SKRIPSI

IMPLEMENTASI ALGORITMA *LOGISTIC REGRESSION* UNTUK KLASIFIKASI UJARAN KEBENCIAN DAN BAHASA KASAR PADA TWITTER BAHASA INDONESIA

Disusun oleh :

NAMA : DADAN DAHMAN WAHIDI

NPM : 2018804303

JURUSAN : SISTEM INFORMASI

Untuk Memenuhi Sebagian Dari Syarat-syarat

Guna Mencapai Gelar Sarjana Komputer

STMIK INSAN PEMBANGUNAN
Jl. Raya Serang Km. 10 Bitung – Tangerang
Website : https:\\www.stmik.ipem.ac.id
Email : info@ipem.ac.id
Telp. (021) 59492836
Fax. (021) 59492837
Tahun Akademik 2021/2022

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INSAN PEMBANGUNAN
SARJANA KOMPUTER JURUSAN SISTEM INFORMASI
2022

TANDA PERSETUJUAN SKRIPSI

NAMA	**: DADAN DAHMAN WAHIDI**
NPM	**: 2018804303**
JURUSAN	**: SISTEM INFORMASI**
JUDUL SKRIPSI	**: IMPLEMENTASI ALGORITMA *LOGISTIC REGRESSION* UNTUK KLASIFIKASI UJARAN KEBENCIAN DAN BAHASA KASAR PADA TWITTER BAHASA INDONESIA**

SKRIPSI INI TELAH DIPERIKSA DAN DISETUJUI :
STMIK INSAN PEMBANGUNAN

Pembimbing Materi	Pembimbing Teknis
_____	_____
NIDN :	NIDN :

Ketua Jurusan Sistem Informasi

_____

NIDN :

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

INSAN PEMBANGUNAN

Jl. Raya Serang Km. 10 Bitung – Tangerang

Tahun Akademik 2021/2022

TANDA PERSETUJUAN SETELAH LULUS UJIAN SKRIPSI

NAMA	: DADAN DAHMAN WAHIDI
NPM	: 2018804303
JURUSAN	: SISTEM INFORMASI
JUDUL SKRIPSI	: IMPLEMENTASI ALGORITMA *LOGISTIC REGRESSION* UNTUK KLASIFIKASI UJARAN KEBENCIAN DAN BAHASA KASAR PADA TWITTER BAHASA INDONESIA

Tangerang,

Penguji I

NIDN :

Penguji II

NIDN :

Mengetahui,

Ketua STMIK Insan Pembangunan

Winanti, S.Kom, MM., M.Kom

NIDN : 0405057702

BERITA ACARA KOMPREHENSIF / SKRIPSI

Berdasarkan Syarat Keputusan Ketua STMIK Insan Pembangunan No……. Tanggal……., pada hari ini telah dilangsungkan Ujian Komprehensif / Skripsi program S-1 STMIK Insan Pembangunan Jurusan Sistem Informasi untuk Tahun Akademik 2021/2022.

1. Nama : DADAN DAHMAN WAHIDI
2. NPM : 2018804303
3. Jenjang Pendidikan : Strata Satu (S-1)
4. Jurusan : Sistem Informasi
5. Judul Skripsi : IMPLEMENTASI ALGORITMA *LOGISTIC REGRESSION* UNTUK KLASIFIKASI UJARAN KEBENCIAN DAN BAHASA KASAR PADA TWITTER BAHASA INDONESIA
6. Ruang / Tempat :
7. Kelulusan dengan nilai :
8. Keterangan :

PANITIA UJIAN

1. …….. Ketua
2. …….. Sekretaris
3. …….. Anggota
4. …….. Anggota

Mengetahui,
Ketua STMIK Insan Pembangunan

Winanti, S.Kom., MM,. M.Kom
NIDN : 0405057702

SURAT PERNYATAAN

Yang bertandatangan dibawah ini :

NAMA	: DADAN DAHMAN WAHIDI
NPM	: 2018804303
JURUSAN	: SISTEM INFORMASI
JUDUL SKRIPSI	: IMPLEMENTASI ALGORITMA *LOGISTIC REGRESSION* UNTUK KLASIFIKASI UJARAN KEBENCIAN DAN BAHASA KASAR PADA TWITTER BAHASA INDONESIA

Dengan ini menerangkan bahwa Skripsi dengan judul tersebut di atas adalah benar hasil karya tulis dan penelitian saya. Oleh karena itu saya bersedia untuk mempertanggungjawabkannya. Apabila dikemudian hari ternyata Skripsi tersebut bukan hasil karya tulis dan penelitian saya siap menerima sanksi dari kampus. Demikian surat pernyataan ini saya buat untuk persyaratan Sidang Skripsi. Terima kasih.

Tangerang, 28 Maret 2022

Yang membuat pernyataan

(Dadan Dahman Wahidi)

KATA PENGANTAR

Segala Puji Syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas limpahan karunia dan rahmat-Nya, maka dapat menyelesaikan penyusunan Skripsi ini. Salam hormat penulis sanjungkan dan haturkan untuk kedua orang tua yang tersayang, karena beliau adalah malaikat dunia yang selalu memberikan motivasi bagi penulis.

Skripsi imi merupakan salah satu syarat yang harus dipenuhi oleh penulis untuk memenuhi sebagian dari syarat-syarat guna mencapai gelar sarjana komputer. Untuk memenuhi persyaratan tersebut, maka penulis telah berusaha menyusun skripsi ini dengan judul "Implementasi Algoritma *Logistic Regression* Untuk Klasifikasi Ujaran Kebencian dan Bahasa Kasar Pada Twitter Bahasa Indonesia".

Penulis menyadari bahwa skripsi ini tidak akan terselesaikan tanpa bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak H. Soebari. Selaku Ketua Yayasan Pendidikan Insan Pembangunan.
2. Ibu Winanti, S.Kom., MM., M.Kom,. Selaku Ketua STMIK Insan Pembangunan
3. Ibu Dr. Dra. Francisca Sestri G., MM. Selaku Puket I Bidang Akademik di STMIK Insan Pembangunan.
4. Ibu Nurasiah, S.Kom., MMSI. Selaku Ketua Jurusan Sistem Informasi STMIK Insan Pembangunan.
5. …….. Selaku Dosen Pembimbing Materi dalam penyusunan skripsi ini.
6. …….. Selaku Dosen Pembimbing Teknis dalam penyusunan skripsi ini.
7. Seluruh Staff Pengajar, yang telah mendidik dan memberikan pengetahuan kepada penulis sehingga penulis banyak memperoleh ilmu pengetahuan untuk penyusunan skripsi ini.

8. Teman serta kerabat dekat khususnya mahasiswa/mahasiswi STMIK Insan Pembangunan

Segala kritik dan saran yang sifatnya membangun sangat penulis harapkan.

Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat dan kebaikan bagi banyak pihak. *Aamiin*

Tangerang, 28 Maret 2022

Penulis

(Dadan Dahman Wahidi)

IMPLEMENTASI ALGORITMA *LOGISTIC REGRESSION* UNTUK KLASIFIKASI UJARAN KEBENCIAN DAN BAHASA KASAR PADA TWITTER BAHASA INDONESIA

Abstrak

Ujaran kebencian (*hate speech*) dan bahasa kasar (*abusive language*) merupakan suatu tindakan negatif yang seringkali terjadi di lingkungan kita. Terlebih lagi dengan adanya teknologi yang semakin maju dan serba *online,* siapa saja bisa melakukan penyebaran ujaran kebencian maupun bahasa kasar melalui media sosial. Sering terjadi pertikaian antara masing-masing pihak yang berkepentingan, salah satu yang sering terjadi melalui media sosial *platform* twitter. Dengan adanya penelitian ini diharapkan dapat membantu kita semua untuk dapat membedakan antara ujaran kebencian dan bahasa kasar, serta lebih bijak lagi dalam ber-media sosial. Pada penelitian ini menggunakan algoritma *Logistic Regression* sebagai *classifier,* penelitian ini melakukan skenario dengan model *word embedding* untuk menemukan akurasi tertinggi yang mungkin dapat dicapai oleh pengklasifikasi.

Kata Kunci: ujaran kebencian*,* bahasa kasar*,* twitter*,* klasifikasi, *logistic regression.*

IMPLEMENTATION OF LOGISTIC REGRESSION ALGORITHM FOR CLASSIFICATION OF HATE SPEECH AND ABUSIVE LANGUAGE IN INDONESIAN TWITTER

Abstract

Hate speech and abusive language are negative actions that often occur in our environment. Moreover, with increasingly advanced technology and all online, anyone can spread hate speech or abusive language through social media. Conflicts often occur between each interested party, one of which often occurs through the social media platform twitter. With this research, it is hoped that it can help us all to be able to distinguish between hate speech and abusive language, and be wiser in using social media. In this study using the Logistic Regression algorithm as a classifier, this study carried out a scenario with a word embedding model to find the highest accuracy that could be achieved by the classifier.

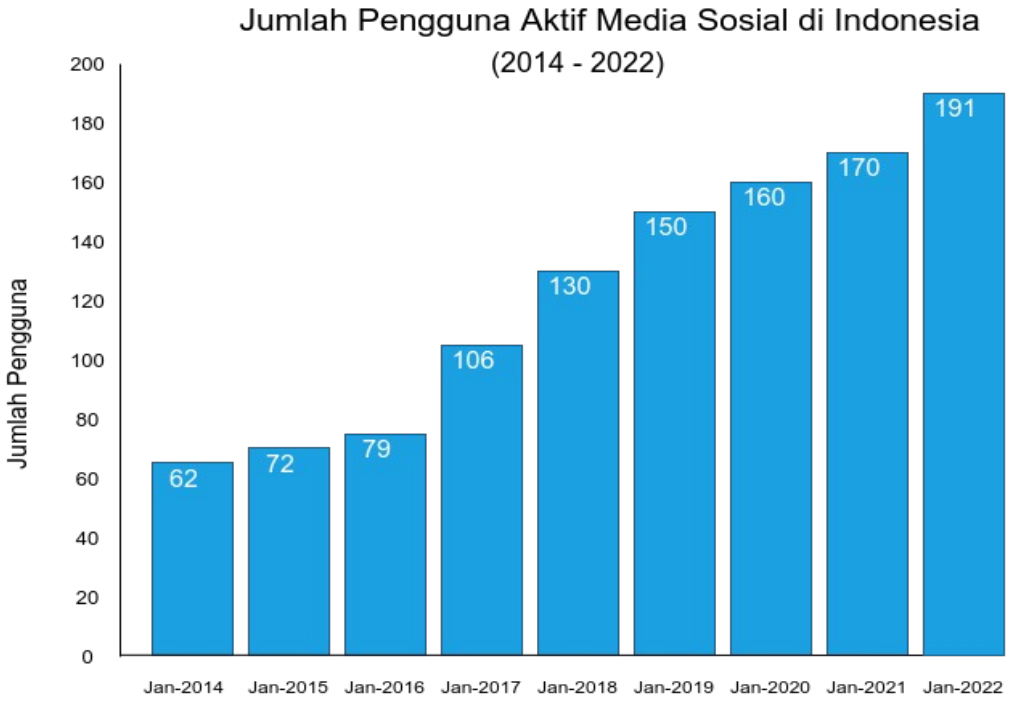
Keywords: hate speech, abusive language, twitter, classification, logistic regression.

DAFTAR ISI:

BAB I
PENDAHULUAN

1.1 Latar Belakang

Media sosial merupakan sebuah media daring yang digunakan satu sama lain yang para penggunanya bisa dengan mudah berpartisipasi, berkomunikasi, berbagi, dan menciptakan berbagai konten tanpa dibatasi oleh ruang dan waktu. Selain memiliki fungsi yang dapat memudahkan berbagai urusan media sosial juga tidak terlepas dari hal-hal negatif yang dapat mempengaruhi pola pikir dan juga pola hidup si pengguna media sosial. Berdasarkan laporan We Are Social, jumlah pengguna aktif media sosial di Indonesia sebanyak 191 juta orang pada Januari 2022. Jumlah itu telah meningkat 12,35% dibandingkan pada tahun sebelumnya yang sebanyak 170 juta orang. Melihat trennya, jumlah pengguna media sosial di Indonesia terus meningkat setiap tahunnya. Walau demikian, pertumbuhannya mengalami fluktuasi sejak 2014-2022. Kenaikan jumlah pengguna media sosial tertinggi mencapai 34,2% pada 2017. Hanya saja, kenaikan tersebut melambat hingga sebesar 6,3% pada tahun lalu. Angkanya baru meningkat lagi pada tahun ini. Adapun, Whatsapp menjadi media sosial yang paling banyak digunakan masyarakat Indonesia. Persentasenya tercatat mencapai 88,7%. Setelahnya ada Instagram dan Facebook dengan persentase masing-masing sebesar 84,8% dan 81,3%. Sementara, proporsi pengguna TikTok dan Telegram berturut-turut sebesar 63,1% dan 62,8%.



GAMBAR 1.1 Jumlah Pengguna Aktif Media Sosial di Indonesia

Ujaran kebencian (*hate speech*) merupakan perbuatan yang dilakukan oleh individu maupun kelompok dengan tujuan ingin menjatuhkan individu atau kelompok lainnya. Provokasi, fitnah, dan hinaan adalah bentuk dari ujaran kebencian. Ujaran kebencian dalam ruang lingkup sosial media sering terjadi dengan konteks atau aspek ras, warna kulit, jenis kelamin, agama, dan sebagainya. (Fauzi & Yuniarti, 2018)

Bahasa kasar (*abusive language*) sering diungkapkan karena kekesalan, emosi, kecewa, atas sebuah peristiwa yang terjadi dengan individu atau kelompok tertentu. Dalam sosial media bahasa kasar sering di implementasikan pada hal-hal yang tergolong ke dalam konteks SARA (suku, agama, ras, dan antar golongan). Kata-kata kasar dalam bahasa Indonesia biasanya diucapkan atau dituliskan untuk menyerang pihak tertentu, mengungkapkan kekesalan, kekecewaan, atau meluapkan emosi terhadap peristiwa tertentu. (Hidayatullah dkk., 2019)

Pada penelitian yang dilakukan oleh (A. T. Haryanto, 2018), terdapat informasi tentang pengguna aktif sosial media di Indonesia. Terdapat sekitar

130 juta penduduk Indonesia yang aktif dalam dunia maya atau sosial media. Twitter adalah *platform* sosial media yang cukup popular dengan fitur andalannya *tweet* dan *re-tweet*. Twitter banyak digunakan penduduk Indonesia sebagai tempat untuk berbagi cerita, informasi, dan pengalaman hidup. Penyebaran berita ataupun informasi melalui twitter dapat dibilang sangat cepat, karena fitur *re-tweet* mampu membuat pengguna lainnya ikut serta dalam menyebarkan berita atau kejadian ke pengguna lainnya.

Pada penelitian (Sudiantoro dkk., 2018) dengan judul analisis sentimen twitter menggunakan *text mining* dan algoritma *Naïve Bayes Classifier*. Tujuan dari penelitian tersebut untuk melakukan klasifikasi terhadap data *tweet* menjadi dua bagian yaitu sentimen positif dan negatif menggunakan data twitter bahasa indonesia terkait pendapat masyarakat terhadap pelaksanaan pilkada Jawa Barat. Data yang diproses sebanyak 300 data *tweet*. Kemudian dibagi untuk data latih dan data uji, data latih 200 data, dan data uji 100 data. Dari 100 data uji yang diklasifikasi, diperoleh 32 data yang bersentimen positif dan 68 data bersentimen negatif. Maka dapat ditarik sebuah kesimpulan bahwa berdasarkan 100 data uji yang telah dilakukan proses klasifikasi dengan hasil 68 data termasuk sentimen negatif, dan akurasi dari *naïve bayes classifier* sebesar 84%.

Berkaitan dengan judul penelitian yang akan diteliti mengenai algoritma *Logistic Regression*, terdapat beberapa referensi terkait diantaranya, pada penelitian (Wesley, 2019) yang berjudul "Implementasi *Machine Learning* pada Sistem Pendeteksi Situs yang Bermuatan Konten Negatif". Dengan menerapkan 5 model utama yait*u Naïve Bayes Classifier, Support Vector Machine, Logistic Regression, Decision Tree,* dan *K-Nearest Neighbor.* Penelitian tersebut menggunakan *Confusion Matrix* untuk proses evaluasi hasil setiap model. Dari hasil training dataset tersebut terlihat bahwa model *Support Vector Machine* memiliki tingkat akurasi paling tinggi dari semua model utama yang ada. Dengan akurasi lebih dari 95 % untuk setiap model menunjukkan model SVM sebagai model yang paling baik untuk penelitian tersebut. Dari

ketiga model SVM di atas, model dengan akurasi tertinggi dipegang oleh model *LinearSVC* dengan akurasi 95.3886%, sedangkan model SVM dengan akurasi terendah adalah model *One Vs Rest* 95.2045%.

Model utama tertinggi kedua adalah *Logistic Regression* yaitu dengan 95.1349%. Model utama tertinggi ketiga adalah model *Naïve Bayes* dengan nilai akurasi di atas 90% dari setiap model. Dari kedua model *Naïve Bayes* tersebut, model *Multinomial Naïve Bayes* merupakan model dengan akurasi tertinggi dengan nilai akurasi mencapai 93.7527%. Sedangkan model *Naïve Bayes* tertinggi kedua adalah model *Gaussian Naïve Bayes* dengan skor akurasi 90.9883%. Sedangkan model dengan akurasi terendah adalah model *K-Nearest Neighbor* dan *Decison Tree* dengan masing-masing akurasi 68.2536% dan 54.2969%.

Berdasarkan masalah yang terdapat pada latar belakang, maka penulis akan melakukan analisa terhadap metode *Logistic Regression* dalam mengklasifikasi *multilabel* ujaran kebencian dan bahasa kasar pada twitter bahasa Indonesia.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas dalam penelitian ini yaitu bagaimana cara mengklasifikasikan *multilabel* ujaran kebencian dan bahasa kasar pada twitter bahasa Indonesia dengan menggunakan algoritma *Logistic Regression* sebagai *classifier*, serta melihat performa *Logistic Regression* dalam mengklasifikasi *multilabel.*

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini, diantaranya:

a. *Dataset* yang diproses sebanyak 13169 tweet (Ibrohim & Budi, 2019).
b. Label yang diproses yaitu *hate speech*, *abusive*, dan *neutral*.
c. Algoritma klasifikasi yang digunakan adalah *Logistic Regression.*

d. Metode *word embedding* yang digunakan adalah TF-IDF Vectorizer.
e. *Output* yang dihasilkan dari penelitian ini adalah klasifikasi *multilabel* ujaran kebencian dan bahasa kasar pada twitter Bahasa Indonesia.

1.4 Tujuan Penelitian

Beberapa tujuan yang ingin dicapai dalam penelitian ini, diantaranya:

a. Implementasi algoritma *Logistic Regression* untuk mengklasifikasi *multilabel* ujaran kebencian dan bahasa kasar pada twitter Bahasa Indonesia.
b. Menghitung akurasi dari algoritma *Logistic Regression* dalam mengklasifikasi *multilabel* ujaran kebencian dan bahasa kasar pada twitter Bahasa Indonesia.
c. Melakukan proses *hyperparameter tuning* untuk menemukan *parameter* terbaik agar mendapatkan model yang mampu memprediksi dengan lebih akurat.
d. Menggunakan teknik *MultiOutput Classifier* dalam menangani klasifikasi *multilabel*.

