LAPORAN HASIL PENELITIAN

PEMANFAATAN METODE KLASIFIKASI NAIVE BAYES UNTUK PENDETEKSI BERITA HOAX PADA ARTIKEL BERBAHASA INDONESIA

Oleh :

Soleman, SKom, M.Kom

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER
UNIVERSITAS BOROBUDUR
JAKARTA
2023

LAPORAN HASIL PENELITIAN

PEMANFAATAN METODE KLASIFIKASI *NAIVE BAYES* UNTUK PENDETEKSI BERITA *HOAX* PADA ARTIKEL BERBAHASA INDONESIA

Oleh :

Soleman, SKom, M.Kom

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER
UNIVERSITAS BOROBUDUR
JAKARTA
2023

LEMBAR IDENTITAS DAN PENGESAHAN
LAPORAN AKHIR PENELITIAN

1	Judul Penelitian	Pemanfaatan Metode Klasifikasi *Naive Bayes* Untuk Pendeteksi Berita *Hoax* Pada Artikel Berbahasa Indonesia
2	Ketua Peneliti :	
	a. Nama	Soleman, SKom, MKom
	b. NIDN	0303098501
	c. Jenis Kelamin	Laki-Laki
	d. Pangkat/Golongan/NIP	-
	e. Jabatan Fungsional	-
	f. Fakultas/Program Studi	Fakultas Ilmu Komputer/Sistem Informasi
	g. Bidang ilmu yang diteliti	Sistem Informasi
3	Jumlah Tim Peneliti	1 (satu) orang
4	Lokasi Penelitian	Jakarta
5	Jangka Waktu Penelitian	6 (enam) bulan
6	Biaya diperlukan	Rp. 10.000.000,-
7	Sumber Dana	Universitas Borobudur

Jakarta, 09 Januari 2023

Mengetahui
Fakultas Ilmu Komputer
Universitas Borobudur
Dekan

Djoko Harsono, SKom, MM, MKom

Ketua Peneliti

Soleman, SKom, MKom

Kepala Lembaga Penelitian dan Pengabdian Masyarakat
LPPM Universitas Borobudur

Dr. Evi Syafrida Nasution, S.Psi, M.Psi

ABSTRAK

PEMANFAATAN METODE KLASIFIKASI *NAÏVE BAYES* UNTUK PENDETEKSI BERITA *HOAX* PADA ARTIKEL BERBAHASA INDONESIA

Berita *hoax* sudah sangat banyak tersebar di internet. Kemudahan dalam membuat dan membagikan informasi merupakan salah satu faktornya. Berita hoax menjadi ancaman dan konsentrasi banyak pihak, muncul masalah dalam mengidentifikasi atau mengklasifikasi karena tidak ada pola yang dapat diidentifikasikan serta gaya penulisan bersifat bebas dan tidak kaku. Sistem pendeteksi sebelumnya belum ditemukannya metode dan atribut yang digunakan untuk klasifikasi berita *hoax* dengan akurasi yang akurat. Atas dasar itu penelitian ini dilakukan, seperti pada kebanyakan klasifikasi berita *hoax* yang dijadikan acuan yaitu dilakukan praproses (*case folding*, *tokenisasi, stemming* dan *stopword removal*), ekstrasi fitur dan penambahan atribut selain dari praproses artikel seperti *website* tempat artikel di publikasi dan status *website* tersebut. Hasil dari penelitian ini didapatkan akurasi sebesar 72% yang ternyata terjadi penurunan 6.6% dibandingkan dengan penelitian sebelumnya yang sebesar 78.6% dikarenakan satu *website* yang hanya mempublikasi satu artikel hoax dan dibiarkan domain *website* tersebut *expired*, dengan begitu terjadi pengurangan terhadap bobot nilai klasifikasi.

Kata kunci: Artikel *Hoax*, klasifikasi Teks*, Naïve Bayes*, Analisa teks, *Union Feature.*

ABSTRACT

UTILIZATION OF NAÏVE BAYES CLASSIFICATION METHOD FOR HOAX NEWS DETECTION IN INDONESIAN LANGUAGE ARTICLES

Hoax news has spread a lot on the internet. The ease of being creative and sharing information is one of the factors. Hoax news is a threat and concentration for many parties, problems arise in identifying or classifying it because there is no identifiable pattern and the writing style is free and not rigid. The previous detection system had not found the methods and attributes used to classify hoax news accurately. It is on this basis that this research was conducted, because most of the hoax news classifications are used as a reference, namely pre-processing (case folding, tokenization, stemming and stopword removal), feature extraction and addition of attributes in addition to pre-processing. articles such as the website where the article was published and the status of the website. . The results of this study obtained an accuracy of 72%, in fact it decreased by 6.6% compared to the previous research which amounted to 78.6% because one website only published one hoax article and allowed the website's domain to expire, resulting in a reduction in the weight of the classification value.

Keywords: Hoax Articles, Text classification, Naïve Bayes, Text analysis, Union Feature.

KATA PENGANTAR

Puji dan Syukur yang sedalam-dalamnya kepada Tuhan Yang Maha Esa, karena hanya dengan rahmat dan karunia-Nya-lah penelitian yang berjudul "Pemanfaatan Metode Klasifikasi *Naïve Bayes* untuk Pendeteksi Berita *Hoax* pada Artikel Berbahasa Indonesia" dapat diselesaikan.

Dalam penyusunan penelitian ini penulis menyampaikan terima kasih yang tulus kepada:

1. Tuhan Yang Maha Esa atas segala petunjuk dan kemudahan-Nya sehingga pada akhirnya penulis dapat menyelesaikan penelitian ini.
2. Prof. Ir Bambang Bernanthos, Msc., selaku Rektor Universitas Borobudur Jakarta.
3. Djoko Harsono.S.Kom., MM., M.Kom. selaku Dekan Fakultas Ilmu Komputer Universitas Borobudur Jakarta.
4. Mansuri S.Kom, M.M.S.I selaku Kaprodi Sistem Informasi Fakultas Ilmu Komputer Universitas Borobudur Jakarta.
5. Ratih Widayanti Kosaman, S.Kom., M.Kom. selaku Sekretaris Fakultas Ilmu Komputer Universitas Borobudur Jakarta.
6. Seluruh staff Fakultas Komputer Universitas Borobudur yang telah banyak memberikan sumbangsih baik tenaga maupun pikiran.
7. Seluruh pihak yang tidak dapat peneliti sebutkan satu persatu, namun telah banyak terlibat membantu proses penelitian ini.

Penulis menyadari bahwa penulisan penelitian ini jauh dari sempurna. Untuk itu penulis berharap mendapatkan kritik dan saran yang dapat berguna untuk membangun demi kesempurnaan tulisan Penelitian ini.

Jakarta, 09 Januari 2023

Soleman, S.Kom., M.Kom.

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR ISI

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pengguna layanan *internet* semakin hari semakin meningkat. Banyak yang memanfaatkan *internet* dalam kehidupan sehari-hari untuk mengisi waktu luang, untuk melakukan pekerjaan dan juga untuk mengetahui informasi terbaru. Menurut hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) setelah melakukan survei penetrasi dan perilaku pengguna *internet* di Indonesia, Jumlah pengguna *internet* pada tahun 2016 adalah 132,7 juta jiwa dan Jumlah pengguna *internet* pada tahun 2017 telah mencapai 143,26 juta jiwa atau setara dengan 54,68% dari total jumlah penduduk Indonesia. Jumlah tersebut menunjukan kenaikan sebesar 10,56 juta jiwa dari hasil survei pada tahun 2016. Tingginya penetrasi ini dilihat dari ketersediaan *fiber optic* dan infrastruktur pendukung lainnya yang menopang aktivitas berinternet pada tahun 2017. Pengguna layanan internet mencakup semua kalangan dari yang muda sampai dengan yang tua, pria dan wanita(APJII, 2017).

Sosiasi Penyelenggara Jasa Internet Indonesia (APJII) mengumumkan hasil survei pengguna internet di Indonesia periode 2019-kuartal II 2020 secara daring pada Senin (9/11) siang. Hasilnya, jumlah pengguna internet di Indonesia hingga kuartal II tahun ini naik menjadi 73,7 persen dari populasi atau setara 196,7 juta pengguna (APJII, 2020). Pesatnya perkembangan tersebut tidak hanya berdampak positif, terdapat juga dampak negatif yang dihasikan. Setiap informasi yang beredar tanpa melewati penyutingan serta validasi kebenaran yang tidak jelas.

Fenomena ini sering dimanfaatkan untuk mencari keuntungan dari menyebarkan informasi hoax. Menurut Presiden Direktur VIVA Media Group, Anindya Novyan Bakrie, Persentase berita hoax di media sosial mencapai 92,40%, disusul aplikasi chatting 62,80%, lalu website 34,90%, sementara untuk media yang sudah kurang diminati seperti televisi hanya 8.70%, media cetak 5%, email 3,10% dan radio 1,20% (Tim VIVA, 2018). Dengan menggunakan dataset yang sama pada penelitian sebelumnya, penulis ingin meneliti dengan menambahkan 2 atribut dalam klasifikasi apakah akan menambah akurasi dalam mendeteksi berita hoax.

Pesatnya perkembangan tersebut tidak hanya berdampak positif, terdapat juga dampak negatif yang dihasikan. Setiap orang bisa dengan mudah memproduksi informasi dan langsung disebarkan di *internet*. Setiap informasi yang beredar tanpa melewati penyutingan serta validasi kebenaran yang tidak jelas. Fenomena ini sering dimanfaatkan untuk mencari keuntungan dari menyebarkan informasi hoax. Setiap ada topik yang sedang *viral* di *internet* maka ada juga informasi *hoax* yang beredar mengikuti topik tersebut. Artikel *hoax* dipublikasi di *website* dengan *domain .com* atau *.net* yang bisa membuat pembaca yakin jika berita tersebut adalah berita sebenarnya.

Tujuan dari berita *hoax* adalah untuk mempengaruhi opini dan pandangan publik dalam suatu hal tertentu. Sebagai contoh dalam politik, berita *hoax* dimanfaatkan untuk menjatuhkan lawan politiknya atau berita *hoax* tentang gempa bumi dan tsunami yang dapat menimbulkan kegelisahan masyarakat setempat. Berita *hoax* sangatlah mengganggu ketentraman masyarakat disegala lapisan, tidak

mengenal usia, jenis kelamin, bahkan tingkat pendidikan. Berita *hoax* dengan mudah tersebar dikarenakan perilaku masyarakat yang bangga menjadi orang pertama kali yang menyebarkan, suka berbagi tetapi malas membaca, gemar mencari sensasi, tidak tahu itu *hoax* dikarenakan tidak mencari tahu kebenarannya terlebih dahulu atau minimnya informasi yang benar dan juga ada yang hanya mengikuti tren. Terkadang ada berita *hoax* di media sosial yang mempunyai *website*, dengan begitu tingkat kepercayaan publik terhadap berita tersebut menjadi lebih tinggi. Karena judul sering provokatif, bombastis dan heboh, pengguna terpikat dan dengan mudah membagikannya ke akun pribadi dan dibaca oleh akun-akun yang terhubung dengannya.

Masyarakat Telematika Indonesia (Mastel) melakukan survei pada 13 Februari 2017 di Jakarta dengan proses survei dilakukan secara *online* dan direspon oleh 1.116 responden dengan rentang usia 25-40 tahun 47,80%, di atas 40 tahun 25,70%, 20-24 tahun 18,40%, 16-19 tahun 7,70% dan di bawah 15 tahun 0,40%. Jumlah responden ini didapatkan dalam waktu 48 jam sejak pertama kali disebarkan ke publik pada 7 Februari 2017. Sebanyak 90,30% responden menjawab bahwa *hoax* adalah berita bohong yang disengaja, 61,60% mengatakan kalau *hoax* adalah berita yang menghasut, 59% berpendapat *hoax* adalah berita yang tidak akurat, 14% menjawab *hoax* sebagai berita ramalan atau fiksi ilmiah, 12% mengatakan *hoax* adalah berita yang menyudutkan pemerintah dan 3% menjawab berita yang tidak saya sukai. Hanya 0,60% responden menjawab tidak tahu. Ketidak jelasan sumber berita 54,10% membuat 83,20% responden langsung memeriksa kebenaran dari berita tersebut atau langsung menghapus dan mendiamkan 15,90%. Hanya 1% dari responden yang menyatakan bahwa mereka langsung meneruskan berita dimaksud (Mastel, 2017). World Wide Web telah menjadi kumpulan besar dokumen dan jumlah dokumen yang tersedia meningkat setiap hari. Menjalankan Naïve Bayes dengan validasi silang 10 kali lipat pada data web yang dipilih memberikan 77% instans yang diklasifikasikan dengan benar dalam nol detik dengan kesalahan absolut relatif 68,9937%. Hal ini menunjukkan kemampuan algoritma Naïve Bayes untuk secara akurat mengklasifikasikan sejumlah besar dokumen web dalam waktu singkat(A. B. Adetunji1, J. P. Oguntoye, O. D. Fenwa1 and N. O. Akande, 2018). Analisis sentimen terhadap berita yang ada di media sosial twitter menggunakan metode naïve Bayes classifier dengan mengklasifikasikan sentimen menjadi positif, dan negatif digunakan untuk melihat bagaimana masyarakat Indonesia khususnya pada media sosial twitter(L. A. Waskito, K. M. Lhaksmana dan D. T. Murdiansyah, 2019).

Mendeteksi berita *hoax* tidaklah mudah, informasi dicampur dan diolah sedemikian rupa sehingga membuat pembaca terkecoh serta dapat membangkitkan kesan sebagai kebenaran baru dan semua orang harus tahu. Perbedaan berita *hoax* dan berita sebenarnya juga sangat sedikit, berita *hoax* merupakan berita sebenarnya yang diubah dengan menambahkan atau mengurangkan kata-kata sehingga menghasilkan makna yang berbeda dengan berita aslinya.

Pada penelitian sebelumnya telah diperoleh hasil akurasi yang cukup tinggi, rata-rata hasil akurasi sekitar 78.6% dengan metode *naïve bayes* untuk klasifikasi artikel (Pratiwi et al., 2017) Dengan menggunakan *dataset* yang sama pada

penelitian sebelumnya, penulis ingin meneliti dengan menambahkan 2 atribut dalam klasifikasi apakah akurasi lebih maksimal dalam mendeteksi berita *hoax*.

1.2. Masalah Penelitian

1.2.1. Identifikasi Masalah

Berdasarkan Latar belakang tersebut, maka permasalahan yang dapat di identifikasi dalam penelitian ini yaitu:

1. Sistem pendeteksi berita *hoax* kurang maksimal dengan hasil proses klasifikasi menghasilkan akurasi 78.6%, sehingga nilai bobot klasifikasi tinggi.
2. Belum ditemukannya kombinasi atribut yang sesuai untuk klasifikasi berita *hoax* dengan metode *naïve bayes*, sehingga berita hoax kurang akurat.

1.2.2. Batasan Masalah

Dalam penelitian ini penulis dalam mendeteksi berita *hoax* dengan klasifikasi menggunakan metode *naïve bayes*, dari sekian banyak masalah yang terdapat pada topik berita *hoax* maka penulis membatasi penelitian ini ialah sebagai berikut:

1. Penelitian ini berfokus pada membangun model penciri artikel berita *hoax* yang dapat digunakan untuk mendeteksi berita *hoax*.
2. Penelitian ini hanya meneliti *text* dalam artikel, tidak termasuk gambar dan judul sebuah berita.
3. Penelitian ini hanya meneliti metode-metode yang dipakai penelitian sebelumnya dan usulan algoritma dalam penelitian ini.
4. *Dataset* yang digunakan adalah *dataset* penelitian sebelumnya.
5. Topik yang diambil menyesuaikan yang terdapat dalam *dataset*.
6. Jumlah *data* yang digunakan mengikuti jumlah *data* pada *dataset*
7. Artikel yang terdapat pada *dataset* adalah artikel berbahasa Indonesia.

1.2.3. Rumusan Masalah

Berdasarkan hasil uraian identifikasi masalah, penulis menetapkan rumusan masalah sebagai berikut:

1. Menggunakan metode apa yang tepat untuk mendeteksi berita *hoax* sehingga menurunnya terhadap bobot nilai klasifikasi ?
2. Kombinasi atribut atau fitur yang bagaimana sehingga mampu meningkatkan keakuratan berita *hoax*?

1.3. Tujuan dan Manfaat Penelitian

1.3.1. Tujuan Penelitian

Tujuan dari penelitian dilakukan ini adalah :

1. Mengembangkan sistem mendeteksi berita *hoax* yang lebih baik atau akurat dengan metode *naïve bayes*.
2. Rancang model klasifikasi berita *hoax* dengan metode *naïve bayes* dan penambahan atribut yang digunakan untuk meningkatkan keakuratan berita *hoax* .

1.3.2. Manfaat Penelitian

Manfaat dari penelitian ini diharapkan dapat :

1. Mencegah konflik yang akan terjadi akibat dari berita *hoax*.
2. Mencegah keresahan masyarakat akibat dari berita *hoax*.
3. Mencegah kerugian pihak tertentu akibat dari berita *hoax*.
4. Mempermudah mengindentifikasi berita hoax apakah suatu artikel *hoax* atau bukan.
5. Meminimalisir penyebaran berita *hoax*.

1.4. Tata Urut Penulisan

BAB I PENDAHULUAN

Menjelaskan secara singkat uraian latar belakang penelitian, masalah yang terdapat ketika penelitian, tujuan dan manfaat dari penelitian yang dilakukan, tata urut penulisan dan daftar pengertian.

BAB II LANDASAN TEORI DAN KERANGKA KONSEP

Menjelaskan secara singkat uraian tentang tinjauan pustaka, tinjauan studi, dan tinjauan objek penelitian yang dijadikan dasar perancangan, serta terdapat pola pikir atau kerangka konsep penulis dalam memecahkan masalah yang ada sehingga dapat disimpulkan ke dalam HipoPenelitian.

BAB III METODOLOGI DAN RANCANGAN PENELITIAN

Meliputi tentang metode dalam penelitian, metode dalam pengumpulan data, teknik instrumentasi, teknik analisis, perancangan, pengujian prototipe, dan langkah-langkah penelitian

BAB IV PENUTUP

Meliputi lembar yang berisi tentang inti dari kesimpulan penulisan Penelitian dan lembar yang berisi tentang saran yang ditujukan kepada peneliti atau pengembang aplikasi berikutnya sehingga dapat meneruskan keterbatasan dalam penelitian saat ini.

1.5. Daftar Pengertian

Penelitian ini mempunyai beberapa istilah yang sering digunakan untuk menjelaskan teknologi yang digunakan, yaitu :

Hoax	:	Menurut Kamus Besar Bahasa Indonesia, 'hoaks' adalah 'berita bohong.' Dalam *Oxford English dictionary*, *'hoax'* didefinisikan sebagai *'malicious deception'* atau 'kebohongan yang dibuat dengan tujuan jahat'.
Naïve Bayes	:	Algoritma atau metode yang di gunakan untuk klasifikasi.
LSTM	:	Algoritma atau metode klasifikasi dari *neural network* yang di gunakan pada penelitian ini.

BAB II
LANDASAN TEORI DAN KERANGKA PEMIKIRAN

2.1.Tinjauan Pustaka

2.1.1. Pengertian *Hoax*.

Berita bohong atau yang lebih dikenal dengan sebutan hoaks (hoax; hocus to trick) didefinisikan sebagai kebohongan yang dibuat secara sengaja untuk menyamarkan kebenaran yang ada untuk melakukan ancaman atau penipuan (Prasetijo, A. B. Isnanto, R. R. Eridani, D. Soetrisno, Y. A.D. Arfan, M. Sofwan dan Aghus, 2018). Dalam KBBI daring, hoaks diartikan: (1) tidak benar, bohong (tentang berita, pesan, dan sebagainya); (2) berita bohong(K. Kominfo, 2022). Dalam perkembangannya, hoaks dapat diartikan sebagai kabar palsu yang sengaja disebar untuk mencari kehebohan publik.

Pengertian *hoax* (hoaks) adalah informasi palsu atau berita yang sebenarnya bisa berisi fakta namun telah dipelintir atau direkayasa. Perkembangan kata *hoax* dari bentuk-bentuk sebelumnya dapat ditelusuri dalam buku "*A Glossary: Or, Collection of Words, Phrases, Names dan Allusions to Customs*", karangan Robert Nares yang terbit pada 1822 di London dimana kata *hoax* mulai dipakai di Inggris pada abad ke-18. Robert Nares menulis bahwa *hoax* berasal dari *hocus*, sebuah kata Latin yang merujuk pada *hocus pocus*. Pada lema (kata atau frasa yang masukan dalam kamus berikut keterangan ringkas) kata *hocus*, Nares menambahkan arti "to cheat" atau "menipu".

