TESIS – IS185401

ANALISIS *CLICKBAIT* PADA JUDUL BERITA BAHASA INDONESIA MENGGUNAKAN WORD2VEC, NODE2VEC, DAN SUPPORT VECTOR MACHINE

NURRIDA AINI ZUHROH
NRP. 05211850010004

Dosen Pembimbing
Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2020

THESIS – IS185401

CLICKBAIT ANALYSIS ON INDONESIAN NEWS TITLE USING WORD2VEC, NODE2VEC, AND SUPPORT VECTOR MACHINE

NURRIDA AINI ZUHROH
NRP. 05211850010004

Supervisor
Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

Department of Information Systems
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember

2020

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom)
di
Institut Teknologi Sepuluh Nopember
Oleh:

NURRIDA AINI ZUHROH
NRP: 05211850010004

Tanggal Ujian: 17 Januari 2020
Periode Wisuda: Maret 2020

Disetujui oleh:
Pembimbing:

1. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D. .....
   NIP: 198201202005012001

Penguji:

1. Dr. Apol Pribadi Subriadi, S.T., M.T. .....
   NIP: 197002252009121001

2. Ahmad Muklason, S.Kom., M.Sc., Ph.D. .....
   NIP: 198203022009121009

Kepala Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas

Dr. Mudjahidin, S.T., M.T.
NIP: 197010102003121001

(Halaman sengaja dikosongkan)

ANALISIS *CLICKBAIT* PADA JUDUL BERITA BAHASA INDONESIA MENGGUNAKAN WORD2VEC, NODE2VEC, DAN SUPPORT VECTOR MACHINE

Nama Mahasiswa	: Nurrida Aini Zuhroh
NRP	: 05211850010004
Dosen Pembimbing	: Nur Aini Rachmawati, S.Kom., M.Sc.Eng., Ph.D

ABSTRAK

Saat ini portal berita daring menjadi salah satu sumber penyedia informasi yang banyak diakses oleh masyarakat Indonesia. Salah satu cara media daring memperoleh pendapatan adalah melalui jumlah *traffic* pengunjung situs. Hal ini menyebabkan persaingan antar media daring semakin ketat untuk meningkatkan jumlah pengunjungnya. Salah satu cara yang dilakukan media untuk menarik pengunjung adalah membuat judul berita yang hiperbola sehingga mendorong seseorang untuk membaca berita tersebut atau biasa disebut *clickbait*. Isi dari *clickbait* cenderung mengecewakan bagi pembaca karena kualitas dari konten yang buruk, kurang informatif, dan dapat menggiring opini publik. Sehingga pada penelitian ini dilakukan pendeteksian berita *clickbait* menggunakan algoritma klasifikasi *support vector machine* (SVM) dengan Word2Vec dan Node2Vec sebagai representasi fitur. Selain itu juga dilakukan analisis dengan menggunakan hasil dari *embedding* Node2Vec untuk mengetahui pola dari berita yang biasa digunakan sebagai *clickbait*. *Dataset* yang digunakan pada penelitian ini diambil dari 7 situs berita yang memiliki jumlah pengunjung terbanyak di Indonesia. Berdasarkan penelitian yang telah dilakukan, diketahui bahwa penggunaan *graph embedding* menunjukkan performa klasifikasi yang lebih baik dibandingkan dengan *word embedding* sebagai representasi fitur dengan nilai *precision* 84% untuk penggunaan *graph embedding* dan 80% untuk penggunaan *word embedding*. Selain itu, dengan penggunaan *word embedding* sebagai representasi fitur diketahui bahwa kata *stopword* pada judul *clickbait* memiliki pengaruh terhadap hasil nilai pengujian. Untuk analisis pola judul berita diketahui bahwa berdasarkan kategori dan tag berita, antara berita *clickbait* dan *nonclickbait* tidak memiliki perbedaan yang signifikan. Berdasarkan hasil pengecekan *similarity* pada berita *clickbait* diperoleh bahwa 40% dari berita yang terklasifikasi *clickbait* memiliki kesamaan dengan berita *nonclickbait*.

Kata kunci : deteksi *clickbait*; *word2vec*; *support vector machine*; *node2vec; word embedding*; *graph embedding*

(Halaman sengaja dikosongkan)

CLICKBAIT ANALYSIS ON INDONESIAN NEWS TITLE USING WORD2VEC, NODE2VEC, AND SUPPORT VECTOR MACHINE

By : Nurrida Aini Zuhroh
Student Identity Number : 05211850010004
Supervisor : Nur Aini Rachmawati, S.Kom., M.Sc.Eng., Ph.D

ABSTRACT

Nowadays, the online news portal is one of the sources of information that is widely accessed by people. One way that online media earn revenue is through numbers of traffic. This causes competition among online media increasingly tight to increase the number of visitors. One way that the media do to attract visitors is to create hyperbole headlines that encourage someone to read the news or commonly called clickbait. The contents of clickbait tend to disappoint the reader because the quality of the content is poor, less informative, and can lead to public opinion. So, in this study, clickbait news detection was performed using the support vector machine (SVM) classification algorithm with Word2Vec and Node2Vec as feature representations. Also, an analysis using the results of Node2Vec embedding was used to determine the pattern of news that is commonly used as a clickbait. The dataset used in this study was taken from 7 news sites that have the most visitors in Indonesia. From this study, it is known that the use of graph embedding shows better performance than word embedding as a feature with a precision value of 84% for the use of graph embedding and 80% for the use of word embedding. Based on the results of the classification using word embedding as a feature representation, it is known that the word stopword in the clickbait title influences the test performance results. For the analysis of news headline patterns, it is known that based on news categories and tags between clickbait and nonclickbait news do not have significant differences. The results of checking the similarity of clickbait news, it was found that 40% of news classified as clickbait has similarities with nonclickbait news.

Keywords : clickbait detection; word2vec; support vector machine; node2vec; word embedding; graph embedding

(Halaman sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas berkat rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan tesis yang berjudul "ANALISIS CLICKBAIT PADA JUDUL BERITA BAHASA INDONESIA MENGGUNAKAN WORD2VEC, NODE2VEC, DAN SUPPORT VECTOR MACHINE". Selama proses penyelesaian tesis ini, penulis memperoleh banyak bimbingan, bantuan, dan dukungan dari berbagai pihak. Sehingga pada kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga yang telah memberikan doa, semangat, motivasi, dan senantiasa memberi dukungan kepada penulis dalam kondisi apapun.
2. Ibu Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D. selaku dosen pembimbing yang telah sabar dan telaten mendukung, membimbing serta membagikan ilmu dan waktunya kepada penulis dalam penyelesaian penelitian ini.
3. Bapak Dr. Apol Pribadi Subriadi dan Bapak Ahmad Muklason, Ph.D., selaku dosen penguji yang telah memberikan kritik dan saran untuk perbaikan penelitian ini.
4. Seluruh bapak ibu dosen dan karyawan di Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.
5. Teman-teman S2 Sistem Informasi angkatan 2018 Gasal dan Keluarga Lab. S2 ADDI yang telah memberikan bantuan dan dukungan kepada penulis.
6. Semua pihak yang belum disebutkan, yang telah membantu dan mendukung penulisan tesis ini.

Penulis menyadari bahwa tesis ini masih jauh dari sempurna, oleh karena itu peneliti menerima kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga penelitian ini dapat memberikan wawasan dan manfaat yang berguna bagi pengembangan ilmu pengetahuan dan pembaca.

Surabaya, Januari 2020

Nurrida Aini Zuhroh

(Halaman sengaja dikosongkan)

DAFTAR ISI

(Halaman sengaja dikosongkan)

DAFTAR GAMBAR

(Halaman sengaja dikosongkan)

DAFTAR TABEL

(Halaman sengaja dikosongkan)

BAB 1
PENDAHULUAN

1.1. Latar Belakang

Perkembangan yang terjadi pada teknologi informasi, telekomunikasi, dan media mendorong masyarakat untuk memasuki era revolusi digital. Saat ini masyarakat dapat dengan mudah dan cepat memperoleh berbagai informasi melalui internet. Perubahan tersebut akhirnya mendorong para pelaku di dunia industri media massa untuk melakukan pengembangan produknya dari cetak beralih ke digital. Saat ini, berbagai portal media daring sudah banyak tersedia dengan menyajikan berbagai informasi yang dapat dibaca oleh masyarakat. Dari hasil riset yang dilakukan oleh HootSuite dan We Are Social dari Januari 2018 hingga Januari 2019, menunjukkan bahwa pengguna internet di Indonesia saat ini meningkat sebanyak 13% dari tahun sebelumnya (Hootsuit and We Are Social, 2019). Tercatat bahwa pengguna internet aktif saat ini mencapai 150 juta penduduk atau 56% dari total populasi dengan jumlah pembaca berita di media daring sekitar 6 juta orang perhari. Berdasarkan data yang dihimpun oleh We Are Social, menunjukkan bahwa rata-rata orang menghabiskan waktunya selama 8 jam 36 menit menggunakan internet setiap harinya (Hootsuit and We Are Social, 2019).

Saat ini, fenomena berita hoaks semakin meningkat, sehingga membuat masyarakat menjadi resah. Dari hasil survei yang dilakukan oleh Lembaga Survei Indonesia (LSI) Denny JA menunjukkan bahwa sebanyak 75% masyarakat merasa khawatir dengan semakin maraknya berita hoaks (LSI Denny JA, 2018). Pada demografi yang dibuat oleh katadata.co.id pada tahun 2018, menunjukkan bahwa tingkat kepercayaan masyarakat Indonesia terhadap media yakni sebesar 68% (katadata.co.id, 2018a). Sedangkan untuk tingkat kepercayaan publik terhadap media sosial mengalami penurunan di tahun 2018 sebanyak 2 poin menjadi hanya 51% saja. Akan tetapi, untuk tingkat kepercayaan publik terhadap jurnalis menunjukkan kenaikan sebesar 5% dari tahun sebelumnya dengan total sebanyak 59% (katadata.co.id, 2018b). Hal ini menunjukkan bahwa secara umum media

daring memiliki pengaruh yang cukup besar dan menjadi salah satu sumber informasi utama dikalangan masyarakat.

Dengan meningkatnya pengguna internet saat ini, berdampak pada semakin banyaknya portal media daring yang tersedia. Dari data pada laman Dewan Pers, tercatat bahwa terdapat 1512 media yang terdaftar baik media daring maupun cetak (Dewan Pers, 2019). Hal ini menyebabkan persaingan antar media daring untuk memperoleh pengunjung semakin ketat. Salah satu cara media daring memperoleh penghasilan adalah melalui jumlah pengguna internet yang mengunjungi situsnya. Hal ini merupakan salah satu cara yang biasa digunakan untuk mengukur tingkat efektifitas suatu portal berita untuk menarik pengunjung di dunia maya. Sehingga beberapa media daring melakukan berbagai strategi untuk meningkatkan jumlah pengunjung yang justru menjurus ke arah negatif, salah satunya adalah *clickbait*. *Clickbait* merupakan istilah yang digunakan pada suatu judul berita yang penulisannya terkesan berlebihan, sensasional, atau membingungkan. Tujuannya adalah untuk memancing pengguna internet melakukan klik pada tautan berita tersebut. *Clickbait* biasanya memiliki cara penulisan yang provokatif yang dapat membuat pengguna menangkap persepsi yang salah terhadap suatu berita (Anand, Chakraborty and Park, 2017). Sehingga berdasarkan hal tersebut, judul *clickbait* juga memiliki potensi melanggar kode etik jurnalistik yang telah ditetapkan oleh Dewan Pers yakni pasal 1 yang berisi "Wartawan Indonesia bersikap independen, menghasilkan berita yang akurat, berimbang, dan tidak beritikad buruk" dan pasal 3 yang berisi "Wartawan Indonesia selalu menguji informasi, memberitakan secara berimbang, tidak mencampurkan fakta dan opini yang menghakimi, serta menerapkan asas praduga tak bersalah".

Judul berita merupakan salah satu unsur yang penting pada suatu berita. Judul mampu memberikan impresi awal yang akan diterima oleh seorang pengguna dan dapat mempengaruhi bagaimana pengguna akan menangkap persepsi dari suatu berita (Agrawal, 2016). *Clickbait* tidak serta merta ada begitu saja, tetapi merupakan strategi dari penulis atau editor untuk membuat judul berita yang dapat menarik perhatian dan memanipulasi emosi pembaca sehingga muncul rasa penasaran. Salah satu teori yang berhubungan dengan *clickbait* adalah *curiosity gap*

(Loewenstein, 1994) yang dikemukakan oleh Loewenstein. Teori tersebut menjelaskan bahwa *curiosity gap* dapat terjadi karena terdapat celah antara sesuatu yang telah diketahui dengan sesuatu yang ingin diketahui oleh individu. Hal ini menimbulkan kesenjangan informasi sehingga menimbulkan dampak secara emosional yang akan membuat individu termotivasi mendapatkan informasi yang hilang untuk mengisi kekurangan tersebut. Hal ini yang dimanfaatkan oleh media untuk menggugah rasa penasaran pembaca untuk mengetahui kejelasan dan menghilangkan rasa penasaran tersebut.

Penelitian awal mengenai deteksi *clickbait* dengan menggunakan metode *machine learning* dilakukan oleh Potthast et al (Potthast et al., 2016) pada tahun 2016. Penelitian tersebut menggunakan *dataset* yang diambil dari sosial media Twitter. *Dataset* yang digunakan sebanyak 2992 *tweet* dan 767 dari data tersebut dikategorikan sebagai *clickbait*. Pada penelitian yang dilakukan oleh (Chakraborty et al., 2016), peneliti melakukan analisis pada 15.000 judul berita clickbait dan non-clickbait menggunakan Stanford CoreNLP untuk menemukan beberapa karakteristik yang membedakan antara judul berita *clickbait* dan non-*clickbait*. Perbedaan yang pertama adalah dari struktur kalimat. Judul berita *clickbait* cenderung memiliki struktur kalimat yang lebih panjang daripada artikel non-*clickbait*. Selain itu pada artikel *clickbait* juga ditemukan banyak penggunaan kata yang hiperbolik dan penggunaan kata-kata slang. Penelitian selanjutnya (Rony, Hassan and Yousuf, 2017) dilakukan dengan menggunakan dua *dataset* yaitu *dataset* headlines yang disediakan oleh (Chakraborty et al., 2016) dan media *corpus* dari data Facebook post menggunakan Facebook Graph API. Salah satu tujuan dari penelitian yang dilakukan adalah untuk melakukan analisis dampak dari *clickbait*. Hasil dari penelitian tersebut menunjukkan bahwa berita *clickbait* lebih banyak memperoleh respon dari pengguna daripada berita non-*clickbait*. Studi terkini mengenai deteksi *clickbait* dilakukan oleh Dong et al (Dong et al., 2019) dengan mengukur kesamaan antara judul dan isi berita menggunakan *cosine similarity*. Studi tersebut menggunakan algoritma Bidirectional Gated Recurrent Unit (BiGRU) sebagai metode klasifikasi.

Saat ini, penelitian mengenai deteksi *clickbait* cukup mendapat perhatian. Akan tetapi, penelitian mengenai deteksi *clickbait* pada artikel berita bahasa Indonesia masih terbatas. Salah satu penelitian mengenai deteksi *clickbait* pada artikel berita bahasa Indonesia dilakukan oleh Maulidi et al. Pada penelitian tersebut, peneliti menggunakan algoritma klasifikasi Naive Bayes pada 1000 judul berita (Maulidi and Palandi, 2018). Sedangkan pada penelitian lain yang dilakukan oleh Yavi, peneliti menggunakan metode Neural Network (NN) pada 800 judul berita (Yavi, 2018). Pada penelitian tersebut ditemukan bahwa judul berita *clickbait* memiliki perubahan pola secara dinamis. Berdasarkan pada penelitian-penelitian sebelumnya, maka pada penelitian ini akan diusulkan klasifikasi menggunakan algoritma dari *machine learning*, yaitu Support Vector Machine (SVM) dengan menggunakan Word2Vec, dan Node2Vec sebagai representasi fitur. Word2Vec merupakan algoritma dari *word embedding* yang digunakan untuk merepresentasikan kata menjadi bentuk vektor. Sedangkan Node2Vec merupakan algoritma dari *graph embedding* yang merepresentasikan *node* dan relasinya dalam bentuk vektor. Penggunaan dari Word2Vec dan Node2Vec sendiri diharapkan dapat meningkatkan performa klasifikasi dari algoritma yang digunakan. Selain itu, pada penelitian ini akan dilakukan analisis pada pola atau topik yang biasa dijadikan sebagai judul berita *clickbait* oleh media daring. Proses analisis ini dilakukan dengan menggunakan hasil dari *graph embedding* dengan menggunakan Node2Vec.

1.2. Rumusan Masalah

Berdasarkan pada latar belakang yang telah dijabarkan, dapat diketahui bahwa penelitian mengenai deteksi *clickbait* untuk artikel berbahasa Indonesia masih minim dan tingkat akurasi yang dihasilkan masih rendah. Sehingga, rumusan masalah yang ingin dijawab pada penelitian ini adalah sebagai berikut:

1. Bagaimana melakukan pendeteksian *clickbait* pada artikel berita bahasa Indonesia menggunakan algoritma klasifikasi SVM dengan penggunaan Word2Vec untuk merepresentasikan teks sebagai fitur dan penggunaan *graph embedding* dari Node2Vec sebagai fitur?

2. Bagaimana kinerja dari penggunaan SVM sebagai algoritma klasifikasi dengan menggunakan *word embedding* dan *graph embedding* sebagai fitur dalam pendeteksian judul berita *clickbait* bahasa Indonesia?
3. Bagaimana hasil analisa dari artikel berita bahasa Indonesia yang terindikasi *clickbait* menggunakan Node2Vec?

1.3. Tujuan

Dari perumusan masalah yang telah didefinisikan, maka tujuan dari penelitian ini adalah (1) melakukan pendeteksian judul *clickbait* pada artikel berita bahasa Indonesia dengan menggunakan algoritma SVM sebagai teknik klasifikasi dengan penggunaan *word embedding* dan *graph embedding* sebagai representasi fitur. (2) Mengukur kinerja dari penggunaan algoritma SVM serta penggunaan *word embedding* dan *graph embedding* sebagai representasi fitur. (3) Melakukan analisis pola atau topik yang terkait dengan judul berita *clickbait* dengan menggunakan Node2Vec.

1.4. Manfaat Penelitian

Manfaat dari penelitian ini adalah memberikan gambaran penggunaan metode untuk mendeteksi judul berita *clickbait* berbahasa Indonesia dengan menggunakan SVM dan membandingkan antara hasil penggunaan *word embedding* dan *graph embedding* sebagai representasi fitur. Selain itu, diharapkan hasil dari penelitian ini dapat digunakan lebih lanjut sebagai evaluasi bagi media daring dalam menulis artikel berita. Dengan melakukan pencarian dan analisis pola dari artikel berita yang terindikasi *clickbait*, dapat diketahui topik berita apa saja yang biasa digunakan oleh media daring dalam penulisan berita *clickbait*.

1.5. Kontribusi Penelitian

Berikut akan dijelaskan mengenai kontribusi teoritis dan praktis pada penelitian ini.

1.5.1. Kontribusi Teoritis

Kontribusi secara teoritis dari penelitian ini adalah memberikan kontribusi pada penelitian tentang analisis pendeteksian *clickbait* untuk judul berita bahasa Indonesia dengan menggunakan metode *machine learning* dengan penggunaan *word embedding* dan *graph embedding* sebagai fitur untuk klasifikasi dan evaluasi dari kinerja klasifikasi. Selain itu juga memberikan gambaran mengenai penggunaan Node2Vec yang menghasilkan *embedding* dan dapat digunakan sebagai fitur untuk klasifikasi serta analisis dari pola judul berita *clickbait*.

1.5.2. Kontribusi Praktis

Kontribusi secara praktis dari penelitian ini adalah tahap awal untuk membantu pengembangan lebih lanjut yang berguna untuk masyarakat dalam menyaring berita yang dibaca di internet dan juga dapat digunakan sebagai evaluasi bagi media daring untuk menyajikan berita yang lebih informatif kepada masyarakat. Selain itu juga, penelitian ini menyediakan *dataset* yang dapat digunakan untuk penelitian selanjutnya.

1.6. Batasan Penelitian

Penelitian ini memiliki ruang lingkup yang akan menjadi batasan dalam penelitian ini. Batasan pada penelitian ini antara lain:

1. Data yang digunakan merupakan kumpulan artikel berita bahasa Indonesia.
2. Data diambil dari situs tribunnews.com, detik.com, liputan6.com, kompas.com, okezone.com, merdeka.com, dan cnnindonesia.com yang diterbitkan pada tahun 2019
3. Dataset *clickbait* yang digunakan terbatas pada judul berita yang mengandung kata kunci begini, ini , (!), (?), (..), ternyata, terungkap.

1.7. Sistematika Penulisan Laporan

Penulisan laporan penelitian dibuat secara sistematis dengan dilakukan sebagai berikut:

Bab 1: Pendahuluan

Bab ini berisi penjelasan terkait latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, kontribusi penelitian, dan sistematika penulisan laporan. Pada bab ini diharapkan mampu memberikan gambaran yang umum dari penelitian yang akan dilakukan.

Bab 2 : Kajian Pustaka

Bab ini berisi mengenai pembahasan teori-teori dari penelitian terkait. Pada bab ini akan digunakan sebagai landasan penelitian.

Bab 3 : Metodologi Penelitian

Bab ini berisi tentang pembahasan tahapan penelitian dan rencana penelitian. Kemudian dijelaskan mengenai detail dari tahapan penelitian beserta jadwal kegiatan pengerjaan penelitian. Selain itu, bab ini digunakan sebagai panduan dalam penyusunan penelitian agar terarah dan sistematis.

Bab 4 : Hasil dan Pembahasan

Bab ini berisi mengenai penjelasan mengenai temuan, hasil dan pembahasan dari hasil penelitian yang dilakukan pada penelitian ini.

Bab 5 : Kesimpulan dan Saran

Bab ini berisi mengenai kesimpulan dan saran yang didapatkan dari penelitian yang dilakukan beserta dengan referensi mengenai penelitian kedepannya.

Daftar Pustaka

Berisi daftar referensi yang digunakan dalam penelitian ini, baik jurnal, buku, maupun artikel.

(Halaman sengaja dikosongkan)

BAB 2
KAJIAN PUSTAKA

Pada bab ini akan dijelaskan mengenai teori-teori yang berhubungan dengan penelitian yang akan dilakukan dan teori yang digunakan pada penelitian yang sudah pernah dilakukan sebelumnya.