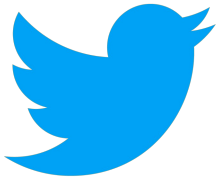
1.5 Manfaat Penelitian

Manfaat pada penelitian ini, diantaranya:

a. Penelitian ini diharapkan mampu memberikan kemudahan bagi masyarakat dan pemerintah dalam pemberantasan *tweet* bermuatan kalimat ujaran kebencian dan bahasa kasar yang menjamur menggunakan sistem klasifikasi ini.
b. Sebagai bahan referensi bagi peneliti lain yang ingin membahas topik yang terkait dengan penelitian ini.

BAB II
TINJAUAN PUSTAKA

2.1 Twitter



GAMBAR 2.1 Logo Media Sosial Twitter

Twitter adalah layanan jejaring sosial daring yang memungkinkan penggunanya untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter, akan tetapi pada tanggal 07 November 2017 bertambah hingga 280 karakter yang dikenal dengan sebutan kicauan (*tweet*). Twitter didirikan pada bulan Maret 2006 oleh Jack Dorsey. Twitter dimiliki dan dioperasikan oleh Twitter, Inc., yang berbasis di San Francisco, dengan kantor dan peladen tambahan terdapat di New York City, Boston, dan San Antonio. Di Twitter, pengguna tak terdaftar hanya bisa membaca *tweet*, sedangkan pengguna terdaftar bisa menulis *tweet* melalui antarmuka situs web, pesan singkat, atau melalui berbagai aplikasi untuk perangkat seluler.

Tingginya popularitas Twitter menyebabkan layanan ini telah dimanfaatkan untuk berbagai keperluan dalam berbagai aspek, misalnya sebagai sarana protes, kampanye politik, sarana pembelajaran, dan sebagai media komunikasi darurat.

2.2 Ujaran Kebencian (*Hate Speech*)

Studi tentang ujaran kebencian dalam beberapa tahun terakhir telah menarik perhatian sejumlah besar sarjana dari berbagai bidang pengetahuan (ahli bahasa, sosiolog, filsuf, sejarawan, psikolog, antropolog, pengacara dan

ilmuwan politik, dan lain-lain (Neshkovska & Trajkova, 2018). Hal yang menjadikan para cendekiawan ingin menangani masalah ini dengan pertimbangan kemungkinan bahwa di masa-masa sulit yang kita hadapi ini, masyarakat di seluruh dunia sedang terpolarisasi secara mendalam dengan begitu banyak perbedaan seperti agama, politik, etnis, dan lain-lain, dan berada dalam keadaan yang terus berubah. Pada akhirnya membuat mereka sangat rentan terhadap kebencian yang dampak buruknya kadang-kadang bisa diluar nalar manusia.

Pertanyaan inti yang muncul dalam membahas ujaran kebencian adalah apa itu ujaran kebencian. Tinjauan literatur yang relevan mengungkapkan bahwa ujaran kebencian adalah fenomena yang kompleks dan sangat diperdebatkan, dan belum ada satu pun definisi yang dapat diterima secara bulat tentang apa sebenarnya ujaran kebencian itu.

Definisi lain yang sedikit lebih luas tentang ujaran kebencian adalah tindakan komunikasi yang dilakukan oleh suatu individu atau kelompok dalam bentuk provokasi, hasutan, ataupun hinaan kepada individu atau kelompok yang lain dalam berbagai aspek seperti ras, warna kulit, jenis kelamin, agama, orientasi seksual, dan lain-lain. (Fauzi & Yuniarti, 2018)

Salah satu definisi yang paling sering dikutip tentang pidato kebencian adalah yang diusulkan oleh Dewan Eropa. Menurut Dewan Eropa “semua bentuk ekspresi yang menyebar, menghasut, mempromosikan atau membenarkan kebencian rasial, *xenophobia*, anti-Semitisme atau bentuk-bentuk kebencian lain berdasarkan intoleransi, termasuk intoleransi yang diungkapkan oleh nasionalisme dan etnosentrisme yang agresif, diskriminasi dan permusuhan terhadap kaum minoritas, migran dan orang-orang yang berasal dari imigran ”berada di bawah payung istilah *hate speech* (Gagliardone et al., 2014).

(Gagliardone et al., 2014) lebih lanjut menyatakan bahwa ujaran kebencian mengungkapkan “sikap diskriminatif, mengintimidasi, tidak setuju, antagonis, dan atau berprasangka terhadap karakteristik tersebut, yang meliputi

gender, ras, agama, etnis, warna kulit, asal kebangsaan, cacat atau orientasi seksual" dan bahwa pidato kebencian adalah dimaksudkan "untuk melukai, tidak memanusiakan, melecehkan, mengintimidasi, merendahkan dan menjadi korban kelompok sasaran, dan untuk menimbulkan ketidakpekaan dan kebrutalan terhadap mereka". Ujaran kebencian (*hate speech*) dapat dilakukan melalui berbagai cara seperti orasi kegiatan kampanye, spanduk atau *banner*, media sosial, penyampaian pendapat dimuka umum (demonstrasi), ceramah keagamaan, media masa cetak maupun elektronik, dan sebagainya.

Masalah pelanggaran atau kejahatan mencemarkan nama baik orang lain, memfitnah, menista dan perbuatan tidak menyenangkan merupakan suatu perbuatan yang melanggar hukum karena meresahkan dan melanggar hak asasi orang lain (Ronny Wuisan, 2018). Perbuatan tersebut tidak hanya dapat dilakukan secara langsung dengan kata-kata di muka umum tetapi juga akhir-akhir ini sering dilakukan di dunia maya atau media sosial, karena di dunia maya masyarakat merasakan kebebasan dalam hal berpendapat maupun mengkritik seseorang yang dianggap tidak akan melanggar hukum dan aman karena tidak berkontak fisik langsung dengan orang lain. Salah satu contoh kasus tentang pelanggaran dalam media sosial yaitu Terdakwa Dhani Ahmad Prasetyo alias Ahmad Dhani. (Jurnal Krisna Law Volume 3, Nomor 2, 2021, 1-13)

Putusan Nomor 370/Pid.Sus/2018/PN.Jkt.Sel merupakan kasus tindak pidana khusus, yakni tindak pidana ujaran kebencian (*hate speech*) yang diperiksa, diadili dan diputus di Pengadilan Negeri Jakarta Selatan yang diucapkan dalam sidang terbuka untuk umum pada hari Senin, tanggal 28 Januari 2019 oleh H. Ratmoho, S.H, M.H. selaku Hakim Ketua, Akhmad Rosidin, S.H., M.H. dan Haruno Patriadi, S.H., M.H. masing-masing selaku Hakim Anggota, dengan identitas terdakwa sebagai berikut:

Nama Lengkap : Dhani Ahmad Prasetyo alias Ahmad Dhani
Tempat lahir : Jakarta
Umur/tgl lahir : 45 tahun/26 Mei 1972

Jenis kelamin : Laki-laki

Kebangsaan : Indonesia

Agama : Islam

Tempat tinggal : Jalan Pinang Emas VII D.4 No.7 RT.008/003, Kelurahan Pondok Pinang, Kecamatan Kebayoran Lama, Jakarta Selatan.

Pada awal mulanya, Terdakwa Dhani Ahmad Prasetyo alias Ahmad Dhani dari tahun 2010 sampai dengan tahun 2014 menggunakan dan mengoperasikan sendiri akun Twitter miliknya yang bernama @ahmaddhaniprast dengan menggunakan komputer PC dirumahnya. Pada tahun 2014 sampai dengan tahun 2017, Dhani Ahmad Prasetyo alias Ahmad Dhani menggunakan HP Iphone 6 dengan nomor HP pribadinya yang terdakwa gunakan khusus untuk media sosial, WhatsApp untuk mengirimkan kalimat kepada Suryopratomo Bimo AT alias Bimo yang kemudian oleh Suryopratomo Bimo AT alias Bimo diunggah ke akun Twitter miliknya yaitu @ahmaddhaniprast.

Peran Suryopratomo Bimo AT alias Bimo ialah bekerja sebagai admin yang tugasnya ialah untuk mengunggah tulisan-tulisan yang dibuat dan dikirimkan oleh Dhani Ahmad Prasetyo alias Ahmad Dhani melalui WhatsApp dari handphone Terdakwa ke nomor handphone Suryopratomo Bimo AT alias Bimo.

Pada tanggal 7 Februari 2017, Dhani Ahmad Prasetyo alias Ahmad Dhani membuat dan mengirimkan tulisan melalui WhatsApp kepada Suryopratomo Bimo AT alias Bimo, kemudian Suryopratomo Bimo AT alias Bimo bertempat di Gg. Edy IV No. 3 Rt. 005/006, Kel. Guntur, Kec. Setiabudi, Jakarta Selatan, menyalin persis seperti apa yang dikirim oleh terdakwa dan mengunggah ke akun Twitter terdakwa yang bernama @ahmaddhaniprast. Bunyi tulisan ialah:

"Yg menistakan Agama si Ahok… Yang diadili KH Ma'ruf Amin...ADP"

(https://Twitter.com/AHMADDHANIPRAST/status/828773795238326273?s=08).

Pada tanggal 6 Maret 2017 Terdakwa juga mengirimkan tulisan melalui Whatsapp kepada saksi Suryopratomo Bimo AT alias Bimo, kemudian saksi Suryopratomo Bimo AT alias Bimo bertempat Gg. Edy IV No. 3 Rt. 005/006, Kel. Guntur, Kec. Setiabudi, Jakarta Selatan, mengunggah tulisan dengan bunyi kalimatnya ialah:

"Siapa saja yg dukung Penista Agama ialah Bajingan yg perlu di ludahi mukanya–ADP"

(https://Twitter.com/AHMADDHANIPRAST/status/838660282222178304?s=08).

Pada tanggal 7 Maret 2017 Terdakwa juga mengirimkan tulisan melalui WhatsApp kepada saksi Suryopratomo Bimo AT alias Bimo, kemudian saksi Suryopratomo Bimo AT alias Bimo mengunggah tulisan dengan bunyi kalimatnya ialah:

"Sila Pertama KETUHANAN YME, PENISTA Agama jadi Gubernur...kalianWARAS???–ADP"

*(*https://Twitter.com/AHMADDHANIPRAST/status/838977634436460544?s=0*)*.

Pada hari Rabu tanggal 8 Maret 2017 sekitar pukul 17.00 WIB di Cilandak Town Square, Jakarta Selatan, saksi Jack Boyd Lapian, Danick Danoko, M. Togar Binda P. Harahap, Retno Hendriastuti, yang tergabung dalam BTP (Bersih Transparan Profesional) *Network* yang merupakan organisasi relawan pendukung Ir. Basuki Tjahaja Purnama, M.M. alias Ahok - Drs. H. Djarot Syaiful Hidayat dalam Pilkada DKI Jakarta 2017, merasa keberatan atas isi Twitter terdakwa diatas, yang dapat menimbulkan kebencian dan perpecahan di masyarakat.

Postingan-postingan terdakwa yang diunggah oleh admin Suryopratomo Bimo AT alias Bimo pada akun Twitter terdakwa yang bernama @ahmaddhaniprast tersebut dapat menimbulkan rasa kebencian atau

permusuhan individu dan/atau kelompok masyarakat tertentu berdasarkan atas suku, agama, ras, dan antar golongan (SARA), karena postingan tersebut disebarkan (di-*share*) yang bisa dibaca oleh orang-orang yang melihat Twitter terdakwa dan mendapat tanggapan tidak baik dari orang-orang yang membaca akun Twitter terdakwa yang bernama @ahmaddhaniprast.

Terdakwa dikenakan Pasal 45A ayat (2) jo. Pasal 28 ayat (2) Undang-Undang Nomor 19 Tahun 2016 Tentang Perubahan Undang-Undang Nomor 11 Tahun 2008 Tentang Informasi dan Transaksi Elektronik jo. Pasal 55 ayat (1) ke-1 Kitab Undang-Undang Hukum Pidana (KUHP).

2.3 Bahasa Kasar (*Abusive Language*)

Bahasa kasar adalah ekspresi yang berisi kata atau frasa kasar/kotor, baik lisan maupun tulisan (Tuarob & Mitrpanont, 2017). Menurut (Tuarob & Mitrpanont, 2017), penyebab digunakannya kata-kata kasar yang tidak terkontrol di sosial media adalah karena tidak adanya alat yang efektif untuk menyaring bahasa kasar di media sosial, kurangnya empati diantara warga negara, dan kurangnya bimbingan orang tua. Bahasa yang kasar di media sosial perlu disaring sehingga tidak ada anak-anak dan remaja yang belajar bahasa kasar dari media sosial yang mereka gunakan (Chen et al., 2012).

Mendeteksi bahasa kasar di media sosial adalah masalah yang sulit untuk dipecahkan (Nobata et al., 2016) mengatakan bahwa mendeteksi suatu bahasa kasar di media sosial tidak bisa hanya menggunakan pencocokan kata. Karena banyak dari netizen biasanya menggunakan ejaan dan tata bahasa kasar yang sangat informal. Terutama dalam teks pendek, mengklasifikasikan teks pendek untuk mendeteksi bahasa kasar lebih sulit untuk diselesaikan.

Misalnya dalam data twitter, ada banyak netizen yang memposting *tweet* menggunakan singkatan, karena terbatasnya jumlah kata yang diizinkan oleh twitter dalam sebuah postingan (Hanafiah et al., 2017) mengatakan bahwa beberapa kata non-formal yang sering digunakan oleh orang indonesia adalah:

kata-kata yang menunjukkan perasaan, pengulangan karakter untuk menekankan makna, menggunakan bahasa gaul, dan mengubah huruf vokal menjadi angka.

2.4 Bahasa Pemrograman Python

GAMBAR 2.2 Logo Pemrograman Python

Python adalah bahasa pemrograman multifungsi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991. Guido Van Rossum menciptakan Python untuk menjadi *interpreter* yang memiliki kemampuan penanganan kesalahan (*exception handling)* dan mengutamakan sintaksis yang mudah dibaca serta dimengerti (*readability*).

Python dirancang untuk memberikan kemudahan yang sangat luar biasa kepada *programmer* baik dari segi efisiensi waktu, maupun kemudahan dalam pengembangan program dan dalam hal kompatibilitas dengan sistem (Qutsiah, Sophan dan Hendrawan, 2016). Beberapa fitur yang terdapat dalam Bahasa Pemrograman Python antara lain memiliki *library* yang luas, dalam distribusi Python telah disediakan modul-modul siap pakai untuk berbagai keperluan.

Python menggunakan indentasi untuk mengelompokkan blok kode, berbeda dengan beberapa bahasa lain yang menggunakan simbol tertentu, misalnya kurung kurawal, atau sintaksis *begin-end*. Sehingga secara visual pun blok kode Python didesain untuk mudah dipahami. Salah satu yang paling dikenal adalah penggunaan titik koma atau *semicolon* (;) tidak wajib di Python dan penggunaan *semicolon* cenderung dianggap bukan cara khas Python (*non-*

pythonic way), meskipun ia tetap dapat digunakan, misalnya untuk memisahkan dua *statement* dalam baris yang sama.

Python juga memilih untuk mengadopsi *dynamic typing* secara opsional, yakni variabel yang dibuat tidak akan diketahui tipenya hingga ia dipanggil pertama kali atau dieksekusi, tidak perlu deklarasi variabel (meskipun dimungkinkan), dan memungkinkan tipe data berubah dalam proses eksekusi program.

Python terus berkembang dalam penggunaannya, sehingga fitur-fitur baru dibutuhkan untuk dikembangkan. Versi 2.0 dirilis Oktober 2000 dengan beberapa pengembangan fitur termasuk *Garbage Collector* dan *Memory Management* yang juga menjadi fitur pada beberapa bahasa pemrograman modern lainnya, di antaranya Java dan C#.

Python 3.0 adalah versi perubahan mayor yang dirilis pada Desember 2008, yang didesain sebagai versi yang tidak *backward-compatible* dengan versi-versi sebelumnya. Beberapa sintaksis yang sebelumnya berjalan di versi 2.x, kini tidak lagi berjalan. Semua hal ini didasarkan pada keinginan bahasa Python yang kembali ke "inti", yakni *readable, consistent & explicit.* Contohnya, fungsi *print* yang sebelumnya adalah *statement* di python 2.x, menjadi *function* di python 3.x.

Saat ini, Python dikelola oleh lembaga non-komersial *Python Software Foundation* (PSF). Namun sebelumnya, GvR dijuluki sebagai *Benevolent Dictator for Life* (BDFL) karena hampir semua keputusan pengembangan Python diambil oleh GvR, berbeda dengan bahasa lain yang misalnya menggunakan voting dan semacamnya. Pasca tahun 2000, dibentuklah beberapa sistem yang memungkinkan Python menjadi lebih substain, misalnya *Python Enhancement Proposals* (PEP) untuk pengembangan Python dan tentunya *Python Software Foundation* (PSF).

Jika PSF menjadi lembaga yang mengelola dan mengadvokasi Python, PEP menjadi panduan dalam pengembangan Python. Beberapa PEP memuat misalnya bagaimana sintaksis dan bagaimana Bahasa Python akan berevolusi,

bagaimana modul akan dinyatakan usang (*deprecated*), dan sebagainya. Setelah kurang lebih 30 tahun dalam pengembangan Python, GvR memutuskan untuk tidak lagi menjabat BDFL pada 12 Juli 2018. Salah satu patokan dalam pengembangan Python adalah PEP 20 yang berjudul Zen of Python. **(**Zen of Python, https://www.python.org/dev/peps/pep-0020/**)**

Python dapat berjalan pada berbagai sistem operasi yang tersedia. Beberapa pemanfaatan bahasa Python di antaranya:

a. *Web development* (*server-side*),
b. *Software development*,
c. *Mathematics & data science*,
d. *Machine learning*,
e. *System scripting*,
f. *Internet of Things (IoT) development*.

