



TESIS – EE185401

***SENTIMENT ANALYSIS PERILAKU PENGGUNA
MEDIA SOSIAL TERHADAP UJIAN NASIONAL
MENGUNAKAN K-MEANS DAN SUPPORT VECTOR
MACHINE (SVM)***

**CHANDRA EKO WAHYUDI UTOMO
07111550060006**

**DOSEN PEMBIMBING
Mochamad Hariadi, S.T., M.Sc., Ph.D
Dr. Surya Sumpeno, S.T., M.Sc**

**PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMASI CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020**



TESIS – EE185401

***SENTIMENT ANALYSIS PERILAKU PENGGUNA
MEDIA SOSIAL TERHADAP UJIAN NASIONAL
MENGUNAKAN K-MEANS DAN SUPPORT VECTOR
MACHINE (SVM)***

CHANDRA EKO WAHYUDI UTOMO

07111550060006

DOSEN PEMBIMBING

Mochamad Hariadi, S.T., M.Sc., Ph.D

Dr. Surya Sumpeno, S.T., M.Sc

PROGRAM MAGISTER

BIDANG KEAHLIAN TELEMATIKA

DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMASI CERDAS

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2020

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh:

CHANDRA EKO WAHYUDI UTOMO
NRP. 07111550060006

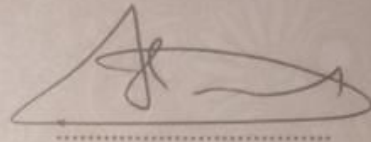
Tanggal Ujian: 15 Januari 2020

Periode Wisuda: Maret 2020

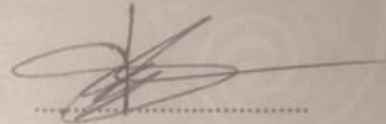
Disetujui oleh:

Pembimbing:

1. Mochamad Hariadi, S.T., M.Sc., Ph.D
NIP: 19691209 1997031002

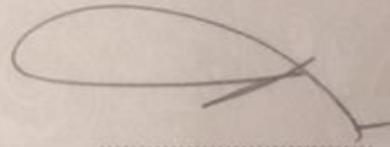


2. Dr. Surya Sumpeno, S.T., M.Sc.
NIP: 19690613 1997021003

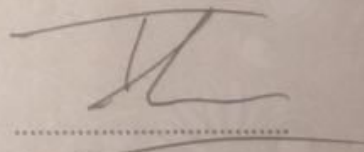


Penguji:

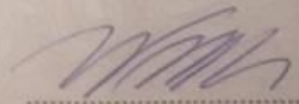
1. Dr. Ir. Endroyono, DEA.
NIP: 196504041991021001



2. Dr. Ista Pratomo, S.T., M.T.
NIP: 197903252003121001



3. Dr. Ir. Wirawan, DEA.
NIP: 196311091989031011



Kepala Departemen Teknik Elektro Fakultas
Teknologi Elektro dan Informasi Cerdas

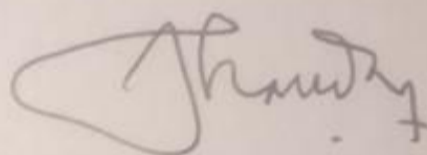
Dedet Candra Riawan, ST., M.Eng., Ph.D
NIP: 197311192000031001

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "***SENTIMENT ANALYSIS PERILAKU PENGGUNA MEDIA SOSIAL TERHADAP UJIAN NASIONAL MENGGUNAKAN K-MEANS DAN SVM***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2020



Chandra Eko Wahyudi Utomo
NRP. 07111550060006

SENTIMENT ANALYSIS PERILAKU PENGGUNA MEDIA SOSIAL TERHADAP UJIAN NASIONAL MENGGUNAKAN K-MEANS DAN SUPPORT VECTOR MACHINE (SVM)

Nama mahasiswa : Chandra Eko Wahyudi Utomo
NRP : 07111550060006
Pembimbing : 1. Mochamad Hariadi, ST., M.Sc., Ph.D
2. Dr. Surya Sumpeno, ST., M.Sc.

ABSTRAK

Program Ujian Nasional (UN) sudah berjalan sejak tahun 2002 sebagai pengganti dari EBTANAS. Dalam pelaksanaannya hingga saat ini mengalami pro dan kontra dari masyarakat. Di satu sisi yang lain, fakta di lapangan menunjukkan perkembangan media sosial terutama jejaring sosial berkembang amat pesat di Indonesia. Menurut hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) 2014, jumlah pengguna internet di Indonesia adalah sebesar 88,1 juta jiwa dari 252,4 juta jiwa jumlah penduduk Indonesia. Dari jumlah pengguna tersebut, ternyata sebagian besar digunakan untuk mengakses media sosial. Oleh karena itu, riset ini berupaya menemukan opini dukungan pengguna salah satu media sosial yaitu *Twitter* terhadap *Ujian Nasional* dengan menggunakan *Support Vector Machine*. Di dalam ilmu komputer, cara menganalisis suatu opini di media sosial dengan topik tertentu disebut dengan *sentiment analysis*. *Sentiment analysis* ini bertujuan untuk menganalisis sentimen publik terhadap ujian nasional berbasis media sosial dengan menggunakan metode *Support Vector Machine*.

Dari hasil *crawling* data penelitian yang sudah dilakukan, diketahui bahwa data *Twitter* bersifat *unstructured*, memiliki variasi tinggi (*variety*) dan sangat besar (*volume*). Berdasarkan kondisi tersebut, digunakan teknik *clustering* menggunakan metode *K-Means* agar diperoleh kelompok data sebelum dilakukan analisis sentimen. Beragamnya data yang sangat besar (124.612 *tweet*) dengan tingkat variasi yang tinggi (banyak digunakan emoji dan karakter huruf, tentu saja memerlukan *tool analyzer* yang sanggup melakukannya. Oleh karena itu, pemrosesan data menggunakan *tool analyzer bigdata* dan pada penelitian ini menggunakan *Spark*. Berdasarkan *clustering* menggunakan *K-Means Methods* diperoleh *elbow* pada *cluster* ke-3 dan hasil klasifikasi menggunakan *SVM* didapatkan 2 kelompok sentimen, yaitu sentimen positif dan sentimen negatif. Dari hasil penelitian yang sudah dilakukan, diperoleh tingkat akurasi hasil klasifikasi dengan Metode *Support Vector Machine (SVM)* mencapai 90 % untuk rasio data *training* dan data *testing* 60:40, lalu 91 % untuk rasio 70:30 dan 91 % untuk rasio 80:20. Hasil riset ini adalah diperoleh sentimen positif dukungan pengguna media sosial terhadap program Ujian Nasional.

Kata kunci: *sentiment analysis*, media sosial, ujian nasional, *K-Means*, *SVM*

SENTIMENT ANALYSIS OF SOCIAL MEDIA USER BEHAVIOR TOWARDS NATIONAL EXAMS USING K-MEANS AND SUPPORT VECTOR MACHINE (SVM)

By : Chandra Eko Wahyudi Utomo
Student Identity Number : 07111550060006
Supervisor(s) : 1. Mochamad Hariadi, ST., M.Sc., Ph.D
2. Dr. Surya Sumpeno, ST., M.Sc.

ABSTRACT

The National Examination Program (UN) has been running since 2002 as substitution of EBTANAS. In its implementation until now experiencing the pros and cons of the community. On the other hand, the facts on the ground show that the development of social media, especially social networking, is growing very rapidly in Indonesia. According to the 2014 Indonesian Internet Service Providers (APJII) survey, the number of internet users in Indonesia was 88.1 million out of 252.4 million people in Indonesia. Of these users, it turns out that most of them are used to access social media. Therefore, this research seeks to find opinions of user support in one of the social media namely Twitter on the National Examination by using the Support Vector Machine. In computer science, how to analyze an opinion on social media with a particular topic is called sentiment analysis. This sentiment analysis aims to analyze public sentiments towards national exams based on social media using the Support Vector Machine method.

From the results of crawling research data that has been done, it is known that Twitter data is unstructured, has a high variation (variety) and very large (volume). Very large variety of data (124,612 tweets) with a high level of variation (widely used emojis and character letters, of course, requires a big data analyzer tool and in this study using Spark. Based on clustering using K-Means Methods obtained elbow in the 3rd cluster and the results of classification using SVM obtained 2 groups of sentiments, namely positive sentiment and negative sentiment. From the results of research that has been done, obtained the accuracy of the classification results with the Support Vector Machine (SVM) method reached 90% for the ratio of training data and testing data 60:40, then 91% for the ratio of 70:30 and 91% for the ratio of 80:20, the results of this research are obtained positive sentiment of social media user support for the National Examination program.

Key words: sentiment analysis, social media, national exams, K-Means, SVM

KATA PENGANTAR

Alhamdulillah, segala puji syukur kehadiran Allah swt yang telah senantiasa memberikan segala anugerah, rahmat dan karunia-Nya sehingga pada akhirnya penulis dapat menyelesaikan Tesis yang berjudul "*Sentiment Analysis* Perilaku Pengguna Media Sosial terhadap Ujian Nasional Menggunakan *K-Means* dan *Support Vector Machine (SVM)*."

Ucapan terima kasih, penghormatan dan penghargaan yang setinggi-tingginya penulis sampaikan kepada Bapak Mochamad Hariadi, S.T., M.Sc., Ph.D, selaku pembimbing pertama dan Bapak Dr. Surya Sumpeno, S.T., M.Sc., selaku pembimbing kedua, yang dengan penuh perhatian dan kesabaran dalam meluangkan waktu, memberikan motivasi dan bimbingan serta semangat dalam penulisan tesis ini.

Dengan terselesaikannya buku tesis ini, perkenankanlah pula penulis mengucapkan terima kasih yang tak terhingga atas bantuan dan kerja sama dari berbagai pihak kepada:

1. Istri tercinta dan tersayang, Ratna Puji Rahayu, atas segala pengorbanan, kesabaran dan perhatiannya demi terselesaikannya studi S2 ini beserta anak-anak yang solih-solihah; Ahmad Rafif Nandra Utama, Ainaraisya Uzma Aqila, Almira Lubna Dhianandra dan Arsy Kirani Adeeva Nandra.
2. Kedua orang tuaku tercinta, atas segala doa dan motivasi serta dukungan sepenuhnya sehingga penulis dapat menyelesaikan studi S2 ini.
3. Kedua mertuaku yang tercinta, atas segala doa dan bantuannya sampai terselesaikannya studi S2 ini.
4. Kementerian Riset, Teknologi dan Pendidikan Tinggi yang telah memberikan kesempatan mendapatkan beasiswa Program Magister Bidang Keahlian Telematika Departemen Teknik Elektro Fakultas Teknologi Elektro dan Informasi Cerdas pada Institut Teknologi Sepuluh Nopember Surabaya.
5. Dr. Wirawan, DEA., selaku Ketua Program Studi Pascasarjana Fakultas Teknologi Elektro, atas arahan dan motivasi serta bimbingannya sehingga penulis dapat menyelesaikan studi S2 ini.

6. Dr. Adhi Dharma Wibawa, S.T., M.T., selaku Koordinator Bidang Keahlian Telematika/*Chief Information Officer (CIO)* sekaligus Dosen Pembimbing Akademik Program Magister (S2) Bidang Keahlian Telematika Departemen Teknik Elektro Fakultas Teknologi Elektro dan Informasi Cerdas pada Institut Teknologi Sepuluh Nopember Surabaya, atas bimbingan, arahan dan motivasinya dalam menyelesaikan perkuliahan maupun penulisan tesis ini.
7. Seluruh Pengajar dan staf Program Studi Magister (S2) Fakultas Teknologi Elektro dan Informasi Cerdas khususnya pada bidang keahlian Telematika, yang telah membagi ilmu pengetahuannya serta membantu kelancaran pengurusan administrasi perkuliahan dan penyelesaian tesis ini.
8. Mahasiswa Program Studi Magister (S2) Telematika/CIO Angkatan 2015 yang selalu kompak dan saling mendukung baik dalam perkuliahan maupun dalam penyelesaian penulisan tesis ini.
9. Penghuni Laboratorium Telematika dan Laboratorium HCCV yang selalu kompak dan memberikan ruang diskusi dalam menyelesaikan penulisan tesis ini.

Dalam penyusunan buku Tesis ini, penulis menyadari masih banyak terdapat kesalahan dan kekurangan. Namun, dalam usaha untuk dapat menyempurnakan buku Tesis ini di masa mendatang, penulis mengharapkan adanya kritikan, ide dan saran yang nantinya buku ini dapat dikembangkan penulis menjadi lebih baik lagi.

Harapan penulis, semoga buku Tesis ini dapat menambah khasanah pengetahuan tentang *data mining* khususnya maupun keilmuan komputer terkini pada umumnya serta membawa manfaat yang seluas-luasnya bagi pembaca dan para akademisi terutama mahasiswa Teknik Elektro ITS Surabaya.

Surabaya, Januari 2020

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN TESIS	i
PERNYATAAN KEASLIAN TESIS	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan	6
1.4 Batasan Masalah	6
1.5 Kontribusi	6
BAB 2 KAJIAN PUSTAKA.....	7
2.1 Kajian Penelitian Terkait	7
2.2 Teori Dasar.....	8
2.2.1 <i>Sentiment Analysis</i>	8
2.2.2 Teori Media Sosial	9
2.2.2.1 Definisi	9
2.2.2.2 Sejarah Media Sosial	10
2.2.2.3 Jenis-Jenis Media Sosial	10
2.2.3 <i>Data Mining</i>	12
BAB 3 METODE PENELITIAN.....	17
3.1 Tahapan Penelitian.....	17
Sedangkan diagram sistem pada penelitian ini adalah sebagai berikut:.....	18
3.2 <i>Input</i> Dokumen	18
3.3 <i>Preprocessing</i>	19

3.4	Ekstraksi Fitur (Pembobotan).....	22
3.5	<i>Clustering</i> Menggunakan Metode K-Means	23
3.6	Analisis Sentimen dengan <i>Support Vector Machine</i>	29
3.7	Evaluasi	30
3.8	Alat dan Bahan	31
3.9	Waktu Penelitian	31
BAB 4 HASIL DAN PEMBAHASAN		33
4.1	<i>Preprocessing</i> Data	33
4.2	<i>Clustering</i> Data.....	35
4.3	Klasifikasi.....	49
4.4	Evaluasi	55
BAB 5 KESIMPULAN		57
DAFTAR PUSTAKA.....		59
LAMPIRAN		

DAFTAR GAMBAR

Gambar 1.1	Alasan Orang Indonesia menggunakan Internet (Hasil Survei APJII 2014)	2
Gambar 3.1	Arsitektur <i>Sentiment Analysis</i>	17
Gambar 3.2	Diagram Sistem Penelitian <i>Sentiment Analysis</i>	188
Gambar 3.3	Tahapan Proses Pengumpulan Data	18
Gambar 3.4	Ilustrasi <i>Case Folding</i>	20
Gambar 3.5	Ilustrasi Tahap <i>Tokenizing</i>	20
Gambar 3.6	Tahap <i>Filtering</i> dalam <i>Preprocessing</i>	21
Gambar 3.7	Proses <i>Pre-Processing</i>	21
Gambar 3.8	Tahapan Pembobotan	22
Gambar 3.9	Kamus Kerja	25
Gambar 3.10	Pelabelan kata berdasarkan kamus	25
Gambar 3.11	Kategori <i>Rule</i>	26
Gambar 3.12	List rule kombinasi kata verb active.....	27
Gambar 3.13	List rule kombinasi kata verb passive	27
Gambar 3.14	List rule kombinasi kata <i>Adjective</i>	28
Gambar 3.15	Program Evaluasi <i>Clustering</i> Menggunakan WSSE	29
Gambar 3.16	<i>Hyperplane</i> memisahkan dua kelas	29
Gambar 3.17	Proses Klasifikasi	30
Gambar 4.1	<i>Script</i> untuk <i>Preprocessing Data</i>	34
Gambar 4.2	Hasil <i>Clustering</i> Menggunakan <i>Spark</i>	36
Gambar 4.3	<i>Script</i> Evaluasi <i>Clustering</i> dengan WSSE pada Databricks.....	38
Gambar 4.4	Hasil <i>Clustering</i> Menggunakan WSSE	39
Gambar 4.5	<i>Elbow</i> terdapat di Cluster 3 pada <i>Clustering</i> Menggunakan WSSE	39
Gambar 4.6	Label Cluster	40
Gambar 4.7	<i>Cluster 0</i> dalam bentuk Worldcloud.....	41
Gambar 4.8	<i>Cluster 1</i> dalam bentuk Worldcloud.....	42
Gambar 4.9	<i>Cluster 2</i> dalam bentuk Worldcloud.....	44
Gambar 4.10	Label <i>Sentiment</i>	45
Gambar 4.11	Sentiment Netral dalam bentuk Worldcloud	46
Gambar 4.12	Sentiment Positif dalam bentuk Worldcloud.....	47
Gambar 4.13	Sentiment Negatif dalam bentuk Worldcloud	48
Gambar 4.14	Klasifikasi Ujian Nasional Menggunakan Metode Naïve Bayes	52
Gambar 4.15	Klasifikasi Ujian Nasional Menggunakan Metode SVM.....	54
Gambar 4.16	Klasifikasi Ujian Nasional Menggunakan Logistic Regression	55

“Halaman Ini Sengaja Dikosongkan”

DAFTAR TABEL

Tabel 2.1	Jejaring Sosial	12
Tabel 3.1	Opini Siswa terhadap Ujian Nasional	19
Tabel 3.2	Confusion Matrix	31
Tabel 4.1	Hasil <i>Preprocessing</i> Data.....	34
Tabel 4.2	<i>Preprocessing Result</i>	35
Tabel 4.3	Clustering Result	36
Tabel 4.4	Label Pengklusteran	40
Tabel 4.5	Cluster 0	41
Tabel 4.6	Cluster 1	42
Tabel 4.7	Hasil Rule-Base Sentiment Score	44
Tabel 4.8	Sentiment Netral.....	46
Tabel 4.9	Sentiment Positif	48
Tabel 4.10	Sentiment Negatif.....	49
Tabel 4.11	Tingkat Akurasi Data Berdasarkan Rasio Perbandingan Menggunakan SVM	55

“Halaman Ini Sengaja Dikosongkan”

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan sistem informasi berjalan sedemikian cepatnya seiring dengan perkembangan teknologi informasi yang mendukungnya. Banyak aplikasi sebagai produk dari teknologi informasi bermunculan di hadapan kita dan itu memberikan banyak pilihan untuk digunakan sesuai dengan kebutuhan hidup manusia dalam kehidupan sehari-hari. Teknologi internet dan jaringan pendukungnya semakin melengkapi pesatnya pertumbuhan dan perkembangan informasi di dunia. Saat ini teknologi jaringan internet sudah memasuki era komputer grid dan menuju masa *internet of things*. Berbagai aplikasi berbasis web sudah banyak dikembangkan, salah satu contohnya adalah aplikasi *chatting* dan media sosial. Perkembangan Internet versi 2.0 ditandai dengan berkembangnya data yang berasal dari pengguna (*user generated content*). Data ini berbentuk blog, forum, dan media sosial. Media sosial terbukti sudah menghancurkan batas-batas antara dunia nyata dengan dunia maya. Komunikasi antar manusia yang terkendala oleh jarak sebelumnya membutuhkan media transportasi sudah berevolusi menggunakan media sosial di internet pada era digital saat ini. Komunikasi ini bahkan telah melunturkan batasan-batasan waktu seiring dengan semakin mudahnya manusia mengakses internet. Manusia dapat berkomunikasi kapan saja dan dimana saja menggunakan aplikasi di media sosial.

