Pemodelan Topik Pada Dakwaan Tindak Pidana Pencurian Di Pengadilan Negeri Serang Menggunakan *Latent Dirichlet Allocation*

Aldho Meidy Tri Putra[1], Sajarwo Anggai[2], Taswanda Taryo[3]
[1,2,3]Program Studi Teknik Informatika S-2, Universitas Pamulang
e-mail: [1]aldhomeidy@gmail.com, [2]dosen02832@unpam.ac.id, [3]dosen02234@unpam.ac.id

***Abstrak*— Penelitian oleh** [1] **yang melakukan pemodelan topik terkait tindak pidana umum menghasilkan nilai koheren terbaik sebesar 0.65895 pada topik ke -4 dengan dataset berupa amar putusan hakim. Berbeda dengan penelitian tersebut, penelitian ini melakukan pemodelan topik terkait tindak pidana pencurian dengan dataset berupa teks dakwaan yang terkandung didalam dokumen putusan hakim. Metode pemodelan topik yang digunakan adalah LDA dengan pembobotan kata TF-IDF. Pada penelitian ini dilakukan ekstraksi fitur menggunakan algoritma N-Grams untuk minimal kata muncul sebanyak 3 dan 5 kata berjenis bigrams dan trigrams, dan dilakukan evaluasi dengan mencari nilai koheren dan perplexity optimal pada rentang 2 hingga 10 topik. Interpretasi topik dilakukan dengan menggunakan bantuan algoritma Word2Vec untuk mencari kata-kata yang mirip dengan kata penyusun suatu topik. Dari percobaan yang telah dilakukan, didapatkan hasil terbaik pada pemodelan tanpa ekstraksi fitur N-Grams dengan nilai koheren sebesar 0.619 untuk 8 topik. Nilai koheren yang didapatkan cukup baik, dan kata-kata yang dihasilkan dapat dengan mudah diinterpretasikan menjadi suatu topik tertentu. Nilai perplexity yang dihasilkan cenderung menunjukan nilai yang terus menurun dari 2 hingga 10 topik yang menandakan bahwa model cukup baik dalam memprediksi suatu data baru. Diharapkan dari penelitian ini dapat dilakukan pengembangan pada kategori tindak pidana yang lebih luas lagi.**

***Kata Kunci*— Pemodelan Topik, Tindak Pidana Pencurian, LDA, N-Gram, TF-IDF, Word2Vec**

I. PENDAHULUAN

Latent Dirichlet Allocation (LDA) adalah salah satu dari sekian metode dalam melakukan pemodelan topik yang cukup populer saat ini. Cukup banyak penelitian yang melakukan pemodelan topik menggunakan LDA dan menghasilkan performa yang cukup baik, salah satunya adalah penelitian yang dilakukan oleh [1] dengan judul "Implementasi Deteksi Topik Putusan Hakim dengan *Latent Dirichlet Allocation (LDA)".* Penelitian tersebut melakukan sebuah pemodelan topik terkait tindak pidana yang ditangani oleh Pengadilan Negeri Sleman pada Tahun 2016 hingga Januari 2020. Penentuan jumlah topik dilakukan dengan mencari nilai koheren tertinggi untuk jumlah topik pada rentang 1 hingga 10, dan didapatkan nilai koheren terbaik pada jumlah topik ke-4 dengan nilai koheren sebesar 0.65895. Penelitian tersebut menggunakan dataset berupa teks amar putusan hakim terkait klasifikasi tindak pidana secara umum seperti pencurian, narkotika, penggelapan, penipuan, perjudian dan lain sebagainya, dengan total data putusan sebanyak 2198 perkara. Penelitian tersebut bertujuan untuk melihat karakteristik perkara pidana di Pengadilan Negeri Sleman pada Tahun 2016 hingga Januari 2020.

Berbeda dengan penelitian yang dilakukan oleh [1] sebelumnya, pada penelitian ini bertujuan untuk mengetahui topik yang dibahas pada teks dakwaan perkara tindak pidana yang diputus oleh Pengadilan Negeri Serang sejak Tahun 2022 hingga Tahun 2024. Kategori tindak pidana yang difokuskan pada penelitian ini terkhusus pada tindak pidana pencurian saja. Data yang digunakan untuk pemodelan pada penelitian ini juga berbeda dengan penelitian yang dilakukan oleh [1] sebelumnya. Penelitian ini menggunakan data teks dakwaan Jaksa Penuntut Umum yang terkandung didalam dokumen putusan hakim, dimana data dakwaan tersebut secara umum berisi kronologi terjadinya peristiwa suatu tindak pidana yang meliputi lokasi dan waktu peristiwa, media yang digunakan untuk melakukan tindak pidana kejahatan, barang bukti tindak pidana, akibat tindak pidana serta informasi – informasi lainnya yang berkaitan dengan kronologi peristiwa tindak pidana [2]. Hal tersebut yang mendasari pemilihan penggunaan teks dakwaan untuk melakukan pemodelan topik pada penelitian ini, sehingga terdapat kemungkinan informasi topik yang akan diperoleh akan lebih bervariasi.

Topik yang dihasilkan oleh LDA merupakan hasil interpretasi dari kumpulan kata-kata yang menyusunnya. Kata-kata yang dihasilkan merupakan hasil dari proses tokenizing, yaitu merubah suatu kalimat menjadi token yang dalam hal ini merupakan sebuah kata. Terkadang kata-kata yang terbentuk bukanlah merupakan suatu kata dalam konteks aslinya pada suatu dokumen, misalnya seperti "sepeda motor". Ketika dilakukan *tokenizing*, maka akan terbentuk dua buah kata, yaitu "sepeda" dan "motor". Padahal konteks dari kata tersebut merujuk kepada sebuah objek kendaraan sepeda motor, bukan objek sepeda dan objek motor. Maka dari itu, pada penelitian ini akan mengimplementasikan teknik N-Gram untuk mendapatkan konteks tersembunyi dari kata-kata didalam suatu kalimat. Interpretasi topik dilakukan dengan menginterpretasikan langsung sebuah topik yang muncul serta dengan menggunakan bantuan algoritma Word2Vec untuk mencari kata yang sejenis dengan kata-kata penyusun topik tersebut, sehingga akan membantu memudahkan dalam melakukan interpretasi topik. Salah satu parameter yang dibutuhkan dalam melakukan pemodelan topik, khususnya menggunakan algoritma LDA adalah jumlah topik. Dalam menentukan jumlah topik yang dimodelkan, penelitian ini menggunakan matriks perplexity dan nilai koheren untuk rentang topik 2 hingga 10 topik. Setiap top ik

akan menghasilkan nilai perplexity dan nilai koheren yang berbeda-beda, dan dari keseluruhan nilai tersebut akan dianalisis nilai perplexity dan koheren terbaik yang merujuk pada jumlah topik tertentu, dan jumlah topik tersebutlah yang akan digunakan sebagai parameter jumlah topik dalam pemodelan menggunakan LDA.

II. METODE PENELITIAN

Pada penelitian ini penulis menggunakan metode *Latent Dirichlet Allocation (LDA)* yang dikombinasikan dengan algoritma *Term Frequency-Inverse Document Frequency (TF-IDF)* untuk pembobotan kata, algoritma *N-Grams* untuk ekstraksi fitur dan *Word2Vec* untuk mencari kata-kata serupa dari kata penyusun topiknya.

