebut LDA menentukan untuk representasi dari setiap topik kata-kata dan nilai dari hasil perhitungan dengan rumus pada persamaan (2) dan (3) pada bab III, pada kesempatan ini peneliti menggunakan kombinasi pengujian jumlah topik 10 dan setting-an lainnya default (iterations, alpha(auto), chunksize dsb) berikut contoh hasil model LDA yang dihasilkan dan kode pemrogramannya pada Gambar 4.8. dan untuk perbandingan waktu untuk setiap percobaan terdapat

pada Tabel 4.2. pemodelan juga disimpan pada sheet HasilLdaModel.

Model LDA 10Topik
0.412**menang" + 0.216**adil" + 0.216**curang" + 0.020**langgar" + 0.020**kader" + 0.020**ulama" + 0.020**gugat" + 0.020**rekonsiliasi" + 0.020**utang" + 0.020**koruptor"
1.0.512**adil" + 0.268**menang" + 0.024**langgar" + 0.024**gugat" + 0.024**curang" + 0.024**kader" + 0.024**ulama" + 0.024**rekonsiliasi" + 0.024**koruptor" + 0.024**utang"
2.0.412**menang" + 0.216**rekonsiliasi" + 0.216**langgar" + 0.020**curang" + 0.020**kader" + 0.020**gugat" + 0.020**adil" + 0.020**ulama" + 0.020**utang" + 0.020**syariah"
3.0.580**menang" + 0.160**ulama" + 0.122**kader" + 0.084**rekonsiliasi" + 0.008**langgar" + 0.008**gugat" + 0.008**curang" + 0.008**adil" + 0.008**utang" + 0.008**koruptor"
4.0.524**menang" + 0.048**langgar" + 0.048**gugat" + 0.048**curang" + 0.048**adil" + 0.048**kader" + 0.048**ulama" + 0.048**utang" + 0.048**rekonsiliasi" + 0.048**koruptor"
5.0.307**syariah" + 0.307**koruptor" + 0.307**utang" + 0.010**menang" + 0.010**langgar" + 0.010**curang" + 0.010**gugat" + 0.010**ulama" + 0.010**kader" + 0.010**adil"
6.0.412**langgar" + 0.412**gugat" + 0.020**menang" + 0.020**curang" + 0.020**kader" + 0.020**ulama" + 0.020**adil" + 0.020**rekonsiliasi" + 0.020**utang" + 0.020**syariah"
7.0.258**kader" + 0.258**menang" + 0.109**utang" + 0.109**ulama" + 0.109**curang" + 0.109**langgar" + 0.010**gugat" + 0.010**adil" + 0.010**rekonsiliasi" + 0.010**koruptor"
8.0.523**curang" + 0.048**menang" + 0.048**langgar" + 0.048**gugat" + 0.048**ulama" + 0.048**kader" + 0.048**adil" + 0.048**rekonsiliasi" + 0.048**utang" + 0.048**koruptor"
9.0.268**menang" + 0.268**rekonsiliasi" + 0.268**gugat" + 0.024**langgar" + 0.024**curang" + 0.024**kader" + 0.024**ulama" + 0.024**adil" + 0.024**utang" + 0.024**koruptor"
Kode Pemrograman
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus, id2word=id2word, num_topics=int(topicnumber), alpha='auto', per_word_topics=True)

Gambar 4.8. Model LDA

Tabel 4.2. Perbandingan Waktu Pemodelan LDA

Jumlah Dokumen	Jumlah Topik	Waktu yang dibutuhkan (detik)
100	10	0.8
1000	10	1.18
6000	10	3.56

c) Penulisan hasil akhir

Setelah proses perhitungan selesai perlu dilakukan evaluasinya dan untuk mempermudah evaluasi dilakukan penulisan ulang hasil akhir agar mudah di evaluasi. Penulisan ulang ini berupa penulisan kembali data-data awal (tanggal, user dan tweet asli) ke dalam field excel lalu di buat dua file berbeda dengan nama sheet persentase topik dan hasil output dimana persentase topik berisi persentase topik mana saja dan berapa persen jumlahnya dalam satu tweet sedangkan hasil output adalah file yang akan di-review berisi topik utama dan keyword untuk setiap dokumen sehingga orang yang baca bisa lebih paham apa penentuan topik yang dihasilkan dari model LDA.