2.1. Kajian Teori

Pada bagian ini akan dijelaskan mengenai teori-teori yang terkait pada penelitian yang akan dilakukan mengenai deteksi *clickbait*.

2.1.1. *Clickbait*

Clickbait merupakan istilah yang merujuk pada suatu konten situs yang menarik pengguna internet untuk melakukan klik pada konten tersebut dengan menggunakan bahasa yang hiperbola atau sensasional dengan kualitas konten yang buruk (Agrawal, 2016). Tujuan dari *clickbait* adalah untuk menarik pengguna internet mengunjungi halaman situs yang terhubung sehingga meningkatkan jumlah pengunjung dari situs tersebut. Pada penelitian yang dilakukan oleh (Reis et al., 2015) dengan melakukan analisis sentimen pada 69.907 judul berita menemukan bahwa judul berita yang masuk pada kategori positif atau negatif yang sangat kuat memiliki kecenderungan lebih menarik bagi pembaca media daring. Salah satu teori yang dapat menjadi dasar dari penggunaan *clickbait* adalah *curiosity gap* (Loewenstein, 1994). Pada penelitian tersebut Loewenstein menjelaskan mengenai *information gap*. Teori tersebut menjelaskan ketika terdapat kesenjangan antara apa yang diketahui dengan apa yang ingin diketahui maka akan memberikan dampak secara emosional pada seseorang. Rasa emosi yang ditimbulkan dari kesenjangan informasi tersebut adalah rasa penasaran atau *curiosity*. Sehingga seseorang akan termotivasi untuk memperoleh informasi dari kesenjangan yang ditimbulkan untuk mengurangi atau menghilangkan rasa penasaran tersebut. Beberapa karakteristik dari judul berita clickbait adalah (Wargadiredja, 2017):

- Biasanya menggunakan kata yang terkesan hiperbolis seperti: Terungkap, Heboh, Viral yang sebenarnya memiliki konten yang bagi pembaca biasa saja.
- Judul yang sengaja dipotong kalimatnya atau yang terkesan misterius dengan menggunakan kata ganti "ini" untuk menyembunyikan nama orang, benda, lokasi, atau suatu pernyataan.

Selain itu, berdasarkan penelitian yang dilakukan oleh (Biyani, Tsioutsiouliklis and Blackmer, 2016), berikut merupakan 8 tipe dari judul berita *clickbait*:

1. *Exaggeration*

   Exaggeration merupakan bentuk *clickbait* dengan judul yang melebih-lebihkan dari isi konten. Contohnya adalah artikel yang dimuat oleh cnnindonesia.com dengan judul "Ahok Akan Jadi Bos, Saham BUMN Kompak Melemah" seperti pada Gambar 1. Judul berita tersebut berlebihan karena isu penunjukkan Ahok masih rencana dan belum jelas akan ditunjuk pada BUMN mana dan melemahnya beberapa saham BUMN tidak ada kaitannya dengan rencana penunjukan Ahok.

Gambar 1. Contoh judul berita *exagerration* (sumber: https://intip.in/dZx2)

2. *Teasing*

   Teasing merupakan *clickbait* yang dibuat dengan menghapus detail dari judul untuk membangun ketegangan atau menggoda (Hadiyat, 2019). Berikut

merupakan contoh judul berita *teasing* yang dimuat oleh suara.com dengan judul berita "Prabowo Meninggal Dunia Kena Serangan Jantung, 1 Hari Tak Makan Urus Pemilu" seperti yang ditunjukkan pada Gambar 2, Prabowo yang dimaksud pada judul berita tersebut bukan merupakan Prabowo yang mencalonkan diri sebagai presiden, tetapi Prabowo yang merupakan salah satu ketua KPPS (Kelompok Penyelenggara Pemungutan Suara).

Gambar 2. Contoh judul berita *teasing* (sumber: https://intip.in/oPtW)

3. *Inflamatory*

Inflamatory merupakan judul berita yang memiliki tujuan untuk membangkitkan rasa marah atau penuh kekerasan dengan penggunaan kata-kata yang tidak pantas atau vulgar (Hadiyat, 2019). Berikut merupakan contoh judul berita *inflamatory* yang dimuat di situs viva.co.id dengan judul "Ribuan Orang Ditangkap Gara-gara Unjuk Rasa Ganti Presiden" seperti pada Gambar 3. Dengan kondisi politik di Indonesia saat ini, secara sekilas pembaca akan mengira bahwa hal tersebut terjadi di Indonesia, hal tersebut dapat menyebabkan kesalahpahaman dan menyulut emosi masyarakat. Padahal judul artikel tersebut dimaksudkan bukan terjadi di Indonesia, tetapi di negara Mesir.

Gambar 3. Contoh judul berita *inflamatory* (sumber: https://intip.in/NGmz)

4. *Formatting*

   Formatting merupakan judul berita *clickbait* yang terlalu sering menggunakan huruf kapital atau tanda baca seperti tanda seru. Berikut merupakan contoh dari judul yang diterbitkan oleh detik.com dengan judul "Ngeri! Tangan Pemuda Cianjur Putus Ditebas Gerombolan Bikers" seperti pada Gambar 4.

Gambar 4. Contoh judul berita *formatting* (sumber: https://intip.in/w7WW)

5. *Graphic*

   Graphic merupakan judul yang mengandung konten yang mengganggu atau menjijikkan dan tidak dapat dipercaya (Hadiyat, 2019).

6. *Bait-and-switch*

   Bait-and-switch merupakan judul *clickbait* yang memiliki konten tidak lengkap atau hal yang termuat pada judul tidak ada di dalam konten dan mengarah ke tautan lain.

7. *Ambigous*

   Ambigous merupakan judul yang tidak jelas dan memiliki makna yang ambigu atau membingungkan dengan tujuan untuk memancing keingintahuan dan rasa penasaran. Berikut merupakan contoh judul berita *ambigous* yang diterbitkan oleh regional.kompas.com dengan judul berita "Dilaporkan ke Bawaslu terkait Acungan Satu Jari, Ini Kata Ridwan Kamil" seperti pada Gambar 5. Berita tersebut berisi mengenai Ridwan Kamil yang mengikuti kegiatan kampanye di luar jam kerja sebagai jam dinas dan menyatakan bahwa acungan satu jari yang disebutkan terkait dengan partai PKB bukan dengan pemilihan presiden untuk mendukung Jokowi.

Gambar 5. Contoh judul berita *ambigous* (sumber: https://intip.in/gcRq)

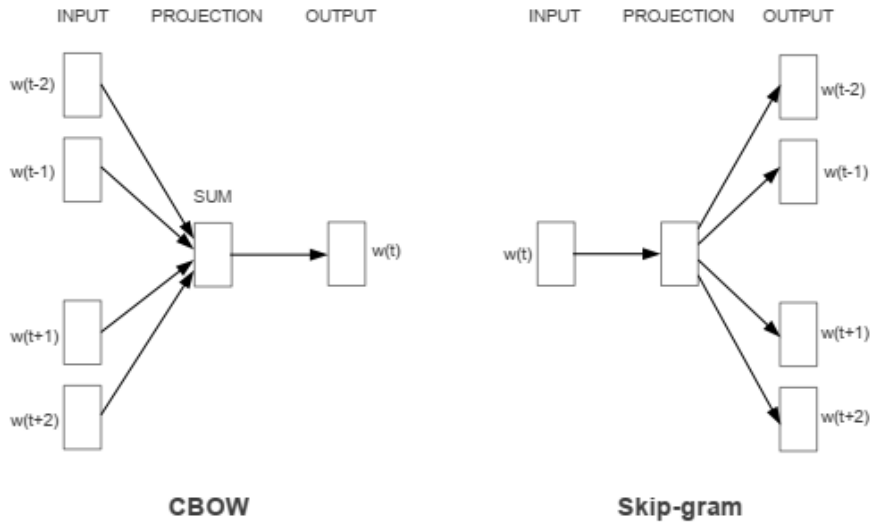
8. *Wrong*

   Wrong merupakan jenis *clickbait* yang salah atau memuat pernyataan yang tidak benar. Berita seperti ini dapat juga dikategorikan sebagai berita *hoax*.

2.1.2. *Word Embedding*

Word Embedding merupakan istilah yang digunakan untuk representasi dari kata-kata yang terdistribusi. Teknik ini sering digunakan pada penggalian teks atau *Natural Language Processing* (NLP) (Li et al., 2015). Cara kerja dari teknik ini adalah dengan merepresentasikan kumpulan kata menjadi suatu vektor. Sehingga dari representasi tersebut dapat diolah lebih lanjut untuk mengetahui konteks dari suatu kalimat atau untuk memperoleh informasi mengenai kesamaan semantik dan sintaksis serta keterkaitannya dengan kata lain.

Salah satu algoritma *word embedding* yang sering digunakan adalah Word2Vec (Mikolov et al., 2013). Teknik ini memiliki 2 model pendekatan yakni, Continous Bag of Words (CBOW) dan Skip-gram. Secara sederhana, Skip-gram melakukan prediksi konteks atau kata sekitarnya berdasarkan kata yang digunakan sebagai masukan, sedangkan CBOW melakukan prediksi kata berdasarkan konteksnya seperti yang ditunjukkan pada Gambar 6.



Gambar 6. Arsitektur model CBOW dan Skip-gram (Mikolov et al., 2013)

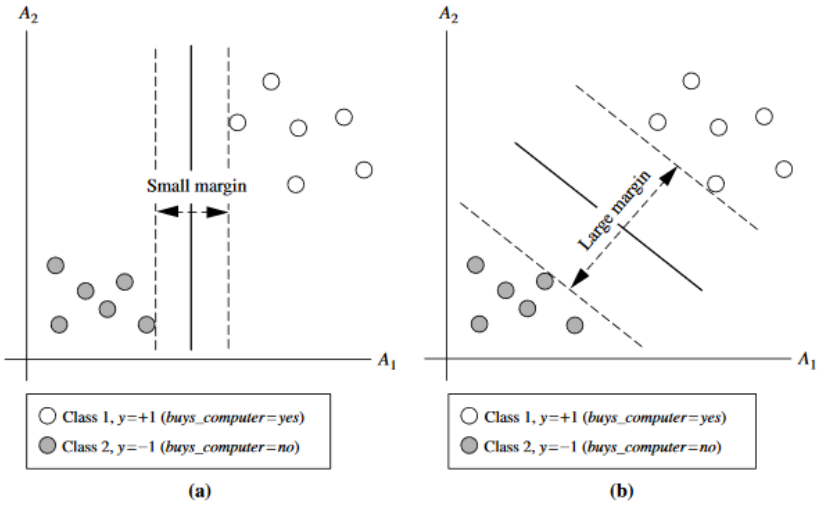
2.1.3. *Machine Learning*

Salah satu teknik yang digunakan untuk melakukan deteksi *clickbait* pada penelitian terdahulu adalah dengan menggunakan algoritma klasifikasi dari *machine learning*. Pada penelitian yang dilakukan oleh Chakraborty et al.

mengenai deteksi *clickbait* dengan menggunakan tiga algoritma dari *machine learning* sebagai perbandingan, yakni Support Vector Machine (SVM), Decision Tree, dan Random Forest (Chakraborty et al., 2016). Pada penelitian tersebut ia menggunakan beberapa fitur seperti pola kata, bahasa yang sering digunakan pada artikel *clickbait*, dan n-*gram*. Dari hasil pengujian yang telah dilakukan, SVM menunjukkan performa paling baik dari ketiga algoritma yang dipilih.

2.1.4.1. Support Vector Machine (SVM)

SVM merupakan salah satu algortima *supervised learning* yang cukup populer digunakan untuk klasifikasi ataupun regresi (Binkhonain and Zhao, 2019). SVM pertama kali dikenalkan oleh Vapnik dan Chervonenkis pada tahun 1964 tetapi mulai mendapat perhatian dari publik mulai tahun 1990an. Cara kerja dari algoritma ini adalah dengan mentransformasikan data *training* menjadi dimensi yang lebih tinggi, dimana algoritma tersebut akan mencari *hyperplane* yang memisahkan data berdasarkan *class* dengan menggunakan *training tuple* yang disebut *support vector* (Han, Kamber and Pei, 2012). Tujuan dari algoritma SVM adalah untuk mencari *hyperplane* dengan *margin* yang terbesar atau biasa disebut dengan *maximum marginal hyperplane* (Han, Kamber and Pei, 2012).



Gambar 7. Perbedaan *small* margin dan *large margin* (Han, Kamber and Pei, 2012)

Pada teori yang dijelaskan oleh Han et al., pada Gambar 7 menunjukkan bahwa kedua *hyperplane* dapat melakukan klasifikasi dengan benar pada *data tuples* yang diberikan. Tetapi dapat diketahui bahwa *hyperplane* dengan *margin* yang lebih lebar memiliki tingkat akurasi klasifikasi yang lebih tinggi daripada *hyperplane* dengan margin yang lebih kecil.

Rumus dari *hyperplane* tersebut dapat ditulis seperti pada persamaan (2.1), dimana W merupakan *weight vector* $W = \{w_1, w_2, \ldots\ldots, w_n\}$; n merupakan jumlah atribut dan b merupakan skalar atau bias.

$$W \cdot X + b = 0 \tag{2.1}$$

Berdasarkan Gambar 7 diasumsikan bahwa terdapat dua atribut masukan, A_1 dan A_2 dengan *training tuple* 2-D (contoh: $X = (x_1, x_2)$). Diketahui bahwa x_1 dan x_2 merupakan atribut nilai dari A_1 dan A_2. Jika diasumsikan bahwa b merupakan *additional weight*, berdasarkan persamaan (2.1) maka w_0 dapat ditulis

$$w_0 + w_1 x_1 + w_2 x_2 = 0 \tag{2.2}$$

Sehingga untuk setiap titik yang ada diatas *hyperplane* memenuhi

$$w_0 + w_1 x_1 + w_2 x_2 > 0 \tag{2.3}$$

Sedangakn, untuk setiap titik yang ada dibawah *hyperplane* memenuhi

$$w_0 + w_1 x_1 + w_2 x_2 < 0 \tag{2.4}$$

Bobot juga dapat disesuaikan, sehingga *hyperplane* untuk mendefinisikan kedua sisi dari margin dapat ditulis dengan persamaan

$$H_1 : w_0 + w_1 x_1 + w_2 x_2 \geq 1 \qquad \text{untuk } y_i = +1 \tag{2.5}$$

$$H_2 : w_0 + w_1 x_1 + w_2 x_2 \leq -1 \qquad \text{untuk } y_i = -1 \tag{2.6}$$

Sehingga dari persamaan (2.5) bahwa setiap *tuple* yang terdapat pada H_1 dimiliki oleh kelas $+1$ dan *tuple* yang berada di bawah H_2 dimiliki oleh kelas -1. Jadi ketika kedua persamaan tersebut digabungkan maka,

$$y_i(w_0 + w_1 x_1 + w_2 x_2) \geq 1, \forall i \tag{2.7}$$

Maka setiap *tuple* yang berada diatas atau dibawah *hyperplane* H_1 dan H_2 yang memenuhi persamaan (2.7) disebut dengan *support vector*.

Berikut merupakan rumus dari linear kernel yang ada di SVM:

$$K(x_i, x_j) = x_i^T x_j \tag{2.8}$$

2.1.4. *Graph Embedding*

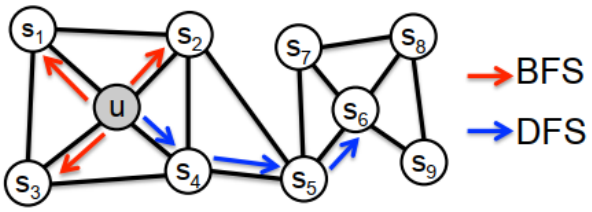
Saat ini *graph analysis* sudah mulai sering digunakan diberbagai studi seperti pada jaringan pertemanan di media sosial dan linguistik untuk menemukan keterkaitan antar kata. Pada penelitian yang dilakukan oleh Goyal et al (Goyal and Ferrara, 2018), mendefinisikan beberapa tugas dari *graph analysis* yang dibagi menjadi empat kategori yaitu: (1) klasifikasi *node*, (2) *link prediction*, (3) *clustering*, dan (4) visualisasi. Klasifikasi *node* memiliki tujuan untuk menentukan label dari suatu *node* berdasarkan *node* yang ada disekitarnya pada suatu jaringan. *Link prediction* digunakan untuk melakukan prediksi hubungan antar *node* yang mungkin terjadi. Sedangkan *clustering* digunakan untuk menemukan kelompok *node* yang memiliki kesamaan dan mengelompokkannya. Visualisasi berfungsi untuk memberikan gambaran mengenai struktur dari suatu jaringan *graph*. Teknik yang dapat digunakan untuk melakukan tugas-tugas tersebut adalah *graph embedding*.

Graph embedding merupakan istilah yang digunakan untuk teknik mentransformasikan *graph* menjadi satu set vektor. Pada penelitian yang dilakukan oleh Goyal et al (Goyal and Ferrara, 2018) membagi metode *graph embedding* berdasarkan tiga kategori, yaitu: (1) *Factorization*, (2) *Random Walk*, dan (3) *Deep Learning*. Pada penelitian ini, metode yang akan digunakan didasarkan pada *Random Walk*. *Random walk* merupakan sebuah algoritma yang menyediakan jalur acak pada suatu *graph* (Neo4J, 2019). Cara kerja dari algoritma ini adalah dengan memulai pada satu *node*, kemudian memilih *node* tetangga untuk memulai penelusuran secara acak atau berdasarkan distribusi dari probabilitas yang sudah ditentukan, dan menyimpan jalur yang dibuat dalam sebuah *list*. Salah satu algoritma dari *graph embedding* yang menggunakan *random walk* adalah Node2Vec.

2.1.2.1. Node2Vec

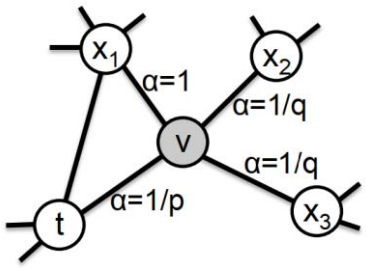
Node2Vec merupakan sebuah algoritma *semi-supervised* yang digunakan untuk pembelajaran fitur pada *node-node* dalam suatu jaringan (Grover and Leskovec, 2016). Cara kerja dari algoritma ini adalah dengan mengubah *node-node*

pada *graph* menjadi vektor. Node2Vec mempelajari pemetaan *node* ke *low-dimensional embeddings* pada suatu jaringan dengan menggunakan *random walks*. *Random walk* bertugas untuk melakukan eksplorasi pada *node-node* di sekitarnya. Prinsip kerja dari *random walk* merupakan penggabungan dari dua strategi *sampling* yakni Breadth-first Sampling (BFS) dan Depth-first Sampling (DFS). Gambar 8 menunjukkan contoh dari BFS dan DFS, untuk BFS ditunjukkan oleh S_1, S_2, S_3 dimana ukuran dari *neigborhood* $k = 3$, sedangkan untuk BFS ditunjukkan dengan node S_4, S_5, S_6.



Gambar 8. Strategi *sampling* BFS dan DFS

Pada Gambar 8 menjelaskan bahwa *node* u dan S_1 tergabung dalam komunitas yang sama, sedangkan *node* u dan S_6 terdapat pada dua komunitas yang berbeda tetapi masih memiliki hubungan keterkaitan pada satu *node* penghubung yang sama. Pada umumnya jaringan di dunia nyata secara sederhana juga memiliki kesamaan dengan prinsip yang dijelaskan pada Gambar 8. Sehingga untuk mengatasi hal tersebut diperlukan algoritma fleksibel yang mempelajari representasi dari suatu *node* dengan berdasarkan prinsip: kemampuan untuk mempelajari representasi dari suatu *node-node* yang saling berdekatan pada satu komunitas yang sama dalam satu *embedding*, dan juga mempelajari representasi dari *node-node* yang memiliki keterkaitan yang sama dan menempatkannya pada *embedding* yang serupa. Sehingga dengan menggunakan prinsip tersebut algoritma pembelajaran fitur mampu menggeneralisasi berbagai domain dan melakukan prediksi.



Gambar 9. Illutrasi dari prosedur *random walk* pada Node2Vec. (Grover and Leskovec, 2016)

Berdasarkan penelitian yang dilakukan oleh Grover et al. (Grover and Leskovec, 2016), pada Gambar 9 diasumsikan bahwa sebuah *random walk* baru saja melalui *edge* (*t*, *v*) dan sekarang berada di *node v*. Kemudian dari *v*, *random walk* perlu menentukan langkah berikutnya untuk mengevaluasi *transition probabilities* π_{vx} pada *node* (*v*, *x*) yang dimulai dari *v*. Untuk menetapkan *transition probabilities* menggunakan rumus: $\pi_{vx} = \alpha_{pq}(t,x) \cdot w_{vx}$, dimana α merupakan *search bias*:

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

Diketahui bahwa *static edge weights* merupakan w_{vx}, misal $\pi_{vx} = w_{vx}$ (Pada kasus *unweighted graphs* $w_{vx} = 1$). d_{tx} menunjukkan jarak dari jalur terpendek antara *node t* dan *x*. Parameter *p* merupakan *return parameter* yang mengontrol probabilitas melewati kembali suatu *node* yang sudah dikunjungi. Semakin tinggi nilai *p* maka kemungkinan untuk melewati *node* yang sudah dikunjungi semakin kecil. Sedangkan *q* merupakan *in-out parameter* yang mengontrol probabilitas untuk mengeksplor bagian *graph* yang belum dikunjungi. Jika nilai $q < 1$ maka *random walk* akan memiliki kecenderungan mengunjungi *node* yang lebih jauh dari *node* awal.

2.1.5. *Confussion Matrix*

Confussion matrix merupakan sebuah metode yang biasa digunakan dalam pengukuran kinerja pada metode klasifikasi. *Confussion matrix* memiliki tiga

penghitungan untuk mengukur kinerja dari klasifikasi yakni, *accuracy*, *precission*, dan *recall* seperti yang ditunjukkan pada Tabel 1. Akurasi merupakan perbandingan antara prediksi yang benar dengan keseluruhan data. *Precission* merupakan perbandingan antara prediksi yang bernilai benar positif dengan keseluruhan hasil yang diprediksi positif. *Recall* merupakan perbandingan antara prediksi benar positif dibandingkan dengan data yang benar positif. Sedangkan f1-score merupakan perbandingan rata-rata antara *precission* dan *recall*.