A. *Term Frequency-Inverse Document Frequency (TF-IDF)*

TF-IDF merupakan metode untuk mempertimbangkan sepenting apa sebuah kata dalam dokumen yang merupakan bagian dari kumpulan dokumen (korpus) [3]. Metode TF-IDF merupakan gabungan dari *Term Frequency (TF)* dan *Inverse Document Frequency (IDF)*. *Term Frequency (TF)* merepresentasikan frekuensi kata t dalam dokumen d, sedangkan *Inverse Document Frequency (IDF)* mengukur berapa banyak informasi suatu istilah atau kata tersedia untuk dokumen dan dihitung dengan mengambil logaritma dari jumlah dokumen disebuah korpus N dibagi dengan jumlah dokumen yang mengandung t. TF-IDF dihitung dengan mencari nilai *TF (Term Frequency)* untuk menentukan banyaknya term pada suatu dokumen. Kemudian dilakukan perhitungan *DF (Document Frequency)* untuk mendapatkan jumlah dokumen dimana term tersebut muncul. Selanjutnya perhitungan *IDF (Inverse Document Frequency)* untuk mengurangi bobot suatu term jika kemunculannya tersebar luas pada setiap dokumen [4].

B. *N-Grams*

Konsep "n-gram" digunakan untuk merujuk ke urutan n item yang berdekatan dalam sampel teks. Item-item ini dapat berupa kata, karakter, atau subkata, tergantung pada tingkat perincian yang diperlukan untuk analisis [5]. Berdasarkan jumlah katanya, N-Grams terbagi menjadi beberapa jenis, yaitu Unigram, Bigrams, Trigrams dan seterusnya sesuai dengan jumlah "n" didalam N-Grams. Unigram terdiri dari satu kata, Bigram dua kata dan Trigram tiga kata. Ketiga jenis N-Grams tersebut umum digunakan dalam proses analisis bahasa alami untuk memahami hubungan antar kata didalam teks [6]. Sebagai contoh, terdapat sebuah kalimat "saya pergi naik bus pariwisata Damri", maka dengan N-Grams akan menghasilkan pemecahan kata sebagai berikut:

1. Unigram
   ["saya", "pergi", "naik", "bus", "pariwisata", "damri"]
2. Bigram
   ["saya_pergi", "pergi_naik", "naik_bus", "bus_pariwisata", "pariwisata_damri"]
3. Trigram
   ["saya_pergi_naik", "pergi_naik_bus", "naik_bus_pariwisata", "bus_pariwisata_damri"]

Jika dilihat, kata "bus", "pariwisata" dan "Damri" pada Unigram memiliki makna masing-masing. Pada Bigrams, kata "bus" dan "pariwisata" yang awalnya memiliki makna tersendiri, namun apabila digabungkan menjadi "bus_pariwisata" memiliki makna baru, yaitu sebuah bus yang khusus digunakan untuk pariwisata. Kemudian pada Trigrams, kata "bus", "pariwisata" dan "Damri" apabila digabungkan menjadi "bus_pariwisata_Damri" memiliki makna baru yaitu sebuah bus yang khusus digunakan untuk pariwisata dengan merek Damri. Penelitian ini akan mengimplementasikan N-Grams berjenis Bigrams dan Trigrams untuk mengekstraksi fitur yang terkandung didalam teks dakwaan nantinya, dan membandingkan hasil pemodelan dari fitur yang tersekstrak.

C. *Latent Dirichlet Allocation (LDA)*

LDA pertama kali dikenalkan oleh [7] dalam jurnalnya yang berjudul "Latent Dirichlet Allocation", yaitu merupakan sebuah metode topic modelling yang digunakan untuk menentukan pola pada sebuah dokumen yang dapat menghasilkan topik. Ide dasar dari LDA adalah menganggap bahwa dokumen direpresentasikan sebagai campuran dari beberapa topik, dimana setiap topiknya dicirikan oleh kata-kata yang terdistribusi. LDA mengidentifikasi informasi topik tersembunyi dalam koleksi dokumen besar menggunakan pendekatan bag of words (cara representasi data teks) yang memperlakukan setiap dokumen sebagai vektor jumlah kata dan direpresentasikan sebagai distribusi probabilitas atas beberapa topik, sementara setiap topik direpresentasikan sebagai distribusi probabilitas atas sejumlah kata [8].

Salah satu parameter yang paling penting dalam pemodelan topik menggunakan LDA adalah menentukan jumlah topik yang akan dimodelkan. Menurut [9], secara umum terdapat empat macam cara dalam menentukan jumlah topik yang akan dimodelkan menggunakan LDA, antara lain:

1. Melalui pengalaman subjektif, mayoritas peneliti menggunakan metode ini untuk menentukan jumlah topik yang akan dimodelkan;

2. Menghitung nilai *perplexity* dengan rentang jumlah yang berbeda;
3. Menghitung nilai fungsi *likelihood*;
4. Melalui metode non-parametrik, yaitu sebuah metode HDP berdasarkan proses *Dirichlet.*

Dari keempat macam cara tersebut, pengalaman subjektif menjadi salah satu cara yang digunakan oleh mayoritas peneliti dengan topik penelitian sejenis. Maka dari itu, pada penelitian ini jumlah topik yang dimodelkan akan ditentukan oleh peneliti dengan memilih rentang jumlah topik tertentu, yang dikombinasikan dengan teknik lain yaitu mencari nilai *perplexity* dan nilai koheren terbaik dari rentang topik yang telah ditentukan.

D. *Word2Vec*

Menurut Makolov didalam [10] *Word2Vec* merupakan salah satu tipe metode dari *Word Embedding* yang dibuat oleh Google. *Word2Vec* menggunakan vektor representasi distribusi dari sebuah kata mewakili makna dari kata tersebut. Terdapat 2 (dua) arsitektur yang digunakan pada *Word2Vec*, yaitu *Continous Bag-of-Words (CBOW)* dan *Skip- gram*. CBOW menghasilkan kata dengan memprediksi kata berdasarkan kata disekitarnya, sedangkan Skip-gram menghasilkan representasi vektor dengan cara memprediksi kata yang ada disekitar sebuah kata [3]. Pada penelitian ini akan menerapkan *Word2Vec* dengan arsitektur CBOW untuk mencari kata-kata yang mirip dengan kata-kata penyusun topiknya.

III. HASIL DAN PEMBAHASAN

A. *Preprocessing*

Preprocessing data dilakukan sebelum data diproses menggunakan metode yang dipakai. Sebagai contoh untuk preprocessing data, akan digunakan cuplikan teks dakwaan pada perkara nomor 2/Pid.B/2024/PN SRG atas nama Rico Yudianto als Oo Bin Partono Alm didalam dataset yang digunakan. Dalam hal ini, teks dakwaan akan melalui 7 tahapan preprocessing, yaitu:

1. *Case Folding*
   Pada tahap ini akan dilakukan penyeragaman jenis huruf seluruh teks dakwaan menjadi huruf kecil
2. *Cleaning*
   Pada tahap ini, seluruh tanda baca, angka dan simbol yang terdapat didalam teks akan dihilangkan dengan cara menggantinya atau replace dengan tanda spasi agar kata sambung.
3. *Remove Single Character*
   Ada kalanya didalam teks dakwaan tersebut terdapat karakter tunggal yang tidak memiliki makna apapun, sehingga untuk mengurangi beban pemrosesan data, pada tahap ini dilakukan penghapusan karakter tunggal
4. *Remove Multiple Space*
   Proses *Cleaning* dan *Remove Single Character* sebelumnya menghasilkan teks dakwaan yang memiliki lebih dari satu spasi (multiple space) antar katanya, sehingga akan mempengaruhi hasil pada tahap *Word Tokenizing* nantinya. Maka dari itu perlu dilakukan penghapusan untuk spasi lebih dari satu
5. *Word Tokenizing*
   Selanjutnya dilakukan proses *Word Tokenizing* untuk memecah teks dakwaan menjadi bentuk token yang dalam hal ini adalah kata
6. *Stopwords Removal*
   Proses *Stopwords Removal* digunakan untuk menghapus seluruh kata – kata yang tidak memiliki makna seperti ‘yang’, ‘dan’, ‘kemudian’ dan sebagainya
7. *Stemming*
   Proses Stemming digunakan untuk mengubah kata atau token menjadi bentuk dasarnya.