Python menyediakan *library* yang meliputi *regular expressions, documentation generation, unit testing, threading, databases, web browsers,* koneksi ke berbagai protokol, *cryptography,* GUI *(graphical user interfaces),* dan lain-lain. (Python Package Index, https://pypi.org/). Sejumlah *library* utama yang digunakan pada penelitian ini diantaranya:

a. Pandas, merupakan *library* di Python yang berlisensi BSD dan *open source* yang menyediakan struktur data dan analisis data yang mudah digunakan. Pandas biasa digunakan untuk membuat tabel, mengubah dimensi data, dan lain sebagainya. Struktur data dasar pada Pandas dinamakan *dataframe*, yang memudahkan kita untuk membaca sebuah *file* dengan banyak jenis format seperti *file* .json, .txt, .csv, dan .tsv. Fitur ini akan menjadikannya *table* dan juga dapat mengolah suatu data dengan menggunakan operasi seperti *join*, *distinct*, *group by*, *agregation*, dan teknik lainnya yang terdapat pada SQL.
b. Numpy, merupakan salah satu *library* pada Python yang berfungsi melakukan proses komputasi numerik. *Array* merupakan kumpulan variabel yang memiliki tipe data yang sama. Numpy menyimpan datanya

dalam bentuk *array*. Bentuk dari numpy *array* adalah multidimensional yang mana dapat berupa 1-dimensi maupun 2-dimensi. *Array* 1-dimensi adalah sekumpulan data yang berisikan nama variabel dan tipe data yang sama yang dapat diakses menggunakan 1 buah *index* saja. Sedangkan *array* 2-dimensi adalah sekumpulan data yang berisikan nama dan tipe data yang sama dimana elemennya dapat diakses menggunakan 2 buah *index* yaitu *index* kolom dan *index* baris.

c. Matplotlib, merupakan *library* visualisasi data *multiplatform* yang dibangun di atas *array* NumPy. Matplotlib disusun oleh John Hunter pada tahun 2002. Matplotlib dirancang agar dapat digunakan seperti MATLAB, dengan kemampuan untuk digunakan dalam Python dengan gratis dan *open-source*. Matplotlib juga dapat digunakan untuk memvisualisasikan data secara 2D maupun 3D di dalam Python dan menghasilkan gambar berkualitas dalam berbagai format.
d. Scikit-learn, menyediakan berbagai pilihan algoritma pembelajaran yang diawasi (*supervised*) dan tidak diawasi (*unsupervised*). Scikit-learn dibuat dengan pola pikir rekayasa perangkat lunak. Desain API intinya berkisar pada kemudahan digunakan, namun kuat, dan tetap mempertahankan fleksibilitas untuk upaya penelitian. Ketangguhan ini membuatnya sempurna untuk digunakan dalam berbagai proyek *end-to-end machine learnin*g apa pun, mulai dari fase penelitian hingga penerapan produksi.
e. NLTK, *Natural Language Toolkit* adalah sebuah *platform* berbasis Python yang dikembangkan untuk memproses data teks. NLTK dilengkapi dengan lebih dari 50 *corpora* dan *lexical resources* seperti Wordnet. Selain itu NLTK juga menyediakan modul untuk *text preprocessing* mulai dari *tokenizing*, *stemming*, *tagging*, dan *lemmatization*.

Implementasi python pada penelitian ini dilakukan pada proses pengolahan data seperti proses *Exploratory Data Analysis, Data Preprocessing, Language Model Training*, *Feature Selection*, klasifikasi *Logistic Regression*, dan *Metrics Evaluation*.

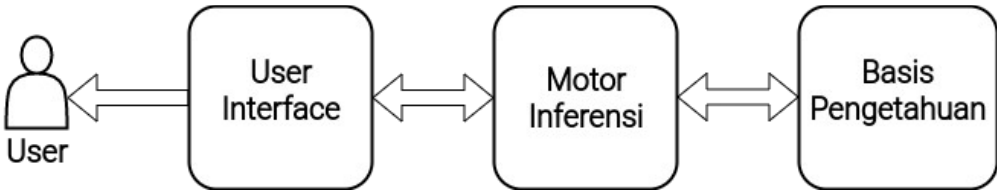
2.5 *Artificial Intelligence*

Artificial Intelligence (AI) atau yang biasa disebut kecerdasan buatan didefinisikan sebagai kecerdasan yang ditunjukkan oleh sebuah mesin atau *software*. Lebih spesifik lagi, menurut (Kusumadewi, 2003) menyatakan kecerdasan buatan merupakan salah satu bagian dalam ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar (*expert system*), permainan komputer (*games*), logika fuzzy, jaringan saraf tiruan (*artificial neural network*), robotika (*robotics*), pengolahan bahasa alami (*natural language processing*), pengenalan pola (*pattern recognition*), dan pengenalan suara (*speech recognition*) (Simarmata, 2006). Lingkup utama kecerdasan buatan diantaranya:

a. Sistem pakar (*expert system*): komputer sebagai sarana untuk menyimpan pengetahuan para pakar sehingga komputer memiliki keahlian menyelesaikan permasalahan dengan meniru keahlian yang dimiliki pakar.
b. Pengolahan bahasa alami (*natural language processing*): *user* dapat berkomunikasi dengan komputer menggunakan bahasa sehari-hari, misal bahasa inggris, bahasa indonesia, bahasa jawa, dll.
c. Pengenalan ucapan (*speech recognition*): manusia dapat berkomunikasi dengan komputer menggunakan suara.
d. Robotika dan sistem sensor.
e. *Computer vision*: menginterpretasikan gambar atau objek-objek tampak melalui komputer.
f. *Intelligent computer-aided instruction*: komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
g. *Game playing*.

Kecerdasan buatan ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer. Pengaplikasian kecerdasan buatan terdiri dari 2 bagian utama yang sangat dibutuhkan, yaitu (Kusumadewi, 2003):

a. Basis Pengetahuan (*Knowledge Base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
b. Motor Inferensi (*Inference Engine*) yaitu kemampuan untuk menarik kesimpulan berdasarkan pengalaman.



GAMBAR 2.3 Konsep Kecerdasan Buatan

Menurut Rusell dan Norvig (2010:2) terdapat empat macam pendekatan dalam AI, yaitu:

a. *Thinking Humanly*, yaitu sistem yang menangkap pemikiran psikologis, misalnya melalui eksperimen.
b. *Acting Humanly*, yaitu sistem dengan pendekatan menirukan tingkah laku manusia.
c. *Thinking Rationally*, yaitu sistem dengan penalaran komputasi.
d. *Acting Rationally*, yaitu sistem yang bertindak untuk mencapai hasil terbaik atau ketika terdapat ketidakpastian, mengeluarkan hasil terbaik yang diharapkan.

TABEL 2.1 Kecerdasan Buatan (Muhammad Dahria, 2008)

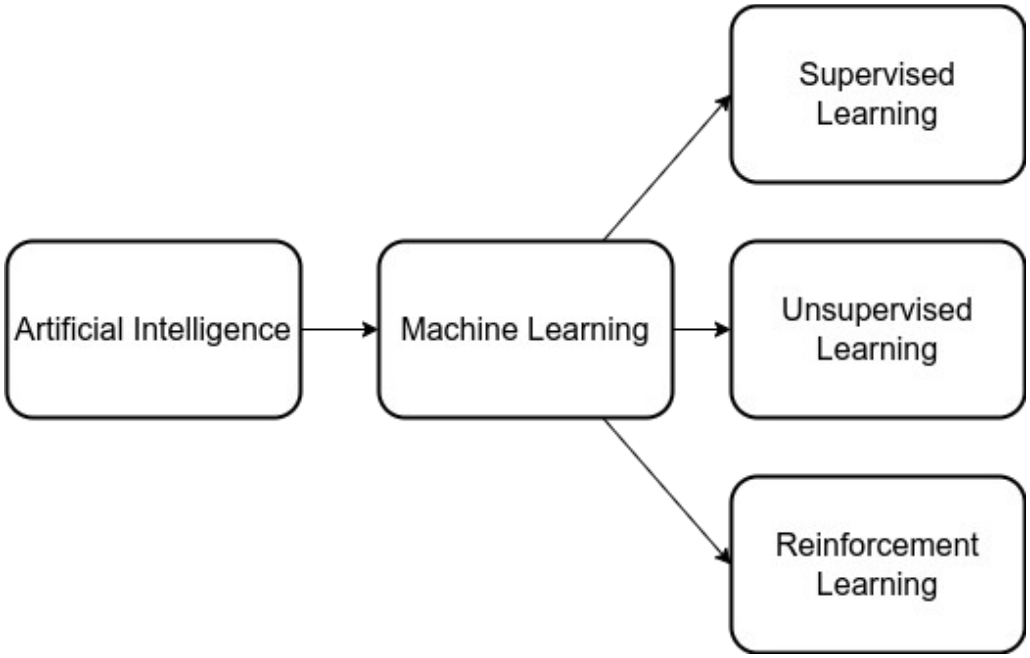
Dimensi	Kecerdasan Buatan	Pemrograman Konvensional
Pemrosesan	Pengetahuan diperoleh dari mekanisme pemrosesan	Digabung dalam satu program sekuensial

Eksekusi	Eksekusi dilakukan secara heuristik dan logis	Secara Algoritma
Sifat Input	Bisa tidak lengkap	Harus lengkap
Manipulasi	Efektif pada basis pengetahuan yang besar	Efektif pada database yang besar
Keterangan	Disediakan	Biasanya tidak disediakan
Fokus	Pengetahuan	Data dan informasi
Struktur	Kontrol dipisahkan dari pengetahuan	Kontrol terintegrasi dengan informasi
Sifat Output	Kualitatif	Kuantitatif
Perubahan	Perubahan pada kaidah dapat dilakukan dengan kaidah yang sedikit	Pada program merepotkan
Kemampuan Menalar	Ya	Tidak

2.6 *Machine Learning*

Seiring berlalunya waktu, mesin pintar atau cerdas perlahan mulai menggantikan dan meningkatkan kemampuan manusia di berbagai bidang. Kecerdasan yang ditunjukkan oleh mesin dikenal dengan kecerdasan buatan (*Artificial Intelligence*) yang merupakan bagian dari ilmu komputer. Kecerdasan buatan merupakan salah satu bidang dalam ilmu komputer yang ditujukan pada pembuatan *software* dan *hardware* yang dapat berfungsi sebagai sesuatu yang dapat berpikir seperti manusia (Sunarya et al., 2015). *Machine learning* merupakan sub dari bidang keilmuan kecerdasan buatan (*Artificial intelligence*). *Machine learning* dapat diartikan sebagai aplikasi komputer dan algoritma matematika yang diadopsi dengan cara pembelajaran yang berasal dari data dan menghasilkan prediksi di masa yang akan datang (Goldberg & Holland, 1988). Proses pembelajaran yang dimaksud adalah suatu usaha dalam memperoleh kecerdasan yang melalui dua tahap antara lain latihan (*training*)

dan pengujian (*testing*). Bidang *machine learning* berkaitan dengan pertanyaan tentang bagaimana membangun program komputer agar meningkat secara otomatis dengan berdasar dari pengalaman. Penelitian terkini mengungkapkan bahwa *machine learning* terbagi menjadi tiga kategori: *Supervised Learning, Unsupervised Learning, Reinforcement Learning* (Somvanshi & Chavan, 2016). Skema keterkaitan *artificial intelligence* dan *machine learning* dapat dijelaskan dalam Gambar 2.4.



GAMBAR 2.4 Skema *Artificial Intelligence* dan *Machine Learning*

Teknik yang digunakan oleh *Supervised Learning* adalah metode klasifikasi di mana kumpulan data sepenuhnya diberikan label untuk mengklasifikasikan kelas yang tidak dikenal. Sedangkan teknik *Unsupervised Learning* dikenal dengan *cluster* dikarenakan tidak ada kebutuhan untuk pemberian label dalam kumpulan data dan hasilnya tidak mengidentifikasi contoh dikelas yang telah ditentukan (Thupae et al., 2018). Teknik *Reinforcement Learning* berada antara *Supervised Learning* dan *Unsupervised Learning*, teknik ini bekerja dalam lingkungan yang dinamis di mana

konsepnya harus menyelesaikan tujuan tanpa adanya pemberitahuan dari komputer secara eksplisit jika tujuan tersebut telah tercapai (Das & Nene, 2017). Penelitian analisis sentimen yang dilakukan akan menggunakan teknik *Supervised Learning* karena data yang digunakan terlebih dahulu diberi label untuk dapat mengklasifikasikan kelas yang tidak dikenal.

2.6.1 Jenis-jenis *Machine Learning*

Algoritma *machine learning* digunakan untuk mengekstrak model yang didapat dari pengolahan data mentah (*raw data*) yang dapat digunakan untuk berbagai tugas dan tujuan. Cara kerja *machine learning* mirip dengan cara manusia belajar. Agar mesin, dalam konteks ini adalah komputer memiliki kemampuan belajar yang sama dengan manusia, maka perlu adanya proses *training* sebelum menganalisis, menilai dan mengambil tindakan.

Secara garis besar, algoritma *machine learning* dibagi menjadi tiga jenis, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Sebelum menyelesaikan masalah menggunakan *machine learning* kita harus memahami ketiga jenis *machine learning* tersebut karena algoritma-algoritma tersebut memiliki fungsi dan tujuan masing-masing.

a. *Supervised Learning, dataset* yang digunakan sudah memiliki label. Label adalah *tag* atau pengenal dari sebuah data. Misalnya terdapat sepotong buah yang memiliki atribut berwarna hijau, berat lebih dari 200 gram, kulitnya keras, berduri, bau yang menyengat, dan isi buahnya manis. Buah yang memiliki karakteristik seperti yang ini dikenali sebagai durian. Maka label dari atribut tersebut adalah durian.
b. *Unsupervised Learning, dataset* yang digunakan tidak memiliki label. Model *unsupervised* melakukan belajar sendiri untuk melabeli atau mengelompokkan data. Contoh kasus untuk *unsupervised* adalah dari data 100 pengunjung sebuah *website*, model akan belajar sendiri untuk mengelompokkan pengunjung. Misalnya dikelompokkan berdasarkan waktu kunjungan, lama kunjungan, jumlah klik, dan sebagainya.

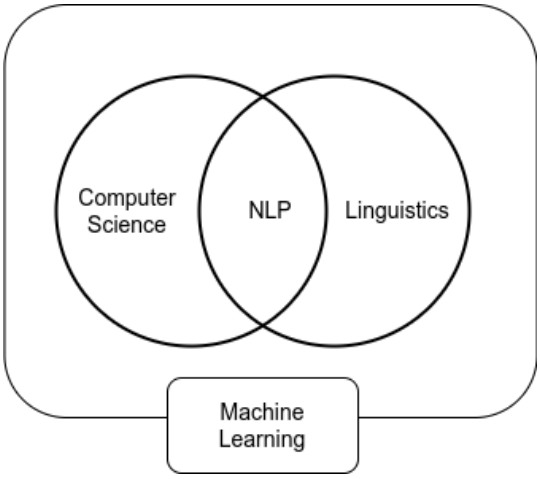
c. *Reinforcement Learning,* adalah model yang belajar menggunakan sistem *reward* dan *penalties*. Alpha Go adalah contoh terkenal dari *reinforcement learning*. Sebuah program yang dikembangkan Google Deepmind untuk memainkan permainan Go, sebuah permainan papan yang berasal dari Cina. Alpha Go mempelajari setiap langkah dalam jutaan permainan Go untuk terus mendapatkan *reward* yaitu memenangkan sebuah permainan.

2.6.2 *Machine Learning Workflow*

Dalam sebuah *machine learning*, ada tahapan-tahapan yang perlu dilalui sebelum *project* bisa diimplementasikan ke tahap produksi. Berikut adalah tahapan-tahapan yang dimaksud menurut buku *Hands on Machine Learning* karya Aurelien Geron:

a. *Exploratory Data Analysis*, bertujuan sebagai analisa awal terhadap data dan melihat bagaimana kualitas data.
b. *Data Preprocessing* dan *Data Cleaning*, tahap dimana data diolah lebih lanjut sehingga data siap digunakan dalam pengembangan *machine learning.*
c. *Model Selection*, pada tahap ini kita mulai memilih model yang akan dipakai serta melakukan optimasi *parameter* dari model tersebut.
d. *Model Evaluation*, lalu kita melakukan evaluasi terhadap model dengan melihat performanya terhadap data *testing*.
e. *Deployment*, ketika model telah dievaluasi, maka model siap untuk dipakai pada tahap produksi.
f. *Monitoring*, model yang telah dipakai dalam tahap produksi masih harus tetap dimonitor untuk menjaga kualitasnya. Pada tahap produksi model bisa saja menemukan data yang tidak dikenali sehingga performa model dapat menurun.

2.7 *Natural Language Processing*



GAMBAR 2.5 Ilustrasi NLP

Natural language Processing adalah suatu cabang ilmu komputer yang berkembang dari studi bahasa dan komputasi *linguistic* dalam cabang kecerdasan buatan (*Artificial Intelligence*). NLP bertujuan untuk mengembangkan dan merancang sebuah aplikasi yang mampu menjadi penengah antara interaksi manusia dan mesin dengan perangkat lain melalui penggunaan bahasa alami (Pustejovsky & Stubbs, 2012). Pada NLP, informasi yang akan digunakan berisi data-data yang tidak terstruktur sehingga diperlukan sebuah proses pengubahan bentuk menjadi data yang terstruktur untuk kebutuhan penelitian (*sentiment analysis*, *topic modelling*, dan lain-lain).

Merujuk pada penelitian yang dilakukan oleh (Mujilahwati, 2016), maka pada penelitian ini akan menggunakan beberapa teknik *preprocessing* yaitu:

2.7.1 *Tokenizing*

Pemotongan *string* input berdasarkan tiap kata penyusunnya. Tokenisasi dilakukan untuk memisahkan setiap kata yang ada pada setiap kata yang menyusun sebuah dokumen. Contoh: “Saya sedang belajar data science”,

setelah dilakukan tokenisasi maka kalimat tersebut akan menjadi “Saya”, “sedang”, “belajar”, “data”, “*science*”.

2.7.2 *Casefolding*

Merupakan salah satu bentuk *text preprocessing* yang paling sederhana dan efektif. *Casefolding* berperan dalam mengkonversi huruf-huruf yang berada didalam dokumen dengan format huruf besar (*uppercase*) menjadi huruf-huruf kecil (*lowercase*). Hanya huruf ‘a’ sampai ‘z’ yang diterima. Karakter selain huruf akan dihilangkan dan dianggap *delimiter*.

2.7.3 *Cleaning*

Tahapan untuk membersihkan *dataset* dari kata-kata yang tidak diperlukan untuk mengurangi *noise* yang ada pada tahap klasifikasi. Kata atau karakter yang akan dihilangkan pada proses *cleaning* seperti: simbol, angka, *link url*, *hashtag* (#), serta *mention* (*@username*).

2.7.4 *Stopwords Removal*

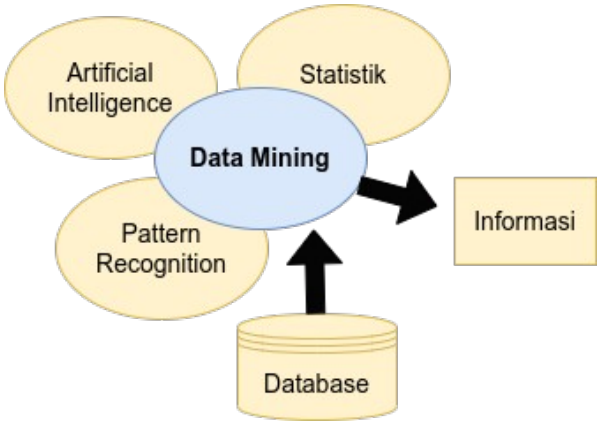
Proses memilah data dengan cara mengambil kata-kata penting dari proses *preprocessing* sebelumnya dengan menggunakan algoritma *stoplist* (membuang kata penting) atau *wordlist* (menyimpan kata penting). *Stopwords* yaitu kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopwords* dalam Bahasa Indonesia adalah ‘yang’, ’dan’, ’di’, ’dari’, dan lain-lain. *Stopwords* menghapus kata-kata yang memiliki informasi rendah dari sebuah teks agar kita dapat fokus pada kata-kata penting lainnya.