Tabel 1. *Confussion matrix*

		Nilai Prediksi	
		True	**False**
Nilai Sebenarnya	**True**	TP (*True Positive*)	FN (*False Negative*)
	False	FP (*False Positive*)	TN (*True Negative*)

Berikut merupakan persamaan rumus dari *confussion matrix*.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.9)$$

$$precission = \frac{TP}{TP+FP} \quad (2.10)$$

$$recall = \frac{TP}{TP+FN} \quad (2.11)$$

$$f - measure = \frac{2*precission*recall}{precission+recall} \quad (2.12)$$

2.2. Rangkuman Penelitian Terdahulu

Penelitian awal mengenai deteksi *clickbait* dilakukan oleh Potthast et al dengan menggunakan melakukan perbandingan pada tiga algoritma klasifikasi yaitu Logistic Regression, Naive Bayes, dan Random Forest. Pada penelitian tersebut diketahui bahwa algoritma Random Forest menunjukkan hasil performa terbaik (Potthast et al., 2016). Pada penelitian lain yang dilakukan oleh Chakraborty

et al, peneliti melakukan analisis pada linguistik pada judul-judul berita *clickbait* yang digunakan sebagai dataset menggunakan Stanford CoreNLP dan menemukan beberapa pola yang membedakan antara judul berita *clickbait* dan non-*clickbait*. Struktur kalimat dari judul berita *clickbait* cenderung lebih panjang, sering menggunakan kata yang termasuk dalam kategori *stopwords*, slang, dan penggunaan kata yang hiperbolis. Sedangkan untuk penelitian mengenai *clickbait* pada judul artikel berita bahasa Indonesia dilakukan oleh Yavi dengan menggunakan algoritma klasifikasi Naive Bayes (Yavi, 2018). Penelitian tersebut menggunakan 1000 data judul berita dengan 800 sebagai data *training* dan 200 sebagai data uji. Berikut menunjukkan mengenai algoritma klasifikasi dan fitur yang sudah pernah digunakan pada penelitian terdahulu seperti pada Tabel 2.

Tabel 2. Rangkuman penelitian terdahulu

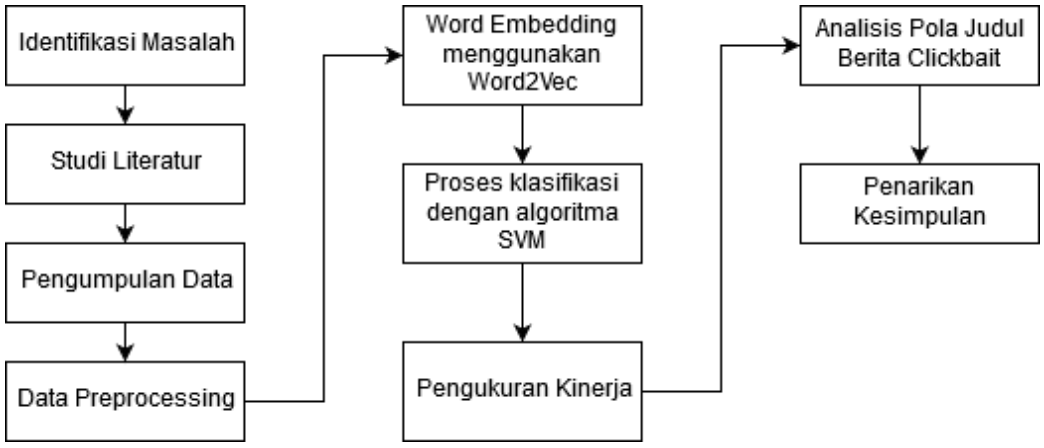
No	Penelitian	Algoritma Klasifikasi	Fitur	Embedding
1.	(Potthast et al., 2016)	- Logistic Regression - Naive Bayes - Random Forest	- *teaser message*/judul - *Link webpage* - *Meta information*	-
2.	(Chakraborty et al., 2016)	- Support Vector Machine - Decision Tree - Random Forest	- Struktur kalimat - Pola kata - Bahasa *clickbait* - N-*gram*	-
3.	(Cao et al., 2017)	- Logistic Regression - Random Forest Regression - Linear Regression - Random Forest Classifier	- *Post text related* - *Target content related* - *Relation between post text and target content*	-
4.	(Pandey and Kaur, 2018)	- Support Vector Machine - Logistic Regression - Random Forest - Multilayer Perceptron	- Komposisi kalimat - Struktur kata - Analisis bahasa yang digunakan - Leksikal - Word Embedding	- Word2Vec - GloVe

5.	(Yavi, 2018)	- Naive Bayes	- Kata yang sering muncul	-
6.	(Wongsap et al., 2018)	- Support Vector Machine - Decission Tree - Naive Bayes	- Kata yang sering muncul - N-*gram*	-
7.	(Kumar et al., 2017)	- Proposed Hybrid Approach - BiLSTM	- Distributed word embedding - Character level word embedding - Document embedding - Pre-trained CNN features	- Word2Vec (CBOW) - Doc2Vec
8.	(Shu et al., 2018)	- Logistic Regression - Decission Tree - Random Forest - XGBoost - AdaBoost - Support Vector Machine - GradBoost	- *Readability score* - *Word dictionary matching* - N-*gram* - *Part-of-speech tags*	-
9.	(Dong et al., 2019)	- Recurrent Neural Network (BiGRU)	- *Latent representation* - *Similarity*	-

BAB 3
METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Pada bagian ini akan menjelaskan mengenai tahapan-tahapan yang akan dilakukan dalam penelitian seperti ditunjukkan pada Gambar 10. Pada proses awal dilakukan identifikasi masalah untuk menentukan perumusan masalah, penetapan tujuan, batasan masalah dan kontribusi dari penelitian yang akan dilakukan. Setelah itu, dilakukan studi literatur dengan mengumpulkan dan mempelajari penelitian terdahulu dan dasar teori yang terkait dengan permasalahan yang akan dibahas pada penelitian ini. Kemudian dilanjutkan dengan pengumpulan data. Pada pengumpulan data, data diambil dari beberapa situs berita daring yang ada di Indonesia. Kemudian dari data yang diperoleh selanjutnya data diolah atau dibersihkan pada tahap *data prerpocessing*. Selanjutnya dari data yang sudah dibersihkan tersebut masuk kedalam proses *word embedding* dengan menggunakan Word2Vec. Setelah itu data akan terlebih dahulu diklasifikasikan secara manual antara judul berita *clickbait* dan *non-clickbait* yang kemudian digunakan untuk pelatihan dengan menggunakan SVM. Dari pelatihan yang telah dilakukan selanjutnya akan dilakukan pengujian dari hasil yang diperoleh. Pada proses selanjutnya dilakukan analisis pada pola judul berita yang diperoleh untuk mendapatkan kesimpulan.



Gambar 10. Tahapan Penelitian

3.1.1. Identifikasi Masalah

Pada identifikasi masalah dilakukan perumusan mengenai pertanyaan yang akan diselesaikan pada penelitian ini yakni mengenai deteksi *clickbait*. Kemudian dilakukan penetapan tujuan, batasan masalah, serta kontribusi dari penelitian yang sudah dijelaskan pada bab sebelumnya. Saat ini fenomena mengenai penggunaan judul berita *clickbait* cukup banyak digunakan oleh beberapa media daring di Indonesia untuk menarik pengguna internet mengunjungi situs dari judul berita tersebut. Judul merupakan salah satu hal yang krusial pada artikel berita, karena dari judul seseorang akan membangun persepsi terhadap suatu berita sebelum membaca secara keseluruhan dari isi berita. Berdasarkan permasalahan yang sudah didefinisikan, tujuan dari penelitian ini adalah untuk melakukan deteksi pada judul berita bahasa Indonesia yang terindikasi *clickbait* dengan menggunakan pendekatan *machine learning* menggunakan algoritma SVM dengan penggunaan *word embedding* menggunakan Word2Vec dan *graph embedding* menggunakan Node2Vec sebagai fitur. Selain itu, hasil dari *graph embedding* juga digunakan untuk menganalisis hasil dari klasifikasi sehingga dapat ditentukan berita seperti apa yang biasa dijadikan bahan untuk *clickbait*.

3.1.2. Studi Literatur

Pada tahap ini dilakukan pengkajian dari berbagai literatur seperti jurnal dari penelitian-penelitian sebelumnya, buku, ataupun dari media lain. Studi literatur ini bertujuan untuk menyusun dasar teori terkait yang akan digunakan sebagai pedoman dalam melakukan penelitian. Studi literatur terkait dengan deteksi *clickbait* sudah dijabarkan pada bab sebelumnya. Pada bab 2 telah dijabarkan mengenai pengertian dari *clickbait*, dasar teori dari algoritma SVM, Word2Vec, dan Node2Vec.

3.1.3. Pengumpulan Data

Pada penelitian ini untuk artikel berita *clickbait* akan diambil dari beberapa situs portal berita daring yang diambil berdasarkan rangking 20 situs di Indonesia

yang memiliki jumlah pengunjung dan total *page views* terbanyak oleh Alexa (Hootsuit and We Are Social, 2019) seperti yang tertera pada Tabel 3.

Tabel 3. Sumber Judul Berita

No.	Nama Situs Berita	Alamat Situs Berita
1.	Tribun News	http://www.tribunnews.com/
2.	Detik Com	https://www.detik.com/
3.	Liputan 6	https://www.liputan6.com/
4.	Kompas	https://www.kompas.com/
5.	Okezone	https://www.okezone.com/
6.	Merdeka	https://www.merdeka.com/
7.	CNN Indonesia	https://www.cnnindonesia.com/

Data-data yang diambil dari berbagai situs berita tersebut meliputi: judul, *tag*, kategori, dan url berita dengan format data csv seperti contoh pada Tabel 4. Pada proses pengumpulan data ini bahasa pemrograman yang digunakan adalah Python dengan menggunakan *library* Scrapy. Scrapy merupakan sebuah *framework* aplikasi yang digunakan untuk melakukan *crawling* dan ekstraksi struktur data pada suatu situs yang dapat digunakan lebih lanjut untuk *data mining*, pemrosesan informasi, dan lain-lain.

Tabel 4. Contoh Hasil Pengumpulan Data

id	judul	tag	kategori	url
1.	Seorang Kakek di Dumai Ditemukan Meninggal di Teras Rumahnya	Meninggal Dunia,Dumai,Polsek Bukit Kapur	Home,Regional,Sumatera	https://www.tribunnews.com/regional/2019/01/01/seorang-kakek-di-dumai-ditemukan-meninggal-di-teras-rumahnya
2.	Wiranto Sebut TNI Akan Kirim Pesawat Hercules Jemput Mahasiswa Papua	Wiranto,Pesawat Hercules,mahasiswa Papua	Home,News,Peristiwa	https://www.liputan6.com/news/read/4058533/wiranto-sebut-tni-akan-kirim-pesawat-hercules-

	jemput-mahasiswa-papua

Dari kumpulan data yang telah diperoleh, dilakukan pelabelan secara manual untuk memberikan label antara judul berita *clickbait* dan non-*clickbait*.

3.1.4. Data *Preprocessing*

Pada tahap ini dilakukan proses pembersihan data yang telah diambil pada tahap pengumpulan data sebelumnya yang diambil dari beberapa situs media daring. Tahap awal dari *preprocesing* adalah mengecek apakah terdapat data yang bernilai null, kosong ataupun hilang yang kemudian data tersebut perlu untuk dihapus. Selanjutnya adalah menghilangkan *noise* atau karakter yang bukan alfabet dan melakukan proses *stemming*. Kemudian pada tahap akhir dilakukan pengecekan apabila terdapat data yang terduplikasi dan menghapus data tersebut.

3.1.5. *Word Embedding*

Setelah melakukan tahap *preprocessing* maka selanjutnya adalah melakukan pembelajaran pada model Word2Vec. Berikut merupakan proses yang dilakukan:

a. **Memproses file ke bentuk *list***, yakni dengan mempersiapkan data yang sudah dibersihkan dalam bentuk *list* yang kemudian akan digunakan oleh model dari Word2Vec.
b. **Pembelajaran model Word2Vec**, pada proses ini dibangun kosa kata dari urutan kalimat yang disediakan dan melakukan inisialisasi model. Pada tahap ini model akan mempelajari bobot dari setiap kata yang akan direpresentasikan sebagai vektor.
c. **Menjalankan model dan melakukan pengujian performa**, setelah proses pembelajaran selesai maka model sudah siap untuk digunakan untuk proses selanjutnya yakni digunakan pada klasifikasi *machine learning*. Untuk pengujian dapat dilakukan dengan mengecek kata yang memiliki kesamaan hubungan.

3.1.6. Proses Klasifikasi

Dari proses sebelumnya sudah diperoleh fitur-fitur yang akan digunakan sebagai masukan pada proses selanjutnya yaitu klasifikasi antara berita *clickbait* dan non-*clickbait* menggunakan algoritma SVM. Pada penelitian yang dilakukan oleh Chakraborty et al., menunjukkan bahwa algoritma SVM mendapatkan nilai akurasi paling tinggi dibandingkan dengan algoritma Decision Tree dan Random Forest (Chakraborty et al., 2016). Dari beberapa penelitian mengenai deteksi *clickbait* yang dilakukan oleh (Chakraborty et al., 2016), (Fu et al., 2017), dan (Pandey and Kaur, 2018) algoritma SVM mampu menunjukkan performa akurasi yang cukup tinggi.

Berikut merupakan tahap-tahap dari klasifikasi dengan menggunakan SVM (Gholami and Fakhari, 2017):

a. Persiapan pola matriks

   Untuk persiapan pola matriks pada klasifikasi digunakan satu set data yang dibagi menjadi dua kelas yakni artikel *clickbait* dan non-*clickbait*. Untuk artikel *clickbait* diberi label 1 dan untuk non-*clickbait* diberi label -1. Kemudian dari data yang sudah dilabeli, dilakukan pembagian menjadi data pelatihan, pengujian, dan validasi dengan perbandingan 40:40:20.

b. Pemilihan fungsi kernel

   Pada penelitian ini fungsi kernel yang digunakan adalah kernel linear. Berdasarkan penelitian yang dilakukan oleh Hsu et al, menyatakan bahwa penggunaan kernel linear lebih cepat dibandingkan dengan kernel lainnya dalam melakukan klasifikasi teks (Hsu, Chang and Lin, 2008).

c. Pemilihan parameter

   Beberapa parameter yang dapat dipilih supaya memperoleh kinerja terbaik yang disarankan oleh perangkat lunak terkenal seperti Weka meliputi: (1) parameter yang terdapat pada fungsi kernel, (2) *the trade-off parameter C*, dan (3) *the ε-insensitivity parameter*.

d. Eksekusi algoritma pelatihan

Pada proses ini dilakukan penerapan algoritma sesuai yang telah didefinisikan pada bab sebelumnya.

e. Klasifikasi

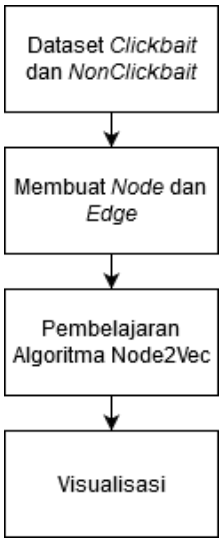
Pada tahap ini dilakukan klasifikasi dan melihat apakah hasil dari proses klasifikasi sudah sesuai. Apabila proses klasifikasi belum sesuai dapat dilakukan pengulangan hingga memperoleh nilai performa yang maksimal.

3.1.7. Pengukuran Kinerja

Selanjutnya, dilakukan pengukuran kinerja dari algoritma klasifikasi untuk penilaian terhadap performa dari algoritma klasifikasi dengan menggunakan *confusion matrix* yang meliputi *precision*, *recall*, dan *f-measure*. Selain itu, pada penelitian ini juga dilakukan pengujian untuk mengetahui peran dari penggunaan *word2vec* sebagai fitur pada algoritma klasifikasi. Cara yang dilakukan pada pengujian ini adalah dengan membandingkan antara penggunaan *word2vec* sebagai fitur dan tidak menggunakan *word2vec* sebagai fitur. Dengan dilakukannya pengujian ini diharapkan dapat diketahui pengaruh dari penggunaan *word2vec* pada performa dari kinerja algoritma klasifikasi.

3.1.8. Analisis Judul Berita *Clickbait*

Pada tahap ini dilakukan analisis pada artikel yang terindikasi *clickbait* dengan menggunakan Node2Vec. Tiap artikel akan direpresentasikan dalam bentuk *node* dan keterkaitannya dengan artikel lain yang sejenis, sehingga pada tahap ini akan terbentuk kelompok-kelompok dari masing-masing *node* yang memiliki keterkaitan. Node-node yang akan didefinisikan pada proses ini terdiri dari: berita yang memiliki atribut katasifat, tagberita, kategoriberita, dan situsberita. Untuk kategori katasifat, diambil dengan melakukan proses *Part-of Speech* (POS) *Tagging*. Pos *tagging* merupakan suatu proses pemberian label pada tiap kata pada suatu kalimat sesuai dengan tag yang ada dalam kategori tata bahasa seperti kata kerja, kata benda, kata sifat, kata keterangan, dan lain-lain.



Gambar 11. Proses *Graph Embedding* dan Visualisasi

Hasil dari pemodelan ini adalah berupa vektor yang merepresentasikan hubungan antar artikel berita yang memiliki keterkaitan yang nantinya dapat divisualisasikan. Analisis ini dilakukan untuk mengetahui topik-topik terkait berdasarkan tag dan kategori berita yang sering digunakan oleh media daring untuk membuat berita *clickbait*. Pada Gambar 11 menunjukkan mengenai tahap-tahap dari proses membuat *graph embedding* yang akan dilakukan.

1. Dataset *clickbait*

   Pada tahap ini, data yang digunakan merupakan kumpulan dari judul berita *clickbait* yang terdiri dari idberita, judulberita, tagberita, kategoriberita, dan situsberita.

2. Membuat *node* dan *edge*

   Pada proses ini dilakukan pembuatan node-node seperti pada Gambar 12, yang terdiri dari:

   - Node berita;
   - Node katasifat: kata sifat diambil dari suku kata yang bersifat adjektiva;
   - Node tagberita;
   - Node kategori;
   - Node situs

3. Pembelajaran algoritma Node2Vec

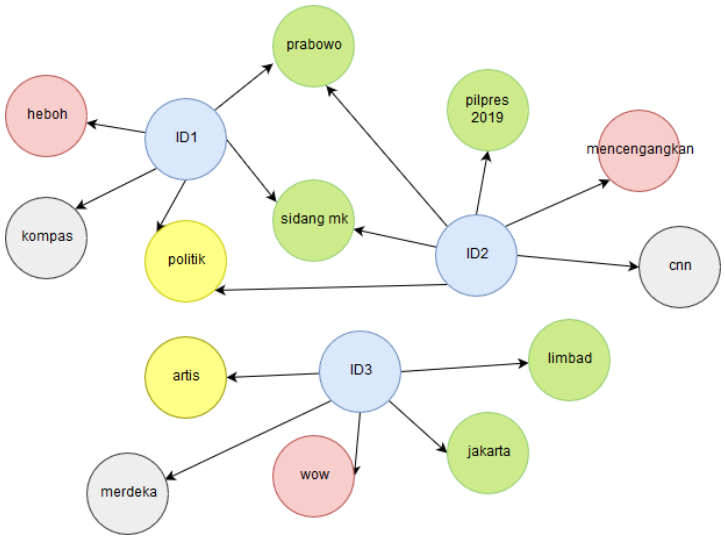
   Pada tahap ini dilakukan eksekusi dari algoritma Node2Vec dengan menggunakan bahasa pemrograman Python.

4. Visualisasi

   Pada tahap ini akan dibuat visualisasi dari hasil *graph embedding* yang diperoleh dari pembelajaran algoritma dengan menggunakan t-SNE.

Tabel 5. Contoh Dataset untuk *Graph Embedding*

idberita	judulberita	katasifat	tagberita	kategori	situs
ID1	heboh video faldo maldini "prabowo tak akan menang pemilu di mk", ini kata bima arya	heboh	prabowo, sidang mk	politik	kompas
ID2	tim prabowo klaim bakal hadirkan saksi mencengangkan di mk	mencengangkan	prabowo, sidang mk, pilpres 2019	politik	cnn
ID3	wow, ternyata limbad miliki gelar profesor dan kuasai 3 bahasa lho!	wow	limbad, jakarta	artis	merdeka



Gambar 12. Contoh *node* dan *edge* yang terbentuk

Keterangan:

Biru : berita

Merah : katasifat

Hijau : tagberita

Kuning : kategori

Abu-abu : situs

3.1.9. Analisis dan Kesimpulan

Tahap akhir pada penelitian ini adalah dengan melakukan analisis dari hasil penelitian yang telah dilakukan dengan mengetahui bagaimana kinerja dari metode yang telah diusulkan dan analisis yang dilakukan pada judul berita *clickbait* dan *nonclickbait*.

(Halaman sengaja dikosongkan)

BAB 4
HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Pada bagian ini akan dijelaskan mengenai hasil yang diperoleh dari penelitian yang dilakukan untuk analisis judul berita *clickbait* dengan menggunakan *word embedding* sebagai fitur untuk klasifikasi dengan menggunakan algoritma SVM. Selain itu juga dilakukan analisis dengan menggunakan Node2Vec.

4.1.1. Pengumpulan Data

Langkah awal yang dilakukan pada penelitian ini adalah dengan melakukan pengumpulan data dari 7 situs berita di Indonesia. Data yang diambil merupakan berita yang diterbitkan mulai dari bulan Januari 2019 sampai dengan September 2019. Pengambilan data dilakukan dengan menggunakan *framework* Scrapy dengan menelusuri berita yang terdapat pada halaman indeks tiap situs. Pada pengambilan data ini, diperolah data sebanyak 49.185 judul berita dan disimpan dalam bentuk csv dengan detail seperti pada Tabel 6. Informasi yang disimpan pada tiap judul berita terdiri dari:

- Judul : merupakan judul berita
- Tag : merupakan tag yang ada pada isi berita
- Kategori : merupakan kategori atau sub-menu dari berita
- url : pranala berita

Tabel 6. Jumlah Data Per Situs

No.	Nama Situs	Indeks URL	Jumlah Data
1.	Tribun News	https://www.tribunnews.com/indeks	4.744
2.	Detik Com	https://news.detik.com/indeks	856
3.	Liputan 6	https://www.liputan6.com/indeks	26.179
4.	Kompas	https://indeks.kompas.com/	9.058
5.	Okezone	https://index.okezone.com/	3.795
6.	Merdeka	https://www.merdeka.com/indeks-berita/	2.771

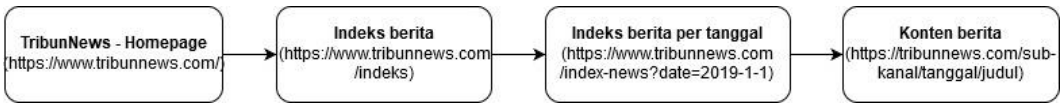
7.	CNN Indonesia	https://www.cnnindonesia.com/indeks	1.782
		Total	**49.185**

Secara umum, halaman indeks memuat daftar dari keseluruhan berita yang telah terbit pada suatu situs dan dapat dilihat berdasarkan tanggal terbit. Selanjutnya, akan dijelaskan lebih lanjut mengenai peta situs untuk memperoleh konten berita. Secara umum, struktur dari peta situs memiliki kesamaan seperti yang ditunjukkan pada Tabel 7.

