P-ISSN : 2622-1594
E-ISSN : 2685-449X

Klasifikasi Teks Humor Bahasa Indonesia Memanfaatkan SVM

Jonie Hermanto
Departemen Sistem Informasi, Institut Sains dan Teknologi Terpadu Surabaya

***Abstrak*—Humor merupakan sebuah topik menarik dari riset di area *Natural Language Processing*, dalam dunia humor banyak sekali ditemukan tipe-tipe humor yang bervariasi namun dengan satu tujuan yaitu menghibur penikmat humor. Tingkat kesulitan pertama dari pengenalan humor adalah perbedaan sense of humor dari tiap orang yang cenderung berbeda beda, kesulitan kedua adalah adanya faktor external seperti situasi yang mempengaruhi derajat atau kadar dari humor tersebut. Berdasarkan hal tersebut, maka terdapat hal-hal yang menjadi kendala selama ini yaitu bagaimana proses dari membedakan kalimat biasa dengan kalimat humor, dan termasuk kategri humor apa jika memang dapat dikatakan kalimat tersebut adalah kalimat humor. Hal ini perlu dilakukan untuk melakukan klasifikasi sehingga dapat ditemukan dan diklasifikasikan secara pasti sehingga semakin mempertajam pengenalan natural language pada komputer dan berguna bagi pengembangan dunia Natural Language Processing ke depan. Klasifikasi terbesar adalah humor verbal dan non verbal, untuk humor non verbal inilah penelitian ini difokuskan, yaitu melakukan klasifikasi humor oneliner, humor sebaris barupa tulisan singkat yang bertujuan menghantarkan sebuah punchline dan premis dalam satu kalimat. Penelitian ini akan berusaha melakukan klasifikasi humor menjadi beberapa kategori dengan menggunakan algoritma SVM dan word2vec, klasifikasi ini nantinya diharapkan memisahkan jenis-jenis humor oneliner menjadi 5 (lima) kategori tipe class humor sesuai cara penyajiannya, dengan dataset yang didapat sekitar 4000 data dari komedian-komedian profesional dan dilakukan proses pengenalan manual oleh ahli di bidangnya. Pada penelitian ini dilakukan skenario uji menggunakan K-Fold Cross Validation dan juga melakukan Persentase Splitting Distribusi antara data training dan testing sehingga dari skenario tersebut mendapatkan nilai akurasi 81%, sehingga penelitian ini dapat dikatakan cukup baik untuk menemukan beberapa kelas humor yang akan menjadi cikal bakal pengenalan komputerisasi humor berbahasa Indonesia.**

***Kata Kunci*—Klasifikasi Teks, Humor Bahasa Indonesia, SVM.**

I. PENDAHULUAN

Penulisan dalam penelitian ini diawali dengan pendahuluan yang berisi tentang gambaran secara singkat mengenai isi penelitian sekaligus memberikan rambu - rambu untuk masuk pada bab-bab berikutnya. Bab ini menjelaskan latar belakang masalah, tujuan dan manfaat penelitian, hipotesis, batasan masalah, dan sistematika pembahasan.

A. *Latar Belakang*

Humor merupakan sebuah topik menarik dari riset di area Natural Language Processing, semakin berkembangnya alogoritma pengenalan bahasa natural memungkinkan komputer untuk lebih memahami makna dari sebuah kata atau kalimat yang diucapkan, dengan dasar algoritma ini maka pengenalan apakah dalam sebuah kalimat mengandung kalimat humor bukanlah sesuatu yang mustahil untuk dilakukan. Tujuan dari pengenalan humor adalah menentukan apakah sebuah kalimat ini termasuk humor atau bukan dengan derajat humor tertentu, kemudian mengklasifikasikan humor yang dikenali sebagai kategori humor apa. Tingkat kesulitan pertama dari pengenalam humor adalah perbedaan sense of humor dari tiap orang yang cenderung berbeda beda, kesulitan kedua adalah adanya faktor external seperti situasi yang mempengaruhi derajat atau kadar dari humor tersebut, sebagai contoh adalah kalimat berikut "*Pak jokowi itu masih saudara dengan saya...setiap ketemu saya beliau selalu bertanya saudara ini siapa ya??*" situasi yang mendasari kalimat diatas termasuk kategori humor adalah bapak Jokowi adalah orang terkenal (Presiden Republik Indonesia).

Menurut penelitan yang pernah dilakukan oleh beberapa penelitian sebelumnya menyatakan bahwa ada tiga pola humor yaitu wordplay, sarcasm, dan irony dan secara general masih banyak ketagori lainnya, namun masih tidak memungkinkan untuk dikelompokkan karena manusia pun masih susah mengenalinya. Dalam riset ini akan dilakukan formulasi sebuah kalimat termasuk humor atau tidak, kemudian menklasifikasikan ke dalam tiga kategori wordplay atau permainan kata, sarcasm atau humor sarkas, dan irony ataupun humor kritik.

B. *Tujuan dan Manfaat*

Adapun tujuan dan manfaat dari penelitian ini adalah menemukan pola kalimat "humor" dan apa yang menyebabkan kalimat ini menjadi sebuah kalimat humor. Serta mengenali humor yang ditemukan termasuk kategori apa, hal ini semakin mempertajam pengenalan natural language pada komputer dan berguna bagi pengembangan dunia Natural Language Processing ke depan. Sehingga para seniman humor utamanya dapat membuat konsep humor dengan lebih tertata rapi.

C. *Hipotesis*

Dalam penelitian ini diharapkan bisa ditemukan pola sebuah kalimat humor dan klasifikasinya bisa dikenali lewat algoritma SVM dan word2vec serta komputer akan mempu

DOI : 10.37823/insight.v3i01.118

mengenali humor dalam bahasa indonesia yang diinputkan, klasifikasi humor dibagi 4 yaitu humor trilogy, humor pembelokan logika, humor plesetan, dan bukan humor dengan target akurasi 65%.

D. Batasan Masalah

Beberapa hal yang perlu diperhatikan pada penelitian berikut adalah batasan dari ruang lingkup penelitian, yang mana batasan dari penelitian adalah berikut ini :

Target yang akan dikenali adalah kumpulan humor pendek berbahasa Indonesia,target yang akan dikenali adalah (humor trilogy, humor sarkastik, humor permainan kata atau plesetan, humor homonym dan humor teka teki, classifier yang akan digunakan hanya Support Vector Machine, Target Akurasi adalah 65%

II. DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori apa saja yang menjadi dasar penelitian klasifikasi teks humor pada Penelitian ini.

A. Humor Recognition and Humor Anchor Extraction

Penulis : Diyi Yang, Alon Lavie, Chris Dyer, Eduard Hovy
Penerbit : Language Technologies Institute, School of Computer Science Carnegie Mellon University.Pittsburgh, PA, 15213, USA
Tahun : 2008

Humor adalah salah satu bagian terpenting dan area puzzle penelitian dalam pemahaman bahasa secara alamiah. Baru-baru ini, komputer telah memiliki peran baru dari yang awalnya hanya dapat menyelesaikan tugas menjadi suatu alat yang cerdas bahkan sekarang lebih berinteraksi secara dinamis dengan orang dan belajar memahami penggunanya. Ketika komputer berhasil melakukan konversi dengan manusia, maka dia dapat menampilkan humor dalam bahasa manusia, ini dapat dipahami dengan baik beserta makna yang benar dari bahasa manusia, dan dengan demikian dapat membuat keputusan yang lebih baik guna meningkatkan *user experience*.