2.7.5 *Stemming*

Stemming adalah langkah selanjutnya dari *preprocessing* untuk merubah setiap kata menjadi bentuk kata dasar, untuk melakukan tahap *stemming* akan digunakan kamus daftar kata berimbuhan, kemudian akan dilakukan perbandingan antar kata yang ada pada *tweet* dengan daftar kata yang ada pada kamus (Nur dan Fithriasari, 2016).

2.8 *Data Mining*

Nama *data mining* sebenarnya mulai dikenal sejak tahun 1990 ketika pekerjaan pemanfaatan data menjadi sesuatu yang penting dalam berbagai bidang, mulai dari bidang akademik, bisnis, hingga medis (Gorunescu, 2011). *Data mining* dapat diterapkan pada berbagai bidang yang mempunyai jumlah data. Tetapi karena wilayah penelitian dengan sejarah yang belum lama, dan belum melewati masa remaja, maka *data mining* masih diperdebatkan posisi bidang pengetahuan yang memilikinya. Maka, Daryl Pregibon menyatakan bahwa "*data mining* adalah campuran dari statistik, kecerdasan buatan, dan riset basis data yang masih berkembang" (Gonunescu, 2011).



GAMBAR 2.5 Akar ilmu *data mining*

Ada istilah lain yang mempunyai makna yang sama dengan *data mining* yaitu *knowledge-discovery in database* (KDD). Memang *data mining* atau KDD bertujuan untuk memanfaatkan data dalam basis data dengan

mengolahnya sehingga menghasilkan informasi baru yang berguna. Seperti diilustrasikan pada Gambar 2.5, jika dilacak akar keilmuannya, ternyata *data mining* mempunya empat akar bidang ilmu sebagai berikut:

a. Statistik

   Bidang ini merupakan akar paling tua, tanpa ada statistik maka *data mining* mungkin tidak ada. Dengan menggunakan statistik klasik ternyata data yang diolah dapat diringkas dalam apa yang umum dikenal sebagai *exploratory data analysis* (EDA). EDA berguna untuk mengidentifikasi hubungan sistematis antarvariabel/fitur ketika tidak ada cukup informasi alami yang dibawanya.

b. Kecerdasan Buatan atau AI

   Bidang ilmu ini berbeda dengan statistik. Teorinya dibangun berdasarkan teknik heuristik sehingga AI berkontribusi terhadap teknik pengolahan informasi berdasarkan pada model penalaran manusia. Salah satu cabang AI, yaitu pembelajaran mesin atau *machine learning,* merupakan disiplin ilmu yang paling penting yang direpresentasikan dalam pembangun *data mining,* menggunakan teknik dimana sistem komputer belajar dengan pelatihan.

c. Pengenalan Pola

   Sebenarnya *data mining* juga menjadi turunan bidang pengenalan pola, tetapi hanya mengolah data dari basis data. Data yang diambil dari basis data untuk diolah bukan dalam bentuk relasi, melainkan dalam bentuk normal pertama sehingga set data dibentuk menjadi bentuk normal pertama. Akan tetapi, *data mining* mempunyai ciri khas yaitu pencarian pola asosiasi dan pola sekuensial.

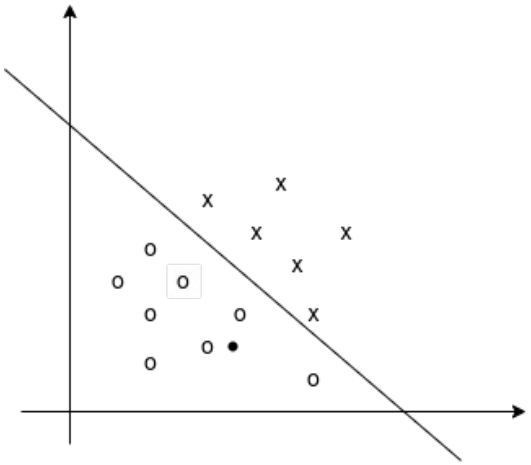
d. Sistem Basis Data

   Akar ilmu keempat dalam bidang *data mining* yang menyediakan informasi berupa data yang akan digali menggunakan metode-metode yang disebutkan sebelumnya.

Dari penjelasan diatas jelas jelas bahwa di satu sisi ada sejumlah data dalam jumlah besar yang secara sistematis belum dieksplorasi, dan di sisi lain, kekuatan teknik komputasi dan ilmu komputer sudah tumbuh secara eksponensial sehingga menyebabkan tekanan pada kebutuhan untuk membuka informasi yang tersembunyi dari data menjadi meningkat. Bidang *data mining* menjadi jawaban untuk menyelesaikan persoalan diatas yang pada awalnya tidak mungkin untuk di deteksi dengan cara tradisional dan hanya menggunakan kemampuan analisis manusia.

2.9 Konsep Klasifikasi

Andaikan set data dengan fitur rata-rata dan standar deviasi digambar dalam diagram kartesius akan menajadi seperti pada Gambar 2.6. Pada gambar tersebut ada 2 kelas yaitu kelas A (o) dan kelas B (x).



GAMBAR 2.6 Diagram pemetaan standar deviasi dan rata-rata

Dari diagram tersebut dapat diamati bahwa data kelas A menyebar di wilayah kiri bawah, sedangkan data kelas B menyebar di wilayah kanan atas.

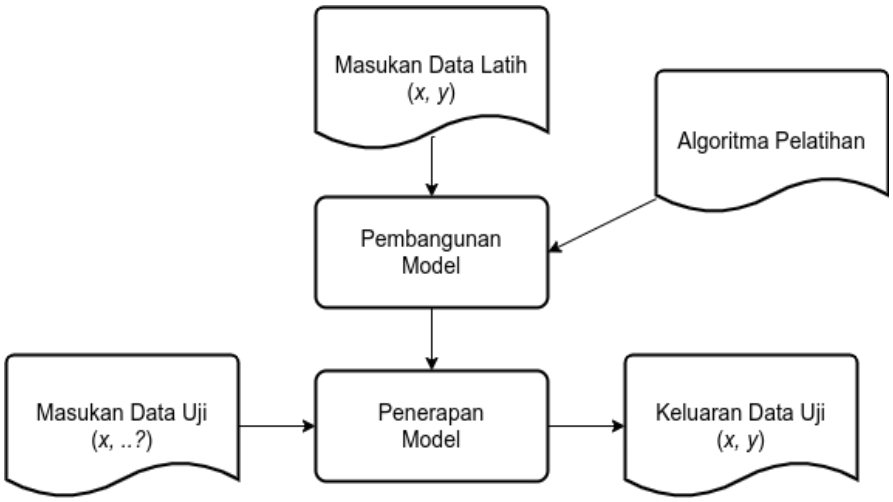
Dapat diamati pula bahwa data dari dua kelas tersebut dapat dipisahkan dengan mudah karena mengelompok dalam wilayah terpisah. Garis lurus yang memisahkan dua kelas berbeda tersebut disebut dengan garis keputusan (*decision line*). Garis tersebut menyatakan bahwa data yang terletak di sisi kiri bawah adalah kelas A, sedangkan yang terletak di sisi kanan atas adalah kelas B. Garis inilah yang memberikan jawaban ketika ada data yang baru, seperti yang disimbolkan oleh simbol titik solid. Dengan mengamati lokasi tiitk tersebut berada di sisi kiri bawah garis keputusan, maka data baru tersebut diprediksi masuk pada kelas A.

Contoh tersebut memberikan gambaran sistem klasifikasi dalam *data mining*. Kuantitas yang digunakan sebagai basis ukuran dalam menilai objek yaitu rata rata dan standar deviasi disebut dengan fitur. Kumpulan dari fitur-fitur yang memberikan deskripsi sebuah objek disebut dengan data. Kumpulan dari sebuah data (objek) disebut dengan set data. Untuk sebuah objek x mempunyai n fitur dinyatakan dengan:

$x = [x1, x2, ..., xn]$

Garis lurus yang memisahkan dua kelas tersebut berperan untuk membagi wilayah fitur menjadi 2 wilayah atau lebih yang berbeda kelas disebut klasifikator. Jika sebuah vektor yang baru yang belum diketahui label kelasnya terletak di wilayah kelas A maka vektor tersebut diprediksi masuk kelas A, jika terletak di kelas B maka vektor tersebut di prediksi masuk ke kelas B. Prediksi tersebut tidak selalu benar, misalnya jika sebuah vektor yang sebenarnya adalah kelas A tetapi hasil prediksi memberikan label bahwa vektor tersebut adalah kelas B, maka hal ini disebut dengan misklasifikasi. Misklasifikasi dapat terjadi dalam sistem klasifikasi yang dibangun akibat masuknya model klasifikator dalam wilayah lokal optimal. Masalah ini dipengaruhi oleh algoritma klasifikasi itu sendiri. Data/vektor yang sudah diketahui sebelumnya untuk label kelas dan digunakan untuk membangun model klasifikator disebut data latih atau *training data*. Data/vektor yang belum diketahui (dianggap belum diketahui) label kelasnya untuk kemudian di

prediksi kelasnya menggunakan model klasifikator yang sudah dibangun disebut dengan data latih atau *testing data.*



GAMBAR 2.7 Diagram klasifikasi

Model yang sudah dibangun pada saat pelatihan kemudian dapat digunakan untuk memprediksi label kelas dari data yang baru yang belum diketahui label kelasnya. Dalam pembangunan model selama proses pelatihan tersebut diperlukan adanya suatu algoritma untuk membangunnya yang disebut sebagai algoritma pelatihan (*learning algorithm*). Ada banya algoritma pelatihan yang sudah dikembangkan oleh para peneliti seperti Decision Tree, K-Nearest Neighbor, Artificial Neural Network, Support Vector Machine, dan sebagainya. Setiap model mempunyai kelebihan dan kekurangan masing-masing. Akan tetapi, semua algoritma mempunyai prinsip yang sama yaitu melakukan suatu pelatihan sehingga di akhir pelatihan model dapat memetakan (memprediksi) setiap vektor masukan ke label kelas keluaran dengan benar.

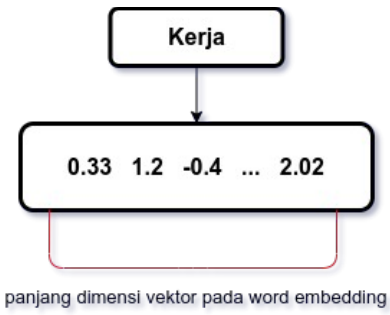
2.9 *Word Embedding*

Word embedding adalah sebuah pendekatan yang digunakan untuk merepresentasikan sebuah kata menjadi sebuah vektor. *Word embedding* merupakan pengembangan komputasi pemodelan kata-kata yang sederhana seperti perhitungan menggunakan jumlah dan frekuensi kemunculan kata

dalam sebuah dokumen. *Word embedding* menggambarkan kedekatan sebuah kata atau dokumen dalam kontekstual sesuai dengan data uji yang digunakan dalam pembentukannya, sehingga seringkali kedekatan tersebut bukan merupakan makna dari sebuah kata.

Dalam *word embedding* dapat digambarkan bahwa setiap "kata" diwakilkan oleh sebuah titik di dalam luasan bidang tertentu, titik-titik ini kemudian akan dipelajari oleh perhitungan *word embedding* dan satu titik akan dipindahkan menjauh atau mendekati titik yang lainnya, berdasarkan kata-kata lain yang mengelilingi titik tersebut. Hal ini dilakukan terus menerus hingga sampai pada sebuah kondisi dimana semua titik tidak dapat dipindahkan lagi mendekati (atau menjauhi) titik yang lainnya. Sehingga hasil akhir dari iterasi ini dapat memberikan sebuah gambaran dimana kata-kata dengan makna yang serupa akan cenderung berada dalam satu area yang sama dalam bidang tersebut atau dengan kata lain. Kata-kata yang ada dalam satu area pada bidang tersebut dan mempunyai jarak kedekatan yang kecil cenderung mempunyai kesamaan.

Metode *Word embedding* mengkonversi "kata" menjadi "vektor" yang berisi angka-angka dengan ukuran yang cukup kecil untuk mengandung informasi yang lebih banyak. Informasi yang didapat akan cukup banyak hingga vektor dapat mendeteksi makna, misal kata "marah" dan "mengamuk" itu lebih memiliki kedekatan nilai dibandingkan kata "marah" dan "bahagia". Illustrasi dari *word embedding* dapat dilihat pada gambar dibawah ini.



GAMBAR 2.8 Ilustrasi *Word Embedding*

2.10 *Term Frequency-Inverse Document Frequency*

Metode TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu *frequency* kemunculan sebuah kata di dalam sebuah dokumen tertentu dan *inverse frequency* dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut.

2.11 *Feature Selection*

Feature Selection atau seleksi fitur merupakan sebuah proses dalam *machine learning* yang menggunakan sekumpulan fitur yang dimiliki oleh data untuk digunakan dalam proses algoritma. (Rokach dkk., 2006) mengatakan bahwa *feature selection* telah menjadi bidang penelitian aktif yang digunakan dalam pengenalan pola, statistik, dan *data mining*. *Feature selection* merupakan salah satu faktor penting yang dapat mempengaruhi tingkat akurasi dari klasifikasi. Karena apabila dalam *dataset* terdapat sejumlah fitur, dimensi *dataset* akan menjadi besar dan dapat menyebabkan rendahnya akurasi yang didapat. Permasalahan utama dalam *feature selection* adalah pengurangan dimensi, dimana awalnya semua atribut diperlukan untuk mendapatkan hasil yang maksimal. Terdapat beberapa alasan mengapa perlu dilakukan pengurangan dimensi menurut (Rokach dkk., 2006):

a. Penurunan biaya pelatihan algoritma (*decreasing the training cost*).
b. Meningkatkan kinerja pelatihan algoritma (*increasing the training performance*).
c. Mengurangi dimensi yang tidak relevan (*reducing irrelevant dimensions*).
d. Mengurangi dimensi yang berlebihan (*reducing redundant dimensions*).

Ide utama dari *feature selection* yaitu memilih subset dari fitur yang ada tanpa transformasi, karena tidak semua fitur relevan dengan masalah. Bahkan ada fitur atau atribut yang mengganggu dan dapat mengurangi akurasi. *Noisy feature* atau fitur yang tidak dipakai tersebut harus dihapus agar dapat meningkatkan akurasi yang diperoleh. Selain itu, fitur atau atribut yang terlalu banyak dapat memperlambat proses komputasi.

Pada penelitian ini, menggunakan cara manual untuk melakukan tahapan seleksi fitur. Yaitu dengan melakukan beberapa percobaan dengan mengkombinasikan *word embedding*, *text preprocessing*, dan model *Logistic Regression*. Hasil dari percobaan tersebut akan menentukan fitur mana yang memiliki akurasi terbaik.

2.12 *Logistic Regression*

Logistic Regression atau Regresi Logistik merupakan salah satu metode yang dapat digunakan untuk mencari hubungan variabel respon yang bersifat berskala nominal atau ordinal dengan dua kategori atau mempunyai skala nominal atau ordinal dengan lebih dari dua kategori dengan satu atau lebih variabel prediktor dan variabel respon yang bersifat kontinyu atau kategorik. *Logistic Regression* juga merupakan hubungan antara regresi logit dan regresi probit. Yang termasuk dalam regresi ini adalah regresi biner (dengan respon Y hanya dua kategori) (Tirta, 2009). Model dari regresi logistik ini, adalah:

$$logit(\pi i)=\ln(\pi i1-\pi i)=\alpha+\beta ix1,i+...+\beta kxk,i,i=1,...,n$$

Menurut (Yudisasanta A. dan Ratna M., 2012) regresi logistik multinomial merupakan regresi logistik yang digunakan saat variabel dependen mempunyai skala yang bersifat *polichotomous* atau multinomial. Skala multinomial adalah suatu pengukuran yang dikategorikan menjadi lebih dari dua kategori. Metode

yang digunakan dalam penelitian ini adalah regresi logistik dengan variabel dependen berskala nominal dengan lima kategori.

2.13 Evaluasi Sistem

Pada tahap klasifikasi tentunya diharapkan agar semua dataset yang ada dapat melakukan proses klasifikasi dengan tepat. Oleh karena itu diperlukan sebuah cara untuk mengukur hasil kerja klasifikasi, hal tersebut dapat dilakukan menggunakan pengujian akurasi, *precision,* dan *recall*.

2.13.1 Akurasi

Sebuah sistem yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua *dataset* dengan benar. Akan tetapi, tidak dapat dipungkiri apabila kinerja suatu sistem tidak bisa bekerja 100% benar. Oleh karena itu, sebuah sistem klasifikasi juga harus diukur kinerjanya. Umumnya cara mengukur kinerja klasifikasi menggunakan *confusion matrix.*

Confusion matrix merupakan tabel yang mencatat hasil kerja klasifikasi. Tabel 2.3 merupakan contoh *confusion matrix* yang melakukan klasifikasi masalah biner (dua kelas) untuk dua kelas, misalnya 0 dan 1. Setiap sel *f*ij dalam matriks menyatakan jumlah *record*/data dari kelas fi yang hasil prediksinya masuk ke kelas j. Misalnya sel *f*11 adalah jumlah data dalam kelas 1 yang secara benar dipetakan ke kelas 1, dan *f*10 adalah data dalam kelas 1 yang dipetakan secara salah ke kelas 0.

TABEL 2.2 *Confusion Matrix*

*f*ij		Kelas hasil prediksi	
		Kelas = 1	Kelas = 0
Kelas asli (*i*)	Kelas = 1	*f*11	*f*10
	Kelas = 0	*f*01	*f*00

Berdasarkan isi data *confusion matrix,* maka dapat diketahui jumlah data dari masing-masing kelas yang diprediksi secara benar yaitu ($f11 + f00$) dan data yang diklasifikasikan secara salah yaitu ($f10 + f01$). Kuantitas *confusion matrix* dapat diringkas menjadi dua nilai, yaitu akurasi dan laju *error*. Dengan mengatahui jumlah data yang diklasifikasi secara benar maka dapat diketahui akurasi hasil prediksi, dan dengan mengetahui jumlah data yang diklasifikasikan secara salah maka diketahui laju *error* dari prediksi yang dilakukan. Dua kuantitas ini digunakan sebagai metrik kinerja klasifikasi. Untuk menghitung akurasi digunakan formula sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah data yang diprediksi secara benar}}{\text{Jumlah prediksi yang dilakukan}}$$

$$= \frac{f11 + f00}{f11 + f10 + f01 + f00}$$

Untuk menghitung laju *error* (kesalahan prediksi) digunakan formula sebagai berikut :

$$\textit{error} = \frac{\text{Jumlah data yang diprediksi secara salah}}{\text{Jumlah prediksi yang dilakukan}}$$

$$= \frac{f11 + f00}{\mathrm{f11} + \mathrm{f10} + \mathrm{f01} + \mathrm{f00}}$$

Semua algoritma klasifikasi berusaha untuk membentuk model yang mempunyai akurasi yang tinggi (laju *error* yang rendah). Umumnya model yang dibangun dapat memprediksi dengan benar semua data yang menjadi data latihnya, tetapi ketika model berhadapan dengan data uji barulah kinerja model dari sebuah algoritma klasifikasi ditentukan.

