TESIS – KI142502

EKSTRAKSI KATA KUNCI *METADATA TWITTER* BERBAHASA INDONESIA DENGAN PENDEKATAN *GRAMMATICAL TAGGING* UNTUK VISUALISASI HUBUNGAN ANTAR FITUR PRODUK

SEPTIYAN ANDIKA ISANTA
5113201037

DOSEN PEMBIMBING
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M.Sc.

PROGRAM STUDI MAGISTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

THESIS – KI142502

KEYWORDS EXTRACTION METADATA TWITTER INDONESIAN LANGUAGE WITH APPROACH GRAMMATICAL TAGGING FOR VISUALIZATION OF RELATIONSHIP BETWEEN PRODUCT FEATURES

SEPTIYAN ANDIKA ISANTA
5113201037

SUPERVISOR
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M.Sc.

MASTER PROGRAM
DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

KATA PENGANTAR

Syukur *Alhamdulillah* penulis hanturkan kehadirat Allah SWT, yang telah memberikan rahmat dan hidayah-Nya kepada penulis, dan tidak lupa *shalawat* serta salam semoga senantiasa terlimpahkan kepada Nabi Besar kita, Muhammad SAW, yang telah membawa tauladan dan pengajaran yang *khasanah* sebagai petunjuk bagi kaum manusia, sehingga dengan begitu penulis dapat menyelesaikan tesis ini yang berjudul **"EKSTRAKSI KATA KUNCI *METADATA TWITTER* BERBAHASA INDONESIA DENGAN PENDEKATAN *GRAMMATICAL TAGGING* UNTUK VISUALISASI HUBUNGAN ANTAR FITUR PRODUK"** dapat terselesaikan dengan baik. Semoga tesis ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan terutama bidang komputasi cerdas dan visualisasi dan dapat memebri kontribusi bagi penelitian selanjutnya.

Dalam kesempatan ini, ijinkan penulis menyampaikan rasa terima kasih kepada:

1. Ibu Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. selaku dosen pembimbing utama atas kesabaran membimbing dan dukungan yang diberikan hingga terselesaikannya tesis ini.
2. Ibu Diana Purwitasari S.Kom., M.Sc. selaku dosen pembimbing kedua atas kesabaran membimbing dan dukungan yang diberikan hingga terselesaikannya tesis ini.
3. Bapak Dr. H. Agus Zainal Arifin, S.Kom, M.Kom, selaku Dekan Fakultas Teknologi Informasi ITS. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D., selaku Koordinator S2 Jurusan Teknik Informatika ITS.
4. Tim Penguji Tesis, Bapak Dr. H. Agus Zainal Arifin, S.Kom., Ibu Isye Arieshanti, S.Kom., M.Phil., dan Ibu Anny Yuniarti, S.Kom., M.Comp.Sc., selaku penguji sidang tesis yang telah memberikan masukan dan arahan.
5. Bapak dan Ibu dosen pascasarjana Teknik Informatika ITS yang telah bersedia dengan sabar mengajar dan memberi bimbingan selama masa kuliah.

6. Mardhatillah Firdayana dan Haikal Zikri Alifiyanda yang selalu memberikan semangat dan motivasi.
7. Kedua orang tua dan mertua, yang memberikan dukungan do'a, moril maupun materil.
8. Keluarga besar Mahasiswa Pascasarjana Teknik Informatika Angkatan 2013, yang banyak memberikan inspirasi dan semangat selama kuliah serta dalam menyelesaikan Tesis ini.
9. Mbak Rini dan Mbak Lina atas bantuan informasi dan adminstrasi yang berkaitan dengan ujian proposal, dan ujian Tesis.
10. Semua teman-teman yang tidak tersebutkan, penulis mengucapkan terima kasih atas bantuannya.

Penulis menyadari bahwa Tesis ini masih jauh dari sempurna, sehingga kritik dan saran dari pembaca sangat penulis harapkan. Akhir kata, Akhir kata, penulis berharap semoga Penelitian ini dapat bermanfaat bagi banyak pihak terutama untuk pengembangan ilmu pengetahuan dan teknologi di bidang Komputasi Cerdas dan Visualisasi.

Surabaya, 28 Januari 2016

Penulis

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

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

Septiyan Andika Isanta

Nrp. 5113201037

Dengan judul :

Ekstraksi Kata Kunci *Metadata* Twitter Berbahasa Indonesia Dengan Pendekatan *Grammatical Tagging* Untuk Visualisasi Hubungan Antar Fitur Produk

Tanggal Ujian : 19-01-2016

Periode Wisuda : Maret 2016

Disetujui oleh:

Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
NIP. 197512202001122002 (Pembimbing 1)

Diana Purwitasari, S.Kom, M.Sc
NIP. 197804102003122001 (Pembimbing 2)

Dr. Agus Zainal Arifin, S.Kom, M.Kom
NIP. 197208091995121001 (Penguji 1)

Isye Arieshanti, S.Kom, M.Phil
NIP. 197804122006042001 (Penguji 2)

Anny Yuniarti, S.Kom, M.Comp.Sc
NIP. 198106222005012002 (Penguji 3)

Direktur Program Pasca Sarjana,

Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

EKSTRAKSI KATA KUNCI *METADATA TWITTER* BERBAHASA INDONESIA DENGAN PENDEKATAN *GRAMMATICAL TAGGING* UNTUK VISUALISASI HUBUNGAN ANTAR FITUR PRODUK

Nama Mahasiswa : Septiyan Andika Isanta
NRP : 5113201037
Pembimbing : Dr.Eng. Chastine Fatichah, S.Kom.,M.Kom.
Diana Purwitasari, S.Kom., M.Sc.

ABSTRAK

Informasi dari sosial media terutama twitter banyak mengandung teks yang menjelaskan fitur suatu produk. Seorang pemilik produk dapat mengamati kata kunci populer dalam twitter secara manual untuk mengetahui fitur produk yang perlu diperbaiki. Proses ini memakan waktu lama mengingat jumlah *tweet* berisi opini suatu produk sangatlah besar. Pengenalan kata kunci dapat dilakukan dengan cara mengekstrak kata yang ada pada suatu tweet dan melakukan perhitungan jumlah kata. Kata dengan jumlah kemunculan terbesar pada umumnya menjadi kata kunci dalam sebuah kumpulan *tweet*. Akan tetapi jumlah kemunculan kata belum tentu menggambarkan fitur sebuah produk. Pemilik produk membutuhkan informasi terkait kata kunci produk yang sedang populer ataupun produk serupa milik kompetitor serta hubungan semantik antara fitur yang menggambarkan kelebihan produk tersebut.

Pada penelitian ini diusulkan suatu metode ekstraksi kata kunci dari metadata twitter dengan pendekatan grammatical tagging. Tahapan dalam metode tersebut yaitu praproses, ekstraksi kata, ekstraksi produk, dan hitung asosiasi serta visualiasi hubungan antara fitur kata-kata produk. Ekstraksi kata bertujuan untuk mendapatkan kandidat kata kunci dari tweet. Tahapan berikutnya adalah ekstraksi produk yang tertulis dalam tweet menggunakan grammatical tagging. Kemudian dilakukan penghitungan asosiasi dengan nilai confidence untuk mencari hubungan antara kata fitur produk yang tertulis dalam tweet produk. Hasil pengujian menggunakan data tweet dari Apple dan Nexus menunjukkan bahwa nilai kemunculan kata lebih berpengaruh dari pada nilai retweet dan nilai favourite. Metode yang diusulkan dapat mengenali kata fitur produk sebuah tweet dengan nilai precision 81.3%, serta dapat mengenali hubungan antara kata fitur produk dengan nilai f-measure 52.5

Kata kunci: ekstraksi, fitur produk, kata kunci, twitter, graph, grammatical tagging, POS Tagging

KEYWORDS EXTRACTION METADATA TWITTER INDONESIAN LANGUAGE WITH APPROACH GRAMMATICAL TAGGING FOR VISUALIZATION OF RELATIONSHIP BETWEEN PRODUCT FEATURES

Student Name	: Septiyan Andika Isanta
NRP	: 5113201037
Supervisor	: Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
	Diana Purwitasari, S.Kom., M.Sc.

ABSTRACT

Information from social media, especially twitter, contain lots of text describing the features of a product. A product owner can manually observe popular keywords in twitter to determine the product features that need to be improved. This process takes a long time due to the enormous number of tweets containing opinions related to the product. Keywords recognition for candidates of product features can be performed by extracting existing words in a tweet and counting the words. The word with the greatest number of occurrences, usually are the keywords in a collection of tweets. However, the number of word occurrences does not necessarily reflect the features of a product. The product owner requires information of keywords related to popular products or similar products belonging to competitors and semantic relationships among the features that reflect the advantages of these products.

In this study, a method is proposed to extract keywords from twitter metadata through grammatical tagging approach. The steps in this method are preprocessing, word extraction, product extraction, calculating the association and visualizing the relationship among the features of the product's words. The word extraction is intended to obtain the keywords candidate from tweets. The next step is the extraction of the product written in a tweet using the grammatical tagging. Afterward, the calculation of the association is performed with the confidence value to find relationships among the words related to the product features written in product tweet. The test results using tweet data from Apple and Nexus revealed that the value of word occurrences is more significant than the value of retweet and favourite. The proposed method can recognize the word of product features in a tweet with precision value of 81.3%, and can recognize relationships among the words of product features with an f-measure value of 52.5%.

Keywords: extraction, product, keyword, twitter, graph, grammatical tagging, POS Tagging

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Twitter merupakan salah satu media sosial yang sedang ramai digunakan pada saat ini, kata yang lagi popular dari twitter biasanya menunjukkan kejadian yang sedang terjadi dan sering disebut sebagai kata kunci. Seorang pemilik produk biasanya mengamati kata kunci apa yang sedang popular dari sebuah produk mereka atau produk kompetitor. Visualisasi kata yang popular tersebut umumnya ditampilkan dalam bentuk grafik batang atau grafik lingkaran. Dengan memvisualisasikan dalam bentuk grafik batang atau grafik lingkaran setiap kata kunci dianggap berdiri sendiri, padahal ada hubungan semantic antara kata kunci tersebut. Sehingga dibutuhkan sebuah visualisasi yang dapat mengetahui hubungan realasi *semantic* antara kata kunci.

Beberapa penelitian melakukan visualisasi hubungan antar kata atau *hashtag* dalam twitter dalam bentuk graph, seperti yang dilakukan oleh Abscal dia melakukan visualisasi hubungan antara *hashtag* bedasarkan *hashtag* tertentu (Abascal-Mena, Lema, & Sedes, 2014). Amma juga melakukan visualisasi topik *tweet* dari twitter stream dalam bentuk graph (Amma, Wada, Nakayama, Akamatsu, Yaguchi, & Naruse, 2014). Pada kedua penelitian tersebut dalam pembuatan *graph* mereka menggunakan fitur frekuensi jumlah kata atau kata kunci yang muncul didalam kumpulan *tweet*.

Sebelum pembentukan relasi antara kata kunci yang ada di kumpulan *tweet*s terlebih dahulu diperlukan metode untuk mengeksktraksi kata kunci dan metode untuk merealisasikan kata kunci tersebut. Pada penelitian yang dilakukan Medvet dan Bartoli, mereka menggunakan metode perbandingan frekuensi kata dengan jumlah *tweet* dan riwayat *tweet* untuk mendapatkan kata kunci yang sedang populer pada suatu event tertentu dari sebuah brand (Medvet & Bartoli, 2012). Dalam penelitian tersebut mereka tidak hanya melakukan pengektrasian 1 keyword, melainkan set of word dari kumpulan *tweet*. Sebagai contoh dalam *tweet* "@.... did you pre order your iPad", "@.... discount 5% for pre order iPad " kata yang penting adalah pre, order dan ipad, sehingga set of word yang dihasilkan

adalah “pre order ipad”. Akan tetapi, jika hanya menggunakan jumlah kemunculan kata dari sebuah kumpulan tweet, kata yang didapatkan belum tentu menggambarkan fitur sebuah produk dari *tweet* tersebut, sehingga dibutuhkan sebuah metode natural language processing (NLP) untuk mengetahui produk yang sedang dibicarakan dalam sebuah *tweet*.

Untuk mendapatkan informasi struktur kata dalam *tweet*, Hasby mengusulkan sebuah metode untuk ekstraksi informasi dalam twitter menggunakan NLP, metode yang digunakan adalah melakukan ekstraksi *named entity recognition* untuk mendapatkan *tag*ging dari *tweet* berbahasa indonesia (Hasby & Khodra, 2013). Selain menggunakan *named entity recognition* untuk mendapatkan *tag*ging dari kalimat bisa menggunakan *POS Tagging* atau biasa disebut *grammatical tag*ging. Alfan Farizki menggunakan metode Hidden Markov Model untuk mendapatkan *grammatical tag*ging dari dokumen news berbahasa indonesia, sehingga dapat mengetahui apakah kata dari sebuah dokumen tersebut adalah kata benda, kata kerja, kata sifat dll (Alfan Farizki Wicaksono, 2010). Setelah mendapatkan *tag*ging dari sebuah kata dalam *tweet*, bagaimana menentukan kata yang menjadi produk didalam *tweet* tersebut. Terdapat sebuah penelitian lain yang melakukan ekstraski fitur produk dalam dokumen review yang dilakukan oleh Qiu, dia mengusulkan metode berbasis *rule* untuk mendapatkan fitur produk dari sebuah kalimat opini (Qiu, 2014). Yang menjadi permasalahan adalah bagaimana menerapkan *rule* untuk melakukan ekstraksi produk dari sebuah *tweet* dalam twitter berbahasa Indonesia.

Permasalahan lain yang muncul adalah bagaimana jika set of word yang penting dari sebuah kumpulan *tweet* lebih dari satu, seperti dalam *tweet* “ Lazada Buka Pre-Order Xiaomi Redmi Note: Redmi Note dibanderol Rp1,99 juta. ” dan “Xiaomi luncurkan Redmi Note di Indonesia, Mi3 tidak akan dijual”, produk yang dikomentari adalah Redmi Note dan Mi3 sedangkan kata kunci yang penting adalah preorder, lazada. Sehingga dibutuhkan suatu metode untuk merepresentasikan hubungan antara kata-kata fitur produk.

Sebagian besar peneliti yang melakukan penelitian menggunakan twitter hanya menggunakan fitur frekuensi *tweet*. Padahal selain fitur frekuensi atau jumlah kata kunci dalam *tweet*, masih banyak metadata twitter yang bisa

digunakan untuk menganalisa *tweet* tersebut. Beberapa yang bisa dipakai untuk mengetahui bobot sebuah *tweet* adalah verified user, *favourite* dan *retweet*. Fitur *favourite* dan *retweet* bisa digunakan untuk mengetahui *tweet* itu penting atau tidak. Karena jika *tweet* tersebut di *favourite* atau di *retweet* orang lain maka *tweet* tersebut dianggap penting oleh orang lain.

Berdasarkan permasalahan tersebut, dalam penilitian ini diusulkan suatu metode untuk ekstraksi kata kunci berdasarkan metadata *tweet* dan content *tweet* dengan pendekatan *grammatical tag*ging untuk visualisasi hubungan antara fitur produk. Dengan adanya usulan tersebut diharapkan pemilik produk dapat mengetahui kata kunci serta produk yang sedang popular ataupun produk serupa milik kompetitor, serta hubungan *semantic* antara fitur yang menggambarkan kelebihan produk tersebut.

1.2 Perumusan Masalah

Dalam penelitian ini, masalah-masalah yang akan diselesaikan dirumuskan sebagai berikut :

1. Bagiamana melakukan pengekstrakan produk dan kata kunci yang tertulis dalam *tweet* berbasis *grammatical tagging* dalam *tweet* berbahasa Indonesia.
2. Bagimana melakukan pembobotan kata kunci berdasarkan fitur jumlah, *favourite* dan *retweet*.
3. Bagaimana memvisualisasikan hubungan antara fitur kata-kata produk.

1.3 Batasan Masalah

Pembahasan dalam penelitian ini dibatasi pada beberapa hal berikut:

1. *Tweet* yang digunakan adalaah *tweet* berbahasa Indonesia
2. *Brand* yang digunakan untuk penelitian adalah Apple, dan Nexus.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengekstraksi kata kunci dan produk dari sebuah *tweet* berdasarakan metadata twiiter dan *grammatical tagging*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini diharapkan akan dapat membantu pemilik produk mengetahui kata kunci serta produk yang sedang popular ataupun produk serupa milik kompetitor, serta hubungan *semantic* antara fitur yang menggambarkan kelebihan produk tersebut.

1.6 Kontribusi Penelitian

Kontribusi penelitian ini yaitu mengusulkan metode untuk ekstraksi serta visualisasi fitur kata-kata produk berdasarkan *metadata* dan content *tweet* berbahasa indonesia dengan pendekatan grammtical *tag*ging pada sebuah tweet produk.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini akan menjelaskan teori-teori dasar yang berkaitan erat dengan topik bahasan dan tinjauan pustaka mengenai metode yang diusulkan.

2.1 *POS Tagging / Grammatical Tagging*

*Grammatical tag*ging atau yang biasa disebut Part-of-Speech *tag*ging merupakan sebuah pekerjaan dalam NLP (Natural Language Processing). Sebagian besar kegiatan yang dilakukan di bidang NLP seperti Information Extraction, Question-Answering, Speech Recognition, Intelligent Tutoring System, Parser, dan yang lainnya membutuhkan sistem *POS Tagging* ini untuk pemrosesan awalnya, termasuk ekstraksi kata kunci.

*Grammatical tag*ging adalah sebuah sistem yang memberikan label kata secara otomatis pada suatu kalimat. Misalkan, ada kalimat "I drink milk". Sistem akan menerima input berupa kalimat tersebut, outputnya adalah:

I_PRP drink_VBZ milk_NN dimana PRP adalah proonoun atau kata ganti, VBZ adalah verb atau kata kerja, dan NN adalah noun atau kata benda.

Ada 3 pendekatan pengklasifikasian kata yang dapat digunakan. Pendekatan pertama adalah pendekatan formal. Pada pendekatan ini, anatomi dari kata digunakan sebagai penentu klasifikasi dari kata. Sebagai contoh, kata yang berakhiran -ing biasanya dapat digolongkan langsung sebagai kata kerja (verb). Pendekatan kedua adalah pendekatan sintaktik. Pendekatan ini menggunakan klasifikasi dari kata lain di dekat kata yang tidak teridentifikasi. Sebagai contoh, kata sifat (adjective) biasanya muncul tepat sebelum kata benda (noun) seperti good camera, bad boy, dan sebagainya. Pendekatan terakhir adalah pendekatan konteks. Pada pendekatan ini, sebuah klasifikasi dipahami maknanya dan digunakan sebagai penentu penggolongan kata. Sebagai contoh, kata benda adalah kata yang merepresentasikan suatu objek. Pendekatan ini sangat sulit diformulasikan. Dan karenanya kurang begitu dipakai dalam pengklasifikasian kata (Jones, 1994). Tabel 3.1 menunjukkan jenis-jenis *tag* dalam *POS Tagging* beserta contohnya.

Tabel 2.1 *POS Tagging*

POS *Tag*	Deskripsi (English)	Deskripsi (Indonesia)	Contoh
OP	Open Parenthesis	Kurung Buka	({[
CP	Close Parenthesis	Kurung Tutup	)}]
GM	Slash	Garis Miring	/
;	Semicolon	Titik Koma	;
:	Colon	Titik Dua	:
“	Quotation	Tanda Kutip	" dan '
.	Sentence Terminator	Tanda Titik	.
,	Comma	Tanda Koma	,
-	Dash	Garis	-
...	Ellipsis	Tanda Pengganti	...
JJ	Adjective	Kata Sifat	Baik, Bagus
RB	Adverb	Kata Keterangan	Sementara, Nanti
NN	Common Noun	Kata Benda	Meja, Kulkas
NNP	Proper Noun	Benda Bernama	Samsung, Sony
NNG	Genitive Noun	Benda Berpemilik	Handphonenya
VBI	Intransitive Verb	Kata Kerja Intransitif	Pergi
VBT	Transitive verb	Kata Kerja Transitif	Membeli
IN	Preposition	Preposisi	Di, Dari, Ke
MD	Modal	Modal	Bisa
CC	Coor- Conjunction	Kata Sambung Setara	Dan, Atau, tetapi
SC	Subor- Conjunction	Kata Sambung Tidak Setara	Jika, Ketika
DT	Determiner	Determiner	Para, Ini, Itu
UH	Interjection	Interjection	Wah, Aduh, Oi
CD	Cardinal Number	Bilangan	1, 2
CDO	Ordinal Numerals	Kata Bilangan Berurut	Pertama, Kedua
CDC	Collective Numerals	Kata Bilangan Kolektif	Berdua
CDP	Primary Numerals	Kata Bilangan Pokok	Satu, Dua, Tiga
CDI	Irregular Numerals	Kata Bilangan Tidak Biasa	Beberapa
PRP	Personal pronoun	Kata Ganti Orang	Saya, Mereka
WP	Wh-pronoun	Kata tanya	Apa, Siapa,
PRN	Number Pronouns	Kata Ganti Bilangan	Kedua-duanya
PRL	Locative Pronouns	Kata Ganti Lokasi	Sini, Situ
NEG	Negation	Negasi	Bukan, Tidak
SYM	Symbol	Simbol	#,%,^,&,*
RP	Particle	Particle	Pun, Kah
FW	Foreign word	Kata Asing	Word

Ada beberapa pendekatan yang bisa digunakan untuk melakukan POS Tagging, yaitu pendekatan berdasar aturan (rule based), pendekatan probabilistik, dan pendekatan berbasis transformasi (transformational based).

