Klasifikasi Ulasan Berdasarkan Divisi Pada Google Play Menggunakan Metode Hierarchical Dirichlet Process Dan Metode Ensemble

Irham Maulani[1] , Chastine Fatichah [2], Arya Yudhi Wijaya[3]

[123]Teknik Informatika, Institut Teknologi Sepuluh Nopember, Indonesia

[1]6025221017@student.its.ac.id; [2]chastine@if.its.ac.id, arya@if.its.ac.id;

INFO ARTIKEL

Sejarah Artikel
Diterima:
Direvisi:
Diterbitkan:

Kata Kunci
Ekstraksi Fitur
Hierarchical Dirichlet Process
Klasifikasi Teks
Topic Modelling
Metode Ensemble

ABSTRAK

Ulasan yang diberikan oleh pengguna pada aplikasi, dewasa ini menjadi umpan balik yang menjadi jembatan penghubung antara pengembang dan pengguna. Pengalaman secara langsung dalam menggunakan aplikasi dapat menjadi masukan yang dapat membuat aplikasi menjadi lebih baik. Ulasan yang dapat menjadi masukan adalah ulasan yang berkualitas baik dan berhubungan secara langsung terhadap pengalaman pengguna. Terdapat banyak ulasan yang diberikan oleh pengguna yang terbawa emosi dan secara cakupan di luar tanggung jawab dan kewenangan oleh tim pengembang perangkat lunak. Beberapa keluhan banyak terarah ke tim operasional dengan masalah seperti keputusan bisnis yang diambil manajemen, promosi yang tidak jelas, tidak responsif nya *customer service*, dan masalah bisnis lainnya. Sehingga dikhawatirkan terjadi pelimpahan hal di luar tanggung jawab pengembang pada masalah yang seharusnya diterima oleh divisi operasional. Data ulasan yang banyak dan kalimat ulasan memiliki arti bias menyulitkan untuk memahami dan memilah ulasan secara manual, sehingga diharapkan klasifikasi secara otomatis membantu dalam pelimpahan masukan secara tepat pada divisi yang bertanggung jawab. Penelitian ini mengusulkan pendekatan klasifikasi menggunakan metode Ensemble pada dua kelas utama, yaitu divisi pengembangan dan divisi operasi. Setiap ulasan di ekstraksi fitur menggunakan metode *Hierarchical Dirichlet Process* (HDP) karena dapat membantu dalam mengelompokkan ulasan yang memiliki karakteristik arti yang secara sentimen ambigu dan emosional ke dalam topik-topik yang relevan. Ulasan diambil dari Google Play dan dilakukan pelabelan secara manual oleh pakar. Hasil penelitian menunjukkan bahwa dengan menggunakan metode *Gradient Boosting* menghasilkan performa yang lebih baik dibandingkan metode klasifikasi Ensemble lainnya yang diuji dengan menggunakan ekstraksi fitur HDP mendapatkan akurasi 0.63, *precision 0.62*, *recall* 0.55 dan *F1 Score* 0.52. Ekstraksi fitur menggunakan HDP memberikan performa yang lebih baik dibandingkan dengan metode pembanding *Latent Dirichlet Allocating* (LDA).

1. PENDAHULUAN

Google Play adalah toko aplikasi digital yang menyediakan aplikasi untuk perangkat Android. Google Play digunakan oleh perusahaan maupun pengembang aplikasi untuk distribusi aplikasi atau game ke pengguna. Salah satu fitur yang ada pada Google Play adalah ulasan, di mana pengguna dapat memberikan ulasan pada aplikasi yang digunakan. Ulasan pada Google Play khususnya, dapat menjadi umpan balik bahkan menjadi jembatan bagi pengguna untuk memberikan keluhan maupun apresiasi pada fitur maupun pembaruan yang

dilakukan oleh pengembang. Umpan balik yang berkualitas pun dapat menjadi data dalam pengumpulan kebutuhan perangkat lunak [1], [2], [3], [4].

Ulasan pun bukan tanpa kekurangan, banyak ulasan yang diberikan pengguna tidak objektif dan banyak membawa emosi di luar pengalaman secara teknis pada aplikasi yang diulas. Pada penelitian yang dilakukan pada penelitian *Actual rating calculation of the zoom cloud meetings app using user reviews on google play store with sentiment annotation of BERT and hybridization of RNN and LSTM* [5] menyebutkan bahwa pada saat Covid-19, banyak siswa yang alih-alih memberikan ulasan terhadap fitur maupun pembaruan yang dilakukan oleh pengembang. Alih-alih memberikan ulasan dan rating yang buruk karena bug maupun karena masalah teknis, memberikan ulasan dan rating yang buruk karena ketidaksukaan mereka dengan melakukan sekolah secara daring. Hal ini berdampak pada sulitnya pengembang mengetahui tentang bug maupun fitur apa yang diinginkan oleh pengguna.