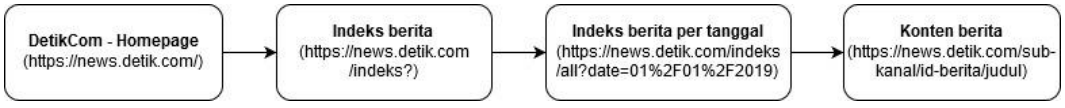
Tabel 7. Peta tautan situs

Nama Situs	Peta Situs
Tribun News	• **Homepage** (https://www.tribunnews.com/) o **Indeks berita** (https://www.tribunnews.com/indeks) ▪ **Indeks berita per tanggal** (https://www.tribunnews.com/index-news?date=2019-1-1) • **Konten Berita** (https://www.tribunnews.com/sub-kanal/tanggal/judul)
Detik Com	• **Homepage** (https://news.detik.com/) o **Indeks berita** (https://news.detik.com/indeks?) ▪ **Indeks berita per tanggal** (https://news.detik.com/indeks/all?date=01%2F01%2F2019) • **Konten Berita** (https://www.tribunnews.com/sub-kanal/id-berita/judul)
Liputan 6	• **Homepage** (https://www.liputan6.com/) o **Indeks berita** (https://www.liputan6.com/news/indeks) ▪ **Indeks berita per tanggal** (https://www.liputan6.com/news/indeks/2019/01/01) • **Konten Berita** (https://www.tribunnews.com/sub-kanal/read/id-berita/judul)
Kompas	• **Homepage** (https://www.kompas.com/) o **Indeks berita** (https://indeks.kompas.com/all/) ▪ **Indeks berita per tanggal** (https://indeks.kompas.com/all/2019-01-01)

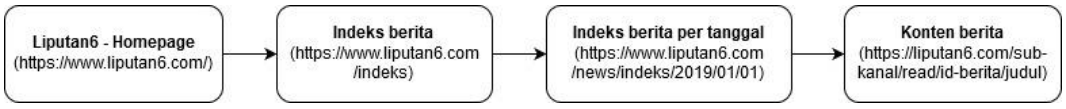
	• **Konten Berita** (https://www.tribunnews.com/sub-kanal/read/tanggal/id-berita/judul)
Okezone	• **Homepage** (https://www.okezone.com/) o **Indeks berita** (https://index.okezone.com/bydate/index) ▪ **Indeks berita per tanggal** (https://index.okezone.com/bydate/index/2019-01-01) • **Konten Berita** (https://www.kanal.okezone.com/read/tanggal/id/id-berita/judul)
Merdeka	• **Homepage** (https://www.merdeka.com/) o **Indeks berita** (https://www.merdeka.com/indeks-berita/) ▪ **Indeks berita per tanggal** (https://www.merdeka.com/indeks-berita/2019/09/16/index.html) • **Konten Berita** (https://www.merdeka.com/sub-kanal/judul)
CNN Indonesia	• **Homepage** (https://www.cnnindonesia.com/) o **Indeks berita** (https://www.cnnindonesia.com/indeks) ▪ **Indeks berita per tanggal** (https://www.cnnindonesia.com/indeks?date=2019/01/01) • **Konten Berita** (https://www.merdeka.com/sub-kanal/id-berita/judul)



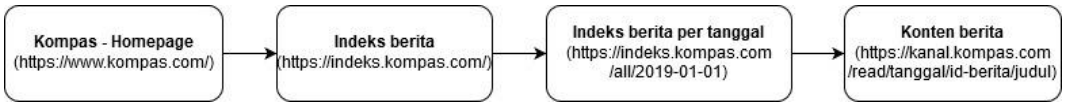
Gambar 13. Peta Situs Tribun News



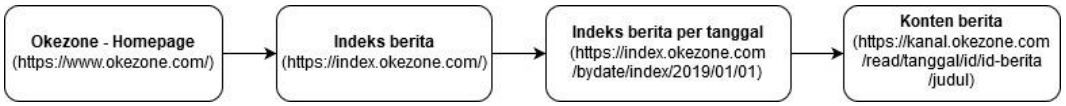
Gambar 14. Peta Situs Detik com



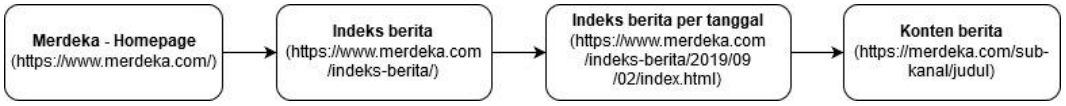
Gambar 15. Peta Situs Liputan6



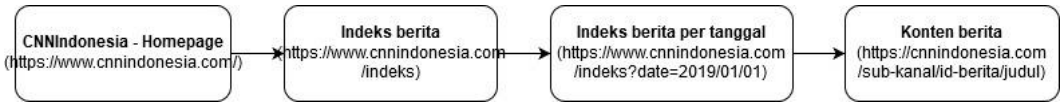
Gambar 16. Peta Situs Kompas



Gambar 17. Peta Situs Okezone



Gambar 18. Peta Situs Merdeka



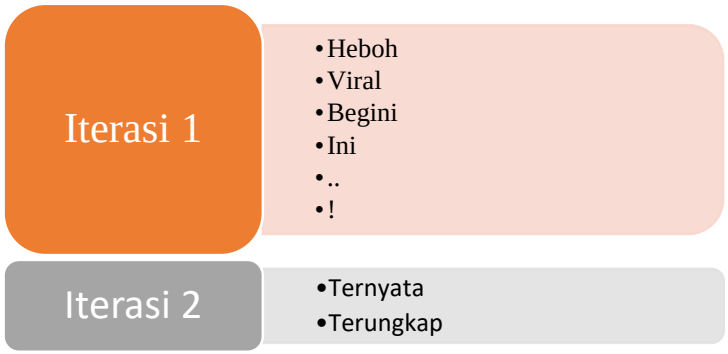
Gambar 19. Peta Situs CNN Indonesia

Setelah diperoleh data yang dibutuhkan, selanjutnya dilakukan pelabelan untuk menentukan judul berita *clickbait*. Cara yang digunakan adalah dengan menentukan kata kunci untuk mengklasifikasi judul berita. Kata kunci yang digunakan diambil berdasarkan dari judul *clickbait* yang memiliki kategori *exaggeration, formatting* dan *ambigous*. Dari kata kunci yang telah ditentukan pada Tabel 8, kemudian dilihat apakah berita yang diperoleh dari kata kunci tersebut termasuk dalam kategori *clickbait* atau bukan.

Tabel 8. Kata Kunci Pencarian *Clickbait* Berdasarkan Kategori

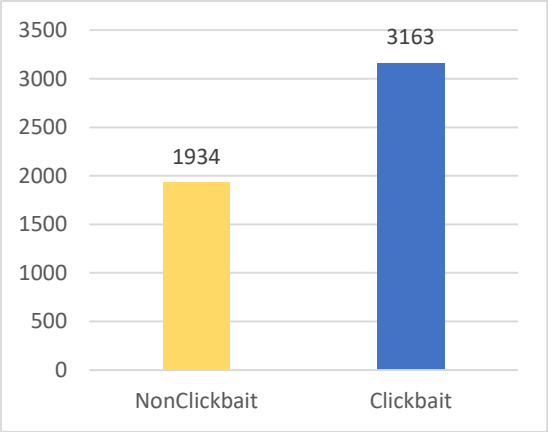
Kategori	Kata Kunci
1. *Exaggerration*	• Heboh • Viral
2. *Formatting*	• ! • ? • ..
3. *Ambigous*	• Begini • Ini

Pada Gambar 20 menunjukkan mengenai iterasi dari proses klasifikasi judul berita *clickbait*. Pada proses iterasi pertama dicari judul berita yang mengandung kata heboh, viral, begini, ini, dan tanda baca. Dari proses tersebut, diperoleh judul berita sebanyak 6586 judul berita. Selanjutnya pada iterasi kedua digunakan kata kunci ternyata dan terungkap. Sehingga dari proses tersebut diperoleh judul berita sebanyak 5717 judul berita. Dari judul yang diperoleh sesuai dengan kata kunci tersebut, dipilih judul berita yang termasuk ke dalam *clickbait* dengan pelabelan yang dilakukan secara manual oleh dua orang.



Gambar 20. Iterasi Klasifikasi Judul *Clickbait*

Dari hasil pelabelan yang dilakukan, diperoleh data sebanyak 1103 judul *clickbait*. Kemudian diambil secara acak 1103 judul berita yang diberi label *nonclickbait*. Berdasarkan penelitian yang dilakukan oleh (Chakraborty et al., 2016) menyatakan bahwa judul berita *clickbait* cenderung menggunakan banyak kata *stopword* pada kalimatnya. Sehingga setelah dilakukan pelabelan secara manual, dihitung perbandingan jumlah kata *stopword* antara judul dengan label *clickbait* dan *nonclickbait* untuk mengetahui jumlah penggunaan *stopword* pada masing-masing label tersebut. Pada judul berita *clickbait*, penggunaan *stopword* lebih banyakdengan jumlah total 3163 kata dibandingkan dengan judul *nonclickbait* dengan jumlah total 1934 seperti grafik yang ditampilkan pada Gambar 21, sehingga untuk proses klasifikasi akan dibandingkan antara penggunaan dataset yang mengandung *stopword* dan yang tidak mengandung *stopword*.



Gambar 21. Grafik Perbandingan Jumlah Penggunaan *Stopword*

4.1.2. *Data Preprocessing*

Pada proses ini, data yang diperoleh dari pengumpulan data dibersihkan. Proses pembersihan dilakukan dengan menghapus baris-baris yang memiliki nilai tidak lengkap atau kosong. Penghapusan pada kolom yang memiliki simbol ataupun karakter yang tidak diperlukan seperti contoh pada Tabel 9.

Tabel 9. Contoh Hasil Penghapusan Karakter yang Tidak Diperlukan

1.	Sebelum	\n Mendagri Nilai Isu Surat Suara Tercoblos Didramatisasi \n
	Sesudah	Mendagri Nilai Isu Surat Suara Tercoblos Didramatisasi
2.	Sebelum	Salah Satu Misteri Ã¢Â□ÂœMoaiÃ¢Â□Â□ di Pulau Paskah Akhirnya Terpecahkan
	Sesudah	Salah Satu Misteri Moai di Pulau Paskah Akhirnya Terpecahkan

Selain itu juga dilakukan proses *stemming*, yakni mentransformasikan kata menjadi kata dasarnya seperti pada Tabel 10. Proses *stemming* ini merupakan salah satu proses penting pada pembersihan data untuk penggunaan Word2Vec, karena *embedding* tersebut disusun berdasarkan kata. Sehingga penggunaan kata dasar akan lebih memperkaya hasil dari *word embedding*.

Tabel 10. Contoh Hasil *Stemming*

Sebelum	lulusan smk di pangandaran ini disebar jadi agen perdamaian
Sesudah	lulus smk di pangandaran ini sebar jadi agen damai

Dari dataset yang diperoleh, satu judul berita memiliki beberapa kategori berita karena data diambil dari *breadcrumbs*. *Breadcrumbs* merupakan suatu navigasi pada situs yang menunjukkan informasi mengenai posisi halaman dimana pengguna saat ini. Sehingga dari beberapa kategori yang diperoleh tersebut dipilih kategori yang paling khusus seperti contoh pada Tabel 11.

Tabel 11. Pemilihan Kategori Berita

Breadcrumbs	**Kategori**
Home,News,Peristiwa	Peristiwa
Home,Nasional,Berita Politik	Politik
detikNews,Berita-jawa-timur,Detail Berita	Jawa Timur

Beberapa dari kategori berita memiliki perbedaan penamaan pada setiap situs, tetapi memiliki konteks yang sama. Sehingga beberapa nama kategori disamakan sesuai dengan konteksnya.

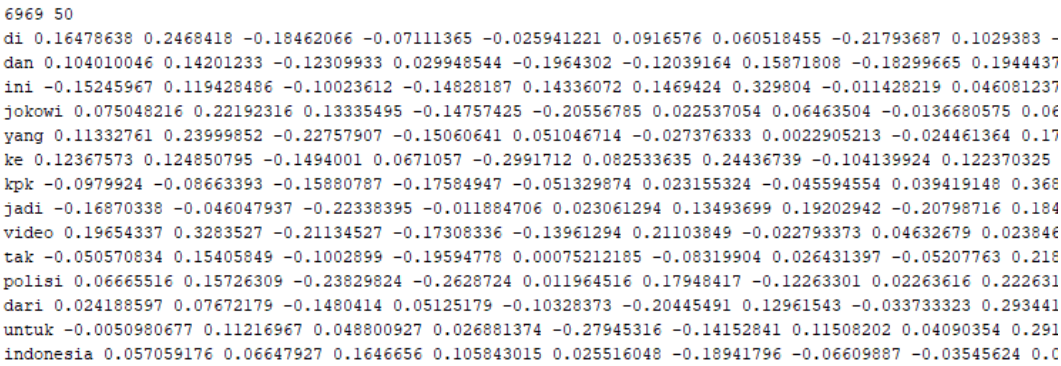
4.1.3. *Word Embedding*

Word embedding merupakan tahap untuk merepresentasikan suatu kata menjadi bentuk vektor. Proses ini menggunakan *library* Gensim yang di dalamnya sudah terdapat *package* Word2Vec. Secara sederhana Word2Vec bekerja dengan mengambil korpus teks sebagai masukan dan menghasilkan vektor sebagai keluaran. Proses yang dilakukan oleh Word2Vec adalah dengan membangun kosakata dari data teks pelatihan dan selanjutnya mempelajari representasi vektor dari kata-kata tersebut. Data yang digunakan pada tahap ini merupakan kumpulan dari keseluruhan judul berita yang telah dibersihkan yakni 49.185 baris judul yang disimpan dalam bentuk csv. Selanjutnya, yang dilakukan pada proses ini adalah dengan memecah tiap baris kalimat menjadi per suku kata dan menyimpannya ke dalam bentuk *list*. Dari proses tersebut, dihitung 10 kata yang sering muncul seperti yang ada pada Tabel 12.

Tabel 12. Kata Yang Sering Muncul Pada Keseluruhan Data

Kata yang sering muncul	Jumlah Kata	Kata yang sering muncul (tanpa stopword)	Jumlah Kata
di	14560	jokowi	3605
dan	5097	kpk	2821
jokowi	3605	video	2568
ini	3580	polisi	2331
yang	3238	indonesia	1741
ke	3211	jakarta	1481
kpk	2821	warga	1320
video	2568	prabowo	1229
tak	2387	rumah	1023
polisi	2331	kota	933

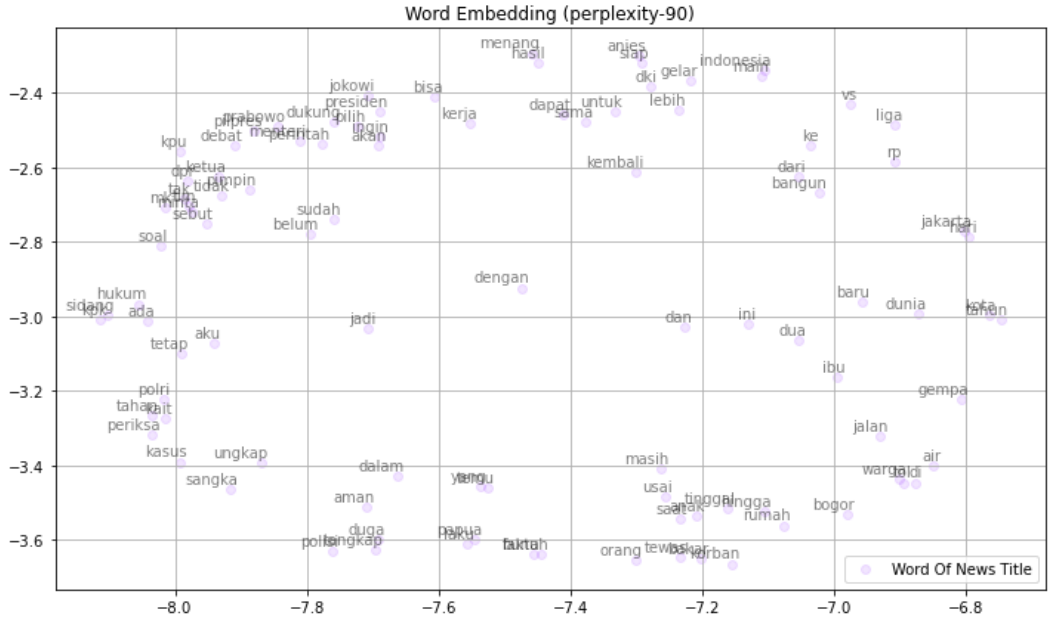
Dari data yang diperoleh, selanjutnya digunakan sebagai data *training* untuk membangun model *word embedding* dengan ukuran dimensi. Pada Gambar 22 merupakan hasil dari pembangunan model menggunakan Word2Vec. Dari gambar tersebut dapat dilihat bahwa pada setiap kata memiliki nilai vektor yang berjumlah 50.



Gambar 22. Hasil Model *Word Embedding*

Dari *word embedding* yang dihasilkan dibuat visualisasi menggunakan T-SNE dengan mengambil 100 kata yang paling sering muncul seperti pada Gambar 23. Pada gambar tersebut dapat dilihat bahwa beberapa kata yang saling berkaitan atau memiliki konteks yang sama berada pada titik yang berdekatan. Contohnya adalah

apabila kita melihat titik pada kata ‘jokowi’ dan titik-titik disekitarnya seperti ‘presiden’, ‘prabowo’, ‘menteri’, ‘pilpres’ masih memiliki hubungan dan konteks antar katanya.



Gambar 23. Visualisasi Kata yang Sering Muncul dari *Word Embedding*

4.1.4. Proses Klasifikasi

Pada tahap klasifikasi judul berita dengan menggunakan SVM, akan dilakukan tiga percobaan klasifikasi dengan menggunakan fitur yang berbeda seperti pada Tabel 13. Fitur yang pertama adalah vektor kata dari judul berita menggunakan *word embedding*. Fitur yang kedua adalah dari judul berita dengan menggunakan *bag of word*. Sedangkan untuk fitur yang ketiga adalah dengan menggunakan *graph embedding*. Pada Percobaan pertama dengan menggunakan fitur *word embedding*, dataset yang digunakan terbagi menjadi dua yakni dataset judul berita yang mengandung *stopword* yang selanjutnya akan disebut dengan DS1 dan dataset judul berita yang tidak mengandung *stopword* yang selanjutnya akan disebut dengan DS2. Hal ini dilakukan untuk mengetahui pengaruh dari penggunaan *stopword* terhadap hasil evaluasi dari SVM. Percobaan kedua menggunakan fitur *bag of word*. Sedangkan untuk percobaan yang ketiga menggunakan fitur dari *graph embedding*.

Tabel 13. Daftar Percobaan Klasifikasi yang Dilakukan

Klasifikasi	**Fitur**	**Dataset**
Percobaan Pertama	*Word Embedding*	DS1
		DS2
Percobaan Kedua	*Bag Of Word*	DS1
Percobaan Ketiga	*Graph Embedding*	Judul berita, Kata Sifat, Tag, Kategori, Situs

Total data yang digunakan pada penelitian ini adalah 1103 judul *clickbait* dan 1103 judul *nonclickbait*. Dari dataset tersebut kemudian dibagi menjadi dua bagian yakni 80% digunakan untuk *data training classifier* dan 20% lainnya digunakan sebagai *data testing classifier*. Klasifikasi SVM dilakukan dengan menggunakan *module* sklearn yang disediakan oleh Python. Untuk membagi dataset digunakan fungsi yang ada pada *module* sklearn yakni `train_test_split`. Fungsi tersebut akan membagi dataset secara random menjadi subset *train* dan *test*. Setelah melakukan pemrosesan dataset, selanjutnya dilakukan percobaan klasifikasi sesuai dengan Tabel 13. Klasifikasi menggunakan SVM dilakukan dengan menggunakan 3 parameter kernel yakni Linear, RBF, dan Poly. Sedangkan parameter lain yang digunakan adalah `C` dan `Gamma`. Parameter `C` merupakan parameter untuk menentukan regulasi atau pemisah kelas dalam *classifier*. Semakin tinggi nilai `C` maka jarak antara *hyperplane* dan kelas akan semakin kecil dan dapat menyebabkan *overfitting*. Sebaliknya, apabila nilai `C` semakin rendah, maka dapat menyebabkan *underfitting* pada modelnya. Pada penelitian ini nilai C yang digunakan adalah 10.0, 20.0, dan 23.0. Parameter `Gamma` merupakan parameter dari kernel RBF yang menentukan sejauh mana pengaruh dari satu *training sample*. Parameter lain yang digunakan pada penelitian ini menggunakan nilai *default* seperti parameter *degree*, coef0, *shrinking*, *probability*, tol, max_iter, decision_function_shape, dan random_state. Parameter degree merupakan parameter yang digunakan pada kernel poly dengan nilai *default*=3. Parameter coef0 merupakan parameter untuk kernel poly dengan nilai *default*=0.0. Parameter *shrinking* merupakan parameter yang berfungsi untuk mengidentifikasi

dan menghapus elemen-elemen yang dibatasi untuk mengatasi masalah optimasi. Nilai *default* dari parameter *shrinking* adalah *True*. Untuk parameter *probability* memiliki nilai *default = False*. Parameter tol merupakan parameter untuk mengatur tingkat toleransi untuk menghentikan suatu kriteria dengan nilai *default* = 1e-3. Parameter max_iter merupakan parameter untuk jumlah maksimal iterasi dari penyelesaian masalah pada SVM dengan nilai *default* = -1 yang berarti *no limit*. Parameter decision_function_shape memiliki nilai *default* = "ovr" atau one-vs-rest *classifier* yang dirancang untuk memodelkan setiap kelas terhadap semua kelas lainnya secara independen. Parameter random_state merupakan dasar dari *pseudo random number generator* yang digunakan ketika mengacak data untuk estimasi probabilitas. Nilai *default* dari parameter tersebut adalah *None* yang berarti *random number generator* merupakan *instance* dari RandomState yang digunakan oleh *numpy.random*.

a. Percobaan Pertama

Pada percobaan pertama, klasifikasi dilakukan dengan menggunakan fitur *word embedding* hasil olahan dari Word2Vec. Untuk perbandingan dari perbedaan dataset yang digunakan antara dataset yang mengandung *stopwords* atau DS1 dan yang tidak mengandung *stopwords* atau DS2 dapat dilihat pada Tabel 14.