Humor recognition bertugas menentukan kalimat mana dalam suatu konteks yang mengungkapkan ekspresi tingkat kelucuan tertentu. *Humor recognition* merupakan sebuah tantangan dalam permasalahan bahasa alami (Attardo, 1994). Pertama, definisi humor universal sulit untuk dicapai, karena orang yang berbeda memegang pemahaman yang berbeda dari bahkan kalimat yang sama sekalipun. Kedua, humor selalu terletak di konteks yang lebih luas bahkan kadang-kadang membutuhkan banyak pengetahuan eksternal untuk dapat mehamic sepenuhnya, contoh : *"The one who invented the door knocker got a No Bell prize" and "Veni, Vidi, Visa: I came, I saw, I did a little shopping"*. Dibutuhkan Konteks Budaya yang lebih besar untuk menunjukkan tingkat kelucuan yang sangat halus dengan mengekspresikan arti dari 2 kalimat tersebut (Raz, 2012), semisal permainan kata, ironi dan sarkasme, akan tetapi disana terdapat beberapa karakterisktik taksonomi formal. Jadi, hampir tidak mungkin untuk merancang algoritma umum yang dapat mengklasifikasikan semua jenis humor, bahkan manusia sekalipun tidak bisa sempurna mengklasifikasi semua itu. Meskipun, ini sangat tidak mungkin untuk melakukan pemahaman terhadap karakteristik humor universal, tetapi kemungkinan masih bisa menunjukkan struktur laten dibalik humor tersebut (Bucaria, 2004; Binstead and Ritchie, 1997). Dalam hal ini, diungkap beberapa struktur laten semantik dibalik humor, memaknai keganjilan, kemabiguan, model fonetik dan pengaruh personal. Sebagai tambahan *Humor Recognition* , melakukan identifikasi *Anchor*, atau menunjukkan kata-kata mana dalam kalimat yang mengandung humor, sehingga termasuk penting untuk memahami fonem dari bahasa humor. *Anchor Extraction* mengacu pada ekstraksi unit semantik (Kata kunci atau frase) yang memungkinkan memiliki nilai humor dalam kalimat. Dalam hal ini di formulasikan *Humor Recognition* sebagai klasifikasi tugas dimana dapat membedakan antara lucu dan tidak lucu, kemudian di eksplorasi latar belakang struktur semantik humornya dari 4 hal: keganjilan/*Incongruity*, keambiguan/ *ambiguity*,efek interpersonal/*interpersonal effect*, dan model fonetik/*phonetic style*.

B. Characterizing Humour: An Exploration of Feature in Humorous Texts

Penulis : Mr. A. Mallikarjuna, Dr. G. Anjan Babu
Penerbit : International Journal of Engineering Research & Technology (IJERT)
Tahun : 11 November 2013

Dalam penelitian ini dikumpulkan artikel yang diterbitkan pada Agustus 2005 – Maret 2006. Yang mana dihasilkan dalam dataset sekitar 2500 artikel berita. Dalam penelitian ini mencoba mengidentifikasi dan mengklasifikasi yang berbasis konten. Dengan memeriksa menggunakan tangan fitur yang berbasis konten yang paling diskrimintif dipelajari selama proses klasifikasi teks, dalam penelitian ini mencoba mengklasifikasikan fitur yang berbasis kontetn tersebut kedalam kelas semantik. Berikut merukan kelas kata yang sering muncul yaitu; *human centric vocabulary, negation* (penolakan), orientasi *negative*, masyarakat professional, dan *human "weakness"*.

Dimulai dengan dataset dijelaskan bahwa orientasi "positif" dan "negatif", penelitian ini menerapkan sistem pengklasifikasian kation yang memiliki kemampuan secara otomatis yang menunjukkan semantik orientasi teks. Secara khusus menggunakan dataset dari 10.662 teks fragmen singkat yang diperkenalkan pada porsi 5331 "positif" dan 5331 "negatif" fragmen dalam pengklasifikasian Naif Bayes. Pada percobaan lintas validasi yaitu menghasilkan sepuluh kali lipat, akurasi dari sistem ditentukan adalah 78,15%, hasil tersebut lebih baik dibandingkan dengan sebelumnya yaitu hasil yang dilaporkan pada dataset yang sama. Menggunakan alat analisis sentimen ini, kita secara otomatis mendapat dua keterangan humor dataset. Menariknya, teks biasa juga cenderung menuju negatif, dengan 56,26% dari campuran kalimat "serius" yang ditentukan sebagai memiliki orientasi negatif. Umumnya, kalimat "serius" pada artikel berita bahkan lebih negatif, dengan 67,60%.Menariknya, dengan menganalisis anotasi, beberapa contoh label sebagai positif tampaknya termasuk kata-kata dengan orientasi negatif, yang kekuatannya adalah mungkin tidak cukup tinggi untuk dipilih sebagai negatif oleh pengklasifikasi otomatis.

 DOI : 10.37823/insight.v3i01.118

C. Making Computer Laugh : Investigations in Automatic Humor Recognition

Penulis : Rada Mihalcea, Carlo Strapparava
Penerbit : International Journal of Engineering Research & Technology (IJERT)
Tahun : 11 November 2013

Humor merupakan elemen penting dalam komunikasi pribadi. Padahal humor hanya dianggap sebagai cara untuk menginduksi hiburan, humor juga memiliki efek positif pada keadaan mental mereka yang menggunakan dan memiliki kemampuan untuk meningkatkan aktivitas mereka. Oleh karena itu komputasi humor layak mendapatkan perhatian khusus, karena memiliki potensi mengubah komputer menjadi kreatif dan alat motivasi untuk aktivitas manusia. Tulisan mengeksplorasi penerapan pendekatan komputasi untuk penggunaan humor verbal. Secara khusus, diteliti apakah teknik klasifikasi otomatis dapat membedakan antara humor dan teks non-humor, dengan disertakan bukti empiris untuk mendukung hipotesis melalui eksperimen dilakukan pada set data yang sangat besar.

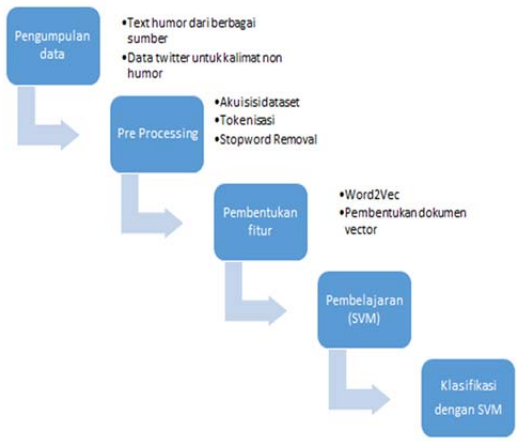
Pembatasan penelitian yang digunakan hanya pada jenis humor *One liners*, dimana merupakan kalimat pendek dengan efek komik dan linguistik struktur yang menarik : sintax yang sederhana, sengaja menggunakan perangkat retorika (Misal aliterasi, rima), dan penggunaan konstruksi bahasa untuk menarik perhatian pembaca. Secara khusus, dalam penelitian ini digunakan 3 dataset negatif yang berbeda: (1) *Reuters* Judul Berita, (2) Peribahasa, (3) Kalaimat dati British National Coorpus (BNC) Hasil klasifikasi ditunjukkan dengan akurasi angka-angka mulai dari 79,15% (One-liners/BNC) ke 96,95% (*One-liners/Reuters*).