Seiring dengan meningkatnya popularitas Google Play, maka jumlah pengguna suatu aplikasi semakin meningkat sehingga jumlah ulasan dari pengguna juga semakin bertambah. Menurut studi yang dilakukan Data.ai dan Amal M. Almana, terdapat 255 juta pengunduhan baru pada aplikasi di seluruh dunia dan 113.1 juta terdapat pada Google Play [6]. Sehingga terdapat banyak ulasan dapat membuat sulit untuk mengevaluasi dan memahami pendapat pengguna di luar buruknya kualitas ulasan yang diberikan oleh pengguna. Para pengembang juga akan kesulitan dalam mencari tahu bagaimana meningkatkan kinerja aplikasi berdasarkan dari ribuan komentar tekstual, baik ulasan yang memiliki kualitas baik yang mengulas pengalaman penggunaan aplikasi ataupun ulasan yang memiliki kualitas buruk yang tidak objektif serta membawa emosi dalam pengetikan. Dalam hal bisnis pun, ulasan Google Play menjadi vital karena ulasan sering digunakan sebagai alat yang efektif dan efisien dalam menemukan informasi terhadap suatu produk atau jasa[6]. Sehingga ulasan dan rating pada Google Play juga banyak ditujukan untuk menunjukkan performa dari sebuah aplikasi bahkan menjadi KPI dalam sebuah tim pengembangan. Sedangkan, dalam ulasan sendiri terdapat banyak ulasan pengguna yang tidak hanya ditujukan pada pengalaman penggunaan aplikasi. Terdapat banyak keluhan yang diarahkan di luar tanggung jawab dari tim pengembangan, seperti keluhan mengenai keputusan bisnis manajemen, pelayanan *customer service* yang tidak responsif, serta promosi yang kurang jelas. Hal ini dikhawatirkan dapat terjadi pelimpahan hal di luar tanggung jawab pengembang.

Terdapat banyaknya data ulasan pada Google Play dan beragamnya makna bias dari ulasan dapat menyebabkan sulitnya menemukan arti sebenarnya. Penggunaan metode *topic modelling* juga memberikan hasil akurasi yang lebih bagus dibandingkan hanya menggunakan metode vectorize tradisional seperti TF-IDF sebagai ekstraksi fitur untuk klasifikasi teks [7]. Kombinasi *topic modelling* dan klasifikasi juga cocok untuk data yang banyak dan terdapat banyak topik, sehingga model dapat melakukan ekstraksi fitur dengan lebih baik tanpa keterlibatan manusia di dalamnya [8], [9].

Metode *Hierarchical Dirichlet Process* (HDP) merupakan metode *topic modelling* yang diajukan pada penelitian ini. Metode HDP merupakan pengembangan dari metode *Latent Dirichlet Allocation* (LDA) di mana metode HDP menghasilkan performa yang lebih bagus dibanding metode LDA dalam melakukan *topic modeling* pada teks, terlebih pada kasus di mana data teks akan bertambah secara terus-menerus [10]. Berbeda dengan LDA, HDP tidak perlu menentukan jumlah topik sehingga data bisa terus bertambah tanpa perlu melatih ulang data [11], [12], [13]. Hasil dari penelitian ini menghasilkan topik yang lebih beragam karena jumlah topik yang bisa berubah secara dinamis berdasarkan data digunakan untuk melatih model. Pada penelitian [14], yang melakukan perbandingan topic modeling pada metode Latent Dirichlet Allocation, Hierarchical Dirichlet Process (HDP), Latent Semantic Analysis (LSA) dengan melakukan klaster pada teks menyimpulkan bahwa HDP dengan metode K-

means menghasilkan akurasi yang cukup bagus dibanding dengan metode LSA, dan akurasi yang hampir mirip dengan LDA pada *batch* data *training* pertama.

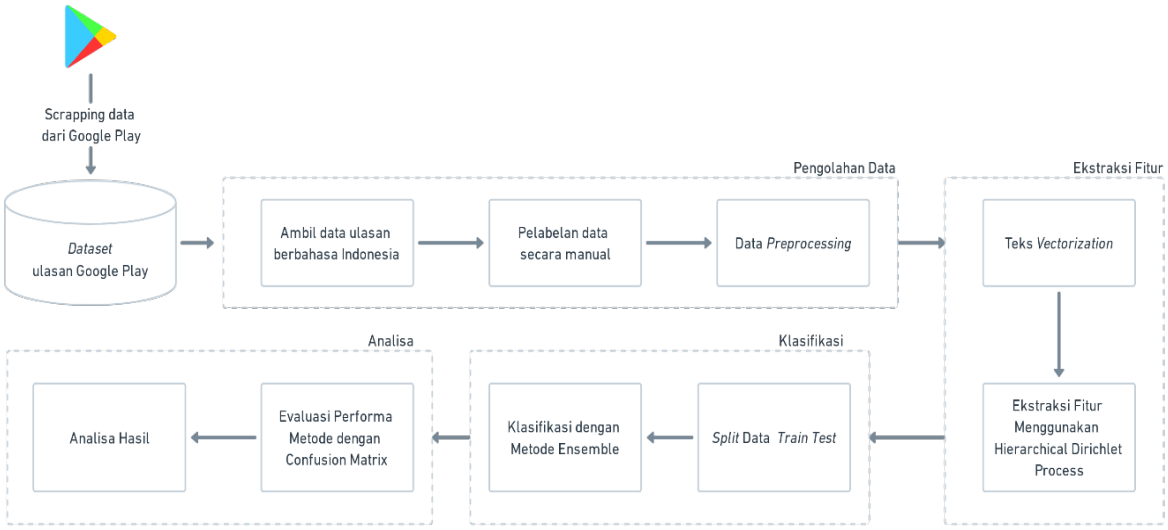
Penelitian ini menggunakan metode Ensemble untuk klasifikasi ulasan. Metode Ensemble adalah istilah umum untuk metode-metode yang menggabungkan beberapa *inducer* untuk membuat keputusan, biasanya dalam *supervise learning*. *Inducer* adalah model atau algoritma *machine learning* yang digunakan sebagai komponen dalam Ensemble. Sebuah *inducer* Ensemble dapat berasal dari berbagai jenis algoritma (misalnya, decision tree, neural network, model regresi linear, dll). Premis utama dari ensemble *learning* adalah bahwa dengan menggabungkan beberapa model, kesalahan dari satu *inducer* kemungkinan akan di kompensasi oleh *inducer* lainnya, dan sebagai hasilnya, kinerja prediksi secara keseluruhan dari ensemble akan lebih baik daripada *inducer* tunggal [15]. Pada penelitian yang dilakukan [16], dilakukan deteksi pada berita untuk menentukan apakah berita tersebut palsu atau tidak. Pada penelitian tersebut, dilakukan komparasi pada beberapa metode, yaitu metode machine learning tradisional (Logistic regression dan KNN), metode Ensemble (*Random Forest*, AdaBoost, dan XGboost), dan Deep learning (CNN dan LSTM). Metode Ensemble XGBoost pada keempat dataset yang digunakan untuk komparasi, mendapatkan performa yang jauh lebih bagus dibanding metode lainnya baik secara akurasi, presisi, recall, dan F1 score. Disusul dengan metode Ensemble boosting lainnya seperti AdaBoost [16]. Pada penelitian yang dilakukan pada [7], [17], [18] yang masing-masing menggunakan teks sebagai teks menyebutkan metode Random Forest, Gradient Boosting Decision Tree, dan XGBoost menghasilkan performa yang bagus dalam melakukan klasifikasi teks. Sehingga pada penelitian ini, tiga metode tersebut digunakan dan dilakukan komparasi untuk mengetahui metode yang menghasilkan performa paling bagus.