Untuk POS Tagging yang menggunakan metode probabilistik, teknik Hidden Markov Model bisa digunakan. Hal ini dikarenakan proses POS Tagging bisa dipandang sebagai proses klasifikasi suatu rangkaian atau urutan tag untuk tiap kata dalam suatu kalimat.

2.2 *Preprocessing* Teks

Preprocessing merupakan tahap yang penting dalam pemrosesan teks untuk memperoleh fitur yang akan diproses pada tahap selanjutnya. Dalam preprocessing yang akan dilakukan terdiri dari beberapa tahapan yaitu normalisasi text twitter, pemecahan kalimat, case folding, tokenizing kata, filtering kata, dan *stemming* menggunakan algoritma *Enhanced Confix Stripping (ECS) Stemmer*.

2.2.1 Normalisasi Teks *Twitter*

Tahapan ini dilakukan bertujuan untuk mengubah text *tweet* menjadi formal text. Dalam normalisasi teks *tweet* terdapat beberapa tahapan yang akan dilakukan yaitu :

1. Memisahkan angka di depan atau di belakang kata.
2. Mengubah angka di dalam text menjadi huruf
3. Menghapus huruf yang berulang
4. Mengubah kata allay menjadi kata formal

2.2.2 *Case folding*

Case folding adalah tahapan proses mengubah semua huruf dalam teks dokumen menjadi huruf kecil semua, serta menghilangkan karakter selain a-z dan dianggap sebagai delimiter.

2.2.3 *Tokenizing*

Tokenizing kata adalah proses pemotongan string input berdasarkan tiap kata yang menyusunnya. Pemecahan kalimat menjadi kata-kata tunggal dilakukan dengan men-scan kalimat dengan pemisah white space (spasi, tab, newline).

2.2.4 *Filtering*

Filtering merupakan proses penghilangan stopword. Stopword adalah kata-kata yang sering kali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Didalam bahasa Indonesia stopword dapat disebut sebagai kata tidak penting, misalnya “di”, “oleh”, “pada”, “sebuah”, “karena” dan lain sebagianya. Dalam penilitian ini daftar stopword yang digunakan sebanyak 1147 kata.

2.2.5 *Stemming*

Stemming adalah proses mencari akar (root) kata dari tiap token kata yaitu dengan pengembalian suatu kata berimbuhan ke bentuk dasarnya (stem). Pada penelitian ini algoritma *stemming* yang digunakan adalah algoritma Algoritma *stemming* kata pada Bahasa Indonesia yang akan digunakan dalam penelitian ini adalah algoritma Enhanced Confix Stripping (ECS) Stemmer.

Algoritma ECS Stemmer ini merupakan algoritma perbaikan dari algoritma Confix Stripping (CS) Stemmer (Arifin, Mahendra, & Ciptaningtyas, 2009). Perbaikan yang dilakukan oleh ECS Stemmer adalah perbaikan beberapa aturan pada tabel acuan pemenggalan imbuhan. Selain itu, algoritma ECS Stemmer juga menambahkan langkah pengembalian akhiran jika terjadi penghilangan akhiran yang seharusnya tidak dilakukan.

2.3 Ekstraksi Kata Kunci *Tweet*

Twitter merupakan layanan jejaring social yang memiliki perbedaan dengan jejaring social media yang lain yaitu memiliki ukuran panjang teks terbatas 140 karakter (Cordeiro, 2012). Batasan tersebut menyebabkan pengguna dengan mudah mengirim *tweet* dengan cepat tentang informasi yang akan

disampaikan. Pengguna bisa mengirim pesan singkat berupa kritik, saran, opini, kabar, berita, suasana hati, peristiwa, fakta dan lain-lain yang tidak terka*tag*orikan. Pesan yang dikirim cenderung secara singkat dan lansgung pada inti dari informasi yang disampaikan.

Saat ini jumlah pengguna twitter telah mencapi 140 juta pengguna aktif yang rata-rata per hari mengirimkan pesan singkat sejumlah 400 juta pesan (Farzindar Atefeh, 2013). Angka-angka tersebut menujukkan bahwa twitter banyak digunakan karena beberapa hal seperti portabilitas, mudah digunakan, berisi pesan singkat dan tidak ada batasan pengguna untuk menyebarkan informasi dalam media tersebut. Dari sekian banyak pesan singkat yang dikirimkan tersebut, terdapat *tweet* yang berisakan opini terhadap sebuah produk dan promo sebuah produk. Untuk mendapatkan kata kunci dalam *tweet* dapat menggunakan metode statistik seprerti yang digunakan oleh peneliti sebelumnya (Medvet & Bartoli, 2012) yaitu menghitung jumlah kemunculan *term* dalam sekumpulan *tweet*, term yang memiliki jumlah kemunculan paling tinggi dapat diartikan sebagai kata kunci dalam sekumpulan *tweet*.

2.4 Ekstraksi Produk dari *Tweet* menggunakan *Hidden Markov Model POS Tagging*

Metode statistik yang digunakan dalam ektraksi kata kunci tidak bisa digunakan untuk mendapatkan produk yang tertulis dalam *tweet* karena untuk mendapatkan produk yang tertulis dalam *tweet* kita perlu tahu *tag* atau makna dari masing-masing kata dalam *tweet*. Oleh sebab itu perlu digunakan *grammatical tag*ging dan *rule*, untuk mendapatkan kata yang menjadi produk dari sebuah *tweet*, algoritma yang dipakai untuk ekstraksi tagging dalam penilitian ini adalah *Hidden Markov Model* (HMM).

Pada kasus Part of Speech *Tag*ging, urutan kelas kata tidak dapat diamati secara langsung sehingga dijadikan sebagai hidden state dan yang menjadi observed state adalah urutan kata-kata (Alfan Farizki Wicaksono, 2010). Dari urutan kata-kata harus dicari urutan kelas kata yang paling tepat. HMM *POS Tagger* menggunakan dua buah asumsi, asumsi yang pertama adalah probabilitas kemunculan suatu kata hanya tergantung pada tag- nya, dan tidak tergantung

dengan kata lain di sekitarnya atau tag lain di sekitarnya. Asumsi yang kedua adalah probabilitas suatu kemunculan tag hanya bergantung dari tag sebelumnya (Rozi, 2013.). Dengan kedua buah asumsi teresebut persamaan dari *Hidden Markov Model* untuk kasus *Part of Speech Tagging* adalah .

$$Tag_n = Max\left(P(word_i|tag_i)\ P(tag_i|tag_{i-1})\right) \quad (2.1)$$

dimana $\mathrm{Tag_n}$ adalah kelas kata yang dicari, $\mathrm{tag_i}$ adalah kelas kata dari word ke *i* yang ada di corpus, $\mathrm{word_i}$ adalah kata yang dicari kelas katanya, $\mathrm{tag_{i-1}}$ adalah kelas kata sebelum kelas kata dari word ke i yang ada di corpus dan P adalah nilai probabilitas. Untuk melakukan perhitungan nilai probabilitas transisi suatu tag dari tag sebelumnya (*transition probability*) $P(tag_i|tag_{i-1})$ dan probabilitas kemiripan suatu kata sebagai sebuah tag (*emission probability*)$P(word_i|tag_i)$ diperlukan koleksi data tweet atau kalimat yang telah diberikan tag sebelumnya (corpus). Untuk menghitungnya dapat digunakan persamaan :

$$P(tag_i|tag_{i-1}) = \frac{C(tag_{i-1},tag_i)}{C(tag_{i-1})} \quad (2.2)$$

$$P(word_i|tag_i) = \frac{C(tag_i,word_i)}{C(tag_i)} \quad (2.3)$$

2.5 Asosiasi Produk dan Kata Kunci Tweet dengan algoritma *Association Rule*

Association *Rule* adalah suatu metode data mining yang bertujuan untuk mencari sekumpulan items yang sering muncul bersamaan. Association *rule* umumnya mengambil bentuk IF-THEN yang menggabungkan beberapa items menjadi satu, misalnya: IF A THEN B. Association *rule* meliputi dua tahap :

1. Mencari kombinasi yang paling sering terjadi dari suatu *itemset*.
2. Mendefinisikan *Condition* dan *Result* (untuk *conditional association rule*)

Dalam menentukan suatu association *rule*, terdapat suatu interestingness measure (ukuran kepercayaan) yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu:

a) *Support* : suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu item/itemset dari keseluruhan transaksi. Ukuran ini akan menentukan apakah suatu item/itemset layak untuk dicari *confidence*-nya (misal, dari seluruh transaksi yang ada, seberapa besar tingkat dominasi yang menunjukkan bahwa item A dan B dibeli bersamaan) dapat juga digunakan untuk mencari tingkat dominasi item tunggal. Adapun rumus yang digunakan untuk menghitung *support* adalah sebagai berikut:

$$s = \frac{\sigma(A, \mathrm{B})}{|\mathrm{T}|}, \tag{2.4}$$

dimana s adalah nilai *support*, $\sigma(A, \mathrm{B})$ total transaksi item A dan B secara bersamaan, dan $|\mathrm{T}|$ adalah jumlah transaksi total.

b) *Confidence* : suatu ukuran yang menunjukkan hubungan antar 2 item secara conditional (misal, seberapa sering item B dibeli jika orang membeli item A). Adapun rumus yang digunakan untuk menghitung *confidence* adalah sebagai berikut:

$$c = \frac{\sigma(\mathrm{A},\mathrm{B})}{\sigma(A)} \tag{2.5}$$

dimana c adalah nilai *confidence*, $\sigma(A, \mathrm{B})$ total transaksi item A dan B secara bersamaan, dan $\sigma(A)$ adalah jumlah transaksi item A.

Pada penelitian ini yang akan diasosiasikan adalah produk dari *tweet* dan kata kunci hasil ekstraksi *tweet*, asosiasi tersebut bertujuan untuk mengetahui seberapa kuat relasi antara produk dan kata kunci dalam sebuah kumpulan *tweet*. Agar dapat menghitung seberapa sering kata kunci muncul dalam *tweet* yang mengandung produk, dapat dilakukan dengan menghitung nilai *confidence* produk *tweet* dan kata kunci. Sehingga kita bisa tahu seberapa besar nilai kedekatan atau relasi antara produk dan kata kunci dalam *tweet*.

2.6 Metode Evaluasi *Precision, Recall,* dan *F-Measure*

Metode evaluasi dilakukan untuk mengalisa kinerja sistem yang diusulkan pada penelitian ini. Kinerja sistem yang diusulkan dengan menggunakan nilai *precision*, *recall*, dan *f-measure*. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sedangkan *recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.

(2.6)

(2.7)

Pada dasarnya, nilai *recall* dan *precision* berada pada rentang antara 0 sampai dengan 1. Oleh karena itu, suatu sistem temu kembali yang baik adalah yang dapat memberikan nilai *recall* dan *precision* mendekati 1.

Nilai *recall* atau *precision* saja belum cukup mewakili kinerja sistem. Oleh karena itu diperlukan metode evaluasi yang mengkombinasikan metode evaluasi *recall* dan *precision* metode evaluasi ini adalah *F-measure*. Formulasi *F-measure* dinyatakan dalam persamaan berikut:

$$\frac{2(Precision \times Recall)}{} \quad (2.8)$$

Pada evaluasi ekstraksi produk dari sebuah tweet, *precision* berarti presentase jumlah produk yang *relevan* dari jumlah produk yang ditemukan. Sedangkan *recall* adalah presentase tingkat keberhasilan system menemukan produk tweet yang relevan dari seluruh koleksi produk tweet yang relevan. Sedangkan pada evaluasi asosiasi produk dan kata kunci, *precision* berarti presentase jumlah asosiasi produk dan kata kunci yang *relevan* dari jumlah asosiasi produk dan kata kunci ditemukan. Sedangkan *recall* adalah presentase tingkat keberhasilan system menmukan asosiasi produk dan kata kunci tweet yang relevan dari seluruh koleksi asosiasi produk dan kata kunci tweet.

BAB III
METODOLOGI PENELITIAN

3.1 Rancangan Penelitian

Secara umum, penelitian ini diawali dengan studi literatur, pengumpulan data, desain sistem, pengujian sistem, analisis hasil, dan penyusunan laporan. Secara lebih detail, penelitian ini dirancang dengan urutan sebagai berikut:

3.2 Studi Literatur

Studi literatur dilakukan untuk mendapatkan dan memahami berbagai infomasi yang terkait dengan penelitian ini. Informasi tersebut diperoleh dari berbagai literatur yang ada, perkembangan, serta metode yang pernah digunakan dalam penelitian sebelumnya. Studi literatur yang dilakukan diharapkan dapat memberikan data, informasi, dan fakta mengenai ekstraksi dan visualisasi produk dan kata kunci *tweet* yang akan dikembangkan. Studi literatur yang dilakukan mencakup pencarian dan mempelajari referensi-referensi yang terkait, seperti:

1. Metode *Grammatical tagging* bahasa indonesia.
2. Metode ekstraksi produk dan kata kunci dalam *tweet*.
3. Metode untuk mendapatkan relasi antara produk dan kata kunci *tweet*.
4. Metode untuk visualisasi hubungan antara fitur kata-kata produk.

3.3 Pengumpulan dan Analisis Data

Dataset yang digunakan dalam penelitian ini adalah data *tweet* atau dokumen *tweet* yang diperoleh dengan memanfaatkan Search API dan Steram API yang disediakan oleh twitter. Sebuah sistem dibangun untuk mengambil data *tweet* tersebut dari twitter dengan menggunakan Search API dan streaming API dengan menambahkan proses deteksi bahasa untuk mendapatkan data bahasa Indonesia dan pembatasan waktu. Topik hanya dibatasi mengenai *tweet* terhadap produk dari brand antara lain : Apple dan Nexus.

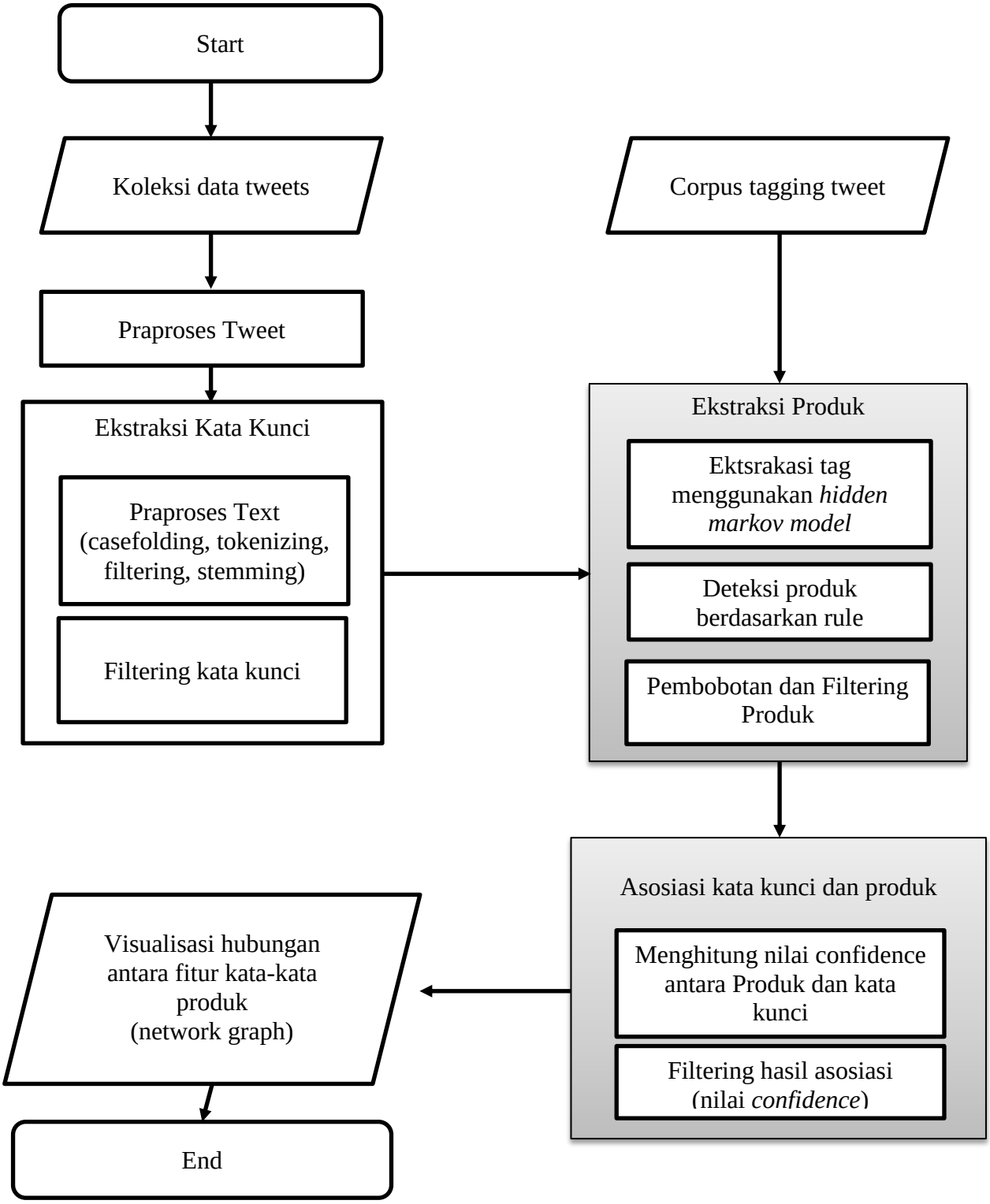
Gambar 3.1 contoh *tweet* yang mengomentari sebuah produk

Gambar 3.1 menunjukkan contoh *tweet* yang mengomentari sebuah produk dari brand Apple. Pada sebuah *tweet*, selain informasi text, kita bisa mendapatkan infomasi jumlah *retweet* dan jumlah *favourite* dari *tweet* tersebut. Contohnya pada *tweet* pertama dari kompas TV, text dalam *tweet* tersebut adalah "Dollar AS perkasa, banderol iphone 6 terkerek naik RP 500 Ribu", jumlah *retweet* 7 dan jumlah *favourite* 4, Berarti *tweet* tesebut sudah di *retweet* oleh 7 orang dan di *favourite* oleh 4 orang.

3.4 Desain Sistem

Secara global desain model sistem yang akan dibuat ditunjukkan pada gambar 3.2. Desain sistem dalam penelitian ini terdiri dari 4 bagian utama yaitu:

1. *Preprocessing tweet*,
2. Ekstraksi kata kunci yang tertulis dalam *tweet*.
3. Ekstraksi produk yang tertulis dalam *tweet* berbasis *grammatical tagging*
4. Asosiasi dan visualisasi hubungan antara fitur kata-kata produk.



Gambar 3.2 Diagram alir sistem

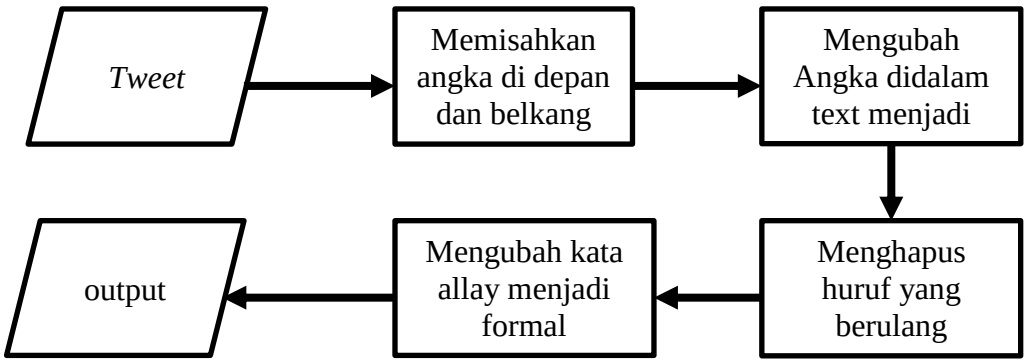
Pada Gambar 3.2, dapat dilihat bahwa proses visualisasi hubungan antara fitur kata-kata produk terdiri dari 5 tahapan. Dimana masing-masing tahap merupakan modul tersendiri yang memanfaatkan output dari modul yang lain. Tahap pertama adalah *prepocessing tweet* kemudian dilanjutkan dengan ekstraksi kata kunci dan ekstraksi produk secara bergantian. Setelah mendapatkan produk dan kata kunci dari *tweet* dilakukan asosiasi dan visualisasi menggunakan graph.

