TUGAS AKHIR – KS184822

PENGELOMPOKAN KATEGORI *TWEET* TERHADAP PENGGUNAAN *E-WALLET* DI INDONESIA MENGGUNAKAN METODE *K-MEANS* DAN *LATENT DIRICHLET ALLOCATION* (LDA)

NUR FIDYAH PERMATASARI
NRP 062115 4000 0087

Dosen Pembimbing
Pratnya Paramitha Oktaviana, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

TUGAS AKHIR – KS184822

PENGELOMPOKAN KATEGORI *TWEET* TERHADAP PENGGUNAAN *E-WALLET* DI INDONESIA MENGGUNAKAN METODE *K-MEANS* DAN *LATENT DIRICHLET ALLOCATION* (LDA)

NUR FIDYAH PERMATASARI
NRP 062115 4000 0087

Dosen Pembimbing
Pratnya Paramitha Oktaviana, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

FINAL PROJECT – KS184822

TWEET CATEGORY CLUSTERING OF *E-WALLET* IN INDONESIA BY USING K-MEANS AND LATENT DIRICHLET ALLOCATION (LDA)

NUR FIDYAH PERMATASARI
NRP 062115 4000 0087

Supervisor
Pratnya Paramitha Oktaviana, S.Si., M.Si.

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

LEMBAR PENGESAHAN

PENGELOMPOKAN KATEGORI *TWEET* TERHADAP PENGGUNAAN *E-WALLET* DI INDONESIA MENGGUNAKAN METODE *K-MEANS* DAN *LATENT DIRICHLET ALLOCATION* (LDA)

TUGAS AKHIR
Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Statistika
pada
Program Studi Sarjana Departemen Statistika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Oleh :
Nur Fidyah Permatasari
NRP. 062115 4000 0087

Disetujui oleh Pembimbing:
Pratnya Paramitha Oktaviana, S.Si., M.Si.
NIP. 1300201405001 ( )

Mengetahui,
Kepala Departemen Statistika

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2019

(Halaman ini sengaja dikosongkan)

PENGELOMPOKAN KATEGORI *TWEET* TERHADAP PENGGUNAAN *E-WALLET* DI INDONESIA MENGGUNAKAN METODE *K-MEANS* DAN *LATENT DIRICHLET ALLOCATION* (LDA)

Nama Mahasiswa : Nur Fidyah Permatasari
NRP : 06211540000087
Departemen : Statistika
Dosen Pembimbing : Pratnya Paramitha Oktaviana, S.Si., M.Si.

Abstrak

Berbagai macam jenis pembayaran elektronik yang ada, salah satu je-nisnya adalah e-wallet. Go-Jek dan Lippo merupakan sebagian dari banyak perusahaan payment gateaway terbesar di Indonesia yang mempunyai sebuah produk e-wallet masing-masing bernama Go-Pay dan OVO. Banyak orang hendak mengajukan pertanyaan, keluhan, atau saran kepada Go-Pay dan OVO melalui Twitter. Guna mempermudah dan mempercepat dalam menanggapi setiap tweet, dilakukan pembentukan kategori tweet yang datanya diperoleh dengan Twitter API. Pada penelitian ini akan mengaplikasikan metode clustering dalam menentukan jenis pertanyaan/keluhan. Metode yang digunakan adalah membanding-kan antara K-Means, Latent Dirichlet Allocation (LDA), K-Means dengan (LDA). Clustering menggunakan K-Means dengan LDA adalah metode terbaik karena menghasilkan nilai silhouette coefficient lebih tinggi. Hasil cluster yang didapatkan untuk e-wallet Go-Pay terdapat 3 yaitu mengenai pembayaran, transaksi saldo, dan juga layanan cashback. Sedangkan cluster untuk e-wallet OVO juga terdapat 3 yaitu berkenaan tentang layanan cashback , e-mail customer service, dan top-up saldo. Selain itu dilakukan pula Social Network Analysis (SNA) yang digunakan untuk menggambarkan struktur komunikasi dan tingkat partisipasi dari setiap pengguna.

Kata Kunci : Clustering, E-Wallet, Go-Pay, K-Means, Latent Dirich-let Allocation, OVO, Social Network Analysis

(Halaman ini sengaja dikosongkan)

TWEET CATEGORY CLUSTERING OF *E-WALLET* USE IN INDONESIA BY USING K-MEANS AND LATENT DIRICHLET ALLOCATION (LDA)

Name : **Nur Fidyah Permatasari**
Student Number : **06211540000087**
Department : **Statistics**
Supervisor : **Pratnya Paramitha Oktaviana, S.Si., M.Si.**

Abstract

Many various types of electronic payments nowadays, one of which is e-wallet. Go-Jek and Lippo are two of the largest payment gateaway companies in Indonesia that have an e-wallet product named Go-Pay and OVO respectively. Many customer wants to ask questions, take complaints, or give suggestions to Go-Pay and OVO via Twitter. In order to make it easier and faster in responding to each tweet, tweet category clustering was formed in which the data was obtained with the Twitter API. In this study will apply the clustering method in determining the type of questions/complaints. The methods are comparing between K-Means, Latent Dirichlet Allocation (LDA) and K-Means with LDA. Clustering using K-Means with LDA is the best method because it produces a higher silhouette coefficient value. The cluster results obtained for Go-Pay e-wallet are 3, payments, balance transactions, and cashback services. Whereas for OVO e-wallet there are also 3 clusters, cashback services, e-mail customer service, and balance top-ups. In addition, Social Network Analysis (SNA) is also applied to describe the communication structure and the level of participation of each user.

Keywords *: **Clustering, E-Wallet, Go-Pay, K-Means, Latent Dirichlet Allocation, OVO, Social Network Analysis***

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Segala puji dan syukur atas kehadirat Allah SWT yang telah melimpahkan segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul "**Pengelompokan Kategori *Tweet* terhadap Penggunaan *E-Wallet* di Indonesia Menggunakan Metode *K-Means* dan *Latent Dirichlet Allocation* (LDA)**" dengan baik dan lancar.

Selama penyusunan laporan, penulis telah menerima bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua, kakak serta adik penulis yang selalu memberikan doa, dukungan, dan motivasi kepada penulis agar tetap berusaha dan sabar dalam menghadapi segala jenis tantangan yang dihadapi dalam proses penyelesaian Tugas Akhir.
2. Ibu Pratnya Paramitha Oktaviana, S.Si., M.Si. selaku dosen pembimbing yang telah meluangkan waktu, mengarahkan, dan membimbing penulis dalam proses penyelesaian Tugas Akhir.
3. Ibu Dr. Dra. Kartika Fithriasari, M.Si. dan Ibu Wiwiek Setya Winahju, M.S. selaku dosen penguji yang telah banyak membantu dan memberikan masukan untuk kesempurnaan Tugas Akhir ini.
4. Bapak Dr.rer pol. Dedy Dwi Prastyo, S.Si, M.Si selaku dosen wali yang telah memberikan arahan dan masukan semenjak penulis mengenyam studi di Departemen Statistika FMKSD ITS hingga penyelesaian laporan Tugas Akhir.
5. Bapak Dr. Suhartono, selaku Kepala Departemen Statistika FMKSD ITS dan Ibu Santi Wulan Purnami, M.Si, Ph. D. selaku Ketua Program Studi Sarjana Departemen Statistika FMKSD ITS yang telah memberikan nasehat dan keyakinan kepada penulis dalam menyelesaikan Tugas Akhir.

6. Seluruh dosen serta karyawan Departemen Statistika ITS yang telah memberikan ilmu, pengalaman, dan semangatnya kepada penulis dan terus mendukung penulis.
7. Teman-teman Vivacious Σ26, Statistika ITS angkatan 2015 yang telah memberikan bantuan dan dukungan kepada penulis selama penyelesaian laporan Tugas Akhir.
8. Semua teman, relasi, dan berbagai pihak yang telah membantu penulis dalam penyelesaian laporan ini.

Penulis menyadari bahwa dalam penulisan dan penyusunan laporan Tugas Akhir ini masih banyak kekurangan dan kelemahan. Oleh karena itu, kritik dan saran membangun akan sangat membantu penulis dalam memperbaikinya di masa yang akan datang. Penulis berharap semoga penelitian Tugas Akhir ini dapat bermanfaat bagi masyarakat dan bagi ilmu pengetahuan.

Surabaya, Juli 2019

Penulis

DAFTAR ISI

Halaman

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

Halaman

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kemajuan ilmu pengetahuan dan teknologi membawa banyak pengaruh terhadap perubahan perilaku dan kebiasaan masyarakat. Hal ini membuat masyarakat membutuhkan sesuatu yang cepat, mudah untuk setiap kegiatan mereka. Kebutuhan ini juga diharapkan masyarakat pada kegiatan ekonomi mereka, salah satunya dalam sistem pembayaran. Sistem pembayaran berbasis teknologi tersebut diadopsi menjadi sistem pembayaran elektronik. Sistem pembayaran ini menawarkan berbagai keuntungan yang saat ini sedang dibutuhkan oleh masyarakat di era globalisasi. Undang-Undang Nomor 23 Tahun 1999 berisi tentang sistem pembayaran mempunyai pengertian yaitu sebuah sistem yang mencakup seperangkat aturan, lembaga, dan mekanisme yang dipergunakan untuk dilakukannya pemindahan dana guna memenuhi kewajiban yang timbul dari suatu kegiatan ekonomi. Sedangkan menurut Daniel (1996), sistem pembayaran adalah suatu jaringan layanan yang memfasilitasi transaksi suatu barang, layanan, dan aset lainnya.

Bank Indonesia (2011) menyatakan bahwa pemakaian uang tunai memiliki kendala dalam hal efisiensi, hal itu dikarenakan adanya pengaruh pengadaan dan pengelolaan (*cash handling*), efisiensi waktu, dan resiko keamanan. Oleh karena itu, Bank Indonesia mengeluarkan kebijakan *Less Cash Society* yang dikeluarkan pada periode tahun 2005-2006. Kebijakan ini bertujuan untuk mengurangi penggunaan instrumen uang tunai yang telah lama diterapkan pada kegiatan transaksi masyarakat. Sistem pembayaran ini pun berkembang menjadi *Electronic Payment System*, suatu sistem pembayaran yang telah dikembangkan oleh perbankan sejalan dengan kemajuan teknologi, salah satunya adalah *Electronic Money* (*E-Money*) yang di-terbitkan oleh Bank Indonesia pertama kali di bulan April 2007. Definisi *e-money* sesuai dengan yang dikeluarkan *Bank for International Settlement* (BIS)

pada bulan Oktober 1996 merupakan produk *stored-value* atau *prepaid* dimana sejumlah nilai uang disimpan dalam suatu media elektronis yang dimiliki seseorang.

Meskipun *e-money* merupakan alat sistem pembayaran yang baru, pertumbuhan pengguna *e-money* tergolong mengalami pertumbuhan pesat setiap tahunnya. Pengguna *e-money* tumbuh signifikan, tercermin dari data volume dan nilai transaksi yang diterbitkan Bank Indonesia. Data yang dikumpulkan Bank Indonesia hingga November 2017 mencatat, nilai transkasi uang elektronik sepanjang tahun 2016 hingga November 2017 mencapai Rp 10,42 triliun. Nilai transaksi itu sudah melampaui nilai transaksi periode yang sama pada tahun 2016 yaitu tercatat Rp 6,31 triliun atau mengalami pertumbuhan menjadi 65 persen. Volume transaksi uang elektronik juga naik 93 persen menjadi 128,51 juta transaksi dari November 2016 yang mencapai 79,22 juta transaksi.Sedangkan untuk jumlah uang elektronik yang beredar pada November 2017 BI mencatat sebesar Rp 113,72 juta atau meningkat 130 persen dengan periode sama tahun sebelumnya yang sebanyak 49,41 juta. Berbeda pada awal penerbitannya, *e-money* saat ini tidak hanya diterbitkan dalam bentuk *chip* yang tertanam pada kartu atau media lainnya *(chip based)*, namun juga telah diterbitkan dalam media lain yaitu suatu media yang saat digunakan untuk bertransaksi akan terkoneksi terlebih dulu dengan *server* penerbit (*server based)* atau lebih dikenal dengan *e-wallet*.

Terdapat beberapa perusahaan *e-wallet* beroperasi di Indonesia salah satunya adalah Go-Pay milik PT Aplikasi Karya Anak Bangsa atau yang lebih dikenal dengan Go-Jek, sebuah perusahaan teknologi Indonesia yang melayani angkutan melalui jasa ojek. Perusahaan ini didirikan pada tahun 2010 di Jakarta oleh Nadiem Makarim (Go-Jek, 2015). Go-Jek tidak ingin berhenti hanya sebagai perusahaan transportasi, namun bertransformasi sebagai sebuah perusahaan *financial technology* (*fintech*) melalui peluncuran *e-wallet* Go-Pay. Go-Pay telah mengantongi izin No. 16/98/DKSP tanggal 17 Juni 2014 dengan tanggal efektif 29 Sep-

tember 2014. Saat ini saldo Go-Pay bisa digunakan untuk berbagai layanan yang ada di aplikasi Go-Jek, seperti membayar biaya Go-Ride, memesan makanan, top up pulsa, dan layanan lainnya. Proses *top up* bisa dilakukan lewat *internet banking*, *mobile banking*, ATM, maupun melalui *driver* Go-Jek. Besarnya peluang dari *e-wallet* juga dilakukan oleh perusahan besar yaitu Lippo. Berada di naungan LippoX, sebuah *e-wallet* diluncurkan pada Maret 2017 bernama OVO. OVO mencoba mengakomodasikan berbagai kebutuhan keuangan tanpa uang tunai dan pembayaran seluler. OVO juga sudah mengantongi izin dari Bank Indonesia dan termasuk dalam Daftar Penyelenggara Uang elektronik yang Telah Memperoleh Izin dari Bank Indonesia Per 21 Januari 2019. Situs resmi Bank Indonesia menyebutkan bahwa nama produk OVO Cash surat dan tanggal izin No. 19/661/DKSP/Srt/B tanggal 7 Agustus 2017 yang memiliki tanggal operasional pada 22 Agustus 2017 (OVO, 2015).

Seiring dengan perkembangan zaman, penggunaan media sosial terus meningkat. Menteri Komunikasi dan Informatika RI, Rudiantara mengatakan bahwa pengguna internet di Indonesia meningkat dua kali lipat dari 2014 hingga 2017 dan saat ini hampir seluruh warga Indonesia menggunakan media sosial. Masyarakat cenderung lebih sering berkomunikasi menggunakan media sosial saat membutuhkan sesuatu karena lebih mudah diakses kapanpun dan dimanapun. Menurut Selamatta S. selaku Direktur Pelayanan Informasi Internasional Ditjen Informasi dan Komunikasi Publik (IKP), situs jejaring sosial yang paling banyak diakses adalah situs jejaring sosial Facebook dan Twitter (Kemenkominfo, 2013). *E-wallet* Go-Pay dan OVO mempunyai masing-masing sebuah akun Twitter yaitu @gopayindonesia untuk Go-Pay dan @ovo_id untuk OVO. Akun tersebut adalah salah satu bentuk layanan pelanggan khusus melalui *online* yang disediakan untuk wadah menanggapi *tweet* tanggapan, pendapat, kritik, saran dan masalah *complaint*. Tanggapan/kritik pada *tweet* tersebut dapat digunakan sebagai data.

Metode statistika yang digunakan untuk analisis data teks dikenal dengan *text mining*. *Text mining* adalah satu cabang dari ilmu *data mining* yang menganalisis suatu data berupa teks. *Text mining* dapat digunakan untuk beberapa proses diantaranya penemuan *rule* baru dengan algoritma pengelompokan, asosiasi, dan *ranking* (Talib, Hanif, Ayesha, dkk., 2016). Dari ketiga fungsi tersebut, *text mining* paling banyak dilakukan pada proses pengelompokan. Terdapat dua jenis metode pengelompokan teks, yaitu *text clustering* dan *text classification*. Perbedaan antara *text clustering* dan *text classification* ada pada penentuan kelompok yang dibentuk. *Text clustering* merupakan proses menemukan sebuah struktur kelompok yang belum terlihat dari sekumpulan dokumen. Sedangkan *text classification* merupakan proses untuk membentuk golongan dari dokumen berdasarkan pada kelas kelompok yang sudah diketahui sebelumnya. Berdasarkan pengertian ini, dapat dinyatakan bahwa proses *clustering* merupakan proses yang tepat dan lebih mudah untuk dilakukan *monitoring*, karena terdapat kelas yang akan terbentuk dalam analisisnya.

Metode *clustering* yang digunakan adalah membandingkan antara *K-Means, Latent Dirichlet Allocation* (LDA), dan *K-Means* dengan LDA. Menurut Zhang dan Fang (2013), metode *K-Means* dapat bekerja secara cepat dengan dataset yang relatif besar dengan berdasar pada jarak *Euclidian*. Sedangkan menurut Campbell, Hindle & Stroulia (2014) metode LDA merupakan metode *topic modeling* yang dapat digunakan dalam melakukan analisis pada dokumen yang berukuran sangat besar. Kedua metode tersebut nantinya akan dilakukan evaluasi dengan membandingkan nilai *silhouette coefficient* pada masing-masing *cluster* yang terbentuk. Penelitian serupa juga dilakukan oleh Indraloka dan Santosa (2017) dengan judul Penerapan *Text Mining* untuk Melakukan *Clustering* Data *Tweet* Shopee Indonesia. Penelitian ini menggunakan metode *K-Means* untuk mengetahui jenis konten *tweet* yang banyak dilakukan *retweet* oleh *followers* Shopee Indonesia. Jannah, Fithriasari, Prastyo, dkk. (2018) pun melakukan penelitian serupa yang berjudul *Text Mining for Identifying and*

Visualizing Topics of Citizen Opinion in Media Centre Surabaya. Penelitian tersebut bertujuan untuk mengidentifikasi topik dari pendapat masyarakat tentang kota Surabaya yang diperoleh dari *Media Center* Surabaya menggunakan metode *K-Means* dengan LDA dan memperoleh hasil *cluster* sebanyak 15. Kemudian Putra & Kusumawardani (2017) melakukan penelitian pemodelan topik dengan metode LDA yang berjudul Analisis Topik Informasi Publik Media Sosial di Surabaya menggunakan Pemodelan *Latent Dirichlet Allocation* (LDA). Hasil yang didapatkan menyatakan bahwa jumlah topik yang terdapat dalam pesan media sosial *Twitter* akun resmi Radio Suara Surabaya FM adalah 4 topik. Sementara Herwanto (2018) melakukan penelitian membandingkan metode gabungan *Ward Hierarchical Clustering* dan LDA dengan dan tanpa pembobotan TF-IDF dalam *Document Clustering* dengan *Latent Dirichlet Allocation* dan *Ward Hierarichal Clustering*. Hasil yang diperloleh menyatakan bahwa dilakukan pembobotan TF-IDF memberikan hasil *cluster* yang lebih baik dengan nilai *silhouette coefficient*.

Oleh karena itu, pada penelitian ini dilakukan penentuan kategori-kategori dari *tweet* yang ditujukan kepada akun Twitter resmi masing-masing perusahaan *e-wallet* Go-Pay dan OVO dengan membandingkan antara metode *K-Means, Latent Dirichlet Allocation* (LDA), dan *K-Means* dengan LDA, dimana parameter yang digunakan untuk membandingkan kinerja kedua metode tersebut adalah *silhouette coefficient*. Kemudian analisis *Social Network Analysis* (SNA) juga dilakukan pada penelitian ini, untuk identifikasi pengguna Twitter yang berpengaruh mengenai layanan *e-wallet* Go-Pay dan OVO. Melalui penelitian ini, diharapkan dapat memberikan masukan kepada perusahaan *e-wallet* terkait kategori dari *tweet* yang ditujukan masyarakat kepada akun Twitter *e-wallet* Go-Pay (@gopayindonesia) dan OVO (@ovo_id) agar dapat mempermudah dan mempercepat dalam menanggapi.

1.2 Rumusan Masalah

Pengguna *e-wallet* dapat memberikan *feedback* secara terbuka, saling berkomentar dalam waktu yang cepat dan tidak ter-

batas terhadap pelayanan dari *e-wallet* Go-Pay dan OVO melalui Twitter resmi Go-Pay (@gopayindonesia) dan OVO (@ovo_id). Jenis komentar tersebut memiliki beragam topik yang kurang jelas. Kondisi ini dapat mengakibatkan perusahaan *e-wallet* melewatkan informasi yang berguna dari sekumpulan topik. Oleh karena itu, pada penelitian ini dilakukan analisis *text mining* dengan membandingkan metode *cluster* yaitu *K-Means, Latent Dirichlet Allocation* (LDA). dan *K-Means* dengan LDA.

Informasi yang didapatkan melalui Twitter tidak dapat menggambarkan struktur komunikasi dan tingkat partisipasi dari setiap pelanggan. Oleh karena itu diperlukan suatu metode yang dapat menilai atau memeriksa pola interaksi pengguna *e-wallet* Go-Pay dan OVO. *Social Network Analysis* (SNA) merupakan salah satu metode untuk menganalisis pola interaksi pengguna *e-wallet* Go-Pay dan OVO.

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut.

1. Mendapatkan karakteristik data *tweet* yang ditujukan kepada akun Twitter resmi masing-masing perusahaan *e-wallet* Go-Pay (@gopayindonesia) dan OVO (@ovo_id).
2. Memperoleh perbandingan hasil analisis *clustering* data *tweet* yang ditujukan kepada akun Twitter resmi masing-masing perusahaan *e-wallet* Go-Pay (@gopayindonesia) dan *e-wallet* OVO (@ovo_id) dengan menggunakan metode *K-Means, Latent Dirichlet Allocation* (LDA), dan *K-Means* dengan LDA.
3. Memvisualisasikan *tweet* menggunakan *wordcloud* yang ditujukan kepada akun Twitter resmi masing-masing perusahaan *e-wallet* Go-Pay (@gopayindonesia) dan OVO (@ovo_id) berdasarkan *cluster* terbaik.
4. Dapat melakukan analisis hasil representasi *graph* yang terbentuk menggunakan *Social Network Analysis* (SNA) untuk mengidentifikasi pengguna Twitter yang berpengaruh mengenai layanan *e-wallet* Go-Pay dan OVO.