Sebagai contoh berikut ini adalah perbedaan teks dakwaan nomor 2/Pid.B/2024/PN SRG atas nama Rico Yudianto als Oo Bin Partono Alm sebelum dan sesudah dilakukan *preprocessing* yang dapat dilihat pada Tabel

Tabel 1 Hasil Preprocessing

Sebelum	Sesudah
Bahwa Terdakwa RICO YUDIANTO Als OO Bin PARTONO (Alm) pada hari Kamis tanggal 14 September 2023 sekira pukul 05.00 WIB, atau setidak-tidaknya pada suatu waktu di bulan September 2023 atau setidak-tidaknya pada tahun 2023, bertempat di Rumah yang beralamat di Lingkungan Sukadamai RT.002 RW.007 Kelurahan Panggung Rawi Kecamatan Jombang Kota Cilegon Provinsi Banten atau setidak-tidaknya pada suatu tempat yang masih	['dakwa', 'rico', 'yudianto', 'als', 'oo', 'bin', 'partono', 'alm', 'kamis', 'tanggal', 'september', 'sekira', 'wib', 'tidak', 'tidak', 'september', 'tidak', 'tidak', 'tempat', 'rumah', 'alamat', 'lingkung', 'sukadamai', 'rt', 'rw', 'lurah', 'panggung', 'rawi', 'camat', 'jombang', 'kota', 'cilegon', 'provinsi', 'banten', 'tidak', 'tidak', 'daerah', 'hukum', 'pengadilan', 'negeri', 'serang', 'wenang', 'periksa', 'adil', 'ambil', 'barang', 'milik', 'orang', 'maksud', 'milik', 'lawan', 'hukum', 'malam', 'rumah',

termasuk dalam daerah hukum Pengadilan Negeri Serang yang berwenang memeriksa dan mengadili, mengambil barang sesuatu, yang seluruhnya atau sebagian milik orang lain, dengan maksud untuk dimiliki secara melawan hukum, pada waktu malam dalam sebuah rumah atau di pekarangan tertutup yang ada rumahnya, yang dilakukan oleh orang yang ada disitu tanpa diketahui atau tanpa dikehendaki yang berhak, yang untuk masuk ke tempat melakukan kejahatan, atau untuk sampai pada barang yang diambilnya dilakukan dengan merusak, memotong, atau memanjat atau dengan memakai anak kunci palsu, perintah palsu atau pakaian jabatan palsu yang dilakukan oleh Terdakwa ...	'pekarang', 'tutup', 'rumah', 'orang', 'situ', 'hendak', 'hak', 'masuk', 'jahat', 'barang', 'ambil', 'rusak', 'potong', 'panjat', 'pakai', 'anak', 'kunci', 'palsu', 'perintah', 'palsu', 'pakai', 'jabat', 'palsu',...]

B. *Implementasi N-Grams*

N-Grams yang diimplementasikan berjenis *bigrams* dan *trigrams*, yang mana *trigrams* merupakan hasil proses ekstrak lanjutan dari *bigrams*, dan *bigrams* itu sendiri merupakan ekstrak lanjutan dari *unigrams*. Sehingga perlu dilakukan ekstraksi secara bertahap mulai dari *unigrams*, bigrams hingga *trigrams*. *Unigrams* sendiri akan menghasilkan informasi yang tersusun dari satu buah kata saja. Token atau kata yang terbentuk pada proses *Word Tokenizing* sebelumnya tersusun dari satu buah kata saja, sehingga tidak perlu dilakukan ekstrak menggunakan *unigrams*. Hasil *Word Tokenizing* sebelumnya dilanjutkan proses ekstraksi menggunakan bigrams dan dari hasil bigrams tersebut diproses kembali menggunakan trigrams. Penelitian ini membandingkan hasil pemodelan dari ekstraksi fitur *Bigrams* dan *Trigrams* dengan masing-masing minimal kemunculan kata sebanyak 3 dan 5 kali, sehingga proses ini akan dijalankan sebanyak 4 kali untuk parameter min_count sebanyak 3 dan 5 serta variasi terakhir adalah token yang dihasilkan tanpa dilakukan ekstraksi dengan N-Grams. Sehingga akan terdapat 5 macam variasi yang akan dimodelkan, yaitu:

1. *Bigrams* dengan minimal kemunculan 3
2. *Bigrams* dengan minimal kemunculan 5
3. *Trigrams* dengan minimal kemunculan 3
4. *Trigrams* dengan minimal kemunculan 5
5. Tanpa N-Grams

Berikut ini adalah contoh token atau kata yang dihasilkan oleh ekstraksi fitur menggunakan N-Grams yang dapat dilihat pada Tabel 2.

Tabel 2 Perbandingan Hasil Token N-Grams

Tanpa N-Grams	***Bigrams, min_count=3***	***Bigrams, min_count=5***	***Trigrams, min_count=3***	***Trigrams, min_count=5***
sepeda	iqbal_junior	unit_handphone	saksi_abdul_hamid	sepeda_motor_hasil_curi
curi	kota_cilegon	merk_samsung	bin_alm_husen	kamis_tanggal_oktober_sekira
saksi	agus_susanto	warna_hitam	jumat_tanggal_oktober_sekira	kragilan_kabupaten_serang
fikri	anggota_polisi	imei_imei	senin_tanggal_oktober_sekira	serang_wenang_periksa_adil
kunci	resor_cilegon	halaman_rumah	tempat_area_pt_doosan	sepeda_motor_hasil
rumah	alm_husen	desa_situterate	indonesia_alamat_proyek_pltu	ambil_rusak_potong_panjat
halaman	heavy_industries	sepeda_motor	jawa_lurah_suralaya_camat	pakai_anak_kunci_palsu
motor	unit_sepeda	kabupaten_serang	serang_wenang_adil_perkara	perintah_palsu_pakai_jabat
jendela	motor_honda	toko_alfamart	pulomerak_kota_cilegon	hukum_malam_rumah_pekarang
kunci	nomor_polisi	pengadilan_negeri	tidak_tidak_daerah_hukum	tutup_rumah_orang_situ

*C*orpus baru dari hasil bigrams akan berisi satu kata yang merupakan token *corpus unigrams* serta dua kata yang merupakan hasil ekstraksi dari *bigrams*. Sedangkan pada *trigrams*, *corpus* baru yang terbentuk akan berisi satu dan dua kata dari *corpus* hasil *bigrams*, karena *trigrams* merupakan ekstraksi lanjutan dari *bigrams*. Sehingga, *trigrams* akan menghasilkan token baru yang tersusun dari satu, dua tiga dan empat kata.

C. Implementasi TF-IDF

Corpus yang dihasilkan 5 (lima) variasi N-Grams sebelumnya dilakukan pembobotan kata menggunakan TF-IDF. TF didapatkan dengan menghitung jumlah frekuensi suatu kata atau *term* pada suatu dokumen didalam *corpus*. Sedangkan DF didapatkan dengan menghitung jumlah dokumen yang mengandung suatu kata atau *term* tertentu didalam *corpus* dan

dilakukan perhitungan *log* dari pembagian total dokumen dengan nilai DF untuk mendapatkan nilai IDF. Berikut ini adalah contoh sampel bobot suatu kata yang dihasilkan dari perhitungan TF-IDF yang dapat dilihat pada Tabel 3.