1.4.1 Tahap *Prepocessing Tweet*

Tahap *prepocessing tweet* bertujuan untuk mengubah text dalam *tweet* menjadi formal text, langkah-langkah yang dilakukan adalah sebagai berikut :

1. Memisahkan angka di depan atau di belakang text dengan text yang mengikutinya, Contohnya “2hari”, menjadi “2 hari”
2. Mengubah angka didalam text menjadi huruf, contohnya “ga2l” menjadi “gagal”
3. Menghapus huruf yang berulang, contohnya adalah kata “tidaaak”, menjadi “tidak”.
4. Mengubah kata allay menjadi kata formal berdasarkan *dictionary* yang didapat dari http://kamusmania.com/component/glossary/Kamus-Alay-9/, contonya “ciyus” menjadi “serius”.

Dari hasil prepocessing *tweet*s akan diperoleh data *tweet*s yang mennggunakan kata-kata formal, data tersebut akan dijadikan sebagai data uji. Flowchart untuk tahap preprocessing *tweet* ditunjukkan dalam Gambar 3.3 dibawah ini.



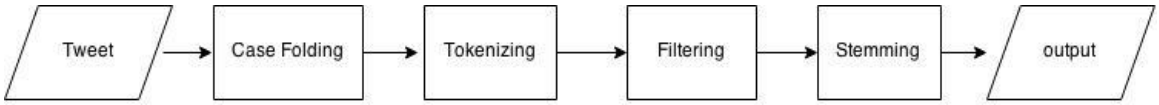
Gambar 3.3 Flowchart preprocessing data *tweet*

1.4.2 Tahap *Prepocessing text* dan Ekstraksi Kata Kunci

Tahap *prepocessing text* yang akan dilakukan sesuai dengan langkah-langkah berikut :

1. *Case folding*, mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter.
2. *Tokenizing*, pemotongan string input berdasarkan tiap kata yang menyusunnya.
3. *Filtering*, adalah tahap mengambil kata - kata penting dari hasil token. Bisa menggunakan stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting). Stoplist / stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Contoh stopwords adalah "yang", "dan", "di", "dari" dan seterusnya.
4. *Stemming*, mencari root kata dari tiap kata hasil filtering. Pada penilitian ini implementasi stemming meggunakan algoritma Enhanced Confix Stripping (ECS) Stemmer dengan bahasa pemrogragaman java.

Dari hasil prepocessing akan diperoleh koleksi term yang akan dijadikan kata kunci dari data *tweet*. Flowchart untuk tahap preprocessing data *tweet* ditunjukkan dalam Gambar 3.4



Gambar 3.4 Flowchart *preprocessing text*

Setalah didapat kata kunci, maka kata kunci tersebut akan difilter bedasarkan jumlah kemunculan, dengan persamaan seperti berikut :

$$w_{kata-kunci} = \frac{\log(|\, D_{term_i} \,|)}{log(|\, D \,|)}, \qquad (3.1)$$

dimana $|D_{term_i}|$ adalah jumlah kemunculan term ke *i* pada total dokumen tweet, dan $|D|$ adalah jumlah total dokumen tweet.

1.4.3 Ektraksi Produk berbasis *Grammatical Tagging*

Untuk mendapatkan produk dari *tweet*, atau produk yang dikomentari terlebih dahulu harus mengetahui *tag* dari masing-masing kata dan relasi informasi tiap kata dalam suatu *tweet*.

a. POS Tagging

Proses ini dilakukan untuk mendapatkan informasi *tag* dari tiap kata yang ada pada *tweet*. Kebanyakan *library POS Tagging* yang ada, digunakan untuk bahasa inggris dan untuk dokumen berita, antara lain *Stanford POS Tagger* dan *linpipe*. Karena dalam penelitian ini data yang digunakan adalah data *tweet* berbahasa indonesia maka diperlukan pembuatan *system POS Tagging* tersendiri, metode yang digunakan adalah *Hidden Makov Model* (HMM) dan ditambahkan 4 buah model *tag* baru yang behubungan dengan *twitter* yaitu *tag* @ untuk *user*, *tag* RT untuk RT, *tag* U untuk url dan *tag* E untuk *emoji*.

Proses *training* pada *POS Tagging* berbasis HMM dilakukan dengan menghitung nilai kimiripan atau kemungkinan suatu kata sebagai sebuah tag (*emission probability)* dan nilai kemungkinan transisi suatu tag dari tag sebelumnya (*transition probability)* dari kumpulan data training. Hasil perhitungan dari proses *training* tersebut yang digunakan sebagai model acuan untuk menentukan *tag* atas suatu kata dalam suatu kalimat dari *data testing*. Tahapan yang dilakukan untuk mendapatkan tag dari sebuah kata adalah :

1. Menghitung nilai kimiripan atau kemungkinan suatu kata sebagai sebuah tag (*emission probability*) menggunakan persamaan 2.2.
2. Menghitung nilai kemungkinan transisi suatu tag dari tag sebelumnya (*transition probability*) menggunakan persamaan 2.3.
3. Menghitung nilai probabilitas suatu tag ke kata, yaitu dengan cara mengkalikan nilai *emission probability* dengan nilai *transition probability*.
4. Dipilih nilai suatu tag dengan nilai maksimal probabilitas yang akan dijidakan tag dari sebuah kata.

Berikut ini contoh *POS Tagging* kalimat *twitter*.

Misal untuk *tweet* “Dollar AS perkasa, banderol iphone 6 terkerek naik RP 500 Ribu”, *POS Tagging* yang dihasilkan adalah:

Dollar_NN AS_NNP perkasa_JJ ,_, banderol_RB iphone_NNP 6_CD terkerek_VB naik_VB RP_NNP 500_CD Ribu_NNP

Dimana NN adalah kata benda(noun), NNP adalah frase kata benda (noun), JJ adalah kata sifat (adjective), CD adalah bilangan pokok (cardinal number) dan VB adalah kata kerja (verb).

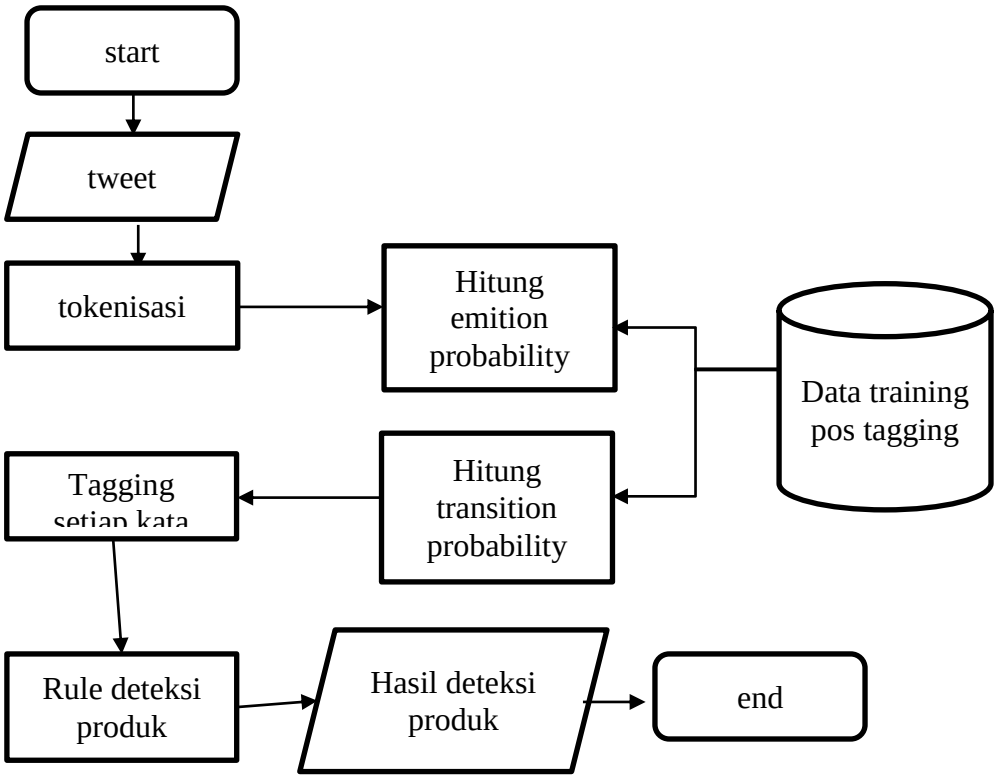
b. Ekstraksi Produk

Untuk menentukan kata mana yang menjadi produk dari sebuah *tweet*, diperlukan *rule* untuk mengolah data hasil proses *POS Tagging*. *Rule* yang digunakan pada penelitian ini untuk deteksi produk yang tertulis dalam sebuah *tweet* ditunjukkan pada Tabel 3.1.

Dalam merancang rule untuk estraksi produk yang tertulis dalam sebuah *tweet*, Groundtruth yang diguanakan sebanyak data 400 *tweet* . Dari data tersebut diamati secara manual mana tag dan kombinasi tag yang sering muncul sebagai produk, tag dan kombinasi tag yang sering muncul sebagai produk digunakan sebagi rule untuk deteksi produk. Berdasarakan hasil pengamatan didapatkan 8 rule yang dapat digunakan untuk mendeteksi produk yang tertulis dalam tweet.

Tabel 3.1 Rule deteksi Produk

No	*Rule*	Arti	Contoh
1	NN	Kata Benda	Laptop, kamera
2	NNG	Kata Benda Berpemilik	macbooknya
3	NNP	Kata Benda Bermakna	Iphone, Android
4	NN NN	Kata Benda diikuti dengan Kata Benda	Galaxy Note
5	NN RB	Kata Benda diikuti dengan Kata keterangan	Iphone 5S, Galaxy s6
6	NN CD	Kata Benda diikuti dengan Number	Iphone 5, Zenfone 5
7	NN NNP	Kata Benda diikuti dengan Kata Benda Bermakna	Redmi Note
8	NNP CD	Kata Benda Bermakna diikuti dengan Number	Note 6



Gambar 3.5 diagram alir ekstraksi produk dari sebuah *tweet* berbasis berbasis *grammatical tagging*

Pada Gambar 3.5, dapat dilihat bahwa proses ekstraksi produk dari sebuah tweet terdiri dari 2 tahapan utama. Tahap pertama adalah ekstraksi *tag* dari masing-masing kata dalam *tweet*, setelah didaptkan tag untuk masing-masing kata selanjutnya dilanjuykan ke tahapan kedua yaitu mendeteksi produk menggunakan *rule* berdasarkan hasil dari tahapan pertamat. Contoh prosess ekstrkasi produk pada sebuath tweet "iphone 6 bagus "dapat dilihat melalui langkah-langkah berikut ini :

1. Tokenisasi tweet menjadi masing-masing kata, sehingga menghasilkan kata iphone, 6 , bagus.
2. Menghitung *emission probability* setiap tag, contohnya tag CD (angka) berapa kemungkinan tag CD setelah tag NN (benda), berdasarkan data training, (data traiing memiliki 3 tag NN, CD, dan JJ)

$P(tag_{CD}|tag_{NN}) =$ 0.7 $P(tag_{jj}|tag_{jj}) =$ 0

$P(tag_{CD}\|tag_{jj}) =$	0.3	$P(tag_{jj}\|tag_{start}) =$	0.7
$P(tag_{CD}\|tag_{cd}) =$	0	$P(tag_{nn}\|tag_{cd}) =$	0.2
$P(tag_{CD}\|tag_{start}) =$	0	$P(tag_{nn}\|tag_{jj}) =$	0.4
$P(tag_{jj}\|tag_{NN}) =$	0.3	$P(tag_{nn}\|tag_{NN}) =$	0.3
$P(tag_{jj}\|tag_{cd}) =$	0.5	$P(tag_{nn}\|tag_{start}) =$	0.8

3. Menghitung *transition probability* setiap kata, contohnya adalah berapa kemungkinan angka 6 memiliki tag CD, berdasarkan data training

$P(word_{iphone}\|tag_{NN}) =$	1	$P(word_{6}\|tag_{JJ}) =$	0
$P(word_{iphone}\|tag_{cd}) =$	0	$P(word_{bagus}\|tag_{NN}) =$	0
$P(word_{iphone}\|tag_{jj}) =$	0	$P(word_{bagus}\|tag_{CD}) =$	0
$P(word_{6}\|tag_{NN}) =$	0	$P(word_{bagus}\|tag_{JJ}) =$	0.9
$P(word_{6}\|tag_{CD}) =$	0.8		

4. Setalah di dapat nilai *transition probability setiap kata, dan emission probability* setiap tag, dihitung nilai kemungkinan tag dengan cara mengkalikan *transition probability* dan *emission probability*

Tabel 3.2 Perhitungan probablitas kata ke tag

Word ke tag	***Transition probability***	**Menghitung *emission probability***	**Nilai probabilitas**
P(iphone,NN)	1	0.8	0.8
P(iphone,CD)	0	0	0
P(iphone,JJ)	0	0.7	0
P(6,CD)	0.8	0.7	0.56
P(6,JJ)	0	0.3	0
P(6,NN)	0	0.3	0
P(bagus,NN)	0	0.2	0
P(bagus,CD)	0	0	0
P(bagus,JJ)	0.9	0.5	0.45

Setiap kata, dipilih tag yang memliki nilai probablitas tertingi, sehingga menghasilkan iphone memiliki tag NN, 6 memiliki tag CD, dan Bagus memiliki tag JJ.

5. Setelah didapatkan semua tag dari masing-masing kata, maka tag tersebut akan dicocokkan dengan rule untuk mengetahui kata yang menjadi produk, berdasarkan tag yang didapat dan rule maka produk dalam tweet tersebut adalah iphone 6.

1.4.4 Pembobotan Produk

Metode yang diusulkan dalam melakukan tahap pembobotan dalam penelitian ini adalah metode pembobotan term, pembobotan jumlah *favourite* , dan pembobotan jumlah *retweet*. Pembobotan jumlah *favourite* dan jumlah *retweet* dimasukkan karena jika *tweet* tersebut di*retweet* dan *favourite* oleh orang lain, maka *tweet* tersebut dianggap penting. Persamaan pembobotannya seperti berikut :

1. **Pembobotan *term tweet***

Pembobotan term *tweet* adalah menghitung jumlah kemunculan term produk yang muncul didalam *tweet*. Perhitungan bobotnya mengikuti persamaan seperti berikut:

$$w_1(term_i) = \frac{|\, D_{term_i} \,|}{|\, D \,|}, \qquad (3.2)$$

dimana $w_1(\mathrm{term_i})$ adalah bobot term produk ke-*i*, $|\mathrm{D_{term_i}}|$ adalah total dokumen *tweet* yang memiliki term produk ke-*i*, dan $|\mathrm{D}|$ adalah total dokumen *tweet*.

2. **Pembobotan *term retweet***

Pembobotan term *retweet* adalah menghitung jumlah *Retweet* dari term produk yang terdapat pada *tweet*. Perhitungan bobotnya mengikuti persamaan seperti berikut:

$$\mathrm{w_2(term_i)} = \frac{\log(\sum_{\mathrm{j} \in \mathrm{tweet}} \mathrm{retweet(term_{i,} tweet_j)})}{\log(\underset{\substack{\mathrm{i} \in \mathrm{term} \\ \mathrm{j} \in \mathrm{tweet}}}{\mathrm{max\_retweet(term_{i,} tweet_j)}})}, \qquad (3.3)$$

dimana $w_2(\mathrm{term_i})$ adalah bobot *retweet* dari term produk ke-*i*, $\mathrm{retweet(term_{i,} tweet_j)}$, adalah jumlah *retweet* dari term produk ke-*i* pada tweet

ke-*j*, dan max_retweet($term_{i,}tweet_j$)adalah nilai maksimal dari jumlah *retweet* sebuah koleksi term ke-*i* di dalam koleksi *tweet*.

3. Pembobotan *term favourite*

Pembobotan term *favourite* adalah menghitung jumlah *favourite* dari term produk yang terdapat pada *tweet*. Perhitungan bobotnya mengikuti persamaan seperti berikut:

$$w_3(term_i) = \frac{\log(\sum_{j \in tweet} favourite(term_{i,} tweet_j))}{\log(\underset{\substack{i \in term \\ j \in tweet}}{max\_favourite(term_{i,} tweet_j)})} \quad (3.4)$$

dimana $w_3(\text{term}_i)$ adalah bobot favourite dari term produk ke-*i*, favourite($term_{i,}tweet_j$), adalah jumlah $favourite$ dari term produk ke-*i* pada tweet ke-*j*, dan max_favourite($term_{i,}tweet_j$) adalah nilai maksimal dari jumlah *retweet* sebuah koleksi term ke-*i* didalam koleksi *tweet.*

4. Bobot total *term*

Berdasarkan 3 pembobotan diatas term, *retweet* dan *favourite* , maka nilai total bobot dari term produk ke i adalah sebagai berikut.

$$w_t(term_i) = (a * w_1(term_i) + b * w_2(term_i) + c * w_3(term_i) \quad (3.5)$$

Nilai bobot $a, b,$ dan c jika di total harus bernilai 1, dimana nanti dalam tahap pengujian dilakukan penentuan bobot *a*, *b*, dan *c* yang optimal, sehingga bisa mengetahui bobot mana yang paling berpengaruh pada ekstrkasi produk.

1.4.5 Asosiasi dan visualisasi hubungan antara fitur kata-kata produk.

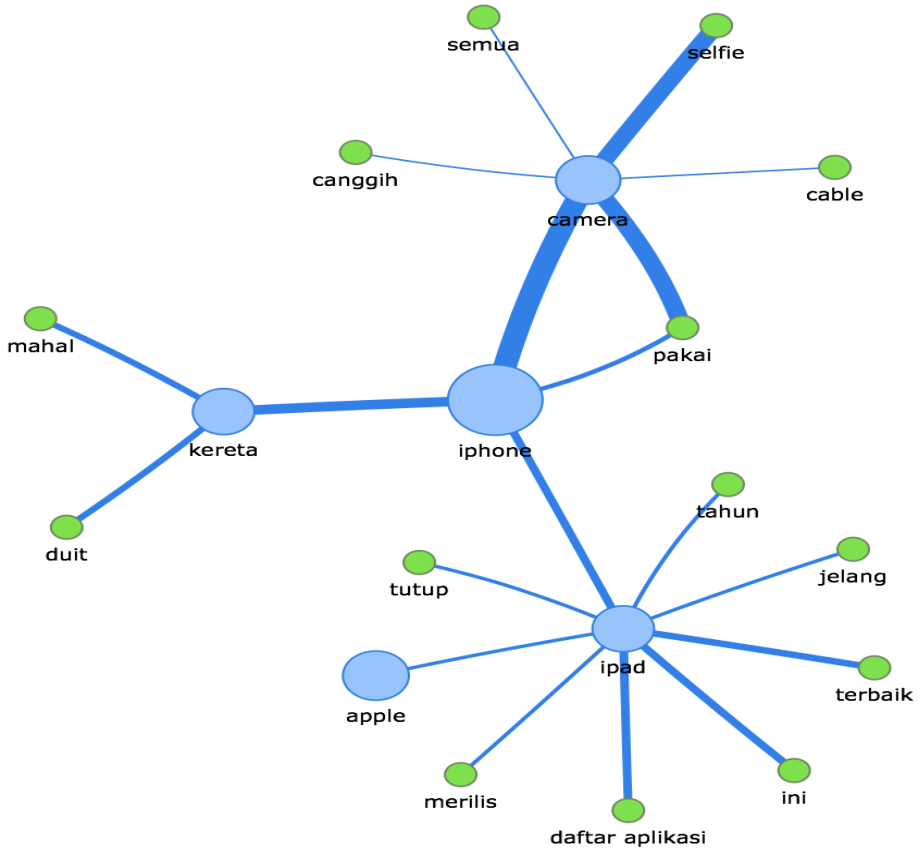
Agar mendapatkan informasi tentang keyword dari sebuah produk, diperlukan suatu metode untuk menghitung kedekatan antar kata. Dengan menghitung nilai *confidence* berdasarkan metode *Association Rule* antara produk dan keyword, kita dapat mengetahui seberapa kuat hubungan antar produk dan keyword tersebut dalam kumpulan *tweet*. Perhitungan *confidence* mengikuti persamaan seperti berikut:

$$c = \frac{s(\text{A,B})}{s(A)} \quad (3.6)$$

Dimana c adalah nilai *confidence* antara produk dan keyword, $s(A, \mathrm{B})$ adalah jumlah *tweet* yang mengandung produk dan kata kunci , dan $s(A)$ adalah jumlah *tweet* yang mengandung produk.

Setelah mendapatkan nilai *confidence* maka dilakukan penyaringan relasi *tag*et dengan kata kunci berdasarkan nilai *confidence* yang diperoleh. Penyaringan tersebut bertujuan untuk mendapatkan relasi produk dengan kata kunci yang benar-benar kuat. Setelah mendapatkan hasil dari penyaringan, dibuat visualisasi dalam bentuk graph, sehingga memudahkan pengguna untuk menganalisanya. Graph yang digunakan dalam peneitian ini adalah graph berbentuk network dangan 2 buah tipe node. Node kesatu adalah produk yang sedang popular dari sebuah brand dan node kedua adalah kata kunci yang sedang popular dari produk tersebut. Berikut ini contoh hasil dari graph relasi produk dan kata kunci.

