



TESIS

METODE *EMBEDDING LSTM* BERBASIS *SIMILARITY CLUSTER LDA* UNTUK KLASIFIKASI PERUBAHAN PERANGKAT LUNAK PADA *MOBILE APP REVIEW*

**ALIFIA PUSPANINGRUM
NRP. 05111650010040**

DOSEN PEMBIMBING

Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng.

NIP: 19741123 200604 1 001

Dr. Chastine Fatichah, S.Kom., M.Kom.

NIP. 19751220 200112 2 002

**PROGRAM MAGISTER
DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M. Kom)
di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:
ALIFIA PUSPANGRUM
Nrp. 05111650010040

Dengan judul:
Metode Embedding LSTM Berbasis Similarity Cluster LDA untuk Klasifikasi Perubahan
Perangkat Lunak pada Mobile App Review

Tanggal Ujian : 20-7-2018
Periode Wisuda : 2017 Genap

Disetujui oleh:

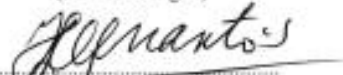
1. Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng.
NIP. 197411232006041001


(Pembimbing I)

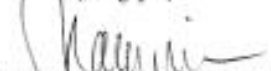
2. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
NIP. 197512202001122002


(Pembimbing II)

3. Prof. Ir. Drs. Ec. Rivanarto Sarno, M.Sc., Ph.D.
NIP. 195908031986011001


(Penguji I)

4. Dr. Ir. Raden Venantius Hari Ginardi, M.Sc.
NIP. 196505181992031003

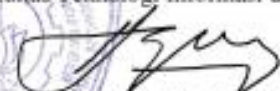

(Penguji II)

5. Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.
NIP. 1984101620081210002


(Penguji III)



Dekan Fakultas Teknologi Informasi dan Komunikasi,


Dr. Agus Zainal Arifin, S.Kom, M.Kom
NIP. 19720809 199512 1 001

METODE EMBEDDING LSTM BERBASIS SIMILARITY CLUSTER LDA UNTUK KLASIFIKASI PERUBAHAN PERANGKAT LUNAK PADA MOBILE APP REVIEW

Nama mahasiswa : Alifia Puspaningrum
NRP : 05111650010040
Pembimbing : Daniel Oranova Siahaan, S.Kom, M.Sc, PD.Eng.
Dr. Chastine Fatichah, S.Kom, M.Kom.

ABSTRAK

Seiring dengan perkembangan aplikasi bergerak, perbaikan dan evolusi perangkat lunak menjadi salah satu hal yang wajib untuk dilakukan. Salah satu input yang dapat digunakan dalam proses tersebut diantaranya adalah pengalaman pengguna dalam menggunakan produk. Review produk dapat menjadi salah satu pendekatan untuk memetakan perubahan yang terjadi pada perangkat lunak. Beberapa jenis kategori perubahan perangkat lunak yang sering digunakan sebagai pemetaan diantaranya adalah *Bug Error*, *Feature Request*, dan *Non Informative*.

Latent Dirichlet Allocation (LDA) merupakan salah satu metode *topic modelling* yang mampu memetakan hidden topik dari suatu dokumen. Namun, LDA memiliki kekurangan dalam mengkategorikan dokumen ke dalam salah satu tipe dari kategori klasifikasi secara langsung. Penelitian ini mengusulkan metode yang menggabungkan LDA dengan *similarity clustering* untuk melabeli dokumen. Selanjutnya, hasil yang diperoleh akan diproses pada proses klasifikasi. Adapun perluasan term list pada saat pengukuran similarity menggunakan metode *Term Frequency – Inverse Cluster Frequency* (TF-ICF). Selain itu, pendeknya kalimat serta sensitivitas semantik menjadi hal yang terkadang menghambat kinerja klasifikasi teks pendek. Oleh karena itu, penelitian ini mengusulkan metode klasifikasi berbasis word embedding dan Long Short Term Memory (LSTM) untuk mengatasi hal tersebut. Adapun ekstraksi fitur yang digunakan adalah melalui Word Embedding. Hasil dari ekstraksi fitur selanjutnya digunakan dalam proses klasifikasi menggunakan metode LSTM.

Performa dari hasil klasifikasi selanjutnya akan dihitung dengan menggunakan precision, recall, dan F1-score untuk pelabelan dokumen serta akurasi untuk klasifikasi dokumen. Hasil uji coba menunjukkan bahwa metode yang diusulkan dapat melakukan pelabelan serta pengklasifikasian dokumen review perangkat lunak. Hal ini ditunjukkan dengan nilai precision, recall dan f-measure dokumen pelabelan terbaik sebesar 56,2%, 49,5%, dan 50,2%, serta akurasi pengklasifikasian terbaik yang mencapai nilai 90,24%.

Kata kunci: Review Mining, Klasifikasi Perubahan Perangkat Lunak, LSA, LSTM, Word Embedding

[Halaman ini sengaja dikosongkan]

EMBEDDING LSTM BASED ON SIMILARITY CLUSTER LDA FOR SOFTWARE CHANGE CLASSIFICATION ON MOBILE APP REVIEW

Nama mahasiswa : Alifia Puspaningrum
NRP : 05111650010040
Pembimbing : Daniel Oranova Siahaan, S.Kom, M.Sc, PD.Eng.
Dr. Chastine Fatichah, S.Kom, M.Kom.

ABSTRACT

Along with the development of mobile applications, software improvement and evolution is an important things to be done. One of inputs that can be used in that process is user experience in using the product. Product reviews can categorize changes that occur in the software. There are several type of software change categorization that are often used, namely: Bug Error, Feature Request, and Non Informative.

Latent Dirichlet Allocation (LDA) is a topic modeling method which is able to map hidden topics from a document. However, LDA gets difficulty to categorize the documents into one type of classification category directly. This research proposes a method that combines LDA with similarity clustering to label the document which will be classified. Term list expansion during the measurement of similarity uses Term Frequency – Inverse Cluster Frequency (TF-ICF) method. In addition, the short sentence and semantic sensitivity becomes a factor inhibits the performance of short text classifications. Therefore, this study proposes a method of classification based on word embedding and Long Short Term Memory (LSTM) to overcome this problem. The feature extraction used is Word Embedding. The results of feature extraction are used in the classification process using the LSTM method then.

The following classification results is calculated by using precision, recall, and F1-score for document labeling and accuracy for document classification. The performance result shows that the proposed method can perform labeling and classification of software review documents well. This is indicated by the best precision, recall and f-measure labeling values of 56.2%, 49.5%, and 50.2%, and the best classification accuracy reaching 90.24%.

Keywords: Review Mining, Software Change Classification, LSA, LSTM, Word Embedding

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, atas segala rizki, berkah, nikmat serta karunia-Nya yang terlimpahkan kepada penulis, sehingga penulis akhirnya dapat menyelesaikan penelitian dengan judul “Metode Embedding LSTM Berbasis Similarity Cluster LDA untuk Klasifikasi Perubahan Perangkat Lunak pada Mobile App Review”.

Penulis juga ingin mengucapkan banyak terimakasih kepada Disadari sepenuhnya bahwa tanpa bantuan dari berbagai pihak, penelitian ini tidak akan terselesaikan dengan hasil seperti sekarang ini. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan terima kasih yang sebesar-besarnya, kepada :

1. Allah SWT atas limpahan nikmat iman, islam, kesehatan, waktu, serta berbagai kemudahan dari arah yang tidak pernah diduga sebelumnya, sehingga penulis dapat menyelesaikan penelitian ini dengan baik.
2. Keluarga tercinta yang tidak hentinya memberikan dukungan materil, do'a yang tulus, serta *belief* bahwa penulis harus senantiasa menjadi yang insan terbaik yang bermanfaat untuk orang sekitar dimanapun penulis berada.
3. BUDI DN dan Politeknik Indramayu yang memfasilitasi banyak hal dalam masa akademik penulis di Teknik Informatika ITS.
4. Bapak Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng. dan Ibu Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. selaku pembimbing yang telah banyak meluangkan waktu untuk dengan sangat sabar mendidik, membimbing, menjadi teman diskusi, serta mengajak penulis untuk berpikir lebih keras dari biasanya dalam menyelesaikan penelitian ini.
5. Bapak Prof. Ir. Drs. Ec. Riyanarto Sarno, M.Sc., Ph.D., Ibu Dr. Ir. Siti Rochimah, M.T., Bapak Dr. Eng. Radityo Anggoro, S.Kom., M.Sc., dan Bapak Dr. Ir. Raden Venantius Hari Ginardi, M.Sc selaku dosen penguji yang telah memberikan banyak saran dan arahan agar penulis mampu lebih baik dalam menyelesaikan penelitian.
6. Bapak Waskitho Wibisono, S.Kom., M.Eng. Ph.D., selaku Kaprodi S2 Teknik Informatika ITS Surabaya yang memfasilitasi mahasiswanya untuk belajar di Lab S2 hingga larut dalam rangka menyelesaikan penelitian.

7. Seluruh staf dosen, staf tata usaha dan karyawan perpustakaan Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember.
8. Tim Jajan; Mba Amelia, Mba Eva, Mas Ilmi, Mba Ulum, Mas Adhi, Mba Yaya, Mba Ozzy, Mba Rozita, Mba Myrna, Mba Pipit, Mba Herna, Mba Udis, Mba Dian yang senantiasa kuat untuk menguatkan, bahagia untuk membahagiakan, jajan untuk menjajani, serta semangat untuk menyemangati dalam menyegerakan tesis, ibadah, juga makan tepat waktu dalam rangka menjaga kesehatan fisik, jasmani, dan tentunya rohani.
9. Rekan-rekan S2 Teknik Informatika atas bantuan dan diskusi selama penelitian.
10. Rekan Non TC; Silmi, Nisa, Sayyida, dan Macan yang senantiasa memberi semangat dan cerita tentang pengalaman-pengalaman menarik baik yang berfaedah maupun kurang berfaedah sehingga membuat penulis tidak jenuh dalam menyelesaikan penelitian.
11. Semua pihak yang tidak dapat dituliskan satu per satu oleh penulis, terima kasih banyak atas doa dan dukungannya.

Semoga Allah SWT senantiasa menyayangi, menguatkan, memampukan, dan menunjukkan jalan yang terbaik atas semua kebaikan yang telah diberikan. Penulis menyadari bahwa laporan penelitian ini tentunya masih jauh dari kesempurnaan. Oleh sebab itu, saran dan kritik sangat diharapkan untuk perbaikan dimasa yang akan datang. Semoga laporan penelitian ini dapat bermanfaat bagi penulis dan pembaca pada umumnya.

“Allah, i do thank You for all of my failed plan, postponed journey, and frustated scenario, because it brings me to people, something, somewhere, and chance better, bigger, and brighter”

Surabaya, Juli 2018

Alifia Puspaningrum

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1. 1. Latar Belakang	1
1. 1. Perumusan Masalah	6
1. 2. Tujuan Penelitian	6
1. 3. Manfaat Penelitian	6
1. 4. Kontribusi Penelitian	6
1. 5. Batasan Masalah	7
BAB 2 DASAR TEORI DAN KAJIAN PUSTAKA.....	9
2. 1. Klasifikasi Perubahan Perangkat Lunak	9
2. 2. Pengolahan Bahasa Manusia.....	10
2. 3. <i>Mobile App Store</i>	17
2. 4. Penelitian Sebelumnya.....	18
2. 5. Kesimpulan Hasil Studi Komparasi.....	22
BAB 3 METODE PENELITIAN.....	25
3. 1. Studi Literatur	25
3. 2. Pengambilan Data	27
3. 3. Perancangan Sistem	28
3.3.1. Pra Proses	30
3.3.2. Modul Perluasan Term List (Expanding Term List).....	33
3.3.3. Proses Pelabelan.....	35
3.3.4. Klasifikasi Dokumen Review	37
3. 4. Perancangan Pengujian	40
BAB 4 HASIL DAN PEMBAHASAN.....	43

4.1. Implementasi Sistem	43
4.1.1 Deskripsi Data Uji	43
4.1.2 Expanding Term List	45
4.1.3 LDA Clustering	51
4.1.4 Hidden Topic Similarity	52
4.1.5 Klasifikasi	52
4.2. Hasil pengujian dan Analisis	53
4.3.1 Hasil Pengujian Pelabelan Dokumen	53
4.3.2 Hasil Pengujian Klasifikasi	61
4.3.3 Analisa Performa	65
BAB 5 KESIMPULAN DAN SARAN	69
5.1 Kesimpulan	69
5.2 Saran	69
DAFTAR PUSTAKA	71
LAMPIRAN	75
BIOGRAFI PENULIS	81

DAFTAR GAMBAR

Gambar 1.1 Grafik unduh aplikasi pada bursa aplikasi bergerak	1
Gambar 1.2 Contoh Review Aplikasi pada Play Store	3
Gambar 2.1 Contoh Word Embedding.....	13
Gambar 3.2 Contoh Word Embedding.....	14
Gambar 2.3 Arsitektur LSTM.....	15
Gambar 2.4 Perbandingan tren <i>Mobile Operating System Market</i>	17
Gambar 3.1 Alur Metodologi Penelitian.....	25
Gambar 3.2 Alur Usulan Metode Penelitian	28
Gambar 3.3 Pra proses sebagai input pada modul topic learning	30
Gambar 3.4 Topic Learning untuk Pelabelan Dokumen.....	35
Gambar 3.5 Klasifikasi dokumen review.....	38
Gambar 3.6 Contoh Hasil Klasifikasi LSTM.....	39
Gambar 4.1 Contoh data hasil <i>crawling</i>	45
Gambar 4.5 Precision Uji Coba Dokumen Keseluruhan	54
Gambar 4.6 Recall Uji Coba Dokumen Keseluruhan	54
Gambar 4.7 F-Measure Uji Coba Dokumen Keseluruhan	55

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Kategori Klasifikasi Dokumen Review	9
Tabel 2.2 Kata kunci pada klasifikasi Maalej	10
Tabel 2.3 Detail dari embedding yang tersedia secara luas	13
Tabel 2.4 Komparasi Penelitian Sebelumnya berdasarkan Metode Deep Learning	20
Tabel 3.1 Tabel Kata Henti	31
Tabel 3.2 List of Word Normalization	31
Tabel 3.3 List of Word Normalization	33
Tabel 3.4 Tabel Contoh Input dan Output LDA	36
Tabel 4.1 Komposisi Kelas dari 2000 Data	44
Tabel 4.2 Term List dari Setiap Kategori oleh (Maalej 2015).....	46
Tabel 4.3 Hasil Perluasan Term List dari Setiap Kategori menggunakan Wordnet	46
Tabel 4.4 Contoh Perhitungan Term Frequency	47
Tabel 4.5 Contoh Perhitungan Inverse Cluster Frequency (ICF)	47
Tabel 4.6 Contoh Perhitungan TF-ICF	48
Tabel 4.7 Perhitungan TF-ICF Kelas Bug Error	49
Tabel 4.8 Perhitungan TF-ICF Kelas Feature Request	50
Tabel 4.9 Perhitungan TF-ICF Kelas Non Informative	51
Tabel 4.10 Konfigurasi untuk memperoleh <i>The Best Term List</i>	56
Tabel 4.11 <i>The Best Term List</i> untuk Proses Pelabelan	57
Tabel 4.12 Hasil Analisa Penggunaan Parameter Terbaik untuk Setiap Data	57
Tabel 4.13 Confussion Matrix untuk Data 475, 773, dan 1522	58
Tabel 4.14 Contoh Kesalahan Klasifikasi Kelas Bug Report	59
Tabel 4.15 Contoh Kesalahan Klasifikasi Kelas Feature Request.....	60
Tabel 4.16 Contoh Kesalahan Klasifikasi Kelas Non Informative	61
Tabel 4.17 Tabel Komposisi Data Training dan Testing	62
Tabel 4.18 Tabel Rerata Akurasi LSTM dan CNN.....	62
Tabel 4.19 Tabel Rerata Akurasi Penggunaan GloVe pada LSTM dan CNN.....	63
Tabel 4.20 Contoh Kesalahan Klasifikasi pada Pelabelan.....	65

[Halaman ini sengaja dikosongkan]

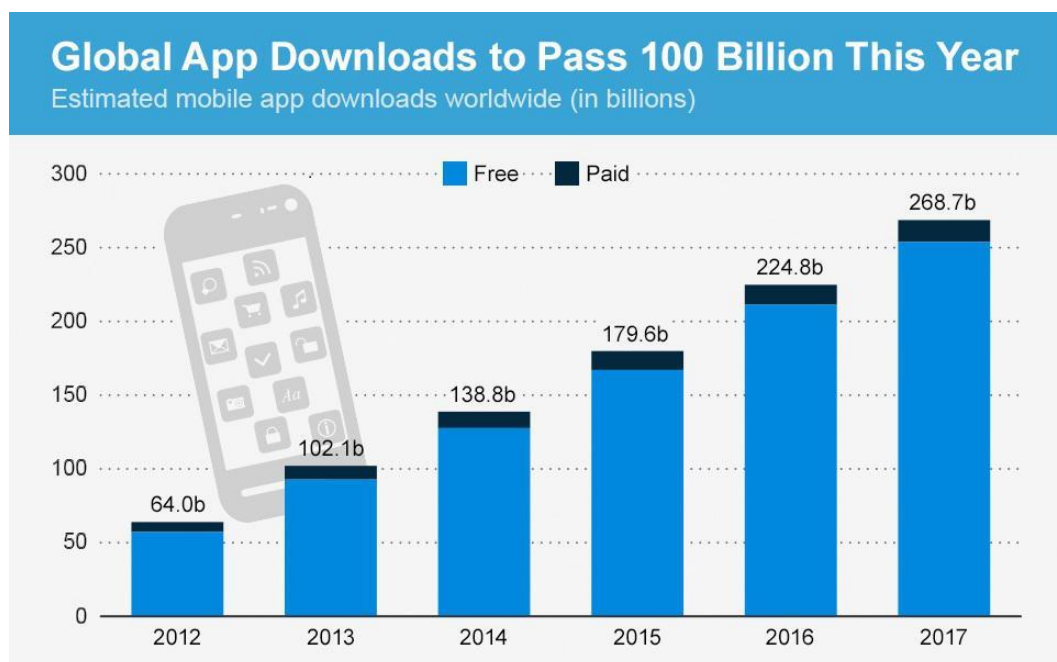
BAB 1

PENDAHULUAN

Pada Bab ini dijelaskan mengenai beberapa hal dasar dalam pembuatan proposal penelitian yang meliputi latar belakang, perumusan masalah, tujuan, manfaat, kontribusi penelitian, dan batasan masalah.

1. 1. Latar Belakang

Saat ini, industri pengembang perangkat bergerak merupakan salah satu industri yang berkembang pesat (Hu & Liu 2004; Chen et al. n.d.). Banyak orang menghabiskan waktu untuk menggunakan perangkat bergerak dengan banyak fitur di dalamnya (Li et al. 2010). Hal ini merupakan salah satu faktor yang mendorong aplikasi bergerak untuk semakin bertambah dari hari ke hari. Hal tersebut selaras dengan infografis yang dideskripsikan pada Gambar 1.1.



Gambar 1.1 Grafik unduh aplikasi pada bursa aplikasi bergerak

Gambar 1.1 menggambarkan bahwa potensi yang dimiliki oleh pengembang aplikasi bergerak sangat luas, ditunjukkan dengan meningkatnya jumlah pengguna yang mengunduh aplikasi selama beberapa tahun kebelakang. Hal tersebut mendorong pengembang untuk senantiasa melakukan perawatan atau perbaikan dari perangkat lunak yang dikembangkan.

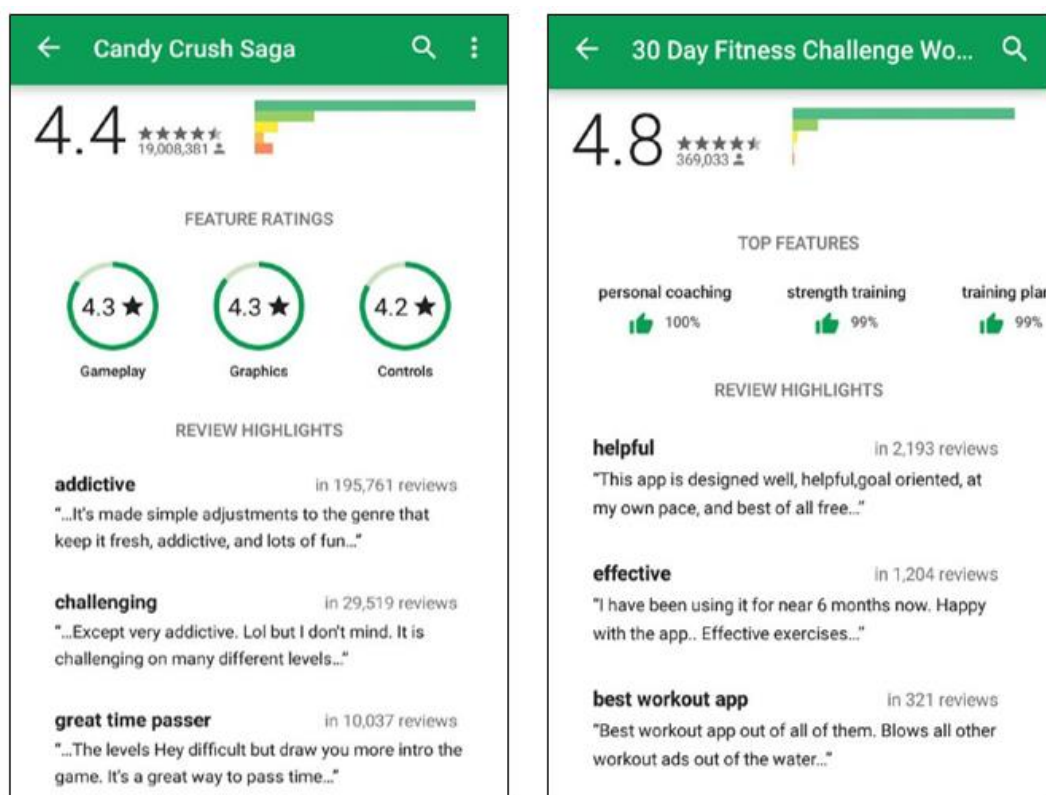
Setiap aplikasi yang telah di-*install* oleh pengguna tidak menutup kemungkinan untuk di-*uninstall*. Hal tersebut mampu dilatarbelakangi oleh banyak hal, seperti tampilan antar muka yang buruk, notifikasi yang mengganggu, proses registrasi yang rumit, dll. Sehingga diperlukan analisa perubahan perangkat lunak yang mampu mendukung dan memetakan proses perbaikan dan evolusi perangkat lunak (Carre & Winbladh 2013).