Tabel 3 Contoh Hasil TF-IDF

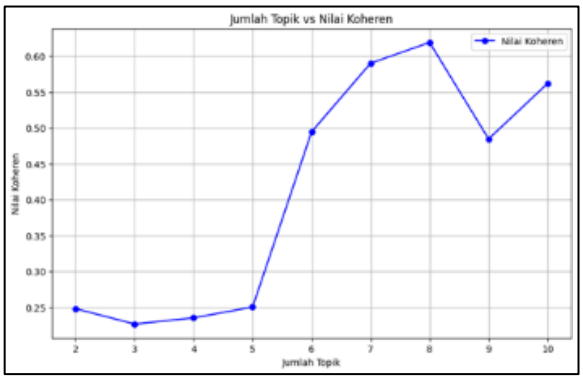
Term	**Bobot**
alm	0.03495635571734482
cilegon	0.0233464415059778
rumah	0.0012114538868503492
alfastio	0.14497895577026576
doosan	0.3803362339308228
heavy	0.4349368673107973
indonesia	0.24604970979849033
industries	0.4349368673107973
mobil	0.011223962886108522
pt	0.11176318566615794
steel	0.11620070021608873
Dst…	

D. *Implementasi LDA dan Evaluasi Model*

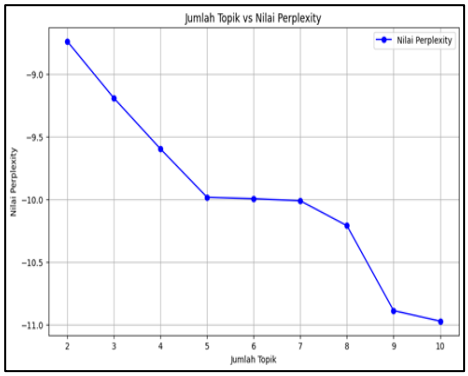
Proses evaluasi model dilaksanakan dengan mengimplementasikan pemodelan menggunakan LDA secara *looping* atau berulang mulai dari 2 topik hingga 10 topik, dan pada masing-masing jumlah topik tersebut akan dihitung nilai koheren dan *perplexity* nya sehingga didapatkan nilai koheren dan *perplexity* untuk setiap jumlah topiknya. Penelitian ini akan melihat performa yang dihasilkan oleh pemodelan topik menggunakan LDA dengan beberapa variasi N-Grams. Proses pemodelan dilakukan sebanyak lima kali sesuai dengan variasi ekstraksi fitur yang akan dicoba, sehingga akan menghasilkan lima macam grafik dan tabel perhitungan nilai koheren dan *perplexity* yang dapat dilihat pada Gambar 1 sampai Gambar 10 dan Tabel 4 sampai Tabel 8.

Tabel 4 Nilai Koheren dan Perplexity tanpa N-Grams

Topik ke	Nilai Koheren	Nilai Perplexity
2	0.2481172182747341	-8.735781925626647
3	0.2266344540392855	-9.189166194656424
4	0.23529213418043385	-9.594946304384973
5	0.25036981069851355	-9.981157909128031
6	0.49463711571698554	-9.992608243612613
7	0.5902331803582618	-10.009646481234347
8	0.6191725889023254	-10.207896652555734
9	0.4847073141834514	-10.885642056388898
10	0.561999193483636	-10.970605351987782



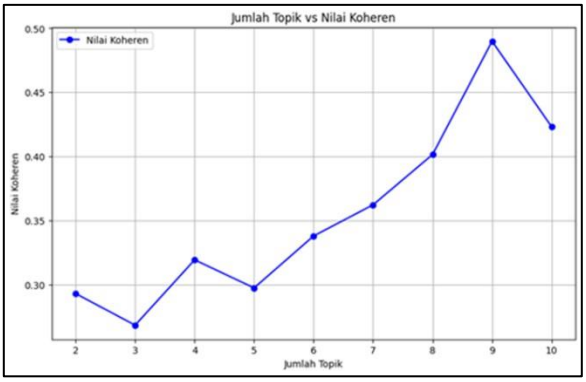
Gambar 1 Grafik Topik vs Koheren tanpa N-Grams



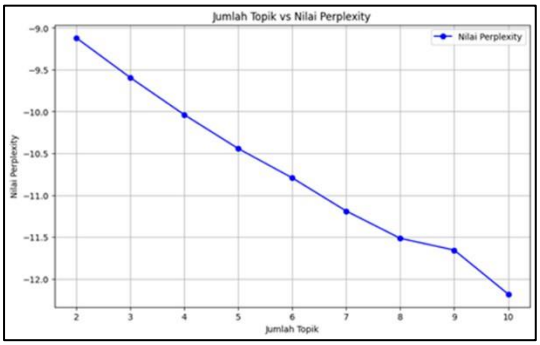
Gambar 2 Grafik Topik vs Perplexity tanpa N-Grams

Tabel 5 Nilai Koheren dan Perplexity dengan Bigrams, min count=3

Topik ke	Nilai Koheren	Nilai Perplexity
2	0.2930608640604421	-9.118702275983486
3	0.2684649352976791	-9.594191997859829
4	0.31938323157997073	-10.038139015150263
5	0.29748773456693667	-10.442876794083855
6	0.33800922404201267	-10.793546808910566
7	0.36230528644853477	-11.18967095057282
8	0.40151940782141765	-11.515848072715537
9	0.48983592936226444	-11.656702938400672
10	0.42325963470313327	-12.183653959465417



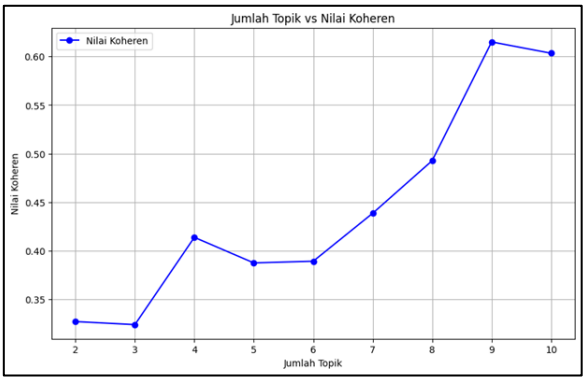
Gambar 3 Grafik Topik vs Koheren Bigrams, min count=3



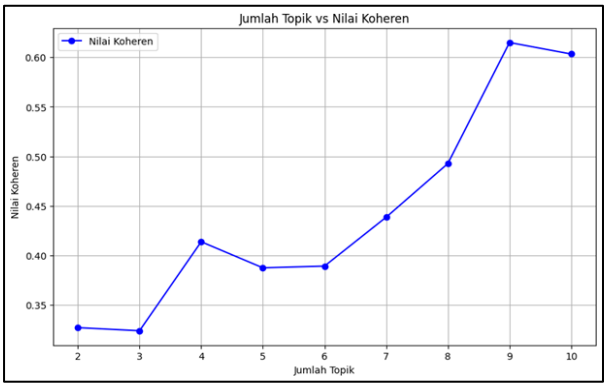
Gambar 4 Grafik Topik vs Perplexity Bigrams, min count=3

Tabel 6 Nilai Koheren dan Perplexity dengan Bigrams, min count=5

Topik ke	Nilai Koheren	Nilai Perplexity
2	0.3270696083174687	-9.057968740105217
3	0.32391944064756334	-9.53014761996805
4	0.41376811000587965	-9.90693963198797
5	0.3874405433653648	-10.257830423678078
6	0.3891945152139284	-10.666187285233962
7	0.4385752288426291	-10.955978545907524
8	0.49270501869770844	-11.138082720515584
9	0.6149073952628307	-10.995415440389538
10	0.6032997795174979	-11.246930974294298