2.13.2 *Precision* Dan *Recall*

Metrik spesifisitas mengukur tingkat kemampuan sistem untuk mengenali data yang sebenarnya negatif. Nilai yang dilibatkan adalah *true negative* dan *false positive.* Data yang masuk dalam proporsi *false positive* adalah data yang sebenarnya negatif tapi dikenali sebagai positif. Metrik ini kurang cocok digunakan dalam bidang pencarian informasi, misalnya mencari data-data dalam dokumen yang relevan sesuai dengan yang diinginkan. Data yang didapatkan dalam pencarian akan terbagi menjadi dua kelompok, yaitu data yang ditemukan yang relevan dan data yang ditemukan tapi tidak relevan. Dalam hal ini biasanya tidak penting untuk mengukur kemampuan sistem dalam mengenali data yang tidak relevan yang dikenali sebagai relevan. Oleh karena itu, metrik spesifisitas tidak digunakan disini. Metrik yang cocok digunakan untuk mengukur pencarian data seperti ini adalah *precision* dan *recall.*

Dalam bidang pencarian informasi, *precision* (disebut juga *positive prediction value*) merupakan metrik untuk mengukur kinerja sistem dalam mendapatkan data relevan yang terbaca (dalam bidang pencarian informasi). Dalam bidang *data mining, precision* adalah jumlah data yang *true positive* (jumlah data positif yang dikenali secara benar sebagai positif) dibagi dengan jumlah data yang dikenali sebagai positif, sedangkan *recall* adalah jumlah data yang *true positive* dibagi dengan jumlah data yang sebenarnya positif (*true positive* + *true negative*). Berikut persamaan yang digunakan untuk menghitung *precision:*

$$Precision = \frac{TP}{TP + FP}$$

Persamaan yang digunakan untuk menghitung *recall* disajikan sebagai berikut:

$$Recall = \frac{TP}{TP + FN}$$

Nilai *precision* dan *recall* biasanya mempunyai hubungan *trade-off* yang terbalik. Untuk meningkatkan *precision* biasanya dibayar dengan menurunkan *recall,* sebaliknya untuk meningkatkan *recall* maka dibayar dengan menurunkan *precision.* Penyajian kinerja sistem dengan *precision* dan *recall* biasanya tidak terpisah. Nilai *precision* dan *recall* dapat digabung dalah satu metrik yaitu F-measure (Rijsbergen, 1979). F-measure merupakan nilai rata-rata harmonik terbobot diantara *precision* dan *recall.* Berikut formula yang umum digunakan:

$$F = 2 \, x \, \frac{precision \, x \, recall}{precision + recall}$$

Nilai yang didapat dari persamaan diatas disebut F1-measure karena *precision* dan *recall* mempunyai bobot yang sama.

2.14 Penelitian Terkait

Penelitian yang pernah dilakukan tentang analisis sentimen pada media sosial twitter. Penelitian terkait dapat dilihat pada Tabel 2.3.

TABEL 2.3 Penelitian Terkait Topik

No	Penelitian	Judul	Tahun	Keterangan
1	Willa Oktinas	"Analisis Sentimen Pada Acara Televisi Menggunakan *Improved K-Nearest Neighbor*"	2017	Pada penelitian ini, pengujian analisis sentimen berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor* menghasilkan *accuracy* tertinggi dengan nilai k=10 sebesar 90%.
2	Ghulam Asrofi Buntoro	"Analisis Sentimen Calon Gubernur Jawa Timur 2018 dengan	2019	Penelitian ini menyimpulkan bahwa algoritma *naive bayes*

		Metode *Naive Bayes Classifier*”		dapat digunakan utuk mencari nilai *probability* dari tweet dengan membaginya dalam tiga kategori, yaitu sentimen positif, negatif dan netral.
3	Ni Made Yeni Dwi Rahayu	“Rancangan Penerapan Metode *Naïve Bayes* dalam Mendeteksi *Hate Speech* di Media Sosial”	2018	Penelitian ini menyimpulkan bahwa proses klasifikasi menjadi lebih akurat jika semakin banyak jumlah *keyword training* yang digunakan.
4	Bagas Prakoso Putra, Budhi Irawan, Casi Setianingsih	“Deteksi Ujaran Kebencian dengan Menggunakan Algortima *Convolutional Neural Network* pada Gambar”	2018	Pada Penelitian ini membuktikan bahwa ukuran *Epoch* dan *Batch size* mempengaruhi kinerja sistem dan hasil *accuracy*, Pada proses pengujian kinerja sistem didapatkan rata-rata *precision* sebesar 99.46%, *recall* sebesar 97.99%, dan *Accuracy* sebesar 99.8%.
5	Aulil Amri	“Implementasi Algoritma *Random Forest* untuk Mendeteksi *Hate Speech* dan *Abusive language* pada Twitter Bahasa Indonesia”	2020	Pada Penelitian ini, kombinasi parameter terbaik adalah pada *feature selection* CO-L1 dengan nilai *n_estimators*=100, *max_depth*=10, *criterion*=”*entropy*”, *min_samples_split*=10, dan *max_features*=”*auto*”. Dengan hasil akurasi sebesar 76,20%.
6	Aini Suri Talita, Aristiawan Wiguna	“Implementasi Algoritma *Long Short Term Memory* (LSTM) untuk Mendeteksi	2019	Berdasarkan hasil penelitian yang dilakukan dengan menggunakan data

		Ujaran Kebencian (*Hate Speech*) pada Kasus Pilpres 2019"		testing 190 kalimat dari 950 kalimat dari dataset, algoritma Long Short Term Memory sudah cukup baik dalam mendeteksi kalimat ujaran kebencian dengan nilai parameter recall mencapai 0.7021.

Penelitian yang pernah dilakukan tentang metode *Logistic Regression*. Penelitian terkait dapat dilihat pada Tabel 2.4.

TABEL 2.4 Penelitian terkait algoritma *Logistic Regression*

No	Penelitian	Judul	Tahun	Keterangan
1	Sherli Yualinda, Elis Hernawati, Dedy Rahman Wijaya	"Aplikasi untuk Memprediksi Kemiskinan Berbasis Data *E-Commerce* Menggunakan Algoritma *Logistic Regression* dan *Sparse Learning Based Feature Selection*"	2020	Aplikasi dapat menampilkan hasil prediksi kemisikinan di suatu daerah dengan menggunakan berbasis *logistic regression* dengan menggunakana algoritma *sparse learning based feature selection*.
2	Ria Indah Fitria, Nur Tulus Ujianto	"Komparasi Algoritma *Logistic Regression* dan *Naive Bayes* untuk Menyeleksi Melamar Pekerjaan Perusahaan Besar Bagi Alumni SMK"	2021	Penelitian ini dilakukan menggunakan komparasi dua algoritma antara lain:, *Algoritma Naive Bayes* dan *Algoritma regresi logistik*. Dari kedua algoritma hasil eksperimen algoritma *Algoritma Naive Bayes* menunjukkan tingkat akurasi sebesar 0,9741, dan algoritma regresi logistik sebesar 0,956 dalam mendeteksi klasifikasi.
3	Rachmat Hendayana	"Penerapan Metode Regresi Logistik Dalam Menganalisis	2012	Penerapan Regresi Logistik dengan nilai duga maksimum

		Adopsi Teknologi Pertanian”		*likelihood* menggunakan *Minitab* dapat direkomendasikan untuk menganalisis adopsi teknologi pertanian pada kasus adopsi teknologi VUB padi.
4	Jefri Junifer Pangaribuan, Henry Tanjaya, Kenichi	“Mendeteksi Penyakit Jantung Menggunakan *Machine Learning* dengan Algoritma *Logistic Regression*”	2021	Perubahan cost dari iterasi ke-1 hingga iterasi ke-76 sebesar 0.33572448 sedangkan pada iterasi ke-76 hingga iterasi ke-152, perubahan cost hanya sebesar 0.0204791, dan terus terjadi penurunan cost hingga iterasi ke-200.
5	Harsih Rianto, Romi Satria Wahono	“*Resampling Logistic Regression* untuk Penanganan Ketidakseimbangan *Class* pada Prediksi Cacat *Software*”	2018	Pada penelitian ini mendapatkan hasil pada eksperimen 10 dataset adalah FP 88.35% dan rata-rata AUC sebesar 0.818.
6	Dery Yuliansyah	“Perbandingan Algortima *Naive Bayes* dengan Regresi Logistik untuk Klasifikasi *Hate Speech*”	2020	Pada Penelitian ini, dari proses cross validation yaitu training dan testing, algoritma Logistic Regression menghasilkan nilai accuracy 62.00%.

BAB III
METODOLOGI PENELITIAN

3.1 Waktu Dan Tempat Penelitian

Penelitian ini dilaksanakan sejak semester ganjil 2022/2023 pada Program Studi Sistem Informasi di Sekolah Tinggi Manajemen dan Ilmu Komputer. Berikut adalah tabel *timeline* penelitian yang dilakukan.

TABEL 3.1 *Timeline* Penelitian

No.	Kegiatan	Waktu Penelitian						
		Mei	Jun	Jul	Agu	Sep	Okt	Nov
1.	Studi Literatur							
2.	Pengumpulan Data							
3.	Preprocessing Data							
4.	Pembuatan Model Klasifikasi Logistic Regression							
5.	Pengujian Model							
6.	Proses Hasil Klasifikasi							
7.	Penulisan Skripsi							

3.2 Pengertian Metodologi Penelitian

David H. Penny (dalam Akhmadi, 2009, hlm. 1) menjelaskan bahwa penelitian adalah pemikiran yang sistematis mengenai berbagai jenis masalah yang pemecahannya memerlukan pengumpulan dan penafsiran kata-kata. Menurut Mohammad Ali (dalam Akhmadi, 2009, hlm. 12) penelitian adalah suatu cara untuk memahami sesuatu dengan melalui penyelidikan atau melalui usaha mencari bukti-bukti yang muncul sehubungan dengan masalah itu, yang dilakukan secara hati-hati sekali sehingga diperoleh pemecahannya.

Dari batasan-batasan di atas dapat diambil kesimpulan bahwa yang dimaksud dengan metodologi penelitian adalah suatu cabang ilmu pengetahuan yang membicarakan/mempersoalkan mengenai cara-cara melaksanakan

penelitian sampai menyusun laporannya berdasarkan fakta-fakta atau gejala-gejala secara ilmiah.

3.3 Identifikasi Masalah

Tahap ini merupakan tahapan awal dari metodologi penelitian untuk menentukan latar belakang dan rumusan masalah serta tujuan dari penelitian ini. Selain itu pada tahap ini juga menentukan batasan-batasan pada penelitian yang dilakukan. Adapun hal-hal yang dilakukan pada tahap identifikasi masalah yaitu sebagai berikut:

a. Latar Belakang

   Latar belakang adalah dasar yang membuat penelitian ini menjadi perlu dilakukan, dalam penelitian ini fenomena ujaran kebencian dan bahasa kasar pada media sosial dan kecenderungan pengguna sosial dalam menyampaikan komentar berupa ujaran kebencian, hinaan, cacian, dan lain-lain.

b. Rumusan Masalah

   Rumusan masalah adalah masalah-masalah yang harus diselesaikan dalam penelitian, contohnya metode apa yang digunakan, dan bagaimana memperoleh data.

c. Batasan Masalah

   Batasan masalah menjadi salah satu yang harus dilakukan pada tahap identifikasi agar penelitian menjadi lebih fokus sehingga tujuan penelitian dapat tercapai.

d. Tujuan Penelitian

   Tujuan penelitian meliputi hal-hal yang ingin dicapai dalam penelitian. Dalam penelitian ini tujuan utamanya yaitu untuk mengklasifikasikan *tweet* apakah tergolong ke dalam ujaran kebencian atau bahasa kasar dan level ujaran kebencian dari *tweet* tersebut apakah kuat, sedang, atau lemah.

3.4 Pengumpulan Data

Pada tahap ini akan menjelaskan tentang bagaimana proses pengumpulan data dilakukan dan referensi apa saja yang digunakan dalam pengumpulan data. Penjelasan lengkapnya sebagai berikut:

3.4.1 Pengumpulan Data Twitter

Data yang diperlukan pada penelitian ini adalah data twitter berupa *tweet* yang memiliki kemungkinan termasuk ke dalam *tweet* ujaran kebencian dan bahasa kasar. Proses pengumpulan data tidak dilakukan dengan cara memilih dan memberi label pada data *tweet* secara manual, karena pada penelitian sebelumnya yang dilakukan oleh (Ibrohim & Budi, 2019) tentang ujaran kebencian dan bahasa kasar telah melakukan *crawling* data *tweet* ujaran kebencian dan bahasa kasar.

(Ibrohim & Budi, 2019) mempersilahkan siapa saja yang ingin melakukan penelitian dengan topik yang berhubungan dan memerlukan dataset tentang ujaran kebencian dan bahasa kasar dapat menggunakan dataset yang sama secara gratis (*free*). Oleh karena itu, peneliti meminta izin kepada (Ibrohim & Budi, 2019) untuk menggunakan dataset beliau dengan cara menghubungi melalui *email*, dan beliau mengizinkan serta memberikan link untuk mengunduh dataset tersebut melalui email.

Dalam penelitian ini, menggunakan dataset ujaran kebencian dan penyalahgunaan bahasa di Twitter dari beberapa penelitian sebelumnya yang terdiri dari (Alfina dkk., 2017, 2018), (Putri, 2018), dan (Ibrohim & Budi, 2019). Selain menggunakan dataset Twitter dari penelitian sebelumnya, (Ibrohim & Budi, 2019) juga merangkak *tweet* untuk memperkaya dataset sehingga dapat mencakup jenis penulisan pidato kebencian dan bahasa kasar yang mungkin belum ada dalam data dari penelitian sebelumnya. Untuk proses anotasi, pada penelitian sebelumnya membangun sistem anotasi berbasis web

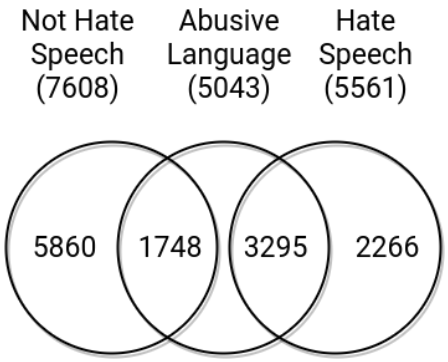
untuk memudahkan dalam membuat anotasi data sehingga dapat mempercepat proses anotasi dan meminimalkan kesalahan anotasi. Kami juga melakukan anotasi standar emas untuk menguji apakah bahasa sudah memahami tugas atau tidak.

Dalam penelitian ini, kami melakukan diskusi dan konsultasi dengan ahli Bahasa (Nurasijah, 2018) untuk mendapatkan pedoman anotasi yang valid dan anotasi standar emas. Data Twitter yang digunakan untuk standar emas berasal dari penelitian sebelumnya (Alfina dkk., 2017; Ibrohim & Budi, 2018) dan buku pedoman ucapan kebencian (Komnas HAM, 2015)

Pada tahap anotasi pertama, berhasil mengumpulkan 16.500 *tweet* dari proses perayapan dan penelitian sebelumnya (Alfina dkk., 2017, 2018; Putri, 2018). Dari fase ini, kita mendapatkan 11.292 (68,44% total *tweet* yang dijelaskan dalam fase pertama) yang terdiri dari 6.187 tweet ucapan tidak kebencian dan 5.105 tweet ucapan benci yang memiliki perjanjian 100% (dataset dapat diandalkan). Menurut (McHugh, 2012), jumlah persentase ini dari dataset yang dapat diandalkan (data dapat digunakan untuk eksperimen penelitian) menunjukkan bahwa penilaian tersebut menetapkan tingkat persetujuan yang meningkat.

Selanjutnya, dalam fase anotasi kedua, tercatat 5.700 *tweet* ucapan kebencian (5.105 *tweet* dari anotasi fase pertama dan 595 *tweet* dari (Ibrohim & Budi, 2019)). Dari proses anotasi dua fase ini, berhasil mendapatkan 13.169 *tweet* yang telah digunakan untuk eksperimen penelitian yang terdiri dari 7.608 *tweet* ucapan tidak membenci (6.187 *tweet* dari anotasi tahap pertama dan 1.421 *tweet* dari (Ibrohim & Budi, 2018)) dan 5.561 *tweet* ucapan kebencian.

Distribusi Bahasa kasar ke *tweet* ucapan tidak membenci pidato dan *tweet* ucapan benci dari *tweet* yang dikumpulkan dapat dilihat pada Gambar 3.1. Dari Gambar 3.1, kita dapat melihat bahwa tidak semua pidato kebencian adalah bahasa yang kasar. Sebaliknya, bahasa yang kasar juga tidak harus berupa pidato kebencian.



Gambar 3.1 Distribusi Data

3.3.2 Studi Literatur

Merupakan tahap untuk memperoleh semua informasi seperti mencari referensi dari jurnal, buku, paper internasional, youtube dan referensi lainnya yang berhubungan dengan penelitian ini. Referensi yang digunakan adalah referensi yang berhubungan dengan penelitian, seperti teori-teori seputar penelitian yang pernah dilakukan sebelumnya.

3.4 Analisa Sistem

Setelah melakukan identifikasi masalah dan pengumpulan data, proses selanjutnya yaitu analisa. Analisa merupakan tahapan untuk mempelajari dan melakukan evaluasi terhadap suatu permasalahan serta bertujuan untuk mengetahui gambaran jelas mengenai penelitian yang dilakukan. Tahapan analisa yang dilakukan sebagai berikut:

3.4.1 Analisa *Data Preparation*

Awalnya dataset yang diberikan oleh (Ibrohim & Budi, 2019) terdiri dari 4 label, dimana ada label untuk *hate speech*, label target *hate speech* yang akan dicapai, label level dari *hate speech*, dan label *abusive*. Namun pada penelitian ini label-label tersebut dibagi menjadi tiga label yaitu label *hate*

speech, *abusive*, dan level *hate speech*. Untuk kelas target *hate speech* yang akan dicapai tidak digunakan dalam penelitian ini.

3.4.2 Analisa *Data Preprocessing*

Kemudian tahap *data preprocessing*. Dimana data dan model yang sudah disiapkan pada tahap sebelumnya akan digunakan pada tahap ini. Dengan menggunakan data yang telah dibuat, dataset akan di proses lagi untuk mendapatkan dataset yang bersih untuk tahap produksi. Tahapan *data preprocessing* yang dilakukan yaitu *casefolding*, *stopwords*, *stemming,* dan *tokenizing*. Tahap pertama dilakukan *casefolding* yaitu dengan mengubah semua huruf yang ada dalam dataset menjadi huruf kecil, hal ini memiliki pengaruh besar untuk tahap produksi. Kemudian *stopwords* yaitu membuang kata yang dianggap tidak memiliki peran yang signifikan pada penelitian, pada tahap ini pula terdapat beberapa teknik data *cleaning*. Tahap ketiga yaitu *stemming* yaitu merubah seluruh kata menjadi kata dasar. Terakhir yaitu *tokenizing* dimana setiap dataset yang terdiri dari sebuah *sentence* atau kalimat, agar dapat diproses dalam model *machine learning* harus ditokenisasi terlebih dahulu, yaitu dengan memisahkan setiap kata yang ada dalam suatu kalimat.