1.4 Manfaat

Berdasarkan permasalahan dan tujuan yang telah dipaparkan, manfaat yang diharapkan dari penelitian ini adalah dapat memberikan kategori dari *tweet* yang ditujukan kepada akun Twitter resmi masing-masing perusahaan *e-wallet* Go-Pay pada @gopayindonesia dan OVO pada @ovo_id sehingga dapat mempercepat dan mempermudah perusahaan dalam menanggapi *tweet* yang masuk. Hasil penelitian berupa banyaknya *cluster* yang terbentuk dari *tweet* yang ditujukan kepada *account* Twitter *e-wallet* dan juga visualisasi SNA ini dapat dijadikan tambahan informasi untuk penelitian selanjutnya yang berkaitan dengan *text clustering*. Selain itu, melalui penelitian ini peneliti mampu memahami pengaplikasian *text mining* di kehidupan nyata.

1.5 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini adalah penelitian menggunakan *tweet* yang ditujukan kepada akun Twitter resmi masing-masing perusahaan *e-wallet* Go-Pay pada @gopayindonesia dan OVO pada @ovo_id pada tanggal 10 Februari 2019 hingga 14 Maret 2019. Metode *clustering* yang digunakan adalah *K-Means, Latent Dirichlet Allocation* (LDA), dan *K-Means* dengan LDA. Lalu dilakukan juga analisis *Social Network Analysis* (SNA) menggunakan *software* Gephi.

(Halaman ini sengaja dikosongkan)

BAB II
TINJAUAN PUSTAKA

2.1 *Text Mining*

Text Mining merupakan proses ketika pengguna berinteraksi dengan kumpulan dokumen dari waktu ke waktu dengan menggunakan kumpulan analisis. *Text Mining* berguna untuk mengekstrak informasi dari data melalui identifikasi dan eksplorasi dari pola yang menarik. Dalam kasus *text mining*, sumber datanya adalah kumpulan dokumen dan pola yang menarik ditemukan pada teks yang tidak terstruktur dalam suatu dokumen (Feldman dan Sanger, 2007). *Text Mining* didefinisikan sebagai ekstrasi non trivia dari informasi yang tersembunyi, sebelumnya belum diketahui, dan berguna dari data berupa teks yang sangat banyak (Waegel, 2006). *Text Mining* biasanya berupa proses tentang struktur penginputan teks, mencari pola dari data teks yang telah terstruktur, dan evaluasi final serta interpretasi dari output (Narrayana & Kumar, 2013). *Text Mining* merupakan cara memperoleh informasi dari sekumpulan dokumen yang tidak terstruktur. Terdapat beberapa hal yang dilakukan dengan *text mining* diantaranya adalah *text preprocessing*, *classification*, dan *clustering*.

2.2 *Text Processing*

Text preprocessing adalah sebuah proses yang penting dari NLP (*Natural Language Processing*) karena karakter, kata, dan kalimat yang diidentifikasi pada tahap ini adalah unit dasar yang diteruskan ke semua tahap pemrosesan lebih lanjut. Tahapan praproses dalam *text mining* merupakan tahapan yang penting dalam menggali informasi dari sebuah dokumen. Operasi *text mining* yang efektif didasarkan pada metodologi pemrosesan data yang canggih (Feldman dan Sanger, 2007). Penggunaan *text preprocessing* yang tepat dapat meningkatkan akurasi pada kasus klasifikasi. Menurut Hidayatullah, Fakhri & Makhrif (2016) penggunaan *stopword removal* dapat meningkatkan akurasi terhadap kasus klasifikasi *tweet* berbahasa Indonesia.

Dalam melakukan *text preprocessing* terdapat beberapa tahap yang dapat dilakukan. Tahapan-tahapan *text preprocessing* secara umum adalah *removing symbols, removing numbers, removing ASCII string, punctuation, tokenization, case folding, stemming,* dan *stopword removal*.

1. *Removing symbol, number, ASCII strings, and punctuation,* merupakan proses penghapusan simbol, nomor, dan tanda baca lainnya dalam *tweet*. *Tweet* mengandung banyak sekali simbol dan tanda baca. Simbol dan tanda baca yang dihapus adalah seperti "# $ % & \ ' ( ) * + , - . / : ; < = > ? @ [\ \ ] ^ _ ` { | } ~ (Jannah, Fithriasari, Prastyo, dkk., 2018).
2. *Tokenization*, merupakan proses pemisahan kalimat menjadi kata, frase, symbol, dan elemen lainnya yang memiliki arti atau disebut token. Dalam *tokenization* terdapat metode yang disebut *N-gram*. *N-gram* merupakan potongan n-karakter dari sebuah kalimat (Cavnar, William & Traker, 1994). *N-gram* merupakan metode paling sederhana untuk menetapkan urutan kata atau probabilitas kata yang akan muncul setelah satu kata (Daniel dan James, 2014). *N-gram* biasa digunakan dalam pemrosesan bahasa dan perkataan. Contoh penggunaan *n-gram* adalah misalkan dari kalimat "cashback error poin nol" dapat ditunjukkan dengan *unigram* yaitu "cashback", "error", "poin" dan "nol".
3. *Case folding*, merupakan proses untuk mengubah kata ke dalam format yang sama, dalam hal ini yaitu menjadi format *lowercase* atau *uppercase* (Hidayatullah, Fakhri & Makhrif, 2016). Pada penilitian ini *case folding* akan dilakukan dengan mengubah ke format *lowercase*.
4. *Stemming*, merupakan proses untuk menemukan kata dasar dari sebuah kata (Tala, 2003). Sistem kerja tahap *stemming* ini adalah menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran) (Ariadi & Fithriasari, 2015). *Stemming* digunakan agar suatu kata sesuai dengan kaidah Bahasa Indonesia yang benar. Tanpa melakukan *stemming* akan terdapat banyak kata berbeda yang bermakna

hampir sama. Hal ini dikarenakan dalam Bahasa Indonesia terdapat banyak sisipan kata.

5. *Stopword removal*, merupakan proses menghapus kata-kata umum dan sering muncul tetapi tidak memiliki pengaruh yang signifikan terhadap makna dari sebuah kalimat. Penggunaan *stopword removal* berguna untuk mengurangi korpus atau jumlah kata tanpa mengurangi informasi dalam kalimat. Beberapa contoh *stopword* dalam Bahasa Indonesia adalah penggunaan kata “dan”, “atau”, “yang”, “itu”, dan lainnya

2.2.1 *Confix Striping Stemmer*

Confix Striping Stemmer atau disebut CS merupakan pendekatan untuk melakukan *stemming* terhadap Bahasa Indonesia. Berikut merupakan urutan penggunaan afiks, dengan tanda kurung siku berarti bahwa afiks tersebut optional (Adriani, Asian & Nazief, 2007).

[[[DP+]DP+]DP+] kata dasar [[+DS][+PP][+P]]

dengan DP (*Derivational Prefixes*) merupakan awalan, DS (*Derivational Suffixes*) merupakan akhiran, PP (*Possessive Pronouns*) merupakan kata ganti kepunyaan, dan P (*Particles*) merupakan partikel.

Langkah-langkah dalam melakukan *stemming* dengan CS adalah sebagai berikut (Adriani, Asian & Nazief, 2007).

1. Pada awal pemrosesan dan pada setiap langkah, dilakukan pemeriksaan kata pada kamus kata dasar. Jika kata tersebut ditemukan, maka dianggap sebagai kata dasar dan seluruh proses dihentikan.
2. Menghilangkan *inflectional suffixes* yang dimulai dari *inflectional particle* (’-kah’, ’-lah’, ’-tah’, ’-pun’) dan dilanjutkan menghilangkan *possessive pronoun* (’-ku’, ’-mu’, ’-nya’). Contohnya kata “bajumulah” akan dipotong menjadi “bajumu” dan kemudian ”baju”, dimana kata ini sudah merupakan kata dasar sehingga proses berhenti.
3. Menghilangkan *derivational suffixes* (’-i’, ’-kan’, ’-an’). Contohnya kata “membelikan” akan dipotong menjadi “mem-

beli", namun karena kata ini bukanlah kata dasar maka proses dilanjutkan ke langkah selanjutnya.

4. Menghilangkan *derivational prefixes* ('be-', 'di-', 'ke-', 'se-', 'me-', 'te-', 'pe-').

a. Proses berhenti jika:
   — Awalan yang teridentifikasi berpasangan dengan akhiran terlarang yang telah dihilangkan pada langkah 3.
   — Awalan yang dideteksi saat ini sama dengan awalan yang telah dihilangkan sebelumnya.
   — Tiga awalan telah dihilangkan.

b. Identifikasi tipe awalan kemudian hilangkan. Awalan terdiri dari dua tipe berikut ini.
   — Standar ('di-', 'ke-', 'se-') dapat dihilangkan langsung dari kata.
   — Kompleks ('be-', 'te-', 'me-', 'pe-') dapat bermorfologi sesuai kata dasar yang mengikutinya (dapat mengubah bentuk asli kata dasar).

c. Mencari kata yang telah dihilangkan awalannya dalam kamus kata dasar. Apabila pencarian tidak ditemukan maka langkah 4 diulang kembali, sedangkan apabila ditemukan maka keseluruhan proses dihentikan.

5. Apabila hingga langkah 4 kata dasar masih belum ditemukan, maka dilakukan proses *recoding*, yaitu menambah atau mengganti huruf awal dari kata yang terpenggal pada proses *stemming*. Contohnya kata "menangkap" dihilangkan awalan "me" sehingga tersisa "nangkap". Kata "nangkap" bukanlah kata dasar yang valid sehingga dilakukan *recoding* menjadi kata "tangkap".
6. Apabila semua langkah tidak berhasil, maka *input* kata dianggap sebagai kata dasar dan algoritma akan mengembalikan kata seperti semula.

2.3 *Term Weighting*

Proses pembobotan pada setiap kata atau *term-weightening* merupakan metode yang digunakan untuk mendapatkan nilai frekuensi dari masing-masing kata di dalam suatu dokumen sehing-

ga diperoleh perbandingan antara kata satu dengan lainnya karena setiap kata memiliki tingkat kepentingan yang berbeda-beda (Abualigah, Khader & Betar, 2016). Salah satu metode yang sering digunakan untuk menghitung bobot dari suatu *term* atau kata di dalam dokumen adalah *Term Frequency-Inverse Document Frequency* (TF-IDF).

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan pengukuran yang digunakan untuk mengukur seberapa penting suatu kata terhadap kumpulan dokumen. *Term Frequency* (TF) merupakan pendekatan paling sederhana untuk menghitung bobot suatu *term* dimana bobot sama dengan jumlah kemunculan sebuah *term* dalam suatu dokumen. Ketika hanya menggunakan *term frequency* akan terdapat permasalahan yaitu suatu *term* akan memiliki tingkat kepentingan yang sama. Misalkan terdapat *term* yang muncul dalam semua dokumen dengan *term* yang hanya muncul dalam beberapa dokumen. *Term Frequency* (TF) berfungsi dalam meringkas kemunculan sebuah kata pada suatu dokumen. Sedangkan *Inverse Document Frequency* (IDF) berfungsi menghitung frekuensi kemunculan sebuah kata pada seluruh kumpulan dokumen (Jannah, Fithriasari, Prastyo, dkk., 2018).

Tahapan dalam melakukan pembobotan dengan metode TF-IDF antara lain sebagai berikut.

1. *Term Weighting*, yaitu suatu proses untuk men-*generate* sebuah nilai pada setiap *term* dengan cara menghitung frekuensi kemunculan *term* dalam dokumen (*d*).
2. *Document Frequency*, yaitu suatu proses untuk menghitung banyaknya dokumen yang mengandung *term* ke *t*.
3. *Inverse Document Frequency*, yaitu suatu proses untuk menghitung nilai *inverse* dari *document frequency*. Rumus yang digunakan adalah sebagai berikut.

$$idf_t = \log_{10} \frac{N}{df_t} \tag{2.2}$$

dimana:

idf_t = *inverse document frequency* pada *term* ke *t*

N = jumlah keseluruhan dokumen
df_t = nilai dari *document frequency* pada *term* ke *t*

4. *Term Frequency-Inverse Document Frequency* (TF-IDF), yaitu proses untuk mendapatkan nilai skor setiap kata terhadap dokumen. Digunakan rumus sebagai berikut.

$$W_{t,d} = Wtf_{t,d} \times idf_t \quad (2.3)$$

dimana:
$W_{t,d}$ = TF-IDF pada *term* ke *t*, dokumen ke *d*
$Wtf_{t,d}$ = *log-frequency weighting* pada *term* ke *t*, dokumen ke *d*
idf_t = *inverse document frequency* pada *term* ke *t*

5. Menghitung nilai skor akhir setiap dokumen dengan menggunakan rumus:

$$Ws_j = \sum_{i=1}^{Nterm} Wtd_{i,j} \quad (2.4)$$

dimana:
Ws_j = skor dari dokumen ke *j*
Nterm = jumlah banyaknya *term*
$Wtd_{,j}$ = nilai TF-IDF pada *term* ke *i*, dokumen ke *j*

Setelah memperoleh bobot dari masing-masing dokumen maka selanjutnya data akan digunakan untuk melakukan *text clustering* dengan variabel berupa bobot kata-kata yang telah diperoleh dari perhitungan TF-IDF.

2.4 *Text Clustering*

Text clustering atau analisis *cluster* merupakan sebuah *unsupervised process* yang digunakan untuk mengelompokkan beberapa data kelompok berupa teks dengan menerapkan algoritma *clustering* yang berbeda-beda. Data yang memiliki jenis dan pola yang sama dapat dikelompokkan menjadi satu *cluster* agar dapat memperoleh hasil yang efektif (Talib, Hanif, Ayesha, dkk., 2016). Terdapat dua jenis metode *clustering* yang umum digunakan, yaitu *Hierarchical Clustering* dan *Partitioned Clustering*. Salah satu contoh metode *Hierarchical Clustering* adalah *Single Linkage Clustering* sedangkan untuk salah satu contoh untuk metode *Partitioned Clustering* adalah *K-Means* (Alfina & Barakbah, 2012).

2.4.1 *K-Means*

K-Means merupakan metode pengelompokan yang dilakukan dengan mengelompokkan data yang memiliki nilai *centroid* atau rata-rata terdekat dalam satu cluster (MacQueen,1967). *K-Means* diperuntukan ketika semua variabel bersifat kuantitatif dan jarak kuadrat euclidean digunakan untuk mengukur ketidaksamaan antar objek (Johnson dan Wichern, 2007). Rumus jarak kuadrat euclidean dari *p* ke *q* adalah sebagai berikut.

$$d^2(p,q) = \sum_{k=1}^{n} (q_{ik} - p_{jk})^2 \qquad (2.5)$$

dimana:

$d^2(p,q)$ = jarak kuadrat euclidean
n = dimensi
p_{ik} = $(p_{i1}, p_{i2}, \ldots, p_{in})$
q_{jk} = $(q_{j1}, q_{j2}, \ldots, q_{jn})$

Algoritma yang digunakan analisis *clustering* dengan metode *K-Means* adalah sebagai berikut (Srivastava & Sahami, 2009).

1. Menentukan jumlah *cluster* yang ingin dibentuk.
2. Menentukan nilai *centroid.* Dalam menentukan nilai *centroid* untuk awal iterasi, nilai awal *centroid* dilakukan secara acak. Sedangkan jika menentukan nilai *centroid* yang merupakan tahap dari iterasi, maka digunakan rumus sebagai berikut.

$$\bar{v}_{ij} = \frac{1}{N_i} \sum_{k=0}^{N_i} x_{kj} \qquad (2.6)$$

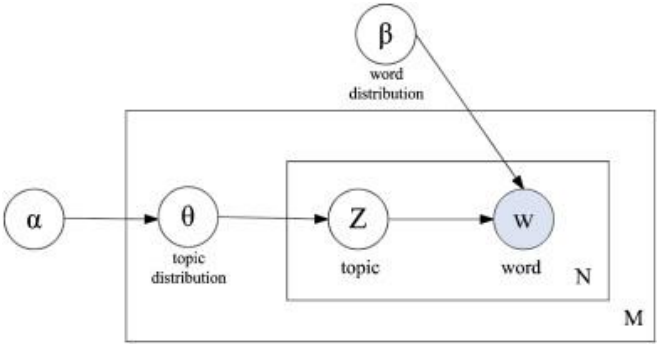
   dengan:
   $\bar{v}_{ij}$ = rata-rata *cluster* ke-*i* untuk variabel ke-*j*
   N_i = jumlah data yang menjadi anggota *cluster* ke-*i*
   i = indeks dari *cluster*
   j = indeks dari variabel
   x_{kj} = nilai data ke-*k* yang ada di dalam *cluster* tersebut untuk variabel ke-*j*
3. Menghitung jarak setiap data ke masing-masing *centroid* menggunakan rumus korelasi antar dua objek (*Euclidean Distance*).

4. Mengelompokkan setiap data berdasarkan jarak terdekat antara data dengan *centroid*. Nilai yang diperoleh dalam keanggotaan data pada *distance* matriks adalah 0 atau 1, dimana nilai 0 untuk data yang dialokasikan ke *cluster* yang lain.

Kembali ke tahap 2 dengan melakukan perulangan hingga nilai *centroid* yang dihasilkan tetap dan anggota *cluster* tidak berpindah ke *cluster* yang lain.

2.4.2 *Latent Dirichlet Allocation* (LDA)

Latent Dirichlet Allocation (LDA) merupakan metode *topic modeling* dan topik analisis yang paling populer saat ini. LDA muncul sebagai salah satu metode yang dipilih dalam melakukan analisis pada dokumen yang berukuran sangat besar. LDA dapat digunakan untuk meringkas, melakukan *clustering*, menghubungkan maupun memproses data yang sangat besar karena LDA menghasilkan daftar topik yang diberi bobot untuk masing-masing dokumen (Campbell, Hindle & Stroulia, 2014). Adapun distribusi yang digunakan untuk mendapatkan distribusi topik per dokumen disebut distribusi Dirichlet, kemudian dalam proses generatif untuk LDA, hasil dari Dirichlet digunakan untuk mengalokasikan kata-kata pada dokumen untuk topik yang berbeda. Dalam LDA, dokumen-dokumen merupakan objek yang dapat diamati, sedangkan topik, distribusi topik per dokumen, penggolongan setiap kata pada topik per dokumen merupakan struktur tersembunyi, maka dari itu, algoritma ini dinamakan *Latent Dirichlet Allocation* (LDA) (Blei, 2012). LDA merupakan model probabilistik generatif dari kumpulan tulisan yang disebut *corpus*. Ide dasar yang diusulkan metode LDA adalah setiap dokumen direpresentasikan sebagai campuran acak atas topik yang tersembunyi, yang mana setiap topik memiliki karakter yang ditentukan berdasarkan distribusi kata-kata yang terdapat di dalamnya. Blei merepresentasikan metode LDA sebagai model probabilistik secara visual seperti pada Gambar 2.1.



Gambar 2.1 Visualisasi Metode LDA sebagai Model Probabilistik

Sesuai visualisasi model di atas, terdapat tiga tingkatan pada LDA Modeling. Parameter α dan β merupakan parameter distribusi topik yang berada pada tingkatan *corpus*, yaitu kumpulan dari M dokumen. Parameter α digunakan dalam menentukan distribusi topik dalam dokumen, semakin besar nilai α dalam suatu dokumen, menandakan campuran topik yang dibahas dalam dokumen semakin banyak. Parameter β digunakan untuk menentukan distribusi kata dalam topik. Semakin tinggi nilai β, maka semakin banyak kata-kata yang ada di dalam topik, sedangkan semakin kecil nilai β, maka semakin sedikit kata-kata yang ada di dalam topik sehingga topik tersebut mengandung kata-kata yang lebih spesifik. Variabel θ_M adalah variabel yang berada di tingkat dokumen *(M)*. Variabel θ merepresentasikan distribusi topik untuk dokumen tertentu. Semakin tinggi nilai θ, maka semakin banyak topik yang ada di dalam dokumen, sedangkan semakin kecil nilai θ, maka dapat dikatakan dokumen tersebut semakin spesifik pada topik tertentu. Variabel Z_N dan W_N adalah variabel tingkat kata *(N)*. Variabel Z dan merepresentasikan topik dari kata tertentu pada sebuah dokumen sedangkan variabel W merepresentasikan kata yang berkaitan dengan topik tertentu yang terdapat dalam dokumen (Blei, 2012).

LDA mengolah kata-kata dalam proses dua tahap: probabilitas kata dalam topik dan probabilitas topik dalam dokumen.

Secara lebih formal, probabilitas kata dalam dokumen dihitung sebagai berikut:

$$p(w_i \mid m) = \sum_{j=1}^{K} p(w_i \mid z_j)\, p(z_j \mid m) \qquad (2.7)$$

LDA mengasumsikan proses generatif berikut untuk *corpus* *M*:

1. Untuk setiap topik $k \in \{1, 2, \ldots, K\}$, sampel distribusi kata $w_k \sim Dirichlet\ (\beta)$
2. Untuk setiap dokumen $m \in \{1, 2, \ldots, M\}$, sampel distribusi topik $\theta_d \sim Dirichlet\ (\alpha)$

Secara umum, LDA bekerja dengan masukan dokumen-dokumen individual dan beberapa parameter, untuk menghasilkan luaran berupa model yang terdiri dari bobot yang dapat di-normalisasi sesuai probabilitas. Probabilitas ini mengacu pada dua jenis, yaitu jenis (a) probabilitas bahwa suatu dokumen spesifik tertentu menghasilkan topik yang spesifik pula dan jenis (b) probabilitas bahwa topik spesifik tertentu menghasilkan kata-kata spesifik dari sebuah kumpulan kosakata. Probabilitas jenis (a), dokumen yang sudah diberi label dengan daftar topik seringkali dilanjutkan hingga menghasilkan probabilitas jenis (b), yang menghasilkan kata-kata spesifik tertentu (Campbell, Hindle & Stroulia, 2014).