2.1.2. Kecerdasan Buatan (*Artificial Intelligence*).

Kecerdasan buatan biasanya dilakukan dengan mengikuti atau meniru karakteristik dan analogi pemikiran kecerdasan manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. *Artificial Intelligence* adalah salah satu cabang ilmu yang berhubungan dengan penggunaan mesin untuk memecahkan masalah yang rumit(E. Kusrini dan L. Taufiq, 2009).

Saat pertama kali ditemukan komputer hanya dipergunakan untuk mengerjakan proses perhitungan. Komputer kemudian berkembang dengan pesat bahkan pada saat ini manusia tidak bisa lepas dari peran dari sebuah komputer. Komputer tidak lagi hanya sebatas alat hitung namun komputer dapat dikembangkan dan diberdayakan untuk menggantikan peran manusia untuk mengerjakan apa yg biasa dikerjakan manusia bahkan performanya bisa melampaui manusia. Dari perkembangan fungsi dari sebuah komputer tersebut maka bisa dikatakan bahwa kecerdasan buatan adalah suatu ilmu komputer yang secara khusus mempelajari bagaimana supaya mesin komputer dapat mempunyai keahlian yang dapat mengerjakan pekerjaan manusia seperti dan sebagus apabila dikerjakan manusia.

Dengan Kemampuan menalar dan pengetahuan yang diberikan kepada komputer maka komputer dapat bertindak sebaik dan selayaknya manusia. Agar komputer menjadi pintar maka kecerdasan buatan (AI) memberikan dua komponen tersebut.

Kecerdasan buatan dapat dilihat dari beberapa sudut pandang yaitu:

1. Dari sudut pandang kecerdasan, kecerdasan buatan adalah bagaimana menjadikan mesin dapat mengerjakan suatu pekerjaan yang awalnya hanya bisa dikerjakan oleh manusia dan membuatnya menjadi pintar.
2. Dari sudut pandang bisnis, Kecerdasan buatan adalah kumpulan dari beberapa alat bantu (*tools*) yang praktis dan untuk menyelesaikan masalah bisnis dengan menggunakan alat-alat bantu tersebut.
3. Dari sudut pandang pemrograman, Kecerdasan buatan adalah studi mengenai pemograman yang meliputi proses pencarian (*search*), simbolik dan pemecahan masalah.
4. Dari sudut pandang penelitian
    a. Menciptakan aplikasi permainan catur, *general problem solving* dan membuktikan suatu teori adalah awal dari penelitian mengenai kecerdasan buatan yang dimulai pada awal tahun 1960-an.
    b. Artificial intelligence adalah nama pada akar dari studi area.

a. Konsep Kecerdasan Buatan.

Dalam kecerdasan buatan terdapat beberapa konsep yang harus dipahami yaitu: (E. Kusrini dan L. Taufiq, 2009):

1. *Turing Test* – Metode Pengujian Kecerdasan
2. Nama *turing test* diambil dari pembuatnya yaitu Alan turing, turing test adalah sebuah metode untuk menguji kecerdasan. Dua objek yang ditanyai dan seorang penanya (manusia) dilibatkan dalam proses uji ini. Yang satu adalah mesin yang akan ditanyai dan yang satunya lagi adalah seorang manusia.Penanya tidak bisa melihat langsung terhadap objek yang akan ditanyai. Membedakan jawaban antara mana jawaban manusia dan mana jawaban komputer berdasarkan jawaban kedua objek tersebut adalah tugas dari penanya. Apabila jawaban komputer dan jawaban manusia tidak dapat dibedakan oleh penanya maka dapat ditarik kesimpulan bahwa CERDAS.
3. Pemrosesan Simbolik
   Mengerjakan proses nonalgoritmik dan simbolik dalam suatu penyelesaian masalah adalah sifat penting dari kecerdasan buatan.
   Tidak berdasarkan pada komputasi matematika atau mengacu kepada rumus adalah kecenderungan manusia dalam menyelesaikan masalah karena manusia lebih bersifat simbolik. Sementara komputer semula diciptakan untuk memproses suatu bilangan atau angka-angka.Kecerdasan buatan merupakan cabang ilmu komputer yang mengerjakan proses nonalgoritmik dan simbolik dalam suatu penyelesaian masalah adalah sifat penting dari kecerdasan buatan.
4. *Heuristic*
   Istilah *heuristic* berasal dari bahasa Yunani yang mempunyai makna menemukan. *Heuristic* adalah cara untuk melakukan proses pencarian (*search*) secara selektif ruang problem, dan memandu proses pencarian yang kita lakukan disepanjang jalur yang memiliki kemungkinan sukses paling besar.

5. Penarikan Kesimpulan
   Kemampuan mempertimbangkan (*reasoning*) atau kemampuan berfikir dicoba dibuat oleh kecerdasan buatan atau AI. Kemampuan berfikir (*reasoning*) termasuk di dalamnya penarikan kesimpulan (*inferencing*) berdasarkan aturan dengan memakai metode *heuristic* atau pencarian lainnya dan berdasarkan fakta-fakta .
6. Pencocokan Pola (*Pattern Matching*)
   Cara kerja kecerdasan buatan adalah dengan menggunakan metode pencocokan pola (*pattern matching*) yang berusaha untuk menjelaskan kejadian (*event*), proses atau objek, dalam hubungan logika atau komputasional.

b. Bahasa Alami

Bentuk representasi dari suatu pesan yang akan dikomunikasikan antara manusia adalah pengertian bahasa alami. Bentuk representasi utamanya adalah berupa ucapan/suara (*spoken language*), akan tetapi sering dinyatakan dalam bentuk tulisan.

Jenis bahasa dapat dipisahkan menjadi (1) bahasa buatan dan (2) bahasa alami. Bahasa buatan merupakan bahasa yang diciptakan secara spesifik untuk memecahkan kebutuhan tertentu, seperti bahasa pemograman komputer atau bahasa pemodelan. Bahasa alami adalah bahasa untuk berkomunikasi sesama manusia yang biasa digunakan, seperti bahasa Inggis, Indonesia, jawa dan sebagainya (Sarkar, 2016).

Pertama kali orang merepresentasikan bahasa untuk rangkaian simbol adalah Chomsky. Dia berhasil membuktikan atau menunjukan bahwa segala sesuatu dapat direpresentasikan memakai cara yang lebih umum atau universal. Dari hasil Chomsky yang mengatur susunan simbol-simbol dan merepresentasikan bahasa sebagai sekumpulan simbol-simbol tersebut maka peluang pemrosesan bahasa secara simbolik dengan komputer peluangnya terbuka, sehingga mencetuskan cabang ilmu baru yaitu *Natural Language Processing* (NLP).

Leksikon dan perbendaharaan kata adalah pembahasan salah satu bidang ilmu *linguistic*. *Linguistic* sendiri adalah cabang ilmu komputer yang secara spesifik mengkaji bagaimana suatu bahasa itu distrukturkan. Leksikon adalah kamus yang mendaftar kata-kata bahasa itu secara alfabet. Perbendaharaan kata adalah sekumpulan kata-kata dan frase-frase yang digunakan dalam bahasa tertentu. Sebagai bagian dari pengkajian bahasa, linguis mendefinisikan semua kata-kata dan frase-frase yang digunakan secara umum kemudian mengorganisasikannya ke dalam sebuah leksikon.

c. Pengolahan Bahasa Alami (*Natural Language Processing*).

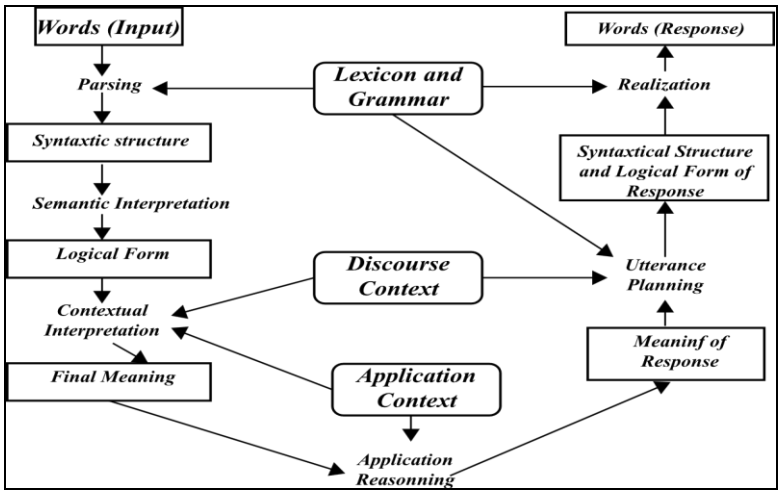
Pemrosesan Bahasa Alami tidak bertujuan untuk mengubah bahasa yang diterima dalam bentuk suara ke dalam data digital dan / atau sebaliknya, tetapi lebih bertujuan untuk memahami makna dari teks yang diberikan dalam format bahasa alami dan merespon dengan tepat, misalnya dengan melakukan tindakan spesifik

atau menampilkan data tertentu. Natural Language Processing (NLP) adalah salah satu bidang kecerdasan buatan (*Artificial Intelligence*) yang mempelajari komunikasi antara manusia dan komputer.

Teknik NLP memungkinkan komputer untuk memproses dan memahami bahasa alami manusia dan memanfaatkannya lebih lanjut untuk memberikan hasil yang bermanfaat. Hal ini membuat NLP berkaitan dengan area *Human-Computer Interaction* (HCI). Hal tersebut terutama berkaitan dengan merancang dan membangun aplikasi dan sistem yang memungkinkan interaksi antara mesin dan bahasa alami dan memanfaatkannya lebih lanjut agar dapat digunakan oleh manusia. NLP didefinisikan sebagai bidang khusus ilmu komputer dan teknik dan kecerdasan buatan dengan akar dalam linguistik komputasional (Sarkar, 2016).

Ketika komputer telah memahami ucapan yang diberikan pengguna, maka komputer dapat melakukan hal-hal yang diharapkan pengguna / tanggapan kembali diungkapkan atau diungkapkan dalam bahasa alami juga. Pelacakan klasik dan teknik pencocokan pola digunakan bersama dengan basis pengetahuan sehingga komputer dapat memahami apa yang dimasukkan pengguna dalam bahasa alami. Agar komputer memahami pertanyaan dalam bahasa alami, komputer harus memiliki pengetahuan analitis dan interpretasi masukan dalam data *knowledgable*-nya. Dalam hubungan ini teknik AI digunakan untuk menampilkan pengetahuan internal dan masukan proses. Komputer harus memahami gramatika dan definisi kata-kata. Untuk mencapai tujuan tersebut dibutuhkan tiga tahap proses. Proses yang pertama adalah parsing atau analisa sintaksis yang memeriksa kebenaran struktur kalimat berdasarkan suatu grammar (tata bahasa) dan *lexicon* (kosa kata) tertentu.

Proses kedua adalah semantic interpretation atau interpretasi semantik yang bertujuan untuk merepresentasikan arti dari kalimat secara contextindependent untuk keperluan lebih lanjut. Sedangkan proses ketiga adalah contextual interpretation atau interpretasi kontekstual yang bertujuan untuk merepresentasikan arti secara *context-dependent* dan menentukan maksud dari penggunaan kalimat. Gambaran sebuah organisasi sistem NLP dapat dilihat pada gambar 1. dibawah ini:



Gambar 1. Gambaran umum sistem NLP

Jenis aplikasi yang bisa dibuat pada bidang bidang *natural language* adalah *text - based application* dan *dialogue-based applications*.

1. *Text - based application*.
   Meliputi berbagai aplikasi yang memproses teks tertulis seperti misalnya buku, berita di surat kabar, *e-mail* dan lain sebagainya. Contoh penggunaan dari *text-based* application ini adalah :
   a. mencari topik tertentu dari buku yang ada pada perpustakaan
   b. memberikan respon atas input yang diberikan
   c. mencari isi dari surat atau *e-mail*
   d. menterjemahkan dokumen dari satu bahasa ke bahasa yang lain
2. *Dialogue-based application*
   Idealnya pedekatan ini melibatkan bahasa lisan atau pengenalan suara,akan tetapi bidang ini juga memasukkan interaksi dengan cara memasukkan teks pertanyaan melalui *keyboard*. Aplikasi yang sering ditemui untuk bidang ini adalah :
   a. sistem tanya jawab, dimana natural language digunakan dalam mendapatkan informasi dari suatu database.
   b. sistem otomatis pelayanan melalui telepon
   c. kontrol suara pada perlatan sistem sistem *problem solving* yang membantu untuk melakukan penyelesaian masalah yang umum dihadapi dalam suatu pekerjaan.

2.1.3. *Scanner* (Analisis Leksikal)

Analisis Leksikal (*Scanner*) merupakan antarmuka antara kode program sumber dan analisa sintaktik (*parser*). Atau dalam pengertiannya adalah sebuah proses yang mendahului parsing sebuah rangkaian karakter. Scanner melakukan pemeriksaan karakter per karakter pada teks masukan, memecah sumber *program* menjadi bagian-bagian yang disebut *Token*. Proses *parsing* akan lebih mudah dilakukan bila *input*nya sudah berupa *token*.Analisis leksikal membuat pekerjaan membuat sebuah *parser* jadi lebih mudah daripada membangun nama setiap fungsi dan variabel dari karakter-karakter yang menyusunnya, dengan analisis leksikal parser cukup hanya berurusan dengan sekumpulan *token* dan nilai sintaksis masing-masing (F. S. Radjatadoe, 2012).

Terlepas dari efisiensi pemrograman yang dapat dicapai dengan penggunaannya, proses kerja analisis leksikal yang membaca lebih dari sekali setiap karakter dari input yang diberikan menjadikan penganalisa leksikal sebagai sub-sistem yang paling intensif melakukan komputasi, terutama bila digunakan dalam sebuah kompilator. Kompilator adalah sebuah program yang membaca suatu program yang ditulis dalam suatu bahasa sumber (*source language*) dan menterjemahkannya ke dalam suatu bahasa sasaran (*target language*). Dalam penguraian struktur kalimat, penganalisa leksikal menganalisa setiap kata dalam kalimat, kemudian menentukan jenis kelas katanya.

Hasil dari penganalisa leksikal ini digunakan oleh penganalisa sintaks yang akan memeriksa urutan simbol-simbol kelas kata tersebut dalam kalimat. Analisa kata dalam kalimat ini dilakukan oleh penganalisa leksikal berdasarkan kecocokan

kata dengan aturan-aturan leksikal berupa ekspresi regular yang sudah didefinisikan.

Tugas dari *scanner* adalah sebagai berikut :

1. Melakukan pembacaan kode sumber dengan merunut karakter demi karakter
2. Mengenali besaran leksik
3. Mentransformasi menjadi sebuah *token* dan menentukan jenis *token*nya.
4. Mengirimkan *token*
5. Membuang/mengabaikan blank dan komentar dalam *program*
6. Menangani kesalahan
7. Menangani tabel simbol

Di dalam aplikasi NLP sistem cerdas yang akan dibuat, yang dimaksud dengan program sumber yang diolah oleh *scanner* adalah berupa kalimat *input* dari pengguna dalam bentuk sms.

Ketika *scanner* menerima input berupa stream karakter kemudian memilah menjadi satuan leksik, satuan leksik tersebut terdiri atas simbol-simbol satuan yang jika dikombinasikan akan mempunyai arti yang berbedabeda. Simbol-simbol yang bisa dipergunakan dalam sebuah bahasa tentunya terbatas jumlahnya, yang membentuk sebuah himpunan dan disebut sebagai abjad (alphabet).

Tata bahasa (grammatika) adalah sekumpulan dari himpunan variabel-variabel, simbol-simbol terminal, simbol non-terminal, simbol awal yang dibatasi oleh aturan-aturan produksi. Aturan produksi adalah pusat dari tata bahasa yang menspesifikasikan bagaimana suatu tata bahasa melakukan transformasi suatu string ke bentuk lainnya. Dalam pembicaraan grammar, anggota alfabet dinamakan simbol terminal atau *token*. Kalimat adalah string yang tersusun atas simbol-simbol terminal. Bahasa adalah himpunan kalimat-kalimat. Anggota bahasa bisa berupa tak berhingga hingga kalimat.

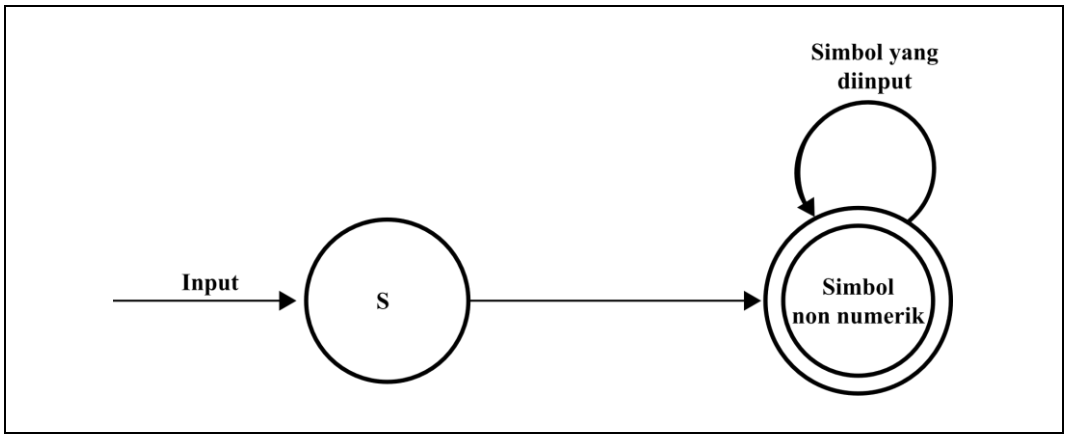
Simbol-simbol berikut adalah symbol terminal :

1. huruf kecil alphabet, misalnya :a, b, c
2. simbol operator, misalnya : +, -, dan ‘
3. simbol tanda baca, misalnya : (,), dan ;

Sedangkan simbol-simbol berikut adalah simbol non terminal :

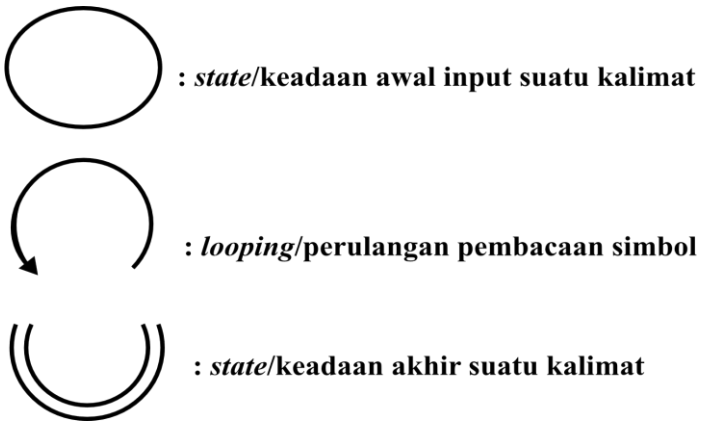
1. huruf besar awal alphabet, misalnya : A, B, C
2. huruf S sebagai simbol awal

Pada saat scanner membaca input, tools yang digunakan untuk menggambarkan perpindahan dari posisi satu ke posisi lainnya adalah diagram transisi. Diagram transisi dapat dilihat pada gambar 2. dibawah ini :



Gambar 2. Diagram transisi

Keterangan gambar :



2.1.4. *Parser* (Analisis Sintaksis)

Parser atau *syntactic analyzer* pada kompilator bahasa pemrograman berfungsi untuk memeriksa kebenaran kemunculan setiap *token*. Pada QA *system*, fungsi dari parser ini agak berbeda karena token yang akan diolah semua memiliki tipe yang sama yaitu berupa kata (*word*). Urutan kemunculan *token* yang berupa kata-kata tersebut akan diolah dengan mengacu pada brain file agar didapatkan makna kalimat yang sesungguhnya. Dengan kata lain, tahap analisa semantik terjadi di bagian brain *file*. Kemampuan dari parser untuk mengolah *token* dan bekerja sama dengan *brain file* inilah yang paling menentukan tingkat kecerdasan dari sebuah *chat bot*.