Menurut hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) 2014, jumlah pengguna internet di Indonesia adalah sebesar 88,1 juta jiwa dari 252,4 juta jiwa jumlah penduduk Indonesia. Dari jumlah pengguna tersebut, ternyata sebagian besar digunakan untuk mengakses media sosial.

Kebiasaan manusia informasi saat ini salah satunya adalah sering melakukan *update* status di dalam sebuah media atau jejaring sosial. Jejaring sosial menjadi alasan utama orang mengakses internet. Berdasarkan data APJII 2014, terdapat 87,4 % dari pengguna internet di Indonesia yang menggunakan jejaring sosial saat mengakses internet. Semuanya itu dalam rangka

berkomunikasi dengan orang lain. Hampir dipastikan basis data yang dimiliki oleh media sosial tersebut sangat berlimpah. Hal ini disebabkan karena seringkali orang melakukan kontak dengan berbagai latar belakang dan tujuan masing-masing. Perkembangan teknologi komunikasi semakin meningkatkan orang untuk saling berinteraksi. Jika dulu hanya dengan berkiriman pesan saja, saat ini interaksi antar manusia sudah dapat dilakukan dengan saling berbicara, bahkan secara nyata dengan telekonferensi. Saat ini bahkan sudah berkembang teknologi 4G dimana orang tidak lagi berbicara mengenai cara berkomunikasi tetapi sudah merambah pada kecepatan waktu dalam berkomunikasi.



Gambar 1.1 Alasan Orang Indonesia menggunakan Internet (Hasil Survei APJII 2014)

Media sosial dapat dimanfaatkan untuk menggali data dan informasi tentang perilaku manusia dalam memenuhi kebutuhan hidupnya. Melalui media massa informasi penting tentang ketertarikan mahasiswa terhadap sesuatu dapat diperoleh. Data dan informasi itu didapatkan dengan cara dilakukan jajak pendapat atau penggalan opini melalui media sosial. Hal itu nantinya diklasifikasi berdasarkan kesamaan dari masing-masing ketertarikan tadi dan diolah menggunakan *artificial intelligent (AI)*.

Ujian Nasional merupakan amanah Undang-Undang Nomor 20 tahun 2003 tentang sistem pendidikan nasional yang bertujuan untuk mengukur pencapaian

kompetensi lulusan pada mata pelajaran tertentu secara nasional dengan mengacu pada standar kompetensi kelulusan (SKL). Namun dalam pelaksanaan program Pemerintah ini mengalami pro dan kontra di dalam masyarakat. Banyak hal yang menjadi penyebabnya, bisa karena pelaksanaannya atau pun esensi penting yang menjadi tujuan ujian nasional tersebut. Sejak munculnya Ujian Nasional (UN) pada tahun 2001/2002 (Ujian Akhir Nasional) yang kemudian diperkuat dengan adanya Peraturan pemerintah Republik Indonesia Nomor 19 tahun 2005, tampaknya UN tidak terlepas dari pro dan kontra. Banyak pihak-pihak masyarakat yang merasa dan berpendapat bahwa ujian nasional tidak perlu dilaksanakan lagi dengan berbagai alasan yang berupa keluhan, ocehan, dan pendapat lainnya, seperti dari persiapan siswa dengan berbagai bimbingan belajar yang merepotkan bagi siswa dan orang tua, tentang berbagai kecurangan, dan bahkan ada yang mengatakan bahwa ujian nasional tidak lebih dari sekedar pembodohan dan tidak ada manfaat secara langsung bagi dunia pendidikan. Sementara di sisi yang lain, banyak pula yang menyarankan agar ujian nasional tetap dipertahankan. Dan tentu saja pemerintah sebagai pemegang kebijakan tetap teguh dengan pendiriannya bahwa ujian nasional harus tetap dilaksanakan dalam rangka pemetaan mutu program satuan pendidikan, dasar seleksi masuk jenjang pendidikan selanjutnya, penentuan kelulusan dan sebagai dasar pemberian bantuan dan binaan dalam rangka peningkatan mutu pendidikan. Pemerintah kini tengah mengkaji kembali pelaksanaan Ujian Nasional (UN). Hal tersebut dikemukakan mantan Menteri Pendidikan dan Kebudayaan (Mendikbud) Muhadjir Effendy dalam rapat kerja dengan Komisi X DPR di Gedung Parlemen Jakarta. "Untuk UN, kami melakukan kajian internal. Pelaksanaan UN dikaji ulang karena saat ini UN tak lagi menjadi penentu kelulusan. Kami akan melihat manfaatnya karena keterbatasan anggaran. Apalagi tahun depan banyak program prioritas lain," kata Mendikbud dalam rapat dengan Komisi X Dewan Perwakilan Rakyat (DPR) di Jakarta, kemarin. (Media Indonesia, 2016).

Opini masyarakat tentang pelaksanaan ujian nasional tersebut sangat beraneka ragam bentuk dan keluhannya. Sering kali opini itu dituangkan dalam media sosial seperti *twitter*, *facebook* dan *instagram* serta bentuk-bentuk media sosial yang lain. Salah satu bahan yang menjadi pro dan kontra adalah ujian

nasional sebagai syarat kelulusan siswa. Walaupun Pemerintah sudah menetapkan bahwa ujian nasional tidak lagi menjadi syarat kelulusan siswa, namun pro kontra opini tentang itu masih saja beredar di kalangan masyarakat. Sejumlah pemerhati meminta agar pelaksanaan UN perbaikan ditinjau ulang karena dinilai menghamburkan uang. "Kalau sedang efisiensi, program penting yang harus diprioritaskan. UN perbaikan harus ditinjau ulang karena program ini mubazir," ungkap pemerhati pendidikan dari Universitas Pendidikan Indonesia Said Hamid Hasan. (Media Indonesia, 2016).

Twitter merupakan salah satu bagian dari media sosial. Volume *twitter* yang besar ini dapat digunakan sebagai sumber data untuk analisis sentimen. Analisis sentimen digunakan untuk menjawab pertanyaan seperti "Berapa persentase respon negatif, positif dan mengenai produk X?", "Aspek apa yang mendapat respon negatif?"

Sentiment Analysis atau *Opinion Mining* (sebagian besar peneliti memiliki pandangan dua istilah ini sama/*interchangeable*) merupakan sebuah cabang penelitian di domain *Text Mining* yang mulai dikenal sejak awal 2002. Riset ini mulai marak semenjak paper dari B. Pang dan L. Lee keluar. Secara umum, *Sentiment analysis* ini dibagi menjadi 2 kategori besar:, yaitu *Coarse-grained sentiment analysis* dan *Fined-grained sentiment analysis*. Untuk *Coarse-grained sentiment analysis* kita mencoba melakukan proses analisis pada level Dokumen. Singkatnya adalah kita mencoba mengklasifikasikan orientasi **sebuah dokumen** secara keseluruhan. Orientasi ini ada 3 jenis: **Positif, Netral, Negatif**. Akan tetapi, ada juga yang menjadikan nilai orientasi ini bersifat kontinu / tidak diskrit. Sedangkan untuk *Fined-grained sentiment analysis*, kategori kedua ini yang sedang **naik daun** sekarang. Maksudnya adalah para peneliti sebagian besar fokus pada jenis ini. Obyek yang ingin diklasifikasi bukan berada pada level dokumen melainkan sebuah **kalimat** pada suatu dokumen. Contohnya pada kalimat "Saya tidak suka *programming*". **Kalimat ini bersentimen negatif**. Lalu, contoh pada kalimat "Hotel yang baru saja dikunjungi sangat indah sekali", kalimat ini bersentimen **positif**.

Sentiment analysis terdiri dari 3 subproses besar. Masing-masing subproses ini bisa kita jadikan bahan atau topik riset secara terpisah karena

masing-masing sub proses ini membutuhkan teknik yang tidak mudah, seperti teknik *Subjectivity Classification*, *Orientation Detection* dan *Opinion Holder and Target Detection*. Untuk teknik *Subjectivity Classification* adalah untuk menentukan kalimat yang merupakan opini. Contohnya, *A bike has 2 wheels VS It is a good bike !* Sedangkan untuk teknik *Orientation Detection*, setelah berhasil diklasifikasi untuk kategori Opini, perlu ditentukan apakah dia **positif**, **negatif**, **netral** ? Seperti pada contoh seperti berikut, "*It is a good bike ! VS ah, It is a bad bike !*" Kemudian, untuk teknik *Opinion Holder and Target Detection* adalah untuk menentukan bagian yang merupakan **Opinion Holder** dan bagian yang merupakan **Target**. Contoh kalimatnya adalah "*Harry said it is a good bike.*"

Media sosial memiliki kontribusi yang sangat besar sebagai sumber data 'tersembunyi' yang sangat cepat berubah dan hadir dalam berbagai formatnya, baik berupa teks, gambar maupun video (multimedia). Sebagai ilustrasi, jumlah besaran data yang dihasilkan di dunia maya secara keseluruhan dalam tahun 2011, diprediksi mencapai 1.8 ZettaByte (10^{21} B), dan jumlah tersebut akan naik dua kali lipat setiap dua tahun [Chen-Liu, 2014]. Prediksi kenaikan tersebut juga diperkuat dengan statistik pengguna media sosial aktif pada 2014 dalam Gambar 2, yang akan terus bertambah. Karena hal tersebut di atas, data yang akan diteliti dalam tesis ini tergolong dalam *big data*. Menurut [Manyika, etal, 2011] hal utama yang membedakan *big data* dengan kumpulan data konvensional terletak pada mekanisme pengelolaannya. Sistem basis data relasional yang saat ini umum digunakan, sudah dirasakan tidak mampu menangani kompleksitas *big data* secara optimal.

1.2 Rumusan Masalah

Selama ini perilaku pengguna media sosial terhadap ujian nasional belum pernah dianalisis. Padahal, komentar pengguna media sosial terhadap ujian nasional direpresentasikan dalam bentuk komentar yang beraneka ragam (*unstructured*). Berbagai komentar pengguna media sosial terhadap ujian nasional tersebut memiliki kalimat yang mengandung sentimen (positif, negatif atau netral). Namun, semua komentar tersebut belum diketahui berapa prosentase sentimen dan tingkat akurasinya. Oleh karena itu, pada riset ini memiliki rumusan

masalah yaitu bagaimana cara mengetahui prosentase sentimen dan tingkat akurasi dari pengguna media sosial terhadap ujian nasional.

1.3 Tujuan

Tujuan dari penelitian adalah untuk menganalisis sentimen publik terhadap ujian nasional berbasis media sosial menggunakan metode *K-Means* dan *Support Vector Machine (SVM)*.

1.4 Batasan Masalah

Penelitian *sentiment analysis* perilaku pengguna media sosial menggunakan *K-Means* dan *SVM* ini dilakukan berdasarkan data yang bersumber dari Twitter saja.

1.5 Kontribusi

Kontribusi yang diharapkan dari hasil penelitian tesis adalah untuk memberikan data riset atas opini masyarakat berdasar media sosial terhadap pelaksanaan ujian nasional dengan proses non-konvensional yang hemat biaya, waktu dan tenaga. Selain itu, dapat diperoleh rekomendasi bagi pihak-pihak terkait mengenai pelaksanaan ujian nasional berdasarkan perilaku pengguna media sosial dari kajian sentimen analysis berbasis *clustering* dan klasifikasi menggunakan *K-Means* dan *SVM*.

BAB 2

KAJIAN PUSTAKA

2.1 Kajian Penelitian Terkait

Kajian penelitian terkait tentang hasil penelitian *sentiment analysis* terdahulu dapat diperoleh berdasarkan data sebagai berikut di bawah ini:

Author	Judul Paper	Metode	Data	Kekurangan
Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan	<i>Thumbsup? Sentiment Classification using Machine Learning</i> (2002)	Naïve Bayes, Maximum Entropy, and SVM	Metode klasifikasi terbaik SVM tingkat akurasi 81,6%	Berbahasa Inggris dengan 2 atribut; positif dan negatif
Franky and Ruli Manurung	<i>Machine Learning-based Sentiment Analysis of Automatic Indonesia Translations of English Movie Reviews</i> . University of Indonesia (2008)	Naïve Bayes, Maximum Entropy, and SVM	Akurasi tertinggi SVM 80,09%	Movie review data berbahasa Inggris diterjemahkan ke dalam bahasa Indonesia
Naradhipa, AR and Purwarianti, A	<i>Sentiment Classification for Indonesian Messagein Social Media</i> (2011)	Machine Learning: SVM	SVM tingkat akurasi 86,66%	Tingkat akurasi Kamus Data rendah
Mesutetla	<i>Sentiment Analysis of Turkish of Political Columns Alt Transfer Learning</i> (2013)	Naive Bayes, Maximum Entropy, dan SVM untuk klasifikasi sentimen berita politik Turki	Akurasi tertinggi NaiveBayes 72,05%, untuk tertinggi kedua Maximum Entropy 69,44% dan SVM 66,01% pada penggunaan tokenisasi biram	Berbahasa Turki dan Inggris dengan 2 atribut; positif dan negatif
Dasgupta, SS., et.al.	<i>Sentiment Analysis of Facebook Data using Hadoop Based Open Source Technologies</i> (2015)	Paket R Sentimen dari CRAN	Tingkat akurasinya 67.6 %	
Petrix Nomleni	<i>Sentiment Analysis Berbasis Big Data</i> (2015)	SVM dan Hadoop	Tingkat akurasi 72 %	

2.2 Teori Dasar

2.2.1 *Sentiment Analysis*

Sentiment analysis atau dalam bahasa Indonesia analisis sentimen telah menjadi bidang penelitian utama sejak awal 2000-an. Dampaknya dapat dilihat di banyak aplikasi praktis, mulai dari menganalisis ulasan produk (Stepanov & Riccardi, 2011) dalam (Habernal, 2014) untuk memprediksi penjualan dan pasar saham menggunakan pemantauan media sosial (Yu, Wu, Chang, & Chu, 2013) dalam (Habernal, 2014). Analisis sentimen adalah sebuah teknik untuk mendeteksi opini terhadap satu subyek (misalnya individu, organisasi ataupun produk) dalam sebuah kumpulan data (Nasukawa, 2003). Pengertian opini sendiri menurut Kamus Besar Bahasa Indonesia adalah pendapat; pikiran; pendirian. Opini di media sosial berupa *mention* dan di situs berita dan log pribadi berupa artikel. *Sentiment analysis* adalah studi komputasi mengenai sikap, emosi, pendapat, penilaian, pandangan dari sekumpulan teks yang fokusnya adalah mengekstraksi, mengidentifikasi atau menemukan karakteristik sentimen dalam unit teks menggunakan metode NLP (*Natural Language Processing*), statistik atau *machine learning*. *Sentiment analysis* merupakan proses klasifikasi dokumen tekstual ke dalam beberapa kelas seperti sentimen *positif* dan *negatif* serta besarnya pengaruh dan manfaat dari sentimen analisis menyebabkan penelitian ataupun aplikasi mengenai *sentiment analysis*. Saat ini perkembangan penelitian *sentiment analysis* mempunyai perkembangan yang sangat pesat. Bahkan di Amerika lebih dari 20 sampai 30 perusahaan memfokuskan pada layanan *sentiment analysis*. Pada dasarnya *sentiment analysis* merupakan klasifikasi, namun dalam implementasinya tidak mudah karena seperti proses klasifikasi biasa dikarenakan terkait penggunaan bahasa dimana terdapat ambiguitas dalam penggunaan kata, tidak adanya intonasi dalam sebuah teks, dan perkembangan dari bahasa itu sendiri. (Nomleni, 2015)

Sentiment analysis bermanfaat juga dalam dunia usaha seperti melakukan analisis tentang sebuah produk yang dapat dilakukan secara cepat serta digunakan sebagai alat bantu untuk melihat respon konsumen terhadap produk tersebut, sehingga dapat membuat langkah-langkah strategis pada tahapan-tahapan berikutnya. (Nomleni, 2015)

Terdapat 2 (dua) pendekatan yang dapat digunakan dalam *sentiment analysis*, yaitu *supervised* dan *unsupervised* (Alhumoud, S. et.al.: 2015). (Medhat, 2014). *The supervised learning* (juga dikenal sebagai pendekatan berbasis *corpus*) berhubungan dengan *machine learning* (Vinodhini and Chandrasekaran, 2012) dan menggunakan algoritma *machine learning* seperti *Support Vector Machine* (SVM), *Naïve Bayes* (NB), *Decision Tree* (D-Tree) and *K-Nearest Neighbors* (KNN) untuk membuat klasifikasi. Lebih jauh, *the supervised learning* meliputi 5 tahapan: membangun dataset, membangun model *classifier*, melatih model, mengevaluasi *classifier*, and menggunakan *classifier*. Pendekatan *unsupervised*, juga dikenal sebagai pendekatan berbasis leksikon (Liu, 2012) and (Alhumoud et al., 2015). Pendekatan ini didasarkan pada leksikon atau kamus kata yang dapat dibuat secara biasa ataupun secara otomatis. (Medhat et al., 2014). Leksikon merupakan koleksi kata-kata beropini yang dinyatakan dengan nilai polarisasi: +1, -1 atau 0 untuk positif, negatif atau netral, secara masing-masing (Shoukry and Rafea, 2012) and (Ravi, 2015).