2.5 *Silhouette Coefficient*

Silhouette coefficient digunakan untuk mengukur persamaan yang terjadi pada suatu objek dengan membandingkan antara *cluster*nya tertentu dengan *cluster* lainnya. Hal yang perlu diketahui ketika menentukan nilai *silhouette* adalah hasil dari partisi atau *clustering result* dan pengelompokan semua kedekatan antar objek (Jannah, Fithriasari, Prastyo, dkk., 2018). Langkah-langkah yang dilakukan dalam menghitung nilai *silhouette* a-dalah sebagai berikut (Han, Kamber & Pei, 2012).

1. Menghitung rata-rata jarak dari suatu dokumen misalkan *i* dengan semua dokumen lain yang berada dalam satu *cluster*.

$$a(i) = \frac{1}{|A|-1} \sum_{j=1}^{i} j \in A_{j \neq i} d(i, j) \quad (2.8)$$

dimana:

j = dokumen lain dalam satu *cluster A*

d(i,j) = jarak antara dokumen *i* dengan *j*

2. Menghitung rata-rata jarak dari dokuman *i* tersebut dengan semua dokumen di *cluster* lain kemudian diambil nilai terkecilnya.

$$d(i,C) = \frac{1}{|A|-1} \sum_{j=1}^{i} j \in C, d(i, j) \quad (2.9)$$

dimana:

d(i,C) = jarak rata-rata dokumen *i* dengan semua objek pada *cluster* lain C dimana $A \neq C$

$$b(i) = \min C \neq A, d(i,C) \quad (2.10)$$

dimana:

b(i) = nilai terkecil dari jarak rata-rata dokumen *i* dengan semua objek pada *cluster* lain *C* dimana $A \neq C$

3. Diperoleh nilai *silhouette* sesuai dengan formula:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (2.11)$$

2.6 *Social Network Analysis* (SNA)

SNA memiliki beberapa definisi, diantaranya: Nooy, Mrvar, dan Batagelj (2005) mendefinisikan bahwa *Social Network Analysis* adalah proses pemetaan dan pengukuran relasi antara orang ke orang, sedangkan Freeman (1979) mendefinisikan sebagai teknik yang fokus mempelajari pola interaksi pada manusia yang tidak terlihat secara eksplisit. Scott (1992) mendefinisikan sebagai sekumpulan metode untuk melakukan investigasi aspek relasi pada struktur sosial. Berdasarkan ketiga definisi tersebut, secara garis besar memiliki kesamaan makna, yaitu mengarah pada proses analisis jaringan sosial berkaitan dengan bentuk struktur dan pola interaksi entitas di dalamnya.

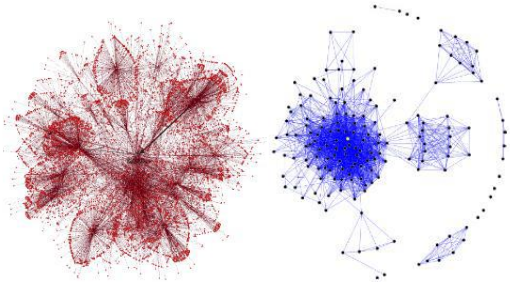
Sebagai contoh pada kasus jejaring sosial *online* di Twitter, yang lebih banyak dianalisis adalah interaksi antar *user* Twitter, dalam hal ini pola interaksi akan dapat menentukan *user* mana yang paling berpengaruh dalam suatu lingkup grup tertentu. SNA dapat memetakan relasi antar orang, organisasi, topik, lokasi, dan intensitas informasi lainnya. *node* di dalam jaringan menggambarkan orang, organisasi, atau entitas informasi. Garis sambungan antar titik menggambarkan relasi antar titik. SNA di dalam teori jaringan terdiri dari *node* dan *edge* (juga disebut relasi, link, atau koneksi). *Node* adalah seorang individu dalam jaringan, dan *edge* adalah hubungan antara *node*. Ada 2 macam graf yaitu *Graf Undirected* dan *Graf Directed*. *Graf undirected* adalah graf yang hubungannya tidak mempunyai orientasi arah. Pada *graf undirected*, nilai antar *node* yang dihubungkan oleh *edge* tidak diperhatikan, yang penting saling berhubungan/berkoneksi maka memiliki nilai. *Graf directed* adalah graf yang setiap hubungan diberikan orientasi arah, dimana edgenya diperhatikan. Contoh dari *Graf Undirected* dan *Graf Directed* dapat ditunjukkan pada Gambar 2.4 dan Gambar 2.5.

Gambar 2.2 Gambaran *Graf Undirected* (Sumber: Cheliotis, 2010)

Gambar 2.3 Gambaran *Graf Directed* (Sumber: Cheliotis, 2010)

SNA dalam Twitter menampilkan peta relasi antar actor di sosial media. *Retweet* menandakan tentang persetujuan (*agreement*), sedangkan *mention* menandakan pertanyaan/diskusi (*discussion*). Ukuran (*metric*) yang digunakan dalam penentuan aktor

dalam penelitian ini adalah *degree centrality*, *betweenness centrality*, dan *closeness centrality*. Bentuk visualisasi dari SNA dapat dilihat dalam Gambar 2.6.



Gambar 2.4 Visualisasi Social Network Analysis
(Sumber: Wasserman & Faust, 1994)

2.6.1 *Degree Centrality*

Degree centrality menghitung jumlah interaksi yang dimiliki oleh sebuah node. Untuk menghitung nilai *degree centrality* milik *node* ke-*i* dapat dilakukan dengan menggunakan rumus sebagai berikut (Wasserman & Faust, 1994).

$$C_D(n_i) = d(n_i) \tag{2.13}$$

dimana:

$d(n_i)$ = banyaknya interaksi yang dimiliki oleh *node* ke-*i* dengan *node* lain di dalam *network*

2.6.2 *Closeness Centrality*

Closeness centrality menghitung jarak rata-rata antara suatu *node* dengan seluruh *node* lain di dalam jaringan atau dalam kata lain mengukur kedekatan sebuah *node* dengan *node* lain. Rumus yang digunakan untuk menghitung kedekatan sebuah *node* dengan *node* lain atau *closeness centrality* adalah sebagai berikut (Wasserman & Faust, 1994).

$$C_c(n_i) = \frac{1}{\sum_{j=1}^{g} d(n_i, n_j)} \tag{2.14}$$

dimana:

$d(n_i, n_j)$ = jumlah jalur terpendek yang menghubungkan *node* ke-*i* dan *node* ke-*j*

2.6.3 *Betweeness Centrality*

Betweeness centrality menghitung seberapa sering sebuah *node* dilewati oleh *node* lain untuk menuju ke sebuah *node* tertentu di dalam jaringan. Nilai ini berfungsi untuk menentukan peran aktor yang menjadi jembatan penghubung interaksi di dalam *network*. Berikut adalah rumus untuk menghitung nilai *betweeness centrality* dari sebuah *node* (Wasserman & Faust, 1994).

$$C_B(n_i) = \sum_{j<k} \frac{g_{jk}(n_i)}{g_{jk}} \tag{2.15}$$

dimana:

$g_{j,k}(n_i)$ = jumlah jalur terpendek dari *node* ke-*j* ke *node* ke-*k* yang melewati *node* ke-*i*.

$g_{j,k}$ = jumlah jalur terpendek dari *node* ke-*j* ke *node* ke-*k*

2.7 *Wordcloud*

Wordcloud merupakan teknik visualisasi yang dilakukan secara sederhana yang bertujuan untuk memberikan informasi terhadap suatu kata yang sering mucul di dalam sebuah dokumen. Secara teknis, *wordcloud* memvisualisasikan kata yang memiliki frekuensi kemunculan paling banyak dalam sebuah dokumen. Dengan menggunakan *word cloud*, gambaran frekuensi kata-kata dapat ditampilkan dalam bentuk yang menarik namun tetap informatif. Kata tersebut ditunjukkan melalui ukuran huruf maupun warna yang menarik perhatian pembaca saat pertama kali melihat bentuk *wordcloud* yang disajikan. Semakin besar ukuran kata dan ketebalan kata yang ditampilkan, maka semakin sering pula kata tersebut muncul di dalam dokumen (Castella, Quim, Sutton, dkk., 2014). *Wordcloud* ini bukan sesuatu yang baru, tetapi penggunaannya masih bisa membuat pembaca tertarik jika *wordcloud* dibuat sekreatif mungkin. Berikut merupakan contoh visualisasi dokumen teks dengan *wordcloud*.

Gambar 2.5 Visualisasi Data dengan *Wordcloud*

2.8 *E-Wallet*

Electronic Wallet merupakan sistem pembayaran non-tunai atau salah satu jenis pembayaran yang disebut *e-money* yang tidak menggunakan media seperti kartu dan melakukan transaksi melalui kanal elektronik. Berbeda dengan kartu debit atau kartu kredit, transaksi menggunakan *e-wallet* tidak secara langsung melainkan melalui pihak ketiga atau intermediari (Amoroso & Magnier-watanabe, 2012). Terdapat beberapa perusahaan *e-wallet* di Indonesia, pada penelitian ini perusahaan *e-wallet* yang terpilih adalah Go-Pay dan OVO.

Go-Pay adalah uang elektronik atau *e-wallet* berupa saldo Go-Jek dan dapat digunakan untuk membayar berbagai layanan Go-Jek dan *merchant* lain. Sebelumnya, Go-Pay bernama Go-Ojek Credit. Go-Pay sudah mengantongi izin No. 16/98/DKSP tanggal 17 Juni 2014 dengan tangga efektif 29 September 2014. Go-Pay sendiri berada di naungan PT Dompet Anak Bangsa. Untuk menggunakan layanan Go-Pay perlu dipastikan bahwa saldo di dalam Go-Pay Go-Jek mencukupi untuk melakukan pembayaran, namun jika saldo tidak mencukupi, Go-Jek menyediakan layanan pembayaran parsial, dimana pelanggan bisa membayar dengan saldo Go-Pay, lalu sisanya bisa dibayarkan dengan uang tunai. Saat ini Go-Pay sudah terintegrasi dengan bank-bank besar di Indonesia demi kemudahan untuk melakukan isi saldo ke dalam Go-Pay. Beberapa bank besar yang menjadi mitra

Go-Jek dalam layanan Go-Pay adalah BCA, Bank Mandiri, Bank BRI, serta pengisian Saldo Via ATM Bersama dan PRIMA.

Gambar 2.6 Logo Go-Pay (Sumber: www.go-jek.com)

Pada Maret 2017 Lippo melalui LippoX mengeluarkan sebuah *e-wallet* yang bernama OVO. OVO merupakan aplikasi smart yang memberikan kesempatan lebih besar mengumpulkan poin di banyak tempat. OVO biasa digunakan untuk bertransaksi di semua *merchant* bertanda *OVO Accepted Here* dan mengumpulkan serta menggunakan *OVO Points* di *merchant* ber-tanda *OVO Zone*. OVO juga sudah mengantongi izin dari Bank Indonesia dan termasuk dalam Daftar Penyelenggara Uang elek-tronik yang Telah Memperoleh Izin dari Bank Indonesia Per 21 Januari 2019. Situs resmi Bank Indonesia menyebutkan bahwa nama produk OVO Cash tanggal izin No. 19/661/DKSP/Srt/B tanggal 7 Agustus 2017 yang memiliki tanggal operasionalnya pada 22 Agustus 2017. OVO sendiri berada pada naungan PT Visionet Internasional. OVO menawarkan beberapa keuntungan bagi penggunanya seperti, promo yang banyak, jumlah *merchant* yang banyak, serta pembayaran yang cepat.

Gambar 2.7 Logo OVO (Sumber: www.ovo.id)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Sumber data yang digunakan dalam penelitian ini adalah sumber data primer yang diambil dari kumpulan *tweet* pengguna Twitter di Indonesia. Akun Twitter yang digunakan dalam analisis kali ini adalah akun Twitter resmi dari *e-wallet* Go-Pay (@gopayindonesia) *e-wallet* dan OVO (@ovo_id). Data tersebut diambil dari tanggal 10 Februari 2019 hingga 14 Maret 2019 dengan menggunakan Twitter API (*Application Programming Interface*). Variabel yang digunakan merupakan kata-kata yang diperoleh dari keseluruhan *tweet* yaitu frekuensi kata *j* yang muncul pada *tweet* ke-*i* pada Tabel 3.1. Frekuensi kata berskala rasio.

3.2 Struktur Data

Struktur data yang digunakan dalam penelitian ini diberikan pada Tabel 3.1.

Tabel 3.1 Struktur Data

Tweet ke	**Kata 1**	**Kata 2**	**….**	**Kata ke-*m***
1	a_{11}	a_{12}	…	a_{1m}
2	a_{21}	a_{22}	…	a_{2m}
3	a_{31}	a_{32}	…	a_{3m}
…	…	…	…	…
n	a_{n1}	a_{n2}	..	a_{nm}

dimana:

a_{ij} = banyak kata ke-*j* muncul pada tweet ke-*i*, $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$.

m = banyak kata

3.3 Langkah Analisis

Langkah analisis yang akan dilakukan pada penelitian ini yaitu sebagai berikut.

1. Mengumpulkan data, data yang digunakan adalah data Twitter terhadap *e-wallet* Go-Pay dan *e-wallet* OVO melalui pencarian

"@gopayindonesia" dan "@ovo_id". Laporan berbentuk data teks yang tidak terstuktur sehingga diperlukan pembersihan terlebih dahulu.

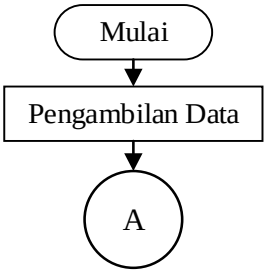
2. *Text preprocessing*, diperlukan dalam membersihkan *noise* sebelum dilakukan analisis.. Adapun langkah-langkah yang dilakukan dalam *text prepocessing* secara umum adalah sebagai berikut.
   a. *Data cleaning*, pembersihan data yang dilakukan pada penelitian ini terdiri dari beberapa tahap diantaranya yaitu sebagai berikut.
      - Melakukan *case folding*, yaitu mengubah semua teks menjadi huruf kecil sehingga data teks memiliki bentuk yang sama yakni non kapital.
      - Menghilangkan angka, semua angka yang terdapat pada *tweet* akan dihapuskan.
      - Menghilangkan *punctuation*, yaitu menghilangkan semua tanda baca seperti ! @ # $ % ^ & * ( ) { } [ ] \ | : " < > ? ; ' , . / `~.
      - *Stemming*, dilakukan dengan mencari kata dasar dari setiap kata yang diperoleh pada *tweet*. *Stemming* dilakukan dengan bantuan *library* Sastrawi pada Python.
      - *Stopword removal*, yaitu menghapus kata-kata yang tidak relevan seperti "dan, ini, itu, dari, atau" dan kata sejenis lainnya.
      - *Tokenizing*, yaitu untuk memecah kalimat aduan dan pertanyaan dalam *tweet* menjadi kata per kata.
   b. Melakukan pembobotan dengan metode TF-IDF, yaitu proses yang digunakan untuk mengetahui jumlah keberadaan suatu *term* di dalam dokumen. *Term* atau kata tersebut kemudian akan diberi bobot berupa *inverse* berdasarkan jumlah kemunculannya agar dapat mengurangi besarnya bilangan frekuensi banyaknya kata tersebut muncul. Melakukan eksplorasi data berupa histogram, data yang telah dibersihkan kemudian dilakukan eksplorasi untuk membe-

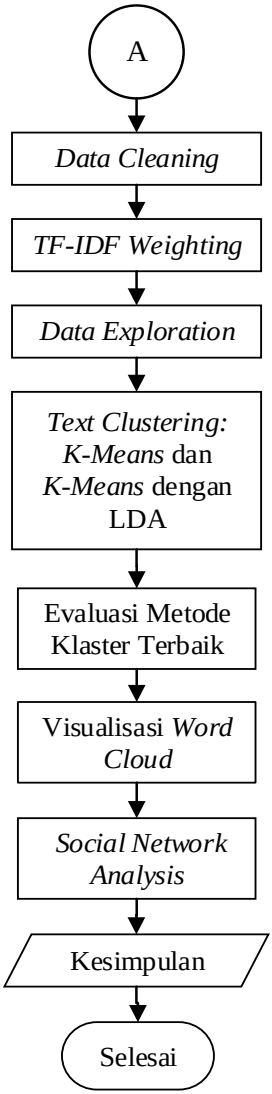
rikan gambaran data *Twitter* secara umum dengan menggunakan pendekatan statistika deskriptif.

3. Melakukan analisis *clustering*.
   a. Analisis *cluster* dengan menggunakan *K-Means*, yaitu menentukan jumlah *cluster* optimum *tweet* dengan memilih jarak antara *centroid* dengan *cluster* yang paling dekat.
   b. Analisis model topik dengan menggunakan *K-Means* dengan LDA, yaitu menentukan jumlah *cluster* optimum berdasarkan data *tweet* dengan menghitung probabilitas kata yang muncul terhadap topik dan probabilitas dokumen terhadap topik.
4. Melakukan perbandingan kedua metode, yaitu menentukan jumlah *cluster* terbaik berdasarkan hasil yang diperoleh pada analisis *cluster* dengan metode *K-Means* dan *K-Means* dengan LDA. Ukuran yang digunakan dalam penentuan jumlah *cluster* terbaik yakni berdasarkan evaluasi nilai *silhouette coefficient*.
5. Melakukan visualisasi hasil dengan menggunakan *wordcloud* pada masing-masing *cluster* yang terbentuk untuk mengetahui kategori apa saja yang terdapat dalam *tweet*.
6. Melakukan *Social Network Analysis.*
7. Interpretasi dan menarik kesimpulan

3.4 Diagram Alir Penelitian

Berikut merupakan diagram alir yang dilakukan pada penelitian.





Gambar 3.1 Diagram Alir Penelitian

BAB IV
ANALISIS DAN PEMBAHASAN

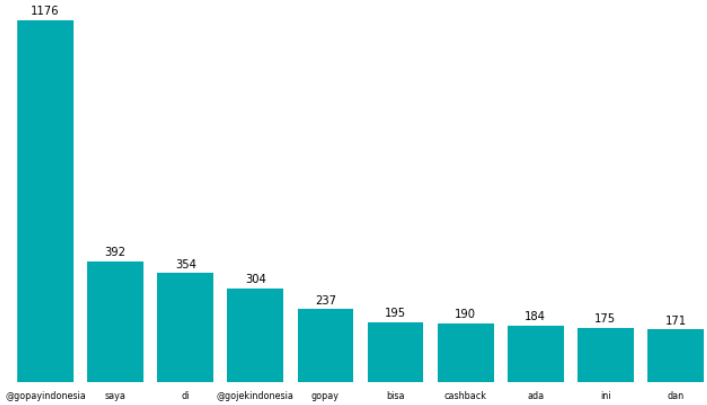
Pada bab analisis dan pembahasan ini akan dibahas mengenai gambaran secara umum karakteristik data *tweet* pengguna akun Twitter terhadap akun *customer service e-wallet* yaitu Go-Pay pada @gopayindonesia dan OVO pada @ovo_id kemudian menetukan *cluster* yang dapat terbentuk. Dalam menentukan hasil *cluster* optimum akan ditentukan dengan membandingkan kinerja dari metode *K-Means, Latent Dirichlet Allocation* (LDA), dan *K-Means* dengan LDA. Selain itu, digunakan pula *Social Network Analysis* (SNA) yang bertujuan untuk menggambarkan struktur komunikasi dan tingkat partisipasi dari setiap pengguna akun Twitter terhadap akun *customer service e-wallet*.

4.1 Karakteristik Data *Tweet E-Wallet* Go-Pay dan OVO

Hasil *crawling* data *tweet* dengan menggunakan Twitter *Application Programming Interface* (API) memperoleh 1182 *tweet* dan 3244 *tweet* yang ditujukan terhadap masing-masing akun layanan *customer care* Go-Pay di akun Twitter @gopayindonesia dan OVO pada @ovo_id. Dari data *tweet* tersebut akan dilihat karakteristik data dan dilakukan *preprocessing* sebelum melakukan analisis *clustering*. Karakteristik data dilihat sebelum dan sesudah *preprocessing* untuk mengetahui apa saja informasi yang terkandung dalam *tweet*.

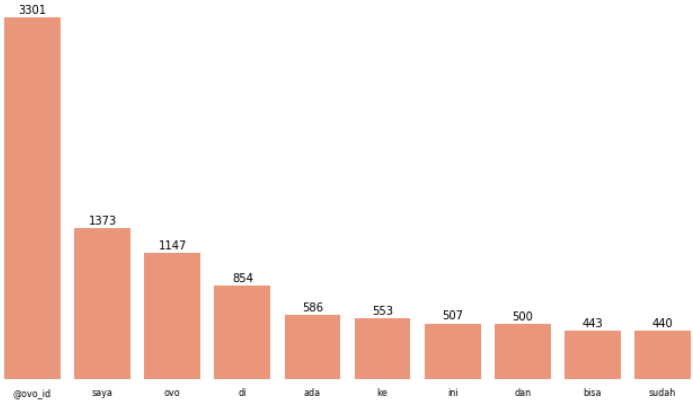
4.1.1 Karakteristik Data *Tweet E-Wallet* Go-Pay dan OVO sebelum *Preprocessing*

Karakteristik data dapat diketahui dengan melihat frekuensi kemunculan kata pada *tweet*. Frekuensi kata yang tinggi berarti kata tersebut sering ditulis oleh pengguna Twitter yang melakukan *tweet* terhadap *e-wallet* Go-Pay pada @gopayindonesia ataupun *e-wallet* OVO pada @ovo_id. Berikut merupakan frekuensi kemunculan kata pada *tweet* dengan menampilkan sepuluh frekuensi kata tertinggi pada masing-masing data yang ditunjukkan dalam diagram batang pada Gambar 4.1 dan Gambar 4.2.