Evaluasi menggunakan sepuluh kali lipat dengan nilai fold 10. Dasar untuk semua percobaan adalah 50%, yang merupakan akurasi klasifikasi yang diperoleh jika label dari "humor" (atau "non-humor") akan ditugaskan secara default untuk semua contoh dalam kumpulan data.

- Heuristics using Humor-specific Features
  Dalam satu set percobaan pertama, mengevaluasi akurasi klasifikasi menggunakan fitur gaya humor spesifik: aliterasi, antonimi, dan gaul/dewasa. threshold ini dipelajari secara otomatis menggunakan pohon keputusan yang diterapkan pada subset kecil dari contoh humor/non-humor (1000 contoh).
- Text Classification with Content Feature
  Percobaan fokus kepada evaluasi fitur berbasis konten untuk Humor Recognition. Hal ini menunjukkan hasil yang diperoleh dengan menggunakan tiga set yang berbeda dari contoh negatif, dengan Naive Bayes dan SVM Teks klasifikasi. Konten humors cenderung sangat mirip dengan teks biasa, meskipun Perbedaan yang cukup akurat masih dapat dibuat dengan menggunakan teknik klasifikasi teks.
- Combining Stylistic and Content Features
  Berdasarkan hasil yang diperoleh dari dua percobaan, maka dirancang percobaan ketiga yang mencoba untuk bersama-sama memanfaatkan fitur gaya dan konten untuk Humor *Recognition*. Fitur Kombinasi dilakukan dengan menggunakan pembelajaran yang ditumpuk, yang mengambil output dari klasifikasi teks, hal itu digabungkan dengan tiga fitur humor-spesifik (aliterasi, antonimi, gaul/dewasa), dan *feeds* baru dibuat menjadi Fitur vektor untuk *machine learning*. mengingat kesenjangan yang relatif besar antara kinerja yang dicapai dengan fitur berbasis konten (teks klasifikasi) dan fitur gaya (humor spesifik heuristik), maka diputuskan untuk melaksanakan pembelajaran tahap kedua dimana pembelajar ditumpuk menggunakan memori berbasis sistem pembelajaran, sehingga fitur kinerja rendah tidak dieliminasi dalam mendukung hasil yang lebih akurat. sehingga digunakanlah *Timbl memory based learner* (Daelemans et al., 2001), dan mengevaluasi klasifikasi menggunakan sepuluh kali lipat lintas. Tingkatan validasi.
  Hasil kombinasi klasifikasi dalam perbaikan tidak bisa secara statistik signifikan (p<0,0005, tes pemasangan t-test) sehubungan dengan klasifikasi terbaik untuk One-liners/Reuters dan One-liners/BNC set data, dengan pengurangan tingkat kesalahan masing-masing relatif 8,9% dan 7,3%.

III. METODE PENELITIAN

Pada bagian ini akan dibahas metode penelitian dari system yang menggambarkan kerangka atau pola dasar dari proses pengenalan humor yang dibangun.



Gambar. 1. Arsitektur dasar

Pada gambar 1 menunjukkan arsitektur dasar dari sistem yang di bangun. Diawali dengan pengumpulan data dari berbagai comedian dan dari twitter untuk data non humor kemudian dilakukan pra proses, dilanjutkan dengan pembentukan fitur, output dari proses ini digunakan untuk melakukan proses klasifikasi dengan SVM.

A. Pengumpulan Data

Pada tahap awal dilakukan pengumpulan data sebagai bahan masukan untuk penelitian ini. Data yang dicari merupakan data humor one liner dengan tipe data berupa teks. Data akan diambil dari berbagai sumber komedian di indonesia. Data yang didapat mencakup humor dalam berbagai tipe kategori aliran.

Total data humor yang diperoleh yaitu 460.868 yang didapat dari 8 komedian berbeda dengan pengambilan secara acak. Rincian jumlah humor dapat dilihat pada Tabel I berikut.

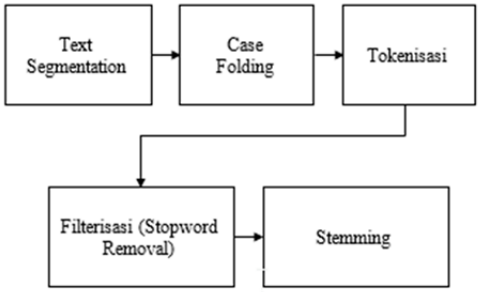
DOI : 10.37823/insight.v3i01.118

TABEL I
RINCIAN JUMLAH HUMOR

Komedian	Jumlah Humor
Tatok	41.974
Ribut Srimulat	37.139
Sule	35.844
Jarwo Kwat	32.977
Isa	25.876
Indro Warkop	24.027
Dargombes Tugu Pahlawan	22.862
Djadi Galajapo	22.571

B. *Pra Proses Data*

Selanjutnya dilakukan tahap praproses data yang diperoleh dari tahap sebelumnya. Tahapan ini merupakan bagian dari tahapan Pra Proses, dimana data akan dilakukan dengan proses akuisisi dataset untuk memecah teks yang akan diambil bagian humornya saja, dari hasil tersebut kemudian akan dilakukan case folding yang mana akan merubah semua kata humor yang diperoleh menjadi berhuruf kecil bahkan karakter lain yang bukan huruf akan dihilangkan. Hasil dari proses tersebut kemudian dilakukan tokenisasi dan filterisasi dengan stopword removal.



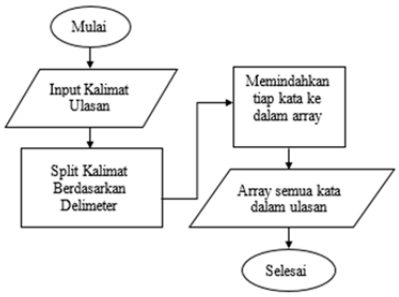
Gambar. 2. Blok Diagram Pra Process

Point utama pada tahapan praproses adalah tokenisasi dan filterisasi, walaupun sebenarnya keseluruhan pra proses ini terdiri teks segmentation untuk memecah teks yang utuh ke dalam setiap baris kalimat dan kemudian case folding untuk mengubah semua kata dalam teks ini menjadi berhuruf kecil sedangkan karakter selain huruf akan dihilangkan. Data yang diperoleh dari proses sebelumnya itu baru akan dilakukan proses tokenisasi dan filterisasi yang kemudian akan dilakukan proses stemming untuk mengembalikan kata ke dalam kata dasarnya. Semua proses ini dilakukan bertujuan untuk mendapatkan bentuk data yang diinginkan sebelum masuk ketahap implementasi. Adapun proses utama dan paling penting dalam pra proses ini adalah proses tokenisasi dan filterisasi, dimana penjelasannya sebagai berikut:

1) *Tokenisasi*

Proses ini dilakukan untuk menghasilkan atau pembentukan *array* dari kata-kata yang ada di daftar humor one liners yang didapat atau dengan kata lain memecah kalimat menjadi token-token atau kata-kata tunggal. Hal ini bertujuan agar data array hasil dari tokenisasi tersebut dapat dijadikan sebagai data masukan pada algoritma yang akan digunakan pada penelitian ini.