2.1.5. *Parsing*

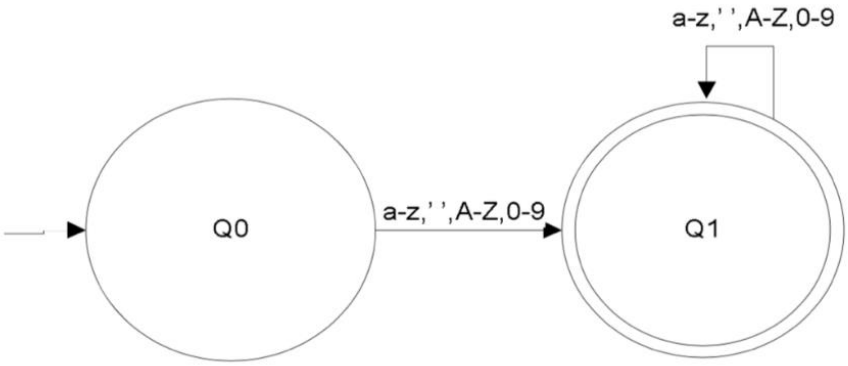
Proses *parsing* tidak hanya dapat dilakukan dalam proses *information retrieval*, melainkan juga pada bidang lain seperti pada pembuatan sebuah *compiler* dan bahasa alami. Sebelumnya perlu diketahui arti dari istilah *parser* yaitu *program*

yang melakukan proses *parsing*. Untuk pemrosesan, dokumen dipisahkan menjadi unit-unit yang lebih kecil misalnya berupa kata, frasa atau kalimat. Unit pemrosesan tersebut disebut sebagai *token*. *Parsing* merujuk pada proses pengenalan *token* yang terdapat dalam rangkaian teks.

Proses *parsing* (penguraian kalimat) juga merupakan proses yang dilakukan untuk menterjemahkan masukan dari pengguna agar dapat dimengerti oleh sistem. Secara *default*, seluruh kalimat masukan pengguna yang dimasukan akan dianggap sebagai kata-kata yang harus ada pada data yang akan dicari. Misal pengguna memasukan kalimat sebagai berikut : "*Penggunaan bahasa alami* "

Maka sistem akan melakukan pencarian data pada dokumen yang memiliki kata bahasa atau alami. Artinya jika terdapat dokumen yang mengandung kata bahasa inggris maka dokumen tersebut dianggap sesuai dengan yang dicari pengguna, karena mengandung kata bahasa. Hal ini membuat pengguna yang ingin mencari informasi mengenai bahasa alami akan mendapatkan hasil yang tidak sesuai dengan yang diinginkan.

Jika pengguna ingin melakukan proses pencarian yang lebih spesifik, agar informasi yang didapat lebih tepat sasaran, maka pengguna harus mengikuti aturan proses penguraian kalimat yang dimiliki sistem dapat dilihat pada gambar 3. Dibawah ini:.



Gambar 3. ***Finite state automata*** **proses** proses *parsing*

Konfigurasi FSA diatas secara formal dinyatakan sebagai berikut :

$$Q = \{Q0, Q1\} \quad (2.1)$$

$$\sum = \{a\text{-}z,\text{ '},A\text{-}Z,0\text{-}9\} \quad (2.2)$$

$$S = Q0 \quad (2.3)$$

$$F = \{Q1\} \quad (2.4)$$

Fungsi transisi yang ada sebagai berikut :

$$\delta(Q0, a\text{-}z|\text{ '}|A\text{-}Z|0\text{-}9) = Q1 \quad (2.5)$$

$$\delta(Q1, a\text{-}z|\text{ '}|A\text{-}Z|0\text{-}9) = Q1 \quad (2.6)$$

Fungsi tersebut bisa disajikan dan dilihat dalam tabe 1.Transisi dibawah ini:

Tabel 1. Transisi

δ	a-z|' '|A-Z|0-9
Q0	Q1
Q1	Q1

Dari diagram state gambar 3. , didapat aturan produksi sebagai berikut:
S ->aA..zA|AA..ZA| 0A-9A|<space>A|a..Z|A..Z|0..9|<space>

A ->aA..zA| AA..ZA |0A-9A|<space>A|a..Z|A..Z|0..9|<space>

Dimana secara formal dapat ditulis :

V = {S, A} (2.7)

T = {a-z,' ' A-Z,0-9} (2.8)

P = {S->aA..zA|AA..ZA|0A..9A|<space>A|a..Z|A..Z|0..9|<space>,(2.9)

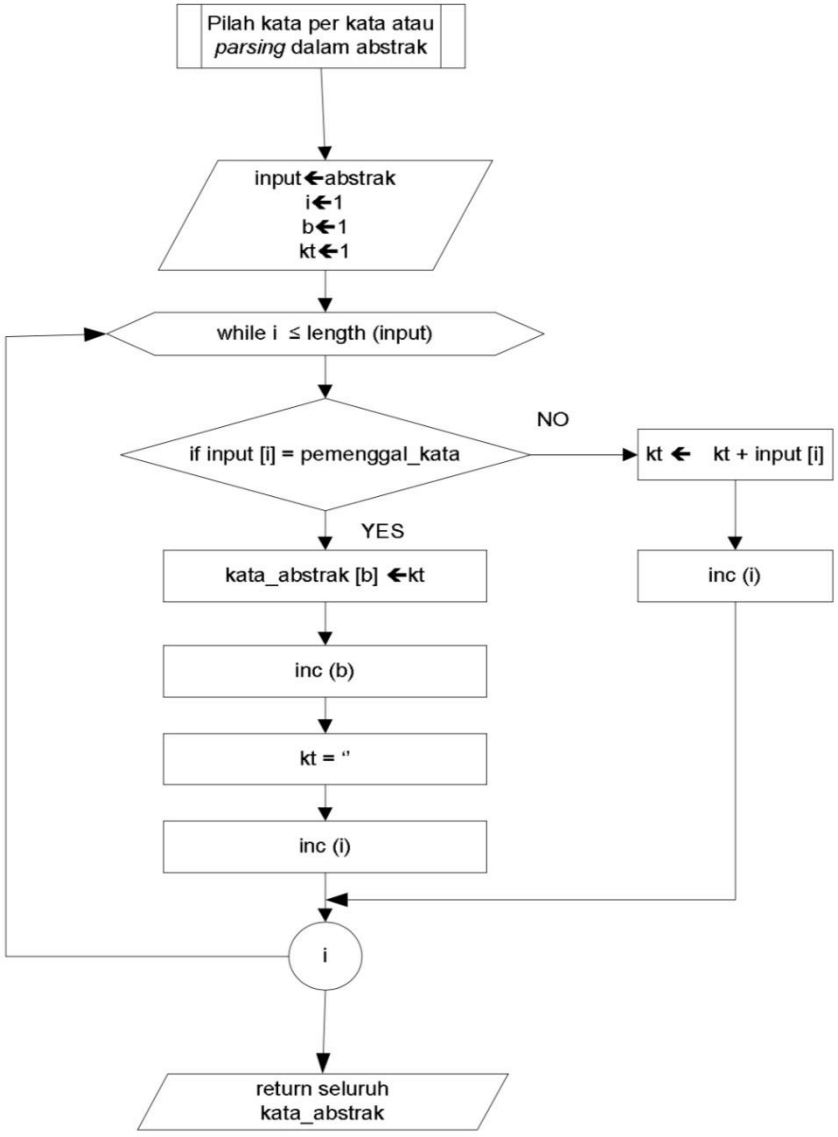
A->aA..zA| AA..ZA |0A..9A|<space>A|a..Z|A..Z|0..9|<space>} (2.10)

S=S (2.11)

Aturan produksi diatas menggunakan *left to right* sehingga jika pengguna memasukan pertanyaan "hai juga", maka proses akan dilakukan dari depan (*left*) ke belakang (*right*), seperti berikut:

S ->hA
S ->haA
S ->haiA
S ->haiA
S ->hai jA
S ->hai juA
S ->hai jugA
S ->hai jugaA
S ->hai juga

Pada proses di atas setiap kalimat atau kata akan dipecah menjadi perbagian karakter dalam bentuk *array*. proses ini digunakan untuk memotong - motong dokumen kata per kata dan menyimpannya pada *array* kata abstrak. *Flowchart* proses *parsing* abstak dapat dilihat pada gambar 4. dibawah ini :

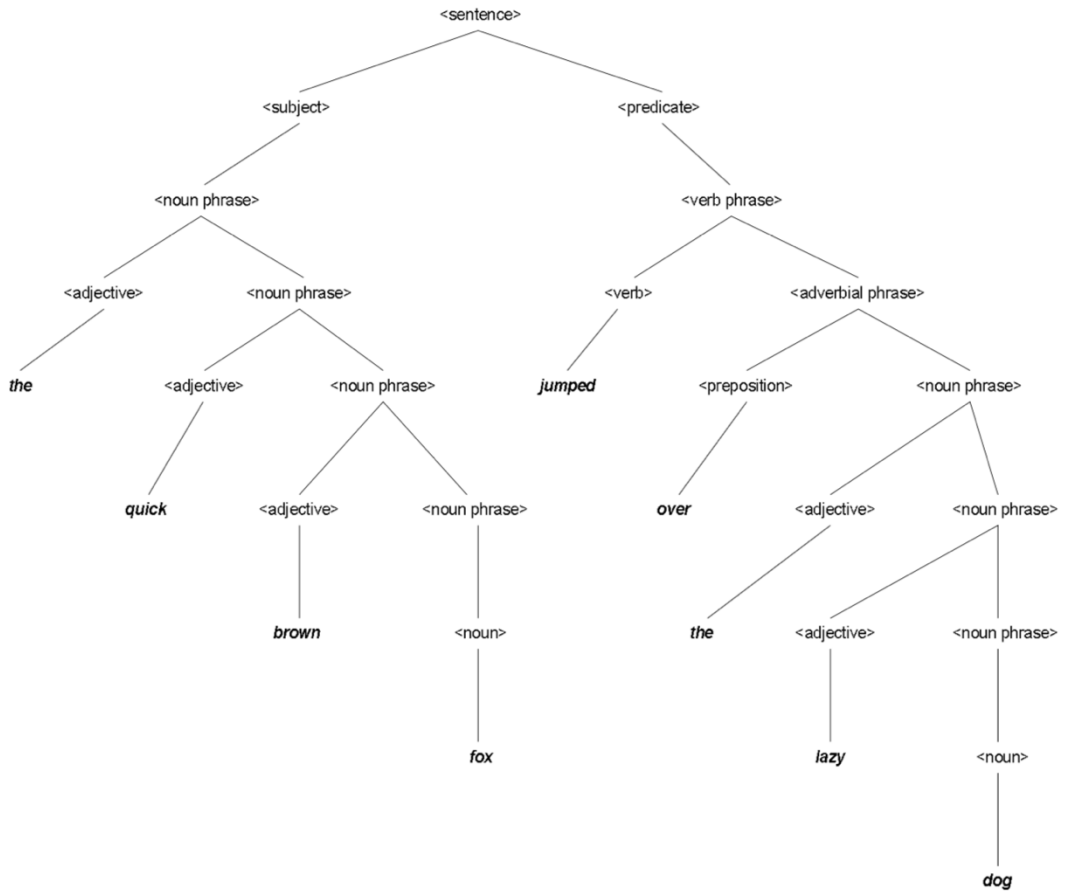


Gambar 4. ***Flowchart parsing***

2.1.6. Pohon Sintaks

Pohon (*tree*) adalah suatu *graph* terhubung tidak sirkuler, yang memiliki satu simpul (*node*)/*vertex* disebut akar (*root*) dan dari situ memiliki lintasan ke setiap simpul. Pohon sintaks/pohon penurunan (*syntax tree/derivation tree/parse tree*) berguna untuk menggambarkan bagaimana memperoleh suatu *string* dengan cara menurunkan simbol-simbol variabel menjadi simbol-simbol terminal. Setiap simbol variabel akan diturunkan menjadi terminal, sampai idak ada yang belum tergantikan. Contoh dapat dilihat pada gambar 5. dimana contoh sebuah *tree* yang menguraikan kalimat dalam bahasa Inggris.

The quick brown fox jumped over the lazy dog



Gambar 5. Pohon sintaks

Proses penurunan atau *parsing* bisa dilakukan dengan cara:

a. Penurunan terkiri (*leftmost derivation*): simbol variabel terkiri yang diperluas terlebih dahulu.
b. Penurunan terkanan (*rightmost derivation*): simbol variabel terkanan yang diperluas terlebih dulu.

2.1.7. *Text Mining*

Text mining merupakan salah satu aplikasi dari *data mining*. *Text mining* juga sering disebut sebagai *Text Data Mining* (TDM) dan *knowladge Discovery in Textual Databases* (KDT). *Text mining* merupakan proses mengesktrak *petterns* dan *knowladge* yang bersifat menarik dan *nontrivial* (penting) dari dokumen-dokumen teks. Pada intinya proses kerja *text mining* sama dengan proses kerja *data mining* pada umumnya hanya saja data yang di *mining* merupakan *text databases*.

Data teks akan diproses menjadi data numerik agar dapat dilakukan proses lebih lanjut. Sehingga dalam *text mining* ada istilah *preprocessing data*, yaitu proses pendahulu yang diterapkan terhadap data teks yang bertujuan untuk menghasilkan data numerik.

Pada proses *preprosesing* merupakan tahap dimana deskripsi di tangani untuk dapat siap diproses memasuki tahap *text mining*. Tahap-tahap tersebut adalah:

1. *Parsing / Tokenizing*
2. *Stopwords Removal / Filtering*
3. *Stemming*
4. *Tagging*
5. *Analyzing*

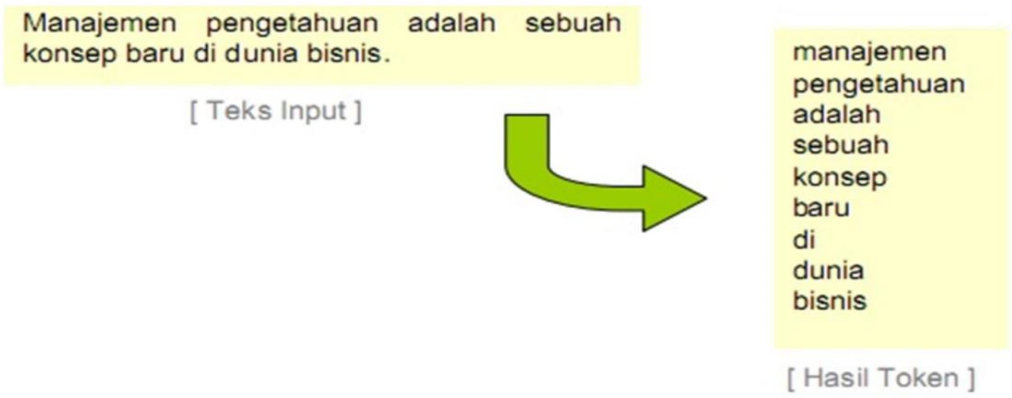
Text mining merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks, yaitu proses penganalisisan teks guna menyarikan informasi yang bermanfaat untuk tujuan tertentu.

Berdasarkan ketidakteraturan struktur data teks, maka proses text mining memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur. Berikut ini adalah bidang-bidang penerapan text mining yang paling popular:

1. *Information extraction* (ekstraksi informasi). Identifikasi terhadap hubungan dan frase-frase kunci dalam *text* dengan mencari urutan yang sudah ditetapkan dalam text menggunakan pencocokan pola.
2. *Topic tracking* (pelacakan topic). Berdasarkan pada profil user dan berbagai dokumen yang dilihat *user*, *text mining* bisa memprediksi dokumen-dokumen lain yang menjadi perhatian/minat *user* tersebut.
3. *Summarization* (peringkasan). Meringkas suatu dokumen untuk menghemat waktu dari sisi si pembaca
4. *Clustering*. Mengelompokkan dokumen-dokumen yang mirip tanpa memiliki kategori yang sudah ditetapkan sebelumnya.
5. *Concept linking*. Menghubungkan berbagai dokumen terkait dengan mengidentifikasi konsep yang digunakan berbsama dan dengan demikian membantu para user untuk menemukan informasi yang barangkali mereka tidak akan temukan dengan menggunakan metode-metode pencarian tradisional.
6. *Question answering*. Menemukan jawaban terbaik pada pertanyaan yang diberikan melalui pencocokan pola berbasis *knowledge*

2.1.8. *Parsing / Token*izing

Parsing adalah sebuah proses yang dilakukan seseorang untuk menjadikan sebuah kalimat menjadi lebih bermakna atau berada dengan cara memecah kalimat tersebut menjadi kata-kata atau frase-frase (“Parsing”). Parsing di dalam pembuatan aplikasi *text mining* ini merupakan proses penguraian deskripsi yang semula berupa kalimat-kalimat berisi kata-kata dan tanda pemisah antara kata seperti titik(.), koma(,), spasi dan tanda pemisah lain menjadi kata-kata saja baik itu berupa kata-kata penting maupun kata- kata tak penting. Secara sederhana proses parsing ini terlihat sebagai proses pengambilan kata jika ketemu tanda spasi namun pada kenyataannya tidak sesederhana itu. Contoh tahap ini dapat dilihat pada gambar 6. berikut ini

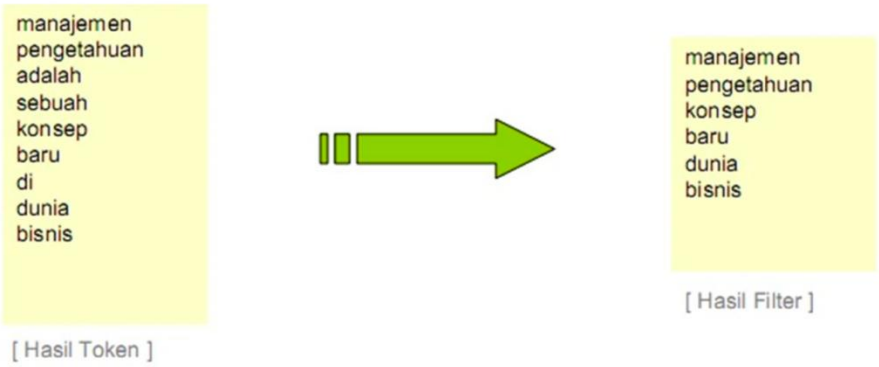


Gambar 6. Tahap *parsing* atau *tokenizing*

2.1.9. *Stopwords Removal / Filtering*

Kebanyakan bahasa resmi di berbagai negara memiliki kata fungsi dan kata sambung seperti artikel dan preposisi yang hampir selalu muncul pada dokumen-dokumen teks. Biasanya kata-kata ini memiliki arti yang lebih di dalam memenuhi kebutuhan seorang *searcher* di dalam mencari informasi. Kata-kata tersebut (misalnya a, an, the on pada bahasa inggris) disebut sebagai *stopwords*. Di dalam bahasa Indonesia stopwords dapat disebut sebagai kata tidak penting misalnya "di", "oleh", "pada", "sebuah", "karena". Sebelum proses *stopwords removal* dilakukan, terlebih dulu dibuat daftar *stopwords* (*stoplist*). Preposisi, kata hubung dan partikel biasanya merupakan kandidat *stoplist*.

Stopwords removal merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil parsing deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak. Jika termasuk di dalam *stoplist* maka kata-kata tersebut akan di-*remove* dari deskripsi sehingga kata-kata yang tersisa di dalam deskripsi di anggap sebagai kata-kata penting atau *keywords*. Tahap *filtering* dapat dilihat pada gambar 7. diabwah ini:



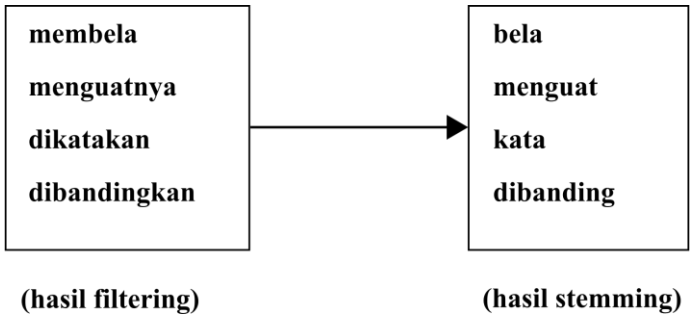
Gambar 7. Tahap *filtering*

2.1.10. *Stemming*

Stemming adalah proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya (*stem*). Proses ini juga

disebut sebagai *conflation*. Proses stemming secara luas sudah digunakan di dalam Information retrieval (pencarian informasi) untuk meningkatkan kualitas informasi yang didapatkan. Kualitas informasi yang dimaksud misalnya untuk mendapatkan hubungan antara *variant* kata yang satu dengan yang lainnya. Sebagai contoh kata "diculik", "menculik" (melakukan tindakan menculik) dan "penculik" (orang yang menculik) yang semula mengandung arti yang berbeda dapat di-*stem* menjadi sebuah kata "culik" yang memiliki arti yang sama sehingga kata-kata diatas saling berhubungan.