Gambar 4.1 Sepuluh Kata dengan Frekuensi Kemunculan Tertinggi pada akun Twitter *E-wallet* @gopayindonesia

Gambar 4.1 menunjukkan bahwa sepuluh kata dengan frekuensi tertinggi adalah "@gopayindonesia", "saya", "di", "@gojekindonesia", "gopay", "bisa", "cashback", "ada", "ini", dan "dan". Kata dengan frekuensi paling tinggi adalah "@gopayindonesia" yaitu sebanyak 1176 kata. Kata ini memiliki frekuensi kemunculan yang sangat tinggi dibandingkan kata-kata selanjutnya. Hal ini terjadi karena kata "@gopayindonesia" merupakan kata yang harus dituliskan sebelum menyampaikan pertanyaan/keluhan kepada *customer service* Go-Pay dan sekaligus adalah *username* akun Twitter *customer service* Go-Pay untuk melakukan komunikasi dengan layanan *customer service e-wallet* Go-Pay di Twitter atau dengan sebutan lain bahwa pengguna Twitter harus melakukan *mention* terhadap akun *customer service* Go-Pay agar dapat melakukan komunikasi. Hal serupa juga terdapat pada frekuensi kemunculan tertinggi nomor empat yaitu kata "@gojekindonesia", diketahui bahwa pengguna akun Twitter yang ingin melakukan komunikasi dengan *customer service* Go-Pay juga tidak jarang melakukan *mention* atau juga ingin berkomunikasi kepada akun Twitter salah satu penyedia layanan Go-Pay yaitu Go-Jek yang merupakan perusahaan penyedia jasa taksi online dengan menawarkan sistem pembayarannya yaitu bisa menggunakan *e-wallet* Go-Pay dan uang tunai.



Gambar 4.2 Sepuluh Kata dengan Frekuensi Kemunculan Tertinggi pada akun Twitter *E-wallet* @ovo_id

Dapat diketahui pada Gambar 4.2 bahwa terdapat sepuluh kata dengan frekuensi tertinggi yaitu "@ovo_id", "saya", "ovo, "di", "ada", "ke", "ini", "dan", "bisa", dan "sudah", "@ovo_id" merupakan kata dengan frekuensi tertinggi, serupa dengan akun @gopayindonesia, hal yang harus dilakukan apabila ingin melakukan komunikasi dengan layanan *customer service* OVO maka diharuskan melakukan *mention* kepada akun Twitter *customer service e-wallet* OVO. Oleh sebab itu "@ovo_id" adalah kata dengan frekuensi kemunculan tertinggi. Selain itu, kata "@ovo_id" berfrekuensi sebanyak 3301 lebih banyak dibandingkan jumlah *tweet* keseluruhan yaitu sebanyak 3244 yang berarti satu pengguna kemungkinan melakukan *mention* dua kali kepada akun Twitter "@ovo_id" atau lebih pada satu *tweet*.

4.1.2 Preprocessing Data *Tweet E-Wallet* Go-Pay dan OVO

Karakterisik data pada Gambar 4.1 dan Gambar 4.2 masih menunjukkan adanya kata-kata yang tidak terlalu memiliki arti penting pada analisis *clustering* ini, seperti kata "@gopayindonesia" dan "@ovo_id" yang wajib dituliskan. Sedangkan kata "di", "ini", "ada", "saya", "ke", dan lain-lain adalah kata-kata yang juga tidak dapat dibiarkan untuk masuk ke proses selanjutnya agar memberikan hasil *clustering* yang maksimal sehingga perlu dilakukan *preprocessing* untuk

membersihkan kumpulan kata-kata yang ada. Tahapan *preprocessing* yang dilakukan antara lain *data cleaning* (menghapus link, menghapus tanda *retweet*, menghapus baris *enter*, menghapus tanda baca, menghapus nomor yang tidak berarti), mengubah kata menjadi *lowercase*, memperbaiki ejaan kata, mencari persamaan kata, menghilangkan *stopwords*, dan melakukan *stemming* (menemukan kata dasar tiap kata). Contoh tahapan *preprocessing* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Ilustrasi Tahapan *Preprocessing* pada Kalimat *Tweet*

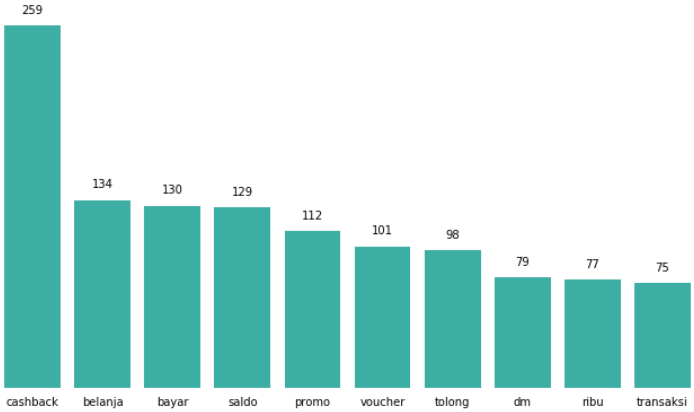
Preprocessing	**Kalimat**	**Keterangan**
Kalimat awal	@gopayindonesia min cashback voucher 40% lotte belum masuk padahal sudah di email ke cs nya 3 minggu lalu tapi tidak ditanggapi. Gimana ya?	-
Menghilangkan *hashtag*, *mention*, *link*, tanda *retweet*, dan angka	min cashback voucher % lotte belum masuk padahal sudah di email ke cs nya minggu lalu tapi tidak ditanggapi. Gimana ya?	Menghilangkan @gopayindonesia
Menghilangkan *punctuation*	min cashback voucher lotte belum masuk padahal sudah di email ke cs nya minggu lalu tapi tidak ditanggapi Gimana ya	Menghilangkan tanda baca titik (.), persen (%), dan tanya (?)
Lowercase	min cashback voucher lotte belum masuk padahal sudah di email ke cs nya minggu lalu tapi tidak ditanggapi gimana ya	Mengubah kalimat ke dalam huruf kecil

Tabel 4.1 (Lanjutan) Ilustrasi Tahapan *Preprocessing* pada Kalimat *Tweet*

Preprocessing	**Kalimat**	**Keterangan**
Persamaan kata	admin cashback voucher lotte belum masuk padahal sudah di email ke customer servis nya minggu lalu tapi tidak ditanggapi gimana ya	Mengganti "min" menjadi "admin" dan "cs" menjadi "customer servis"
Stemming	admin cashback voucher lotte belum masuk padahal sudah di email ke customer servis nya minggu lalu tapi tidak tanggap gimana ya	Mencari kata dasar untuk kata "ditanggapi"
Stopword	admin cashback voucher lotte belum masuk email customer servis tidak tanggap	Menghilangkan kata "padahal", "di", "ke", "nya", "minggu", "lalu", "tapi", "gimana", "ya"

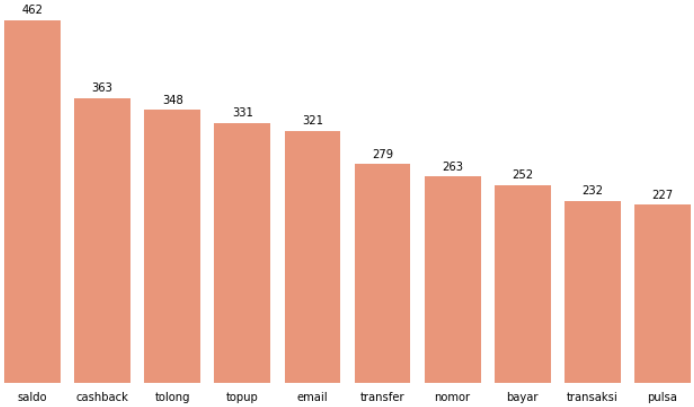
4.1.3 Karakteristik Data *Tweet E-Wallet* Go-Pay dan OVO setelah *Preprocessing*

Tahapan *preprocessing* menghilangkan kata dan karakter yang tidak berarti. Jumlah *tweet* setelah dilakukan *preprocessing* berkurang menjadi 846 *tweet* untuk akun *e-wallet* Go-Pay dan 2460 *tweet* untuk akun OVO. Hal ini dapat terjadi karena terdapat *tweet* yang mengandung kata-kata tidak berarti. Setelah kata-kata tersebut hilang, maka hanya akan tersisa kata-kata penting untuk mengetahui karakteristik *tweet* yang ditujukan ke @gopayindonesia dan juga @ovo_id dan sebagai data untuk analisis *clustering* pada penelitian ini. Frekuensi kemunculan kata-kata setelah dilakukan *preprocessing* pada data *e-wallet* Go-Pay dan *e-wallet* OVO tersebut dapat ditunjukkan dengan diagram batang pada Gambar 4.3 dan Gambar 4.4.



Gambar 4.3 Sepuluh Kata dengan Frekuensi Kemunculan Tertinggi Setelah *Preprocessing* pada akun Twitter *E-wallet* @gopayindonesia

Gambar 4.3 menunjukkan bahwa kata dengan frekuensi tertinggi adalah "cashback", dan selanjutnya diikuti oleh kata "belanja", "bayar", "bayar", "saldo", "promo", "voucher", "tolong", "dm", "ribu", dan "transaksi". Kata-kata ini menunjukkan bahwa *tweet* ini membahas jenis layanan dan produk yang dimiliki oleh *e-wallet* Go-Pay sebagai perusahaan *e-wallet* yaitu seperti layanan cashback, promo, saldo dan voucher yang ditawarkan. Kemudian terdapat kata "tolong" dan "dm", kata-kata ini dapat menunjukkan bahwa *tweet* mengandung pembahasan terkait layanan *customer service e-wallet* Go-Pay seperti meminta tolong untuk menyelesaikan permasalah terkait layanan *e-wallet* Go-Pay melalui pengiriman *direct message* atau DM, yaitu mengirimkan pesan langsung dengan tanpa batas karakter atau *unlimited text* yang hanya dapat dilihat oleh pengirim pesan dan yang dituju yaitu akun Twitter *e-wallet* Go-Pay. Fitur ini dimanfaatkan banyak pengguna agar mendapatkan keleleluasaan dalam menyampaikan informasi berupa pertanyaan ataupun keluhan yang terjadi pada *e-wallet* Go-Pay yang mereka miliki, dan juga sering digunakan agar penyampaian informasi yang bersifat *privacy* seperti nomor telepon, domisili, nama akun, alamat *e-mail* ataupun jumlah saldo akun tidak tersebar luas.



Gambar 4.4 Sepuluh Kata dengan Frekuensi Kemunculan Tertinggi Setelah *Preprocessing* pada akun Twitter *E-wallet* @ovo_id

Gambar 4.4 dapat diketahui bahwa kata dengan frekuensi tertinggi adalah "saldo", dan selanjutnya diikuti oleh kata "cashback", "tolong", "topup", "email", "transfer", "nomor", "bayar", "transaksi", dan "pulsa". Kata-kata ini menunjukkan bahwa *tweet* ini membahas jenis layanan dan produk yang dimiliki oleh *e-wallet* OVO seperti layanan saldo, cashback, *top-up* dan lain-lain. Kemudian terdapat kata "tolong" dan "email". Kata-kata tersebut diketahui bahwa dapat bersinggungan apabila terdapat dalam satu *tweet*, yang memiliki arti *customer* ingin meminta tolong untuk menyelesaikan permasalah terkait layanyan *e-wallet* OVO melalui pengiriman *email*. Sedangkan kata "nomor" dan "pulsa" juga terdapat di frekuensi kata kemunculan tertinggi nomor tujuh dan sepuluh karena arti "nomor" dan "pulsa" bersinggungan dengan salah satu layanan *e-wallet* OVO, yaitu layanan melakukan transaksi isi pulsa.

4.1.4 *Term Weighting* Data *Tweet E-Wallet* Go-Pay dan OVO

Transfomasi kata dilakukan dengan mengubah data teks menjadi data numerik. Trasformasi kata dilakukan agar data *tweet* dapat dianalisis lebih lanjut pada analisis *clustering*. Kumpulan kata disajikan dalam bentuk frekuensi kemunculan masing-masing kata pada

tweet. Setiap kata dilakukan pembobotan untuk membedakan nilai frekuensinya dengan kata yang lain.

Hal yang pertama kali dilakukan yaitu menghitung kemunculan kata pada per *tweet*. Kemunculan kata pada *tweet* ditunjukkan dengan *Document-Term Matrix*. Contoh DTM untuk data ini ditunjukkan oleh Tabel 4.2.

Tabel 4.2 *Document-Term Matrix*

***Tweet* ke-**	**Kata**				
	cashback	**rekening**	**tolong**	⋯	**voucher**
1	0	0	0	⋯	0
2	0	0	2	⋯	0
⋮	⋮	⋮	⋮	⋱	⋮
845	0	0	0	⋯	0
846	0	1	0	⋯	0

Tabel 4.2 menunjukkan frekuensi kemunculan kata pada masing-masing *tweet*. Misalnya, pada *tweet* kedua kemunculan kata "tolong" sebanyak dua kali. Pada *tweet* ke - 490 kemunculan kata "cashback" dan kata "voucher" masing-masing sebanyak satu kali dan seterusnya. Kata-kata ini menjadi variabel penelitian dan frekuensi kemunculan kata menjadi nilai masing-masing variabel. Jumlah kata yang digunakan sebagai variabel adalah sebanyak 37 kata untuk akun Twitter Go-Pay dan 46 kata untuk akun Twitter OVO.

Setelah mendapatkan *Document–Term Matrix*, akan dilakukan pembobotan dengan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Pembobotan dilakukan untuk mendapatkan tingkat kepentingan suatu kata. TF-IDF dilakukan dengan menghitung nilai *Term Frequency* (TF), yaitu frekuensi kemunculan kata pada *tweet*. Selanjutnya menghitung nilai *Document Frequency* (DF) dan *Inverse Document Frequency* (IDF). *Document Frequency* yaitu jumlah *tweet* yang mengandung kata ke-*i*. *Inverse Document Matrix* (IDF) dapat dihitung dengan persamaan (2.1). Tabel 4.3 menampilkan contoh perhitungan *Document Frequency* dan *Inverse Document Frequency*.

Tabel 4.3 Perhitungan DF dan IDF

Kata	***Tweet* ke-**					**DF**	**IDF**
	1	**2**	⋯	845	846		
cashback	0	0	⋯	0	0	219	$\log\left(\frac{846}{219}\right) = 0,587$
rekening	0	0	⋯	0	1	24	$\log\left(\frac{846}{24}\right) = 1,547$
tolong	0	2	⋯	0	0	84	$\log\left(\frac{846}{84}\right) = 1,003$
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮
voucher	0	0	⋯	0	0	73	$\log\left(\frac{846}{73}\right) = 1,064$

Tabel 4.3 menunjukkan contoh nilai DF dan IDF dari masing-masing kata. Kata "cashback" memiliki nilai DF sebesar 219 yang berarti kata "cashback" muncul dalam 219 *tweet*. Semakin tinggi nilai DF makan nilai IDF akan semakin kecil. Kata "rekening" muncul dalam 24 *tweet* sehingga memiliki nilai DF sebesar 24 dan menghasilkan nilai IDF sebesar 1,570. Nilai ini lebih besar dibandingkan nilai IDF "cashback". Bobot kata "cashback" lebih kecil karena terdapat pada lebih banyak *tweet* sedangkan bobot kata "rekening" memiliki bobot yang lebih besar dikarenakan terdapat pada lebih sedikit dokumen.

Selanjutnya adalah dilakukan perhitungan TF-IDF dengan menggunakan persamaan (2.2), yaitu dengan melakukan perkalian antara nilai TF setiap kata-*i* pada masing-masing *tweet* dengan nilai IDF dari kata ke-*i*. Berikut merupakan hasil perkalian antara nilai TF setiap kata-*i* pada masing-masing *tweet* dengan nilai IDF dari kata ke-*i* atau hasil TF-IDF yang ditunjukkan pada Tabel 4.4.

Tabel 4.4 Perhitungan TF- IDF

Tweet ke-	Kata				
	cashback	**rekening**	**tolong**	⋯	**voucher**
1	0	0	0	⋯	0
2	0	0	2,054	⋯	0
⋮	⋮	⋮	⋮	⋱	⋮
845	0	0	0	⋯	0
846	0	1,570	0	⋯	0

Tabel 4.4 menunjukkan hasil perhitungan TF-IDF pada setiap kata untuk setiap *tweet*. Nilai TF-IDF untuk kata "tolong" pada *tweet* kedua yaitu sebesar 2,054 diperoleh dari nilai TF yaitu 2 dan nilai IDF sebesar 1,027, sehingga TF-IDF diperoleh sebesar 2 × 1,027 = 2,054. Perhitungan nilai TF-IDF untuk kata lainnya juga dilakukan dengan cara demikian. Pada analisis *clustering*, nilai TF-IDF akan digunakan sebagai variabel prediktor.

4.2 Analisis *Clustering* Data *Tweet E-Wallet* Go-Pay dan OVO

Analisis *clustering* bertujuan untuk mengelompokkan *tweet* terhadap akun Twitter *e-wallet* yaitu Go-Pay pada @gopayindonesia dan OVO pada @ovo_id. Analisis *Clustering* dilakukan dengan menggunakan dua metode yaitu *K-Means* dan *K-Means* dengan *Latent Dirichlet Allocation* (LDA). Hasil dari kedua metode tersebut akan didapatkan jumlah *cluster* optimal pada masing-masing metode. Lalu, jumlah *cluster* optimal juga ditetapkan berdasarkan evaluasi hasil *cluster* yang menggunakan nilai *silhouette coefficient*.

4.2.1 Analisis *Clustering* Data *Tweet E-Wallet* Go-Pay Menggunakan Metode *K-Means*

Metode *K-Means* dilakukan untuk *clustering* atau mengelompokkan *tweet* ke dalam beberapa *cluster* berdasarkan variabel, dalam hal ini adalah kata-kata yang terdapat pada *tweet*. Sebelum melakukan *clustering* ditentukan nilai *K* oleh peneliti dengan *K* adalah jumlah *cluster*. Setelah menentukan nilai *K* maka ditentu-

kan titik *centroid* sejumlah *K* secara acak. Kemudian dihitung nilai *euclidean* antara keseluruhan *tweet* dengan masing-masing titik *centroid*. Nilai *euclidean* terkecil berarti *tweet* termasuk ke dalam kelompok tersebut. Ilustrasi perhitungan jarak kuadrat *euclidean* ditunjukkan pada Tabel 4.5.

Tabel 4.5 Perhitungan Jarak Kuadrat *Euclidean*

***Tweet* ke**	**Kata**	
	cashback	**rekening**
1	0,000	0,696
2	0,000	0,552
3	0,419	0,000
4	0,809	0,303
5	0,461	0,000

Tabel 4.5 merupakan data ilustrasi untuk perhitungan nilai jarak kuadrat *euclidean*. Pada data tersebut akan dilakukan *clustering* terhadap *tweet* berdasarkan variabel Cashback dan Rekening. Pertama ditentukan jumlah *cluster* atau nilai *K* yang digunakan dalam *K-Means*. Misalkan menggunakan nilai *K* = 2 yang berarti akan dibentuk 2 *cluster* dari 5 *tweet* pada Tabel 4.5. Selanjutnya adalah menentukan 2 titik *centroid* secara acak. Dipilih titik *centroid* pertama pada variabel Cashback = 0 dan Rekening = 0,6, titik *centroid* kedua pada pada variabel Cashback = 0.6 dan Rekening = 0,15. Kemudian yaitu dihitung jarak kuadrat *euclidean* pada masing-masing *tweet* terhadap titik *centroid*.

$$d^2(\text{tweet}1, \text{centroid}1) = (0-0)^2 + (0,696-0,6)^2 = 0,0092$$
$$d^2(\text{tweet}1, \text{centroid}2) = (0-0,6)^2 + (0,696-0,15)^2 = 0,6581$$
$$d^2(\text{tweet}2, \text{centroid}1) = (0-0)^2 + (0,552-0,6)^2 = 0,0023$$
$$d^2(\text{tweet}2, \text{centroid}2) = (0-0,6)^2 + (0,552-0,15)^2 = 0,5216$$
$$d^2(\text{tweet}3, \text{centroid}1) = (0,419-0)^2 + (0-0,6)^2 = 0,5356$$
$$d^2(\text{tweet}3, \text{centroid}2) = (0,419-0,6)^2 + (0-0,15)^2 = 0,0552$$
$$d^2(\text{tweet}4, \text{centroid}1) = (0,809-0)^2 + (0,303-0,6)^2 = 0,7427$$

$$d^2(\text{tweet}\,4, \text{centroid}\,2) = (0,809-0,6)^2 + (0,303-0,15)^2 = 0,0671$$
$$d^2(\text{tweet}\,5, \text{centroid}\,1) = (0,461-0)^2 + (0-0,6)^2 = 0,5725$$
$$d^2(\text{tweet}\,5, \text{centroid}\,2) = (0,461-0,6)^2 + (0-0,15)^2 = 0,0418$$

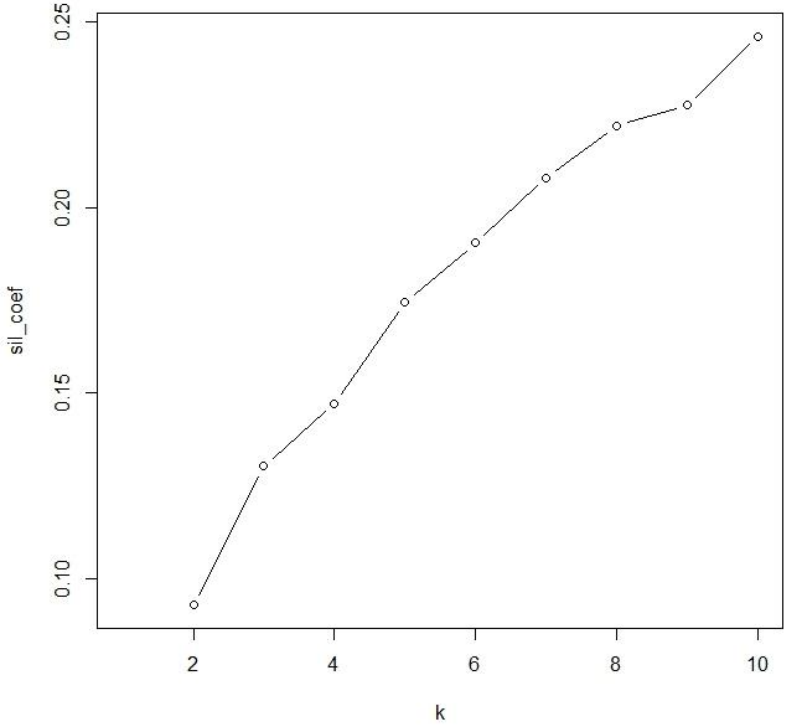
Setelah melakukan perhitungan jarak kuadrat *euclidean* selanjutnya ditentukan data *tweet* tersebut masuk ke dalam kelompok titik *centroid* mana. Penentuan tersebut dilihat berdasarkan nilai terkecil pada jarak kuadrat *euclidean*. Berdasarkan perhitungan maka *tweet* 1 termasuk ke dalam *centroid* 1, *tweet* 2 masuk ke dalam *centroid* 1, *tweet* 3 termasuk ke dalam anggota *centroid* 2, *tweet* 4 masuk ke dalam *centroid* 2, dan *tweet* 5 masuk ke dalam anggota *centroid* 2. Setelah terbentuk kelompok antara *centroid* 1 dan *centroid* 2 selanjutnya kembali dilakukan penentuan titik *centroid*.

$$\overline{v}_{1,\text{cashback}} = \frac{0+0}{2} = 0$$
$$\overline{v}_{1,\text{rekening}} = \frac{0,696+0,552}{2} = 0,624$$
$$\overline{v}_{2,\text{cashback}} = \frac{0,419+0,809+0,461}{3} = 0,563$$
$$\overline{v}_{2,\text{rekening}} = \frac{0+0,303+0}{3} = 0,101$$

Berdasarkan perhitungan di atas maka terbentuk titik *centroid* yaitu titik *centroid* pertama pada variabel Cashback = 0 dan variabel Rekening = 0,624. Titik *centroid* kedua adalah pada variabel Cashback = 0,563 dan variabel Rekening = 0,101. Dengan menggunakan titik *centroid* yang baru kembali dihitung jarak kuadrat *euclidean* masing-masing *tweet* dengan kedua titik *centroid* yang baru. Jika tidak terjadi perubahan anggota pada masing-masing titik *centroid* maka proses *clustering* dengan metode *K-Means* selesai. Berdasarkan ilustrasi pada Tabel 4.5 diperoleh hasil *clustering* yaitu untuk *cluster* 1 terdiri dari *tweet* 1 dan *tweet* 2. Kemudian *cluster* 2 terdiri dari *tweet* 3, *tweet* 4, dan *tweet* 5.