Analisa pengalaman pengguna dalam menggunakan produk perlu untuk dianalisa lebih lanjut, sehingga informasi yang diberikan oleh pengguna dalam bentuk review produk dapat diolah untuk dijadikan masukan oleh pengembang aplikasi (Liu et al. 2013). Review produk juga dapat membantu desainer untuk memahami kebutuhan serta preferensi pengguna dalam memutuskan untuk membeli atau tidak (Liu et al. 2007). Sejak tahun 2003, peneliti mulai menganalisa dan mengembangkan model inovatif dalam mengembangkan analisa review produk yang saat ini dikenal sebagai penggalian review (Hu et al. 2004), (Liu et al. n.d.), (Xianghua et al. 2013).

Review produk secara umum berbeda dengan survei data konvensional dan pengumpulan data melalui kuisioner atau interview. Review ditulis secara umum berdasarkan kemauan pengguna, sudut pandang ketertarikan, dengan menggunakan bahasa pribadi, serta tidak ada pertanyaan yang menuntun mereka dalam memberikan jawaban. Secara umum, review terhadap produk seringkali lebih beragam dikarenakan ekspresi pengguna yang bervariasi. Selain itu, beberapa review terdiri dari kalimat yang singkat namun cenderung lebih berpusat pada preferensi pengguna (Liu et al. 2013).

Salah satu platform yang juga menggunakan review produk sebagai langkah evaluasinya adalah aplikasi bergerak. Platform berbasis aplikasi bergerak seperti Apple dan Android dapat memberikan kesempatan kepada penggunanya untuk memberikan review dan menuliskan review terhadap aplikasi yang mereka gunakan (Panichella et al. 2015). Review ini dapat dijadikan sebagai jembatan penghubung antara pengembang dan pengguna (Chen et al. n.d.). Melalui jembatan penghubung tersebut, pengguna dapat memberikan informasi yang relevan untuk membantu pengembang dalam melakukan perbaikan atau pun evolusi perangkat lunak, seperti

implementasi fitur terbaru, perbaikan terhadap eror, atau pun peningkatan fitur atau fungsionalitas yang telah ada (Maalej 2015).



Gambar 1.2 Contoh Review Aplikasi pada Play Store

Review produk yang diberikan oleh pengguna tentunya membutuhkan suatu metode untuk memetakan input sehingga dapat menjadi sistem pendukung dalam mengambil keputusan. (Maalej 2015) mengklasifikasikan dokumen user review ke dalam beberapa kategori, seperti bug reports, feature request, user experience, dan rating untuk mengekspresikan pengalaman pengguna dalam menggunakan aplikasi.

Seperti yang ditunjukkan pada Gambar 1.2, review produk terdiri dari banyak dokumen yang direpresentasikan dengan *natural language*. Sehingga proses klasifikasi secara manual tidak akan efektif dan efisien (Panichella et al. 2015). Di satu sisi lain, terdapat banyak penelitian yang telah membahas pemrosesan review online secara otomatis agar proses pengklasifikasian review online dari pengguna dapat lebih terstruktur dan dapat dijadikan sebagai pendukung keputusan desainer untuk melakukan perbaikan perangkat lunak.

Beberapa metode telah berhasil dikembangkan untuk mengklasifikasi suatu review ke dalam salah satu jenis dari kategori perubahan perangkat lunak baik

dengan menggunakan metode *supervised* atau pun *unsupervised*. Metode *supervised* seperti Support Vector Machine (Basari et al. 2013), Logistic Regression (Hamdan et al. 2015) yang diterapkan pada penggalian review produk membutuhkan data dan pelabelan dalam jumlah besar yang menyebabkan penemuan *ground truth* dari dokumen yang dianalisa tidak dapat dilakukan secara manual. Selain itu, metode ini masih menggunakan fitur statistik yang berbasis pada kemunculan term sebagai fitur pada proses klasifikasi sehingga tidak mempertimbangkan aspek semantik dari dokumen. Adapun metode *unsupervised* pun menunjukkan hasil yang tidak terlalu signifikan (Zhai et al. 2011). Salah satu metode *unsupervised* yang kerap digunakan dalam penggalian review online adalah topic modelling. Latent Dirichlet Allocation (LDA) menjadi salah satu metode yang paling banyak dikembangkan dalam membentuk model dari topik yang ada (Brody & Elhadad 2010). LDA mampu memetakan *hidden topic* dari suatu dokumen. Namun, LDA memiliki kekurangan dalam mengkategorikan *hidden topic* ke dalam salah satu tipe dari kategori klasifikasi. Tidak hanya itu, LDA tidak memanfaatkan informasi mengenai distribusi dari kata yang ada (Zhai et al. 2011). Sedangkan, dalam beberapa kondisi, hasil dari LDA perlu untuk digeneralisasi menjadi kategori tertentu yang telah didefinisikan sebelumnya. Oleh karena itu, dalam pengembangannya, LDA perlu menambahkan informasi mengenai domain tertentu untuk meningkatkan performanya (Wu & Chien 2010). Selain itu, solusi ini memiliki kecenderungan untuk memunculkan permasalahan Out of Vocabulary (OOV) disebabkan terbatasnya informasi setiap kategori. (Suadaa 2016) mengusulkan LDA untuk mengkluster dokumen dan menggunakan TF-ICF (Reed 2006) untuk melabeli kluster berdasarkan skor tertinggi untuk setiap kluster. Tidak hanya itu, TF-ICF memiliki kemampuan untuk memunculkan term-term yang berpengaruh pada setiap kluster.

Lexical similarity merupakan salah satu strategi untuk membangun taksonomi dari dokumen dengan menggunakan beberapa metode pengukuran similaritas yang mampu menunjukkan hasil yang baik dibandingkan beberapa metode lainnya (Zhai et al. 2011). Hal tersebut selaras dengan ide (Fu et al. 2015) dalam mengembangkan semi-supervised LDA berbasis *similarity calculation*. Namun, LDA masih merepresentasikan dokumen teks sebagai *bag of words (BoW)*.

Hal ini membuat metode ini kurang efektif dalam menganalisa *fine-grained semantic* pada pemodelan teks pendek. *Fine-grained* pada penggalan review termasuk dalam proses identifikasi review, perhitungan intensiti dan sentimen, serta mengidentifikasi aspek dari suatu review (Liu et al. 2015).

Pada beberapa tahun terakhir, Deep Learning mampu menarik perhatian pada 20 sektor dari pengolahan bahasa manusia, yang secara umum terbagi ke dalam dua area besar (Wang et al. 2016). Sektor pertama adalah mempelajari word embedding dengan melatih training pada model bahasa (Bengio et al. 2003), (Mikolov, Corrado, et al. 2013), (Mikolov, Sutskever, et al. 2013), sedangkan sektor kedua adalah melakukan komposisi semantik untuk memperoleh tingkat representasi dari frasa atau kalimat (Collobert et al. 2011). Salah satu pendekatan deep learning yang mampu secara otomatis mempelajari fitur yang dideskripsikan dalam bentuk vektor adalah Recurrent Neural Networks (RNN). RNN dengan menggunakan word embedding mampu sukses diaplikasikan dalam kasus penggalan review tanpa memerlukan modifikasi proses di dalamnya. Namun, RNN memiliki kelemahan dalam memahami keterhubungan dari suatu sequence yang terpisah dalam jarak yang cukup jauh. Long short term memory (LSTM) pada RNN didesain untuk mampu menutupi kekurangan RNN dalam memodelkan dependensi term yang cukup jauh (Hochreiter & Schmidhuber 1997). Hal ini membuat metode ini sukses diterapkan pada berbagai sektor, diantaranya pemodelan bahasa, pengenalan suara, dan pemahaman bahasa yang diucapkan.

Penelitian ini bertujuan untuk mengembangkan klasifikasi dalam penggalan review pengguna aplikasi bergerak untuk perbaikan serta evolusi perangkat lunak. Secara spesifik, penelitian ini mengusulkan metode LDA berbasis *similarity cluster* untuk memberikan pelabelan dokumen. Hasil dari pasangan dokumen dan label selanjutnya akan diekstrak menjadi *word vector* dengan menggunakan word embedding dan digunakan sebagai input untuk proses klasifikasi perubahan proses perangkat lunak menggunakan metode LSTM. Hal tersebut mampu mendorong penelitian ini agar mampu mengklasifikasi review pada aplikasi bergerak sesuai dengan kategori-kategori analisa perubahan perangkat lunak.

1. 1. Perumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara melabeli topik untuk setiap dokumen review berdasarkan jenis klasifikasi perubahan perangkat lunak
2. Bagaimana cara mengklasifikasi review pengguna menjadi kategori perubahan perangkat lunak secara presisi dan akurat?

1. 2. Tujuan Penelitian

Tujuan yang akan dicapai dalam penelitian ini adalah membangun metode klasifikasi perubahan perangkat lunak dengan menggunakan metode embedding LSTM berbasis *similarity cluster* LDA untuk membantu proses perbaikan atau evolusi perangkat lunak.

1. 3. Manfaat Penelitian

Manfaat dari penelitian ini diantaranya adalah untuk mengembangkan metode klasifikasi Word Embedding Long Short Term Memory (LSTM) berbasis LDA *similarity cluster* dalam melakukan pengklasifikasian perubahan perangkat lunak. Hal tersebut diharapkan dapat membantu pengembang perangkat lunak dalam melakukan perbaikan atau evolusi perangkat lunak.

1. 4. Kontribusi Penelitian

Kontribusi yang diharapkan dari penelitian ini adalah pengklasifikasian review untuk membantu pengembang dalam melakukan perbaikan atau evolusi perangkat lunak. Sebelum melakukan pengklasifikasian, dibutuhkan pelabelan topik. Penelitian ini menggunakan metode Latent Dirichlet Allocation (LDA) untuk penemuan *hidden topic*. *Hidden topic* yang dihasilkan selanjutnya akan dilakukan clustering sesuai dengan topik klasifikasi berdasarkan kemiripan (*word similarity*). Untuk mengatasi keterbatasan term list, penelitian ini akan mengekspansi term list yang ada dengan cara mengambil term-term penting pada setiap kluster dengan menggunakan metode Term Frequency – Inverse Cluster Frequency (TF-ICF). Output dari proses ini adalah dokumen dan label topik sesuai dengan kategori perubahan perangkat lunak.

Pada proses klasifikasi, dokumen maupun label topik akan diekstrak menjadi fitur dengan menggunakan Word Embedding. Sehingga fitur yang akan digunakan pada proses klasifikasi adalah nilai rerata untuk masing-masing vektor kata (*Mean Representation Vector*). Fitur ini lah selanjutnya digunakan pada proses klasifikasi dengan menggunakan LSTM. Hal tersebut diharapkan dapat memberikan hasil dan analisa yang lebih baik dibandingkan metode klasifikasi lainnya untuk dapat memberikan pemetaan kategori perubahan proses perangkat lunak.

1. 5. Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Dataset dalam penelitian ini adalah dokumen review dari aplikasi Waze dan Accuweather yang berada di Google Play Store
2. Dokumen review yang akan diproses adalah dokumen berbahasa inggris
3. Kelas pada klasifikasi yang digunakan mengacu kepada kategori yang dikembangkan oleh (Maalej 2015), yaitu *bug report*, *feature request*, dan *non informative*.

[Halaman ini sengaja dikosongkan]

BAB 2

DASAR TEORI DAN KAJIAN PUSTAKA

Bab ini merupakan pembahasan dari referensi terkait yang telah dilakukan dalam menyelesaikan permasalahan sesuai dengan uraian pada latar belakang. Bab ini diawali dengan menjabarkan hal-hal yang diterapkan pada metode yang diusulkan, kelemahan yang terdapat pada penelitian sebelumnya, komparasi penelitian sebelumnya. Selanjutnya dilanjutkan dengan kelebihan dari metode yang akan digunakan untuk menyelesaikan permasalahan deteksi ketidaklengkapan referensi, metode usulan.

2. 1. Klasifikasi Perubahan Perangkat Lunak

Standar ISO/IEC 14764 mendefinisikan perbaikan perangkat lunak sebagai suatu kegiatan yang diperlukan untuk menyediakan dukungan terhadap suatu sistem perangkat lunak, meliputi kegiatan sebelum pembuatan atau pun setelah pembuatan (Fu et al. 2015).

Terdapat banyak jenis klasifikasi perbaikan perangkat lunak, salah satunya adalah analisa yang diusulkan oleh (Maalej 2015) yang didenifinisikan ke dalam beberapa definisi pada tabel 2.1

Tabel 2.0.1 Kategori Klasifikasi Dokumen Review

Kategori	Kata Kunci
Bug Reports	Kategori yang mendeskripsikan permasalahan pada aplikasi yang perlu untuk diperbaiki, seperti error pada fungsional atau permasalahan terkait dengan performa
Feature Request	Kategori yang mendeskripsikan usulan atau saran untuk meningkatkan performa dari aplikasi pada versi selanjutnya
User Experience	Kategori yang merefleksikan pengalaman pengguna terhadap aplikasi dan fiturnya pada kondisi tertentu
Ratings	Refleksi singkat dari rating dalam bentuk teks

Pada proses perhitungan frekuensi, terdapat list dari term-term klasifikasi oleh Maalej, yang dijelaskan pada Tabel 2.2

Tabel 2.0.2 Kata kunci pada klasifikasi Maalej

Kategori	Kata Kunci
Bug Reports	Bug, fix, problem, issue, defect, crash, solve
Feature Request	Add, please, could, would, hope, improve, miss, need, prefer, request, should, suggest, want, wish
User Experiences	Help, support, assist, when, situation
Ratings	Great, good, nice, very, cool, love, hate, bad, worst

2. 2. Pengolahan Bahasa Manusia

Pemrosesan bahasa alami merupakan teknik untuk mengajarkan komputer dalam memahami maksud dari kata-kata yang digunakan oleh manusia. Metode ini lah yang kemudian diadaptasi oleh bidang Rekaya Perangkat Lunak salah satu nya dalam memproses kategori perubahan dari suatu produk perangkat lunak.

2.2.1. Pre Processing

Salah satu teknik pemrosesan bahasa alami yang kerap digunakan pada pra proses adalah teknik-teknik berikut, yaitu:

- **Tokenisasi**
Pada tahap ini, input teks dokumen dipecah menjadi unit atomis terkecil. Biasanya unit tersebut berupa kata-kata atau kalimat atau paragraf.
- **Normalisasi**
Merubah semua huruf menjadi huruf kecil
- **Penghilangan Tanda Baca**
Menghilangkan tanda baca pada kalimat
- **Stemming**
Stemming memiliki peran untuk menjadikan teks menjadi kata dasar.
- **Stopwords Removal**
Stopwords Removal memiliki peran untuk menghapus kata henti dalam bahasa inggris.
- **Spelling Correction**
Proses ini memiliki fungsi untuk menyempurnakan kalimat yang memiliki kesalahan dalam penulisan.

2.2.2. Term Frequency – Inverse Cluster Frequency (TF-ICF)

TF-ICF (Reed 2006) merupakan salah satu pembobotan term berdasarkan informasi dari dokumen-dokumen pada suatu kluster. Hal tersebut membuat metode ini mampu mengetahui informasi mengenai nilai dari suatu term pada suatu term. Secara umum TF-ICF melihat frekuensi term terhadap kluster dengan menggunakan persamaan 3 dimana nilai ICF pada term i dipengaruhi oleh jumlah cluster yang ada dan jumlah cluster yang mengandung term i (cf_i).

$$ICF_i = 1 + \log \frac{c}{cf_i} \quad (2.1)$$

Sedangkan setiap term i pada setiap kluster akan dihitung nilai bobot TF-ICF nya dengan menggunakan persamaan 4, dimana tf_{ji} adalah frekuensi term i pada cluster j dan ICF_i adalah nilai icf term tersebut.

$$TF - ICF_j = tf_{ji} \times ICF_i \quad (2.2)$$

2.2.3. Latent Dirichlet Allocation (LDA)

LDA (Blei et al. 2003) merupakan model yang mengkategorisasikan kata yang muncul pada korpus dari dokumen-dokumen berdasarkan topik-topik yang telah ditentukan sebelumnya. Untuk setiap dokumen d pada suatu korpus D , LDA mengasumsikan proses-proses berikut:

1. Panjang sample dokumen N_d dari suatu distribusi Poisson Poiss (ξ)
2. Ambil distribusi topik $\vec{\theta}_d$ dari suatu Distribusi Dirichlet Dir (α)
3. Untuk setiap n kata pada N_d kata:
 - a. Pilih suatu topik $z_{d,n}$ dari distribusi multinomial Mult ($\vec{\theta}_d$).
 - b. Pilih suatu kata $w_{d,n}$ dari distribusi multinomial Mult ($\vec{\varphi}_{z_{d,n}}$) dengan distribusi topik ($\vec{\varphi}_{z_{d,n}}$) dari distribusi Dir (β).

Menurut Blei et al, LDA didefinisikan sebagai berikut:

$$p(w, z / \alpha, \beta) = p(w / \alpha, \beta) p(z / \alpha). \quad (2.3)$$

Parameter α dan β merupakan model dari parameter, w adalah kata, dan z adalah topik, $p(z | \alpha)$ adalah probabilitas dari topik z yang terdapat di dokumen d , dan $p(w | z, \beta)$ adalah probabilitas huruf w ada di topik z . Namun komputasi dengan menggunakan persamaan 2.3 cukup sulit untuk dikendalikan. Sehingga Griffiths menggunakan perkiraan persamaan sebagai berikut:

$$p(z_i = k / z_{-i}, w) \propto \frac{n_{-i,k}^{w_i} + \beta}{n_{-i,k} + W\beta} \cdot \frac{n_{-i,k}^{d_i} + \alpha}{n_{-i}^{d_i} + T\alpha} \quad (2.4)$$

Pada persamaan 2.4, $n_{-i}^{(\cdot)}$ merupakan jumlah dari topik z_i , $n_{-i}^{w_i}$ adalah jumlah topik z yang berhubungan dengan huruf w_i , $n_{-i}^{d_i}$ adalah jumlah topik z yang berhubungan dengan dokumen d_i , W adalah jumlah huruf berbeda yang dilakukan pre proses, dan T adalah jumlah topik.

2.2.4. Similarity Clustering

Similarity Clustering merupakan metode yang digunakan untuk mengkluster dokumen berdasarkan perhitungan kemiripan antar dokumen (*semantic similarity*). *Semantic similarity* merupakan metode pengukuran yang mendefinisikan setiap dokumen atau term yang memiliki jarak diantaranya berdasarkan atas makna atau arti secara semantik. Terdapat dua jenis perhitungan similaritas, yaitu berdasarkan sumber daya yang telah ada, seperti thesaurus dan berdasarkan pada penyebaran kata pada suatu corpus.

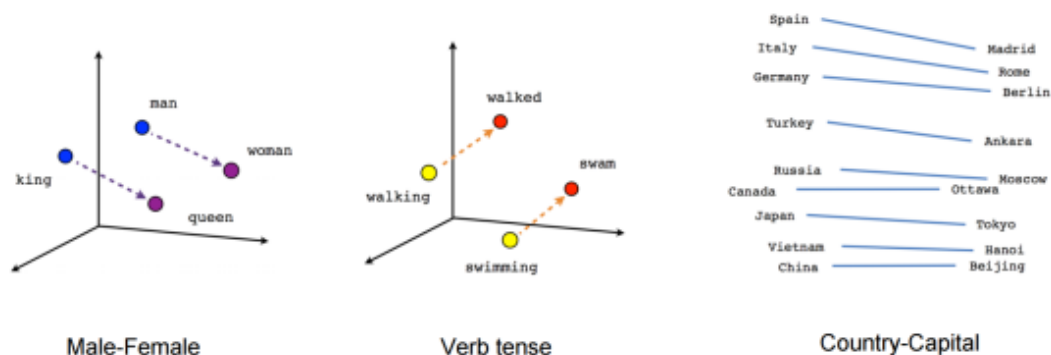
$$Similarity(w_i, w_j) = \frac{\sum_{m=1}^K w_1^m w_2^m}{\sqrt{\sum_{m=1}^K (w_1^m)^2} \sqrt{\sum_{m=1}^K (w_2^m)^2}} \quad (2.5)$$

dimana similarity distance mengukur jarak kedekatan antara $word_1$ dan $word_2$. Nilai maksimal dari jarak similariti adalah 1 yang berarti benar-benar sama, dan nilai minimal adalah -1 yang berarti benar-benar berbeda. Pada kasus sistem kemu kembali informasi, nilai dari similarity dari dua dokumen berada dalam range 0 dan 1 karena term frequency (pembobotan tf-idf) tidak memungkinkan menghasilkan nilai -1.