Selain itu *stemming* juga dapat digunakan untuk mengurangi ukuran dari suatu ukuran *index file*. Misalnya dalam suatu deskripsi terdapat *variant* kata "memberikan", "diberikan", "memberi" dan "diberi" hanya memiliki akar kata (stem) yaitu "beri". Ukuran *file* daftar *index* yang semula berjumlah lima *record* akan di-*reduce* sehingga menjadi satu *record* saja. Gambaran tahap *stemming* dapat dilihat pada gambar 8. dibawah ini :



Gambar 8. Tahap s*temming*

2.1.11. *Analyzing*

Tahap *analyzing* merupakan tahap penentuan seberapa jauh keterhubungan antara kata-kata dengan dokumen yang ada. *Analyzing Stemming* untuk Bahasa Indonesia Menggunakan *Confix Stripping Stemmer*.

Confix Stripping (CS) *stemmer* adalah metode stemming untuk Bahasa Indonesia yang diperkenalkan oleh Jelita Asian. *Stemmer* ini merupakan pengembangan dari metode *stemming* untuk Bahasa Indonesia yang diperkenalkan oleh Nazief dan Adriani (1996). Algoritma *stemming* Nazief dan Adriani ini dikembangkan berdasarkan pada aturan morfologi Bahasa Indonesia yang mengelompokkan dan mengenkapsulasi imbuhan-imbuhan, termasuk di dalamnya adalah awalan (*prefix*), sisipan 17 (*infix*), akhiran (*suffix*) dan gabungan awalan-akhiran (*confixes*).

Algoritma ini menggunakan kamus kata dasar dan mendukung recoding, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih. Kombinasi awalan dan akhiran yang dilarang dapat dilihat pada tabel 2.

Tabel 2. Kombinasi awalan atau akhiran yang dilarang

Awalan (*prefix*)	Akhiran (*Suffix*) yang tidak diperbolehkan
Be-	-i
di-	-an
Ke-	-i, -kan
Me-	-an
Se-	-i, -kan
Te-	-an

Algoritma *stemmer* Nazief dan Adriani mengelompokkan imbuhan ke dalam beberapa kategori sebagai berikut:

1. *Inflection suffixes* yakni kelompok-kelompok akhiran yang tidak merubah bentuk kata dasar. Sebagai contoh, kata "duduk" yang diberikan akhiran "-lah" akan menjadi "duduklah". Kelompok ini dapat dibagi menjadi dua :
   a. *Particle* (P) atau partikel, yakni termasuk di dalamnya "-lah", "- kah", "-tah", dan "-pun".
   b. *Possessive Pronoun* (PP) atau kata ganti kepunyaan, termasuk di dalamnya adalah "-ku" , "-mu", dan "-nya".
2. *Derivation Suffixes* (DS) yakni kumpulan akhiran asli Bahasa Indonesia yang secara langsung ditambahkan pada kata dasar. Termasuk di dalamnya adalah akhiran "-i", "-kan", dan "-an".
3. *Derivation Prefixes* (DP) yakni kumpulan awalan yang dapat langsung diberikan pada kata dasar murni, atau pada kata dasar yang sudah mendapatkan penambahan sampai dengan 2 awalan. Termasuk di dalamnya adalah awalan yang dapat bermorfologi ("me-", "be-", "pe-", dan "te-") serta awalan yang tidak bermorfologi ("di-", "ke-" dan "se-")

Berdasarkan pengklasifikasian imbuhan-imbuhan di atas, maka bentuk kata berimbuhan dalam Bahasa Indonesia dapat dimodelkan sebagai berikut:[DP + [DP + [DP +]]] Kata Dasar [[+ DS][+ PP][+ P]]Dengan batasan-batasan sebagai berikut :

1. Tidak semua kombinasi diperbolehkan, sebagai contoh pada kata yang diberi awalan "di-", maka penambahan akhiran "-an" tidak diperkenankan. Kombinasi-kombinasi imbuhan yang tidak diperbolehkan dapat dilihat pada Tabel 2.
2. Penggunaan imbuhan yang sama secara berulang tidak diperkenankan.
3. Jika suatu kata hanya terdiri dari satu atau dua huruf, maka proses stemming tidak dilakukan.
4. Penambahan suatu awalan tertentu dapat mengubah bentuk asli kata dasar, ataupun awalan yang telah diberikan sebelumnya pada kata dasar bersangkutan (bermorfologi). Sebagai contoh, awalan "me-" dapat berubah menjadi "meng-", "men-", "meny-", dan "mem-". Oleh karena itu, diperlukan suatu aturan yang mampu mengatasi masalah morfologi ini.

Pada dasarnya, algoritma *confix stripping stemmer* dikembangkan dari algoritma *stemming* yang dibuat oleh Nazief dan Adriani, dengan beberapa

penambahan aturan tertentu yang telah terbukti mampu meningkatkan kinerja *stemmer* tersebut. Algoritma *stemmer* yang diperkenalkan Nazief dan Adriani didefinisikan sebagai berikut:

1. Kata yang hendak di-stemming dicari terlebih dahulu pada kamus. Jika kata ditemukan dalam kamus, berarti kata tersebut sudah berbentuk kata dasar, jika tidak maka tahap selanjutnya dilakukan.
2. Hilangkan inflectional particle P ("-lah", "-kah", "-tah", "-pun") dan kata ganti kepunyaan atau possessive pronoun PP ("-ku", "-mu", "-nya").
3. Hilangkan derivation suffixes DS ("-i", "-kan", atau "-an").
4. Hilangkan derivation prefixes DP {"di-","ke-","se-","me-","be-","pe","te-" dengan iterasi maksimum adalah 3 kali:
   a. Langkah 4 berhenti jika:
      Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada Tabel 2.. Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya. Tiga awalan telah dihilangkan.
   b. Identifikasikan tipe awalan dan hilangkan. Awalan ada dua tipe:
      Standar: "di-", "ke-", "se-" yang dapat langsung dihilangkan dari kata. Kompleks: "me-", "be-", "pe", "te-" adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel 2. untuk mendapatkan pemenggalan yang tepat..
   c. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan
5. Apabila setelah langkah 4 kata dasar masih belum ditemukan, maka proses recording dilakukan dengan mengacu pada aturan pada Tabel 2. recoding dilakukan dengan menambahkan karakter recoding di awal kata yang dipenggal. Pada Tabel 2., karakter recoding adalah huruf kecil setelah tanda hubung ('-') dan terkadang berada sebelum tanda kurung.
6. Jika semua langkah gagal, maka input kata yang diuji pada algoritma ini dianggap sebagai kata dasar.

Dilihat segi performa, *stemmer* ini mampu memberikan hasil yang lebih baik jika dibandingkan dengan algoritma stemmer untuk Bahasa Indonesia seperti yang dikembangkan oleh Arifin dan Setiono (2002) , Vega (2001), Ahmad (1996), dan Idris (2001). Hal ini wajar, mengingat algoritma ini merupakan algoritma yang paling kompleks. Jelita Asian kemudian mengembangkan algoritma CS *stemmer*, dengan menambahkan beberapa perbaikan yang bertujuan untuk meningkatkan hasil *stemming* yang diperoleh. Perbaikan yang ditambahkannya adalah sebagai berikut:

1. Menggunakan kamus kata dasar yang lebih lengkap.
2. Memodifikasi dan menambahkan aturan pemenggalan untuk tipe awalan yang kompleks (memodifikasi aturan pada Tabel 2. sesuai modifikasi pada Tabel 3., dan menambahkan aturan pada Tabel 4 ke Tabel.2).
3. Menambahkan aturan stemming untuk kata ulang dan bentuk jamak, semisal pada "buku-buku" yang seharusnya di-stemming menjadi "buku". Caranya, adalah dengan melakukan pemisahan menjadi dua sub-kata, yang masing-masing di-*stemming*. Apabila *stemming* memberikan kata dasar yang sama,

maka *output* kata dasarnya adalah hasil *stemming* tersebut. Namun apabila hasil *stemming* 2 sub-kata ini berbeda, maka dapat disimpulkan bahwa input adalah kata ulang semu, dan tidak memiliki bentuk kata dasar lagi.

4. Mengubah urutan *stemming* untuk beberapa kasus tertentu. Algoritma *stemmer* Nazief dan Adriani akan menghilangkan akhiran terlebih dahulu, baru diikuti penghilangan awalan. Menurut Jelita Asian, cara ini tidak selalu berhasil pada beberapa kata. Oleh karena itu, diberikan beberapa aturan yang akan mengubah urutan *stemming*, dimana penghilangan awalan dilakukan terlebih dahulu, lalu diikuti penghilangan akhiran. Aturan ini disebut rule *precedence*, dan berlaku jika suatu kata memiliki pasangan awalan-akhiran "be-lah", "be-an", "me-i", "di-i", "pe-i", atau "te-i".

Pada tabel 3., tabel 4., tabel 5., simbol C merupakan konsonan, simbol V menandakan vokal, simbol A merupakan vokal atau konsonan, dan simbol P merepresentasikan partikel atau *fragmen* dari suatu kata, misalnya "er".

Tabel 3. Aturan pemenggalan awalan *stemmer*

Aturan	Format Kata	Pemenggalan
1	berV...	ber-V...| ber-rV...
2	berCAP...	ber-CAP...dimana C!='r' & P!='er'
3	berCAerV...	ber-CaerV...dimana C!='r'
4	belajar	bel-ajar
5	beC1erC2 ...	be-C1erC2... dimana C1!={'r'|'l'}
6	terV...	ter-V...| te-rV...
7	terCerV...	ter-CerV... dimana C!='r'
8	terCP...	ter-CP... dimana C!='r' dan P!='er'
9	teC1erC2...	te-C1erC2... dimana C1!='r'
10	me{l|r|w|y}V...	me-{l|r|w|y}V...
11	mem{b|f| v}...	mem-{b|f|v}...
12	mempe{r|l}...	mem-pe...
13	mem{rV|V}...	me-m{rV|V}... | me-p{rV|V}...
14	men{c|d|j|z}...	men-{c|d|j|z}...
15	menV...	me-nV... | me-tV
16	meng{g|h|q}...	meng-{g|h|q}...
17	mengV...	meng-V... | meng-kV...
18	menyV...	meny-sV...
19	mempV...	mem-pV... dimana V!='e'
20	pe{w|y}V...	pe-{w| y}V...
21	perV...	per-V...| pe-rV...
23	perCAP...	per-CAP... dimana C!='r' dan P!='er'
24	perCAerV...	per-CAerV... dimana C!='r'

25	pem{b|f|V}...	Pem-{b|f|V}...
26	pem{rV|V}...	Pe-m{rV|V}... | pe-p{rV|V}...
27	pen{c|d|j|z}...	pen-{c|d|j|z}...
28	penV...	pe-nV... | pe-tV...
29	peng{g|h|q}...	peng-{g|h|q}...
30	pengV...	peng-V... | peng-kV...
31	penyV...	peny-sV...
32	pelV...	pe-lV... kecuali "pelajar" yang menghasilkan "ajar"
33	peCerV...	per-erV... dimana C!={r|w|y|l|m|n}
34	peCP...	pe-CP... dimana C!={r|w|y|l|m|n} dan P!='er'

Tabel 4. Modifikasi aturan pada tabel 3.

Aturan	Format Kata	Pemenggalan
12	mempe...	mem-pe
16	Meng{g|h|q|K}...	meng-{g|h|q|k}...

Tabel 5. Tambahan aturan untuk tabel 3.

Aturan	Format Kata	Pemenggalan
35	terC$_1$erC$_2$ ...	ter-C$_1$erC$_2$... dimana C$_1$!='r'
36	peC$_1$erC$_2$ ...	pe-C$_1$erC$_2$... dimana C$_1$!={r|w|y|l|m|n}

Algoritma *confix stripping* (CS) *stemmer* yang dikembangkan Jelita Asian bekerja sebagai berikut :

1. Cek *rule precedence*, apabila bernilai benar maka lakukan penghilangan awalan terlebih dahulu. Apabila bernilai salah, maka penghilangan akhiran dilakukan terlebih dahulu.
2. Lakukan *recoding* apabila diperlukan.
3. Cek apakah terdapat tanda hubung ('-') yang menandakan bahwa input kata tersebut adalah kata ulang. Jika benar, maka lakukan proses *stemming* pada potongan kata di sebelah kiri dan kanan tanda hubung tersebut. Apabila *stemming* dua kata ini memberikan hasil yang sama, maka kata dasar kata ulang tersebut adalah hasil *stemming* yang didapatkan.
4. Jika ketiga proses di atas gagal, maka input kata yang di-*stemming* dianggap sebagai kata dasar. Pada setiap perpindahan langkah, dilakukan proses pencarian *output stemming* ke kamus. Apabila ditemukan di kamus, maka algoritma ini berhenti. Berikut adalah contoh proses *stemming* pada kata "menangkapnya" dengan menggunakan *confix stripping stemmer*:
   a. *Rule precedence* bernilai false karena tidak ditemukan kombinasi awalan dan akhiran yang tidak diperbolehkan.Pemenggalan akhiran dilakukan terlebih dahulu, dan menyisakan kata "menangkap".

b. Pemenggalan awalan menyisakan kata “nangkap”.
c. Karena “nangkap” tidak terdapat di kamus, maka lakukan recoding dengan mengganti ‘n’ dengan ‘t’ (aturan 15 pada Tabel 3.).
d. Kata “tangkap” ada di kamus, oleh karena itu proses berhenti.
Kata dasar “menangkapnya” adalah “tangkap”.

Sayangnya, setelah melakukan beberapa analisa dan percobaan kecil, ditemukan beberapa contoh kata yang tidak dapat di-stemming. Analisa atas beberapa kata yang gagal di-stemming tersebut adalah sebagai berikut:

1. Kurangnya aturan pemenggalan awalan untuk kata-kata dengan format “mem+p...”. Hal ini terjadi pada kata “mempromosikan”, “memproteksi”, dan “memprediksi”.
2. Kurangnya aturan pemenggalan awalan untuk kata-kata dengan format “men+s...”. Hal ini terjadi pada kata “mensyaratkan”, dan “ m e n s yu k u r i ”.
3. Kurang relevannya aturan 17 untuk pemenggalan awalan pada kata- kata dengan format “menge+kata dasar”, seperti pada kata “mengerem”.
4. Kurang relevannya aturan 30 untuk pemenggalan awalan pada kata- kata dengan format “penge+kata dasar”, seperti pada kata “pengeboman”.
5. Kurangnya aturan pemenggalan awalan untuk kata-kata dengan format “peng+k...” seperti pada kata “pengkajian”.
6. Adanya elemen pada beberapa kata dasar yang menyerupai suatu imbuhan. Kata-kata seperti “pelanggan”, “perpolitikan”, dan “pelaku” gagal di-stemming karena akhiran “-an”, “-kan” dan “-ku” seharusnya tidak dihilangkan.

Berdasarkan kegagalan-kegagalan ini, maka untuk memperbaiki kesalahan algoritma CS *stemmer* dengan menambah beberapa perbaikan, diantaranya adalah sebagai berikut:

1. Merevisi aturan 19 pada tabel 3. agar *stemming* berhasil pada kata-kata dengan format “mem+p...”. revisi aturan ini dapat dilihat pada tabel 6.
2. Merevisi aturan 14 pada tabel 3. agar *stemming* berhasil pada kata-kata dengan format “men+s...”. revisi aturan ini dapat dilihat pada tabel 6.
3. Merivisi aturan 17 pada tabel 3. agar *stemming* berhasil pada kata-kata dengan format “menge+...”. revisi aturan ini dapat dilihat pada Tabel 6.
4. Merivisi aturan 30 pada tabel 3. agar *stemming* berhasil pada kata-kata dengan format “penge+...”. revisi aturan ini dapat dilihat pada tabel 6.
5. Merevisi aturan 29 pada tabel 3. agar *stemming* berhasil pada kata-kata dengan format “peng+k...”. revisi aturan ini dapat dilihat pada tabel 6.
6. Menambahkan suatu algoritma tambahan untuk mengatasi kesalahan pemenggalan akhiran yang seharusnya tidak dilakukan. Algoritma ini disebut *loop* Pengembalian Akhiran, dan dilakukan apabila proses *recoding* gagal.

Algoritma *loop* Pengembalian Akhiran dideskripsikan sebagai berikut:

1. Kembalikan seluruh awalan yang telah dihilangkan sebelumnya, sehingga menghasilkan model kata seperti berikut: [DP+[DP+[DP]]] + Kata Dasar. Pemenggalan awalan dilanjutkan dengan proses pencarian di kamus kemudian dilakukan pada kata yang telah dikembalikan menjadi model tersebut.
2. Kembalikan akhiran sesuai dengan urutan model. Ini berarti bahwa pengembalian dimulai dari DS (“-i”, “-kan”, “-an”), lalu PP(“-ku”, “-mu”, “-nya”), dan terakhir adalah P (“-lah”, “-kah”, “-tah”, “-pun”). Untuk setiap

pengembalian, lakukan langkah 3) hingga 5) berikut. Khusus untuk akhiran "-kan", pengembalian pertama dimulai dengan "k", baru kemudian dilanjutkan dengan "an".

3. Lakukan pengecekan di kamus. Apabila ditemukan, proses dihentikan. Apabila gagal, maka lakukan proses pemenggalan awalan berdasarkan aturan pada Tabel 3. (dengan revisi Tabel 6.).
4. Lakukan recoding apabila diperlukan.
5. Apabila pengecekan di kamus tetap gagal setelah recoding, maka awalan awalan yang telah dihilangkan dikembalikan lagi.

Tabel 6. Revisi untuk tabel 3.

Aturan	Format Kata	Pemenggalan
14	men{c|d|j|s|z}...	men-{c|d|j|s|z}...
17	mengV...	meng-V... |meng-kV...| (mengV- ...jika V='e')
19	mempA...	mem-pA... dengan A!='e'
29	pengC...	peng-C...
30	pengV...	peng-V... |peng-kV... | (pengV-...jika V='e')

B. *Web Mining.*

Web mining adalah ekstraksi pola-pola penting dan bermanfaat namun tersimpan secara implisit pada kumpulan data yang relatif besar pada layanan *world wide web*. *Web mining* teridiri atas tiga bagian yaitu: *web content mining*, *web structure mining*, dan *web usage mining*.

Web content mining adalah suatu proses otomatis untuk menemukan informasi yang berguna dari dokumen atau *data*. Pada prinsipnya teknik ini mengekstraksi kata kunci yang terkandung pada dokumen. Isi data *web* antara lain dapat berupa teks, citra, audio, video, metadata, dan hyperlink. Ada dua strategi yang umum digunakan: pertama langsung melakukan mining terhadap data, dan kedua melakukan pencarian serta mengimprove hasil pencarian seperti layaknya *search engine*.

Web struncture mining dikenal juga sebagai *web log mining* adalah teknik yang digunakan untuk menemukan struktur link dari hyperlink dan membangun rangkuman *website* dan halaman *web*. Salah satu manfaatnya adalah untuk menentukan pagerank pada suatu halaman *web*.

Web usage mining adalah teknik untuk mengenali perilaku pelanggan dan struktur *web* melalui informasi yang diperoleh dari *log*, *click stream*, *cookies*, dan *query*. Berbagai *tool* yang sudah ada antara lain *WebLogMiner* yang melakukan mining terhadap data log. Teknik yang lebih canggih digunakan untuk melakukan *OLAP*. Manfaat *web usage mining* adalah untuk kustomosasi halaman berdasarkan profil pengguna, menentukan ketertarikan pelanggan terhadap produk tertentu, dan menentukan target market yang sesuai. Menurut Han dan Kamber (pakar *data*

mining), *web* juga memberikan tantangan besar untuk penemuan pengetahuan yang efisien dan efektif:

1. Web terlalu besar untuk melakukan data *mining* yang efektif. *Web* sangat besar dan tumbuh dengan sangat cepat sehingga sangat sulit bahkan untuk sekedar diukur. Karena ukuran *size* nya yang unik, maka tidak lah layak untuk membuat *data warehouse* untuk me-replikasi,menyimpan, dan mengintegrasikan semua *data* yang ada di *web*, yang akhirnya membuat pengumpulan dan integrasi data menjadi suatu tantangan tersendiri.
2. *Web* sangatlah kompleks. Kompleksitas halaman *web* jauh lebih besar disbanding dengan suatu halaman dalam koleksi dokumen teks tradisional. Halaman-halaman *web* kurang terpadu strukturnya. Halaman-halaman *web* mengandung gaya penulisian dan variasi konten yang jauh lebih banyak disbanding dengan buku, artikel atau dokumen teks tradisional lainnya.
3. *Web* terlalu dinamis. *Web* adalah sumber informasi yang sangat dinamis. Tidak hanya tumbuh dengan cepat, tetapi kontennya juga terus di-*update* secara konstan. *Blog*, artikel berita, pasar saham, laporan cuaca, skor olah raga, harga, iklan-iklan perusahaan, dan banyak jenis informasi lainnya di-*update* secara *regular* di *web*.
4. *Web* tidaklah spesifik pada suatu *domain* tertentu. *Web* menyajikan keragaman komunitas yang sangat luas dan menghubungkan miliaran *computer*. Para pengguna *web* memiliki latar belakang yang berbeda-beda, minat yang berbeda, dan tujuan penggunaan web yang berbeda. Kebanyakan pengguna mungkin tidak memiliki mengetahuan yang baik mengenai struktur jaringan informasi dan mugngkin tidak sadar tentang biaya besar dalam pencarian tertentu yang mereka lakukan.
5. *Web* memiliki segalanya. Hanya sebagian kecil informasi di *web* yang benar-benar relevan atau bermanfaat bagi seseorang (atau untuk suatu tugas). Menurut laporan bahwa 99 persen informasi di *web* sama sekali tidak berguna bagi 99 persen pengguna *web*. Meskipun hal ini kelihatannya kurang jelas, memanng benar bahwa orang tertentu pada umumnya tertarik hanya pada sebagian kecil saja dari *web*, sedangkan sisanya di *web* berisi informasi yang tidak menarik bagi pengguna tersebut dan mungkin membanjiri hasil-hasil yang diinginkan. Menemukan porsi *web* yang benar-benar relevan terhadap seseorang dan tugas yang sedang dilakukan adalah isu yang sedang menonjol dalam riset yang terkait *web*.

