

Klasifikasi Laporan Keluhan Pelayanan Publik Berdasarkan Instansi Menggunakan Metode LDA-SVM

by Muhammad Alkaff

Submission date: 05-Apr-2023 12:03PM (UTC+0700)

Submission ID: 2056344745

File name: A.3.d.1-Klasifikasi Laporan Keluhan Pelayanan Publik Berdasarkan Instansi Menggunakan Metode LDA-SVM.pdf (470.47K)

Word count: 7425

Character count: 45213

KLASIFIKASI LAPORAN KELUHAN PELAYANAN PUBLIK BERDASARKAN INSTANSI MENGGUNAKAN METODE LDA-SVM

Muhammad Alkaff^{*1}, Andreyan Rizky Baskara², Irham Maulani³

^{1,2,3}Universitas Lambung Mangkurat, Banjarmasin

Email : ¹m.alkaff@ulm.ac.id, ²andreyan.baskara@ulm.ac.id, ³irhammaulani@gmail.com

*Penulis Korespondensi

(Naskah masuk: 10 Juli 2021, diterima untuk diterbitkan: 17 November 2021)

Abstrak

Sebuah sistem layanan untuk menyampaikan aspirasi dan keluhan masyarakat terhadap layanan pemerintah Indonesia, bernama Lapo! Pemerintah sudah lama memanfaatkan sistem tersebut untuk menjawab permasalahan masyarakat Indonesia terkait permasalahan birokrasi. Namun, peningkatan volume laporan dan pemilahan laporan yang dilakukan oleh operator dengan membaca setiap keluhan yang masuk melalui sistem menyebabkan sering terjadi kesalahan dimana operator meneruskan laporan tersebut ke instansi yang salah. Oleh karena itu, diperlukan suatu solusi yang dapat menentukan konteks laporan secara otomatis dengan menggunakan teknik Natural Language Processing. Penelitian ini bertujuan untuk membangun klasifikasi laporan secara otomatis berdasarkan topik laporan yang ditujukan kepada instansi yang berwenang dengan menggabungkan metode Latent Dirichlet Allocation (LDA) dan Support Vector Machine (SVM). Proses pemodelan topik untuk setiap laporan dilakukan dengan menggunakan metode LDA. Metode ini mengekstrak laporan untuk menemukan pola tertentu dalam dokumen yang akan menghasilkan keluaran dalam nilai distribusi topik. Selanjutnya, proses klasifikasi untuk menentukan laporan agensi tujuan dilakukan dengan menggunakan SVM berdasarkan nilai topik yang diekstraksi dengan metode LDA. Performa model LDA-SVM diukur dengan menggunakan confusion matrix dengan menghitung nilai akurasi, presisi, recall, dan F1 Score. Hasil pengujian menggunakan teknik split train-test dengan skor 70:30 menunjukkan bahwa model menghasilkan kinerja yang baik dengan akurasi 79,85%, presisi 79,98%, recall 72,37%, dan Skor F1 74,67%.

Kata kunci: *lapo, laporan, topic modeling, klasifikasi, latent dirichlet allocation, support vector machine*

CLASSIFICATION OF PUBLIC SERVICE COMPLAINT REPORTS BASED ON AGENCY USING LDA-SVM METHOD

Abstract

A service system to convey aspirations and complaints from the public against Indonesia's government services, named Lapo! The Government has used the Government for a long time to answer the problems of the Indonesian people related to bureaucratic problems. However, the increasing volume of reports and the sorting of reports carried out by operators by reading every complaint that comes through the system cause frequent errors where operators forward the reports to the wrong agencies. Therefore, we need a solution that can automatically determine the report's context using Natural Language Processing techniques. This study aims to build automatic report classifications based on report topics addressed to authorized agencies by combining Latent Dirichlet Allocation (LDA) and Support Vector Machine (SVM). The topic-modeling process for each report was carried out using the LDA method. This method extracts reports to find specific patterns in documents that will produce output in topic distribution values. Furthermore, the classification process to determine the report's destination agency carried out using the SVM based on the value of the topics extracted by the LDA method. The LDA-SVM model's performance is measured using a confusion matrix by calculating the value of accuracy, precision, recall, and F1 Score. The test results using the train-test split technique with a 70:30 show that the model produces good performance with 79.85% accuracy, 79.98% precision, 72.37% recall, and 74.67% F1 Score

Keywords: *lapo, report, topic modeling, classification, latent dirichlet allocation, support vector machine*

1. PENDAHULUAN

Gerakan manajemen kualitas yang di pelopori oleh W. Edwards Deming seorang falsafah manajemen dengan manajemen mutu terpadu, selalu menganjurkan pentingnya orientasi terhadap pelanggan. Karena pelangganlah yang menentukan kualitas pada barang maupun jasa yang digunakan (Obsorne & Gaebler, 1992). Orientasi terhadap pelanggan dalam konteks manajemen publik dimaknai secara luas adalah bagaimana birokrasi pemerintah dapat menghasilkan kebijakan publik (*public interest*) dan dapat di pertanggungjawabkan ke publik. Konsep pelanggan dalam pelayanan publik lebih tepat disebut dengan *citizen* atau warga negara. Warga negara berbeda dengan pelanggan, dimana pelanggan dapat berhenti terhadap suatu produk atau jasa yang tidak memuaskan. Sedangkan warga negara tidak memiliki pilihan lain selain menyampaikan keluhan dalam bentuk pengaduan (Sagaf, al., 2018).

Pelayanan publik menjadi tolak ukur keberhasilan pelaksanaan tugas dan pengukuran kinerja pemerintah melalui birokrasi. Akan tetapi, fenomena pelayanan dan pengaduan publik oleh birokrasi pemerintah sarat dengan permasalahan. Misalnya, prosedur yang berbelut-larut, ketidakpastian waktu dan harga yang menyebabkan pelayanan menjadi sulit dijangkau (Maryam, 2016). Pengelolaan pengaduan pelayanan publik belum dikelola secara efektif dan terintegrasi. Masing-masing organisasi penyelenggara mengelola pengaduan secara parsial dan tidak terkoordinir dengan baik, akibatnya terjadi duplikasi dan aduan yang bukan masuk wewenang organisasi penyelenggara tersebut. Untuk itu pemerintah Republik Indonesia membentuk Sistem Pengelolaan Pengaduan Pelayanan Publik Nasional (SP4N) – Layanan Aspirasi dan Pengaduan Online Rakyat (LAPOR!) (Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, n.d.).