Tujuan dari penelitian ini adalah untuk melakukan klasifikasi dari jenis ulasan apa saja yang diberikan oleh pengguna sehingga dapat diketahui permasalahan apa yang dihadapi oleh pengguna sehingga penugasan untuk perbaikan diharapkan dapat lebih mudah diarahkan ke divisi yang bertanggung jawab. Klasifikasi akan dilakukan dengan melakukan ekstraksi fitur pada teks ulasan dengan menggunakan ekstraksi fitur menggunakan metode HDP setelah didapat nilai dari hasil ekstraksi fitur akan dilakukan klasifikasi menggunakan metode klasifikasi Ensemble.

Dataset yang digunakan adalah *dataset* ulasan pada aplikasi di Google Play berbahasa Indonesia. Data ulasan akan menggunakan beberapa aplikasi populer yang menyediakan layanan berbasis jasa seperti pemesanan transportasi dan perdagangan secara elektronik di mana terdapat transaksi antara pelanggan dan pihak penyedia layanan. Pembagian kelas akan berdasar dari beberapa layanan yang disediakan layanan berbasis jasa secara umum, yaitu bagian pengembangan aplikasi (*development*) yang berhubungan dengan kualitas aplikasi seperti aplikasi lambat, *crash*, tampilan yang menyulitkan. Dan bagian operasi (*support*) yang berhubungan dengan bisnis secara umum seperti lamanya proses pengembalian dana, *customer service* yang lambat, dan keputusan bisnis yang mengganggu pengalaman penggunaan jasa.

2. METODE

Pada penelitian ini dilakukan beberapa tahap dalam perancangan sistem untuk klasifikasi ulasan aplikasi pada Google Play. Seperti pada gambar 1, tahap akan dibagi menjadi empat bagian besar yaitu Pengolahan Data, Ekstraksi Fitur, Klasifikasi dan Analisa hasil.



Gambar 1. Diagram Alir Perancangan Sistem

Pengambilan Data

Pada penelitian ini data yang digunakan merupakan data primer yang dikumpulkan dari ulasan berbahasa Indonesia pada aplikasi yang terdapat di Google Play. Data ulasan dikumpulkan pada beberapa aplikasi popular berbasis jasa pada pelanggan yang berbasis di Indonesia. Pengumpulan data menggunakan teknik *scrapping* menggunakan Bahasa *Pemrograman* Python dengan bantuan *library* google-play-scrapper. Beberapa aplikasi yang akan diambil ulasannya adalah aplikasi dengan peringkat tiga teratas pada setiap kategori yang menyediakan jasa, yaitu kategori perjalanan & lokal, belanja, dan keuangan di mana di ketiga kategori ini secara intens menyediakan jasa yang terhubung antara konsumen dan admin di mana terdapat transaksi keuangan di dalamnya. Daftar aplikasi yang akan di *scrapping* beserta jumlah ulasan dapat dilihat pada Tabel 1.

Tabel 1. Daftar Aplikasi Berdasarkan Kategori

Kategori Aplikasi	Nama Aplikasi	Jumlah Ulasan
Perjalanan & Lokal	Grab	11.149.827
	Gojek	5.362.412
	Traveloka	1.700.773
Belanja	Shopee	11.943.865
	Lazada	22.188.575
	Akulaku	2.882.625
Keuangan	Dana	3.878.593
	BRImo BRI	1.137.371
	Easycash – Kredit Dana Online	1.307.963
Total Data		**61.552.004**

Pengolahan Data

Pengolahan data akan terbagi menjadi tiga bagian, yaitu mengambil data ulasan yang sudah dilakukan *scrap* dan mengambil parameter apa yang relevan untuk penelitian, pelabelan data secara manual berdasarkan kelas klasifikasi divisi, dan data *preprocessing*.