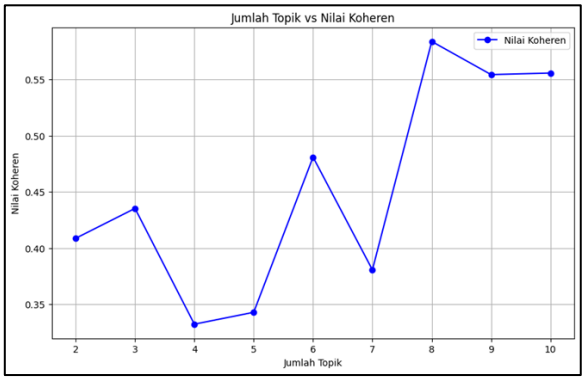
Gambar 5 Grafik Topik vs Koheren Bigrams, min count=5



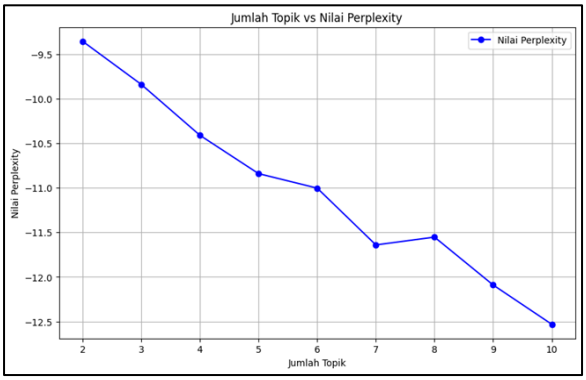
Gambar 6 Grafik Topik vs Perplexity Bigrams, smin count=5

Tabel 7 Nilai Koheren dan Perplexity dengan Trigrams, min count=3

Topik ke	Nilai Koheren	Nilai Perplexity
2	0.408635181254814	-9.355009664159036
3	0.43526622277830923	-9.839032460247886
4	0.33216327901648474	-10.409913187407362
5	0.3427361214273598	-10.840286905400198
6	0.48093882998291443	-11.002838978386375
7	0.3805615898140724	-11.640258283395957
8	0.5839453188686285	-11.552816808203977
9	0.5544297486988508	-12.08997252639737
10	0.5558473810742034	-12.53116435607626



Gambar 7 Grafik Topik vs Koheren Trigrams, min count=3

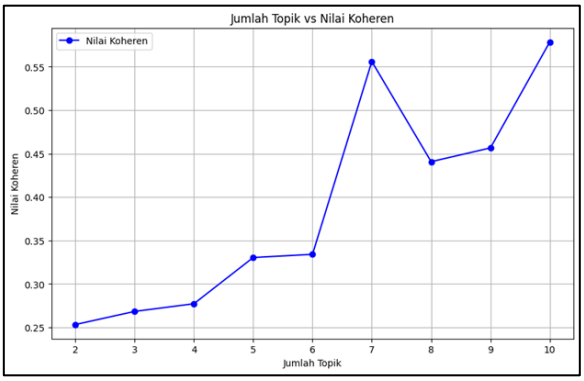


Gambar 8 Grafik Topik vs Perplexity Trigrams, min count=3

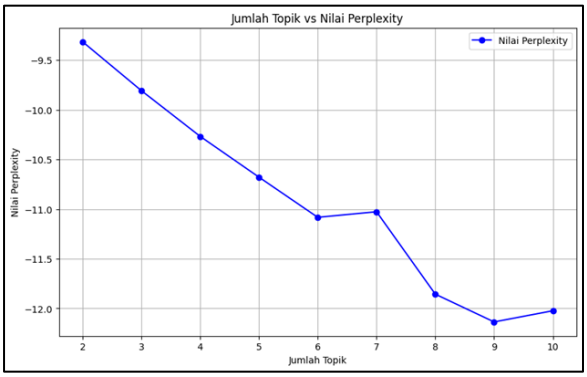
Tabel 8 Nilai Koheren dan Perplexity dengan Trigrams, min count=5

Topik ke	Nilai Koheren	Nilai Perplexity
2	0.253296448225739	-9.3160215228842
3	0.2684016575986146	-9.807898572669608
4	0.2772368110845515	-10.268262050510412
5	0.3304411139645142	-10.678886794441805
6	0.33412099419064806	-11.083132982265369
7	0.5560709506275436	-11.026906850434285
8	0.44075913316639	-11.856698548118906
9	0.45657577564061547	-12.136193214680944

10	0.5784338723795288	-12.023528600442194



Gambar 9 Grafik Topik vs Koheren Trigrams, min count=5



Gambar 10 Grafik Topik vs Perplexity Trigrams, min count=5

Dapat dilihat, nilai koheren dan *perplexity* yang dihasilkan cukup bervariasi. Untuk nilai *perplexity* dapat disimpulkan dari keseluruhan teknik menghasilkan nilai *perplexity* yang cenderung selalu menurun mulai dari topik 2 hingga 10. Sedangkan nilai koheren menghasilkan nilai yang naik turun disetiap topik dan tekniknya. Dari setiap variasi tersebut akan dilakukan pemodelan topik menggunakan LDA dengan jumlah topik yang memiliki nilai koheren tertinggi (blok warna hijau pada setiap tabel). Topik yang dihasilkan dari pemodelan dapat dilihat pada contoh hasil untuk pemodelan tanpa N-Grams pada Tabel 9 berikut.

Tabel 9 Hasil Pemodelan Topik tanpa Ekstraksi Fitur dengan N-Grams

Topik Ke -	**Kata Penyusun Topik**
1	('handphone', 0.002957299), ('sdr', 0.0027210678), ('alan', 0.0027008643), ('pt', 0.0025319194), ('motor', 0.002426708), ('sepeda', 0.002258881), ('jin', 0.0022292896), ('merek', 0.00217138), ('iman', 0.002101991), ('muhamad', 0.0019735324)
2	('toko', 0.0018913377), ('motor', 0.0018525808), ('sdr', 0.00184836), ('akbar', 0.0017771891), ('joni', 0.0017685372), ('dpo', 0.0017627636), ('sepeda', 0.0017005363), ('sandi', 0.0014537161), ('suhada', 0.0014254997), ('kontra', 0.0013691034)
3	('sdr', 0.0031971326), ('korban', 0.0028961594), ('pt', 0.0027246661), ('mobil', 0.0024294637), ('nomor', 0.0023765604), ('kambing', 0.0023737196), ('motor', 0.002320844), ('sepeda', 0.0020473583), ('riki', 0.0019127575), ('imei', 0.0017768848)])
4	('sdr', 0.0042789425), ('als', 0.003482713), ('motor', 0.002649515), ('dpo', 0.00241765), ('sepeda', 0.002392642), ('cilegon', 0.0020635093), ('rudi', 0.002032109), ('atm', 0.0019563437), ('no', 0.0019058052), ('korban', 0.0018697439)])
5	('sdr', 0.0028965208), ('toko', 0.0019711494), ('handphone', 0.0019548298), ('als', 0.0019259994), ('korban', 0.0019077238), ('burung', 0.0018417371), ('rokok', 0.0017844068), ('dpo', 0.0017120325), ('alias', 0.0016983579), ('mobil', 0.0016622544)])
6	('korban', 0.0031731843), ('als', 0.0026553525), ('sdr', 0.0025889024), ('motor', 0.0023811655), ('sepeda', 0.0022478963), ('handphone', 0.0019887686), ('warung', 0.0019533099), ('muhamad', 0.001945475), ('buah', 0.001930889), ('dpo', 0.001930787)])
7	('als', 0.002032749), ('hendra', 0.0018954523), ('handphone', 0.0018134773), ('motor', 0.0017706455), ('merek', 0.0016879772), ('besi', 0.0016872898), ('dpo', 0.0016640893), ('buah', 0.001621414), ('sepeda', 0.0016025084), ('kamar', 0.0015041864)])
8	('sdr', 0.002855405), ('toko', 0.0023223255), ('sepeda', 0.0018726646), ('handphone', 0.0018506938), ('motor', 0.0018031723), ('hp', 0.0017147282), ('sumiati', 0.0016600571), ('ac', 0.0015993632), ('alfamart', 0.0015704519), ('tom', 0.0014495624)])