Aplikasi Laporan! adalah layanan penyampaian semua aspirasi dan pengaduan masyarakat Indonesia melalui beberapa kanal pengaduan yaitu *website* www.lapor.go.id, SMS 1708, *twitter* @lapor1708 dan aplikasi *smartphone android*. Aplikasi Laporan! yang semula diinisiasikan dan dikembangkan oleh Unit Kerja Presiden Bidang Pengawasan dan Pengendalian Pembangunan (UKP-PPP atau UKP4) kini dikelola oleh Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi (Kementerian PANRB). Laporan! telah ditetapkan sebagai Sistem Pengelolaan Pengaduan Pelayanan Publik Nasional (SP4N) berdasarkan Peraturan Presiden nomor 76 tahun 2013 dan Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 3 tahun 2015 yang merealisasikan kebijakan *no wrong door policy* yang berarti menjamin pengaduan dari manapun dan jenis apapun akan disalurkan kepada penyelenggara pelayanan publik yang berwenang menanganinya

(Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, n.d.) (Megawati, 2015).

Aplikasi Laporan! telah terhubung dengan 34 Kementerian, 96 Lembaga, dan 493 Pemerintah daerah di Indonesia. Jumlah pelapor per Januari 2019 sebanyak 801.257 pengguna. Total laporan yang masuk sebanyak 1.389.891 yang terbagi dalam beberapa kanal Laporan!, dimana *website* sebagai kanal terbanyak dalam menampung laporan (Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, n.d.).

Kota Banjarmasin adalah salah satu kota di Indonesia dengan pertumbuhan pengguna aplikasi Laporan! yang tinggi, pada bulan Juli sampai Desember 2017 aplikasi Laporan! di kota Banjarmasin telah menjaring 115 laporan perharinya. Pada bulan Januari tahun 2019, laporan di Kota Banjarmasin tumbuh hingga 275 laporan perhari yang akan didisposisi ke semua instansi di Kota Banjarmasin yaitu 44 instansi yang dapat dilihat pada lampiran 2. Dengan antusiasme warga yang tinggi serta kecepatan penyelenggara pelayanan publik dalam menindaklanjuti keluhan yang berasal dari aplikasi Laporan!, Kota Banjarmasin pun menerima penghargaan sebagai tiga kota terbaik pengguna aplikasi Laporan!. Pemerintah Kota Banjarmasin juga termasuk dalam 10 bagian dari kementerian negara, lembaga negara dan pemerintah daerah yang menjadi percontohan penerapan aplikasi Laporan! merujuk dari hasil monitoring yang dilakukan oleh Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi dan Kantor Staf Presiden (Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, n.d.)¹.

Berdasarkan data yang telah dipaparkan, dengan jumlah laporan khususnya di Kota Banjarmasin beserta jumlah instansi yang banyak tersebut tentu akan sulit untuk pengelola dalam mengelola laporan-laporan yang masuk perharinya dengan cara konvensional. Pengelola akan sulit dalam mengelompokkan dan menyalurkan laporan warga negara kepada penyelenggara pelayanan publik yang berwenang menanganinya. Keadaan tersebut menyebabkan lambatnya laporan tersebut untuk ditindaklanjuti oleh instansi terkait, untuk saat ini waktu yang dibutuhkan untuk pengelola dalam memverifikasi laporan untuk diteruskan ke penyelenggara pelayanan publik terkait adalah 3 hari kerja (Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, n.d.).

Dari 20 data yang diambil secara acak pada aplikasi Laporan! halaman Kota Banjarmasin yang dapat terdapat 70% laporan yang masuk terdapat kesalahan disposisi oleh admin aplikasi Laporan! Kota Banjarmasin. Hal ini tidak sesuai dengan kebijakan *no wrong door policy* yang merupakan salah satu

¹ Kumparan, "Memanfaatkan LAPOR!, Pemkot Banjarmasin Mendapat Penghargaan", diakses dari <https://kumparan.com/banjarihis/menjaring-keluhan-berujung-penghargaan-pada-tanggal-23-Januari-2019-pukul-02:00>.

latar belakang dibuatnya aplikasi Laporan!. Pengkategorian manual yang harus dipilih oleh pelapor saat melakukan laporan juga dapat menyebabkan kesalahan pemilihan kategori laporan ditambah dengan kategori yang terlalu banyak dan pilihan yang sering tidak selaras dengan keluhan yang ingin dilaporkan oleh pelapor juga dapat menyebabkan kesalahan laporan tersebut untuk diteruskan ke penyelenggara pelayanan publik yang berwenang.

Oleh sebab itu, perlu dikembangkannya prototype pengklasifikasian laporan secara otomatis berdasarkan isi laporan. Salah satu metode dalam klasifikasi teks adalah menggunakan *Natural Language Processing* (NLP). NLP memiliki beberapa teknik yaitu *topic modeling*, *sentiment analysis*, dan *information extraction*. Dan *topic modeling* merupakan teknik yang cocok berdasarkan permasalahan yang sudah dijabarkan sebelumnya dimana *dataset* laporan keluhan masyarakat akan dilakukan *feature extraction* dan hasilnya akan dilakukan klasifikasi berdasarkan instansi (Young, et al., 2018).

Berdasarkan penelitian terdahulu tentang implementasi *Clasification of Indonesian News Articles based on Latent Dirichlet Allocation* berhasil melakukan klasifikasi terhadap dokumen berita berbahasa Indonesia dengan metode LDA dengan akurasi rata-rata 70% terhadap 10 arsip artikel secara keseluruhan sebagai *data testing* dan pada arsip ke 5 akurasi mencapai 100% (Kusumaningrum, et al., 2016).

Pada penelitian yang dilakukan (Li, et al., 2011) metode *Latent Dirichlet Allocation* melakukan klasifikasi terhadap data dengan dimensi yang tinggi dan data dalam teks yang besar menggunakan metode LDA dan SVM juga melakukan perbandingan terhadap metode lainnya dan mendapatkan akurasi sebesar 91.3%, lebih tinggi dibandingkan metode *Weight of Evidence for Text* (WET) yang memiliki akurasi 90.3% dan metode *Mutual Information* (MI) yang memiliki akurasi 77.8%. Serta, penelitian ini juga menggabungkan metode LDA dan SVM untuk mengklasifikasi dokumen tesk dan mendapatkan *macro-P* dengan hasil 0.89279, *macro-R* dengan hasil 0.889472, *macro-F1* dengan hasil 0.889523 dan *micro-F1* dengan hasil 0.890751 yang lebih baik dibandingkan penggabungan metode MI dan SVM dengan hasil *macro-P* sebesar 0.863596, *macro-R* sebesar 0.865328, *macro-F1* sebesar 0.86463 dan *micro-F1* sebesar 0.872053. Dimana *macro-P* merupakan *precision* sistem secara keseluruhan pada data yang berbeda, *macro-R* adalah hasil *recall* sistem secara keseluruhan pada data yang berbeda, *macro-F1* adalah hasil perhitungan kombinasi *harmonic* dari *precision* dan *recall*, *micro-F1* adalah hasil perhitungan kombinasi individu dari *precision* dan *recall*.