1. Ambil Ulasan

Data yang telah di dapat setelah di *scrapping*, akan diolah lebih lanjut. Data teks akan diproses menggunakan bahasa pemrograman lalu akan diambil sehingga berbentuk seperti Tabel 2.

Tabel 2. Struktur Data Setelah Pengolahan Data

Nama Aplikasi	Ulasan
Grab	Mulai gak *worth*, beli makanan pake biaya pemesanan lah, biaya kemasan lah. Padahal …

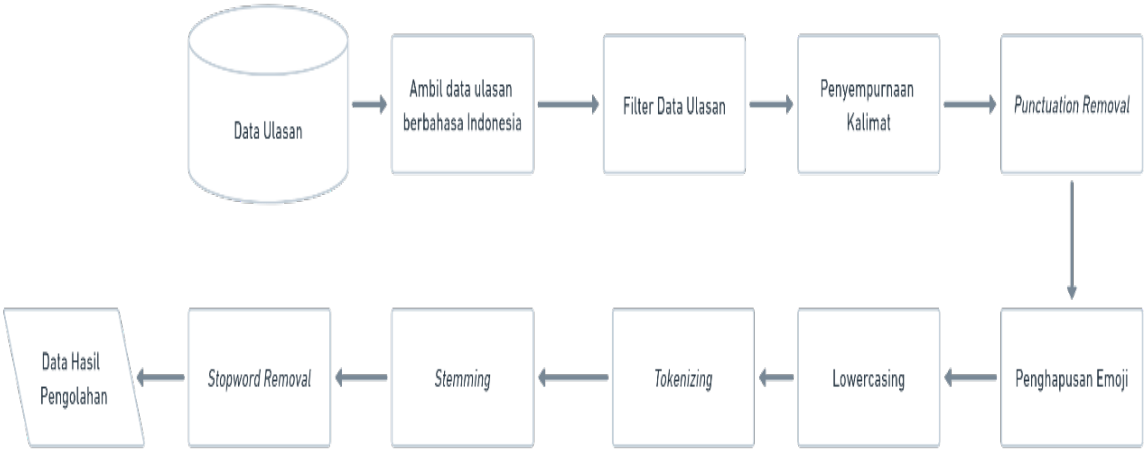
2. Pelabelan Data

Data akan dilakukan pelabelan berdasarkan divisi yang bertanggung jawab. Di mana untuk pelabelan nya sendiri akan dilakukan oleh ahli tata Bahasa Indonesia yang akan menjabarkan secara umum tentang kalimat ulasan data dan ahli rekayasa perangkat lunak yang akan melakukan labeling pada data. Tahapan awal untuk pelabelan data yaitu pelabelan secara manual pada seratus data pada masing-masing kelas. Di mana 100 data diambil sebagai minimal merujuk pada [19] yang mengatakan minimal data yang dibutuhkan tiap kelas klasifikasi adalah lebih dari 75 data. Pelabelan pada data ulasan akan terdapat dua kelas yaitu:

- Divisi *Developer*: bertanggung jawab pada ulasan yang bersifat teknis dalam teknologi dengan kata kunci seperti "*crash*", "*white screen*", "*hang*" atau masalah yang tertuju pada aplikasi nya seperti "aplikasi berat", "aplikasi lemot".
- Divisi Operasional: bertanggung jawab pada ulasan yang berhubungan dengan bisnis secara langsung dengan kata kunci seperti "harga mahal", "pemesanan kurang memuaskan", "pengembalian dana lambat".

Data *Preprocessing*

Data preprocessing adalah proses yang dibutuhkan bagi sebuah data yang berupa teks. Data preprocessing bertujuan agar teks yang diinginkan berkurang kompleksitas tanpa mempengaruhi subtansi dan informasi yang terkandung di dalam teks tersebut [20]. Urutan pada *preprocessing* dapat dilihat pada Gambar 2. Di mana dapat ulasan yang telah disimpan diambil, dan akan dilakukan *filter* data ulasan yang memenuhi kalimat sesuai KBBI. Lalu dimasukan ke penyempurnaan data. Setelah dilakukan penyempurnaan, ulasan akan dilakukan *punctuation removal.* Tahapan berikutnya yaitu dilakukan *lowercasing*. Lalu dilakukan *tokenizing* yang selanjutnya dilakukan *stemming* dan *stopword removal*. Data ini akan disimpan untuk selanjutnya dilakukan proses ekstraksi fitur



Gambar 2. Diagram Alir *Preprocessing*

Langkah *preprocessing* kalimat dengan sampel ulasan dapat dilihat pada Tabel 3.