2.2.5. Word Embedding

Word embedding merupakan kumpulan nama dari pemodelan bahasa dan teknik ekstraksi fitur pada natural language processing (NLP) dimana setiap kata atau phrasa dari suatu kosakata akan dipetakan menjadi vektor yang berupa bilangan real. Word embedding kerap digunakan dalam neural networks, reduksi dimensi pada matriks kemunculan kata, model probabilistik, dll. Metode word embedding

ini juga digunakan sebagai input untuk meningkatkan performa pada pengolahan bahasa manusia seperti parsing sintaktik dan analisa sentimen.



Gambar 2.1 Contoh Word Embedding

(Mikolov, Sutskever, et al. 2013) mengusulkan dua model log-linear untuk menghitung word embeddings dari suatu dataset secara efisien, yaitu bag-of-words dan skip gram. Continuous bag-of words (CBOW) model memprediksi kata saat ini berdasarkan konteks kata. Sedangkan skip-gram memprediksi kata-kata yang berada disekitar kata yang diberikan sesuai dengan kedekatan antara masing-masing vektor kata, seperti yang terlihat pada Gambar 2.1. Saat ini, terdapat banyak ringkasan dari word embedding yang dapat digunakan secara publik, beberapa diantaranya dijelaskan pada Tabel 2.3

Tabel 2.0.3 Detail dari embedding yang tersedia secara luas

Embeddings	Senna	GloVe	Word2Vec
Training Corpus	Wikipedia	Wikipedia/Gigaword	Google News
Dimensionality	50	50	300
Size of Vocab.	130.000	400.000	3.000.000

Senna (Semantic/Syntactic extraction using a neural network architecture) merupakan pengembangan Collobert. Word embedding ini telah dilatih melalui Wikipedia selama dua bulan. Senna juga dapat digunakan sebagai bagian dari part-of-speech (POS) tags, name entity recognition (NER), semantic role labelling (SLR), dan syntactic parsing (PSG).

GloVe (Global Vector) dikembangkan oleh Pennington. Word embedding ini mengusulkan algoritma berbasis unsupervised untuk memperoleh representasi dari

word vector. GloVe pada dasarnya merupakan metode log-bilinear dengan membobotkan least-squares yang telah dilatih melalui enam milyar token corpus yang dikonstruksikan menggunakan Wikipedia2014 dan Gigaword5, dengan kosakata sebanyak 400.000 kata yang sering muncul.

Word2Vec merupakan perangkat yang menyediakan implementasi efisien dalam continuous bag-of-words. Word embeddings ini merupakan bagian dari Google News dataset yang mengandung 300 dimensi vektor untuk tiga juta kata dan frasa.

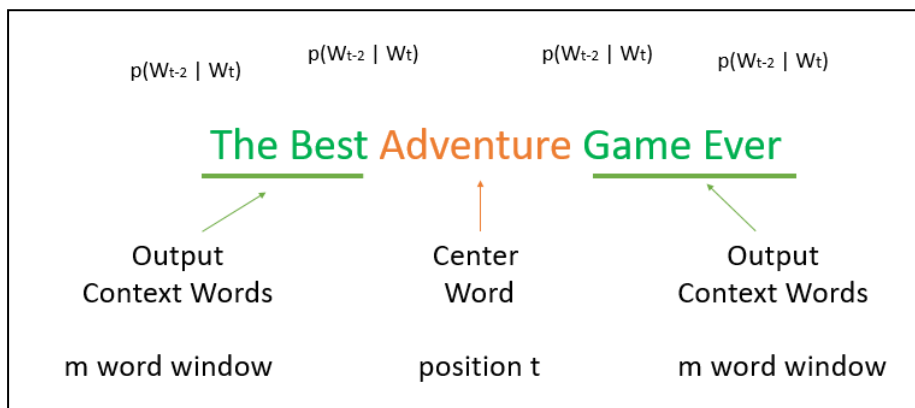
Proses penemuan nilai untuk masing-masing vektor diperoleh dari perbandingan antara suatu kata terhadap seluruh kata dengan menggunakan persamaan (2.6)

$$P(\text{context} / w_t) \tag{2.6}$$

Dengan nilai loss function pada persamaan (2.7)

$$J = 1 - p(w_{-t} / w_t) \tag{2.7}$$

Salah satu contoh penggunaannya ditunjukkan pada Gambar



Gambar 3.2 Contoh Word Embedding

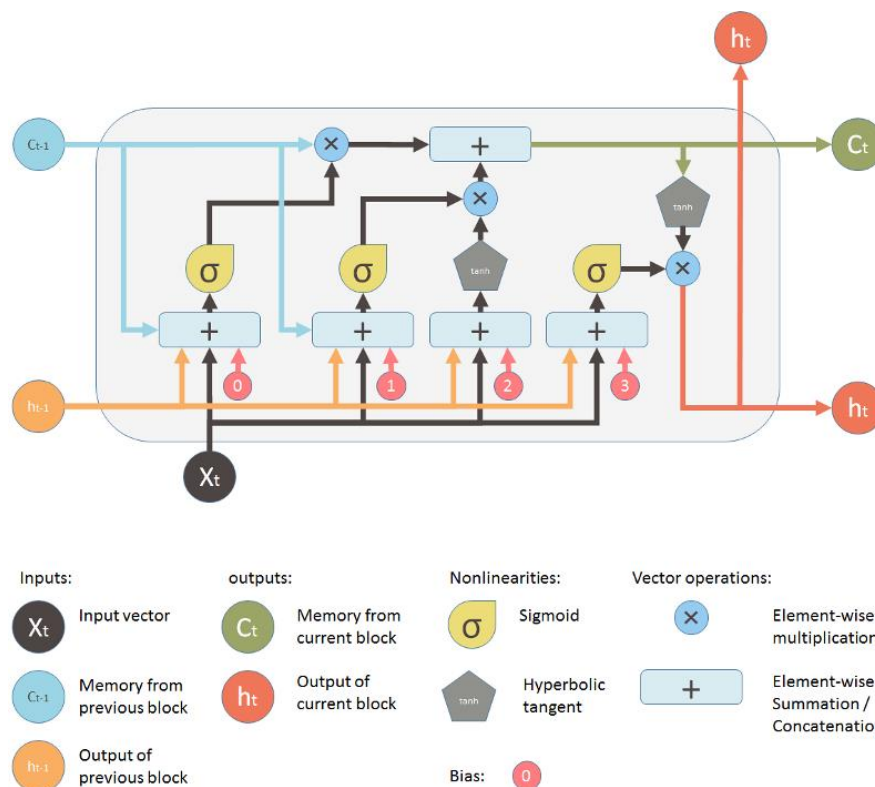
Nilai dari $p(w_{t+j} / w_t)$ dedefinisikan dengan persamaan (2.8)

$$P(o | c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^v \exp(u_w^T v_c)} \tag{2.8}$$

Dimana o adalah output dari indeks kata, c adalah pusat dari indeks kata, v_c dan u_o adalah nilai vektor “tengah” dan “luar” dari kata c dan o . Nilai softmax dengan menggunakan c untuk memperoleh nilai probability dari o . Nilai yang sama cenderung memiliki nilai vektor yang sama. Hasil yang muncul merupakan suatu kumpulan vektor-vektor untuk setiap jenis tipe kata. Hal ini membuat lebih mudah untuk memprediksi kata yang muncul sebagai context words.

2.2.6. Long Short Term Memory

LSTM (Hochreiter & Schmidhuber 1997) adalah arsitektur recurrent neural network (RNN) yang didesain untuk memodelkan keterhubungan antara term yang memiliki interval yang jauh. LSTM telah digunakan secara luas dalam pengolahan bahasa manusia seperti pada analisa sentimen, parsing sintaksis, kategorisasi dokumen yang memiliki ukuran yang panjang, dll. Secara umum, arsitektur dari LSTM digambarkan pada Gambar 2.3



Gambar 2.3 Arsitektur LSTM

LSTM terdiri dari empat elemen yaitu memory cell c , input gate i untuk mengontrol arus input yang masuk ke dalam neuron, output gate o untuk mengontrol efek dari aktivasi neuron pada neuron lainnya, dan forget gate f yang membuat neuron berada dalam status reset dari statusnya saat ini. Secara umum, LSTM terdiri dari beberapa fungsi berikut:

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}) \quad (2.9)$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}) \quad (2.10)$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}) \quad (2.11)$$

$$u_t = \tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}) \quad (2.12)$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \quad (2.13)$$

$$h_t = o_t \odot \tanh(c_t) \quad (2.14)$$

dimana W_k , U_k adalah bobot matrik antara dua hidden layer yang berurutan, antara input dan hidden layer, dan antara dua cell activation yang berurutan, masing-masing, terhubung dengan gate k (contoh: input, output, forget, dan cell), dan b_k adalah nilai vektor bias terkait. Simbol \odot menyatakan nilai produk untuk masing-masing elemen dari dua vektor. Nilai fungsi gate σ merupakan aktivasi sigmoid, dan g dan h adalah aktivasi dari cell input dan cell output, biasanya bernilai tanh.

2.2.1. Evaluasi

Terdapat banyak macam-macam metode evaluasi yang dapat digunakan untuk mengukur kehandalan dari metode yang diusulkan. Proposal ini mengusulkan metode pengukuran dengan menggunakan Presisi, Recall, dan F-1 score. Adapun proses perhitungan dari presisi dan recall ditentukan dari prediksi informasi terhadap nilai sebenarnya yang direpresentasikan dengan True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN).

- Presisi adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2.15)$$

- Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.16)$$

- F1 score adalah

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.17)$$

- Accuracy adalah

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.18)$$

2. 3. *Mobile App Store*

Layanan konten digital yang melingkupi toko daring untuk produk-produk seperti musik, permainan, aplikasi, dll yang dapat diakses baik melalui web, aplikasi telepon genggam, dll. Terdapat banyak *mobile app store* yang dapat digunakan oleh pengguna. Namun, diantara sekian banyak Google Play yang dikembangkan Google dan App Store yang dikembangkan oleh Apple menduduki posisi tertinggi dalam pengunduhan aplikasi beberapa tahun terakhir. Mobile app store berkembang pesat beberapa tahun terakhir dibandingkan dengan mobile app store lainnya seperti Windows Store, Amazon AppStore, dll. Hal tersebut seiring dengan yang ditunjukkan pada Gambar 2.4



Gambar 2.4 Perbandingan tren *Mobile Operating System Market*

Gambar 2.4 menjelaskan mengenai tren pengunduhan aplikasi pada bursa aplikasi bergerak. Meskipun kedua bursa ini memperlihatkan peningkatan, namun dapat terlihat bahwa Google Play menempati posisi yang lebih tinggi dibandingkan dengan App Store dalam pengunduhan. Meskipun Google Play rilis tiga bulan setelah Apple's App Store diluncurkan pada Oktober 2008, Google Play menawarkan sistem operasi yang lebih mudah untuk para pengembang

menawarkan aplikasinya. Selain itu, Android OS menawarkan sejumlah perangkat keras yang lebih terjangkau dibandingkan dengan Apple.

2. 4. Penelitian Sebelumnya

Pada perbandingan penelitian ini, penulis akan membandingkan metode-metode berdasarkan metode klasifikasi serta beberapa metode yang berada pada ruang lingkup penggalian review. Secara garis besar, proses klasifikasi dapat dilakukan dengan banyak metode.

(Maalej 2015) melakukan beberapa pendekatan probabilistik untuk mengklasifikasikan review pengguna dalam empat tipe informasi, yaitu laporan error, permohonan fitur, pengalaman pengguna, serta rating. Untuk memperoleh kategori tersebut, penulis menggunakan metadata pengguna berupa rating serta text review untuk diklasifikasikan dengan menggunakan NLP dan teknik analisa sentimen. Ketika dikombinasikan dengan NLP, precision dari klasifikasi performa yang baik. Hal ini membuktikan bahwa metode yang diusulkan dapat membantu pemilik, pengembang, dan pengguna aplikasi untuk mengumpulkan, memfilter, dan memproses review pada suatu aplikasi. Namun, pada saat proses training data masih membutuhkan pelabelan data dengan bantuan manusia sehingga akan riskan ketika author bukan pengembang aplikasi yang professional. Selain itu, penggunaan string matching yang belum terlalu efektif dibandingkan beberapa metode lain seperti LDA, LSI, dll.

(Panichella et al. 2015) melakukan penelitian tentang taksonomi untuk mengklasifikasikan umpan balik aplikasi ke dalam kategori yang relevan dengan perbaikan atau pun evolusi perangkat lunak. Adapun pendekatan yang dilakukan menggabungkan beberapa teknik berikut: (1) Pengolahan Bahasa Natural, (2) Analisa Teks, dan (3) Analisa sentimen untuk mengklasifikasi umpan balik aplikasi secara otomatis dalam beberapa kategori. Proses pengolahan umpan balik tidak hanya mengandalkan fitur yang diperoleh dari metadata, tetapi juga memasukan unsur *linguistic rule*, analisa teks, sentimen analisis untuk memperoleh informasi mengenai umpan balik yang diberikan oleh pengguna. Paper ini juga melakukan filtrasi umpan balik yang kurang informatif dengan menggunakan sistem perankingan terhadap hasil klasifikasi yang telah diperoleh. Namun, sebaiknya

Kategorisasi dari taksonomi dapat dikembangkan melalui penggunaan pemodelan dari topik-topik yang dapat dikolaborasikan dengan clustering kalimat. Selain itu, struktur dari taksonomi dapat dikembangkan dengan mempertimbangkan tidak hanya struktur kalimat dan sentimen analisis, tetapi juga maksud dari topik yang diusung oleh umpan balik tersebut.

(Fu et al. 2015) melakukan penelitian tentang penggunaan domain software change untuk membuat label pada model semi supervised LDA. Model ini juga menggunakan signifier document dari ahli yang digunakan sebagai dokumen eksternal yang akan diproses similarity antara dokumen ahli dengan dokumen uji. Sehingga proses pertimbangan hidden topic mempertimbangkan aspek similarity antara keduanya. Hal ini tidak hanya dapat mengatasi permasalahan jumlah topik yang ada pada LDA tetapi juga menyediakan sample label pada proses training data. Berdasarkan data percobaan, metode ini dapat secara otomatis mengklasifikasikan sebagian besar pesan perubahan yang merekam bagaimana suatu perangkat berubah.

(Wang et al. 2016) melakukan penelitian terkait Word embedding yang mampu secara semantik menghubungkan kata-kata yang memiliki keterkaitan. Word embedding yang memiliki kedekatan selanjutnya diklaster menjadi kategori yang sama sebagai expanded matrix. Kemudian expanded matrix dan matrix sebenarnya digabungkan untuk selanjutnya diklasifikasi dengan menggunakan CNN. Adapun word embedding yang digunakan diantaranya Senna, GloVe, dan Word2Vec. Setiap dari word embedding dibandingkan kehandalannya terhadap metode yang diusulkan. Hasil percobaan menunjukkan bahwa metode yang diusulkan efektif untuk mengklasifikasi dimana Word2Vec menunjukkan performa terbaik dalam menyediakan kosa kata pada Word Embedding.

(Zhai et al. 2011) melakukan penelitian tentang klastering dengan menggunakan distribusi similarity. Dengan menggunakan distribusi kesamaan antar kata, penulis hendak menyelesaikan permasalahan mengenai adanya kata-kata yang bermakna sama namun direpresentasikan menggunakan kata-kata berbeda oleh pengguna. Oleh karena itu, proses ekstraksi fitur dari produk menggunakan teknik cluster atau pengelompokan sinonim dari kata-kata yang ada. Metode yang diusulkan ini dapat memberikan performa yang baik serta mampu secara otomatis

mengidentifikasi beberapa contoh pelabelan. Pada prosesnya digunakan beberapa pengukuran kemiripan yang merupakan pengembangan dari varian *Least Common Subsumer* (LCS) diantaranya algoritma Jcn, Res, Lin. Namun, untuk dataset yang dianalisa, penulis memilih Jcn karena memiliki performa yang terbaik dibandingkan lainnya.

(Liu et al. 2015) melakukan penelitian tentang penerapan pengembangan model RNN dan word embedding yang sukses diterapkan untuk mengklasifikasi review. Adapun model yang dibandingkan adalah Elman-type RNN, Jordan-type RNN, Long Short Term Memory (LSTM), Bidirectionally. Fine-tuning of Embedding. Adapun word embedding yang digunakan diantaranya adalah SENNA Embeddings, Google Embeddings, Amazon Embeddings. Hasil menunjukkan bahwa ekstraksi review dengan menggunakan word embeddings dapat meningkatkan performa. LSTM RNN mampu menunjukkan performa terbaik pada restoran dataset dan berada di posisi kedua pada dataset laptop dibandingkan dengan metode lainnya dan metode conditional random fields (CRF).

Tabel 2.0.4 Komparasi Penelitian Sebelumnya berdasarkan Metode Deep Learning

Judul	Metode	Label	Dataset	Hasil
Bug Report, Feature Request, or Simply Praise? On Automatically Classifying App Reviews (Maalej 2015)	NLP dan teknik analisa sentimen	laporan eror, permohonan fitur, pengalaman pengguna, serta rating.	1.1juta review dari 1100 aplikasi yang ada di App store dan 146.057 review dari 40 aplikasi yang ada di Google Store	precision dari klasifikasi memperoleh nilai 70 – 95% dan recall 80 – 90%.
How can i improve my app? Classifying user reviews for software	Pengolahan Bahasa Natural, Analisa Teks,	feature request, reviewon asking, problem discovery,	Aplikasi AngryBirds, Dropbox, dan Evernote yang diambil dari	precision dari klasifikasi memperoleh nilai 75% dan recall 74%

<p>maintenance and evolution (Panichella et al. 2015)</p>	<p>dan Analisa sentimen</p>	<p>solution proposal, information seeking, information giving</p>	<p>Apple Store serta TripAdvisor, PicsArt, Pinterest, Whatsapp yang diambil dari Android Google Play</p>	
<p>Fine-grained Reviewon mining with Recurrent Neural Network (RNN) and word embedding (Liu : 2015)</p>	<p>Model RNN dan Word Embedding</p>	<p>Laptop, Restaurant</p>	<p>SemEval-2014 task 4: aspect-based sentiment analysis evaluation campaign</p>	<p>LSTM memperoleh precision dari klasifikasi memperoleh nilai 83.64% dan recall 81.39% pada dataset restaurant dan 82.80% dan 81.22% untuk dataset laptop</p>
<p>Clustering Product Features for reviewon mining (Zhai: 2011)</p>	<p>Similarity Clustering</p>	<p>hometheather (H), Insurance (I), Mattress(M), Car (C), and Vacuum(V)</p>	<p>Reviews from commercial company that provides sentiment analysis services</p>	<p>rerata entropy dari purity memperoleh nilai 2.56% dan recall 0.33%</p>
<p>Semantic Expansion using Word embedding clustering and</p>	<p>Semantic Clustering, Word Embedding, dan</p>	<p>business, computers, culture-arts-entertainment, education-</p>	<p>Google Snippets, TREC</p>	<p>85% untuk dataset Google Snippets dan 95.73% untuk dataset TREC</p>

Convolutional Neural Network for Improving Short Text Classification (Wang : 2015)	Convolutional Neural Network	science, engineering, heart, politics-society, sports		
Automated classification of software change messages by semi supervised LDA (Fu : 2014)	Semi Supervised LDA	Corrective, Adaptive, Perfective (Swanson Software Change Classification)	Change Log dari 5 open source projects, diantaranya adalah Bugzilla, Wireshark, Boost, Firebird, dan Python	Rerata F-measure 65% – 80% dan accuracy 65% - 75%

2. 5. Kesimpulan Hasil Studi Komparasi

Tabel 2.4 menampilkan perbandingan metode berdasarkan analisa klasifikasi. Berdasarkan tabel-tabel tersebut, terdapat beberapa metode dalam memproses data pada penggalian review, penelitian ini akan berfokus dalam mengembangkan beberapa metode, diantaranya adalah *similarity clustering* yang terinspirasi dari paper (Zhai: 2011). Metode ini dapat meningkatkan performa dari Latent Dirichlet Allocation (LDA) dalam menggeneralisasi *hidden topic* menjadi kelas label untuk setiap dokumen. Selain itu, performa dari term list yang telah didefinisikan sebelumnya, akan diluaskan dengan menggunakan TF-ICF. Dokumen yang telah terlabeli selanjutnya dapat digunakan sebagai input pada proses klasifikasi. Adapun klasifikasi akan menggunakan metode LSTM, seperti yang telah dilakukan oleh paper (Liu : 2015). Pada proses ekstraksi fitur yang akan digunakan pada proses klasifikasi, word2vec akan digunakan sebagai proses pembentukan vektor. Hal ini

berdasarkan pada kesuksesan word2vec pada paper (Wang et al. 2016) dibandingkan beberapa tipe word embedding lainnya.

