SKRIPSI

ANALISIS SENTIMEN UNTUK MENGETAHUI ELEKTABILITAS TOKOH POLITIK MENGGUNAKAN METODE *MULTINOMIAL NAÏVE BAYES*

SENTIMENT ANALYSIS TO MEASURE POLITICIANS ELECTABILITY USING MULTINOMIAL NAÏVE BAYES

RYAN SURYOTOMO

14/364147/PA/15915

PROGRAM STUDI ILMU KOMPUTER

DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2018

SKRIPSI

ANALISIS SENTIMEN UNTUK MENGETAHUI ELEKTABILITAS TOKOH POLITIK MENGGUNAKAN METODE *MULTINOMIAL NAÏVE BAYES*

SENTIMENT ANALYSIS TO MEASURE POLITICIANS ELECTABILITY USING MULTINOMIAL NAÏVE BAYES

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Ilmu Komputer

RYAN SURYOTOMO

14/364147/PA/15915

PROGRAM STUDI ILMU KOMPUTER

DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2018

HALAMAN PENGESAHAN

SKRIPSI

ANALISIS SENTIMEN UNTUK MENGETAHUI ELEKTABILITAS TOKOH POLITIK MENGGUNAKAN METODE *MULTINOMIAL NAÏVE BAYES*

Telah dipersiapkan dan disusun oleh :

RYAN SURYOTOMO

14/364147/PA/15915

Telah dipertahankan didepan Tim Penguji

pada tanggal 24 Mei 2018

Susunan Tim Penguji

Faizah, S.Kom., M.Kom	Sigit Priyanta, S.Si., M.Kom., Dr
Pembimbing	Ketua Penguji
	Arif Nurwidiyantoro, S.Kom., M.Cs
	Angggota Penguji

Mengetahui
a.n. Dekan FMIPA-UGM
Wakil Dekan Bidang Akademik dan Kemahasiswaan

Dr.rer.nat. Nurul Hidayat Aprilita, M.Si.
NIP. 197304071998031002

PERNYATAAN

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 24 Mei 2018

Ryan Suryotomo

HALAMAN MOTTO DAN PERSEMBAHAN

"Karena sesungguhnya sesudah kesulitan itu ada kemudahan.Sesungguhnya sesudah kesulitan itu ada kemudahan"

(QS. Alam Nasyroh: 5-6)

"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupanya"

(QS. Al-Baqarah: 286)

"You are what you think all day long"

(Ralph Waldo Emerson)

Karya ini penulis persembahkan untuk
Orang tua dan kakak-kakak tercinta,
Seluruh keluarga besar tersayang,
Teman-teman seperjuangan Ilmu Komputer 2014
dan segenap pembaca sekalian.

PRAKATA

Puji syukur kehadirat Allah SWT atas limpahan anugrah, rahmat, karunia , serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik.

Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak *(alm)* Ir. Otto Dwi Utomo dan Ibu *(almh)* Dra. Putri Suryandari yang telah memberikan doa, semangat, didikan , kasih sayang serta pelajaran hidup yang tak terhingga yang penulis tidak dapat membalasnya dan tidak akan melupakanya.
2. Kakak Tommy Yoga Pratama dan Okky Prasetyo Utomo dan keluarga besar yang senantiasa memberikan semangat dan nasehat serta membantu secara moril dan materiil.
3. Bapak Nur Rokhman, S.Si, M.Kom selaku dosen pembimbing akademik yang telah membantu dan membimbing penulis saat mengalami masalah akademik.
4. Ibu Faizah, S.Kom, M.Kom selaku dosen pembimbing tugas akhir yang senantiasa berkenan meluangkan waktunya untuk memberikan arahan dan bimbingan kepada penulis dalam penyusunan tugas akhir ini.
5. Bapak/Ibu Dosen Ilmu Komputer UGM yang telah membimbing dan membantu penulis selama menjalani kuliah di Universitas Gadjah Mada
6. Aji, Anang , Angger , Ardi Bagus, Bily M. Fachri,Ade, Dudi,Ipang, Rudi, Naufal Abiyyu,Ojan , Rafif, dan Azzis yang sudah penulis anggap seperti saudara sendiri yang selalu memberikan semangat , saling mengingatkan dan selalu memberi warna dalam kehidupan kampus. Kalian adalah sahabat terbaik selama kuliah.
7. Mbak Em,Mbak Ikvi, Mbak Norma, Mbak Putri, Mbak Amel, Mbak Denis, Mas Deni, Mas Tepen, Mbak Luna, Mbak Tya, Mbak Tea, Mbak Kiky, Mas Haikal dan kakak tingkat di Ilmu Komputer yang lain yang selalu

memberikan cerita dan nasehat untuk penulis dalam menyelesaikan tugas akhir ini.

8. Farhan, Felix , Edgar,Jennie , Ida, Astuti, Tama,Harvey,Fajar,Tebu, Kresna, Mas Jason, Mas Goldi, Mas Yudhi, Mas Dhani, Mas Wildan, Mas Budi, Mas Dhayu dari Laboraturium Sistem Cerdas dan Sistem Komputer Jaringan yang senantiasa membantu penulis saat membutuhkan bantuan.
9. Teman-teman divisi Kewirausahaan HIMAKOM 2014/2015. Mbak Feby, Agif, Fatah, Fajar, Roni, Ruli, Suci yang selalu menyemangati penulis.
10. Teman-teman PH HIMAKOM 2015/2016 Mas Micco, Mas Hary, Rubila, Prabowo, Angger, Bily, Ruli, Harvey.
11. Teman-teman peradaban jogja UGM. Imas, Tazia, Dimas, Icha, Gita, Meli,dan Yunita yang selalu memberikan semangat dan doa.
12. Teman-teman KLIMAKS SMA Negeri 1 Sragen yang sudah menjadi sahabat sendiri bagi penulis.Terimakasih atas kelucuan, semangat dan doanya.
13. Teman-teman cerdas cermat Erra, Nana, Yulia, Acuy yang selalu memberikan semangat kepada penulis dan menjadi partner kulineran.
14. TIM KKN JTG-88 Sepat, Masaran, Sragen. Terimakasih banyak.
15. Teman-teman HIMAKOM UGM yang menjadi bagian dari masa berjuang berorganisasi di kampus.
16. Teman-teman Ilmu Komputer UGM 2014 yang telah berjuang dari awal bersama, menuntut ilmu bersama di kampus Universitas Gadjah Mada.
17. Pihak-pihak lain yang tidak dapat disebutkan satu per satu.

Akhir kata penulis berharap semoga skripsi ini dapat memberikan manfaat bagi kita semua, terutama bagi perkembangan ilmu pengetahuan serta perkembangan Ilmu Komputer dan Teknologi Informasi

Yogyakarta, 24 Mei 2018

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN…….iii
PERNYATAAN……iv
HALAMAN MOTTO DAN PERSEMBAHAN……...vi
PRAKATA……...vii
DAFTAR ISI…….....ix
DAFTAR TABEL……xii
DAFTAR GAMBAR……..xiv
INTISARI……xvi
ABSTRACT……...xvii
BAB I : PENDAHULUAN ……...1
1.1. Latar Belakang Masalah……...1
1.2. Rumusan Masalah…….....3
1.3. Batasan Masalah……...3
1.4. Tujuan Penelitian……..……3
1.5. Manfaat Penelitian……4
1.6. Sistematika penulisan……5
BAB II : TINJAUAN PUSTAKA…….…….6
BAB III : LANDASAN TEORI……...11
3.1 Text Mining…….11
3.2 Twitter……..12
3.3 Analisis Sentimen……...13
3.4 Preprocessing……..13
3.4.1 Tokenisasi…….13
3.4.2 Stopword Removal……...14
3.4.3 Stemming……..15
3.4.4 Case Folding…….18
3.5 Regular Expression…….....18
3.6 *Term Frequency-Inverse Document Frequency*…….19
3.7 *Chi Square*……..21
3.8 Multinominal Naïve Bayes…….22
3.9 Evaluasi Performa……..24
3.9.1 Akurasi…….25
3.9.2 Presisi……...25

3.9.3 Recall……26
3.9.4 F-Measure……26
3.9.5 Cross Validation……26
3.10 *Positive Versus Tiotal (PvT)*……27
3.11 *Share of Volume*……...28
BAB IV : ANALISIS DAN PERANCANGAN…….…..…….29
4.1. Analisis Permasalahan……..…..……29
4.2. Rancangan Umum Sistem…….…....29
4.3. Rancangan Data……...…...…31
4.3.1. Dataset ……..…...…31
4.3.2. Data stopword ……..35
4.3.3. Data kata dasar ……....35
4.4. Perancangan Sistem Klasifikasi Sentimen…….….....35
4.4.1. Rancangan Case Folding ……37
4.4.2. Rancangan Regular Expression ……..38
4.4.3. Rancangan Tokenisasi ……42
4.4.4. Filtering Data.. ……43
4.4.5. Perancangan Stopword removal ……43
4.4.6. Perancangan Stemming ……..44
4.4.7. Perancangan Seleksi Fitur ……..48
4.4.8. Seleksi Fitur TF-IDF ……..48
4.4.9. Seleksi Fitur Chi Square ……48
4.4.10. Pelabelan Data……49
4.4.11. Perancangan Training……49
4.4.12. Perancangan Testing…..……51
4.4.13. Pengujian model klasifikasi sentimen……53
4.4.14. Perhitungan Elektabilitas……54
4.5. Rancangan Skenario Pengujian…….….....55
4.5.1. Rancangan Perbandingan pengujian performa fitur top-n ……..55
4.5.2. Rancangan Perbandingan pengujian rumus elektabilitas ……56
BAB V : IMPLEMENTASI……...…….57
5.1. Lingkungan Implementasi……..……...57
5.2. Implementasi Sistem……..……...57
5.3. Implementasi Pelabelan Data……..…….....57

5.4. Implementasi Preprocessing……..……....58
5.4.1 Implementasi Case Folding…… ……..58
5.4.2 Implementasi Regular Expression.…… ……..58
5.4.3 Implementasi Tokenisasi…….…… ……60
5.5 Implementasi Filtering……..58
5.5.1 Stopword Semoval…….…….…….59
5.5.2 Stemming……..…….…….…….61
5.6 Implementasi Seleksi Fitur…..……..62
5.6.1 Implementasi Seleksi Fitur TF-IDF...…….…….62
5.6.2 Implementasi Seleksi Fitur chi square…...…….…….63
5.7 Implementasi Training dan Pengujian Model Klasifikasi Sentimen……64
5.8 Implementasi Testing Model Klasifikasi Sentimen…….…….66
5.9 Implementasi Perhitungan Elektabilitas PvT……73
5.10 Implementasi Perhitungan Elektabilitas SoV…….....74
BAB VI : HASIL DAN ANALISA……...……..75
6.1 Hasil Preprocessing……..……...……75
6.2 Hasil Filtering…….……..……...……76
6.3 Hasil Seleksi Fitur TF-IDF….……..……...……76
6.4 Hasil Seleksi Fitur Chi Square….……..…….....77
6.5 Hasil Pengujian Perbandingan Fitur Top-n…..……..…….....77
6.6 Hasil Klasifikasi Sentimen Top-n Pada Data Tes…..……..……88
6.7 Hasil Perhitungan Elektabilitas……..……..……93
BAB VII : SARAN DAN KESIMPULAN……...……100
6.1 Kesimpulan……..……...……100
6.2 Saran……..…….……..……...…...101
DAFTAR PUSTAKA……102
LAMPIRAN……...105

DAFTAR TABEL

Tabel 2.1 : Perbandingan penelitian yang sudah ada……..8
Tabel 3.1 : Contoh Tokenisasi……..14
Tabel 3.2 : Contoh Stopword Removal……15
Tabel 3.3 : Contoh Kegagalan algoritma ECS……..…16
Tabel 3.4 : Contoh Data yang digunakan dalam Proses Seleksi Fitur...……..20
Tabel 3.5 : Contoh Perhitungan TF-IDF……..20
Tabel 3.6 : Contoh Perhitungan Chi Square Positif…….22
Tabel 3.7 : Contoh *Confusion Matrix*……...25
Tabel 3.8 : Contoh Perhitungan *Positive versus total*..……27
Tabel 4.1 : Daftar kata kunci yang digunakan…….34
Tabel 4.2 : Komposisi jumlah data tiap tokoh politik…….34
Tabel 4.3 : Ilustrasi proses *case folding*..……38
Tabel 4.4 : Ilustrasi proses *URLs removal*….…….39
Tabel 4.5 : Ilustrasi proses *Bracket removal*……...……40
Tabel 4.6 : Ilustrasi proses *mention removal*……..……40
Tabel 4.7 : Ilustrasi proses *hashtag removal*……..……41
Tabel 4.8 : Ilustrasi proses *non-alphanumeric character removal*...……41
Tabel 4.9 : Ilustrasi proses *RT removal.*……..……42
Tabel 4.10 : Contoh data testing yang akan melalui proses klasifikasi..……52
Tabel 4.11 : Contoh perhitungan probabilitas prior……52
Tabel 4.12 : Contoh hasil perhitungan probabilitas kondisional untuk masing-masing token pada tiap kelas……..53
Tabel 4.13 : Perbandingan jumlah fitur top-n tiap tokoh politik……55

Tabel 6.1 : Hasil Klasifikasi Sentimen Data Testing Tokoh Agus Yudhoyono……88
Tabel 6.2 : Hasil Klasifikasi Sentimen Data Testing Tokoh Ahok……89
Tabel 6.3 : Hasil Klasifikasi Sentimen Data Testing Tokoh Anies Baswedan…….…....89
Tabel 6.4 : Hasil Klasifikasi Sentimen Data Testing Tokoh Gatot Nurmantyo…….……90
Tabel 6.5 : Hasil Klasifikasi Sentimen Data Testing Tokoh Hary Tanoe…….…….....90
Tabel 6.6 : Hasil Klasifikasi Sentimen Data Testing Tokoh Jusuf Kalla…….…….....91
Tabel 6.7 : Hasil Klasifikasi Sentimen Data Testing Tokoh Jokowi…….……91
Tabel 6.8 : Hasil Klasifikasi Sentimen Data Testing Tokoh Prabowo Subianto……...92
Tabel 6.9 : Hasil Klasifikasi Sentimen Data Testing Tokoh Ridwan Kamil…….……92
Tabel 6.10 : Hasil Klasifikasi Sentimen Data Testing Tokoh Zulkifli Hasan…….……...93
Tabel 6.11 : Hasil Elektabilitas PvT Tokoh Politik Tanpa Seleksi Fitur…….……..94
Tabel 6.12 : Hasil Elektabilitas PvT Tokoh Politik Dengan Seleksi Fitur chi square…...94
Tabel 6.13 : Hasil Elektabilitas PvT Tokoh Politik Dengan Seleksi Fitur TF-IDF……...95
Tabel 6.14 : Hasil Rata-Rata Elektabilitas PvT Tokoh Politik …….……96
Tabel 6.15 : Hasil Elektabilitas SoV Tokoh Politik Tanpa Seleksi Fitur…….……97
Tabel 6.16 : Hasil Elektabilitas SoV Tokoh Politik Dengan Seleksi Fitur TF-IDF…..…..97
Tabel 6.17 : Hasil Elektabilitas SoV Tokoh Politik Dengan Seleksi Fitur chi square…...98
Tabel 6.18 : Hasil Rata-Rata Elektabilitas SoV Tokoh Politik …….……98

DAFTAR GAMBAR

Gambar 3.1 : Proses dalam Text Mining……11
Gambar 3.2 : Pseudocode Training dan Testing Multinomial Naïve Bayes……..24
Gambar 3.3 : Ilustrasi *k-fold cross validation*……27
Gambar 4.1 : Cuplikan data berita……..……32
Gambar 4.2 : Cuplikan data tweet…….…….32
Gambar 4.3 : Diagram Alur Scrapping Data Twitter setiap kata kunci…….....33
Gambar 4.4 : Diagram Alur Crawling data berita setiap kata kunci…….....33
Gambar 4.5 : Diagram alur sistem klasifikasi sentimen……...……..36
Gambar 4.6 : Diagram *preprocessing*……...…….37
Gambar 4.7 : Diagram alur *regular expression*……...……...38
Gambar 4.8 : Diagram alur *stopword removal*……...……44
Gambar 4.9 : Diagram alur *stemming* (1)……...……45
Gambar 4.10 : Diagram alur *stemming* (2)……...……..46
Gambar 4.11 : Diagram alur *stemming* (3)……...……..47
Gambar 4.12 : Diagram alur perancangan *training* ……...……50
Gambar 4.13 : Diagram alur perancangan *testing* ……...……..51
Gambar 4.14 : Diagram alur perhitungan elektabilitas tiap tokoh ……....54
Gambar 4.15 : Skenario pengujian perbandingan fitur top-n ……...…….55
Gambar 4.16 : Skenario pengujian perbandingan rumus elektabilitas……...56
Gambar 5.1 : Cuplikan data yang telah diberikan label……...……..58
Gambar 5.2 : Kode proses case folding……...……...58
Gambar 5.3 : Kode program regular expression ……...……59
Gambar 5.4 : Kode program tokenisasi……...……...60
Gambar 5.5 : Kode program stopword removal……...……..61
Gambar 5.6 : Kode program stemming……...……...61
Gambar 5.7 : Kode program seleksi fitur TF-IDF……...……...62
Gambar 5.8 : Kode program seleksi fitur chi square……...……...63
Gambar 5.9 : Kode program training dan pengujian model klasifikasi…….....65
Gambar 5.10: Kode program untuk mengimpor data training, testing dan fitur....67

Gambar 5.11 : Kode program untuk ekstraksi fitur……..…….68
Gambar 5.12 : Kode program untuk menghitung probabilitas prior……...…...69
Gambar 5.13 : Kode program untuk menghitung probabilitas kondisional……...69
Gambar 5.14 : Kode program untuk mencari setiap elemen yang sama dengan suatu nilai tertentu pada variabel classlabel……...……70
Gambar 5.15 : Kode program untuk mencari indeks kata-kata pada data test yang hanya terdapat di vocabulary……...……...70
Gambar 5.16 : Bagian utama program klasifikasi sentimen……...……...72
Gambar 5.17 : Kode program perhitungan elektabilitas PvT……...……..73
Gambar 5.18 : Kode program perhitungan elektabilitas SoV……...……74
Gambar 6.1 : Data sebelum tahap *preprocessing*……...……75
Gambar 6.2 : Data sesudah tahap *preprocessing*……...……76
Gambar 6.3 : Data sesudah tahap filtering……...……..76
Gambar 6.4 : Cuplikan kata fitur hasil TF-IDF……...……...77
Gambar 6.5 : Cuplikan kata fitur hasil chi square……...……...77
Gambar 6.6 : Hasil Performa Model Tokoh Agus Yudhoyono…….....78
Gambar 6.7 : Hasil Performa Model Tokoh Ahok……...…….....79
Gambar 6.8 : Hasil Performa Model Tokoh Anies Baswedan ……...……...80
Gambar 6.9 : Hasil Performa Model Tokoh Gatot Nurmantyo…….....81
Gambar 6.10 : Hasil Performa Model Tokoh Hary Tanoe ……...82
Gambar 6.11 : Hasil Performa Model Tokoh Jusuf Kalla ..……...83
Gambar 6.12 : Hasil Performa Model Tokoh Jokowi …...……....84
Gambar 6.13 : Hasil Performa Model Tokoh Prabowo Subianto …….....85
Gambar 6.14 : Hasil Performa Model Tokoh Ridwan Kamil…….....86
Gambar 6.15 : Hasil Performa Model Tokoh Zulkifli Hasan…….....87
Gambar 6.16 : Perbandingan performa model tanpa dan dengan seleksi fitur…...88

INTISARI

ANALISIS SENTIMEN UNTUK MENGETAHUI ELEKTABILITAS TOKOH POLITIK MENGGUNAKAN METODE *MULTINOMIAL NAÏVE BAYES*

Oleh

Ryan Suryotomo
14/364147/PA/15915

Salah satu tolak ukur kandidat tokoh politik yang akan mengikuti pilgub, pilkada atau pemilu adalah elektabilitas. Sekarang ini, metode untuk mengukur elektabilitas tokoh politik masih dilakukan secara konvensional dan tidak objektif sehingga hasilnya kurang merepresentasikan tokoh politik tersebut.Sementara metode yang lebih modern dan objektif seperti analisis sentimen menggunakan data Twitter dan berita untuk mengukur elektabilitas masih sedikit dilakukan. Data Twitter dan berita dipilih karena dapat mempengaruhi opini publik tokoh politik.

Pada penelitian ini dilakukan analisis sentimen menggunakan data tweet dan berita dari masing-masing tokoh politik untuk mengetahui elektabilitasnya menggunakan metode *Multinomial Naïve Bayes*. Tokoh politik yang digunakan dalam penelitian adalah 10 tokoh politik yang dianggap populer di Indonesia. Dataset yang digunakan berjumlah 16.523 data *training* dan 6.550 data *testing*.Data tweet didapatkan menggunakan *tool tweetcatcher* dan berita didapatkan dari 3 situs berita di Indonesia yaitu *tribunnews.com*, *tempo.co*, dan *viva.co.id* menggunakan *tools scrapper* dalam kurun waktu 17 November 2016 sampai 1 November 2017. Setelah data terkumpul, dilakukan tahap *preprocessing* dan *filtering*. Lalu dilakukan seleksi *top-n* kata fitur menggunakan metode c*hi square* dan TF-IDF. Selanjutnya adalah pembentukan model klasifikasi dan proses testing dengan membandingkan hasil elektabilitas tiap tokoh politik tanpa seleksi fitur dan dengan seleksi fitur *chi square* dan TF-IDF.

Hasil penelitian ini menunjukan bahwa nilai performa model menggunakan metode seleksi fitur *chi square* lebih tinggi dengan rata-rata nilai akurasi 85,24% , presisi 88,84% , *recall* 91,65% dan *f-measure* 90,17% dibandingkan dengan menggunakan metode seleksi fitur TF-IDF dengan rata-rata nilai akurasi 78,11% , presisi 87,41%, recall 87,79% dan *f-measure* 87,54% serta jika dibandingkan tanpa seleksi fitur dengan nilai rata-rata akurasi 74,69% , presisi 87,40%, *recall* 84,88% dan *f-measure* 84,72%.

Kata kunci: analisis sentimen, elektabilitas, multinomial naïve bayes, *chi square*, TF-IDF

ABSTRACT

SENTIMENT ANALYSIS TO MEASURE POLITICIANS ELECTABILITY USING MULTINOMIAL NAÏVE BAYES

by

Ryan Suryotomo
14/364147/PA/15915

One of politician candidates benchmark to join in election is electability. Recently, the method to measure politicians electability was done conventionally and not objectively, so the result were less representative to the politicians figure. Meanwhile, method that was more modern or objective like sentiment analysis to measure the electability was less used. Twitter and news data were chosen because its could influence politicians public opinions.

Sentiment analysis was performed in this research with tweet and news data for every politicans to measure the electability by using Multinomial Naïve Bayes algorithm. The number of politicians used in this research were 10 politicians that considered as popular politicians in Indonesia. The data set consists of 16.523 training data and 6.550 testing data. The tweet data were collected by using tweetcatcher tool and the news data were collected from 3 news site : tribunnews.com, tempo.co, and viva.co.id by suing scrapper tool in the period of 17th November 2016 until 1st November 2017. Once collected, processing and filtering phase were performed. Then, top-n word features were performed by using chi square and TF-IDF algorithm. The next phase was forming classification models and testing process that compared electabilities result of each politicians with chi square and TF-IDF feature selection or without feature selection.

The result of this research showed that average performance of chi square features selection model was higher with 85,24% accuracy, 88,84% precision, 91,65% recall and 90,17% f-measure compared to TF-IDF features selection which had average value of 78,11% accuracy , 87,41% precision , 87,79% recall and 87,54 f-measure and without features selection model which had average value of 74,69% accuracy , 87,40% precision , 84,88% recall and 84,72% f-measure.

Keyword : sentiment analysis, electability, multinomial naïve bayes, chi square, TF-IDF

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Facebook dan Twitter merupakan sosial media yang populer digunakan oleh masyarakat untuk menyampaikan pendapatnya di dunia maya.Karena kebebasan berpendapat itulah berbagai informasi dapat tersebar ke seluruh dunia dalam waktu yang singkat dan sangat cepat. Sampai tahun 2010, Twitter telah mencatat 1,4 miliar relasi sosial, 4262 *trending topic*, dan 106 juta tweet (Kwak et al, 2010). Informasi di dalam Twitter tersebar secara langsung dan tidak langsung dengan proses tweet secara langsung atau me-retweet. Hal ini dapat memberikan dampak yang besar bagi individu karena memuat berbagai opini dan pandangan terhadap seseorang. Oleh karena itu berbagai metode analisis sentimen telah dilakukan untuk mengidentifikasi perilaku tersebut (Ramteke et al, 2016).

Pada pemilihan presiden 2014 dengan kandidat presiden Prabowo Subianto dan Joko Widodo, Twitter memberikan dampak dan peran yang sangat penting. Disini semua portal berita online, koran, bahkan Twitter sibuk dan fokus menggambarkan bagaimana kepribadian calon presiden saat itu. Selain kepribadian calon presiden, sosial media juga berfokus pada penyebaran informasi yang terkait dengan pengalaman dan integritas kandidat.Hal ini menunjukan bahwa sosial media telah menjadi bagian integral penting dalam demokrasi Indonesia , karena publik beramai ramai memberikan opini mereka terhadap kualitas kepemimpinan dan integritas kandidat Presiden melalui Twitter. Meskipun saat itu tidak banyak media cetak yang berpihak kepada kandidat calon presiden tertentu, Twitter tetap dianggap dapat menjadi media yang lebih relevan dalam mempengaruhi opini publik saat pemilihan presiden (Hermawan, 2016) .

Setiap berita memiliki konstruksi realitas politik yang berbeda-beda tergantung dinamika eksternal dan internal masing-masing, serta pengkonstruksian yang dipilih. Namun, berita memiliki keterkaitan yang sama yaitu kesadaran memilih bahasa dan simbol politik, fakta dan pengemasan pesan , dan kesediaan memberi ruang untuk merilisnya yang akan digunakan dalam pembentukan opini

publik (Hamad, 2004).Objektifitas berita merupakan sesuatu yang penting untuk menentukan suatu berita kredibel atau tidak dalam pembentukan opini publik. Akurasi kesesuaian isi berita dengan kejadian yang terjadi menjadi tolak ukur objektifitas tersebut. Berita memiliki tingkat akurasi kesesuaian sebesar 82,6% sehingga berita dapat menjadi sumber data yang objektif (Juditha, 2013).

Elektabilitas adalah tolak ukur tingkat keterpilihan dari calon tersebut berdasarkan kriteria pilihan saat akan maju kedalam pemilihan umum. Elektabilitas seseorang dipengaruhi oleh opini pendukung calon presiden bagaimana prospek nominasi kedepanya. Dengan banyak opini yang positif terhadap calon presiden, semakin banyak massa yang mendukung dan meningkatkan elektabilitas mereka (Abramowitz, 1989). Sebuah penelitian pernah dilakukan untuk mengetahui apakah tweet pada Twitter memiliki hubungan terhadap pemilu parlemen di Jerman saat itu. Dari hasilnya didapat bahwa tweet dari Twitter tersebut mencerminkan hasil dari pemilu tersebut (Tumasjan et al, 2010).

Telah banyak metode statistik yang dilakukan untuk memprediksi elektabilitas tokoh ataupun partai politik. Misalnya dengan survey, namun hasil survei yang dihasilkan oleh lembaga survey terkadang tidak sesuai dengan kenyataan dan seringkali digunakan untuk mengarahkan opini sehingga tidak netral oleh salah satu kandidat (Lestari et al, 2017). Terdapat juga *quick count* , namun *quick count* membutuhkan waktu yang lama dan dilaksanakan setelah pemilihan umum.Pada penelitian yang lain, metode analisis sentimen dengan berbagai algoritma telah diusulkan untuk mengetahui dan mengevaluasi elektabilitas tokoh atau partai politik yang dilaksanakan sebelum pemilu (Lestari et al, 2017).

Untuk melakukan analisis sentimen diperlukan suatu algoritma pengklasifikasian. Salah satu metode pengklasifikasian yang dapat digunakan adalah Multinomial Naïve Bayes. Multinomial Naïve Bayes mengadopsi prinsip Bayesian, dimana pendistribusian atau suatu dokumen dilakukan model parameter yang spesifik. Parameter dapat dipelajari dengan memaksimalkan kemiripan data yang dilabeli (Zhao et al., 2016). Multinomial Naïve Bayes merepresentasikan dokumen dalam bentuk set kejadian kata beserta jumlah frekuensi kata fitur dalam

dokumen. Saat menghitung probabilitas dari dokumen, dilakukan perkalian nilai probabilitas masing masing kejadian kata fitur (McCallum & Nigam, 1998).

Performa model klasifikasi menjadi bagian penting dalam proses klasifikasi. Hal ini menunjukan seberapa akurat sistem dapat mengklasifikasikan data dengan benar. Salah satu metode untuk meningkatkan akurasi dengan seleksi fitur. Seleksi fitur adalah proses mereduksi fitur-fitur yang dianggap tidak relevan dalam proses klasifikasi yang akan menimbulkan *overfitting*. Jika seleksi fitur TF-IDF memperhitungkan jumlah kemunculan fitur saja, seleksi fitur *chi square* menggunakan metode statistika untuk mengukur independensi sebuah term dengan kategorinya, tidak sebatas kemunculan fitur saja. Hal ini membuat performa model *chi square* lebih baik dari TF-IDF (Lestari et al, 2017).

Karena pemilih yang menggunakan Twitter dan berita memiliki karakteristik kritis, mandiri, independen, rasional dan pro perubahan (Sukendar, 2017) penelitian ini mengusulkan topik analisis sentimen pada data Twitter dan berita untuk mengetahui elektabilitas tokoh politik dalam periode waktu tertentu berdasarkan sentimen (positif dan negatif) masing masing tokoh politik. Algoritma klasifikasi sentimen yang digunakan adalah Multinomial Naïve Bayes.

1.2. Rumusan Masalah

Elektabilitas menjadi tolak ukur bagi tokoh politik yang akan maju kedalam pemilihan gubernur,kepala daerah atau pemilihan umum.Namun,metode yang dilakukan untuk mengukur elektabilitas seperti survey dan *quick count* memiliki kelemahan seperti membutuhkan proses yang lama, hasilnya terkadang tidak objektif dan dilaksanakan setelah pemilihan umum.Sementara, metode ilmiah seperti analisis sentimen menggunakan data Twitter dan berita tidak banyak dilakukan padahal Twitter dan berita dapat menggiring opini publik yang dapat mempengaruhi elektabilitas tokoh politik.Oleh karena itu, pada penelitian ini akan digunakan metode analisis sentimen untuk mengukur elektabilitas tokoh politik dengan harapan dapat diperoleh hasil dengan cepat dan lebih objektif.