3.4.3 Analisa TFIDF Vectorizer

Pada tahap ini akan dilakukan proses seleksi fitur terhadap data yang akan digunakan nantinya. Seleksi fitur berfungsi untuk menemukan fitur-fitur apa saja yang penting dan kurang penting di dalam data. Sehingga setelah dilakukannya proses seleksi fitur diharapkan hasil dari seleksi fitur adalah fitur-fitur terbaik yang dapat menghasilkan performa terbaik utuk proses klasifikasi. Proses *feature selection* dimulai dari dataset awal yang masih berupa sebuah kalimat. Untuk dapat diproses oleh model *word embeddin*g, dataset ditokenisasi terlebih dahulu ke dalam bentuk *word* (kata). Setelah ditokenisasi,

data akan diubah menjadi sebuah vektor, dengan panjang dimensi maksimal 256. Sekarang setiap data telah menjadi vektor dengan maksimum dimensi 256. Setelah vektor data didapatkan, maka data akan melalui tahap *encoding*. Sebuah tahap pemrosesan untuk setiap kalimat yang akan diubah menjadi sebuah kata dan kemudian di *encode* menjadi vektor. Hasil dari proses *encoding* yaitu hasil dari rata-rata nilai untuk setiap vektor dalam setiap kata dan kalimat, nilai inilah yang akan digunakan untuk proses training.

3.4.4 Analisa *Dataset Splitting*

Pada tahap ini peneliti mencoba melakukan pembagian dataset yaitu data uji dan data latih, mencari proporsi terbaik dari setiap data. Tahap ini bertujuan agar tidak terjadi sebuah *data leakage*, atau kebocoran data.

3.4.5 Analisa *K-Fold Cross Validation*

Metode pembelajaran mesin sering gagal membuat model data karena mempelajari fitur tertentu dari set pelatihan, yang tidak ada dalam set pengujian. Ini terjadi jika model terlalu cocok dengan data pelatihan, tetapi tidak dapat menggeneralisasi dalam sampel baru. Pada tahap ini penulis mencoba mencari validasi silang terbaik agar dataset terbagi ke beberapa silangan secara merata.

3.4.6 Analisa *Grid Search CV*

Analisa untuk menemukan pencarian parameter terbaik dalam sebuah pemodelan *machine learning*. *Grid Search* adalah pencarian lengkap berdasarkan subset ruang *hyperparameter* yang ditentukan menggunakan nilai minimal *(lower bound*/batas bawah), nilai maksimal (*upper bound*/batas atas), dan jumlah angka (Syarif et al., 2016). *Grid Search* membagi jangkauan

parameter yang akan dioptimalkan ke dalam *grid* dan melintasi semua titik untuk mendapatkan parameter yang optimal. *Grid Search* mengoptimalkan parameter SVM menggunakan teknik *cross validation* sebagai metrik kinerja. Tujuannya adalah untuk mengidentifikasi kombinasi *hyperparameter* yang baik sehingga *classifier* dapat memprediksi data yang tidak diketahui secara akurat. Menurut Lin et al., (2008). Teknik *cross validation* dapat mencegah masalah overfitting.

Pada penelitian Syarif et al. (2016) *Grid Sreach* sebagai optimasi parameter terbukti dapat meningkatkan akurasi pada SVM dibanding dengan *Genetic Algorithm*. Pada penelitian Deshwal & Sharma (2019) diterapkannya *Grid Sreach* pada *dataset* kanker payudara dengan model SVM memberikan hasil yang jauh lebih baik dari pada tidak diterapkan *Grid Sreach*. Pada penelitian Eliana et al. (2019) *Finding Anomalies Around the Mean* (FAM) yang dikombinasikan dengan *Grid Search* mampu memberikan kinerja yang baik, menunjukkan ketelitian yang tinggi, dan mampu memberikan hasil akurasi yang besar yaitu sebesar 92,63%.

3.4.7 Analisa *Multioutput Classifier*

Pada kasus klasifikasi *multilabel* berbeda dengan multikelas dan klasifikasi biner yang hanya mengevaluasi satu kombinasi pada setiap label kelas. Pada *multilabel* klasifikasi akan menggunakan sebuah algortima khusus dan yang penulis gunakan pada penelitian ini yaitu *multiouput classifier.*

3.4.8 *Analisa Logistic Regression*

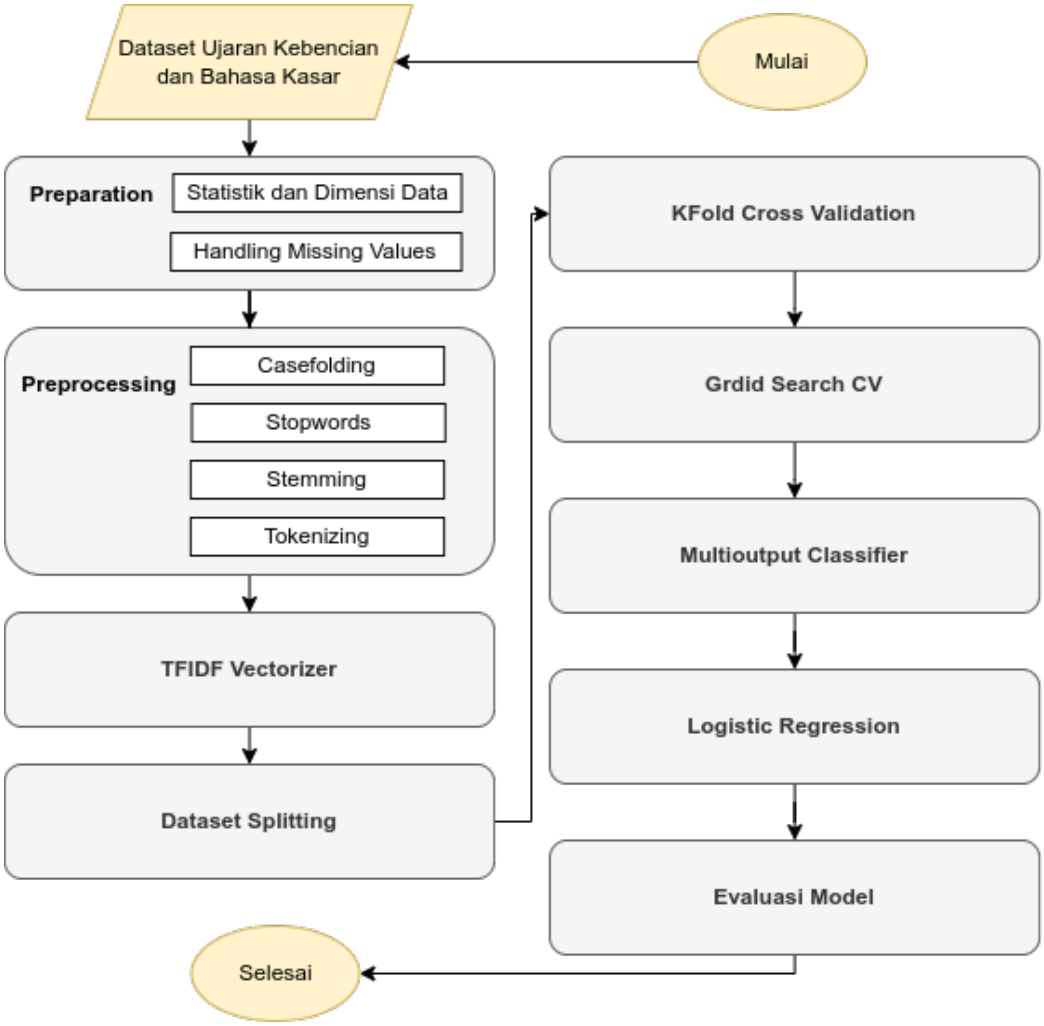
Pada tahap ini peneliti mencoba mengimplementasikan algoritma klasifikasi *logistic regression.* Mencari parameter terbaik dari algoritma dan menemukan akurasi maksimum.

3.4.9 Analisa Evaluasi Model

Evaluasi model adalah tahapan untuk menguji kinerja dari suatu mesin yang telah dibangun. Untuk menghitung akurasi dari model *machine learning* yang telah dibuat dan mengidentifikasi klasifikasi *Logistic Regression* menggunakan *precision* dan *recall.*

3.5 Perancangan Sistem

Gambar 3.2 merupakan diagram alir sistem secara keseluruhan. Proses diawali dengan *input* data ujaran kebencian dan bahasa kasar. Data tersebut kemudian masuk pada tahap *preparation* untuk dilakukan teknik *exploratory data analysis* agar siap masuk pada tahap *preprocessing*.



GAMBAR 3.2 Diagram Perancangan Sistem

Pada tahap ini sistem dirancang berdasarkan hasil analisa yang telah dilakukan pada proses sebelumnya dengan tujuan untuk memudahkan dalam pembuatan model yang terstruktur dengan baik. Dalam melakukan

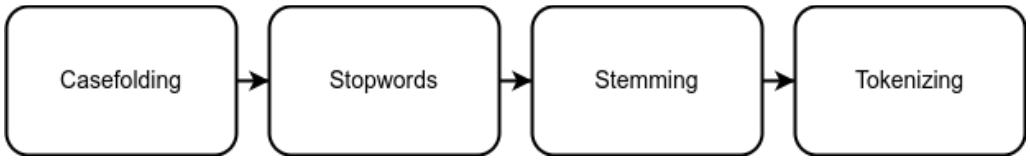
perancangan, peneliti menggunakan Jupyter notebook sebagai tempat untuk mengimplementasikan kode python. Dalam membuat *notebook* untuk proses perancangan model, tahapan-tahapan dalam perancangan dibuat ke dalam empat *notebook*. Dilakukannya pemisahan terhadap setiap proses bertujuan agar tahap demi tahap dalam pemodelan dapat terstruktur dan juga dapat mempermudah perubahan terhadap salah satu *notebook* apabila terjadi kesalahan dalam pengkodean. Dalam membangun sistem ini penulis menggunakan beberapa *library* utama dari python diantaranya numpy, pandas, scikit learn, dan matplotlib. Tahapan dalam merancang model pada penelitian ini terdiri dari empat tahap yaitu:

3.5.1 Perancangan *Data Preparation*

Pada tahap ini dilakukan persiapan data, mulai dari membaca data, kemudian melihat dimensi dan statistik data. Pada tahap ini pun dilakukan sebuah proses *exploratory data analysis* atau pemahaman menyeluruh terhadap data yang akan kita di proses. Kemudian mendeteksi apakah ada *missing values* dalam set data, dan mendeteksi nilai kosong karena akan mengganggu pada tahap pembangunan model *machine learning.*

3.5.2 Perancangan *Data Preprocessing*

Data Preprocessing merupakan tahap dimana teks akan diseragamkan bentuk dan format agar dapat dipersiapkan menjadi data yang dapat diolah pada tahap selanjutnya. *Data Preprocessing* meliputi *casefolding*, *tokenizing*, *stopwords* dan *stemming*. Gambaran tahapan *Data Preprocessing* dapat di lihat pada Gambar 3.3.



GAMBAR 3.3 Tahapan *Data Preprocessing*

a. *Casefolding*

Pada tahap ini, semua huruf akan diubah menjadi *lowercase* atau huruf kecil. Berikut adalah langkah-langkah *casefolding:*

1) Memeriksa ukuran setiap karakter dari awal sampai akhir karakter.
2) Jika ditemukan karakter yang menggunakan huruf kapital (*uppercase*), maka huruf tersebut akan diubah menjadi huruf kecil.

b. *Stopwords*

Kata-kata yang sering muncul secara umum dan kurang relevan dilakukan untuk mengubah kata berimbuhan dari setiap kata yang sudah disaring menjadi kata dasar dengan teks akan dihapus. Tahap ini akan menghapus kata-kata yang tidak bermakna dan tidak memiliki pengaruh terhadap analisis sentimen.

c. *Stemming*

Stemming adalah tahap mencari *root* (dasar) kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengambilan berbagai bentukan kata kedalam suatu representasi yang sama.

d. *Tokenizing*

Tokenizing dalam penelitian ini merupakan tahapan dalam memecah string atau input terhadap suatu teks berdasarkan tiap kata yang menyusunnya beserta menghilangkan delimeter seperti tanda titik (.), koma (,), spasi dan karakter angka dan tanda baca yang ada pada dokumen (*tweets*).

3.5.3 Perancangan TFIDF Vectorizer

Karena pada proses klasifikasi data yang bisa diproses oleh algoritma *machine learning* adalah data yang sudah di transformasi dalam bentuk

numerik, oleh karena itu dilakukan *language model training* menggunakan TFIDF. Pada proses ini data akan di ubah ke dalam setiap bentuk vektor kata.

3.5.4 Perancangan *Logistic Regression*

Proses pembangunan model pembelajaran mesin dilakukan menggunakan algortima *logistic regression,* algoritma yang cukup terkenal dikalangan *machine learning engineer* dan menjadi cikal bakal lahirnya algoritma *deep learning.*

3.5.5 Evaluasi Model

Pengujian adalah tahapan untuk menguji kinerja dari suatu mesin yang telah dibangun. Untuk menghitung akurasi dari model *machine learning* yang telah dibuat dan mengidentifikasi klasifikasi *Logistic Regression* menggunakan *precision, recall,* dan F1-Score.

Akurasi merupakan persentase dari total sentimen yang benar dikenali. Perhitungan akurasi dilakukan dengan cara membagi jumlah data sentimen yang benar dengan total data dan data uji. Untuk menghitung nilai akurasi digunakan persamaan dibawah ini:

$$\text{Akurasi} = \frac{\text{TP+TN}}{\text{TP+FN+FN+TN}} \text{ x } 100\ \%$$

Precision merupakan perbandingan jumlah data relevan yang ditemukan terhadap jumlah data yang ditemukan. Untuk menghitung nilai precision digunakan persamaan dibawah ini:

$$Precission = \frac{TP}{TP+FP}$$

Recall merupakan perbandingan jumlah materi relevan yang ditemukan terhadap jumlah materi yang relevan. Untuk menghitung nilai recall digunakan persamaan dibawah ini :

$$Recall = \frac{TP}{TP+FN}$$

F1-Score merupakan parameter tunggal ukuran keberhasilan retrieval yang menggabungkan recall dan precision. Untuk menghitung nilai F-measure dugunakan persamaan dibawah ini :

$$F1\text{-}Score = 2 * \frac{Precission*Recall}{Precission+Recall}$$

3.6 Implementasi

Implementasi adalah proses penerapan dari hasil perancangan ke dalam sebuah sistem. Proses ini membutuhkan perangkat pendukung berupa perangkas keras dan perangkat lunak.

3.6.1 Perangkat Lunak

Perangkat lunak yang akan digunakan dalam penelitian ini yaitu menggunakan Python. Python merupakan Bahasa pemrograman *open source* dan telah menduduki posisi 4 yang paling sering digunakan di seluruh dunia (Triasanti, 2000) dan memiliki *library* yang luas, dalam distibusi Python telah menyediakan modul-modul siap pakai untuk berbagai keperluan. IDE yang digunakan pada penelitian ini yaitu Jupyter Notebook, IDE ini mudah digunakan dan memiliki banyak *plugin* untuk memudahkan dalam pembangunan model *machine learning.*

TABEL 3.2 *Software* yang digunakan

Software
Operating System Linux 64 Bit (MX Linux Debian Base)
Python Programming
Jupyter Notebook
Pandas

Numpy
Matplotlib
Seaborn
Scikit-learn
NLTK
Wordcloud
Draw.io

3.6.2 Perangkat Keras

Perangkat keras yang akan digunakan dalam penelitian ini yaitu Laptop Lenovo Ideapad 320 dengan dengan spesifikasi sebagai berikut:

TABEL 3.3 *Hardware* yang digunakan

Hardware
Host: 80XU Lenovo ideapad 320-14AST
Kernel: 5.10.0-15-amd64
Shell: bash 5.1.4
Resolution: 1366x768
DE: lightdm-xsession
WM: Fluxbox
CPU: AMD A4-9120 RADEON R3 2C+2G (2) @ 2.200GHz
GPU: AMD ATI Radeon R2/R3/R4/R5 Graphics
Memory RAM: 4GiB

3.7 Kesimpulan Dan Saran

Bagian kesimpulan merupakan tahap penentuan kesimpulan terhadap hasil pengujian yang telah dilakukan. Hal tersebut bertujuan untuk mengetahui apakah penerapan algoritma yang telah dilakukan menggunakan metode *Logistic Regression* berhasil dan mengetahui tingkat akurasinya. Pada bagian saran berisi kemungkinan pengembangan yang dapat dilakukan terhadap penelitian ini.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Sumber Data

Data yang diperlukan pada penelitian ini adalah data twitter berupa *tweet* yang memiliki kemungkinan termasuk ke dalam *tweet* ujaran kebencian dan bahasa kasar. Proses pengumpulan data tidak dilakukan dengan cara memilih dan memberi label pada data *tweet* secara manual, karena pada penelitian sebelumnya yang dilakukan oleh (Ibrohim & Budi, 2019) tentang ujaran kebencian dan bahasa kasar telah melakukan *crawling* data *tweet* ujaran kebencian dan bahasa kasar.

(Ibrohim & Budi, 2019) Mempersilahkan Siapa Saja Yang Ingin Melakukan Penelitian Dengan Topik Yang Berhubungan Dan Memerlukan Dataset Tentang Ujaran Kebencian Dan Bahasa Kasar Dapat Menggunakan Dataset Yang Sama Secara Gratis (*free*). Oleh Karena Itu, Peneliti Meminta Izin Kepada (Ibrohim & Budi, 2019) Untuk Menggunakan Dataset Beliau Dengan Cara Menghubungi Melalui *Email*, Dan Beliau Mengizinkan Serta Memberikan Link Untuk Mengunduh Dataset Tersebut Melalui Email.

4.2 *Data Preparation*

Pada tahap ini dilakukan persiapan data, mulai dari mempersiapkan library yang dibutuhkan, membaca data dalam format .csv, kemudian melihat dimensi data, dan statistik data. Pada tahap ini pun dilakukan sebuah proses *exploratory data analysis* atau pemahaman menyeluruh terhadap data yang akan di proses. Bahasa pemrograman yang penulis implementasikan yaitu Python karena banyak digunakan dalam proses membangun arsitektur *machine learning* dan *data mining.*

GAMBAR 4.1 ***Flowcart Data Preparation***

4.2.1 *Import Library*

Library utama yang digunakan pada penelitian ini diantaranya adalah sebagai berikut:

a. Numpy, yaitu library yang akan digunakan untuk kebutuhan *scientific* dan matematika.
b. Pandas, yaitu *library* yang digunakan untuk manipulasi data seperti membuat tabel, mengubah dimensi data, mengecek data, dan sebagainya. Pandas mampu membaca berbagai format file seperti .txt, .csv, .tsv.
c. Matplotlib, yaitu *library* yang digunakan untuk membuat grafik plot sesuai kebutuhan.
d. Scikit-learn, yaitu *library* berbagai metode dan algoritma yang digunakan dalam *machine learning.*

4.2.2 *Load Dataset*

Pada bagian ini penulis akan memanggil *dataset* yang sudah disiapkan dengan nama file data.csv, Berikut adalah kode programnya:

```
path = '../dataset/data.csv'
dataset = pd.read_csv(path, encoding='latin-1')
```

Kode tersebut jika di eksekusi akan menghasilkan variabel baru yang dapat diproses yaitu:

a. Variabel path: berisi path atau folder data yang digunakan.
b. Variabel data: berisi keseluruhan data.