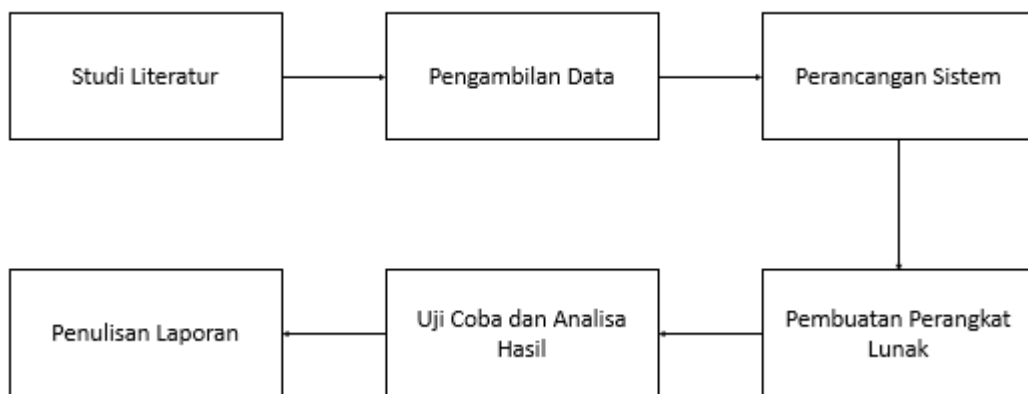
Pada proses pengklasifikasian dokumen review produk, terdapat banyak jenis kategori yang dapat digunakan. Penelitian ini akan menggunakan tipe klasifikasi perubahan proses perangkat lunak yang diusulkan oleh (Maalej 2015) yang memetakan kategori perubahan proses perangkat lunak ke dalam beberapa tipe, yaitu *bug error*, *feature request*, *user experience*, dan *ratings*.

[Halaman ini sengaja dikosongkan]

BAB 3

METODE PENELITIAN

Bab ini akan memaparkan tentang metodologi penelitian yang digunakan pada penelitian ini, yang terdiri dari (1) studi literatur, (2) pengambilan data, (3) perancangan sistem, (4) pembuatan perangkat lunak, (5) uji coba dan analisa hasil, dan (5) penulisan laporan. Ilustrasi alur metodologi penelitian dapat dilihat pada Gambar 3.1



Gambar 3.1 Alur Metodologi Penelitian

Penjelasan tahapan metode penelitian pada Gambar 3.1 akan diterangkan secara terperinci pada sub bab berikut.

3. 1. Studi Literatur

Penelitian diawali dengan proses pengkajian yang berkaitan dengan topik penelitian yang diambil. Pada penelitian ini, referensi yang digunakan diperoleh dari jurnal yang memiliki hubungan dengan klasifikasi perubahan perangkat lunak, penggalian review, metode LDA, Cluster based similarity, Word Embedding dan LSTM. Dari studi literatur yang telah dilakukan, diperoleh informasi yang berhubungan dengan penelitian, diantaranya sebagai berikut:

1. Banyak nya review atau review yang terdapat pada aplikasi bergerak mengandung banyak informasi yang berhubungan dengan kebutuhan pengguna

2. Proses penggalian review membutuhkan metode klasifikasi yang baik agar pengembang dapat memperoleh banyak masukan untuk mengembangkan aplikasi
3. (Maalej 2015) mengklasifikasikan review pengguna kedalam beberapa kategori, meliputi *bug report*, *feature request*, *user experience*, dan *rating*
4. Term Frequency – Inverse Cluster Frequency (TF-ICF) merupakan metode pembobotan kata untuk mengetahui bobot suatu kata pada suatu kluster berdasarkan kemunculan term pada cluster.
5. Latent Dirichlet Allocation (LDA) adalah model probabilistik yang mengelompokkan bagian dari suatu data yang memiliki kemiripan. Biasanya LDA digunakan pada suatu dataset dalam bentuk teks. LDA melihat setiap dokumen memiliki berbagai topik dalam suatu nilai probabilitas tertentu.
6. Pelabelan menggunakan metode kedekatan berdasarkan nilai kemiripan antar kata (*word similarity*) dapat diperoleh melalui banyak cara, salah satunya adalah dengan menggunakan Wordnet.
7. Evaluasi pelabelan menggunakan formula Precision, Recall, dan F-Measure untuk menganalisa dokumen yang berhasil di-*retrieve* oleh sistem
8. Ekstraksi fitur dengan menggunakan metode Word Embedding. Setiap kata dan topik akan dicari nilai fiturnya menggunakan rerata nilai representasi setiap kata dalam bentuk vektor (*Mean Representation Vector*).
9. Long Short Term Memory (LSTM) merupakan metode pengembangan Recurrent Neural Network (RNN) yang mampu mengingat nilai dari suatu input dalam interval tertentu.
10. Terdapat suatu karakteristik dari review aplikasi yang umum ditemukan pada kasus penggalian review, diantaranya adalah:
 - a. Tidak menggunakan struktur tata bahasa yang baku
Teks pada dataset penggalian review tidak dapat dipastikan memiliki struktur bahasa yang baik, seperti tidak mengandung subjek atau predikat secara eksplisit.
 - b. Memungkinkan terdapat kata-kata yang jarang ditemukan secara umum

Tidak adanya tuntutan menggunakan bahasa formal membuat pengguna bebas menuliskan review nya, salah satunya adalah menggunakan kata-kata baru (*slang words*) yang mampu memicu *Out of Vocabulary* (OOV)

- c. Penggunaan emoji/emoticon pada dokumen review.

Pemberian review pada aplikasi di App Store mendukung *encoding character* UTF-8. Penggunaan *encoding character* tersebut mengakibatkan pengguna Twitter dapat menggunakan blok karakter emoji atau blok karakter ornamen di *character set* UTF-8. Hal ini mengakibatkan pengguna dapat lebih bebas dalam mengungkapkan ekspresi dan juga menjadi tantangan tambahan dalam proses ekstraksi informasi.

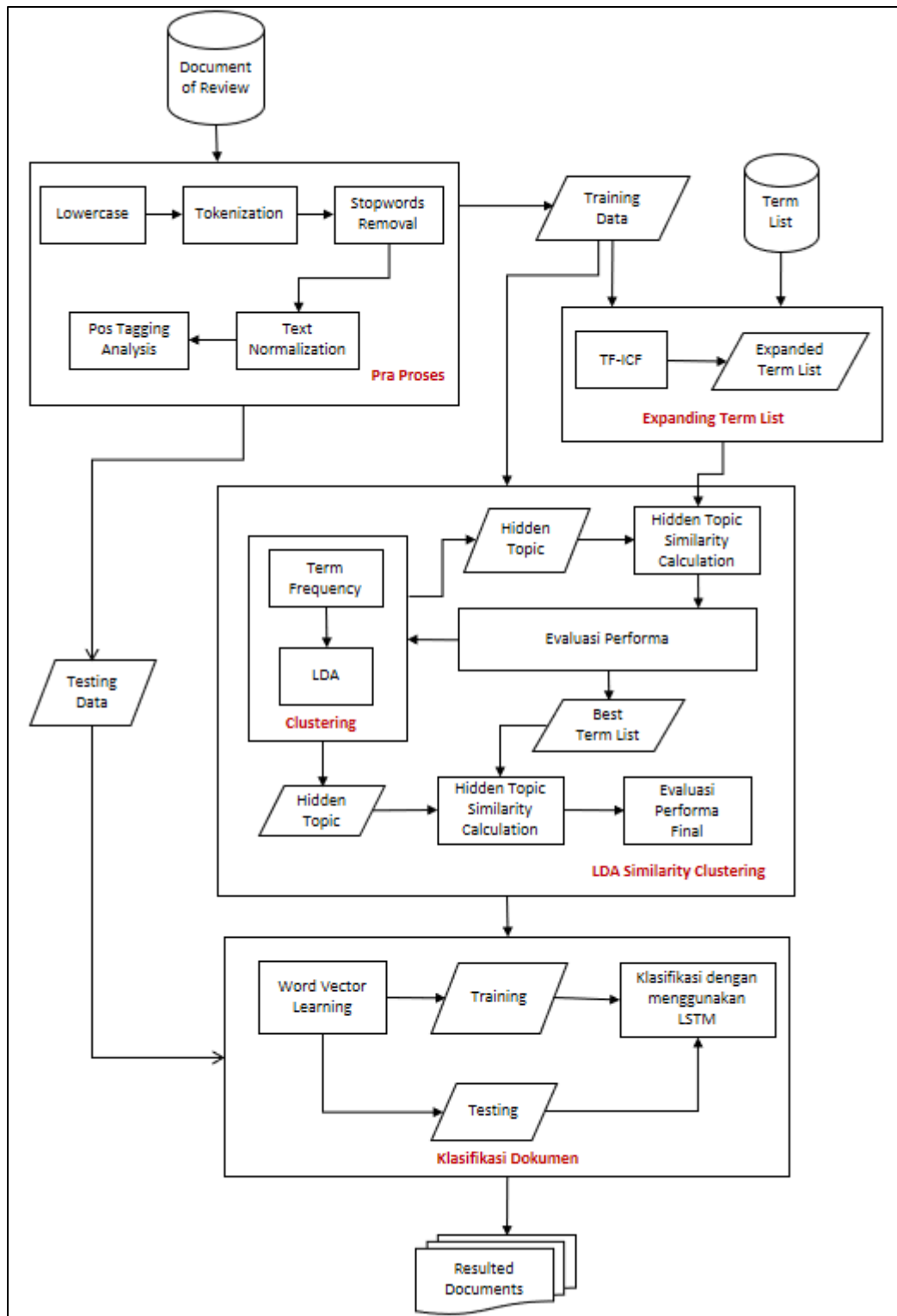
- 11. Metode lain yang dapat digunakan untuk menganalisa data review adalah dengan memanfaatkan aturan bahasa atau dengan memanfaatkan relasi antar kata dalam kalimat. Dari studi literatur, dapat disimpulkan juga mengenai kondisi saat ini bahwa:

- a. Sebagai salah satu sumber masukan bagi pengembang perangkat lunak, penggalian review dapat digunakan untuk menggali berbagai tipe kebutuhan dari pengguna atas suatu aplikasi yang digunakan
- b. Dalam melakukan proses penemuan aspek serta sentimen dari suatu review, dibutuhkan suatu proses yang dapat menganalisa jika terdapat review yang mengandung lebih dari satu aspek atau sentimen.

3. 2. Pengambilan Data

Data review yang diproses pada penelitian ini adalah review dari aplikasi Waze dan Accuweather yang tersedia di Google App Store. Penelitian ini hanya menganalisa review pengguna yang memiliki rating satu dan dua. Hal tersebut berhubungan dengan asumsi bahwa rating-rating tersebut dapat menggambarkan kategori-kategori yang akan dianalisa oleh sistem. Setiap pengguna memiliki kebebasan untuk mengekspresikan pendapatnya mengenai suatu produk, sehingga hal tersebut memungkinkan adanya lebih dari satu kategori pada suatu kalimat. Sehingga penelitian ini memfokuskan untuk menganalisa 2000 kalimat dari 1318 review.

3.3. Perancangan Sistem



Gambar 3.2 Alur Usulan Metode Penelitian

Penelitian ini memiliki empat tahapan besar yang akan diimplementasikan. Tahapan-tahapan besar tersebut selanjutnya akan mengeksekusi modul-modul kecil dengan tugas dan fungsinya masing-masing. Adapun rancangan sistem penelitian dideskripsikan pada Gambar 3.2. Secara umum, model rekomendasi dibagi menjadi empat modul besar, yaitu: pra-proses, *expanding term list*, proses pelabelan, dan klasifikasi dokumen sesuai yang dideskripsikan pada Gambar 3.2.

Data yang diperoleh pertama-tama akan diolah agar dapat diproses dalam model-model selanjutnya melalui modul pertama, modul pra proses. Modul pertama bertugas untuk membersihkan review yang akan digunakan pada modul selanjutnya baik dari noise ataupun struktur teks yang kurang relevan untuk diproses. Dengan demikian, diharapkan modul yang dihasilkan pada tahap selanjutnya mampu bekerja dengan baik.

Modul selanjutnya adalah modul *expanding term list*. Model ini memiliki tujuan untuk menanggulangi terjadinya *Out-of-Vocabulary*, yaitu kondisi dimana pendeklarasian *term list* pada setiap kategori terbatas sehingga memungkinkan adanya term kategori yang tidak dideskripsikan. Dengan demikian, diharapkan modul ini dapat mengoptimalkan proses pada modul proses pelabelan terutama pada saat perhitungan *hidden topic similarity*. Setelah itu, data hasil pra proses dan *expanding term list* masuk ke dalam modul proses labeling. Modul ini memiliki dua sub modul besar, yaitu *clustering* dan *hidden topic similarity*.

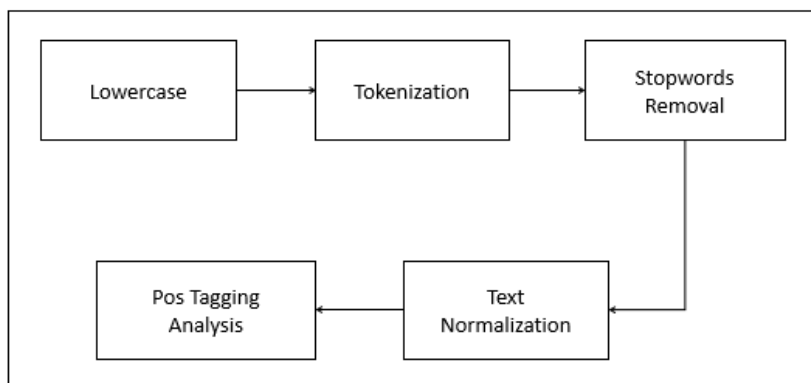
Pada modul ini, setiap dari dokumen dilabeli berdasarkan kategori perubahan proses perangkat lunak. Sehingga label kelas yang akan digunakan pada proses pelabelan dan klasifikasi adalah atribut kategori yaitu *bug report*, *feature request*, dan *non informative*.

Ketiga jenis kelas dalam hal ini kategori tersebut berbeda dengan kategori yang telah dituliskan pada studi literatur. Hal tersebut disebabkan peneliti hendak menggabungkan kelas *user experience* dan *rating* menjadi satu jenis kelas karena kedua jenis kategori ini mengandung pernyataan yang tidak jauh berbeda dalam merepresentasikan setiap kelasnya. Setelah mendapatkan evaluasi pada tahap performa, penulis selanjutnya memilih term list terbaik untuk setiap kelas. Hal tersebut dilakukan dengan asumsi, dengan mengambil term list terbaik setiap kelas, diharapkan akan diperoleh hasil performa evaluasi akhir yang baik pula.

Setelah diperoleh dan dievaluasi, setiap dari data tersebut diproses pada modul klasifikasi untuk memperoleh hasil analisa klasifikasi untuk masing-masing dokumen. Jika hasil dari klasifikasi mampu menunjukkan hal yang baik, maka sistem ini mampu dijadikan sebagai rekomendasi kepada pengembang perangkat lunak dalam melakukan perbaikan atau pun evolusi perangkat lunak.

3.3.1. Pra Proses

Pada modul ini dilakukan proses untuk mempersiapkan dan memastikan dokumen yang diproses telah siap untuk digunakan pada modul selanjutnya. Arsitektur yang dipilih merupakan arsitektur pra proses umum untuk mengekstrak suatu informasi. Pada model ini, modul pra proses memiliki beberapa proses. Berikut ini adalah penjelasan mengenai fungsi dan contoh pemrosesan review yang dilakukan pada setiap proses pada modul pra proses.



Gambar 3.3 Pra proses sebagai input pada modul topic learning

Adapun penjelasan dari setiap proses di Gambar 3.4 diantaranya adalah:

1. Lowercase

Proses ini memiliki fungsi untuk mengubah huruf pada teks review ke dalam bentuk huruf kecil. Pada teks review

'This App Runs So Smoothly And I Rarely Have Issues With It Any more'

Akan berubah menjadi

'this app runs so smoothly and i rarely have issues with it any more'

2. Tokenization

Tokenisasi memiliki peran untuk memecah kalimat, paragraf, dokumen menjadi unit terkecil dalam pemrosesan bahasa atau dikenal dengan token atau potongan kata. Pada penelitian ini, tokenization dilakukan dengan menggunakan fungsi `word_tokenize` pada library NLTK. Pada tahap tokenisasi, review

```
'this app runs so smoothly and i rarely have issues  
with it any more'
```

akan dipecah menjadi

```
{this, app, runs, so, smoothly, and, i, rarely, have,  
issues, with, it, any, more' }
```

3. Stopwords Removal

Stopwords Removal memiliki peran untuk menghapus kata henti dalam bahasa Inggris. Adapun stopwords yang akan dihilangkan pada tahap ini adalah:

Tabel 3.0.1 Tabel Kata Henti

for, a, of, the, and, to, in, as, are, is, at, on, be, i, you, they, we, she, he, it, if, up, also, your, their, its, her, his, that, this, those, these, there here an or now why who what was were am can not so when which me
--

Pada teks review

```
'this app runs so smoothly and i rarely have issues  
with it any more'
```

Akan berubah menjadi

```
{app, runs, smoothly, rarely, issues}
```

4. Text Normalization

Text normalization bertujuan untuk membuang dan merubah beberapa kata menjadi kata yang seharusnya dan bukan merupakan singkatan. Selain itu, sub modul ini pun akan membuang angka, tanda-tanda baca seperti '.', '!', '?', '...', dll.

Tabel 3.0.2 List of Word Normalization

Base Form	Normalized Form
n't	not

didnt	did not
wo	will
'd	would
ive	i have
re	are
cant	can not
's	is
wouldnt	would not
ca	can
havent	have not
&	and
'm	am
dont	do not
alot	lot
hasnt	has not
've	have

5. Part of Speech Tagging

Part of speech dalam bahasa Indonesia disebut sebagai Kelas Kata. Kelas kata adalah golongan kata dalam satuan bahasa berdasarkan kategori bentuk, fungsi dan makna dalam sistem tata bahasa. Penelitian ini menggunakan POS Tag yang disediakan oleh NLTK. Penelitian ini tidak menggunakan proses stemmer atau pun lemmatizer secara independen. Hal tersebut disebabkan oleh penggunaan stemmer berjenis Snowball, Porter, maupun Lancaster belum dapat menghasilkan keluaran yang maksimal. Hasil stemming terkadang memotong kata sehingga kata yang dihasilkan tidak dapat diproses pada modul selanjutnya. Sedangkan Lemmatizer mampu bekerja dengan baik namun memiliki kekurangan dalam mengatasi adjective superlative dan comparative, plural noun, serta adverb of manner. Oleh karena itu, pada penelitian ini, peneliti menambahkan fungsi POS Tagging Rules untuk mengatasi kekurangan tersebut.

Tabel 3.0.3 List of Word Normalization

Tag	Action
Adjective Superlative (JJR) Adjective Superlative (JJS)	do adjective lemmatization by using WordNetLemmatizer from WordNet
Noun Plural (NNS)	make it be singular noun by using inflect library
Adverb of manner (RB)	remove the 'ly' in the end of text
Else	Do verb lemmatization by using WordNetLemmatizer from WordNet

Selanjutnya hasil preprocessing digunakan sebagai input pada modul expanding term list dan short text classification. Gambar 3.2 menunjukkan bahwa pra proses digunakan sebagai input pada sub model proses topic modeling dan modul expanding term list.

3.3.2. Modul Perluasan Term List (Expanding Term List)

Terdapat tiga jenis kelas yang digunakan pada penelitian ini, yaitu: *bug report*, *feature request*, dan *non informative (user experience serta rating)*. Hal tersebut disebabkan karena pada data yang diperoleh, term yang dimiliki oleh user experience dan rating sering kali beririsan. Hal tersebut menyebabkan pengekspresian ke dua kelas tersebut sulit dipisahkan.