Gambar 3.6 menunjukkan bawha produk iphone, camera, dan selfie dengan sedang banyak diperbincangkan dalam kumpulan tweet. Kata kunci yang lain seperti cable, canggih dan semua lebih sedikit di perbicangkan dari camera selfie. Itu berarti bahwa pada periode tweet tersebut banyak orang yang membicarakan selfie menggunakan camera iphone. Contoh tweet yang mengndung kata iphone, camera dan selfie adalah "pakai iphone tapi selfie pakai back camera. pastu tangan pulak block. selfie pebend a? https://t.co/emy9kwqdea" dan ""weh iphone xde camera depan ye" "ada npe kau ckp cm tu" "ye la aku tgk smua selfie pakai cermin je""",



Gambar 3.6 Contoh hasil visualisasi hubungan antara fitur kata-kata produk pada tweet Apple

3.5 Metode Pengujian

Hasil uji coba akan dievaluasi sehingga dapat dilihat kinerja metode yang diajukan. Ukuran evaluasi yang digunakan adalah *precision*, dan *recall*. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. dihitung berdasarkan perbandingan hasil yang dihasilkan oleh sistem dan pakar, persamaannya adalah :

$$Precision = \frac{true\ positive}{true\ positive + false\ positive}, \quad (3.5)$$

Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.

$$Recall = \frac{true\ positive}{true\ positive + false\ negatif}, \quad (3.6)$$

F-measure adalah nilai *harmonic mean* dari *precision* dan *recall*.

$$F - measure = 2\ \frac{Precision * Recall}{Precision + Recall}, \quad (3.6)$$

Hal-hal yang akan dievaluasi adalah hasil ekstraksi produk dan kata kunci, uji coba *threshold*, uji coba nilai *a,b,c* dan *k* pada pembobotan, dan uji coba hasil asosiasi untuk pembentukan *graph*.

Evaluasi hasil ekstraksi produk dan kata kunci adalah dengan menggunakan nilai *precision*, *recall,* dan f-meausure. Evaluasi ini bertujuan untuk mengetahui apakah sebarapa baik hasil ekstraksi produk dan kata kunci oleh sistem.

Evaluasi Uji coba *threshold* menggunakan nilai *threshold* yang berbeda-beda pada waktu penyaringan produk, kata kunci *tweet* dan asosiasi antara terget *tweet* dan kata kunci *tweet*s. Evaluasi Uji coba ini bertujuan untuk mendapatkan nilai *threshold* optimal yang akan digunakan dalam penyaringan produk, kata kunci *tweet*, dan asosiasi antara terget *tweet* dan kata kunci *tweet*s.

Evaluasi Uji coba *a, b,* dan *c* pada pembobotan menggunakan nilai *a, b,* dan *c* yang berbeda-beda. Skenario ini bertujuan untuk mendapatkan nilai *a, b,* dan *c* yang optimal. Nilai *a, b,* dan *c* digunakan dalam pembobotan produk agar dapat mengetahui bobot mana yang paling berpengaruh pada pengekstrasian produk dari sebuah *tweet*.

Evaluasi hasil asosiasi produk dan kata kunci adalah dengan menggunakan nilai f-meausure dari pemfilteran nilai *confidence*. Skenario ini bertujuan untuk mengetahui apakah seberapa baik hasil relasi produk dan kata kunci yang dihasilkan oleh sistem.

BAB IV
IMPLEMENTASI DAN PENGUJIAN

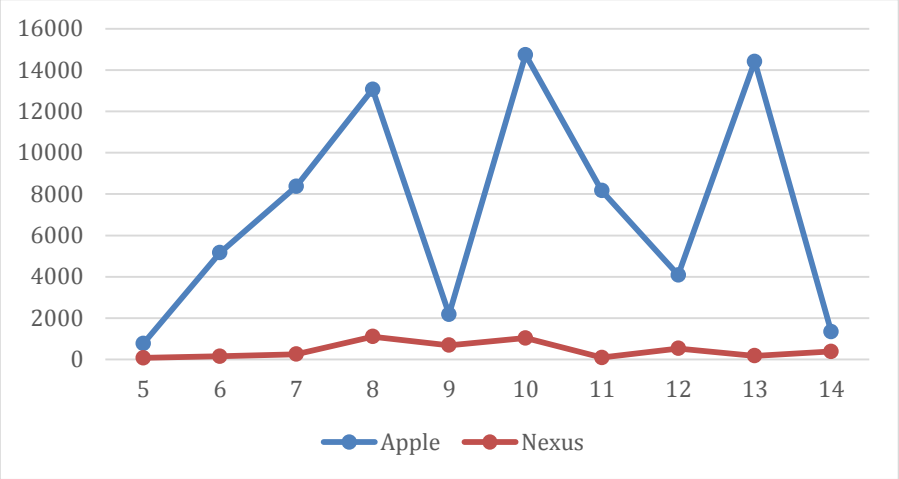
Pada sub bab ini akan membahas tentang implementasi, pengujian dan pembahasan terkait penelitian yang diusulkan. Tahapan implementasi yang dilakukan sesuai dengan alur pada Gambar 3.1 yang terdiri dari prepossessing *tweet*, *prepocessing text* dan ekstraksi kata kunci, ekstraksi produk dari sebuah *tweet* berbasis *grammatical tag*ging, pembobotan dan pemfilteran produk dan asosiasi serta visualisasi hubungan antara fitur kata-kata produk. Tahap berikutnya adalah pengujian dari implementasi yang telah dilakukan. Skenario pengujian sesuai dengan skenario yang telah direncanakan sebelumnya pada subbab 3.5 tentang perancangan pengujian. Tahapan tekahir dari bab ini adalah pembahasan tentang hasil dan evaluasi ekstraksi kata kunci dari metadata twitter berbahasa indonesia dengan pendekatan *grammatical tag*ging untuk visualisasi hubungan antara fitur kata-kata produk.

4.1 Data Uji Coba

Data uji coba yang digunakan dalam penelitian ini berasal dari data *tweet* atau dokumen *tweet* dengan memanfaatkan search API yang disediakan oleh twitter. Sebuah aplikasi dibangun untuk mengambil data *tweet* tersebut dari twitter dengan menggunakan search API dengan query languange ID yang berarti *tweet* yang berbahasa Indonesia yang diambil. Data *tweet* hanya dibatasi topik-topik produk dari brand antara lain : Apple dan Nexus. Data *tweet* dikumpulkan atau dikoleksi selama 10 hari dari tanggal 5 Desember 2015 sampai 14 Desember 2015. Seluruh data *raw tweet*s disimpan dalam format *json,* yang berisi seluruh metadata twitter..

Grafik 4.1 adalah hasil Streaming API selama 10 hari dari data Twitter untuk seluruh produk dari brand Apple dan Nexus. Dari Grafik 4.1 dapat dilihat pada kedua brand memiliki jumlah *tweet* yang cukup banyak setiap harinya. Jumlah *tweet* terkait produk dari Apple Samsung sangat tinggi pada tanggal 10 Desember sebesar 14.375. Jumlah *tweet* untuk Nexus tertinggi adalah 1.032 pada tanggal 10 Desember. Sedangkan untuk rata-rata *tweet* masing-masing produk

adalah Apple sebesar 72.286 *tweet*, dan Nexus sebesar 4.486 *tweet*, seluruh data tersebut akan digunakan dalam ujicoba.



Gambar 4.1 Jumlah data selama 10 hari semua brand

Tabel 4.1 adalah hasil contoh representasi data *tweet* dari brand nexus. Dari tabel tersebut terdapat dapat diketahui user yang melakukan *tweet*, text yang *tweet*, jumlah *tweet* di *retweet* dan jumlah *tweet favourite* , tidak semua *tweet* memiliki nilai *retweet* dan nilia *favourite* .

Tabel 4.1 Contoh representasi data *tweet* apple

User	*Tweet*	Jumlah *retweet*	Jumlah *favourite*
@AlamTekno	**iphone 6s** & 6s plus **berbalut emas** dibanderol rp 160 juta	0	0
@saramahayuddin	pakai **iphone** tapi **selfie** pakai back **camera**. pastu tangan pulak block. selfie pebenda? https://t.co/emy9kwqdea	6494	0
@info_jakarta	3 games moba terbaik di **ipad**, **iphone**, dan ipod touch https://t.co/8qofcv2408	2	0
@detikinet	ini daftar aplikasi terbaik iphone dan ipad https://t.co/xvkks84riq	6	2
@infokuisnetwork	ontes catat tanggalnya lazada1212 berhadiah **iphone 6s**. #lazada1212	4	8

Tabel 4.1 Contoh representasi data *tweet* apple (lanjutan)

User	*Tweet*	Jumlah *retweet*	Jumlah *favourite*
@Sheqal_Rosli	**iphone** users semakin meningkat	0	1
@syg__malaysia	**iphone** tak dapat beli ilmu. ilmu dapat beli **iphone**, rumah, **kereta**. sabar, jangan sia-siakan usia muda hanya untuk perkara tak penting	26372	0

4.2 Implementasi Metode

Metode dalam penelitian ini diimplementasikan dengan didukung oleh hardware dan software dengan spesifikasi berikut ini.

a. Perangkat Keras

Perangkat keras yang digunakan untuk implementasi dan pengujian adalah satu buah laptop dengan spesifikasi processor Intel Core i5-2410M 2.3 GHz, RAM 8 Gb.

b. Perangkat Lunak

Perangkat lunak yang digunakan pada tahapan implementasi dan pengujian adalah Sistem operasi OS X 10.11 64 bit dan aplikasi NetBeans 8.0.3 dengan JDK versi 8.

Metode yang diusulkan terdiri atas empat bagian utama yaitu: praproses (preprocessing), ekstraksi kata kunci, ekstraksi produk, dan Asosiasi dan visualisasi hubungan antara fitur kata-kata produk.

4.2.1. Implementasi Tahap Praproses tweet

Pada tahap praproses *tweet* dimulai dengan membaca data *tweet* dari hasil *serach API* yang telah disimpan dalam format json seperti pada Gambar 4.1. Metada *tweet* yang digunakan dalam pengujian ini adalah text, *retweet*_count dan *favourite*s_count, untuk metadata lainnya seperti location, coordinate tidak digunakan. sebeleum dilakukan praproses terlebih dahulu dibuat sebuah system untuk mengekstrak value dari meta data text, *retweet*_count dan *favourite* s_count.

{"metadata":{"result_type":"recent","iso_language_code":"in"},"in_reply_to_status_id_str":null,"in_reply_to_status_id":null,"created_at":"Mon Oct 12 15:28:37 +0000 2015","in_reply_to_user_id_str":null,"source":"dlvr.it","***retweet*_count**":6,"*retweet*ed":false,"geo":null,"in_reply_to_screen_name":null,"is_quote_status":false,"id_str":"653593119351439362","in_reply_to_user_id":null,"**favorite_count**":1,"id":653593119351439362,"**text**":"Nexus 4 Bisa Pakai Android 6.0 Marshmallow http://t.co/pa1wL1XlGj","place":null,"lang":"in","favorited":false,"possibly_sensitive":false,"coordinates":null,"truncated":false,"entities":{"urls":[{"display_url":"dlvr.it/CQbXWp","indices":[43,65],"expanded_url":"http://dlvr.it/CQbXWp","url":"http://t.co/pa1wL1XlGj"}],"hash*tag*s":[],"user_mentions":[],"symbols":[]},"contributors":null,"user":{"utc_offset":25200,"friends_count":326,"profile_image_url_https":"https://pbs.twimg.com/profile_images/378800000685890581/9176627ad02dad36dd783dbfbf4bdc98_normal.png","listed_count":5,"profile_background_image_url":"http://abs.twimg.com/images/themes/theme13/bg.gif","default_profile_image":false,"*favourite*s_count":1,"description":"Just an ordinary man with extraordinary dreams, and cat lovers too. Follow me and get news update about internet world with my own style.","created_at":"Fri Dec 11 16:45:58 +0000 2009","is_translator":false,"profile_background_image_url_https":"https://abs.twimg.com/images/themes/theme13/bg.gif","protected":false,"screen_name":"tenovid","id_str":"96160815","profile_link_color":"93A644","is_translation_enabled":false,"id":96160815,"geo_enabled":false,"profile_background_color":"B2DFDA","lang":"en","has_extended_profile":false,"profile_sidebar_border_color":"EEEEEE","profile_text_color":"333333","verified":false,"profile_image_url":"http://pbs.twimg.com/profile_images/378800000685890581/9176627ad02dad36dd783dbfbf4bdc98_normal.png","time_zone":"Jakarta","url":"http://t.co/BOLQR6YJ5D","contributors_enabled":false,"profile_background_tile":false,"profile_banner_url":"https://pbs.twimg.com/profile_banners/96160815/1388453537","entities":{"description":{"urls":[]},"url":{"urls":[{"display_url":"napnipnop.com","indices":[0,22],"expanded_url":"http://www.napnipnop.com/","url":"http://t.co/BOLQR6YJ5D"}]}},"statuses_count":18811,"follow_request_sent":false,"followers_count":1705,"profile_use_background_image":true,"default_profile":false,"following":false,"name":"Tendi Noviandi","location":"Bekasi, West Java, Indonesia","profile_sidebar_fill_color":"FFFFFF","notifications":false}}

Gambar 4.2 Contoh Raw JSON *Tweet* dari Hasil Search API

Setiap text tweet dilakukan proses untuk memformalkan text tweet, langkah-langkah yang dilakukan untuk memformalkan text tweet terdapat 4 langkah yaitu:

1. Memisahkan angka di depan atau di belakang text dengan text yang mengikutinya. Contohnya “2hari”, menjadi 2 hari

2. Mengubah angka didalam text menjadi huruf, contohnya “ga2l” menjadi gagal
3. Menghapus huruf yang berulang, contohnya adalah kata “tidaaak”, menjadi tidak.
4. Mengubah kata allay menajdi kata formal berdasarkan *dictionary* yang didapat dari http://kamusmania.com/component/glossary/Kamus-Alay-9/, contonya “ciyus” menjadi serius.

Dari hasil *prepocessing text tweet*s akan diperoleh data text *tweet*s yang menggunakan kata-kata formal, data tersebut akan dijadikan sebagai text yang akan di proses ke tahapan selanjutnya yaitu ektraksi kata kunci dan ekstraksi produk. Text *tweet* sebelum dan sesudah praproses untuk brand Apple dapat dilihat pada tabel 4.2 dan 4.3 sedangkan untuk brand nexus dapat dilihat pada tabel 4.4 dan 4.5.

Tabel 4.2 Contoh data tweet apple sebelum praproses

No	Tweet
1	iphone 6s & 6s plus berbalut emas dibanderol rp 160 juta https://t.co/idyfdpxgv0
2	iphone 6s & 6s plus berbalut emas dibanderol rp 160 juta: tanpa embel-embel edisi khusus pun, harga iphone https://t.co/rvvadqsfvv
3	iphone tak dapat beli ilmu. ilmu dapat beli iphone, rumah, kereta. sabar, jangan **sia2kan** usia muda hanya untuk perkara tak penting
4	rt @saramahayuddin: pakai iphone tapi selfie pakai back camera. pastu tangan pulak block. selfie pebenda? https://t.co/emy9kwqdea
5	maksa kali buat ngecas iphone, hape aja canggih , cas ? https://t.co/cys9sgvpuz
6	alhamdulillah, akhirnya aku bisa nebus **iphone6** aku besok sebelum masuk kerja, semua berkat usaha aku berdagang ini
7	**ciyus**, ini daftar aplikasi terbaik iphone dan ipad https://t.co/pwmdkhfeqg
8	ini daftar aplikasi terbaik iphone dan ipad: jelang tutup tahun, Apple selalu merilis daftar aplikasi dan game https://t.co/4eabitubyb

Tabel 4.2 Contoh data *tweet* apple sebelum praproses (lanjutan)

No	Tweet

No	Tweet
9	aku tak pernah jeles dgn org yang bawa kereta mahal, ada iphone, ada duit banyak. yg aku jeles dengan org yang kurus, makan banyak
10	iphone ni memang **canggiiih**, camera lawan apa semua. cable paling babi antara semua handphone

Tabel 4.3 Contoh data *tweet* apple sesudah praproses

No	Tweet
1	iphone 6s & 6s plus berbalut emas dibanderol rp 160 juta https://t.co/idyfdpxgv0
2	iphone 6s & 6s plus berbalut emas dibanderol rp 160 juta: tanpa embel-embel edisi khusus pun, harga iphone https://t.co/rvvadqsfvv
3	iphone tak dapat beli ilmu. ilmu dapat beli iphone, rumah, kereta. sabar, jangan **siasiakan** usia muda hanya untuk perkara tak penting
4	rt @saramahayuddin: pakai iphone tapi selfie pakai back camera. pastu tangan pulak block. selfie pebenda? https://t.co/emy9kwqdea
5	maksa kali buat ngecas iphone, hape aja canggih , cas ? https://t.co/cys9sgvpuz
6	alhamdulillah, akhirnya aku bisa nebus **iphone 6** aku besok sebelum masuk kerja, semua berkat usaha aku berdagang ini
7	**serius**, ini daftar aplikasi terbaik iphone dan ipad https://t.co/pwmdkhfeqg
8	ini daftar aplikasi terbaik iphone dan ipad: jelang tutup tahun, Apple selalu merilis daftar aplikasi dan game https://t.co/4eabitubyb
9	aku tak pernah jeles dgn org yang bawa kereta mahal, ada iphone, ada duit banyak. yg aku jeles dengan org yang kurus, makan banyak
10	iphone ni memang **canggih**, camera lawan apa semua. cable paling babi antara semua handphone

Tabel 4.2 menunjukkan 10 contoh tweet yang berisikan produk apple sebelum praprosess, sedangkan di Tabel 4.3 menunjukkan 10 contoh *tweet* sesudah di praprosess. Dapat dilihat bahwa terdapat kata-kata yang berubah, antara lain sia2kan menjadi siasiakan 3, ciyus menjadi serius, iphone6 menjadi iphone 6 dan canggiiih menjadi canggih.

Tabel 4.4 Contoh data *tweet* nexus sebelum praproses

No	Tweet
1	adu gahar smartphone nexus 6p & moto x pure edition
2	google setop jual **nexus6** : tak sampai **2bulan** setelah merilis nexus 5x dan nexus 6p, google menyetop penjual.
3	lg nexus 5x, smartphone klasik & tangguh: lg nexus 5x dirancang dengan model klasik dengan tampilan sederhana https://t.co/r25hrkzgk8
4	hisense pureshot, pureshot+ nexus , lg g3, lg g4, dll @smartfrenworld @metro_tv #unlimitedsmartfren #goforit https://t.co/ik0ow8qjsr ∗1 = 0 0
5	nexus 6p international #giveaway androidauth graajkumaar
5	huawei nexus terbaru akan disemati snapdragon 820: horisonshenzhen, (prlm).- untuk perangkat huawei nexus https://t.co/9xgx8ljhiz
7	camera test: nexus 6p vs **iphone6** - http://t.co/lwy5y1nkli http://t.co/ymhyernk5k #googlenexus6 #nexus6
8	**nexus4** bisa pakai android 6.0 marshmallow http://t.co/hqpcz8t2x3
9	motorola nexus 6p review https://t.co/b0ufnrepdy
10	sementara itu kamar sebelah udah banyak yang ngepost foto nexus 6p atau 5x, nggak perlu nunggu resmi di indonesia :)

Tabel 4.5 Contoh data tweet nexus sesudah praproses

No	Tweet
1	adu gahar smartphone nexus 6p & moto x pure edition
2	google setop jual **nexus 6** : tak sampai **2 bulan** setelah merilis nexus 5x dan nexus 6p, google menyetop penjual.
3	lg nexus 5x, smartphone klasik & tangguh: lg nexus 5x dirancang dengan model klasik dengan tampilan sederhana https://t.co/r25hrkzgk8
4	hisense pureshot, pureshot+ nexus , lg g3, lg g4, dll @smartfrenworld @metro_tv #unlimitedsmartfren #goforit https://t.co/ik0ow8qjsr ∗1 = 0 0
5	nexus 6p international #giveaway androidauth graajkumaar
6	huawei nexus terbaru akan disemati snapdragon 820: horisonshenzhen, (prlm).- untuk perangkat huawei nexus https://t.co/9xgx8ljhiz

Tabel 4.5 Contoh data tweet nexus sesudah praproses (lanjutan)

No	Tweet
7	camera test: nexus 6p vs **iphone 6** - http://t.co/lwy5y1nkli http://t.co/ymhyernk5k #googlenexus6 #nexus6
8	**Nexus 4** bisa pakai android 6.0 marshmallow http://t.co/hqpcz8t2x3
9	motorola nexus 6p review https://t.co/b0ufnrepdy
10	sementara itu kamar sebelah udah banyak yang ngepost foto nexus 6p atau 5x, nggak perlu nunggu resmi di indonesia :)

Tabel 4.4 menunjukkan 10 contoh tweet yang berisikan produk nexus sebelum praprosess, sedangkan di Tabel 4.5 menunjukkan 10 contoh *tweet* sesudah di praprosess. Dapat dilihat bahwa terdapat kata-kata yang berubah, antara lain note5 menjadi note 5, iphone6 menjadi iphone 6 dan nexus4 menjadi nexus 4.