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Tokoh politik yang digunakan pada penelitian ini berjumlah 10 tokoh politik yang dianggap populer di Indonesia saat ini.
2. Penelitian ini menggunakan data berita yang diambil dari 3 portal berita yaitu viva.co.id, tempo.co, tribunnews.com yang berisi tanggal dan judul berita dan tidak berfokus pada proses pengambilan data/crawling.
3. Data tweet diambil melalui Twitter API dengan kata kunci pencarian nama tokoh politik seperti “jokowi”,”ridwan kamil”,”ahok”,dan lain sebagainya .
4. Data set diambil dalam kurun waktu 17 Oktober 2016 – 19 Oktober 2017 untuk dataset berita dan 19 Oktober 2017-1 November 2017 untuk dataset tweet
5. Pelabelan data dilakukan secara manual berdasarkan konteks dataset dan hanya menggunakan sentimen positif dan negatif

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Untuk mengetahui elektabilitas tokoh politik dengan analisis sentimen terhadap data tweet dan berita dengan menggunakan metode Multinomial Naïve Bayes
2. Untuk mengetahui perbandingan performa model klasifikasi menggunakan seleksi fitur *chi square* dan TF-IDF dan tanpa seleksi fitur

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Dapat digunakan oleh tokoh politik untuk mengetahui dan mengevaluasi tingkat elektabilitasnya apabila ingin mendaftarkan diri dalam pemilu dan lain sebagainya.
2. Dapat digunakan sebagai alternatif metode survei konvensional dan *quick count* dalam mengetahui elektabilitas tokoh politik
3. Dapat digunakan untuk mengevaluasi dan menyusun strategi kampanye
4. Dapat digunakan untuk memprediksi kekuatan politik suatu partai atau tokoh politik

1.6. Sistematika Penulisan

Penelitian ini terdiri dari tujuh bab dengan sistematika masing-masing bab adalah sebagai berikut :

1. BAB I PENDAHULUAN

   Bab ini memuat latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan sistematika penulisan

2. BAB II TINJAUAN PUSTAKA

   Bab ini memuat beberapa penelitian terdahulu yang terkait pada topik permasalahan, metode yang digunakan dan menjadi bahan referensi dalam penelitian ini. Tinjauan pustaka berkisar antara topic-topik yang berkaitan dengan analisis sentiment

3. BAB III LANDASAN TEORI

   Bab ini memuat teori-teori yang digunakan dalam penelitian ini. Dalam bab ini, dijelaskan pula persamaan-persamaan yang digunakan dalam penelitian, serta teori-teori yang mendukung penelitian ini.

4. BAB IV ANALISIS DAN RANCANGAN

   Bab ini memuat rancangan penelitian serta analisis permasalahan, arsitektur sistem secara umum dan metode pengujian.

5. BAB V IMPLEMENTASI

   Bab ini memuat spesifikasi *hardware* dan *software* yang digunakan dan hasil implementasi kode sistem yang dikembangkan berdasarkan perancangan yang dilakukan beserta penjelasanya.

6. BAB VI HASIL DAN PEMBAHASAN

   Bab ini memuat rangkuman hasil penelitian dan pengujian berupa hasil dari proses-proses yang dilakukan dalam melakukan analisis sentiment berserta permasalahan yang dihadapi dan pembahasanya.

7. BAB VII PENUTUP

   Bab ini berisi kesimpulan penelitian yang telah dilakukan disertai saran untuk pengembangan penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

Ramteke, *et al*. (2016) melakukan penelitian untuk memprediksi hasil pemilihan presiden di Amerika berdasarkan sentimen analisis dari data Twitter. Data yang digunakan berupa tweet berbahasa Inggris dengan membandingkan metode Multinominal Naïve Bayes dan *Support Vector Machine* (SVM). Dari hasil yang didapat metode Multinomial Naïve Bayes mendapatkan hasil nilai akurasi sebesar 97%.dan nilai f-measure sebesar 94%. Metode untuk mengukur elektabilitas menggunakan *positive versus total* (PvT).

Dalam mengukur elektabilitas, Bermingham & Smeaton. (2011) menggunakan rumus Share of Volume (SoV) yaitu dengan membandingkan sentimen positif seorang tokoh politik dengan keseluruhan sentimen positif. Metode sentimen analisis diterapkan terhadap 4 jenis dataset yang berbeda yaitu *time-based, sample size-based, cummulative dan manual*. SoV memiliki kelebihan yaitu hasilnya mudah dibandingkan dengan hasil polling yang didapatkan. Ternyata ke-empat hasil data Twitter tersebut menunjukan hasil yang mirip dengan hasil *polling*.

Jika Ramteke, et al. (2016) melakukan penelitian untuk memprediksi hasil pemilihan presiden di Amerika Serikat, Lestari, et al. (2017) meneliti analisis sentimen tentang opini pilkada DKI 2017 pada dokumen Twitter. Dokumen tersebut terkadang memuat unsur non-tekstual seperti adanya emoji. Emoji biasanya digunakan untuk mengungkapkan perasaan seseorang. Algoritma yang digunakan adalah Naïve Bayes dengan melakukan pembobotan non-tekstual (emoji) dan pembobotan tekstual. Hasil dari penelitian ini berupa sentimen positif dan negatif yang sudah dinormalisasi dengan *Min-Max normalisation* dengan nilai akurasi yang didapat yaitu 68,52% untuk pembobotan tekstual dan 75,93% untuk pembobotan non-tekstual. Dengan ini diketahui bahwa pembobotan non-tekstual berpengaruh terhadap akurasi dan pengklasifikasian yang didapat.

Selain itu, Virgo. (2018) juga melakukan penelitian analisis sentimen untuk memprediksi hasil Pilkada DKI 2017 oleh pasangan Anies-Sandi dan Ahok-Djarot. Algoritma yang digunakan adalah *Multinomial Naïve Bayes* dengan deteksi Buzzer

tanpa seleksi fitur. Hasil dari penelitian ini , sistem dapat mengklasifikasikan sentimen untuk pasangan Ahok-Djarot dengan akurasi sebesar 77,28% dan pasangan Anies-Sandi dengan akurasi sebesar 79,70%.

Metode Naïve Bayes juga dilakukan oleh Hidayatullah & SN (2014) untuk melakukan analisis sentimen dan klasifikasi tokoh publik pada Twitter. Tokoh publik yang dipilih adalah tokoh publik yang dianggap layak dan memiliki kemampuan untuk menjadi pemimpin.Naïve Bayes dikombinasikan dengan fitur sehingga dapat mendeteksi negasi dan menggunakan pembobotan *Term Frequency* dan TF-IDF. Selain metode Naïve Bayes juga digunakan metode *Support Vector Machine* (SVM) . Hasil klasifikasi berupa sentimen positif dan negatif dengan kategori tokoh politik berdasar kapabilitas, integritas, dan akseptabilitas tokoh tersebut. Dari proses pengklasifikasian sentimen dan kategori tokoh politik memang SVM lebih unggul dari Naïve Bayes.

Tokoh publik memang menarik untuk diteliti untuk mendapatkan bagaimana opini masyarakat terhadap mereka.Hal ini membuat Hayatin et al (2014) melakukan sebuah penelitian terhadap 6 tokoh publik yang terkenal melalui media Twitter. Fokus penelitian ini adalah membuat sebuah sistem yang dapat otomatis mengekstrak opini terhadap tokoh publik tersebut berdasarkan 2 fitur novel *hater* dan *lover*. Metode yang digunakan masih mengunakan Naïve Bayes dengan pembobotan TF-IDF. Novel feature dapat digunakan untuk merepresentasikan opini terhadap tokoh tersebut. *Hater* untuk yang tidak mendukung tokoh tersebut dan *Lover* untuk yang mendukung tokoh tersebut. Tahapan untuk melakukan sentimen analisis terhadap tokoh publik ini berupa *preprocessing*, pembobotan, pengklasifikasian dan penentuan sentimen. Penelitian ini menghasilkan presisi sebesar 99% , *recall* 75% dan akurasi 76,67%.

Proses pembobotan fitur juga berpengaruh terhadap nilai akurasi suatu metode. Metode pembobotan menggunakan *Chi Square* dianggap lebih baik dibanding *Term Frequence* menurut Ling et al (2014). Hal ini dikarenakan seleksi fitur *Chi Square* memperhitungkan frekuensi fitur yang tidak diharapkan dan diharapkan. Sementara, *Term Frequence* hanya menghitung frekuensi dari fitur yang diharapkan saja. Salah satu tujuan penggunaan pembobotan *Chi Square* adalah

untuk menghilangkan fitur pengganggu dalam klasifikasi yang mempengaruhi akurasi nantinya.

Perbedaan penelitian ini dengan penelitian yang sudah ada adalah pada penelitian ini menggunakan dataset 10 tokoh politik sementara Hayatin et al.(2014) hanya menggunakan 6 tokoh publik. Metode klasifikasi menggunakan Multinomial Naïve Bayes sementara Hidayatullah & SN (2014), Hayatin et al (2014) dan Ling et al (2014) menggunakan Naïve Bayes. Metode seleksi fitur menggunakan TF-IDF dan *chi square* sementara Virgo (2018) tidak menggunakan seleksi fitur, Lestari et al (2017) hanya *chi square* saja. Rumus perhitungan elektabilitas yang digunakan PvT dan SoV, sedangkan Ramteke et al(2016) hanya menggunakan PvT saja dan Bermingham & Smeaton. (2011) hanya menggunakan SoV saja. Detail perbandingan penelitian yang sudah ada ditampilkan dalam tabel 2.1

Tabel 2.1 Perbandingan Penelitian yang Sudah Ada

No	Peneliti	Topik	Metode	Perbedaan
1.	Ramteke et al (2016)	Prediksi pemilu presiden di Amerika Serikat meggunakan analisis sentimen dari data Twitter	-Analisis Sentimen -SVM dan Multinominal Naïve Bayes -PvT	-Dataset -Metode Klasifikasi -Metode seleksi fitur *Chi Square* -Elektabilitas SoV
2.	Lestari et al (2017)	Analisis sentimen untuk mengetahui opini masyarakat terhadap Pilkada DKI 2017	-Analisis Sentimen -Naïve Bayes -Pembobotan emoticon	-Metode klasifikasi -Metode seleksi fitur *Chi Square* -Metode seleksi fitur TF-IDF -Dataset

Tabel 2.1 Perbandingan Penelitian yang Sudah Ada (lanjutan)

No	Peneliti	Topik	Metode	Perbedaan
3.	Hidayatullah & SN (2014)	Analisis sentimen dan klasifikasi tokoh publik berdasarkan data di Twitter	-Analisis Sentimen -Naïve Bayes dan SVM -Kategori klasifikasi tokoh public	-Dataset -Metode klasifikasi -Elektabilitas tokoh politik -Metode seleksi fitur *Chi Square*
4.	Virgo (2018)	Analisis Sentimen dan Klasifikasi Buzzer untuk prediksi Pilkada DKI 2017	-Analisis Sentimen -Multinomial Naïve Bayes & Gaussian Naïve Bayes -Deteksi Buzzer	-Dataset -Seleksi Fitur Chi Square dan TF-IDF -Elektabilitas tokoh politik
5.	Hayatin et al (2014)	Analisis sentimen terhadap tokoh publik berdasarkan data di Twitter	-Analisis Sentimen -Naïve Bayes -Pembobotan TF-IDF	-Bahasa dari data set -Metode klasifikasi -Jumlah tokoh publik -Metode Seleksi fitur *Chi Square*
6.	Ling et al (2014)	Analisis sentimen data telepon genggam dengan Naïve Bayes dan Pembobotan Chi Square	-Analisis Sentimen -Naïve Bayes -Metode seleksi fitur Chi Square	-Dataset -Metode klasifikasi -Metode seleksi fitur TF –IDF

Tabel 2.1 Perbandingan Penelitian yang Sudah Ada (lanjutan)

No	Peneliti	Topik	Metode	Perbedaan
7.	Bermingham & Smeaton. (2011)	Analisis sentimen untuk memprediksi hasil pemilihan presiden Irish berdasarkan data tweet	-Analisis Sentimen -SVM -MNB - Elektabilitas SoV	-Dataset -Seleksi fitur *chi square* dan TF-IDF -Elektabilitas PvT

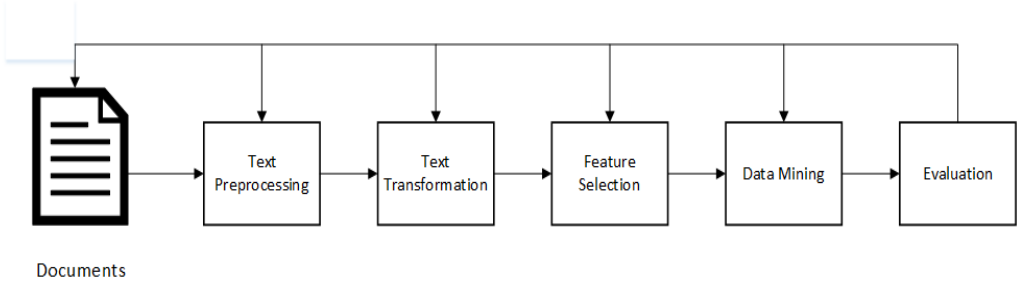
BAB III
LANDASAN TEORI

3.1 Text Mining

Informasi berupa teks adalah informasi yang penting dan cara mendapatkanya pun mudah. Informasi tersebut dapat kita cari dari berbagai sumber seperti buku, surat kabar, website ataupun email. Teks merupakan sebuah hamparan Bahasa , baik dalam pembicaraan ataupun tuisan

Text mining adalah suatu bidang didalam data mining dimana penggalian informasi dan pengelolaan sekumpulan dokumen menggunakan tools analisis. Gagasan utama dari text mining adalah mengetahui cakupan atau topik dari permasalahan dalam teks. Pengembalian informasi dari teks (text mining) antara lain dapat meliputi kategorisasi teks atau dokumen, analisis sentimen , pencarian topik yang lebih spesifik, serta spam filtering. Text mining penting dalam analisis sentimen sebagai pengidentifikasi emosional suatu pernyataan, sehingga banyak studi tentang analisis sentimen dilakukan. (Manning et al, 2008).

Berbeda dengan data mining yang biasanya memproses structured data, text mining biasanya digunakan untuk memproses *unstructured* atau minimal semi-structured data. Akibatnya text mining mempunyai tantangan tambahan yang tidak ditemui di data mining seperti struktur data yang kompleks dan tidak lengkap, arti yang tidak jelas dan tidak standard dan bahasa yang berbeda serta translasi yang tidak akurat.Oleh karena itu biasanya *Natural Language Processing* digunakan untuk memproses unstructured data text tersebut (Adiwijawa, 2006).



Gambar 3.1 Proses dalam Text Mining

Gambar 3.1 menjelaskan bagaima proses yang terjadi di dalam text mining. Proses text mining dimulai dengan text preprocessing yaitu pengubahan bentuk dokumen menjadi data struktur dengan cara seperti tokenisasi, *stop word removal* , *stemming*, normalisasi dan lain sebagainya. Dari hasil text processing selanjutnya dilakukan text transformation untuk menemukan fitur fitur yang tersimpan didalam data sesuai kebutuhan yang diperlukan. Setelah diketahui fitur fiturnya selanjutnya dilakukan feature selection untuk menentukan fitur yang berpengaruh atau tidak dalam pemodelan data menggunakan perankingan atau pembobotan fitur. Setelah itu, digunakan berbagai algoritma data mining untuk menemukan informasi atau pola yang menarik dari data yang terpilih. Selanjutnya dilakukan evaluasi model apakah pola atau informasi yang dihasilkan bertentangan dengan fakta atau hipotesa sebelumnya (Adiwijawa, 2006)

3.2 Twitter

Twitter adalah *micro-blogging* media sosial dimana user dapat memposting pesan singkat (maksimal 140 karakter) ke dunia maya yang biasa disebut tweet. Twitter memiliki lebih dari 250 juta pengguna bulanan dan terus bertambah setiap saat. Kepopuleran Twitter tidak hanya menarik perhatian masyarakat pada umumnya tetapi juga para peneliti yang ingin menggali informasi didalamnya dengan berbagai topic (Kwak et al, 2010)

Twitter menawarkan sebuah *Application Programming Interface* (API) yang dapat digunakan untuk mencrawling atau mengumpulkan data dari Twitter dengan mudah. Untuk data yang dapat dicrawling/diambil dari Twitter dapat berupa user profile, trending topik, dan tweet itu sendiri. Akan tetapi Twitter API dibagi menjadi 3 kategori :

1. Search API : search/tweets : Hanya dapat mengambil data tweet/search dalam 7 hari terakhir
2. 30-Day search API : Dapat mengambil data tweet/search dalam 30 hari terakhir
3. Full-archive search API : Dapat mengambil data tweet/search dari awal tahun 2006

3.3 Analisis Sentimen

Analisis sentimen adalah studi komputasi mengunakan natural language processing dari opini , sentimen dan emosi yang diekspresikan dalam teks. Tugas dasar dari dalam analisis sentiment adalah mengelompokan polaritas dari teks yang ada dalam dokumen, kalimat atau pendapat. Polaritas mempunyai pengertian apakah teks didalam dokumen, kalimat atau pendapat memiliki aspek positif atau negatif (Ling et al, 2014).

Apa yang orang lain pikirkan selalu menjadi bagian penting dari suatu informasi dalam proses *decision-making*. Menggunakan berbagai benda atau layanan bukan satu satunya alasan orang orang mencari tahu informasi atau mengekspresikan pendapat secara online. Kebutuhan akan informasi politik juga menjadi faktor lain. Sebagai contohnya, didalam pemilihen presiden Amerika tahun 2006 terdapat lebih dari 60 juta kampanye online dari pengguna internet terhdap calon presiden (Pang & Lee, 2008)

3.4 Preprocessing

Preprocessing adalah tahap penting yang dilakukan untuk membersihkan data atau merubah data menjadi bentuk data yang terstruktur. Proses membersihkan data meliputi pengecekan data yang tidak konsisten, menghapus data yang terduplikat dan mengkoreksi kesalahan yang terjadi saat penulisan teks (Wikarsa dan Thair, 2016).

Tahapan didalam preprocessing yang biasa dilakukan terdiri dari proses tokenisasi , stopword removal, dan stemming. Proses penghapusan URL , karakter khusus dalam Twitter, simbol dan tanda baca serta case folding juga dilakukan dalam tahap ini (Hidayatullah dan SN, 2014) .

3.4.1 Tokenisasi

Tokenisasi adalah proses pemotongan string input berdasarkan tiap kata penyusunnya.Pada prinsipnya adalah memisahkan setiap kata yang menyusun suatu dokumen. Pada proses ini dilakukan penghilangan angka, tanda baca dan karakter selain huruf alphabet, karena karakter-karakter tersebut dianggap sebagai pemisah

kata (delimiter) dan tidak memiliki pengaruh terhadap pemrosesan teks. Pada tahapan ini juga dilakukan proses *case folding*, dimana semua huruf diubah menjadi huruf kecil. Cleaning adalah proses membersihkan dokumen dari komponen-komponen yang tidak memiliki hubungan dengan informasi yang ada pada dokumen, seperti tag html, link, dan script , dan sebagainya (Ling et al, 2014). Contoh proses tokenisasi ditunjukan pada tabel 3.1

Tabel 3.1 Contoh Tokenisasi

Contoh Kalimat	Hasil Tokenisasi
Jokowi dan SBY akan bersaing di pilpres 2019	"Jokowi", "dan", "SBY", "akan", "bersaing" , "di" , "pilpres", "2019"

3.4.2 *Stopword Removal*

Stopword removal adalah tahap pemilihan kata kata penting dari hasil token, yaitu kata apa saja yang akan digunakan untuk mewakili dokumen. *Stopword* adalah kata kata yang tidak deskriptif (tidak penting) yang dapat dibuang dengan pendekatan *bag of words* (database kumpulan kata kata yang tidak deskriptif/tidak penting), kemudian kalau hasil tokenisasi itu ada yang merupakan kata tidak penting dalam database tersebut, maka hasil tokenisasi itu dibuang.Biasanya performa text mining ataupun *information retrieval* dapat ditingkatkan dengan *stopword removal* ini. Contoh dari stopwords adalah "aku", "mereka","di","ada" ,"atom" dan seterusnya. Sementara beberapa metode yang biasa digunakan untuk stopword removal adalah The Classic Method, Zipf's law, Mutual Information,dan Term Based Random Sampling (Vijayani et al, 2015). Contoh proses stopword removal ditunjukan pada tabel 3.2

Tabel 3.2 Contoh Stopword Removal

Contoh kalimat	Hasil stopword removal
Jokowi dan SBY akan bersaing di pilpres 2019	“Jokowi”, “SBY” , “bersaing” , “pilpres” , “2019”

3.4.3 Stemming

Stemming adalah proses pengubahan bentuk kata menjadi kata dasar atau tahap mencari root kata dari tiap hasil. Dengan dilakukanya proses stemming setiap kata berimbuhan akan berubah menjadi kata dasar , dengan demikian dapat lebih mengoptimalkan proses teks mining. Terdapat 2 poin penting yang dipertimbangkan dalam proses stemming :

1. Kata yang tidak memiliki makna yang sama lebih baik disimpan terpisah
2. Bentuk morfologi dari suatu kata yang memiliki makna dasar yang sama lebih baik dipetakan kedalam stem yang sama

Dua aturan ini cukup baik digunakan dalam teks mining atau language processing. Stemming biasanya dipertimbangkan sebagai *recall-enhancing device*. Untuk bahasa yang relatif simpel morfologinya, stemming tidak dapat bekerja optimal dibandingkan untuk bahasa yang kompleks morfologinya. Kebanyakan eksperimen stemming ini diaplikasikan untuk bahasa inggris (Vijayani et al, 2015).

Pada bahasa Indonesia terdapat kompleksitas pada variasi imbuhan yang menjadi pembentukan kata dasarnya. Algoritma stemming yang pertama kali digunakan adalah Algoritma Nazief et al., (2007), mengacu pada algoritma *Porter Stemmer* yang digunakan pada bahasa inggris. Selanjutnya muncul algoritma baru yang meminimalisir kekurangan kekurangan yang adaAlgortima *Config Stripping Stemmer* dianggap menjadi algoritma stemming yang paling efektif saat itu meskipun masih ada kesalahan stemming yang terjadi. Namun seiring berjalanya waktu muncul algoritma *Enchanced Confix Stripping Stemmer* (ECS) yang menjadi perbaikan dari algoritma *Confix Stripping Stemmer* (Anggara, 2013).

Perbaikan yang dilakukan oleh ECS *Stemmer* adalah perbaikan beberapa aturan pada tabel referensi pemenggalan imbuhan. Selain itu, algoritma ECS *Stemmer*. Selain itu algoritma ECS stemmer juga menambahkan langkah pengembalian akhiran jikalau terjadi penghilangan akhiran yang semestinya tidak dilakukan. Akan tetapi, ECS *Stemmer* masih memiliki kelemahan, diantaranya keterbatasan dalam menstemming kata yang memiliki sisipan, dan kekurangan masalah *overstemming* dan *understemming* (Tahitoe & Purwitasari, 2010). Oleh sebab itu , Tahitoe & Purwitasari (2010) melakukan penelitian untuk memperbaiki kesalahan stemming yang dilakukan oleh algortima ECS *Stemmer*.

Berikut detail Algoritma Perbaikan *Enchanced Confix Stripping Stemmer* Pengembangan algoritma stemming *Enchanced Confix Stripping Stemmer* dilakukan karena masih banyak kesalahan yang stemming pada algoritma *Enhanced Confix Stripping Stemmer*, antara lain ditunjukan pada tabel 3.3 :

Tabel 3.3 Contoh Kegagalan Algoritma ECS

(Tahitoe dan Purwitasari, 2010)

Tipe Kesalahan	Contoh Kasus		
	Awal	*Stemming*	Seharusnya
Sisipan	Temaram	temaram	Taram
Overstemming	Penyidikan	Sidi	Sidik
Understemming	Mengalami	Alami	Alam
Nama Orang	Gumai	Guma	Gumai
Kesalahan beberapa aturan pemenggalan	Menyatakan	Menyatakan	Nyata
Kata gabungan	Diberitahu	Diberitahu	Beritahu

Dalam pengembangan tersebut, dilakukan beberapa hal untuk membuat algoritma *Enhanced Confix-Stripping Stemmer* lebih optimal, antara lain :

1. Sisipan

Imbuhan dalam bahasa Indonesia mengenal adanya sisipan, yang terdiri dari “er”, “el”, “em” dan “in”. Aturan yang dibuat sebelumnya hanya mengenal imbuhan yang berupa awalan dan akhiran. Untuk itu ditambahkan aturan reduksi untuk sisipan guna memperbaiki kesalahan *stemming* untuk kata yang memiliki sisipan. Proses reduksi sisipan dilakukan setelah proses reduksi awalan dan akhiran selesai dilakukan.

2. Kesalahan aturan pemenggalan

Dilakukan revisi pada beberapa aturan yang masih menimbulkan kesalahan *stemming*

3. Kata gabungan

Ditambahkan langkah untuk melakukan pengecekan keberadaan kata turunan dalam algoritma ECS *Stemmer*. Proses ini dilakukan apabila tidak ditemukan bentuk dasar dari kata yang dimasukkankan pada kamus kata dasar setelah proses reduksi awalan dan akhiran selesai dilakukan. Ketika kata dasar tidak ditemukan, maka proses *stemming* kata yang dimasukkan diulangi sekali lagi. Namun kali ini yang dilakukan adalah pengecekan keberadaan kata gabungan. Hal tersebut dilakukan setelah proses reduksi awalan dan akhiran. Masing-masing kata pada kata turunan yang dimasukkan tentu saja harus terdapat pada kamus kata dasar yang dipergunakan.

4. Akhiran serapan bahasa asing

Untuk melakukan reduksi akhiran yang berasal dari serapan bahasa asing, yang perlu dilakukan tentu saja adalah melakukan pendaftaran akhiran serapan bahasa asing ke dalam tabel aturan pemenggalan imbuhan. Akhiran serapan bahasa asing tersebut, yakni ”-wan”, “- wati”, ”-is”, ”-isme”, dan “-isasi”

5. Nama orang, *overstemming*, dan *understemming*

Overstemming dan *understemming* terjadi jika hasil *stemming* dari suatu *term* berjumlah lebih dari satu. Pengembangan ECS ini dapat memilih hasil *stemming* berdasarkan koleksi dokumen yang digunakan

3.4.4 Case Folding

Sebuah dokumen teks memuat berbagai karakter (baik huruf, tanda baca, maupun angka). Huruf-huruf didalam dokumen tersebut juga bisa berupa huruf besar ataupun huruf kecil. *Case folding* dilakukan untuk membuat huruf seragam yaitu dibuat menjadi huruf kecil saja. Karakter selain huruf akan dihilangkan. Tujuan dari proses ini untuk menghilangkan karakter-karakter selain huruf pada saat pengambilan informasi (Fathan & SN, 2014)

3.5 Regular Expression

Regular expression (regex) adalah sebuah pola penggambaran dari sejumlah text. Nama regex berasal dari salah satu teori matematika dengan nama yang sama. Regex secara jelas memisahkan pola dari text disekitarnya dan tanda bacanya. (Goyvaerts, 2007). Contoh penggunaan regular expression adalah sebagai berikut :

1. “[\(\[].*?[\)\]]” yang merupakan regular expression untuk pengilangan bracket
2. “http\S+” yang merupakan regular expression untuk menghapus URL
3. “[^A-Za-z0-9]+” yang merupakan regular expression untuk membuat dokumen berisi hanya *alpha numeric* yaitu huruf dan angka saja
4. "@\S+" "#\S+" yang merupakan regular expression untuk mengapus mention dan hastag dari dokumen

Regular expression dapat diubah-ubang tergantung setiap kondisi text dalam dokumen yang diinginkan.

3.6 *Term Frequency-Inverse Document Frequency*

Term frequency adalah total frekuensi munculnya sebuah kata term dalam *corpus*.Untuk menghitung term frequency, melibatkan jumlah semua kejadian dari kata dalam semua dokumen dalam *corpus*. Untuk lebih jelasnya , rumus untuk mencari nilai *term frequency* dapat dilihat pada persamaan (3.1)

$$tf(ti, dj) = \frac{fij}{\max\{f(w,d):w \in d\}} \quad (3.1)$$

dimana :

fij = frekuensi kemunculan kata t_i pada dokumen d_j

$max\{f(w,d):w \in d\}$=nilai maksimum yang dihitung menggunakan frekuensi dari seluruh *term* yang muncul pada dokumen *dj*

Inverse document frequency (IDF) adalah nilai yang menyatakan bahwa semakin jarang sebuah term muncul dalam dokumen-dokumen yang ada didalam *corpus*, maka semakin relevan *term* tersebut. Metode IDF ditambahkan karena *term frequency* dinilai terlalu sederhana dalam mengukur tingkat pentingnya sebuah term karena tidak melibatkan informasi secara global dalam *corpus*. IDF dapat membantu dalam mebedakan satu dokumen dengan dokumen-dokumen lainya (Siddiqi & Sharan, 2015)

$$Idf\,(t_i,d_j) = \log\left(\frac{|N|}{1+|\{d \in D: ti \in d\}|}\right) \quad (3.2)$$

dimana :

|N| = jumlah total seluruh dokumen

$|\{d \in D: ti \in d\}|$ = banyaknya dokumen dimana suatu kata (t_i) muncul

Untuk menghitung TF-IDF, maka hal yang dilakukan adalah mengalikan nilai dari *term frequency* dengan nilai IDF dari suatu term tersebut.
Rumus dari TF-IDF dapat dilihat pada persamaan (3.3)

$$\text{TF-IDF}(ti,dj) = tf(ti,dj)\ x\ idf(ti,dj) \quad (3.3)$$

Dimana :

TF-IDF(t_i,d_j)= bobot TF-IDF kata ke-i dalam dokumen d_j

Tf(t_i,d_j) = *term frequency* kata ke-i dalam dokumen d_j

Idf(t_i,d_j) = *inverse document frequency* kata ke-i dalam dokumen d_j

Contoh data yang digunakan dalam proses seleksi fitur dapat dilihat pada tabel 3.4. Contoh perhitungan TF-IDF dapat dilihat pada tabel 3.5

Tabel 3.4 Contoh Data yang Digunakan dalam Proses Seleksi Fitur

Data Latih	No	Berita	Sentimen
	1	lemah agus yudhoyono	negatif
	2	agus yudhoyono sambang wapres jk makassar	positif
	3	agus yudhoyono temu kerja sesuai bakat bakat kerja	positif

Tabel 3.5 Contoh Perhitungan TF-IDF

jumlah N = 3								
Term			DF			IDF		
agus	3		-0.29	jk		1		0.40
yudhoyono	3		-0.29	makassar		1		0.40
lemah	1		0.40	sambang		1		0.40
bakat	1		0.40	wapres		1		0.40
kerja	1		0.40	sesuai		1		0.40
				temu		1		0.40
Term	tf-d1	tf-d2	tf-d3	idf	tfidf1	tfidf2	tfidf3	tf-idf
agus	0.33	0.16	0.125	-0.29	-0.096	-0.046	-0.036	-0.178
yudhoyono	0.33	0.16	0.125	-0.29	-0.096	-0.046	-0.036	-0.178
lemah	0.33	0	0	0.40	0.132	0	0	0.132
bakat	0	0	0.25	0.40	0	0	0.1	0.1
kerja	0	0	0.25	0.40	0	0	0.1	0.1
jk	0	0.16	0	0.40	0	0.064	0	0.064
makassar	0	0.16	0	0.40	0	0.064	0	0.064
sambang	0	0.16	0	0.40	0	0.064	0	0.064
wapres	0	0.16	0	0.40	0	0.064	0	0.064
sesuai	0	0	0.125	0.40	0	0	0.05	0.05
temu	0	0	0.125	0.40	0	0	0.05	0.05

3.7 Chi Square

Seleksi fitur digunakan untuk mereduksi fitur yang tidak relevan dalam proses klasifikasi. Seleksi fitur *chi square* menggunakan teori statistika untuk menguji independensi sebuah term dengan kategorinya.