4.2.4 Evaluasi

Tahapan ini me-review bagaimana kinerja penentuan topik LDA bekerja dengan menggunakan rumus akurasi sebagaimana dijelaskan pada persamaan (9) pada Bab III, berikut hasil penilaiannya Table 4.3.

Tabel 4.3. Evaluasi LDA

Jumlah Dokumen	Jumlah Topik Salah	Jumlah Topik Benar	Skor Akhir
100	10	90	0,9
1000	99	901	0,901
6000	595	5405	0,900833333

4.3 Pembahasan

Proses pendeteksian topik dimulai dengan mengambil data, proses pengambilan data menggunakan API Tweepy dengan query “pemilu” jumlah data yang diambil adalah total 6000 dan dari total 6000 di pisah menjadi 3 pengujian dengan pengujian-1 100 data pengujian-2 1000 data dan pengujian-3 6000 data. Untuk data yang disimpan adalah date, user dan tweet. Semua data disimpan di file txt.

File txt diolah (preprocessing) dengan tujuan meningkatkan keakurasian dengan membuang kata dan mengubah ke bentuk awal gambar 4.2 hingga 4.5. Tahapan-tahapan preprocessingnya mulai dari lowercase (pengembalian ke bentuk huruf nonkapital) dengan fungsi python, tokenisasi (pemenggalan kata) menggunakan fungsi python, stopwords (penghilangan huruf non alfabet dan beberapa kata yang dianggap akan mengganggu hasil akhir) menggunakan stopwords dari library Nltk berbahasa Indonesia dan ditambahkan stopwords ciptaan peneliti, dan diakhiri dengan stemming (pengembalian ke bentuk awal) dengan menggunakan bantuan library Sastrawi.

Setelah data kata-kata didapatkan proses berlanjut dengan pembuatan corpus oleh Gensim dengan pemberian angka per kata oleh Gensim lalu disimpan dalam list variabel corpus. Dari data corpus tersebut LDA dengan settingan default dan jumlah topik 10 dihasilkan model untuk 10 topik, model tersebut berisi persentase terbentuknya topik dari kata-kata pembentuknya. Lalu dilakukan penentuan topik dengan cara menghitung persentase topik untuk setiap tweetnya, persentase tertinggi dianggap sebagai topik utama dari suatu tweet, dan karena ada kemungkinan tidak ada topik yang dominan pada suatu tweet maka untuk meningkatkan nilai akurasi dari evaluasi maka jika ada tweet yang topik utamanya memiliki persentase kurang dari 0.2 dianggap tidak dapat dideteksi.

Setelah topik ditentukan secara otomatis dilakukan evaluasi, evaluasi bersifat manual dan dilakukan oleh manusia dengan cara membandingkan tweet asli hasil topik yang ditentukan oleh sistem, lalu dihitung jumlah benar dan salah dari semua data. Dari total keseluruhan data dari 3 kali pengujian didapatkan rata-rata kebenaran 90%..

5. Kesimpulan dan Saran

5.1 Kesimpulan

Setelah dilakukan penelitian dengan judul Sistem Deteksi Topik Politik pada Twitter Menggunakan Algoritma Latent Dirichlet Allocation. Didapatkan dengan proses preprocessing dengan bantuan library Nltk untuk stopword dan Sastrawi untuk stemming, dan proses perhitungan LDA dengan bantuan Gensim dihasilkan kesimpulan bahwa dengan settingan dasar

LDA pada library Gensim dan jumlah topik 10 untuk pengujian 100, 1000 dan 6000 data dihasilkan rata-rata 90% benar untuk deteksi topik LDA, nilai tersebut juga masih dapat berubah tergantung seberapa bagus input ataupun optimalisasi yang dilakukan pada model sehingga LDA dirasa dapat digunakan untuk mendeteksi topik pada Twitter dengan topik politik.