4.2.2. Ekstraksi kata kunci

Pada tahap ekstraksi fitur kata kunci *tweet* terdapat 2 proses tahapan, tahapan pertama adalah *prepocessing text* yang meliputi *Case folding*, *Tokenizing*, *Filtering* dan *Stemming*. Tahapan kedua adalah pemfilteran kata kunci dengan berdasarkan kemunculan kata di dalam *tweet*, persamaan yang digunakan adalah persamaan 3.1.

Pemfilteran kata kunci digunakan untuk mereduksi dimensi kata kunci yang didapat pada tahapan selanjutnya sehingga akan mempecepat komputasi dan hanya kata kunci yang sering muncul yang akan dipakai. Kata kunci yang melebihi nilai *threshold* akan digunakan sebagai kata kunci yang akan diasosiasikan dengan produk yang diperoleh pada tahapan ekstraksi produk.

Tabel 4.6 menunjukkan contoh data kata kunci hasil ektraksi dari 7123 kata kunci yang didapat dari *tweet* yang mengandung produk Apple. Dari pengamatan pada Tabel 4.6, tidak semua kata kunci berhubungan dengan produl brand Apple, hanya beberapa yang berhubungan antara lain mahal, cable, canggih dan selfie. Sedangkan untuk yang lainnya tidak berhubungan dengan produk Apple. Pengujian *threshold* untuk produk brand Apple akan dijelaskan pada subbab 4.3.1 c. Kata “pakai” memiliki bobot tertinggi, padahal kata pakai tidak

langsung berhubungan dengan produk, dikarenakan kata pakai muncul pada 11434 *tweet*.

Tabel 4.6 Contoh hasil ekstraksi kata kunci *tweet* brand Apple

Kata Kunci	Bobot
cable	0.59
canggih	0.59
mahal	0.63
merilis	0.66
tahun	0.66
terbaik	0.67
semua	0.70
selfie	0.76
pakai	0.85

Tabel 4.7 menunjukkan beberapa data kata kunci hasil ekstraksi dari 790 kata kunci yang didapat yang mengandung produk nexus. Dari pengamatan pada Tabel 4.7, tidak semua kata kunci berhubungan dengan produl brand Nexus, hanya beberapa yang berhubungan antara lain snapdragon, androidauth, pureshot dan lg. Sedangkan untuk yang lainnya tidak berhubungan dengan produk nexus. Pengujian *threshold* untuk produk brand nexus akan dijelaskan pada subbab 4.3.2 c. Kata lg memiliki bobot tertinggi, dikarenakan kata pakai muncul pada 1017 *tweet*.

Tabel 4.7 Contoh hasil ekstraksi kata kunci *tweet* brand Nexus

Kata Kunci	Bobot
snapdragon	0.35
merilis	0.71
jual	0.72
androidauth	0.74
#giveaway	0.75

Tabel 4.7 Contoh hasil ekstraksi kata kunci *tweet* brand Nexus (lanjutan)

Kata Kunci	Bobot
international	0.75
pureshot	0.76
lg	0.82

4.2.3. Ekstraksi Produk dari *Tweet* menggunakan Hidden Markov Model POS Tagging

Pada tahapan ektraksi kata yang menjadi produk didalam *tweet* terdapat 2 proses tahapan agar dapat mengetahui produk apa yang sedang dibicarakan dalam sebuah *tweet*, yaitu ekstrakasi *tag* menggunakan *grammatical tag*ging dan penyeleksian *term* yang didapat menggunakan rula yang di buat pada tabel 3.1 Ekstraksi *tag* berbasis *grammatical* menggunakan metode *Hidden Markov Model* yang telah dijelaskan pada subbab 2.5. Data training yang digunakan untuk ekstraksi tag berjumlah 7534 tweet. Tahapan yang dlikukan pada waktu ektsrkasi tag adalah :

1. Menghitung nilai kimiripan atau kemungkinan suatu kata sebagai sebuah tag (*emission probability*).
2. Menghitung nilai kemungkinan transisi suatu tag dari tag sebelumnya (*transition probability*).
3. Menghitung nilai probabilitas suatu tag ke kata, yaitu dengan cara mengkalikan nilai *emission probability* dengan nilai *transition probability*.
4. Dipilih nilai suatu tag dengan nilai maksimal probabilitas yang akan dijidakan tag dari sebuah kata.

Tabel 4.8 menunjukkan hasil extraksi *tag* dari *tweet* mengandung produk brand Apple dan tabel 4.10 menunjukkan hasil extraksi *tag* dari *tweet* yang mengandung produk brand nexus.

Berdasarkan tabel 4.8 setiap kata dari *tweet* yang mengandung produk Apple memiliki *tag*ing, dari *tag*ging tersebut dilakuakan pencocokan *rule* berdasarkan tabel 3.1, sehingga bisa didapatkan produk dari sebuah *tweet*. Setelah didapatkan produk maka dilakukan perhitungan pembobotan produk tersebut berdasarkan nilai kemunculan, nilai *retweet* dan nilai *favourite* seperti pada

persamaan 3.5. hasil dari nilai persamaan tersebut akan digunakan untuk memfilter produk mana yang akan digunakan pada tahapan selanjutnya asosiasi produk dan *tweet*. Pemfilteran terebut bertujuan untuk mereduksi data dan memilih produk yang penting.

Tabel 4.8 Contoh hasil Grammatical *tagging* data *tweet* apple

No	Tweet
1	**iphone_NNP 6s_CD** &_SYM 6s_CD plus_FW berbalut_VBT emas_NN dibanderol_VBT rp_FW 160_CD juta_NN https://t.co_idyfdpxgv0_U
2	**iphone_NNP 6s_CD** &_SYM 6s_CD plus_FW berbalut_VBT emas_NN dibanderol_VBT rp_FW 160_CD juta:_NN tanpa_IN embel-embel_NN edisi_NN khusus_JJ pun_RP ,_PRP harga_NN **iphone_NNP** https://t.co_rvvadqsfvv_U
3	**iphone_NNP** tak_NEG dapat_MD beli_VBT ilmu_NN ._. ilmu_NN dapat_MD beli_VBT **iphone_NNP** ,_WP rumah_NN ,_PRP **kereta_NN** ._. sabar_JJ ,_PRP jangan_NEG siasiakan_VBT usia_NN muda_JJ hanya_RB untuk_SC perkara_NN tak_NEG penting_JJ
4	rt_RT @saramahayuddin:_@ pakai_VBI iphone_NNP tapi_CC selfie_VBI pakai_VBI back_FW **camera_NN** ._. pastu_RB tangan_NN pulak_NN block_FW ._. selfie_VBI pebenda_NN ?_SYM https://t.co_emy9kwqdea_U
5	maksa_NN kali_NN buat_IN ngecas_VBT **iphone_NNP** ,_PRP hape_VBI aja_RB canggih_JJ ,_PRP cas_NN ?_SYM https:__t.co_cys9sgvpuz_U
6	alhamdulillah_UH ,_, akhirnya_RB aku_PRP bisa_MD nebus_VBT **iphone_NNP 6_CD** aku_PRP besok_NN sebelum_RB masuk_VBI kerja_VBI ,_WP semua_CDI berkat_NN usaha_VBI aku_PRP berdagang_VBT ini_DT
7	serius_JJ ,_, ini_DT daftar_NN aplikasi_NN terbaik_JJ **iphone_NNP** dan_CC ipad_NN https:__t.co_pwmdkhfeqg_U

Tabel 4.8 Contoh hasil *Grammatical* tagging data *tweet* apple (lanjutan)

No	Tweet
8	ini_DT daftar_NN aplikasi_NN terbaik_JJ **iphone_NNP** dan_CC **ipad**_NN jelang_VBI tutup_VBI tahun_NN ,_PRP **Apple_NNP** selalu_RB merilis_VBT daftar_NN aplikasi_NN dan_CC game_FW https://t.co_4eabitubyb_U
9	aku_PRP tak_NEG pernah_RB jeles_JJ dgn_CC org_NN yang_SC bawa_VBT kereta_NN mahal_JJ ,_PRP ada_VBT **iphone_NN** ,_PRP ada_VBT duit_NN banyak_CDI ._. yg_SC aku_PRP jeles_VBI dengan_SC org_NN yang_SC kurus_JJ ,_PRP makan_VBT banyak_CDI
10	**iphone_NNP** ni_DT memang_RB canggih_JJ ,_PRP **camera_NN** lawan_RB apa_WP semua_CDI ._. cable_VBT paling_RB babi_NN antara_IN semua_CCI handphone_FW

Tabel 4.9 Contoh hasil ekstraksi produk pada *tweet* Apple

Produk	*Rule* yang digunakan	Bobot
iphone	NNP	0.48
apple	NNP	0.24
camera	NN	0.23
iphone 6s	NNP CD	0.22
kereta	NNP	0.21
iphone 6	NNP CD	0.20

Tabel 4.9 menujukkan beberapa contoh hasil dari ekstraksi prouk, *rule* yang digunakan serta nilai bobot produk dari tweet Apple dapat dilihat pada tabel 4.9. nilai bobot terbesar ada di prdouk iphone sedangkan nilai terkecil ada di produk iphone 6. Semakin besar nilai bobot produk, itu berarti produk tersebut semakin penting. Visualissi dapat dilihat pada gambar 4.2

Berdasarkan tabel 4.10 setiap kata dari *tweet* yang mengandung produk nexus memiliki *tag*iing, dari *tag*ging tersebut dilakuakan pencocokan *rule* bedasrakan tabel 3.1, sehingga bisa didapatkan produk dari sebuah *tweet*. Setelah didapatkan kata yang menjadi produk maka dilakukan perhitungan pembobotan produk berdarakan persamaan 3.5. hasil dari nilai persamaan tersebut akan

digunakan untuk memfilter produk mana yang akan digunakan pada tahapan selanjutnya asosiasi produk dan kata kunci. Pemfilteran terebut bertujuan untuk mereduksi data dan memilih produk yang penting.

Tabel 4.10 Contoh hasil *Grammatical tagging tweet* brand Nexus

No	Tweet
1	adu_VBI gahar_NNP smartphone_FW **nexus_NNP 6p_CD** &_SYM moto_VBT x_NNP pure_NN edition_NNP https://t.co_gaxqwjklfq_U
2	**google_NNP** setop_RB jual_VBI **nexus_NNP 6_CD** :_: tak_NEG sampai_VBI 2_CD bulan_NN setelah_RB merilis_VBT **nexus_NNP 5x_CD** dan_CC **nexus_NNP 6p_CD** ,_PRP **google_NNP** menyetop_VBT penjual_NN ._
3	lg_RB **nexus_NNP 5x_CD** ,_PRP smartphone_FW klasik_NN &_SYM tangguh:_NN lg_RB **nexus_NNP 5x_CD** dirancang_NN dengan_SC model_NN klasik_NN dengan_SC tampilan_NN sederhana_JJ https://t.co_r25hrkzgk8_U
4	hisense_VBI pureshot_RB ,_PRP pureshot+_RB **nexus_NNP** ,_WP lg_RB g3_NN ,_PRP lg_RB g4_RB ,_PRP dll_FW @smartfrenworld_@ @metro_tv_@ #unlimitedsmartfren_# #goforit_# https:__t.co_ik0ow8qjsr_U
5	**nexus_NNP 6p_CD** international_NN #giveaway_# androidauth_NNP https://t.co_d879fe2frg_U graajkumaar_JJ
6	huawei_NNP **nexus_NNP** terbaru_JJ akan_MD disemati_VBT snapdragon_JJ 820:_NN horisonshenzhen_NN ,_PRP (prlm).-_OP untuk_IN perangkat_NN huawei_NNP nexus_NNP https://t.co_9xgx8ljhiz_U
7	camera_NN test:_NN **nexus_NNP 6p_CD** vs_CC **iphone_NNP 6_CD** -_- http:__t.co_lwy5y1nkli_U http://t.co_ymhyernk5k_U #googlenexus6_# #nexus6_#
8	**nexus_NNP 4_CD** bisa_MD pakai_VBI android_NN 6.0_CDP marshmallow_NNP, http://t.co_hqpcz8t2x3_U
9	motorola_NNP **nexus_NNP 6_CD** review_NN https://t.co_b0ufnrepdy_U
10	sementara_RB itu_DT kamar_NN sebelah_PRL udah_RB banyak_CDI yang_SC ngepost_VBT foto_NN **nexus_NNP 6p_CD** atau_CC 5x_RB ,_PRP nggak_NEG perlu_RB nunggu_VBT resmi_JJ di_IN indonesia_NNP :)_E

Tabel 4.11 Contoh hasil ekstraksi produk pada tweet *brand* Nexus

Produk	*Rule* yang digunakan	Bobot
nexus	NNP	0.47
Nexus 6p	NNP CD	0.42
google	NNP	0.27
Nexus 6	NNP CD	0.25
Nexus 5x	NNP CD	0.18

Tabel 4.11 menujukkan beberapa contoh hasil dari ekstraksi produk, *rule* yang digunakan serta nilai bobot produk dari brand nexus dapat dilihat pada tabel 4.11. nilai bobot terbesar ada di produk nexus dan nexus 6p sedangkan nilai terkecil ada di produk nexus 5x. Semakin besar nilai bobot produk, itu berarti produk tersebut semakin penting. Visualisi dapat dilihat pada gambar 4.3

4.2.4. Asosiasi dan visualisasi hubungan antara fitur kata-kata produk.

Setalah didapatkan kata kunci dan produk dari *tweet* dilkukakan asosiasi produk dengan kata kunci supaya dapat mengetahui apakah kedua kata tersebut berhubungan atau tidak. Perhitungan asosiasi menggunakan nilai *confidence* dari metode *Association Rule*. Tahapan untuk mendapatkan nilai *confidence* dari relasi produk dan kata kunci adalah sebagai berikut :

1. Meghitung jumlah tweet yang mengandung produk
2. Meghitung jumlah tweet yang mengandung produk dan kata kunci secara bersamaan
3. Meghitung nilai confidence kata kunci ke produk, dengan cara membagi jumlah tweet yang mengandung produk dan kata kunci secara bersamaan dengan jumlah tweet yang mengadung produk

Semakin tinggi nilai *confidence* tersebut maka produk dan kata kunci sering muncul bersamaan didalam sekumpulan *tweet*, hal tersebut dapat berarti bahwa relasi antara produk dan kata kunci tersebut kuat. Dalam pemilihan relasi yang akan digunakan untuk visualisasi graph di lakukan pemfilteran nilai *confidence*, pemfilteran tersebut bertujuan hanya relasi yang kuat saja yang akan digunakan dalam *visualisasi graph*. Jika nilai *confidence* yang dihasilkan dari

relasi kata kunci dan produk melebihi nilai *threshold* yang ditentukan, maka kata kunci dan produk tersebut akan divisualisasikan dalam bentuk graph.

Dalam pembuatan graph, node untuk produk di beri warna biru, sedangkan untuk kata kunci diberi warna hijau. Besar kecilnya node itu berdasarkan pada nilai bobot w_t dan besar kecilnya garis tergantung pada nilai *confidence* antara produk dan kata kunci.

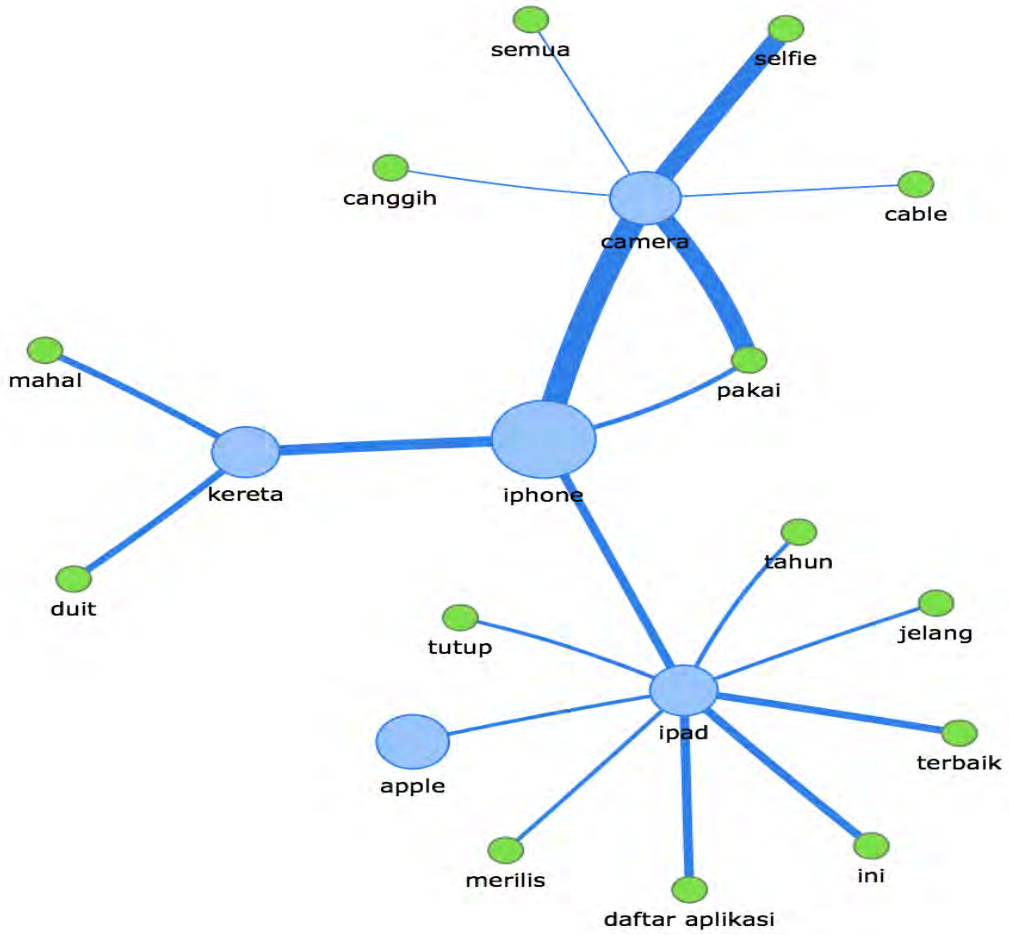
Tabel 4.12 menunjukkan beberapa contoh nilai *confidence* dari produk dan kata kunci *tweet* yang mengandung produk Apple. Berdasarkan data tersebut relasi antara produk dan kata kunci yang mempunyai hubungan paling kuat adalah kamera dan selfie dengan nilai *confidence* 0.81, sedangkan relasi antara produk dengan produk yang mempunyai hubungan paling kuat lainnya adalah kamera dengan iphone dengan nilai *confidence* 0.98. setalah didapatkan nilai *confidence* produk dengan kata kunci atau produk lainnya, selanjutnya akan dibuat *graph* dengan model *network* untuk memvisualisasikan hubungan antara produk dengan kata kunci atau produk lainnya seperti yang dapat dilihat pada gambar 4.2.

Tabel 4.12 Contoh asosiasi roduk dengan kata kunci atau produk lain *tweet* Apple

Produk	**Kata Kunci / Produk**	***confidence***
Camera	Iphone	0.98
camera	selfie	0.81
iphone	pakai	0.3
iphone 6s	emas	0.15
iphone 6s	berbalut	0.15

Contoh *tweet* yang mengandung asosiasi kata camare, iphone dan selfie adalah “pakai iphone tapi selfie pakai back camera. pastu tangan pulak block. selfie pebend a? https://t.co/emy9kwqdea” dan “"weh iphone xde camera depan ye" "ada npe kau ckp cm tu" "ye la aku tgk smua selfie pakai crmin je"””, sedangkan yang mengandung asosiasi kata iphone 6s, berbalut dan emas adalah “iphone 6s & 6s plus berbalut emas dibanderol rp 160 juta https://t.co/enilc5bh8p” dan “jd.id hadirkan lux iphone 6s berbalut emas https://t.co/brh2wyjrjy”.

Tidak semua kata yang berelasi sesuai memiliki makna, contohnya pada gambar 4.2 semua merupakan kata kunci dari camera. Padahal semua dan camera tidak mempunyai hubungan dalam arti kata secara langsung. Semua menjadi kata kunci camera, karena kata semua dan kata camera sering muncul bersamaan dama 1 tweet.