Penyeleksian fitur *chi square* dilakukan dengan cara mengurutkan setiap berdasarkan fitur berdasarkan hasil seleksi fitur *chi square* dari nilai yang terbesar hingga terkecil. Nilai seleksi fitur Chi Square yang lebih besar dari nilai signifikan menunjukan penolakan hipotesis independensi. Sedangkan jika dua peristiwa menunjukan dependen, maka fitur tersebut menyerupai atau sama dengan label kategori sesuai pada kategori (Ling et al, 2014). Rumus *chi square* dapat dilihat pada persamaan (3.4)

$$X^2(D,t,c) = \sum_{et=\{0,1\}} \sum_{ec=\{0,1\}} \frac{(N_{etec} - E_{etec})^2}{E_{etec}} \quad (3.4)$$

dimana :

$\chi 2\ (D,t,c)$ = merupakan nilai Chi Square dari term t untuk kelas c

N_{etec} = *observed value* (jumlah term t pada kelas c)

E_{etec} = *expected value*

Sementara contoh untuk mencari nilai salah satu *expected value* dapat dilihat pada persamaan (3.5)

$$E_{11} = N \times P(t) \times P(c) = N \times \frac{N11+N10}{N} \, x \, \frac{N11+N01}{N} \quad (3.5)$$

dimana :

N = jumlah dokumen

N_{11} = Jumlah kemunculan term t pada kelas c

N_{10} = Jumlah kemunculan term t pada kelas bukan c

N_{11} = Jumlah kelas c yang memuat term t

N_{01} = Jumlah kelas c yang tidak memuat term t

Tahap selanjutnya dari Chi Square adalah melakukan perangkingan terhadap nilai chi square masing masing term.Contoh perhitungan Chi Square dapat dilihat pada tabel 3.6 dengan contoh data yang digunakan pada tabel 3.4

Tabel 3.6 Contoh Perhitungan Chi Square Positif

Term	Observed	Expected	Chi-Square
agus	2	2	0.0
yudhoyono	2	2	0.0
lemah	0	2/3	0.6666666666666666
bakat	2	4/3	0.3333333333333334
kerja	2	4/3	0.3333333333333334
jk	1	2/3	0.1666666666666667
makassar	1	2/3	0.1666666666666667
sambang	1	2/3	0.1666666666666667
wapres	1	2/3	0.1666666666666667
sesuai	1	2/3	0.1666666666666667
Temu	1	2/3	0.1666666666666667

3.8 Multinominal Naïve Bayes

Multinomial Naïve Bayes merupakan metode pembelajaran probabilitas. Multinomial Naïve Bayes diterapkan tanpa memperhitungkan urutan kata dan informasi yang ada dalam kalimat atau dokumen secara umum. Dalam menghitung peluang sebuah kata i masuk kedalam kategori j dapat dilakukan dengan mengunakan perhitungan *likelihood*/*conditional probability* data uji yang ditambahkan dengan angka satu untuk menghindari nilai nol (Manning et al, 2008). Probabilitas suatu dokumen d dalam kelas c dihitung menggunakan rumus pada persamaan 3.6

$$P(c|d) \propto P(c) \prod_{1 \le k \le nd} P(t_k|c) \quad (3.6)$$

Dimana P(tk|c) adalah probabilitas kondisional dari term tk yang muncul dalam kelas c. P(c) adalah probabilitas *prior* dari dokumen yang muncul pada kelas c. Didalam klasifikasi text tujuan kita adalah menemukan kelas terbaik dari suatu dokumen. Kelas terbaik didalam klasifikasi Naïve Bayes merupakan *maximum a posteriori* (MAP) yang rumusnya terdapat pada persamaan 3.7

$$C_{map} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{1 \le k \le nd} P(t_k|c) \qquad (3.7)$$

Didalam persamaan (3.7) banyak probabilitas kondisional yang dikalikan, satu untuk setiap $1 \le k \le n_d$. Untuk mendapatkan performa komputasi yang lebih baik, logaritma digunakan untuk menggantikan probabilitas perkalian. Kelas dengan hasil probabilitas log yang tinggi adalah yang baik. Rumus perbaikan ini ditampilkan pada persamaan (3.8)

$$C_{map} = \underset{c \in C}{\operatorname{argmax}} [\log P(c) + \sum_{1 \le k \le nd} \log P(t_k|c) \qquad (3.8)$$

Dari persamaan diatas, setiap parameter kondisional $\log P(t_k|c)$ merupakan bobot yang mengindikasi seberapa bagus indikator t_k untuk c. Semakin sering kelasnya muncul semakin akan semakin baik menjadi kelas yang benar daripada kelas yang jarang muncul. Jumlah log prior dan bobot menjadi bukti bahwa terdapat dokumen dalam suatu kelas.

Untuk memulai menghitung klasifikasi Multinomial Naïve Bayes, dihitung probabilitas prior dari suatu kelas menggunakan rumus pada persamaan (3.9)

$$P(c) = \frac{Nc}{N} \qquad (3.9)$$

Dimana N_c adalah jumlah dokumen pada kelas C dan N adalah jumlah seluruh dokumen. Probabilitas kondisional P(t|c) dihitung berdasarkan frequency term t yang termasuk dalam kelas c. Rumus probabilitas kondisional P(t|c) ditampilkan dalam persamaan (3.10)

$$P(t|c) = \frac{Tct}{\sum_{t;\in V} Tct\prime} \qquad (3.10)$$

Dimana T_{ct} merupakan jumlah kemunculan t dalam dokumen *training* dari kelas c, termasuk kemunculan ganda suatu term dalam dokumen. Namun terdapat kelemahan dalam persamaan ini yaitu dihasilkanya nilai 0 untuk term yang tidak muncul dalam *training* data. Unutk mengatasinya dilakukan persamaan Laplace yang dituliskan pada persamaan 3.11

$$P(t|c) = \frac{Tct+1}{\sum_{t\prime \in V}(Tct'+1)} = \frac{Tct+1}{\sum_{t\prime \in V}(Tct')+B\prime} \qquad (3.11)$$

Dimana B = |V| yaitu jumlah term dalam vocabulary.

Pseudo code dari Multinomial Naïve Bayes ditunjukan pada gambar 3.2

TrainMultinomialNB(C, D)
1 $V \leftarrow$ ExtractVocabulary(D)
2 $N \leftarrow$ CountDocs(D)
3 **for each** $c \in$ C
4 **do** $Nc \leftarrow$ CountDocsInClass(D, c)
5 *prior* [c] $\leftarrow Nc/N$
6 *textc* $\leftarrow$ ConcatenateTextOfAllDocsInClass(D, c)
7 **for each** $t \in V$
8 **do** $Tct \leftarrow$ CountTokensOfTerm(*textc*, t)
9 **for each** $t \in V$
10 **do** *condprob*[t][c] $\leftarrow \frac{Tct+1}{\sum_{t' \in V}(Tct'+1)}$
11 **return** V, *prior*, *condprob*

ApplyMultinomialNB(C, V, *prior*, *condprob*, d)
1 $W \leftarrow$ ExtractTokensFromDoc(V, d)
2 for each $c \in$ C
3 do *score*[c] $\leftarrow$ *prior* [c]
4 for each $t \in W$
5 do *score*[c] += *condprob*[t][c]
6 return arg max$c \in$C *score*[c]

Gambar 3.2 Pseudo Code Training dan Testing Multinomial Naïve Bayes

3.9 Evaluasi Performa

Evaluasi digunakan untuk mengukur kinerja suatu sistem, untuk penelitian ini evaluasi digunakan untuk mengetahui akurasi metode klasifikasi yang digunakan. Ada beberapa teknik evaluasi untuk mengukur keakuratan dan keefektifan suatu sistem klasifikasi diantaranya *precision*, *recall*, dan *f-measure* (Ling, 2014) .

Model klasifikasi yang dibuat adalah pemetaan baris data dengan keluaran data dengan keluaran sebuah hasil prediksi kelas/target dari data tersebut. Hasil klasifikasi memiliki dua kelas keluaran yang biasa direpresentasikan dala {0,1}, {+1,-1}, atau {positif,negatif} (Rianto, 2016).

Dalam proses evaluasi klasifikasi terdapat 4 kemungkinan yang terjadi dalam proses klasifikasi suatu baris data. Jika data positif dan diprediksi positif akan

dihitung sebagai true positif, tetapi jika data diprediksi negatif maka akan dihitung sebagai false negatif.Jika data negatif dan diprediksi negatif akan dihitung sebagai true negative, tetapi jika data tersebut diprediksi positif maka akan dihitung sebagai false positif. Hasil klasifikasi tersebut di representasikan kedalam matriks yang disebut *confusion matrix* (Fawcett, 2005).Contoh *confusion matrix* ditunjukan pada tabel 3.7

Tabel 3.7. Contoh *Confusion Matrix*

Confusion Matrix		Nilai Prediksi	
		True	False
Nilai Sebenarnya	True	TP (True Positif)	FN (False Negatif)
	False	FP (False Positif)	TN (True Negatif)

3.9.1 Akurasi

Akurasi adalah jumlah proporsi prediksi yang benar. Adapun rumus perhitungan akurasi dapat dilihat pada persamaan 3.12 (Fawcett, 2005)

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.12}$$

3.9.2 *Presisi*

Presisi adalah proporsi jumlah dokumen teks yang relevan terkenali diantara semua dokumen teks yang terpilih oleh sistem. Rumus presisi dapat dilihat pada persamaan 3.13 (Fawcett, 2005)

$$\text{Presisi} = \frac{TP}{TP+FP} \tag{3.13}$$

3.9.3 *Recall*

Recall adalah proporsi jumlah dokumen teks yang relevan terkenali diantara semua dokumen teks relevan yang ada pada koleksi. Rumus dapat dilihat pada persamaan 3.14 (Fawcett, 2005)

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.14}$$

3.9.4 *F-Measure*

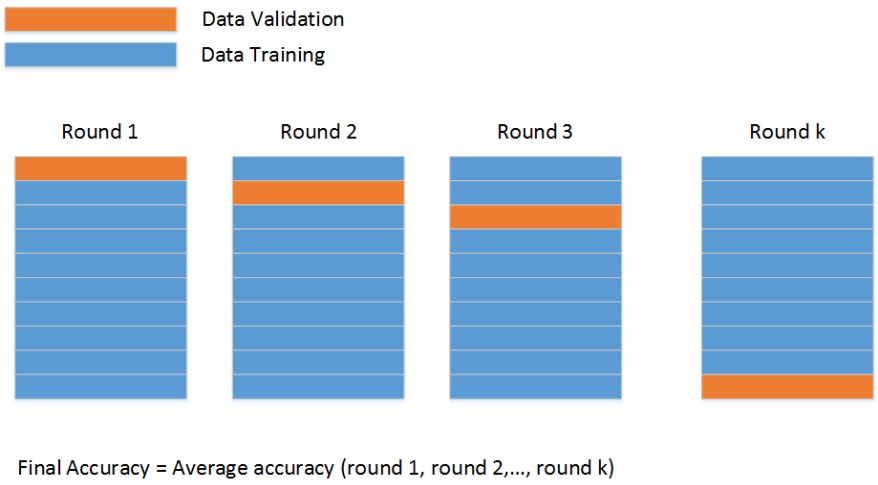
F-Measure adalah nilai yang mewakili seluruh kinerja sistem yang merupakan rata-rata harmonic dari presisi dan recall. rumus *f-measure* dapat dilihat pada persamaan 3.15 (Fawcett, 2005)

$$\text{f-measure} = 2 \text{ x } \frac{Presisi\ x\ Recall}{Presisi+Recall} \tag{3.15}$$

3.9.5 *Cross Validation*

Cross validation adalah metode statistika untuk melakukan evaluasi dan membandingkan algoritma pembelajaran dengan cara membagi data menjadi dua bagian, yaitu data latih dan data uji.

Dalam *k-fold cross validation*, data training dibagi sejumlah k *subset* menjadi D_1, D_2,…., D_k dimana setiap subset memiliki ukuran data yang sama. Kemudian percobaan dilakukan sebanyak k-kali dengan tiap iterasi ke-i, D_i menjadi data validation, sedangkan data latih adalah subset lain selain D_i. Gambar 3.3 mengilustrasikan bagaimana proses pembagian data latih (*training set*) dan data validation (*validation set*) pada k-fold cross validation. Nantinya akan didapatkan akurasi tiap iterasi dan di rata rata untuk mendapatkan *final accuracy*.



Gambar 3.3. Ilustrasi *K-Fold Cross Validation*

3.10. Positive Versus Total (PvT)

Positive versus Total (PvT) adalah cara untuk menghitung elektabilitas tokoh dengan membandingkan jumlah sentiment positif dari masing-masing tokoh dengan jumlah total sentimen (positif dan negatif) nya (Ramteke J. et al, 2016). Rumus PvT dapat dilihat pada persamaan (3.16)

$$\text{Rasio} = P/T \tag{3.16}$$

Dimana :

P : jumlah sentimen positif hasil klasifikasi tiap tokoh politik

T : jumlah seluruh sentimen hasil klasifikasi tiap tokoh politik

Contoh perhitungan *positive versus total* dapat dilihat pada tabel 3.8

Tabel 3.8 Contoh Perhitungan *Positive versus Total*

Tokoh	Positif	Negatif	Total	Elektabilitas
Agus	221	97	318	0.6949
Anies	650	274	924	0.7034
Ahok	515	166	681	0.7562

3.11. Share of Volume (SoV)

Share adalah cara menghitung elektabilitas tokoh politik dengan membandingkan jumlah sentimen positif seorang tokoh politik dengan total sentimen positif keseluruhan tokoh politik. SoV memiliki keuntungan bahwa hasilnya bisa dibandingkan dengan mudah dengan hasil presentasi *polling* (Bermingham & Smeaton, On Using Twitter to Monitor Political Sentiment and Predict Election Result, 2011). Rumus perhitungan SoV ditunjukan pada persamaan (3.17)

$$\text{SoV(x)} = \frac{|Rel\ (x)|}{\sum_{i=1}^{n} |Rel(i)|} \tag{3.17}$$

Dimana :

SoV(x) : *share of volume* untuk tokoh politik x

Rel (i) = Jumlah tweet positif dari tokoh dari masing-masing tokoh politik

BAB IV

ANALISIS DAN PERANCANGAN

4.1. Analisis Permasalahan

Setiap partai politik yang ingin mengajukan kandidatnya untuk mengikuti pemilihan gubernur, kepala daerag, atau pemilihan umum membutuhkan tolak ukur citra para kandidatanya di mata masyarakat. Salah satu tolak ukur tersebut adalah elektabilitas. Perhitungan elektabilitas tokoh politik biasanya menggunakan metode survey atau *quick count* . Namun dua metode ini memiliki kekuranganya masing-masing. Survey merupakan metode konvensional, membutuhkan waktu lama untuk mengumpulkan data , pengolahan data dan analisisnya, dan belum tentu hasil akurasinya tinggi karena survey bisa saja tidak objektif. Sementara *quick count* yang merupakan metode modern, membutuhkan biaya yang mahal untuk pelaksanaanya, selain itu *quick count* dilakukan saat pemilu berlangsung sehingga tokoh politik tidak bisa menentukan bagaimana strategi pemilu atau politik yang baik. Sementara itu, metode analisis sentimen lebih ilmiah ,memiliki akurasi yang tinggi ,serta biayanya tidak mahal.

Analisis sentimen disini dilakukan untuk mendapatkan sentimen positif atau negatif dari 10 tokoh politik yang sudah ditentukan yaitu Agus Yudhoyono, Prabowo Subianto, Ahok, Jokowi, Anies Baswedan, Gatot Nurmantyo, Jusuf Kalla, Hary Tanoe, Ridwan Kamil, dan Zulkifli Hasan. Metode seleksi fitur *chi square* dan TF-IDF akan dilakukan untuk membandingkan metode seleksi fitur manakah yang memiliki akurasi yang lebih baik.Setelah itu, dari ke 10 tokoh politik tersebut akan dihitung elektabilitas tokoh politiknya masing masing dengan rumus *positif versus total* dan *share of volume*.

4.2. Rancangan Umum Sistem

Penelitian ini bertujuan untuk mengetahui bagaimana elektabilitas seorang tokoh politik berdasarkan analisis sentimen dari portal berita dan media sosial (Twitter) berbahasa Indonesia. Untuk itu dibutuhkan sistem yang mampu melakukan klasifikasi sentimen.

Proses klasifikasi sentimen dilakukan dengan menggunakan data tweet yang didapatkan dengan menggunakan tools Chorus Tweet Catcher yang memuat Twitter API sehingga bisa dilakukan pencarian dengan berupa kata kunci yang tertera dalam tabel 4.1 Sementara data berita didapatkan dari portal berita online viva.co.id,tempo.co dan tribunnews.com dengan menggunakan tools Scrapper untuk mendapatkan judul berita dan tanggal berita. Dataset ini diambil dalam kurun waktu 17 November 2016 sampai 1 November 2017. Tweet dan berita yang sudah diambil kemudian dilabeli secara manual. Data yang sudah dilabeli kemudian dibaca dan dimasukan kedalam proses *preprocessing* data. Preprocessing data dimaksudkan untuk mengurangi *noise* pada data dengan mengganti dan menghilangkan fitur-fitur yang tidak diperlukan.Setelah *preprocessing* selanjutnya data dimasukan kedalam proses seleksi fitur. Seleksi fitur dilakukan untuk memilih sejumlah fitur *top-n* yang dianggap sudah merepresentasikan keseluruhan data sehingga hasilnya klasifikasi tidak *overfitting*.Setelah itu, algoritma Multinomial Naïve Bayes digunakan untuk melakukan proses klasifikasi sentimen.

Proses pelatihan dan pengujian model kemudian dilakukan untuk melihat seberapa bagus model yang dihasilkan pada proses klasifikasi sentiment yang mengklasifikasikan data menjadi kelas positif dan negatif. Pengujian model dilakukan dengan emnggunakan metode *10-fold cross validation*. Parameter keberhasilan terletak pada akurasi yang dihasilkan dari pengujian tersebut.

Setelah semua tweet dan berita terklasifikasi positif dan negatif, hasil klasifikasi sentimen tersebut digunakan untuk menghitung nilai elektabilitas dari masing-masing tokoh politik. Perhitungan nilai elektabilitas tokoh politik dilakukan dengan dua cara yaitu :

1. *Positive versus total*, dengan membagi jumlah sentimen positif dan jumlah total sentimen (jumlah data) dari tokoh politik tersebut.
2. *Share of Volume*, dengan membagi jumlah sentimen positif seorang tokoh politik dengan total sentimen positif dari keseluruhan tokoh politik.

Perhitungan nilai elektabilitas dilakukan untuk data yang menggunakan seleksi fitur TF-IDF dan chi square dan tanpa seleksi fitur.Hasil perhitungan nilai elektabilitas PvT dan SOV akan dirata-rata untuk mendapatkan nilai elektabilitas akhir dari masing-masing tokoh politik.

4.3. Perancangan Data

Data yang akan digunakan meliputi dataset (berita dan tweet), data *stopword* dan data kata dasar yang akan digunakan dalam sistem

4.3.1. Dataset (berita dan tweet)

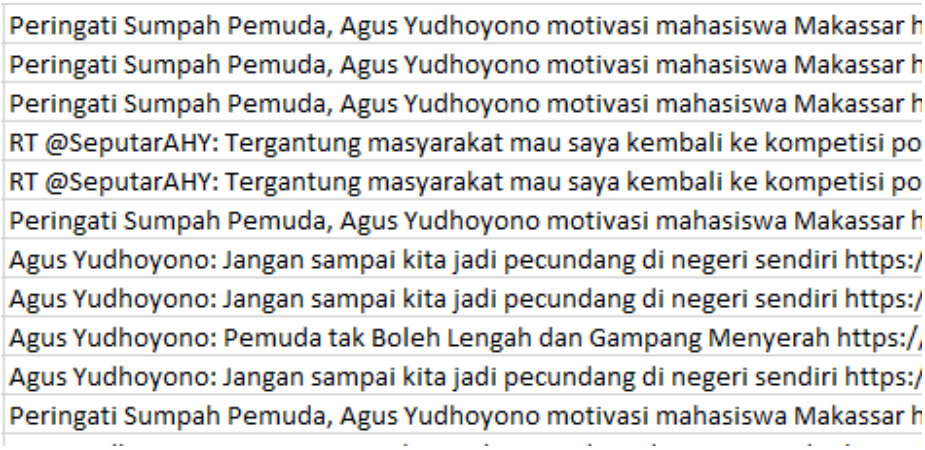
Penelitian ini menggunakan data berita yang diambil dari 3 situs berita *online* berbahasa Indonesia. Data didapatkan dengan teknik *scrapping* menggunakan *tools scrapper* untuk mengambil judul berita dan tanggal berita. Berikut adalah situs-situs yang digunakan sebagai sumber data berita penelitian ini:

1. tribunnews.com
2. viva.co.id
3. tempo.co

Sementara itu, data tweet didapatkan dengan teknik scrapping dengan menggunakan *tools chorus* dengan memasukan kata kunci berupa nama tokoh politik. Dataset diambil dalam rentang waktu 17 Oktober 2016 - 19 Oktober 2017 untuk dataset judul berita dan 19 Oktober 2017 – 1 November 2017 untuk dataset Twitter. Setelah data di *scrapping*, masing masing data berita dan tweet dimasukan kedalam suatu file berekstensi .csv. Total jumlah data yang digunakan dalam penelitian ini sebanyak 23.001 data tweet dan berita. Berikut cuplikan data berita dapat dilihat pada gambar 4.1 dan data tweet dapat dilihat pada gambar 4.2

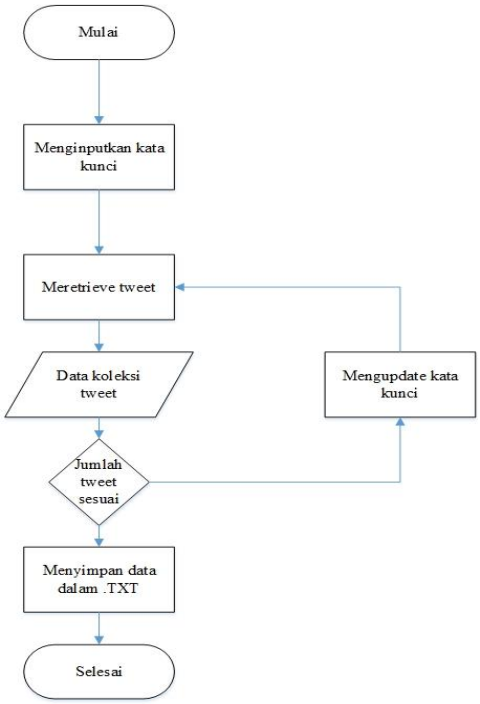
PDIP Anggap Pertemuan Jokowi dan SBY sebagai Politik Makana	30 Okt 2017 | 08:32 WIB
Puluhan Orang Deklarasikan Cak Imin-AHY Jadi Capres-Cawapres	29 Okt 2017 | 17:30 WIB
Teka-teki Politikus Muda Pendamping Khofifah di Pilgub Jatim	28 Okt 2017 | 18:22 WIB
AHY Mau Nyapres, Ini Saran dari PDIP	22 Okt 2017 | 23:15 WIB
Jika Gandeng AHY Jadi Wapres, Jokowi Diprediksi Menang Telak	22 Okt 2017 | 20:13 WIB
PPP Munculkan Nama AHY Jadi Pendamping Khofifah	19 Okt 2017 | 16:21 WIB
AHY Pamerkan 'Surat Cinta' dari Ahok	17 Okt 2017 | 19:58 WIB
Dede Yusuf Benarkan Spanduk AHY Bertebaran Terkait Pilpres	2 Jun 2017 | 21:44 WIB
Kalah di DKI, AHY Tetap Dianggap Anak Kandung Demokrat	2 Jun 2017 | 15:56 WIB
Roy Suryo: Aspirasi Kader Ingin AHY Maju ke Pilpres 2019	31 Mei 2017 | 14:15 WIB
Demokrat Beri Sinyal AHY Siap Maju Pilpres	20 Mei 2017 | 12:54 WIB
Politikus Demokrat: Akar Rumput Ingin AHY Maju Pilpres	10 Mei 2017 | 13:20 WIB

Gambar 4.1 Cuplikan Data Berita

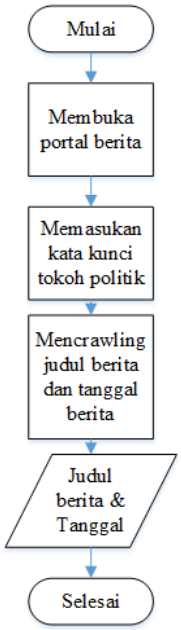


Gambar 4.2 Cuplikan Data Tweet

Pengambilan data tweet dimulai dengan memasukan kata kunci pada tabel 4.1 dalam *tools chorus*. *Chorus* akan mengumpulkan tweet dalam kurun waktu maksimal 7 hari sebelum tanggal pencarian. Tweet yang sudah terkumpul akan disimpan dalam file berformat .txt. Jika jumlah tweet belum memenuhi maka akan digunkan kata kunci pencarian yang lain. Pengambilan data berita dimulai dengan membuka portal berita dan memasukan kata kunci pada tabel 4.1 dalam kolom pencarian. Berita hasil pencarian kemudian dicrawling menggunakan tools scrapper untuk mendapatkan judul dan tanggal berita untuk masing-masing tokoh politik. Detail proses pengambilan data tweet dan berita ditunjukan pada gambar 4.3 dan 4.4



Gambar 4.3 Diagram Alur Scraping Data Twitter setiap Kata Kunci dengan Chorus



Gambar 4.4 Diagram Alur Crawling Data Berita setiap kata kunci dengan Scrapper

Daftar kata kunci yang digunakan untuk proses pengambilan data dapat dilihat pada tabel 4.1

Tabel 4.1 Daftar Kata Kunci Yang Digunakan

Tokoh Politik	**Kata Kunci**
Prabowo Subianto	“Prabowo Subianto”
Basuki Tjahaja Purnama	“Ahok”
Joko Widodo	“Jokowi” , “Joko Widodo”
Anies Baswedan	“Anies Baswedan” ,”Anies”
Gatot Nurmantyo	“Jenderal Gatot”,”Gatot Nurmantyo”
Jusuf Kalla	“Yusuf Kalla”,“Jusuf Kalla”
Hary Tanoe	“Hary Tanoe”
Ridwan Kamil	“Ridwan Kamil”
Zulkifli Hasan	“Zulkifli Hasan”
Agus Yudhoyono	“Agus Yudhoyono”

Kata kunci tiap tokoh memiliki jumlah yang berbeda, ada yang 1 ada yang 2, hal ini dikarenakan ada tokoh yang menggunakan 1 kata kunci saja dataset yang didapat masih sedikit.Dataset yang sudah didapatkan kemudian dibagi menjadi data *training* dan data *testing* dengan rasio 70%:30% yaitu 16.035 data *training* dan 6.966 data *testing*.Pembagian data *training* dan *testing* dapat dilihat pada tabel 4.2

Tabel 4.2 Komposisi Jumlah Data Tiap Tokoh Politik

Tokoh Politik	Jumlah data *training*			Jumlah data *testing*
	Total	Positif	Negatif	
Agus Yudhoyono	770	647	123	318
Ahok	1375	848	527	681
Anies Baswedan	2120	1472	648	923
Gatot Nurmantyo	1756	533	1223	755
Hary Tanoe	899	793	106	385
Jusuf Kalla	1026	781	245	453
Jokowi	3312	2438	873	1420
Ridwan Kamil	1943	1457	486	833
Zulkifli Hasan	1568	1403	162	671
Prabowo Subianto	1221	1018	189	525

4.3.2. Data *stopword*

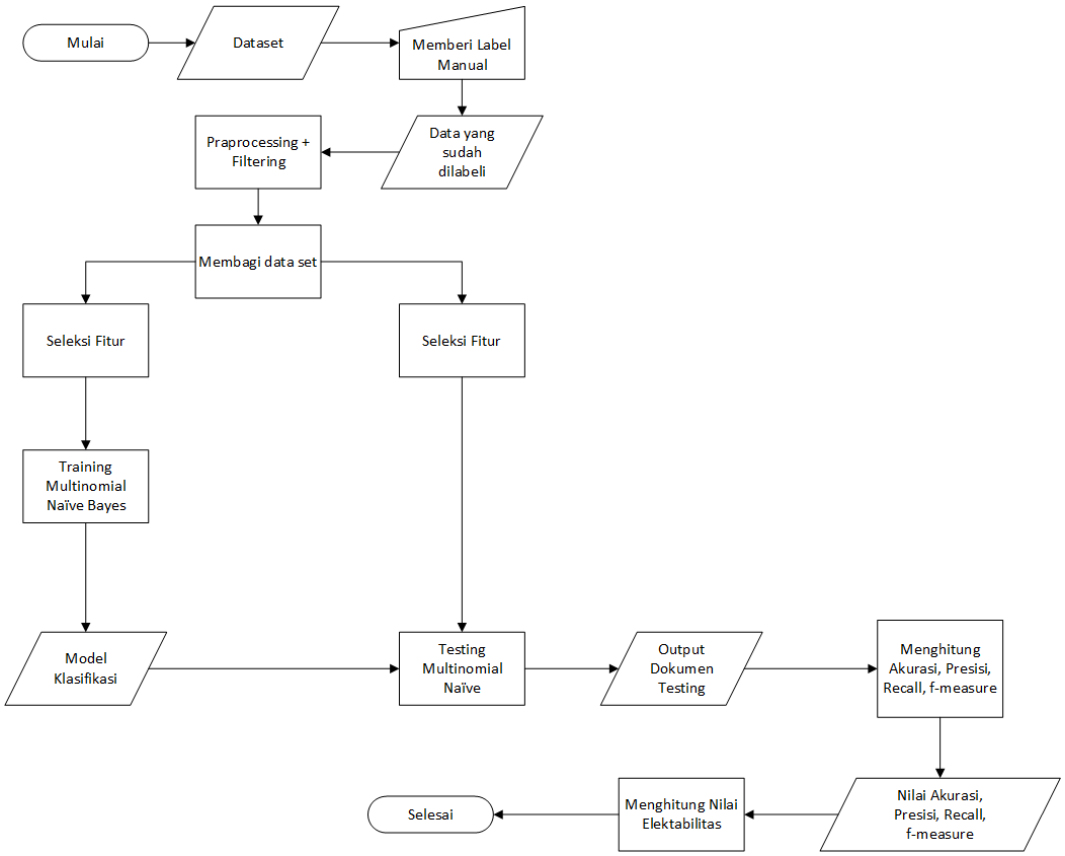
Data *stopword* yang akan digunakan pada proses *stopword removal* berasal dari penelitian oleh Tala pada tahun 2003. Daftar stopword yang digunakan berjumlah 363 kata. Penghilangan *stopword* bertujuan untuk menghilangkan kata-kata yang kurang representatif dan kurang efektif jika digunakan dalam proses klasifikasi sentiment.

4.3.3. Data Kata Dasar

Data kata dasar ini digunakan dalam proses *stemming* atau pemotongan kata dasar berimbuhan yang ada pada tahap *filtering*. Kata-kata dasar yang digunakan diambil dari modul Sastrawi yang berjumlah 29.932 kata berbahasa Indonesia.