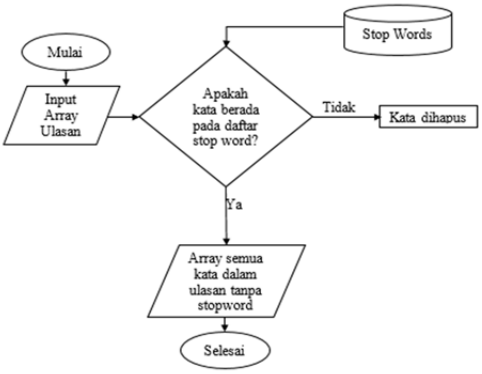
Pada tahap tokenisasi dilakukan suatu proses pemecahan tiap kalimat ulasan menjadi bentuk kata-kata yang terpisah satu sama lain atau disebut juga dengan token. Hal in bertujuan agar dapat diamati makna setiap kata dalam humor onliners sehingga dapat ditentukan apakah humor tersebut termasuk kategori yang mana seperti pada Gambar 3 menunjukkan diagram alir proses tokenisasi pada suatu kalimat humor.



Gambar. 3. Diagram Alir Proses Tokenisasi

2) *Filterisasi*

Data-data humor yang didapat pasti akan mengandung karakter, kata-kata yang tidak memiliki makna atau bahkan juga dapat ditemukan kata-kata yang tidak baku, kata henti, dan kata penghubung. Semua kata-kata atau hal-hal tersebut diatas akan membuat data humor sulit untuk diolah ke tahapan yang lebih lanjut. Oleh karena itu, pada tahapan ini harus dilakukan penghilangan kata-kata tersebut, sehingga untuk memperingkas pemecahannya atau filterisasi ini dilakukan berdasarkan tanda spasi. Hal ini dilakukan agar sistem dapat memproses data dengan lebih efektif dan efisien. Adapun Diagram alir yang menunjukkan tentang proses filterisasi yang akan dilakukan oleh sistem ditunjukkan seperti Gambar 4 berikut.



Gambar. 4. Diagram Alir Proses Filterisasi

Sedangkan Daftar data stopwords yang digunakan untuk tahapan filterisasi ini ada sebanyak 793 yang terdiri dari karakter atau kata yang tidak memiliki makna, adapun contoh dari daftar stopword tersebut dapat dilihat pada tabel II berikut ini.

TABEL II
CONTOH DAFTAR DATA STOPWORDS

No	Stopwords
1	Â !
2	#
3	$
…	…
793	yang

DOI : 10.37823/insight.v3i01.118

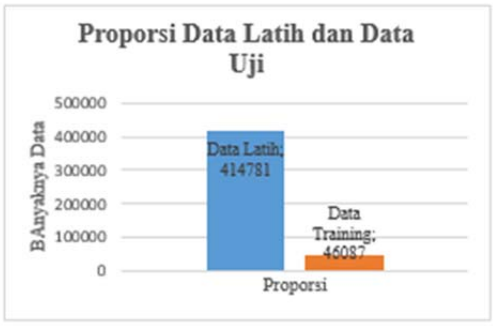
C. *Ekstraksi Fitur*

Dalam pengolahan teks diperlukan ekstraksi kata menjadi numerik karena berdasarkan prinsipnya komputer saat ini tidak dapat mengolah data teks kecuali data yang bersifat numerik, sehingga ekstraksi fitur ini dapat digunakan untuk menggali informasi-informasi yang penting dan dapat dikatakan potensial yang direpresentasikan sebagai vektor fitur.

Pada tahap ini setiap kata dirubah menjadi representasi vektor yang mewakili polaritas suatu kata. Pada saat ini terdapat beberapa macam metode untuk menjadikan kata menjadi suatu vektor. Salah satu metode yang sedang berkembang sekarang yaitu metode *Word To Vec,* Metode ini bertugas untuk merubah sebuah kata menjadi suatu vektor. Vektor yang dihasilkan merupakan konteks dari kata tersebut yang memperhatikan peluang kata yang muncul disekitarnya. Pada perkembangannya *Word To Vec* telah dijadikan sebagai dasar dari penelitian NLP. *Word2vec* ini juga memiliki beberapa jenis antara lain *Bag of Word* (BOW), *Continuous Bag of Word* (CBOW), *Skip gram negative sampling*, atau yang terbaru yaitu *Global Vector* (GloVe). Dengan vektor ini, dapat dapat diolah dengan mudah untuk proses selanjutnya.

D. *Pemisahan Data Testing dan Training*

Tahapan selanjutnya pada arsitektur penelitian ini adalah melakukan pemisahan data yaitu antara data training dan data testing. Dimana, pada penelitian ini akan digunakan data latih atau data training sebanyak 90%, sedangkan data uji atau data testing sebanyak 10% dari total data keseluruhan humor 460.868 record.



Gambar. 5. Proporsi Data Latih dan Data Training

Dari gambar 5 telah ditunjukkan proporsi pembagian data latih dan data uji, dimana dari keseluruhan data yang diperoleh maka sebanyak 414.718 record data humor akan digunakan sebagai data latih, sedangkan sebanyak 46.868 record data humor digunakan sebagai data latih yang contohnya dapat dilihat pada tabel 3.4. Sehingga kesemuanya itu memenuhi persentase 90:10 antara data latih dan data uji.

E. *Library*

Pada penelitian ini digunakan *library* yang ada pada Python sebagai pembentuk model *deep learning* yang akan dibangun. Terdapat tiga *library* utama yang digunakan pada penelitian ini, antara lain *Keras*, *Sckit-learn*, dan *Gensim*. *Library* tersebut menyediakan fungsi-fungsi untuk dapat membangun model *deep learning* maupun SVM dengan mudah. Adapun penjelasan dan kegunaan dari 3 (tiga) library utama yang digunakan pada penelitian ini adalah sebagai berikut:

1) *Library Keras* merupakan salah satu *library* yang khusus dibuat untuk mengolah data dengan menggunakan model *deep learning*.
2) Kemudian, untuk membentuk model jaringan SVM digunakan library bernama *Sckit-learn*.
   Sama halnya dengan *Keras*, library ini dibuat khusus untuk membentuk model jaringan syaraf tiruan dengan salah satu modelnya yaitu SVM.
3) Dan yang terakhir, untuk dapat merubah kata menjadi vektor maka digunakan *library Gensim*.
   Library tersebut menyediakan model *Word To Vec* sehingga dapat digunakan dengan mudah untuk merubah kata menjadi representasi vektor kata.