C. Data *Mining*

Data *Mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Patut diingat bahwa kata *mining* sendiri berarti usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar. Karena itu *Data Mining* sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, statistik dan *database*. *Data mining* adalah proses menerapkan metode ini untuk data dengan maksud untuk mengungkap pola-pola tersembunyi. Dengan arti lain *Data mining* adalah proses untuk penggalian pola-pola dari data. *Data mining* menjadi alat yang semakin penting untuk

mengubah data tersebut menjadi informasi. Hal ini sering digunakan dalam berbagai praktek profil, seperti pemasaran, pengawasan, penipuan deteksi dan penemuan ilmiah. Telah digunakan selama bertahun-tahun oleh bisnis, ilmuwan dan pemerintah untuk menyaring volume *data* seperti catatan perjalanan penumpang penerbangan, data sensus dan supermarket *scanner data* untuk menghasilkan laporan riset pasar.

Alasan utama untuk menggunakan *data mining* adalah untuk membantu dalam analisis koleksi pengamatan perilaku. *Data* tersebut rentan terhadap *collinearity* karena diketahui keterkaitan. Fakta yang tak terelakkan *data mining* adalah bahwa subset/set data yang dianalisis mungkin tidak mewakili seluruh *domain*, dan karenanya tidak boleh berisi contoh-contoh hubungan kritis tertentu dan perilaku yang ada di bagian lain dari domain . Untuk mengatasi masalah semacam ini, analisis dapat ditambah menggunakan berbasis percobaan dan pendekatan lain, seperti *Choice Modelling* untuk data yang dihasilkan manusia. Dalam situasi ini, yang melekat dapat berupa korelasi dikontrol untuk, atau dihapus sama sekali, selama konstruksi desain eksperimental.

Beberapa teknik yang sering disebut-sebut dalam literatur *Data Mining* dalam penerapannya antara lain: *clustering*, *classification*, *association rule mining*, *neural network*, *genetic algorithm* dan lain-lain. Yang membedakan persepsi terhadap Data *Mining* adalah perkembangan teknik-teknik *Data Mining* untuk aplikasi pada *database* skala besar. Sebelum populernya *Data Mining*, teknik-teknik tersebut hanya dapat dipakai untuk data skala kecil saja.

Berikut Fungsi - Fungsi Umum *Data Mining:*

1. *Assosiation*, adalah proses untuk menemukan aturan assosiatif antara suatu kombinasi item dalam suatu waktu
2. *Sequence*, proses untuk menemukan aturan assosiatif antara suatu kombinasi item dalam suatu waktu dan diterapkan lebih dari satu periode
3. *Clustering*, adalah proses pengelompokan seumlah *data*/obyek ke dalam kelompok data sehingga setiap kelompok berisi data yang mirip
4. *Classification*, proses penemuan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui.
5. *Regretion*, adalah proses pemetaan data dalam suatu nilai prediksi
6. *Forecasting*, adalah proses pengestimasian nilai prediksi berdasarkan pola-pola di dalam sekumpulan data.
7. *Solution*, adalah proses penemuan akar masalah dan problem solving dari persoalan bisnis yang dihadapkai atau paling tidak sebagai informasi dalam pengambilan keputusan.

Terdapat beberapa proses dalam data mining adalag sebagai berikut:

1. Pembersihan *data* dan integritas data (*Cleaning* & *Integration*): Proses ini digunakan untuk membuang data yang tidak konsisten dan bersifat noise dari data yang terdapat di berbagai basisdata yang mungkin berbeda format maupun platform yang kemudian dinintegrasikan dalam satu database *datawarehouse.*

2. Seleksi dan transformasi *data* (*selection and transformation*): *Data* yang ada dalam *database datawarehouse* kemudian direduksi untuk mendapatkan hasil yang akurat.
3. Penambangan data (*data mining*): Data yang telah ditransformasi, kemudian ditambang dengan berbagai teknik. Proses *data mining* adalah proses mencari pola atau informasi menarik dalam *data* terpilih dengan menggunkan fungsi-fungsi tertentu. Fungsi atau algoritma dalam *data mining* sangat bervariasi, dimana pemilihannya bergantung pada tujuan dan proses pencarian pengetahuan secara menyeluruh.
4. Evaluasi pola dan presentasi pengetahuan: Tahap ini merupakan bagian dari proses pencarian pengetahuan yang mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya. Langkah terakhir KDD adalah mempresentasikan pengetahuan dalam bentuk yang mudah dipahami pengguna.

D. *TF-IDF.*

Pada dokumen yang besar, skema yang paling sukses dan secara luas digunakan untuk pemberian bobot term adalah skema pembobotan atau Term Weighting TF-IDF. Kelemahan scoring dengan Jaccard coefficient adalah tidak disertakannya frekuensi suatu term dalam suatu dokumen, maka diperlukan skoring dengan kombinasi Term Weighting TF-IDF. Hal yang perlu diperhatikan dalam pencarian informasi dari koleksi dokumen yang heterogen adalah pembobotan term. Term dapat berupa kata, frase atau unit hasil indexing lainnya dalam suatu dokumen yang dapat digunakan untuk mengetahui konteks dari dokumen tersebut, maka untuk setiap kata tersebut diberikan indikator, yaitu term weight (Informatikalogi, 2016, Pembobotan Kata atau Term Weighting TF-IDF).

TF (Term Frequency) adalah frekuensi dari kemunculan sebuah term dalam dokumen yang bersangkutan. Semakin besar jumlah kemunculan suatu *term* (TF tinggi) dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. Pada Term Frequency (TF), terdapat beberapa jenis formula yang dapat digunakan :

1. *TF biner (binary TF),* hanya memperhatikan apakah suatu kata atau *term* ada atau tidak dalam dokumen, jika ada diberi nilai satu (1), jika tidak diberi nilai nol (0).
2. *TF* murni *(raw TF),* nilai *TF* diberikan berdasarkan jumlah kemunculan suatu *term* di dokumen. Contohnya, jika muncul lima (5) kali maka kata tersebut akan bernilai lima (5).
3. *TF logaritmik*, hal ini untuk menghindari dominansi dokumen yang mengandung sedikit *term* dalam *query*, namun mempunyai frekuensi yang tinggi.

$$TF = \begin{cases} 1 + \log(f_{t,d}), & f_{t,d}>0 \\ 0, & f_{t,d}=0 \end{cases} \qquad (2.12)$$

Dimana nilai ft,d adalah frekuensi *term* (t) pada document (d). Jadi jika suatu kata atau *term* terdapat dalam suatu dokumen sebanyak 5 kali maka diperoleh bobot = 1 + log (5) = 1.699. Tetapi jika term tidak terdapat dalam dokumen tersebut, bobotnya adalah nol (0).

4. *TF* normalisasi, menggunakan perbandingan antara frekuensi sebuah *term* dengan nilai maksimum dari keseluruhan atau kumpulan frekuensi *term* yang ada pada suatu dokumen.

$$TF = 0.5 + 0.5 * \left[\frac{f_{t.d}}{\max(f_{t',d:t',d\in d})}\right] \quad (2.13)$$

DF (Inverse Document Frequency) merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan. *IDF* menunjukkan hubungan ketersediaan sebuah *term* dalam seluruh dokumen. Semakin sedikit jumlah dokumen yang mengandung *term* yang dimaksud, maka nilai *IDF* semakin besar. Sedangkan untuk *Inverse Document Frequency* (*IDF*) dihitung dengan menggunakan formula sebagai berikut:

$$IDF_j = \log(D/df_j) \quad (2.14)$$

Dimana D adalah jumlah semua dokumen dalam koleksi sedangkan dfj adalah jumlah dokumen yang mengandung *term* (tj). Jenis formula TF yang biasa digunakan untuk perhitungan adalah TF murni (*raw* TF). Dengan demikian rumus umum untuk *Term Weighting TF-IDF* adalah penggabungan dari formula perhitungan *raw* TF dengan formula *IDF* dengan cara mengalikan nilai *TF* dengan nilai IDF:

$$w_{ij} = tf_{ij} * idf_j \quad (2.14)$$

$$w_{ij} = tf_{ij} * \log(\frac{D}{idf_j}) \quad (2.15)$$

Dimana Wij adalah bobot *term* (tj) terhadap dokumen (di). Sedangkan tfij adalah jumlah kemunculan *term* (tj) dalam dokumen (di). D adalah adalah jumlah semua dokumen yang ada dalam *database* dan dfj adalah jumlah dokumen yang mengandung *term* (tj) (minimal ada satu kata yaitu *term* (tj)).

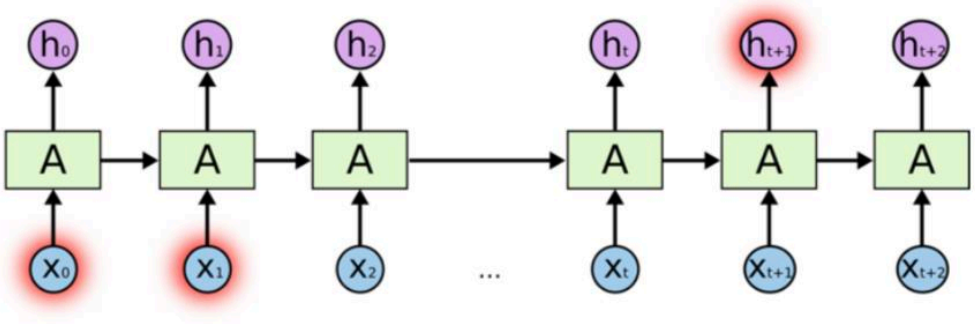
Berapapun besarnya nilai tfij, apabila D = dfj, maka akan didapatkan hasil 0 (nol), dikarenakan hasil dari log 1, untuk perhitungan *IDF*. Untuk itu dapat ditambahkan nilai 1 pada sisi *IDF*, sehingga perhitungan bobotnya menjadi sebagai berikut :

$$w_{ij} = tf_{ij} * \log(\frac{D}{idf_j}) + 1 \quad (2.16)$$

E. LSTM *(Long Short Term Memory).*

Long Short Term Memory (LSTM) merupakan sebuah evolusi dari arsitektur RNN, dimana pertama kali diperkenalkan oleh Hochreiter & Schmidhuber pada tahun 1997. Hingga penelitian ini dilakukan banyak para peneliti yang terus

mengembangkan arsitektur LSTM di berbagai bidang seperti dalam bidang *speech recognition* dan *forecasting*.



Gambar 9. Memori pada RNN

Pada gambar 9. menjelaskan RNN memiliki kekurangan, kekurangan itu dapat dilihat pada inputan X_0 , X_1 memiliki rentang informasi yang sangat besar dengan X_t , X_{t+1} sehingga ketika X_{h+1} memerlukan informasi yang relevan dengan X_0 , X_1 RNN tidak dapat untuk belajar menghubungkan informasi karena memori lama yang tersimpan akan semakin tidak berguna dengan seiringnya waktu berjalan karena tertimpa atau tergantikan dengan memori baru. Berbeda dengan RNN, LSTM tidak memiliki kekurangan tersebut karena LSTM dapat mengatur memori pada setiap masukannya dengan menggunakan *memory cells* dan *gate units.*

F. Klasifikasi.

Klasifikasi adalah sebuah proses untuk menemukan sebuah model yang menjelaskan dan membedakan konsep atau kelas data dengan tujuan memperkirakan kelas dari suatu objek yang kelasnya tidak diketahui. Dalam klasifikasi, diberikan sejumlah record yang dinamakan data latih, yang terdiri dari beberapa atribut yang dapat berupa kontinu ataupun kategoris, salah satu atribut menunjukkan kelas untuk record. Tujuan dari klasifikasi adalah untuk:

1. Menemukan model dari data latih yang membedakan record kedalam kategori atau kelas yang sesuai, model tersebut kemudian digunakan untuk mengklasifikasikan *record* yang kelasnya belum diketahui sebelumnya pada testing set.
2. Mengambil keputusan dengan memprediksi suatu kasus, berdasarkan hasil klasifikasi yang diperoleh.

Untuk mendapatkan model, harus dilakukan analisis terhadap data latih, Sedangkan data uji digunakan untuk mengetahui tingkat akurasi dari model yang telah dihasilkan. Klasifikasi dapat digunakan untuk memprediksi nama atau nilai kelas dari suatu objek data.

Penggunaan model menguraikan pengklasifikasian data yang akan diuji ataupun objek yang belum diketahui. Adapun parameter keberhasilan dari model yang terdiri dari:

1. Label yang telah diketahui dari data latih dibandingkan dengan hasil klasifikasi dari model.

2. Nilai akurasi adalah persentase dari kumpulan data uji yang diklasifikasikan secara tepat oleh model.
3. Data uji tidak sama dengan data latih.
4. Jika sesuai, gunakan model untuk mengklasifikasi data record yang label kelasnya belum diketahui.

G. *Confusion Matrix.*

Confusion matrix adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data mining atau Sistem Pendukung Keputusan. Pada pengukuran kinerja menggunakan confusion matrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi (*Ilmukomputer.com, 2018, Pengujian Dengan Confusion Matrix*). Keempat istilah tersebut adalah.

TP (*True Positive*) adalah kelas yang diprediksi positif dan benar.
TN (*True Negatif*) adalah kelas yang diprediksi negatif dan benar.
FP (*False Positive*) adalah kelas yang diprediksi positif dan salah.
FN (*False Negatif*) adalah kelas yang diprediksi negatif dan salah.

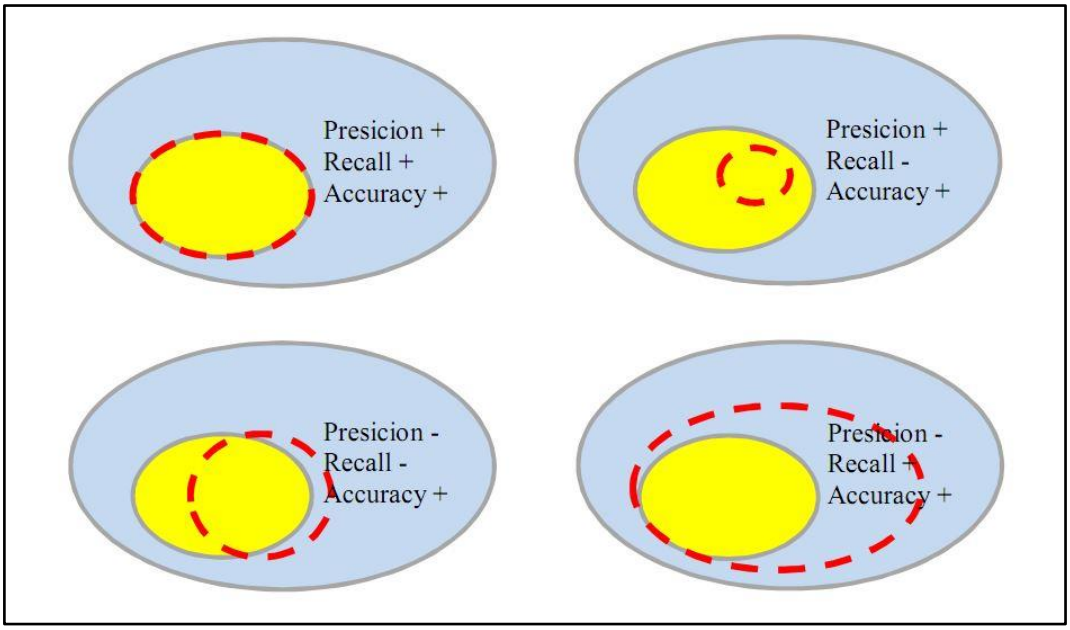
Tabel 7. *Confusion matrix*

		Kelas Prediksi	
		Positif	Negatif
Observasi	Positif	TP	FN
	Negatif	FP	TN

Sehingga akurasi dari klasifikasi dapat diperoleh dari penjumlahan true positif dan true negative dibagi total untuk melihat kinerja secara keseluruhan dengan rumus berikut:

$$\text{Akurasi} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}} \qquad (2.12)$$

Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) dapat diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. *Recall* menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem.



Gambar 10. Perbandingan *accuracy recall* dan *presicion*

H. *Naive Bayes.*

Naive Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan setiap frekuensi dan kombinasi nilai dari dataset yang diberikan. *Naive Bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai *output*. Keuntungan penggunaan *Naive Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*Training Data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. *Naive Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan.

Persamaan dari teorema *Bayes* adalah:

Di mana	:
X	: Data dengan class yang belum diketahui
H	: HipoPenelitian data merupakan suatu class spesifik
P(H|X)	: Probabilitas hipoPenelitian H berdasar kondisi X (posteriori probabilityas)
P(H)	: Probabilitas hipoPenelitian H (prior probabilitas)
P(X|H)	: Probabilitas X berdasarkan kondisi pada hipoPenelitian H
P(X)	: Probabilitas X

2.2.Tinjauan Studi.