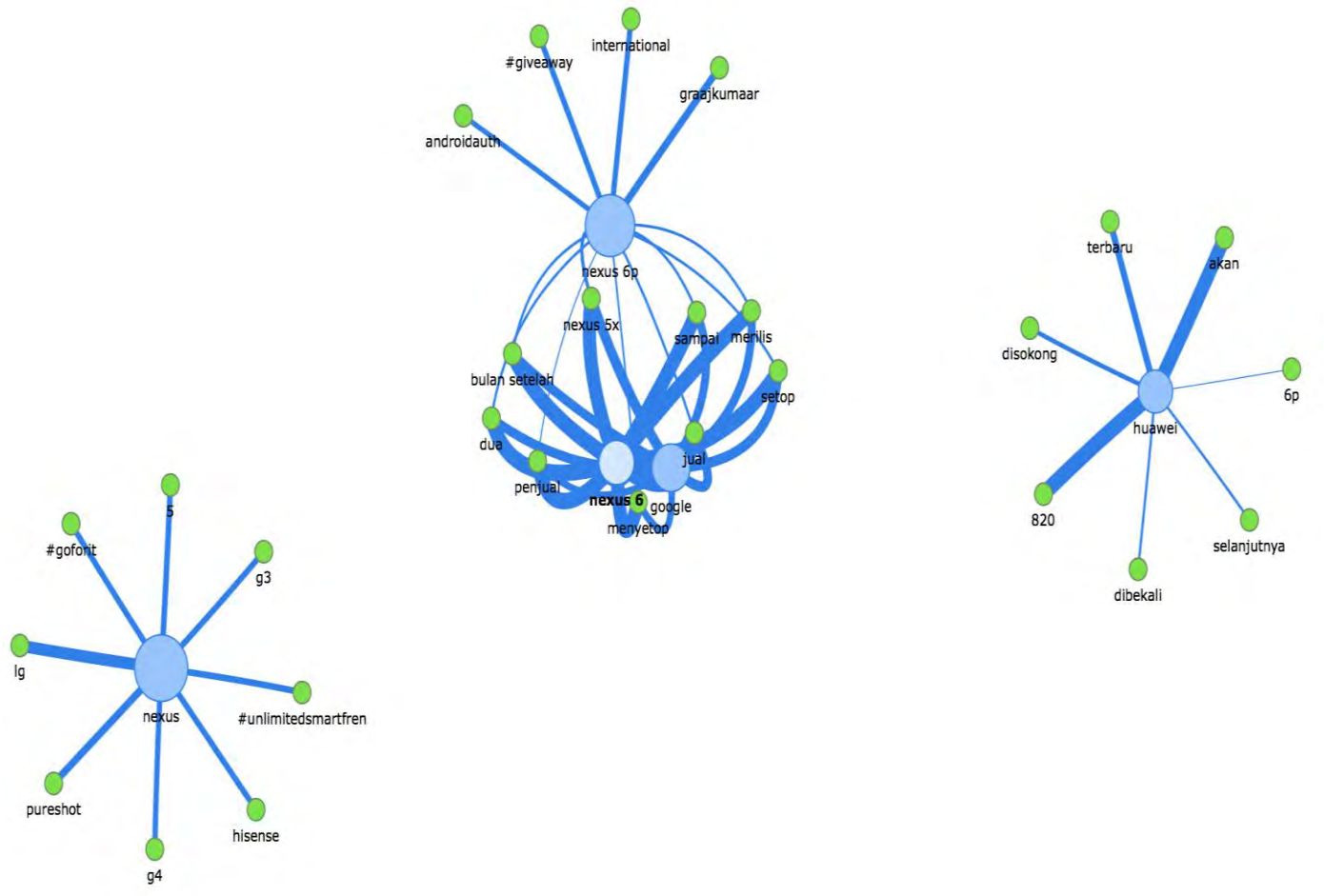
Gambar 4.3 Contoh hasil visualisasi hubungan antara fitur kata-kata produk pada tweet Apple

Tabel 4.13 menunjukkan beberapa contoh nilai *confidence* dari produk dan kata kunci *tweet* yang mengandung produk nexus. Berdasarkan data tersebut relasi antara produk dan kata kunci yang mempunyai hubungan paling kuat adalah nexus 6p dan merilis dengan nilai *confidence* 0.5, sedangkan relasi antara produk dengan produk yang mempunyai hubungan paling kuat lainnya adalah google dengan nexus 5x dengan nilai merilis 0.51. setalah didapatkan nilai *confidence* produk dengan kata kunci atau produk lainnya, selanjutnya akan dibuat graph

dengan model network untuk memvisualissikan hubungan antara produk dengan kata kunci atau produk lainnya seperti yang dapat dilihat pada gambar 4.3.

Tabel 4.13 Contoh asoiasi produk dengan kata kunci *tweet* Nexus

Produk	Kata Kunci / Produk	*Confidence*
google	nexus 5x	0.51
nexus 6p	merilis	0.5
google	merilis	0.46
nexus	pureshot	0.4
nexus	#unlimitedsmartfren	0.37
nexus 6p	#giveaway	0.36
nexus 5	nexus 5x	0.30



Gambar 4.4 Contoh Hasil visualisasi hubungan antara fitur kata-kata produk pada tweet Nexus

Contoh tweet yang mengandung asosiasi kata google dengan merilis, nexus 5x, dan nexus 6p dengan merilis adalah “google setop jual nexus 6 : tak sampai 2 bulan setelah merilis nexus 5x dan nexus 6p, google menyetop penjual.”, sedangkan yang mengandung asosiasi kata nexus 6p, #giveaway adalah “nexus 6p international #giveaway androidauth https://t.co/d879fe2frg graajkumaar”.

4.3 Uji Coba dan Analisa

Pengujian dilakukan agar dapat mengetahui performance system yang dibuat. Groundtruth yang digunakan dalam penlitian ini, didapatkan secara manual dari pengamatan data. Terdapat 4 macam ujicoba dan analisa yang dilakukan dalam penelitian dan data yang akan digunakan dalam pengujian adalah 2 kumpulan data.

Data yang akan digunakan dalam pengujian yaitu :

a. *Tweet* Brand Apple

Data *tweet* yang digunakan adalah *tweet* dengan keyword produk dari Apple sebanyak 72.286 *tweet*, *tweet* dengan total *tweet* yang mengandung nilai *retweet* sebanyak 33651 *tweet* dan yang mengandung nilai *favourite* sebanyak 2955 *tweet*.

b. *Tweet* Brand Nexus

Data *tweet* yang digunakan adalah *tweet* dengan keyword produk dari nexus sebanyak 4.486 *tweet*, *tweet* dengan total *tweet* yang mengandung nilai *retweet* sebanyak 306 *tweet* dan yang mengandung nilai *favourite* sebanyak 105 *tweet*.

Terdapat dua buah data training dalam penelitian ini, data training pertama adalah data training yang di guanakan sebagai corpus POS *tagging*, sebanyak 7534 *tweet*, data tersebut berisi text tweet yang sudah di *tagging* secara manual. Data training kedua adalah data yang digunakan dalam membentuk rule, sebanyak 400 *tweet* yang berisikan produk dari brand. Dari data tersebut di amati secara mana kombinasi tag yang sering muncul sebagai produk, kombinasi tag yang sering muncul akan digunakan sebagi rule untuk deteksi produk.

Skenarion ujicoba dan analisa yang akan digunakan dalam penelitian ini adalah

a. Uji Coba Nilai *Threshold* pada pemfilteran kata kunci.

Penentuan nilai *threshold* dalam pemfilteran kata kunci menentukan seberapa banyak kata yang akan diproses. Semakin besar nilai *threshold* maka akan semakin sedikit kata yang diproses, dan begitu pula sebaliknya semakin kecil nilai *threshold* maka semakin banyak kata yang diproses.

Uji coba ini bertujuan untuk mengetahui nilai *threshold* yang memberikan hasil pemfilteran kata kunci paling baik. Sehingga bisa menurunkan dimensi kata kunci yang akan digunakan dalam proses pembentukan asosiasi kata kunci dan hasil ekstraksi produk, hasil uji coba dapat dlihat dari nilai *f-measure* masing-masing kelompok. Parameter nilai *threshold* yang digunakan adalah kelipatan nilai 0.1, mulai dari 0.1 sampai dengan 1. Untuk mendapatkan nilai precision, recall, dan *f-measure,* hasil yang didapat dari sistem akan dibandingkan dengan hasil secara manual.

b. Uji Coba Nilai *a,b,*dan *c* pada pembobotan produk serta nilai Nilai *Threshold* pada pemfilteran produk

Penentuan nilai *a, b,* dan *c*, pembobotan persamaan 3.4 menentukan seberapa pengaruh nilai kemunculan, nilai *favourite* dan nilai *retweet* pada sejumlah *tweet*. Sedangkan penentuan nilai *threshold* dalam pemfilteran produk seberapa banyak kata yang akan diproses. Semakin besar nilai *threshold* maka akan semakin sedikit kata yang diproses, dan begitu pula sebaliknya semakin kecil nilai *threshold* maka semakin banyak kata yang diproses. Pengujian nilai a (kemunculan), b (*retweet*), c (*favourite*), dan *threshold* dilalukan bersamaan karena jika dilakukan terpisah tidak bisa mengetahui nilai a, b, c yang optimal, kerena dilakukan bersamaan sehingga kombinasi yang dilakukan adalah 500 pengujian untuk masing-masing dataset.

Uji coba ini bertujuan untuk mengetahui nilai a, b, c, dan *threshold* yang memberikan hasil ekstraksi produk paling baik, hasil ujicoba dapat dilihat dari nilai *f-measure* masing-masing kelompok. Parameter nilai *threshold* yang digunakan adalah kelipatan nilai 0.1, mulai dari 0.1 sampai

dengan 1 untuk masing-masing nilai *threshold*. Untuk mendapatkan nilai precision, recall, dan *f-measure,* hasil yang didapat dari sistem akan dibandingkan dengan hasil secara manual.

c. Uji Coba Nilai *Threshold* pada asosiasi.

Penentuan nilai *threshold* dalam asosiasi antara produk dan kata kunci *tweet*s nilai *threshold* menentukan seberapa kuat relasi produk dan kata kunci, semakin besar maka relasinya akan semakin kuat, dan begitu pula sebaliknya.

Uji coba ini bertujuan untuk mengetahui *threshold* yang memberikan hasil asosiasi produk dana kata kunci paling baik, hasil ujicoba dapat dilihat dari nilai *f-measure* masing-masing kelompok. Parameter nilai *threshold* yang digunakan adalah kelipatan nilai 0.05, mulai dari 0.05 sampai dengan 1. Untuk mendapatkan nilai precision, recall, dan *f-measure,* hasil yang didapat dari sistem akan dibandingkan dengan hasil secara manual.

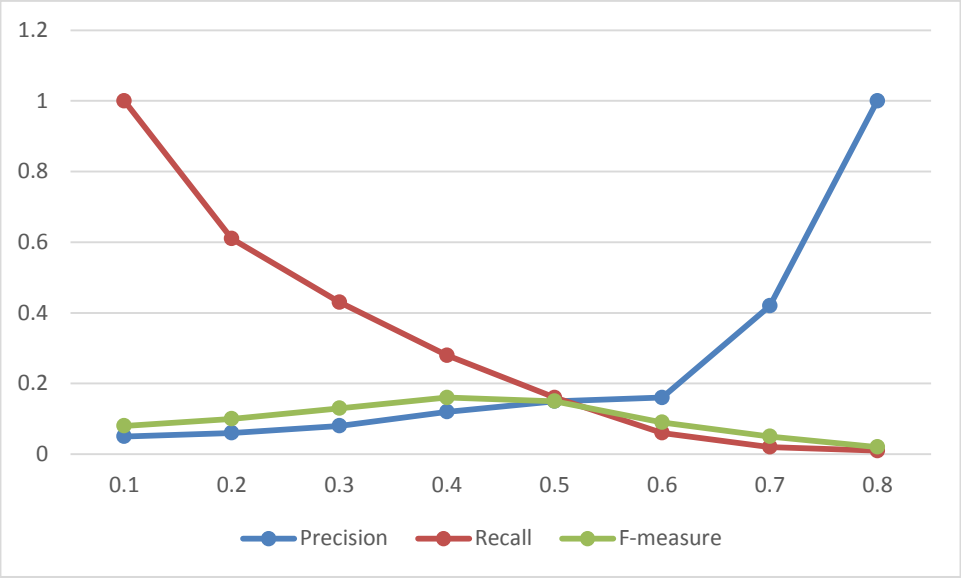
d. Analisa Visualisasi hubungan antara fitur kata-kata produk

Analisa visualisasi hubungan antara fitur kata-kata produk digunakan untuk mengetahui apakah bentuk atau model *graph* yang digunakan yaitu model *network* mampu merepresentasikan data secara baik dan mudah di pahami oleh user.

4.3.1 Pengujian dan Analisas*Tweet* Brand Apple

a. Uji coba dan analisa nilai *threshold* pada pemfilteran kata kunci.

Tujuan dari pengujian ini adalah untuk mengetahui nilai *threshold* pemfilteran kata kunci yang menghasilkan kata kunci yang terbaik. Uji coba menggunakan data *tweets* yang mengandung keyword dari brands Apple sebanyak 72.286 *tweets*. Pemberian beberapa variasi *threshold* terhadap pemfilteran kata kunci bertujuan untuk nilai mendapatkan *threshold* optimal, nilai *threshold* dimulai dari 0.1 sampai 1.0 dengan kenaikan 0.1. Selanjutnya kata kunci yang didaptakan untuk setiap nilai *threshold* akan dihitung validasinya nilai *Precission, Recall,* dan *F-Measure.*



Gambar 4.5 Grafik nilai precison, recall dan *f-measure* pada pemfilteran *threshold* pada pemfilteran kata kunci dengan *tweet* brand Apple

Semakin besar nilai *threshold* maka kata kunci yang dihasilkan semakin sedikit, dari grafik 4.2 nilai *threshold* maksimal adalah 0.8 jika melebihi 0.8 maka tidak akan menghasilkan kata kunci. berdasarkan grafik 4.2 nilai *precission* akan cenderung meningkat jika nilai *threshold* semakin besar. Berkebalikan dengan nilai *recall*, nilai *recall* cendurung menurun jika *threshold* semakin besar. Untuk nilai *f-measure* akan meningkat jika nilai *threshold* tidak melebihi 0.4, jika lebih dari 0.4 maka nilai *threshold* akan menurun jadi nilai optimal untuk *threshold f-measure* adalah 0.4. *Precission, recall dan f-measure* nilainya hampir sama atau selisihnya kecil jika nilai *threshold* yang digunakan adalah 0.5, itu berarti jumlah kata kunci yang didapatkan tingkat ketepatan antara kata kunci benar yang berhasil di dapat dan tingkat keberhasilan sistem dalam menemukan kata kunci benar nilainya sama atau selisihnya kecil, sehingga nilai *threshold* yang digunakan pada pemfilteran kata kunci *tweet* yang mengandung produk dari bran Apple adalah 0.5 dengan nilai *precission* = 0.15 , *recall* = 0.158, *dan f-measure* = 0.154.

b. Uji coba dan analisa nilai *a,b,*dan *c* pada pembobotan produk serta nilai Nilai *Threshold* pada pemfilteran produk

Tujuan dari pengujian ini adalah untuk mengetahui nilai nilai *a* (kemunculan), *b* (*retweet*), *c* (*favourite*) pada pembobotan, dan *threshold* pemfilteran bobot yang dapat menghasilkan hasil ekstraksi produk dari sebuah *tweet* yang terbaik. Uji coba menggunakan data *tweet*s yang mengandung keyword dari brands Apple sebanyak 72.286 *tweet*s. Pemberian beberapa variasi *a ,b,* dan *c* pada pembobotan produk serta nilai *threshold* terhadap pemfilteran bobot bertujuan untuk mendapatkan nilai *a, b, c* dan *threshold* yang optimal. Nilai nilai *a, b, c* dan *threshold* dimulai dari 0.1 sampai 1.0 dengan kenaikan 0.1. Selanjutnya kumpulan produk yang dihasilkan untuk setiap kombinasi *a, b, c* dan *threshold* akan dihitung validasinya nilai menggunakan nilai *f-measure*, jumlah kombiasi nilai *a, b, c* dan *threshold* sebanyak 507.

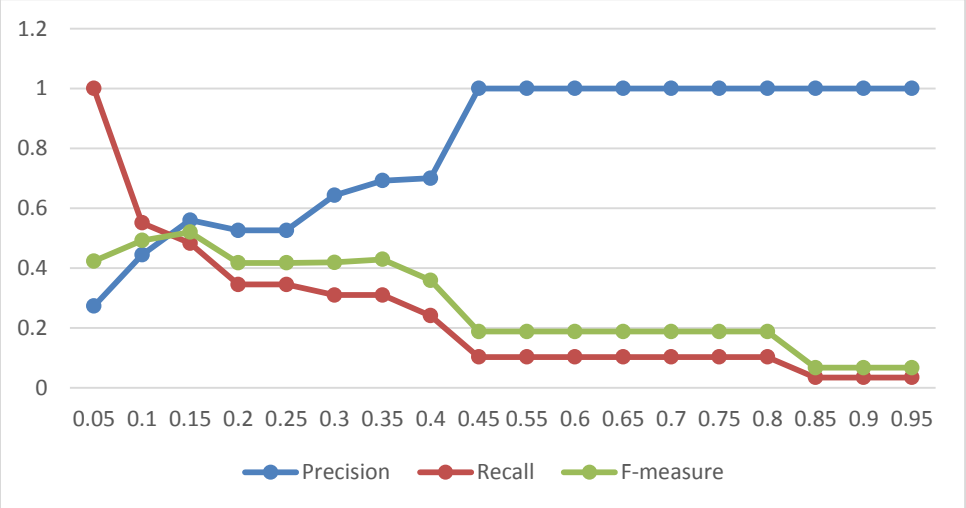
Tabel 4.14 Ujicoba Nilai *a, b,* dan *c* pada pembobotan produk serta nilai Nilai *Threshold* pada pemfilteran produk Apple

Nilai *a (w1)*	**Nilai *b (w2)***	**Nilai *c (w3)***	**Nilai *threshold* bobot**	**Precsion**	***Recall***	***F-measure***
0.8	**0.2**	**0**	**0.2**	0.857	0.171	**0.286**
0.9	0.1	0	0.1	0.429	0.171	0.245
0.9	0	0.1	0.1	0.714	0.143	0.238
0.7	0	0.3	0.2	0.333	0.171	0.226
0.6	0.2	0.2	0.3	0.417	0.142	0.218
0.7	0.2	0.1	0.2	0.219	0.2	0.209
0.5	0.4	0.1	0.4	0.385	0.143	0.208
0.5	0.1	0.4	0.3	0.25	0.17	0.203
0.6	0.3	0.1	0.3	0.25	0.17	0.203
.....	.....	.....	.....	.....	.....	.....
0	0.9	0.1	0.1	0.022	1	0.043
0	0.8	0.2	0.2	0.015	0.371	0.029

Tabel 4.14 menampilkan 10 nilai kombinasi *a, b, c* dan *threshold* yang memilkin nilai *f-measure* tertinggi dan 2 nilai *f-measure* terendah yang nilainya bukan 0, untuk kesuluruhan kombinasi *a, b, c* dan *threshold* dapat dilihat di lampiran. Dimana *f-measure* tertinggi diperoleh saat nilai *a* = 0.8, *b* = 0.2 , *c* = 0 dan nilai *threshold* = 0.2, hal ini menunjukkan bahwa nilai kombinasi tersebut memiliki hasil yang lebih baik dari hasil kombinasi lainnya. Nilai *a* = 0.8, *b* = 0.2, dan *c* = 0 menunjukkan bahwa nilai kemunculan berpengaruh lebih besar pada dari pada nilai *retweet* dan *favourite* pada etraksi produk.

c. Uji coba dan analisa nilai *threshold* pada asosiasi.

Tujuan dari pengujian ini adalah untuk mengetahui nilai *threshold* yang akan digunakan pada asosiasi produk dengan kata kunci atau kata kunci lainnya, penggunaan nialai *threshold* agar dapat menghasilkan kombinasi produk dengan kata kunci yang terbaik. Uji coba menggunakan data *tweets* yang mengandung keyword dari brands Apple sebanyak 72.286 *tweets*. Pemberian beberapa variasi *threshold* terhadap pemfilteran asosiasi produk dengan kata kunci atau kata kunci lainnya bertujuan untuk nilai mendapatkan *threshold* optimal, nilai *threshold* dimulai dari 0.1 sampai 1.0 dengan kenaikan 0.1. Selanjutnya hasi asosiasi yang didaptakan untuk setiap nilai *threshold* akan dihitung validasinya nilai Precission, *Recall*, dan *F-measure*.