Dalam melakukan *clustering K-Means* pada data *tweet* terhadap akun @gopayindonesia, hal yang pertama dilakukan adalah me-

nentukan nilai *K* optimal. Untuk menentukan nilai *K* terbaik dapat ditentukan dengan nilai *silhouette coefficient*. Semakin tinggi nilai *silhouette coefficient* menunjukkan bahwa nilai *K* semakin baik digunakan untuk melakukan analisis *clustering*. Pada penelitian ini akan dilakukan percobaan untuk nilai *K* optimal. Penentuan jumlah *cluster* optimum dilakukan dengan *K* mulai dari 2 hingga *K* = 10 dan diperoleh grafik dari nilai *silhouette coefficient* yang dapat dilihat pada Gambar 4.5



Gambar 4.5 Grafik *Silhoutte Coefficient* dari Metode *K-Means* pada Data *E-wallet* Go-Pay

Tampak pada Gambar 4.5 dengan jumlah *cluster* (*K*) sebanyak 10 memiliki nilai *silhouette coefficient* yang paling tinggi yaitu sebesar 0,246. Oleh karena itu, dapat disimpulkan bahwa jumlah *cluster* op-

timum untuk metode *K-Means* adalah sebanyak 10 *cluster*. Nilai *silhouette coefficient* yang diperoleh tersebut cukup rendah, artinya sebagian besar *tweet* berada di antara dua *cluster* sehingga kurang jelas harus dimasukkan ke dalam *cluster* yang mana. Dengan jumlah *cluster* sebanyak 10 maka diperoleh distribusi *tweet* terhadap *cluster* yang ditunjukkan pada Tabel 4.6.

Tabel 4.6 Distribusi *Tweet* pada *Cluster* untuk Data *E-wallet* Go-Pay

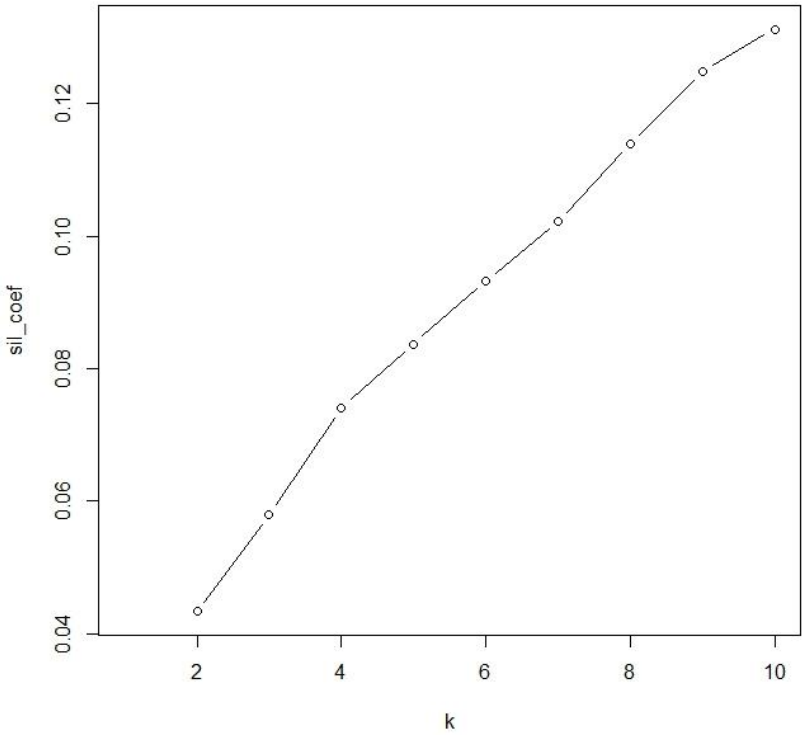
Cluster	**Jumlah T*weet***	**Deskripsi**
1	219	*Email*, Gojek
2	47	Tolong
3	62	Tolong, Cek, Keluh
4	57	*Cashback*
5	75	Bayar, *Cashback*
6	91	*Voucher, Cashback*
7	89	*Cashback*
8	81	Promo, *Cashback*
9	47	*Top-up*, Saldo
10	78	Saldo, Isi

Berdasarkan Tabel 4.7 diketahui jumlah *tweet* terbanyak berada pada *cluster* 1 yaitu sebanyak 219 *tweet* dan jumlah *tweet* paling sedikit ada pada *cluster* 2 dan *cluster* 9 sejumlah masing-masing 47 *tweet*. Dengan menggunakan jumlah *cluster* sebanyak 10 diperoleh bahwa *cluster* 1 membicarakan kata "email". Contoh pembahasan pada *cluster* 1 seperti *tweet* yang dilayangkan pengguna mengenai bagaimana kelanjutan aduan pengguna yang sudah dilaporkan melalui *email customer service* Go-Pay. Pada *cluster* 2 merupakan *cluster* yang membahas terkait permintaan tolong. Pada *Cluster* 3 merupakan kelompok yang membahas tentang permintaan tolong dan mengecek serta keluhan pengguna. *Cluster* 4 merupakan kelompok yang membahas terkait *cashback*. *Cluster* 5 merupakan kelompok yang membahas tentang *cashback* dan pembayaran menggunakan Go-Pay. *Cluster* 6 merupakan kelompok yang membahas ucapan *cashback* dan *voucher*. *Cluster* 7 merupakan kelompok yang juga membahas tentang *cash-*

back. *Cluster* 8 merupakan kelompok yang membahas tentang promo Go-Pay serta *cashback*. *Cluster* 9 membahas tentang permasalahan terkait *top-up* saldo Go-Pay. *Cluster* 10 membahas seputar isi saldo Go-Pay. Dapat dilihat bahwa terdapat beberapa kelompok dengan deskripsi yang terulang dan hampir sama seperti *cluster* 4, *cluster* 5, *cluster* 6, *cluster* 7, dan *cluster* 8 yaitu sama-sama terdapat dan membahas kata "cashback" yang berarti pada *cluster* 4, *cluster* 5, *cluster* 6, *cluster* 7, dan *cluster* 8 sedang membahas tweet pelanggan yang bertanya tentang promo *cashback* atau berkeluh mengenai *cashback* yang belum didapatkan pengguna *e-wallet* Go-Pay. Sedangkan *cluster* 9 dan *cluster* 10 juga memiliki deskripsi yang hampir sama mengenai *top-up* dan isi saldo, jadi pada kedua *cluster* tersebut memiliki bahasan yang sama seperti membahas keluhan pelanggan yang ingin mengetahui bagaimana cara melakukan *top-up* atau isi saldo pada *e-wallet* Go-Pay. Lalu pada *cluster* 2 dan *cluster* 3 juga terjadi hal serupa yaitu kedua *cluster* tersebut bermunculan kata "tolong" yang berindikasi *tweet* oleh pengguna *e-wallet* kepada Twitter *e-wallet* Go-Pay membahas mengenai permintaan tolong untuk menanggapi persoalan keluhan sesuatu tentang *e-wallet* Go-Pay yang dimiliki pengguna. Hal ini dapat terjadi dikarenakan nilai *silhouette coefficient* yang dihasilkan oleh metode *K-Means* masih relatif kecil.

4.2.2 Analisis *Clustering* Data *Tweet E-Wallet* OVO Menggunakan Metode *K-Means*

Analisis *clustering* menggunakan *K-Means* pada data *tweet* terhadap akun Twitter *e-wallet* OVO *username* @ovo_id juga dilakukan dengan cara yang sama seperti halnya proses *clustering* menggunakan *K-Means* pada *e-wallet* Go-Pay di sub-bab sebelum. Lalu hal yang pertama dilakukan merupakan penetuan jumlah *cluster* atau nilai *K* optimal dengan menggunakan nilai *silhouette coefficient*. Semakin besar nilai *silhouette coeficient* maka nilai *K* tersebut semakin baik untuk membuat *cluster*. Dilakukan percobaan yang sama seperti data *e-wallet* Go-Pay yaitu menggunakan nilai $K = 2$ hingga $K = 10$. Grafik nilai *silhouette coefficient* untuk nilai *K* yang telah ditentukan dari metode *K-Means* pada *e-wallet* OVO ditunjukkan pada Gambar 4.6.



Gambar 4.6 Grafik *Silhoutte Coefficient* dari Metode *K-Means* pada Data *E-wallet* OVO

Pada Gambar 4.6 menunjukkan bahwa dengan jumlah *cluster* (*K*) sebanyak 2 hingga 10 memiliki nilai *silhouette coefficient* yang paling tinggi sebesar 0,131 pada jumlah *cluster* 10 atau *K* = 10. Oleh karena itu, dapat disimpulkan bahwa jumlah *cluster* optimum untuk metode *K-Means* adalah sebanyak 10 *cluster*. Nilai *silhouette coefficient* yang diperoleh tersebut cukup rendah, dapat diindikasikan bahwa sebagian besar *tweet* berada di antara dua *cluster* sehingga kurang jelas harus dimasukkan ke dalam *cluster* yang mana. Dengan jumlah *cluster* sebanyak 10 maka diperoleh distribusi dan deskripsi *tweet* terhadap masing-masing *cluster* yaitu *cluster* 1 hingga *cluster* 10 yang ditunjukkan pada Tabel 4.7.

Tabel 4.7 Distribusi *Tweet* pada *Cluster* untuk Data *E-wallet* OVO

Cluster	**Jumlah T*weet***	**Deskripsi**
1	180	*Top-up*
2	242	Transfer, Rekening
3	220	*Cashback*
4	140	Nomor, Tiket, *Email*
5	247	Saldo
6	85	Tunggu, *Email*
7	141	*Customer*, Servis
8	257	Bayar, Akun
9	185	Pulsa
10	763	*Email*, Tolong

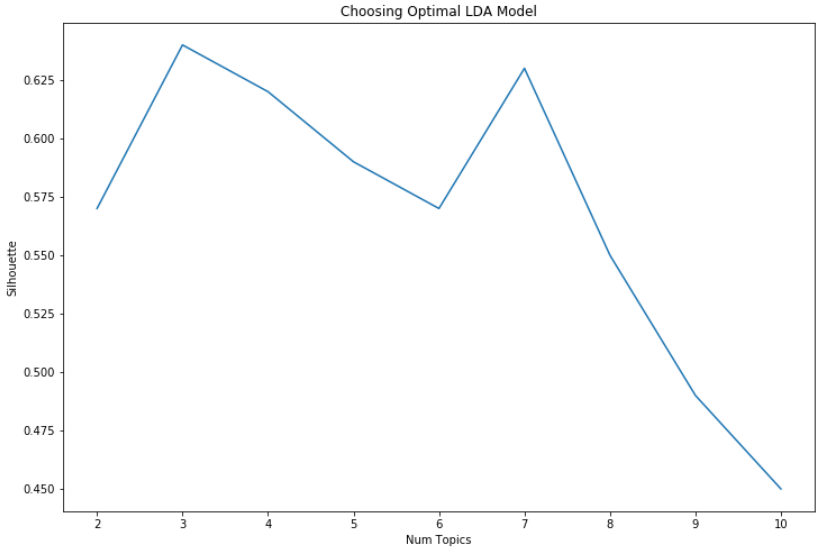
Berdasarkan Tabel 4.8 diketahui dengan menggunakan jumlah *cluster* sebanyak 10 diperoleh jumlah *tweet* terbanyak berada pada *cluster* 10 yaitu sebanyak 763 *tweet* dan jumlah *tweet* paling sedikit ada pada *cluster* 6 sebanyak 85 *tweet*. Kedua *cluster* tersebut tampak berisi pembahasan tentang menanyakan berapa lama lagi menunggu untuk respon aduan yang telah di-kirim melalui *email customer service e-wallet* OVO salah satunya seperti aduan *cashback* yang belum didapatkan pengguna. Di *Cluster* 1 membicarakan tentang *top-up*. Pada *cluster* 2 merupakan *cluster* yang membahas terkait transfer melalui rekening bank untuk *e-wallet* OVO. *Cluster* 3 merupakan kelompok membahas tentang *cashback*. Lalu *Cluster* 4 merupakan kelompok yang membahas terkait nomor tiket yang didapatkan ketika telah melakukan aduan melalui *email customer service* OVO. *Cluster* 5 merupakan kelompok yang membahas tentang saldo. *Cluster* 7 merupakan kelompok yang membahas tentang *customer service*. *Cluster* 8 merupakan kelompok yang membahas tentang akun OVO dan pembayaran *e-wallet* OVO. *Cluster* 9 membahas tentang permasalahan terkait pulsa yang merupakan salah satu fitur pada OVO. Terlihat bahwa beberapa kelompok dengan deskripsi yang terulang dan hampir sama. Hal

ini terjadi karena nilai *silhouette coefficient* masih kecil. Selanjutnya yaitu dilakukan analisis menggunakan metode *K-Means* dengan *Latent Dirichlet Allocation* (LDA) pada masing-masing data *e-wallet* Go-Pay dan OVO.

4.2.3 Analisis *Clustering* Data *Tweet E-Wallet* Go-Pay Menggunakan Metode *K-Means* dengan LDA

Pada analisis *cluster* menggunakan metode *K-Means* masih terdapat kekurangan pada evaluasi model yaitu pada nilai *silhouette coefficient* yang masih relatif kecil dan masih banyak terdapat beberapa kelompok atau *cluster* dengan deskripsi *tweet* yang terulang dan hampir sama. Pada penelitian ini, diusulkan penggunaan metode *Latent Dirichlet Allocation* (LDA) yang diharapkan dapat meningkatkan performa dari *clustering*. LDA merupakan metode untuk menemukan topik dari sekumpulan dokumen. Setiap topik yang terbentuk terdiri dari kata-kata yang menjadi karakteristik topik tersebut. Kata pertama yang muncul pada model suatu topik merupakan kata yang paling menjelaskan suatu topik tersebut.

Tahap awal dalam melakukan pemodelan dokumen dengan LDA adalah dengan menentukan jumlah topik yang akan dibentuk. Jumlah topik yang dibentuk ditentukan oleh peneliti. Pada penelitian ini akan dilakukan percobaan terhadap jumlah topik sebanyak 2 topik hingga 10 topik. Topik terbaik ditentukan berdasarkan nilai *silhouette coefficient* paling tinggi. Semakin tinggi *silhouette coefficient* maka harapannya semakin baik pula penentuan jumlah topik atau distribusi pada setiap topik yang didapatkan. Selain itu juga dapat dilakukan dengan mengevaluasi kata-kata yang muncul pada topik dan memastikan bahwa tidak lebih dari satu topik membahas hal yang sama. Apabila jumlah topik yang terpilih menggunakan metode LDA ini menghasilkan distribusi *tweet* yang optimal maka deskripsi dari masing-masing topik juga akan relevan sebagaimana ingin diketahui informasi atau permasalahan apa saja yang dihadapi oleh masing-masing perusahaan *e-wallet* yaitu dalam penelitian ini adalah *e-wallet* Go-Pay maupun *e-wallet* OVO pada rentang waktu 10 Februari 2019 hingga 14 Maret 2019. Nilai *silhouette coefficient* masing-masing topik ditunjukkan pada Gambar 4.7.



Gambar 4.7 Grafik *Silhoutte Coefficient* dari Metode LDA pada Data *E-wallet* Go-Pay

Tampak pada Gambar 4.7 menunjukkan bahwa jumlah topik sebanyak 3 memiliki nilai *silhouette coefficient* tertinggi yaitu 0,640. Oleh karena itu digunakan jumlah topik sebanyak 3 untuk membentuk model LDA. Dengan jumlah topik sebanyak 3 maka diperoleh model LDA sebagai berikut. Model LDA sebagai berikut didapatkan berdasarkan persamaan (2.7).

Tabel 4.8 Model LDA untuk Data *E-wallet* Go-Pay

Topik	Model
Topik 1	bayar * 0,178 + dm * 0,097 + gojek * 0,063 + cek * 0,059 + email * 0,057 +…
Topik 2	tolong * 0,16 + saldo * 0,156 + transaksi * 0,118 + topup * 0,091 + proses * 0,063 + …
Topik 3	cashback * 0,282 + belanja * 0,146 + promo * 0,117 + voucher * 0,116 + alfamart * 0,071 + …

Pada Tabel 4.8 diketahui model LDA yang terbentuk pada masing-masing topik. Kata yang memiliki bobot paling tinggi menunjukkan karakteristik dari topik tersebut. Maka dari itu, pada topik 1 yang menjadi kata dengan bobot tertinggi adalah kata "bayar". Pada topik 2 yang menjadi kata dengan bobot tertinggi adalah kata "tolong". Pada topik 3 diketahui bahwa kata yang paling penting adalah kata "cashback". Selanjutnya dengan adanya model tersebut dapat diperoleh probabilitas masing-masing *tweet* terhadap topik yang terbentuk. Probabilitas masing-masing *tweet* terhadap topik ditunjukkan pada Tabel 4.9.

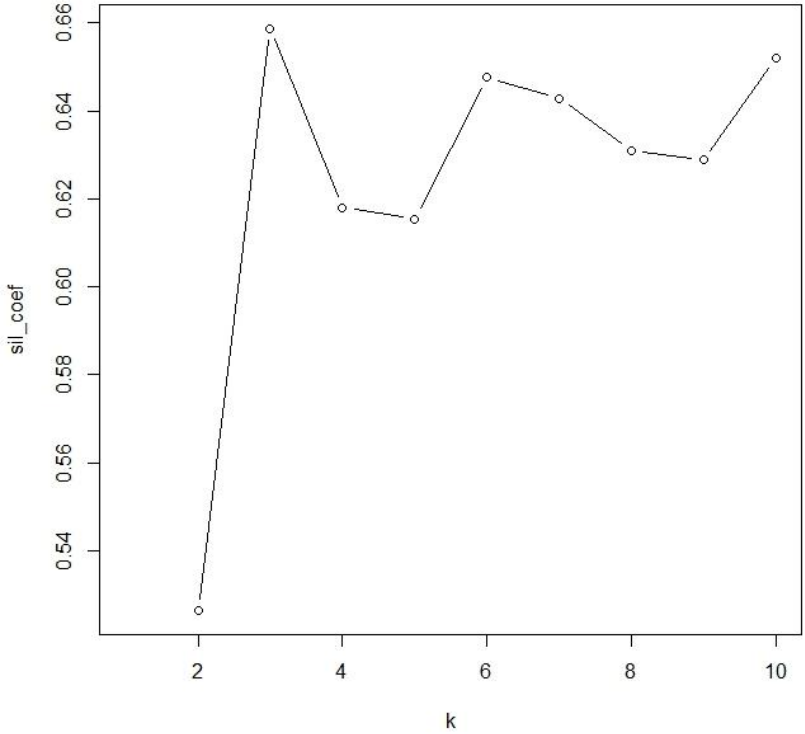
Tabel 4.9 Probabilitas *Tweet* Terhadap Topik LDA pada Data *E-wallet* Go-Pay

***Tweet* ke**	**Topik 1**	**Topik 2**	**Topik 3**
1	0,167	0,666	0,1667
2	0,202	0,715	0,083
3	0,081	0,864	0,055
⋮	⋮	⋮	⋮
844	0,425	0,111	0,464
845	0,589	0,327	0,083
846	0,179	0,654	0,168

Pada tahap ini variabel awal yang berjumlah 37 menjadi topik berjumlah 3. Tabel 4.9 adalah probabilitas setiap *tweet* terhadap topik yang terbentuk pada LDA yaitu sebanyak 3 topik. Misalkan pada *tweet* pertama memiliki probabilitas paling besar pada topik 2 dan probabilitas yang rendah pada Topik 1 maupun Topik 3. Probabilitas topik ini selanjutnya digunakan untuk melakukan analisis *clustering* dengan metode *K-Means* untuk mengelompokkan data *tweet* berdasarkan kesamaan probabilitas topik.

Saat melakukan *clustering* dengan *K-Means* terlebih dahulu ditentukan nilai *K* optimal. Percobaan dilakukan dengan menggunakan nilai $K = 2$ hingga 10. Nilai *K* optimal dihitung dengan ni-

lai *silhouette coefficient*. Grafik *silhouette coefficient* untuk *K* optimal ditunjukkan oleh Gambar 4.8.