Dari topik yang dihasilkan tersebut, selanjutnya dilakukan interpretasi topik dengan melihat kata-kata yang menyusun setiap topik tersebut. Untuk memudahkan proses interpretasi, peneliti mencari kata-kata yang mirip untuk setiap kata penyusun topik tersebut menggunakan algoritma *Word2Vec*.

E. Implementasi *Word2Vec*

Pada penelitian ini *Word2Vec* digunakan untuk membantu interpretasi topik dengan cara mencari kata yang mirip dari kata-kata penyusun sebuah topik. Jenis arsitektur *Word2Vec* yang digunakan pada penelitian ini adalah *CBOW* dengan nilai *vector size* = 100, *window* = 5 dan minimal kemunculan kata = 2. Berikut ini adalah contoh hasil dari pencarian kata menggunakan *Word2Vec* untuk pemodelan tanpa ekstraksi fitur N-Grams yang dapat dilihat pada pada Tabel 10.

Tabel 10 Contoh Hasil Word2Vec untuk pemodelan tanpa N-Grams (TN)

Topik	Kata Penyusun	Kata Mirip (*Word2Vec*)
1	handphone, sdr, alan, pt, motor…	• 'handphone': 'hp' (0.8755), 'oppo' (0.8619), 'handhone' (0.7730), 'redmi' (0.7650), 'samsung' (0.7526), 'ak' (0.7348), 'infinix' (0.7332), 'as' (0.7259), 'laptop' (0.7177), 'rose' (0.7062) • 'sdr': 'dpo' (0.8662), 'ing' (0.8515), 'mi' (0.7770), 'iyan' (0.7635), 'andro' (0.7536), 'jhandi' (0.7483), 'oyok' (0.7432), 'herman' (0.7414), 'majid' (0.7392), 'irul' (0.7346) • 'alan': 'iman' (0.9593), 'jaenal' (0.9426), 'hilman' (0.9206), 'wahyudi' (0.8831), 'jamadi' (0.8547), 'rinaldi' (0.8239), 'jumadi' (0.8071), 'jumaedi' (0.8030), 'jin' (0.7760), 'ri' (0.7692) • 'pt': 'indonesia' (0.8240), 'doosan' (0.8207), 'steel' (0.8156), 'wilmar' (0.7920), 'pundi' (0.7881), 'heavy' (0.7868), 'uniwood' (0.7660), 'industries' (0.7571), 'langgeng' (0.7446), 'nabati' (0.7428) • 'motor': 'ride' (0.6757), 'yamaha' (0.6651), 'scoopy' (0.6613), 'vixion' (0.6591), 'mio' (0.6582), 'honda' (0.6564), 'kendara' (0.6407), 'ci' (0.6226), 'jupiter' (0.6160), 'titip' (0.6030) • …
2	toko, motor, sdr, akbar, joni…	• 'toko': 'indomaret' (0.8328), 'alfamart' (0.8304), 'aneka' (0.8116), 'ruko' (0.8045), 'frozen' (0.7608), 'darizki' (0.7333), 'filla' (0.7185), 'panunggulan' (0.6972), 'baby' (0.6966), 'gudang' (0.6864) • 'motor': 'ride' (0.6757), 'yamaha' (0.6651), 'scoopy' (0.6613), 'vixion' (0.6591), 'mio' (0.6582), 'honda' (0.6564), 'kendara' (0.6407), 'ci' (0.6226), 'jupiter' (0.6160), 'titip' (0.6030) • 'sdr': 'dpo' (0.8662), 'ing' (0.8515), 'mi' (0.7770), 'iyan' (0.7635), 'andro' (0.7536), 'jhandi' (0.7483), 'oyok' (0.7432), 'herman' (0.7414), 'majid' (0.7392), 'irul' (0.7346) • 'akbar': 'adam' (0.9274), 'transaksi' (0.9030), 'debet' (0.8745), 'nama' (0.8683), 'memo' (0.8465), 'via' (0.8416), 'biaya' (0.8269), 'brilink' (0.8101), 'bertulisakan' (0.8086), 'gumilar' (0.8016) • 'joni': 'adi' (0.8287), 'son' (0.8082), 'amran' (0.8031), 'win' (0.7929), 'erwin' (0.7888), 'yu' (0.7652), 'prabu' (0.7499), 'dedi' (0.7496), 'permana' (0.7290), 'agil' (0.7265) • …
3	sdr, korban, pt, mobil, nomor…	• 'sdr': 'dpo' (0.8662), 'ing' (0.8515), 'mi' (0.7770), 'iyan' (0.7635), 'andro' (0.7536), 'jhandi' (0.7483), 'oyok' (0.7432), 'herman' (0.7414), 'majid' (0.7392), 'irul' (0.7346) • 'korban': 'siti' (0.8114), 'aliyah' (0.7654), 'karto' (0.7247), 'binti' (0.7222), 'mahmudah' (0.7222), 'soleha' (0.7189), 'trisni' (0.7186), 'sanip' (0.7070), 'ayu' (0.7053), 'alwi' (0.6953) • 'pt': 'indonesia' (0.8240), 'doosan' (0.8207), 'steel' (0.8156), 'wilmar' (0.7920), 'pundi' (0.7881), 'heavy' (0.7868), 'uniwood' (0.7660), 'industries' (0.7571), 'langgeng' (0.7446), 'nabati' (0.7428)

		• 'mobil': 'up' (0.8507), 'tronton' (0.8449), 'pick' (0.8269), 'truk' (0.7816), 'toyota' (0.7705), 'supir' (0.7599), 'carry' (0.7521), 'engkel' (0.7394), 'dump' (0.7340), 'bak' (0.7314) • 'nomor': 'gz' (0.7729), 'yk' (0.7729), 'cux' (0.7645), 'sporty' (0.6988), 'xeon' (0.6900), 'ns' (0.6880), 'gj' (0.6873), 'bh' (0.6715), 'tu' (0.6712), 'be' (0.6689) • …
dst…		

Setelah didapatkan kata-kata yang mirip dari kelima variasi teknik, lalu dilakukan analisa teknik mana yang akan dipilih untuk diinterpretasikan, berdasarkan jumlah topik terbaiknya, nilai koheren terbaiknya, penyusun kata topiknya serta hasil kemiripan kata penyusun topiknya dengan *Word2Vec*. Untuk memudahkan analisa, dilakukan visualisasi data perbandingan dalam bentuk tabel yang dapat dilihat pada Tabel 11. Sebagai sampel untuk kata penyusun topik, diambil pada Topik 1 (pertama) saja dan 5 top words saja.