Jika ingin melihat variabel-variabel dalam data, bisa menggunakan perintah sebagai berikut:

```
dataset.keys()
```

Kode tersebut akan mengeluarkan output sebagai berikut:

```
Index(['Tweet', 'HS', 'Abusive', 'HS_Individual', 'HS_Group', 'HS_Religion',
'HS_Race', 'HS_Physical', 'HS_Gender', 'HS_Other', 'HS_Weak',
'HS_Moderate', 'HS_Strong'],
  dtype='object')
```

Data tersebut mempunyai 13 kolom data, atau jika ingin melihat *shape* dari data bisa dilakukan perintah sebagai berikut:

```
dataset.shape
```

Akan menghasilkan output (13169, 13) yang artinya *dataset* memiliki 13169 baris kalimat dan terdiri atas 13 kolom label.

Dataset yang diperlukan pada penelitian ini adalah data twitter berupa *tweet* yang termasuk ke dalam *tweet* ujaran kebencian dan bahasa kasar. Awalnya dataset yang diberikan oleh (Ibrohim & Budi, 2019) terdiri dari 4 label, dimana ada label untuk *hate speech*, label target *hate speech* yang akan dicapai, label level dari *hate speech*, dan label *abusive*. Namun pada penelitian ini label-label tersebut dibagi menjadi tiga label yaitu label *hate speech*, *abusive*, dan *neutral*. Untuk kelas target *hate speech* yang akan dicapai tidak digunakan dalam penelitian ini. Pada tahap ini peneliti akan melakukan proses pembuatan variabel baru dengan nama *neutral* yaitu kolom yang berisi kalimat yang dianggap bukan *hate* dan *abusive,* Berikut kode programnnya:

```
dataset = dataset.drop(['HS_Individual', 'HS_Group', 'HS_Religion',
'HS_Race', 'HS_Physical', 'HS_Gender', 'HS_Other','HS_Weak',
            'HS_Moderate', 'HS_Strong'], axis=1)
```

```
dataset['Neutral'] = (dataset['Abusive'] == 0) & (dataset['HS'] ==
0).astype(int)

dataset['Neutral'].replace({False: 0, True: 1}, inplace=True)
```

Pandas bisa digunakan untuk menampilkan sebagian isi dari dataset dengan menggunakan perintah sebagai berikut:

```
print(dataset.head(15))
```

Output dari kode tersebut akan menampilkan 15 baris data teratas.

TABEL 4.1 Contoh dari tabel data

Tweet	HS	Abusive	Neutral
- disaat semua cowok berusaha melacak perhatian gue. loe lantas remehkan perhatian yg gue kasih khusus ke elo. basic elo cowok bego ! ! !'	1	1	0
RT USER: USER siapa yang telat ngasih tau elu? edan sarap gue bergaul dengan cigax jifla calis sama siapa noh licew juga'	0	1	0
41. Kadang aku berfikir, kenapa aku tetap percaya pada Tuhan padahal aku selalu jatuh berkali-kali. Kadang aku merasa Tuhan itu ninggalkan aku sendirian. Ketika orangtuaku berencana berpisah, ketika kakakku lebih memilih jadi Kristen. Ketika aku anak ter	0	0	1
USER USER AKU ITU AKU\n\nKU TAU MATAMU SIPIT TAPI DILIAT DARI MANA ITU AKU'	0	0	1
USER USER Kaum cebong kapir udah keliatan dongoknya dari awal tambah dongok lagi hahahah'	1	1	0
USER Ya bani taplak dkk \xf0\x9f\x98\x84\xf0\x9f\ x98\x84\xf0\x9f\x98\x84'	1	1	0
deklarasi pilkada 2018 aman dan anti hoax warga dukuh sari jabon	0	0	1
Gue baru aja kelar re-watch Aldnoah Zero!!! paling kampret emang endingnya! 2 karakter utama cowonya kena friendzone bray! XD URL	0	1	0

Nah admin belanja satu lagi port terbaik nak makan Ais Kepal Milo, Ais Kepal Horlicks atau Cendol Toping kaw kaw. ð??; ; Doket mano tu ? Gerai Rojak Mertuaku - Taipan 2 (depan TWINS BABY & ROMANTIKA / Bank Islam Senawang) ð???	0	0	1
USER Enak lg klo smbil ngewe'	0	1	0
Setidaknya gw punya jari tengah buat lu, sebelom gw ukur nyali sama bacot lu \xf0\x9f\x98\x8f'	1	1	0
USER USER USER USER BANCI KALENG MALU GA BISA JAWAB PERTANYAAN KAMI DARI 2 HARI LALU.... NYUNGSEP KOE USER URL	1	1	0
Kalo belajar ekonomi mestinya jago memprivatisasi hati orang. Duh.. ironi USER	0	0	1
Aktor huruhara 98 Prabowo S ingin lengserkan pemerintahan Jokowi.... Nyata	1	0	0
USER Bu guru enakan jadi jablay atau guru esde sih.\nKayaknya menikmati jadi pecun ini guru.'	1	1	0

Didapatkan hasil akhir dari persiapan dataset dengan menggunakan library pandas sebanyak 13169 data tweet adalah 5860 *tweet* yang masuk dalam kelas Neutral, 3295 tweet kelas yang termsuk HS dan Abusive, 2266 *tweet* kelas HS, dan 1748 *tweet* termasuk kelas Abusive.

	HS	Abusive	Neutral	Tweet
0	0	0	1	5860
3	1	1	0	3295
2	1	0	0	2266
1	0	1	0	1748

GAMBAR 4.2 Distribusi kelas

4.2.4 *Handling Missing Values*

Sangat umum dalam kasus di dunia nyata terjadinya kehilangan data (*missing value*) atau data yang tidak lengkap. Penyebab kasus data seperti ini biasanya :

- Field kosong pada saat survey
- Terjadi data *corrupt*
- *Measurements not applicable*(?)

Data yang hilang biasanya diwakili dengan indikator Nan atau Null. Masalahnya adalah kebanyakan algoritma yang ada, tidak bisa menangani *missing value* sehingga kita harus menangani nilai-nilai seperti itu sebelum memasukkannya ke dalam model *machine learning.*
Ada beberapa teknik dalam menangani *missing value* :

a. Menghilangkan sample atau *feature* (bobot) yang memiliki banyak *missing data* . Tetapi beresiko ketika sample atau bobot memiliki banyak informasi yang relevan.
b. Mengganti data yang hilang dengan beberapa *pre-built estimator* (data penduga) seperti *Imputer Class* pada Scikit Learn. Kita akan sesuaikan data yang kosong dengan memperkirakan data yang kosong tersebut. Salah satu cara yang paling umum adalah dengan menggantinya dengan nilai rata-rata dari sisa sample atau *Feature* (bobot).

Library pandas dapat melihat apakah dalam dataset terdapat *missing value* atau tidak dengan menggunakan kode program sebagai berikut:

```
print(dataset.isna().sum())
```

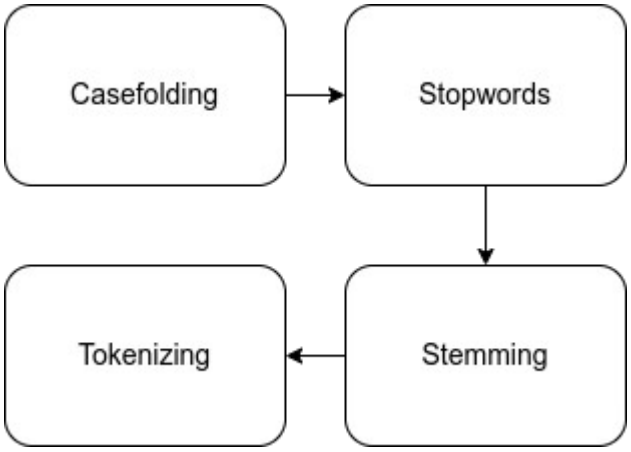
Jika program tersebut dieksekusi maka akan menghasilkan output sebagai berikut:

```
Null value :
Tweet       0
HS          0
Abusive     0
Neutral     0
dtype: int64
```

Terlihat semua kolom pada dataset memiliki angka nol yang artinya dataset pada penelitian ini sudah bersih dari *missing value* dan nilai *null*,

4.3 *Data Preprocessing*

Tahapan *preprocessing* perlu dilakukan karena beberapa kalimat *tweet* yang didapatkan tidak sepenuhnya menggunakan kata baku dan menggunakan bahasa indonesia yang baik. *Preprocessing* dilakukan menggunakan bantuan *library* pada bahasa pemrograman Python. *Preprocessing* data dilakukan dengan tahap *Casefolding*, *Stopwords Stemming*, dan *Tokenizing* sehingga menghasilkan data bersih dan siap untuk lanjut pada proses berikutnya. Berikut tahap *preprocessing* pada penelitian ini:



GAMBAR 4.2 Tahap*an Data Preprocessing*

4.3.1 *Casefolding*

Casefolding berperan dalam mengkonversi huruf-huruf yang berada didalam dokumen dengan format huruf besar (*uppercase*) menjadi huruf-huruf

kecil (*lowercase*). Hanya huruf ‘a’ sampai ‘z’ yang diterima. Pada tahap ini terdapat 3 fungsi, yaitu:

a. Konversi huruf besar menjadi huruf kecil.
b. Menghapus *unnecessary character*.
c. Menghapus *nonalphanumeric*.

Berikut kode programnya:

```
def lowercase(text):
    return text.lower()

def remove_unnecessary_char(text):
    text = re.sub('\n',' ',text)
    text = re.sub('rt',' ',text)
    text = re.sub('user',' ',text)
    text = re.sub('((www\.[^\s]+)|(https?://[^\s]+)|(http?://[^\s]+))',' ',text)
    text = re.sub('  +', ' ', text)
    return text

def remove_nonaplhanumeric(text):
    text = re.sub('[^0-9a-zA-Z]+', ' ', text)
    return text

def casefold(text):
    text = lowercase(text)
    text = remove_nonaplhanumeric(text)
    text = remove_unnecessary_char(text)
    return text

dataset['Casefolding'] = tqdm(dataset['Tweet'].apply(casefold))
```

Tabel 4.2 *Casefolding*

Tweet	Casefolding
- disaat semua cowok berusaha melacak perhatian gue. loe lantas remehkan perhatian yg gue kasih khusus ke elo. basic elo cowok bego ! ! !'	disaat semua cowok berusaha melacak perhatian gue loe lantas remehkan perhatian yg gue kasih khusus ke elo basic elo cowok bego
RT USER: USER siapa yang telat ngasih tau elu?edan sarap gue bergaul dengan cigax jifla calis sama siapa noh licew juga'	siapa yang telat ngasih tau elu edan sarap gue bergaul dengan cigax jifla calis sama siapa noh licew juga
41. Kadang aku berfikir, kenapa aku tetap percaya pada Tuhan padahal aku selalu	41 kadang aku berfikir kenapa aku tetap percaya pada tuhan padahal

jatuh berkali-kali. Kadang aku merasa Tuhan itu ninggalkan aku sendirian. Ketika orangtuaku berencana berpisah, ketika kakakku lebih memilih jadi Kristen. Ketika aku anak ter	aku selalu jatuh berkali kali kadang aku merasa tuhan itu ninggalkan aku sendirian ketika orangtuaku berencana berpisah ketika kakakku lebih memilih jadi kristen ketika aku anak ter
USER USER AKU ITU AKU\n\nKU TAU MATAMU SIPIT TAPI DILIAT DARI MANA ITU AKU'	aku itu aku n nku tau matamu sipit tapi diliat dari mana itu aku
USER USER Kaum cebong kapir udah keliatan dongoknya dari awal tambah dongok lagi hahahah'	kaum cebong kapir udah keliatan dongoknya dari awal tambah dongok lagi hahahah
USER Ya bani taplak dkk \xf0\x9f\x98\ x84\xf0\x9f\x98\x84\xf0\x9f\x98\x84'	ya bani taplak dkk xf0 x9f x98 x84 xf0 x9f x98 x84 xf0 x9f x98 x84
deklarasi pilkada 2018 aman dan anti hoax warga dukuh sari jabon	deklarasi pilkada 2018 aman dan anti hoax warga dukuh sari jabon
Gue baru aja kelar re-watch Aldnoah Zero!!! paling kampret emang endingnya! 2 karakter utama cowonya kena friendzone bray! XD URL	gue baru aja kelar re watch aldnoah zero paling kampret emang endingnya 2 karakter utama cowonya kena friendzone bray xd url
Nah admin belanja satu lagi port terbaik nak makan Ais Kepal Milo, Ais Kepal Horlicks atau Cendol Toping kaw kaw. ð??; ; Doket mano tu ? Gerai Rojak Mertuaku - Taipan 2 (depan TWINS BABY & ROMANTIKA / Bank Islam Senawang) ð???	nah admin belanja satu lagi po terbaik nak makan ais kepal milo ais kepal horlicks atau cendol toping kaw kaw doket mano tu gerai rojak me uaku taipan 2 depan twins baby amp romantika bank islam senawang
USER Enak lg klo smbil ngewe'	enak lg klo smbil ngewe
Setidaknya gw punya jari tengah buat lu, sebelom gw ukur nyali sama bacot lu \ xf0\x9f\x98\x8f'	setidaknya gw punya jari tengah buat lu sebelom gw ukur nyali sama bacot lu xf0 x9f x98 x8f
USER USER USER USER BANCI KALENG MALU GA BISA JAWAB PERTANYAAN KAMI DARI 2 HARI LALU.... NYUNGSEP KOE USER URL	banci kaleng malu ga bisa jawab pe anyaan kami dari 2 hari lalu nyungsep koe url
Kalo belajar ekonomi mestinya jago memprivatisasi hati orang. Duh.. ironi USER	kalo belajar ekonomi mestinya jago memprivatisasi hati orang duh ironi
Aktor huruhara 98 Prabowo S ingin lengserkan pemerintahan Jokowi....	aktor huruhara 98 prabowo s ingin lengserkan pemerintahan jokowi

Nyata	nyata
USER Bu guru enakan jadi jablay atau guru esde sih.\nKayaknya menikmati jadi pecun ini guru.'	bu guru enakan jadi jablay atau guru esde sih nkayaknya menikmati jadi pecun ini guru

Pada Tabel 4.2. kolom sebelah kiri berisikan teks yang akan diproses dengan *Casefolding*, sedangkan kolom sebelah kanan merupakan gambaran hasil dari proses *Casefolding*. Terlihat jelas dikolom sebelah kanan bahwa teks sudah bersih huruf besar.

4.3.2 *Stopwords*

Stopwords merupakan proses memilah data dengan cara mengambil kata-kata penting dari proses *preprocessing.* Pada tahap ini terdapat 2 fungsi *stopwords*, yaitu:

a. Normalisasi kata tidak baku menjadi baku.
b. Membung kata yang dianggap tidak penting pada penelitian ini.

Berikut kode program untuk melakukan teknik *stopwords:*

```
alay_dict_map = dict(zip(alay_dict['original'],
                         alay_dict['replacement']))

def normalize_alay(text):
    return ' '.join([alay_dict_map[word] if word in alay_dict_map else word for
word in text.split(' ')])

def remove_stopword(text):
    text = ' '.join(['' if word in id_stopword.stopword.values else word for word in
text.split(' ')])
    text = re.sub('  +', ' ', text)
    text = text.strip()
    return text

def stopword(text):
    text = normalize_alay(text)
    text = remove_stopword(text)
    return text
```

```
dataset['Stopwords'] = tqdm(dataset['Casefolding'].apply(stopword))
```

TABEL 4.3 ***Stopwords***

Casefolding	Stopwords
disaat semua cowok berusaha melacak perhatian gue loe lantas remehkan perhatian yg gue kasih khusus ke elo basic elo cowok bego	cowok berusaha melacak perhatian lantas remehkan perhatian kasih khusus basic cowok bego
siapa yang telat ngasih tau elu edan sarap gue bergaul dengan cigax jifla calis sama siapa noh licew juga	telat tau edan sarap bergaul licew
41 kadang aku berfikir kenapa aku tetap percaya pada tuhan padahal aku selalu jatuh berkali kali kadang aku merasa tuhan itu ninggalkan aku sendirian ketika orangtuaku berencana berpisah ketika kakakku lebih memilih jadi kristen ketika aku anak ter	41 kadang berpikir percaya tuhan jatuh berkali kali kadang tuhan meninggalkan orang tuaku berencana berpisah kakakku memilih kristen anak ter
aku itu aku n nku tau matamu sipit tapi diliat dari mana itu aku	ku tau matamu sipit
kaum cebong kapir udah keliatan dongoknya dari awal tambah dongok lagi hahahah	kaum cebong kafir dongoknya dungu haha
ya bani taplak dkk xf0 x9f x98 x84 xf0 x9f x98 x84 xf0 x9f x98 x84	bani taplak kawan kawan
deklarasi pilkada 2018 aman dan anti hoax warga dukuh sari jabon	deklarasi pilihan kepala daerah 2018 aman anti hoaks warga dukuh sari jabon
gue baru aja kelar re watch aldnoah zero paling kampret emang endingnya 2 karakter utama cowonya kena friendzone bray xd url	selesai re watch aldnoah zero kampret 2 karakter utama cowoknya kena friendzone bro xd
nah admin belanja satu lagi po terbaik nak makan ais kepal milo ais kepal horlicks atau cendol toping kaw kaw doket mano tu gerai rojak me uaku taipan 2 depan twins baby amp romantika bank islam senawang	admin belanja po terbaik nak makan ais kepal milo ais kepal horlicks cendol toping kau kau doket gerai rozak me uaku taipan 2 kembar baby amp romantika bank islam senawang
enak lg klo smbil ngewe	enak ngewe
setidaknya gw punya jari tengah buat lu	jari ukur nyali bacot

sebelom gw ukur nyali sama bacot lu xf0 x9f x98 x8f	
banci kaleng malu ga bisa jawab pe anyaan kami dari 2 hari lalu nyungsep koe url	banci kaleng malu pe anyaan 2 nyungsep koe
kalo belajar ekonomi mestinya jago memprivatisasi hati orang duh ironi	belajar ekonomi mestinya jago memprivatisasi hati orang aduh ironi
aktor huruhara 98 prabowo s ingin lengserkan pemerintahan jokowi nyata	aktor huru hara 98 prabowo si lengserkan pemerintahan jokowi nyata
bu guru enakan jadi jablay atau guru esde sih nkayaknya menikmati jadi pecun ini guru	bu guru enakan jablay guru sekolah dasar kayaknya menikmati pecun guru

Pada Tabel 4.3. kolom sebelah kiri berisikan teks yang akan diproses dengan *Stopwords Removal*, sedangkan kolom sebelah kanan merupakan gambaran hasil dari proses *Stopwords Removal*. Terlihat jelas dikolom sebelah kanan bahwa teks sudah bersih dari tanda baca.