Gambar 4.6 Grafik nilai precison, *recall* dan *f-measure* pada asosiasi produk dengan kata kunci pada *tweet* yang mengandung produk Apple

Semakin besar nilai *threshold* maka hasil asosiasi produk dengan kata kunci yang dihasilkan semakin sedikit, pada grafik 4.3 nilai *threshold* maksimal adalah 0.95 jika melebihi 0.8 maka tidak akan menghasilkan hasil asosiasi. berdasarkan grafik 4.2 nilai *precission* akan cenderung meningkat jika nilai *threshold* semakin besar. Berkebalikan dengan nilai *recall*, nilai *recall* cendurung menurun jika *threshold* semakin besar. Untuk nilai *f-measure* akan meningkat jika nilai *threshold* tidak melebihi 0.15, jika lebih dari 0.15 maka nilai *threshold* akan menurun jadi nilai optimal untuk *threshold f-measure* adalah 0.15. *Precission, recall dan f-measure* nilainya hampir sama atau selisihnya kecil jika nilai *threshold* yang digunakan adalah 0.15, itu berarti jumlah tingkat ketepatan antara hasil asosiasi benar yang di dapat dan tingkat keberhasilan sistem dalam menemukan asosiasi benar nilainya sama atau selisihnya kecil, sehingga nilai *threshold* yang digunakan pada pemfilteran asosiasi asosiasi produk dengan kata kunci *tweet* yang mengandung produk dari bran Apple adalah 0.15 dengan nilai *precission* = 0.56 , *recall* = 0.48, *dan f-measure* = 0.52.

d. Analisa visualisasi hubungan antara fitur kata-kata produk

Tujuan dari Analisa visualisasi hubungan antara fitur kata-kata produk adalah untuk mengetahui apakah bentuk atau model graph yang digunakan yaitu model network mampu merepresentasikan data secara baik. Tabel 4.14 menenjukkan nilai *confidence* antara produk dengan kata kunci atau produk lainnya. Nilai *confidence* tertinggi adalah relasi antara camera dengan iphone 0.98, dan nilai *confidence* terendah adalah iphone6s dengan berbaliut yaitu 0.15.

Tabel 4.15 Ujicoba *confidence* asosisai antara produk dengan kata lunci atau produk pada *tweet* brand Apple.

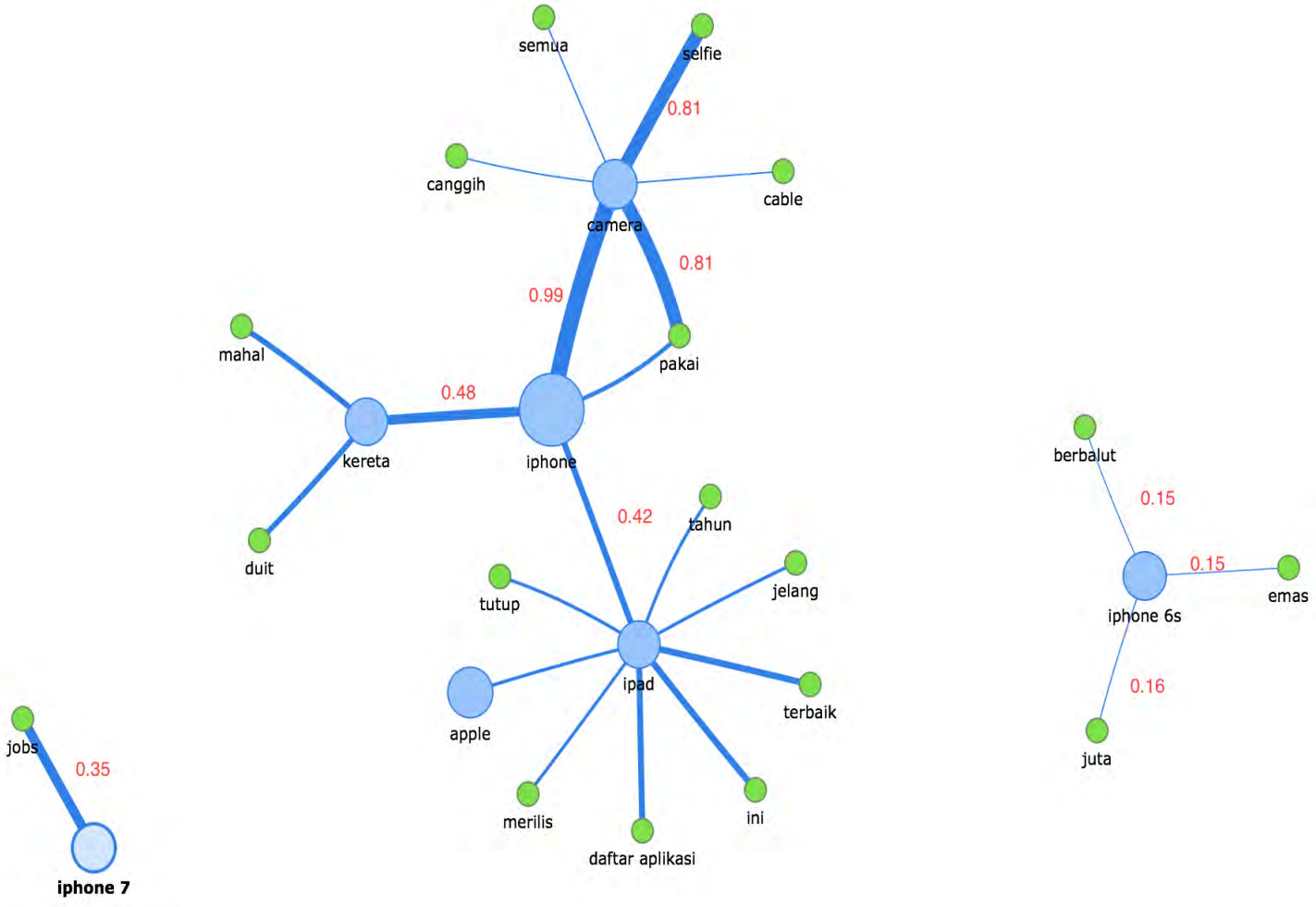
Produk	Kata Kunci / Produk	*confidence*
iphone 6s	berbalut	0.15
iphone 6s	emas	0.15
iphone 6s	juta	0.16

Tabel 4.15 Ujicoba *confidence* asosisai antara produk dengan kata lunci atau produk pada *tweet* brand Apple. (lanjutan)

Produk	Kata Kunci / Produk	*confidence*
camera	canggih	0.17
camera	cable	0.17
camera	semua	0.17
ipad	tutup	0.26
ipad	jelang	0.26
ipad	tahun	0.26
iphone	pakai	0.31
iphone 6	mau	0.33
iphone 7	jobs	0.35
kereta	duit	0.38
kereta	mahal	0.38
ipad	daftar aplikasi	0.42
ipad	iphone	0.42
ipad	terbaik	0.42
iphone 6	kuisnya	0.43
iphone 6	#mydreamdate	0.43
iphone 6	ikutan	0.43
iphone 6	disini	0.43
ipad	ini	0.43
kereta	iphone	0.58
camera	selfie	0.81
camera	pakai	0.81
camera	iphone	0.99

Gambar 4.4 menunjukkan 3 kelompok network, kelompok pertama memilki pusat graph iphone, kelompok kedua memilki pusat graph iphone 6s, dan kelompok ketiga memilki pusat graph iphone 7. Pada kelompok pertama iphone berelasi kuat dengan camera ditunjukkan dengan egde yang tebal (nilai *confidence* 0.98), dan camera juga berelasi kuat dengan selfie dengan

egde yang tebal (nilai *confidence* 0.81), dari kumpulan *tweet* yang membicarakan camera sebanyak 3557 *tweet* yang juga membicarakan iphone 3512 *tweet*, dan yang membicarakan juga selfie 2889 *tweet*. Sehingga dapat ditarik kesimpulan *tweet* yang membicarakan camera sebegian besar pasti juga membicarakan iphone dan selfie. Pada kelompok ke dua dengan iphone 6s berelasi dengan emas dan berbalut. Nilai edge untuk berbalut dan emas kurang lebih sama yaitu 0.15. dari kumpulan *tweet* membicarakan iphone 6s sebanyak 4078 *tweet* yang juga membicarakan berbalut dan emas 619 *tweet*. Sehingga dapat ditarik kesimpulan *tweet* yang membicarakan iphone 6s sebegian kecil pasti juga membicarakan berbalut dan emas iphone



Gambar 4.7 Hasil visualisasi hubungan antara fitur kata-kata produk pada tweet Apple

Tabel 4.16 Contoh tweet yang membicarakan iphone camera dan selfie secara bersamaan

No.	Tweet
1	pakai **iphone** tapi **selfie** pakai back **camera**. pastu tangan pulak block. selfie pebenda? https://t.co/emy9kwqdea
2	"weh **iphone** xde **camera** depan ye" "ada npe kau ckp cm tu" "ye la aku tgk smua **selfie** pakai crmin je""
3	rt @saramahayuddin: pakai **iphone** tapi **selfie** pakai back **camera**. pastu tangan pulak block. selfie pebenda? https://t.co/emy9kwqdea
4	rt @monstermasa17: "weh **iphone** xde **camera** depan ye" "ada npe kau ckp cm tu" "ye la aku tgk smua **selfie** pakai crmin je""
5	rt @jangan_bash: pakai **iphone** tapi **selfie** pakai back **camera**. pastu tangan pulak block camera. selfie pebenda? https://t.co/hszltlizdg

Tabel 4.17 Contoh *tweet* yang membicarakan iphone 6s, berbalut, dan emas secara bersamaan

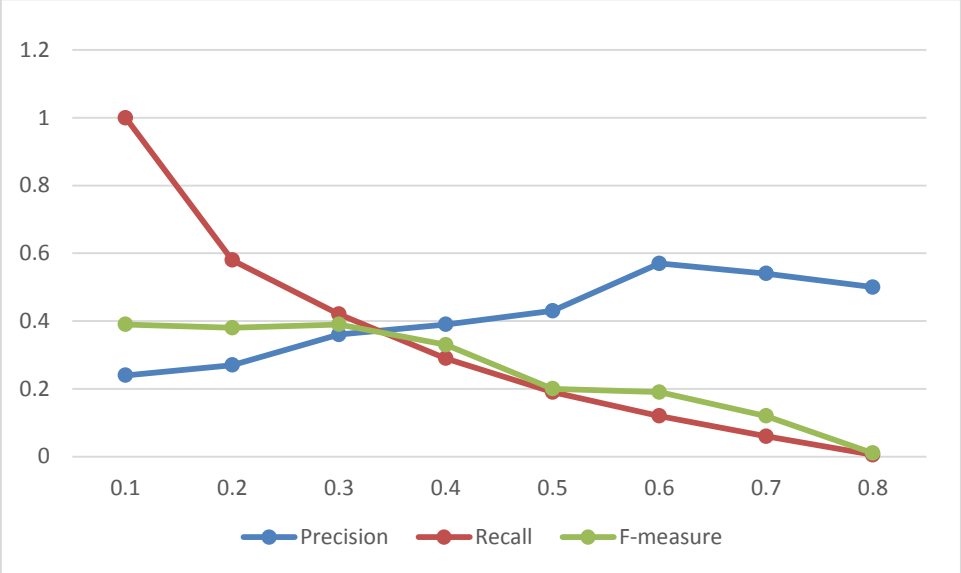
No.	Tweet
1	**iphone 6s** & 6s plus **berbalut emas** dibanderol rp 160 juta https://t.co/enilc5bh8p
2	**iphone 6s** & 6s plus **berbalut emas** dibanderol rp 160 juta: tanpa embel-embel edisi khusus pun, harga iphone sud... https://t.co/ledddcxwav
3	jd.id hadirkan lux **iphone 6s berbalut emas** https://t.co/brh2wyjrjy
4	rt @detikcom: **iphone 6s** & 6s plus **berbalut emas** dibanderol rp 160 juta https://t.co/q88gom1kqy via @detikinet

4.3.2 Pengujian dan Analisas Tweet Brand Nexus

a. Uji coba dan analisa nilai *threshold* pada pemfilteran kata kunci.

Tujuan dari pengujian ini adalah untuk mengetahui nilai *threshold* pemfilteran kata kunci yang menghasilkan kata kunci yang terbaik. Uji coba menggunakan data *tweets* yang mengandung keyword dari brands nexus sebanyak 4.486 *tweets*. Pemberian beberapa variasi *threshold* terhadap pemfilteran kata kunci bertujuan untuk nilai mendapatkan *threshold* optimal, nilai *threshold* dimulai dari 0.1 sampai 1.0 dengan kenaikan 0.1. Selanjutnya

kata kunci yang didaptakan untuk setiap nilai *threshold* akan dihitung validasinya nilai *Precission, Recall,* dan *F-Measure.*



Gambar 4.8 Grafik nilai *precision, recall* dan *f-measure* pada pemfilteran *threshold* pada pemfilteran kata kunci dengan *tweet* brand nexus

Semakin besar nilai *threshold* maka kata kunci yang dihasilkan semakin sedikit, dari grafik 4.4 nilai *threshold* maksimal adalah 0.8 jika melebihi 0.8 maka tidak akan menghasilkan kata kunci. berdasarkan grafik 4.4 nilai *precission* akan cenderung meningkat jika nilai *threshold* semakin besar. Berkebalikan dengan nilai *recall*, nilai *recall* cendurung menurun jika *threshold* semakin besar. Untuk nilai *f-measure* akan meningkat jika nilai *threshold* tidak melebihi 0.3, jika lebih dari 0.3 maka nilai *threshold* akan menurun jadi nilai optimal untuk *threshold f-measure* adalah 0.3. *Precission, recall dan f-measure* nilainya hampir sama atau selisihnya kecil jika nilai *threshold* yang digunakan adalah 0.3, itu berarti jumlah kata kunci yang didapatkan tingkat ketepatan antara kata kunci benar yang berhasil di dapat dan tingkat keberhasilan sistem dalam menemukan kata kunci benar nilainya sama atau selisihnya kecil, sehingga nilai *threshold* yang digunakan pada pemfilteran kata kunci *tweet* yang mengandung produk dari bran nexus adalah 0.3 dengan nilai *precission* = 0.36 , *recall* = 0.42, *dan f-measure* = 0.39.

b. Uji coba dan analisa nilai *a,b,*dan *c* pada pembobotan produk serta nilai Nilai *Threshold* pada pemfilteran produk

Tujuan dari pengujian ini adalah untuk mengetahui nilai nilai a (kemunculan), b (*retweet*), c (*favourite*) pada pembobotan, dan *threshold* pemfilteran bobot yang dapat menghasilkan hasil ekstraksi produk atau produk dari sebuah *tweet* yang terbaik. Uji coba menggunakan data *tweet*s yang mengandung keyword dari brands nexus sebanyak 4.486 *tweet*s. Pemberian beberapa variasi *a, b,* dan *c* pada pembobotan produk serta nilai *threshold* terhadap pemfilteran bobot bertujuan untuk mendapatkan nilai *a, b, c* dan *threshold* yang optimal. Nilai nilai *a, b , c* dan *threshold* dimulai dari -0.1 sampai 1.0 dengan kenaikan 0.1. Selanjutnya produk yang dihasilkan untuk setiap kombinasi *a, b, c* dan *threshold* akan dihitung validasinya nilai menggunakan nilai *F-measure*, jumlah kombiasi nilai *a, b, c* dan *threshold* sebanyak 500.

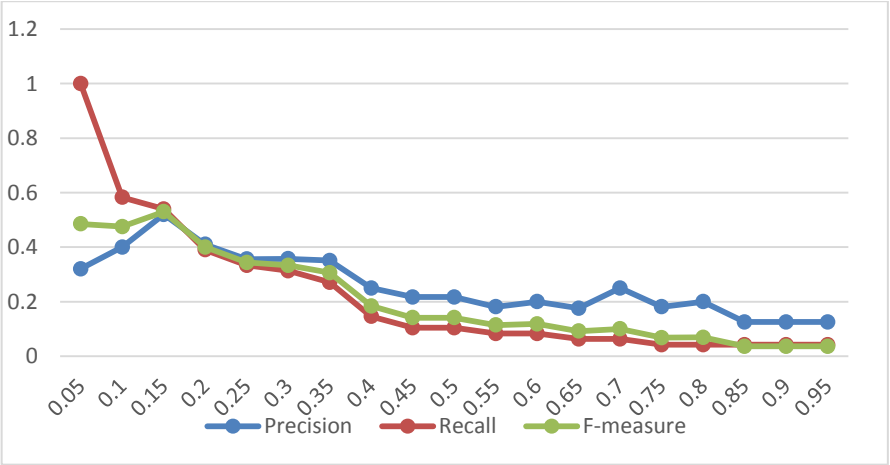
Tabel 4.18 Ujicoba Nilai *a,b,*dan *c* pada pembobotan produk serta nilai *Threshold* pada pemfilteran produk Nexus.

Nilai *a (w1)*	**Nilai *b (w2)***	**Nilai *c (w3)***	**Nilai *threshold* bobot**	**Precsion**	***Recall***	***F-measure***
0.6	**0.2**	**0.2**	**0.2**	0.769	0.625	**0.689**
0.8	0.1	0.1	0.1	0.625	0.625	0.625
0.8	0	0.2	0.1	0.727	0.5	0.59
0.6	0	0.4	0.2	0.875	0.438	0.58
0.9	0.1	0	0.1	0.778	0.438	0.56
0.9	0	0.1	0.1	0.778	0.438	0.56
0.6	0.1	0.3	0.2	0.778	0.438	0.56
0.7	0.1	0.2	0.2	1	0.375	0.55
0.8	0.2	0	0.1	0.45	0.625	0.526
.....	.....	.....	.....	.....	.....	.....
0.4	0.5	0.1	0.5	0.333	0.063	0.105
0.4	0.5	0.1	0.5	0.333	0.063	0.105

Tabel 4.18 menampilkan 10 nilai kombinasi *a, b, c* dan *threshold* yang memilkin nilai *f-measure* tertinggi dan 2 nilai *f-measure* terendah yang nilainya bukan 0, untuk kesuluruhan kombinasi *a, b, c* dan *threshold* dapat dilihat di lampiran. Dimana *f-measure* tertinggi diperoleh saat nilai *a* = 0.6, *b* = 0.2 , *c* = 0.2 dan nilai *threshold* = 0.2, hal ini menunjukkan bahwa nilai kombinasi tersebut memiliki hasil yang lebih baik dari hasil kombinasi lainnya. Nilai *a* = 0.6, *b* = 0.2, dan *c* = 0.2 menunjukkan bahwa nilai kemunculan berpengaruh lebih besar pada dari pada nilai *retweet* dan *favourite* pada ekstraksi produk.

c. Uji coba dan analisa nilai *threshold* pada asosiasi.

Tujuan dari pengujian ini adalah untuk mengetahui nilai *threshold* yang akan digunakan pada asosiasi produk dengan kata kunci atau kata kunci lainnya, penggunaan nialai *threshold* agar dapat menghasilkan kombinasi produk dengan kata kunci yang terbaik. Uji coba menggunakan data *tweet*s yang mengandung keyword dari brands nexus sebanyak 4.486 *tweets*. Pemberian beberapa variasi *threshold* terhadap pemfilteran asosiasi produk dengan kata kunci atau kata kunci lainnya bertujuan untuk nilai mendapatkan *threshold* optimal, nilai *threshold* dimulai dari 0.1 sampai 1.0 dengan kenaikan 0.1. Selanjutnya hasi asosiasi yang didaptakan untuk setiap nilai *threshold* akan dihitung validasinya nilai Precission, *Recall*, dan *F-measure*.