Tabel 11 Perbandingan Hasil Word2Vec dari seluruh variasi pemodelan

Kode/ Alias	Algoritma	Jumlah Topik	Nilai Koheren	Kata Penyusun Topik	Hasil *Word2Vec*
NB-3	NGrams – Bigrams, min count = 3	9	0.489835929 36226444	sdr andi putra sepeda_motor roni	sdr: dpo, wak_dpo, semang, hardiansyah, wak andi: husni, abdul_muhamad, anwar, sdr_ayi, iyan_kuis putra: fahmi, hartono_sdr, bin_rahmadi, bajul, iwan sepeda_motor: saudara_dedi, motor, mio, honda_beat, honda_scoopy roni: sdr_majid, teguh, tawar, putra, sdr_bule
NB-5	NGrams – Bigrams, min count = 5	9	0.614907395 2628307	dpo sdr sepeda_motor mulyadi unit_handphone	dpo: sdr, warsim, uki, bin_juwanda, herman_dpo sdr: dpo, warsim, iway, rohman_alias, herman_dpo sepeda_motor: saudara_dedi, motor, kb, nopol_dy, em, titip mulyadi: khotim_mastomi, agustiawan, usman, satria, aat unit_handphone: ak, merk_samsung, merk_oppo, imei_imei, merek_oppo
NT-3	NGrams – Trigrams., min count = 3	8	0.583945318 8686285	dpo anak sdr rumah saksi_samin	dpo: sdr, alias_iyong, sdr_miran, sdr_pirdaus, sdr_oyok anak: indra, romli, motor_pt_chandra, scoopy_nopol_un, duk sdr: dpo, sdr_miran, daftar_cari_orang, sdr_rifa, sdr_pirdaus rumah: selanjunya, kontra, roby_adul, sulasmi, beralamat saksi_samin: sadar, dada, pt_anugrah_terang_persada, pegang_tangan, kepala
NT-5	NGrams – Trigrams, min count = 5	10	0.578433872 3795288	anak irwan sdr angga tower	anak: indra, saudara_ridho, romli, kiki_maulana, billi_anak irwan: fida, ilham, supriyadi, mudah_lari, bujuk

					sdr: dpo, wak_dpo, abdul_muhamad_husni, daftar_cari_orang, alias_iyong angga: binawan, dpo_kamis_tanggal, tuntut_berkas_pisah, dpo_res_reskrim, tanggal_juli tower: atap, atap_toko, belah, jebol, tembok
TN	Tanpa N-Grams	8	0.6191725889023254	handphone sdr alan pt motor	handphone: hp, oppo, handhone, redmi, samsung sdr: dpo, ing, mi, iyan, andro alan: iman, jaenal, hilman, wahyudi, jamadi pt: Indonesia, Doosan, steel, wilmar, pundi motor: ride, Yamaha, scoopy, vixion, mio

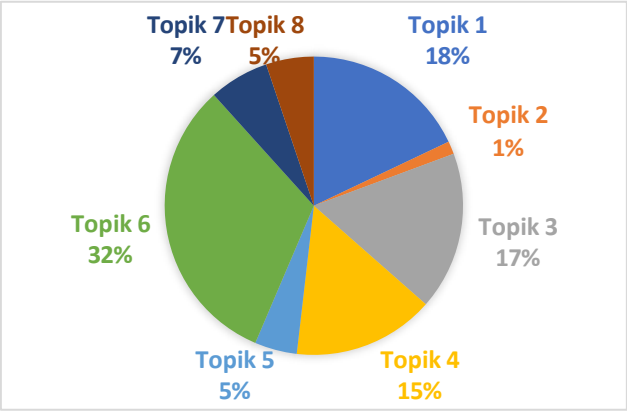
Berdasarkan Tabel 11, nilai koheren tertinggi didapatkan oleh pemodelan TN dengan nilai 0.619 (pembulatan), dan diikuti pemodelan NB-5 dengan nilai 0.615 (pembulatan). Selisih dari kedua pemodelan tersebut cukup rendah, hanya sebesar 0.04 saja. Maka dari itu variasi pemodelan lainnya diabaikan, dan dilakukan analisa lebih lanjut untuk pemodelan NB-5 dan TN. Pada pemodelan NB-5, koheren tertinggi didapatkan pada jumlah topik 9, sedangkan pada TN koheren tertinggi didapatkan pada jumlah topik 8. Semakin banyak topik yang diekstraksi tidak menjadi penanda bahwa model yang dihasilkan lebih baik. Pada pemodelan topik, terdapat kemungkinan terjadinya *over-clustered* atau semacam topik yang tumpang tindih, karena dua atau lebih topik yang memiliki karakteristik hampir sama, namun dianggap berbeda sehingga terklaster menjadi dua atau lebih topik. Maka dari itu, untuk mengatasi hal tersebut, peneliti mempertimbangkan untuk memilih jumlah topik yang lebih sedikit dengan nilai koheren tertinggi, yaitu pemodelan TN. Kata penyusun topik yang dihasilkan dari kedua pemodelan tersebut juga secara umum cukup mudah untuk dilakukan interpretasi topik walaupun pada NB-5 terdapat beberapa kata penyusun topik yang kurang logis jika dianggap sebagai suatu token / kata yang berdiri sendiri seperti unit_handphone. Untuk kemiripan kata yang dihasilkan dari Word2Vec, juga secara umum keduanya menghasilkan kata-kata yang mirip dengan kata utamanya. Namun jika dibandingkan untuk kata yang sama misalnya motor dan sepeda_motor, pemodelan dengan TN menghasilkan kata yang lebih sesuai yaitu ride, Yamaha, scoopy, vixion dan mio yang merupakan nama-nama merek motor. Sedangkan kata yang mirip dengan motor yang dihasilkan pada pemodelan NB-5 yaitu saudara_dedi, motor, kb, nopol_dy, em dan titip, sekilas terlihat kurang memiliki makna yang relevan dengan kata sepeda_motor. Maka dari itu dari kedua variasi pemodelan tersebut, peneliti menyimpulkan pemodelan TN sebagai variasi pemodelan terbaik untuk selanjutnya dilakukan interpretasi topik. Dengan melihat kata-kata penyusun topik serta kata-kata yang mirip dengan kata penyusun topik tersebut, dapat diinterpretasikan 8 (delapan) topik yang dibahas pada dakwaan tindak pidana pencurian yang telah putus pada Pengadilan Negeri Serang yang dapat dilihat pada Tabel 12.

Tabel 12 Hasil Interpretasi Topik

Topik	Kata Penyusun Topik	Interpretasi Topik	Keterangan
1	handphone, sdr, alan, pt, motor, sepeda, jin, merek, iman, muhamad	Pencurian Gadget (HP, Laptop berbagai merek) dan Kendaraan Roda Dua (Motor, Sepeda berbagai merek) yang kemungkinan terkait dengan lingkungan perusahaan.	Cukup jelas
2	toko, motor, sdr, akbar, joni, dpo, sepeda, sandi, suhada, kontra	Pencurian di Toko atau Minimarket (Indomaret, Alfamart) yang melibatkan kendaraan roda dua, dengan pelaku berstatus DPO, dengan lokasi perencanaan / persembunyian di kontrakan atau sejenisnya.	Kata kontra merupakan *stemming* dari kontrakan, dikontrakan dan sejenisnya
3	sdr, korban, pt, mobil, nomor, kambing, motor, sepeda, riki, imei	Pencurian kendaraan roda empat (truk, pick-up, mobil) dan kendaraan roda dua yang melibatkan perusahaan besar.	Kata kambing bersifat anomali, sehingga diabaikan