2.2 .2 Teori Media Sosial

2.2.2.1 Definisi

Media sosial adalah sebuah media *online*, dengan para penggunanya bisa dengan mudah berpartisipasi, berbagi dan menciptakan isi meliputi blog, jejaring sosial, wiki, forum dan dunia virtual. Blog, jejaring sosial dan Wiki merupakan bentuk media sosial yang paling umum digunakan oleh masyarakat di seluruh dunia. Menurut Antony Mayfield dari iCrossing, media sosial adalah mengenai menjadi manusia biasa. Manusia biasa yang saling membagi ide, bekerja sama dan berkolaborasi untuk menciptakan kreasi, berfikir, berdebat, menemukan orang yang bisa menjadi teman baik, menemukan pasangan dan membangun sebuah komunitas. Intinya, menggunakan media sosial menjadikan kita sebagai diri sendiri. Selain kecepatan informasi yang bisa diakses dalam hitungan detik, menjadi diri sendiri dalam media sosial adalah alasan mengapa media sosial berkembang pesat. Tak terkecuali, keinginan untuk aktualisasi diri dan kebutuhan menciptakan *personal branding*. Teknologi-teknologi web baru memudahkan semua orang untuk membuat dan yang terpenting menyebarluaskan konten

mereka sendiri. Post di Blog, *tweet*, atau video di *YouTube* dapat direproduksi dan dilihat oleh jutaan orang secara gratis. Pemasang iklan tidak harus membayar banyak uang kepada penerbit atau distributor untuk memasang iklannya. Sekarang pemasang iklan dapat membuat konten sendiri yang menarik dan dilihat banyak orang (Zarrella, 2010: 2). Andreas Kaplan dan Michael Haenlein mendefinisikan media sosial sebagai "sebuah kelompok aplikasi berbasis internet yang membangun di atas dasar ideologi dan teknologi Web 2.0, dan yang memungkinkan penciptaan dan pertukaran *user-generated content*".

2.2.2.2 Sejarah Media Sosial

Kemunculan situs jejaring sosial ini diawali dengan adanya inisiatif untuk menghubungkan orang-orang dari seluruh belahan dunia. Situs jejaring sosial pertama, yaitu *Sixdegrees.com* mulai muncul pada tahun 1997. Situs ini memiliki aplikasi untuk membuat profil, menambah teman dan mengirim pesan. Tahun 1999 dan 2000 muncul situs sosial *Lunarstorm*, *LiveJournal*, *Cyword* yang berfungsi memperluas informasi secara searah. Tahun 2001, muncul *Ryze.com* yang berperan untuk memperbesar jejaring bisnis. Tahun 2002, muncul *Friendster* sebagai situs anak muda untuk saling berkenalan dengan pengguna lain. Tahun 2003, muncul situs sosial interaktif lain menyusul kemunculan *Friendster*, *Flickr*, *R*, *Youtube*, *Myspace*. Hingga akhir tahun 2005, *Friendster* dan *Myspace* merupakan situs jejaring sosial yang paling diminati. Lalu para pengguna sosial media beralih ke *facebook* yang sebenarnya telah dibuat pada tahun 2004, tetapi baru saja *booming* pada tahun 2006. Tahun 2006, kemunculan *twitter* ternyata menambah jumlah pemakai media sosial, *Twitter* merupakan *microblog* yang memiliki batasan karakter tulisan bagi penggunanya, yaitu 140 karakter. Lalu setelah lahirnya *Twitter* muncul jejaring sosial lain seperti *Path*, *Instagram* yang hanya bisa diakses melalui perangkat *iOs* atau *Android*.

2.2.2.3 Jenis-Jenis Media Sosial

Media sosial teknologi mengambil berbagai bentuk termasuk majalah, forum internet, *webblog*, blog sosial, *microblogging*, wiki, podcast, foto atau gambar, video, peringkat dan *bookmark* sosial. Dengan menerapkan satu set teori-

teori dalam bidang media penelitian (kehadiran sosial, media kekayaan) dan proses sosial (*self-presentation, self-disclosure*) Kaplan dan Haenlein menciptakan skema klasifikasi untuk berbagai jenis media sosial dalam artikel Horizons Bisnis mereka diterbitkan dalam 2010. Menurut Kaplan dan Haenlein ada 6 (enam) jenis media sosial:

a. Proyek Kolaborasi

Website mengizinkan usernya untuk dapat mengubah, menambah, ataupun *remove* konten – konten yang ada di website ini. Contohnya wikipedia.

b. *Blog dan microblog*

User lebih bebas dalam mengekspresikan sesuatu di blog ini seperti curhat ataupun mengkritik kebijakan pemerintah. Contohnya *Twitter, Blogspot, Tumblr, Path* dan lain-lain.

c. Konten

Para *user* dari pengguna *website* ini saling meng-*share* konten – konten media, baik seperti video, *e-book*, gambar dan lain-lain. Contohnya *Youtube*.

d. Situs jejaring sosial

Aplikasi yang mengizinkan *user* untuk dapat terhubung dengan cara membuat informasi pribadi sehingga dapat terhubung dengan orang lain. Informasi pribadi itu bisa seperti foto-foto. Contoh *Facebook, Path, Instagram* dan lain-lain.

e. *Virtual game world*

Dunia virtual dimana mereplikasikan lingkungan 3D, di mana *user* bisa muncul dalam bentuk avatar-avatars yang diinginkan serta berinteraksi dengan orang lain selayaknya di dunia nyata, contohnya game *online*.

f. *Virtual social world*.

Dunia virtual yang di mana penggunaanya merasa hidup di dunia virtual, sama seperti virtual game world, berinteraksi dengan yang lain. Namun, *Virtual Social World* lebih bebas, dan lebih ke arah kehidupan, contohnya *second life*.

Tabel 2.1 Jejaring Sosial

No	Jejaring Sosial	Jumlah Member	Keterangan
1	Facebook	845.000.000	Pengguna > 13 Tahun
2	Qzone	480.000.000	Pengguna China daratan (berbahasa mandarin)
3	Twitter	300.000.000	Microblogging terpopuler didunia
4	Habbo	200.000.000	Pengguna > 13 tahun
5	Renren	160.000.000	Situs utama di China
6	Badoo	133.000.000	Situs umum untuk pecarian jodoh, populer di Amerika dan Eropa
7	LinkedIn	120.000.000	Untuk pembisnis, pengguna >18 tahun
8	Bebo	117.000.000	Pengguna > 13 tahun
9	Vkontakte	111.578.500	Berbahasa rusia, untuk umum
10	Tagged	100.000.000	Untuk segala usia

Sumber: <https://sites.google.com>

2.2.3 Data Mining

Menurut (Luthfi, 2009) *Data Mining* merupakan metode pengolahan data berskala besar oleh karena itu *data mining* ini memiliki peranan penting dalam bidang industri, keuangan, cuaca, ilmu dan teknologi. Secara umum kajian *data mining* membahas metode-metode seperti, *clustering*, klasifikasi, regresi, seleksi variabel, dan *market basket analisis*. (Luthfi, 2009) mendefinisikan data mining sebagai penggalian informasi yang berguna dari gudang data yang besar. *Data mining* dapat disebut juga dengan *pattern recognition* merupakan pengolahan data untuk menemukan pola yang tersembunyi dari data tersebut. Hasil dari pengolahan data dengan metode *data mining* ini dapat digunakan untuk mengambil keputusan di masa depan. Pada umumnya *data mining* digunakan untuk data yang berskala besar dan banyak diaplikasikan di berbagai bidang kehidupan baik industri, kesehatan, pendidikan, perdagangan dan masih banyak lainnya.

Beberapa kelompok *Data Mining* berdasarkan tugas yang dapat dilakukan adalah sebagai berikut:

- a. Deskripsi
Digunakan oleh peneliti untuk mendeskripsikan pola dan kecenderungan yang sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.
- b. Optimalisasi
Optimalisasi dapat dikatakan hampir sama dengan klasifikasi, akan tetapi yang membedakannya adalah variabel target, pada estimasi variabel target lebih ke arah numerik daripada ke arah kategori.
- c. Prediksi
Digunakan untuk memprediksi hasil di masa yang akan datang. Hampir sama dengan klasifikasi dan estimasi.
- d. Klasifikasi
Pada klasifikasi, mempunyai target variabel kategori, sebagai contoh dalam penggolongan pendapatan terdapat tiga kategori yaitu kategori pendapatan rendah, sedang dan tinggi.
- e. Pengklusteran
Dilakukan dengan cara pengelompokkan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Dalam pengklusteran ini tidak terdapat variabel target seperti halnya pada klasifikasi.
- f. Asosiasi
Berfungsi untuk menemukan atribut yang muncul pada satu waktu.

Menurut (Dahlan Abdullah, 2015) dijelaskan bahwa karakteristik *Data Mining* adalah sebagai berikut:

- a. *Data mining* berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
- b. *Data mining* biasa menggunakan data yang sangat besar. Biasanya data yang besar digunakan untuk membuat hasil lebih dipercaya.

- c. *Association rule mining* adalah teknik *mining* untuk menemukan aturan asosiatif antara suatu kombinasi item. Contoh dari aturan asosiatif dari analisis pembelian di suatu pasar swalayan adalah bisa diketahui berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu.
- d. *Classification* merupakan proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan setiap konsep atau kelas data, yang bertujuan untuk dapat memperkirakan kelas dari suatu objek dengan label yang tidak diketahui (*unsupervised*).
- e. *Decision tree* merupakan salah satu metode *classification* yang paling populer dengan alasan karena mudah untuk diinterpretasi oleh manusia, sehingga setiap percabangan menyatakan kondisi yang harus dipenuhi dan tiap ujung pohon menyatakan kelas data.
- f. *Clustering*, melakukan pengelompokan data tanpa berdasarkan kelas data tertentu. Biasanya *clustering* dapat dipakai untuk memberikan label pada kelas data yang belum diketahui itu, metode *unsupervised learning*.
- g. *Neural Network*, merupakan jaringan syaraf buatan yang terlatih dan dapat dianggap sebagai pakar dalam kategori informasi yang akan dianalisis untuk memproyeksi situasi baru dari ketertarikan informasi.

2.2.4 K-Means Clustering API berbasis RDD

Clustering adalah permasalahan *unsupervised learning* di mana bertujuan untuk mengelompokkan subset entitas dengan satu sama lain berdasarkan beberapa gagasan kesamaan. *Clustering* sering digunakan untuk analisis eksplorasi dan / atau sebagai komponen penghubung *supervised learning* secara hierarkis (di mana pengklasifikasi atau model regresi yang berbeda dilatih untuk masing-masing *cluster*). Pada *Spark* terdapat paket-paket pemroses data dan salah satunya adalah *spark.mllib*. Paket *spark.mllib* yang mendukung model yang dimaksud adalah *K-Means*, Campuran gaussian, *Power iteration clustering (PIC)*, Alokasi *Dirichlet Laten (LDA)*, Membagi dua *K-Means* dan *Streaming K-Means*.

K-means adalah salah satu algoritma pengelompokan yang paling umum digunakan yang mengelompokkan titik-titik data menjadi sejumlah kelompok yang telah ditentukan. Implementasi *spark.mllib* termasuk varian paralel dari

metode *k-means* ++ yang disebut *k-means* //. Implementasi di *spark.mllib* memiliki parameter seperti; *k* adalah jumlah *cluster* yang diinginkan. Perhatikan bahwa dimungkinkan untuk lebih sedikit dari *k* cluster yang dikembalikan, misalnya, jika ada lebih sedikit dari *k* titik berbeda untuk mengelompok. Lalu parameter lainnya seperti *maxIterations*, adalah jumlah iterasi maksimum untuk dijalankan dan parameter *initializationMode*, untuk menentukan inisialisasi acak atau inisialisasi melalui *k-means* //. Juga ada parameter *initializationSteps*, untuk menentukan jumlah langkah dalam *k-means* // algoritma. Parameter *Epsilon*, untuk menentukan ambang jarak di mana kami menganggap *k-means* telah konvergen. Dan parameter yang terakhir adalah *initialModel*, adalah set opsional pusat *cluster* yang digunakan untuk inisialisasi. Jika parameter ini diberikan, hanya satu kali yang dijalankan.

Dalam contoh berikut setelah memuat dan mem-*parsing* data, menggunakan objek *K-Means* untuk mengelompokkan data menjadi dua kelompok. Jumlah *cluster* yang diinginkan diteruskan ke algoritma. Selanjutnya menghitung *Within Set Sum of Squared Error* (WSSSE). Untuk dapat mengurangi ukuran kesalahan ini dengan meningkatkan *k*. Bahkan *k* optimal biasanya adalah di mana ada "siku" di grafik WSSSE. Secara detail *script* untuk menjalankan *K-Means* menggunakan scala adalah sebagai berikut:

```
import org.apache.spark.mllib.clustering.{KMeans, KMeansModel}
import org.apache.spark.mllib.linalg.Vectors

// Load and parse the data
val data = sc.textFile("data/mllib/kmeans_data.txt")
val parsedData = data.map(s => Vectors.dense(s.split('
').map(_.toDouble))).cache()

// Cluster the data into two classes using KMeans
val numClusters = 2
val numIterations = 20
val clusters = KMeans.train(parsedData, numClusters, numIterations)

// Evaluate clustering by computing Within Set Sum of Squared Errors
val WSSSE = clusters.computeCost(parsedData)
println("Within Set Sum of Squared Errors = " + WSSSE)

// Save and load model
```

```
clusters.save(sc,  
"target/org/apache/spark/KMeansExample/KMeansModel")  
val sameModel = KMeansModel.load(sc,  
"target/org/apache/spark/KMeansExample/KMeansModel")  
  
Find full example code at  
"examples/src/main/scala/org/apache/spark/examples/mllib/KMeansExample.scala" in the Spark repo.
```

2.2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar *SVM* sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* (Duda & Hart tahun 1973, Cover tahun 1965, Vapnik 1964, dsb.), kernel diperkenalkan oleh Aronszajn tahun 1950, dan demikian juga dengan konsep-konsep pendukung yang lain. Akan tetapi hingga tahun 1992, belum pernah ada upaya merangkaikan komponen-komponen tersebut.

Prinsip utama *SVM* adalah untuk mencari pemisah linier atau *hyperplane* di ruang pencarian yang dapat memisahkan kelas yang berbeda. Mungkin ada beberapa *hyperplane* itu pisahkan kelas, tetapi yang dipilih adalah *hyperplane* di mana jarak normal salah satu data poin adalah yang terbesar, sehingga menggambarkan margin maksimum pemisahan.

Menurut (Medhat, W. and Hasan, A. : 2014) klasifikasi teks sangat cocok untuk *SVM* karena dari sifat jarang teks, di mana beberapa fitur tidak relevan, tetapi mereka cenderung saling berkorelasi dan umumnya diatur dalam kategori yang dapat dipisahkan secara linier.

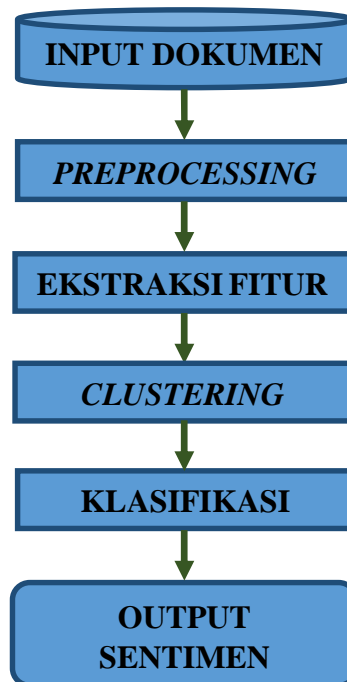
BAB 3

METODE PENELITIAN

Bab ini akan menjelaskan tentang metodologi yang digunakan dalam penelitian. Bagian-bagian yang dijelaskan pada bab ini berupa tahapan penelitian yang dilakukan dan menjelaskan alur dari metodologi penelitian tentang *sentiment analysis* perilaku pengguna media sosial menggunakan teknik *clustering K-Means* dan teknik klasifikasi *Support Vector Machine (SVM)*.

3.1 Tahapan Penelitian

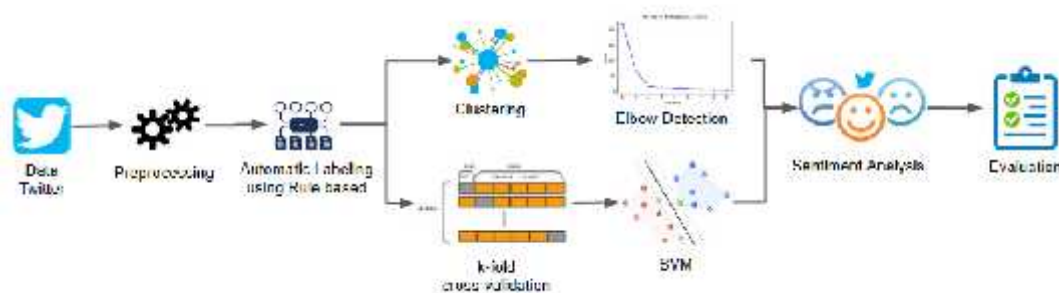
Berikut adalah tahapan umum yang akan dilakukan pada penelitian ini:



Gambar 3.1 Arsitektur *Sentiment Analysis*

Tahapan penelitian secara umum diawali dengan *input* dokumen berupa data twitter dan dilanjutkan dengan proses *preprocessing* data untuk menyiapkan data twitter sebelum diproses menggunakan teknik clustering dan teknik klasifikasi. Tahap akhir dari penelitian ini adalah keluaran dokumen berupa hasil sentiment (positif dan negatif).