Tabel 14. Perbedaan Dataset Yang Digunakan

DS1	**DS2**
jadi panglima wilayah timur **ini** strategi soekarwo menang demokrat	panglima wilayah timur strategi soekarwo menang demokrat
lagi asyik isap sabu residivis begal sadis tembak mati polisi	asyik isap sabu residivis begal sadis tembak mati polisi
ini pria **yang ingin** temu nick furry **belum** hilang **di akhir** avengers infinity war	pria temu nick furry hilang avengers infinity war

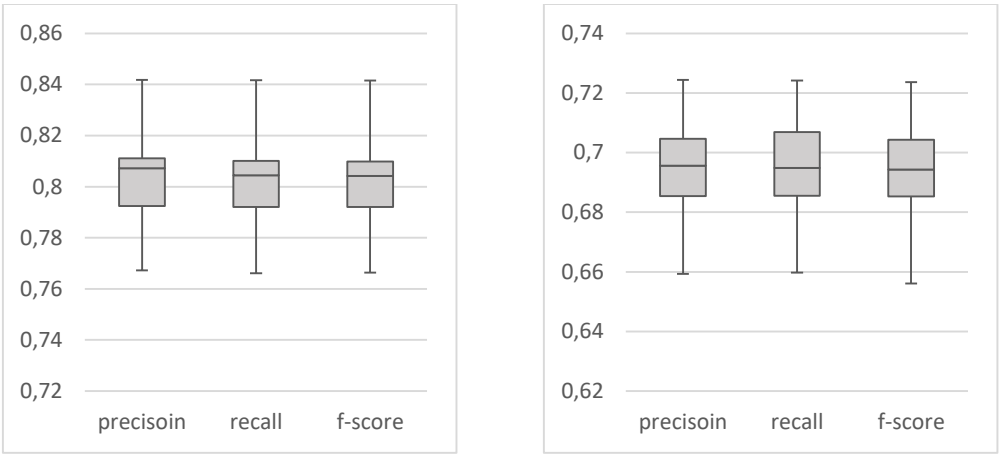
Dari percobaan klasifikasi yang telah dilakukan dengan menggunakan kedua dataset pada DS1 dan DS2, diperoleh hasil dari evaluasi yang dilakukan pada beberapa parameter yang berbeda dapat dilihat pada Tabel 15.

Tabel 15. Hasil Evaluasi Percobaan Pertama

Parameter	DS1			DS2		
	Precision	Recall	F-Score	Precision	Recall	F-Score
Kernel: Linear C=10.0	0.785933	0.784863	0.784371	0.678549	0.677793	0.676892
Kernel: Linear C=20.0	0.76839	0.767127	0.766312	0.687493	0.686775	0.685904
Kernel: Linear C=23.0	0.787153	0.78585	0.785123	0.678562	0.677996	0.677335
Kernel: RBF C=10.0	0.798711	0.798346	0.797877	0.684029	0.683913	0.683171
Kernel: RBF C=20.0	0.798582	0.798085	0.797446	0.68962	0.689391	0.688745
Kernel: RBF C=23.0 Gamma='scale'	**0.803929**	**0.803206**	**0.802723**	**0.694662**	**0.694674**	**0.693886**
Kernel: Poly C=10.0	0.738226	0.737616	0.736804	0.688878	0.688698	0.687799
Kernel: Poly C=20.0	0.794947	0.794553	0.794004	0.686331	0.686278	0.685457
Kernel: Poly C=23.0	0.798503	0.798208	0.797652	0.684029	0.683913	0.683171

Percobaan pengujian dilakukan sebanyak 50 kali iterasi yang kemudian diambil rata-rata seperti yang dituliskan pada Tabel 15. Berdasarkan dari hasil percobaan tersebut, kombinasi parameter yang menghasilkan rata-rata nilai evaluasi paling baik adalah dengan menggunakan Kernel RBF dengan C=23.0 dan *gamma='scale'*. Dari nilai tersebut, selanjutnya dibuat diagram boxplot berdasarkan dari hasil pengujian dengan rata-rata nilai paling tinggi seperti pada Gambar 24.

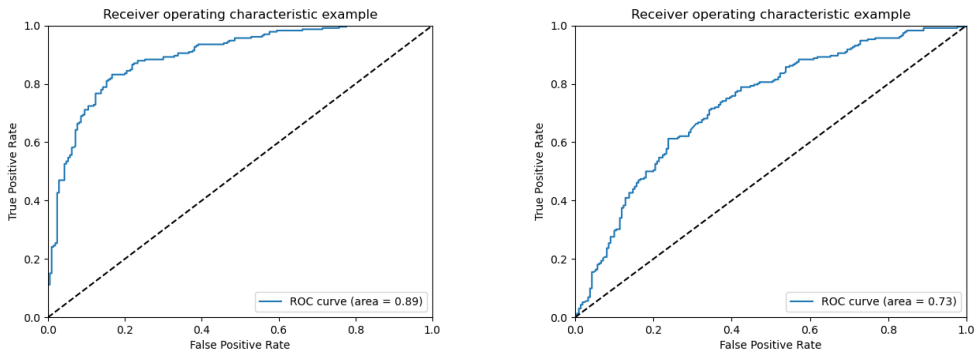
(a) (b)



Gambar 24. Diagram Boxplot Nilai Terbaik Percobaan Pertama. (a) Boxplot dengan dataset DS1. (b) Boxplot dengan dataset DS2.

Pengukuran selanjutnya adalah menghitung nilai ROC dari klasifikasi SVM untuk mengetahui probabilitas dari *classifier* untuk mengklasifikasikan dengan benar data yang *true positive* terhadap data yang diklasifikasikan sebagai *true negative* dengan kurva yang digambarkan seperti pada Gambar 25.

(a) (b)

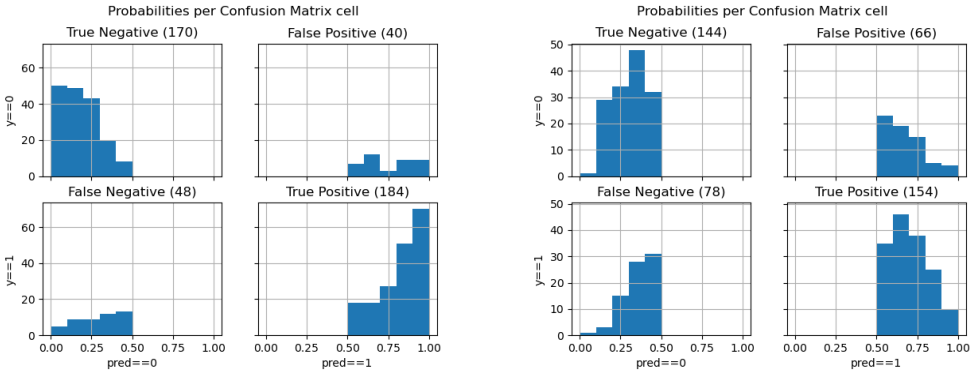


Gambar 25. Kurva ROC Percobaan Pertama; (a) Kurva ROC menggunakan dataset DS1; (b) Kurva ROC menggunakan dataset DS2

Dari gambar grafik tersebut dapat diketahui bahwa nilai probabilitas dari ROC untuk klasifikasi dengan dataset DS1 adalah 0.89 dan nilai ROC untuk klasifikasi dengan dataset DS2 adalah 0.73. Hasil tersebut menunjukkan bahwa probabilitas dari *classifier* untuk mengklasifikasikan dataset secara benar adalah 89% untuk dataset DS1 dan sebesar 73% untuk dataset DS2.

Selanjutnya, dibuat grafik histogram untuk mengetahui distribusi dari data *test* untuk nilai *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) seperti pada Gambar 26. Dari gambar tersebut dapat diketahui bahwa jumlah data *test* yang diklasifikasikan kedalam TP, FP, TN, FN adalah 184, 40, 170, 48 untuk klasifikasi dengan dataset DS1. Sedangkan untuk klasifikasi dengan dataset DS2 jumlah dari TP, FP, TN, FN adalah 154, 66, 144, 78.

(a) (b)



Gambar 26. Histogram *Confussion Matrix* Percobaan Pertama; (a) Klasifikasi dengan dataset DS1; (b) Klasifikasi dengan dataset DS2

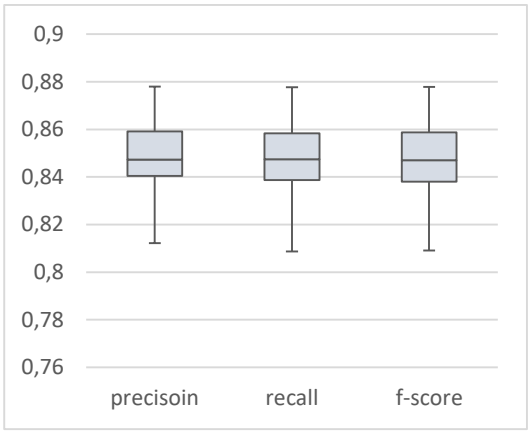
b. Percobaan Kedua

Pada percobaan kedua, fitur yang digunakan adalah *bag of word* dengan menggunakan *library* sklearn yakni `sklearn.feature_extraction.text`. *Class* yang digunakan adalah `CountVectorizer` dan menggunakan beberapa parameter yakni `max_features=1500`, `min_df=5`, dan `max_df=0.7`. Parameter `max_features` merupakan parameter yang membatasi jumlah fitur yang berarti membatasi penggunaan kata yang paling banyak muncul. Parameter `min_df` merupakan parameter yang menunjukkan nilai minimal kata yang harus muncul pada dokumen untuk dimasukkan ke dalam fitur. Sedangkan `max_df` merupakan parameter yang membatasi nilai maksimum suatu kata muncul pada dokumen, karena ketika suatu kata muncul terlalu sering, maka kata tersebut dianggap kurang berarti. Hasil evaluasi dari percobaan ketiga dapat dilihat pada Tabel 16.

Tabel 16. Hasil Evaluasi Percobaan Kedua

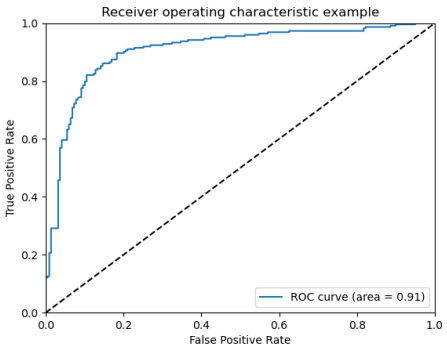
Parameter	Precision	Recall	F-Score
Kernel: Linear C=10.0	0.783987	0.783977	0.783537
Kernel: Linear C=20.0	0.774106	0.773957	0.773551
Kernel: Linear C=23.0	0.772132	0.772203	0.771583
Kernel: RBF C=10.0	**0.848711**	**0.848063**	**0.847791**
Kernel: RBF C=20.0	0.845884	0.84514	0.844745
Kernel: RBF C=23.0 Gamma='scale'	0.843897	0.843132	0.84266
Kernel: Poly C=10.0	0.81304	0.792261	0.788635
Kernel: Poly C=20.0	0.812708	0.792438	0.789019
Kernel: Poly C=23.0	0.810838	0.789378	0.785786

Evaluasi dilakukan sebanyak 50 kali dengan mengambil nilai rata-rata dari 50 iterasi yang telah dilakukan menggunakan masing-masing parameter seperti pada Tabel 16. Dari hasil evaluasi tersebut, nilai rata-rata evaluasi paling baik diperoleh dengan parameter kernel RBF dan C=10.0. Selanjutnya, dibuat diagram boxplot dari pengujian yang dilakukan dengan penggunaan parameter tesebut seperti pada Gambar 27. Berdasarkan hasil evaluasi tersebut dapat dilihat bahwa rata-rata nilai yang diperoleh lebih baik dibandingkan dengan menggunakan fitur *word embedding*.



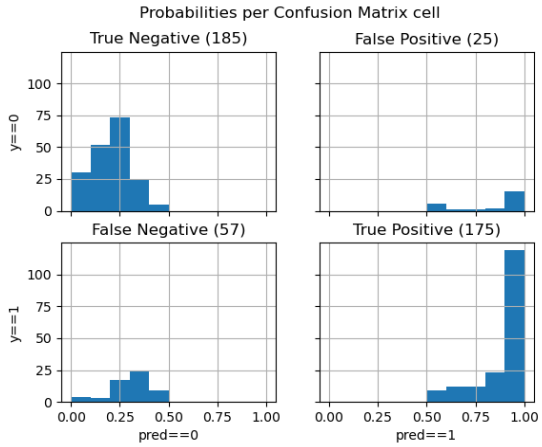
Gambar 27. Diagram Boxplot Nilai Terbaik Percobaan Kedua

Evaluasi selanjutnya adalah penghitungan nilai ROC dan menampilkan kurva yang digambarkan seperti pada Gambar 28. Dari gambar tersebut dapat diketahui bahwa nilai probabilitas dari ROC adalah 0.91. Berdasarkan dari hasil tersebut, berarti probabilitas dari *classifier* SVM dapat mengklasifikasikan dengan benar adalah 91%.



Gambar 28. Kurva ROC Percobaan Kedua

Selanjutnya, untuk histogram dari *confussion matrix* pada percobaan kedua dapat dilihat pada Gambar 29. Berdasarkan gambar tersebut, dapat dilihat bahwa jumlah dari masing-masing nilai untuk TP, FP, TN, FN secara berturut-turut adalah 175, 25, 185, 57. Dari diagram tersebut dapat diketahui bahwa distribusi dari TP dengan nilai prediksi = 1 memiliki jumlah yang cukup tinggi.



Gambar 29. Histogram *Confussion Matrix* Percobaan Ketiga

c. Percobaan Ketiga

Pada percobaan yang ketiga ini, fitur yang digunakan untuk klasifikasi merupakan vektor *graph embedding* dari Node2Vec dengan ukuran dimensi 16. Pada perobaan pertama dan kedua dataset yang digunakan sebagai fitur merupakan judul berita, sedangkan pada percobaan ketiga ini dataset yang digunakan sebagai fitur meliputi judul berita, kata sifat yang diambil dari judul berita, tag berita, kategori dan situs. Pada percobaan klasifikasi ini, dilakukan beberapa klasifikasi dengan kombinasi fitur yang berbeda-beda untuk mengetahui fitur mana yang paling berpengaruh terhadap nilai evaluasi.

1) Fitur Judul dan Kata Sifat

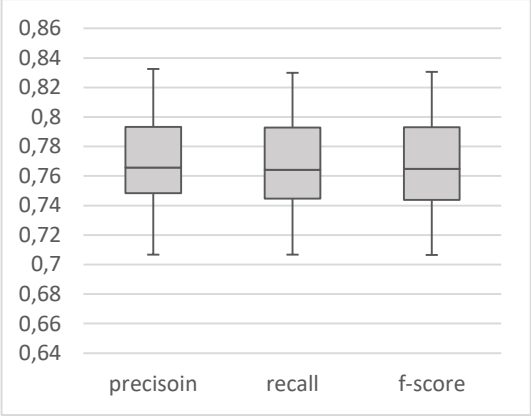
Dari hasil klasifikasi menggunakan fitur judul berita dan kata sifat berikut merupakan hasil evauasi yang diperoleh seperti pada Tabel 17.

Tabel 17. Hasil Evaluasi dengan Fitur Judul dan Kata Sifat

Parameter			**Precision**	**Recall**	**F-Score**
Kernel	**C**	**Gamma**			
Linear	10	-	0.75821	0.752874	0.753042
Linear	20	-	0.753827	0.751398	0.750592
Linear	23	-	0.760577	0.757126	0.757285
RBF	10	Scale	**0.767685**	**0.766046**	**0.765367**

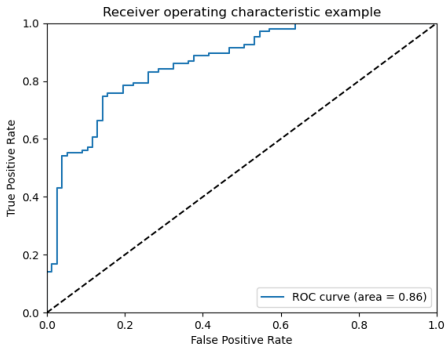
RBF	20	Scale	0.756981	0.755964	0.75564
RBF	23	Scale	0.761392	0.761098	0.760345
Poly	10	-	0.755369	0.754183	0.753705
Poly	20	-	0.747457	0.748207	0.746455
Poly	23	-	0.750105	0.749606	0.748649

Berdasarkan hasil evaluasi pada Tabel 17, nilai rata-rata evaluasi terbaik adalah dengan menggunakan parameter kernel RBF, C=10, dan Gamma=’scale’. Berikut merupakan diagram boxplot dari 50 kali iterasi pengujian yang dilakukan dengan menggunakan parameter tersebut seperti pada Gambar 30.



Gambar 30. Diagram Boxplot Nilai Terbaik Percobaan Ketiga (Fitur Judul dan Kata Sifat)

Kemudian dilakukan penghitungan untuk memperoleh nilai ROC dan dibuat grafik seperti yang ditunjukkan pada Gambar 31.



Gambar 31. Kurva ROC Percobaan Ketiga (Fitur Judul dan Kata Sifat)

Berdasarkan grafik tersebut, nilai ROC yang diperoleh adalah 0.86. Sehingga, dari grafik tersebut diketahui bahwa nilai dari probabilitas *classifier* dapat mengklasifikasi dengan benar adalah 86%.

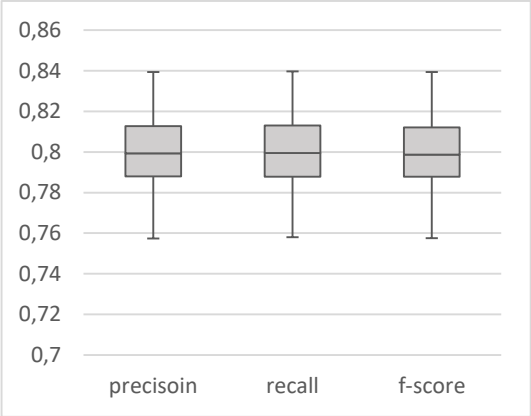
2) Fitur Judul dan Tag Berita

Berikut merupakan hasil dari evaluasi klasifikasi dengan menggunakan fitur judul dan tag berita yang disajikan pada Tabel 18.

Tabel 18. Hasil Evaluasi dengan Fitur Judul dan Tag Berita

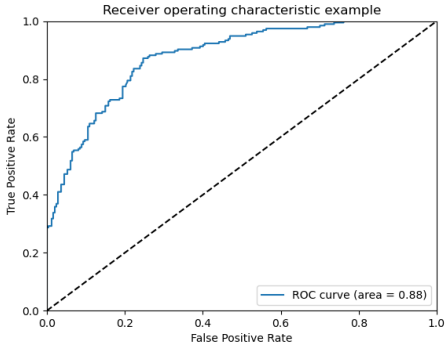
Parameter			**Precision**	**Recall**	**F-Score**
Kernel	**C**	**Gamma**			
Linear	10	-	0.772466	0.772124	0.771411
Linear	20	-	0.77077	0.76997	0.769506
Linear	23	-	0.768756	0.768137	0.767389
RBF	10	Scale	**0.80144**	**0.801236**	**0.800825**
RBF	20	Scale	0.797738	0.7976	0.797191
RBF	23	Scale	0.799586	0.799485	0.798974
Poly	10	-	0.791107	0.790493	0.789756
Poly	20	-	0.791915	0.79171	0.791257
Poly	23	-	0.788255	0.788351	0.787792

Berdasarkan hasil evaluasi pada Tabel 18, nilai rata-rata evaluasi terbaik diperoleh dengan menggunakan kernel RBF, C=10, dan Gamma=’scale’. Hasil dari nilai evaluasi dengan menggunakan fitur judul dan tag berita menunjukkan performa yang lebih baik dengan nilai *precision* 0.80 dibandingkan dengan penggunaan fitur judul dan kata sifat dengan nilai *precision* 0.77. Selanjutnya adalah membuat diagram boxplot dari 50 kali iterasi pengujian yang dilakukan dengan menggunakan parameter yang memperoleh hasil terbaik tersebut yang digambarkan seperti pada Gambar 32.



Gambar 32. Diagram Boxplot Nilai Terbaik Percobaan Ketiga (Fitur Judul dan Tag Berita)

Kemudian dilakukan penghitungan nilai ROC dan dibuat grafik seperti pada Gambar 33. Sehingga, dari grafik tersebut diketahui bahwa nilai dari probabilitas *classifier* dapat mengklasifikasi dengan benar adalah 88%.



Gambar 33. Kurva ROC Percobaan Ketiga (Fitur Judul dan Tag Berita)

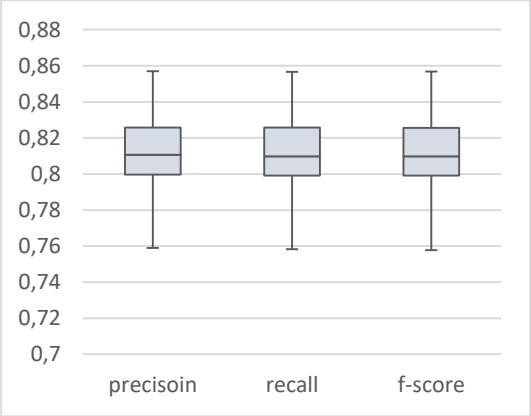
3) Fitur Judul dan Kategori

Pengujian selanjutnya adalah klasifikasi dengan menggunakan fitur judul dan kategori berita, berikut merupakan hasil evaluasi yang diperoleh seperti pada Tabel 19. Berdasarkan hasil evaluasi pada Tabel 19, rata-rata nilai evaluasi terbaik diperoleh dengan menggunakan parameter kernel RBF, C=20, dan gamma=’scale’. Hasil nilai evaluasi dengan menggunakan fitur judul dan kategori memperoleh nilai sedikit lebih baik yakni 0.81 dibandingkan dengan penggunaan fitur judul dan tag berita dengan nilai 0.80.