Modul ini merupakan modul yang ditujukan untuk mengatasi permasalahan *Out-of-Vocabulary* yang mungkin timbul pada saat pendeklarasian term list. Berikut adalah deskripsi untuk masing-masing kelas:

1. *Bug reports* mendeskripsikan permasalahan dari aplikasi yang akan dikoreksi, seperti kegagalan fungsional atau permasalahan performa, seperti yang diungkapkan pada review berikut:

```
'slow when starting and why does it need the permissions to read phone calls'
```
2. *Feature requests* mendeskripsikan permintaan pengguna untuk mengatasi fungsionalitas yang bermasalah, konten yang tidak ada, atau berbagi ide mengenai saran yang dapat meningkatkan performa aplikasi untuk update selanjutnya, seperti yang diungkapkan pada review berikut:

'it should have an option to hide video and news category which are not of much use'

3. *Non informative* mengombinasikan pengalaman pengguna dan rating yang menjelaskan mengenai ekspresi pengguna ketika menggunakan aplikasi, seperti yang diungkapkan pada review berikut:

'starting to regret purchasing'

Oleh karena itu, penelitian ini memperluas term list dengan menggunakan *Term Frequency-Inverse Document Frequency* (TF- ICF). Metode ini digunakan untuk mengambil term-term penting setiap kelas. Adapun formula dari TF-ICF dijelaskan pada Equation 1 dan Equation 2.

$$TF - ICF = TF_{t,i} \times ICF_t \quad (3.1)$$

$$TF - ICF = TF_{t,i} \times \log\left(\frac{N}{CF_t}\right) \quad (3.2)$$

$TF_{t,i}$ = jumlah term pada kelas i

N = jumlah kelas

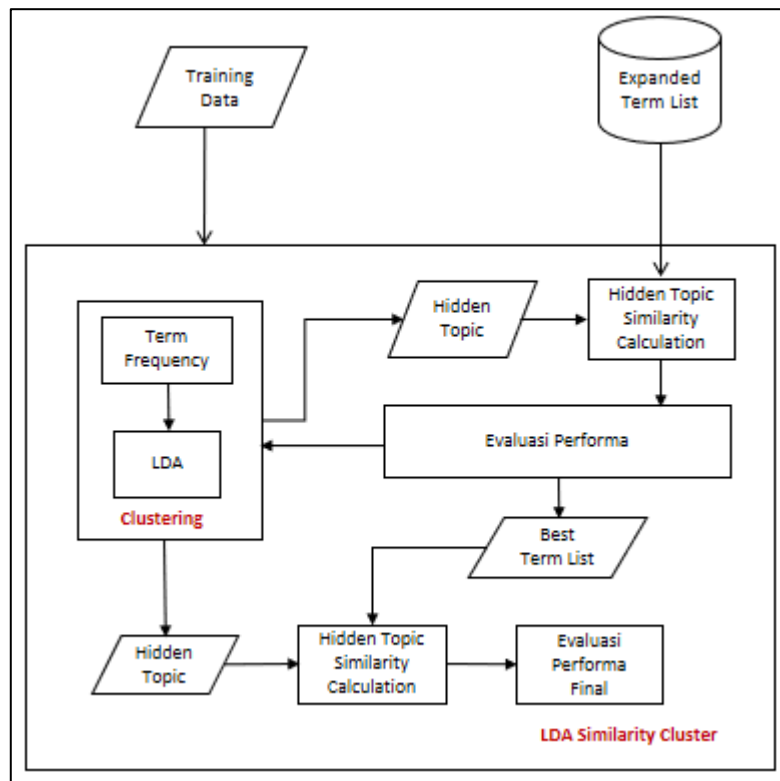
CF_t = jumlah kelas yang mengandung term t

Terdapat langkah-langkah dari TF-ICF dalam menemukan term list tambahan. Hal tersebut dijelaskan pada pseudocode berikut:

1. Ambil review yang telah di pra proses pada modul sebelumnya
2. Berdasarkan kelasnya masing-masing, hitung nilai dari Term Frequency (TF)
3. Hitung Inverse Cluster Frequency (ICF)
4. Hitung nilai TF-ICF
5. Untuk masing-masing kelas, analisa term-term penting sesuai nilai TF-ICF dan justifikasi yang telah didefinisikan sebelumnya.

Aturan tersebut berdasarkan pada analisa bahwa ekspresi setiap kelas akan berbeda-beda. Meskipun begitu, setiap kelas memiliki kecenderungan yang sama untuk mengekspresikan permasalahan, kebutuhan, atau kesan dari aplikasi yang dikembangkan. Oleh karena itu penggunaan penambahan term list berdasarkan kelas yang telah ditandai sebelumnya diharapkan mampu meningkatkan performa.

3.3.3. Proses Pelabelan



Gambar 3.4 Topic Learning untuk Pelabelan Dokumen

Setelah melewati modul pra proses dan expanding term list, selanjutnya proses pelabelan dari setiap dokumen diinisialisasi. Tujuan utama dari modul ini adalah memberikan label pada setiap dokumen sesuai dengan kelas *review*-nya. Input pada proses ini adalah hasil dari modul pra proses. Sedangkan secara umum, output yang dihasilkan dari modul ini adalah dokumen yang telah terlabeli. Adapun diagram alir pada modul ini direpresentasikan pada Gambar 3.5.

Secara umum modul topic learning terdiri dari tiga sub modul besar, yaitu Clustering dengan menggunakan LDA, *Hidden Topic Similarity*, dan Evaluasi Performa. Berikut adalah penjelasan untuk masing-masing sub modul.

1. Topic Modelling (LDA)

Sub modul Topic Modelling dengan menggunakan LDA ini berfungsi sebagai metode untuk menemukan hidden topic yang berada pada suatu dokumen review. Sehingga, dengan menggunakan metode ini setiap dokumen review dapat direpresentasikan menjadi beberapa topik. Input dari proses ini adalah hasil dari pra

proses. Sedangkan output dari sub modul ini adalah hidden topic untuk masing-masing dokumen. Secara umum terdapat beberapa langkah untuk memodelkan hidden topic dari suatu dokumen, yaitu:

1. Input yang akan dianalisa adalah kata-kata dari dokumen hasil pra proses
2. Tentukan nilai topics untuk setiap kata pada dokumen secara random
3. Hingga iterasi maksimal, hitung nilai probabilitas topik untuk setiap kata
4. Hitung probabilitas kata untuk setiap topik dan probabilitas dokumen untuk setiap topik
5. Cari nilai probabilitas dokumen pada setiap topik tertinggi untuk setiap dokumen

Pada implementasinya, LDA menggunakan term frekuensi untuk membangun vektor dari setiap token kata dengan menggunakan modul CountVectorizer dan selanjutnya dianalisa dengan menggunakan LDA model yang dikonstruksi dengan menggunakan Librari Gensim. Adapun output dari proses ini adalah hidden topics yang ditunjukkan pada Tabel 3.4

Tabel 3.0.4 Tabel Contoh Input dan Output LDA

app run smooth rare have issue any more	(4, '0.200*"issue" + 0.200*"smooth" + 0.200*"run"')
--	---

2. Hidden Topic Similarity Clustering

Sub modul topic clustering berfungsi untuk mengelompokkan setiap dokumen ke dalam salah satu dari tiga kelas berdasarkan nilai similaritas dari hidden topic setiap dokumen dengan term list yang ada. Metode ini mempertimbangkan kemiripan setiap *hidden topic* dengan jenis kelas yang telah ditentukan sebelumnya. Input dari sub modul ini adalah *hidden topic* yang diperoleh dari sub modul clustering LDA. Adapun output dari proses ini adalah label dokumen. Kakas bantu yang digunakan untuk mengelompokkan setiap dokumen adalah WordNet similarity. Nilai yang dihasilkan berada pada rentang [-1 1], dimana 1 menyatakan kedua term benar-benar sama dan -1 yang menyatakan benar-benar beda atau tipe kata yang tidak kompatibel. Secara umum terdapat beberapa langkah untuk mengelompokkan

suatu dokumen terhadap salah satu dari kelas yang ada, proses terdiri dari empat langkah, yaitu:

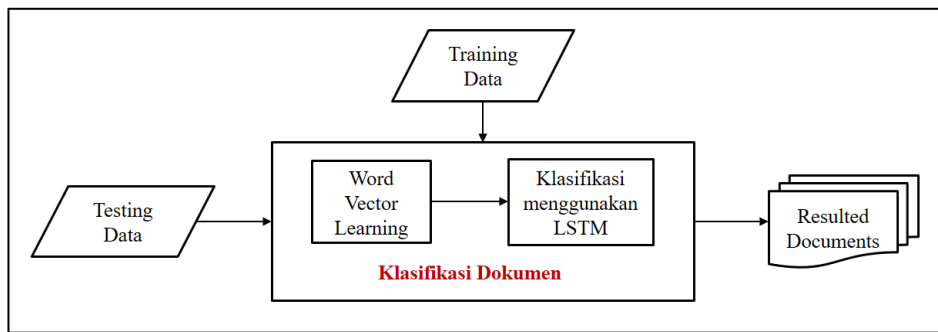
1. Untuk setiap dokumen i^{th} , hitung nilai similarity dengan menggunakan wordnet terhadap term list masing-masing kategori seperti yang terlihat pada Tabel 3.4
2. Untuk setiap dokumen i^{th} , nilai similarity dihitung dengan menggunakan rumus similarity
3. Nilai similarity yang diperoleh kemudian dicari reratanya sesuai dengan kelas klaster masing-masing
4. Hasil rerata setiap kelas selanjutnya diranking. Kelas yang memiliki nilai similarity tertinggi selanjutnya akan menjadi label untuk dokumen tersebut, seperti yang terlihat pada Tabel 3.5

3. Clustering Evaluation

Sub modul clustering evaluation berfungsi untuk memeriksa apakah kelas yang telah dihasilkan berisi dokumen yang homogen. Skenario mula-mula adalah mencoba mengevaluasi hasil kelas menggunakan *silhouette index*. Namun, nilai silhouette yang dihasilkan tidak optimal sehingga tidak dapat dijadikan sebagai acuan. Hal tersebut disebabkan oleh kemungkinan penyebaran term setiap kelas yang tidak terlalu jauh berbeda. Oleh karena itu, perhitungan performa dilakukan dengan mencocokkan hasil dan *ground truth* yang diperoleh dari annotator. Sehingga proses evaluasi dihitung dengan menggunakan precision, recall, dan f-measure. Adapun hasil dari proses pelabelan yang selanjutnya digunakan pada tahap klasifikasi dokumen review adalah yang bernilai True Positive.

3.3.4. Klasifikasi Dokumen Review

Setelah melewati modul pelabelan dokumen, data diproses untuk mengetahui performa dari proses pada modul klasifikasi. Tujuan utama dari modul ini adalah mengklasifikasi setiap dokumen review ke dalam jenis kategori perubahan perangkat lunak. Input pada proses ini adalah dokumen dari hasil *topic learning*. Sedangkan, output yang dihasilkan dari modul ini adalah hasil performa klasifikasi.



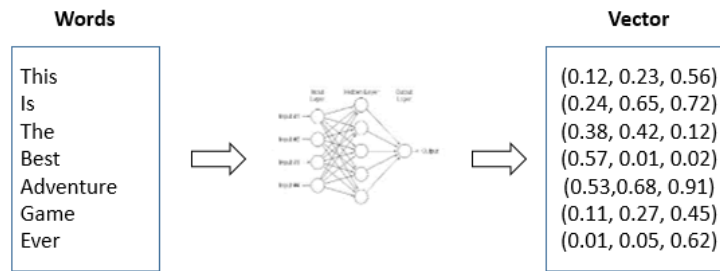
Gambar 3.5 Klasifikasi dokumen review

Seperti yang ditunjukkan pada Gambar 3.5, input yang digunakan pada modul ini adalah *Labeled Cluster* yang berisi dokumen dan label nya. Adapun diagram alir pada modul ini direpresentasikan pada Gambar 3.5. Berikut adalah penjelasan untuk masing-masing sub modul.

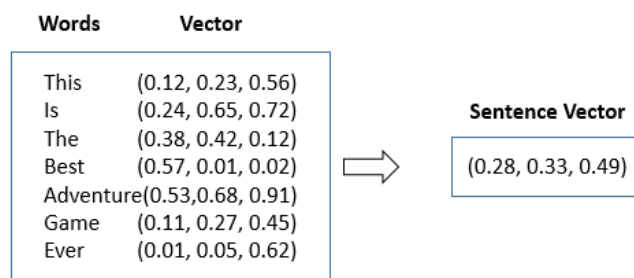
1. Word + Topic Vector Learning

Sub modul ini berfungsi untuk menghasilkan nilai fitur yang digunakan pada proses klasifikasi. Pada modul ini setiap dokumen beserta label diproses menjadi word dan topic vector sebagai nilai fitur. Adapun nilai fitur diperoleh dari rerata nilai vektor dari representasi word embedding untuk setiap kata. Input dari proses ini adalah hasil dari modul topic learning yaitu topik, dokumen, dan label. Adapun word embedding yang digunakan diinisialisasi dengan menggunakan GloVe. Tensorflow digunakan untuk mengimplementasikan model Neural Network yang digunakan. Secara umum terdapat beberapa langkah untuk memperoleh fitur vector, proses terdiri dari tiga langkah, yaitu:

1. Untuk setiap dokumen i^{th} akan dilatih dengan menggunakan model neural network untuk memaksimalkan kemungkinan kondisional dari konteks kata yang diberikan
2. Terapkan model ke setiap kata untuk memperoleh corresponding vector



3. Hitung nilai vektor dari suatu kalimat dengan merata-ratakan vektor dari setiap kata



Pada proses implementasi, word2vec dijalankan dengan menggunakan bantuan library tensorflow yang merupakan library open source untuk menyelesaikan permasalahan di sektor kecerdasan buatan dan Keras yang merupakan library model dari neural network.

2. Classification using LSTM

LSTM memproses hasil dari word embedding. Embedding vector dari aspek yang telah ada dipelajari selama proses training. Hal tersebut diharapkan dapat memetakan informasi-informasi penting dari suatu kalimat. Output dari proses ini adalah evaluasi klasifikasi.

1	Epoch 1/3
2	16750/16750 [=====] - 58s - loss: 0.5186 - acc: 0.7263
3	Epoch 2/3
4	16750/16750 [=====] - 58s - loss: 0.2946 - acc: 0.8825
5	Epoch 3/3
6	16750/16750 [=====] - 58s - loss: 0.2291 - acc: 0.9126
7	Accuracy: 86.36%

Gambar 3.6 Contoh Hasil Klasifikasi LSTM

Proses perhitungan performa dari klasifikasi yang terlihat pada Gambar 3.6 dilakukan dengan menggunakan tensorflow dan keras seperti yang dilakukan pada proses word embedding. Accuracy dijadikan sebagai formula yang biasa digunakan

untuk mengevaluasi performa klasifikasi dari metode deep learning dengan menggunakan teknik binary classification.

3. 4. Perancangan Pengujian

Pada proses pengujian, penelitian ini melakukan beberapa skenario pada beberapa modul untuk mengetahui performa dari sistem yang dihasilkan. Adapun skenario tersebut adalah:

1. Pengujian ukuran data dan pendeteksian satu dan multi label
 - a. 474 data untuk pendeteksian satu label (diambil dari 1200 data)
 - b. 773 data untuk pendeteksian satu label (diambil dari 2000 data)
 - c. 1522 data untuk pendeteksian multi label (diambil dari 2000 data)

2. Penerapan batas ambang dari nilai TF-ICF per kategori

Setiap kategori memiliki skala batas ambang yang berbeda, peneliti hendak mengetahui dampak dari penerapan batas ambang yang berbeda terhadap performa pelabelan dokumen dengan skenario berikut:

- a. Batas ambang 15% dari nilai maksimal TF-ICF per kategori
- b. Batas ambang 30% dari nilai maksimal TF-ICF per kategori
- c. Batas ambang 45% dari nilai maksimal TF-ICF per kategori
- d. Batas ambang 60% dari nilai maksimal TF-ICF per kategori
- e. Batas ambang 75% dari nilai maksimal TF-ICF per kategori

3. Penerapan Similarity Distance untuk perhitungan Hidden Topic Similarity

Terdapat banyak jenis distance similarity yang diberikan oleh WordNet. Pada penelitian kali ini, peneliti hendak menguji coba performa beberapa jenis distance similarity terhadap performa pelabelan dokumen dengan skenario berikut:

- a. Pengukuran similarity distance menggunakan Wu Palmer Similarity
- b. Pengukuran similarity distance menggunakan Path Similarity

4. Penerapan metode Classifier dalam mengklasifikasi dokumen review

Setelah dilabeli, dokumen-dokumen review yang memiliki kategori short text document diklasifikasi dengan menggunakan beberapa metode untuk mengetahui performa dari setiap metode classifier dengan skenario berikut:

- a. Tanpa Pre Trained Embedding + LSTM

- b. Tanpa Pre Trained Embedding + CNN
- c. GloVe + LSTM
- d. GloVe + CNN

[Halaman ini sengaja dikosongkan]

BAB 4

HASIL DAN PEMBAHASAN

Pada pembahasan ini diberikan pemaparan mengenai implementasi sistem serta pengujian dari sistem berdasarkan skenario yang telah dirancang pada Bab Tiga. Proses implementasi dilakukan berdasarkan tahapan yang telah diberikan pada pembahasan sebelumnya. Selanjutnya pengujian sistem dilakukan dengan beberapa kondisi yang disesuaikan dengan skenario pengujian. Dari hasil pengujian yang telah didapatkan, selanjutnya diberikan pembahasan dan analisa dari setiap pengujian yang dilakukan dengan tujuan untuk mendapatkan hasil dari penelitian, sehingga mendapatkan kesimpulan yang diberikan pada pembahasan selanjutnya.

4.1. Implementasi Sistem

4.1.1 Deskripsi Data Uji

Dataset yang digunakan pada pengujian proses pelabelan adalah *review* yang sudah dilabeli. Adapun data yang digunakan *review* dari aplikasi Waze dan Accuweather diperoleh dari Google Play Store. Dokumen yang dianalisa berbahasa Inggris dan memiliki rating satu dan dua. Hal tersebut disebabkan oleh kemungkinan dokumen kelas *bug report*, *feature request*, dan *non informative* yang memiliki sentimen negatif dan berating rendah.

Adapun pada implementasinya, setiap *review* dipecah per kalimat. Hal tersebut berdasarkan pertimbangan bahwa suatu *review* mengandung lebih dari satu kelas. Selain memiliki sentimen yang cenderung sama, setiap kelas memiliki penyebaran term yang relatif sama. Hal tersebut menyebabkan nilai silhouette setiap kelas yang dihasilkan tidak terlalu baik serta perbedaan antar kluster sangat tipis. Sehingga untuk mengevaluasi performa dari metode yang dikembangkan, maka penelitian ini dibantu oleh sejumlah annotator untuk melabeli dokumen yang diproses. Adapun proses anotasi yang dilakukan adalah dengan melakukan pelabelan satu dokumen oleh tiga annotator. Pada pengolahan data training, penulis menguji coba performa model pada input data sebanyak 1200 dan 2000. Hal tersebut disebabkan tidak hanya oleh proses pelabelan oleh annotator yang

dilakukan bertahap, tetapi juga untuk mengamati perubahan performa metode yang diusulkan jika besar data berubah.

1. 1200 Data

Adapun komposisi kelas dari 474 data yang dilabeli oleh tiga annotator, terdiri dari 359 data kelas Bug Report, 20 data kelas Feature Request, dan 95 data kelas Non Informative. Sejumlah 474 data selanjutnya akan digunakan untuk mengklasifikasi dokumen ke dalam one class.

2. 2000 Data

Adapun komposisi kelas dari dokumen ini dijelaskan pada Tabel 4.2

Tabel 4.1 Komposisi Kelas dari 2000 Data

	Bug Report	Feature Request	Non Informative	Total
3 Annotator Sepakat	587	31	155	773
2 Annotator Sepakat [Training]	878	76	568	1522
2 Annotator Sepakat [Testing]	322	56	229	607

Dengan menggunakan data ini, penulis hendak membandingkan apakah metode yang diusulkan mampu digunakan untuk mengklasifikasikan one class atau multi-class. Sehingga, 773 dokumen akan digunakan sebagai uji coba klasifikasi one class. Sedangkan 1903 data (gabungan data training dan testing) akan digunakan dalam klasifikasi multi-class.

Dataset dokumen teks dibedakan menjadi dua jenis dataset yang berbeda sesuai dengan proses yang telah dijelaskan pada pembahasan sebelumnya, yaitu dataset pada proses pelabelan dan dataset pada proses klasifikasi.

Dalam proses pelabelan, dokumen dicocokkan dengan hasil anotasi oleh annotator yang bertugas untuk membuat *ground truth* setiap dokumen. Proses pertama yang dilakukan pada proses pelabelan adalah melakukan pra proses dari dataset dokumen yang ada. Setelah melewati tahap pra proses, dataset yang diperoleh digunakan sebagai input proses expanding term list. Setelah menambah beberapa term list, modul proses pelabelan selanjutnya dimulai dengan menjalankan sub modul LDA cluster. Hasil hidden topic yang diperoleh pada LDA

cluster selanjutnya dihitung nilai similarity nya dengan hasil expanding term list. Nilai similarity terbesar selanjutnya dijadikan label untuk dokumen tersebut. Setelah memperoleh prediksi label, selanjutnya metode evaluasi digunakan untuk menghitung performa dari proses pelabelan. Selain itu, dokumen yang relevan digunakan sebagai input pada proses selanjutnya. Dokumen yang terdeteksi benar selanjutnya akan digunakan pada tahap klasifikasi. Metode klasifikasi yang digunakan pada penelitian kali ini adalah metode Long Short Term Memory yang pertama-tama mengekstrasi fitur dari input dengan menggunakan Word Embedding.

	rating	timestamp	app version	device	user name	title	review text
1	1	1389244712013	1.22.1	?	Joe Palumbo	Galaxy 2 to LG G2	was okay but n
3	5	1387253953766	1.21.0	?	deepak katyal	?	good
4	5	1394861720660	1.26.0	?	Yurel Chattam	Fun Its	amazing
5	4	1388459124656	1.15.1	HTC	Desire HD	A Google User	? When u need help
6	5	1385019389457	1.19.0	?	Meet Marchi	Lovely	Nice
7	4	1396153166214	1.27.0	Samsung	Galaxy S3	Ronnie McIntosh Sr	Addictive
8	5	1390098817305	1.22.1	?	Francesca Hernandez	I could play this	everyday
9	5	1395896627858	1.27.0	?	Yusuf Mamedov	Cool	Nice
10	3	1393047902346	1.25.0	?	Zoe Marie	Last update sucked!	Not very happy!