4.4. Perancangan Sistem Klasifikasi Sentimen

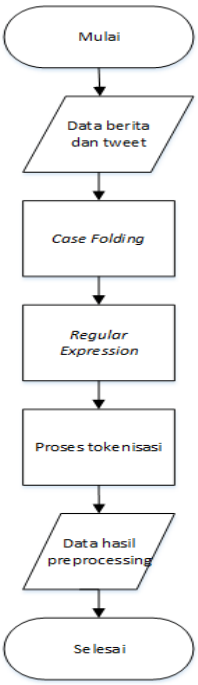
Pada penelitian ini akan dilakukan klasifikasi sentimen dari kesepuluh tokoh politik. Perancangan sistem klasifikasi sentimen ini terdiri dari beberapa langkah yaitu *preprocessing*, *filtering* seleksi fitur dan klasifikasi sentimen menggunakan algoritma klasifikasi Multinomial Naïve Bayes. Tahapan dari sistem klasifikasi sentiment dapat dilihat pada gambar 4.5



Gambar 4.5 Diagram Alur Sistem Klasifikasi Sentimen

Pada gambar 4.5 apabila dataset sudah terkumpul, dataset akan dilabeli secara manual berdasarkan konteks berita. Setelah data dilabeli akan masuk kedalam proses *preprocessing* dan *filtering*. Data tersebut kemudian dibagi menjadi 2 yaitu data *training* dan *testing*. Pada data *training* akan dilakukan seleksi fitur untuk mendapatkan fitur-fitur yang merepresentasikan data *training*. Setelah itu akan dilakukan proses klasifikasi menggunakan Multinomial Naïve Bayes untuk mendapatkan performa model data *training*. Setelah didapat model dengan performa terbaik akan dilakukan proses klasifikasi menggunakan Multinomial Naïve Bayes untuk data *testing*. Pada data *training* akan diukur performa akurasi , presisi, recall dan f-measure. Setelah itu hasil klasifikasi pada data *testing* akan dikenakan rumus PvT dan SoV untuk menghitung nilai elektabilitas setiap tokoh politik. Hasil PvT akan memuat hasil normalisasi yang merupakan elektabilitas seorang tokoh politik jika dibandingkan dengan Sembilan tokoh yang lain

Tahap pertama dalam klasifikasi sentimen setelah pelabelan data adalah *preprocessing*. *Preprocessing* dibutuhkan karena data tweet dan berita yang diambil masih memuat berbagai macam tanda baca dan besar-kecil teks belum seragam. Tahapan dari proses *preprocessing* dataset dapat dilihat pada gambar 4.6



Gambar 4.6 Diagram Alir *Preprocessing*

4.4.1. Rancangan Case Folding

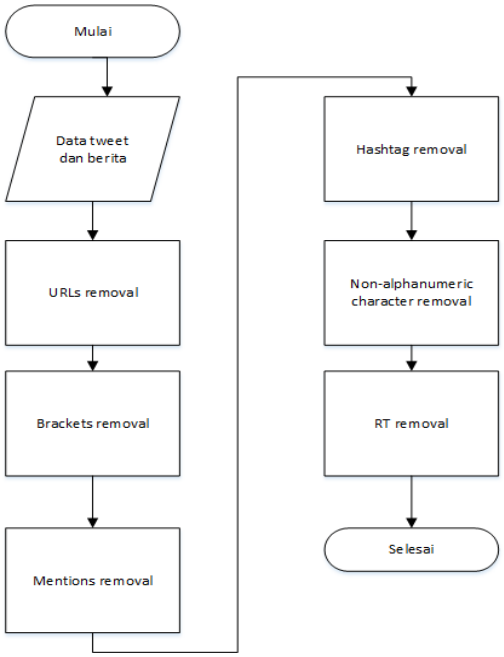
Case folding adalah proses penyamaan case dalam sebuah tweet. Tidak semua teks tweet dan berita konsistem dalam penggunaan huruf besar-kecil. Oleh karena itu peran case folding dibutuhkan dalam mengkonversi keseluruhan teks dalam tweet dan berita menjadi suatu bentuk standard. Dalam hal ini, bentuk standard nya adalah huruf kecil. Ilustrasi proses *case folding* ditunjukan pada tabel 4.3.

Tabel 4.3 : Ilustrasi Proses *Case Folding*

Data Sebelum Proses *Case Folding*	Data Sesudah Proses *Case Folding*
Peringati Sumpah Pemuda, Agus Yudhoyono motivasi mahasiswa Makassar[…] https://t.co/BWlL9bRSom#AHYsohibPemuda #il	peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar[…] https://t.co/bwll9brsom #ahysohibpemuda #il
RT @SeputarAHY: Tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. Sy melihat 2019 bukan tujuan akhir	rt @seputarahy: tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir
RT @wadjah_doeloe: Agus Yudhoyono: Jangan sampai kita jadi pecundang di negeri sendiri https://t.co/YUslZ91rsE #AHYsohibPemuda	rt @wadjah_doeloe: agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri https://t.co/yuslz91rse #ahysohibpemuda

4.4.2. Rancangan Penghapusan Karakter

Didalam teks berita dan tweet hasil *scrapping* terdapat karakter yang sebenarnya tidak terlalu dibutuhkan dalam klasifikasi sentimen dan dapat membuat akurasi klasifikasinya rendah. Karakter tersebut antara lain tanda baca, mention, URL dan lain sebagainya. Oleh karena itu regular expression digunakan untuk menghapus karakter-karakter tersebut. Proses Reguler expression yang digunakan dalam penelitian ini dapat dilihat pada gambar 4.7.



Gambar 4.7 Diagram Alir *Regular Expression*

1. URLs removal

URL digunakan untuk menunjukan alamat dari suatu sumber, seperti dokumen, file dan gambar yang terdapat di internet. URL tidak dibutuhkan dalam melakukan klasifikasi sentiment. Oleh karena itu URL-URL tersebut harus dihapus. *URLs removal* adalah suatu proses untuk menghapus URL-URL dalam dataset. Ilustrasi proses URLs removal ditunjukan pada tabel 4.4

Tabel 4.4 : Ilustrasi Proses *URLs Removal*

Data Sebelum Proses *URLs Removal*	**Data Sesudah Proses *URLs Removal***
peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar […] https://t.co/bwll9brsom #ahysohibpemuda #il	peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar [...] #ahysohibpemuda #il
rt @seputarahy: tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir	rt @seputarahy: tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir
rt @wadjah_doeloe: agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri https://t.co/yuslz91rse #ahysohibpemuda	rt @wadjah_doeloe: agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri #ahysohibpemuda

2. Brackets removal

Dalam hasil *scrapping* tweet dan berita terkadang teks yang terlalu panjang tidak di scrap semua, sehingga diganti dengan *bracket* […]. Selain itu bracket berfungsi sebagai keterangan lokasi saat tweet dibuat, keterangan bersama user lain saat tweet dibuat, serta penanda tweet memiliki gambar. Karena tidak relevan dalam proses klasifikasi sentiment, maka tanda kurung beserta isinya akan dihapus dalam proses *bracket removal*. Ilustrasi proses *bracket removal* ditunjukan pada tabel 4.5

Tabel 4.5 : Ilustrasi Proses *Bracket Removal*

Data Sebelum Proses *bracket removal*	Data Sesudah Proses *bracket removal*
peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar [...] #ahysohibpemuda #il	peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar #ahysohibpemuda #il
rt @seputarahy: tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir	rt @seputarahy: tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir
rt @wadjah_doeloe: agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri #ahysohibpemuda	rt @wadjah_doeloe: agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri #ahysohibpemuda

3. Mentions removal

Mention adalah suatu cara untuk membuat link terhadap suatu akun Twitter. Cara ini biasanya digunakan ketika kita akan membalas tweet atau ingin menandai tweet kepada seseorang. Dalam proses klasifikasi sentiment keberadaan *mention* tidak dibutuhkan. Sehingga , *mention* akan dihapus dalam proses *mentions removal*. Ilustrasi proses *mentions removal* ditunjukan pada tabel 4.6

Tabel 4.6 : Ilustrasi Proses *Mentions Removal*

Data Sebelum Proses *mentions removal*	Data Sesudah Proses *mentions removal*
peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar #ahysohibpemuda #il	peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar #ahysohibpemuda #il
rt @seputarahy: tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir	rt tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir
rt @wadjah_doeloe: agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri #ahysohibpemuda	rt agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri #ahysohibpemuda

4. Hashtag removal

Hashtag merupakan tanda "#" yang biasa disematkan untuk mendukung tweet yang dibuat atau bisa menjadi bentuk dukungan kampanye terhadap sesuatu. Dalam klasifikasi sentimen , hashtag tidak dibutuhkan sehingga perlu dihapus dalam proses *hashtag removal*. Ilustrasi proses *hashtag removal* dapat dilihat pada tabel 4.7

Tabel 4.7 : Ilustrasi Proses *Hashtag Removal*

Data Sebelum Proses *hashtag removal*	**Data Sesudah Proses *hashtag removal***
peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar #ahysohibpemuda #il	peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar
rt tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir	rt tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir
rt agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri #ahysohibpemuda	rt agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri

5. Non-alphanumeric character removal

Karakter *non-alphanumeric* adalah karakter-karakter selain huruf (*uppercase* dan *lowercase*) , angka, dan *white space* (spasi, tab atau enter). Karakter-karakter *non-alphanumeric* dapat berupa tanda baca, seperti koma, titik, tanda tanya, titik dua dan lainya. Karakter non-alphanumeric akan dihapus pada *non-alphanumeric character removal*. Ilustrasi proses *non-alphanumeric character removal* ditunjukan pada tabel 4.8

Tabel 4.8 : Ilustrasi Proses *Non-Alphanumeric Character Removal*

Data Sebelum Proses *non-alphanumeric character removal*	Data Sesudah Proses *non-alphanumeric character removal*
peringati sumpah pemuda, agus yudhoyono motivasi mahasiswa makassar	peringati sumpah pemuda agus yudhoyono motivasi mahasiswa makassar
rt tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak. sy melihat 2019 bukan tujuan akhir	rt tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak sy melihat 2019 bukan tujuan akhir ahy nn
rt agus yudhoyono: jangan sampai kita jadi pecundang di negeri sendiri	rt agus yudhoyono jangan sampai kita jadi pecundang di negeri sendiri

6. RT Removal

Retweet atau RT berfungsi untuk mengulang kembali tweet yang telah ada agar dapat dibagikan kembali kepada pengikut-pengikut yang ada di Twitter. Karena didalam klasifikasi sentiment RT tidak diperlukan maka RT akan dihapus dalam *RT removal*. Ilustrasi proses *RT removal* dapat dilihat pada tabel 4.9

Tabel 4.9 : Ilustrasi Proses RT *Removal*

Data Sebelum Proses *RT Removal*	Data Sesudah Proses *RT Removal*
peringati sumpah pemuda agus yudhoyono motivasi mahasiswa makassar	peringati sumpah pemuda agus yudhoyono motivasi mahasiswa makassar
rt tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak sy melihat 2019 bukan tujuan akhir ahy	tergantung masyarakat mau saya kembali ke kompetisi politik atau tidak sy melihat 2019 bukan tujuan akhir ahy
rt agus yudhoyono jangan sampai kita jadi pecundang di negeri sendiri	agus yudhoyono jangan sampai kita jadi pecundang di negeri sendiri

4.4.3. Rancangan Tokenisasi

Tokenisasi adalah tahap yang mengubah teks yang semula terdiri dari kalimat-kalimat menjadi kata-kata yang menyusun kalimat tersebut. Teks yang sudah masuk tokenisasi sudah merupakan teks yang bersih.

4.4.4. Filtering data

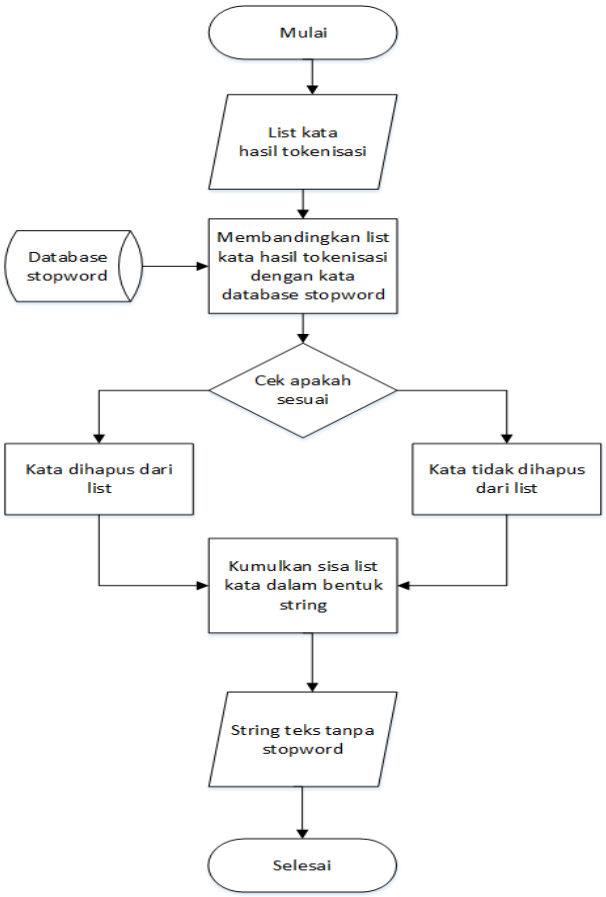
Setelah dilakukan *preprocessing* data, selanjutnya *list* hasil tokenisasi akan melalui proses *filtering* data. Tahap *filtering* data terdiri dari proses *stopword removal* dan *stemming* .

4.4.5. Perancangan stopword removal

Proses *stopword removal* adalah proses pembuangan kata-kata yang kurang bermakna atau kurang representatif dalam mewakili dokumen pada proses klasifikasi sentimen, termasuk angka-angka yang ada pada dokumen teks. Dalam proses penghapusan *stopword* , disertakan daftar kata stopword yang nanti akan jadi acuan dalam menghilangkan kata-kata yang sesuai dalam dokumen teks.

Pada gambar 4.8, dapat dilihat proses penghapusan *stopword* yang berisi langkah-langkah sebagai berikut:

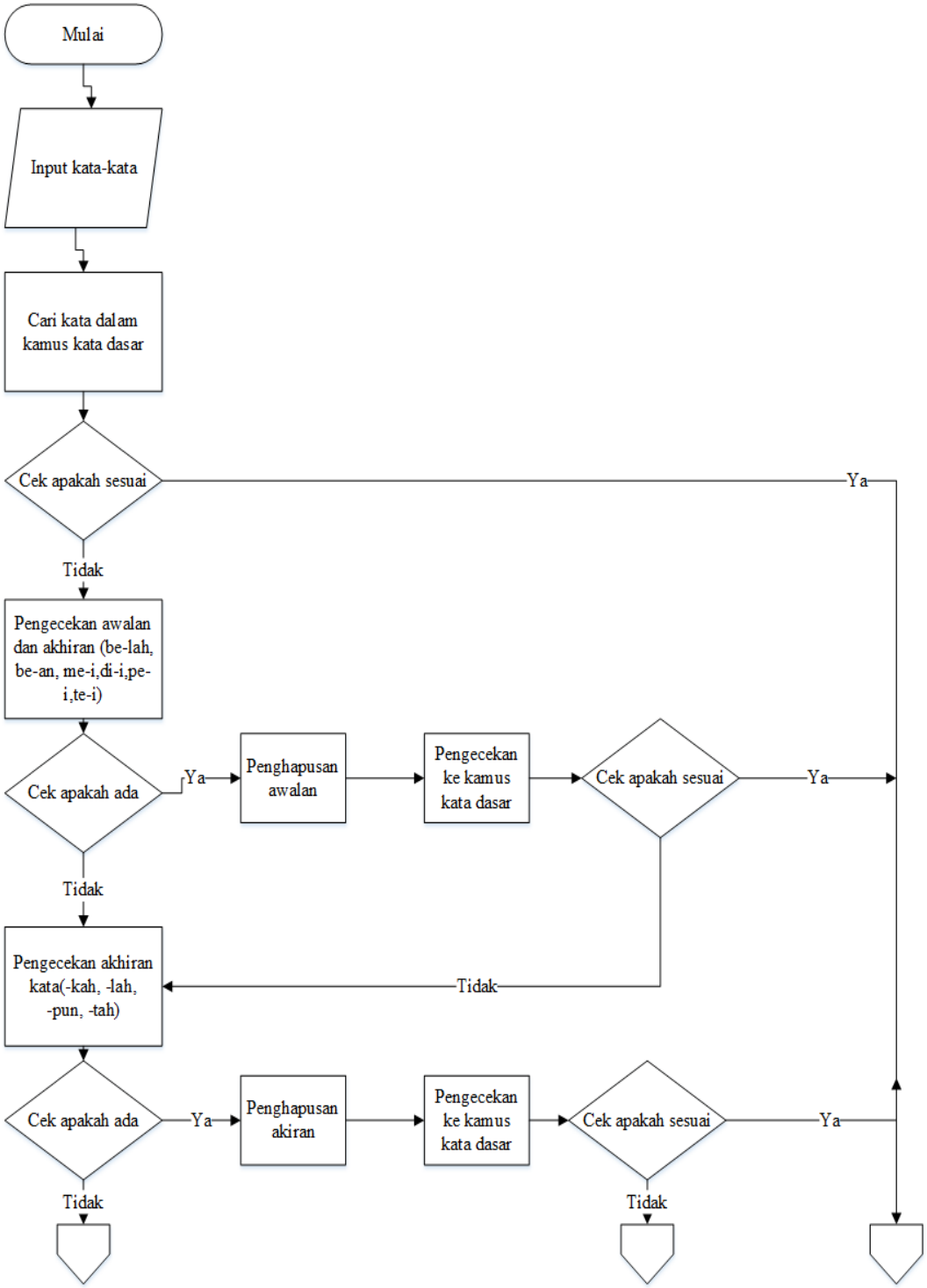
1. Kata kata hasil tokenisasi yang tersimpan dalam bentuk *list* dimasukan sebagai *input*
2. List kata-kata dalam dokumen dicocokan dengan daftar *stopword* yang juga disimpan dalam bentuk *list*
3. Jika kata dalam dokumen teks yang dibandingkan ada dalam daftar *stopword* maka kata dalam dokumen akan dihapus, jika tidak ada maka kata tersebut tidak dihapus.
4. Sisa kata yang ada akan disatukan lagi dalam bentuk *string*



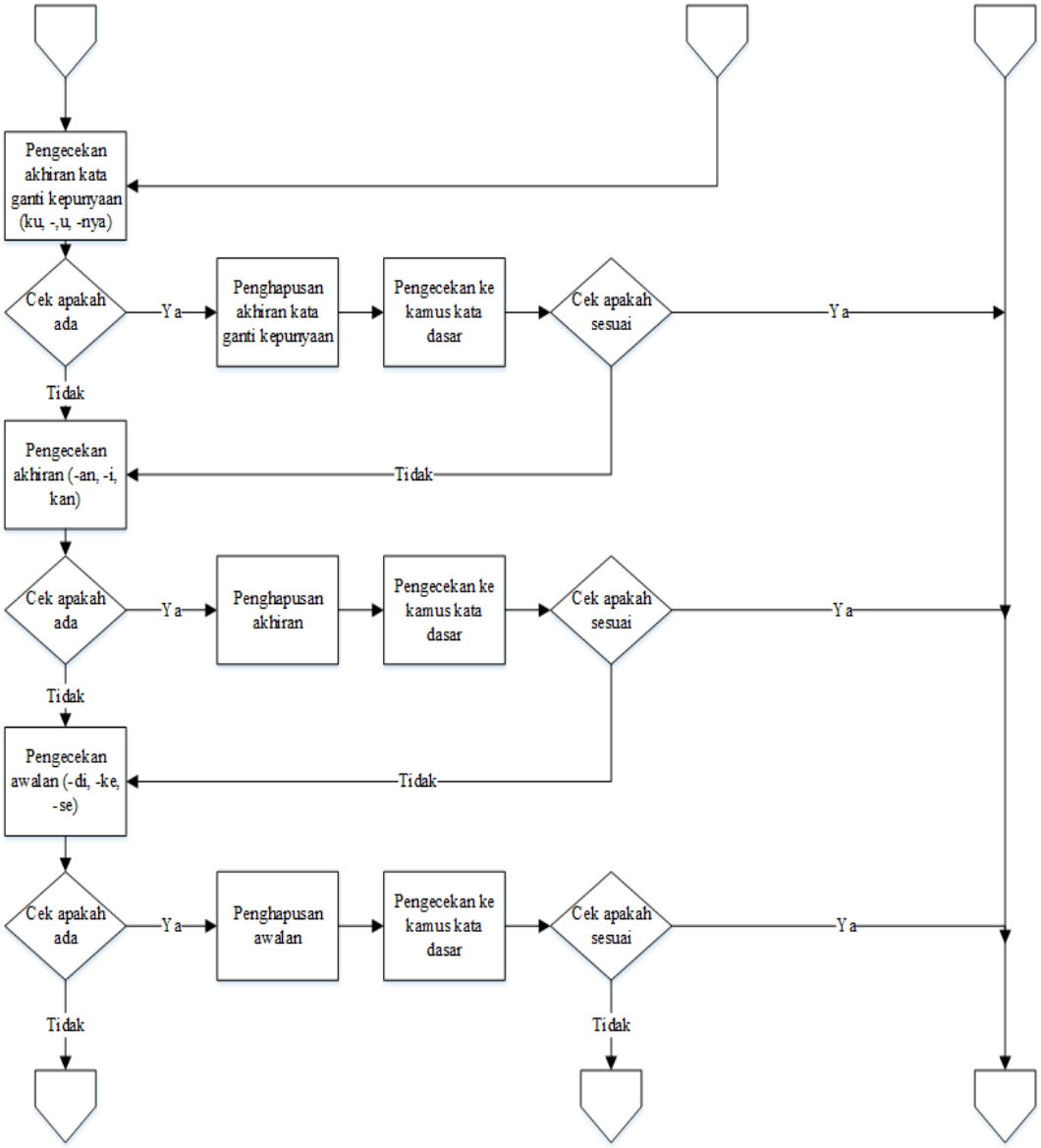
Gambar 4.8 Diagram Alir *Stopword Removal*

4.4.6. Perancangan stemming

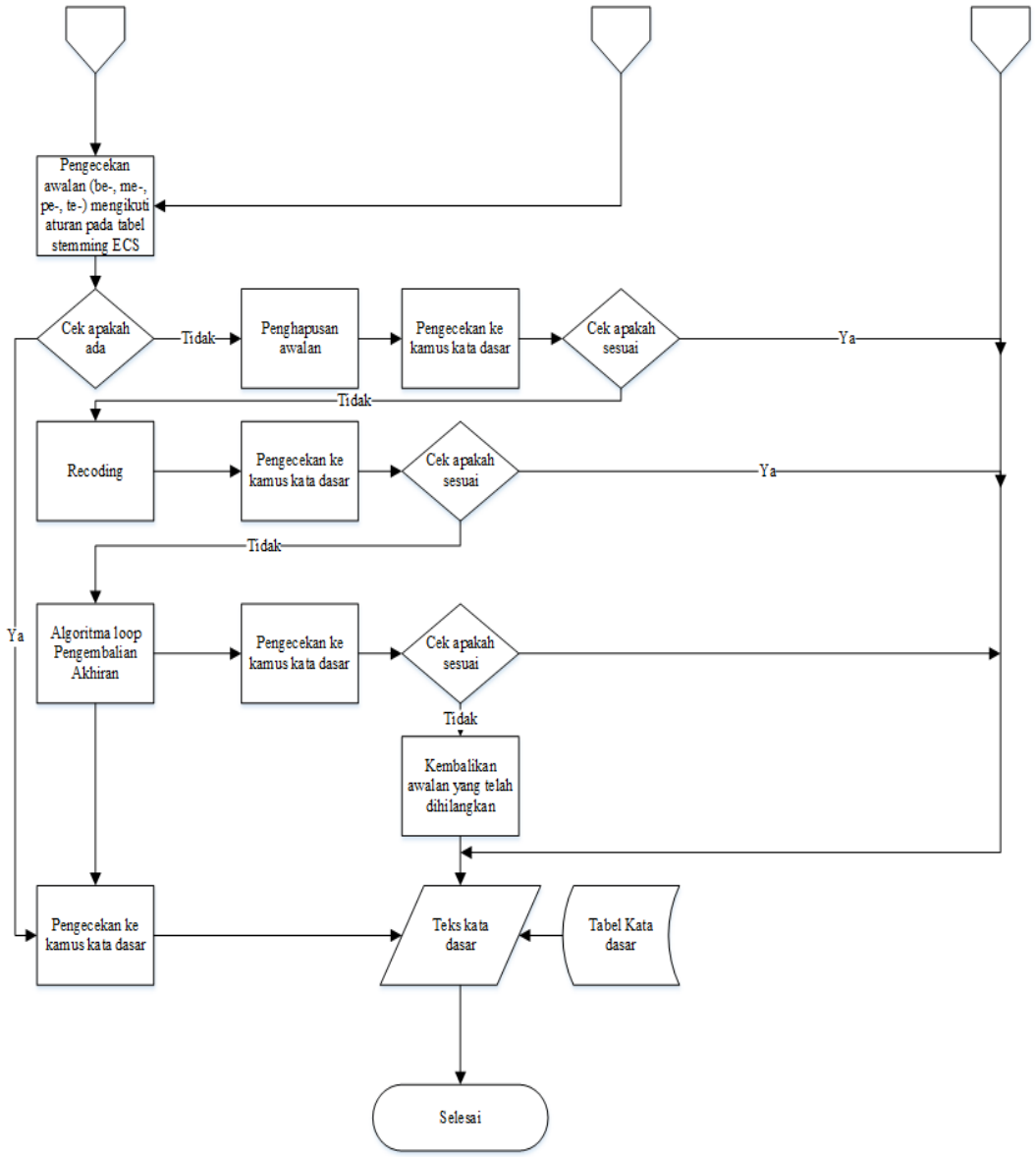
Tahap *stemming* bertujuan untuk menghasilkan kata dasar dari berbagai macam kata berimbuhan pada dokumen teks. Dalam penelitian ini digunakan algoritma perbaikan dari *Enhanced Confix-Stripping Stemmer* hasil penelitian Tahitoe dan Purwitasari (2010) yang kemudian dikemas dalam *library* bernama Sastrawi. Diagram alir proses stemming menggunakan *library* Sastrawi dapat dilihat pada gambar 4.9, 4.10, 4.11



Gambar 4.9 Diagram Alir *Stemming* (1)



Gambar 4.10 Diagram Alir *Stemming* (2)



Gambar 4.11 Diagram Alir *Stemming* (3)

Pada proses *stemming* , masukan yang diberikan pada sistem berupa *string* teks dokumen, dan output yang dihasilkan dari proses adalah berupa string berisi kata-kata yang sudah mengalami pemotongan imbuhan.

4.4.7. Perancangan Seleksi fitur

Proses seleksi fitur dalam penelitian ini menggunakan 2 metode yaitu *chi square* dan TF-IDF. Pada penelitian ini akan dicari performa seleksi fitur yang lebih baik antara *chi square* dan TF-IDF.Proses seleksi fitur diawali dengan melakukan pembobotan masing masing metode seleksi fitur *chi square* dan TF-IDF pada seluruh kata yang ada pada dokumen teks. Setelah itu dilakukan pengurutan nilai bobot hasil *chi square* dan TF-IDF terbesar. Akan dilakukan beberapa kali pengujian dengan memvariasikan jumlah kata fitur yang akan digunakan supaya dapat membentuk model klasifikasi terbaik dengan nilai performansi optimal untuk masing masing tokoh politik. Jumlah variasi kata fitur antar tokoh politik akan berbeda beda karena setiap tokoh memiliki dataset yang jumlahnya berbeda juga.

4.4.8. Seleksi fitur TF-IDF

Tahap ini dilakukan setelah dokumen teks mengalami proses penghilangan *stopword* dan *stemming*. Dalam menentukan nilai TF-IDF, diperlukan perhitungan term frequency, inverse document frequency dan nilai TF-IDF sudah dibahas pada sub bab 3.6. Langkah-langkah yang dilakukan adalah :

1. Menghitung TF yaitu frekuensi kemunculan kata(ti) pada dokumen (dj). Hasil frekuensi kemunculan kata dibagi dengan frekuensi maksimum dari seluruh kata didalam dokumen. Rumus untuk menghitung TF dapat dilihat pada persamaan 3.1
2. Menghitung *document frequency*, yaitu banyaknya dokumen dimasa suatu kata (ti) muncul
3. Menghitung nilai inverse document frequency menggunakan persamaan 3.2
4. Menghitung nilai TF-IDF dari masing masing kata pada dokumen dengan persamaan 3.3

4.4.9. Seleksi fitur chi square

Tahap ini juga dilakukan setelah dokumen teks mengalami proses penghilangan *stopword* dan *stemming*. Dalam menentukan nilai *chi square* diperlukan perhitungan nilai *Observed dan Expected Value* yang sudah dibahas pada sub bab 3.7. Langkah langkah yang dilakukan adalah sebagai berikut :

1. Menghitung nilai *Observed Value* yaitu jumlah kemunculan kata suatu kelas.
2. Mencari nilai Expected value menggunakan perkalian N x P(t) x P(c)
3. Setelah itu nilai (*Observed Value – Expected Value)*^{2} dan dibagi dengan nilai *Expected Value*

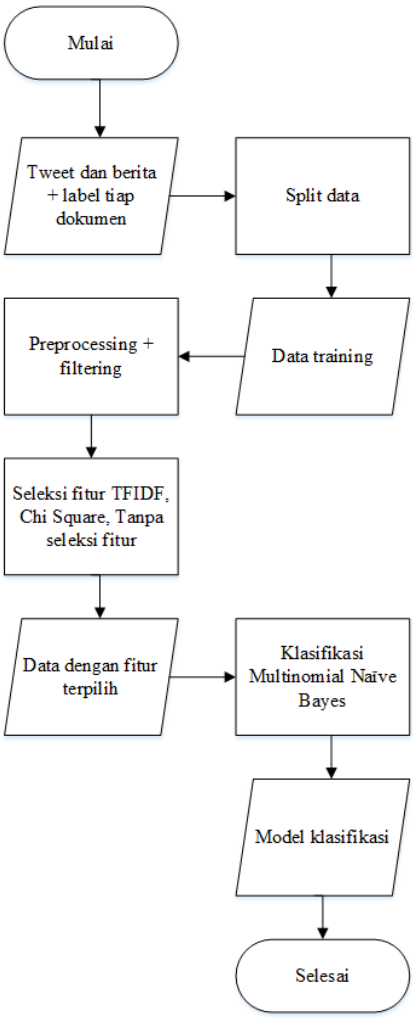
4.4.10. Pelabelan Data

Pelabelan data adalah pemberian kategori pada masing-masing teks dokumen berita dan tweet. Label pada teks berita dibutuhkan pada saat proses training data , yaitu pada saat pembentukan model klasifikasi. Selain itu, label pada data juga dibutuhkan untuk mengukur performa dari klasifikasi, dengan cara membandingkan hasil label yang diberikan pada data secara manual dengan label hasil prediksi klasifikasi. Polaritas yang digunakan dalam pelabelan data ini adalah positif dan negatif. Pelabelan data dilakukan terhadap 10 tokoh politik yang sudah dijelaskan pada tabel 4.1

Pemberian label dilakukan secara manual. Data-data yang telah terkumpul dalam file berekstensi .csv dengan *header* bernama berita dibaca satu persatu. Selanjutnya diberikan kolom dengan *header* baru bernama sentimen. Dalam kolom tersebut diisikan label berupa polaritas dari teks yaitu positif atau negatif. Dalam suatu teks berita atau tweet, paling tidak berisi salah satu atau beberapa info penting yang dapat menjadi pertimbangan penentuan polaritas yaitu sentimen dari kata-kata penyusun teks. Pelabelan juga dibantu oleh ahli dari jurusan Sastra Indonesia untuk mevalidasi. Setelah pelabelan dilakukan akan dilakukan validasi ulang untuk tiap tiap polaritas yang diberikan setiap teks setiap tokoh politik.