Gambar 4.8 Grafik *Silhoutte Coefficient* dari Metode *K-Means* dengan LDA pada *E-wallet* Go-Pay

Gambar 4.8 menunjukkan bahwa dengan jumlah *cluster* sejumlah 3 memiliki nilai *silhouette coefficient* yang paling tinggi yaitu 0,659. Oleh karena itu, disimpulkan bahwa jumlah *cluster* atau kelompok optimum untuk metode *K-Means* dengan menggunakan LDA adalah sebanyak 3 *cluster*. Nilai *silhouette coefficient* yang diperoleh meningkat dengan menggunakan LDA. Dengan jumlah *cluster* sebanyak 3 maka diperoleh distribusi *tweet* terhadap *cluster* yang ditunjukkan pada Tabel 4.10.

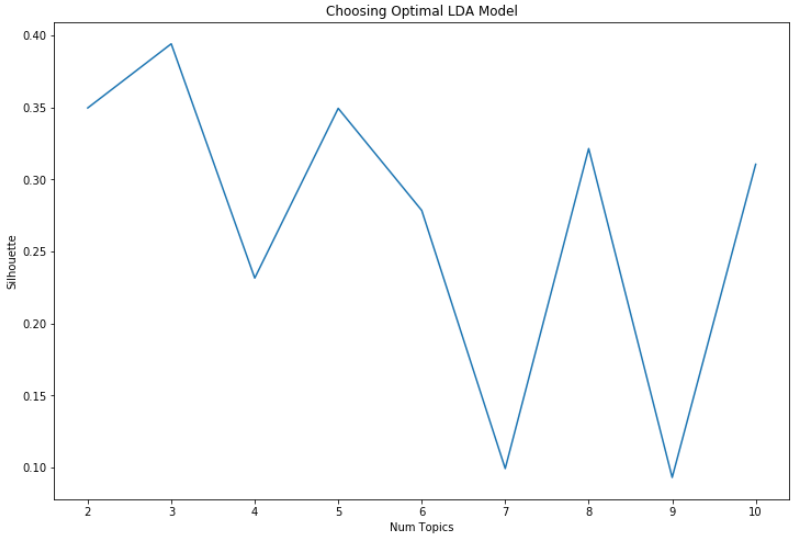
Tabel 4.10 Distribusi *Tweet* pada *Cluster* untuk Data *E-wallet* Go-Pay pada Metode *K-Means* dengan LDA

Cluster	Jumlah T*weet*	Deskripsi
1	274	Pembayaran
2	267	Saldo, Tolong, *Top-up*, Transaksi
3	305	*Cashback*, Promo, Belanja

Berdasarkan Tabel 4.10 diketahui jumlah *tweet* terbanyak ada pada *cluster* 3 yaitu sebanyak 305 *tweet* dan jumlah *tweet* paling sedikit ada pada *cluster* 2 sejumlah 267 *tweet*. Dengan menggunakan jumlah *cluster* sebanyak 3 diperoleh bahwa *cluster* 1 membicarakan tentang pembayaran menggunakan *e-wallet* Go-Pay. Pada *cluster* 2 merupakan *cluster* yang membahas terkait saldo Go-Pay. Lalu *Cluster* 3 merupakan kelompok yang membahas tentang *cashback*.

4.2.4 Analisis *Clustering* Data *Tweet E-Wallet* OVO Menggunakan Metode *K-Means* dengan LDA

Penggunaan metode LDA juga dilakukan pada data OVO. Dalam melakukan LDA terlebih dahulu ditentukan jumlah topik yang optimal. Pada penelitian ini akan dilakukan percobaan terhadap jumlah topik sebanyak 2 topik hingga 10 topik pada data *tweet* OVO. Topik terbaik kembali ditentukan berdasarkan *silhouette coefficient* yang paling tinggi. Semakin tinggi nilai *silhouette coefficient* maka harapannya semakin baik pula penetuan jumlah topik atau distribusi *tweet* pada tiap topik yang didapatkan. Selain itu juga dapat dilakukan dengan melalukan evaluasi kata-kata yang muncul pada topik dan memastikan bahwa tidak lebih dari satu topik membahas hal yang sama. Apabila jumlah topik yang terpilih menggunakan metode LDA ini menghasilkan distribusi *tweet* yang baik atau optimal maka deskripsi dari masing-masing topik juga akan relevan. Berikut adalah nilai *silhouette coefficient* untuk masing-masing jumlah topik sebanyak 2 topik hingga 10 topik ditunjukkan pada Gambar 4.9.



Gambar 4.9 Grafik *Silhoutte Coefficient* dari Metode LDA pada Data *E-wallet* OVO

Gambar 4.9 menunjukkan bahwa jumlah topik sebesar 3 topik yang memiliki nilai *silhouette coefficient* paling tinggi yaitu 0,394. Oleh karena itu digunakan jumlah topik sebanyak 3 untuk membentuk model LDA. Dengan menggunakan jumlah topik sebanyak 3 diperoleh model LDA. Model LDA sebagai berikut berdasarkan persamaan (2.7).

Tabel 4.11 Model LDA untuk Data *E-wallet* OVO

Topik	Model
Topik 1	cashback * 0,115 + bayar * 0,081 + pulsa * 0,08 + saldo * 0,071 + tolong * 0,07 +…
Topik 2	email * 0,136 + nomor * 0,114 + customer * 0,083 + tiket * 0,081 + servis * 0,07 +…
Topik 3	topup * 0,122 + saldo * 0,12 + transfer * 0,096 + akun * 0,064 + aplikasi * 0,058 +…

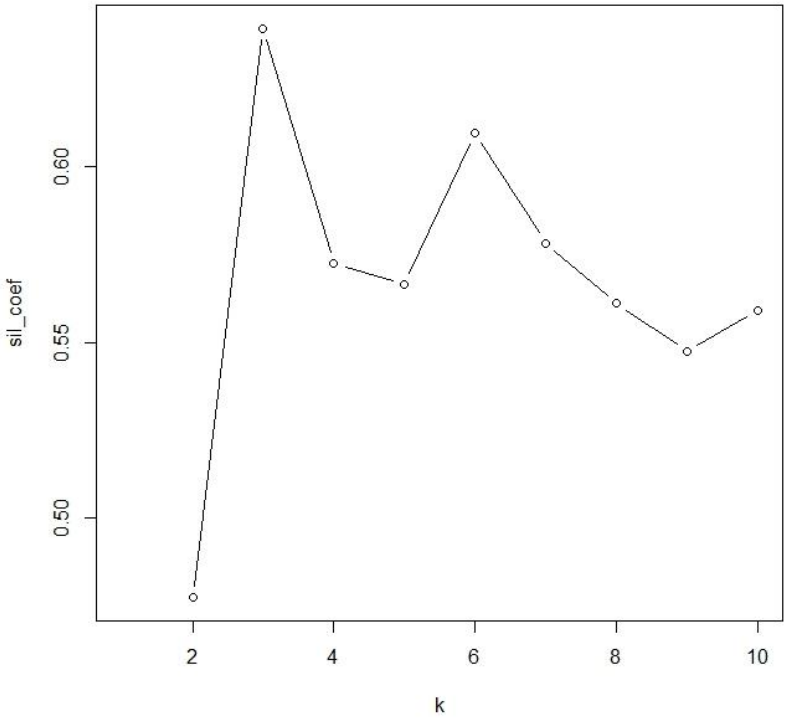
Berdasarkan Tabel 4.11 diketahui model LDA yang terbentuk adalah sebanyak 3 topik. Pada topik 1 yang menjadi kata dengan bobot tertinggi adalah kata "cashback". Pada topik 2 yang menjadi kata dengan bobo tertinggi adalah kata "email". Pada topik 3 diketahui bahwa kata yang paling penting adalah kata "top-up". Selanjutnya dengan model tersebut diperoleh probabilitas masing-masing *tweet* terhadap topik yang terbentuk. Probabilitas masing-masing *tweet* terhadap topik ditunjukkan pada Tabel 4.12.

Tabel 4.12 Probabilitas *Tweet* Terhadap Topik LDA pada Data *E-wallet* OVO

***Tweet* ke**	**Topik 1**	**Topik 2**	**Topik 3**
1	0,055	0,279	0,666
2	0,037	0,802	0,161
3	0,042	0,915	0,043
⋮	⋮	⋮	⋮
2458	0,666	0,264	0,070
2459	0,666	0,265	0,069
2460	0,888	0,055	0,056

Pada tahap ini telah dilakukan membuat variabel awal berjumlah 46 menjadi topik berjumlah 3. Tabel 4.12 merupakan probabilitas setiap *tweet* terhadap topik yang terbentuk pada LDA. Misalkan pada *tweet* 1 memiliki probabilitas paling besar pada topik 3 dan probabilitas yang rendah pada topik yang lain. Probabilitas topik ini selanjutnya digunakan untuk melakukan *clustering* dengan metode *K-Means* untuk mengelompokkan data *tweet* berdasarkan kesamaan probabilitas topik.

Dalam melakukan *clustering* dengan *K-Means* terlebih dahulu ditentukan nilai *K* optimal. Percobaan kembali dilakukan dengan menggunakan nilai $K = 2$ hingga $K = 10$. Nilai *K* optimal dihitung dengan nilai *silhouette coefficient* kembali. Berikut adalah grafik nilai *silhouette coefficient* untuk nilai *K* optimal ditunjukkan oleh Gambar 4.10.



Gambar 4.10 Grafik *Silhoutte Coefficient* dari Metode *K-Means* dengan LDA pada Data *E-wallet* OVO

Tampak pada Gambar 4.10 menunjukkan bahwa dengan jumlah *cluster* (*K*) sebanyak 3 menghasilkan nilai *silhouette coefficient* yang paling tinggi yaitu sebesar 0,639. Oleh karena itu, dapat disimpulkan bahwa jumlah *cluster* optimum untuk metode *K-Means* dengan menggunakan LDA adalah sebanyak 3 *cluster*. Dengan penggunaan LDA nilai *silhouette coefficient* yang diperoleh pada *clustering* meningkat. Hal ini menunjukkan bahwa hasil *cluster* dengan metode LDA lebih baik. Dengan jumlah *cluster* sebanyak 3 maka diperoleh jumlah *tweet* dan deskripsi terhadap masing-masing *cluster* yang ditunjukkan pada Tabel 4.13.

Tabel 4.13 Distribusi *Tweet* pada *Cluster* untuk Data *E-wallet* OVO pada Metode *K-Means* dengan LDA

Cluster	Jumlah T*weet*	Deskripsi
1	906	*Email*, *Customer*, Servis
2	679	Saldo, *Top-up*
3	875	*Cashback*

Berdasarkan Tabel 4.13 diketahui jumlah *tweet* terbanyak ada pada *cluster* 1 yaitu sebanyak 906 *tweet* dan jumlah *tweet* paling sedikit ada pada *cluster* 2. Dengan menggunakan jumlah *cluster* sebanyak 3 diperoleh bahwa *cluster* 1 merupakan kelompok yang membahas mengenai *email*, yang kebanyakan muncul dikarenakan pengguna menanyakan kelanjutan respon atau balasan dari *customer service* OVO yang telah dikirim melalui *e-mail customer service* OVO. Pada *Cluster* 2 merupakan kelompok yang membahas tentang *top-up* saldo. *Cluster* 3 merupakan kelompok yang membahas tentang *cashback* OVO.

4.2.5 Evaluasi Metode *Clustering* Terbaik

Evalusi metode *cluster* terbaik dilakukan dengan melihat *silhouette coefficient* pada masing-masing metode *cluster*. Nilai *silhouette coefficient* yang tinggi menunjukkan bahwa *cluster* semakin baik untuk mengelompokkan dokumen. *Cluster* yang baik merupakan *cluster* yang mampu mengelompokkan dokumen/*tweet* sehomogen mungkin. Tabel 4.14 dan 4.15 menunjukkan nilai *silhouette coefficient* untuk masing-masing metode *clustering* yang digunakan dalam penelitian ini.

Tabel 4.14 Evaluasi *Cluster* Terbaik pada Data *E-wallet* Go-Pay

Metode	*Sillhoeuette Coefficient*
K-Means	0,246
LDA	0,640
K-Means dengan LDA	0,659

Tabel 4.14 menunjukkan bahwa nilai *silhouette coefficient* untuk *cluster* dengan metode *K-Means* pada data *e-wallet* Go-Pay adalah sebesar 0,246. Metode LDA memperoleh nilai *silhouette coefficient* 0,640, sedangkan metode *K-Means* dengan LDA memperoleh nilai *silhouette coefficient* yang lebih besar yaitu 0,659. Hal ini menunjukkan bahwa penggunaan LDA mampu meningkatkan kebaikan model *clustering K-Means*. Sehingga, metode *clustering* terbaik pada data *tweet e-wallet* Go-Pay adalah metode *cluster K-Means* dengan LDA.

Tabel 4.15 Evaluasi *Cluster* Terbaik pada Data *E-wallet* OVO

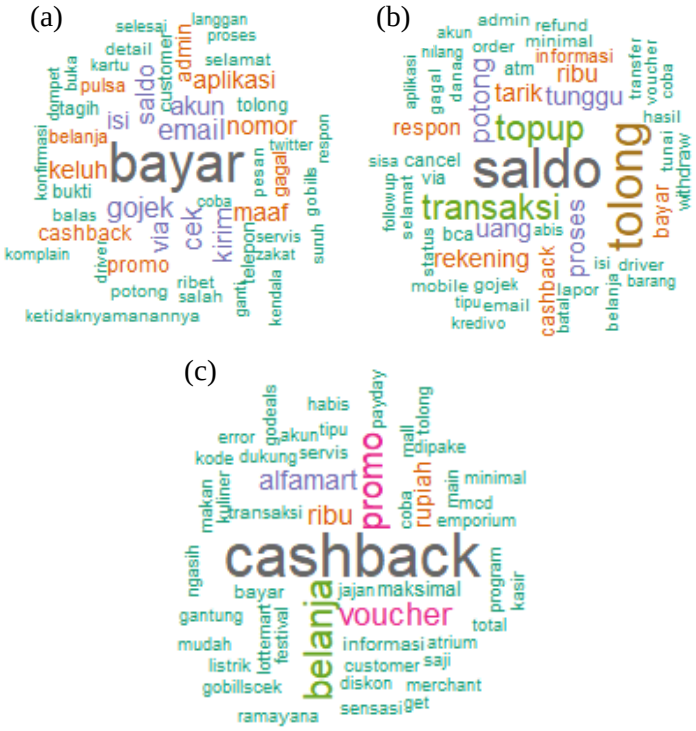
Metode	***Sillhoeuette Coefficient***
K-Means	0,131
LDA	0,394
K-Means dengan LDA	0,639

Tabel 4.15 menunjukkan nilai *silhouette coefficient* untuk *cluster* dengan metode *K-Means*, LDA dan *K-Means* dengan LDA pada data *e-wallet* OVO. Nilai *silhouette coefficient* untuk metode *K-Means* adalah sebesar 0,131. nilai *silhouette coefficient* untuk LDA adalah 0,394, sedangkan nilai *K-Means* dengan LDA adalah 0,639. Berdasarkan data tersebut menunjukkan bahwa pengguanaan LDA mampu meningkatkan kebaikan model *clustering K-Means*. Sehingga, pada data *e-wallet* OVO metode *cluster* terbaik yang digunakan adalah metode *cluster K-Means* dengan LDA.

4.3 Visualisasi *Tweet* Berdasarkan *Cluster* Terbaik

Berdasarkan metode terbaik yang telah diperoleh pada subbab sebelum, yaitu metode *K-Means* dengan *Latent Dirichlet Allocation* (LDA), selanjutnya pada subbab ini dilakukan visualisasi pada setiap *cluster* yang diperoleh menggunakan *wordcloud*. Visualisasi dilakukan guna mengetahui karakteristik dari *tweet* yang ditujukan kepada akun Twitter *customer service e-wallet* Go-Pay dan OVO. Besarnya ukuran *font* suatu kata menandakan frekuensi kemunculan dari kata yang bersangkutan. Dengan demikian, semakin besar ukuran *font* suatu kata maka frekuensi kemunculan kata tersebut semakin tinggi atau semakin sering muncul. Berikut

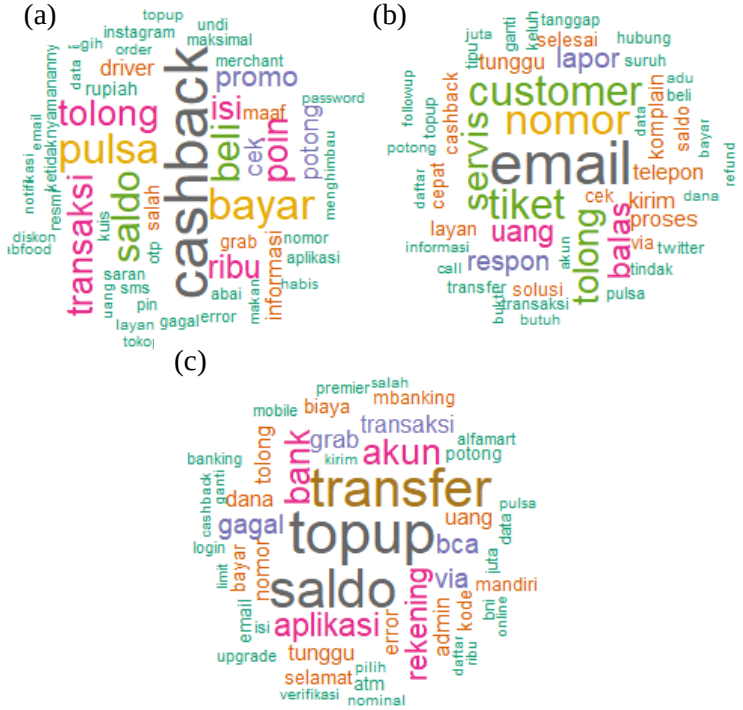
merupakan *wordcloud* dari setiap *cluster* untuk Twitter *e-wallet* Go-Pay pada @gopayindonesia.



Gambar 4.11 *WordCloud Cluster* 1 (a), *Cluster* 2 (b) dan *Cluster* 3 (c) Data *E-wallet* Go-Pay

Pada Gambar 4.11 yang menampilkan *word cloud* dari *cluster* 1, *cluster* 2, dan *cluster* 3 untuk data *e-wallet* Go-Pay menunjukkan bahwa pada *cluster* 1 terdapat kata-kata "bayar", "gojek", "cek", "e-mail". Kata-kata ini berhubungan dengan pembayaran menggunakan Go-Pay seperti apakah suatu layanan dapat dibayar dengan Go-Pay. Kemudian tampak pada *cluster* 2 terdapat kata-kata yaitu kata "saldo", "tolong", "transaksi", "topup", dan seterusnya. Kata-kata tersebut memiliki hubungan dengan permasalahan saldo pada Go-Pay seperti

permasalahan terkait saldo saat melakukan transaksi, permintaan tolong untuk melalukan pengecekan saldo, permasalahan terkait *top-up* saldo Go-Pay dan lainnya. Pada *cluster* 3 terdapat kata-kata "cashback", "voucher", "promo", "belanja", dan lainnya. Keseluruhan kata-kata ini berhubungan dengan pem-bahasan *cashback* seperti menanyakan berapa jumlah *cashback* yang diperoleh saat belanja, promo *cashback* yang diberikan Go-Pay, dan seterusnya. Sehingga, berdasarkan *wordcloud* terdapat tiga permbahasan utama terkait penggunaan layanan Go-Pay yaitu membahas tentang metode pembayaran dengan Go-Pay, saldo Go-Pay dan *cashback* Go-Pay.