Sedangkan diagram sistem pada penelitian ini adalah sebagai berikut:

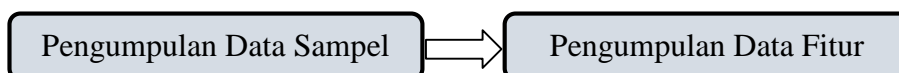


Gambar 3.2 Diagram Sistem Penelitian *Sentiment Analysis*

3.2 *Input Dokumen*

Tahap awal untuk melakukan proses *sentiment analysis* adalah pengumpulan data. Sumber data penelitian ini diambil dari *Twitter*. Pengambilan data dari *twitter* cukup mudah dilakukan karena *Twitter* menyediakan *API* (*Application Programming Interface*) (Inc., 2016) yang ditujukan kepada pengembang sistem dalam rangka memudahkan pengambilan data dari *Twitter*. Pengambilan data lebih dikenal dengan istilah *crawling* data dilakukan dengan menambang data tentang Ujian Nasional di *Twitter* melalui *API Twitter*. Data ini diperoleh dengan memasukkan kata kunci Ujian Nasional atau #UjianNasional pada *API* (*Application Programming Interface*) *Twitter*. Data yang dihasilkan berupa data komentar orang di *Twitter* tentang Ujian Nasional berformat *.txt* dan *CSV*.

Setelah data berhasil dikumpulkan menjadi sebuah dataset, tahap selanjutnya adalah pelabelan. Tahap ini terdiri dari dua proses yaitu proses Pengumpulan Data *Training* dan Pengumpulan Data *Testing*. Data *Training* adalah data yang digunakan untuk melatih sistem agar mampu mengenali pola yang sedang dicari, sedangkan data *testing* adalah data yang digunakan untuk menguji hasil pelatihan yang sudah dilakukan.



Gambar 3.3 Tahapan Proses Pengumpulan Data

Pada tahap ini dilakukan proses mengumpulkan data sampel dan data fitur yang akan digunakan pada penelitian baik data pelatihan (*Training*) maupun data pengujian (*Testing*). Data sampel dari penelitian ini adalah orang yang memberikan opini melalui media sosial *Twitter*. Sedangkan pengumpulan data fitur yang akan digunakan untuk proses klasifikasi adalah opini orang terhadap ujian nasional.

Tabel 3.1 Opini Siswa terhadap Ujian Nasional

No	Opini Siswa terhadap Ujian Nasional
1	Hpskan UN RT @saididu: Apakah kolaborasi lembaga bimbingan + sekolah ini yg ikut mempengaruhi agar ujian nasional dihapuskan ? #nanyserius
2	RT @vinna_puspita: Siapa yg mau ganti'in aku ujian nasional nnti ? Aku ga mau takut :) :D
3	RT @byanmega: Ujian Nasional taun depan 20 paket Hunt
4	Ternyata Nembak cewek itu lebih sulit dari pada ujian nasional...!!!
5	yang kelas 2 jangan seneng dulu mau jadi kelas 3, nanti pas ujian nasional lo kesel sendiri jadi kelas 3 #okesip

3.3 Preprocessing

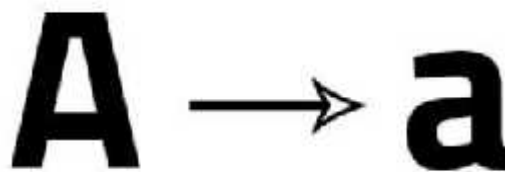
Setelah dilakukan pelabelan data, tahap selanjutnya adalah *preprocessing*. *Preprocessing* merupakan tahap awal dari penelitian ini. Menurut (Fauzi, M.A., 2018) pada tahap ini data mentah (*raw data*) akan diproses dari bentuk yang belum terstruktur menjadi data yang terstruktur untuk keperluan lebih lanjut, seperti *sentiment analysis*, *document clustering*, dan lain sebagainya. *Preprocessing* memiliki beberapa tahap, di antaranya adalah *Lexical Analysis*, *Stop-word Removal*, *Phrase Detection*, *Stemming & Lemmatization*, *Weighting*, *Indexing*. Namun, studi kasus yang digunakan pada penelitian ini adalah pada data *twitter*, sehingga struktur bahasa yang ada tidak sesuai dengan tata bahasa Indonesia. Sehingga, hanya beberapa langkah dari *preprocessing* yang dilakukan pada penelitian ini. Tahap *preprocessing* merupakan tahap dimana data disiapkan agar menjadi data yang siap dimasukkan ke dalam sistem untuk dianalisis. Berikut ini adalah penjelasan secara rinci tentang tahap *preprocessing*, yaitu:

a. *Cleaning*

Tahapan atau proses membersihkan dokumen teks yang masuk dari kata-kata yang tidak diperlukan untuk mengurangi *noise* pada proses klasifikasi seperti karakter *html*, kata kunci, ikon, *hashtag* dan lain-lain.

b. *Case Folding*

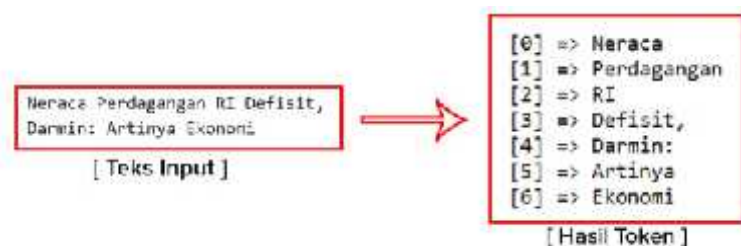
Tahapan atau proses untuk menyeragamkan bentuk huruf dan menghilangkan tanda baca. Contoh karakter yang dibuang adalah tanda seru, tanda tanya, koma dan titik. *Case Folding* adalah proses penyamaan besar kecil huruf dalam sebuah dokumen. Hal ini dilakukan untuk memudahkan tahap selanjutnya. Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu peran *Case Folding* dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar yang ditentukan. (Shabirin, I., 2017)



Gambar 3.4 Ilustrasi *Case Folding*

c. *Tokenizing*

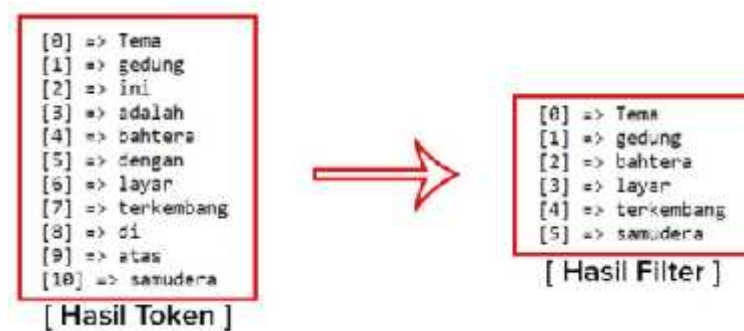
Tahapan *tokenizing* adalah tahap pemotongan string input berdasarkan kata yang menyusunnya. Atau dapat didefinisikan *tokenizing* adalah mengubah kalimat menjadi kumpulan kata-kata. Pada proses ini dilakukan penghilangan angka, tanda baca dan karakter selain huruf alfabet, karena karakter-karakter tersebut dianggap sebagai pemisah kata (*delimiter*) dan tidak memiliki pengaruh terhadap pemrosesan teks. Contoh dari *tokenizing* sebagaimana tercantum pada (Shabirin, I., 2017) adalah sebagai berikut:



Gambar 3.5 Ilustrasi Tahap *Tokenizing*

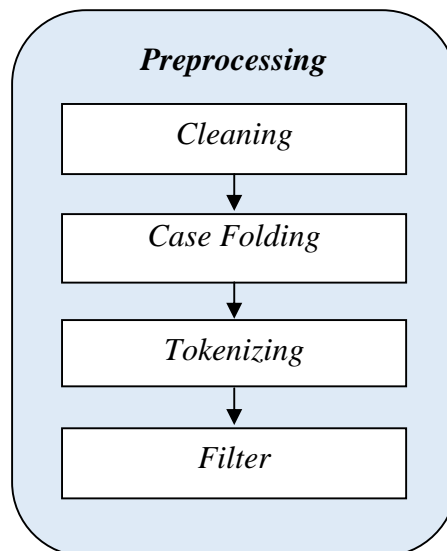
d. *Filtering*

Tahapan ini untuk memilih data pada *twitter* yang berbahasa Indonesia saja. Jika ada data yang berbahasa Indonesia tidak baku maka diganti sinonimnya berupa kata baku yang sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Tahap *Filtering* adalah tahap mengambil kata-kata penting dari hasil token pada tahap sebelumnya. Untuk memilih kata-kata penting, menggunakan *stop word list*. *Stop word list* adalah list kata-kata yang tidak dapat digunakan sebagai perwakilan dokumen, contohnya: “adalah”, “sebuah”, “di”, “dan”, dan seterusnya. (Shabirin, I., 2017)



Gambar 3.6 Tahap *Filtering* dalam *Preprocessing*

Proses *preprocessing* secara lengkap dapat dilihat sebagaimana gambar di bawah ini:



Gambar 3.7 Proses *Preprocessing*

3.4 Ekstraksi Fitur (Pembobotan)

Pembobotan kata adalah suatu mekanisme untuk memberikan skor terhadap frekuensi kemunculan sebuah kata dalam dokumen teks. Salah satu metode populer untuk melakukan pembobotan kata adalah *TF-IDF* (*Term Frequency – Inverse Document Frequency*). Pada tahapan pembobotan ini fitur yang digunakan adalah unigram dengan pembobotan menggunakan *Term Presense* (TP), *Term Frequency* (TF), *Term Frequency-Inverse Document Frequency* (TF-IDF). Kata dan simbol direpresentasi ke dalam bentuk vektor, dimana tiap kata atau simbol dihitung sebagai satu fitur. Adapun perhitungan bobot yang digunakan adalah:

1. *Feature Term Frequency (TF)*

$$\vec{d} := (n_1(d), n_2(d), \dots, n_m(d)) \quad (1)$$

2. *Feature Term Presence (TP)*

$$n_i(d) = 1, \text{ jika fitur } f_i \text{ ada di dokumen } d \quad (2)$$

$$n_i(d) = 0, \text{ jika fitur } f_i \text{ tidak ada di dokumen } d \quad (3)$$

3. *Term Frequency – Inverse Document Frequency (TF – IDF)*

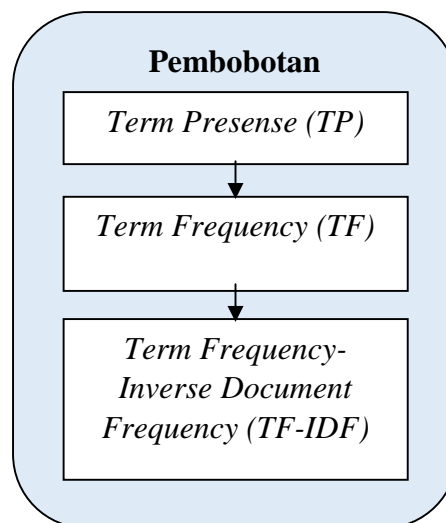
$$n_i(d) = df_i \cdot \log D/df_i \quad (4)$$

Dimana:

df_i adalah banyaknya dokumen yang mengandung fitur i (kata) yang dicari;

D adalah jumlah dokumen;

Setelah perhitungan bobot tiap *term* dilakukan, selanjutnya proses penentuan kelas sentimen yang memberikan argumen maksimum dengan membandingkan nilai dari ketiga kelas sentimen tersebut.



Gambar 3.8 Tahapan Pembobotan

3.5 Clustering Menggunakan Metode K-Means

Clustering merupakan sebuah metode analisis data yang bersifat eksploratif yang dapat menangani pengelompokan objek yang mempunyai kesamaan satu sama lain (Barakbah, A.R., Arai, K., 2004). *Clustering* merupakan salah satu metode *machine learning* untuk menangani *unsupervised data*. Metode ini biasanya digunakan untuk menganalisis persebaran suatu data dan beberapa fungsi pengelompokan lainnya. Beberapa metode yang menjadi acuan dalam penelitian ini di antaranya:

a. K-Means

K-Means clustering merupakan metode *clustering*/pengelompokan data yang terkenal simpel dan cepat. *K-Means clustering* adalah metode *clustering* yang mengelompokkan semua data yang dimiliki ke dalam k cluster, dimana nilai k sudah ditentukan sebelumnya. *K-Means* mengelompokkan data berdasarkan jarak vektor/parameter dari tiap data ke vektor/parameter dari pusat *cluster* (*centroid*) yang sudah ditentukan sebanyak k, dan mengelompokkan data-data ke pusat *cluster* yang terdekat. Algoritma metode *K-Means* menurut (Shabirin, I., 2017) adalah pertama, pilih secara acak vektor data yang akan digunakan sebagai *centroid* awal sebanyak k. Kedua, cari *centroid* yang paling dekat dari setiap data dengan cara menghitung jarak setiap data dengan setiap *centroid cluster* dan ketiga, hitung ulang untuk menentukan *centroid* baru dari setiap *cluster* dengan menghitung rata-rata nilai vektor semua data dalam *cluster* tersebut. Lakukan langkah b dan c hingga *centroid* tidak mengalami perubahan lagi (tidak ada data yang berpindah *cluster* lagi) atau perubahan *centroid* lebih kecil dari nilai *error/threshold* yang ditetapkan.

b. Bisecting K-Means

K-Means clustering bisecting adalah metode *divisive hierarchical cluster* yang menggunakan *K-Means* untuk mengelompokkan dua anak kluster C1 dan C2. Dalam proses penentuan pemisahan terbaik, *K-Means* membagi dua kelompok yang berukuran seragam. Algoritma untuk pengelompokan *K-Means Bisecting* menurut (Reddy, C.K., & Aggarwal, C.C., 2016) adalah sebagai berikut: (1) Pilih *cluster parent* untuk di-split sebagai C, (2) Pilih 2 *centroid* secara

random dari anggota C, (3) *Assign* titik yang tersisa pada *subcluster* yang paling dekat menggunakan perhitungan jarak yang telah dihitung sebelumnya, (4) Hitung kembali *centroid* dan lanjutkan proses *cluster* hingga *convergence*, (5) Hitung *dissimilarity between cluster* dari 2 *subcluster* menggunakan *centroid*, (6) Ulangi langkah (2) hingga iterasi *I* selesai, (7) Pilih *centroid* dari *subcluster* dengan *dissimilarity between cluster* paling besar, (8) Pisahkan C menjadi C1 dan C2 dengan *centroid* tersebut, (9) Pilih *cluster* yang lebih besar di antara C1 dan C2 untuk dijadikan sebagai *parent cluster*, dan (10) Ulangi mulai dari langkah (1) hingga *cluster K* didapatkan.

3.6 Rule-based Sentiment Score

Rule-based Sentiment Score merupakan bagian dari *sentiment analysis* untuk memahami, mengekstrak, dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Pada penelitian ini, merujuk pada penelitian yang dilakukan oleh (Kurniawan, R.H., 2017) yang menggunakan *rule-based* sebelumnya yang telah dikembangkan. Pada penelitian ini, dalam melakukan klasifikasi sentimen, menggunakan 3 kelas, yaitu positif, negatif, dan netral. Untuk melakukan analisis sentimen, perlu adanya data kamus sentimen kata dan *rule*. Kamus sentimen kata digunakan sebagai referensi sentimen kata dan *rule* digunakan sebagai teknik perhitungan sentimen. Berikut ini penjelasan langkah perhitungan sentimen menggunakan *rule-based*.

3.6.1 Kamus Sentimen Kata

Kamus sentimen kata digunakan untuk memberi nilai pada masing – masing kata. Nilai ini berupa angka 1, -1 dan 0 dengan 1 adalah positif, -1 adalah negatif, dan 0 adalah netral. Selain nilai sentimen, kamus sentimen kata juga menyimpan tipe kata untuk mempermudah sistem dalam membuat rule penilaian sentimen opini nantinya. Berikut ini kamus sentimen kata yang telah dikembangkan dari penelitian yang dilakukan oleh (Kurniawan, R.H., 2017):

word	type	value	word	type	value
berlanjut	verba	1	dibaca	verba-di	1
berlawanan	verba	-1	dibahas	verba-di	1
berlebihan	verba	-1	dibalas	verba-di	1
berlindung	verba	1	dibalik	preposisi	0
bermain	verba	0	dibanding	verba-di	0
bermakna	verba	1	dibandingkan	verba-di	0
bermanfaat	verba	1	dibangun	verba-di	1
bermasalah	verba	-1	dibantu	verba-di	1
bermobil	verba	0	dibarengi	verba-di	0
bermuara	verba	0	dibatalkan	verba-di	-1
bernama	verba	0	dibatasi	verba-di	-1
berontak	verba	-1	dibawa	verba-di	0
berorientasi	verba	0	dibawah	preposisi	-1
berpengalaman	verba	1	dibayang	verba-di	-1
berpengaruh	verba	1	dibela	verba-di	1

Gambar 3.9 Kamus Kata (Kurniawan, R.H., 2017)

3.6.2 Pemberian Label pada Kata

Sebelum dilakukan proses *opinion mining* dengan implementasi *rule impresi*, kata-kata hasil dari *preprocessing* harus diberikan label, berdasarkan kamus sentimen kata. Dikarenakan keterbatasan kosa kata pada kamus *sentiment word* maka beberapa kata yang tidak terdapat pada *database* kamus sentimen kata akan mendapatkan tipe kata *unknown word* dan nilai sentimennya 0 / netral.