4.4.11. Perancangan Training

Setelah proses pelabelan data, preprocessing dan seleksi fitur menggunakan TF-IDF dan *chi square* . maka data training sudah siap untuk dijadikan masukan pada proses pembuatan model klasifikasi. Langkah-langkah proses training klasifikasi tweet dan berita menggunakan metode Multinomial Naïve Bayes dapat dilihat pada gambar 4.12



Gambar 4.12 Diagram Alir Perancangan *Training*

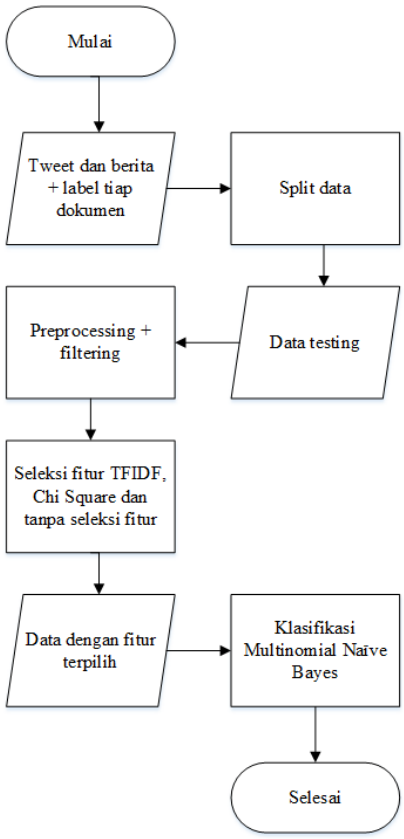
Pada gambar 4.12 dijelaskan proses training dari klasifikasi Multinomial Naïve bayes. Berikut penjelasan dari gambar :

1. Dataset berita dan tweet diberi label manual untuk mendapatkan label dari masing-masing teks dokumen berita
2. Dataset kemudian dibagi untuk memisahkan antara kumpulan data *training* dan data *testing*. Data training yang akan digunakan selanjutnya melalui proses preprocessing dan filtering.
3. Setelah melalui *preprocessing* dan *filtering* , dihitung bobot dari semua kata yang ada pada dokumen. Dicari top-n kata dengan bobot TF-IDF dan chi square tertinggi untuk dijadikan kumpulan fitur yang akan dipakai dalam klasifikasi.

4. Dibuat model klasifikasi menggunakan algoritma Multinomial Naïve Bayes yang merujuk pada sub bab 3.8 dari data *training*

4.4.12. Perancangan Testing

Model yang telah didapat dari proses *training* digunakan dalam proses testing pada dokumen testing. Langkah-langkahnya dapat dilihat pada gambar 4.13



Gambar 4.13 Diagram Alir Perancangan *Testing*

Berikut penjelasan dari proses *testing* pada gambar 4.13 :

1. Dataset dibagi untuk memisahkan antara kumpulan data *training* dan data *testing*. Data testing yang akan digunakan selanjutnya melalui proses preprocessing dan filtering.
2. Dengan menggunakan model yang telah dibuat pada proses *training*, dilakukan proses testing sesuai sub bab 3.8 pada dokumen testing yang memuat kata-kata hasil seleksi fitur menggunakan algoritma Multinomial Naïve Bayes
3. Didapatkan hasil klasifikasi yang merupakan output dari proses testing yang selanjutnya akan dipakai untuk perhitungan elektabilitas.

Contoh data testing yang akan melalui proses klasifikasi Multinomial Naïve Bayes ditunjukan pada tabel 4.10

Tabel 4.10 : Contoh Data Testing Yang Akan Melalui Proses Klasifikasi

Data Latih	No	Berita	Sentimen
	1	lemah agus yudhoyono	negatif
	2	agus yudhoyono sambang wapres jk makassar	positif
	3	agus yudhoyono temu kerja sesuai bakat bakat kerja	positif
Data testing	4	agus yudhoyono temu wapres jk	?

Probabilitas *prior* dihitung dengan membagi jumlah suatu kelas pada data latih dengan jumlah seluruh data latih. Rumus perhitungan probabilitas prior ditunjukan pada persamaan (3.9). Hasilnya akan ditunjukan pada tabel 4.11

Tabel 4.11 : Contoh Hasil Perhitungan Probabilitas *Prior*

P(pos)	2/3
P(neg)	1/3

Untuk setiap kelas yang ada, probabilitas kondisional untuk masing-masing token(kata yang unik) pada data test dihitung dengan menggunakan persamaan (3.10)

Dengan menggunakan persamaan (3.10), probabilitas kondisional untuk masing masing token pada data test dihitung dan hasilnya ditunjukan pada pada tabel 4.12. Nilai yang digunakan dalam tabel 4.12 merupakan jumlah frekuensi kemunculan kata didalam data latih pada tabel 4.10. Pada tabel 4.12, nilai 14 merupakan jumlah seluruh kata dalam kelas positif, nilai 3 merupakan jumlah seluruh kata dalam kelas negatif , nilai 11 merupakan jumlah kata yang unik dalam data latih.

Tabel 4.12 : Contoh Hasil Perhitungan Probabilitas Kondisional Untuk Masing-Masing Token Pada Tiap Kelas

Kata	P(kata | positif)	P(kata | negatif)
temu	$\frac{1+1}{14+11} = \frac{2}{25}$	$\frac{0+1}{3+11} = \frac{1}{14}$
agus	$\frac{2+1}{14+11} = \frac{3}{25}$	$\frac{1+1}{3+11} = \frac{2}{14}$
yudhoyono	$\frac{2+1}{14+11} = \frac{3}{25}$	$\frac{1+1}{3+11} = \frac{2}{14}$
wapres	$\frac{1+1}{14+11} = \frac{2}{25}$	$\frac{0+1}{3+11} = \frac{1}{14}$
jk	$\frac{1+1}{14+11} = \frac{2}{25}$	$\frac{0+1}{3+11} = \frac{1}{14}$

Hasil perhitungan probabilitas kondisional setiap token kemudian dilakukan untuk masing-masing kelas nya dan dikalikan dengan probabilitas *prior* kelas tersebut. Hasilnya berupa probabilitas suatu kelas pada suatu data yang ditunjukan dibawah ini :

P(positif|n4) = 3/25 * 3/25 * 2/25 * 2/25 * 2/25 * 2/3 = 4.9152e-6

P(negatif|n4) = 2/14 * 2/14 * 1/14 * 1/14 * 1/14 * 1/3 = 2.4791e-6

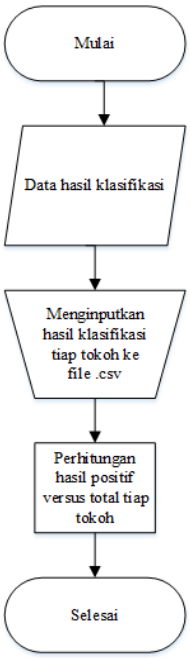
Jika probabilitas suatu kelas pada suatu dokumen lebih besar dibandingkan probabilitas kelas lainya pada dokumen tersebut, maka dokumen tersebut diklkasifikasikan sesuai dengan probabilitas terbesar tersebut. Dikarenakan nilai dari P(positif|n4) lebih besar dibanding P(negatif|n4) maka dokumen tersebut diklasifikasikan sebagai positif

4.4.13. Pengujian model klasifikasi sentimen

Dalam penelitian ini digunakan metode pengujian *k-fold cross validation* dengan nilai k=10. Proses evaluasi dengan k-fold cross validation dilakukan dengan membagi dataset menjadi 10 bagian. Diambil 1/10 data sebagai data *validation* dan sisanya menjadi data training. Untuk lebih jelasnya dapat dilihat pada gambar 3.3. Evaluasi performa dari klasifikasi sentimen ini didapatkan dengan membandingkan hasil klasifikasi dengan data yang diberi label secara manualdengan menghitung presisi, *recall*, akurasi dan *f-measure* nya.

4.4.14. Perhitungan Elektabilitas

Setelah proses *testing* dilakukan dan didapatkan hasil klasifikasi sentimen tiap tokoh, selanjutnya hasil klasifikasi dari masing-masing tokoh tersebut disimpan dalam file berekstensi .csv. Setelah itu dilakukan perhitungan nilai elektabilitas tiap tokoh dengan menggunakan rumus *positive versus total* dan *share of volume* pada sub bab 3.10 dan 3.11. Diagram alir perhitungan elektabilitas tokoh politik dapat dilihat pada gambar 4.14



Gambar 4.14 Diagram Alir Perhitungan Elektabilitas Tiap Tokoh

4.5. Rancangan Skenario Pengujian

Terdapat 2 tahap skenario pengujian dalam penelitian ini yaitu perbandingan penggunaan jumlah fitur *top-n* dalam proses seleksi fitur TF-IDF dan *chi square* untuk masing masing tokoh politik yang bertujuan untuk mengatahui performa fitur *top-n* mana yang terbaik. Serta, perbandingan hasil nilai elektabilitas menggunakan *positive versus total* dan *share of volume* untuk masing masing tokoh politik.

4.5.1.Rancangan Perbandingan pengujian performa fitur top-n

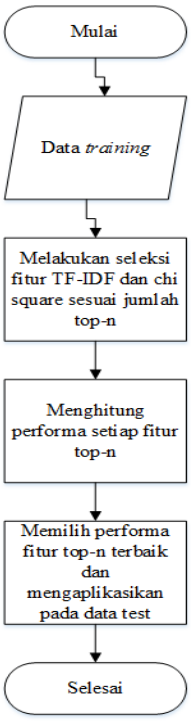
Untuk mengetahui performa fitur top-n yang mana yang terbaik dilakukan penggunaan perbandingan fitur top-n setiap tokoh politik.Jumlah perbandingan fitur *top-n* setiap tokoh akan berbeda-beda karena data *training* setiap tokoh politik

juga berbeda-beda. Detail perbandingan jumlah fitur *top-n* setiap tokoh politik dapat dilihat pada tabel 4.13

Tabel 4.13 Perbandingan Jumlah Fitur Top-N Tiap Tokoh Politik

	Perbandingan jumlah fitur top-n		
Prabowo Subianto	200	400	600
Ahok	400	800	1200
Joko Widodo	1000	2000	3000
Anies Baswedan	500	1000	1500
Gatot Nurmantyo	200	400	600
Jusuf Kalla	350	700	1050
Hary Tanoe	200	400	800
Ridwan Kamil	500	1000	1500
Zulkifli Hasan	150	300	450
Agus Yudhoyono	200	400	600

Diagram alur skenario pengujian perbandingan jumlah fitur dapat dilihat pada gambar 4.15

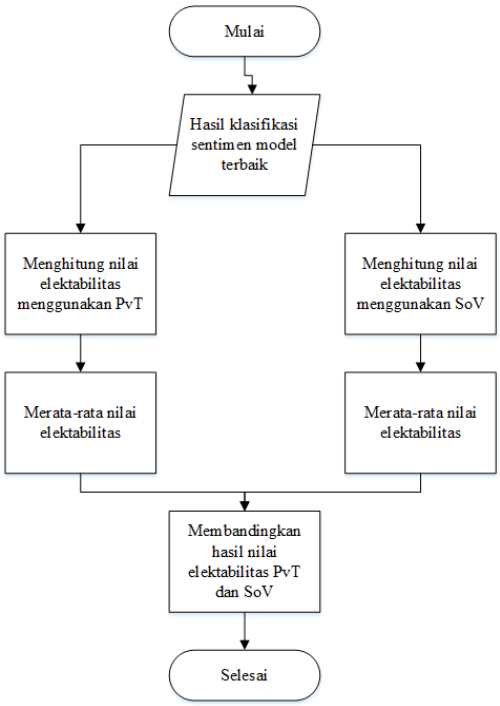


Gambar 4.15 Skenario Pengujian Perbandingan Fitur Top-N

Pada gambar 4.15, proses seleksi fitur TF-IDF dan *chi square* dilakukan terhadap data *training* untuk mendapatkan kata fitur yang merepresentasikan data *training* tersebut. Jumlah perbandingan kata fitur yang digunakan untuk masing-masing tokoh politik sesuai dengan tabel 4.13. Perbedaan jumlah perbandingan kata fitur masing-masing tokoh politik dikarenakan perbedaan jumlah data *training* untuk masing-masing tokoh politik saat pengambilan data. Setelah proses seleksi fitur, dilakukan pengujian performa untuk setiap perbandingan kata fitur dan akan dipilih fitur *top-n* yang memiliki performa yang paling baik.

4.5.2.Rancangan Perbandingan pengujian rumus elektabilitas

Setelah model klasifikasi terbaik didapatkan, maka model klasifikasi terbaik tersebut diterapkan pada data *testing*. Hasil klasifikasi sentimen pada data *testing* kemudian akan digunakan untuk mencari nilai elektabilitas masing masing tokoh politik. Skenario pengujian dilakukan untuk perbandingan nilai elektabilitas tiap tokoh politik menggunakan rumus *positive versus total* dan *share of volume*.Diagram alur skenario pengujian mencari nilai elektabilitas menggunakan positive versus total dan share of volume dapat dilihay pada gambar 4.16



Gambar 4.16 Skenario Pengujian Perbandingan Rumus Elektabilitas

BAB V

IMPLEMENTASI

5.1 Lingkungan Implementasi

Implementasi dari sistem ini menggunakan bahasa pemrograman python. Adapun peralatan dan bahan yang digunakan dalam implementasi ini adalah sebagai berikut :

1. Prosesor : Intel core i3-3217U CPU @1.80GHz
2. RAM : 4.00 GB, 64-bit Windows 10 pro OS
3. Harddisk : 500GB

Spesifikasi perangkat lunak yang digunakan untuk proses implementasi analisis sentimen menggunakan metode Multinomial Naïve Bayes adalah sebagai berikut :

1. Sistem Operasi : 64-bit Windows 10 pro
2. Bahasa pemrograman : Python 3.6
3. IDE : Spyder

5.2 Implementasi Sistem

Pada bagian ini akan dijelaskan implementasi dari sistem yang sebelumnya sudah dirancang. Implementasi sistem terdiri dari *preprocessing*, *filtering*, seleksi fitur TF-IDF dan *chi square*, implementasi pemodelan klasifikasi menggunakan algoritma klasifikasi *Multinomial Naïve Bayes,* implementasi pengujian dengan data testing, implementasi pengujian model dengan *10-fold cross validation* dan perhitungan elektabilitas tiap tokoh. Implementasi *preprocessing* terdiri dari proses *case folding*, *regular expression*, tokenisasi. Sementara *filtering* terdiri dari *stopword removal* dan *stemming*.

5.3 Implementasi Pelabelan Data

Seperti yang sudah dijelaskan sebelumnya pada sub bab 4.3.1 bahwa dataset yang akan digunakan sebagai data *training* dan *testing* berjumlah 23.073 data yang dikumpulkan dalam rentang waktu 17 November 2016 – 1 November 2017. Pelabelan dilakukan untuk menentukan polaritas dari masing-masing berita dan

tweet yang sudah dikumpulkan. Label berita akan dipakai untuk menentukan polaritas untuk proses *training* dan untuk pengukuran performansi dengan cara mencocokan data hasil pemberian label manual dengan hasil dari klasifikasi yang sudah dilakukan. Pelabelan data dilakukan dengan membaca isi dari tiap berita dan tweet dan mengkategorikan pada sentiment positif atau negatif. Gambar 5.1 menunjukan cuplikan data yang telah diberikan label.

agus yudhoyono suka seringkali hindar	neg
agus yudhoyono hidup selalu indah indah hidup kenang	pos
agus harimurti yudhoyono aku hebat wali kota makassar	pos
agus yudhoyono sambang wapres jk makassar	pos
hari temu sby presiden jokowi gilir agus yudhoyono temu wapr	pos

Gambar 5.1 Cuplikan Data Yang Telah Diberikan Label

5.4 Implementasi Preprocessing

Tahap *preprocessing* adalah tahap yang dilakukan sebelum dilakukan proses klasifikasi. Dalam *preprocessing* terdapat 3 proses yang dilakukan, yaitu proses *case folding* , *regular expression* dan tokenisasi. Hasil dari *preprocessing* ini akan digunakan sebagai masukan dari proses *filtering*

5.4.1 Implementasi Case Folding

Proses *case folding* adalah penyeragaman *case* dari teks dalam dataset menjadi *lowercase*. Proses *case folding* ditunjukan pada gambar 5.2

```
1 import pandas as pd
2 df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\datatrain.csv',
3 encoding = "latin1" )
4
5 df['Berita'] = df.Berita.apply(lambda x : x.lower())
```

Gambar 5.2 Kode Proses Case Folding

Pada gambar 5.2, baris ke-2 dilakukan proses pembacaan dokumen .csv yang memuat berita dan tweet tiap tokoh politik yang sudah dikumpulkan. Lalu pada baris ke-5 dilakukan pengubahan format teks pada kolom Berita yang bertujuan untuk mengubah *case* pada semua teks menjadi *lowercase*

5.4.2 Implementasi Regular Expression

Setelah melakukan *case folding* , tahap selanjutnya adalah menggunakan *regular expression* untuk melakukan berbagai proses yang sudah dijelaskan pada sub bab 4.4.2. Proses penggunaan *regular expression* ditunjukan pada gambar 5.3

```python
import csv
import re

with open('datatrain.csv', encoding='windows-1252') as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')

    tweets = []

    for row in readCSV:
        tweet = row[0]
        tweets.append(tweet)

preprocessed_tweets = []

for tweet in tweets:
    tweet = re.sub(r"http\S+", "", tweet) # URLs Removal
    tweet = re.sub(r"[\(\[].*?[\)\]]", "", tweet) # Brackets Removal
    tweet = re.sub(r"@\S+", "", tweet) # Mentions Removal
    tweet = re.sub(r"#\S+", "", tweet) # Hashtag Removal
    tweet = re.sub(r"[^A-Za-z0-9]+", " ", tweet) # Alphanumeric Only
    tweet = re.sub(r"rt ", "", tweet) # RT Removal
    preprocessed_tweets.append(tweet)
```

Gambar 5.3 Kode Program Regular Expression

Pada gambar 5.3, baris ke-1 digunakan untuk memanggil library untuk membaca file berektens .csv .Baris ke-2 digunakan untuk memanggil library re untuk melakukan proses *regular expression*. Baris ke-4 dan 5 dilakukan proses pembacaan file berekstensi .csv. Pada baris ke-7 sampai 11 dilakukan pengambilan konten pada kolom ke-0 yang berisi berita atau tweet untuk selanjutnya disimpan dalam *list* tweets. Selanjutnya ,setiap konten berita atau tweet dari *list* tweets dilakukan proses perulangan *regular expression* pada baris ke-15 sampai ke 22. Penggunaan regular expression yang pertama adalah *URLs removal*. Pada baris ke-16 ,setiap konten *list* tweets dibaca, apakah memuat URL atau tidak. URL biasanya ditandai dengan http pada bagian awal URL tersebut. Pada baris ke-17 ,*list* tweets setelah proses *URLs removal* dibaca, apakah memuat tanda kurung (baik biasa maupun kurung siku) atau tidak. Jika ada maka akan dihapus beserta isinya. Pada baris ke-18 , *list* tweets setelah proses *brackets removal* dibaca, apakah memuat mention atau tidak. Mention biasanya ditandai dengan "@" dibagian awal mention. Jika ada maka akan dihapus beserta kata yang berpasangan denganya. Pada baris ke-19, *list* tweets setelah proses *mention removal* dibaca, apakah memuat *hashtag* atau tidak. *Hashtag* biasanya ditandai dengan "#". Jika ada "#" maka akan dihapus beserta kata yang berpasangan dengan nya. Pada baris ke-20 , *list* tweets setelah proses *hashtag removal* akan dibaca, apakah memuat kakarter *non-alpha numeric* atau tidak. Jika ada maka akan dihapus. Pada baris ke-21 , *list* tweets setelah proses

non-alphanumeric removal akan dibaca, apakah memuat retweet atau tidak. Retweet biasa ditandai dengan "rt". Jika ada, maka akan dihapus. Setelah proses *RT removal*, *list* tweets disimpan pada *list* baru preprocessed_tweets yang berisi tweet atau berita bersih yang sudah mendapat *regular expression*.

5.4.3 Implementasi Tokenisasi

Proses tokenisasi dilakukan untuk mengubah teks yang semula terdiri dari kalimat-kalimat menjadi kata-kata yang menyusun kalimat tersebut. Proses tokenisasi dapat dilihat pada gambar 5.4

```
1 import pandas as pd
2 import nltk
3
4 df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\datatrain.csv',
5 encoding = "latin1" )
6
7 df['Berita'] = df.Berita.apply(lambda x : nltk.word_tokenize(x))
```

Gambar 5.4 Kode Program Tokenisasi

Ditunjukan pada gambar 5.4, baris ke-4 dan 5 dilakukan proses pembacaan file dokumen .csv yang berisi kumpulan tweet dan berita. Kemudian proses tokenisasi dilakukan dengan cara mengimport *library* nltk yang dilakukan pada baris ke-2. Kemudian fungsi tokenisasi dari *library* nltk dipanggil pada baris ke-7 untuk melakukan proses tokenisasi pada setiap teks berita atau tweet.

5.5 Implementasi Filtering

Setelah dilakukan proses *preprocessing* pada dataset mentah yang sudah dikumpulkan, selanjutnya dataset akan melalui tahap *filtering*. Tahap filtering yang dilakukan terdiri dari 2 proses, yaitu *stopword removal* dan *stemming*.

5.5.1 Stopword Removal

Pada proses tokenisasi, teks pada dataset sudah dipecah menjadi kata-kata yang semula menyusun kalimat teks tersebut. Selanjutnya dilakukan pembuangan kata-kata *stopword* untuk mengurangi jumlah kata-kata yang tidak cukup representative untuk dijadikan fitur pada proses klasifikasi sentimen, seperti kata-kata penghubung, kata ganti kepunyaan, dan lain sebagainya. Sebelumnya daftar *stopword* didapatkan dari penelitian yang dilakukan oleh Tala (2003) yang telah dikumpulkan dalam suatu dokumen berformat.txt. Cuplikan kode program proses *stopword removal* ditunjukan pada gambar 5.5.

```
1  import pandas as pd
2
3  df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\datatrain.csv',
4  encoding = "latin1" )
5
6  kata = [line.strip() for line in open('stopwords.txt','r')]
7
8  def hilangkanstopword(teks):
9      remove = []
10     for word in teks:
11         for minus in kata:
12             if word in minus:
13                 remove.append(word)
14     sentence = [x for x in teks if x not in remove]
15     sentencecomplete = ' '.join(sentence)
16     return sentencecomplete
17 df['Berita'] = df.Berita.apply(lambda x : hilangkanstopword(x))
```

Gambar 5.5 Kode Program Stopword Removal

Pada gambar 5.5, baris ke-3 dan 4 adalah proses pembacaan file dokumen .csv. Pada baris ke 6, dilakukan pembacaan file *stopwords.txt* yang kemudian daftarnya disimpan dalam variable kata. Setelah itu pada fungsi *hilangkanstopword* yang ditunjukan pada baris ke-9 sampai 14 dilakukan pengecekan, apakah kata-kata pada teks tersebut termasuk kata yang ada dalam daftar *stopword* atau tidak.Jika termasuk, maka kata tersebut akan dihilangkan. Setelah itu kata-kata dalam bentuk token yang tidak dihapus akan dijadikan *string* kembali per berita atau tweet oleh kode baris ke 15.

5.5.2 Stemming

Implementasi *stemming* pada penelitian ini dilakukan dengan menggunakan *library stemmer* teks berbahasa Indonesia bernama Sastrawi. Sastrawi merupakan *library stemmer open source* yang dapat digunakan dengan cara memanggil fungsi *StemmerFactory*. Kode program proses *stemming* dapat dilihat pada gambar 5.6

```
1  import pandas as pd
2  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
3
4  df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\datatrain.csv',
5  encoding = "latin1" )
6
7  factory = StemmerFactory()
8  stemmer = factory.create_stemmer()
9  df['Berita'] = df.Berita.apply(lambda x : stemmer.stem(x))
```

Gambar 5.6 Kode Program Stemming

Pada gambar 5.6. baris ke-2 dilakukan untuk mengimport *library* Sastrawi. Pada baris ke-4 dilakukan pembacaan file yang sudah mengalami proses *stopword removal* . Pada baris ke-9 dilakukan proses stemming dengan menggunakan *library*

Sastrawi pada kolom *Berita* dengan aturan *stemming* yang sudah dijelaskan pada sub bab 4.5.2

5.6 Implementasi Seleksi Fitur

Metode seleksi fitur yang digunakan pada penelitian ini menggunakan TF-IDF dan *chi square*. Penjelasan tentang kedua metode ini sudah terpapar dalam sub bab 3.6 dan 3.7. Dalam pengambilan fitur, diambil sejumlah n kata dengan bobot TF-IDF dan *chi square* tertinggi atau biasa disebut dengan metode *top-n*. Dalam penelitian ini akan dicari nilai n terbaik untuk masing-masing tokoh politik untuk model yang optimal.

5.6.1 Implementasi Seleksi Fitur TF-IDF

Cuplikan kode implementasi seleksi fitur TF-IDF dapat dilihat pada gambar 5.7.

```
1  import pandas as pd
2  from sklearn.feature_extraction.text import TfidfVectorizer
3
4  df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\datatrain.csv',
5  encoding = "latin1" )
6  X = df.Berita
7
8  tfidf_vectorizer = TfidfVectorizer(min_df = 1)
9  tfidf_matrix = tfidf_vectorizer.fit_transform(X)
10 fn = tfidf_vectorizer.get_feature_names()
11 sums = tfidf_matrix.sum(axis=0)
12 data = []
13 for col, fn in enumerate(fn):
14     data.append( (fn, sums[0,col] ))
15
16 ranking = pd.DataFrame(data, columns=['fn','rank'])
17 ranking_sort = ranking.sort_values('rank',inplace=True, ascending=False)
18 list_fitur = ranking['fn'].tolist()
19 top_n = list_fitur[:450]
```

Gambar 5.7 Kode Program Seleksi Fitur TF-IDF

Dalam proses seleksi fitur TF-IDF pada gambar 5.7, digunakan *library* TF-IDF yang dipanggil pada baris ke-2. Pada baris ke-4 dan 5 dilakukan pembacaan dokumen berekstensi .csv. Pada baris ke-6 dilakukan pembacaan isi dari kolom *Berita* yang kemudian disimpan dalam variable *X*. Lalu fungsi *TfidfVectorizer* dipanggil pada baris ke-8 dengan parameter *min-df* bernilai 1. *Min_df* adalah batas minimal dalam pembangunan kosa kata. Ketika membangun kosa kata, perlu diperhatikan frekuensi kemunculan dari suatu kata fitur. Penggunaan *min_df* yang bernilai 1 berarti digunakan batas minimal kemunculan suatu kata fitur pada dokumen dalam dataset adalah 1. Selanjutnya pada baris ke-9 dilakukan

pembobotan menggunakan *library TfidfVectorizer* dari data yang disimpan dalam variabel *X* . Pada baris ke-10 digunakan untuk mendapatkan kata-kata yang didapatkan dari proses seleksi fitur TF-IDF. Pada baris ke-11 dilakukan perhitungan nilai bobot tiap fitur yang dihasilkan. Pada baris ke-12 sampai 14 fitur dan bobot hasil TF-IDF dimasukan kedalam *list* baru bernama data. Pada baris ke-15 variabel *list data* diubah menjadi tipe *data frame* dan disimpan dalam variabel *ranking* dengan judul kolom *fn* untuk nama fitur dan *rank* untuk bobot fitur. Setelah itu variable *ranking* di sorting secara descending untuk mengurutkan nilai bobot yang terbesar pada baris ke-16 dan 17. Selanjutnya diambil fitur dengan nilai bobot tertinggi sesuai jumlah inputan pada top n pada baris ke-18 dan 19

5.6.2 Implementasi Seleksi Fitur Chi Square

Cuplikan kode implementasi proses seleksi fitur *chi square* dapat dilihat pada gambar 5.8

```
1  import numpy as np
2  import pandas as pd
3  from sklearn.feature_extraction.text import CountVectorizer
4  from sklearn.preprocessing import LabelBinarizer
5
6  df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\datatrain.csv',
7  encoding = "latin1" )
8  X = df.Berita
9  y = df.Sentimen
10 vect = CountVectorizer()
11 X_dtm = vect.fit_transform(X)
12 X_dtm = X_dtm.toarray()
13 pd.DataFrame(X_dtm,columns=vect.get_feature_names())
14
15 y_binarized = LabelBinarizer().fit_transform(y)
16
17 observed = np.dot(y_binarized.T, X_dtm)
18
19 class_prob = y_binarized.mean(axis=0).reshape(1,-1)
20 feature_count = X_dtm.sum(axis=0).reshape(1,-1)
21 expected = np.dot(class_prob.T,feature_count)
22 chisq = (observed-expected)**2/expected
23 chisq_score = chisq.sum(axis=0).tolist()
24
25 fitur =[]
26 fitur = vect.get_feature_names()
27
28 hasil_akhir = list(zip(fitur,chisq_score))
29 fitur_terpilih = sorted(hasil_akhir,key=lambda x: x[1], reverse=True)[:5]
```

Gambar 5.8 Kode Program Seleksi Fitur Chi Square

Pada gambar 5.8 , baris ke-6 dan 7 dilakukan pembacaan file berekstensi .csv.Pada baris ke-8 dilakukan pengambilan isi dari kolom *Berita* dan disimpan dalam variabel *X*. Pada baris ke-9 dilakukan pengambilan isi dari kolom *Sentimen* dan disimpan dalam variabel *y* . Setelah itu dilakukan pemanggilan fungsi

CountVectorizer untuk menghitung kemunculan fitur dalam variabel *X* disetiap dokumen dan diubah kedalam bentuk *array X_dtm* pada baris ke-10 sampai 12. Pada baris ke 13 tipe *array X_dtm* diubah kedalam tipe *dataframe* yang berisi nama fitur dan jumlah kemunculanya dimasing-masing dokumen. Setelah itu pada baris ke-15 nilai dari varibel *y* diubah kedalam bentuk angka biner 0 untuk neg dan 1 untuk pos menggunakan fungsi *LabelBinarizer*. Setelah itu dihitung nilai *observed* dari perkalian tiap kelas dan fitur.Pada baris ke-19 dihitung nilai probabilitas tiap kelas. Pada baris ke-20 dihitung nilai kemunculan masing masing fitur diseluruh dokumen. Pada baris ke-21 dihitung nilai *expected* yang merupakan perkalian dari probabilitas tiap kelas dengan kemunculan fitur diseluruh dokumen. Selanjutnya dihitung nilai chi square tiap fitur yang merupakan nilai (*observed-expected*)^2 dibagi dengan *expected* pada baris ke- 22 yang kemudian hasilnya disimpan dalam bentuk *list* pada baris ke-23. Selanjutnya fitur-fitur yang ada disimpan kedalam *list* baru dengan nama fitur pada baris ke-25 sampai 26. Pada baris ke-28 dibuat variabel *hasil_akhir* dengan tipe *list* yang berisi nama fitur beserta bobot *chi square* dan kemudian dilakukan sorting dan diambil top-n terbaik pada baris ke-29.