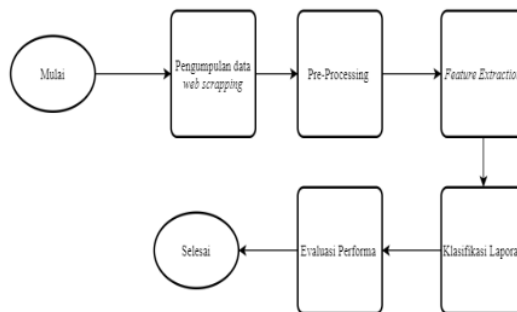
Pada penelitian yang dilakukan (Sheshasaayee & Thailambal, 2017) melakukan perbandingan metode klasifikasi pada konten web dalam bentuk teks dan metode SVM mendapatkan akurasi sebesar 97.40%, lebih tinggi dibandingkan metode *Naïve Bayes* yang memiliki akurasi sebesar 90.20% dan metode *Random Forest* dengan akurasi 88% dalam melakukan klasifikasi terhadap teks.

Pada penelitian yang dilakukan (Cui, et al., 2014) yang bertujuan untuk mengklasifikasi teks berita dengan melakukan perbandingan metode *Naïve Bayes*, SVM dan kombinasi LDA dan SVM. Hasilnya kombinasi metode LDA dan SVM mendapatkan nilai presisi sebesar 96.16%, lebih besar dibandingkan metode *Naïve Bayes* dengan *feature extraction* yang hanya menghasilkan presisi 69.78% dan metode SVM dengan *feature extraction* yang menghasilkan presisi sebesar 94.55%.

Pada penelitian yang dilakukan (George, et al., 2017) yang melakukan survey pada metode klasifikasi dan metode *feature extraction* dan merujuk pada penelitian terdahulu menyebutkan bahwa kombinasi metode LDA dan SVM menunjukkan performa yang lebih baik dibandingkan kombinasi metode lain seperti *Naïve Bayes* dan LSA.

Berdasarkan penelitian terkait di atas dapat disimpulkan bahwa metode *Latent Dirichlet Allocation* dan *Support Vector Machine* merupakan kombinasi yang menghasilkan performa tinggi dalam mengklasifikasi teks. Oleh karena itu, kombinasi kedua metode tersebut akan diterapkan ke dalam *prototype* klasifikasi laporan keluhan pelayanan publik berdasarkan instansi untuk mengetahui performa dalam menyelesaikan masalah yang telah dipaparkan.

2. METODE PENELITIAN



Gambar 1. Perancangan Prototype

Metode penelitian yang dilakukan pada penelitian ini yang dapat dilihat pada gambar 1 adalah sebagai berikut :

1. Pengambilan Data
2. Text Pre-Processing
3. Feature Extraction

4. Klasifikasi Laporan
5. Evaluasi Performa

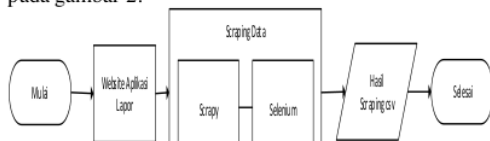
Yang akan dijelaskan pada pont-point berikut :

2.1. Pengambilan Data

Pengambilan laporan di aplikasi Lapo! menggunakan teknik *web scraping*. *Web Scraping* adalah proses pengumpulan data dan informasi yang berasal dari internet, umumnya halaman *website* berbasis *HTML*, secara otomatis dimana data tersebut akan dianalisis sesuai tujuan dan kebutuhan (Slamet, et al., 2018). *Tools* yang digunakan dalam *web scraping* ini adalah *framework* Scrapy dengan bahasa pemrograman Python, Python merupakan bahasa pemrograman *interpreter* yang merupakan *high-level language* dan berparadigma *object-oriented*. Bahasa Pemrograman ini banyak digunakan dalam keperluan *machine learning* dan hal otomatisasi seperti *scraping data* (Harrington, 2020).

Scrapy adalah *framework* berbasis Python yang digunakan dalam melakukan *web scraping*. Scrapy dapat memudahkan pengguna dalam melakukan *scraping data* dibandingkan menulis kode *scraping* dari awal (Loukas, 2016). Dikarenakan *website* Lapo!.go.id kota Banjarmasin membutuhkan pengguna untuk *login* sebelum melihat detail data laporan dan halaman *login* berbentuk *ajax* yaitu teknik yang umum digunakan untuk mengirim dan meminta data ke *server* secara *asynchronous* di *client-side*, maka proses *scraping* membutuhkan *library* tambahan untuk membuka *browser* dan dapat melakukan *login* secara otomatis. *Library* tambahan yang digunakan adalah Selenium.

Selenium adalah perangkat lunak yang digunakan untuk melakukan otomatisasi untuk banyak keperluan, seperti melakukan *software testing* untuk *website*. Selenium juga mendukung *ajax* dalam mengirim maupun mendapatkan data (Gojarea, et al., 2015). Sehingga dengan menggunakan Selenium, *login* di aplikasi Lapo! dapat dilakukan bersamaan dengan proses *scraping* secara otomatis. Halaman yang akan diproses berada pada url www.lapo!.go.id/instansi/pemerintah-kota-banjarmasin-done. Di halaman ini data yang tersedia adalah laporan yang telah selesai dan telah ditutup oleh pengelola dan telah diberi label sesuai instansi yang berwenang. Proses *scraping data* dapat dilihat pada gambar 2.



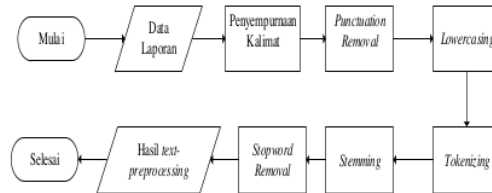
Gambar 2. Alur *scraping data*

Berikut adalah penjelasan diagram alir proses *scraping data*:

1. Membuka *website* aplikasi Lapo! halaman Kota Banjarmasin sebagai sumber data.
2. Proses *scraping* akan dilakukan dimana didalam proses ini terdapat *framework* Scrapy sebagai *tools* untuk mengambil data dan Selenium untuk rekayasa *browser* untuk melakukan proses seperti klik tombol, menuliskan informasi data untuk melakukan *login*.
3. Hasil proses *scraping data* akan disimpan ke dalam bentuk file CSV. Dimana data ini akan menjadi *data training* dan *data testing*.