IV. PERANCANGAN DAN IMPLEMENTASI

Dalam penerapan Penelitian ini dibutuhkan data sebagai bahan masukan dari algoritma yang akan dijalankan. Data yang digunakan dalam penelitian kali ini menggunakan dataset yang dikumpulkan secara manual dari sumber pelaku bisnis komedi (pelawak) di indonesia. Dataset terdiri dari humor pendek dalam berbagai kategori. Tiap masukkan pada dataset disajikan dalam row excel dan diberikan pelabelan manual untuk menentukan kolompok kategori dari humor one liner yang berhasil dikumpulkan. Pengumpulan data non humor didapat dari data twitter yang juga berupa statement kalimat pendek. *Dataset* ini disimpan dalam bentuk *file csv* untuk mempermudah penyimpanan dan penggunaan. Supaya dataset dapat digunakan secara optimal, maka dilakukan pengurangan atribut data dan menghilangkannya. Data skor dari ulasan dipisahkan dari dataset kemudian disimpan di lain dokumen, sedangkan beberapa model humor yang tidak memiliki banyak sampling data dikeluarkan dari contoh dataset.

Selanjutnya, dikarenakan data skor yang didapatkan merupakan skor dari 0 sampai 4, maka data dirubah menjadi 5 kelas yaitu :

1) 0 : Humor presepsi linguistik (humor dari presepsi bahasa multi tafsir)
2) 1: humor missdefinitif (humor dari definisi yang disalah artikan)
3) 2: humor comparison (humor dari membandingkan hal yang berlawanan maupun senada namun salah tujuan)
4) 3: humor missleading quote (humor quote " kata2 motivasi " yang disalah arti)
5) 4: bukan humor

Data diluar kategori diatas dikesampingkan karena hal ini juga mengatisipasi adanya kerancuan dalam mengklasifikasi ulasan. Untuk menghindari ketidakseimbangan data, maka data yang digunakan harus berdistibusi secara seragam agar diperoleh hasil yang maksimal. Oleh karena itu, dipilih salah satu dataset yang memiliki sebaran data yang paling baik dari 8 dataset humor tersebut.

Data humor yang didapat tidak dapat langsung digunakan, hal ini dikarenakan data tersebut masih berupa kalimat utuh sehingga diperlukan tahapan pra proses dengan tujuan mendapat data yang siap diolah.

Fase pra-proses merupakan fase yang penting untuk menentukan kualitas proses selanjutnya). Tujuan utama fase

DOI : 10.37823/insight.v3i01.118

pra-proses adalah untuk mendapatkan bentuk data siap olah untuk diproses oleh data mining dari data awal yang berupa data tekstual. Fitur-fitur fase pra-proses terdiri dari beberapa tahap, dimulai dari pemilihan humor dalam dokumen yang digunakan (humor yang mengandung kata kata umum dihilangkan).

Proses klasifikasi dengan menggunakan SVM ini pada prinsipnya mencari *hyperplane* atau garis yang memisahkan data antar kelas atau kategorinya dan yang memiliki margin terbesar.

Dua tahap praproses data yaitu tahap tokenisasi dan filterisasi.

1) Tokenisasi

Pada tahap tokenisasi dilakukan suatu proses pemecahan tiap kalimat ulasan menjadi bentuk kata-kata yang terpisah satu sama lain atau disebut juga dengan token. Hal ini bertujuan agar dapat diamati makna setiap kata dalam humor onliners sehingga dapat ditentukan apakah humor tersebut termasuk kategori 0,1,2,3 atau 4.

2) Filterisasi

Proses selanjutnya pada praproses data yaitu tahap filterisasi. Pada tahap ini kata-kata atau karakter yang tidak memiliki arti seperti kata penghubung, kata henti, imbuhan dan lain sebagainya dihilangkan dari ulasan. Selain itu kata-kata pada ulasan buku dilakukan proses *case folding* (perubahan huruf capital menjadi huruf kecil) agar dapat. Hal ini dilakukan agar sistem dapat memproses data dengan lebih efektif dan efisien.

Algoritma 1. Preprocessing

```
01: def preprocessing(artikel,stopwords):
02: result = word_tokenize(artikel)
03:     result = [word.lower() for word in result
        if word.lower() not in stopwords]
04:     result = [word for word in result if
        not(word.startswith('@')                and
        word.startswith('http://')              and
        word.startswith('<br        />')       and
        word.startswith('<br>')) ]
05: return result
```

Agar dapat dengan mudah dibentuk implementasi sistem ulasan, maka dalam prosesnya menggunakan modul *Word To Vec* untuk merubah kata menjadi vektor-vektor kata.

Algoritma 2. Implementasi Tahapan Modul WordToVec

```
01: # import genism
02: # import numpy as np
03: # import matplotlib.pyplot as plt
04: # from sklearn.manifold import TSNE
05:     #model                                  =
    gensim.models.Word2Vec.load("wiki.id.word2vec.
    model")
06:     # # word = input("Inputkan kata yang akan
    dicari : ")
07:     # print("Top 10 Similar Words :")
08:     # for d in model.similar_by_word(word):
09:     #  print(d[0]  +  ",  similarity  :  "  +
    str(d[1]))
```

Untuk kata yang tidak berada pada Word to vec data, maka kata dibentuk secara acak, kemudian hasil dari vektor yang ditemukan dilakukan perhitungan rata rata.

Algoritma 3. Implementasi Tahapan Diluar Modul WordToVec

```
01: # mendapatkan vektor kata
02: Embed = model.wv[“buku”]
03: Print(embed)
```

Ketika tahapan klasifikasi dengan word to vec telah dilakukan, maka akan dilakukan selanjutnya dengan tahapan pembobotan kata TF-IDF sebagai representasi kata.

Algoritma 4. Build Dataset Pembobotan

```
01: def
    build_dataset(dataset_path,stopword,word2vec,wo
    rd2vec_dim):
02: x_train = []
03: y_train = []
04: with open(dataset_path,mode='r') as reader:
05: ctr = 0
06: for l in reader:
07: ctr += 1
08: temp = l.split("~")
09: data = temp[0].strip().lower()
10: label = temp[1]
11: #print(str(ctr)+". "+data+"~"+str(label))
12: target = int(label)
13: X = preprocessing(data,stopwords)
14: x_train.append(meanTransform(X,word2vec,word2v
    ec_dim))
15: y_train.append(target)
16: return np.array(x_train),np.array(y_train)
```

Definisi fungsi build dataset diatas mendapat parameter berupa lokasi dari dataset yang digunakan, stopword,word2vec dari fungsi word2vec dan dimensi dari word2vec itu sendiri. Berikut adalah potongan code yang merupakan main atau potongan program utama dimana seluruh proses dijalankanm dalam code ini terjadi proses dari fungsi diatas.