5.7 Implementasi Training dan Pengujian Model Klasifikasi Sentimen

Tahap pertama dari proses klasifikasi sentimen adalah pembuatan model klasifikasi menggunakan data training. Kode program pembuatan model klasifikasi dapa dilihat pada gambar 5.9

```
from sklearn.cross_validation import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.cross_validation import cross_val_score
import pandas as pd

dn = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\datatrain.csv
encoding = "latin1" )
X = dn.Berita.tolist()
y = dn.Sentimen



df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\fitur.csv',
encoding = "latin1" )
dg = df.Berita.tolist()


X2=[]
for i in X:
    i+=" "
    temp=""
    for j in dg:
        j+=" "
        if j in i:
            temp+=j
            temp+=" "
    X2.append(temp[:-2])

X_train, X_test, y_train, y_test = train_test_split(X2, y ,test_size = 0.3, random_state=1)

vect = CountVectorizer()
vect.fit(X_train)
X_train_dtm = vect.transform(X_train)
X_train_dtm = vect.fit_transform(X_train)
X_train_dtm

X_test_dtm = vect.transform(X_test)
X_test_dtm
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
y_pred_class = nb.predict(X_test_dtm)
X2_dtm = vect.fit_transform(X2)

from sklearn import metrics
aaa = metrics.accuracy_score(y_test, y_pred_class)
con_mat = metrics.confusion_matrix(y_test,y_pred_class)
scores = cross_val_score(nb,X2_dtm,y, cv=10)
print(scores)
```

Gambar 5.9 Kode Program Training Dan Pengujian Model Klasifikasi

Pada gambar 5.9, pada baris ke-1 digunakan *library train_test_split* untuk membagi data training. Pada baris ke-3 digunakan *library cross_val_score* untuk melakukan k-fold cross validation. Pada baris ke 6 sampai 10 dilakukan pembacaan file berektensi .csv dan kemudian dilakukan pengambilan konten kolom *Berita* dan simpan dalam variabel *X* serta pengambilan konten kolom *Sentimen* dan disimpan dalam variabel *y*. Pada baris ke-13 sampai 15 dilakukan pembacaan kata fitur dalam file berekekstensi .csv. Selanjutnya pada baris ke 18-27 dilakukan pencocokan kata fitur dengan teks yang ada di variabel *X* apabila kata fitur ada didalam variabel X maka kata tersebut akan dioutputkan, jika tidak ada maka kata tidak akan dioutputkan. Setelah itu teks tersebut disimpan dalam list X2. Pada baris ke-29

dilakukan pemanggilan fungsi *train_test_split* untuk pembagian data *training* dari data yang disimpan dalam variabel X2 dan y dengan rasio pembagian datanya 30% untuk data *validation* dan masing-masing disimpan dalam variabel *X_train*, *X_test*, *y_train, y_test*. Setelah itu pada baris ke-32 sampai 36 , data *X_train* dihitung kemunculan seluruh fiturnya dan disimpan dalam variabel *X_train_dtm* dalam bentuk *term document matrix*. Pada baris ke-38 dan 39 , data *X_test* dihitung kemunculan fitur nya dan diubah kedalam bentuk *term document matrix* . Pada baris ke-40 di gunakan *library Multinomial NB* untuk proses klasifikasi. Pada baris ke-42 dipanggil fungsi *Multinomial NB* untuk memahami *vocabulary* data dari X_train_dtm dan y_train dan disimpan variabel *nb*. Selanjutnya pada baris ke-43, dari hasil pemahaman *vocabulary* data *X_train* dan *y_train* , digunakan fungsi *predict* untuk memprediksi hasil sentimen dari *X_test_dtm* dan hasilnya disimpan dalam variabel *y_pred_class*. Pada baris ke-45 dipanggil *library metric* untuk mengukur akurasi hasil prediksi. Pada baris ke-46 dipanggil fungsi *metrics.accuracy_score* untuk menghitung akurasi antara hasil sentimen data tes *(y_test)* dengan hasil prediksi klasifikasi (*y_pred_class*) yang kemudian disimpan dalam variabel *akurasi_model*. Untuk evaluasi model yang dihasilkan, dilakukan perhitungan *confusion matrix* dan *k-fold cross validation* pada baris ke-47 dan 48. *Confusion matrix* dibuat dengan memanggil fungi *metrics.confusion_matrix* dari hasil sentimen data tes (*y_test*) dengan hasil prediksi klasifikasi (*y_pred_class*). Untuk melakukan *k-fold cross validation* dipanggil fungsi cross_val_score dengan metode klasfikasi *Multinomial Naïve Bayes* dengan menggunakan data validation *X_test_dtm* dan hasil prediksi klasifikasinya (*y_pred_class*) dengan jumlah fold sebanyak 10 dan hasilnya disimpan dalam variabel *scores*.

5.8 Implementasi Testing Model Klasifikasi Sentimen

Setelah didapatkan model klasifikasi terbaik selanjutnya dilakukan proses klasifikasi sentimen terhadap data testing. Hal yang dilakukan pertama dalam klasifikasi sentimen data testing adalah mengimpor data latih, data tes dan fitur. Proses mengimpor data latih,data tes dan fitur ditunjukan pada gambar 5.10

```
import csv
import numpy as np
import pandas as pd

with open('contoh_train_ahy.csv', encoding='windows-1252') as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')

    train_tweets = []
    classlabels = []

    for row in readCSV:
        train_tweet = row[0]
        classlabel = row[1]
        train_tweets.append(train_tweet)
        classlabels.append(classlabel)
with open('contoh_testing_ahy.csv', encoding='windows-1252') as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    test_tweets = []

    for row in readCSV:
        test_tweet = row[0]
        test_tweets.append(test_tweet)

df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\dataset\fitur.csv',
encoding = "latin1" )
dn = df.Berita.tolist()

train_tweets2=[]
for i in train_tweets:
    i+=" "
    temp=""
    for j in dn:
        j+=" "
        if j in i:
            temp+=j
            temp+=" "
    train_tweets2.append(temp[:-2])
```

Gambar 5.10 Kode Program Untuk Mengimpor Data Training,Testing, Dan Fitur

Pada gambar 5.10, Pada baris ke-5 sampai 15 dilakukan pembacaan file data *training* dengan format file .csv. Tweet dan berita pada data *training* kemudian disimpan dalam *list* baru dengan variabel *train_tweets*, sementara label-label pada data *training* disimpan dalam list baru dengan variabel *classlabels*. Pada barus ke-16 sampai 22 dilakukan pembacaan file data *testing* dengan format file .csv. Tweet dan berita pada data *testing* kemudian disimpan dalam *list* baru dengan variabel *test_tweets*. Pada baris ke-24 sampai 26 dilakukan pembacaan file fitur dengan format file .csv dan diubah tipe data *dataframe* menjadi *list* dan disimpan dalam variabel *dn*. Pada baris ke-28 sampai 37, dilakukan pencocokan kata dalam variabel *dn* yang berisi fitur-fitur terpilih dengan tweet atau berita yang ada didalam data *training*. Jika kata penyusun tweet atau berita dalam data training termuat dalam *dn* maka kata tersebut akan di *keep* , jika kata tidak ada maka tidak dipakai. Kata-kata yang di *keep* kemudian disimpan dalam variabel baru dengan nama *train_tweets2*.

Setelas proses mengimpor data *training*, data *test* dan fitur selesai, selanjutnya dilakukan proses ekstraksi fitur. Untuk kode program ekstraksi fitur dapat dilihat pada gambar 5.11

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 vectorizer_train = CountVectorizer()
3 vectorizer_test = CountVectorizer()
4
5
6 vectorizer_train.fit(train_tweets2)
7 vectorizer_test.fit(test_tweets)
8
9
10 vector_train_vocab = vectorizer_train.vocabulary_
11 vector_test_vocab = vectorizer_test.vocabulary_
12
13 vector_train = vectorizer_train.transform(train_tweets2)
14 vector_test = vectorizer_test.transform(test_tweets)
15
16 vector_train_array = vector_train.toarray()
17 vector_test_array = vector_test.toarray()
18
19 x1, y1 = vector_train.shape
20 x, y = vector_test.shape
```

Gambar 5.11 Kode Program Untuk Ekstraksi Fitur

Pada gambar 5.11, pada baris ke-1 sampai 7 digunakan *library CountVectorizer* untuk mengubah data dalam *train_tweets2* dan *test_tweets* menjadi bentuk matriks sesuai dengan jumlah tokenya. Proses ini sebagai ekstraksi fitur agar data berubah ke bentuk vektornya sehingga dapat digunakan dalam algoritma klasifikasi. Selanjutnya, hasilnya ditampilkan dalam bentuk kosakata atau *vocabulary* bertipe data *dictionary* pada baris ke-10 dan 11. Pada baris ke-13 sampai 17 dilakukan pengubahan data *training* dan data *testing* menjadi bentuk array yang masing-masing token pada tiap elemen array sesuai dengan kosakata dan berisikan jumlah kemunculan token pada tweet atau berita tersebut. Setelah proses ekstraksi fitur dilakukan perhitungan probabilitas prior dari masing-masing kelas. Kode program perhitungan probabilitas prior ditunjukan pada gambar 5.12

```
1 npos = 0
2 nneg = 0
3
4 for classlabel in classlabels:
5     if classlabel == "pos":
6         npos += 1
7     elif classlabel == "neg":
8         nneg += 1
9
10
11 ppos = npos/(npos+nneg)
12 pneg = nneg/(npos+nneg)
```

Gambar 5.12 Kode Program Untuk Menghitung Probabilitas Prior

Algortima Multinomial Naïve Bayes digunakan dalam melakukan klasifikasi sentimen. Oleh karena itu, diperlukan perhitungan probabilitas *prior* yang didasarkan pada persamaan (4.1). Pada gambar 5.12, terdapat 2 jenis probabilitas *prior* yang akan dihitung sesuai dengan label kelas masing-masing, yakni positif dan negatif. Pada baris ke-1 sampai 8 , dilakukan perulangan untuk membaca *classlabel*, jika *classlabel* bernilai *pos* maka *npos* ditambah 1, jika bernilai *neg* maka nneg ditambah 1. Selanjutnya probabilitas *prior* positif didapatkan dengan membagi jumlah data *training* yang berlabel *pos* dengan jumlah keseluruhan data *training*. Probabilitas *prior* negatif didapatkan dengan membagi jumlah data training yang berlabel *neg* dengan jumlah keseluruhan data *training*. Hal ini ditunjukan pada baris ke-11 dan 12. Selain perhitungan probabilitas *prior*,algoritma Multinomial Naïve Bayes juga memerlukan perhitungan probabilitas kondisinal. Kode program untuk menghitung probabilitas kondisional ditunjukan pada gambar 5.13.

```
1 def conditional_probability(labels):
2     results = []
3     count = search_through(labels,make_slice[:])
4
5 sorted_vector_test_vocabs = sorted(vector_test_vocab, key=vector_test_vocab.__getitem__)
6     for sorted_vector_test_vocab in sorted_vector_test_vocabs:
7         for key2, value2 in vector_train_vocab.items():
8             if sorted_vector_test_vocab == key2:
9                 count2 = search_through(labels,value2)
10                calc = (count2 + 1)/(count + y1)
11                results.append(calc)
12
13    return results
```

Gambar 5.13 Kode Program Untuk Menghitung Probabilitas Kondisional

Pada gambar 5.13, proses perhitungan probabilitas kondisional terletak pada fungsi *conditional_probability*. Fungsi *search_through* pada baris ke 3 digunakan untuk menghitung jumlah label yang kemudian disimpan kedalam variabel *count*.

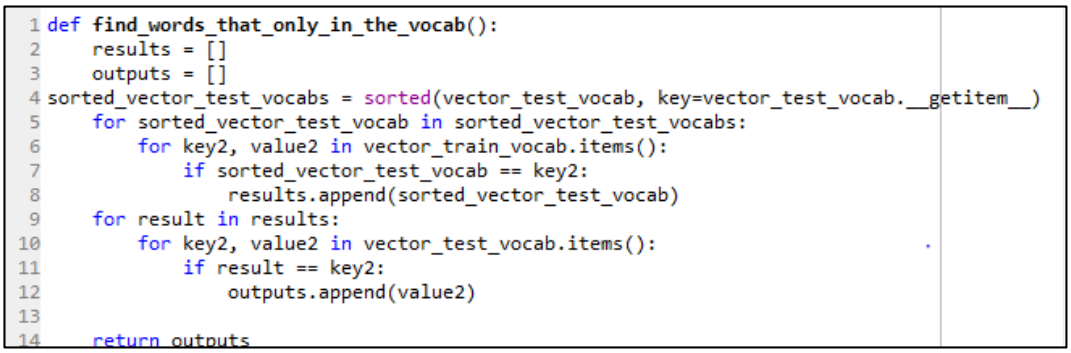
Fungsi *sorted* digunakan untuk mengurutkan variabel *dictionary vector_test_vocab* berdasarkan jumlah kemunculan terkecil pada masing-masing elemen variabel tersebut. Hasilnya kemudian dimasukan kedalam variabel *sorted_vector_test_vocab* pada baris ke-5. Pada baris ke-6 sampai 11, *sorted_vector_test_vocabs* di iterasi untuk setiap elemenya. Pada setiap iterasi tersebut, variabel *dictionary vector_train_vocabs* juga diiterasi untuk setiap itemnya. Jika suatu elemen pada *sorted_vector_test_vocabs* sama dengan *key* pada *vector_train_vocabs*,maka fungsi *search_through* digunakan untuk menghitung. Hasilnya kemudian dimasukan kedalam variabel *count2*. Selain itu , probabilitas kondisional dihitung berdasarkan pada persamaan 4.2 yaitu menggunakan rumus *(count2 + 1)/(count + y1)* yang kemudian hasilnya disimpan dalam variabel *calc*. Setelah hasilnya dalam variabel calc dimasukan kedalam *array results*. Fungsi *conditional_probability* mengembalikan variabel *results* melalui *statement return* pada baris ke 12.

Pada fungsi *search_through* , label yang sama akan dicari untuk setiap elemen dengan nilai tertentu pada variabel *array classlabels*. Hal ini dilakukan dengan mengiterasi variabel *classlabels* pada baris ke 3 pada gambar 5.14. Ketika label yang ingin dicari sama dengan elemen pada variabel *array classlabels* maka fungsi numpy.sum digunakan untuk menjumlahkan setiap elemen pada variabel array *vector_train_array*. Dimana barisnya sesuai iterasi saat itu dan kolomnya berdasarkan input pengguna.

```
1 def search_through(labels, value):
2     count = 0
3     for position, classlabel in enumerate(classlabels):
4         if labels == classlabel:
5             count += np.sum(vector_train_array[position,value])
6     return count
```

Gambar 5.14 Kode Program Untuk Mencari Setiap Elemen Yang Sama Dengan Suatu Nilai Tertentu Pada Variabel Classlabel

Dalam proses klasifikasi, data *testing* harus memuat kata yang hanya ada didalam data *training* saja.Kode program untuk mencari indeks kata-kata data testing yang hanya terdapat pada di *vocabulary* ditunjukan pada gambar 5.15



```
1  def find_words_that_only_in_the_vocab():
2      results = []
3      outputs = []
4  sorted_vector_test_vocabs = sorted(vector_test_vocab, key=vector_test_vocab.__getitem__)
5      for sorted_vector_test_vocab in sorted_vector_test_vocabs:
6          for key2, value2 in vector_train_vocab.items():
7              if sorted_vector_test_vocab == key2:
8                  results.append(sorted_vector_test_vocab)
9      for result in results:
10         for key2, value2 in vector_test_vocab.items():
11             if result == key2:
12                 outputs.append(value2)
13
14     return outputs
```

Gambar 5.15 Kode Program Untuk Mencari Indeks Kata-Kata Pada Data Test Yang Hanya Terdapat Di Vocabulary

Pada gambar 5.15 baris ke-4, fungsi sorted digunakan untuk mengurutkan variabel *dictionary vector_test_vocab* berdasarkan jumlah kemunculan terkecil pada masing-masing elemenvariabel tersebut. Hasilnya kemudian dimasukkan ke dalam variabel '*sorted_vector_test_vocabs*'. Variabel '*sorted_vector_test_vocabs*' kemudian diiterasi untuk setiap elemennya. Pada setiap iterasi tersebut, variabel *dictionary* '*vector_train_vocabs*' juga diiterasi untuk setiap *item*nya seperti yang ditunjukkan pada baris 5 dan 6. Jika suatu elemen pada '*sorted_vector_test_vocabs*' sama dengan *key* pada '*vector_train_vocabs*', maka elemen '*sorted_vector_test_vocabs*' yang sama tersebut dimasukkan ke dalam variabel array '*results*'. Hal ini ditunjukkan pada baris ke-5 sampai 8. Ketika seluruh iterasi sebelumnya telah selesai, dilakukan perulangan untuk setiap elemen pada variabel '*results*'. Untuk setiap iterasi dalam perulangan, variabel *dictionary* '*vector_test_vocabs*' juga diiterasi untuk setiap *item*-nya seperti yang ditunjukkan pada baris 9 dan 10. Jika suatu elemen pada '*results*' sama dengan *key* pada '*vector_test_vocabs*', maka *value* dari '*vector_test_vocabs*' yang sama tersebut dimasukkan ke dalam variabel array '*outputs*'. Proses ini ditunjukkan pada baris 11 dan 12. Fungsi *find_words_that_only_in_the_vocab* mengembalikan variabel '*outputs*' melalui *statement return* pada baris 1.

Proses terakhir adalah proses pengklasifikasian sentiment. Kode program untuk proses klasifikasi sentiment ditunjukan pada gambar 5.16

```
poscount = 0
negcount = 0
hasil =[]
c = find_words_that_only_in_the_vocab()
resultpos = np.array(conditional_probability("pos"))
resultneg = np.array(conditional_probability("neg"))

for index, test_tweet in enumerate(test_tweets):
    a = np.array(vector_test_array[index,c])

    sum_pos = np.prod(resultpos**a)*ppos
    sum_neg = np.prod(resultneg**a)*pneg


    if (sum_pos >= sum_neg):
        hasil.append("pos")
        poscount += 1
    elif (sum_neg > sum_pos):
        hasil.append("neg")
        negcount += 1
b=list(zip(test_tweets,hasil))
```

Gambar 5.16 Bagian Utama Program Klasifikasi Sentimen

Pada gambar 5.16 , fungsi *find_words_that_only_in_vocab* dipanggil dan disimpan pada variabel *c*. pada baris ke-5 dan 6 dipanggil fungsi *conditional probability* untuk menghitung *conditional probability* dari label *pos* dan *neg* yang hasilnya kemudian diubah kedalam bentuk array dan disimpan pada variabel *resultpos* dan *resultneg*.Pada baris ke-8 sampai 9,untuk setiap elemen pada variabel *array test_tweet* , pada setiap iterasi perulanganya, variabel *a* didefinisikan sebagai *array* yang berisikan elemen dari *array vector_test_array* dengan index baris sesuai dengan iterasi perulangan dan index kolom sesuai dengan variabel *c* yang berisi token kata yang hanya ada di *vocabulary*. Variabel *a* ini berisi banyaknya kemunculan suatu kata pada data test. Masih dalam perulangan , probabilitas suatu kelas pada data *testing* dihitung dan didefinisikan dalam variabel *sum_pos* dan *sum_neg* pada baris ke-11 dan 12. Untuk menghitung probabilitas kelas positif pada data testing,variabel *sum_pos* didapat dengan memanggil fungsi *np.prod* terhadap variabel *resultpos* yang dipangkatkan dengan variabel *a*. Hasilnya kemudian dikalikan dengan probabilitas prior positif yang ditunjukan dengan variabel *ppos*.Fungsi *np.prod* adalah mengalikan setiap elemen pada array. Untuk menghitung probabilitas kelas negatif pada data testing,variabel *sum_neg* didapat dengan memanggil fungsi *np.prod* terhadap variabel *resultneg* yang dipangkatkan dengan variabel *a*. Hasilnya kemudian dikalikan dengan probabilitas prior negatif yang ditunjukan dengan variabel pneg. Pada baris ke-15 sampai 20 dilakukan

penentuan setiap data dalam data *testing* diklasifikasikan kedalam kelas positif atau negatif. Jika nilai *sum_pos* lebih besar dari *sum_neg* maka data tersebut akan diklasifikasikan kedalam kelas positif. Sebaliknya, jika *sum_neg* lebih besar dari *sum_pos* maka data tersebut diklasifikasikan kedalam kelas negatif. Pada baris ke-21 dibuat *list* yang berisi data *testing* beserta hasil pengklasifikasianya dan disimpan dalam variabel *b*.

5.9 Implementasi Perhitungan Elektabilitas PvT

Setelah data *testing* diklasifikasikan, selanjtnya dihitung elektabilitas tiap tokoh politiknya. Kode program perhitungan elektabilitas PvT tiap tokoh ditunjukan pada gambar 5.17.

```
1  import pandas as pd
2  df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\elektabilitas.csv')
3  tokoh_politik= ['ahok' , 'anies', 'prabowo', 'zulkifli',
4  'ahy', 'gatot', 'jk', 'hary', 'ridwan', 'jokowi' ]
5  hasil = []
6  for  column in df.iteritems():
7      a = column[1].tolist()
8      y2 = a.count('neg')
9      y1 = a.count('pos')
10     total = y1+y2
11     PvT1 = y1/total
12     hasil.append(PvT1)
13
14 total_elektabilitas = sum(hasil)
15 df = pd.DataFrame(hasil)
16 normalisasi = df/total_elektabilitas*100
17 hasil_norm = normalisasi[0].tolist()
18 elektabilitas_norm = list(zip(tokoh_politik,hasil_norm))
```

Gambar 5.17 Kode Program Perhitungan Elektabilitas Pvt

Pada gambar 5.17 , perhitungan elektabilitas didasarkan pada konsep *positive versus total* yang sudah dijelaskan pada sub bab 3.10. Pada baris ke-2 dilakukan pembacaan dokumen file berektensi .csv yang berisi hasil klasifikasi sentimen tiap tokoh politik yang disimpan dalam variabel *df*. Setelah itu pada baris ke-3 dan 4 dibuat sebuah *list* baru yang berisi nama-nama tokoh politik yang digunakan. Pada baris ke-6 sampai 12 proses *positive versus total* dilakukan. Setiap kolom didalam variabel *df* akan diiterasi. Pada setiap iterasinya setiap kolom diubah menjadi list dan dihitung jumlah nilai positifnya lalu dimasukan keladam variabel *y1* dan dihitung jumlah nilai negatifnya lau dimasukan kedalam variabel *y2*. Setelah itu dihitung total data nya dengan menjumlahkan jumlah y1 dan y2. Nilai *positive versus total* tiap tokoh kemudian dihitung dengan rumus *y1/total* yang kemudian disimpan dalam variabel PvT1. Hasil keseluruhan nilai elektabilitas setiap tokoh

politik kemudian disimpan dalam *list hasil*. Pada baris ke-14 sampai 18 dilakukan perhitungan nilai normalisasi elektabilitas ke sepuluh tokoh politik. Total elektabilitas yang didapat setiap tokoh dijumlahkan dan disimpan dalam variabel *total_elektabilitas*. Nilai normalisasi tiap tokoh politik dihitung dengan membagi nilai elektabilitas tiap tokoh dengan total elektabilitasnya kemudian hasilnya disimpan dalam *list* baru dengan nama varibel *hasil_norm*. Kemudian dibuat list baru yang berisi nama tokoh politik beserta hasil normalisasi elektabilitasnya.

5.10. Implementasi perhitungan elektabilitas SoV

Kode implementasi perhitungan elektabilitas SoV tiap tokoh politik ditunjukan pada gambar 5.18

```
1  import pandas as pd
2
3  df = pd.read_csv(r'E:\scan Ryan\Ryan Document\MATKUL\SMT 7\SKRIPSWEET\elektabilitas.csv')
4  tokoh_politik= ['ahok' , 'anies', 'prabowo', 'zulkifli',
5  'ahy', 'gatot', 'jk', 'hary', 'ridwan', 'jokowi' ]
6  y1_total = []
7  hasil = []
8  for  column in df.iteritems():
9      a = column[1].tolist()
10     y1 = a.count('pos')
11     y1_total.append(y1)
12     total_pos = sum(y1_total)
13
14 df = pd.DataFrame(y1_total)
15 elektabilitas = df/total_pos
```

Gambar 5.18 Kode Program Perhitungan Elektabilitas Pvt

Pada gambar 5.18 , baris ke-3 digunakan untuk mengimpor file berekstensi .csv yang berisi hasil klasifikasi sentiment tiap tokoh politik. Pada baris ke-4 sampai 5 dibuat *list* berisi nama-nama tokoh politik. Pada barus ke-8 sampai 12 dilakukan perhitungan total jumlah sentiment positf setiap tokoh dan dimasukan kedalam variabel *list y1_total* yang kemudian dihitung total jumlah sentiment positif kesepuluh tokoh yang disimpan pada variabel *total_pos*. Pada baris ke-14 sampai 15 dihitung nilai elektabilitas tiap tokoh politik dengan membagi jumlah sentimen positif tiap tokoh dalam variabel *df* dengan *total_pos*

BAB VI
HASIL DAN ANALISA

6.1 Hasil Preprocessing

Tahap preprocessing yang dilakukan adalah *case folding*, *regular expression* dan tokenisasi. Pada gambar 6.1 ditunjukan data sebelum dilakukan tahap preprocessing dan gambar 6.2 adalah data sesudah mengalami tahap *preprocessing*

2	RT @denirisman: Alexis Sumbang Pemasukan Pajak 30 M, Anies: Lalu Pelanggarannya Dibiarkan? | SwaMedium https://t.co/8t0JALzo6c lewat @swame[...]
3	RT @KedahTawakal: Tidak Perlu Tunggu Hasil Raperda, Anies-Sandi Segera Cabut Izin Reklamasi https://t.co/GF6UF4p6nc
4	RT @Umnia77: Ada Apa Dengan PKB Zaman Now..???n#hey2 nPKB Tantang Anies Beberkan Bukti2 Pelanggaran Alexis nhttps://t.co/D1SFHwEGER
5	RT @KedahTawakal: Setelah Alexis, Anies Janji Tutup Semua Tempat Prostitusi di Jakarta https://t.co/7vhzVhCpcP
6	RT @republikaonline: Pajak Alexis, Anies: Gak Halal, Gak Berkah https://t.co/IyisGi1zZb
7	RT @roninpribumi: KAU ini maunya apa @kompascom? Kemarin blow up tantangan Ahok ke Anies tutup Alexis. nGiliran ditutup beneran sok2an pedu[...]
8	RT @PrijantoRabbani: GNPF Ulama Dukung Pernyataan Anies Baswedan Soal Pribumi https://t.co/kZ4JICyeEc
9	Izin Alexis Dihentikan, Anies: Kita ingin Uang Halal dari Kerja yang Halal https://t.co/yXHpubkNgC via
10	RT @maspiyuuu: GERAK ANIES MENUTUP RUMAH BORDIL https://t.co/wuBjWZXa6P https://t.co/vxAFCw4H0O
11	RT @VIVAcoid: Anies Sebut 10 Proyek Infrastruktur Tak Punya Amdal Lalin https://t.co/gDaFRlUvA1

Gambar 6.1 Data Sebelum Tahap *Preprocessing*

Data yang menjadi masukan tahap *preprocessing* ini akan melalui tahap *case folding* yaitu tahap mengubah *case* keseluruhan teks menjadi *lowercase* , *regular expression* yaitu penghilangan URL, *bracket*, *mention*, *hashtag*, *RT*, *non-alphanumeric character* dan yang terakhir tokenisasi yaitu memecah kalimat-kalimat dalam berita menjadi kata-kata yang menyusun kalimat-kalimat tersebut. Pada gambar 6.2 ditunjukan hasil tahap *preprocessing*