Kata Hasil <i>Preprocessing</i>	Tipe	Value Sentimen
diperparah	Verb_di	-1
jokowi	Noun	0
mengeluarkan	Verb	-1
kebijakan bebas visa	Noun	0

Gambar 3.10 Pelabelan Kata berdasarkan Kamus (Kurniawan, R.H., 2017)

3.6.3 Rule Sentimen

Setelah *database sentimental word dictionary*, dan pemberian label pada kata, proses desain *rule* dimulai. *Rule* digunakan untuk memberikan aturan penilaian sentimen komentar. Proses ini tidak menggunakan algoritma khusus, melainkan dengan teknik impresi. Teknik ini lebih sederhana dibandingkan menggunakan algoritma. Teknik impresi lebih condong ke penganalisisan susunan kata pada suatu kalimat (Kurniawan, R.H., 2017).

Teknik ini menganalisis letak kata sifat, kata kerja, dan preposisi pada suatu kalimat. Preposisi adalah kata yang merangkaikan kata-kata atau bagian kalimat dan biasanya diikuti oleh nomina atau pronomina, contohnya tidak, belum, sangat, dll.

Pada penelitian ini menggunakan penelitian yang dilakukan oleh (Kurniawan, R.H., 2017). Pada penelitian ini menggunakan 51 *rule*, yang dibagi menjadi 3 kategori sebagai berikut:

Nama Kategori	Keterangan
Verb Aktive	<i>Rule</i> kombinasi kata yang menjadikan kata kerja aktif sebagai titik fokusnya
Verb Passive	<i>Rule</i> kombinasi kata yang menjadikan kata kerja pasif sebagai titik fokusnya
Adjective	<i>Rule</i> kombinasi kata yang menjadikan kata sifat sebagai titik fokusnya

Gambar 3.11 Kategori *Rule*

a. 20 *rule* kombinasi kata pada *Verb Active*

Daftar 20 *rule* kombinasi kata pada kategori *verb active* dapat dilihat pada tabel di bawah ini:

No	Kombinasi
1	<i>verb</i>
2	<i>verb</i> + <i>noun</i>
3	<i>verb</i> + <i>noun</i> + <i>adj</i>
4	<i>verb</i> + <i>adj</i>
5	<i>verb</i> + <i>adj</i> + <i>noun</i>
5	<i>pre</i> + <i>verb</i>
7	<i>pre</i> + <i>verb</i> + <i>noun</i>
8	<i>pre</i> + <i>verb</i> + <i>noun</i> + <i>adj</i>
9	<i>pre</i> + <i>verb</i> + <i>adj</i>
10	<i>pre</i> + <i>verb</i> + <i>adj</i> + <i>noun</i>
11	<i>noun</i> + <i>verb</i>
12	<i>noun</i> + <i>verb</i> + <i>noun</i>
13	<i>noun</i> + <i>verb</i> + <i>noun</i> + <i>adj</i>
14	<i>noun</i> + <i>verb</i> + <i>adj</i>
15	<i>noun</i> + <i>verb</i> + <i>adj</i> + <i>noun</i>
16	<i>noun</i> + <i>pre</i> + <i>verb</i>
17	<i>noun</i> + <i>pre</i> + <i>verb</i> + <i>noun</i>
18	<i>noun</i> + <i>pre</i> + <i>verb</i> + <i>noun</i> + <i>adj</i>
19	<i>noun</i> + <i>pre</i> + <i>verb</i> + <i>adj</i>
20	<i>noun</i> + <i>pre</i> + <i>verb</i> + <i>adj</i> + <i>noun</i>

Gambar 3.12 *List Rule* Kombinasi Kata *Verb Active*

b. 20 rule kombinasi kata pada *Verb Passive*

No	Kombinasi
1	<i>verb_di</i>
2	<i>verb_di</i> + <i>noun</i>
3	<i>verb_di</i> + <i>noun</i> + <i>adj</i>
4	<i>verb_di</i> + <i>adj</i>
5	<i>verb_di</i> + <i>adj</i> + <i>noun</i>
6	<i>pre</i> + <i>verb_di</i>
7	<i>pre</i> + <i>verb_di</i> + <i>noun</i>
8	<i>pre</i> + <i>verb_di</i> + <i>noun</i> + <i>adj</i>
9	<i>pre</i> + <i>verb_di</i> + <i>adj</i>
10	<i>pre</i> + <i>verb_di</i> + <i>adj</i> + <i>noun</i>
11	<i>noun</i> + <i>verb_di</i>
12	<i>noun</i> + <i>verb_di</i> + <i>noun</i>
13	<i>noun</i> + <i>verb_di</i> + <i>noun</i> + <i>adj</i>
14	<i>noun</i> + <i>verb_di</i> + <i>adj</i>
15	<i>noun</i> + <i>verb_di</i> + <i>adj</i> + <i>noun</i>
16	<i>noun</i> + <i>pre</i> + <i>verb_di</i>
17	<i>noun</i> + <i>pre</i> + <i>verb_di</i> + <i>noun</i>
18	<i>noun</i> + <i>pre</i> + <i>verb_di</i> + <i>noun</i> + <i>adj</i>
19	<i>noun</i> + <i>pre</i> + <i>verb_di</i> + <i>adj</i>
20	<i>noun</i> + <i>pre</i> + <i>verb_di</i> + <i>adj</i> + <i>noun</i>

Gambar 3.13 *List Rule* Kombinasi Kata *Verb Passive*.

c. 11 *rule* kombinasi frasa pada *Adjective*

No	Kombinasi
1	Adj
2	adj + verb
3	adj + verb + noun
4	pre + adj
5	pre + adj + verb
6	pre + adj + verb + noun
7	noun + adj + verb
8	noun + adj + verb + noun
9	noun + pre + adj
10	noun + pre + adj + verb
11	noun + pre + adj + verb + noun

Gambar 3.14 *List Rule* Kombinasi Kata *Adjective*

3.7 *K-Fold Cross Validation*

Cross-validation (CV) adalah metode statistik yang dapat digunakan untuk mengevaluasi kinerja model atau algoritma dimana data dipisahkan menjadi dua subset yaitu data proses pembelajaran dan data validasi / evaluasi. Model atau algoritma dilatih oleh subset pembelajaran dan divalidasi oleh subset validasi. Selanjutnya pemilihan jenis CV dapat didasarkan pada ukuran dataset. Biasanya *CV K-Fold* digunakan karena dapat mengurangi waktu komputasi dengan tetap menjaga keakuratan estimasi. (Wibowo, A., 2019)

Tahap terakhir *Clustering* adalah menguji algoritma *clustering* menggunakan *WSSSE (Within Set Sum of Squared Errors)*. Tujuan dari metode ini adalah untuk mengetahui berapa banyak "*cluster*" yang optimal untuk suatu dataset. *Script* pengujian ditunjukkan pada Gambar 2.

```

databricks ClusteringEvaluationWithWSSSE (Scala)

import scala.collection.mutable.ListBuffer

var listBufferWSSSE = new ListBuffer[(Int, Double)]()

for( numCluster <= 2 to 36){
  var model = KMeans.train(vectors, numCluster, numIterations)
  var WSSSE = model.computeCost(vectors)
  println("Within Set Sum of Squared Errors untuk " + numCluster + " Class = " + WSSSE)
  var value = (numCluster, WSSSE)
  ListBufferWSSSE += value
}

val listWSSSE = ListBufferWSSSE.toList
val WSSSEDF = ListBufferWSSSE.toDF("Num of Cluster", "WSSSE")

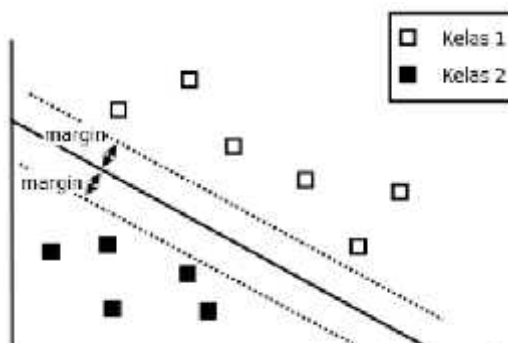
display(WSSSEDF)

```

Gambar 3.15 Program Evaluasi *Clustering* Menggunakan WSSSE

3.6 Analisis Sentimen dengan *Support Vector Machine*

Support Vector Machine adalah seperangkat metode pembelajaran terbimbing (*supervised learning*) yang menganalisis data dan mengenali pola, digunakan untuk klasifikasi dan analisis regresi (Saraswati, 2011). Klasifikasi ini dilakukan dengan mencari *hyperplane* atau garis pembatas (*decision boundary*) yang memisahkan antara satu kelas dengan kelas yang lain. Dalam konsep ini, *Support Vector Machine* berusaha mencari *hyperplane* terbaik diantara fungsi yang tidak terbatas jumlahnya. Fungsi yang tidak terbatas dalam pencarian *hyperplane* di metode *Support Vector Machine* merupakan sebuah keuntungan, dimana pemrosesan pasti akan selalu bisa dilakukan bagaimanapun data yang dimiliki (Hadna, 2016). *Hyperplane* dapat dilihat pada Gambar 2.



Gambar 3.16 *Hyperplane* Memisahkan Dua Kelas

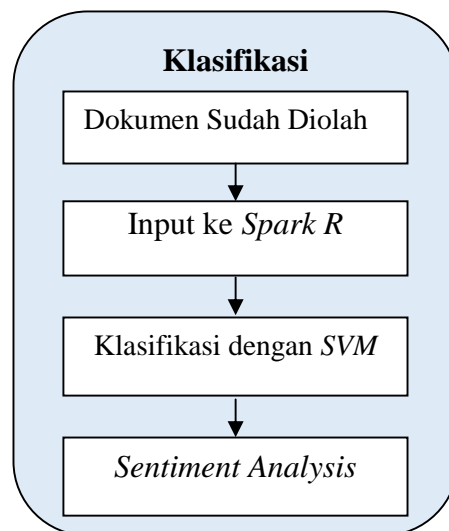
Data yang digunakan pada metode *Support Vector Machine* dengan $x_i \in \mathbb{R}^d$ dan label dinotasikan dengan $y_i \in \{1,2\}$ untuk $i = 1,2, \dots, l$ yang mana l adalah banyaknya data.

Support Vector Machine dapat diformulasikan ke dalam Persamaan (3) untuk $y_i = +1$ dan Persamaan (4) untuk $y_i = -1$.

$$x_i * w + b \geq +1 \quad (3)$$

$$x_i * w + b \leq -1 \quad (4)$$

Pada penelitian ini dokumen yang sudah diolah dimasukkan ke dalam *Spark*, lalu data itu diolah menggunakan algoritma *SVM*. Untuk mendapatkan hasil klasifikasi terbaik, diujikan menggunakan 3 (tiga) kelas sentimen dan selanjutnya dibandingkan nilainya dari ketiga kelas tersebut.



Gambar 3.17 Proses Klasifikasi

3.7 Evaluasi (*Evaluation*)

Evaluation merupakan tahap memberikan penilaian terhadap suatu hasil dari proses klasifikasi. Penilaian tersebut yang nantinya menjadi pengukuran performa dari metode yang diajukan. Sebelum menggunakan metode evaluasi, terlebih dahulu dibuat *confusion matrix* untuk membedakan solusi optimal selama proses *training* dari klasifikasi. *Confusion matrix* dapat dilihat pada gambar di tabel ini:

	Actual Positive Class	Actual Negative Class
Predicted Positive Class	True positive (<i>tp</i>)	False negative (<i>fn</i>)
Predicted Negative Class	False positive (<i>fp</i>)	True negative (<i>tn</i>)

Tabel 3.2 *Confusion Matrix* (Hossin, M. & Sulaiman, M.N., 2015)

Baris pada tabel menunjukkan *predicted class* (hasil sentimen dari *SVM*), sementara kolom pada tabel menunjukkan *actual class* (label pada data *testing*). Metode evaluasi yang digunakan untuk evaluasi ini terdapat 2, yaitu *precision* dan *recall*.

1. *Precision*

Precision digunakan untuk mengukur *Positive Class* yang berhasil diprediksi secara tepat dari keseluruhan *Positive Class*. Rumus yang digunakan untuk mengukur *precision* adalah sebagai berikut:

$$P = \frac{tp}{tp + fp}$$

dimana: *tp* = *True Positive*

fp = *False Positive*

2. *Recall*

Recall digunakan untuk mengukur bagian *Positive Class* yang terklasifikasi secara benar.

$$R = \frac{tp}{tp + fn}$$

dimana: *tp* = *True Positive*

tn = *True Negative*

3.8 Alat dan Bahan

Berikut adalah alat yang dibutuhkan dalam penelitian:

1. *Laptop* minimal RAM 4 GB sebagai media untuk pemrosesan data.
2. Data sampel hasil *download* data yaitu data riset *twitter*.
3. Alat tulis untuk melakukan analisis data.
4. Kamera untuk dokumentasi kegiatan penelitian.

3.9 Waktu Penelitian

Penelitian ini rencananya akan dilakukan selama kurun waktu selama 10 bulan yaitu sampai dengan akhir Desember 2017.

“Halaman Ini Sengaja Dikosongkan”

Pada tahap *preprocessing data* melalui Spark, semua rangkaian proses dilakukan dalam 1 eksekusi dan hasilnya berupa data hasil *preprocessing* dalam bentuk scala.

Pada tahap *preprocessing* dilakukan *case folding*, untuk menyamakan besar huruf. Kemudian dilakukan *cleaning*, untuk membersihkan dari huruf selain abjad. Kemudian proses *tokenizing*, memecah kalimat menjadi kata. Kemudian tahap *filtering* untuk menghilangkan kata yang termasuk *stopword*. Berikut ini hasil dari tahap *preprocessing*:

Tabel 4.1 Hasil *Preprocessing* Data

<i>Raw Data</i>	<i>Preprocessed Data</i>
Ujian kehidupan jauh lbh bikin stres drpd ujian nasional! *tsah	ujian kehidupan jauh lbh bikin stres drpd ujian nasional tsah
ini lebih mendebarkan dibanding Detik-detik Ujian Nasional (-, — °) aseeeeeem !!!	ini lebih mendebarkan dibanding detikdetik ujian nasional aseeeeeem
Berdoa bareng" saat mau ujian nasional #DontYouRemember	berdoa bareng saat mau ujian nasional dontyouremember

Preprocessing data awalnya menggunakan pemrograman berbasis PHP, namun data mentah harus dipecah-pecah terlebih dahulu agar dapat diproses satu per satu. Pada penelitian ini awalnya digunakan program aplikasi berbasis PHP untuk penyelesaian *preprocessing*-nya.

Program aplikasi tersebut memiliki kelemahan yaitu lambat dalam melakukan pemrosesan data dimana data mentah harus dipecah-pecah menjadi 100 *row* agar dapat di-*running*. Sebenarnya hasil *preprocessing* keluar dengan baik. Namun, ketika sampai data terkumpul hingga 1.000 data, kemampuan komputer mengalami hambatan karena proses *preprocessing* data berhenti. Di satu sisi, *dataset* yang dimiliki berjumlah 124.612 *row* yang berasal dari *Twitter*. Secara kecepatan tentu saja memerlukan waktu yang sangat lama hanya untuk memproses *preprocessing* data saja. Permasalahan ini terpecahkan oleh *tool analyzer Spark*. Karena di Spark memang sudah disediakan untuk pemrosesan

data dalam kapasitas data yang sangat besar (*big data*). Di dalam *Spark* sudah terpasang berbagai *tool analyzer* dengan sudah dilengkapi berbagai metode *machine learning*.

Preprocessing yang berjalan di atas *tool analyzer Spark* terdiri dari beberapa tahap yang meliputi (1) *Lower Case*, (2) *Remove Mention*, (3) *Remove link*, (4) *Remove Hashtag*, (5) *Remove Retweet*, (6) *Emoji diubah menjadi teks*, (7) *Remove duplicate character*, (8) *Remove punctuation*. Di bawah ini merupakan ilustrasi hasil *preprocessing* menggunakan *Spark*:

Tabel 4.2 *Preprocessing Result*

Raw Text	
<p>RT @StorySMU: *ujian nasional* izin ke toilet sebentar ke pengawas. Padahal kunci jawaban di selipin di atas pintu WC #storySMU 211424704777560000</p>	
Preprocessing	
<p>USER_MENTION ujian nasional izin ke toilet sebentar ke pengawas. padahal kunci jawaban di selipin di atas pintu wc storysmu</p>	
<p> : Lower Case</p> <p> : Remove Mention</p> <p> : Remove Link</p> <p> : Remove Hashtag</p> <p> : Remove Retweet</p> <p> : Emoji -> Text</p> <p> : Remove Duplicate Character</p> <p> : Remove Punctuation</p>	

4.2 Clustering Data

Sebagaimana sudah dijelaskan pada bagian metodologi di bab sebelumnya, riset *sentiment analysis* perilaku pengguna media sosial akhirnya menggunakan metode clustering. Metode ini digunakan untuk menyelesaikan masalah data yang bersifat *unstructured*. Data *twitter* ternyata sangat beragam dan bervariasi. Beragamnya data perlu dikelompokkan untuk memudahkan analisis sentimennya. Teknik *clustering* dipilih untuk mengatasi permasalahan tersebut. Teknik *Clustering* juga menggunakan *tool analyzer Spark* dengan mencari *elbow* terbaik untuk *cluster 2* hingga *cluster 50*.

Sedangkan untuk tahap *clustering* data, diperoleh hasil sebagai berikut:

```

10 *komentar: USER_MENTION kamu & USER_MENTION yang belajar dengan senang gila mau jadi belajar nanti pas ujian nasional ini bakal sendiri jadi kelas" "label":0
11 *komentar: smpek skarang masih inget bnget wktu minta doa buat ujian nasional sama rikaendingnya kita nanges breng" "label":0
12 *komentar: kamu pasti bisa hahaha USER_MENTION kk tanggal aku mau ujian nasional pleaseeee semangaaat nyaa aku gugup mau ngomong apa" "label":0
13 *komentar: besok ak liburan USER_MENTION ujian nasional telah usaiUSER_MENTION ujian hidup sedang dimulai" "label":1
14 *komentar: USER_MENTION brb pindah ke cinaUSER_MENTION cina larang siswi peserta ujian nasional pakai bra LINK" "label":1
15 *komentar: kk tanggal aku mau ujian nasional pleaseeee semangaaat nyaa aku gugup mau ngomong apa stres" "label":2
16 *komentar: yassalam kaya ngadepin soal ujian nasionalo" "label":2

```

Gambar 4.2 Hasil *Clustering* Menggunakan *Spark*

Data hasil eksekusi di atas merupakan data hasil *preprocessing* dan *clustering* dalam 2 label, yaitu positif dan negatif menggunakan metode *K-Means*. Hasil *clustering* pada Spark diperoleh untuk kelas positif dilabeli dengan angka 0 dan untuk kelas negatif dilabeli dengan angka 1 pada masing-masing kalimat *tweet*.