2.2. Text Pre-Processing

Text pre-processing adalah proses yang dibutuhkan bagi sebuah data yang berupa teks. *Text pre-processing* bertujuan agar teks yang diinginkan berkurang kompleksitasnya tanpa mempengaruhi substansi dan informasi yang terkandung di dalam teks tersebut (Denny & Spirling, 2018).



Gambar 3. Alur *Text Pre-Processing*

Alur *text pre-processing* pada gambar 3 dimulai ketika data laporan di *inputkan*.

a. *Penyempurnaan Kalimat*

Penyempurnaan kalimat pada data dilakukan secara manual untuk menyempurnakan kata yang disingkat atau hilangnya huruf yang membuat kalimat tidak sempurna dan ambigu, lalu diubah dengan kata yang sudah sesuai secara kaidah baik penulisan maupun arti yang sesuai dengan KBBI. Contohnya seperti kata “d” menjadi “di”, “ad” menjadi “ada” secara manual.

b. *Punctuation removal*

Punctuation removal adalah proses menghilangkan komponen-komponen yang tidak relevan dan tidak mengandung informasi. Seperti tanda baca (“”, “?”, “}”, “()”) dan *white space* berlebih. Contoh dari *punctuation* dapat dilihat pada kalimat berikut “benarkah?” menjadi “benarkah”.

c. *Lowercasing*

Lowercasing yaitu tahap menyamakan semua kalimat dalam teks menjadi huruf kecil. Contoh dari *lowercasing* dapat dilihat pada kalimat berikut “Lapo!” menjadi “lapo!”.

d. *Tokenizing*

Proses pemotongan kalimat menjadi bagian-bagian yang disebut *token*. Contoh dari *tokenizing* dapat dilihat pada tabel 1.

Tabel 1. Keluaran dan masukan *tokenizing*

Masukan	Keluaran
lapor pak tolong	["lapor", "pak",
pak diperiksa hak	"tolong", "pak",
kepemilikan tanah	"diperiksa", "hak",
dibelakang	"kepemilikan", "tanah", "d
bangunan sdn	ibelakang", "bangunan",
basirih 3 karena	"sdn", "basirih", "3",
ilegal listriknya	"karena", "ilegal",
	"listriknya]

e. *Stemming*

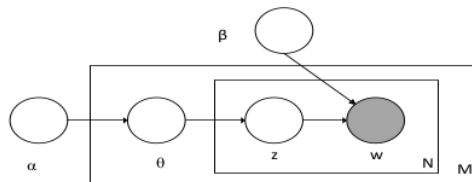
Proses mengubah kata berimbuhan menjadi kata dasar. Contoh dari *stemming* dapat dilihat pada kalimat "kepemilikan" menjadi "milik".

f. *Stopword removal*

Merupakan tahap terakhir pada tahap *text pre-processing*. Tahap ini akan menghapus kata yang kurang penting dan tidak memiliki informasi yang dibutuhkan. Contoh kata yang termasuk *stopword* adalah "lah", "kah", "pun", "tah", "ku", "mu", "nya" (Tala, 2003).

2.3. *Feature Extraction*

Setelah tahap *text pre-processing* selesai, tahap berikutnya teks akan dilakukan *feature extraction* dan pembobotan kata menggunakan metode *Latent Dirichlet Allocation (LDA)*. *Latent Dirichlet Allocation (LDA)* metode yang diusulkan oleh Blei et al pada tahun 2003 (Blei, et al., 2003) adalah model probabilistik *generative* dari koleksi data diskrit seperti *korpus* teks. Ide dasarnya adalah dokumen dipresentasikan sebagai campuran acak topik *laten* (tidak terlihat), dimana topik-topik ditandai oleh distribusi kata dan LDA akan menganggap bahwa setiap dokumen dapat diwakili oleh distribusi probabilistik topik laten. LDA akan mempresentasikan topik dengan probabilitas kata, dimana kata yang memiliki probabilitas yang paling tinggi di setiap topiknya akan mempresentasikan topik tersebut.



Gambar 4. Grafik representasi model probabilistik LDA

Pada gambar 4 dapat dilihat persegi panjang adalah *plates* yang menggambarkan replika. Dimana *plate* terluar menggambarkan dokumen, dan *plate* di dalam

menggambarkan pilihan topik yang berulang dan kata-kata di dalam dokumen. α parameter yang mewakili distribusi topik pada tingkat *corpus* dalam dokumen, β parameter yang mewakili distribusi kata pada tingkat *corpus* dalam topik.

Metode LDA terdiri dari 2 tahapan. Tahapan pertama yaitu tahap inialisasi dan tahapan kedua yaitu tahap *Gibbs Sampling*. Pada tahap inialisasi, setiap kata unik yang sudah didapatkan dari dokumen diberikan topik secara acak. Selanjutnya pada tahapan kedua, dilakukan iterasi untuk setiap kata menggunakan metode *Gibbs Sampling*. Untuk mencari perkiraan probabilitas posterian topik pada sebuah kata menggunakan persamaan (1) berikut.

$$p(z_i = t | z_{-i}, w_i) \propto \frac{n_{-i,j}^{(w)} + \beta}{n_{-i,t}^{(w)} + W\beta} \frac{n_{-i,t}^{(m_i)} + \alpha}{m_i + T\alpha} \quad (1)$$

Dimana $n_{-i,t}^{(w)}$ adalah jumlah kata w_i yang memiliki topik t . $n_{-i,t}^{(w)}$ Adalah jumlah total semua kata pada topik t . $n_{-i,t}^{(m_i)}$ adalah jumlah kata yang memiliki topik t pada dokumen m_i . n_{-i} adalah total kata keseluruhan pada dokumen m_i .