Tabel 19. Hasil Evaluasi dengan Fitur Judul dan Kategori

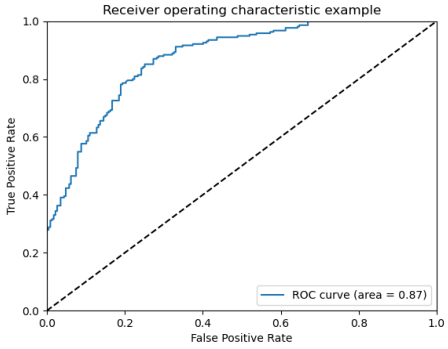
Parameter			Precision	Recall	F-Score
Kernel	**C**	**Gamma**			
Linear	10	-	0.769498	0.764133	0.762502
Linear	20	-	0.775704	0.769472	0.7677
Linear	23	-	0.775928	0.770058	0.768122
RBF	10	Scale	0.810718	0.809182	0.808492
RBF	20	Scale	**0.810993**	**0.810301**	**0.809931**
RBF	23	Scale	0.809502	0.809031	0.8086
Poly	10	-	0.806087	0.80477	0.804358
Poly	20	-	0.804321	0.803632	0.803323
Poly	23	-	0.807307	0.806705	0.806461

Berikut merupakan diagram boxplot dari 50 kali iterasi pengujian yang dilakukan dengan menggunakan parameter tersebut seperti pada Gambar 34. Hasil dari nilai evaluasi tersebut menunjukkan performa yang lebih baik dibandingkan dengan fitur judul kata sifat dan judul tag berita.



Gambar 34. Diagram Boxplot Nilai Terbaik Percobaan Ketiga (Fitur Judul dan Kategori)

Selanjutnya, dilakukan penghitungan nilai ROC dan dibuat grafik seperti pada Gambar 35. Sehingga, dari grafik tersebut diketahui bahwa nilai dari probabilitas *classifier* dapat mengklasifikasi dengan benar adalah 88%.



Gambar 35. Kurva ROC Percobaan Ketiga (Fitur Judul dan Kategori)

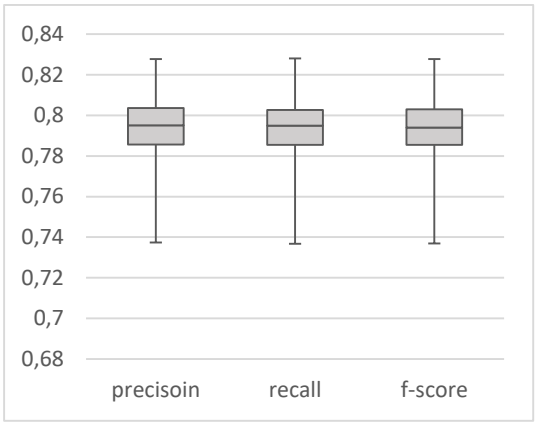
4) Fitur Judul, Kata Sifat, dan Tag Berita

Berikut merupakan hasil dari evaluasi klasifikasi dengan menggunakan fitur judul, kata sifat, dan tag berita seperti pada Tabel 20.

Tabel 20. Hasil Evaluasi dengan Fitur Judul, Kata Sifat, dan Tag

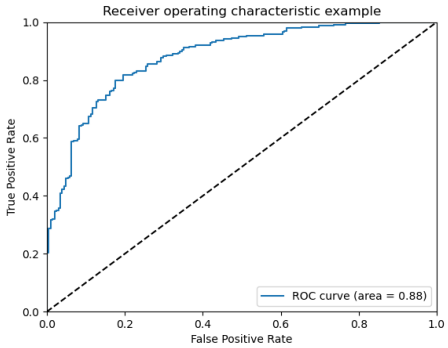
Parameter			**Precision**	**Recall**	**F-Score**
Kernel	**C**	**Gamma**			
Linear	10	-	0.771069	0.77046	0.769873
Linear	20	-	0.765092	0.76467	0.764246
Linear	23	-	0.770619	0.77015	0.769576
RBF	10	Scale	0.792484	0.791729	0.791189
RBF	20	Scale	0.789055	0.788944	0.788445
RBF	23	Scale	**0.79297**	**0.792823**	**0.792469**
Poly	10	-	0.786776	0.786518	0.78608
Poly	20	-	0.786392	0.786218	0.785975
Poly	23	-	0.780595	0.780359	0.779731

Berdasarkan hasil evaluasi pada Tabel 20, rata-rata nilai evaluasi terbaik diperoleh dengan menggunakan parameter kernel RBF, C=23, dan gamma=’scale’. Berikut merupakan diagram boxplot dari 50 kali iterasi pengujian yang dilakukan dengan menggunakan parameter tersebut seperti pada Gambar 36.



Gambar 36. Diagram Boxplot Nilai Terbaik Percobaan Ketiga (Fitur Judul, Kata Sifat, dan Tag Berita)

Evaluasi selanjutnya dilakukan dengan menghitung nilai ROC dan dibuat grafik seperti pada Gambar 37. Sehingga, dari grafik tersebut diketahui bahwa nilai dari probabilitas *classifier* dapat mengklasifikasi dengan benar adalah 88%.



Gambar 37. Kurva ROC Percobaan Ketiga (Fitur Judul, Kata Sifat, dan Tag Berita)

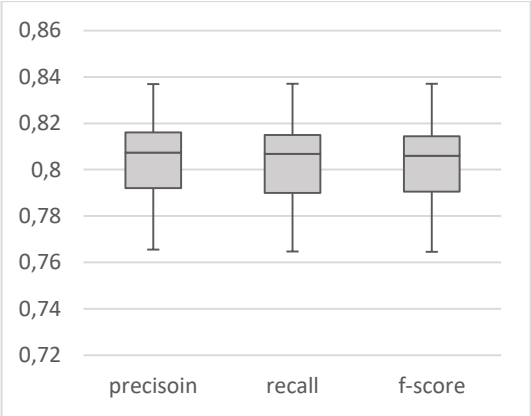
5) Fitur Judul, Tag Berita, dan Kategori

Berikut merupakan hasil dari evaluasi klasifikasi dengan menggunakan fitur judul, kata sifat, dan tag berita seperti pada Tabel 21. Berdasarkan dari hasil evaluasi pada Tabel 21, rata-rata nilai evaluasi terbaik diperoleh dengan menggunakan parameter kernel RBF, C=20, dan gamma=’scale’.

Tabel 21. Hasil Evaluasi dengan Fitur Judul, Tag, dan Kategori

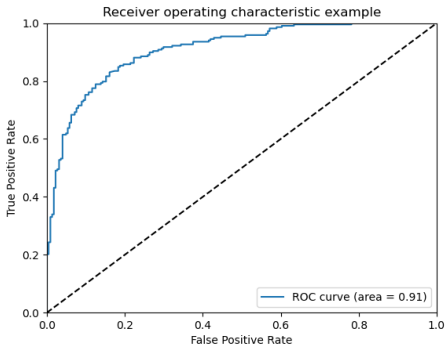
Parameter			Precision	Recall	F-Score
Kernel	**C**	**Gamma**			
Linear	10	-	0.770241	0.768734	0.767956
Linear	20	-	0.775385	0.774345	0.773505
Linear	23	-	0.770364	0.769245	0.768429
RBF	10	Scale	0.802893	0.802138	0.801668
RBF	20	Scale	**0.80456**	**0.804186**	**0.803751**
RBF	23	Scale	0.80245	0.802039	0.801661
Poly	10	-	0.793703	0.7935	0.793061
Poly	20	-	0.795982	0.795873	0.795196
Poly	23	-	0.798829	0.798753	0.798435

Berikut merupakan diagram boxplot dari 50 kali iterasi pengujian yang dilakukan dengan menggunakan parameter tersebut seperti yang ditampilkan pada Gambar 38.



Gambar 38. Diagram Boxplot Nilai Terbaik Percobaan Ketiga (Fitur Judul, Tag, dan Kategori)

Evaluasi selanjutnya dilakukan penghitungan nilai ROC dan dibuat grafik seperti pada Gambar 39. Sehingga, dari grafik tersebut diketahui bahwa nilai dari probabilitas *classifier* dapat mengklasifikasi dengan benar adalah 91%.



Gambar 39. Kurva ROC Percobaan Ketiga (Fitur Judul, Tag, dan Kategori)

6) Fitur Judul, Kata Sifat, Tag Berita, Kategori, dan Situs

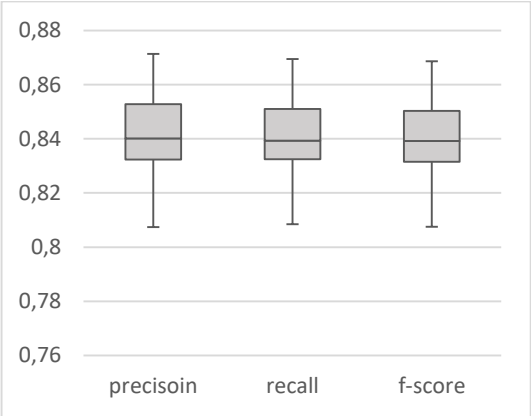
Dari pengujian yang dilakukan, berikut merupakan hasil pengujian yang diperoleh dengan menggunakan keseluruhan fitur, seperti pada Tabel 22.

Tabel 22. Hasil Evaluasi dengan Semua Fitur

Parameter			**Precision**	**Recall**	**F-Score**
Kernel	**C**	**Gamma**			
Linear	10	-	0.804218	0.802674	0.802043
Linear	20	-	0.800439	0.799806	0.799185
Linear	23	-	0.800427	0.799444	0.798864
RBF	10	Scale	0.838138	0.838268	0.837638
RBF	20	Scale	0.837905	0.837321	0.836865
RBF	23	Scale	**0.840879**	**0.84055**	**0.840106**
Poly	10	-	0.835983	0.834977	0.834419
Poly	20	-	0.832643	0.831453	0.830939
Poly	23	-	0.834337	0.832875	0.832816

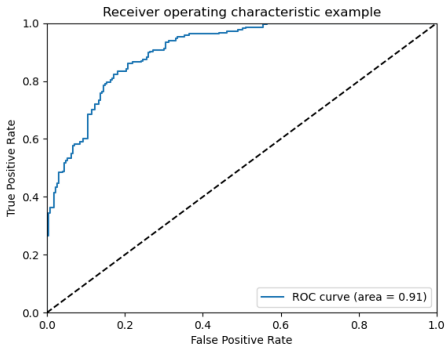
Berdasarkan evaluasi yang dilakukan, hasil rata-rata nilai evaluasi paling baik diperoleh dengan menggunakan parameter kernel RBF, C=23.0, dan gamma=’scale’. Berikut merupakan diagram boxplot dari 50 kali iterasi

pengujian yang dilakukan dengan menggunakan parameter tersebut seperti pada Gambar 40.



Gambar 40. Diagram Boxplot Nilai Terbaik Percobaan Ketiga (Semua Fitur)

Evaluasi selanjutnya dengan dilakukan penghitungan nilai ROC dan dibuat grafik seperti pada Gambar 41. Sehingga, dari grafik tersebut diketahui bahwa nilai dari probabilitas *classifier* dapat mengklasifikasi dengan benar adalah 91%.



Gambar 41. Kurva ROC Percobaan Ketiga (Semua Fitur)

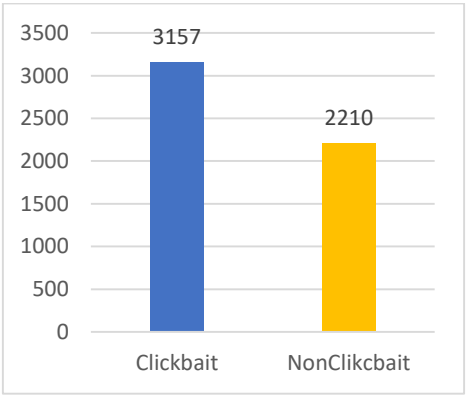
Dari hasil evaluasi-evaluasi yang telah dilakukan, dapat diketahui bahwa rata-rata nilai evaluasi yang diperoleh lebih baik dibandingkan dengan penggunaan *word embedding* sebagai fitur. Selain itu, dari beberapa percobaan pengujian dengan kombinasi beberapa fitur dapat diketahui bahwa fitur yang memiliki pengaruh terbesar terhadap hasil evaluasi dari klasifikasi adalah fitur kategori dan tag berita. Berikut merupakan rangkuman hasil nilai evaluasi tertinggi yang diperoleh dari masing-masing percobaan yang ditampilkan pada Tabel 23.

Tabel 23. Rangkuman Hasil Evaluasi

Fitur	Precision	Recall	F-Score
Word2Vec	0.803929	0.803206	0.802723
Bag Of Word	0.848711	0.848063	0.847791
Node2Vec	0.840879	0.84055	0.840106

d. Analisis Data

Berdasarkan pengujian dan evaluasi yang telah dilakukan, selanjutnya analisis untuk melihat perbedaan dari distribusi penggunaan *stopword* pada hasil klasifikasi yang diperoleh dari penggunaan SVM dengan *word embedding* sebagai representasi fitur. Berdasarkan hasil klasifikasi dari *machine learning*, diperoleh judul berita yang masuk pada kategori *clickbait* sebanyak 1044 judul berita dan *nonclickbait* 1162 judul berita. Dari hasil tersebut dihitung total banyaknya kata yang termasuk dalam *stopword* pada masing-masing label klasifikasi seperti pada Gambar 42.



Gambar 42. Grafik Perbandingan Penggunaan Kata *Stopword*

Dari hasil penghitungan tersebut, berikutnya dilakukan penghitungan untuk memperoleh 10 kata yang paling banyak muncul pada judul berita hasil dari pengklasifikasian antara berita *clickbait* dan *nonclickbait* seperti yang ditunjukkan pada Tabel 24. Berdasarkan Gambar 42 dan Tabel 24 dapat dilihat bahwa berita *clickbait* menggunakan lebih banyak kata yang masuk dalam kategori *stopword* dibandingkan dengan berita *nonclickbait*.

Tabel 24. Kata yang sering muncul pada data *clickbait* dan *nonclickbait*

Clickbait		NonClickbait	
Kata	**Jumlah**	**Kata**	**Jumlah**
ini	594	di	343
di	242	dan	91
yang	168	jokowi	85
begini	92	ke	70
dan	92	polisi	64
jadi	76	kpk	59
dengan	70	dari	59
viral	60	tak	53
soal	51	indonesia	52
kata	48	akan	50

Dari 10 kata yang paling sering muncul pada judul berita *clickbait*, 8 diantaranya termasuk dalam kata *stopword*. Sedangkan, untuk data yang diklasifikasikan sebagai *nonclickbait* dari 10 kata yang paling sering muncul 6 diantaranya adalah kata *stopword*. Sehingga, kata *stopword* yang terdapat pada kalimat judul berita *clickbait* mempengaruhi nilai hasil evaluasi kinerja dari klasifikasi dengan nilai *precision* 80% untuk klasifikasi dengan dataset yang mengandung kata *stopword* dan nilai *precision* 69% untuk klasifikasi dengan dataset yang tidak mengandung kata *stopword*.

4.1.5. Analisis Judul Berita dengan Node2Vec

Sebelum masuk ke dalam proses *embedding* dengan menggunakan Node2Vec, tahap awal yang dilakukan adalah melakukan POSTagging pada judul berita untuk memperoleh kata sifat dari judul yang nantinya akan digunakan sebagai *node*. Proses POSTagging menggunakan *library* NLTK dan menggunakan *class* `CRFTagger`. *Corpus* yang digunakan adalah *corpus* model tag bahasa Indonesia. Proses yang dilakukan adalah dengan memecah judul berita yang kemudian akan diberi label sesuai dengan kategori jenisnya. Untuk kata dengan kategori adjektiva

atau kata sifat dilabeli dengan label ‘JJ’. Pada Tabel 25 merupakan contoh hasil dari proses POSTagging.

Tabel 25. Hasil POSTagging Judul Berita

	Teks / Hasil POSTagging	Adj
1.	bukti baru celaka maut tol cipularang truk nyata tidak over kapasitas tapi ('bukti', 'NN'), **('baru', 'JJ')**, ('celaka', 'SC'), ('maut', 'NN'), ('tol', 'NN'), ('cipularang', 'NN'), ('truk', 'NN'), **('nyata', 'JJ')**, ('tidak', 'NEG'), ('over', 'VB'), ('kapasitas', 'NN'), ('tapi', 'CC')	baru,nyata
2.	viral pin emas anggota dprd biaya fantastis ('viral', 'NN'), ('pin', 'NN'), ('emas', 'NN'), ('anggota', 'NN'), ('dprd', 'NN'), ('biaya', 'NN'), **('fantastis', 'JJ')**	fantastis
3.	viral bakar surat suara di papua istana itu bukan dokumen penting **('viral', 'JJ')**, ('bakar', 'VB'), ('surat', 'NN'), ('suara', 'NN'), ('di', 'IN'), ('papua', 'CD'), ('istana', 'NN'), ('itu', 'PR'), ('bukan', 'NEG'), ('dokumen', 'NN'), **('penting', 'JJ')**	viral,penting

Dataset yang digunakan sebagai node terdiri dari idberita, katasifat, tagberita, kategori, situs, dan status. Selanjutnya, dilakukan proses *query* untuk membuat *node* dari dataset yang sudah dikategorikan ke dalam *clickbait* dan *nonclickbait* dan menyimpannya ke dalam Neo4J seperti pada Tabel 26. Dari tabel tersebut diketahui bahwa untuk *node* `:Berita` memiliki atribut idberita, judul, dan status. Sedangkan untuk *node* `:KataSifat` memiliki atribut katasifat, *node* `:TagBerita` memiliki atribut tag, `:Kategori` memiliki atribut kategori, dan `:Situs` memiliki atribut situs.

Tabel 26. *Query Create Node*

Node	Query

:Berita	`create_berita_query = '''`
	`UNWIND {nodes} as node`
	`CREATE (n:Berita {idberita: node.idberita, judul: node.judul, status: node.status})`
	`'''`

:KataSifat	`create_katasifat_nodes = '''`
	`UNWIND {nodes} as node`
	`CREATE (n:KataSifat {katasifat:`

	node.katasifat}) '''
:TagBerita	create_tagberita_nodes = ''' UNWIND {nodes} as node CREATE (n:TagBerita {tag: node.nama}) '''
:Kategori	create_kategori_nodes = ''' UNWIND {nodes} as node CREATE (n:Kategori {kategori: node.nama}) '''
:Situs	create_situs_nodes = ''' UNWIND {nodes} as node CREATE (n:Situs {situs: node.situs}) '''

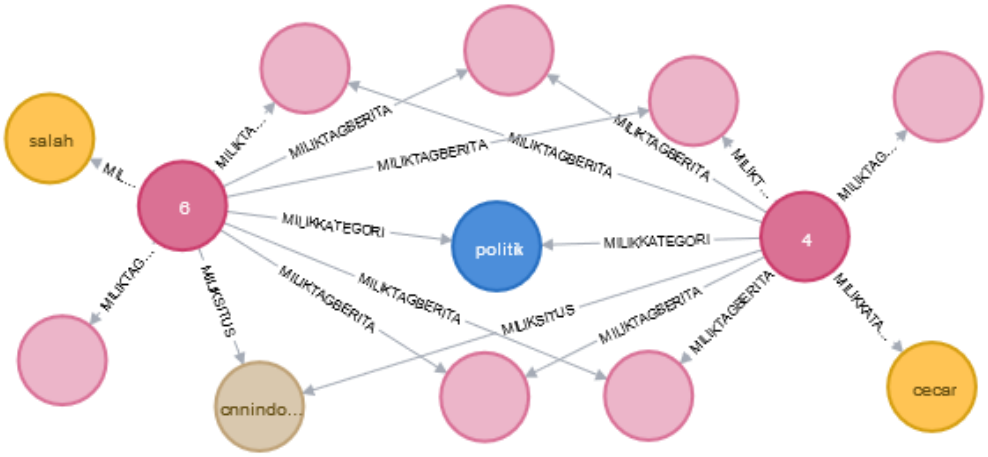
Setelah semua *node-node* tersebut dibuat, selanjutnya adalah membuat *edge* atau relasi antar *node*. Pada Tabel 27 menunjukkan *query* untuk membuat relasi antar *node*. Relasi yang dibuat antara lain adalah :MILIKKATASIFAT untuk relasi berita dan kata sifat, :MILIKTAGBERITA untuk relasi berita dan tag berita, :MILIKKATEGORI untuk relasi berita dan kategori, :MILIKSITUS untuk relasi berita dan situs.

Tabel 27. *Query Create Relationship*

Node	Query
:Berita → :KataSifat	create_berita_judultag_relationship = ''' UNWIND {records} as record MATCH (a:Berita) where a.idberita=record.idberita MATCH (b:KataSifat) where b.katasifat in record.katasifat CREATE (a)-[:MILIKKATASIFAT]->(b) '''
:Berita → :TagBerita	create_berita_tag_relationship = ''' UNWIND {records} as record MATCH (a:Berita) where a.idberita=record.idberita MATCH (b:TagBerita) where b.tag in record.tag CREATE (a)-[:MILIKTAGBERITA]->(b) '''
:Berita → :Kategori	create_berita_kategori_relationship = ''' UNWIND {records} as record MATCH (a:Berita) where a.idberita=record.idberita MATCH (b:Kategori) where b.kategori in record.kategori

	CREATE (a)-[:MILIKKATEGORI]->(b) '''
:Berita → :Situs	create_berita_situs_relationship = ''' UNWIND {records} as record MATCH (a:Berita) where a.idberita=record.idberita MATCH (b:Situs) where b.situs in record.situs CREATE (a)-[:MILIKSITUS]->(b) '''

Dari hasil *query* tersebut diperoleh *node* sebanyak 6636 dan *edge* sebanyak 12949 relasi. Pada Gambar 43 merupakan contoh hasil *node* dan *edge* yang telah dibuat. Secara berturut-turut warna ungu, kuning, ungu muda, biru, dan coklat adalah *node* :Berita, :KataSifat, :TagBerita, :Kategori, dan :Situs.