Tabel 4.3 *Clustering Result*

Cluster	Tweet
0	smpek skarang masih inget bnget wktu minta doa buat ujian nasional sama rikaendingnya kita nanges breng
	kamu pasti bisa hahaha USER_MENTION kk tanggal aku mau ujian nasional pleaseeee semangaaat nyaa aku gugup mau ngomong apa
1	besok ak liburan USER_MENTION ujian nasional telah usaiUSER_MENTION ujian hidup sedang dimulai
	USER_MENTION brb pindah ke cinaUSER_MENTION cina larang siswi peserta ujian nasional pakai bra LINK
2	kk tanggal aku mau ujian nasional pleaseeee semangaaat nyaa aku gugup mau ngomong apa stres
	yassalam kaya ngadepin soal ujian nasionalo

Berdasarkan hasil pelabelan menggunakan teknik *clustering* dengan metode *K-Means*, diperoleh bahwa ternyata kalimat positif ketika diolah dengan *Spark* dapat berlabel negatif. Hal ini menarik untuk dikaji karena berbeda *sentiment analyst* konvensional terutama yang menggunakan pendekatan leksikon semantik. Berdasarkan penelitian-penelitian sebelumnya, sebuah kalimat dapat dilabeli (positif atau negatif) minimal oleh beberapa ahli bahasa untuk memastikan bahwa kalimat itu bersentimen positif, negatif atau netral. Pelabelan otomatis pada *Spark* didasarkan pada jumlah kata di dalam suatu kalimat tertentu. Kalimat tertentu dapat berlabel positif dikarenakan struktur kalimat memiliki kelengkapan persyaratan struktur kalimat (terdiri dari Subjek, Predikat, Objek dan Keterangan). Betapapun definisi suatu kata bernilai negatif namun karena jumlah katanya banyak, maka kalimat tersebut bernilai positif. Dengan *clustering*, algoritma hanya menggunakan nilai fitur TF-IDF saja. Jadi, *labelling*-nya hanya berdasarkan nilai TF-IDF yang notabene dikomputasi oleh program.

Teknik *clustering* pada *Spark* untuk riset *sentiment analysis* tentang ujian nasional menghasilkan 3 *cluster* terbaik dari sebaran jumlah *cluster* antara 2 hingga 50 *cluster*. Tabel 4.3 menunjukkan informasi bahwa terdapat 3 kelompok hasil *clustering* menggunakan *K-Means* pada *Spark*. *Cluster* data diartikan kelompok, dengan demikian pada dasarnya analisis *cluster* akan menghasilkan sejumlah *cluster* (kelompok). Analisis ini diawali dengan pemahaman bahwa sejumlah data tertentu sebenarnya mempunyai kemiripan di antara anggotanya. Karena itu, dimungkinkan untuk mengelompokkan anggota-anggota yang mirip atau mempunyai karakteristik yang serupa tersebut dalam satu atau lebih dari satu *cluster*. Dalam menentukan hasil *clustering* peneliti menggunakan hitungan jumlah kata dari masing-masing kalimat sehingga terbentuk 3 *cluster*. Semua proses tersebut dilakukan dalam tool analyzer *Spark*.

Tahap akhir dari *Clustering* adalah menguji algoritma *clustering* dengan menggunakan *WSSSE* (*Within Set Sum of Squared Errors*). Pengujian ini dilakukan untuk memastikan bahwa proses *clustering* benar-benar valid. Pengujian algoritma *clustering* dengan menggunakan *WSSSE* sudah sering digunakan dalam *analysis* pada *Spark*.

```

import scala.collection.mutable.ListBuffer

var listBufferWSSSE = new ListBuffer[(Int, Double)]()

for( numCluster <- 2 to 50){
  var model = KMeans.train(vectors, numCluster, numIterations)
  var WSSSE = model.computeCost(vectors)
  println("Within Set Sum of Squared Errors untuk " + numCluster + " Class = " + WSSSE)
  var value = (numCluster, WSSSE)
  listBufferWSSSE += value
}

val listWSSSE = listBufferWSSSE.toList
val WSSSEDF = listBufferWSSSE.toDF("Num of Cluster", "WSSSE")

display(WSSSEDF)

```

Gambar 4.3 Script Evaluasi *Clustering* dengan *WSSSE* pada *Databricks*

Script dalam menentukan uji algoritma *clustering* dengan menggunakan *WSSSE* adalah sebagai berikut dibawah ini:

```

for( numCluster <- 2 to 50){

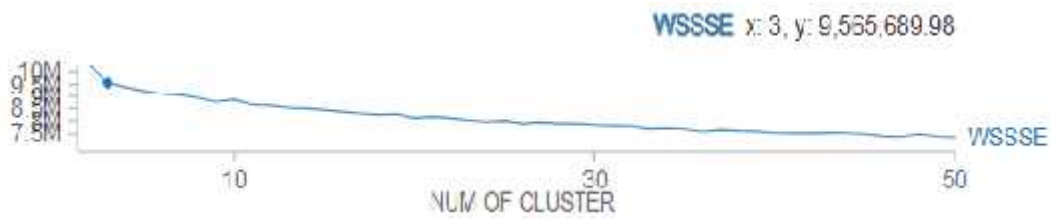
  var model = KMeans.train(vectors, numCluster, numIterations)
  var WSSSE = model.computeCost(vectors)
  println("Within Set Sum of Squared Errors untuk " + numCluster +
  Class = "+WSSSE)
  var value = (numCluster, WSSSE)
  listBufferWSSSE += value
}

val listWSSSE = listBufferWSSSE.toList
val WSSSEDF = listBufferWSSSE.toDF("Num of Cluster", "WSSSE")

display(WSSSEDF)

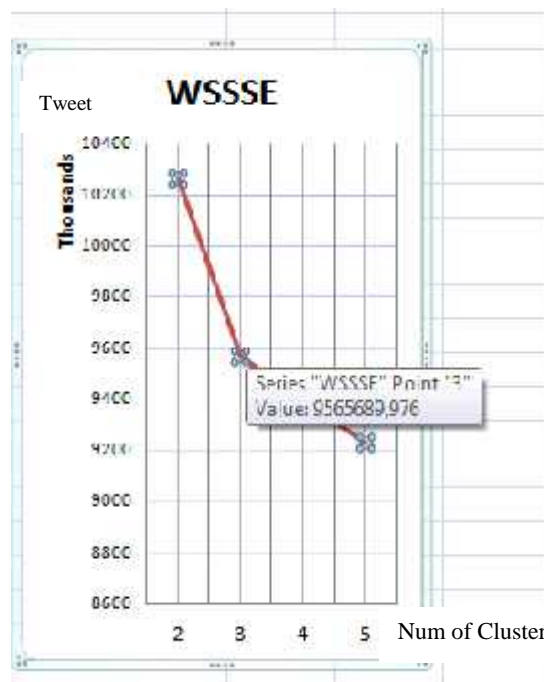
```

Proses ini mengelompokkan data berdasarkan fitur yang telah dihasilkan pada proses sebelumnya, data agregasi. Pengelompokan data ini menggunakan metode *K-Means* yang diulangi secara iteratif dengan nilai *n* mulai dari 2 hingga 50. Kemudian dicari nilai *n* yang optimal menggunakan metode *elbow*. Metode *elbow* ini menginterpretasikan dan memvalidasi konsistensi dalam *cluster analysis* untuk menentukan jumlah *cluster* optimal. Dari hasil metode *elbow* yang digunakan pada rentang *cluster* 2 hingga 50, didapatkan *cluster* optimal pada *cluster* ke-3.



Gambar 4.4 Hasil *Clustering* Menggunakan *WSSSE*

Maksud dari metode ini adalah untuk mengetahui berapa banyak “*Cluster*” yang paling optimal untuk suatu *dataset*. Kebanyakan jumlah “*Cluster*” paling optimal berada tepat pada bagian “*elbow*” atau siku, dimana grafik dapat dilihat bagian “*Elbow/siku*” berada di *cluster* 3.



Gambar 4.5 *Elbow* terdapat di *Cluster* 3 pada *Clustering* Menggunakan *WSSSE*

Pada pengujian algoritma *clustering* dengan menggunakan *WSSSE* (*Within Set Sum of Squared Errors*) diperoleh hasil bahwa untuk *elbow* terbaik terletak pada *cluster* 3.

Proses *crawling data* berhasil mengumpulkan data *tweet* sebanyak 124.612. Kemudian data diproses dalam pengelompokan dan mendapatkan nilai siku dari 3. Hasil pengelompokan dengan 3 *centroid* akan digunakan sebagai label sentimen. Rangkaian proses ini dapat berjalan dengan baik pada *platform big*

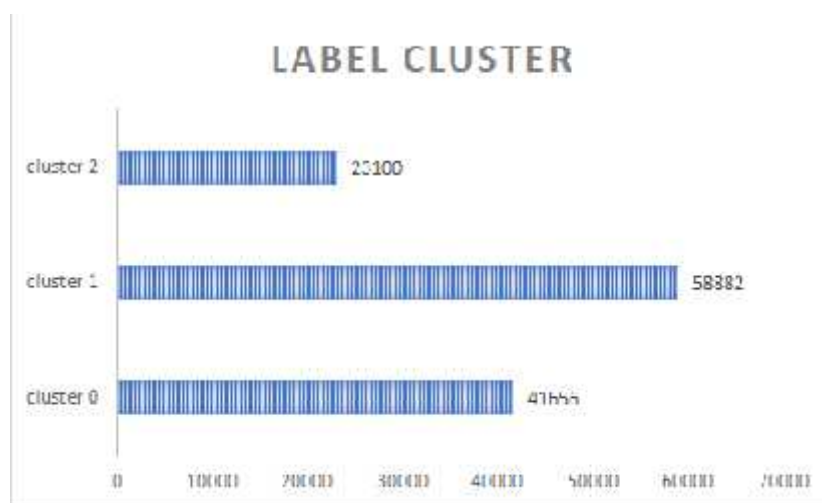
data, sehingga menghemat waktu pemrosesan. Namun, hasil yang diperoleh belum mampu menunjukkan keakuratan sentimen label. Proses pemisahan *cluster* cenderung didasarkan pada isi *tweet* dan panjang pendek *tweet*. Sebagai contoh adalah kluster 2, *tweet* yang diklasifikasikan sebagai kluster 2 memiliki *tweet* yang cenderung pendek. Maka *tweet* di *cluster* 2 tidak memiliki *user_mention*. Meskipun *cluster* 1 dan 0 memiliki *user_mention*, perbedaannya adalah bahwa *cluster* 1 memiliki konten yang lebih pendek, sedangkan *cluster* 0 memiliki konten yang lebih panjang.

Dari 3 *cluster* ini didapat hasil evaluasi sebagai berikut:

Tabel 4.4 Label Pengklusteran

<i>Tweet</i>	<i>Cluster</i>
USER_MENTION ranking tp ujian nasional nilai bagus nyoo wee p dituker ma nasi soto di kantinnya pak gik. wkwkwk	0
USER_MENTION iya kak. udah pengumuman. nem gatau kak. tapi jumlah ujian nasional	1
denger ost. madagaskar jadi keinget pas ujian nasional bhs ing	2

Hasil dari *clustering* kemudian divisualisasikan menjadi grafik di bawah ini. Grafik tersebut menunjukkan bahwa *cluster* yang mempunyai anggota paling besar adalah *cluster* 1. Kemudian, *cluster* yang terbesar kedua adalah *cluster* 0. Kemudian, *cluster* yang terkecil adalah *cluster* 2.



Gambar 4.6 Label *Cluster*

Hasil pembobotan pada ekstraksi fitur untuk *cluster* 1 diperoleh hasil *Term* dan *Frequency* sebagaimana tertera berikut di bawah ini:

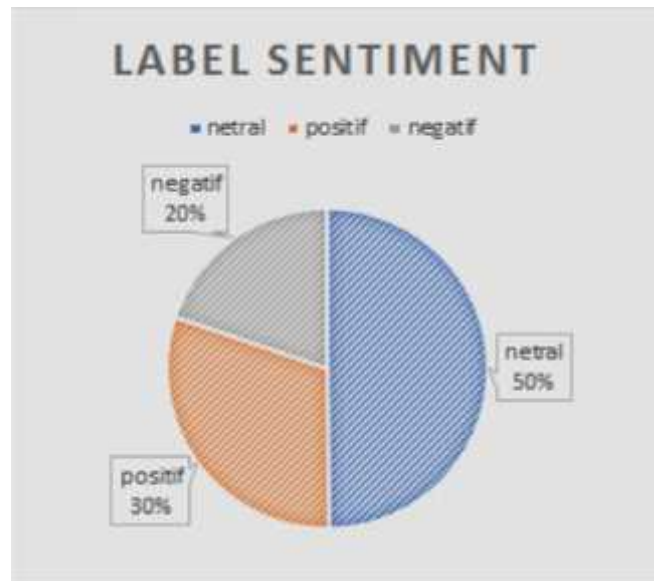
Tabel 4.6 *Cluster* 1

<i>Term</i>	<i>Frequency</i>
ujian	61511
Nasional	57399
<i>Link</i>	12267
Lulus	5276
Un	4833
Soal	3681
Sd	3194
Sma	2928
Mau	2900
Smp	2636

Pada *cluster* 1, *term* "ujian" dan *term* "nasional" tetap memiliki frekuensi yang terbesar dibandingkan dengan kata-kata yang lain seperti ditampilkan pada *wordcloud* di atas. *Term* "ujian" pada *Cluster* 1 memiliki frekuensi kata yang muncul sebanyak 61.511 dan *term* "nasional" memiliki frekuensi kata yang muncul sebanyak 57.399. Sehingga dua kata tersebut juga mendominasi pada *cluster* 1.

Sementara itu, *cluster* 2 memuat kata di antaranya kata "gue", "gak", "ya", "ada", "kok", "sd", "smp", "smk", "sma", "snmptn". *Cluster* 2 cenderung mengandung kata-kata tidak baku dan bahasa keseharian. Sehingga sedikit banyak semakin tampak *cluster* 2 memuat kata-kata tidak baku dan tidak sesuai dengan kaidah ejaan bahasa Indonesia. Seringkali kata-kata seperti itu diikuti dengan karakter-karakter atau emoji-emoji, yang tentunya sudah dihilangkan pada tahap

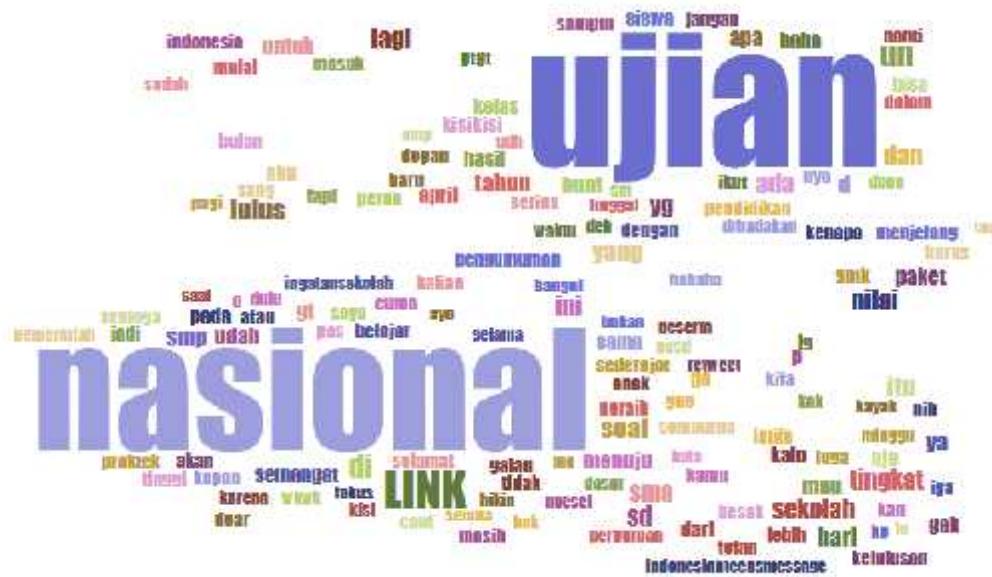
Hasil dari *sentiment* kemudian divisualisasikan menjadi grafik berikut di bawah ini:



Gambar 4.10 Label *Sentiment*

Grafik tersebut menunjukkan bahwa *sentiment* yang paling banyak adalah sentiment netral. Kemudian sentiment positif dan terakhir adalah sentiment negatif. Berdasarkan grafik di atas diperoleh informasi bahwa kalimat-kalimat yang terdapat di media sosial *Twitter* banyak berisi kalimat-kalimat tidak bersentimen daripada kalimat-kalimat yang bersentimen. Terbukti terdapat 50% sentiment yang berlabel netral dan 50% yang bersentimen, baik sentimen positif maupun sentimen negatif. Rata-rata pengguna *Twitter* mengirimkan pesan yang tidak berpihak kaitannya dengan pro dan kontra mengenai Ujian Nasional. Mereka umumnya mencuplik kata-kata "ujian nasional" sebagai bentuk ungkapan pembeda atas sesuatu yang terjadi pada diri mereka.

Sentimen netral rata-rata didominasi dengan kata yang tidak mengandung sentimen di antaranya "indonesianteensmessage", "sma", "yang", "di", "smp". Seperti ditunjukkan di Gambar 4.11 terlihat dalam bentuk *world cloud*.



Gambar 4.11 *Sentiment* Netral dalam Bentuk *World Cloud*

Berdasarkan hasil klasifikasi, diperoleh *sentiment* netral pada pengguna media sosial *twitter* seperti ditunjukkan oleh *wordcloud* dengan kata-kata dominan antara lain: ujian, nasional, link, UN, lulus, soal, sekolah, tingkat, sma dan nilai.