Gambar 4.1 Contoh data hasil *crawling*

Gambar 4.1 menunjukkan data yang pertama kali diproses, dimana data ini adalah data review dari suatu aplikasi yang terdiri dari banyak review terhadap aplikasi atau selanjutnya disebut dengan dokumen. Tidak semua bagian dari data akan diproses. Karena tidak adanya metode pembobotan, maka bagian yang diambil untuk diproses hanya bagian review text. Setiap dokumen review text akan disimpan ke dalam database dengan identifier ID dan id_kategori yang merupakan representasi dari nama dataset yang sedang dianalisa.

4.1.2 Expanding Term List

Penelitian kali ini menggunakan strategi perluasan term (*expanding term list*) untuk mengatasi keterbatasan dari term yang telah didefinisikan sebelumnya. Proses ini dilakukan terhadap data yang telah melewati tahap pra proses. Adapun metode yang digunakan adalah dengan menggunakan pembobotan Term Frequency – Inverse Cluster Frequency (TF-ICF) untuk menemukan term-term yang merepresentasikan setiap dokumen.

Tabel 4.0.2 Term List dari Setiap Kategori oleh (Maalej 2015)

Category	Term List
Bug Report	'bug', 'fix', 'problem', 'issue', 'defect', 'crash', 'solve'
Feature Request	'add', 'please', 'could', 'would', 'hope', 'improve', 'miss', 'need', 'prefer', 'request's', 'should', 'suggest', 'want', 'wish', 'allow', 'complaint', 'improvement', 'instead', 'lacks', 'maybe', 'must', 'want', 'will'
Non Informative	'help', 'support', 'assist', 'when', 'situation', 'great', 'good', 'nice', 'very', 'cool', 'love', 'hate', 'bad', 'worst'

Sebelum diperluas, terdapat term list yang sebelumnya telah didefinisikan oleh (Maalej 2015) sebagaimana yang ditunjukkan pada Tabel 4.4. Pada penelitian kali ini, peneliti telah menguji coba performa penambahan term list dengan melakukan WordNet similarity berdasarkan synonym set (synset) yang dihasilkan setiap term pada tiap kategori. Namun, tidak semua term memberikan hasil synset yang sesuai. Oleh karena itu, berdasarkan threshold dan justifikasi peneliti, maka hanya beberapa sinonim dari tiap term yang diambil, seperti yang ditunjukkan pada Tabel 4.5

Tabel 4.0.3 Hasil Perluasan Term List dari Setiap Kategori menggunakan Wordnet

Category	Term List
Bug Report	'bug', 'fix', 'problem', 'issue', 'defect', 'crash', 'solve', 'repair', 'problem', 'error', 'resolve', 'clear'
Feature Request	'add', 'please', 'could', 'would', 'hope', 'improve', 'miss', 'need', 'prefer', 'request', 'should', 'suggest', 'want', 'wish', 'allow', 'complaint', 'improvement', 'instead', 'lack', 'maybe', 'must', 'want', 'will', 'give', 'lend', 'bring', 'better', 'choose', 'propose', 'should'
Non Informative	'help', 'support', 'assist', 'when', 'situation', 'aid', 'serve', 'great', 'good', 'nice', 'very', 'cool', 'love', 'hate', 'bad', 'worst'

Setelah melakukan ekspansi dengan menggunakan WordNet similarity, selanjutnya peneliti hendak menambahkan term list dengan menggunakan TF-ICF.

Perhitungan Term Frequency (TF)

Proses yang pertama kali dijalankan dalam proses ini adalah membentuk kamus kata yang berisi kata-kata unik pada corpus dokumen. Proses pembentukan kamus ini, tidak memperhatikan kata henti. Kata-kata unik ini selanjutnya dihitung kemunculannya tiap dokumen pada klaster yang sama. Oleh karena itu, masing-masing kelas memungkinkan memiliki nilai term frekuensi yang berbeda. Adapun contoh hasil dari tahap ini dicontohkan pada Tabel 4.6

Tabel 4.0.4 Contoh Perhitungan Term Frequency

id_term	term	tf1	tf2	tf3
2	rain	8	0	0
5	app	45	2	20
6	say	15	0	0
10	start	8	0	2
11	do	42	0	5
249	would	1	3	0
611	should	1	4	0

Berdasarkan Tabel 4.6 tf1 menunjukkan jumlah kemunculan term pada cluster 1, tf2 menunjukkan jumlah kemunculan term pada cluster 2, dan tf3 menunjukkan jumlah kemunculan term pada cluster 3.

Perhitungan Inverse Cluster Frequency (ICF)

Proses ini akan menghitung nilai dari ICF yang mempertimbangkan kemunculan dari setiap term pada cluster berdasarkan fungsi logaritmik. Adapun contoh dari perhitungan ICF dijelaskan pada Tabel 4.7

Tabel 4.0.5 Contoh Perhitungan Inverse Cluster Frequency (ICF)

id_term	term	tf1	tf2	tf3	icf
2	rain	8	0	0	1,099
5	app	45	2	20	0
6	say	15	0	0	1,099
10	start	8	0	2	0,405
11	do	42	0	5	0,405
249	would	1	3	0	0,405
611	should	1	4	0	0,405

Tabel 4.7 menunjukkan data yang memiliki nilai ICF 0. Hal tersebut disebabkan karena term tersebut muncul di semua klaster. Sehingga ketika fungsi logaritmik menghitung hasil pembagian jumlah klaster dengan klaster kemunculan term, maka akan menghasilkan nilai $\log(1)$, yaitu nol. ICF menunjukkan bahwa semakin sedikit klaster dimana term muncul, maka nilai ICF akan cenderung besar, seperti pada id_term 2.

Penentuan Term-Term Penting Tiap Klaster

Proses pada tahap ini dimulai dengan menghitung nilai TF-ICF untuk setiap klaster. Selanjutnya proses penentuan term-term penting dipengaruhi oleh nilai TF-ICF yang dihasilkan.

Tabel 4.0.6 Contoh Perhitungan TF-ICF

id_term	term	tf1	tf2	tf3	Icf	tf_icf1	tf_icf2	tf_icf3
2	rain	8	0	0	1,099	8,789	0	0
5	app	45	2	20	0	0	0	0
6	say	15	0	0	1,099	16,479	0	0
10	start	8	0	2	0,405	3,244	0	0,811
11	do	42	0	5	0,405	17,029	0	2,027
249	would	1	3	0	0,405	0,405	1,216	0
611	should	1	4	0	0,405	0,405	1,622	0

Berdasarkan Tabel 4.8 tf_icf1 menunjukkan nilai suatu term pada cluster 1, tf_icf2 menunjukkan nilai suatu term pada cluster 2, dan tf_icf3 menunjukkan nilai suatu term pada cluster 3. Tabel tersebut menunjukkan bahwa nilai frekuensi suatu term di tiap klaster mampu mempengaruhi nilai ICF dalam menghasilkan nilai akhir TF-ICF. Sehingga semakin tinggi nilai frekuensi suatu term, selama nilai ICF nya tidak nol, maka nilai TF-ICF yang dihasilkan semakin besar. Hal tersebut senada dengan hipotesa peneliti untuk mengambil term-term setiap klaster dengan menggunakan TF-ICF karena semakin term muncul di klaster yang terbatas, maka semakin besar kemungkinan term tersebut mampu merepresentasikan klaster. Sehingga, nilai TF-ICF tertinggi selanjutnya dijadikan sebagai penambahan term pada cluster. Namun, pada penelitian ini, peneliti melakukan beberapa justifikasi terhadap hasil yang diperoleh untuk menentukan term cluster, yaitu:

- Term tidak akan digunakan meskipun nilai TF-ICF tinggi namun memiliki interval yang rendah dengan kluster lainnya

Hal tersebut berdasarkan pertimbangan bahwa, rendahnya nilai TF-ICF menunjukkan term tersebut tidak terlalu dapat merepresentasikan kluster nya. Sehingga, hal tersebut memungkinkan bahwa term tersebut muncul di kluster yang berbeda pada saat proses testing

- Term tidak akan digunakan jika mengandung nama produk dari suatu barang atau aplikasi

Hal tersebut berdasarkan pertimbangan bahwa, penggunaan term yang merupakan nama produk akan menyulitkan pada proses hidden topic similarity dan menambah waktu komputasi.

Pada penelitian ini, selain menguji coba performa *expanded term list* dengan menggunakan WordNet similarity, peneliti juga menguji coba performa penambahan 15%, 30%, 45%, 60%, dan 75% dari term yang telah dikelompokan per kluster. Adapun penetapan term diperoleh dari nilai batas ambang setiap kluster yang berbeda-beda.

id_term	term	tf1	tf2	tf3	icf	tficf1	tficf2	tficf3	Normalisasi	
257	error	38	0	0	1,09861	41,7472	0	0	10	15%
126	even	58	0	18	0,405465	23,517	0	7,29837	5,633192	45%
198	unable	20	0	0	1,09861	21,9722	0	0	5,263155	60%
314	take	49	0	9	0,405465	19,8678	0	3,64918	4,759074	
9	slow	17	0	0	1,09861	18,6764	0	0	4,473689	
136	connect	46	0	2	0,405465	18,6514	0	0,81093	4,467701	
140	connection	42	0	2	0,405465	17,0295	0	0,81093	4,079196	
6	say	40	0	16	0,405465	16,2186	0	6,48744	3,884955	75%
137	network	35	0	4	0,405465	14,1913	0	1,62186	3,399342	
171	try	35	0	16	0,405465	14,1913	0	6,48744	3,399342	
65	stop	33	0	7	0,405465	13,3803	0	2,83825	3,205077	
141	server	33	0	1	0,405465	13,3803	0	0,405465	3,205077	
459	turn	33	0	4	0,405465	13,3803	0	1,62186	3,205077	
224	receive	11	0	0	1,09861	12,0847	0	0	2,894733	
345	blank	11	0	0	1,09861	12,0847	0	0	2,894733	
466	calculate	10	0	0	1,09861	10,9861	0	0	2,631578	

Tabel 4.7 Perhitungan TF-ICF Kelas Bug Error

Tabel 4.7 menunjukkan bahwa term list yang digunakan pada kelas Bug Error menunjukkan bahwa representasi term-term kelas ini merupakan term yang memiliki frekuensi yang tinggi dibandingkan dengan frekuensi term di kelas lainnya. Sehingga, kecenderungan term ini muncul kembali pada dokumen review lainnya besar.

id_term	term	tf1	tf2	tf3	icf	tficf1	tficf2	tficf3	Normalisasi	
1399	biker	0	2	0	1,09861	0	2,19722	0	10	15%
1433	personal	0	2	0	1,09861	0	2,19722	0	10	
1437	ppl	0	2	0	1,09861	0	2,19722	0	10	
1405	backup	0	3	1	0,405465	0	1,2164	0,405465	5,536087	45%
1381	recheck	0	1	0	1,09861	0	1,09861	0	5	60%
1383	vertical	0	1	0	1,09861	0	1,09861	0	5	
1387	fee	0	1	0	1,09861	0	1,09861	0	5	
1390	glance	0	1	0	1,09861	0	1,09861	0	5	
1391	custom	0	1	0	1,09861	0	1,09861	0	5	
1392	uv	0	1	0	1,09861	0	1,09861	0	5	
1393	age	0	1	0	1,09861	0	1,09861	0	5	
1394	appearance	0	1	0	1,09861	0	1,09861	0	5	
1395	determine	0	1	0	1,09861	0	1,09861	0	5	
1396	mustang	0	1	0	1,09861	0	1,09861	0	5	
1397	interstate	0	1	0	1,09861	0	1,09861	0	5	
1398	rest	0	1	0	1,09861	0	1,09861	0	5	
1406	sunrise	0	1	0	1,09861	0	1,09861	0	5	
1407	sunset	0	1	0	1,09861	0	1,09861	0	5	

Tabel 4.8 Perhitungan TF-ICF Kelas Feature Request

Gambar 4.8 menunjukkan bahwa term list yang digunakan pada kelas Feature Request menunjukkan bahwa representasi term-term kelas ini merupakan term yang memiliki frekuensi yang rendah meskipun hanya muncul pada kelas tersebut. Hal tersebut disebabkan karena term yang tinggi di kelas ini cenderung lebih tinggi di kelas lain, sehingga term tersebut harus dibuang. Sedikit nya frekuensi ini membuat pelabelan dokumen review kelas ini cenderung sulit dibandingkan dengan kelas lainnya.

id_term	term	tf1	tf2	tf3	icf	tficf1	tficf2	tficf3		
105	problem	18	0	25	0,405465	7,29837	0	10,1366	10	15%
56	pay	22	0	24	0,405465	8,92023	0	9,73116	9,600024	
617	accurate	9	0	15	0,405465	3,64918	0	6,08197	6,00001	45%

729	money	2	0	15	0,405465	0,81093	0	6,08197	6,00001	
1553	rely	0	0	5	1,09861	0	0	5,49305	5,419026	60%
526	before	11	0	12	0,405465	4,46011	0	4,86558	4,800012	
910	buy	2	0	12	0,405465	0,81093	0	4,86558	4,800012	
255	many	8	0	11	0,405465	3,24372	0	4,46011	4,400006	
1480	regret	0	0	4	1,09861	0	0	4,39444	4,335221	
1488	customer	0	0	4	1,09861	0	0	4,39444	4,335221	
1501	prefer	0	0	4	1,09861	0	0	4,39444	4,335221	
1522	residential	0	0	4	1,09861	0	0	4,39444	4,335221	
1530	impress	0	0	4	1,09861	0	0	4,39444	4,335221	
1554	sort	0	0	4	1,09861	0	0	4,39444	4,335221	
1296	review	1	0	9	0,405465	0,405465	0	3,64918	3,600004	75%
1521	vegas	0	0	3	1,09861	0	0	3,29583	3,251416	
1537	police	0	0	3	1,09861	0	0	3,29583	3,251416	
1541	recommend	0	0	3	1,09861	0	0	3,29583	3,251416	
1623	might	0	0	3	1,09861	0	0	3,29583	3,251416	
1642	must	0	0	3	1,09861	0	0	3,29583	3,251416	
680	actual	5	0	8	0,405465	2,02732	0	3,24372	3,200008	
875	download	4	0	8	0,405465	1,62186	0	3,24372	3,200008	

Tabel 4.9 Perhitungan TF-ICF Kelas Non Informative

Tabel 4.9 menunjukkan bahwa term list yang digunakan pada kelas Non Informative menunjukkan bahwa representasi beberapa term kelas ini merupakan term yang memiliki frekuensi yang hampir mirip dibandingkan dengan frekuensi term di kelas Bug Error. Sehingga, kecenderungan term ini muncul kembali pada dokumen review kelas Bug Error relatif besar.

4.1.3 LDA Clustering

Metode clustering yang digunakan pada penelitian kali ini adalah Latent Dirichlet Allocation (LDA). Hasil yang diperoleh dari proses ini adalah hidden topic dan nilai probabilitasnya dari dokumen yang diinputkan.

Hidden topic yang diperoleh merupakan hasil identifikasi topik dari kata, dokumen, maupun keseluruhan corpus. Jumlah dari hidden topic yang dikeluarkan tidak ditentukan sebelumnya. Hal tersebut disebabkan karena panjang dokumen yang cenderung berbeda-beda. Setelah memperoleh hidden topic untuk masing-masing dokumen, maka nilai similaritas dari hidden topic dihitung dengan menggunakan term list yang telah didefinisikan sebelumnya.

4.1.4 Hidden Topic Similarity

Proses hidden topic similarity merupakan tahapan terakhir pada proses pelabelan. Input dari proses ini adalah term list yang diperoleh dari proses *expanding term list* dan hidden topic per dokumen.

Selanjutnya, setiap hidden topic akan dihitung nilai similaritas setiap clusternya. Nilai similaritas cluster tertinggi dijadikan label dari dokumen tersebut. Adapun skala nilai yang dihasilkan adalah 1 dan 0. Dimana semakin mendekati nilai 1, maka hidden topic tersebut benar-benar mirip dan kemungkinan besar mampu merepresentasikan cluster tersebut. Sedangkan sebaliknya, nilai 0 merepresentasikan bahwa hidden topic sangat tidak mirip dan tidak dapat merepresentasikan cluster. Pada penelitian ini, setiap nilai -1 yang dihasilkan pada proses perhitungan kelas akan dijadikan nilai 0. Hal tersebut dengan pertimbangan bahwa nilai -1 dapat mempengaruhi nilai keseluruhan dari similaritas. Sehingga term yang memiliki -1 hanya dinyatakan sebagai term yang tidak memiliki kemiripan.

Penelitian ini hendak menguji coba performa perhitungan similaritas antara dua kata dengan menggunakan Wu Palmer dan Path Similarity karena pengukuran ini memiliki kesamaan dalam cara kerja yang berbasis thesaurus.

4.1.5 Klasifikasi

Proses ini mengelola input yang diperoleh dari proses pelabelan. Data yang diperoleh sebelum diklasifikasi dihitung nilai vektor untuk mempermudah proses klasifikasi. Word Embedding akan digunakan untuk memperoleh representasi vektor dari kata-kata setiap dokumen. Adapun nilai vektor yang digunakan adalah nilai rerata dari vektor kata-kata yang ada dalam suatu dokumen. Word Embedding menggunakan perhitungan berdasarkan statistik kemunculan kata dari suatu corpus yang telah disediakan oleh Word Embedding (*Pre-trained word vectors*). Pada penelitian ini, GloVe digunakan sebagai word embedding.

Setelah memperoleh vektor yang digunakan sebagai fitur pada proses klasifikasi, selanjutnya LSTM digunakan sebagai metode klasifikasi. Pada penelitian ini, dokumen diklasifikasi apakah termasuk klaster tertentu atau tidak.

Hal ini dikenal dengan nama *binary classification*. Pada kasus ini, setiap dokumen diuji sebanyak tiga kali, yaitu (1) Bug Report atau tidak, (2) Feature Request atau tidak, dan (3) Non Informative atau tidak. Selain itu, metode yang diusulkan menggunakan Convolutional Neural Network (CNN) sebagai metode pembandingan.

4.2. Hasil pengujian dan Analisis

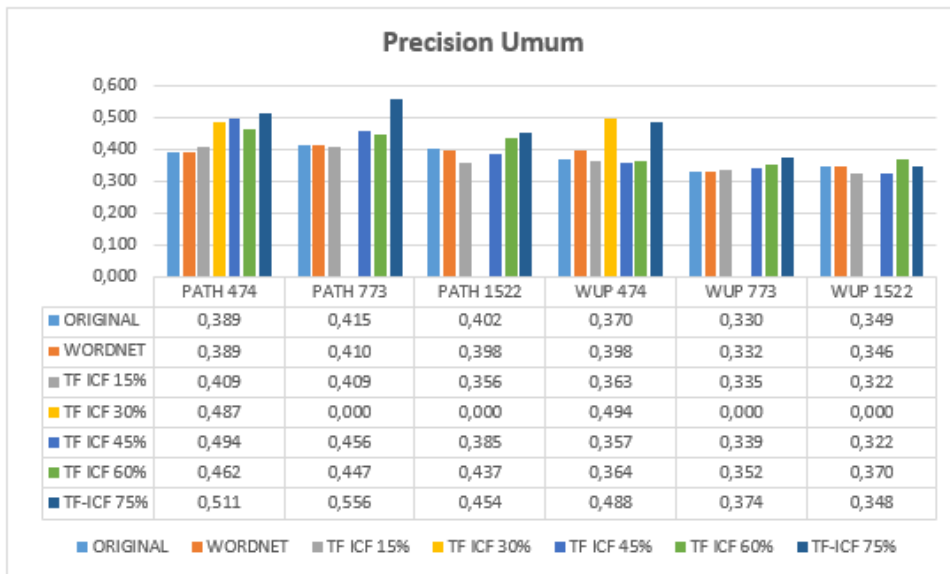
Setelah memperoleh expanding term list, modul proses pelabelan selanjutnya dimulai dengan menjalankan sub modul LDA clustering. Hasil hidden topic yang diperoleh pada LDA cluster, selanjutnya dihitung nilai similarity nya terhadap hasil expanding term list. Nilai similarity terbesar selanjutnya dijadikan label untuk dokumen tersebut. Terdapat beberapa kombinasi rule yang dilakukan pada skenario pengujian klasterisi, yaitu:

1. Penggunaan dua metode perhitungan similarity untuk mengetahui performa expanded term list pada proses pelabelan dokumen sebanyak 15%, 30%, 45%, dan 60% yaitu Wu Palmer dan Path Similarity (Skenario Pengujian 1, 2, dan 3)
2. Klasifikasi dokumen review yang telah dilabeli dengan menggunakan metode Long Short Term Memory (LSTM) dan Convolutional Neural Network (CNN) dengan menggunakan pre trained Word Vector GloVe (Skenario Pengujian 4).