Setelah batas maksimal iterasi yang ditentukan pada *Gibbs Sampling* telah tercapai, selanjutnya dilakukan perhitungan nilai θ (*theta*) dan nilai Φ (*phi*). Tahap ini bertujuan sebagai menentukan pembobotan kata pada isi laporan untuk mengetahui topik apa yang terkandung pada kata di dalam laporan. Hasil yang didapatkan adalah nilai θ (*theta*) dimana *theta* adalah distribusi nilai topik terhadap dokumen dan nilai Φ (*phi*) merupakan distribusi nilai kata terhadap topik. Nilai *theta* ini akan digunakan pada pembobotan dalam melakukan klasifikasi. Untuk menghitung nilai θ (*theta*) digunakan persamaan (2) dan untuk menghitung nilai Φ (*phi*) menggunakan persamaan (3).

$$\theta^{(d_i)} = \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,t}^{(d_i)} + T\alpha} \quad (2)$$

$$\varphi_j^w = \frac{n_{-i,j}^{(w)} + \beta}{n_{-i,t}^{(w)} + W\beta} \quad (3)$$

Contoh *topic modeling* pada *dataset* dengan jumlah tiga topik dapat dilihat pada tabel 2.

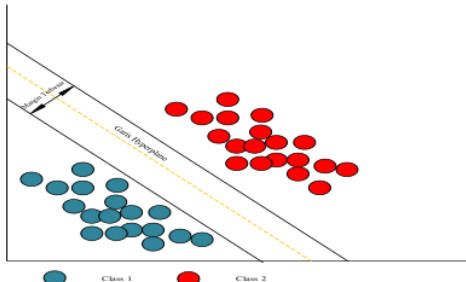
Tabel 2. Contoh *topic modeling* dengan LDA

Node label	Topi k 1	Topi k 2	Topi k 3
Mohon untuk ditertibkan parkir-parkir liar di sepanjang trotoar jalan Belitung Laut trotoar yang bagus dan luas digunakan oleh oknum-oknum	0.03	0.10	0.15

Node label	Topik 1	Topik 2	Topik 3
untuk memarkirkan kendaranya			

2.4. Klasifikasi

Klasifikasi berdasarkan label instansi akan dilakukan menggunakan metode *Support Vector Machine*. *Support Vector Machine* adalah metode *supervised* yang akan mencoba untuk mencari fungsi optimal yang digunakan untuk pemisah antar kelas (*hyperplane*) dua kelas data yang berbeda dan akan menghasilkan *model* yang terbaik untuk data di masa depan. SVM akan mencoba mencari nilai maksimum margin antara dua kelas data dari sampel data yang akan diujikan yang dapat dilihat pada gambar 5.



Gambar 5. Garis pemisah antar kelas (*hyperplane*) di metode SVM

Untuk data yang kompleks dan *non-linear*, SVM memiliki teknik yang bernama *kernel trick*. Dimana kernel akan *transformasi* data menjadi *N dimensional space*. Beberapa kernel yang ada di SVM adalah *linear*, *polynomial*, *radial basis function* (RBF) dan *sigmoid*. SVM sudah banyak diterapkan dalam studi kasus pengenalan pola dan pengkategorian teks dan menghasilkan akurasi yang sangat tinggi (Sheshasaayee & Thailambal, 2017)

Kernel SVM yang digunakan pada penelitian adalah kernel *Radial Basis* merujuk penelitian Kaur Guvir and Kaur Parvinder (Kaur & Kaur, 2017), kernel *Radial Basis* merupakan kernel yang menghasilkan akurasi yang baik dalam melakukan klasifikasi teks dibandingkan kernel linear. Kernel *Radial Basis Function* (RBF) merupakan fungsi yang nilainya tergantung dari jarak asal sebuah data dengan data lainnya di *hyperplane*. Kernel RBF dapat dihitung menggunakan persamaan (4).

$$K(X_1, X_2) = \text{exponent}(-\gamma \|X_1 - X_2\|^2) \quad (4)$$

Dimana $\|X_1 - X_2\|$ merupakan Euclidian distance antara data X_1 dan data X_2 .

Klasifikasi akan menggunakan nilai theta yang dihasilkan metode LDA sehingga laporan dapat diklasifikasi berdasarkan label instansi. Contoh dari klasifikasi SVM pada *dataset* dapat dilihat pada tabel 3.

Tabel 3. Contoh Klasifikasi dengan SVM

Node label	Topik 1	Topik 2	Topik 3	Label
Mohon untuk ditingkatkan parkir-parkir liar di sepanjang trotoar jalan Belitung Laut trotoar yang bagus dan luas digunakan oleh oknum-oknum untuk memarkirkan kendaranya	0.03	0.10	0.15	Dinas Perhubungan
ingin menyampaikan bahwa kegiatan proyek pembuatan trotoar di sepanjang jalan ayani km 2 sangat asal asalan banyak yang tidak dibuatkan saluran pembuangan air	0.07	0.30	0.03	Dinas Pekerjaan Umum dan Perumahan Rakyat

2.5 Evaluasi Performa

Evaluasi performa metode menggunakan *Confusion Matrix*, pengujian dilakukan dengan membandingkan hasil prediksi kombinasi metode LDA dan SVM dengan data laporan masyarakat yang telah didapat sebelumnya (*data testing*). Dalam melakukan *Confusion Matrix* dibutuhkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Dimana matriks yang digunakan adalah *accuracy*, *precision*, *recall*, dan *F1 Score*.

3. Hasil dan Pembahasan

3.1. Pengumpulan dan Pemilahan Data

Data yang dikumpulkan pada penelitian ini merupakan laporan masyarakat Kota Banjarmasin yang telah diberi label selesai oleh admin aplikasi Laporan!. Pengumpulan data menggunakan teknik *scraping* dimana teknik ini akan mengambil informasi yang ada pada setiap komponen website Laporan! dalam bentuk HTML. Scrapy yang merupakan *framework* yang digunakan untuk melakukan proses *scraping* web Laporan! mengenali setiap komponen HTML pada website menggunakan XPath atau CSS Selector dan Selenium digunakan untuk berinteraksi dengan website Laporan! secara otomatis seperti *login* dan membuka halaman baru pada saat proses *scraping*.

Terdapat 1436 jumlah laporan dan 44 jumlah instansi yang terdapat pada data hasil *scraping*. Persebaran jumlah laporan berdasarkan instansi. Dimana Dinas Pekerjaan Umum dan Perumahan Rakyat merupakan instansi dengan jumlah laporan yang terbanyak dengan jumlah

(Syarif, et al., 2016). Atas dasar ini maka menulis akan melakukan optimasi model dengan mencari nilai *hyperparameters* dan diharapkan dapat menaikkan akurasi model dalam melakukan prediksi laporan. Optimasi akan dilakukan dengan mencari nilai *hyperparameters* pada metode *latent dirichlet allocation* dan *support vector machine*.

1. Optimasi metode *Latent Dirichlet Allocation*

Merujuk pada (Blei, et al., 2003) terdapat beberapa parameter yang mempengaruhi performa metode yaitu jumlah iterasi, jumlah topik, α (alpha), dan β (beta). Sehingga dengan melakukan percobaan dengan mengubah nilai tersebut dan melakukan percobaan hingga mendapatkan akurasi yang optimal dalam melakukan klasifikasi laporan.