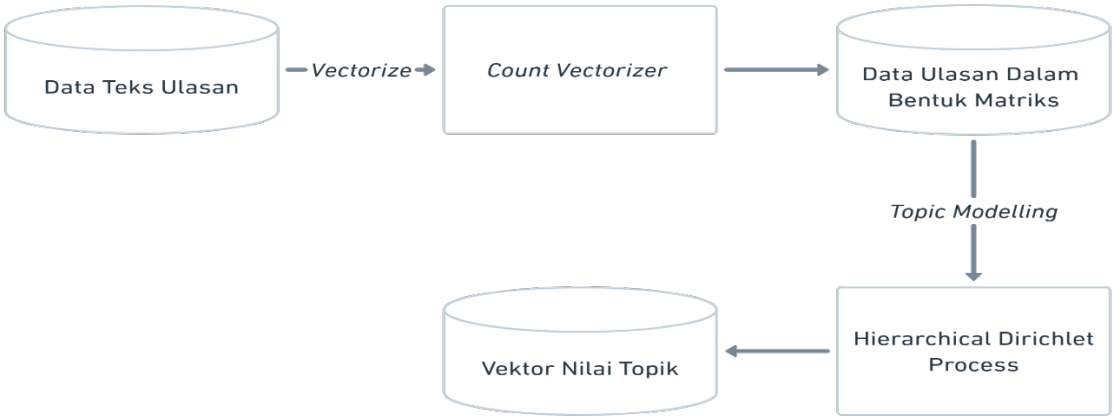
Tabel 3. Proses *Preprocessing Input* dan *Output* ulasan

Proses	***Input***	***Output***
Filter Data Ulasan	Prses pinjaman cpt dana cicilan limit bsr praktis buat belanja, kapan saja. Terima kasih akulaku Semoga limitnya di tambah terus 🤣🤣🤣	Prses pinjaman cpt dana cicilan limit bsr praktis buat belanja, kapan saja. Terima kasih akulaku Semoga limitnya di tambah terus 🤣🤣🤣
Penyempurnaan Kalimat	Prses pinjaman cpt dana cicilan limit bsr praktis buat belanja, kapan saja. Terima kasih akulaku Semoga limitnya di tambah terus 🤣🤣🤣	Proses pinjaman cepat dana cicilan limit besar praktis buat belanja, kapan saja. Terima kasih akulaku Semoga limitnya di tambah terus 🤣🤣🤣
Punctuation Removal	Proses pinjaman cepat dana cicilan limit besar praktis buat belanja, kapan saja. Terima kasih akulaku Semoga limitnya di tambah terus 🤣🤣🤣	Proses pinjaman cepat dana cicilan limit besar praktis buat belanja kapan saja Terima kasih akulaku Semoga limitnya di tambah terus 🤣🤣🤣

Penghapusan *Emoji*	Proses pinjaman cepat dana cicilan limit besar praktis buat belanja kapan saja Terima kasih akulaku Semoga limitnya di tambah terus 🤣🤣🤣	Proses pinjaman cepat dana cicilan limit besar praktis buat belanja kapan saja Terima kasih akulaku Semoga limitnya di tambah terus
Lowercasing	Proses pinjaman cepat dana cicilan limit besar praktis buat belanja kapan saja Terima kasih akulaku Semoga limitnya di tambah terus	proses pinjaman cepat dana cicilan limit besar praktis buat belanja kapan saja terima kasih akulaku Semoga limitnya di tambah terus
Tokenizing	proses pinjaman cepat dana cicilan limit besar praktis buat belanja kapan saja terima kasih akulaku semoga limitnya di tambah terus	["proses", "pinjaman", "cepat", "dana", "cicilan", "limit", "besar", "praktis", "buat", "belanja", "kapan", "saja", "terima", "kasih", "akulaku", "semoga", "limitnya", "di", "tambah", "terus"]
Stemming	["proses", "pinjaman", "cepat", "dana", "cicilan", "limit", "besar", "praktis", "buat", "belanja", "kapan", "saja", "terima", "kasih", "akulaku", "semoga", "limitnya", "di", "tambah", "terus"]	["proses", "pinjam", "cepat", "dana", "cicil", "limit", "besar", "praktis", "buat", "belanja", "kapan", "saja", "terima", "kasih", "akulaku", "semoga", "limit", "di", "tambah", "terus"]
Stopword	["proses", "pinjam", "cepat", "dana", "cicil", "limit", "besar", "praktis", "buat", "belanja", "kapan", "saja", "terima", "kasih", "akulaku", "semoga", "limit", "di", "tambah", "terus"]	["pinjam", "cepat", "dana", "cicil", "limit","belanja", "terima", "kasih", "akulaku", "semoga", "limit", "tambah"]

Ekstraksi Fitur

Proses ekstraksi fitur sehingga data ulasan akan menjadi nilai vektor yang dapat diklasifikasikan dapat dilihat pada Gambar 3.



Gambar 3. Diagram Alir Ekstraksi Fitur

1. Vectorizer

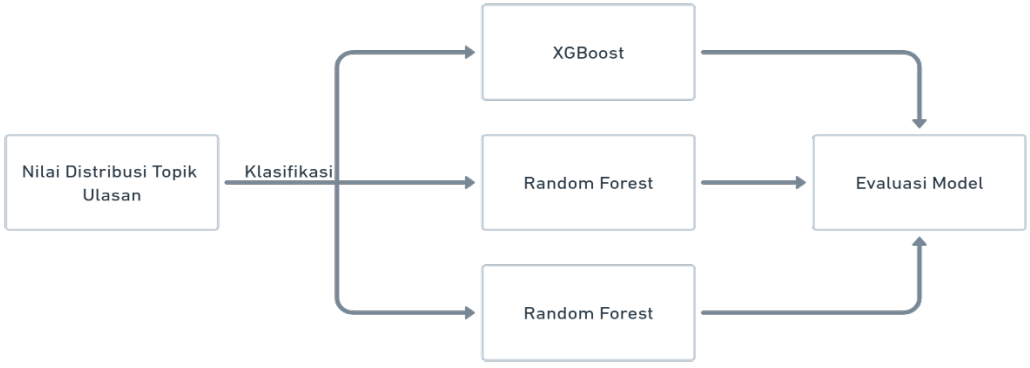
Pada tahap ini akan dilakukan perubahan data ulasan yang telah di *preprocessing* menjadi data matriks. Tahapan ini menggunakan metode *count vectorizer* untuk mendapatkan vektor kata [21], [22].