Algoritma dan Segmen Program 5. Main Function

```
01: $status=$this->data_user-
    >checkUserLogin($email, $id);
01: dim = 200
02:     word2vec_model=
        gensim.models.Word2Vec.load("wiki.id.word2v
        ec.model")
03:     stopwords=
        open("stopwords.txt",encoding="utf8").read(
        ).splitlines()
04:     X,y=
        build_dataset("dataset.txt",stopwords,word2
        vec_model,dim)
05: n_fold = 10 #paramter k-fold
06: kf = KFold(n_splits=n_fold)
07: ctr = 0
08: acc = 0.0
09: rec = 0.0
10: prec = 0.0
11: f1 = 0.0
12: for train_index, test_index in kf.split(X):
13:     ctr = ctr + 1
14:     X_train = X[train_index]
15:     Y_train = y[train_index]
16:     X_test = X[test_index]
17:     clf                                    =
    svm.SVC(decision_function_shape='ovr')
18:     clf.fit(X_train, Y_train)
19:     y_pred = clf.predict(X_test)
```

 DOI : 10.37823/insight.v3i01.118

```
20:         temp_acc                                =
        accuracy_score(y_true=y[test_index],y_pred=
        y_pred)
21:         temp_rec                                =
        recall_score(y_true=y[test_index],
        y_pred=y_pred,average='micro')
22:         temp_prec                               =
        accuracy_score(y_true=y[test_index],
        y_pred=y_pred)
23:         temp_f1= f1_score(y_true=y[test_index],
        y_pred=y_pred,average='micro')
24:      acc = acc + temp_acc
25:      rec = rec + temp_rec
26:      prec = prec + temp_prec
27:      f1 = f1 + temp_f1
28:         print("Fold               #"+str(ctr)+".
        Precision/Recall/F1-Score/Accuracy   :"   +
        str(temp_prec)+"/"+str(temp_rec)+"/"+str(te
        mp_f1)+"/"+str(temp_acc))
29:     print("Average  Recall/Precision/F1-Score  :
        "
        +str(prec/ctr)+"/"+str(rec/ctr)+"/"+str(f1/
        ctr) )
30: print("")
31: print("Average Accuracy : " + str(acc/ctr))
32: print("")
```

Pada segmen diatas dilakukan inti dari proses klasifikasi SVM dan print output accuracy dengan beberapa variasi parameter untuk mencari nilai optimal, juga dilakukan nilai iterasi FOLD sesuai nilai dari variable K-Fold.

Untuk menunjukkan tahap implementasi pada proses penelitian maka setiap langkah dari kegiatan pra proses yang dilakukan oleh sistem akan dicontohkan dengan beberapa pengambilan sampel dibawah ini:

Kalimat:

Hasil survei menunjukkan, ternyata 50% orang pacaran itu karena mereka saling jatuh cinta Yang 50% lagi karena mereka saling suka dan sayang~

1) Tokenisasi (memberikan nilai kepada setiap kata untuk mewakili kata tersebut):

0	*Hasil*	10	*saling*	20	*dan*
1	*survei*	11	*jatuh*	21	*sayang*
2	*menunjukkan*	12	*cinta*	22	*~*
3	*ternyata*	13	*yang*		
4	*50%*	14	*50%*		
5	*orang*	15	*lagi*		
6	*pacaran*	16	*karena*		
7	*itu*	17	*mereka*		
8	*karena*	18	*saling*		
9	*mereka*	19	*suka*		

Dari kalimat sebelumnya, maka setiap kata akan dipecah dan diberikan bobot nilai dari angka "0" sampai dengan tak terhingga (mewakili jumlah katanya) sehingga semua kata akan memiliki angka perwakilannya masing masing.

2) Array Ulasan (Hasil dari pembobotan tokenisasi):

```
    0      1          2           3        4      5
["Hasil" "survei" "menunjukkan" "ternyata" "50%" "orang"
    6      7      8         9          10     11      12
"pacaran" "itu" "karena" "mereka" "saling" "jatuh" "cinta"
   13     14     15       16      17       18      19    20
 "yang" "50%" "lagi" "karena" "mereka" "saling" "suka" dan

   21
sayang ~]
```

Gambar. 6. Array Ulasan

Dari tiap kata yang sudah diberikan bobot nilai sebagai perwakilan kata tersebut, kemudian terbentuklah array kata ulasan seperti contoh diatas.

3) Filterisasi (Melakukan cek setiap kata apakah mengandung stopword):

```
    0      1          2           3        4      5
["Hasil" "survei" "menunjukkan" "ternyata" "50%" "orang"
    6      7      8         9          10     11      12
"pacaran" "itu" "karena" "mereka" "saling" "jatuh" "cinta"
   13     14     15       16      17       18      19    20
 "yang" "50%" "lagi" "karena" "mereka" "saling" "suka" "dan"
   21
sayang ~]
```

Gambar. 7. Filterisasi

Semua kata yang sudah menjadi array kata ulasan akan dilakukan pengecekan dengan database stopwords yang sudah ada. Jika terdapat maka kata tersebut akan dihapus dan tidak di proses ke tahapan selanjutnya. Ada beberapa kata yang dihapus berikut ini:

TABEL III
DATA KATA HUMOR YANG MENGANDUNG STOPWORDS PADA KALIMAT 1

Token	Kata	Alasan dihapus
2	Menunjukkan	Dihapus karena ada dalam stopword
3	Ternyata	Dihapus karena ada dalam stopword
4	50%	Dihapus karena mengandung karakter tanda "%"
5	Orang	Dihapus karena ada dalam stopword
7	Itu	Dihapus karena ada dalam stopword
8	Karena	Dihapus karena ada dalam stopword
9	Mereka	Dihapus karena ada dalam stopword
10	Saling	Dihapus karena ada dalam stopword
13	Yang	Dihapus karena ada dalam stopword
14	50%	Dihapus karena mengandung karakter tanda "%"
15	Lagi	Dihapus karena ada dalam stopword
16	Karena	Dihapus karena ada dalam stopword
17	Mereka	Dihapus karena ada dalam stopword
18	Saling	Dihapus karena ada dalam stopword
20	Dan	Dihapus karena ada dalam stopword

4) Array Ulasan tanpa Stopwords (Hasil dari filterisasi):

```
    0      1          6        11       12      19      21
["Hasil" "survei" "pacaran" "jatuh" "cinta" "suka" sayang ~]
```

Gambar. 8. Array Ulasan tanpa Stopwords

Semua kata yang array kata ulasan tanpa stopword dikumpulkan dan akan dilakukan proses ke tahapan selanjutnya.

5) Penentuan *Hyperplane* dan pemberian nilai label kelas dari dataset yang ditemukan di atas dengan menggunakan library phyton.

```
0,A,-0.018843938,0.09809821,0.04288118,0.057057357,
0.026773896,0.055061192,0.000926503,0.043578719,-
……..
92742, 0.079109626,-0.03323493,   -0.022569793,
0.05327705, 0.021026058, 0.035782193, -0.055192674,
-0.094050223,0.023088206 ,0.001335762, -0.005019553,
```

Dari matrik angka diatas dapat dijelaskan bahwa label kelas yang digunakan adalah range {+1,-1} dari dataset.

DOI : 10.37823/insight.v3i01.118

Sedangkan hyperplane pada kalimat humor tersebut hanya menggunakan 2 kelas yaitu kelas A dan kelas B, sedangkan yang diperoleh pada humor tersebut termasuk atau cenderung dalam kelas A bidang pembatas.

Setelah mendapatkan nilai matrik yang dilakukan pada tahapan pra proses, maka matrik tersebut akan diolah dan ditestingkan dengan berbagai skenario.