Permasalahan berita hoax tentu menjadi perhatian para peneliti baik di Indonesia maupun di dunia hal ini tercermin dari beberapa penelitian yang telah dilakukan sebelumnya. Penelitian – penelitian tersebut diantaranya dapat dilihat pada tabel 8. Berikut ini :

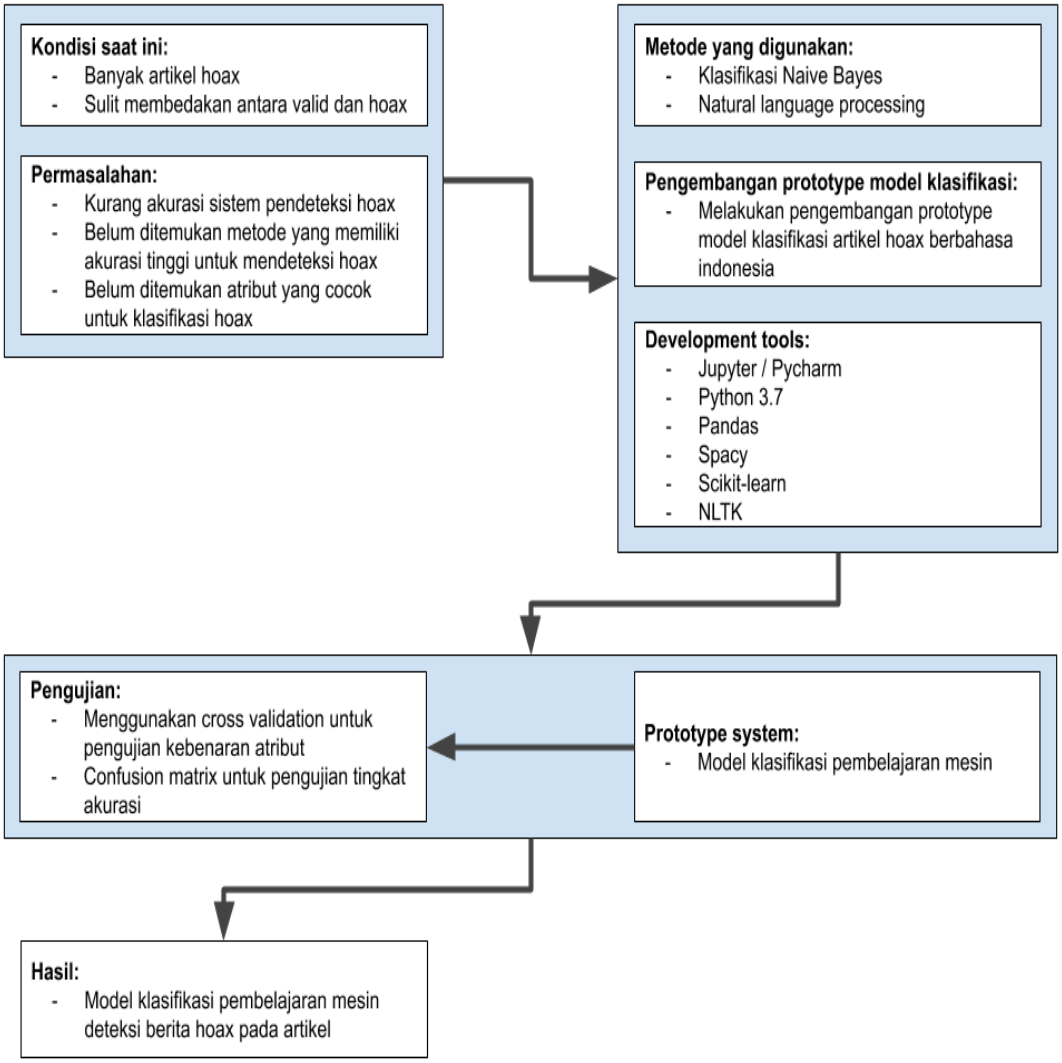
Tabel 8. Tinjauan studi

Nama peneliti	Penelitian	Metode	Hasil
(Pratiwi et al., 2017)	Study of hoax news detection using naïve bayes classifier in Indonesian language	Membuat dataset berita hoax dan membuat sistem deteksi hoax menggunakan Naive Bayes	Berdasarkan tiga kali random pada training dan testing dataset diperoleh rata-rata tertinggi pada 70% training set dan 30% testing set dengan akurasi 78,6% hoax presisi adalah 67,1% presisi valid adalah 91,6%, recall hoax adalah 89,4% dan recall valid adalah 71,4.
(Prasetijo, A. B. Isnanto, R. R. Eridani, D. Soetrisno, Y. A.D. Arfan, M. Sofwan dan Aghus, 2018)	Hoax detection system on Indonesian news sites based on text classification using SVM and SGD	Klasifikasi berita hoax berbahasa Indonesia dengan representasi vektor teks berdasarkan Term Frequency dan frekuensi document. Teknik klasifikasi menggunakan, Support Vector Machine dan Stochastic Gradient Descent.	Menggunakan SGD dengan kernel dengan modified-huber kernel dapat meningkatkan akurasi dan ketepatan SVM masing-masing sekitar 4% dan 20%. Keakuratan TF-IDF lebih baik dikombinasikan dengan SGD ketika sampel klasifikasi hoax tidak memiliki ketentuan khusus, sebaliknya, ia memiliki pola unik dari sebuah portal berita yang sama.
(Wapna et al., 2019)	Fake News Detection Using Naive Bayes Classifier	Pendekatan sederhana mendeteksi berita palsu menggunakan klasifikasi Naïve Bayes yang diuji menggunakan kumpulan data posting berita di Facebook.	Sistem deteksi hoax dengan akurasi 74% hoax dan menyebutkan bahwa masalah berita palsu atau hoax dapat diselesaikan dengan metode pembelajaran mesin.
(F. Rahutomo, I. Y. R. Pratiwi dan D. M. Ramadhani, 2019)	Eksperimen pada Sistem Klasifikasi Berita Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin	Fitur term frequency dan algoritma klasifikasi naïve Bayes dengan menggunakan komponen library PHP-ML atau PHP-Machine Learning.	Berdasarkan hasil uji coba secara statis, sistem ini menghasilkan akurasi sebesar 82,6% dan pengujian secara dinamis persentase kesesuaian dengan sistem 68,33%.

(Chen et al., 2014)	Email Hoax Detection System Using Levenshtein Distance Method	Deteksi email hoax dengan memasukan metode pencocokan text menggunakan metode Levenshtein Distance measure.	Sistem ini mampu memberikan nilai prediksi positif tinggi 0,96 tetapi tidak memiliki kemampuan untuk mengidentifikasi email asli
(Thota, 2018)	Fake News Detection: A Deep Learning Approach	Perbandingan tingkat kesamaan text.	Menggunakan model Tf-IDF - Dense neural network (DNN) yang tersetel halus, mampu mengungguli arsitektur model yang ada sebesar 2,5% dan kami mampu mencapai akurasi 94,21% pada data uji.

2.3. Kerangka Konsep

Berdasarkan identifikasi masalah, tujuan penelitian, kajian teori, studi dari penelitian sebelumnya dan juga topik penelitian maka dapat dibangun kerangka konsep penelitian tentang model klasifikasi berita *hoax* pada artikel berbahasa Indonesia seperti terlihat pada gambar 11. dibawah berikut ini:



Gambar 11. Kerangka konsep

2.4. HipoPenelitian

Diduga dengan menambahkan 2 atribut yang diluar dari atribut hasil *preprocessing* kalimat artikel seperti atribut *website* asal artikel dan status *website* pada metode yang digunakan penelitian sebelumnya maka akan lebih sesuai untuk metode klasifikasi *naïve bayes* dan menghasilkan akurasi lebih baik.

BAB III
METODOLOGI DAN RANCANGAN

3.1. Metode Penelitian

Maksud dari penelitian ini adalah menambahkan atribut untuk mengklasifikasikan atrikel berita valid dan berita *hoax* pada penelitian sebelumnya yang menggunakan metode klasifikasi *naïve bayes* agar lebih akurat. Untuk mengetahuinya perlu dilakukan pengujian atribut terhadap metode klasifikasi *naïve bayes*. Hal ini dilakukan untuk mengetahui apakah ada pengaruh penambahan atribut terhadap metode klasifikasi *naïve bayes*.

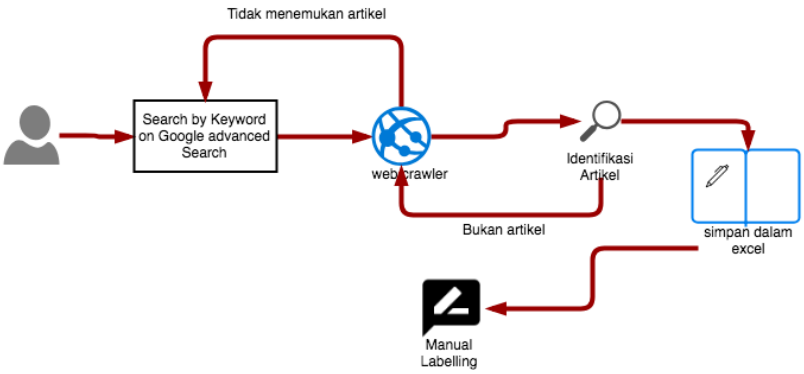
Berdasarkan maksud dan ruang lingkup penelitian ini, penelitian ini dilakukan dengan menggunakan metode eksperimen. Metode eksperimen ini dilakukan peneliti dengan memanipulasi kondisi sesuai dengan kebutuhan permasalahan yang dihadapi di dalam penelitian. Dengan memanipulasi kondisi ini,nantinya hasil dari penelitian ini akan menghasilkan algoritma dengan tingkat akurasi yang lebih tinggi dari penelitian sebelumnya.

3.2. Metode Pengumpulan Data

Dataset dalam penelitian ini menggunakan data berdasarkan penelitian yang dilakukan (Pratiwi, Asmara dan Rahutomo, 2017), yang terdiri dari 250 data artikel dalam bahasa Indonesia. Artikel hoax dan bukan hoax yang terdiri dari 10 topik yang berbeda dan setiap topik terdiri dari 25 berita. Topik berita tersebut adalah :

a. Makan lele menyebabkan sel kanker.
b. Aku puntur dengan jarum menyebabkan stroke.
c. Iphone 6 mudah dibengkokan.
d. Reog Ponorogo dibakar di Filipina.
e. Simpatisan Aksi 212 dilarang masuk masjid Istiqlal.
f. Sikat gigi dari rambut babi.
g. Permen dot mengandung narkoba.
h. Pokemon berarti “Aku Yahudi”
i. Foto Awan Berdoa Di Pemakaman Uje
j. Munarman pengacara Freeport

Metode pengumpulan yang dilakukan dapat dilihat pada gambar 12. bawah ini:



Gambar 12. Pengumpulan Data

Langkah pertama ialah mencari dengan kata kunci di google advance search kemudian sistem melakukan *webcrawler*, kemudian sistem mengidentifikasi artikel, apabila menemukan artikel maka dilanjutkan proses selanjutnya namun sebaliknya apabila tidak ditemukan artikel akan memberikan notifikasi bahwa tidak ditemukan artikel dengan keyword tersebut.

1. *Web Crawler*
   Web crawler adalah suatu metode untuk mencari atau menelurusuri informasi suatu halaman atau kumpulan halaman. Tidak hanya menelusuri, akah tetapi juga mengambil informasi dari halaman. Fungsi utama sebuah *web crawler* adalah untuk mencari atau mengambil informasi dari suatu halaman(Pratiwi et al., 2017).
2. Indetifikasi Artikel
   Identifikasi artikel adalah untuk menentukann apakah suatu *website* mengandung sebuah artikel. Artikel berita harus berbentuk cerita kronologi atau harus setidaknya terdiri dari dua paragraf atau laporan 8 baris. Kondisi ini diatur untuk membuat pengecualian pada halaman yang mungkin terdiri dari rangkuman (yang biasanya terdapat link yang mengarah ke *website* selanjutnya).
   Untuk memastikan jika hasil pencarian seperti berita dan *blog review* mengandung *file pdf*, dokumen, gambar dll adalah dengan memakai *google advanced search*. Sebagai contoh kita dapat mengetikan kata pencarian "Dot mengandung babi". Pada url *google*, akan muncul detail *URL* yang kemudian dapat di *copy* pada *source code coding*. Berikut adalah contohnya:
   https://www.google.co.id/search?q=dot+mengandung+babi&tbm=nws&source=lnms&sa=X&ved=0ahUKEwj6lvqjsLXcAhUTcCsKHYTrC04Q_AUIDCgD&biw=1546&bih=903&dpr=2
3. *HTML Tag Removal*.
   Tag HTML dan tanda baca pada kalimat dihilangkan menggunakan HTML *parser* dengan *DOM* (*document Object Model)*. Setiap file HTML dapat dipetakan menggunakan *DOM*. Selain tag <*p*> dan </ *p*>pada file akan dihapus, sistem hanya akan mengambil konten pada tag tag hingga .Untuk menentukan teks utama artikel suatau *web* adalah dalam tag ... </ p>, jumlah maksimum karakter harus dibatasi. Namun, pada penelitian ini, artikel yang akan diteliti setidaknya harus memiliki setidaknya 500 karakter atau lebih, sehingga dapat membantu untuk memutuskan bagian mana artikel itu dimulai dan diakhiri.
4. Input file ke *Microsoft Excel*
   Setelah artikel dilakukan proses *HTML tag removal* langkah selanjutnya sistem akan memasukan ke file hasil *HTML tag removal*tersebut ke dalam *Microsoft Excel* file.
5. Menandai file tersebut dengan label *hoax* atau valid.
   Setelah semua *file* tersebut berada pada *file excel* kemudian dilakukan validasi manual terhadap setiap *file*. Terdapat 3 orang *reviewer* yang akan melakukan validasi dengan cara melihat setiap artikel dari portal berita

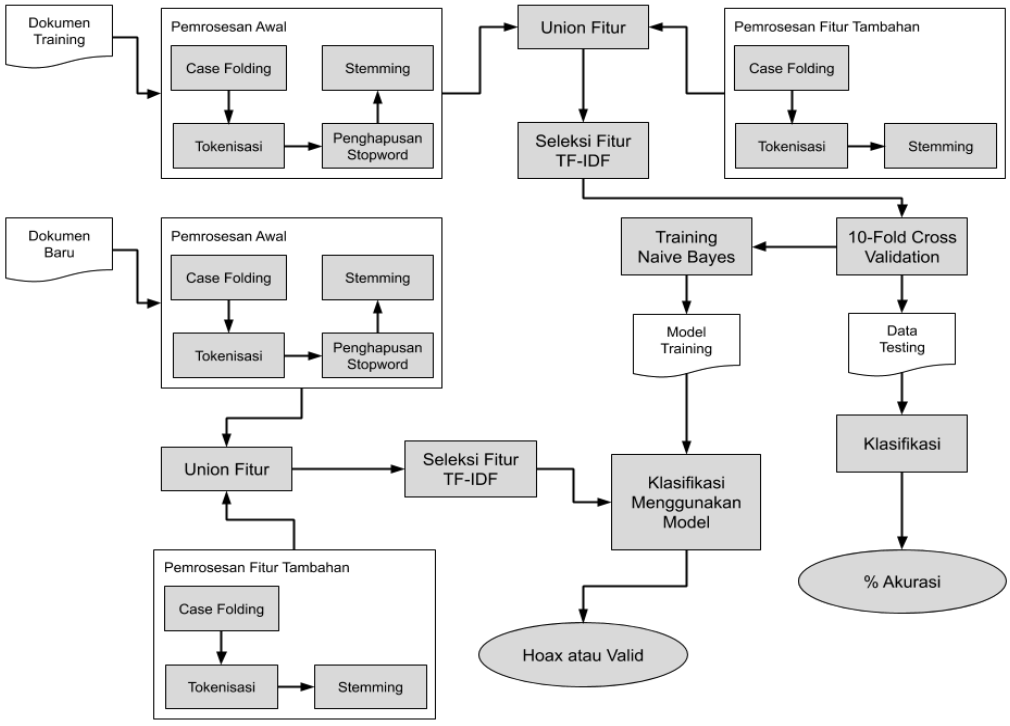
mainstream di Indonesia, dari hasil 3 orang tersebut dibuat *voting* yang akan dijadikan label pada setiap *file* tersebut.

3.3. Sistem Usulan

Pada Penelitian ini penulis mengusulkan tambahan atribut dalam klasifikasi teks berita *hoax* yang dapat digunakan untuk mendeteksi apakah suatu artikel berita tersebut *hoax* atau tidak. Atribut yang ditambahkan adalah *website* yang mempublikasi artikel dan status *website* yang masih aktif atau tidak, atribut tersebut akan diterapkan pada klasifikasi naïve bayes. Di Indonesia terdapat seitar 43.000 *website* yang mengklaim sebagai portal berita menurut catatan Dewan Pers. *Website* yang terverifikasi sebagai portal berita resmi tidak mencapai 300 *website.* Dari perbandingan jumlah *website* dan yang *website* telah terverifikasi dapat disimpulkan terdapat puluhan ribu yang berpotensi menyebarkan berita hoax. Informasi yang berasal dari website tidak terverifikasi seperti blog pribadi perlu diwaspadai karena kemungkinan berita tersebut adalah *hoax* (Yunita dan Kominfo, 2017).

Sistem yang diusulkan pada penelitian ini dibagi menjadi dua tahap yaitu tahap pelatihan dan pengujian. Pada tahap pelatihan digunakan untuk menciptakan model klasifikasi artikel berita itu *hoax* atau bukan, sedangkan dalam tahap pengujian untuk mengklasifikasikan apakah artikel atau dokumen masukan tersebut *hoax* atau bukan.

Pada pembangunan model, terdapat 4 tahap utama yaitu praproses, ekstraksi fitur, seleksi fitur atau penambahan fitur dan pelatihan. Setiap tahap dapat dilihat pada gambar 13. berikut ini :



Gambar 13. Rancangan usulan sistem

3.3.1. Praproses

Teks bisa dalam berbagai bentuk dari daftar kata-kata, hingga kalimat ke beberapa paragraf dengan karakter khusus. Seperti halnya masalah *data sciense*, memahami pertanyaan yang ditanyakan akan menginformasikan langkah apa yang dapat digunakan untuk mengubah kata-kata menjadi fitur numerik dengan menggunakan algoritma pembelajaran mesin. Tahap praproses adalah tahapan dimana aplikasi melakukan seleksi data yang akan diproses pada setiap dokumen. Praproses digunakan untuk merubah teks yang tidak terstruktur menjadi token representasi yang siap dimodelkan oleh algoritma klasifikasi. Praproses terdiri dari pemrosesan leksikal dan pengubahan kata ke fitur kata. Pemrosesan leksikal meliputi: (1) *case folding*, (2) tokenisasi, (3) penghapusan *stopword*, dan (4) *stemming*.

1. ***Case Folding***
   Menurut Errisa dalam penelitiannya mengatakan bahwa case folding adalah proses yang dilakukan untuk menghapus karakter selain huruf dalam dokumen pada saat pengambilan informasi (Errisa & Ayu, 2016). Dengan fitur ini kita dapat secara otomatis mengubah semua huruf pada teks menjadi huruf kecil semua.
2. **Tokenisasi**
   Proses tokenisasi adalah proses memisahkan kalimat menjadi kata atau frase (Errisa & Ayu, 2016). Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca serta memfilter berdasarkan panjang teks. Pada penelitian ini proses tokenisasi yang dilakukan ada memisahkan setiap kata dari kalimat menggunakan spasi, kemudian kata tersebut disimpan dalam sebuah array atau larik.
3. **Penghapusan *Stopword***
   Dalam *NLP* (*Natural Language Processing*) *stopword* merupakan kata yang diabaikan dalam pemrosesan, kata-kata ini biasanya disimpan ke dalam *stop lists*. Karakteristik utama dalam pemilihan *stopword* biasanya adalah kata yang mempunyai frekuensi kemunculan yang tinggi misalnya kata penghubung seperti “dan”, “atau”, “tapi”, “akan” dan lainnya. Tidak ada aturan pasti dalam menentukan *stopword* yang akan digunakan, penentuan *stopword* bisa disesuaikan dengan kasus yang sedang diselesaikan. Tujuan utama dalam penerapan proses *Stopword Removal* adalah mengurangi jumlah kata dalam sebuah dokumen yang nantinya akan berpengaruh dalam kecepatan dan peforma dalam kegiatan *NLP*. Proses ini bertujuan untuk mengurangi volume kata. *Stopwords* dapat berupa kata depan, kata penghubung, dan kata pengganti (Errisa & Ayu, 2016). Fiter *stopword* bahasa indonesia ini penulis menggunakan filter *stopword* bahasa Indonesia yang dibuat oleh *Sastrawi*.
4. ***Stemming***
   Stemming adalah proses mengubah kata berimbuhan menjadi kata dasar. Aturan-aturan bahasa diterapkan untuk menanggalkan imbuhan-imbuhan tersebut (Errisa & Ayu, 2016). Pada penelitian ini penulis menggunakan *Sastrawi stemmer* karena *Sastrawi stemmer* menerapkan algoritma yang berbasis Nazief dan Adriani, kemudian ditingkatkan oleh Algoritma *CS*

(*Confix Stripping*), kemudian ditingkatkan lagi oleh algortima *ECS* (*Enhanced Confix Stripping*), lalu ditingkatkan lagi oleh *Modified ECS*. Dengan menggunakan algoritma-algoritma tersebut, banyak persoalan *stemming* berhasil diatasi:

- mencegah *overstemming* dengan kamus kata dasar.
- mencegah *understemming* dengan aturan-aturan tambahan.
- kata bentuk jamak berhasil distem: Buku-buku -> buku.

3.3.2. Ekstraksi Fitur

Ekstraksi fitur adalah proses mengekstrak seluruh fitur kata yang terdapat dalam dokumen latih. Keluaran dari proses ini adalah kumpulan kata yang dijadikan penciri dokumen berita *hoax* dan bukan *hoax* (Errisa & Ayu, 2016). Fitur kata ini di dapatkan dari proses tokenisasi.

3.3.3. Seleksi Fitur

Seleksi fitur dilakukan sebelum proses klasifikasi terhadap *dataset*. Pada penelitian ini penulis menggunakan algoritma *TF-IDF* (*Term Frequency – Inverse Document Frequency*) yang digabungkan dengan algoritma *SGD* (*Stochastic Gradient Descent*) karena menurut penelitian dari (Agung & Yosua) mengatakan bahwa keakuratan *TF-IDF* lebih baik dikombinasikan dengan *SGD* ketika sampel klasifikasi tidak memiliki ketentuan khusus, sebaliknya, ia memiliki pola unik di penyedia portal berita yang sama.

3.3.4. Klasifikasi

Pengkalsifikasian data dilakukan terhadap dataset berdasarkan atribut yang dimiliki oleh dataset berita *hoax* tersebut. Pada penelitian ini penulis menggunakan klasifikasi *naïve bayes*.