Gambar 43. Hasil *Node* dan *edge*

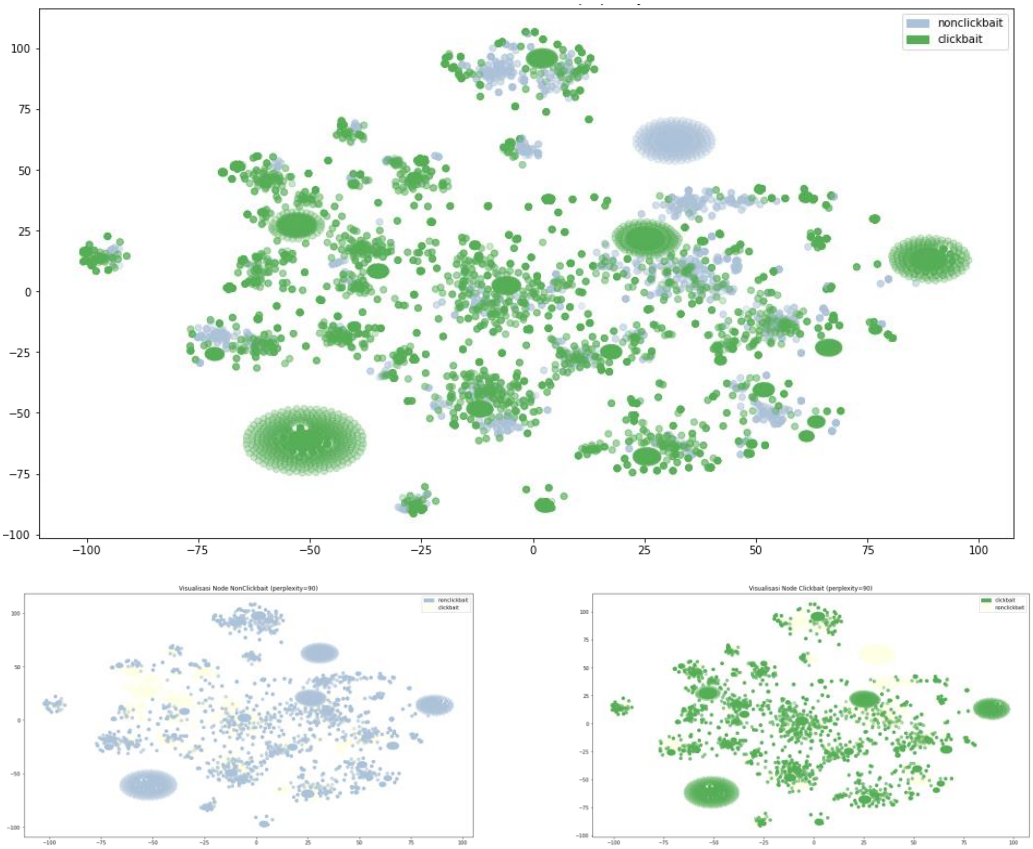
Proses selanjutnya adalah membuat daftar *edgelist*, yakni daftar keseluruhan relasi yang ada di dalam *graph* dan hasilnya akan digunakan sebagai masukan pada Node2Vec. Beberapa parameter yang digunakan untuk proses dari Node2Vec adalah `-d=16` yang merupakan jumlah dimensi dari *embedding*. Parameter `-l=10` yang merupakan panjang dari proses jalan yang dilakukan dari *node* sumber. Parameter `-r=20` merupakan jumlah perjalanan yang dilakukan dari tiap *node* sumber. Parameter `-dr` yang menunjukkan bahwa *graph* merupakan *directed graph*. Contoh hasil dari *graph embedding* dapat dilihat pada Gambar 44. Pada baris

pertama menunjukkan bahwa terdapat 6636 baris node yang memiliki panjang vektor dengan dimensi 16.

```
6636 16
6420 -0.195275 -0.392017 0.842264 -0.701083 1.07902 0.0997727 0.598018 0.570257 -0.419239 -1.07643 -1.1861 0.0493989 -0.557004 0.393388 -0.301216 -0.828151
665 -0.131785 -0.498085 1.06705 -0.766006 1.45317 0.0546795 0.7207 0.324395 -0.47702 -1.2937 -1.5403 0.439168 -0.747173 0.251352 -0.341714 -1.10577
3331 -0.428215 -0.463357 0.871961 -0.689077 1.25732 0.0166039 0.609552 0.575071 -0.53688 -1.30391 -1.34872 0.169902 -0.815967 0.198242 -0.119146 -0.956657
5288 -0.216134 -0.333251 0.758368 -0.599217 1.12509 0.129933 0.519667 0.485236 -0.406519 -0.941471 -1.08732 0.0516871 -0.563432 0.399495 -0.236079 -0.759176
1217 -0.55215 0.146947 0.704482 0.117665 0.558553 0.486987 -0.0678036 0.180614 0.762351 -1.36419 -0.658093 -0.0783192 0.149787 0.422325 -0.644164 -1.634
6634 -0.314427 0.0698723 0.574113 -0.345427 0.221351 0.992304 0.368525 0.326725 0.437958 -1.28409 -0.553607 -0.677359 -0.335133 0.55014 -0.570096 -1.75476
605 -0.0325021 -1.10821 0.53267 -0.235898 0.758065 0.853341 0.444506 0.998317 0.276259 -1.21033 -0.56878 -0.219426 -0.247922 0.544297 -0.855135 -1.97385
6662 -0.123865 -0.165885 0.183333 0.126663 0.573788 0.508748 0.0541983 0.016584 -0.484832 -0.732456 -0.527214 0.0872858 -0.300809 0.307599 -0.751479 -1.0257
1176 0.0216374 0.236904 0.414045 -0.0159181 0.270563 1.22309 -0.0737272 -0.238036 0.596551 -1.10039 -0.937201 -0.836397 -0.377469 -0.182716 -1.25144 -1.5250
5489 0.00310011 0.075459 0.311086 -0.163708 0.220528 1.07596 0.114764 -0.0182103 0.509282 -0.780086 -0.850682 -0.650249 -0.246291 -0.10335 -1.10976 -1.52346
3033 -0.0818109 -0.0121315 0.230257 -0.366002 0.119969 0.889582 0.0471118 -0.00379617 0.525533 -0.939351 -0.634866 -0.504739 -0.315992 0.0641512 -0.98029 -1
2532 0.0406835 0.343273 0.000686874 0.000183099 0.0618581 0.770696 -0.255217 -0.208777 0.884732 -0.593456 -0.924698 -0.642286 -0.585179 -0.597594 -2.20161 -
1466 -0.597869 0.765233 0.474967 -0.121051 0.908432 0.787391 0.219245 -0.385859 0.240446 -1.06343 -0.687872 -0.642297 -0.463823 1.3732 -0.798907 -1.76248
1317 -0.348907 -0.0937187 0.191679 -0.0453506 0.411473 0.635496 1.49015 0.404268 -0.00756529 -0.946903 -0.553986 -0.729211 -0.300941 0.40186 -0.0399282 -1.7
2266 -0.675895 0.758792 -0.429037 -0.210453 -0.146923 0.50037 1.44247 0.627885 0.137757 -0.607261 -0.520797 -0.43347 -0.727403 0.169066 -0.254815 -1.69111
```

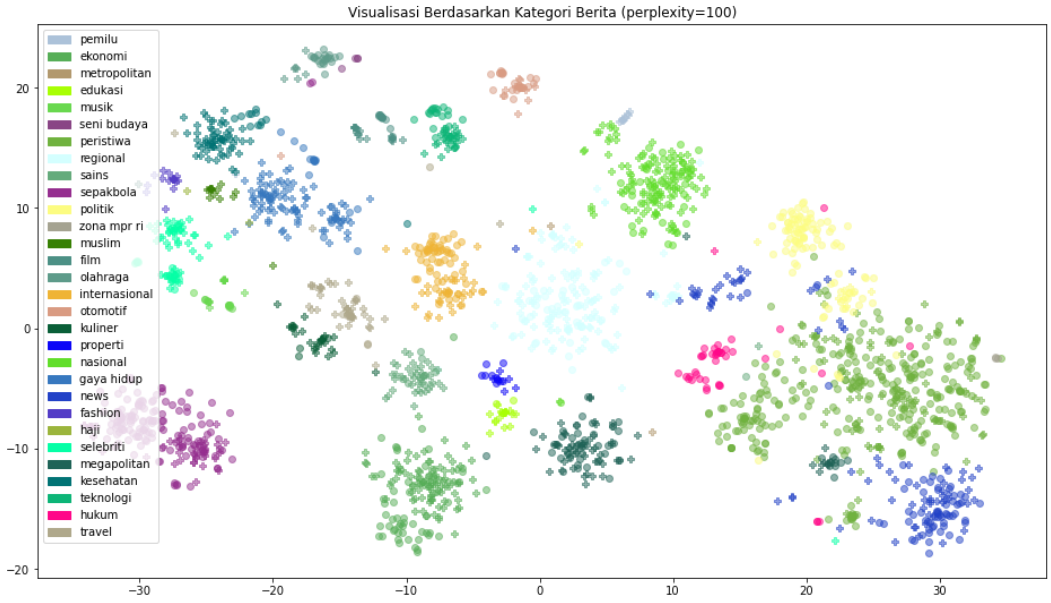
Gambar 44. Hasil *Graph Embedding*

Dari hasil *embedding* tersebut selanjutnya dilakukan analisis dengan memvisualisasikan vektor dari *node-node* yang terkait berdasarkan klasifikasi *clickbait* dan *nonclickbait*. Visualisasi dilakukan dengan menggunakan t-SNE. T-SNE merupakan teknik non-linear untuk mereduksi dari *high-dimensional* dataset menjadi *low-dimensional* dataset. Algoritma t-SNE bekerja dengan menghitung probabilitas kesamaan dari titik atau poin pada ruang *high-dimensional* dengan titik atau poin yang ada pada ruang *low-dimensional*.



Gambar 45. Visualisasi Node *Cickbait* dan *NonClickbait* (Perplexity=90)

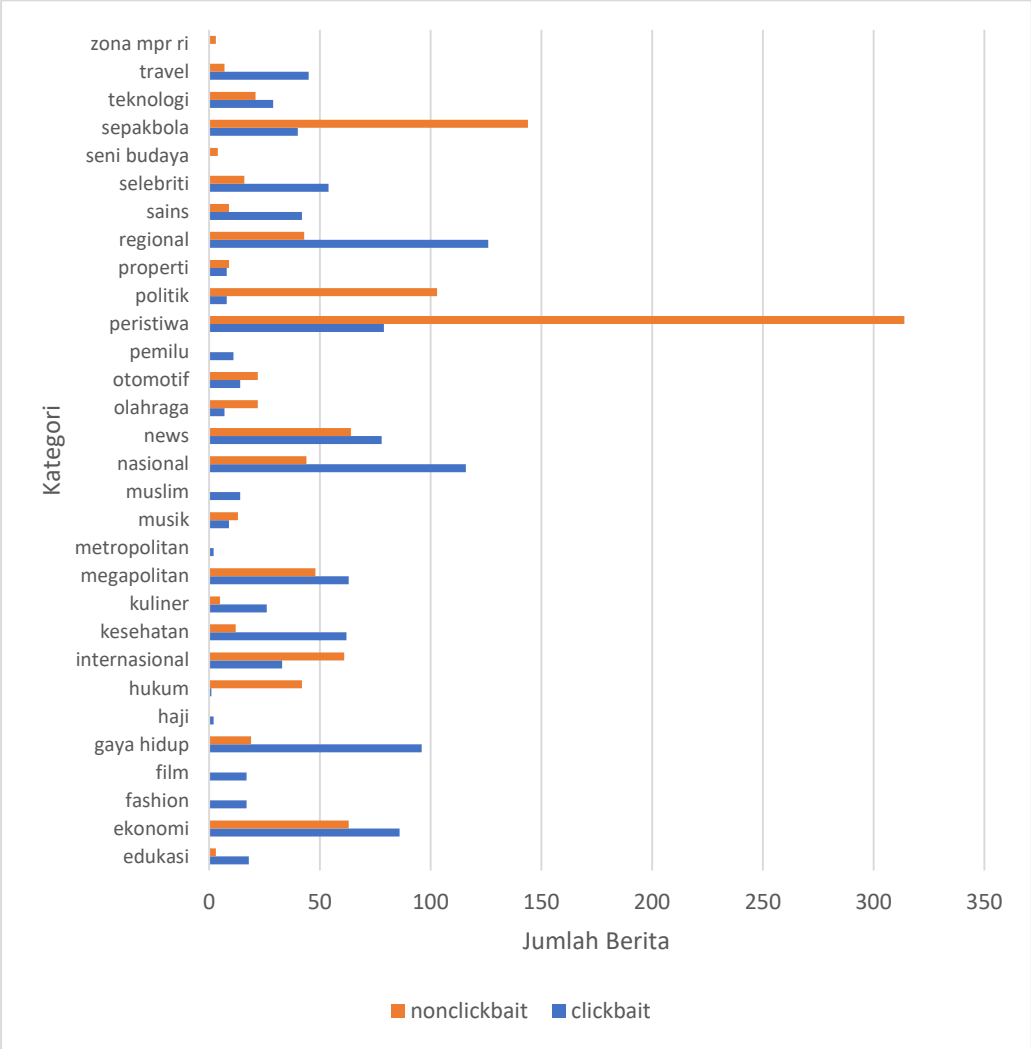
Pada Gambar 45 menunjukkan visualisasi dari vektor *node* dengan keterangan, biru untuk *nonclickbait* dan hijau untuk *clickbait*. Berdasarkan hasil dari visualisasi tersebut vektor representasi dari hubungan poin-poin antara *clickbait* dan *nonclickbait* memiliki kemiripan satu sama lain sehingga terdapat poin antara *clickbait* dan *nonclickbait* yang saling bertumpang tindih. Berdasarkan gambar tersebut diketahui bahwa poin-poin hasil dari *node embedding* yang digunakan antara *clickbait* dan *nonclickbait* memiliki banyak kesamaan atau kemiripan sehingga hal ini menjadi salah satu yang menyebabkan nilai evaluasi pada proses klasifikasi menggunakan SVM menghasilkan nilai yang kurang maksimal karena hanya sedikit poin-poin yang benar-benar memiliki perbedaan antara *clickbait* dan *nonclickbait*.



Gambar 46. Visualisasi Berdasarkan Kategori Berita

Pada Gambar 46, merupakan visualisasi dari vektor Node2Vec yang dibagi berdasarkan kategori berita. Poin dengan simbol “o” merupakan poin yang masuk dalam klasifikasi *nonclickbait*, sedangkan poin dengan simbol “+” merupakan poin untuk *clickbait*. Pada hasil visualisasi tersebut diketahui bahwa *node* yang masuk ke dalam label *clickbait* tersebar pada berbagai kategori sehingga sulit untuk diambil kesimpulan. Hal tersebut menunjukkan bahwa tidak ada perbedaan yang

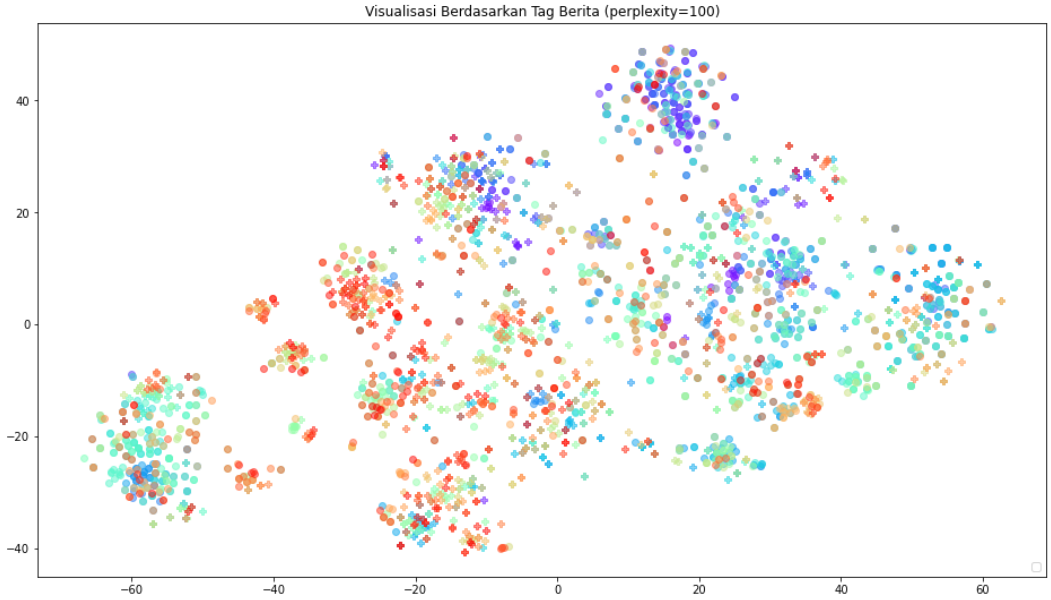
signifikan pada kategori yang digunakan antara berita *clickbait* dan *nonclickbait*. Oleh karena itu, dilakukan penghitungan untuk mengetahui banyaknya jumlah berita *clickbait* dan *nonclickbait* pada masing-masing kategori seperti pada Gambar 47. Berdasarkan grafik tersebut, lima kategori yang paling banyak muncul pada berita *clickbait* adalah regional, nasional, gaya hidup, ekonomi, dan peristiwa. Sedangkan, untuk berita *nonclickbait* adalah peristiwa, sepakbola, politik, *news*, dan ekonomi.



Gambar 47. Jumlah Berita *Clickbait* dan *NonClickbait* per Kategori

Selanjutnya, dibuat visualiasi berdasarkan tag berita seperti pada Gambar 48. Berdasarkan visualisasi tersebut antara *node* dengan label *clickbait* dan *nonclickbait* memiliki banyak kesamaan dan tidak memiliki perbedaan yang signifikan. Dari

gambar tersebut dapat diketahui bahwa tiap berita memiliki variasi tag yang berbeda. Hal ini dikarenakan berita dengan topik yang sama belum tentu memiliki tag yang sama, karena dari masing-masing situs memiliki ketentuan penggunaan tag yang berbeda-beda. Oleh karena itu, untuk mengetahui jumlah dari tag berita yang digunakan, berikut disajikan 10 besar tag berita yang sering muncul pada berita *clickbait* dan *nonclickbait* seperti pada Tabel 28.



Gambar 48. Visualisasi Berdasarkan Tag Berita

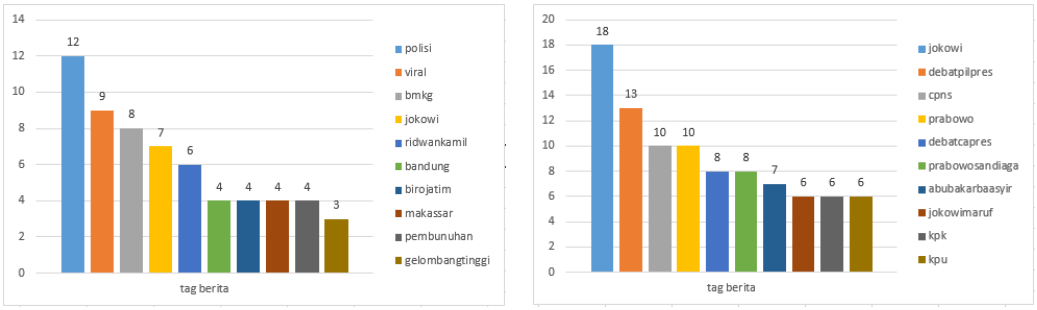
Tabel 28. 10 Besar Tag Berita pada Berita *Clickbait* dan *NonClickbait*

Clickbait		**NonClickbait**	
Tag Berita	**Jumlah Berita**	**Tag Berita**	**Jumlah Berita**
Jokowi	46	Jokowi	68
Prabowo	20	Pilpres	68
Kesehatan	20	Pemilu	58
Viral	19	Program TV News	41
Polisi	18	KPK	39
KPK	16	Liputan SCTV	30
Pilpres	15	Liga Inggris	29
Debat Pilpres	15	Merdeka.com	27
BMKG	12	Debat Capres	23

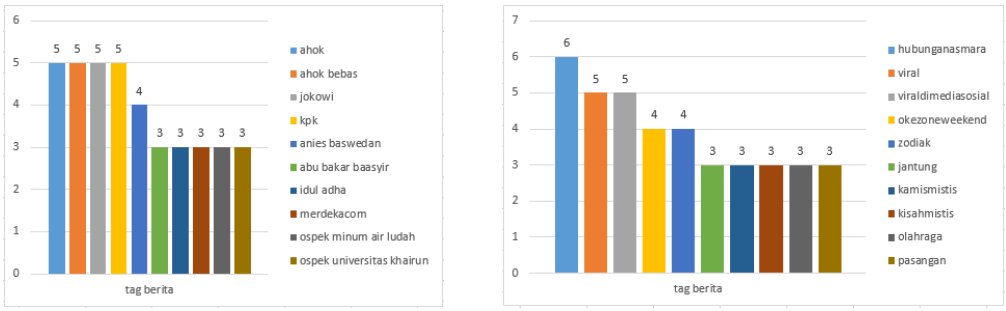
Debat Capres	11	Manchester United	22

Berdasarkan pada 5 kategori yang paling banyak muncul pada berita *clickbait* dan *nonclickbait*, selanjutnya dihitung jumlah dari tag yang sering muncul pada kategori-kategori tersebut. Gambar 49 menunjukkan 10 besar dari tag berita yang sering muncul pada berita *clickbait* dari masing-masing kategori.

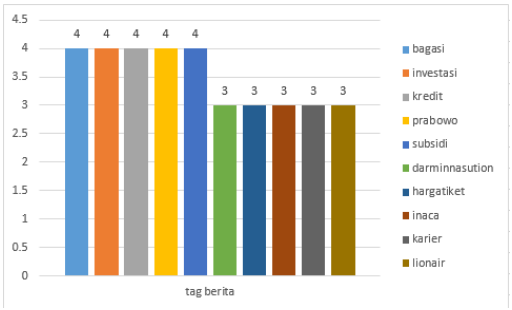
(a) (b)



(c) (d)



(e)



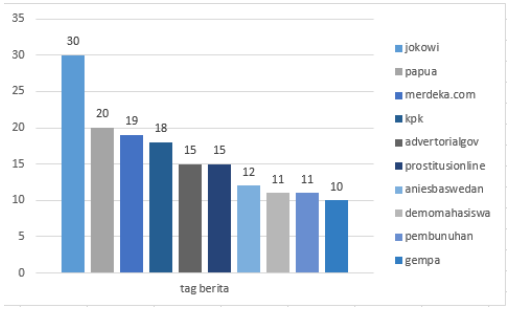
Gambar 49. 10 Besar Tag Berita pada Berita *Clickbait* (a) Kategori Regional (b) Kategori Nasional (c) Kategori Peristiwa (d) Gaya Hidup (e) Ekonomi

Berdasarkan data tersebut, tag berita pada kategori regional, nasional, dan peristiwa memiliki beberapa tag dengan konteks yang sama seputar pemilu atau pilpres. Topik pembahasan yang diangkat untuk berita *clickbait* bisa bervariasi dan

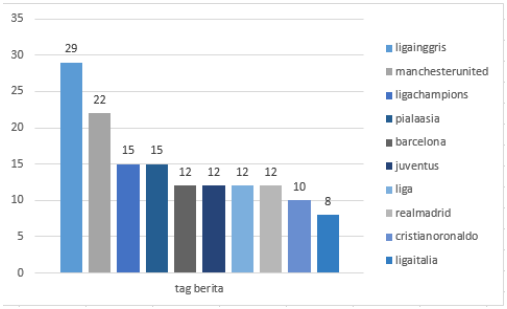
tergantung pada berita yang sedang hangat pada saat itu. Misalkan, karena dataset yang digunakan pada penelitian ini diperoleh dari berita yang diterbitkan pada tahun 2019 dan merupakan tahun untuk pemilihan umum presiden Indonesia, sehingga beberapa berita *clickbait* memiliki keterkaitan dengan momen tersebut.