2. Hierarchical Dirichlet Process

Metode HDP, merupakan metode yang diajukan oleh [23]. Metode HDP merupakan pengembangan dari metode LDA di mana metode HDP menghasilkan performa yang lebih bagus dibanding metode LDA dalam melakukan topic modeling pada teks, terlebih pada kasus di mana data teks akan bertambah secara terus-menerus. Berbeda dengan LDA, HDP tidak perlu menentukan jumlah topik sehingga data bisa terus bertambah tanpa perlu melatih ulang data [11], [12], [13]. Tahap ini akan dilakukan pembobotan kata menjadi persebaran distribusi nilai topik terhadap dokumen. Nilai ini akan digunakan pada pembobotan sebagai masukan dari proses klasifikasi.

Klasifikasi

Data dilakukan *split* 70:30, pada masing-masing 70% untuk data training dan 30% untuk data testing. Setelah itu data dilakukan klasifikasi ke dalam dua kelas yaitu kelas *developer* dan kelas *operation*. Metode klasifikasi yang dipakai adalah metode Ensemble, di mana akan dilakukan klasifikasi menggunakan tiga model yang berbeda yaitu metode Random Forest, Gradient Boosting Decision Tree, dan XGBoost. Alur pada klasifikasi dapat dilihat pada Gambar 4.



Gambar 4. Diagram Alir Klasifikasi

Skenario klasifikasi dengan memasukan vektor kata akan dilakukan seperti Tabel 4.

Tabel 4. Contoh Data Ulasan Hasil Ekstraksi Topik HDP Beserta Label Klasifikasi

Ulasan	Topik 1	Topik 2	Topik 3	Topik 4	Topik 5	Topik 6	Label
Aplikasi yg sangat tdk rekomendasi,tdk sesuai dengan iklan.. Penagihnya sangat meresahkan dan tidak ramah sama sekali kepada pelanggan,	0.03	0.10	0.15	0.30	0.35	0.07	Operasional
Mengapa setiap masuk aplikasi selalu keluar sendiri? padahal memori hp saya tidak sedang penuh, aplikasi sedang tdk ada update, cache juga sudah dihapus, sudah di reinstall	0.07	0.30	0.03	0.15	0.10	0.35	*Developer*

Analisis

Evaluasi performa metode menggunakan *Confusion Matrix*, pengujian dilakukan dengan membandingkan hasil prediksi metode HDP dan metode Ensemble dengan data ulasan Google Play 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).

HASIL DAN PEMBAHASAN

Data yang telah dikumpulkan dengan menggunakan teknik *scrapping* menghasilkan data ulasan dengan jumlah 4.500. Data tersebut dilakukan *filtering* di mana beberapa kalimat tidak memiliki arti seperti hanya terdapat kalimat "ok", *emoticon*, "jos", dan ulasan yang tidak mencapai dua kalimat yang berarti. Sehingga data yang tidak memiliki arti ini dihilangkan sehingga jumlah data menjadi 2.009 data.

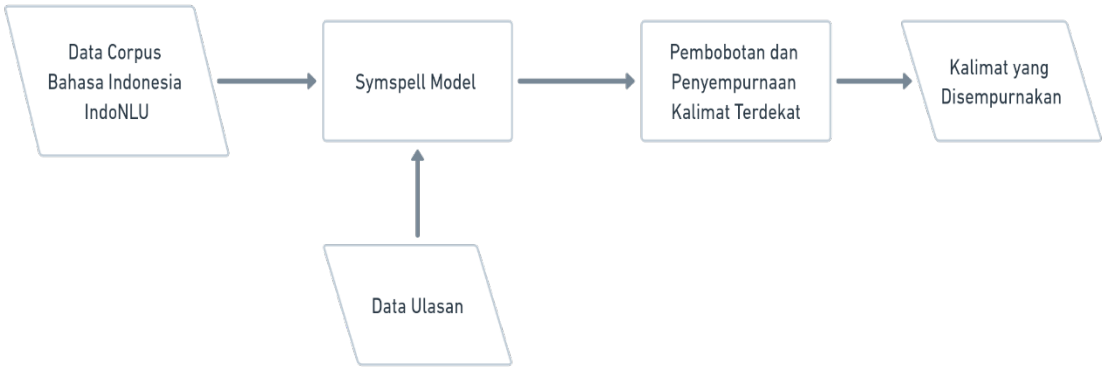
Pelabelan data dilakukan oleh empat orang annotator berusia di atas 20 tahun yang menggunakan acuan yang telah dibuat dari hasil diskusi oleh tenaga ahli di bidang bahasa Indonesia bersama tenaga ahli di bidang Rekayasa Perangkat Lunak. Proses pelabelan tersebut menghasilkan data terbagi menjadi dua label, yaitu *Operation* yang berjumlah 1205 data dan *Developer* yang berjumlah 799 data. Acuan dari pelabelan dapat dilihat pada Tabel 5.