Tabel 4.8 *Sentiment* Netral

<i>Term</i>	<i>Frequency</i>
Ujian	66876
Nasional	60598
<i>Link</i>	13642
Un	7163
Lulus	4685
Soal	4651
Sekolah	4151
Tingkat	4117
Sma	4034
Nilai	3902

Sementara, *sentiment* positif rata-rata didominasi dengan kata yang mengandung sentimen positif di antaranya “keselamatan”, “semoga”, “amin”, “semangat”



Gambar 4.12 Visualisasi Sentimen Positif dalam Bentuk *Word Cloud*

Berdasarkan hasil klasifikasi, diperoleh *sentiment* positif pada pengguna media sosial *twitter* seperti ditunjukkan oleh *wordcloud* dengan kata-kata dominan antara lain: ujian, nasional, *link*, UN, lulus, sukses, nilai, amin, UN, kelas dan paket. Untuk *term* tertinggi pada *sentiment* positif adalah *term* "Ujian" dan "Nasional". Kedua kata tersebut sangat sering disebut dalam cuitan pengguna *twitter*. Sedangkan kata-kata positif yang mengarah pada *sentiment* positif adalah *term* "link", "UN", "lulus", "sukses", "nilai", kata doa "amin". Juga ada *term* "Un", "kelas" dan *term* "paket" yang disebut sebanyak lebih dari 2.000 kali. Walaupun kalimat yang menjadi komentar adalah kalimat yang belum tentu menyatakan dukungan pasti ke program Ujian Nasional. Namun, setidaknya dari analisis data menggunakan *SVM* dapat diketahui bahwa mayoritas *sentimen* pengguna media sosial *twitter* mengarah pada *sentimen* positif.

Berdasarkan hasil klasifikasi SVM pula, diperoleh *sentiment* negatif pada pengguna media sosial *twitter* seperti ditunjukkan oleh *wordcloud* dengan kata-kata dominan antara lain: ujian, nasional, *link*, UN, lulus, sukses, nilai, amin, UN, kelas dan paket. Untuk *term* tertinggi pada *sentiment* negatif adalah term "Ujian" dan "Nasional". Kedua kata tersebut juga sangat sering disebut dalam cuitan pengguna *twitter* dengan sentimen negatif. Sedangkan kata-kata negatif yang mengarah pada *sentiment* negatif adalah *term* "link", "UN", "soal", "lebih", "tidak", "gak".

Tabel 4.10 *Sentiment* Negatif

Term	Frequency
Ujian	27317
Nasional	24383
Link	4783
Un	2719
Soal	2169
Lebih	2119
Tidak	2114
Belajar	2065
Jam	1859
Gak	1688

4.3 Klasifikasi

Tujuan riset ini adalah untuk menganalisis sentimen publik terhadap ujian nasional dengan menggunakan *Support Vector Machine (SVM)*. Setelah proses *clustering*, analisis dilanjutkan dengan melakukan klasifikasi data *Tweet* dengan menggunakan *SVM*. Adapun alur risetnya adalah sebagai berikut:

Dari data mentah dibagi dua *dataset*-nya menjadi data *training* dan sampel. Untuk data *training* dibuat pemodelan, lalu pemodelan tersebut digunakan

ke data sample untuk menentukan labelnya, lalu untuk setiap data sampel dibandingkan label lama dan label baru. Kemudian langkah selanjutnya dibuat rata-rata untuk tingkat *error*-nya dan diubah ke nilai akurasi.

Script untuk klasifikasi menggunakan Metode Naive Bayes seperti ditunjukkan pada *script* di bawah ini:

```
import org.apache.spark.mllib.feature.HashingTF
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.mllib.classification.NaiveBayes

import scala.collection.mutable.ArrayBuffer
import scala.util.{Success, Try}

import org.apache.spark.sql.{Row, SparkSession}
import org.apache.spark.sql.types.{IntegerType, StringType,
StructField, StructType}

val tweetsDF =
sqlContext.read.json("/FileStore/tables/tweets/2_class_labeled_tweet.json")
val spark : SparkSession =
SparkSession.builder.master("local[*]").getOrCreate

val tweetsRDD = tweetsDF.rdd

val bagOfWord = tweetsRDD.map(
  row => {
    val label = row.getLong(0).toDouble
    val tweets = row.getString(1)
    (label, tweets.split(" ").toSeq)
  }
)

val splits = bagOfWord.randomSplit(Array(0.8, 0.2), seed = 11L)
val training = splits(0).cache()
val test = splits(1)

val hashingTF = new HashingTF(2000)

val training_labeled = training.map(
  t => (t._1, hashingTF.transform(t._2))
).map(
  x => new LabeledPoint((x._1).toDouble, x._2)
)

def time[R](block: => R): R={
  val t0 = System.nanoTime()
  val result = block
  val t1 = System.nanoTime()
  println("\n\nElapsed time: " + (t1 - t0)/1000000 + "ms")
  result
}
```



```

println("\n\n***** Training *****\n\n")
val model = time { NaiveBayes.train(training_labeled, 1.0) }

println("\n\n***** Testing *****\n\n")
var testing_labeled = test.map(
  t => (t._1, hashingTF.transform(t._2), t._2)
).map(
  x => (new LabeledPoint((x._1).toDouble, x._2), x._3)
)

val predictionAndLabel = time{
  testing_labeled.map(
    p => {
      val labeledPoint = p._1
      val text = p._2
      val features = labeledPoint.features
      val actual_label = labeledPoint.label
      val predicted_label = model.predict(features)
      (actual_label, predicted_label, text)
    }
  )
}

val accuracy = 1.0 * predictionAndLabel.filter(x => x._1 ==
x._2).count() / test.count()

println("Training and Testing Complete, accuracy is = " +
accuracy)
println("\nSome Predictions:\n")

predictionAndLabel.take(10).foreach(
  x => {
    println("-----")
    println("Text = " + x._3)
    println("Actual Label = " + (if (x._1 == 0) "positive"
else "negative"))
    println("Predicted Label = " + (if (x._2 == 0) "positive"
else "negative"))
    println("-----")
    println("\n\n")
  }
)

***** Training ***** Elapsed time: 2931ms
***** Testing ***** Elapsed time: 7ms Training and
Testing Complete, accuracy is = 0.8219299661126351 Some Predictions: --
-----

```

Hasil dari *running script* diatas dapat diketahui seperti pada tampilan berikut di bawah ini:

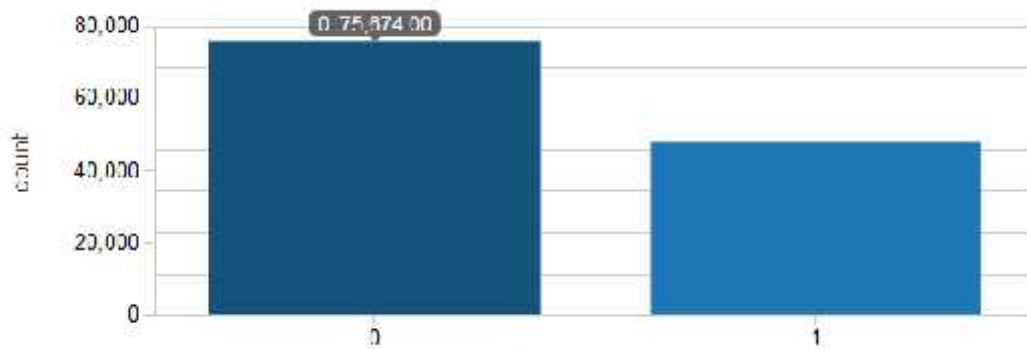
```

var allDataLabeled = bagOfWord.map(
  t => (t._1, hashingtf.transform(t._2), t._2)
).map(
  x => (new LabeledPoint((x._1).toDouble, x._2), x._3)
).map(
  p => {
    val labeledPoint = p._1
    val text = p._2
    val features = labeledPoint.features
    val predicted_label = model.predict(features)
    (predicted_label, text)
  }
)

var allDataLabelFrame = spark.createDataFrame(allDataLabeled.toDF("label", "tweet").groupBy("label").count())
display(allDataLabelFrame)

```

Berdasarkan analisis dengan metode *Naive Bayes* didapati hasil jumlah sentimen kelas positif adalah 75.674 dan jumlah kelas negatif adalah 47.964.



Gambar 4.14 Klasifikasi Ujian Nasional Menggunakan Metode *Naive Bayes*

Sedangkan untuk klasifikasi menggunakan *SVM* seperti tampak pada *script* di bawah ini:

```

import org.apache.spark.mllib.feature.HashingTF
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.mllib.classification.{SVMModel, SVMWithSGD}
import org.apache.spark.mllib.optimization._
import
org.apache.spark.mllib.evaluation.BinaryClassificationMetrics

import scala.collection.mutable.ArrayBuffer
import scala.util.{Success, Try}

import org.apache.spark.sql.{Row, SparkSession}
import org.apache.spark.sql.types.{IntegerType, StringType, StructField, StructType}

val tweetsDF =
sqlContext.read.json("/FileStore/tables/tweets/2_class_labeled_tweet.json")

```

```

val spark : SparkSession =
SparkSession.builder.master("local[*]").getOrCreate

val tweetsRDD = tweetsDF.rdd

val bagOfWord = tweetsRDD.map(
  row => {
    val label = row.getLong(0).toDouble
    val tweets = row.getString(1)
    (label, tweets.split(" ").toSeq)
  }
)

val splits = bagOfWord.randomSplit(Array(0.8, 0.2), seed =
11L)
val training = splits(0).cache()
val test = splits(1)

val hashingTF = new HashingTF(2000)

val training_labeled = training.map(
  t => (t._1, hashingTF.transform(t._2))
).map(
  x => new LabeledPoint((x._1).toDouble, x._2)
)

def time[R](block: => R): R={
  val t0 = System.nanoTime()
  val result = block
  val t1 = System.nanoTime()
  println("\n\nElapsed time: " + (t1 - t0)/1000000 + "ms")
  result
}

println("\n\n***** Training *****\n\n")
val svmAlg = new SVMWithSGD()
svmAlg.optimizer
  .setNumIterations(100)
  .setRegParam(0.1)
  .setUpdater (new SquaredL2Updater)
val model = time { svmAlg.run(training_labeled) }
model.setThreshold(0)

println("\n\n***** Testing *****\n\n")
var testing_labeled = test.map(
  t => (t._1, hashingTF.transform(t._2), t._2)
).map(
  x => (new LabeledPoint((x._1).toDouble, x._2), x._3)
)

val predictionAndLabel = time{
  testing_labeled.map(
    p => {
      val labeledPoint = p._1
      val text = p._2
      val features = labeledPoint.features
      val actual_label = labeledPoint.label
      val score = model.predict(features)
    }
  )
}

```

```

        var predicted_label = 0
        if(score > 0){
            predicted_label = 1
        }else{
            predicted_label = 0
        }
        (actual_label, predicted_label, text)
    }
}
)
}

val accuracy = 1.0 * predictionAndLabel.filter(x => x._1 ==
x._2).count() / test.count()

println("Training and Testing Complete, accuracy is = " +
accuracy)
println("\nSome Predictions:\n")

predictionAndLabel.take(10).foreach(
  x => {
    println("-----")
    println("Text = " + x._3)
    println("Actual Label = " + (if (x._1 == 0) "positive"
else "negative") )
    println("Predicted Label = " + (if (x._2 == 0) "Positive"
else "Negative" ) )
    println("-----")
    println("\n\n")
  }
)
)
***** Training ***** Elapsed time: 60698ms
***** Testing ***** Elapsed time: 7ms Training and
Testing Complete, accuracy is = 0.7886073906729062 Some Predictions: --
-----

```

Hasil dari *running script* diatas adalah:

```

databricks [2Class]SVMClassifier (Scala)

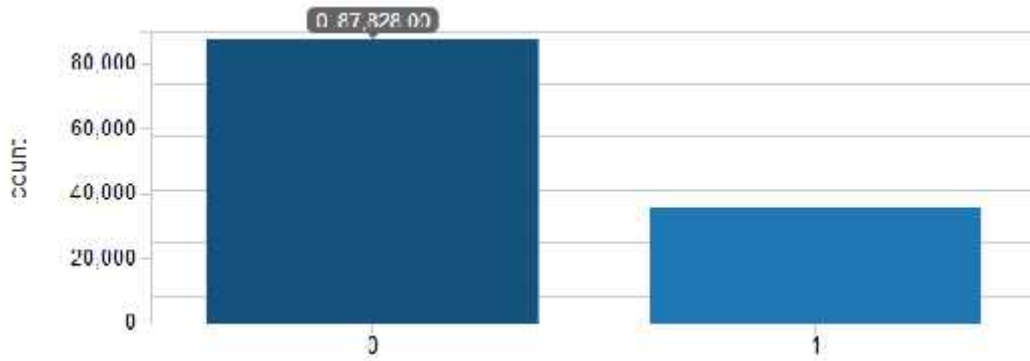
var allDataLabelTest = mapOfDataLabel.map(
  t => (t._1, hashingTF.transform(t._2), t._2)
).map(
  s => (new LabelledPoint((s._1).toFloat, s._2), s._3)
).map(
  p => {
    val label = p._1
    val text = p._2
    val features = LabelledPoint.features
    val predicted_label = model.predict(features)
    (predicted_label, text)
  }
)

var allDataLabelTrain = sparkContext.parallelize(allDataLabelTest).repartition(2).sortBy("label").count()

display(allDataLabelTrain)

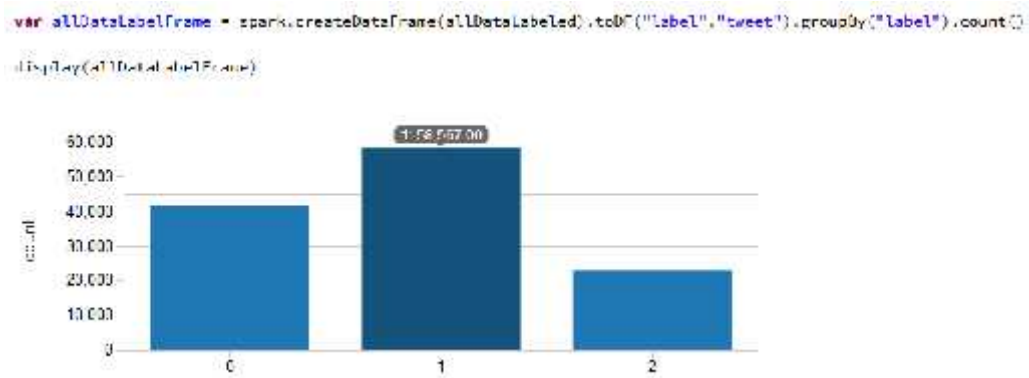
```

Berdasarkan analisis dengan metode *SVM* didapati hasil jumlah sentimen kelas positif adalah 87.828 dan jumlah kelas negatif adalah 35.800.



Gambar 4.15 Klasifikasi Ujian Nasional Menggunakan Metode *SVM*

Berdasarkan analisis dengan metode *SVM* didapati hasil jumlah kelas positif adalah 87.828 dan jumlah kelas negatif adalah 36.784.



Gambar 4.16 Klasifikasi Ujian Nasional Menggunakan *Logistic Regression*

Berdasarkan analisis dengan metode *Logistic Regression* didapati hasil jumlah sentimen kelas netral adalah 39.245, jumlah kelas positif adalah 58.567 dan jumlah kelas negatif adalah 26.800.

4.4 Evaluasi

Evaluasi klasifikasi dalam spark menggunakan *precision recall*. Tingkat akurasi diukur dengan *precision recall*. Pada penelitian ini mencoba membandingkan dua metode, yaitu metode *Logistic Regression* dan *SVM*. Sehingga pada penelitian ini menghasilkan hasil evaluasi sebagai berikut.

Tabel 4.11 Tingkat Akurasi Data Berdasarkan Rasio Perbandingan Menggunakan SVM

No	Metode	Rasio	Akurasi
1	Logistic Regression	60:40	0.84
		70:30	0.85
		80:20	0.856
2	SVM	60:40	0.90
		70:30	0.91
		80:20	0.91

Berdasarkan tabel 4.11 di atas, pada klasifikasi menggunakan *SVM* diperoleh tingkat akurasi 90 % untuk rasio data *training* dan data *testing* 60:40, lalu tingkat akurasi 91 % untuk rasio data *training* dan data *testing* 70:30 serta tingkat akurasi 91 % untuk rasio data *training* dan data *testing* 80:20. Sebagai pembandingan pada metode yang berbeda, peneliti mencoba membandingkan tingkat akurasi pada klasifikasi yang menggunakan metode *Logistic Regression*. Tingkat akurasi pada klasifikasi menggunakan *Logistic Regression* diperoleh tingkat akurasi 84 % untuk rasio data *training* dan data *testing* 60:40, lalu tingkat akurasi 85 % untuk rasio data *training* dan data *testing* 70:30 serta tingkat akurasi 86 % untuk rasio data *training* dan data *testing* 80:20. Fakta di atas semakin memperkuat referensi bahwa klasifikasi dengan menggunakan metode *SVM* adalah paling baik dibandingkan metode lainnya untuk klasifikasi *sentiment analysis*. Terbukti klasifikasi *sentiment analysis* pengguna media sosial *twitter* menggunakan *SVM* pada *tool analyzer Spark* memperoleh tingkat akurasi 90 %.

BAB 5

KESIMPULAN

Berdasarkan hasil penelitian diperoleh jumlah data yang sangat besar (124.612 *tweet*) dengan variasi kalimat yang sangat tinggi dan belum terkelompokkan. Untuk itu, dilakukan *clustering* data menggunakan Spark dengan metode *K-Means*. Hasil *clustering* diperoleh hasil terbentuk dan terdeteksi *elbow* optimal pada *elbow* 3 yang berarti bahwa terdapat 3 *cluster* untuk *clustering* pada $n=2$ sampai dengan $n=50$ *clustering* menggunakan metode *K-Means*. Sedangkan untuk pengklasifikasian data pada media sosial *twitter* menggunakan *Support Vector Machine (SVM)* dalam Spark. Pada klasifikasi data *tweet* menggunakan metode SVM diperoleh hasil jumlah sentimen kelas positif adalah 87.828 dan jumlah kelas negatif adalah 35.800. Berdasarkan analisis data diperoleh bahwa hasil penelitian ini diperoleh sentimen positif pengguna media sosial dalam pelaksanaan program ujian nasional dengan menggunakan metode *SVM*.