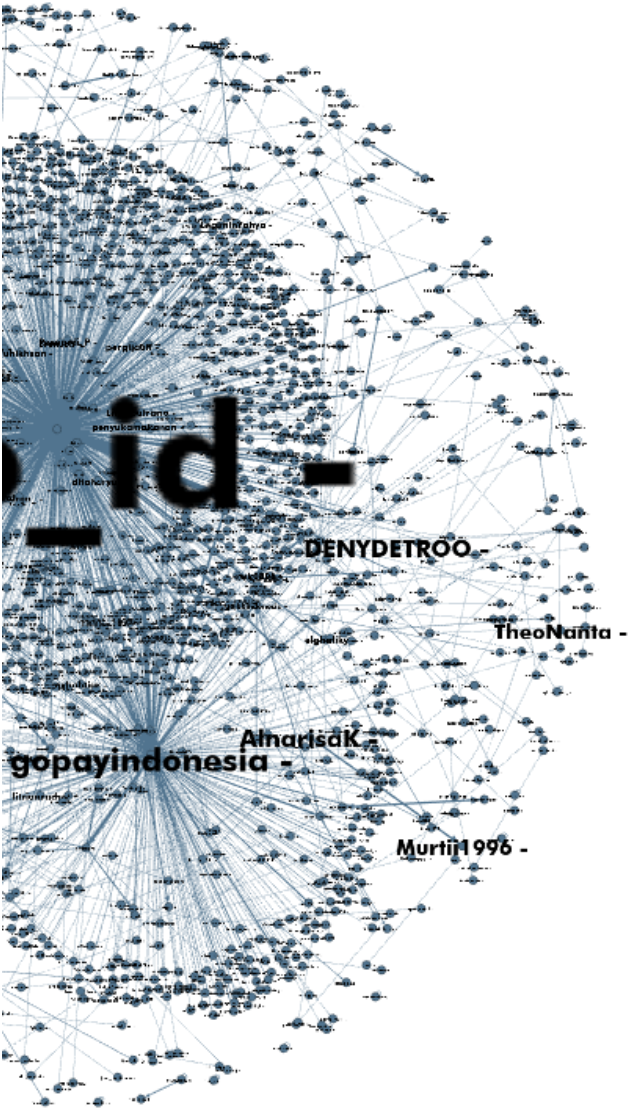


Gambar 4.12 *WordCloud Cluster* 1 (a), *Cluster* 2 (b) dan *Cluster* 3 (c) Data *E-wallet* OVO

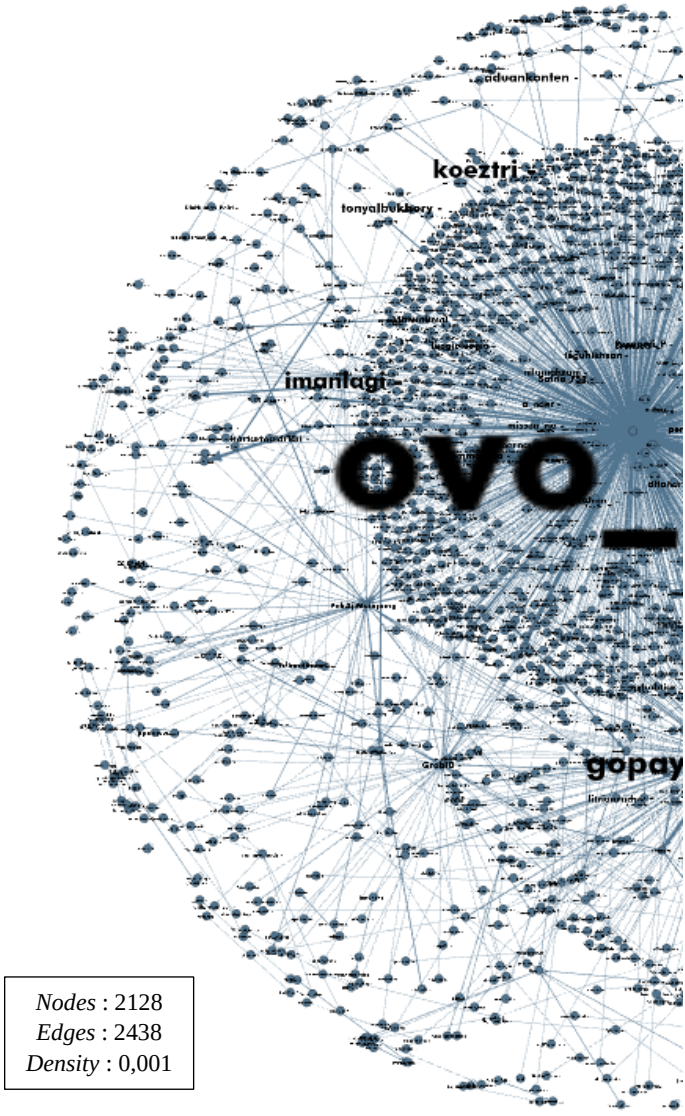
Pada Gambar 4.12 menunjukkan *wordcloud* dari *cluster* 1, *cluster* 2, dan *cluster* 3 untuk data *e-wallet* OVO. Pada *cluster* 1 terdapat kata-kata "cashback", "beli", "bayar", dan lainnya. Kata ini memiliki hubungan arti dengan *cashback* yang diberikan oleh OVO seperti *cashback* yang diberikan ketika melakukan pembayaran, dan lainnya. Pada *cluster* 2 terdapat kata-kata "email", "customer", "servis", "tiket", "nomor", "tolong", "respon", dan seterusnya. Kata-kata tersebut berhubungan dengan pelayanan *customer service* OVO tentang pelanggan yang menanyakan bagaimana kelanjutan aduan yang pernah telah disampaikan melalui *email customer service* OVO, lalu nomor tiket terkait aduan yang disampaikan, respon dari *customer service*, dan seterusnya. Pada *cluster* 3 terdapat kata-kata yaitu kata "transfer", "topup", "saldo", "rekening", "akun", "bank", dan lainnya. Kata-kata tersebut memliki hubungan dengan pembahasan saldo seperti *top up* saldo, transfer saldo, saldo pada akun dan rekening, dan seterusnya. Berdasarkan ketiga *wordcloud* tersebut maka *tweet* terkait penggunaan *e-wallet* OVO memiliki bahasan tiga hal yaitu *cashback* OVO, *e-mail customer service* OVO, dan saldo OVO.

4.4 *Social Network Analysis* (SNA) Data *Tweet E-Wallet* Go-Pay dan OVO

Setelah dilakukan analisis *clustering* dengan menggunakan metode *K-Means*, *Latent Dirichlet Allocation* dan *K-Means* dengan *Latent Dirichlet Allocation* maka selanjutnya dilakukan analisis dengan menggunakan *Social Network Analysis* (SNA) untuk mengetahui informasi mengenai Twitter perusahaan *e-wallet* Go-Pay dan OVO dengan rentang waktu 10 Februari 2019 hingga 14 Maret 2019. Informasi yang didapatkan dari jejaring sosial/*platform* Twitter kurang dapat memberikan gambaran struktur komunikasi dan tingkat partisipasi dari setiap pelanggan. Oleh karena itu diperlukan suatu metode yang dapat menilai atau memeriksa pola interaksi pelanggan *e-wallet* Go-Pay dan OVO. Berikut adalah visualisasi SNA data *e-wallet* Go-Pay dan OVO yang ditunjukkan pada Gambar 4.13.



Gambar 4.13 Visualisasi *Social Network Analysis* (SNA) Data *E-wallet* Go-Pay dan OVO



Gambar 4.13 (Lanjutan) Visualisasi *Social Network Analysis* (SNA) Data *E-wallet* Go-Pay dan OVO

Dapat diketahui bahwa pada Gambar 4.13 didapatkan jumlah *nodes*, *edges* dan *density* masing-masing adalah sebanyak 2128 *nodes*, 2438 *edges* dan *density* sebesar 0,001. *Density* adalah kepadatan *graph* suatu *network* yang menunjukkan jumlah hubungan yang hadir dalam suatu kelompok. Ketika menghitung *density*, SNA melihat bagaimana erat hubungan seseorang yang satu dengan yang lain. Nilai *density* sebesar 0,001 menunjukkan bahwa *network e-wallet* Go-Pay dan OVO memiliki kepadatan yang kecil atau renggang. *Density* yang kecil atau renggang ini terjadi karena rendahnya interaksi antar *account*, baik berupa *mention*, *quote retweet*, atau *reply* yang dilakukan antar *node* dalam jaringan. Berikut adalah hasil analisis *centrality* dari Gambar 4.13 yang ditunjukkan pada Tabel 4.16.

Tabel 4.16 Analisis *Centrality* dari Jaringan *E-wallet* Go-Pay dan OVO

Centrality	***Username***	***Score***
Degree Centrality	@ovo_id	992
	@gopayindonesia	309
	@GrabID	58
Closeness Centrality	@ovo_id	0,829
	@imanlagi	0,8
	@akhlishdiaz	0,8
Betweeness Centrality	@ovo_id	30811
	@koeztri	2130
	@istitoha	1330,45

Analisis *centrality* pada Tabel 4.16 bertujuan untuk menemukan *account* yang paling berperan dalam sebuah *network*. *Metric* yang digunakan dalam penentuan *centrality* ini adalah *degree centrality*, *closeness centrality* dan *betweenness centrality*. Analisis *degree centrality* menentukan *account* yang paling berperan berdasarkan banyaknya *edge* atau hubungan yang terjadi antara sebuah *node* dengan *node* yang lainnya. Akun @ovo_id memiliki nilai *degree centrality* tertinggi yaitu 992. Hal ini dapat diartikan bahwa @ovo_id memiliki hubungan atau interaksi yang berupa *mention*, *quote retweet*, atau *reply* antar *node* lain sebanyak 992 kali. Sedangkan, yang kedua adalah *account* @gopayindonesia dengan nilai *degree centrality* sebesar 309, dan yang ketiga adalah *account* @GrabID dengan nilai *degree*

centrality sebesar 58. Selain kedua account *e-wallet* Go-Pay dan OVO, *account* @GrabID memiliki dampak yang juga besar pada *graph* karena diketahui bahwa *account* tersebut merupakan *account* resmi salah satu penyedia layanan *e-wallet* OVO yang juga sebuah penyedia jasa taksi online dengan salah satu sistem pembayaran menggunakan *e-wallet* OVO.

Analisis *closeness centrality* digunakan untuk melihat *node-node* yang dapat menjangkau *node* lainnya dengan jalur yang lebih pendek. Semakin mendekati 1 atau sama dengan 1, maka semakin dekat *node* tersebut dengan *node* lain. Agar lebih mudah dipahami dan mengurangi bias, maka semua *node* yang memiliki nilai *closeness centrality* sebesar 1 tidak diperhatikan. Terdapat tiga *account* dengan nilai *closeness centrality* tertinggi, yaitu @ovo_id, @imanlagi, dan @akhlishdiaz. Nilai *closeness centrality* untuk peringkat pertama yaitu 0,829 lalu peringkat kedua dan ketiga memiliki nilai yang sama yaitu sebesar 0,8. Peringkat kedua pada analisis *closeness centrality* diduduki oleh *account* @imanlagi yang merupakan diketahui seorang *influencer* pada *platform* Twitter. *Account* menjadi peringkat kedua karena sering melakukan promo yaitu dengan cara mention kepada *account e-wallet* OVO. Hal ini merupakan pemilihan yang tepat oleh perusahaan *e-wallet* OVO dalam menentukan *promoter* atau *account* yang dapat membantu perusahaan melakukan *promotion* layanan *e-wallet* sehingga *account* @imanlagi dapat dengan cepat menyebarkan info mengenai *e-wallet* OVO kepada *followers account* tersebut yang notabene seorang *influencer* memiliki jumlah *followers* banyak. Sedangkan account ketiga pada analisis *closeness centrality* adalah @akhlishdiaz, diketahui bahwa *account* tersebut sering melakukan *mention* untuk melayangkan pertanyaan maupun keluhan terhadap *e-wallet* OVO. *Account* tersebut terlihat aktif melayangkan pertanyaan/ keluhan sebanyak total 3 dan mendapatkan *feedback* oleh *e-wallet* OVO. Selisih nilai peringkat pertama dan kedua, ketiga yaitu sebesar 0,029. Hal ini mengakibatkan *node-node* dengan nilai *closeness centrality* 0,829 lebih cepat dan lebih mudah dalam berkomunikasi dengan *node* lain ketika melakukan *mention*, *quote retweet*, atau *reply* tanpa melalui banyak perantara yang dilalui.

Analisis *betweenness centrality* bertujuan untuk mengetahui posisi *node* dalam *network*, dimana *node* tersebut tidak boleh hilang. Jika *node* tersebut hilang maka akan terjadi gangguan komunikasi dalam *network*. *Node* dengan *username* @ovo_id memiliki nilai *betweenness centrality* tertinggi dalam *network*. Artinya, @ovo_id adalah *account* yang mendominasi sebagai penghubung atau jembatan dari seluruh aliran informasi dalam percakapan dibandingkan dengan akun @gopayindonesia. Hal tersebut juga dapat terlihat berdasarkan Gambar 4.13 bahwa jaringan dari akun @ovo_id terlihat lebih besar daripada akun dari @gopayindonesia.

(Halaman ini sengaja dikosongkan)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut.

1. Sejumlah 1182 *tweet* yang ditujukan kepada akun Twitter *customer service e-wallet* Go-Pay (@gopayindonesia) pada tanggal 10 Februari 2019 hingga 14 Maret 2019 setelah dilakukan *preprocessing* kata yang paling banyak muncul adalah kata "cashback" sebanyak 259 kali dan selanjutnya diikuti oleh kata "belanja", "bayar", "bayar", "saldo", "promo", "voucher", "tolong", "dm", "ribu", dan "transaksi". Sedangkan sejumlah 3244 *tweet* pada yang ditujukan kepada akun Twitter *customer service e-wallet* OVO (@ovo_id) kata yang paling banyak muncul adalah kata "saldo" sebanyak 462 kali dan selanjutnya diikuti oleh kata "cashback", "tolong", "topup", "e-mail", "transfer", "nomor", "bayar", "transaksi", dan "pulsa".
2. Metode *K-Means* dengan *Latent Dirichlet Allocation* (LDA) adalah metode terbaik untuk mengelompokkan *tweet* yang ditujukan kepada akun Twitter *customer service e-wallet* Go-Pay (@gopayindonesia) maupun OVO (@ovo_id). Metode ini dipilih karena memiliki nilai *silhouette coefficient* yang lebih tinggi dibandingkan dengan metode *K-Means*. Nilai *silhouette coefficient* yang diperoleh masing-masing sebesar 0,659 dan 0,639, artinya *cluster* yang dihasilkan telah kompak dan telah terpisahkan dengan baik antara *cluster* yang satu dengan yang lainnya. Analisis c*lustering* dengan metode terbaik memperoleh tiga kategori *tweet* yang ditujukan kepada akun Twitter *customer service e-wallet* Go-Pay (@gopayindonesia) dan tiga kategori *tweet* yang ditujukan kepada akun Twitter *customer service e-wallet* OVO (@ovo_id).
3. Melalui *wordcloud* berdasarkan *cluster* dengan metode terbaik, dapat diketahui visualisasi kategori dan kata yang sering

muncul dari *tweet* pengguna kepada akun Twitter *customer service* pada akun *e-wallet* Go-Pay (@gopayindonesia) dan akun *e-wallet* OVO (@ovo_id).

4. Berdasarkan hasil *graph* dari *Social Network Analysis* (SNA) terlihat bahwa akun Twitter *customer service e-wallet* OVO (@ovo_id) mendominasi sebagai penghubung atau jembatan dari seluruh aliran informasi.

5.2 Saran

Saran yang diberikan oleh peneliti terkait analisis yang telah dilakukan adalah sangat diperlukan ketelitian dan pemahaman lebih mendalam mengenai permasalahan saat melakukan *preprocessing data* karena hasil yang diperoleh dari proses tersebut, sangat mempengaruhi hasil analisis *cluster* yang terbentuk. Diperlukan juga analisis mengenai penerapan teknik kombinasi kata lainnya dalam penelitian selanjutnya. Sedangkan untuk melakukan *Social Network Analysis* (SNA) disarankan untuk melakukan analisis menggunakan data *crawling tweet* dengan menggunakan *keyword* permasalahan yang ingin dianalisis dan bukan merujuk ke salah satu akun Twitter, agar analisis yang dihasilkan lebih baik dan *relevant*. Saran yang dapat diberikan kepada perusahaan penyedia layanan *e-wallet* adalah mempertimbangkan hasil proses *tweet* berdasarkan *cluster* yang terbentuk. *Cluster* untuk *e-wallet* Go-Pay terdapat 3 yaitu mengenai pembayaran, transaksi saldo, dan juga layanan *cashback*. Sedangkan *cluster* untuk *e-wallet* OVO juga terdapat 3 yaitu berkenaan tentang layanan *cashback*, *e-mail customer service*, dan *top-up* saldo. Saldo dan *cashback* merupakan dua permasalahan yang sering muncul di kedua perusahaan *e-wallet* Go-Pay dan OVO, maka dari itu disarankan kepada perusahaan untuk mempertimbangkan hasil penelitian sehingga dapat mempermudah dalam menangani maupun meningkatkan layanan *e-wallet* di Indonesia.

DAFTAR PUSTAKA

Abualigah, L. M., Khader, A. T., & Betar, M. A. (2016). Unsupervised Feature Selection Technique Based on Genetic Algorithm for Improving the Text Clustering. *International Conference on Computer Science and Information Technology (CSIT).* 7(1), 5-6.

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M. M., & Williams, H. E. (2007). Stemming Indonesian: A Confix-Stripping Approach. *ACM Transactions on Asian Language Information Processing (TALIP).* 6(4), 1-33.

Agustina, A. (2017). *Analisis Dan Visualisasi Suara Pelanggan Pada Pusat Layanan Pelanggan Dengan Pemodelan Topik Menggunakan Latent Dirichlet Allocation (LDA) Studi Kasus: PT. Petrokimia Gresik*. Surabaya: Institut Teknologi Sepuluh Nopember.

Alfina, T., Santosa, B., & Barakbah, A. R. (2012). Analisa Perbandingan Metode Hierarchical Clustering, K-Means dan Gabungan Keduanya dalam Cluster Data. *Jurnal Teknik ITS*. 1(1), 1-5.

Amoroso, D. L., & Magnier-watanabe, R. (2012). Building a Research Model for Mobile Wallet Consumer Adoption: The Case of Mobile Suica in Japan. *Journal of Theoretical and Applied Electronic Commerce Research*. 7(1), 94–110.

Ariadi, D., & Fithriasari, K. (2015). Klasifikasi Berita Indonesia Menggunakan Bayesian Classification dan Support Vector Machine dengan Confix Stripping Stemmer. *Jurnal Sains dan Seni ITS*. 4(2). 20.

Bank Indonesia. (2011). *Sistem Pembayaran di Indonesia*. Retrieved Februari 28, 2019, from https://www.bi.go.id/id-/sistem-pembayaran/di-indonesia/Contents/Default.aspx

Blei, D. (2012). Probabilistic Topic Models. *Communications of the ACM*. 55(4), 77–84.

Campbell J.C, Hindle, A. & Stroulia, E. (2014). Latent Dirichlet Allocation: Extracting Topics from Software Engineering Data. In M. Bird & T. Zimmermann (Eds.). *The Art and Science of Analyzing Software Data (1 ed.).* Burlington, MA: Morgan Kaufmann.

Castella, Quim, Sutton, & Charles. (2014). Word Storm: Multiples of Wordclouds for Visual Comparison of Documents. In Proceedings *of the 23rd International Conference on World Wide Web.* 665–676.

Cavnar, William & Trenkle, J. (1994). N-Gram-Based Text Categorization. Proceedings of the *Third Annual Symposium on Document Analysis and Information Retrieval*. 161-175.

Daniel, J. and James, H., M. (2014). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. New Jersey: Prentice Hall.

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook : Advanced Approaches in Analyzing Unsrtructed Data.* New York: Cambridge University Press.

Freeman, L., C. (1979). *Centrality In Social Networks: I. Conceptual Clarification*, *Social Networks*. 1(3), 215-239.

Go-Jek. (2015). *Profil Go-Jek Indonesia.* Jakarta: Go-Jek Indonesia.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques.* USA: Morgan Kaufmann.

Herwanto, G. B. (2018). Document Clustering dengan Latent Dirichlet Allocation dan Ward Hierarichal Clustering. *Jurnal Pseudocode Universitas Gajah Mada*. 5(2), 8.

Hidayatullah, Fakhri & Maarif, Muhammad. (2017). Pre-processing Tasks in Indonesian Twitter Messages. *Journal of Physics Conference Series*. 10(5), 35-47.

Jannah, S. Z., Fithriasari, K., Prastyo, D. D., & Iriawan, N. (2018). Text Mining for Identifying and Visualizing Topics of Citizen Opinion in Media Centre Surabaya. *International Conference on Theretical and Applied Statistics*, p. 82.

Johnson, R. A., & Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis (6th ed.)*. United States of America: Pearson Prentice Hall.

Juwita, R., & Irhamah. (2018). *Klasifikasi Kelas Risiko Pasien Pneumonia menggunakan Regresi Logistik Ordinal, Hybrid Regresi Logistik Ordinal - Algoritma Genetika dan Naive Bayes Classification.* Surabaya: Institut Teknologi Sepuluh Nopember.

MacQueen, J. B., 1967. *Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th.* Berkeley: University of California Press

Narayana, B. L., & Kumar, S. P. (2015). A New Clustering Technique on Text. *International Journal of Science Engineering and Advance Technology (IJSEAT).* III(3), 69-71.

Nooy, W., Mrvar, A., & Batagelj, V. (2005) *Exploratory Social Network Analysis with Pajek*. Cambridge: Cambridge University Press.

OVO. (2015). *About Us OVO.* Jakarta: OVO.

Putra, I. M. K. B., & Kusumawardani, R. P. (2017). *Clustering and Visualizing Surabaya Citizen Aspirations by Using Text Mining.* Surabaya: Jurnal Teknik Institut Teknologi Sepuluh Nopember.

Scott, J. (1992). *Social Network Analysis*. Newbury Park CA: Sage.

Srivastava, A., & Sahami, M. (2009). *Text Mining Classification, Clustering, and.* USA: Taylor and Francis Group, LLC.

Tala, F. Z. (2003). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. M.Sc*. Amsterdam: Universiteti van Amsterdam.

Talib, R., Hanif, M. K., Ayesha, S., & Fatima, F. (2016). Text Mining: Techniques, Applications and Issues. *International Journal of Advanced Computer Science and Applications (IJACSA)*. 7(11), 414-418.

Waegel, D. (2006) The Development of Text-Mining Tools and Algorithms. *Distinguished Honors in Computer Science*. Ursinus College.

Zhang, C., & Fang, Z. (2013). An Improved K-Means Clustering Agorithm. *Journal of Information & Computational Science*. 10(1), 193-199.