Penghitungan tag berita yang sering muncul juga dilakukan pada 5 kategori yang sering ditemui pada berita *nonclickbait*. Gambar 50 menunjukkan grafik dari 10 besar tag berita yang sering muncul pada berita *nonclickbait* dari masing-masing 5 kategori teratas.

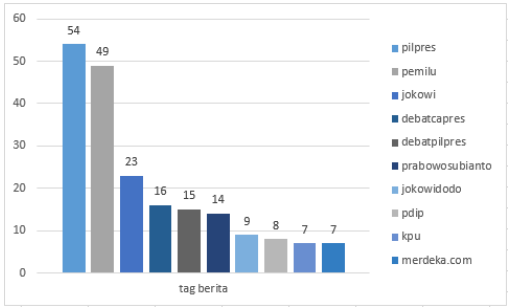
(a)



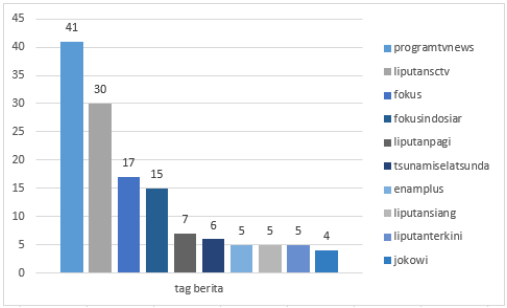
(b)



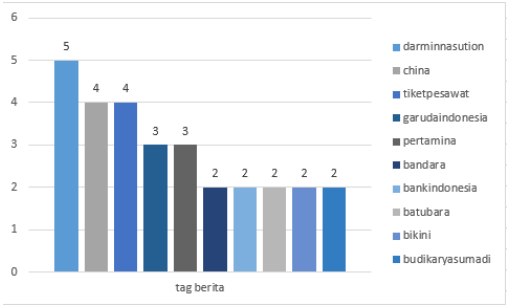
(c)



(d)

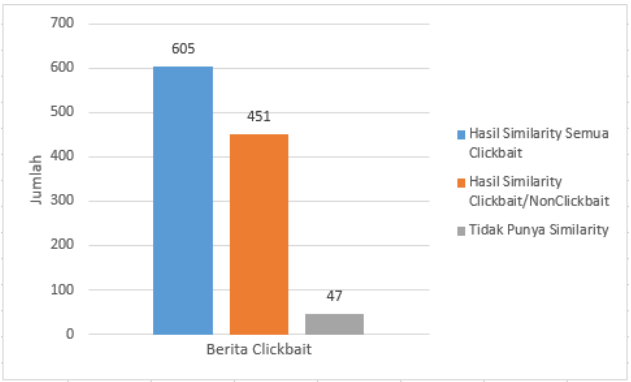


(e)



Gambar 50. 10 Besar Tag Berita pada Berita *NonClickbait* (a) Kategori Peristiwa (b) Kategori Sepak Bola (c) Kategori Politik (d) Kategori News (e) Kategori Ekonomi

Pada analisis selanjutnya, dilakukan pengecekan *similarity* pada berita *clickbait* dan *nonclickbait* berdasarkan dari hasil *graph embedding* yang telah dibuat dengan menggunakan Node2Vec. Dari hasil pengecekan *similarity* tersebut, berikut pada Gambar 51 merupakan grafik dari hasil pengecekan *similarity* untuk berita yang dikategorikan *clickbait*.



Gambar 51. Hasil Pengecekan *Similarity* Berita *Clickbait*

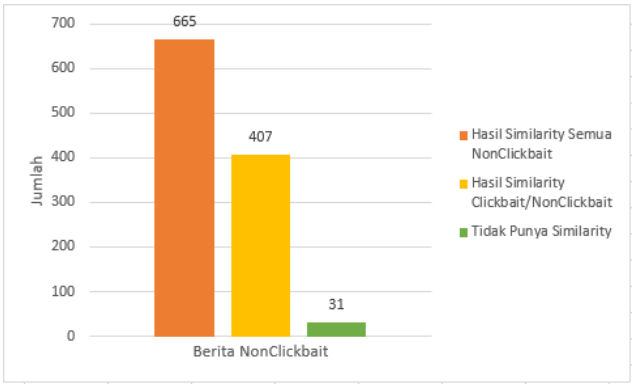
Berdasarkan grafik pada Gambar 51 mengenai jumlah dari hasil pengecekan *similarity* pada judul berita *clickbait*. Warna biru menunjukkan jumlah dari judul berita *clickbait* yang memiliki hasil *similarity* keseluruhannya juga *clickbait*. Warna oranye menunjukkan jumlah dari judul berita *clickbait* yang hasil *similarity*nya terdapat judul *nonclickbait*. Sedangkan warna abu-abu menunjukkan judul berita *clickbait* yang tidak memiliki *similarity* dengan judul berita lain. Berdasarkan dari grafik tersebut menunjukkan bahwa dari 1103 judul berita *clickbait*, sekitar 40%

nya masih memiliki kesamaan dengan judul berita *nonclickbait*. Berdasarkan dari hasil pengecekan *similarity* pada berita *clickbait*, berikut ditampilkan contoh dari hasil pengecekan *similarity* untuk beberapa judul berita *clickbait* seperti pada Tabel 29. Pada nomor yang pertama merupakan contoh dari judul berita *clickbait* yang semua hasil *similarity*-nya adalah judul berita yang juga *clickbait*. Contoh nomor kedua merupakan contoh dari berita *clickbait* yang memiliki hasil *similarity* dengan judul berita *nonclickbait*. Dari 3 hasil yang diperoleh, salah satunya adalah judul dengan kategori *nonclickbait*. Untuk contoh nomor ketiga merupakan judul berita *clickbait* yang tidak memiliki *similarity* dengan berita lain.

Tabel 29. Contoh Hasil Pengecekan *Similarity* pada Berita *Clickbait*

1.	**Judul: Jadi 'Panglima' Wilayah Timur, Ini Strategi Soekarwo Menangkan Demokrat**	***Score***
	Hasil Similarity **(Semua Hasil *Clickbait*)**:	
	• Ribuan Pekerja di Jatim Dukung Jokowi-Ma'ruf Amin, Ini Alasannya (clickbait)	0.93
	• Jokowi: Saya Lihat Pangudi Luhur Bukan Asal Sandiaga Uno, tapi... (clickbait)	0.93
	• Saat Mobil Jokowi Dicegat Reog dan Jaranan di Tengah Jalan (clickbait)	0.92
2.	**Judul: Maju Lagi Jadi Capim, Ini Alasan Komisioner KPK Laode M Syarif**	
	Hasil Similarity (***Clickbait* dan *NonClickbait***)	
	• Dilaporkan Karena Dugaan Langgar Etik, Ini Kata Deputi Pencegahan KPK (clickbait)	0.95
	• Silaturahmi ke Ponpes, Capim KPK Irjen Firli Bantah Langgar Kode Etik (nonclickbait)	0.91
	• 2 Hari Tak Ada Kabar, Warga Madiun Ini Ternyata Tewas Kecelakaan (clickbait)	0.91
3.	Judul Berita **Tidak Memiliki *Similarity***: **Heboh Kabar Bupati Ngamuk di Kafe Makassar**	

Setelah melakukan pengecekan *similarity* pada berita *clickbait*, selanjutnya dilakukan juga pengecekan *similarity* pada berita *nonclickbait* yang hasilnya digambarkan pada grafik Gambar 52.



Gambar 52. Hasil Pengecekan *Similarity* Berita *NonClickbait*

Gambar 52 menunjukkan grafik untuk jumlah dari hasil pengecekan *similarity* untuk judul berita *nonclickbait*. Warna Oranye menunjukkan jumlah dari judul *nonclickbait* yang keseluruhan hasil *similarity* nya merupakan *nonclickbait*. Sedangkan untuk warna kuning menunjukkan jumlah dari judul berita *nonclickbait* yang hasil *similarity* nya terdapat judul *clickbait*. Warna hijau merupakan jumlah dari judul *nonclickbait* yang tidak memiliki *similarity* sama sekali dengan judul berita lain. Berdasarkan dari data grafik tersebut, menunjukkan bahwa sekitar 37% dari judul berita *nonclickbait* memiliki kesamaan dengan judul berita *clickbait*.

Tabel 30. Contoh Hasil Pengecekan *Similarity* pada Berita *NonClickbait*

1.	**Judul: Tim Forensik Temukan Bekas Jeratan di Leher Mayat Terbakar**	***Score***
	Hasil Similarity **(Semua Hasil *NonClickbait*)**:	
	• Hasil TelusurHasil webPembunuh Wanita dalam Kardus Divonis 14 Tahun (nonclickbait)	0.98
	• Divonis Seumur Hidup, Oknum TNI Pembunuh Pacar Ajukan Banding (nonclickbait)	0.97
2.	**Judul: Densus 88 Pantau Sel Tidur Teroris Jelang Hari Bebas Ba'asyir**	
	Hasil Similarity (***Clickbait* dan *NonClickbait***)	
	• Ba'asyir Dianjurkan Jalani Fisioterapi Tiga Kali Sepekan (nonclickbait)	0.94
	• Yusril: Abu Bakar Baasyir Seharusnya Tanda Tangani Ikrar, tapi... (clickbait)	0.91
	• Jokowi: Dokumen Strategi Cegah Korupsi Berdebu Jika Tak Jalan (nonclickbait)	0.91

3. Judul Berita **Tidak Memiliki *Similarity***:

Kenalkan Teknologi Pertanian Baru, Yanmar Sasar Asia Tenggara (nonclickbait)

Berikut pada Tabel 30 merupakan contoh dari hasil pengecekan *similarity* untuk berita *nonclickbait*. Contoh nomor satu merupakan contoh dari judul berita *nonclickbait* yang keseluruhan hasil *similarity*-nya juga *nonclickbait*. Sedangkan, nomor dua merupakan contoh dari berita *nonclickbait* yang hasil *similarity*-nya terdapat judul berita *clickbait*. Pada contoh nomor dua, dari 3 judul berita yang dihasilkan oleh pengecekan *similarity*, salah satu diantaranya adalah judul berita *clickbait*. Contoh ketiga merupakan berita *nonclickbait* yang tidak memiliki *similarity* sama sekali dengan berita lain.

(Halaman ini dikosongkan)

BAB 5
KESIMPULAN DAN SARAN

Bab ini menjelaskan mengenai kesimpulan berdasarkan hasil dan pembahasan yang telah dilakukan pada bab sebelumnya. Saran yang diperoleh dari kesimpulan juga disampaikan agar berguna sebagai masukan dalam pengembangan penelitian di masa datang.

5.1 Kesimpulan

Berdasarkan tahap yang telah dilakukan sejak awal penelitian dapat ditarik kesimpulan berupa rangkuman dari proses pengumpulan data, *pre processing*, pembuatan model Word2Vec, Klasifikasi SVM, dan analisis menggunakan Node2Vec.

1. *Word embedding* dengan menggukan Word2Vec menghasilkan vektor yang dapat digunakan sebagai fitur dalam proses klasifikasi menggunakan SVM. Penggunaan *stopword* yang ada pada dataset mempengaruhi nilai evaluasi dari klasifikasi yang dilakukan. Hal ini dikarenakan susunan kata yang digunakan pada judul berita *clickbait* lebih banyak menggunakan kata yang masuk dalam kategori *stopword*. Dari hasil klasifikasi pada dataset yang digunakan, kata *stopword* muncul sebanyak 3157 pada judul berita yang diklasifikasikan menjadi *clickbait*. Sedangkan untuk judul berita yang diklasifikasikan sebagai *nonclickbait*, kata *stopword* muncul sebanyak 2210.
2. Pada hasil evaluasi untuk perbandingan penggunaan fitur *word embedding*, *bag of word*, dan *graph embedding* menunjukkan bahwa klasifikasi dengan menggunakan fitur *bag of word* dan *graph embedding* memiliki nilai lebih baik dibandingkan dengan penggunaan fitur *word embedding*. Pada pengujian dengan menggunakan fitur dari graph embedding, fitur yang paling berpengaruh terhadap hasil dari klasifikasi adalah fitur tag dan kategori berita.

3. Pada analisis pola judul berita menggunakan node2vec, dari hasil visualisasi dapat diketahui bahwa antara kategori dan tag berita yang digunakan antara berita *clickbait* dan *nonclickbait* tidak memiliki perbedaan yang signifikan. Sehingga menyebabkan hasil evaluasi dengan penggunaan fitur yang disebutkan menghasilkan nilai performa yang kurang maksimal. Berdasarkan penghitungan jumlah kategori yang sering muncul pada berita *clickbait*, diperoleh 5 kategori yang sering muncul yakni regional, nasional, gaya hidup, ekonomi, dan peristiwa. Sedangkan untuk tag berita yang sering muncul pada berita *clickbait* meliputi jokowi, prabowo, kesehatan, viral, polisi, kpk, pilpres, debat pilpres, bmkg, debat capres. Berita yang kebanyakan digunakan sebagai *clickbait* biasanya tergantung pada momen-momen berita tersebut diterbitkan. Pada penelitian ini, dataset yang digunakan adalah berita yang diunggah pada momen pemilihan umum presiden, sehingga hal ini menyebabkan beberapa tag berita tersebut kebanyakan berkaitan dengan momen tersebut. Sehingga, berita *clickbait* yang dibuat tidak memiliki pola yang statis tetapi tergantung pada momen-momen besar atau viral yang sedang terjadi. Berdasarkan dari hasil pengecekan *similarity* pada berita *clickbait*, diketahui bahwa 40% dari berita kategori *clickbait* memiliki *similarity* dengan berita *nonclickbait*.

5.2 Saran

Berdasarkan permasalahan yang muncul pada saat melakukan penelitian terdapat saran yang dapat dijadikan pertimbangan untuk melakukan penelitian selanjutnya. Saran penelitian tersebut aalah sebagai berikut:

1. Dataset *clickbait* yang digunakan tidak terbatas pada penyaringan kata kunci yang ditentukan pada penelitian ini.
2. Pengelompokan lebih umum untuk data tag berita dan kategori sehingga tag berita yang memiliki konteks yang sama dapat dikelompokkan menjadi satu.

DAFTAR PUSTAKA

Agrawal, A., 2016. Clickbait Detection using Deep Learning. In: *2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016)*. Dehradun, pp.268–272.

Anand, A., Chakraborty, T. and Park, N., 2017. We used neural networks to detect clickbaits: You won't believe what happened next! In: *Jose J. et al. (eds) Advances in Information Retrieval. ECIR 2017. Lecture Notes in Computer Science*. Springer, Cham, pp.541–547.

Binkhonain, M. and Zhao, L., 2019. A review of machine learning algorithms for identification and classification of non-functional requirements. *Expert Systems with Applications: X*, [online] 1, p.100001. Available at: <https://linkinghub.elsevier.com/retrieve/pii/S2590188519300010>.

Biyani, P., Tsioutsiouliklis, K. and Blackmer, J., 2016. "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*. pp.94–100.

Cao, X., Le, T., Jason and Zhang, 2017. Machine Learning Based Detection of Clickbait Posts in Social Media. [online] Available at: <http://arxiv.org/abs/1710.01977>.

Chakraborty, A., Paranjape, B., Kakarla, S. and Ganguly, N., 2016. Stop Clickbait : Detecting and Preventing Clickbaits in Online News Media. In: *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. San Francisco: IEEE.

Dewan Pers, 2019. *Data Perusahaan Pers*. [online] Available at: <https://dewanpers.or.id/data/perusahaanpers> [Accessed 24 Apr. 2019].

Dong, M., Yao, L., Wang, X., Benatallah, B. and Huang, C., 2019. Similarity-Aware Deep Attentive Model for Clickbait Detection. In: *Advances in Knowledge Discovery and Data Mining*. [online] Springer, Cham. Available at: <http://dx.doi.org/10.1007/978-3-030-16145-3_5>.

Fu, J., Liang, L., Zhou, X. and Zheng, J., 2017. A Convolutional Neural Network

for Clickbait Detection. In: *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, pp.6–10.

Gholami, R. and Fakhari, N., 2017. *Support Vector Machine: Principles, Parameters, and Applications*. 1st ed. [online] *Handbook of Neural Computation*. Elsevier Inc. Available at: <http://dx.doi.org/10.1016/B978-0-12-811318-9.00027-2>.

Goyal, P. and Ferrara, E., 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, [online] 151, pp.78–94. Available at: <https://doi.org/10.1016/j.knosys.2018.03.022>.

Grover, A. and Leskovec, J., 2016. Node2Vec: Scalable Feature Learning for Networks. In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Hadiyat, Y.D., 2019. Clickbait di Media Online Indonesia. *Pekommas*, 4(1), pp.1–10.

Han, J., Kamber, M. and Pei, J., 2012. *Data Mining Concepts and Techniques Third Edition*. Elsevier Inc.

Hootsuit and We Are Social, 2019. *Digital 2019: Indonesia*. [online] Available at: <https://datareportal.com/reports/digital-2019-indonesia> [Accessed 28 Apr. 2019].

Hsu, C.-W., Chang, C.-C. and Lin, C.-J., 2008. A Practical Guide to Support Vector Classification and Regression. *BJU international*, [online] 101(1), pp.1396–1400. Available at: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.

katadata.co.id, 2018a. *Bagaimana Kepercayaan Publik Terhadap Media?* [online] Available at: <https://databoks.katadata.co.id/datapublish/2018/05/08/bagaimana-kepercayaan-publik-terhadap-media> [Accessed 24 Apr. 2019].

katadata.co.id, 2018b. *Kepercayaan Publik Terhadap Media Sosial Pada 2018 Turun*. [online] Available at: <https://databoks.katadata.co.id/datapublish/2018/05/15/kepercayaan-publik-terhadap-media-sosial-pada-2018-turun> [Accessed 24 Apr. 2019].

Kumar, V., Khattar, D., Gairola, S., Lal, Y.K. and Varma, V., 2017. Identifying Clickbait: A Multi-Strategy Approach Using Neural Networks. [online] pp.1225–1228. Available at: <http://arxiv.org/abs/1710.01507%0Ahttp://dx.doi.org/10.1145/3209978.3210144>.

Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X. and Chen, E., 2015. Word Embedding Revisited: A New RepresentationLearning and Explicit Matrix Factorization Perspective. In: *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*. pp.3650–3656.

Loewenstein, G., 1994. The Psychology of Curiosity: A Review and Reinterpretation. *Psychological Bulletin*, 116(1), pp.75–98.

LSI Denny JA, 2018. *LSI Denny JA: Mayoritas Masyarakat Khawatir Terhadap Hoaks*. [online] Available at: <https://databoks.katadata.co.id/datapublish/2018/10/24/lsi-denny-ja-mayoritas-masyarakat-khawatir-terhadap-hoaks> [Accessed 28 Apr. 2019].

Maulidi, R. and Palandi, J.F., 2018. Penerapan Neural Network Backpropagation untuk Klasifikasi Artikel Clickbait.

Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. [online] pp.1–12. Available at: <http://arxiv.org/abs/1301.3781>.

Neo4J, 2019. *The Random Walk Algorithm*. [online] Available at: <https://neo4j.com/docs/graph-algorithms/current/algorithms/random-walk/> [Accessed 12 May 2019].

Pandey, S. and Kaur, G., 2018. Curious to Click It?-Identifying Clickbait using Deep Learning and Evolutionary Algorithm. In: *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, pp.1481–1487.

Potthast, M., Köpsel, S., Stein, B. and Hagen, M., 2016. Clickbait Detection. In: *Ferro N. et al. (eds) Advances in Information Retrieval. ECIR 2016. Lecture Notes in Computer Science*. Springer, Cham, pp.810–817.

Reis, J., Benevenuto, F., de Melo, P.O.S.V., Prates, R., Kwak, H. and An, J., 2015.

Breaking the News: First Impressions Matter on Online News. *CoRR*, [online] abs/1503.0. Available at: <http://arxiv.org/abs/1503.07921>.

Rony, M.M.U., Hassan, N. and Yousuf, M., 2017. Diving Deep into Clickbaits: Who Use Them to What Extents in Which Topics with What Effects? In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. [online] ACM, pp.232--239. Available at: <http://arxiv.org/abs/1703.09400>.

Shu, K., Wang, S., Le, T., Lee, D. and Liu, H., 2018. Deep Headline Generation for Clickbait Detection.

Wargadiredja, A.T., 2017. *ClickUnbait Memerangi Judul Artikel Bombastis di Jagat Media Daring Kita*. [online] Available at: <https://www.vice.com/id_id/article/a378mz/clickunbait-memerangi-judul-artikel-bombastis-di-jagat-media-daring-kita> [Accessed 20 May 2019].

Wongsap, N., Prapphan, T., Lou, L., Kongyoung, S., Jumun, S. and Kaothanthong, N., 2018. Thai Clickbait Headline News Classification and its Characteristic. In: *2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES)*. IEEE, pp.1–6.

Yavi, A.F., 2018. Klasifikasi Artikel Berbahasa Indonesia untuk Mendeteksi Clickbait menggunakan metode Naive Bayes. In: *Journal of Information and Technology*. pp.141–147.

BIODATA PENULIS

Nurrida Aini Zuhroh, lahir di Malang pada tanggal 28 Mei 1995. Penulis telah menempuh pendidikan formal di SDN 3 Sukowilangun Malang, SMPN 2 Sumberpucung Malang, dan SMK Telkom Malang. Pada tahun 2012, penulis melanjutkan pendidikan Strata-1 di Universitas Telkom, dengan jurusan Sistem Informasi dan kelompok keahlian *Enterprise System Development*. Penulis lulus pada tahun 2016 dengan Tugas Akhir berjudul "Membangun Aplikasi Kemahasiswaan Berbasis Web Modul Pengelolaan Kegiatan Himpunan pada Sisi Kemahasiswaan menggunakan Metode *Iterative* dan *Incremental*". Pada penelitian ini, penulis mengambil konsentrasi dalam Akuisisi Data dan Diseminasi Informasi (ADDI) dengan topik penelitian mengenai *text mining* pada judul berita bahasa Indonesia. Segala kritik dan saran yang membangun dapat disampaikan melalui nurrida.aini@gmail.com.