- Jumlah Iterasi

Jumlah iterasi dibutuhkan setiap dokumen dalam setiap prosesnya hingga proses modeling dapat konvergen (Blei, et al., 2003). Merujuk pada jurnal A Prior Setting that Improves LDA in both Document Representation and Topic Extraction (Ding & Jin, 2019), nilai jumlah iterasi yang digunakan adalah 50. Oleh sebab itu, dalam penelitian ini nilai jumlah iterasi akan menggunakan nilai 50 sebagai nilai awal dalam meningkatkan tingkat akurasi.

- Jumlah topik

Pada saat menggunakan metode LDA, diharuskan untuk memberikan nilai tetap untuk jumlah topik (Sarkar, et al., 2019), karena jumlah topik dapat memberikan pengaruh pada kualitas *topic modeling* (Ding & Jin, 2019) maupun terhadap tingkat akurasi (Arun, et al., 2010). Namun, menentukan nilai topik jadi permasalahan disebabkan tidak adanya metode yang benar-benar tepat dalam menentukan nilai topik (Wallach, et al., 2009). Merujuk Blei et al yang memasukan rentang nilai topik pada klasifikasi dengan nilai 50. Maka, dalam percobaan ini angka yang digunakan akan menggunakan nilai dari percobaan Blei et al (Blei, et al., 2003) sebagai nilai awal dalam meningkatkan akurasi dan akan meningkatkan nilai karena jumlah topik lebih besar lebih baik dibandingkan nilai yang lebih kecil (Wallach, et al., 2009).

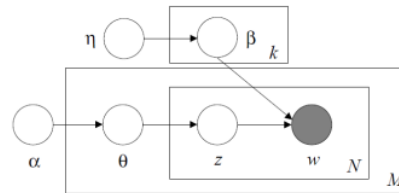
- Alpha (α)

Nilai α merupakan salah satu *hyper-parameters* yang mempengaruhi performa dari metode LDA. α merupakan parameter yang mewakili distribusi topik pada tingkat *corpus* dalam dokumen pada persebaran topik (Beleites, et al., 2012) (Ding & Jin, 2019). Merujuk pada On Smoothing and Inference for Topic Models (Asuncion, et al., 2009), nilai alpha yang baik diperlukan untuk menaikkan perfroma metode. Nilai α yang digunakan pada penelitian ini adalah 0.1 sebagai nilai awal dalam meningkatkan akurasi metode

(Asuncion, et al., 2009) (Teh, et al., 2007) (Hoffman & Blei, 2010).

- Beta (β)

Nilai β merupakan salah satu *hyper-parameters* yang mempengaruhi performa dari metode LDA. β merupakan Parameter yang mewakili distribusi kata pada tingkat *corpus* dalam topik (Blei, et al., 2003) (Ding & Jin, 2019). Merujuk pada (Asuncion, et al., 2009), nilai beta yang baik diperlukan untuk menaikkan perfroma metode. Merujuk pada (Blei, et al., 2003), untuk mengubah parameter distribusi kata pada beta, maka yang diatur nilainya adalah eta (η) yang dapat dilihat pada gambar 6. Eta adalah distribusi kata pada setiap nilai beta (Hoffman & Blei, 2010).



Gambar 6. Representasi model probabilistik LDA yang dihaluskan

Nilai η yang digunakan pada penelitian ini adalah 0.1 sebagai nilai awal dalam meningkatkan akurasi metode (Teh, et al., 2007) (Asuncion, et al., 2009) (Hoffman & Blei, 2010).

2. Optimasi metode *Support Vector Machine*

Optimasi pada metode *support vector machine* akan dilakukan menggunakan algoritma *grid search* untuk menentukan *hyper-parameters* terbaik dalam melakukan klasifikasi data laporan. *Grid search* merupakan algoritma pencarian penuh berdasarkan *sub-set hyper-parameter* yang telah ditentukan. Dimana *hyper-parameters* tersebut ditentukan berdasarkan nilai paling minimal (terendah) sampai ke nilai maksimal (tertinggi). Namun, algoritma ini memerlukan komputasi yang tinggi (Syarif, et al., 2016).

SVM memiliki *hyper-parameters* yang relatif lebih sedikit dibandingkan metode *machine learning* lainnya, dan beberapa *hyper-parameters* bergantung pada kernel yang digunakan, sehingga pencarian optimasi *hyper-parameters* dapat menggunakan teknik seperti algoritma pencarian (Mantovani, et al., 2015) sehingga penggunaan algoritma *grid search* akan lebih mudah diterapkan dengan komputasi yang lebih sedikit. *Grid search* akan selalu dijalankan pada setiap perubahan parameter pada metode LDA sehingga pada metode SVM, dapat menghasilkan akurasi yang optimal pada setiap data hasil *topic modeling* metode LDA pada saat dilakukan optimasi. C (*cost*) merupakan *hyper-parameters* pada metode SVM dimana C merupakan nilai yang menentukan jumlah penalti akibat

kesalahan dalam klasifikasi data. Semakin besar nilai C maka akan memberikan penalti yang besar terhadap kesalahan dalam klasifikasi. Kernel yang digunakan pada penelitian ini adalah kernel *radial basis function*, dimana pada kernel ini terdapat parameter yang mempengaruhi akurasi yaitu γ dimana γ merupakan bagaimana data *training* dipetakan kedalam ruang input berdimensi tinggi (*feature space*) dan menentukan bagaimana pengaruh setiap data yang ada (Syarif, et al., 2016) (Kusumaningrum, 2017). Nilai γ dan C akan dimulai dari 0.001 sampai 10.000 dengan kenaikan nilai 10.

c. Pengujian model dengan parameter rujukan

Pengujian dilakukan dengan *hyper-parameters* berdasarkan rujukan dari jurnal-jurnal yang telah dipaparkan diatas, *hyper-parameters* dapat dilihat pada tabel 10.

Tabel 10. Daftar nilai *hyper-parameters* rujukan juml terkait

Matriks	Hyper-parameters)	Nilai
LDA	Alpha	0.1
	Beta	0.1
	Jumlah Iterasi	50
	Jumlah Topik	50
SVM	C	10
	Gamma	1

Hasil performa metode dapat dilihat pada tabel 11.