Gambar 4.9 Grafik nilai *precision, recall* dan *f-measure* pada asosiasi produk dengan kata kunci pada *tweet* yang mengandung produk Nexus

Semakin besar nilai *threshold* maka hasil asosiasi produk dengan kata kunci yang dihasilkan semakin sedikit, pada grafik 4.5 nilai *threshold* maksimal adalah 0.95 jika melebihi 0.8 maka tidak akan menghasilkan hasil asosiasi. berdasarkan grafik 4.5 nilai *precission* akan cenderung meningkat jika nilai *threshold* semakin besar. Berkebalikan dengan nilai *recall*, nilai *recall* cendurung menurun jika *threshold* semakin besar. Untuk nilai *f-measure* akan meningkat jika nilai *threshold* tidak melebihi 0.15, jika lebih dari 0.15 maka nilai *threshold* akan menurun jadi nilai optimal untuk *threshold f-measure* adalah 0.15. *Precission, recall dan f-measure* nilainya hampir sama atau selisihnya kecil jika nilai *threshold* yang digunakan adalah 0.15, itu berarti jumlah tingkat ketepatan antara hasil asosiasi benar yang di dapat dan tingkat keberhasilan sistem dalam menemukan asosiasi benar nilainya sama atau selisihnya kecil, sehingga nilai *threshold* yang digunakan pada pemfilteran asosiasi asosiasi produk dengan kata kunci *tweet* yang mengandung produk dari brand nexus adalah 0.15 dengan nilai *precission* = 0.51, *recall* = 0.54, dan *f-measure* = 0.53.

d. Analisa visualisasi hubungan antara fitur kata-kata produk

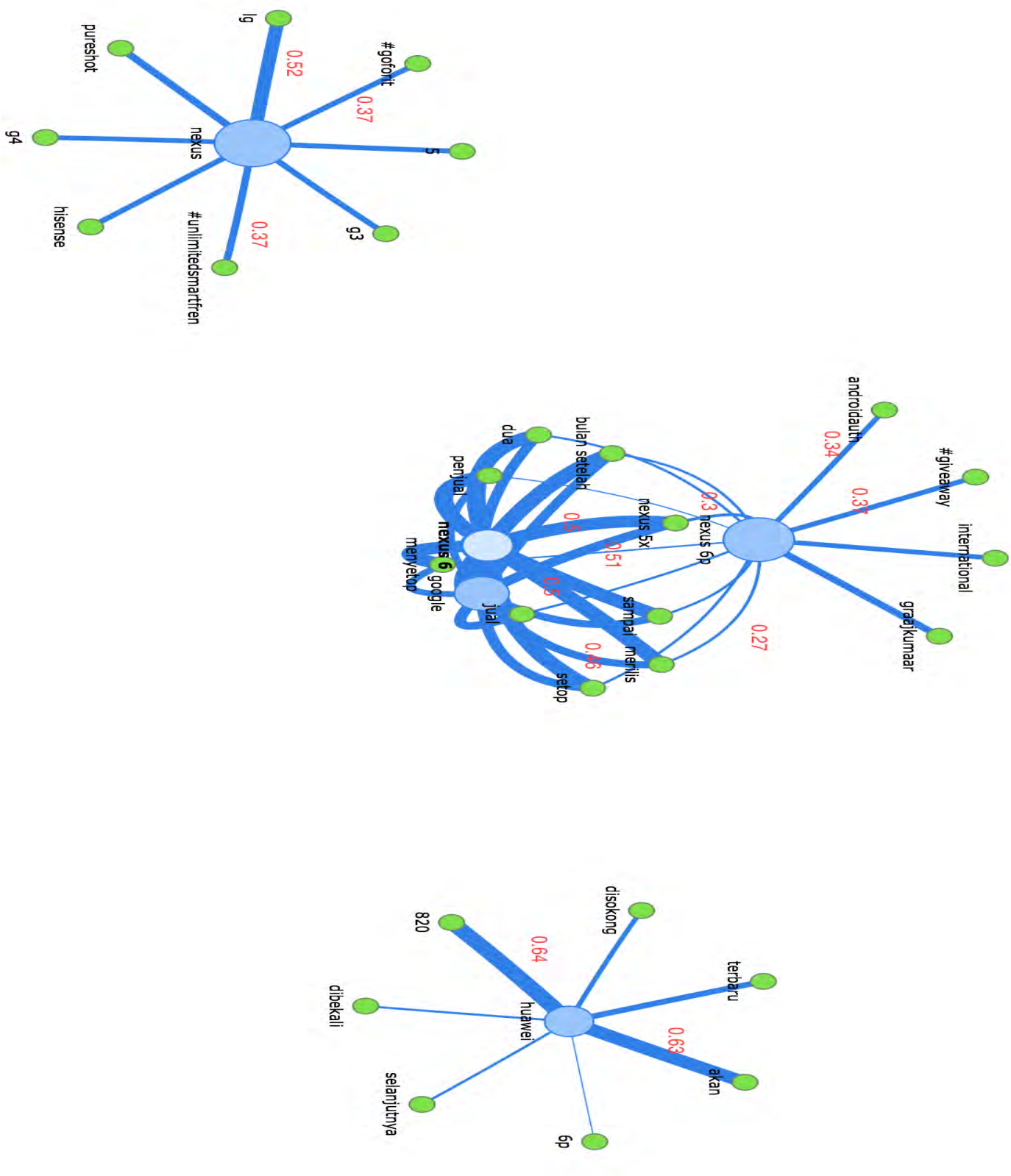
Tujuan dari Analisa visualisasi hubungan antara fitur kata-kata produk adalah untuk mengetahui apakah bentuk atau model graph yang digunakan yaitu model *network* mampu merepresentasikan data secara baik. Tabel 4.19 menunjukkan nilai *confidence* antara produk dengan kata kunci atau produk lainnya. Nilai *confidence* tertinggi adalah relasi antara Huawei dengan 820 yaitu 0.2, dan nilai *confidence* terendah adalah Huawei dengan 6p yaitu 0.151.

Tabel 4.5 Nilai *confidence* asosisai antara produk dengan, Kata Kunci / produk pada *tweet* brand Nexus.

Produk	Kata Kunci / Produk	*confidence*
nexus 6p	merilis	0.27
nexus 6p	dua	0.27
nexus 6p	nexus 5x	0.30

Tabel 4.19 Nilai *confidence* asosisai antara produk dengan, Kata Kunci / produk pada *tweet* brand Nexus (lanjutan)

Produk	Kata Kunci / Produk	*confidence*
nexus 6p	androidauth	0.34
nexus 6p	#giveaway	0.37
nexus 6p	international	0.37
google	penjual	0.39
huawei	terbaru	0.40
nexus	pureshot	0.41
google	menyetop	0.41
nexus 6p	graajkumaar	0.41
nexus 6	penjual	0.43
nexus 6	menyetop	0.44
google	bulan setelah	0.46
google	sampai	0.46
google	merilis	0.46
google	dua	0.46
google	setop	0.48
google	setop	0.48
google	jual	0.48
nexus 6	nexus 5x	0.50
nexus 6	sampai	0.50
nexus 6	merilis	0.50
nexus 6	dua	0.50
google	nexus 5x	0.51
nexus	lg	0.52
nexus 6	setop	0.53
nexus 6	jual	0.54
huawei	akan	0.63
huawei	820	0.64
nexus	pureshot	0.41



Gambar 4.10 Hasil visualisasi hubungan antara fitur kata-kata produk *tweet* Nexus

Gambar 4.5 menunjukkan 3 kelompok network, kelompok pertama memilki pusat graph nexus 6p, kelompok kedua memilki pusat graph huawei, dan kelompok ketiga memilki pusat graph nexus. Pada kelompok pertama google berelasi dengan merilis ditunjukkan dengan egde yang ketebalanya sedang (nilai *confidence* 0.46), dan merilis juga berelasi dengan nexus 6 (nilai *confidence* 0.5), dan nexus 6p (nilai *confidence* 0.27). Sehingga dapat ditarik kesimpulan bahwa *tweet* yang membahas google merilis nexus 6 lebih banyak dari pada *tweet* yang mebahas google merilis nexus 6p. Pada kelompok ke dua dengan huawei berelasi dengan akan dan 820, dengan ketebalan dan nilai *edge* yang hampir sama yaitu 0.63 dan 0.64. Dari kumpulan *tweet* yang membicarakan huawei sebanyak 321 *tweet* yang juga membicarakan 820 dan akan 206 *tweet*. Sehingga dapat ditarik kesimpulan *tweet* yang membicarakan huawei sebagian besar pasti juga membicarakan akan dan 820.

Hasil dari pengujian brand Apple dan Nexus menunjukkan bahwa nilai kemunculan lebih berpengaruh pada eksttraksi produk, dari pada nilai retweet dan nilai favourite. Nilai *Precision* dan *Recall* untuk pemfilteran dan hasil ekstraksi Produk kata kunci berbanding terbalik, karena terlalu banyak kata yang didapatkan sedangkan kata yang benar-banar kata-kunci atau atau produk tidak terlalu banyak. Pada brand *Apple*, untuk kata kunci yang didapat adalah 7123 kata, sedangkan yang benar adalah 323 kata, untuk produk 1601 yang relevan hanya 35. Sedengkan pada brand *Nexus*, untuk kata kunci yang didapat adalah 790 kata, sedangkan yang benar adalah 191 kata, untuk produk 98 yang benar hanya 14. Hasil *ekstraksi* produk yang didapat tidak terlalu bagus, dikarenakan ekstraksi *POS Tagging* yang dihasilkan menggunakan metode *Hidden Makov Model* pada *twiiter* kurang begitu bagus, karena terdapat beberapa yang bukan kata benda di asumsuikan kata benda. Itu dikarenakan metode *Hidden Makov Model* selain menghitung probabilitas kata ke sebuah *tag*, juga menghitung probabilitas *tag* ke *tag* lainnya, sehingga jika sebuah kata tidak memilki data set *tag*, maka bisa dihasilkan *tag* yang salah.

4.4 Kendala Uji Coba

Dalam melakukan uji coba dan validasi, terdapat beberapa kendala yang mempengaruhi hasil pengujian di atas. Beberapa kendala tersebut dapat dijabarkan sebagai berikut :

1. Pada *tweet* yang didapatkan terdapat *tweet* berbahasa malaysia meskipun dilakukan pemfilteran *tweet* bahasa Indonesia menggunakan twitter api.
2. Pada *tweet* yang didapatkan terdapat *tweet* yang tidak sesuai dengan *keyword* hal itu disebabkan karena *tweet* tersebut mengandung salah satu *keyword* dari *brand* yang kita cari sebagai has*tag*, tapi isinya bukan mengenai *brand* yang dicari. Sehingga harus dilakukan pemfilteran lagi.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut :

1. Penggunaan Metode *Grammatical Tagging* digabungkan dengan *rule*, dapat mengestraksi produk yang tertulis dalam sebuah *tweet*.
2. Dari hasil ujicoba pembobotan pada ekstraksi produk yang tertulis dalam sebuah *tweet*, nilai kemunculan lebih berpengaruh besar dari pada nilai *retweet* dan nilai *favourite* dengan nilai *precision* 81.3 % .
3. Dari hasi ujicoba asosiasi produk dengan kata kunci atau produk lain yang tertulis dalam sebuah *tweet*, didapatkan nilai *f-measure* 52.5 %, mengindikasikan bahwa dapat mengenali hubungan antara produk dengan kata kunci.

5.2 Saran

Berdasarkan hasil penelitian ini, saran yang dapat diberikan agar diperoleh hasil yang lebih baik adalah :

1. Diperlukan perbaikan di metode ekstraksi *Tagging*, sehingga dapat mengekstraksi produk dengan lebih baik.
2. Diperlukan suatu metode untuk menyeleksi *tweet* yang tidak sesuai dengan kata kunci yang dimasukkan.

LAMPIRAN

LAMPIRAN 1. – NILAI THRESHOLD PADA PEMFILTERAN KATA KUNCI TWEET APPLE

Threshold filter	Jumlah Term	Jumlah Term yang berhubungan dengan produk	Precssion	Recall	F-measure
0.1	7123	323	0.05	1	0.08
0.2	3502	197	0.06	0.61	0.10
0.3	1689	140	0.08	0.43	0.13
0.4	757	91	0.12	0.28	0.16
0.5	340	51	0.15	0.16	0.15
0.6	119	19	0.16	0.06	0.09
0.7	19	8	0.42	0.02	0.05
0.8	3	3	1	0.01	0.02
0.9	1	0	0	0	0
1.0	0	0	0	0	0

LAMPIRAN 2. – NILAI THRESHOLD PADA PEMFILTERAN KATA KUNCI TWEET NEXUS

Threshold filter	Jumlah Term	Jumlah Term yang berhubungan dengan produk	Precssion	Recall	F-measure
0.1	790	191	0.24	1	0.389
0.2	400	111	0.28	0.58	0.376
0.3	224	81	0.36	0.43	0.390
0.4	141	55	0.39	0.28	0.331
0.5	84	36	0.42	0.19	0.261
0.6	40	23	0.58	0.12	0.199
0.7	24	13	0.54	0.06	0.120
0.8	2	1	0.5	0.005	0.001
0.9	1	0	0	0	0
1.0	0	0	0	0	0

LAMPIRAN 3. – HASIL NILAI UJICOBA A, B, C DAN THRESHOLD PADA PEMFILTERAN EKSTRAKSI PRODUK PADA TWEET APPLE

Bobot *a (w1)*	**Bobot *b (w2)***	**Bobot *c (w3)***	**Monimal bobot produk**	**Produk yang didapat**	**Produk yang relevan**	**Precsion**	**Recall**	**F-Measure**
0.8	0.2	0	0.2	7	6	0.857	0.171	0.286
0.9	0.1	0	0.1	14	6	0.429	0.171	0.245
0.9	0	0.1	0.1	7	5	0.714	0.143	0.238
0.7	0	0.3	0.2	18	6	0.333	0.171	0.226
0.6	0.2	0.2	0.3	12	5	0.417	0.143	0.213
0.7	0.2	0.1	0.2	32	7	0.219	0.200	0.209
0.5	0.4	0.1	0.4	13	5	0.385	0.143	0.208
0.5	0.1	0.4	0.3	24	6	0.250	0.171	0.203
0.6	0.3	0.1	0.3	24	6	0.250	0.171	0.203
0.7	0.1	0.2	0.2	25	6	0.240	0.171	0.200
0.4	0.3	0.3	0.4	25	6	0.240	0.171	0.200
0.6	0.1	0.3	0.3	6	4	0.667	0.114	0.195
0.5	0.5	0	0.4	18	5	0.278	0.143	0.189
0.6	0	0.4	0.2	50	8	0.160	0.229	0.188
0.8	0	0.2	0.1	53	8	0.151	0.229	0.182
0.6	0.1	0.3	0.2	53	8	0.151	0.229	0.182
0.5	0.2	0.3	0.3	31	6	0.194	0.171	0.182
0.6	0.3	0.1	0.2	99	12	0.121	0.343	0.179
0.5	0	0.5	0.3	22	5	0.227	0.143	0.175
----	----	----	----	----	----	----	----	----
0	0.6	0.4	0.1	1414	35	0.025	1.000	0.048
0.1	0.7	0.2	0.1	1421	35	0.025	1.000	0.048
0.2	0.8	0	0.1	1448	35	0.024	1.000	0.047
0	0.7	0.3	0.1	1475	35	0.024	1.000	0.046
0.1	0.8	0.1	0.1	1488	35	0.024	1.000	0.046
0	0.8	0.2	0.1	1528	35	0.023	1.000	0.045
0.1	0.9	0	0.1	1572	35	0.022	1.000	0.044
0	1	0	0.1	1597	35	0.022	1.000	0.043
0	0.9	0.1	0.1	1601	35	0.022	1.000	0.043

LAMPIRAN 4. – HASIL NILAI UJICOBA A, B, C DAN THRESHOLD PADA PEMFILTERAN EKSTRAKSI PRODUK PADA TWEET NEXUS

Bobot *a (w1)*	**Bobot *b (w2)***	**Bobot *c (w3)***	**Monimal bobot produk**	**Produk yang didapat**	**Produk yang relevan**	**Precsion**	**Recall**	**F-Measure**
0.6	0.2	0.2	0.2	13	10	0.769	0.625	0.690
0.8	0.1	0.1	0.1	16	10	0.625	0.625	0.625
0.8	0	0.2	0.1	11	8	0.727	0.500	0.593
0.6	0	0.4	0.2	8	7	0.875	0.438	0.583
0.9	0.1	0	0.1	9	7	0.778	0.438	0.560
0.9	0	0.1	0.1	9	7	0.778	0.438	0.560
0.6	0.1	0.3	0.2	9	7	0.778	0.438	0.560
0.6	0.3	0.1	0.2	13	8	0.615	0.500	0.552
0.7	0.1	0.2	0.2	6	6	1.000	0.375	0.545
0.8	0.2	0	0.1	22	10	0.455	0.625	0.526
0.7	0.2	0.1	0.2	7	6	0.857	0.375	0.522
0.7	0	0.3	0.2	7	6	0.857	0.375	0.522
0.7	0.3	0	0.2	11	7	0.636	0.438	0.519
0.6	0.4	0	0.3	8	6	0.750	0.375	0.500
0.5	0	0.5	0.2	16	8	0.500	0.500	0.500
0.5	0.1	0.4	0.2	20	9	0.450	0.563	0.500
0.7	0	0.3	0.1	24	10	0.417	0.625	0.500
0.5	0.4	0.1	0.3	9	6	0.667	0.375	0.480
0.5	0.3	0.2	0.3	9	6	0.667	0.375	0.480
----	----	----	----	----	----	----	----	----
0	0.5	0.5	0.8	1	1	1.000	0.063	0.118
0	0.4	0.6	0.8	1	1	1.000	0.063	0.118
0	0.3	0.7	0.8	1	1	1.000	0.063	0.118
0	0.2	0.8	0.8	1	1	1.000	0.063	0.118
0	0.1	0.9	0.8	1	1	1.000	0.063	0.118
0	0	1	0.8	1	1	1.000	0.063	0.118
0.2	0.8	0	0.7	2	1	0.500	0.063	0.111
0.1	0.9	0	0.8	2	1	0.500	0.063	0.111
0.4	0.5	0.1	0.5	3	1	0.333	0.063	0.105

LAMPIRAN 5. – HASIL NILAI UJICOBA NILAI *THRESHOLD* PADA ASOSIASI TWEET APPLE

Nilai threshold confidence	Jumlah asosiasi yang didapat	Jumlah asosiasi yang benar	Precsion	Recall	F-measure
0.05	106	29	0.273	1	0.423
0.1	36	16	0.444	0.551	0.492
0.15	25	14	0.56	0.483	0.52
0.2	19	10	0.526	0.345	0.417
0.25	19	10	0.526	0.345	0.417
0.3	14	9	0.643	0.310	0.419
0.35	13	9	0.692	0.310	0.429
0.4	10	7	0.7	0.241	0.359
0.45	3	3	1	0.103	0.188
0.55	3	3	1	0.103	0.188
0.6	3	3	1	0.103	0.188
0.65	3	3	1	0.103	0.188
0.7	3	3	1	0.103	0.188
0.75	3	3	1	0.103	0.188
0.8	3	3	1	0.103	0.188
0.85	1	1	1	0.0345	0.067
0.9	1	1	1	0.0345	0.067
0.95	1	1	1	0.0345	0.067
1	0	0	0	0	0

LAMPIRAN 6. – HASIL NILAI UJICOBA NILAI *THRESHOLD* PADA ASOSIASI TWEET NEXUS

Nilai threshold confidence	Jumlah asosiasi yang didapat	Jumlah asosiasi yang benar	Precsion	Recall	F-measure
0.05	150	48	0.32	1	0.485
0.1	70	28	0.4	0.583	0.475
0.15	52	27	0.519	0.563	0.54
0.2	48	19	0.396	0.396	0.396
0.25	45	16	0.356	0.333	0.344
0.3	42	15	0.357	0.313	0.333
0.35	37	13	0.351	0.271	0.306
0.4	28	7	0.25	0.146	0.184
0.45	23	5	0.217	0.104	0.141
0.5	23	5	0.217	0.104	0.141
0.55	22	4	0.181	0.083	0.114
0.6	20	4	0.2	0.083	0.118
0.65	17	3	0.176	0.063	0.092
0.7	12	3	0.25	0.063	0.1
0.75	11	2	0.181	0.042	0.068
0.8	10	2	0.2	0.042	0.069
0.85	8	1	0.125	0.042	0.036
0.9	8	1	0.125	0.042	0.036
0.95	8	1	0.125	0.042	0.036

DAFTAR PUSTAKA

Abascal-Mena, R., Lema, R., & Sedes, F. (2014). From tweet to graph: Social network analysis for semantic information extraction. *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*, (pp. 1 - 10).

Alfan Farizki Wicaksono, A. P. (2010). HMM Based Part-of-Speech Tagger for Bahasa Indonesia . *Proceeding of the Fourth Internationul MALINDO Workshop (MALINDO2010)* .

Amma, K., Wada, S., Nakayama, K., Akamatsu, Y., Yaguchi, Y., & Naruse, K. (2014). Visualization of spread of topic words on Twitter using stream graphs and relational graphs. *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*, (pp. 3-6).

Arifin, A., Mahendra, I., & Ciptaningtyas, H. (2009). Enhanced Confix Stripping Stemmer and Ants Algorithm for Classifying News Document in Indonesian Language. *Proceeding of International Conference on Information & Communication Technology and Systems (ICTS).*

Cordeiro, M. (2012). Twitter event detection: Combining wavelet analysis and topic inference summarization. *Doctoral Symposium on Informatics Engineering, DSIE*.

Farzindar Atefeh, W. K. (2013). A Survey of Techniques for Event Detection in Twitter. Computational Intelligence.

Hasby, M., & Khodra, M. (2013). Optimal path finding based on traffic information extraction from Twitter. *ICT for Smart Society (ICISS), 2013 International Conference on* , (pp. 1 - 5).

Jones, B. (1994). Can punctuation help parsing . *In 15 th International Conference on Computational Linguistics* . Kyoto, Japan .

Jurafsky, D. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.*

Medvet, E., & Bartoli, A. (2012, Dec). Brand-Related Events Detection, Classification and Summarization on Twitter. *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on* , (pp. 297 - 302).

Qiu, G. L. (2014). Opinion word expansion and target extraction through double propagation. *Computational linguistics*, (pp. 9-27).

Wibisono, Y. (2008). Penggunaan Hidden Markov Model untuk Kompresi Kalimat. . *Thesis. Program Magister Informatika. Institut Teknologi Bandung.* .

BIOGRAFI PENULIS

Septiyan Andika Isanta lahir di Lumajang pada tanggal 25 September 1989. Pada September tahun 2011 Penulis telah menyelesaikan studi S1 sebagai Sarjana Komputer (S.Kom) di Jurusan Teknik Informatika, Universitas Muhammadiyah Malang. Setelah itu Penulis bekerja di beberapa perusahaan di kota Malang, pada tahun 2013 Penulis mendapat kesempatan untuk melanjutkan studi S2 di Program Pascasarjana Teknik Informatika ITS Surabaya. Pada Januari 2015 Penulis telah telah mengikuti ujian Tesis sebagai syarat mendapatkan gelar Magister Komputer di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Penulis memiliki minat riset di bidang *Sentiment Analysis*, *Dokumen processing* dan *text mining*.

Email korespondensi : septiyan.andika@gmail.com