```
['alexis', 'sumbang', 'pemasukan', 'pajak', '30', 'm', 'anies', 'lalu',
'pelanggarannya', 'dibiarkan', 'swamedium', 'lewat']
['tidak', 'perlu', 'tunggu', 'hasil', 'raperda', 'anies', 'sandi', 'segera',
 'cabut', 'izin', 'reklamasi']
['ada', 'apa', 'dengan', 'pkb', 'zaman', 'now', 'n', 'npkb', 'tantang', 'anies',
 'beberkan', 'bukti2', 'pelanggaran', 'alexis', 'n']
['setelah', 'alexis', 'anies', 'janji', 'tutup', 'semua', 'tempat', 'prostitusi',
 'di', 'jakarta']
['pajak', 'alexis', 'anies', 'gak', 'halal', 'gak', 'berkah']
['kau', 'ini', 'maunya', 'apa', 'kemarin', 'blow', 'up', 'tantangan', 'ahok', 'ke',
 'anies', 'tutup', 'alexis', 'ngiliran', 'ditutup', 'beneran', 'sok2an', 'pedu']
['gnpf', 'ulama', 'dukung', 'pernyataan', 'anies', 'baswedan', 'soal', 'pribumi']
['izin', 'alexis', 'dihentikan', 'anies', 'kita', 'ingin', 'uang', 'halal', 'dari',
'kerja', 'yang', 'halal', 'via']
['gerak', 'anies', 'menutup', 'rumah', 'bordil']
['anies', 'sebut', '10', 'proyek', 'infrastruktur', 'tak', 'punya', 'amdal', 'lalin']
```

Gambar 6.2 Data Sesudah Tahap *Preprocessing*

6.2 Hasil Filtering

Pada tahap filtering, data berbentuk token yang sudah dihasilkan dari tahap *preprocessing* akan melalui tahap stopword removal dan stemming. Stopword removal adalah penghilangan kata-kata yang ada pada list stopword dan stemming adalah penghilangan imbuhan kata atau dengan kata lain mengubah semua kata menjadi bentuk kata dasar dari masing-masing kata. Data sebelum tahap filtering dapat dilihat pada gambar 6.2, kemudian keluaran dari tahap filtering ditunjukan pada gambar 6.3

2	alexis sumbang pasu pajak anies langgar biar swamedium
3	tunggu hasil raperda anies sandi cabut izin reklamasi
4	pkb zaman now npkb tantang anies kan bukti langgar alexis
5	alexis anies janji tutup prostitusi jakarta
6	pajak alexis anies halal berkah
7	kau kemarin blow tantang ahok anies tutup alexis ngiliran tutup beneran sok pedu
8	gnpf ulama dukung nyata anies baswedan pribumi
9	izin alexis henti anies uang halal halal via
10	gerak anies tutup rumah bordil
11	anies proyek infrastruktur amdal lalin

Gambar 6.3 Data Sesudah Tahap *Filtering*

6.3 Hasil Seleksi Fitur TF-IDF

Seleksi fitur TF-IDF seperti yang telah dipaparkan pada sub bab 5.6.1 , setelah dihitung bobot TF-IDF dari setiap kata dalam teks, dicari sejumlah n-kata dengan bobot TF-IDF tertinggi dari seluruh data *training* untuk dijadikan kata fitur

yang nantinya akan dipakai untuk membentuk model klasifikasi. Berikut cuplikan daftar kata fitur yang berisi kata-kata dengan bobot TF-IDF yang ditunjukan pada gambar 6.4

Term	Nilai TF-IDF
agus	-0.178
yudhoyono	-0.178
lemah	0.132
bakat	0.1
kerja	0.1
jk	0.064

Gambar 6.4 Cuplikan Kata Fitur Hasil TF-IDF

6.4 Hasil Seleksi Fitur Chi Square

Seleksi fitur *chi square* seperti yang telah dipaparkan pada sub bab 5.6.2 , setelah dihitung nilai *chi square* dari setiap kata dalam teks, dicari sejumlah n-kata dengan nilai *chi square* tertinggi dari seluruh data *training* untuk dijadikan kata fitur yang nantinya akan dipakai untuk membentuk model klasifikasi. Berikut cuplikan daftar kata fitur yang berisi kata-kata dengan nilai *chi square* yang ditunjukan pada gambar 6.5

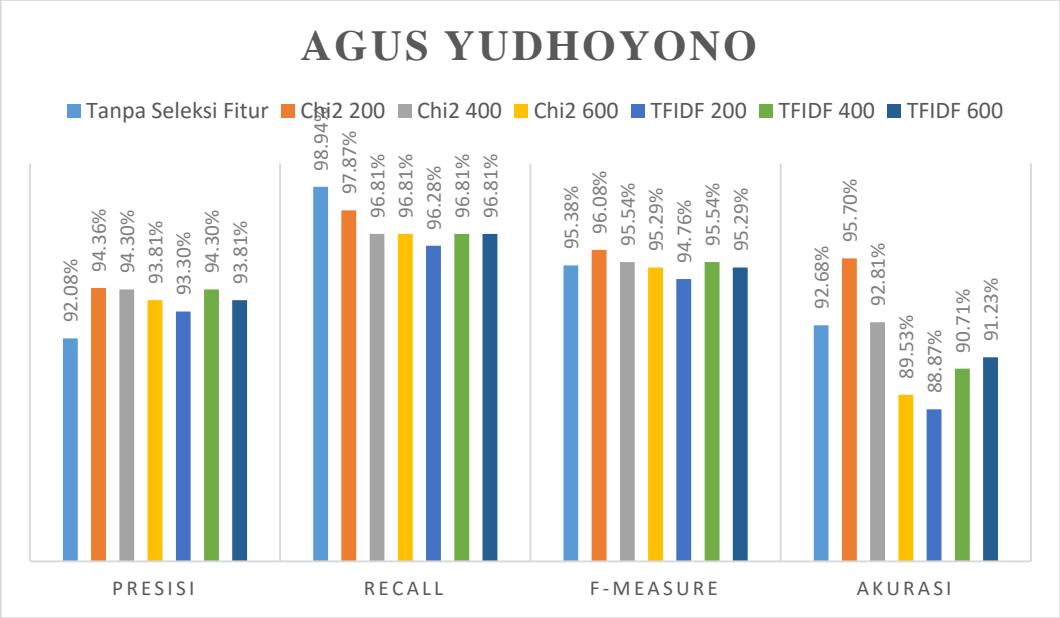
Term	Nilai Chi Square
agus	0
yudhoyono	0
lemah	0.666666667
bakat	0.333333333
kerja	0.333333333
jk	0.166666667

Gambar 6.5 Cuplikan Kata Fitur Hasil Chi Square

6.5 Hasil Pengujian Perbandingan Fitur Top-n

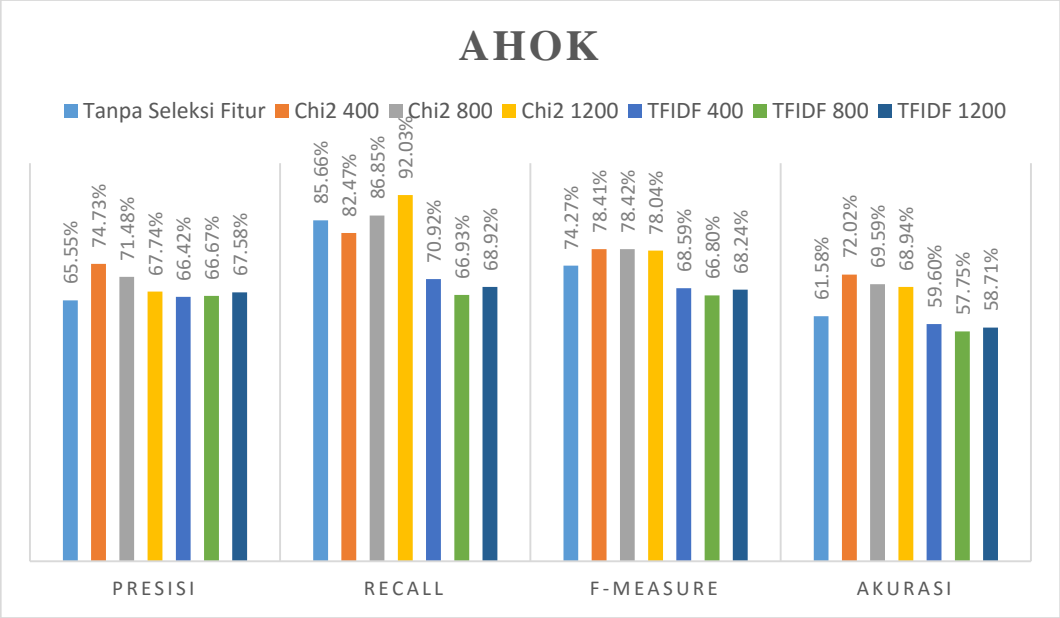
Dalam menentukan model terbaik dilakukan penentuan jumlah kata fitur yang digunakan, dilakukan percobaan dengan pengukuran peformansi untuk memutuskan berapa jumlah kata fitur yang akan dipakai. Berikut ditampilkan hasil

pengukuran performansi dengan memvariasikan jumlah kata fitur tiap-tiap tokoh politik pada model klasifikasi sentimen menggunakan Multinomial Naïve Bayes. Variasi jumlah fitur tiap tokoh berbeda-beda dikarenakan jumlah data *training* setiap tokoh juga berbeda-beda.Hasil pengujian performa model klasifikasi untuk tokoh Agus Yudhoyono dapat dilihat pada gambar 6.6



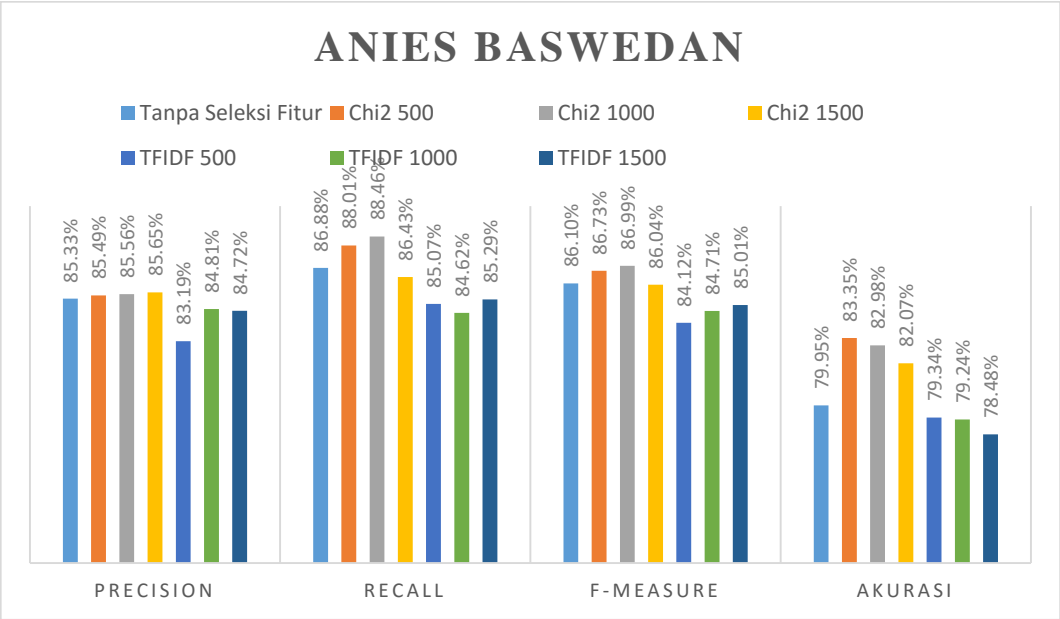
Gambar 6.6 Hasil Performa Model Tokoh Agus Yudhoyono

Pada gambar 6.6 , variasi jumlah fitur untuk tokoh Agus Yudhoyono yang digunakan adalah 200, 400 dan 600 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=600 dimana model dapat mengklasifikasikan 91,23% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=200 dimana model dapat mengklasifikasikan 95,07% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Ahok ditunjukan pada gambar 6.7



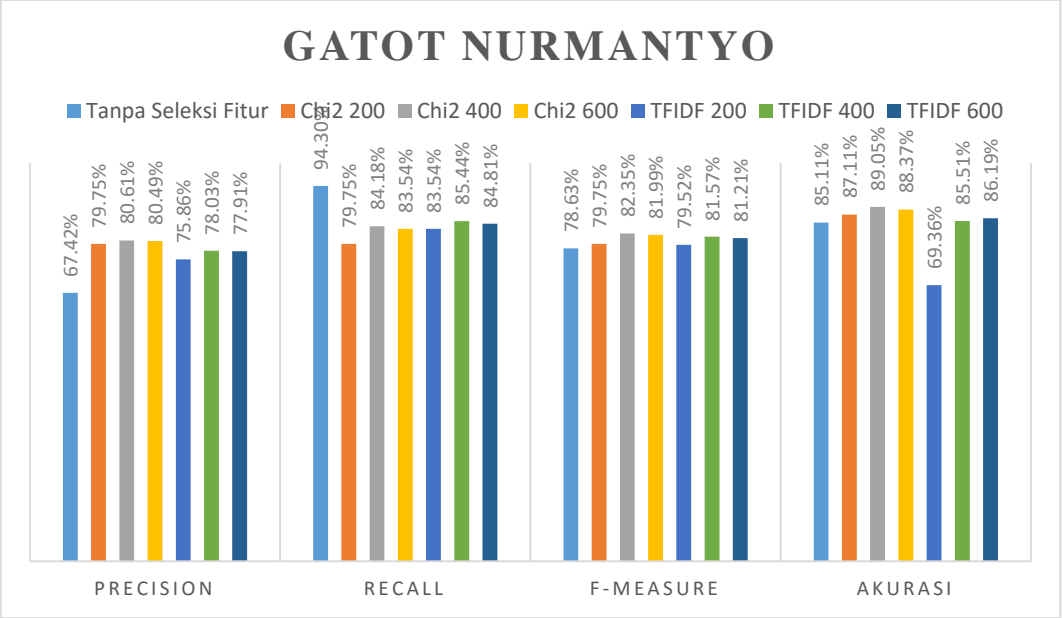
Gambar 6.7 Hasil Performa Model Tokoh Ahok

Pada gambar 6.7 , variasi jumlah fitur untuk tokoh Ahok yang digunakan adalah 400, 800 dan 1200 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=400 dimana model dapat mengklasifikasikan 59,60% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=400 dimana model dapat mengklasifikasikan 72,02% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Anies Baswedan ditunjukan pada gambar 6.8



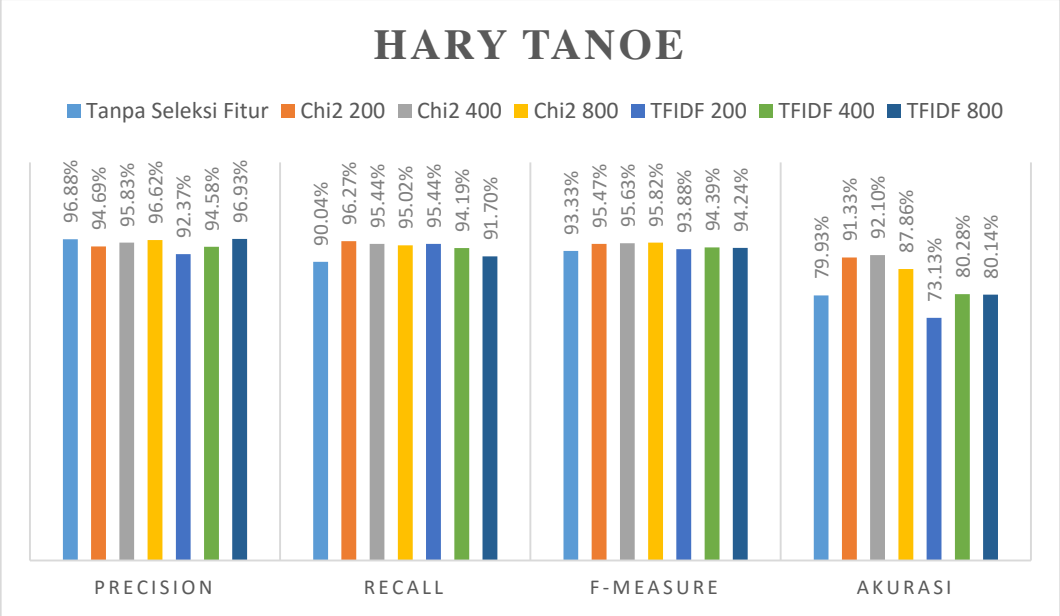
Gambar 6.8 Hasil Performa Model Tokoh Anies Baswedan

Pada gambar 6.8 , variasi jumlah fitur untuk tokoh Anies Baswedan yang digunakan adalah 500, 1000 dan 1500 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=500 dimana model dapat mengklasifikasikan 79,34% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=500 dimana model dapat mengklasifikasikan 83,35% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Gatot Nurmantyo ditunjukan pada gambar 6.9



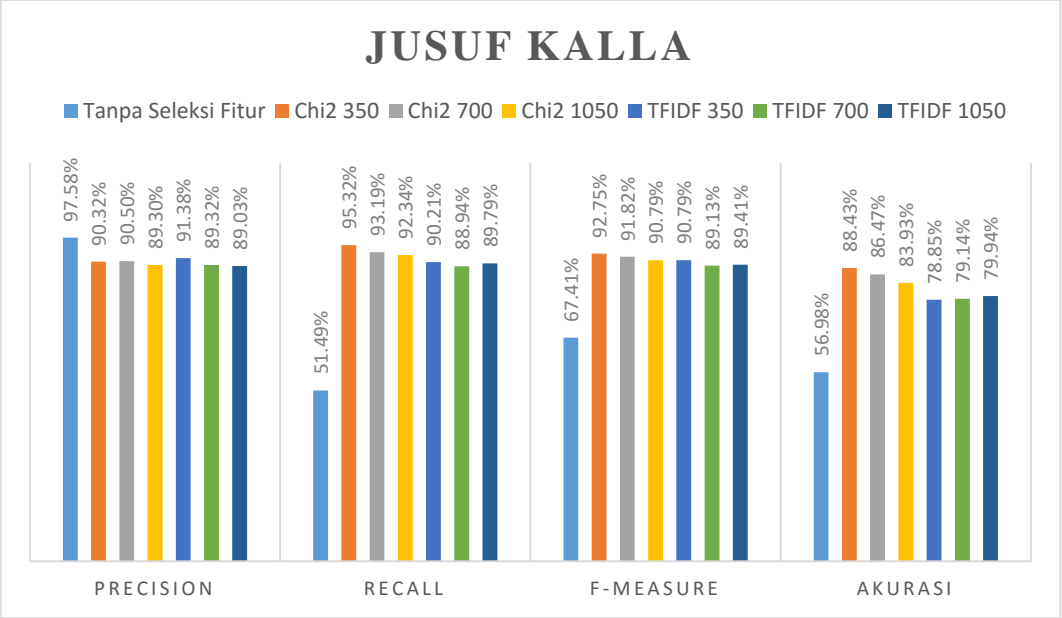
Hasil Performa Model Tokoh Gatot Nurmantyo

Pada gambar 6.9 , variasi jumlah fitur untuk tokoh Gatot Nurmantyo yang digunakan adalah 200, 400 dan 600 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=600 dimana model dapat mengklasifikasikan 86,19% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=400 dimana model dapat mengklasifikasikan 87,11% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Hary Tanoe ditunjukan pada gambar 6.10



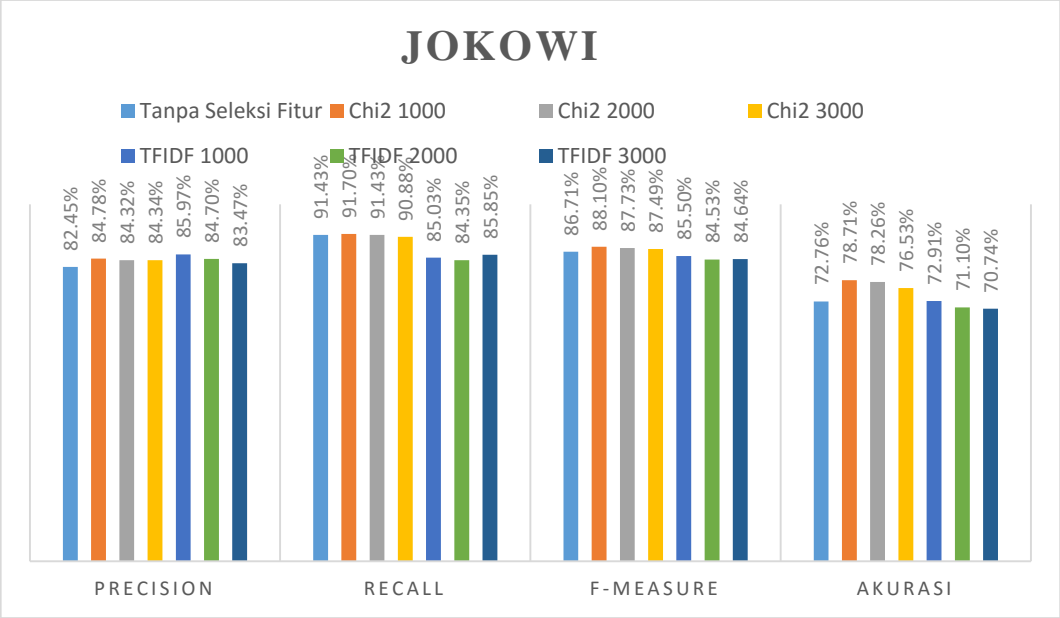
Gambar 6.10 Hasil Performa Model Tokoh Hary Tanoe

Pada gambar 6.10 , variasi jumlah fitur untuk tokoh Hary Tanoe yang digunakan adalah 200, 400 dan 800 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=400 dimana model dapat mengklasifikasikan 80,28% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=400 dimana model dapat mengklasifikasikan 91,33% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Jusuf Kalla ditunjukan pada gambar 6.11



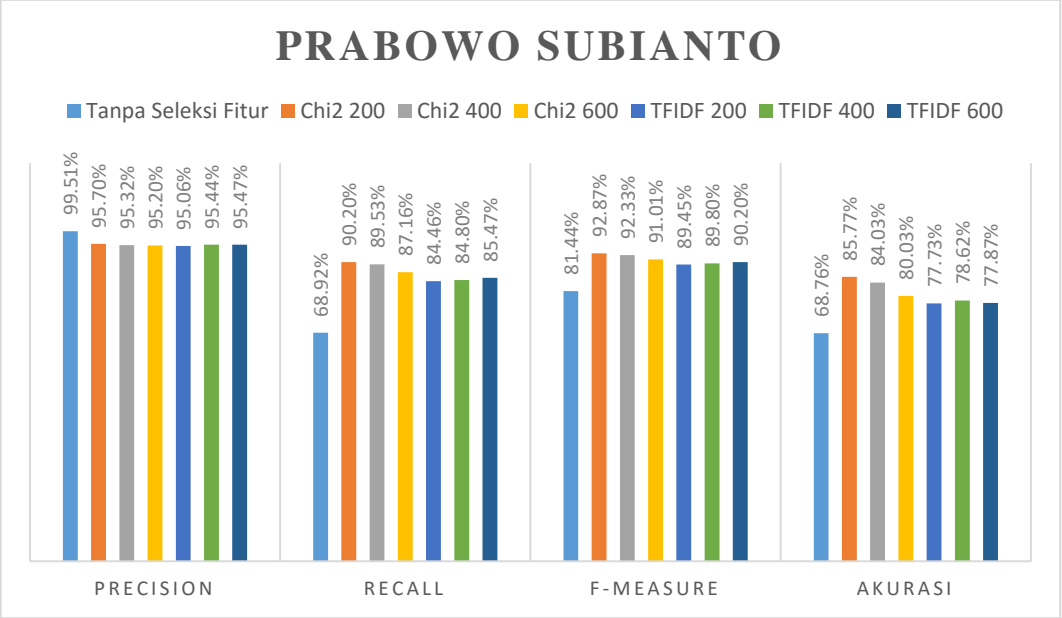
Gambar 6.11 Hasil Performa Model Tokoh Jusuf Kalla

Pada gambar 6.11 , variasi jumlah fitur untuk tokoh Jusuf Kalla yang digunakan adalah 350, 700 dan 1050 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=1050 dimana model dapat mengklasifikasikan 79,94% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=350 dimana model dapat mengklasifikasikan 88,43% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Jokowi ditunjukan pada gambar 6.12



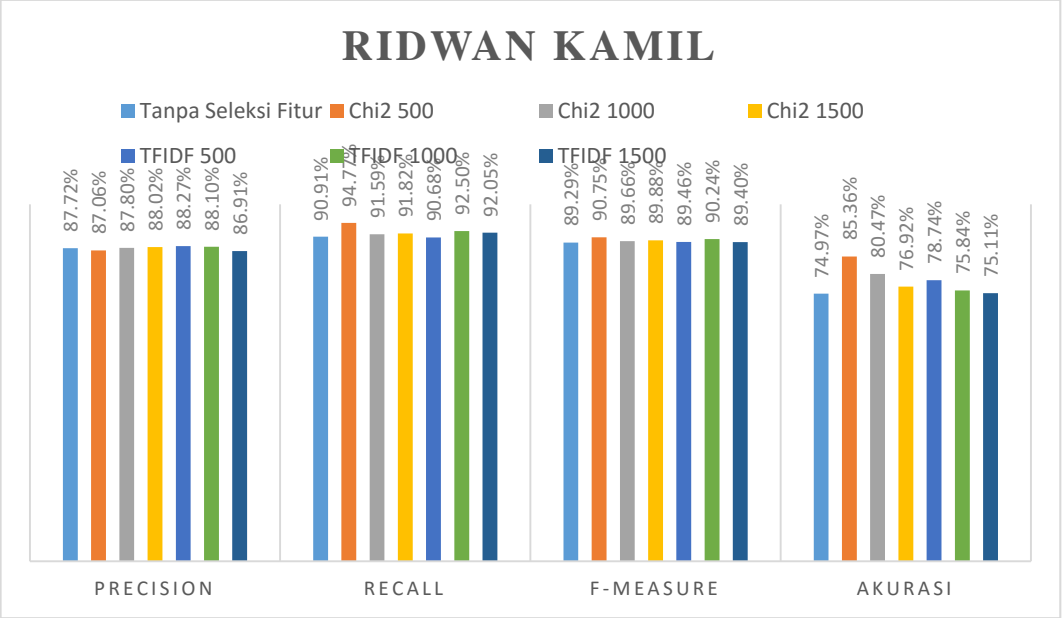
Gambar 6.12 Hasil Performa Model Tokoh Jokowi

Pada gambar 6.12 , variasi jumlah fitur untuk tokoh Jokowi yang digunakan adalah 1000, 2000 dan 3000 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=1000 dimana model dapat mengklasifikasikan 72,91% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=1000 dimana model dapat mengklasifikasikan 78,71% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Prabowo Subianto ditunjukan pada gambar 6.13



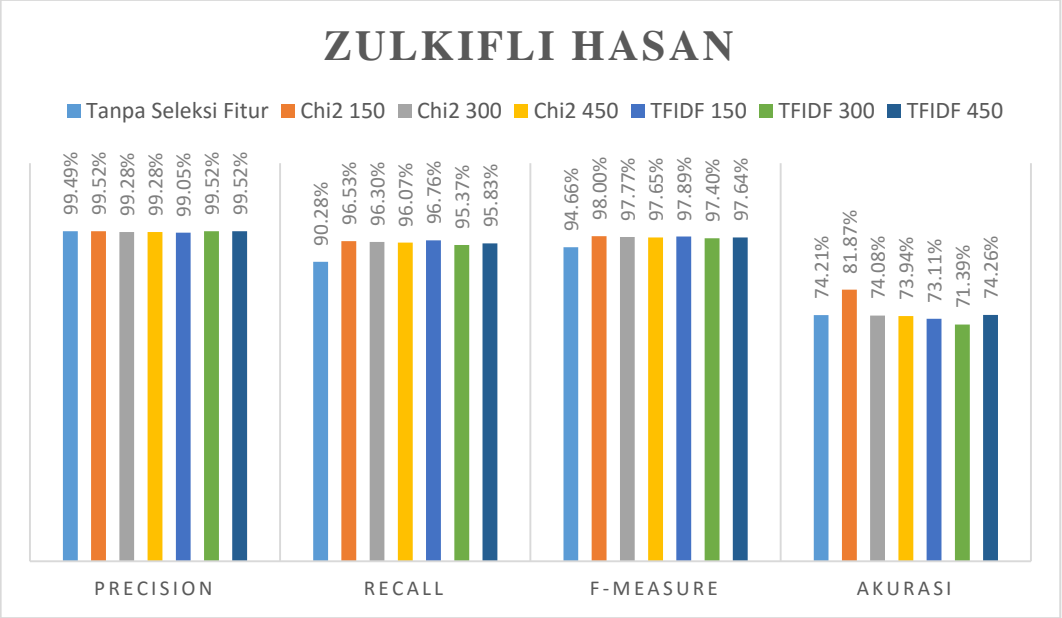
Gambar 6.13 Hasil Performa Model Tokoh Prabowo Subianto

Pada gambr 6.13 , variasi jumlah fitur untuk tokoh Prabowo Subianto yang digunakan adalah 200, 400 dan 600 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=400 dimana model dapat mengklasifikasikan 78,62% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=200 dimana model dapat mengklasifikasikan 85,77% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Ridwan Kamil ditunjukan pada gambar 6.14



Gambar 6.14 Hasil Performa Model Tokoh Ridwan Kamil

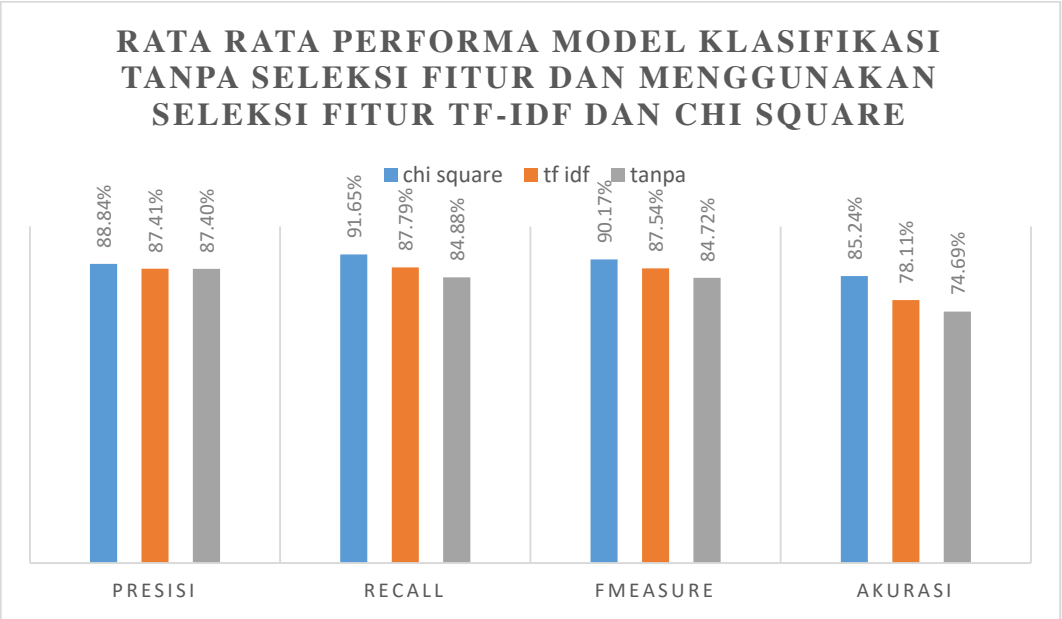
Pada gambar 6.14 , variasi jumlah fitur untuk tokoh Ridwan Kamil yang digunakan adalah 500, 1000 dan 1500 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=500 dimana model dapat mengklasifikasikan 78,74% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=500 dimana model dapat mengklasifikasikan 85,36% dari data *training* secara benar. Selanjutnya, hasil pengujian performa model klasifikasi untuk tokoh Ridwan Kamil ditunjukan pada gambar 6.15



Gambar 6.15 Hasil Performa Model Tokoh Zulkifli Hasan

Pada gambar 6.15 , variasi jumlah fitur untuk tokoh Zulkifli Hasan yang digunakan adalah 150, 300 dan 450 kata fitur.Dari percobaan diatas, ditemukan bahwa jumlah kata fitur hasil proses seleksi fitur TF-IDF yang paling optimal adalah pada jumlah n=450 dimana model dapat mengklasifikasikan 74,26% dari data *training* secara benar. Sementara jumlah kata fitur hasil proses seleksi fitur chi square yang paling optimal adalah pada jumlah n=150 dimana model dapat mengklasifikasikan 81,87% dari data *training* secara benar.

Dari ke 10 tokoh politik tersebut dipilih jumlah kata fitur yang menunjukan model terbaik dari masing masing tokoh kemudian dirata-rata nilai akurasi , presisi, recall, dan fmeasure untuk menunjukan perbandingan hasil model klasifikasi tanpa seleksi fitur, dan menggunakan seleksi fitur TF-IDF dan *chi square*. Hasilnya perbandinganya ditunjukan pada gambar 6.16



Gambar 6.16 Perbandingan performa model klasifikasi tanpa seleksi fitur dan menggunakan seleksi fitur TF-IDF dan chi square

Dari gambar 6.16 dapat dilihat bahwa rata-rata, proses seleksi fitur meningkatkan akurasi dari klasifikasi model jika dibandingkan dengan tanpa menggunakan seleksi fitur yang hanya menghasilkan akurasi model 74,69%. Dari hasil percobaan diatas, dengan menggunakan seleksi fitur *chi square*,model memiliki performa klasifikasi yang lebih baik dari seleksi fitur TF-IDF yaitu 78,11% untuk model yang menggunakan seleksi fitur *chi square* dan 85,24% untuk model yang menggunakan seleksi fitur TF-IDF.

6.6 Hasil Klasifikasi Sentimen Top-n Pada Data Tes

Hasil klasifikasi sentimen untuk data *testing* tokoh agus yudhoyono ditunjukan pada tabel 6.1

Tabel 6.1 Hasil Klasifikasi Sentimen Data Testing Tokoh Agus Yudhoyono

	Agus Yudhoyono		
	Tanpa seleksi fitur	Chi square (n=200)	TF-IDF (n=600)
Pos	221	230	188
Neg	97	88	130

Dari hasil yang didapat, diketahui dari 318 total data testing, klasifikasi tanpa seleksi fitur, 221 data diklasifikasikan sebagai positif dan 97 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=200), 230 data diklasifikasikan sebagai positif dan 88 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 600), 188 data diklasifikasikan sebagai positif dan 130 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Ahok ditunjukan pada tabel 6.2

Tabel 6.2 Hasil Klasifikasi Sentimen Data Testing Tokoh Ahok

	Ahok		
	Tanpa seleksi fitur	Chi square (n=400)	TF-IDF (n=400)
Pos	592	515	513
Neg	89	166	168

Dari hasil yang didapat, diketahui dari 681 total data testing, klasifikasi tanpa seleksi fitur, 592 data diklasifikasikan sebagai positif dan 89 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=400), 515 data diklasifikasikan sebagai positif dan 166 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 400), 513 data diklasifikasikan sebagai positif dan 168 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Anies Baswedan ditunjukan pada tabel 6.3

Tabel 6.3 Hasil Klasifikasi Sentimen Data Testing Tokoh Anies Baswedan

	Anies Baswedan		
	Tanpa seleksi fitur	Chi square (n=500)	TF-IDF (n=500)
Pos	649	701	657
Neg	274	222	266

Dari hasil yang didapat, diketahui dari 923 total data testing, klasifikasi tanpa seleksi fitur, 649 data diklasifikasikan sebagai positif dan 274 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=500), 701 data diklasifikasikan sebagai positif dan 222 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 500), 657 data diklasifikasikan sebagai positif dan 266 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Gatot Nurmantyo ditunjukan pada tabel 6.4

Tabel 6.4 Hasil Klasifikasi Sentimen Data Testing Tokoh Gatot Nurmantyo

	Gatot Nurmantyo		
	Tanpa seleksi fitur	Chi square (n=400)	TF-IDF (n=600)
Pos	284	207	208
Neg	471	548	547

Dari hasil yang didapat, diketahui dari 755 total data testing , Untuk klasifikasi tanpa seleksi fitur , 284 data diklasifikasikan sebagai positif dan 471 data diklasifikasikan sebagai negatif. Untuk klasifikasi menggunakan seleksi fitur *chi square* (n=400) , 207 data diklasifikasikan sebagai positif dan 548 data diklasifikasikan sebagai negatif. Untuk klasifikasi menggunakan seleksi fitur TF-IDF (n = 600) , 208 Data diklasifikasikan sebagai positif dan 547 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Hary Tanoe ditunjukan pada tabel 6.5

Tabel 6.5 Hasil Klasifikasi Sentimen Data Testing Tokoh Hary Tanoe

	Hary Tanoe		
	Tanpa seleksi fitur	Chi square (n=400)	TF-IDF (n=400)
Pos	282	330	332
Neg	103	55	53

Dari hasil yang didapat, diketahui dari 385 total data testing, klasifikasi tanpa seleksi fitur, 282 data diklasifikasikan sebagai positif dan 103 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=400), 330 data diklasifikasikan sebagai positif dan 55 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 400), 332 data diklasifikasikan sebagai positif dan 53 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Jusuf Kalla ditunjukan pada tabel 6.6

Tabel 6.6 Hasil Klasifikasi Sentimen Data Testing Tokoh Jusuf Kalla

	Jusuf Kalla		
	Tanpa seleksi fitur	Chi square (n=350)	TF-IDF (n=1050)
Pos	142	327	311
Neg	311	126	142

Dari hasil yang didapat, diketahui dari 453 total data testing, klasifikasi tanpa seleksi fitur, 142 data diklasifikasikan sebagai positif dan 311 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=350), 327 data diklasifikasikan sebagai positif dan 126 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 350), 311 data diklasifikasikan sebagai positif dan 142 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Jokowi ditunjukan pada tabel 6.7

Tabel 6.7 Hasil Klasifikasi Sentimen Data Testing Tokoh Jokowi

	Jokowi		
	Tanpa seleksi fitur	Chi square (n=1000)	TF-IDF (n=1000)
Pos	1148	1164	1116
Neg	272	256	304

Dari hasil yang didapat, diketahui dari 1420 total data testing, klasifikasi tanpa seleksi fitur, 1148 data diklasifikasikan sebagai positif dan 272 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=1000), 1164 data diklasifikasikan sebagai positif dan 256 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 1000), 1116 data diklasifikasikan sebagai positif dan 304 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Prabowo Subianto ditunjukan pada tabel 6.8

Tabel 6.8 Hasil Klasifikasi Sentimen Data Testing Tokoh Prabowo Subianto

	Prabowo Subianto		
	Tanpa seleksi fitur	Chi square (n=200)	TF-IDF (n=400)
Pos	100	217	210
Neg	425	308	315

Dari hasil yang didapat, diketahui dari 525 total data testing, klasifikasi tanpa seleksi fitur, 100 data diklasifikasikan sebagai positif dan 425 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=200), 217 data diklasifikasikan sebagai positif dan 308 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 400), 210 data diklasifikasikan sebagai positif dan 315 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Ridwan Kamil ditunjukan pada tabel 6.9

Tabel 6.9 Hasil Klasifikasi Sentimen Data Testing Tokoh Ridwan Kamil

	Ridwan Kamil		
	Tanpa seleksi fitur	Chi square (n=500)	TF-IDF (n=500)
Pos	705	671	679
Neg	128	162	154

Dari hasil yang didapat, diketahui dari 833 total data testing, klasifikasi tanpa seleksi fitur, 705 data diklasifikasikan sebagai positif dan 128 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=500), 671 data diklasifikasikan sebagai positif dan 162 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 1000), 679 data diklasifikasikan sebagai positif dan 154 data diklasifikasikan sebagai negatif.

Hasil klasifikasi sentimen untuk data *testing* tokoh Zulkifli Hasan ditunjukan pada tabel 6.10

Tabel 6.10 Hasil Klasifikasi Sentimen Data Testing Tokoh Zulkifli Hasan

	Zulkifli Hasan		
	Tanpa seleksi fitur	Chi square (n=150)	TF-IDF (n=450)
Pos	381	549	535
Neg	290	122	136

Dari hasil yang didapat, diketahui dari 671 total data testing, klasifikasi tanpa seleksi fitur, 381 data diklasifikasikan sebagai positif dan 290 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur *chi square* (n=150), 549 data diklasifikasikan sebagai positif dan 122 data diklasifikasikan sebagai negatif. Klasifikasi menggunakan seleksi fitur TF-IDF (n = 150), 535 data diklasifikasikan sebagai positif dan 136 data diklasifikasikan sebagai negatif.

6.7 Hasil Perhitungan Elektabilitas

Setelah model klasifikasi sentimen diterapkan pada data *testing* tiap tokoh politik, selanjutnya dilakukan perhitungan nilai elektabilitas tiap tokoh politik menggunakan rumus *positive versus total* dan *share of volume* yang sudah dijelaskan pada sub bab 3.10 dan 3.11. Hasil perhitungan elektabilitas tokoh politik menggunakan *positive versus total* untuk hasil klasifikasi pada data testing tanpa seleksi fitur dapat dilihat pada tabel 6.11

Tabel 6.11 Hasil Elektabilitas PvT Tokoh Politik Tanpa Seleksi Fitur

Tanpa Seleksi Fitur		
Tokoh Politik	Hasil normalisasi (%)	Hasil PvT (%)
Basuki Tjahaja Purnama	14.24494	86.93098
Anies Baswedan	11.52203	70.31419
Prabowo Subianto	3.121237	19.04762
Zulkifli Hasan	9.304402	56.78092
Agus Yudhoyono	11.3881	69.49686
Gatot Nurmantyo	6.163926	37.61589
Jusuf Kalla	5.136605	31.34658
Hary Tanoe	12.00257	73.24675
Rdiwan Kamil	13.86852	84.63385
Joko Widodo	13.24767	80.84507

Pada tabel 6.11, hasil PvT tertinggi diperoleh tokoh Basuki Tjahaja Purnama dengan hasil PvT sebesar 86.93%, sementara hasil PvT terendah diperoleh tokoh Prabowo Subianto dengan elektabilitas PvT sebesar 19.05%. Hasil normalisasi PvT tertinggi diperoleh tokoh Basuki Tjahaja Purnama dengan hasil normalisasi sebesar 14.24% dan hasil normalisasi PvT terendah diperoleh tokoh Prabowo Subianto dengan hasil normalisasi PvT sebesar 3.12%

Hasil perhitungan elektabilitas tokoh politik menggunakan *positive versus total* untuk hasil klasifikasi pada data *testing* dengan seleksi fitur *chi square* dapat dilihat pada tabel 6.12

Tabel 6.12 Hasil Elektabilitas Pvt Tokoh Politik Dengan Seleksi Fitur chi square

Seleksi Fitur Chi Square		
Tokoh Politik	Hasil normalisasi (%)	Hasil PvT (%)
Basuki Tjahaja Purnama	10.88286	75.62408
Anies Baswedan	10.92947	75.948
Prabowo Subianto	5.94817	41.33333
Zulkifli Hasan	11.77424	81.81818
Agus Yudhoyono	10.40839	72.32704
Gatot Nurmantyo	3.945539	27.41722
Jusuf Kalla	10.38801	72.18543
Hary Tanoe	12.33491	85.71429
Rdiwan Kamil	11.59206	80.55222
Joko Widodo	11.79635	81.97183

Pada tabel 6.12, hasil PvT tertinggi diperoleh tokoh Hary Tanoe dengan hasil PvT sebesar 85.71%, sementara hasil PvT terendah diperoleh tokoh Gatot Nurmantyo dengan elektabilitas PvT sebesar 27.42%. Hasil normalisasi PvT tertinggi diperoleh tokoh Hary Tanoe dengan hasil normalisasi sebesar 12.33% dan hasil normalisasi PvT terendah diperoleh tokoh Gatot Nurmantyo dengan hasil normalisasi PvT sebesar 3.94%

Hasil perhitungan elektabilitas tokoh politik menggunakan *positive versus total* untuk hasil klasifikasi pada data *testing* dengan seleksi fitur TF-IDF dapat dilihat pada tabel 6.13