6) Training 2 data, Testing 8 data

Matrik yang berbetuk angka yang telah dihasilkan dari tahapan SVM sebelumnya akan dilakukan embedding word untuk memprediksi bobot konteks output di sekitar inputannya dari data data trainig sebanyak 2 dan data yang akan di testingkan sebanyak 8 data tersebut.

```
Training Data : 2
Testing Data : 8
training started...
Epoch is: 1 and Cost is: 0.010979535026426552
Weight :
[-0.01139422 -0.01700199 -0.00939682 -0.0082468  -0.00951075 -0.00712243
 .....
-0.00473361 -0.00777177 -0.01999999]

Epoch is: 2 and Cost is: 0.010979491108352316
Weight :
[-0.0113942  -0.01700195 -0.0093968  -0.00824678 -0.00951073 -0.00712242
 .....
-0.0047336  -0.00777175 -0.01999995]

training finished.
weights are:
[-0.0113942  -0.01700195 -0.0093968  -0.00824678 -0.00951073 -0.00712242
-0.01220762 -0.00830968 -0.01084535 -0.01393827 -0.00934584 -0.00998126
 .....
-0.00952417 -0.00866163 -0.00872508 -0.00594313 -0.00844585 -0.0051645
-0.0047336  -0.00777175 -0.01999995]
```

Gambar. 9. Contoh Training 2 Testing 8

Dari data diatas dapat dijelaskan bahwa bahwa pada data training sebanyak 2 dan data testing sebanyak 8 maka proses training yang dilakukan kepada dataset terjadi sebanyak 2 epoch dengan nilai cost function terakhir 0.010979491108352316 untuk menghasilkan nilai matrik yang tidak berubah atau nilai error minimum sehingga pembelajaran menjadi optimal.

V. UJI COBA

Pada akhirnya proses training dan testing akan menunjukkan hasil ujicoba dimana ujicoba yang dilakukan dengan *K-Fold Cross Validation* yang bertujuan menemukan kombinasi optimal dari data training dan testing dari dataset yang diuji. Untuk menguji tingkat validitas matrik yang diperoleh dari hasil konstruksi yang dipakai pada tahapan ini adalah membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya yang disebut juga dengan Metode Confusion Matrix, sehingga kinerja sistem klasifikasi dapat ditemukan kesimpulan seberapa baik sistem dalam mengklasifikasikan data.

Untuk penentuan akurasi uji tersebut pada uji coba dengan 5 fold dengan ilustrasi seperti berikut.

1. Fold Ke = 1
   Precision = 0.7881016042780749
   Recall = 0.7881016042780749
   F1-Score = 0.7881016042780749
   Confusion Matrix =
   [[940 8 2 10]
   [110 170 1 0]
   [31 9 39 0]
   [140 2 0 34]]
   Accuracy = 0.7881016042780749
2. Fold Ke = 2
   Precision = 0.821524064171123
   Recall = 0.821524064171123
   F1-Score = 0.821524064171123
   Confusion Matrix =
   [[942 13 0 5]
   [70 190 0 0]
   [52 5 51 0]
   [116 6 0 46]]
   Accuracy = 0.821524064171123

Kemudian perhitungan itu dilakukan terus hingga keseluruhan fold sesuai dan selesai keseluruhan. Setelah itu, dapat dilakukan proses perhitungan rata-rata akurasi yang disajikan pada tabel IV.

TABEL IV
HASIL UJI COBA DENGAN FOLD SEBANYAK 5

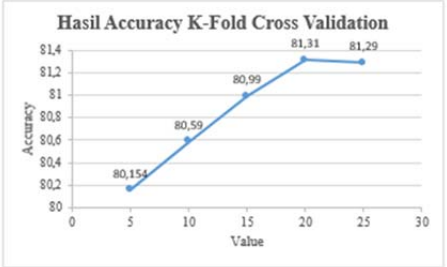
Fold	Precision	Recall	F1 Score	Accuracy
#1	0.78810160	0.78810160427	0.788101604	0.788101
#2	0.82152406	0.82152406417	0.821524064	0.821524
#3	0.84425133	0.84425133689	0.844251336	0.844251
#4	0.75652173	0.75652173913	0.756521739	0.756521
#5	0.79732441	0.79732441471	0.797324414	0.797324
Average	0.801544	0.801544	0.801544	0.801544

Hasil uji coba dengan menggunakan fold sebanyak 5 (lima), didapatkan nilai accuracy, average, F1-Score, average recall, dan nilai average precision sebesar 0,801544.

TABEL V
HASIL ACCURACY K-FOLD CROSS VALIDATION

No	K Value	Accuracy
1	5	80,154
2	10	80,59
3	15	80,99
4	20	81,31
5	25	81,29

Perhitungan akurasi dengan K-Fold Cross Validation ini dilakukan sebanyak 5 Value Fold yang berbeda sehingga didapat hasil percobaan seperti gambar pada tabel V. Dari serangkaian percobaan diatas didapat nilai optimal accuracy pada nilai K = 20 yaitu sebesar 81.31%



Gambar. 10. Grafik Hasil Accuracy K-Fold Cross Validation

DOI : 10.37823/insight.v3i01.118

Sedangkan Gambar 10 diatas menunjukkan grafik perbandingan antara 5 value fold yang diujikan pada tabel V yang mana informasi yang dapat diambil adalah akurasi uji optimalnya.

Langkah ujicoba selanjutnya adalah dengan melakukan evaluasi terhadap hipotesa dengan perhitungannya menggunakan persentase splitting data training dan testing. Contoh dari penentuan akurasi uji dengan splitting.

Percentage = 60 Training – 40 Testing
Confusion Matrix =
[[1903 16 1 6]
[190 331 4 0]
[100 12 86 1]
[259 11 0 72]]
Column A
True Positive = 1903
False Positive = 549
False Negative = 23
Precision = 0,776101
Recall = 0,988058
F1 Score = 0,869347
Support = 1926

Perhitungan dilakukan terus hingga ditemukan nilai akurasi seperti berikut All True Postiif = 2392, Total Data 2992 dengan akurasi 0,80.

TABEL VI
SPLITTING TRAINING DAN TESTING SEBESAR 60-40

Percentage: 60-40				
#	Precision	Recall	F1 Score	Support
0	0.78	0.99	0.87	1926
1	0.89	0.63	0.74	525
2	0.95	0.43	0.59	199
3	0.92	0.21	0.34	342
Accuracy		0.80		2992
Acro Avg	0.88	0.57	0.64	2992
Weighted Avg	0.82	0.80	0.77	2992

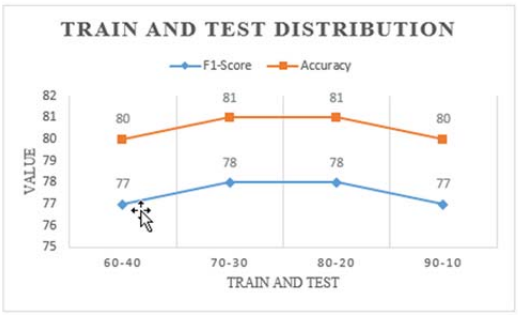
Pada tabel VI dapat dilihat hasil precision, recall, F1-Score, dan support dengan menggunakan data training sebesar 60% dan data testing sebesar 40%, didapatkan nilai acro average, nilai wighted average sebesar, nilai support atau yang diprediksi betul sebanyak 2992, dan nilai akurasi sebesar 0,80.