Tabel 11. Hasil *confusion matrix* pada percobaan optimasi metode menggunakan *hyper-parameters* rujukan

Matriks	Hasil (K-Fold)	Hasil (<i>training-test split</i> 70 : 30)
<i>accuracy</i>	70.33 %	71.94 %
<i>precision</i>	63.86 %	68.19 %
<i>recall</i>	60.59 %	62.73 %
<i>F1 Score</i>	60.46 %	64.78 %

Dapat dilihat terdapat kenaikan performa secara keseluruhan matrix dibandingkan metode yang menggunakan *hyper-parameters* bawaan scikit-learn.

3.6. Percobaan Optimasi Model

Percobaan berikutnya akan dilakukan dengan mengatur *hyper-parameters* pada metode LDA secara manual dan melihat hasil performa pada saat dilakukan klasifikasi. Pada setiap percobaan dengan *dataset* yang sudah dipaparkan, *hyper-parameters* SVM selalu didapatkan 10 untuk C dan 1 untuk γ dari hasil pencarian menggunakan algoritma *grid search*. Sehingga optimasi metode secara manual hanya pada metode LDA. *Hyper-parameters* rujukan akan menjadi nilai awal dalam menjalankan percobaan dan nilai *hyper-parameters* yang ditampilkan merupakan perubahan yang menyebabkan peningkatan pada performa metode. Perubahan *hyper-parameters* akan dihentikan jika terjadi penurunan performa pada saat nilai *hyper-parameters* dinaikan dan diturunkan:

3.7. Hasil Akurasi Optimasi Model

Setelah melakukan optimasi metode dengan mengatur nilai *hyper-parameter* pada metode *latent dirichlet allocation* dan *support vector machine*, maka didapatkanlah nilai performa dengan *hyper-parameters* optimal pada *dataset* metode LDA dengan Jumlah topik = 61, Jumlah iterasi = 51, Alpha (α) = 0.6, Beta (β) = 0.001 dan pada metode SVM $C = 10$ dan $\gamma = 1$ pada metode K-Fold yang dapat dilihat pada Tabel 12.

Pada metode *training-test split* 70:30, metode LDA dapat dioptimasi lagi *hyper-parameters* nya sehingga dapat mendapatkan akurasi sampai 79.85% dan semua matrix meningkat, namun dengan mengubah nilai *hyper-parameters* LDA tersebut, hasil pada K-Fold menurun. Nilai *hyper-parameters* yang digunakan pada metode LDA Jumlah topik = 61, Jumlah iterasi = 51, Alpha (α) = 0.6, Beta (β) = 0.001 dan pada metode SVM $C = 10$ dan $\gamma = 1$. Hasil pengujian tanpa K-Fold dapat dilihat pada tabel 12.

Tabel 12. Hasil *confusion matrix* pada percobaan setelah percobaan dengan optimasi *hyper-parameter*

Matriks	K-Fold	Split-test
<i>accuracy</i>	76.62 %	79.86 %
<i>precision</i>	76.21 %	79.98 %
<i>recall</i>	70.73 %	72.38 %
<i>F1 Score</i>	69.85 %	74.68 %

Berikut salah satu hasil klasifikasi dengan 5 sampel data dengan ketepatan klasifikasi 3 data dan kesalahan klasifikasi 2 data yang dapat dilihat pada tabel 13 berikut :

Tabel 13. Hasil klasifikasi dari sampel data uji

Laporan	Distribusi nilai topik	Label	Prediksi
Tadi saya ikuti acara lapor baiman show di fakultas fisip	[0.01315789 0.03508772 0.01315789 0.01315789]	1 (Dinas Pekerjaan dan Perumahan Rakyat)	1 (Dinas Pekerjaan dan Perumahan Rakyat)
Saya ingin melaporkan mengenai lampu traffic light ...	[0.01209974 0.01119403 0.01119403 0.01217532]	2 (Dinas Perhubungan Kota Banjarasin)	3 (Satpol PP dan Pemadam Kebakaran)
Kami warga kelurahan surgu mufti mengadu	[0.01209677 0.01209677 0.01209677 0.01227708]	4 (Dinas Lingkungan Hidup)	4 (Dinas Lingkungan Hidup)
Kepada yth dinas pupr kota banjarmasin saya ingin melaporkan....	[0.00670992 0.02748414 0.01691332 0.03993109]	1 (Dinas Pekerjaan dan Perumahan Rakyat)	1 (Dinas Pekerjaan dan Perumahan Rakyat)
Malam minggu saya bawa keluarga ke siring bekantan mau santai	[0.01287554 0.03433476 0.01307748 0.04494619]	3 (Satpol PP dan Pemadam Kebakaran)	1 (Dinas Pekerjaan dan Perumahan Rakyat)

4. KESIMPULAN DAN SARAN

4.1 KESIMPULAN

Berdasarkan dengan penelitian yang telah dilakukan adapun simpulan yang dapat ditarik adalah sebagai berikut:

1. Dapat dibangunnya prototipe berbentuk webservice yang dibangun menggunakan bahasa pemrograman Python dengan framework Masonite. Pada proses feature extraction digunakan metode LDA topic modeling dan pada proses klasifikasi digunakan metode SVM menggunakan library Scikit-learn.

2. Penerapan Metode LDA dan SVM dapat mengklasifikasikan data uji laporan dengan benar sesuai instansi dengan performa yang cukup baik yaitu *accuracy* sebesar 76.18%, *precision* sebesar 75.78%, *recall* sebesar 70.44% dan *F1 Score* sebesar 69.36% dengan menggunakan K-Fold dengan Fold berjumlah 10. Dan *accuracy* sebesar 79.85%, *precision* sebesar 79.98%, *recall* sebesar 72.37% dan *F1 Score* sebesar 74.67% training-test split 70 : 30.

4.2 SARAN

Adapun saran yang dapat diberikan oleh peneliti adalah sebagai berikut:

1. Menambahkan *data training* menjadi lebih banyak sehingga model dapat memberikan performa yang lebih baik.
2. Menambahkan data yang lebih seragam dan jumlah laporan lebih terdistribusi dengan baik per-instansi. Sehingga klasifikasi dapat dilakukan pada semua instansi.
3. Terdapat fungsi yang dapat mengubah kalimat ber-bahasa daerah secara otomatis tanpa diubah secara manual.

DAFTAR PUSTAKA

ARUN, R., SURESH, V., MADHAVAN, C. E. V. & MURTY, M. N., 2010. *On Finding the Natural Number of Topics with*. Berlin, s.n.

ASUNCION, A., WELLING, M., SMYTH, P. & TEH, Y. W., 2009. *On Smoothing and Inference for Topic Models*. Montreal, s.n.