Untuk merepresentasikan sebuah kelas dokumen, terdapat karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi yang berguna untuk menjelaskan bahwa peluang masuknya sampel karakteristik tertentu kedalam kelas *posterior*. Klasifikasi *naïve bayes* diasumsikan bahwa ada atau tidaknya ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya. Persamaan dari teorema bayes adalah (E. Kusrini dan L. Taufiq, 2009):

$$P(H|X)=(P(X|H)P(H))/(P(X)) \quad (3.1)$$

Di mana nilai X adalah data kelas yang belum diketahui, H adalah hipoPenelitian X pada label tertentu, P(H|X) adalah probabilitas H berdasarkan kondisi X (*posteriori*), P(H) adalah probabilitas H (prior), P(X|H) adalah probabilitias X.

3.4. Pengukuran Akurasi.

Dalam menguji keefektifan suatu klasifikasi dibutuhkan suatu pengukuran evaluasi. Pengukuran tersebut didapatkan dalam sebuah set *confusion matrix* (Sarkar). Kondisi yang digunakan dalam pengujian adalah sebagai berikut:

1. *True Positive* adalah kondisi berita *hoax* diklasifikasi sebagai *hoax*,

2. *True Negative* adalah kondisi berita bukan *hoax* diklasifikasikan sebagai bukan *hoax*,
3. *False Positive* adalah kondisi berita *hoax* diklasifikasikan sebagai bukan *hoax*,
4. *False Negative* adalah kondisi berita bukan *hoax* diklasifikasikan sebagai berita *hoax*.

3.5. Pengujian Sistem dan Analisis Sistem

3.5.1. Pengujian Sistem

Pengujian sistem yang dilakukan pada penelitian ini untuk melihat performansi atribut tambahan pada klasifikasi *naïve bayes* dalam melakukan prediksi terhadap data testing. Performansi diukur dengan melakukan perbandingan antara hasil testing yang diklasifikasikan oleh sistem dengan data testing yang sebelumnya telah diberi label.

3.5.2. Tujuan Pengujian Sistem

Tujuan dilakukannya pengujian ini adalah:

a. Menganalisis pengaruh perbandingan data training dan data testing terhadap performansi sistem.
b. Menganalis pengaruh jumlah dataset dalam pembentukan model klasifikasi
c. Menganalisis hasil klasifikasi yang didapatkan

3.5.3. Skenario Pengujian Sistem

Pada penelitan ini menulis menggunakan metode *10-fold cross validation* seperti yang dapat dilihat pada gambar 2., metode ini bertujuan untuk :

a. Perbandingan jumlah dataset Pengujian perbandingan jumlah dataset dilakukan dengan mengubah jumlah dataset yang digunakan pada setiap pengujian. Analisa dilakukan dengan membandingkan akurasi yang didapat pada setiap model dengan jumlah dataset yang berbeda.
b. Perbandingan komposisi data training dan data testing Pengujian perbandingan komposisi data training dan data testing dilakukan untuk melihat pengaruh pada model yang dibuat oleh sistem klasifikasi. Pengujian ini dilakukan dengan mengubah komposisi data training dan data testing kemudian membandingkan dan menganalisa hasil akurasi dari masing-masing komposisi.
c. Analisis hasil klasifikasi pada sistem klasifikasi deteksi hoax

3.6. *Instrumentasi*

Instrumen penelitian adalah alat bantu yang digunakan oleh peneliti untuk mengumpulkan data penelitian. Pada penelitian ini penulis menggunakan beberapa instrumen untuk mendukung penelitian yang dilakukan.

3.6.1. Perangkat Lunak

Berikut beberapa perangkat lunak yang digunakan dalam penelitian ini.

1. Sistem Operasi Windows 8
2. Jupyter / Pycharm
3. Python versi 3.7
4. Scrapy untuk *web spider crawler*
5. Scikit-Learn (library *machine-learning*) versi 0.16.1
6. Sastrawi (*library* untuk *stemming* & penghapusan *stopword*)
7. Pandas untuk *import file .csv*

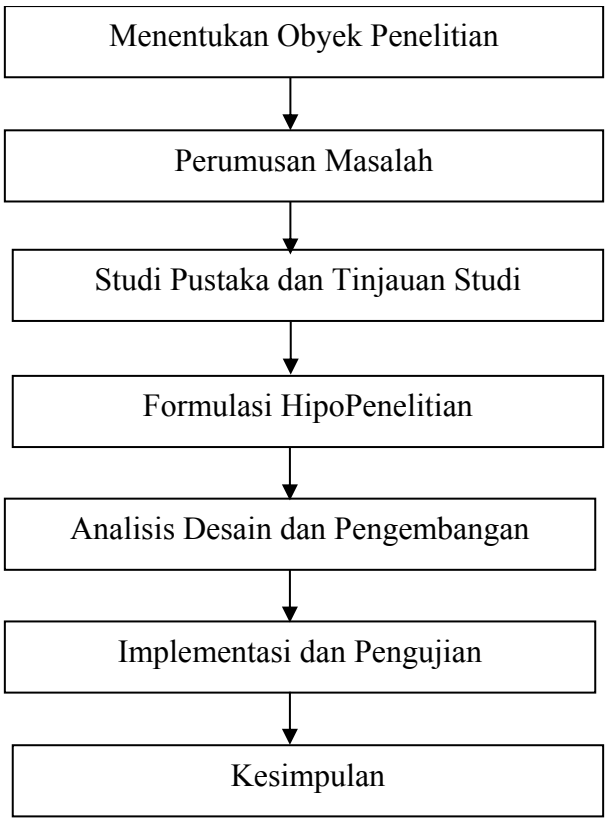
3.6.2. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah macbook yang memiliki spesifikasi sebagai berkut :

Processor	: Intel(R) Core i5
Memory	: 8 GB RAM
Hard Disk	: 250 GB

3.7. Langkah-Langkah Penelitian

Untuk mencapai tujuan yang diharapkan, beberapa langkah-langkah penelitian yang dilakukan. Bagan alur proses langkah-langkah peneltian untuk lebih jelasnya dapat lihat di gambar 14. berikut ini :



Gambar 14. Langkah-langkah penelitian

3.7.1. Menentukan Obyek Penelitian

Topik penelitian dipilih berdasarkan keprihatinan penulis terhadap banyaknya berita *hoax* yang ada di Indonesi. Berangkat dari hal tersebut maka penulis melakukan beberapa kajian terhadap beberapa penelitian yang telah dilakukan oleh beberapa orang namun penulis masih belum menemukan akurasi dari yang tinggi dari solusi yang mereka tawarkan kalaupun ada maka masih terdapat kekurangan – kekurangan. Dari penjabaran tersebut maka penulis bisa menarik kesimpulan bahwa penelitian pada topik ini masih sangat terbuka lebar.

3.7.2. Perumusan Masalah

Perumusan masalah pada penelitian ini adalah memeriksa semua literatur yang berkaitan dengan topik yang penulis usulkan kemudian menganalisa setiap solusi yang ditawarkan serta kesimpulan dari setiap penelitian tersebut. Dari langkah – langkah tersebut kemudian penulis merumuskan masalah – masalah yang dihadapi dalam penelitan ini serta mencoba meniliti solusi yang mungkin bisa ditawarkan.

3.7.3. Studi Pustaka dan Tinjauan Studi

Studi pustaka dilakukan dengan melakukan penelitian terhadap kepustakaan yang berhubungan dengan obyek penelitian antisipasi berita hoax menggunakan *natural language processing* dan *machine learning*. Bahan-bahan studi pustaka yang digunakan untuk menjadi referensi dalam penelitian ini. Tinjauan studi dilakukan untuk menambah wawasan serta acuan dalam mengerjakan penelitian ini.

3.7.4. Formulasi HipoPenelitian

Formulasi hipoPenelitian pada penelitian ini bertujuan untuk meningkatkan akurasi dari sistem atau metode mengklasifikasikan suatu artikel itu *hoax* atau bukan. Penerapan metode ini diharapkan dapat memberikan kemudahan dalam mendeteksi apakah suatu artikel itu merupakan berita *hoax* atau bukan.

3.7.5. Analisis Desain dan Pengembangan

Pada tahap ini dianalisa beberapa kriteria serta beberapa algoritma yang telah dipakai dan dibuktikan oleh penelitian sebelumnya kemudian mencari suatu cara dalam meningkatkan akurasi dari algoritma klasifikasi teks. Analisa penulis ialah dengan menambahkan atribut, atribut yang sudah ada akan digabungkan dengan atribut yang baru maka dapat dicapai peningkatan akurasi.

3.7.6. Implementasi dan Pengujian

Pengujian dari analisis bertujuan untuk mengetahui hasil penelitian dari metode yang digunakan. Pengujian ini untuk mengetahui performasi dan tingkat akurasi dari metode klasifikasi berita hoax. Untuk melakukan analisa klasifikasi berita hoax secara garis besar dilakukan dengan beberapa tahap, tahap pertama ialah praproses, tahap kedua ekstrasi fitur, tahap seleksi fitur dan tahap klasifikasi menggunakan metode yang diusulkan dalam penelitian ini.

Pengukuran akurasi pada penelitian ini menggunakan *confusion matrix*. Pengujian akurasi klasifikasi dengan dilakukan dengan beberapa eksperimen seperti eksperimen pada extrasi fitur dan seleksi fitur, untuk mendapatkan hasil yang lebih akurat.

3.7.7. Kesimpulan

Berdasarkan penelitian dengan berbagai cara klasifikasi dan gabungan antar atribut diharapkan dapat meningkatkan akurasi klasifikasi dengan metode *naïve bayes*. Agar suatu dokumen dapat berguna dan dapat dijadikan pembentukan model penciri suatu artikel *hoax* maka harus melalui beberapa tahapan diantaranya praproses dan esktrasi fitur.

BAB IV
PEMBAHASAN DAN HASIL PENELITIAN

4.1. Analisa Sistem

Analisis sistem dilakukan untuk mengetahui kebutuhan yang diperlukan untuk membangun sistem prediksi berita hoax. Selain itu, analisis sistem juga akan berguna untuk perancangan sistem prediksi berita hoax tersebut. Pada sistem prediksi berita hoax ini dibuat berdasarkan pola atau gaya penulisan yang sering digunakan dalam berita hoax dan dalam berita yang valid atau bukan hoax. Proses yang akan dilakukan untuk membuat sistem prediksi adalah proses pelatihan yang akan dilanjutkan dengan proses pengujian untuk mendapatkan hasil prediksinya. Data berita yang digunakan pada sistem ini terbatas yaitu terdiri dari 250 data artikel dalam bahasa Indonesia. Artikel hoax dan bukan hoax yang terdiri dari 10 topik yang berbeda dan setiap topik terdiri dari 25 berita.

Tabel 9. Data penelitian

ID	Articles	Tagging	Website	Status
1	Isu bahwa ikan lele mengandung sel kanker di jejaring sosial dan berita dari mulut ke mulut terus menyebar. Dampak dari isu tersebut para ibu-ibu enggan membeli ikan lele. Waspada Online berhasil merangkum komentar ibu-ibu yang biasanya membeli ikan lele untuk konsumsi rutin	*valid*	waspada.co.id	aktif
2	Ikan lele merupakan salah satu makanan favorit di Indonesia. Selain harganya murah, rasanya juga sangat enak. Meski demikian, ada sebagian masyarakat yang takut menikmati masakan dari ikan air tawar tersebut. Mereka beranggapan jika ikan lele penyebab kanker dan penyakit lainnya. Namun apakah anggapan yang menyatakan lele mengandung kanker tersebut benar?. Berikut ini penjelasannya. Habitat dan Kehidupan Ikan Lele.....	*valid*	transferfactorformula.com	aktif
3	Kepala Bagian Penerangan Umum (Kabagpenum) Polri Kombes Pol Martinus Sitompul memantah isu terjadi pelemparan Alquran oleh petugas jaga di Rutan Mako Brimob Cabang Salemba, Jakarta Pusat, Jumat (10/11/2017). Rutan Mako Brimob Cabang Salemba kerusuhan…….	*valid*	bacaberitaupdate.com	mati

…	…….	……..	…….	…… …
…	…….	……..	…….	…… …
…	…….	……..	…….	…… …
248	Rumah Tahanan (Rutan) Mako Brimob, Kelapa Dua, Depok, Jawa Barat, dikabarkan rusuh, Jumat (10/11/2017) sore. Kabar ini ramai dibicarakan di media sosial. "Mako Brimob rusuh tadi sore. Kabarnya 3 blok ruang tahanan terbakar & pintu2nya hancur. Menurut sumber saya, sebabnya karena sekelompok Polisi di sana melempar Al-Qur'an dan buku2 hadits milik tahanan," kicau pemilik akun Twitter @CondetWarrior……..	*hoax*	harianumum.com	aktif
249	Malam peringatan ulang tahun ke-90 Kolese Kanisius di Hall D JIExpo Kemayoran, Jakarta Utara, Sabtu (11/11/2017) lalu terus menjadi perbincangan hangat beberapa hari terakhir…….	*hoax*	netralnews.com	aktif
250	Tahun 2012 kita dihebohkan dengan munculnya kuas atau sikat yang bebahan bulu babi sampai-sampai MUI mengeluarkan fatwa haram menggunakan sikat bulu babi, namun baru-baru ini muncul kembali info sikat gigi yang berbahan bulu babi. Informasi bagi anda kaum Muslim agar lebih teliti dalam memilih sikat gigi, karena ada produk pembersih gigi yang menggunakan bahan dari yang tidak halal……..	*hoax*	pelangimuslim.com	mati

Pada dataset yang akan digunakan terdapat beberapa perbandingan dari proses *tagging* manual dan memeriksa status *website*. Proses *tagging* valid atau hoax dilakukan dengan menggunakan sistem *voting* 3 *reviewer*.

Tabel I0. Proses *voting label valid* atau *hoax*

No.	Artikel	Rev-1	Rev-2	Rev-3	Hasil
1	Isu bahwa ikan lele mengandung sel kanker di jejaring sosial dan berita dari mulut ke mulut terus menyebar. Dampak dari isu tersebut para ibu-ibu	*Valid*	*Valid*	*Valid*	*Valid*

	enggan membeli ikan lele. Waspada Online berhasil merangkum komentar ibu-ibu yang biasanya membeli ikan lele untuk konsumsi rutin				
2	Ikan lele merupakan salah satu makanan favorit di Indonesia. Selain harganya murah, rasanya juga sangat enak. Meski demikian, ada sebagian masyarakat yang takut menikmati masakan dari ikan air tawar tersebut. Mereka beranggapan jika ikan lele penyebab kanker dan penyakit lainnya. Namun apakah anggapan yang menyatakan lele mengandung kanker tersebut benar?. Berikut ini penjelasannya. Habitat dan Kehidupan Ikan Lele.....	*Valid*	*Valid*	*Hoax*	*Valid*
3	Kepala Bagian Penerangan Umum (Kabagpenum) Polri Kombes Pol Martinus Sitompul memantah isu terjadi pelemparan Alquran oleh petugas jaga di Rutan Mako Brimob Cabang Salemba, Jakarta Pusat, Jumat (10/11/2017). Rutan Mako Brimob Cabang Salemba kerusuhan…….	*Valid*	*Valid*	*Valid*	*Valid*
…	…….	……	……	……	……
…	…….	……	……	……	……
…	…….	……	……	……	……
248	Rumah Tahanan (Rutan) Mako Brimob, Kelapa Dua, Depok, Jawa Barat, dikabarkan rusuh, Jumat (10/11/2017) sore. Kabar ini ramai dibicarakan di media sosial. "Mako Brimob rusuh tadi sore. Kabarnya 3 blok ruang tahanan terbakar & pintu2nya hancur. Menurut sumber saya, sebabnya karena sekelompok Polisi di sana melempar Al-Qur'an dan buku2 hadits milik tahanan," kicau pemilik akun Twitter @CondetWarrior……..	*Hoax*	*Hoax*	*Hoax*	*Hoax*
249	Malam peringatan ulang tahun ke-90 Kolese Kanisius di Hall D JIExpo Kemayoran, Jakarta Utara, Sabtu (11/11/2017) lalu terus menjadi perbincangan hangat beberapa hari terakhir…….	*Hoax*	*Hoax*	*Hoax*	*Hoax*

250	Tahun 2012 kita dihebohkan dengan munculnya kuas atau sikat yang bebahan bulu babi sampai-sampai MUI mengeluarkan fatwa haram menggunakan sikat bulu babi, namun baru-baru ini muncul kembali info sikat gigi yang berbahan bulu babi. Informasi bagi anda kaum Muslim agar lebih teliti dalam memilih sikat gigi, karena ada produk pembersih gigi yang menggunakan bahan dari yang tidak halal……..	*Hoax*	*Hoax*	*Hoax*	*Hoax*

Hasil dari manual tagging dari 250 dataset artikel yang digunakan dalam penelitian dapat dilihat pada tabel 11. dibawah ini :

Tabel 11. Perbandingan *valid dan hoax*

Tagging	**Total**
Artikel Valid	159
Artikel Hoax	91

Untuk mengetahui status *website* aktif atau tidak, penulis mengunjungi *link website* yang ada dalam dataset pada penelitian sebelumnya satu per satu secara manual.

Tabel 12. Perbandingan status *website*

Status	**Total**
Aktif	192
Mati	58

Terdapat banyak jenis *website* yang telah tidak bisa di akses atau mati. terdapat beberapa *website* yang diblock oleh google, *webserver* mati dan juga domain yang sudah habis masa berlakunya.

Gambar 15.Contoh *website* yang telah mati

Artikel pada *dataset* tersebut akan di-*import* ke dalam *prototype* dengan menggunakan *libraries* pandas pada *pyhton* untuk dilakukan proses selanjutnya seperti preprocessing dan klasifikasi.

Out[140]:

	Articles	Tagging	Website	Status
0	Jakarta, Di jejaring sosial, banyak beredar in...	valid	health.detik.com	aktif
1	Isu bahwa ikan lele mengandung sel kanker di j...	valid	waspada.co.id	aktif
2	Bagi penikmat kuliner dengan bahan dasar ikan ...	valid	tongkatmaduraasli.com	mati
3	Ikan lele merupakan salah satu makanan favorit...	valid	transferfactorformula.com	mati
4	Ikan lele merupakan bahan makanan yang cukup p...	valid	tribunnews.com	aktif
5	"Dalam sesuap daging ikan lele, terkandung 3.0...	hoax	regional.kompas.com	aktif
6	Bahaya Mengkonsumsi Ikan Lele Yang Mengandung ...	hoax	mediamasha.com	aktif
7	Di jejaring sosial banyak beredar informasi y...	hoax	beritalive.com	aktif
8	Jakarta Sebuah artikel yang cukup viral di in...	valid	crystalsweb.com	aktif
9	Pada dasarnya tidak ada makanan yang membawa s...	valid	deherba.com	aktif

Gambar 16. *Import dataset* ke *prototype*

4.2. Implementasi

Pada tahap ini berisi hasil implementasi dari proses yang telah dirancang sebelumnya yaitu *preprocessing*, Ekstraksi fitur, seleksi fitur, dan implemenasi proses klasifikasi untuk prediksi berita *hoax*.

4.2.1 Implementasi Proses *Stemming*.

Dalam proses *stemming* digunakan untuk mengubah kata-kata dalam kalimat *dataset* menjadi kata dasar dalam morfologi bahasa Indonesia. Langkah-langkah proses *stemming* yaitu, hasil proses *cleaning* yang disimpan dalam bentuk xlsx kemudian dipanggil untuk dijadikan masukan pada proses *stemming*. *List content* dari seluruh dataset dipanggil dan dijadikan satu kemudian *list content* ini diganti nama variabelnya menjadi *df*, untuk memudahkan proses menulis kode.

Dalam penelitian ini, proses *stemming* dilakukan dengan menggunakan *Stemmer Spacy* yang merupakan pustaka *stemming Python open-source*. Pada proses ini hasil dari proses *stemming* akan di tampilkan ke layar dengan kata yang sebelum dirubah menjadi kata dasar untuk keperluan evaluasi apakah *stemmer* ini berkerja dengan baik atau tidak. Contoh kalimat yang telah melalui proses *stemming* adalah seperti terlihat pada Gambar 17. dibawah ini :

```
Out[20]: ['Ikan => Ikan',
          'lele => lele',
          'merupakan => rupa',
          'salah => salah',
          'satu => satu',
          'makanan => makan',
          'favorit => favorit',
          'di => di',
          'Indonesia => Indonesia',
          '. => .',
          'Selain => Selain',
          'harganya => harganya',
          'murah => murah',
          '  =>  ',
          'rasanya => rasa',
```

Gambar 17. Hasil proses *stemming*

Dari gambar 17. diatas dapat dilihat bahwa hasil dari proses *stemming* menggunakan *stemmer spacy* berjalan dengan baik. Hal ini dibuktikan dengan kata "merupakan" berubah menjadi "rupa" yang memang merupakan kata dasar dari kata "merupakan".