TABEL VII
DISTRIBUSI DATA TRAINING DAN TESTING

No	Train and Test Distribution	F1-Score	Akurasi
1	60-40	77	80
2	70-30	78	81
3	80-20	78	81
4	90-10	77	80

Perhitungan akurasi dengan Splitting percentage dilakukan sebanyak 4 model pembagian distribusi sehingga didapat hasil percobaan seperti gambar pada tabel VII. Dari serangkaian percobaan dapat diketahui bahwa accuracy terbaik pada distribusi data 70-30, dan 80-20.

Dari kedua tabel VII maka dapat dibentuk grafik perhitungan yang ditunjukkan pada gambar 6.2 yang mana nilai ini menujukkan akurasi terbaik berada pada data perbandingan 70-30 dan 80.20 dengan tingakt akurasi 81%.



Gambar. 11. Grafik Hasil Akurasi Distribusi Data Training dan Testing

VI. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan selama ini, maka dapat disimpulkan beberapa hasil percobaan adalah:

1) Penelitian pada kalimat humor untuk menentukan pola sebuah kalimat humor dan klasifikasinya bisa dikenali lewat algoritma *Support Vector Machine* (SVM), akan tetapi tidak hanya menggunakan SVM saja akan tetapi diperlukan *Word To Vec* untuk ekstraksinya.
2) Penelitian mengenali 5 (lima) jenis klasifikasi humor dengan metode SWM menggunakan Library Phyton ini mampu mengenali humor dalam bahasa Indonesia dengan tingkat akurasi diatas 80%.
3) Target penelitian berdasarkan hipotesa awal penelitian yang menyatakan tingkat akurasi 65% dapat tercapai, karena setelah dilakukan penelitian nilai akurasinya lebih tinggi.
4) Penelitian klasifikasi teks humor bahas Indonesia ini dapat dikategorikan sudah cukup baik, dikarenakan penelitian ini akan menjadi cikal bakal atau dasar acuan pada penelitian selanjutnya.
5) Perlu dicoba juga pada penelitian selanjutnya jika jumlah dataset dikembangkan dan didapat dari ahli di bidangnya sebisa mungkin dispesialisasi sesuai aliran komedi dari para ahli humor

DAFTAR PUSTAKA

[1] A. Mallikarjuna, and Dr. G. Anjan Babu. “Characterizing Humour: An Exploration of Features in Humorous Texts”, International Journal of Engineering Research & Technology (IJERT),1, 2, 347-354.
[2] Aseel Addawood, Jodi Schneider, and Masooda Bashir. 2017. Stance Classification of Twitter Debates: The Encryption Debate as A Use Case
[3] Barbara Mikkelson, David P. Mikkelson. Snopes [Online] tersedia di: http://www.snopes.com [Diakses pada tanggal 25 Juni 2017]
[4] Binsted, K. (1996). Machine humour: An implemented model of puns. PhD the
[5] Binsted, K., Bergen, B., and McKay, J. (2003). Pun and non-pun humour in second-language learning. In Workshop Proceedings, CHI 2003, Fort Lauderdale, Florida.
[6] Binsted, K., Pain, H., and Ritchie, G. (1997). Children’s evaluation of computer-generated punning riddles. Pragmatics and Cognition, 5(2):305–354.
[7] Chiara Bucaria. 2004. Lexical and syntactic ambiguity as a source of humor: The case of newspaper headlines, International Journal of Humor Research, (17–3), 279–309
[8] David O. Siegmund. Applications of conditional probability [Online] Tersedia di: [*Diakses* tanggal 13 Febuari 2018] https://www.britannica.com/science/probability-theory/Applications-of-conditional-probability
[9] Diyi Yang, Alon Lavie, Chris Dyer, Eduard Hovy. 2015. *Humor Recognition and Humor Anchor Extraction,* Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 2367–2376

 DOI : 10.37823/insight.v3i01.118

[10] Errissya Rasywir, Ayu Purwarianti. 2016. Eksperimen pada Sistem Klasifikasi *Berita* Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin
[11] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of HLT-NAACL*, 252-259
[12] Kevin P. Murphy. 2006. Naive *Bayes classifier*
[13] Lefcourt, H.M., (2005). Humor. *Handbook of Positive Psychology*. Editor: Snyder, C.R and Lopez, S.J. Oxford University Press.
[14] Lewis, at al. 2004. Handbook Of Emotion Second Edition. New York: *Springer-Verlag.*
[15] Miller, G. A. 1995. WordNet: a lexical database for English. Communications of the ACM, 38(11), 39–41.
[16] Palupi, Dian. 2014. *Bentuk dan Fungsi Humor dalam Serial Drama Komedi Extra Francais Karya Whitney Barros.* Diss.Universitas Negeri Yogyakarta:5. Mulyani (dalam Palupi, 2014: 5)
[17] Rad Mihalcea and Carlo Strapparava. 2005. *Making Computers Laugh: Investigations in Automatic Humor Recognition,* Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 531–538.
[18] Rada Mihalcea., Carmen Banea., Janyce Wiebe, and Samer Hassan. 2008 *Multilingual Subjectivity Analysis Using Machine Translation,* Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 127–135.
[19] Rahmanadji, D. (2007). Sejarah, Teori, Jenis dan Fungsi Humor. *Jurnal Bahasa dan Seni No. 2.,* hlm 213-221.
[20] Rahmanadji, D. 2009: *Sejarah, Teori, Jenis, Dan Fungsi Humor.* Jurnal Jurusan Seni dan Desain Fakultas Sastra, Universitas Negeri Malang : 15
[21] Ruch, W. (2007).The Sense of Humor: Explorations of a Personality Characteristic. Berlin: Walter de Gruyter.
[22] Sawedi. 2012. *Bentuk Penggunaan Bahasa Humor Dalam Bahasa Banggai.* dalam http://journal Eprintn Februari 2012. 311408023 (Diunduh tanggal 22 April 2015). *Man Out of His Humor* (dalam Sawedi, 2012: 20).
[23] Thamrin Dahlan. 2016 Bukan Hoax
[24] Tomas Mikolov., Kai Chen., Greg Corrado, and Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space.* See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/ 234131319
[25] Tristan A. Bekinschtein,1 Matthew H. Davis,1 Jennifer M. Rodd,2 and Adrian M. Owen. 2011. *Why Clowns Taste Funny: The Relationship between Humor and Semantic Ambiguity,* The Journal of Neuroscience, 31(26):9665–9671
[26] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea. 2017. Automatic Detection of Fake News
[27] Wahyudi Hazimah. 2012. "Sistem Deteksi Retinopati Diabetik Menggunakan Support Vektor Machine". Tesis. Magister Sistem Informasi. Universitas Diponegoro. Semarang
[28] Wan Hazimah Wan Ismail. 2005. Text Categorization Using Naive Bayes Algorithm
[29] Wijana Kartun, " Study Tentang Permainan Bahasa (Yogyakarta: Ombak, 2004)
[30] Rada Mihalcea and Carlo Strapparava. 2005. *Making Computers Laugh: Investigations in Automatic Humor Recognition,* Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 531–538.

 DOI : 10.37823/insight.v3i01.118