5.2 Saran

Penulis memiliki beberapa saran untuk peneliti serupa yang berkaitan dengan algoritma Latent Dirichlet Allocation untuk penentuan topik pada Twitter atau teks pendek lainnya, yaitu:

1. Untuk pengambilan data menggunakan API Tweepy lebih baik dalam waktu dimana dalam 7 hari ke belakang topik banyak dibicarakan;
2. Untuk meningkatkan akurasi tweet yang diambil bisa diseleksi terlebih dahulu agar sesuai yang diinginkan, bisa juga meningkatkan hasil dengan tidak mengambil tweet yang me-retweet tweet lainnya;
3. Untuk library stopwords dan stemming Bahasa Indonesia yang ada (Nltk(Indonesia)) dan Sastrawi masih sering terdapat kurang tepatnya, alangkah baiknya menambahkan kondisi lebih/term pada kode pemrogramannya agar corpus yang dihasilkan lebih baik; dan
4. Untuk penentuan jumlah topik optimal, iterasi optimal, dan parameter lainnya dapat menggunakan looping dan fungsi bawaan Gensim perhitungan coherence dan mengambil nilai tertinggi dengan fungsi sebagai berikut, `CoherenceModel(model, text, dictionary, coherence='c_v')`.

Daftar Pustaka:

- Allan, J. (2002) Topic Detection and Tracking: Event-based Information Organization. New York: Springer Science+Business Media.
- Blei, D.M., Y. Ng, Andrew., & I.J, Michael. (2003). “Latent Dirichlet Allocation.” Machine Learning Research Volume 3, pp.993-1022.
- Budiardjo, M. (2007). Dasar-Dasar Ilmu Politik. Jakarta: PT Gramedia Pustaka Utama.
- Campbell, J.C., Hindle, A., & Stroulia, E. (2015). “Latent Dirichlet Allocation: Extracting Topics from Software Engineering Data.” The Art and Science of Analyzing Software Data, pp.139-159.
- Fitriasih, M. & Kusumaningrum, R. (2019). “Analisis Klasifikasi Opini Tweet pada Media Sosial Twitter Menggunakan Latent Dirichlet Allocation (LDA).” Seminar Nasional Teknologi Informasi dan Komunikasi 2019 (SENTIKA2019), PP.177-186
- Gil, P. (2019). “What is Twitter & How Does it Work?.” lifewire.com. Diambil dari <https://www.lifewire.com/what-exactly-is-twitter-2483331>. Diakses pada 22 juni 2019.

- Kengken, R.I. (2014). *Pemodelan Topik untuk Media Sosial Menggunakan Latent Dirichlet Allocation Studi Kasus: Analisis Tren Berita dalam Media Sosial*. Yogyakarta: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada
- Kusumaningrum, R., Wei, H., Manurung, R. & Murni, A. (2014) "Integrated Visual Vocabulary in Latent Dirichlet Allocation-Based Scene Classification for IKONOS Image." *Journal of Applied Remote Sensing*, pp.1-18.
- Pang, B., Lee, L. & Vaithyanathan, S. (2002). "Thumbs Up ? Sentiment Classification Using Machine Learning Techniques." *Proceddings Of The Conference On Empirical Methods In Natural Language Processing (EMNLP)*, pp.79-86.
- Pratomo, Y. (2019). "Untuk Pertama Kali Twitter Ungkap Jumlah Pengguna Harian," *Kompas.com*. Diambil dari <https://tekno.kompas.com/read/2019/02/09/11340027/untuk-pertama-kali-twitter-ungkap-jumlah-pengguna-harian>. Diakses 22 April 2019.
- Rosen, A. (2017). "Tweeting Made Easier." *blog.twitter.com*. Diambil dari https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html. Diakses pada 22 Juni 2019.
- Saputro, D.A. (2019). *Implementasi Metode Latent Dirichlet Allocation (LDA) untuk Menentukan Topik Teks Berita*. Kediri: Fakultas Teknik Informatika Universitas Nusantara PGRI