LAMPIRAN

Lampiran 1. Data *Tweet* Akun Twitter *Customer Service E-wallet* Go-Pay

***Tweet* ke**	***Text***
1	banyak diskon non, yuk merapat @gopayindonesia @PromoMakanan @gojekindonesia https://t.co/5DNDNj6E3K
2	@gojekindonesia Keterlaluan dan kuno banget ini @gopayindonesia narik dana saja bisa 2x24 jam kayak jaman batu aja kalian
3	@gopayindonesia saya sudah DM tolong dibantu donk min
4	@gopayindonesia tolong ya min, dana saya sudah 1x24 jam belum masuk ke rekening penarikan dari saldo gopay. Sudah saya DM barbuk
5	@gopayindonesia maximum TopUp saldo gopay berapakah ya min per-hari
⋮	⋮
1177	Ini lagi ada promo apa sih ? Kok bisa cashback 100% ? <U+0001F924> Tau gini gua belanja banyak tadi.. @gojekindonesia @gopayindonesia @alfamart https://t.co/FmEaWtK7Ie
1178	@ismissyouu @gojekindonesia @gopayindonesia @alfamart khusus buat sobat misqin <U+0001F606>
1179	@gojekindonesia @gopayindonesia tiket 07206978, sya punya 2 device kenapa dhitung 1 device? Padahal merchant lain fine aja tuu, CB tetep masuk krn 2 device beda & 2 Akun beda juga Lah solaria doang yg g masuk Aneh aturannya! Sya bsa kash bukt
1180	@samuelchrstns @gojekindonesia @gopayindonesia @alfamart Kok gokil sihh wkwk
1181	@nainanina wes tak mention seko 7 jam lalu, tetep ra respon, wqwq...gimana @linkaja mo ngalahin @ovo_id dan @gopayindonesia nya @gojekindonesia kl keluhanku di balas minta maaf aja juga engga :(
1182	@gopayindonesia @akhlishdiaz @gojekindonesia Iyaa perlu ditingkatkan untuk salah satu hal ini. Masalah satu ini ovo jauh lebih unggul. Transfer dana ovo ke rekening mana aja tanpa charge. Gopay 2.500

Lampiran 2. Data *Tweet* Akun Twitter *Customer Service E-wallet* OVO

***Tweet* ke**	***Text***
1	@ZacRydo @ShineCard @ovo_id saya sih di 11x24 jam sudah ada diberikan solusi, ktnya sih uangnya akan di trf dan masuk ke rek bank kita 1 hari kerja....jd bsk. Kita liat bsk yah
2	@UwaisyMH @MediaKonsumenID @ovo_id Sama! Case: 1083190 udah 10Ã—24jam
3	@ShineCard @ovo_id Saya aja sudah 10Ã—24 jam belum ada kabar
4	ini komplain masalah uang transfer blm kelar2. Uang saya 1 juta gantung gak jelas kemana. Tanya @ovo_id dilempar ke email, di email jawabnya masih proses yg gak jelas service levelnya sampai kpn. Case#1083190 kapok deh ama OVO cc @ojkindonesia
5	ini sampai kapan yah hrs menunggu, sudah 1 minggu uangku blm balik2. Kirim email ke cs @ovo_id blm ada balasannya case aku https://t.co/16x2IFuh7q
⋮	⋮
3240	@GrabID @ovo_id min, saya belum order tapi ko saldo saya udah kekurangin min? Tolong tindak lanjutnya sedih woy bisa buat beli nasi padang ini 16rebu :(((https://t.co/6p1pWYom0n
3241	Ini gmn sih @GrabID. Ga dpt driver krn katanya busy kok ovo poinnya udh dipotong. Jgn gt lah cari duitnya @ovo_id @GrabID
3242	@ovo_id isi pulsa lewat ovo ,pulsa nggak masuk No referensi OG11150048 Uang juga belum balik ke OVO
3243	@ovo_id ini sy isikan pulsa tmn status berhasil di OVO tpi pulsa gak masuk2. Ini sdh ke dua kalinya kayak gini. Kapok2 deh beli pulsa di ovo https://t.co/orbjk5ZOXb
3244	Subuh bangun kedengeran suara ujan. Kudu ke priok lagi karna ada audit. So harus memilih ngegrab ke ragunan dari rumah. Terima kasih Tuhan udah ciptain ovopoints<U+0001F602> @GrabID @ovo_id https://t.co/VcL8tzr5ys

Lampiran 3. *Syntax* Karakteristik Data

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
from collections import Counter

df = pd.read_csv("gopay.csv")
text=df['text']

mpl.rcParams['figure.figsize']=(12.0,6.0)
mpl.rcParams['font.size']=10
mpl.rcParams['savefig.dpi']=100
mpl.rcParams['figure.subplot.bottom']=.1

kata = [s.lower().split() for s in text if s]
noline_ = [sublist for l in kata for sublist in l]

counts1 = dict(Counter(noline_).most_common(10))
labels1, values1 = zip(*counts1.items())

indSort1 = np.argsort(values1)[::-1]
labels1 = np.array(labels1)[indSort1]
values1 = np.array(values1)[indSort1]
indexes1 = np.arange(len(labels1))

mybar = plt.bar(indexes1, values1, color='#E9967A')
# get rid of the frame
for spine in plt.gca().spines.values():
    spine.set_visible(False)
# remove all the ticks and directly label each bar with respective
value
plt.tick_params(top='off', bottom='off', left='off', right='off',
labelleft='off', labelbottom='on')
# direct label each bar with Y axis values
```

```
for bari in mybar:
    height = bari.get_height()
    plt.gca().text(bari.get_x() + bari.get_width()/2,
bari.get_height() + 40, str(int(height)), ha='center', color='black',
fontsize=10)
# add labels
plt.xticks(indexes1, labels1)
plt.show()
```

Lampiran 4. *Syntax Preprocessing* Data

```
import pandas as pd
import re
import nltk
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from collections import OrderedDict

data = pd.read_csv("gopay.csv")
text = data['text']

#Menghapus Link
data_link = []
for line in text:
    result = re.sub(r"http\S+", " ", line)
    data_link.append(result)

#Menghapus Retweet
data_rt = []
for line in data_link:
    result = re.sub(r"RT", " ", line)
    data_rt.append(result)

#Menghapus Username
data_uname = []
for line in data_rt:
    result = re.sub(r"@\S+", " ", line)
    data_uname.append(result)

#Menghapus Baris Baru
data_line=[]
for line in data_uname:
    result=re.sub("\n"," ",line)
```

```
    data_line.append(result)

#Menghapus Angka
data_num=[]
for line in data_line :
    result=re.sub("\d"," ",line)
    data_num.append(result)

#Menghapus Hashtag
data_hashtag=[]
for line in data_num :
    result=re.sub(r"#\S+","",line)
    data_hashtag.append(result)

#Menghapus Emoticon
data_emoticon=[]
for line in data_hashtag :
    result = re.sub(r'<.*?>',"",line)
    data_emoticon.append(result)

#Menghapus Punctuation
data_punc=[]
for line in data_emoticon :
    result=re.sub(r"[^\w\s]"," ",line)
    data_punc.append(result)

#Menghapus Spasi Berlebih
data_doublespace=[]
for line in data_punc :
    result=re.sub(r'\s+',' ',line)
    data_doublespace.append(result)

#Case Folding
data_casef = []
for line in data_doublespace:
```

```
    a = line.lower()
    data_casef.append(a)

#Stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()
df_stemmed = map(lambda x: stemmer.stem(x), data_casef)
data_stemmed = list(df_stemmed)

data_stemmed_gopay=pd.DataFrame(data_stemmed)
data_stemmed_gopay.to_csv('data_stemmed_gopay.csv',
header=True)

#Sinonim Kata
kata = {"trf":"transfer","transf ":"transfer","rek.":"rekening", "rek
":"rekening", "gak":"tidak","service":"servis",
        "drive ":"driver", "driverr":"driver","terimakasih":"terima
kasih","trims":"terima kasih","cepet":"cepat",
        "complaint":"komplain", "dtg":"datang", "dmn":"dimana",
"diem":"diam", "errorrr":"error", "diklaksonin":"klakson",
        "gaada":"tidak ada","gabisa":"tidak bisa", "gatau":"tidak
tahu","hour":"jam","komplen":"komplain","lwt":"lewat",
        "trimakasih":"terima kasih","makasih":"terima kasih",
"makas":"terima kasih", "mhn":"mohon", "mksh":"terima kasih",
        "mlm":"malam", "malem":"malam", "mnit":
"menit","mnt":"menit","msk":"masuk","nungguin":"tunggu",
"nunggunya":"tunggu",
        "nunggu":"tunggu", "nyambung":"sambung",
"nyampe":"sampai", "ojol":"ojek online", "renti":"berhenti",
        "responnya":"respon","supirnya":"sopir",
"supir":"sopir","tambahin":"tambah",
"tdk":"tidak","tetep":"tetap",
        "tgl":"tanggal", "thanks":"terima kasih", "thank":"terima
kasih", "thank you":"terima kasih", "thx":"terima kasih",
        "tks":"terima kasih", "tlng":"tolong", "tlg":"tolong",
```

"tq":"terima kasih","terimaqasih":"terima kasih",
"trima": "terima kasih", "trims": "terima kasih", "terimakasih":"terima kasih","twittet":"twitter",
"cash back":"cashback", "cash ":"cashback", "m banking":"mbanking","acct":"akun","account":"akun","no.":"nom or",
"nomer":"nomor","wd ":"withdraw ","witdraw":"withdraw","atmnya":"atm",
"nomornya":"nomor","nmr":"nomor","i banking":"ibanking", "go deals":"godeals","go jek":"gojek","go-jek":"gojek",
"top up":"topup","top-up":"topup", "go pay":"gopay","cashbacknya":"cashback", "casback":"cashback",
"cashbak":"cashback","slorespon":"slowrespon", "kesbek":"cashback", "kesbeknya":"cashback",
"vocer":"voucher", "voucer":"voucher", "vocher":"voucher","vouchernya":"voucher", "fail":"gagal", "failed":"gagal",
"ticket":"tiket","rb":"ribu","tansaction":"transaksi", "notif":"notifikasi", "notifikasinya":"notifikasi",
"notifnya":"notifikasi", "notification":"notifikasi","msk":"masuk","app ":"aplikasi ", "application":"aplikasi",
"appnya":"aplikasi","bls":"balas", "promonya":"promo", "pedes":"pedas", "bales":"balas","discount":"diskon",
"transferer":"transfer","tari":"tarik","maks ":"maksimal ","maximum":"maksimal", " max":" maksimal","jsm":"jam",
"gopaynya":"gopay","gopaypayday":"gopay payday","duit":"uang","promony":"promo",

"ovonya":"ovo","brg":"barang","whatsaplikasi":"whatsapp","trx": "transaksi","tranfer":"transfer","toped":"tokopedia",
"follow up":"followup","tokennya":"token","tlp ":"telepon ","telp ":"telepon ", "telpon":"telepon",

"teleponon":"telepon","ditelpon":"telpon","bbrp":"beberapa","brp":"berapa"," rp":" rupiah","nanya":"tanya",
"tanyanya":"tanya","membeli":"belanja","beli":"belanja",

"telfon":"telepon","tf":"transfer","temen":"teman","tarikk":"tarik","system":"sistem","service":"servis","response":"respon",

"rekeningbank":"rekening","rekeningtuju":"rekening","rebu":"ribu","premier":"premiere","point":"poin","points":"poin",

"piye":"bagaimana","perhati":"perhatian","ovopoints":"ovopoint","notifikasiikasi":"notifikasi","jarungan":"jaringan",
"nila":"nilai","merespon":"respon"," mayan":"lumayan","maksimalimal":"maksimal","maksimalimum":"maksimal",
"maksimalud":"maksimal","lelet":"lemot","jt":"juta"," info":" informasi","ig ":"instagram ","hub":"hubung",
"haplikasiy":"aplikasi","hape":"hp","gagaled":"gagal","feb":"februari ","eror":"error","debet":"debit",
"cust ":"customer","cs":"customer servis","mrchant":"merchant","dibalas":"balas","teriterima":"terima",

"cinemaksimalx":"cinemaxx","center":"centre","apk":"aplikasi","adminnya":"admin","adm":"admin","adminin":"admin",
"aji mumpung":"ajimumpung", "alfa ":"alfamart "," ilang":"hilang","minimum":"minimal", "tarikk":"tarik",
"driverku":"driver","rekpon":"rekening ponsel","vchr":"voucher","cashbackga":"cashback","lamaa":"lama","rekg":"rekening",
"terimahkasih":"terima kasih","maintenancenya":"maintenance","servicenya":"servis","complain":"komplain",
"gopayyy":"gopay","kasij":"kasih","tulung":"tolong","qr code":"barcode","promox":"promo","lotte mart":"lottemart",

```
      "ilotte":"lottemart","lotte ":"lottemart ","lottema ":"lottemart
","transaction":"transaksi","bantuan":"tolong",

"bantu":"tolong","narik":"tarik","struknya":"struk","tolongin":"tol
ong","direfund":"refund","onlen":"online",
      "mksih":"terima
kasih","ngedm":"dm","tanyak":"tanya","development":"perkemba
ngan","hubungung":"hubung",

"kasihiiiii":"kasih","gopipay":"gopay","cbnya":"cashback","cb
":"cashback ","cb nya":"cashback",
      "go bills":"gobills","go ridenya":"goride","go
jek":"gojek","go food":"gofood","go pulsa":"gopulsa","go
tix":"gotix"}

def replace_all(text, dic):
   for i, j in dic.items():
      text = text.replace(i, j)
   return text

dic = OrderedDict(kata)
data_change = []
for line in data_stemmed:
   result = replace_all(line, dic)
   data_change.append(result)

data_change_gopay=pd.DataFrame(data_change)
data_change_gopay.to_csv('data_change_gopay.csv',
header=False)

#Stopwords
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
stopWords = set(stopwords.words('indonesian'))
```

```
stopword = open("idn stopwords.txt","r").read()
stopword = set(stopword.split())
not_stopword = {}
new_stopword = set([word for word in stopword if not word in not_stopword])

data_stop = []
for line in data_change:
   word_token = nltk.word_tokenize(line)
   word_token = [word for word in word_token if not word in stopword]
   data_stop.append(" ".join(word_token))

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = CountVectorizer(min_df=0.02)
x = vectorizer.fit_transform(data_stop)
DTM = pd.DataFrame(x.toarray(),columns = vectorizer.get_feature_names())
DTM['kalimat_tweet']=data_stop
DTM['sum_features'] = DTM.sum(axis=1)
DTM = DTM.loc[DTM['sum_features'] != 0].drop(['sum_features'], axis=1)
DTM.to_csv('DTM5_gopay.csv')

vectorizer = TfidfVectorizer(min_df=0.02)
vec = vectorizer.fit_transform(data_stop)
TFIDF = pd.DataFrame(vec.toarray(), columns = vectorizer.get_feature_names())
TFIDF['kalimat_tweet']=data_stop
TFIDF['sum_features'] = TFIDF.sum(axis=1)
TFIDF = TFIDF.loc[TFIDF['sum_features'] != 0].drop(['sum_features'], axis=1)
TFIDF.to_csv('TFIDF_gopay.csv')
```

Lampiran 5. *Syntax* Metode *K-Means*

```
library(dplyr)
library(tidyr)
library(caret)
library(RColorBrewer)
library(ggplot2)
library(factoextra)
library(NbClust)
library(wordcloud)
library(wordcloud2)
library(tm)
library(cluster)

TFIDF_gopay<-read.csv("C:/FIDYS/KULIAH/SMT 8/TUGAS
AKHIR/SYNTAX/PYTHON/TFIDF_gopay.csv")
df_clust_gopay<-TFIDF_gopay%>%select(-c(kalimat_tweet,
X))

##K-Means Evaluation
Kmeans <- function(data)
{
sil_coef <- matrix()
for (k in 2:10)
{
set.seed(12)
KMeans <- kmeans(data, centers = k, nstart = 100)
silcoef <- silhouette(KMeans$cluster, dist(data))
sil_coef[k] <- summary(silcoef)$avg.width
}
win.graph()
plot(sil_coef, xlab = "k", type = "b")
win.graph()
```

```
list(Sil_Coef = sil_coef)
}
Kmeans(df_clust_gopay)

set.seed(12)
kmeans_gopay<-kmeans(df_clust_gopay, 10, nstart = 20)
kmeans_gopay$size
silcoef <- silhouette(kmeans_gopay$cluster,
dist(df_clust_gopay))
summary(silcoef)$avg.width

TFIDF_gopay$clust<-kmeans_gopay$cluster
gopay_cluster<-TFIDF_gopay%>%select(c(kalimat_tweet,
clust))

#wordcloud
for (i in 1:10){
  docs<-
Corpus(VectorSource(gopay_cluster$kalimat_tweet[gopay_clust
er$clust==i]))
  dtm <- TermDocumentMatrix(docs)
  m <- as.matrix(dtm)
  v <- sort(rowSums(m),decreasing=TRUE)
  d <- data.frame(word = names(v),freq=v)
  wordcloud(words = d$word, freq = d$freq, min.freq = 1,
          max.words=50, random.order=FALSE, rot.per=0.35,
          scale=c(2,0.5), colors=brewer.pal(8, "Dark2"))
}
```

Lampiran 6. *Syntax* Metode *Latent Dirichlet Allocation* (LDA)

```
import numpy as np
import pandas as pd
import re, nltk, gensim

# Sklearn
from sklearn.decomposition import LatentDirichletAllocation, TruncatedSVD
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import GridSearchCV
from pprint import pprint

# Plotting tools
import pyLDAvis
import pyLDAvis.sklearn
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore",category=DeprecationWarning)

df = pd.read_csv("TFIDF_gopay.csv")
data = df['kalimat_tweet']

vectorizer = CountVectorizer(analyzer='word', min_df=0.02)
data_vectorized = vectorizer.fit_transform(data)

sil=[]
topik=[2, 3, 4, 5, 6, 7, 8, 9, 10]
for i in topik:
    lda_model = LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
            evaluate_every=-1, learning_decay=0.7, learning_method='online',
```

```
        learning_offset=10.0,          max_doc_update_iter=100,
max_iter=15,
        mean_change_tol=0.001,
        n_components=i,
        n_jobs=1,          n_topics=None,          perp_tol=0.1,
random_state=10,
        topic_word_prior=None,          total_samples=1000000.0,
verbose=0)
   lda_output = lda_model.fit_transform(data_vectorized)
   # Get dominant topic for each document
   dominant_topic      =      np.argmax(df_document_topic.values,
axis=1)
   ss = silhouette_score(data_vectorized, dominant_topic)
   sil.append(ss)
   print('topik :', i, 'Silhouette :', ss)

# Show graph
plt.figure(figsize=(12, 8))
plt.plot(topik, sil)
plt.title("Choosing Optimal LDA Model")
plt.xlabel("Num Topics")
plt.ylabel("Silhoutte")
plt.show()

#lda_model      =      LatentDirichletAllocation(n_components=3,
max_iter=15, learning_decay=0.5, random_state=10)
lda_model      =      LatentDirichletAllocation(batch_size=128,
doc_topic_prior=None,
        evaluate_every=-1,                learning_decay=0.7,
learning_method=None,
        learning_offset=10.0,          max_doc_update_iter=100,
max_iter=15,
        mean_change_tol=0.001,
        n_components=3,
        n_jobs=1,
```

```
        n_topics=None, perp_tol=0.1, random_state=10,
        topic_word_prior=None,        total_samples=1000000.0,
verbose=0)

lda_output = lda_model.fit_transform(data_vectorized)

# Show top n keywords for each topic
def  show_topics(vectorizer=vectorizer,  lda_model=lda_model,
n_words=20):
   keywords = np.array(vectorizer.get_feature_names())
   topic_keywords = []
   for topic_weights in lda_model.components_:
      top_keyword_locs = (-topic_weights).argsort()[:n_words]
      topic_keywords.append(keywords.take(top_keyword_locs))
   return topic_keywords

topic_keywords        =        show_topics(vectorizer=vectorizer,
lda_model=lda_model, n_words=15)

# Topic - Keywords Dataframe
df_topic_keywords = pd.DataFrame(topic_keywords)
df_topic_keywords.columns  =  ['Word  '+str(i)  for  i  in
range(df_topic_keywords.shape[1])]
df_topic_keywords.index  =  ['Topic  '+str(i)  for  i  in
range(df_topic_keywords.shape[0])]
df_topic_keywords

cmp              =              lda_model.components_              /
lda_model.components_.sum(axis=1)[:, np.newaxis]
def print_top_words(model, feature_names, n_top_words):
   for topic_id, topic in enumerate(cmp):
      print('\nTopic %d:' % int(topic_id + 1))
      print(''.join([feature_names[i] + ' * ' + str(round(topic[i], 3))
           +' + ' for i in topic.argsort()[:-n_top_words - 1:-1]]))
```

```
n_top_words = 5
feature_names = vectorizer.get_feature_names()
print_top_words(lda_model, feature_names, n_top_words)

lda_output = lda_model.transform(data_vectorized)

topic_prob = pd.DataFrame(lda_output)
topic_prob['kalimat_tweet'] = df['kalimat_tweet']
topic_prob

topic_prob.to_csv('topic prob gopay 3.csv')
```

Lampiran 6. Surat Pernyataan Data

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FMKSD ITS:

Nama : Nur Fidyah Permatasari

NRP : 06211540000087

menyatakan bahwa data yang digunakan dalam Tugas Akhir/ Thesis ini merupakan data sekunder yang diambil dari ~~penelitian / buku/ Tugas Akhir/ Thesis/~~ publikasi lainnya yaitu:

Sumber : Twitter API (*Application Program Interface*)

Keterangan : Data *tweet* dengan *keyword* "@gopayindonesia" dan data *tweet* dengan *keyword* "@ovo_id"

Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui
Pembimbing Tugas Akhir

Surabaya, Juli 2019

Pratnya Paramitha Oktaviana, S.Si., M.Si.
NIP. 1300201405001

Nur Fidyah Permatasari
NRP. 062115 40000087

*(coret yang tidak perlu)

BIODATA PENULIS

Penulis yang akrab disapa Fidy memiliki nama lengkap Nur Fidyah Permatasari. Penulis lahir di Surabaya pada tanggal 29 Juni 1997 dan merupakan anak kedua dari tiga bersaudara. Hingga kini putri dari pasangan Bapak Kresna Herlambang dan Ibu Titin Sumarti ini berdomisili di Surabaya. Penulis menempuh pendidikan formal di SD Negeri MA II Surabaya, dilanjutkan menempuh pendidikan di SMP Negeri 35 Surabaya, dan di SMA Negeri 9 Surabaya. Kemudian penulis diterima sebagai Mahasiswa Departemen Statistika, Fakultas Matematika, Komputasi, dan Sains Data (FMKSD), Institut Teknologi Sepuluh Nopember di Surabaya melalui jalur SBMPTN pada tahun 2015. Selama masa perkuliahan, penulis aktif dalam berbagai organisasi kampus seperti Ikatan Himpunan Mahasiswa Statistika Indonesia (IHMSI) 2016/2018 sebagai Bendahara Divisi Komunikasi dan Informasi, Himpunan Mahasiswa Statistika ITS (HIMASTA-ITS) 2017/2018 sebagai Staf Divisi PERS, Ketua Biro Departemen Komunikasi dan Informasi HIMASTA-ITS periode 2017/2018, sebagai Konseptor Kreatif GERIGI-ITS pada tahun 2017. Penulis juga berkesempatan pada tahun 2018 melakukan Studi Ekskursi di Dalian University of Technology, China. Penulis sangat terbuka akan kritik dan saran terkait hasil Laporan Tugas Akhir ini dengan menghubungi penulis melalui *e-mail* penulis fidyahpnur@gmail.com.