Tabel 6.13 Hasil Elektabilitas PvT Tokoh Politik Dengan Seleksi Fitur TF-IDF

Seleksi Fitur TF-IDF		
Tokoh Politik	Hasil normalisasi (%)	Hasil PvT (%)
Basuki Tjahaja Purnama	11.27863	75.3304
Anies Baswedan	10.65737	71.18093
Prabowo Subianto	5.988888	40
Zulkifli Hasan	11.93761	79.73174
Agus Yudhoyono	8.851502	59.1195
Gatot Nurmantyo	4.124797	27.54967
Jusuf Kalla	10.27894	68.65342
Hary Tanoe	12.91111	86.23377
Rdiwan Kamil	12.20425	81.51261
Joko Widodo	11.7669	78.59155

Pada tabel 6.13, hasil PvT tertinggi diperoleh tokoh Hary Tanoe dengan hasil PvT sebesar 86.23%, sementara hasil PvT terendah diperoleh tokoh Gatot Nurmantyo dengan elektabilitas PvT sebesar 27.55%. Hasil normalisasi PvT tertinggi diperoleh tokoh Hary Tanoe dengan hasil normalisasi sebesar 12.91% dan hasil normalisasi PvT terendah diperoleh tokoh Gatot Nurmantyo dengan hasil normalisasi PvT sebesar 4.12%

Hasil perhitungan rata-rata elektabilitas tokoh politik menggunakan *positive versus total* untuk hasil klasifikasi pada data *testing* tanpa seleksi fitur, dengan seleksi fitur TF-IDF dan *chi square* dapat dilihat pada tabel 6.14

Tabel 6.14 Hasil Rata-Rata Elektabilitas PvT Tokoh Politik

Rata-Rata Elektabilitas		
Tokoh Politik	Hasil normalisasi (%)	Hasil PvT (%)
Basuki Tjahaja Purnama	12.13548	79.29515
Anies Baswedan	11.03629	72.48104
Prabowo Subianto	5.019432	33.46032
Zulkifli Hasan	11.00542	72.77695
Agus Yudhoyono	10.216	66.98113
Gatot Nurmantyo	4.744754	30.86093
Jusuf Kalla	8.601186	57.39514
Hary Tanoe	12.4162	81.7316
Rdiwan Kamil	12.55494	82.23289
Joko Widodo	12.27031	80.46948

Pada tabel 6.14, berdasarkan rata-rata elektabilitas tanpa seleksi fitur, dengan seleksi fitur TF-IDF dan seleksi fitur *chi square*, hasil PvT tertinggi diperoleh tokoh Ridwan Kamil dengan hasil PvT sebesar 82.23%, sementara hasil PvT terendah diperoleh tokoh Gatot Nurmantyo dengan elektabilitas PvT sebesar 30.86%. Hasil normalisasi PvT tertinggi diperoleh tokoh Ridwan Kamil dengan hasil normalisasi sebesar 12.55% dan hasil normalisasi PvT terendah diperoleh tokoh Gatot Nurmantyo dengan hasil normalisasi PvT sebesar 4.74%

Hasil perhitungan elektabilitas tokoh politik menggunakan *share of volume* untuk hasil klasifikasi pada data testing tanpa seleksi fitur dapat dilihat pada tabel 6.15

Tabel 6.15 Hasil Elektabilitas SoV Tokoh Politik Tanpa Seleksi Fitur

Tokoh Politik	Hasil SoV (%)
Basuki Tjahaja Purnama	13.14387
Anies Baswedan	14.40941
Prabowo Subianto	2.220249
Zulkifli Hasan	8.459147
Agus Yudhoyono	4.90675
Gatot Nurmantyo	6.305506
Jusuf Kalla	3.152753
Hary Tanoe	6.261101
Rdiwan Kamil	15.65275
Joko Widodo	25.48845

Pada tabel 6.15, hasil SoV tertinggi diperoleh tokoh Joko Widodo dengan hasil SoV sebesar 25.49% dan hasil SoV terendah diperoleh tokoh Prabowo Subianto dengan hasil SoV sebesar 2.22%

Hasil perhitungan elektabilitas tokoh politik menggunakan *share of volume* untuk hasil klasifikasi pada data testing menggunakan seleksi fitur chi square dapat dilihat pada tabel 6.16

Tabel 6.16 Hasil Elektabilitas SoV Tokoh Politik Dengan Seleksi Fitur TF-IDF

Tokoh Politik	Hasil SoV (%)
Basuki Tjahaja Purnama	10.80227
Anies Baswedan	13.83449
Prabowo Subianto	4.421984
Zulkifli Hasan	11.26553
Agus Yudhoyono	3.958728
Gatot Nurmantyo	4.379869
Jusuf Kalla	6.548747
Hary Tanoe	6.990945
Rdiwan Kamil	14.29775
Joko Widodo	23.49968

Pada tabel 6.16, hasil SoV tertinggi diperoleh tokoh Joko Widodo dengan hasil SoV sebesar 23.50% dan hasil SoV terendah diperoleh tokoh Agus Yudhoyono dengan hasil SoV sebesar 3.96%

Hasil perhitungan elektabilitas tokoh politik menggunakan *share of volume* untuk hasil klasifikasi pada data testing menggunakan seleksi fitur TF-IDF dapat dilihat pada tabel 6.17

Tabel 6.17 Hasil Elektabilitas SoV Tokoh Politik Dengan Seleksi Fitur chi square

Tokoh Politik	Hasil SoV (%)
Basuki Tjahaja Purnama	10.48666
Anies Baswedan	14.27408
Prabowo Subianto	4.418652
Zulkifli Hasan	11.17899
Agus Yudhoyono	4.683364
Gatot Nurmantyo	4.215027
Jusuf Kalla	6.658522
Hary Tanoe	6.719609
Rdiwan Kamil	13.66321
Joko Widodo	23.70189

Pada tabel 6.17, hasil SoV tertinggi diperoleh tokoh Joko Widodo dengan hasil SoV sebesar 23.70% dan hasil SoV terendah diperoleh tokoh Gatot Nurmantyo dengan hasil SoV sebesar 4.21%

Hasil perhitungan rata-rata elektabilitas tokoh politik menggunakan *share of volume* untuk hasil klasifikasi pada data *testing* tanpa seleksi fitur, dengan seleksi fitur TF-IDF dan *chi square* dapat dilihat pada tabel 6.18

Tabel 6.18 Hasil Rata-Rata Elektabilitas SoV Tokoh Politik

Tokoh Politik	Hasil SoV (%)
Basuki Tjahaja Purnama	11.4776
Anies Baswedan	14.17266
Prabowo Subianto	3.686961
Zulkifli Hasan	10.30122
Agus Yudhoyono	4.516281
Gatot Nurmantyo	4.966801
Jusuf Kalla	5.453341
Hary Tanoe	6.657219
Rdiwan Kamil	14.5379
Joko Widodo	24.23001

Pada tabel 6.18, berdasarkan rata-rata elektabilitas SoV, hasil SoV tertinggi diperoleh tokoh Joko Widodo dengan hasil SoV sebesar 24.23% dan hasil SoV terendah diperoleh tokoh Prabowo Subianto dengan hasil SoV sebesar 3.68%

BAB VII
SARAN DAN KESIMPULAN

7.1 Kesimpulan

Berdasarkan hasil pengamatan, pengujian dan analisis pada hasil yang diperoleh, kesimpulan yang dapat diambil adalah sebagai berikut:

1. Analisis sentimen untuk mengetahui elektabilitas tokoh politik menggunakan metode Multinomial Naïve Bayes telah berhasil dijalankan.
2. Metode seleksi fitur terbukti mampu meningkatkan akurasi model klasifikasi sentimen, yaitu sebesar 10,55% untuk chi square dan 3,42% untuk TF-IDF
3. Metode seleksi fitur *chi square* memiliki performa klasifikasi yang lebih baik dengan akurasi 85,24% ,presisi 88,84% ,*recall* 91,65% ,*fmeasure* 90,17% dibandingkan dengan metode seleksi fitur TF-IDF dengan akurasi 78,11% , presisi 87,41%, *recall* 87,79% , *fmeasure* 87,54% serta dibandingkan dengan metode tanpa seleksi fitur dengan akurasi 74,69%, presisi 87,40%, *recall* 84,88%, fmeasure 84,72%
4. Berdasarkan rata-rata nilai elektabilitas dengan rumus *positive versus total*, elektabilitas tertinggi diperoleh oleh tokoh Ridwan Kamil dengan elektabilitas 82.23% dan elektabilitas terendah diperoleh oleh Gatot Nurmantyo dengan elektabilitas 30.86%
5. Berdasarkan rata-rata nilai elektabilitas dengan rumus *share of volume*, elektabilitas tertinggi diperoleh oleh tokoh Jokowi dengan elektabilitas 24,23% dan elektabilitas terendah diperoleh oleh Prabowo Subianto dengan elektabilitas 3,68%

7.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut :

1. Membandingkan performa klasifikasi dengan metode seleksi fitur yang lain seperti Information Gain, Mutual Information, dan lain sebagainya dengan harapan mendapatkan performa klasifikasi yang lebih baik
2. Membandingkan dengan metode klasifikasi yang lain seperti SVM , KNN dan lain sebagainya untuk mendapatkan perbandingan nilai performansi
3. Menggunakan rumus yang lain untuk menghitung nilai elektabilitas tokoh politik selain *positive versus total* dan *share of volume*
4. Menghitung variabel lain seperti tingkat popularitas tokoh politik ,partai, kemenangan tokoh politik dalam pemilu sebelumnya untuk mempertimbangkan elektabilitas tokoh politik.
5. Kata kunci pencarian untuk masing-masing tokoh politik menggunakan variasi jumlah yang sama
6. Menggunakan teknik untuk mengatasi ketidak seimbangan jumlah label pada data latih

DAFTAR PUSTAKA

Abramowitz, A. I. (1989). Viability, Electability, and Candidate Choice in a Presidential Primary Election : A Test of Competing Models. *The Journal of Politics*, 977-992.

Adiwijawa, I. (2006). *Text Mining dan Knowledge Discovery*. EMC Coporation.

Anggara, N., Romadhony, A., & Suliiyo, M. D. (2013). *Implementasi Modifikasi Algoritma Enchanced Confix Stripping Stemmer pada Teks Bahasa Indonesia*. Bandung: Telkom University.

Bermingham, A., & Smeaton, A. F. (2011). On Using Twitter to Monitor Political Sentiment. *In Proceeding of IJCNLP conference*. Chiang Mai, Thailand.

Bermingham, A., & Smeaton, A. F. (2011). On Using Twitter to Monitor Political Sentiment and Predict Election Result. *Sentiment Analysis where AI meets Psychology (SAAIP) Workshop at the International Joint Conference for Natural Language Processing (IJCNLP)*. Chiang Mai, Thailand.

Fathan, A., & SN, A. (2014). Analisis Sentimen dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter. *Seminar Nasional Informatika*, 115-122.

Fawcett, T. (2005). *An Introduction to ROC Analysis*. California: Elsevier.

Goyvaerts, J. (2007). *Regular Expressions : The Complete Tutorial*.

Hamad, I. (2004). *Konstruksi Realitas Politik dalam Media Massa : Sebuah Studi Critical Discourse Analysis terhadap berita-berita Politik*. Jakarta: Granit.

Hayatin, N., Mentari, M., & Izzah, A. (2014). Opinion Extraction of Public Figure Based on Sentiment Analysis in Twitter. *Journal of Engineering*, 9-14.

Hermawan, A. (2016). *Framing The 2014 Indonesian Presidential Candidates in Newspapers and on Twitter*. Arizona: The University of Arizona.

Juditha, C. (2013). Akurasi Berita dalam Jurnalisme Online (Kasus Dugaan Korupsi Mahkamah Konstitusi di Portal Berita Detiknews). *Jurnal Pekommas, Vol. 16 No. 3*, 145-154.

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a Social Network or a News Media? *Proceedings of the 19th international conference on World wide web* (pp. 591-600). Raleigh, North Carolina, USA: ACM.

Lestari, A. R., Perdana, R. S., & Fauzi, M. A. (2017). Analisis Sentimen Tentang Opini Pilkada DKI 2017 Pada Dokumen twitter Berbahasa Indonesia

Menggunakan Naive Bayes dan Pembobotan Emoji. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1718-1724.

Lestari, A. R., Perdana, R. S., & Fauzi, M. A. (2017). Analisis Sentimen Tentang Opini Pilkada DKI 2017 Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes dan Pembobotan Emoji. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1718-1724.

Ling, J., Kencana, I. P., & Oka, T. B. (2014). Analisis Sentimen Menggunakan Metode Naive Bayes Classifier dengan Seleksi Fitur Chi Square. *E-Jurnal Matematika*, 92-99.

Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*. London: Cambridge University Press.

McCallum, A., & Nigam, K. (1998). A comparison of event Models for Naive Bayes Text Classification. *Proceedings in Workshop on Learning for Text Categorization* (pp. 41-48). AAAI'98.

Nazief, B., Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S., & Williams, H. E. (2007). Stemming Indonesian : A confix-stripping approach. *Journal ACM Transactions on Asian Language Informations Processing (TALIP)*, 1-33.

Pang, B., & Lee, L. (2008). *Opinion Mining and Sentiment Analysis*. Netherlands: now Publishers Inc.

Ramteke, J., Shah, S., Godhia, D., & Shaikh, A. (2016). Election Result Prediction Using Twitter sentiment Analysis. *Inventive Computation Technologies International Conference*. Coimbatore, India: IEEE.

Ramteke, J., Shah, S., Godhia, D., & Shaikh, A. (2016). Election Result Prediction Using Twitter Sentiment Analysis. 1-5.

Rianto, B. (2016). *Implementasi dan Perbandingan Metod Prapemrosesan pada Analisis Sentimen Gubernur DKI Jakarta Menggunakan Metode Support Vector Machine dan Naive Bayes*. Yogyakarta: Universitas Gadjah Mada.

Rossellini, R. G. (2012). *Perbandingan Metode Pembobotan Term Menggunakan Term Frequency Chi Square Dan Term Frequency Inverse Documen Frequency Pada Text Mining*. Bandung: Telkom University.

Siddiqi, S., & Sharan, A. (2015). Keyword and Keyphrase Extraction Techniques : A Literature Review. *International Journal of Computer Applications*, 19-23.

Sukendar, M. U. (2017). Pemilihan Presiden, Media Sosial dan Pendidikan Politik. *Jurnal IKON Prodi D3 Komunikasi Massa*, 74-79.

Tahitoe, A. D., & Purwitasari, D. (2010). *Implementasi Modifikasi Enchanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming*. Surabaya: Institut Teknologi Sepuluh Nopember (ITS).

Trisedya, B. D. (2009). *Pemanfaatan Dokumen Unlabeled pada Klasifikasi Topik Berbasis Naïve Bayes dengan Algoritma Expectation Maximization*. Depok: Universitas Indonesia.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting Elections with Twitter:. *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media* (pp. 178-183). Jerman: Technische Universität München.

Vijayani, D. S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks* , 7-16.

Virgo, F. G. (2018). *Analisis Sentimen dan Deteksi Buzzer di Twitter dalam Prediksi Pilkada DKI Jakarta 2017*. Yogyakarta: Perpustakaan FMIPA UGM.

Wikarsa, L., & Thair, S. N. (2016). A text mining application of emotion classifications of Twitter's users using Naïve Bayes method. *Wireless and Telematics (ICWT), 2015 1st International Conference*. Manado: IEEE.

Zhao, J., Shah, A., & Oshershon, D. (2009). On the provenance of judgments of conditional probability. *COGNITION*, 26-36.

Zhao, L., Huang, M., Yao, Z., Su, R., Jiang, Y., & Zhu, X. (2016). Semi-Supervised Multinomial Naive Bayes for Text Classification. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (pp. 2877-2883). Beijing, China: AAAI.

LAMPIRAN

A. *List Stopword*

Stopwords			
bermula	ada	diperlihatkan	kan
bersama	adalah	diperlukan	kapan
bersama-sama	adanya	diperlukannya	kapankah
bersiap	adapun	dipersoalkan	kapanpun
bersiap-siap	agak	dipertanyakan	karena
bertanya	agaknya	dipunyai	karenanya
bertanya-tanya	agar	diri	kasus
berturut	akan	dirinya	kata
berturut-turut	akankah	disampaikan	katakan
bertutur	akhir	disebut	katakanlah
berujar	akhiri	disebutkan	katanya
berupa	akhirnya	disebutkannya	ke
besar	aku	disini	keadaan
betul	akulah	disinilah	kebetulan
betulkah	amat	ditambahkan	kecil
biasa	amatlah	ditandaskan	kedua
biasanya	anda	ditanya	keduanya
bila	andalah	ditanyai	keinginan
bilakah	antar	ditanyakan	kelamaan
bisa	antara	ditegaskan	kelihatan
bisakah	antaranya	ditujukan	kelihatannya
boleh	apa	ditunjuk	kelima
bolehkah	apaan	ditunjuki	keluar
bolehlah	apabila	ditunjukkan	kembali
buat	apakah	ditunjukkannya	kemudian
bukan	apalagi	ditunjuknya	kemungkinan
bukankah	apatah	dituturkan	kemungkinannya
bukanlah	artinya	dituturkannya	kenapa
bukannya	asal	diucapkan	kepada
bulan	asalkan	diucapkannya	kepadanya
bung	atas	diungkapkan	kesampaian
cara	atau	dong	keseluruhan
caranya	ataukah	dua	keseluruhannya
cukup	ataupun	dulu	keterlaluan
cukupkah	awal	empat	ketika
cukuplah	awalnya	enggak	khususnya
cuma	bagai	enggaknya	kini
dahulu	bagaikan	entah	kinilah
dalam	bagaimana	entahlah	kira
dan	bagaimanakah	guna	kira-kira

dapat	bagaimanapun	gunakan	kiranya
dari	bagi	hal	kita
daripada	bagian	hampir	kitalah
datang	bahkan	hanya	kok
dekat	bahwa	hanyalah	kurang
demi	bahwasanya	hari	lagi
demikian	baik	harus	lagian
demikianlah	bakal	haruslah	lah
dengan	bakalan	harusnya	lain
depan	balik	hendak	lainnya
di	banyak	hendaklah	lalu
dia	bapak	hendaknya	lama
diakhiri	baru	hingga	lamanya
diakhirinya	bawah	ia	lanjut
dialah	beberapa	ialah	lanjutnya
diantara	begini	ibarat	lebih
diantaranya	beginian	ibaratkan	lewat
diberi	beginikah	ibaratnya	lima
diberikan	beginilah	ibu	luar
diberikannya	begitu	ikut	macam
dibuat	begitukah	ingat	maka
dibuatnya	begitulah	ingat-ingat	makanya
didapat	begitupun	ingin	makin
didatangkan	bekerja	inginkah	malah
digunakan	belakang	inginkan	malahan
diibaratkan	belakangan	ini	mampu
diibaratkannya	belum	inikah	mampukah
diingat	belumlah	inilah	mana
diingatkan	benar	itu	manakala
diinginkan	benarkah	itukah	manalagi
dijawab	benarlah	itulah	masa
dijelaskan	berada	jadi	masalah
dijelaskannya	berakhir	jadilah	masalahnya
dikarenakan	berakhirlah	jadinya	masih
dikatakan	berakhirnya	jangan	masihkah
dikatakannya	berapa	jangankan	masing
dikerjakan	berapakah	janganlah	masing-masing
diketahui	berapalah	jauh	mau
diketahuinya	berapapun	jawab	maupun
dikira	berarti	jawaban	melainkan
dilakukan	berawal	jawabnya	melakukan
dilalui	berbagai	jelas	melalui
dilihat	berdatangan	jelaskan	melihat
dimaksud	beri	jelaslah	melihatnya
dimaksudkan	berikan	jelasnya	memang

dimaksudkannya	berikut	jika	memastikan
dimaksudnya	berikutnya	jikalau	memberi
diminta	berjumlah	juga	memberikan
dimintai	berkali-kali	jumlah	membuat
dimisalkan	berkata	jumlahnya	memerlukan
dimulai	berkehendak	justru	memihak
dimulailah	berkeinginan	kala	meminta
dimulainya	berkenaan	kalau	memintakan
dimungkinkan	berlainan	kalaulah	memisalkan
dini	berlalu	kalaupun	memperbuat
dipastikan	berlangsung	kalian	mempergunakan
diperbuat	berlebihan	kami	memperkirakan
diperbuatnya	bermacam	kamilah	memperlihatkan
dipergunakan	bermacam-macam	kamu	mempersiapkan
mempertanyakan	pertama	semacam	ungkapnya
mempunyai	pertama-tama	semakin	untuk
memulai	pertanyaan	semampu	usah
memungkinkan	pertanyakan	semampunya	usai
menaiki	pihak	semasa	waduh
menambahkan	pihaknya	semasih	wah
menandaskan	pukul	semata	wahai
menanti	pula	semata-mata	waktu
menanti-nanti	pun	semaunya	waktunya
menantikan	punya	sementara	walau
menanya	rasa	semisal	walaupun
menanyai	rasanya	semisalnya	wong
menanyakan	rata	sempat	yaitu
mendapat	rupanya	semua	yakin
mendapatkan	saat	semuanya	yakni
mendatang	saatnya	semula	yang
mendatangi	saja	sendiri	siapapun
mendatangkan	sajalah	sendirian	sini
menegaskan	saling	sendirinya	sinilah
mengakhiri	sama	seolah	soal
mengapa	sama-sama	seolah-olah	soalnya
mengatakan	sambil	seorang	suatu
mengatakannya	sampai	sepanjang	sudah
mengenai	sampai-sampai	sepantasnya	sudahkah
mengerjakan	sampaikan	sepantasnyalah	sudahlah
mengetahui	sana	seperlunya	supaya
menggunakan	sangat	seperti	tadi
menghendaki	sangatlah	sepertinya	tadinya
mengibaratkan	satu	sepihak	tahu

mengibaratkannya	saya	sering	tahun
mengingat	sayalah	seringnya	tak
mengingatkan	se	serta	tambah
menginginkan	sebab	serupa	tambahnya
mengira	sebabnya	sesaat	tampak
mengucapkan	sebagai	sesama	tampaknya
mengucapkannya	sebagaimana	sesampai	tandas
mengungkapkan	sebagainya	sesegera	tandasnya
menjadi	sebagian	sesekali	tanpa
menjawab	sebaik	seseorang	tanya
menjelaskan	sebaik-baiknya	sesuatu	tanyakan
menuju	sebaiknya	sesuatunya	tanyanya
menunjuk	sebaliknya	sesudah	tapi
menunjuki	sebanyak	sesudahnya	tegas
menunjukkan	sebegini	setelah	tegasnya
menunjuknya	sebegitu	setempat	telah
menurut	sebelum	setengah	tempat
menuturkan	sebelumnya	seterusnya	tengah
menyampaikan	sebenarnya	setiap	tentang
menyangkut	seberapa	setiba	tentu
menyatakan	sebesar	setibanya	tentulah
menyebutkan	sebetulnya	setidak-tidaknya	tentunya
menyeluruh	sebisanya	setidaknya	tepat
menyiapkan	sebuah	setinggi	terakhir
merasa	sebut	seusai	terasa
mereka	sebutlah	sewaktu	terbanyak
merekalah	sebutnya	siap	terdahulu
merupakan	secara	siapa	terdapat
meski	secukupnya	siapakah	terdiri
meskipun	sedang	terhadap	
meyakini	sedangkan	terhadapnya	
meyakinkan	sedemikian	teringat	
minta	sedikit	teringat-ingat	
mirip	sedikitnya	terjadi	
misal	seenaknya	terjadilah	
misalkan	segala	terjadinya	
misalnya	segalanya	terkira	
mula	segera	terlalu	
mulai	seharusnya	terlebih	
mulailah	sehingga	terlihat	
mulanya	seingat	termasuk	
mungkin	sejak	ternyata	
mungkinkah	sejauh	tersampaikan	

nah	sejenak	tersebut	
naik	sejumlah	tersebutlah	
namun	sekadar	tertentu	
nanti	sekadarnya	tertuju	
nantinya	sekali	terus	
nyaris	sekali-kali	terutama	
nyatanya	sekalian	tetap	
oleh	sekaligus	tetapi	
olehnya	sekalipun	tiap	
pada	sekarang	tiba	
padahal	sekecil	tiba-tiba	
padanya	seketika	tidak	
pak	sekiranya	tidakkah	
paling	sekitar	tidaklah	
panjang	sekitarnya	tiga	
pantas	sekurang-kurangnya	tinggi	
para	sekurangnya	toh	
pasti	sela	tunjuk	
pastilah	selagi	turut	
penting	selain	tutur	
pentingnya	selaku	tuturnya	
per	selalu	ucap	
percuma	selama	ucapnya	
perlu	selama-lamanya	ujar	
perlukah	selamanya	ujarnya	
perlunya	selanjutnya	umum	
pernah	seluruh	umumnya	
persoalan	seluruhnya	ungkap	