Tabel 5. Acuan Pelabelan Data Ulasan

Operation	***Developer***
Bersifat keluhan terhadap bisnis	Bersifat keluhan terhadap hal teknis
Mengandung kalimat keluhan terhadap layanan yang diberikan	Mengandung kalimat keluhan tentang performa perangkat lunak

Terdiri dari kalimat bersifat umum	Terdiri dari kalimat teknis dan serapan dari Bahasa Inggris seperti "*bug*, *lag*, *username*, *password*, *update*, *error*, *hack*, *crash*, *UI*"

Penyempurnaan Kalimat pada data ulasan menggunakan teknik *Spell Checker*, di mana kalimat yang terdapat pada ulasan akan dilakukan pengecekan dan penyempurnaan kalimat dengan metode *spelling corrector* secara otomatis. Alur dalam penyempurnaan kalimat dapat dilihat pada Gambar 5 (tambahkan gambar output kalimat yang disempurnakan).



Gambar 5. Diagram Alir Penyempurnaan Kalimat

Kalimat yang telah dilakukan penyempurnaan kalimat, selanjutnya dilakukan tahap *preprocessing*.

Ekstraksi fitur dilakukan dengan *vectorizer* yang selanjutnya dilakukan proses HDP untuk mendapatkan distribusi topik antar dokumen. Di mana kalimat yang sering muncul pada topik-topik yang diproses pada metode HDP dapat dilihat pada Gambar 6.

Gambar 6. *Wordcloud* Pada Setiap Topik Kalimat

Hasil dari vektor pada topik yang dihasilkan HDP akan dimasukan ke proses klasifikasi pada metode ensemble. Hasil dari klasifikasi dengan metode Ensemble menggunakan ekstraksi fitur HDP menghasilkan nilai yang dapat dilihat pada Tabel 6.

Tabel 6. Perbandingan Performa Metode Ekstraksi Fitur dan Klasifikasi

Ekstraksi Fitur	Metode	Akurasi	Precision	Recall	F1 Score
HDP	XGBoost	0.61	0.57	0.54	0.52
	Random Forest	0.60	0.55	0.53	0.52
	Gradient Boosting	0.63	0.62	0.55	0.52
LDA	XGBoost	0.58	0.55	0.55	0.55
	Random Forest	0.55	0.53	0.53	0.53
	Gradient Boosting	0.62	0.60	0.57	0.56

Dari Tabel 6, dapat dilihat pengujian dilakukan menggunakan ekstraksi fitur HDP dan LDA lalu dilakukan klasifikasi dengan metode Ensemble. Pada pengujian, metode HDP memiliki nilai yang lebih baik pada akurasi dan *precision* dibandingkan metode LDA yang memiliki nilai *recall* dan *F1 Score* yang lebih baik pada semua metode Ensemble.
Pada Tabel 7 dapat dilihat hasil dari *confusion matrix* ketiga metode Ensemble dengan ekstraksi fitur HDP.

Tabel 7. Hasil *Confusion Matrix*

		Prediksi	
Metode	**Aktual**	**Developer**	**Operation**
XGBoost	**Developer**	53	185
	Operation	48	317
Random Forest	**Developer**	59	179
	Operation	62	303
Gradient Boosting	**Developer**	49	189
	Operation	32	333

Pada Tabel 8, dapat dilihat skenario menggunakan data *training* sebagai validasi pengujian untuk melihat model apakah *overfitting* atau tidak.

Tabel 8. Perbandingan Performa Pada Metode Dengan Data Training

Ekstraksi Fitur	Metode	Akurasi	Precision	Recall	F1 Score
HDP Dengan Data Training	XGBoost	0.67	0.74	0.60	0.57
	Random Forest	0.71	0.81	0.65	0.64

	Gradient Boosting	0.66	0.75	0.59	0.56
LDA Dengan Data Training	XGBoost	0.90	0.91	0.89	0.90
	Random Forest	0.99	0.98	0.99	0.99
	Gradient Boosting	0.66	0.75	0.59	0.56

Dapat dilihat pada Tabel 8, rentang nilai performa pengujian dengan data *testing* dan data *training* sebagai validasi pengujian menghasilkan nilai yang tidak terlalu jauh. Sehingga dapat disimpulkan untuk model tidak terdapat *overfitting* dibandingkan dengan menggunakan metode LDA yang menghasilkan nilai yang jauh dibandingkan pengujian dengan data *testing* sehingga bisa didapatkan bahwa model LDA menghasilkan *overfitting*,

KESIMPULAN

Kesimpulan dari penelitian ini adalah pengklasifikasian data teks ulasan pada Google Review berdasarkan divisi yang terlibat, menghasilkan perbandingan kinerja antara model HDP sebagai ekstraksi fitur dengan metode klasifikasi ensemble sebagai berikut: Metode XGBoost, Random Forest dan Gradient Boosting menghasilkan nilai yang konsisten pada situasi hasil test yang diujikan dibandingkan dengan metode pembanding LDA sebagai ekstraksi fitur. HDP yang merupakan metode pengembangan dari LDA dapat menghasilkan hasil konsisten pada klasifikasi ulasan pada aplikasi di Google Play dan menunjukkan bahwa dengan menggunakan metode Gradient Boosting, ekstraksi fitur menggunakan HDP menghasilkan performa yang lebih baik dibandingkan metode klasifikasi ensemble lainnya yang diuji dengan keseluruhan performa akurasi 0.63, precision 0.62, recall 0.55 dan F1 Score 0.52. Ekstraksi fitur menggunakan HDP memberikan performa yang lebih tinggi dibandingkan dengan metode LDA dan HDP memberikan nilai yang tidak terpaut jauh pada saat pengujian menggunakan data *training* dibandingkan dengan LDA. Nilai yang tidak terpaut jauh tersebut menunjukkan HDP dalam melakukan ekstraksi fitur pada ulasan tidak terjadi *overfitting*.