BARDENET, R., BRENDEL, M., KEGL, B. & SEBAG, M., 2013. *Collaborative hyperparameter tuning*. Atlanta, s.n.

BELEITES, C. dkk., 2012. Sample Size Planning for Classification Models. *Analytica Chimica Acta*, Volume 760, pp. 25-33.

BLEI, M. D., NG, Y. A. & JORDAN, I. M., 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, Volume 3, pp. 993-1022.

CLAESEN, M. & MOOR, B. D., 2015. *Hyperparameter Search in Machine Learning*. Agadir, s.n.

CUI, L. dkk., 2014. A Hierarchy Method Based on LDA and SVM for News Classification. *IEEE International Conference on Data Mining Workshop*, Volume 2015, pp. 60-64.

DENNY, M. J. & SPIRLING, A., 2018. Text Preprocessing For Unsupervised Learning: Why It. *Political Analysis*, 26(2), pp. 168-189.

DING, J. & JIN, W., 2019. *A Prior Setting that Improves LDA in both Document Representation and Topic Extraction*. Budapest, s.n.

GEORGE, M., SOUNDARABAI, B. & KRISHNAMURTHI, K., 2017. Impact of Topic Modelling Methods and Text Classification Techniques in Text Mining: a Survey. *International Journal of Advances in Electronics and Computer Science*, 4(3).

GOJAREA, S., JOSHI, R. & GAIGAWARE, D., 2015. Analysis and Design of Selenium WebDriver Automation Testing. *2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)*, Volume 50, pp. 341-346.

HARRINGTON, A. N., 2020. *Hands-on Python Tutorial*. Chicago: Loyola University of Chicago.

HOFFMAN, M. D. & BLEI, D. M., 2010. Online Learning for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems*, Volume 23, pp. 856-864.

JAMES, G., WITTEN, D., HASTIE, T. & TIBSHIRANI, R., 2017. *An Introduction to Statistical Learning with Applications in R*. New York: Springer Science+Business.

KAUR, G. & KAUR, E. P., 2017. Novel Approach of Text Classification by SVM-RBF Kernel and Linear SVC. *International journal of Advance Research, Ideas, and Inovations in Technology*, 3(3).

KAUR, J. & BUTTAR, P. K., 2016. A Systematic Review on Stopword Removal Algorithms. *International Journal on Future Revolution in Computer Science & Communication Engineering*, 150(4), pp. 15-17.

Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, n.d. *Lapor!*. [Online] Available at: <https://www.lapor.go.id/> [Accessed 29 11 2019].

Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, n.d. *Tentang LAPOR!*. [Online] Available at: <https://www.lapor.go.id/tentang> [Accessed 29 11 2019].

KUHN, M. & JOHNSON, K., 2013. Over-Fitting and Model Tuning. In: *Applied Predictive Modeling*. New York: Springer Science+Business, pp. 69-71.

KUSUMANINGRUM, A. P., 2017. *Optimization Of Support Vector Machine Parameters Using*

- Genetic Algorithm For Microarray Data Classification*. s.l., s.n.
- KUSUMANINGRUM, R., WIEDJAYANTO, M., ADHY, S. & SURYONO, S., 2016. Classification of Indonesian News Articles based on Latent Dirichlet. in *Proceedings of the 2016 International Conference on Data and Software Engineering (ICoDSE)*.
- LI, K. dkk., 2011. Multi-class text categorization based on LDA and SVM. *Procedia Engineering*, Volume 15, pp. 1963-1967.
- LOUKAS, D. K., 2016. *Learning Scrapy*. Birmingham: Packt Publishing.
- MANTOVANI, R. G. dkk., 2015. *Effectiveness of Random Search in SVM hyper-parameter tuning*. Killarney, s.n.
- MARYAM, N. S., 2016. Mewujudkan Good Governance Melalui Pelayanan Publik. *Jurnal Ilmu Politik dan Komunikasi*, VI(1), p. 2.
- MEGAWATI, C., 2015. Analisis Aspirasi dan Pengaduan Di Situs Laporkan dengan Menggunakan Text Mining. pp. 1-3.
- OBSORNE, D. & GAEBLER, T., 1992. Dalam: *Reinventing Government*. California: Addison-Wesley Publ. Co., p. 24.
- PEDREGOSA, F. dkk., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, Volume 12, pp. 2825-2830.
- SAGGAF, S., SAID, M. M. & SAGGAF, W. S., 2018. Dalam: *Reformasi Pelayanan Publik di Negara Berkembang*. s.l.:SAH MEDIA, pp. 26-27.
- SARKAR, S. dkk., 2019. Application of Optimized Machine Learning Techniques for Prediction of Occupational Accidents. *Computers and Operations Research*, Volume 106, pp. 210-224.
- SHEHASAAYEE, A. & THAILAMBAL, G., 2017. Comparison of Classification Algorithms in Text Mining. *International Journal of Pure and Applied Mathematics*, 116(22), pp. 425-433.
- SLAMET, C. dkk., 2018. Web Scraping and Naïve Bayes Classification for Job Search Engine. *IOP Conference Series: Materials Science and Engineering*, Volume 288.
- SYARIF, I., WILLS, G. & BENNET, A. P., 2016. SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. *Telecommunication Computing Electronics and Control (TELKOMNIKA)*, 14(4), pp. 1502-1509.
- TALA, F. Z., 2003. A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia.
- TEH, Y. W., NEWMAN, D. & WELLING, M., 2007. *A Collapsed Variational Bayesian Inference*. Vancouver, s.n.
- VABALAS, A., GOWEN, E., POLIAKOFF, E. & CASSON, A. J., 2019. Machine learning algorithm validation with a Limited Size Sample. *Plos One*, 14(11).
- WALLACH, H. M., MIMNO, D. & MCCALLUM, A., 2009. *Rethinking LDA: Why Priors Matter*. Vancouver, s.n.
- WANG, B. & GONG, N. Z., 2018. Stealing Hyperparameters in Machine Learning. *IEEE Symposium on Security and Privacy*, pp. 36-52.
- YOUNG, T., HAZARIKA, D., PORIA, S. & CAMBRIA, E., 2018. Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Computational Intelligence Magazine*, 13(3), pp. 55-75.

Klasifikasi Laporan Keluhan Pelayanan Publik Berdasarkan Instansi Menggunakan Metode LDA-SVM

ORIGINALITY REPORT

15%

SIMILARITY INDEX

15%

INTERNET SOURCES

7%

PUBLICATIONS

%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

5%

★ 123dok.com

Internet Source

Exclude quotes On

Exclude matches Off

Exclude bibliography On