4.2.2 Implementasi Proses *Case Folding*

Pada proses *casefolding*, variabel hasil *stemming* akan dilakukan proses *casefolding* sehingga semua kata dalam *dataset* berubah menjadi kata dengan huruf kecil. Hasil *casefolding* dapat dilihat pada gambar 18. dibawah ini :

```
Out[22]: ['ikan',
          'lele',
          'rupa',
          'salah',
          'satu',
          'makan',
          'favorit',
          'di',
          'indonesia',
```

Gambar 18. Hasil *casefolding*

4.2.3 Implementasi Proses Penghilangan *Stopword* dan Tanda Baca

Pada proses penghilangan *stopword*, dibutuhkan modul *Spacy*. Modul *Spacy* menyediakan beberapa *corpora teks*, salah satunya adalah *Stopwords Corpus.* Selain kata-kata umum, ada juga kelompok kata *stopword* yang memiliki posisi penting dalam morfologi dan tidak bisa berdiri sendiri. Stopword yang akan dihilangkan dapat dilihat gambar 19. dibawah ini :

```
Out[29]: ['baik',
          'sepantasnyalah',
          'haruslah',
          'dahulu',
          'boleh',
          'kalaupun',
          'bertanya',
          'misalnya',
          'begitupun',
          'pertama',
```

Gambar 19. *Stopword*

Jumlah keseluruhan stopword bahasa Indonesia ada 757 kata. Setelah menghilangkan *stopword* pada kalimat maka proses selanjutnya adalah mengilangkan tanda baca pada kalimat. Tanda baca yang akan dihilangkan dapat dilihat pada gambar 20. dibawah ini :

```
Out[31]: '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

Gambar 20. Tanda baca

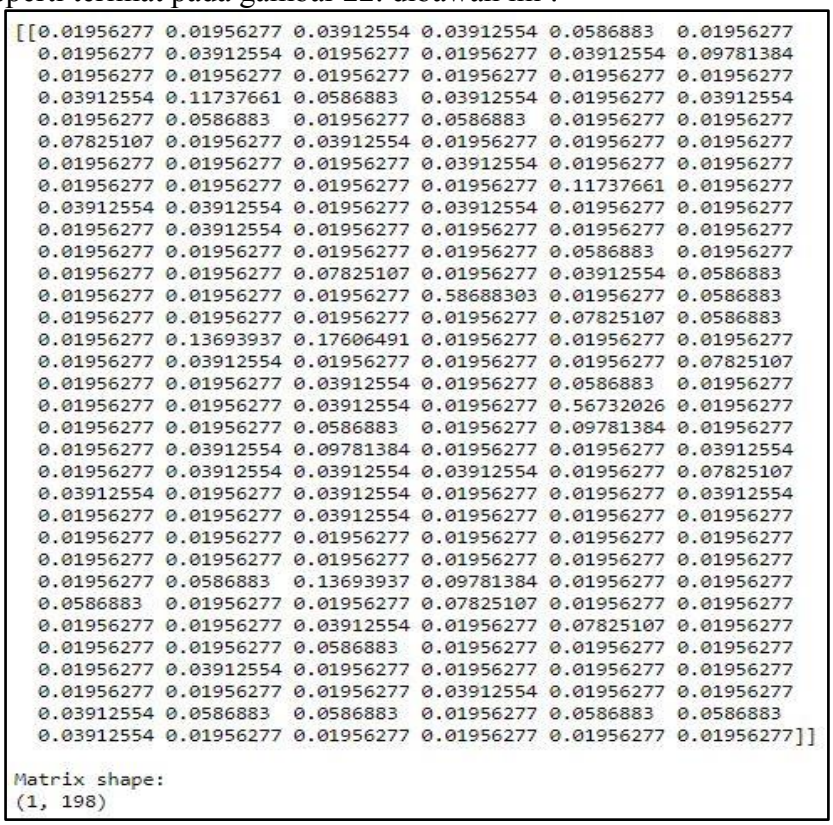
4.2.4 Implementasi Proses *Tokenisasi*

Pada proses tokenisasi adalah proses dimana mengubah sebuah kalimat menjadi *unigram* kata. Dalam proses ini dibutuhkan modul *spacy NLP*. Meski Python memiliki kemampuan untuk melakukan tugas-tugas *Natural Language Processing* dasar, namun tidak cukup powerful untuk melakukan tugas-tugas standar NLP, maka dari itu muncullah modul *spacy*. Modul ini menyediakan berbagai fungsi dan *wrapper*, serta *corpora* standar baik itu mentah atau *pre-processed*. Pada proses ini pula penulis mencoba untuk menggabungkan beberapa implementasi sebelumnya seperti *stemming*, *case folding*, *stopword removal* menjadi satu fungsi. Untuk proses tokenisasi adalah seperti terlihat pada gambar 21. dibawah ini :

Gambar 21. Hasil *tokenisasi* dan penggabungan algoritma *preprocessing*

4.2.5 Implementasi Proses Pembobotan *TF-IDF*

Pada proses pembobotan *TF-IDF* digunakan modul pustaka *scikit-learn* untuk mengekstraksi teks dengan menggunakan *TfidfVectorizer*. Hasil dari pembobotan TF-IDF berupa matriks. Untuk hasil proses pembobotan *TF-IDF* adalah seperti terlihat pada gambar 22. dibawah ini :



Gambar 22. Matriks pembobotan *TF-IDF*

Gambar 22. menunjukkan hasil *TF-IDF* yang telah dilakukan. Matriks *TF-IDF* yang dihasilkan berukuran 1 x 198 dengan hanya menggunakan satu artikel contoh. Matriks dari total keseluruhan data artikel adalah 250 x 5853, data website adalah 250 x 205 dan data status adalah 250 x 2. Setelah memanggil nilai label atau tagging dan data matriks *TF-IDF* selanjutnya dilakukan proses klasifikasi.

4.2.6 Implementasi Klasifikasi Berita *Hoax*

Pada proses ini penulis akan menggabungkan semua fitur *preprocessing text*, tokenisasi dan ekstrasi fitur ke dalam sebuah fungsi. Fungsi tersebut akan digunakan bersamaan dengan proses klasifikasi menggunakan *pipeline*. Fitur-fitur dan fungsi yang telah dibuat sebelumnya akan dimasukan ke dalam *pipeline* ini. *Pipeline* pertama berisi fitur hasil dari *preprocessing* artikel, *Pipeline* kedua berisi fitur hasil dari *preprocessing website* dan ketiga berisi fitur hasil dari *preprocessing* status dari usulan atribut penulis. Sebelum melakukan proses klasifikasi, ketiga *pipeline* tersebut akan di-*union* dengan menggunakan fungsi *union pipeline* pada *pipeline* gabungan dan hasil *union* akan digunakan untuk proses klasifikasi pada *pipeline* gabungan.

```
pipe = Pipeline([("cleaner", predictors()),
                 ('vectorizer', tfvectorizer),
                 ('to_dense', DenseTransformer())])
pipeZ1 = Pipeline([("cleaner", predictorsZ1()),
                 ('vectorizer', vectorizer),
                 ('to_dense', DenseTransformerZ1())])
pipeZ2 = Pipeline([("cleaner", predictorsZ2()),
                 ('vectorizer', vectorizer),
                 ('to_dense', DenseTransformerZ2())])
customPipeline = Pipeline([
    ('feat_union', FeatureUnion(transformer_list=[
        ('pl1', pipe),
        ('pl2', pipeZ1),
        ('pl3', pipeZ2)
     ]))
    ])
```

Gambar 23. Implementasi pembuatan *pipeline*

Pada gambar 23. terlihat bahwa pada *pipeline* terdapat beberapa langkah berurutan yang akan dieksekusi oleh sistem. Urutan pertama ialah *cleanner* yang akan berfungsi untuk *text cleaning* seperti *case folding,* yang kedua ialah ekstraksi fitur menggunakan TF-IDF atau tokenisasi, dalam proses ini terdapat proses *stemming*, *stopword removal*, *string punctuation*. Pada proses kedua ini akan menghasilkan matriks TF-IDF yang akan ditransformasikan ke dalam *dense matriks* pada urutan ketiga agar dapat dilakukan perhitungan *FeatureUnion*.

4.2.7 Implementasi Klasifikasi dengan Algoritma *Naive Bayes*

Implementasi proses klasifikasi berita dengan algoritma *Naïve Bayes* ialah dengan cara menambahkan fungsi classifier pada *pipeline* dengan fungsi algoritma

Naïve Bayes seperti yang dapat dilihat pada potongan *script* pada gambar 24. di bawah ini :

```
customPipeline = Pipeline([
    ('feat_union', FeatureUnion(transformer_list=[
        ('pl1', pipe),
        ('pl2', pipeZ1),
        ('pl3', pipeZ2)
     ])),
    ('classify', GaussianNB())
    ])
```

Gambar 24. Implementasi klasifikasi dengan *naïve bayes*

4.2.8 Implementasi Pembentukan Model Klasifikasi.

Implementasi klasifikasi berita adalah proses dimana fungsi – fungsi yang sebelumnya telah dibangun digunakan untuk melatih model menggunakan data berita. Pada proses ini semua proses *preprocessing*, seleksi fitur dan pembelajaran itu sendiri dijalankan. Hasil keluaran dari proses ini ialah suatu model yang dapat kita simpan dan kita gunakan untuk klasifikasi berita. *Script* untuk menjalankan semua proses atau fungsi tersebut dapat dilihat pada gambar 25 berikut ini :

```
X_train, X_test, y_train, y_test = train_test_split(X, ylabels, test_size=0.3, random_state=42)
trainProcess = customPipeline.fit(X_train, y_train)
trainProcess.score(X_test,y_test)
```

Gambar 25. Implementasi pembentukan model klasifikasi

4.3. Pengujian

4.3.1. Persiapan Data

Untuk melakukan prediksi berita hoax harus dipastikan bahwa berita yang disimpan merupakan suatu artikel berita dengan minimum 2 paragraf, bukan hanya sebuah *headline* berita saja. Untuk pengujian pada penelitian ini menggunakan data yang telah disimpan pada tahap pelatihan dan pengujian kemudian dibagi dokumen menjadi dua bagian, satu bagian untuk pelatihan dan satu bagian lainnya digunakan untuk pengujian.

4.3.2. *Preprocessing* dan Klasifikasi

Data yang disiapkan dilakukan *preprocessing* dengan tujuan untuk menghasilkan fitur prediksi berupa matriks frekuensi berita dengan akurasi terbaik yang akan digunakan untuk proses perhitungan pelatihan untuk prediksi. Untuk mendapatkan matriks frekuensi relatif berita dengan akurasi terbaik, sistem melakukan pemilihan fitur dengan menghasilkan data *unigram* kata dengan pembobotan *TF-IDF*.

4.3.3. Pengujian *Confusion Matrix*

Pengujian *confusion matrix* akan dilakukan uji coba dengan merubah perbandingan antara *data test* dan *data train*. Pada pengujian terdapat beberapa kondisi sebagai berikut: *True Positive* adalah kondisi berita *hoax* diklasifikasi

sebagai *hoax*, *True Negative* adalah kondisi berita bukan *hoax* diklasifikasikan sebagai bukan *hoax*, *False Positive* adalah kondisi berita *hoax* diklasifikasikan sebagai bukan *hoax*, *False Negative* adalah kondisi berita bukan *hoax* diklasifikasikan sebagai berita *hoax*.

Tabel 13. Pengujian dengan *test* 10 : *train* 90

Data Test	10%
Data Train	90%
True Positive (TP)	11
True Negative (TN)	6
False Positive (FP)	4
False Negative (FN)	4
Accuracy	68%

$$Accuracy = \frac{(TP + TN)}{(TP+TN+FP+FN)} * 100\% = \frac{(11 + 6)}{(11+6+4+4)} * 100\% = 68\% \qquad (4.1)$$

Pengujian *confusion matrix* menggunakan perbandingan 10:90 menghasilkan akurasi 68% dengan nilai *True Positive (TP)* adalah 11, *True Negative (TN)* adalah 6, *False Positive (FP)* adalah 4, *False Negative (FN)* adalah 4.

Tabel 14. Pengujian dengan *test* 20 : *train* 80

Data Test	20%
Data Train	80%
True Positif (TP)	25
True Negatif (TN)	11
False Positif (FP)	7
False Negatif (FN)	7
Accuracy	72%

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100\% = \frac{(25+11)}{(25+11+7+7)} * 100\% = 72\% \qquad (4.2)$$

Pengujian *confusion matrix* menggunakan perbandingan 20:80 menghasilkan akurasi 72% dengan nilai *True Positive (TP)* adalah 25, *True Negative (TN)* adalah 11, *False Positive (FP)* adalah 7, *False Negative (FN)* adalah 7.

Tabel 15. Pengujian dengan *test* 30 : *train* 70

Data Test	30%
Data Train	70%
True Positif (TP)	41
True Negatif (TN)	13
False Positif (FP)	15
False Negatif (FN)	6
Accuracy	72%

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100\% = \frac{(41+13)}{(41+13+15+6)} * 100\% = 72\% \quad (4.3)$$

Pengujian *confusion matrix* menggunakan perbandingan 30:70 menghasilkan akurasi 72% dengan nilai *True Positive (TP)* adalah 41, *True Negative (TN)* adalah 13, *False Positive (FP)* adalah 15, *False Negative (FN)* adalah 6.

Tabel 16. Pengujian dengan *test* 40 : *train* 60

Data Test	40%
Data Train	60%
True Positif (TP)	55
True Negatif (TN)	16
False Positif (FP)	22
False Negatif (FN)	7
Accuracy	71%

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100\% = \frac{(55+16)}{(55+16+22+7)} * 100\% = 71\% \quad (4.4)$$

Pengujian *confusion matrix* menggunakan perbandingan 40:60 menghasilkan akurasi 71% dengan nilai *True Positive (TP)* adalah 55, *True Negative (TN)* adalah 16, *False Positive (FP)* adalah 22, *False Negative (FN)* adalah 7.

Tabel 17. Pengujian dengan *test* 50 : *train* 50

Data Test	50%
Data Train	50%
True Positif (TP)	68
True Negatif (TN)	21
False Positif (FP)	27
False Negatif (FN)	9
Accuracy	71,2%

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100\% = \frac{(68+21)}{(68+21+27+9)} * 100\% = 71{,}2\% \quad (4.5)$$

Pengujian *confusion matrix* menggunakan perbandingan 50:50 menghasilkan akurasi 71,2% dengan nilai *True Positive (TP)* adalah 68, *True Negative (TN)* adalah 31, *False Positive (FP)* adalah 27, *False Negative (FN)* adalah 9.

Tabel 18. Perbandingan hasil uji coba

Test : Train	*Accuracy*	*True Positif (TP)*	*True Negatif (TN)*	*False Positif (FP)*	*False Negatif (FN)*
10% : 90%	68%	11	6	4	4
20% : 80%	72%	25	11	7	7
30% : 70%	72%	41	13	15	6
40% : 60%	71%	55	16	22	7
50% : 50%	71,2%	68	31	27	9

Dari hasi perbandingan uji coba pada Tabel 18. dapat disimpulkan bahwa perbandingan data test dan data train yang menghasilkan nilai terbaik adalah 20:80 dan 30:70 dengan tingkat akurasi 72%.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari penelitian dan pengujian yang dilakukan dapat dikesimpulkan bahwa akurasi terbaik dihasilkan dari *data train* 80%, *data test* 20% dan *data train* 70%, *data test* 30% dihasilkan sistem pada proses klasifikasi dengan penambahan atribut adalah 72%. Dan turunnya akurasi dikarenakan penurunan bobot klasifikasi dari *website*, setiap *website* penyebar *hoax* hanya mempublikasi satu topik dan satu berita dan kemudian domain *website* tersebut dibiarkan *expired* atau mati. Berita *hoax* mengikuti tren terkini, penyebar *hoax* hanya akan menggunakan satu *website* untuk mempublikasi sedikit artikel *hoax*.

5.2 Saran

Pada penelitian ini terdapat batasan-batasan masalah dan masih banyak kekurangan pada sistem yang telah dikembangkan pada penelitian ini. Beberapa hal yang dapat dikembangkan pada penelitian lebih lanjut yaitu:

1. Mendeteksi berita *hoax* dengan *data train* yang sedikit, dikarenakan *hoax* mengikuti tren berita terkini.
2. Mendeteksi *hoax* selain teks, banyak berita hoax tersebar dengan menggunakan gambar yang kemudian dijelaskan dengan artikel yang tidak sesuai fakta.
3. Menambahkan atribut waktu seperti setiap jam berapa artikel hoax akan dipublikasi atau jarak antara publikasi berita *valid* dan berita *hoax* butuh beberapa menit.

DAFTAR PUSTAKA

A. B. Adetunji1, J. P. Oguntoye, O. D. Fenwa1 and N. O. Akande. (2018). *Web Document Classification Using Naïve Bayes. Journal of Advances in Mathematics and Computer Science, 29(6), 1–11. https://doi.org/10.9734/jamcs/2018/34128*

APJII. (2017). Asosiasi Penyelenggara Jasa Internet Indonesia. https://www.apjii.or.id/content/read/39/342/Hasil-Survei-Penetrasi-dan-Perilaku-Pengguna-Internet-Indonesia-2017

APJII. (2020). siaran Pers: Pengguna Internet Indonesia Hampir Tembus 200 Juta di 2019 – Q2 2020. https://blog.apjii.or.id/index.php/2020/11/09/siaran-pers-pengguna-internet-indonesia-hampir-tembus-200-juta-di-2019-q2-2020/

Chen, Y. Y., Yong, S.-P., & Ishak, A. (2014). *Email Hoax Detection System Using Levenshtein Distance Method. Journal of Computers, 9(2). https://doi.org/10.4304/jcp.9.2.441-446*

E.Kusrini dan L. Taufiq. (2009). Algoritma Data Mining. Andi.

F.Rahutomo, I. Y. R. Pratiwi dan D. M. Ramadhani. (2019). *Eksperimen Naïve Bayes* Pada Deteksi Berita Hoax Berbahasa Indonesia. December. https://doi.org/10.33299/jpkop.23.1.1805

F.S.Radjatadoe. (2012). Analisis Leksikal (Analisis Leksikal/Analisis Linier/Pembacaan Sekilas (Scanner)). https://farmysetiawan.wordpress.com/2012/05/09/analisis-leksikal-analisis-leksikalanalisis-linierpembacaan-sekilas-scanner/

K.Kominfo. (2022). Pengertian Hoax dan Cara Menangkalnya. https://diskominfo.badungkab.go.id/artikel/42985-pengertian-hoax-dan-cara-menangkalnya

L.A. Waskito, K. M. Lhaksmana dan D. T. Murdiansyah. (2019). Analisis Sentimen Terhadap Pemilihan Presiden Indonesia 2019 Pada Media Sosial Twitter Menggunakan Metode *Naïve Bayes*. *Proceeding of Engineering, 6(2), 9753–9764.*

Mastel. (2017). Hasil Survey Mastel Tentang Wabah Hoax Nasional. In Mastel.

Prasetijo, A. B. Isnanto, R. R. Eridani, D. Soetrisno, Y. A.D. Arfan, M. Sofwan dan Aghus. (2018). *Hoax detection system on Indonesian news sites based on text classification using SVM and SGD. Proceedings - 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2017, 2018-Januari, 45–49. https://doi.org/10.1109/ICITACEE.2017.8257673*

Pratiwi, I. Y. R., Asmara, R. A., & Rahutomo, F. (2017). *Study of hoax news detection using naïve bayes classifier in Indonesian language. 2017 11th International Conference on Information & Communication Technology and System (ICTS), February, 73–78. https://doi.org/10.1109/ICTS.2017.8265649*

Sarkar, D. (2016). *Text Analytics with Python. https://doi.org/10.1007/978-1-4842-2388-8*

Thota, A. (2018). SMU Data *Science Review Fake News Detection* : *A Deep Learning Approach Fake News Detection : A Deep Learning Approach. 1(3)*.

Tim VIVA. (2018). Anindya Bakrie: Penyebar Hoax Terbanyak itu Media Sosial –

VIVA.

Wapna, Y. S., Aiprasad, P. S., Athsalya, B. V, & Handrakanth, S. C. (2019). *Fake News Detection using Naïve Bayes Classifier. 11(04), 214–217.*

Yunita dan Kominfo. (2017). Cara Mengatasi Berita "*Hoax*" di Dunia Maya. https://kominfo.go.id/content/detail/8949/ini-cara-mengatasi-berita-hoax-di-dunia-maya/0/sorotan_media.