REFERENSI

[1] L. Al-Safoury, A. Salah, and S. Makady, "Integrating User Reviews and Issue Reports of Mobile Apps for Change Requests Detection," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, 2022, doi: 10.14569/IJACSA.2022.0131248.

[2] A. A. Al-Subaihin, F. Sarro, S. Black, L. Capra, and M. Harman, "App Store Effects on Software Engineering Practices," *IEEE Transactions on Software Engineering*, vol. 47, no. 2, pp. 300–319, Feb. 2021, doi: 10.1109/TSE.2019.2891715.

[3] J. Dąbrowski, E. Letier, A. Perini, and A. Susi, "Analysing app reviews for software engineering: a systematic literature review," *Empir Softw Eng*, vol. 27, no. 2, Mar. 2022, doi: 10.1007/s10664-021-10065-7.

[4] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, "A survey of app store analysis for software engineering," *IEEE Transactions on Software Engineering*, vol. 43, no. 9, pp. 817–847, Sep. 2017, doi: 10.1109/TSE.2016.2630689.

[5] Md. J. Islam, R. Datta, and A. Iqbal, "Actual rating calculation of the zoom cloud meetings app using user reviews on google play store with sentiment annotation of BERT and hybridization of RNN and LSTM," *Expert Syst Appl*, vol. 223, p. 119919, 2023, doi: https://doi.org/10.1016/j.eswa.2023.119919.

[6] A. M. Almana and A. A. Mirza, "The Impact of Electronic Word of Mouth on Consumers' Purchasing Decisions," 2013.

[7] Y. Huang, R. Wang, B. Huang, B. Wei, S. L. Zheng, and M. Chen, "Sentiment Classification of Crowdsourcing Participants' Reviews Text Based on LDA Topic Model," *IEEE Access*, vol. 9, pp. 108131–108143, 2021, doi: 10.1109/ACCESS.2021.3101565.

[8] N. Alghamdi, S. Khatoon, and M. Alshamari, "Multi-Aspect Oriented Sentiment Classification: Prior Knowledge Topic Modelling and Ensemble Learning Classifier Approach," *Applied Sciences (Switzerland)*, vol. 12, no. 8, Apr. 2022, doi: 10.3390/app12084066.

[9] M. George, P. B. Soundarabai, K. Krishnamurthi, M. Scholar, A. Professor, and A. Professor, "IMPACT OF TOPIC MODELLING METHODS AND TEXT CLASSIFICATION TECHNIQUES IN TEXT MINING: A SURVEY," 2017. [Online]. Available: http://iraj.in

[10] Y. W. Teh, D. Newman, and M. Welling, "A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation," *Adv Neural Inf Process Syst*, pp. 1353–1360, 2007, doi: 10.7551/mitpress/7503.003.0174.

[11] W. Hong, X. Zheng, J. Qi, W. Wang, N. Zheng, and Y. Weng, "FinancialFlow: Visual analytics of financial news based on hierarchical dirichlet process," in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2018, pp. 375–380. doi: 10.1109/YAC.2018.8406403.

[12] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes," 2005.

[13] C. Wang, J. Paisley, and D. M. Blei, "Online Variational Inference for the Hierarchical Dirichlet Process," 2011.

[14] J. Chakraborty, G. Thopugunta, and S. Bansal, "Data Extraction and Integration for Scholar Recommendation System," in *Proceedings - 12th IEEE International Conference on Semantic Computing, ICSC 2018*, Institute of Electrical and Electronics Engineers Inc., Apr. 2018, pp. 397–402. doi: 10.1109/ICSC.2018.00079.

[15] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, Jul. 2018, doi: https://doi.org/10.1002/widm.1249.

[16] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods," *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/8885861.

[17] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Syst Appl*, vol. 57, pp. 232–247, Sep. 2016, doi: 10.1016/j.eswa.2016.03.045.

[18] F. Giannakas, C. Troussas, A. Krouska, C. Sgouropoulou, and I. Voyiatzis, "XGBoost and Deep Neural Network Comparison: The Case of Teams' Performance," in *Intelligent Tutoring Systems*, A. I. Cristea and C. Troussas, Eds., Cham: Springer International Publishing, 2021, pp. 343–349.

[19] C. Beleites, U. Neugebauer, T. Bocklitz, C. Krafft, and J. Popp, "Sample size planning for classification models," *Anal Chim Acta*, vol. 760, no. June 2012, pp. 25–33, 2013, doi: 10.1016/j.aca.2012.11.007.

[20] M. J. Denny and A. Spirling, "Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do about It," *Political Analysis*, vol. 26, no. 2, pp. 168–189, 2018, doi: 10.1017/pan.2017.44.

[21] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011, [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[22] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nat Methods*, vol. 17, no. 3, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.

[23] S.-M. J. Wong, M. Dras, and M. Johnson, "Topic Modeling for Native Language Identification," *Proceedings of the Australasian Language Technology Association Workshop 2011*, pp. 115–124, 2011, [Online]. Available: http://www.aclweb.org/anthology/U/U11/U11-1015