4	sdr, als, motor, dpo, sepeda, cilegon, rudi, atm, no, korban	Pencurian yang terjadi di wilayah Kota Cilegon dan sekitarnya dengan pelaku berstatus DPO yang kemungkinan berkaitan dengan ATM dan melibatkan kendaraan roda dua	Cukup jelas
5	sdr, toko, handphone, als, korban, burung, rokok, dpo, alias, mobil	Pencurian yang dilakukan disebuah tempat usaha dengan objek pencurian barang sembako dan hewan dengan pelaku berstatus DPO	Cukup jelas
6	korban, als, sdr, motor, sepeda, handphone, warung, muhamad, buah, dpo	Pencurian yang dilakukan disebuah tempat usaha dengan objek pencurian kendaraan dan barang elektronik dengan pelaku berstatus DPO	Kata buah merujuk pada satuan jumlah, bukan sebagai objek buah
7	als, hendra, handphone, motor, merek, besi, dpo, buah, sepeda, kamar	Pencurian barang pribadi dan didalam ruangan pribadi (rumah/kamar tidur/kamar mandi)	Kata 'besi' mungkin merupakan *noise* karena tidak ada kata lain yang mengacu kata tersebut pada topik, sehingga diabaikan
8	sdr, toko, sepeda, handphone, motor, hp, sumiati, ac, alfamart, tom	Pencurian properti / fasilitas minimarket	Cukup jelas

Dari keseluruhan topik yang terbentuk, kemudian dilakukan inverensi dokumen untuk melihat suatu dokumen didalam dataset termasuk dalam topik ke berapa berdasarkan nilai probabilitasnya untuk setiap topiknya dan dari total 500 dokumen dakwaan didapatkan presentase klaster topik yang terbentuk yang dapat dilihat pada Gambar 11.



Gambar 11 Presentase Topik pada seluruh Dokumen

IV. KESIMPULAN

Berdasarkan hasil percobaan dan analisia, maka didapatkan kesimpulan sebagai berikut:

1. Algoritma LDA dapat digunakan untuk melakukan pemodelan topik yang dibahas pada suatu dataset teks dakwaan dengan melakukan preprocessing teks terlebih dahulu yang dilanjutkan dengan proses ekstraksi fitur (opsional) menggunakan algoritma N-Grams, lalu pembobotan kata menggunakan algoritma TF-IDF dan implementasi algoritma LDA untuk jumlah topik tertentu;
2. Evaluasi model dilakukan dengan mencari nilai koheren tertinggi dan perplexity terendah yang dihasilkan pemodelan pada rentang topik 2 hingga 10 topik;
3. Secara keseluruhan, algoritma N-Grams dengan beberapa ukuran parameter yang digunakan mampu meningkatkan performa pemodelan yang dibuktikan dengan meningkatnya nilai koheren yang dihasilkan dari beberapa percobaan. Namun, token atau kata-kata penyusun topik yang dihasilkan dengan pemodelan menggunakan N-Grams dengan parameter tertentu masih menghasilkan kata-kata yang kurang informatif dibandingkan dengan token yang dihasilkan tanpa menggunakan N-Grams. Hal ini kemungkinan terjadi karena jumlah dataset yang digunakan masih kurang, yang

mempengaruhi tingkat kemunculan suatu token yang berdampingan dianggap sebagai sebuah informasi tersendiri.

4. Performa pemodelan terbaik didapatkan dengan melakukan pemodelan topik menggunakan LDA dengan pembobotan kata menggunakan TF-IDF dan tanpa dilakukan ekstraksi fitur N-Grams dengan topik yang diekstrak berjumlah 8 topik, nilai koheren sebesar 0,619 dan perplexity sebesar -10.2078.
5. Nilai koheren optimal yang didapatkan cukup baik dan kata-kata yang dihasilkan dapat dengan mudah untuk diinterpretasikan kedalam suatu topik tertentu.
6. Nilai perplexity yang dihasilkan cenderung menunjukan nilai yang terus menurun dari topik 2 hingga 10 untuk semua variasi percobaan dan nilai terbaik sebesar -10.2078 didapatkan pada jumlah topik 8. Hal ini menandakan bahwa model yang dihasilkan cukup baik dalam memprediksi data baru.

UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih kepada Universitas Pamulang khususnya Program Studi Teknik Informatika S-2, Rektor Universitas Pamulang, Direktur Pasca sarjana Universitas Pamulang, Ketua Program Studi Program Studi Teknik Informatika S-2 Universitas Pamulang, Dosen Pembimbing pada penelitian in, teman-teman dan kerabat penulis atas dukungan baik langsung maupun secara tidak langsung, sehingga penulis dapat menyelesaikan penelitian yang berjudul "PEMODELAN TOPIK PADA DAKWAAN TINDAK PIDANA PENCURIAN DI PENGADILAN NEGERI SERANG MENGGUNAKAN LATENT DIRICHLET ALLOCATION" ini dengan baik. Semoga dengan adanya penelitian ini dapat memberikan dampak positif, terutama dalam bidang *data science* dan dapat menjadi penunjang bagi penelitian selanjutnya.

DAFTAR PUSTAKA

[1] F. D. Nisrina, "Implementasi Deteksi Topik Putusan Hakim Dengan Latent Dirichlet Allocation (LDA)," pp. 1–64, 2020, [Online]. Available: https://dspace.uii.ac.id/handle/123456789/23847

[2] R. Indonesia, "Kitab Undang-Undang Hukum Acara Pidana (KUHAP) No. 8 Tahun 1981," *Kuhap*, p. 871, 1981.

[3] A. H. Dani, E. Y. Puspaningrum, and R. Mumpuni, "Studi Performa TF-IDF dan Word2Vec Pada Analisis Sentimen Cyberbullying," *Router J. Tek. Inform. dan Terap.*, vol. 2, no. 2, pp. 94–106, 2024, [Online]. Available: https://doi.org/10.62951/router.v2i2.76

[4] G. H. A. R. Noer, "Implementasi Algoritma Naïve Bayes dan TF-IDF Dalam Analisis Sentimen Data Ulasan (Studi Kasus: Ulasan Review Aplikasi E-commerce Shopee di Situs Google …," UIN Syarif Hidayatullah Jakarta, 2023. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/68747%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/68747/1/GERALD HALIM AL RASYID NOER-FST.pdf

[5] R. P. F. Afidh and Syahrial, "Pemodelan Topik Menggunakan n-Gram dan Non-negative Matrix Factorization," *J. Inf. dan Teknol.*, vol. 5, no. 1, pp. 265–275, 2023, doi: 10.60083/jidt.v5i1.385.

[6] R. Nurhidayat and K. E. Dewi, "Penerapan Algoritma K-Nearest Neighbor Dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek," *Komputa J. Ilm. Komput. dan Inform.*, vol. 12, no. 1, pp. 91–100, 2023, doi: 10.34010/komputa.v12i1.9458.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003, doi: 10.1016/B978-0-12-411519-4.00006-9.

[8] Alfrida Rahmawati, Najla Lailin Nikmah, Reynaldi Drajat Ageng Perwira, and Nur Aini Rakhmawati, "Analisis topik konten channel YouTube K-pop Indonesia menggunakan Latent Dirichlet Allocation," *Teknologi*, vol. 11, no. 1, pp. 16–25, 2021, doi: 10.26594/teknologi.v11i1.2155.

[9] R. Ma and Y. J. Kim, "Tracing the evolution of green logistics: A latent dirichlet allocation based topic modeling technology and roadmapping," *PLoS One*, vol. 18, no. 8 August, pp. 1–20, 2023, doi: 10.1371/journal.pone.0290074.

[10] F. Gunawan, I. Cholissodin, and P. P. Adikara, "Pemerolehan Informasi Artikel terkait Covid-19 dengan menggunakan Metode Vector Space Model dan Word2Vec untuk Query Expansion," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 3, pp. 960–968, 2021, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/8690