Untuk tingkat akurasi pada klasifikasi menggunakan *SVM* diperoleh tingkat akurasi 90 % untuk rasio *data training* dan *data testing* 60:40, lalu tingkat akurasi 91 % untuk rasio *data training* dan *data testing* 70:30 serta tingkat akurasi 91 % untuk rasio *data training* dan *data testing* 80:20. Dengan demikian, hasil riset ini semakin menegaskan bahwa klasifikasi pada *sentiment analysis* perilaku pengguna media sosial dengan menggunakan metode SVM menghasilkan tingkat akurasi yang lebih baik daripada menggunakan metode yang lain dalam *machine learning*.

“Halaman Ini Sengaja Dikosongkan”

DAFTAR PUSTAKA

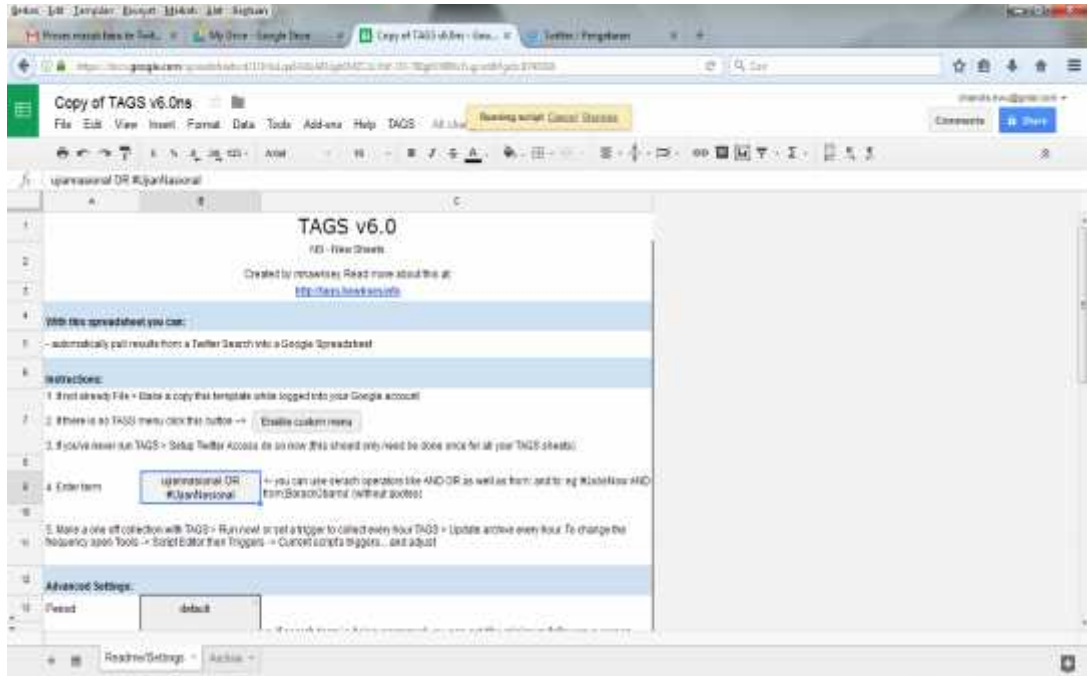
- Alhumoud, S., Albuhaire, T., and Altuwaijri, M., 2015, *Arabic Sentiment Analysis using WEKA a Hybrid Learning Approach*. 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), IEEE
- Alwi, Hasan, 2007, KBBI, edisi ketiga, Jakarta: Balai Pustaka
- APJII, 2014, *Profil Pengguna Internet di Indonesia 2014*, Puskakom UI, Jakarta
- B. Pang, L. Lee, 2008, "Opinion Mining and Sentiment Analysis, Foundations and Trends in Information Retrieval", vol. 2, No. 1–2, pp 1-135
- Barakbah, A.R., & Arai, K., 2004, *Determining Constraints of Moving Variance to Find Global Optimum and Make Automatic Clustering*, Industrial Electronics Seminar (IES) 2004. Surabaya
- Bholowalia, P., & Kumar, A., 2014, EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN
- Chen, M., Mao, S., & Liu, Y., 2014, *Big data: A survey*. Mobile Networks and Applications, 19 (2), 171-209
- Fauzi, M.A., 2018, Text Pre-Processing, Lecture UB, malifauzi.lecture.ub.ac.id/files/2018/08/Text-Pre-Processing-v2-1.pptx, diakses 25 Januari 2019
- Gemilang, H.T., et al., 2014, Indonesian president candidates 2014 sentiment analysis by using Twitter data, Proc. 2014 Int. Conf. ICT Smart Soc., "Smart Syst. Platf. Dev. City Soc. GoeSmart 2014", ICISS 2014, pp. 101-104
- Glass K and Colbaugh R, 2012, Estimating the sentiment of social media content for security informatics applications, Security Informatics 2012: 1(3)
- Habernal, I. e. (2014). *Supervised sentiment analysis in Czech social media* (Vol. 50). Plzen, Czech Republic: Information Processing and Management.
- Hossin, M. & Sulaiman, M.N., 2015, *A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS*, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.2, March 2015

- Jain, A.P., & Dandannavar, P., 2016, *Application of Machine Learning Techniques to Sentiment Analysis*, 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)
- Jianqiang, Z. and Xiaolin, G.U.I., 2017, Comparison Research on Text Preprocessing Methods on Twitter Sentiment Analysis, vol. 5, 2017
- Krouska, A., Troussas, C., and Virvou, M., 2016, "The effect of preprocessing techniques on Twitter sentiment analysis," pp 1-5
- Kurniawan, R.H., 2017, *Real Time Opinion Mining of Social Media about Indonesian Government Policy*, Tugas Akhir Sarjana Terapan Politeknik Elektronika Negeri Surabaya, Surabaya
- Lampe, C., Ellison, B., & Steinfield, C., 2008, *Changes in Use and Perception of Facebook*. Proceedings of the 2008 ACM conference on Computer supported cooperative work (pp. 721-730). New York: ACM
- Luthfi, K. d., 2009,. *Algoritma Data Mining*. Yogyakarta: Penerbit Andi (CV Andi Offset.
- Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., 2011, *Big data: The next frontier for innovation, competition, and productivity*, ... &McKinsey Global Institute
- Martineau, J. and Finin, T., 2009, Delta TFIDF: An Improved Feature Space for Sentiment Analysis, no. May 2009
- Medhat, W. Hassan, A. Korashy H. 2014. *Sentiment analysis algorithms and applications: A survey*, Ain Shams Engineering Journal (5). P. 1093–1113. [Online] Available from: <http://www.sciencedirect.com/science/article/pii/S2090447914000550>. [Accessed: 28 Aug 2015]
- Messias, J., et al., 2017, An Evaluation of Sentiment Analysis for Mobile Devices, Soc. Netw. Anal. Min., vol. 7, no. 1, p. 20
- Mihanovi , A., etal, 2014, *Big Data and Sentiment Analysis using KNIME: Online Reviews vs. Social Media*, MIPRO 2014, Opatija, Croatia
- Nasrullah, N., 2015, *Media Sosial Perspektif Komunikasi, Budaya, dan Siosioteknologi*, Cetakan Pertama, Simbiosis Rekatama Media

- Nomleni, P., et al., 2013, *Sentiment Analysis Berbasis Big Data*, Teknik Elektro ITS, Surabaya
- Pang, B., Lee, L., dan Vaithyanathan, S., 2002, *Thumbs up? Sentiment Classification using Machine Learning Techniques*, in *Proceedings of The ACL-02 Conference on Empirical Methods in Natural Language Processing*, Volume 10, Pp. 79-86, Morristown, NJ, USA
- Pak, A., dan Paurobek, P., 2010, *Twitter as a Corpus for Sentiment Analysis and Opinion Mining*, Universite de Paris-Sud, Laboratoire LIMSI-CNRS.
- Ray, S., 2017, *Understanding Support Vector Machine algorithm from examples (along with code)*,
<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>, Analytics Vidhya, diakses 26 Januari 2019
- Reddy, C.K., & Aggarwal, C.C., 2016, *Data Clustering*, Chapman and Hall/CRC Teknokeras. (2014, Oktober 28). *Kecerdasan Buatan (Artificial Intelligence) di Hadoop*. Retrieved from <https://openbigdata.wordpress.com>.
- Vongsingthong, S., Wisitpongphan, N., 2014, *Classification of University Students' Behaviors in Sharing Information on Facebook*, 11th International Joint Conference on Computer Science and Software Engineering (JCSSE).
- Wibowo, A., 2019, *10 Fold-Cross Validation*, MTI BINUS UNIVERSITY,
<https://mti.binus.ac.id/2017/11/24/10-fold-cross-validation>, diakses 26 Januari 2019
<http://greenmetric.ui.ac.id/overall-ranking-2015/>
<https://sites.google.com>
<http://www.merdeka.com/uang/di-5-media-sosial-ini-orang-indonesia-pengguna-terbesar-dunia.html>
<http://www.pewinternet.org/2015/01/09/social-media-update-2014/>
<http://mediaindonesia.com/news/read/72891/ujian-nasional-dikaji-ulang/2016-10-19#sthash.pX7UNA8M.dpuf>, diakses pada 16 Agustus 2017.
<http://mediaindonesia.com/news/read/72891/ujian-nasional-dikaji-ulang/2016-10-19#sthash.pX7UNA8M.dpuf>, diakses pada 16 Agustus 2017.

“Halaman Ini Sengaja Dikosongkan”

Lampiran 1. Proses *Crawling* Data pada *Twitter*



“Halaman Ini Sengaja Dikosongkan”

Lampiran 2: Tahapan Processing Data Menggunakan Spark

1. Tahapan 1: Create Cluster

Create Cluster

New Cluster

0 Workers: 0 GB Memory, 0 Cores, 0 CPUs
1 Driver: 60 GB Memory, 0.88 CPUs, 1 GBU

Cluster Name
Cluster Name

Databricks Runtime Version
3.5 LTS (includes Apache Spark 2.7.1, Scala 2.11)

Python Version
3

Instance

Free 60GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For more configuration options, please upgrade your Databricks subscription.

Instances **Spark**

Availability Zone
us-west-2c

2. Tahapan 2: Upload File JSON

Create Table (create)

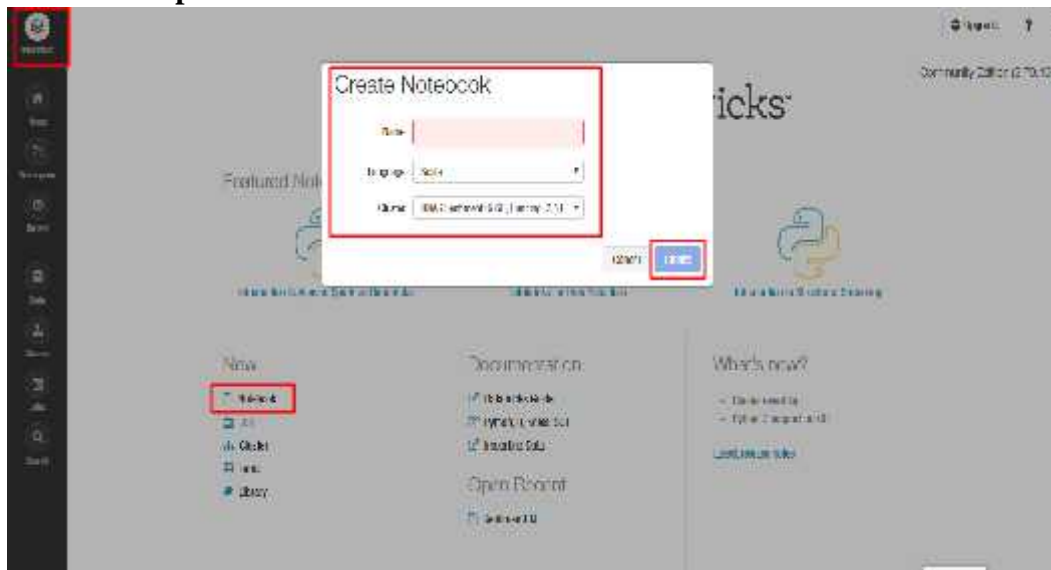
Create New Table

Data source
 S3 **S3** Spark Data Sources

Upload to DBFS
File Storage Location

File

3. Tahapan 3: Create Notebook



Lampiran 3. Program Evaluation Clasification MultiClass Using Bahasa Pemrograman Scala

```
package com.muhalbian.spark.jobs

import com.muhalbian.spark.util._

import org.apache.spark.{SparkContext, SparkConf}
import org.apache.spark.sql._

import org.apache.spark.mllib.feature.HashingTF
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.mllib.classification.{LogisticRegressionModel,
LogisticRegressionWithLBFGS}
import org.apache.spark.mllib.evaluation.MulticlassMetrics

import scala.collection.mutable.ArrayBuffer
import scala.util.{Success, Try}

import org.apache.spark.sql.{Row, SparkSession}
import org.apache.spark.sql.types.{IntegerType, StringType, StructField,
StructType}
import org.apache.spark.mllib.util.MLUtils

object Evaluation extends StreamUtils {

def main(args: Array[String]): Unit = {

val SparkSession = getSparkSession(args)
import SparkSession.implicits._

val tweetsDF = SparkSession.read.json("/home/blade1/Documents/spark-
sentiment-clustering/db/resentiment_agg.json")

val tweetsDFTesting = SparkSession.read.json("/home/blade1/Documents/spark-
sentiment-clustering/db/chandra_training_res_2.json")

//val spark : SparkSession = SparkSession.builder.master("local[*]").getOrCreate

val tweetsRDD = tweetsDF.rdd
val TestingRDD = tweetsDFTesting.rdd
```

```

//println(TestingRDD)
// tweetsRDD.collect().foreach(println)

val bagOfWord = tweetsRDD.map(
row => {
val label = row.getLong(0)
val tweets = row.getString(2)
(label, tweets.split(" ").toSeq)
}
)

val bagOfWordtest = TestingRDD.map(
row => {
val label = row.getLong(0)
val tweets = row.getString(1)
(label, tweets.split(" ").toSeq)
} )

val splits = bagOfWord.randomSplit(Array(0.8, 0.2), seed = 11L)
val training = splits(0).cache()
val test = bagOfWordtest

val hashingTF = new HashingTF(2000)

val training_labeled = training.map(
t => (t._1, hashingTF.transform(t._2))
).map(
x => new LabeledPoint((x._1).toDouble, x._2)
)

def time[R](block: => R): R={
val t0 = System.nanoTime()
val result = block
val t1 = System.nanoTime()

```

```

println("\n\nElapsed time: " + (t1 - t0)/1000000 + "ms")
result
}

println(training_labeled)
//training_labeled.collect().foreach(println)

println("\n\n***** Training *****\n\n")

// Run training algorithm to build the model
val model = new LogisticRegressionWithLBFGS()
.setNumClasses(3)
.run(training_labeled)

println("\n\n***** Testing *****\n\n")

val predictionAndLabels = test.map(
x => {
val prediction = model.predict(hashingTF.transform(x._2))
(prediction, x._1.toDouble)
}
)

println(predictionAndLabels)
//start evaluation with matric
// Instantiate metrics object
val metrics = new MulticlassMetrics(predictionAndLabels)

// Confusion matrix
println("Confusion matrix:")
println(metrics.confusionMatrix)

// Overall Statistics
val accuracy = metrics.accuracy
println("Summary Statistics")
println(s"Accuracy = $accuracy")

```

```

// Precision by label
val labels = metrics.labels
labels.foreach { l =>
println(s"Precision($l) = " + metrics.precision(l))
}

// Recall by label
labels.foreach { l =>
println(s"Recall($l) = " + metrics.recall(l))
}

// False positive rate by label
labels.foreach { l =>
println(s"FPR($l) = " + metrics.falsePositiveRate(l))
}

// F-measure by label
labels.foreach { l =>
println(s"F1-Score($l) = " + metrics.fMeasure(l))
}

// Weighted stats
println(s"Weighted precision: ${metrics.weightedPrecision}")
println(s"Weighted recall: ${metrics.weightedRecall}")
println(s"Weighted F1 score: ${metrics.weightedFMeasure}")
println(s"Weighted false positive rate: ${metrics.weightedFalsePositiveRate}")

//
// val accuracy = 1.0 * predictionAndLabels.filter(x => x._1 == x._2).count() /
test.count()

println("Training and Testing Complete, accuracy is = " + accuracy)
println("\nSome Predictions:\n")
} }

```

BIOGRAFI PENULIS

Nama : Chandra Eko Wahyudi Utomo
Tempat & Tanggal Lahir : Jember, 26 Juni 1979
Agama : Islam
Pekerjaan : Pranata Komputer di Universitas
Jember
Alamat Kantor : Jl. Kalimantan 37 Sumpersari Jember Jawa Timur
68121
Alamat Rumah : Jl. Pahlawan 234 Wuluhan Jember Jawa Timur
68162
E-mail : chandra.uptti@unej.ac.id



Riwayat Pendidikan

- TK Dharma Wanita Wuluhan, Jember 1983 - 1985
- MI Islamiyah Dukuhdempok Wuluhan, Jember 1985 - 1991
- SMP Negeri 1 Wuluhan, Jember 1991 - 1994
- SMA Negeri 2 Jember 1994 - 1997
- S1 Fisika FMIPA Universitas Brawijaya, Malang 1997 - 2004
- S2 Telematika Fakultas Teknik Elektro dan Informasi
Cerdas ITS, Surabaya 2015 - 2020
- Penerima Beasiswa BPPDN Kementrian Riset, Teknologi
dan Pendidikan Tinggi