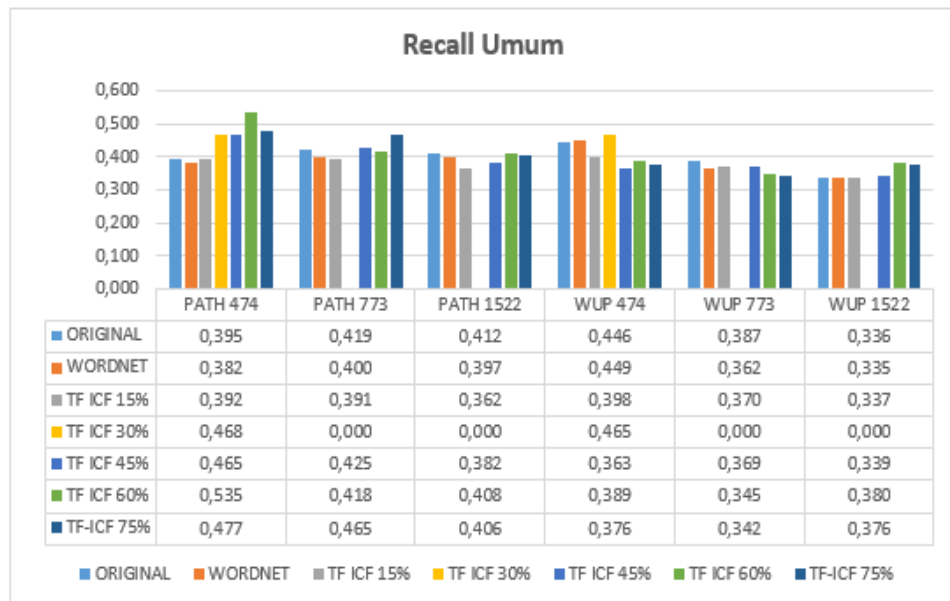
Pengujian kinerja dilakukan dengan membandingkan hasil metode dengan hasil anotasi yang telah dilakukan oleh annotator yang telah dijelaskan pada tahap sebelumnya.

4.3.1 Hasil Pengujian Pelabelan Dokumen

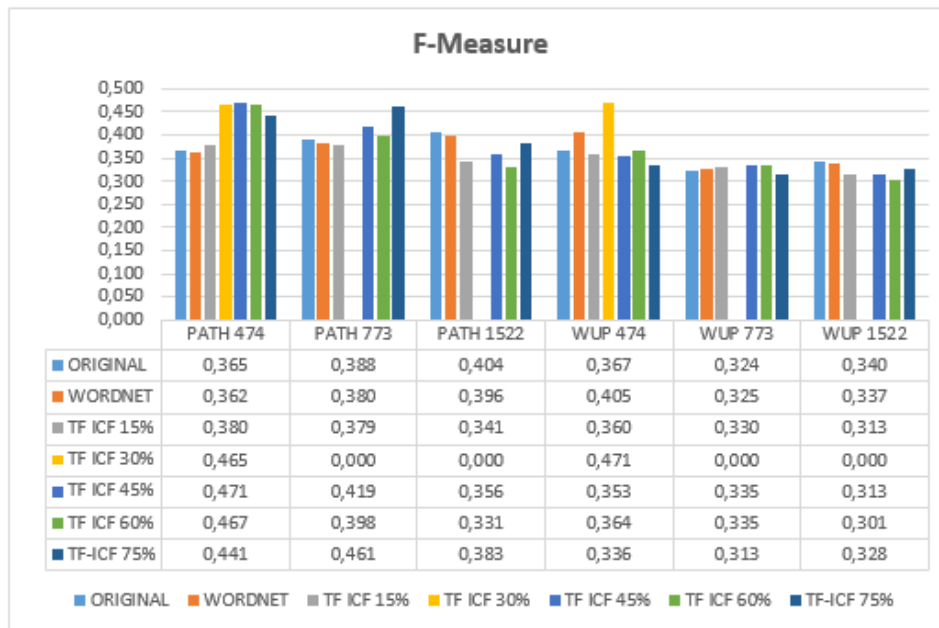
Hasil pengujian pelabelan dokumen merupakan hasil yang diperoleh dari gabungan uji coba pada skenario 1, 2, dan 3. Berdasarkan performa untuk setiap kategori yang ditunjukkan pada tabel di atas, penulis hendak menganalisa performa metode yang diusulkan secara umum. Adapun performa dari rerata nilai yang ditunjukkan pada visualisasi precision, recall, dan f-measure pada Gambar 4.5, 4.6, dan 4.7



Gambar 4.2 Precision Uji Coba Dokumen Keseluruhan



Gambar 4.3 Recall Uji Coba Dokumen Keseluruhan



Gambar 4.4 F-Measure Uji Coba Dokumen Keseluruhan

Secara umum, tabel diatas menunjukkan hasil pelabelan tiga jenis data. Data 474 dan 773 merupakan dokumen yang berisi data yang telah disepakati oleh tiga annotator. Performa dokumen yang terbaik secara umum terlihat pada Data 773 Penambahan Term Expansion sebanyak 75% dengan nilai precision 0,556, recall 0,465, dan f-measure 0,461.

Berdasarkan gambar di atas, dapat terlihat bahwa semakin banyak data yang digunakan, kecenderungan performa yang ditampilkan semakin baik. Hal tersebut disebabkan karena setiap data yang dilabeli telah disepakati oleh ketiga annotator. Sehingga, confidence level setiap dokumen cenderung tinggi. Namun, hal tersebut tidak sepenuhnya bekerja dengan baik pada Data 1522. Data ini terdiri dari data yang disepakati oleh dua annotator atau lebih, sehingga memungkinkan suatu dokumen memiliki label lebih dari satu. Faktor ini yang mampu menjadi salah satu penyebab. Kurang tingginya confidence level setiap dokumen dibandingkan dengan karakteristik data sebelumnya membuat performa tidak sebaik Data 474 dan Data 774.

Jika membandingkan performa similaritas terhadap hasil pelabelan, maka dapat terlihat bahwa path mampu lebih baik dibandingkan dengan wu palmer similarity. Kedua metode similariti ini dihitung berdasarkan pada kamus/tesaurus.

Path similarity menghitung jarak terdekat antara kata dengan kata lain pada struktur hierarkis, dimana semakin jauh jarak, maka semakin dinyatakan berbeda. Namun, Wu Palmer similarity tidak memerlukan fitur dalam jalur terpendek untuk mencari hubungan kedua kata, tetapi metode ini melakukan pencarian terdalam dari ancestor pada taksonomi yang sama, bukan yang paling dekat dengan kedua indra.

Jika dilihat dari penambahan data pada setiap TF-ICF, terlihat bahwa pola kecenderungan secara umum adalah adanya peningkatan performa pada setiap penambahan data. Namun, terdapat beberapa kondisi dimana penambahan persentase dari penambahan data cenderung menurunkan performa dari pelabelan. Hal tersebut kemungkinan besar disebabkan oleh dua faktor, yaitu tidak adanya data yang memenuhi batas interval, misal penambahan TF-ICF 30% pada Data 773 dan Data 1522. Karena tidak ada penambahan data dari ketiga kelas, maka penulis mengasumsikan nilai nya 0. Adapun faktor kedua yang memungkinkan mempengaruhi performa adalah ketidak-tepatan term list untuk diimplementasikan pada data tersebut, seperti yang ditunjukkan pada penggunaan penambahan TF-ICF 45% ke 60%.

Berdasarkan analisa secara umum, TF-ICF mampu memberikan penambahan term yang mampu berpengaruh terhadap peningkatan performa hampir di setiap data. Namun, peningkatan yang diberikan belum mampu secara maksimal terlihat, sehingga penulis hendak menguji coba untuk mengambil term list terbaik untuk setiap kelas (bug error, feature request, dan non informative) untuk di uji coba di setiap data. Adapun penggabungan term list menggunakan pengaturan berikut:

Tabel 4.0.10 Konfigurasi untuk memperoleh *The Best Term List*

Konfigurasi	Nama
Metode Similaritas	Path Similarity
Term List	
Bug Error	Path 773 – TF ICF 75%
Feature Request	Path 474 – TF ICF 60%
Non Informative	Path 1522 – TF ICF 60%

Parameter ini selanjutnya digunakan untuk mengamati performa pelabelan dokumen dengan skenario ukuran dokumen.

Tabel 4.0.11 *The Best Term List* untuk Proses Pelabelan

Kategori	Term List
Bug Report	'bug', 'fix', 'problem', 'issue', 'defect', 'crash', 'solve', 'repair', 'problem', 'error', 'resolve', 'clear', 'server', 'update', 'get', 'map', 'no', 'turn', 'off', 'current', 'sometimes', 'work', 'unable', 'gps', 'wrong', 'even', 'connect', 'slow', 'signal', 'constant', 'say', 'connection', 'try'
Feature Request	'add', 'please', 'could', 'would', 'hope', 'improve', 'miss', 'need', 'prefer', 'request', 'should', 'suggest', 'want', 'wish', 'allow', 'complaint', 'improvement', 'instead', 'lacks', 'maybe', 'must', 'want', 'will', 'give', 'lend', 'bring', 'better', 'choose', 'propose', 'should', 'biker', 'video', 'news', 'category', 'rather', 'cache', 'recheck', 'label', 'vertical', 'may', 'detail', 'fee', 'glance', 'custom', 'uv', 'age', 'appearance', 'determine', 'mustang', 'obvious', 'interstate', 'rest', 'hohway', 'whell', 'above', 'special', 'dropdown', 'instead', 'backup'
Non Informative	'help', 'support', 'assist', 'when', 'situation', 'aid', 'serve', 'great', 'good', 'nice', 'very', 'cool', 'love', 'hate', 'bad', 'worst', 'problem', 'pay', 'accurate', 'money', 'rely', 'before', 'buy', 'many', 'regret', 'customer', 'prefer', 'residential', 'impress', 'sort'

Selanjutnya, karakteristik yang diambil dari aspek-aspek terbaik ini akan dijadikan parameter yang digunakan untuk melihat apakah terdapat perubahan performa yang diperoleh untuk setiap data.

Tabel 4.0.12 Hasil Analisa Penggunaan Parameter Terbaik untuk Setiap Data

		Sebelum	Data 475	Data 773	Data 1522
Bug Report	Precision	0,553	0,783	0,784	0,740
	Recall	0,641	0,870	0,875	0,688
	F-Measure	0,585	0,824	0,827	0,713
Feature Request	Precision	0,184	0,400	0,452	0,461
	Recall	0,135	0,129	0,146	0,153
	F-Measure	0,114	0,195	0,220	0,230

Non Informative	Precision	0,349	0,432	0,452	0,350
	Recall	0,307	0,461	0,464	0,572
	F-Measure	0,311	0,446	0,458	0,434
Average Score	Precision	0,362	0,538	0,562	0,517
	Recall	0,361	0,487	0,495	0,471
	F-Measure	0,337	0,488	0,502	0,459
Correct Labeled Data		-	330	544	884

Berdasarkan tabel di atas, dapat terlihat bahwa dibandingkan dengan metode sebelumnya, perubahan term list mampu berpengaruh cukup signifikan pada performa seluruh data. Performa terbaik ditunjukkan pada Data 773 dengan skor Precision 0,562, Recall 0,495, dan F-Measure 0,502. Perolehan ini secara umum menunjukkan bahwa term list yang telah didefinisikan mampu mendeskripsikan Data 773 lebih baik dibandingkan dengan data yang lainnya. Adapun analisa performa untuk setiap Data akan dijelaskan pada tabel berikut:

Tabel 4.0.13 Confussion Matrix untuk Data 475, 773, dan 1522

		475 DATA			773 DATA		
		Prediksi Sistem			Prediksi Sistem		
		A	B	C	A	B	C
Label	A	281	7	35	460	8	58
	B	35	8	19	55	14	27
	C	43	5	41	72	9	70

		1522 DATA		
		Prediksi Sistem		
		A	B	C
Label	A	650	26	269

	B	94	35	100
	C	134	15	199

Kelas Bug Error atau yang ditandai oleh huruf A pada tabel di atas memiliki performa terbaik pada Data 773 dengan nilai Precision 0,784, Recall 0,875, dan F-measure 0,827. Jika dibandingkan dengan kelas lainnya, maka kelas bug error merupakan kelas dengan performa terbaik. Hal tersebut tidak hanya didukung dari jumlah data yang banyak, namun hasil analisa pembentukan term expansion dengan TF-ICF menunjukkan bahwa nilai TF-ICF setiap term yang mendeskripsikan kelas ini sangat tinggi dibandingkan kelas lainnya. Hal ini mengindikasikan bahwa *distinct term* untuk kelas ini sangat baik sehingga dapat secara optimal dalam proses pelabelan yang ditujukan dengan tingginya nilai True Positive untuk seluruh data.

Tabel 4.0.14 Contoh Kesalahan Klasifikasi Kelas Bug Report

Id	Review	Similarity			Prediksi Sistem	Label Sebenarnya
		Bug Report	Feature Request	Non Informative		
365	i drive a diesel so regular gas prices do not help me .	0,111	0,105	0,126	3	1

Tabel 4.13 menunjukkan salah satu kesalahan klasifikasi dari kelas bug report yang terprediksi oleh model sebagai kelas non informative. Jika diperhatikan secara sintaktik dan makna secara umum, review tersebut menjelaskan tentang ke-kurang bermanfaat salah satu fitur dari aplikasi. Namun, dikarenakan model yang digunakan hanya memperhatikan aspek similaritas terhadap term yang telah didefinisikan sebelumnya maka review tersebut diprediksi menjadi kelas yang salah.

Kelas Feature Request atau yang ditandai oleh huruf B pada tabel di atas memiliki performa terbaik pada Data 1522 dengan nilai Precision 0,461, Recall 0,153, dan F-measure 0,230. Kelas ini merupakan kelas yang memiliki data yang paling sedikit dibandingkan dengan kelas lainnya. Hal tersebut membuat proses

pembentukan term expansion dengan TF-ICF pada kelas ini tidak sebaik kelas lainnya. Pada proses penemuan term tambahan, nilai tf dari kelas ini cenderung sedikit. Hal ini membuat tingkat kepercayaan terhadap term tambahan semakin menurun ketika diimplementasikan pada proses pelabelan. Seperti yang ditunjukkan pada tabel di atas bahwa skor true positif kelas ini cenderung kecil dibandingkan skor false negatif dan true negatif nya. Kecenderungan ini mengindikasikan bahwa term yang mendeklarasikan kelas ini belum dapat bekerja secara optimal dalam proses pelabelan.

Tabel 4.0.15 Contoh Kesalahan Klasifikasi Kelas Feature Request

Id	Review	Similarity			Prediksi Sistem	Label Sebenarnya
		Bug Report	Feature Request	Non Informative		
10	would it be possible to cache last loaded forecasts and such as to not use data to recheck a forecast or if data is not available .	0,059	0,055	0,059	1	2

Tabel 4.14 menunjukkan salah satu kesalahan klasifikasi dari kelas feature request yang terprediksi oleh model sebagai kelas bug report. Jika diperhatikan secara sintaktik dan makna secara umum, terdapat kata ‘*would it be possible ...*’ pada review tersebut yang menjelaskan tentang saran yang dapat digunakan untuk pengembangan fitur dari aplikasi. Namun, dikarenakan model yang digunakan hanya memperhatikan aspek similaritas dan mengabaikan pola kemunculan setiap kelas terutama kelas feature request terhadap term yang telah didefinisikan sebelumnya maka review tersebut diprediksi menjadi kelas yang salah.

Kelas Non-Informative atau yang ditandai oleh huruf C pada tabel di atas memiliki performa terbaik pada Data 773 dengan nilai Precision 0,452, Recall 0,464, dan F-measure 0,458. Kelas ini memiliki data yang lebih banyak dari kelas feature request namun lebih sedikit dibandingkan dengan bug error. Penyebaran kelas ini membuat proses pembentukan term expansion dengan TF-ICF sulit

diperoleh. Hal tersebut juga memperkuat pernyataan bahwa banyaknya jumlah data tidak selalu berbanding lurus dengan tingginya tingkat confidence dari term untuk kelas ini. Penyebab nya adalah banyak term yang digunakan pada kelas ini juga digunakan oleh kelas lainnya terutama kelas bug error. Hal tersebut juga diperkuat dengan hasil pada tabel di atas. Tabel di atas menunjukkan data yang terdapat pada kelas non informative banyak yang misklasifikasi terhadap kelas bug error. Hampir berimbangnya nilai yang terdeteksi benar dan misklasifikasi untuk setiap kelas mengindikasikan bahwa term list yang digunakan kurang deskriptif dan mewakili kelas ini. Hal tersebut membuat nilai False Negative yang ditunjukkan warna merah paling banyak disumbang oleh kelas Non Informative.

Tabel 4.0.16 Contoh Kesalahan Klasifikasi Kelas Non Informative

Id	Review	Similarity			Prediksi Sistem	Label Sebenarnya
		Bug Report	Feature Request	Non Informative		
1122	that is as accurate as it gets .	0.102	0.087	0.097	1	3

Tabel 4.15 menunjukkan salah satu kesalahan klasifikasi dari kelas non informative yang terprediksi oleh model sebagai kelas bug report. Jika diperhatikan secara sintaktik dan makna secara umum, ekspresi kepuasan dari fitur pada aplikasi. Namun, dikarenakan model yang digunakan hanya memperhatikan aspek similaritas terhadap term yang telah didefinisikan sebelumnya maka review tersebut diprediksi menjadi kelas yang salah. Pada proses pelabelan, jenis kata yang lebih dominan berpengaruh pada perhitungan pelabelan adalah sebagai berikut

- Jumlah Kata Benda : 2577
- Jumlah Kata Kerja : 865
- Jumlah Kata Sifat : 993
- Lain-lain : 867

4.3.2 Hasil Pengujian Klasifikasi

Hasil pengujian klasifikasi dokumen merupakan hasil yang diperoleh dari skenario 4. Modul ini mengklasifikasi dokumen yang telah terlabeli benar pada proses sebelumnya. Metode yang digunakan pada modul ini merupakan pengklasifikasian Long Short Term Memory (LSTM) dengan menggunakan word embedding. Hasil yang diperoleh selanjutnya dibandingkan dengan metode deep learning lainnya, yaitu Convolutional Neural Network (CNN) 1D Max Pooling.

Tabel 4.0.17 Tabel Komposisi Data Training dan Testing

		Training	Testing
DATA 330	Bug Error	281	70
	Feature Request	8	2
	Non Informative	41	10
	Total	330	82
DATA 544	Bug Error	460	115
	Feature Request	14	3
	Non Informative	70	18
	Total	544	136
DATA 884	Bug Error	650	162
	Feature Request	35	9
	Non Informative	199	50
	Total	884	221

Pada penelitian ini, klasifikasi yang dilakukan adalah binary classification. Dokumen diklasifikasi apakah dokumen termasuk kelas tertentu atau tidak. Pada kasus ini, setiap dokumen dihitung sebanyak tiga kali, yaitu (1) Bug Report atau tidak, (2) Feature Request atau tidak, dan (3) Non Informative atau tidak. Hal tersebut dilakukan untuk melihat performa metode klasifikasi dalam mengklasifikasikan dokumen multi label.

Tabel 4.0.18 Tabel Rerata Akurasi LSTM dan CNN

LSTM	CNN
------	-----

		300	544	884	300	544	884
		Data	Data	Data	Data	Data	Data
Bug Report	Accuracy	85,37	84,56	93,10	84,15	88,24	82,18
	Precision	100,00	100,00	100,00	97,14	97,39	84,57
	Recall	85,37	84,56	93,10	86,08	89,60	95,80
Feature Request	Accuracy	98,78	97,79	94,83	97,56	97,79	91,95
	Precision	50,00	0,00	0,00	0,00	0,00	11,11
	Recall	100,00	0,00	0,00	0,00	0,00	14,29
Non Informative	Accuracy	87,80	86,76	69,66	86,59	89,71	75,86
	Precision	0,00	0,00	0,00	10,00	50,00	48,00
	Recall	0,00	0,00	0,00	33,33	64,29	60,00
Average	Accuracy	90,65	89,71	85,86	89,43	91,91	83,33
	Precision	50,00	33,33	33,33	35,71	49,13	47,89
	Recall	61,79	28,19	31,03	39,80	51,30	56,70

Tabel 4.0.19 Tabel Rerata Akurasi Penggunaan GloVe pada LSTM dan CNN

		LSTM + GloVe			CNN + Glove		
		300	544	884	300	544	884
		Data	Data	Data	Data	Data	Data
Bug Report	Accuracy	87,80	88,24	91,38	86,59	90,44	79,31
	Precision	100,00	98,26	96,30	100,00	94,78	80,86
	Recall	87,50	88,98	94,55	86,42	93,97	96,32
Feature Request	Accuracy	97,56	97,79	94,83	97,56	97,79	94,83
	Precision	0,00	0,00	0,00	0,00	0,00	0,00
	Recall	0,00	0,00	0,00	0,00	0,00	0,00
Non Informative	Accuracy	87,80	88,97	79,31	87,80	88,97	77,01
	Precision	0,00	44,44	46,00	10,00	27,78	44,00
	Recall	0,00	61,54	71,88	50,00	71,43	64,71
Average	Accuracy	91,06	91,67	88,51	90,65	92,40	83,72
	Precision	33,33	47,57	47,43	36,67	40,85	41,62

Recall	29,17	50,17	55,47	45,47	55,13	53,68
--------	-------	-------	-------	-------	-------	-------

Berdasarkan Tabel 4. 17 Performa metode secara umum menunjukkan bahwa LSTM mampu lebih unggul dibandingkan dengan CNN dalam mengklasifikasi dokumen baik yang menggunakan GloVe maupun tidak. Tabel 4.18 menunjukkan bahwa penggunaan Pre Trained GloVe mampu meningkatkan performa klasifikasi.

Jika melihat performa metode untuk setiap kelas, dapat dilihat bahwa pada kelas Bug Report, 884 Data menjadi data yang memperoleh nilai akurasi klasifikasi terbaik yaitu 93,10%. Kelas ini menunjukkan bahwa, semakin bertambahnya data, tidak membuat nilai akurasi dari klasifikasi berkurang. Hal ini menunjukkan bahwa meskipun data yang digunakan pada data 884 merupakan data yang multi label, LSTM mampu dengan optimal mengklasifikasi. Namun, penambahan data dari 300 data menjadi 544 data menunjukkan penurunan performa dari kedua metode klasifikasi, meskipun LSTM mampu menunjukkan performa yang lebih baik dibandingkan dengan CNN. Selain itu, perolehan skor precision dan recall pada kelas ini memperlihatkan bahwa nilai true positive yang dimiliki oleh kelas ini lebih besar dibandingkan dengan nilai false negative atau pun true negative. Sehingga baik LSTM maupun CNN mampu mengklasifikasi kelas ini dengan baik.

Pada kelas Feature Request, tidak ada perbedaan yang signifikan antara penggunaan metode CNN atau LSTM. Selain itu, data 300 menunjukkan bahwa LSTM mampu menunjukkan performa akurasi terbaik dibandingkan dengan CNN serta data lainnya yang diuji coba, yaitu sebesar 98,78%. Namun, 884 Data menunjukkan bahwa dengan bertambahnya data, nilai akurasi dari klasifikasi sedikit berkurang. Hal ini disebabkan oleh jumlah true negative dari kedua metode ini cenderung meningkat. Jika diamati, skor precision dan recall dari kelas ini merupakan salah satu skor kelas terburuk meskipun nilai akurasi yang diperoleh sangat baik. Nilai akurasi yang baik tersebut didukung oleh nilai true negative yang baik, tetapi tidak untuk true positive. Seluruh data uji dan kedua metode menunjukkan bahwa ketidakseimbangan data dari proses training model mampu mempengaruhi kemampuan komputer untuk memahami pola kalimat pada kelas feature request.

Pada kelas Non Informative, uji coba yang dilakukan pada 544 data menunjukkan performa terbaik dengan menggunakan CNN dibandingkan uji coba menggunakan data lainnya. Sedangkan uji coba dengan menggunakan 884 menunjukkan performa terburuk untuk ke empat metode, terutama metode LSTM. Salah satu faktor yang menungkinan kelas ini memiliki performa yang paling rendah dibandingkan kelas lainnya adalah karena kecenderungan kelas non informative yang lebih mudah terdeteksi ke dalam kelas lainnya. Jika dibandingkan performa antara metode LSTM dan CNN, terlihat bahwa sepiintas metode CNN mampu mengungguli metode LSTM. Hal tersebut disebabkan oleh kemampuan LSTM yang hanya mampu mendeteksi true positive dan true negative. Sedangkan, pada data 884 nilai dari true positive lebih kecil dibandingkan dengan true negative. Sehingga nilai akurasi yang dihasilkan pun kurang optimal.

Adapun perbandingan antara LSTM dan CNN pada proses *running* program adalah CNN mampu dua atau tiga kali lebih cepat untuk mengklasifikasi dokumen dibandingkan dengan LSTM.

4.3.3 Analisa Performa

Metode yang diusulkan untuk pelabelan dokumen mampu dengan baik melabeli kelas bug error, namun kurang begitu optimal dalam pelabelan dengan kategori feature request dan non informative. Salah satu metode yang dapat dilakukan selanjutnya adalah dengan tidak lagi fokus pada penambahan term list. Hal tersebut disebabkan oleh kemungkinan salah satu faktor yang membuat term list tidak berpengaruh secara signifikan adalah karena term list hanya mempertimbangkan kemunculan term namun mengabaikan posisi term atau struktur dari kalimat tersebut.

Tabel 4.0.20 Contoh Kesalahan Klasifikasi pada Pelabelan

<p><i>“would it be possible to cache last loaded forecasts and such as to not use data to recheck a forecast or if data is not available”</i> – Feature Request Review</p>
--

Selain itu, penggunaan semantic similarity yang digunakan mampu dengan baik dalam proses pelabelan. Namun, tidak semua review dapat terlabeli dengan baik. Jika diperhatikan dari Tabel 4.18, review tersebut merupakan kesalahan

klasifikasi dari review feature request. Meskipun review tersebut dengan jelas menyatakan kategori feature request, namun aspek similaritas saja dirasa tidak cukup. Sehingga berdasarkan posisi kata dan hubungan antar kata pada suatu kalimat seharusnya dijadikan salah satu bahan pertimbangan untuk melabeli dokumen, seperti linguistic rule kata '*would it be possible ...*' untuk menyatakan kelas feature request. Oleh karena itu, selain memperhatikan aspek semantik, sebaiknya metode selanjutnya dapat mengoptimalkan aspek sintaktik dalam kalimat, misalnya dengan menggunakan linguistic rule atau word embedding dalam tahap pra proses atau proses pelabelan dokumen. Hal tersebut diharapkan mampu memberikan performa yang lebih baik dari proses pelabelan.

Metode klasifikasi yang diusulkan dapat dengan optimal mengklasifikasi dokumen terutama ke dalam kelas Bug Report. Hal tersebut disebabkan oleh data training bug report lebih besar dibandingkan kelas lainnya yang membuat proses learning model mampu lebih mudah mengenali kelas ini. Berbeda dengan kelas Bug Report, uji coba pada kelas Feature Request dan Non Informative menunjukkan bahwa metode yang diusulkan maupun metode pembanding kurang cakap dalam mengklasifikasi dokumen ke dalam masing-masing kelas. Salah satu faktor yang memungkinkan terjadinya hal tersebut diantaranya adalah ketidakseimbangan data dan penggunaan term dalam mengekspresikan pernyataan setiap kelas. Ketidakseimbangan kelas membuat proses pelatihan kelas dengan jumlah lebih sedikit menjadi kurang maksimal, sehingga ia menjadi data yang terklasifikasi sebagai kelas yang dominan. Hal tersebut membuat term dan topik dari kelas lebih kecil kemungkinan dia lebih susah terdeteksi.

Oleh karena itu, untuk memastikan permasalahan ketidakseimbangan data, penulis menguji coba untuk mengklasifikasi data yang terdiri dari kelas feature request dan non informative. Hasil yang diperoleh dari uji coba tersebut ternyata model tetap mengklasifikasikan data feature request ke dalam kelas non informative. Di sisi lain, peneliti juga hendak menguji coba untuk mengklasifikasikan data yang terdiri dari kelas feature request dan beberapa data tentang start up company. Ternyata hasil menunjukkan bahwa jika komposisi perbandingan data kedua kelas tersebut seimbang, maka data testing dari feature

request dapat terdeteksi seluruhnya. Ketika data training dari kelas feature request dikurang setengahnya, maka baik CNN maupun LSTM tidak dapat mendeteksi, namun penggunaan GloVe dapat mendeteksi satu dari tiga data testing. Namun ketika data kelas feature request yang tersisa dikurangi kembali setengah, semua metode uji coba tidak dapat mendeteksi kecuali metode Glove + LSTM. Hal ini menunjukkan bahwa bagi kelas feature request, dibutuhkan tidak hanya data yang seimbang tetapi juga penggunaan term yang lebih bervariasi yang mencerminkan karakteristik kelas tersebut. Sehingga dibutuhkan metode untuk mengenerate sintetik term. Dengan menggunakan sintetik term pada kelas yang seimbang dengan kelas lainnya, diharapkan dapat muncul term-term yang membuat kinerja pengklasifikasian kelas ini lebih optimal.

[Halaman ini sengaja dikosongkan]

BAB 5

KESIMPULAN DAN SARAN

Pada bab terakhir ini, ditarik beberapa kesimpulan yang didapat dari hasil penelitian dan saran-saran yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan atau riset selanjutnya

5.1 Kesimpulan

Berdasarkan metode yang telah diimplementasikan dan hasil uji coba yang diperoleh, maka dapat ditarik beberapa kesimpulan:

1. Pelabelan dokumen dapat dengan baik dilakukan dengan menggunakan metode LDA Similarity dengan menggunakan Term Expansion berbasis TF ICF. terutama untuk klasifikasi kelas Bug Error dengan nilai 78,4% untuk precision, 87,5% untuk recall, dan 82,7% untuk F-Measure, meningkat lebih dari 25% dari sebelumnya.
2. Klasifikasi dokumen dapat dengan baik dilakukan dengan menggunakan Long Short Term Memory (LSTM) dengan Pre Trained GloVe terutama untuk klasifikasi kelas Bug Error.
3. Seperti ditunjukkan pada skenario pengujian ke-4, performa hasil klasifikasi terbaik diperoleh pada 544 data dengan rerata nilai akurasi sebesar 92,4%.

5.2 Saran

Beberapa saran atas pengerjaan tesis ini guna pengembangan lebih lanjut diantaranya adalah:

1. Penggunaan semantic similarity dapat dikuatkan dengan mempertimbangkan posisi kata serta hubungan kata pada kalimat dengan menggunakan beberapa metode seperti word embedding, posisi kata pada linguistic rule, serta pembobotan kata (term weighting)
2. Untuk mengatasi permasalahan ketidakseimbangan data pada proses klasifikasi, perlu adanya teknis selanjutnya yang mampu mempersiapkan dokumen terutama pada kelas feature request agar dapat dikenali dan mengatasi ketidakseimbangan data yang terjadi, seperti penggunaan metode oversampling, undersampling, atau pembuatan data sintetik yang lebih optimal.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Basari, A.S.H. et al., 2013. Reviewon Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization. *Procedia Engineering*, 53, pp.453–462.
- Bengio, Y. et al., 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3, pp.1137–1155.
- Blei, D.M., Ng, A.Y. & Jordan, M.I., 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, pp.993–1022.
- Brody, S. & Elhadad, N., 2010. An Unsupervised Aspect-Sentiment Model for Online Reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*. pp. 804–812.
- Carre, L.V.G. & Winbladh, K., 2013. Analysis of User Comments : An Approach for Software Requirements Evolution. In *35th International Conference on Software Engineering (ICSE) 2013*. pp. 582–591.
- Chen, N. et al., AR-Miner : Mining Informative Reviews for Developers from Mobile App Marketplace. , pp.767–778.
- Collobert, R. et al., 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12, pp.2493–2537.
- Fu, Y. et al., 2015. Automated Classification of Software Change Messages by Semi-supervised Latent Dirichlet Allocation. *Journal of Information and Software Technology*, 57, pp.369–377.
- Hamdan, H., Bellot, P. & Bechet, F., 2015. Lsislif : CRF and Logistic Regression for Reviewon Target Extraction and Sentiment Polarity Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. pp. 753–758.
- Hochreiter, S. & Schmidhuber, J., 1997. Long Short Term Memory. *Neural Computation*, 9, pp.1735–1780.
- Hu, M. & Liu, B., 2004. Mining and Summarizing Customer Reviews. In *Proceeding KDD '04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 168–177.

- Hu, M., Liu, B. & Street, S.M., 2004. Mining and Summarizing Customer Reviews.
- Li, B. et al., 2010. A Unified Graph Model for Sentence-based Review on Retrieval. , (July), pp.1367–1375.
- Liu, B. et al., Review on Observer : Analyzing and Comparing Reviews on the Web.
- Liu, P., Joty, S. & Meng, H., 2015. Fine-grained Review Mining with Recurrent Neural Networks and Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pp. 1433–1443.
- Liu, Y. et al., 2013. Computer-Aided Design Identifying helpful online reviews : A product designer ' s perspective. *Computer-Aided Design*, 45(2), pp.180–194. Available at: <http://dx.doi.org/10.1016/j.cad.2012.07.008>.
- Liu, Y., Lu, W.F. & Loh, H.T., 2007. KNOWLEDGE DISCOVERY AND MANAGEMENT FOR PRODUCT DESIGN THROUGH TEXT MINING – A CASE STUDY OF ONLINE INFORMATION. , (August), pp.1–12.
- Maalej, W., 2015. Bug Report , Feature Request , or Simply Praise? On Automatically Classifying App Reviews. , pp.116–125.
- Mauczka, A. et al., 2012. Tracing Your Maintenance Work – A Cross-Project Validation of an Automated Classification Dictionary for Commit Messages. In *15th International Conference, FASE 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS*. pp. 301–315.
- Mikolov, T., Sutskever, H., et al., 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*. pp. 1–9.
- Mikolov, T., Corrado, G., et al., 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv*, pp.1–12.
- Panichella, S., Sorbo, A. Di & Guzman, E., 2015. How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution. , (2), pp.281–290.
- Pennington, J., Socher, R. and Manning, C., 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

- Swanson, E.B., 1972. The Dimension of Maintenance.
- Wang, P. et al., 2016. Semantic Expansion using Word Embedding Clustering and Convolutional Neural Network for Improving Short Text Classification. *Journal of Neurocomputing*, 174, pp.806–814.
- Wu, M.-S. & Chien, J.-T., 2010. A New Topic-Bridged Model for Transfer Learning. In *Acoustics Speech and Signal Processing*. pp. 5346–5349.
- Xianghua, F. et al., 2013. Knowledge-Based Systems Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. *Knowledge-Based Systems*, 37, pp.186–195. Available at: <http://dx.doi.org/10.1016/j.knosys.2012.08.003>.
- Zhai, Z. et al., 2011. Clustering Product Features for Reviewon Mining. In *Proceeding WSDM '11 Proceedings of the fourth ACM international conference*. pp. 347–354.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Lampiran 1 : Perhitungan Precision, Recall, Akurasi LDA

LSTM
DATA
300

		predict	
		1	0
actual	1	70	12
	0	0	0
Akurasi		0,853659	
Precision		0,853659	
Recall		1	
F-measure		0,921053	

		predict	
		1	0
actual	1	0	2
	0	0	80
Akurasi		0,97561	
Precision		0	
Recall		#DIV/0!	
F-measure		#DIV/0!	

		predict	
		1	0
actual	1	0	10
	0	0	72
Akurasi		0,878049	
Precision		0	
Recall		#DIV/0!	
F-measure		#DIV/0!	

DATA
544

		predict	
		1	0
actual	1	115	21
	0	0	0
Akurasi		0,845588	
Precision		0,845588	
Recall		1	
F-measure		0,916335	

		predict	
		1	0
actual	1	0	3
	0	0	133
Akurasi		0,977941	
Precision		0	
Recall		#DIV/0!	
F-measure		#DIV/0!	

		predict	
		1	0
actual	1	0	18
	0	0	118
Akurasi		0,867647	
Precision		0	

Recall	#DIV/0!
F-measure	#DIV/0!

DATA
884

		predict	
		1	0
actual	1	162	12
	0	0	0
Akurasi		0,931034	
Precision		0,931034	
Recall		1	
F-measure		0,964286	

		predict	
		1	0
actual	1	0	9
	0	0	165
Akurasi		0,948276	
Precision		0	
Recall		#DIV/0!	
F-measure		#DIV/0!	

		predict	
		1	0
actual	1	0	124
	0	0	50
Akurasi		0,287356	
Precision		0	
Recall		#DIV/0!	
F-measure		#DIV/0!	

CNN
DATA
300

		predict	
		1	0
actual	1	70	0
	0	12	0
Akurasi		0,853659	
Precision		1	
Recall		0,853659	
F-measure		0,921053	

		predict	
		1	0
actual	1	0	2
	0	0	80
Akurasi		0,97561	
Precision		0	
Recall		#DIV/0!	
F-measure		#DIV/0!	

		predict	
		1	0
actual	1	1	9
	0	0	72
Akurasi		0,890244	
Precision		0,1	

Recall	1
F-measure	0,181818

DATA
544

		predict	
		1	0
actual	1	115	0
	0	20	0
	Akurasi	0,851852	
	Precision	1	
	Recall	0,851852	
	F-measure	0,92	

		predict	
		1	0
actual	1	0	2
	0	0	133
	Akurasi	0,985185	
	Precision	0	
	Recall	#DIV/0!	
	F-measure	#DIV/0!	

		predict	
		1	0
actual	1	2	0
	0	22	96
	Akurasi	0,816667	
	Precision	1	
	Recall	0,083333	
	F-measure	0,153846	

DATA
884

		predict	
		1	0
actual	1	135	27
	0	6	5
	Akurasi	0,809249	
	Precision	0,833333	
	Recall	0,957447	
	F-measure	0,891089	

		predict	
		1	0
actual	1	0	9
	0	0	165
	Akurasi	0,948276	
	Precision	0	
	Recall	#DIV/0!	
	F-measure	#DIV/0!	

		predict	
		1	0
actual	1	19	31
	0	29	95
	Akurasi	0,655172	
	Precision	0,38	
	Recall	0,395833	
	F-measure	0,387755	

Lampiran 3 : Perhitungan Precision, Recall, FM, Akurasi LDA

BUG REPORT

	PATH 474			PATH 773			PATH 1522			WUP 474			WUP 773			WUP 1522		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
ORIGINAL	0,552	0,786	0,648	0,562	0,807	0,663	0,548	0,613	0,578	0,682	0,770	0,724	0,661	0,568	0,611	0,688	0,592	0,636
WORDNET	0,552	0,783	0,647	0,560	0,802	0,660	0,560	0,614	0,586	0,696	0,772	0,732	0,685	0,570	0,622	0,704	0,589	0,641
TF ICF 15%	0,571	0,788	0,662	0,583	0,809	0,677	0,567	0,613	0,589	0,682	0,768	0,723	0,701	0,569	0,628	0,726	0,591	0,652
TF ICF 30%	0,674	0,823	0,741	0	0	0	0	0	0	0,696	0,820	0,753	0	0	0	0	0	0
TF ICF 45%	0,696	0,820	0,753	0,593	0,819	0,688	0,565	0,633	0,597	0,694	0,764	0,727	0,699	0,572	0,629	0,722	0,592	0,651
TF ICF 60%	0,691	0,816	0,748	0,630	0,828	0,716	0,267	0,701	0,386	0,735	0,767	0,751	0,710	0,578	0,637	0,292	0,621	0,397
TF-ICF 75%	0,760	0,832	0,795	0,663	0,878	0,755	0,449	0,674	0,539	0,437	0,797	0,565	0,497	0,573	0,532	0,492	0,606	0,543

FEATURE REQUEST

	PATH 474			PATH 773			PATH 1522			WUP 474			WUP 773			WUP 1522		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
ORIGINAL	0,100	0,167	0,125	0,129	0,200	0,157	0,171	0,217	0,191	0,050	0,333	0,087	0,020	0,250	0,036	0,000	0,000	0,000
WORDNET	0,100	0,125	0,111	0,129	0,148	0,138	0,171	0,167	0,169	0,136	0,333	0,194	0,020	0,167	0,035	0,000	0,000	0,000
TF ICF 15%	0,150	0,136	0,143	0,129	0,108	0,118	0,197	0,049	0,079	0,050	0,200	0,080	0,039	0,200	0,066	0,053	0,023	0,032
TF ICF 30%	0,250	0,263	0,256	0	0	0	0	0	0	0,300	0,261	0,279	0	0	0	0	0	0
TF ICF 45%	0,300	0,261	0,279	0,258	0,195	0,222	0,276	0,061	0,100	0,050	0,100	0,067	0,039	0,182	0,065	0,053	0,022	0,031

TF ICF 60%	0,200	0,500	0,286	0,323	0,062	0,104	0,539	0,074	0,130	0,050	0,167	0,077	0,157	0,098	0,120	0,447	0,052	0,093
TF-ICF 75%	0,500	0,108	0,177	0,516	0,113	0,186	0,461	0,096	0,158	0,700	0,097	0,171	0,431	0,092	0,152	0,250	0,042	0,072

NON INFORMATIVE

	PATH 474			PATH 773			PATH 1522			WUP 474			WUP 773			WUP 1522		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
ORIGINAL	0,516	0,233	0,321	0,555	0,250	0,345	0,486	0,408	0,443	0,379	0,235	0,290	0,310	0,343	0,326	0,359	0,415	0,385
WORDNET	0,516	0,239	0,327	0,542	0,250	0,342	0,463	0,409	0,434	0,361	0,241	0,289	0,292	0,349	0,318	0,333	0,416	0,370
TF ICF 15%	0,505	0,250	0,334	0,516	0,256	0,342	0,303	0,425	0,354	0,358	0,227	0,278	0,264	0,340	0,297	0,188	0,398	0,256
TF ICF 30%	0,537	0,317	0,398	0	0	0	0	0	0	0,484	0,315	0,382	0	0	0	0	0	0
TF ICF 45%	0,484	0,315	0,382	0,516	0,261	0,346	0,315	0,453	0,372	0,326	0,225	0,266	0,278	0,353	0,311	0,190	0,404	0,259
TF ICF 60%	0,495	0,290	0,366	0,387	0,364	0,375	0,504	0,450	0,475	0,305	0,234	0,265	0,188	0,361	0,247	0,371	0,466	0,413
TF-ICF 75%	0,274	0,491	0,351	0,490	0,402	0,442	0,452	0,450	0,451	0,326	0,233	0,272	0,195	0,362	0,254	0,301	0,480	0,370

[Halaman ini sengaja dikosongkan]

BIOGRAFI PENULIS

Alifia Puspaningrum lahir di Rembang, Jawa Tengah pada tanggal 28 Mei 1993. Pendidikan yang telah ditempuh adalah SDN Karanganyar 1 (1998 - 2004), SMPN 2 Sindang (2004 - 2007), SMA Unggulan Da'I An-Nur (2007 - 2010), dan Pendidikan Ilmu Komputer Universitas Pendidikan Indonesia (UPI). Rumpun Mata Kuliah (RMK) yang diambil oleh penulis adalah Rekayasa Perangkat Lunak (RPL). Alat komunikasi yang disediakan oleh penulis adalah surel: alifia.puspaningrum@gmail.com