4.3.3 *Stemming*

Stemming adalah langkah selanjutnya dari *preprocessing* untuk merubah setiap kata menjadi bentuk kata dasar. Pada penelitian ini menggunakan algoritma Nazief dan Adriani, yang kemudian diimplementasikan pada pemrograman python dengan menggunakan *library* Sastrawi.

Berikut kode program pada tahap *stemming:*

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stemming(text):
   return stemmer.stem(text)

dataset['Stemming'] = tqdm(dataset['Stopwords'].apply(stemming))
```

TABEL 4.4 *Stemming*

Stopwords	Stemming
cowok berusaha melacak perhatian lantas remehkan perhatian kasih khusus basic cowok bego	cowok usaha lacak perhati lantas remeh perhati kasih khusus basic cowok bego
telat tau edan sarap bergaul licew	telat tau edan sarap gaul licew
41 kadang berpikir percaya tuhan jatuh berkali kali kadang tuhan meninggalkan orang tuaku berencana berpisah kakakku memilih kristen anak ter	41 kadang pikir percaya tuhan jatuh kali kali kadang tuhan tinggal orang tua rencana pisah kakak pilih kristen anak ter
ku tau matamu sipit	ku tau mata sipit
kaum cebong kafir dongoknya dungu haha	kaum cebong kafir dongok dungu haha
bani taplak kawan kawan	bani taplak kawan kawan
deklarasi pilihan kepala daerah 2018 aman anti hoaks warga dukuh sari jabon	deklarasi pilih kepala daerah 2018 aman anti hoaks warga dukuh sari jabon
selesai re watch aldnoah zero kampret 2 karakter utama cowoknya kena friendzone bro xd	selesai re watch aldnoah zero kampret 2 karakter utama cowok kena friendzone bro xd
admin belanja po terbaik nak makan ais kepal milo ais kepal horlicks cendol toping kau kau doket gerai rozak me uaku taipan 2 kembar baby amp romantika bank islam senawang	admin belanja po baik nak makan ais kepal milo ais kepal horlicks cendol toping kau kau doket gerai rozak me uaku taipan 2 kembar baby amp romantika bank islam senawang
enak ngewe	enak ngewe
jari ukur nyali bacot	jari ukur nyali bacot
banci kaleng malu pe anyaan 2 nyungsep koe	banci kaleng malu pe anyaan 2 nyungsep koe
belajar ekonomi mestinya jago memprivatisasi hati orang aduh ironi	ajar ekonomi mesti jago privatisasi hati orang aduh ironi
aktor huru hara 98 prabowo si lengserkan pemerintahan jokowi nyata	aktor huru hara 98 prabowo si lengser perintah jokowi nyata
bu guru enakan jablay guru sekolah dasar kayaknya menikmati pecun guru	bu guru enak jablay guru sekolah dasar kayak nikmat pecun guru

Pada Tabel 4.4. kolom sebelah kiri berisikan teks yang akan diproses dengan *Stemming*, sedangkan kolom sebelah kanan merupakan gambaran hasil dari proses *Stemming*. Terlihat jelas dikolom sebelah kanan bahwa masing-masing kata sudah diubah kedalam bentuk dasar kata tersebut.

4.3.4 *Tokenizing*

Tokenisasi dilakukan untuk memisahkan setiap kata yang ada pada teks yang menyusun sebuah dokumen. Proses tokenizing pada penelitian ini menggunakan library NLTK dari python.

Berikut kode program pada tahap *tokenizing:*

```
dataset["Tokenizing"] = [word_tokenize(Tweet.lower()) for Tweet in
tqdm(dataset.Stemming.astype(str))]
```

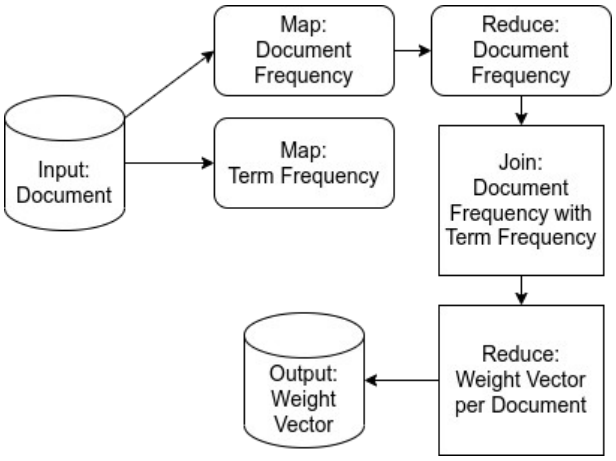
TABEL 4.5 *Tokenizing*

Stemming	Tokenizing
cowok usaha lacak perhati lantas remeh perhati kasih khusus basic cowok bego	['cowok', 'usaha', 'lacak', 'perhati', 'lantas', 'remeh', 'perhati', 'kasih', 'khusus', 'basic', 'cowok', 'bego']
telat tau edan sarap gaul licew	['telat', 'tau', 'edan', 'sarap', 'gaul', 'licew']
41 kadang pikir percaya tuhan jatuh kali kali kadang tuhan tinggal orang tua rencana pisah kakak pilih kristen anak ter	['41', 'kadang', 'pikir', 'percaya', 'tuhan', 'jatuh', 'kali', 'kali', 'kadang', 'tuhan', 'tinggal', 'orang', 'tua', 'rencana', 'pisah', 'kakak', 'pilih', 'kristen', 'anak', 'ter']
ku tau mata sipit	['ku', 'tau', 'mata', 'sipit']
kaum cebong kafir dongok dungu haha	['kaum', 'cebong', 'kafir', 'dongok', 'dungu', 'haha']
bani taplak kawan kawan	['bani', 'taplak', 'kawan', 'kawan']
deklarasi pilih kepala daerah 2018 aman anti hoaks warga dukuh sari jabon	['deklarasi', 'pilih', 'kepala', 'daerah', '2018', 'aman', 'anti', 'hoaks', 'warga', 'dukuh', 'sari', 'jabon']
selesai re watch aldnoah zero kampret 2 karakter utama cowok	['selesai', 're', 'watch', 'aldnoah', 'zero', 'kampret', '2', 'karakter', 'utama',

kena friendzone bro xd	'cowok', 'kena', 'friendzone', 'bro', 'xd']
admin belanja po baik nak makan ais kepal milo ais kepal horlicks cendol toping kau kau doket gerai rozak me uaku taipan 2 kembar baby amp romantika bank islam senawang	['admin', 'belanja', 'po', 'baik', 'nak', 'makan', 'ais', 'kepal', 'milo', 'ais', 'kepal', 'horlicks', 'cendol', 'toping', 'kau', 'kau', 'doket', 'gerai', 'rozak', 'me', 'uaku', 'taipan', '2', 'kembar', 'baby', 'amp', 'romantika', 'bank', 'islam', 'senawang']
enak ngewe	['enak', 'ngewe']
jari ukur nyali bacot	['jari', 'ukur', 'nyali', 'bacot']
banci kaleng malu pe anyaan 2 nyungsep koe	['banci', 'kaleng', 'malu', 'pe', 'anyaan', '2', 'nyungsep', 'koe']
ajar ekonomi mesti jago privatisasi hati orang aduh ironi	['ajar', 'ekonomi', 'mesti', 'jago', 'privatisasi', 'hati', 'orang', 'aduh', 'ironi']
aktor huru hara 98 prabowo si lengser perintah jokowi nyata	['aktor', 'huru', 'hara', '98', 'prabowo', 'si', 'lengser', 'perintah', 'jokowi', 'nyata']
bu guru enak jablay guru sekolah dasar kayak nikmat pecun guru	['bu', 'guru', 'enak', 'jablay', 'guru', 'sekolah', 'dasar', 'kayak', 'nikmat', 'pecun', 'guru']

Pada Tabel 4.5 kolom sebelah kiri berisikan teks yang akan diproses dengan *Tokenize*, sedangkan kolom sebelah kanan merupakan gambaran hasil dari proses *Tokenize*. Terlihat jelas dikolom sebelah kanan bahwa teks sudah terpisah menjadi satuan kata.

4.4 *Term Frequency-Inverse Document Frequency*



GAMBAR 4.3 Diagram alir TFIDF

Setelah melewati tahap *preprocessing*, tahap selanjutnya yang akan dilakulan adalah pemberian bobot pada setiap kata yang ada pada tweet dengan menggunakan TFIDF. *Term Frequency* (TF) digunakan untuk menghitung seberapa sering sebuah kata muncul pada tweet dan *Inverse Document Frequency* (IDF) untuk pemberian bobot pada kata tertentu yang banyak terkandung di dalam dokumen.

TABEL 4.6 TFIDF

Kode	Tweet	HS	Abusive	Neutral
D1	prabowo kalah sebut bantu jokowi citra ratap pilu	1	0	0
D2	takut azan iblis	0	0	1
D3	goblok bani cebong tukang tipu jilat kuasa tahu gerak bayar pakai nasi bungkus propaganda nasi bungkus gagal	1	1	0
D4	cebong sewot	1	1	0
D5	2 gerak tekan kerja keras total tingkat potensi bangsa	0	0	1

Data yang sudah melewati tahap *preprocessing* melakukan proses TFIDF. Berikut merupakan *flowchart* proses TF-IDF untuk mendapatkan nilai dari bobot setiap kata.

TABEL 4.7 Perhitungan TF

Term	D1	D2	D3	D4	D5
prabowo	1	0	0	0	0
kalah	1	0	0	0	0
sebut	1	0	0	0	0
bantu	1	0	0	0	0
jokowi	1	0	0	0	0
citra	1	0	0	0	0
ratap	1	0	0	0	0
pilu	1	0	0	0	0
takut	0	1	0	0	0
azan	0	1	0	0	0
iblis	0	1	0	0	0
goblok	0	0	1	0	0
bani	0	0	1	0	0
cebong	0	0	1	1	0
tukang	0	0	1	0	0
tipu	0	0	1	0	0
jilat	0	0	1	0	0
kuasa	0	0	1	0	0
tahu	0	0	1	0	0
gerak	0	0	1	0	1
bayar	0	0	1	0	0
pakai	0	0	1	0	0
nasi	0	0	1	0	0
bungkus	0	0	1	0	0
propaganda	0	0	1	0	0
gagal	0	0	1	0	0

sewot	0	0	0	1	0
2	0	0	0	0	1
tekan	0	0	0	0	1
kerja	0	0	0	0	1
keras	0	0	0	0	1
total	0	0	0	0	1
tingkat	0	0	0	0	1
potensi	0	0	0	0	1
bangsa	0	0	0	0	1

Setelah didapatkan nilai TF dari masing-masing *tweet,* maka proses selanjutnya adalah menghitung nilai DF dari setiap kata pada *tweet.*

TABEL 4.8 Perhitungan DF

Term	**D1**	**D2**	**D3**	**D4**	**D5**	**DF**
prabowo	1	0	0	0	0	1
kalah	1	0	0	0	0	1
sebut	1	0	0	0	0	1
bantu	1	0	0	0	0	1
jokowi	1	0	0	0	0	1
citra	1	0	0	0	0	1
ratap	1	0	0	0	0	1
pilu	1	0	0	0	0	1
takut	0	1	0	0	0	1
azan	0	1	0	0	0	1
iblis	0	1	0	0	0	1
goblok	0	0	1	0	0	1
bani	0	0	1	0	0	1
cebong	0	0	1	1	0	2
tukang	0	0	1	0	0	1
tipu	0	0	1	0	0	1
jilat	0	0	1	0	0	1
kuasa	0	0	1	0	0	1

tahu	0	0	1	0	0	1
gerak	0	0	1	0	1	2
bayar	0	0	1	0	0	1
pakai	0	0	1	0	0	1
nasi	0	0	1	0	0	1
bungkus	0	0	1	0	0	1
propaganda	0	0	1	0	0	1
gagal	0	0	1	0	0	1
sewot	0	0	0	1	0	1
2	0	0	0	0	1	1
tekan	0	0	0	0	1	1
kerja	0	0	0	0	1	1
keras	0	0	0	0	1	1
total	0	0	0	0	1	1
tingkat	0	0	0	0	1	1
potensi	0	0	0	0	1	1
bangsa	0	0	0	0	1	1

Setelah mendapatkan nilai DF dari masing-masing kata tahap selanjutnya adalah perhitungan untuk mendapatkan nilai IDF.

TABEL 4.9 Perhitungan IDF

Term	D1	D2	D3	D4	D5	DF	D/DF	IDF
prabowo	1	0	0	0	0	1	5	0.698970
kalah	1	0	0	0	0	1	5	0.698970
sebut	1	0	0	0	0	1	5	0.698970
bantu	1	0	0	0	0	1	5	0.698970
jokowi	1	0	0	0	0	1	5	0.698970
citra	1	0	0	0	0	1	5	0.698970
ratap	1	0	0	0	0	1	5	0.698970
pilu	1	0	0	0	0	1	5	0.698970
takut	0	1	0	0	0	1	5	0.698970
azan	0	1	0	0	0	1	5	0.698970

iblis	0	1	0	0	0	1	5	0.698970
goblok	0	0	1	0	0	1	5	0.698970
bani	0	0	1	0	0	1	5	0.698970
cebong	0	0	1	1	0	2	2.5	0.397940
tukang	0	0	1	0	0	1	5	0.698970
tipu	0	0	1	0	0	1	5	0.698970
jilat	0	0	1	0	0	1	5	0.698970
kuasa	0	0	1	0	0	1	5	0.698970
tahu	0	0	1	0	0	1	5	0.698970
gerak	0	0	1	0	1	2	2.5	0.397940
bayar	0	0	1	0	0	1	5	0.698970
pakai	0	0	1	0	0	1	5	0.698970
nasi	0	0	2	0	0	1	5	0.698970
bungkus	0	0	2	0	0	1	5	0.698970
propaganda	0	0	1	0	0	1	5	0.698970
gagal	0	0	1	0	0	1	5	0.698970
sewot	0	0	0	1	0	1	5	0.698970
2	0	0	0	0	1	1	5	0.698970
tekan	0	0	0	0	1	1	5	0.698970
kerja	0	0	0	0	1	1	5	0.698970
keras	0	0	0	0	1	1	5	0.698970
total	0	0	0	0	1	1	5	0.698970
tingkat	0	0	0	0	1	1	5	0.698970
potensi	0	0	0	0	1	1	5	0.698970
bangsa	0	0	0	0	1	1	5	0.698970

Selanjutnya menghitung nilai dari TFIDF.

TABEL 4.10 Perhitungan TFIDF

Term	D1	D2	D3	D4	D5
prabowo	0.698970	0	0	0	0
kalah	0.698970	0	0	0	0
sebut	0.698970	0	0	0	0

bantu	0.698970	0	0	0	0
jokowi	0.698970	0	0	0	0
citra	0.698970	0	0	0	0
ratap	0.698970	0	0	0	0
pilu	0.698970	0	0	0	0
takut	0	0.698970	0	0	0
azan	0	0.698970	0	0	0
iblis	0	0.698970	0	0	0
goblok	0	0	0.698970	0	0
bani	0	0	0.698970	0	0
cebong	0	0	0.397940	0.397940	0
tukang	0	0	0.698970	0	0
tipu	0	0	0.698970	0	0
jilat	0	0	0.698970	0	0
kuasa	0	0	0.698970	0	0
tahu	0	0	0.698970	0	0
gerak	0	0	0.397940	0	0.397940
bayar	0	0	0.698970	0	0
pakai	0	0	0.698970	0	0
nasi	0	0	1.397940	0	0
bungkus	0	0	1.397940	0	0
propaganda	0	0	0.698970	0	0
gagal	0	0	0.698970	0	0
sewot	0	0	0	0.698970	0
2	0	0	0	0	0.698970
tekan	0	0	0	0	0.698970
kerja	0	0	0	0	0.698970
keras	0	0	0	0	0.698970
total	0	0	0	0	0.698970
tingkat	0	0	0	0	0.698970
potensi	0	0	0	0	0.698970
bangsa	0	0	0	0	0.698970

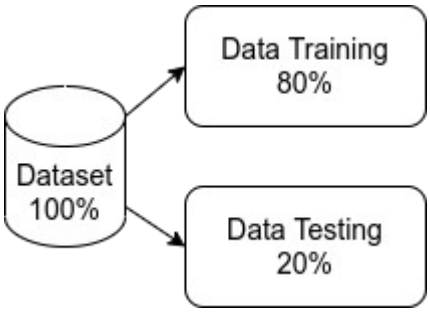
Tujuan dari pembobotan ini adalah untuk melakukan penyaringan kata yang memenuhi kriteria *threshold.* Pada penelitian ini akan digunakan ambang batas dengan rentang nilai range 0 sampai 1 (Chamidah & Kunci, 2012). Agar bisa membawa range nilai hasil pembobotan kata TFIDF sesuai dengan nilai hasil *threshold,* maka digunakan metode normalisasi *min-max scaler.*

Pada penelitian ini penulis tidak melakukan tahapan perhitungan TFIDF secara manual, namun menggunakan teknologi *library scikit-learn* dari bahasa pemrograman python, dengan memanggil modul sebagai berikut:

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(max_features=5000, ngram_range=(0,1))
X = tfidf.fit_transform(data['Tweet'].values.astype('U'))
```

Buat variabel tfidf sebagai *instance* dari *class* TfidfVectozier, kemudian masukan parameter max_features sebanyak 5000 maka akan melakukan *generate* sebanyak 5000 kolom vektor TFIDF, masukan parameter rentang kata dari 1 sampai 4 kata per vektor.

4.5 Dataset Splitting



GAMBAR 4.3 Proses *data splitting*

Pada bagian ini digunakan untuk membagi dataset menjadi dua bagian yaitu *data training* dan *data testing. Data training* akan digunakan sebagai data latih, yaitu menggunakan 80% dari data. Sementara *data testing* digunakan

untuk data uji dan dalam penelitian ini menggunakan 20% dari data. Berikut adalah kode programnya:

```
from sklearn.model_selection import train_test_split

X = vector_w2v.wv
y = data[['HS', 'Abusive', 'Neutral']]
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    stratify=y,
                                                    shuffle=True,
                                                    test_size=0.2,
                                                    random_state=42)
```

Kode diatas jika di eksekusi akan menghasilkan 2 variabel tambahan yaitu variabel X akan diisi dengan vektor TFIDF yang sudah dilakukan perhitungan pada tahap sebelumnya, dan variabel y diisi dengan kolom label pada dataset. Sementara 4 variabel tambahan lainnya yaitu X_train, X_test, y_train, dan y_test yang akan diproses dalam *training* dan *testing* model

.

Dengan dilakukannya *dataset splitting* dengan 20% data uji dan 80% data latih maka akan menghasilkan dimensi data sebagai berikut:

1. X_train = 10535 baris data dan 5000 kolom data.
2. X_test = 2634 baris data dan 5000 kolom data.
3. y_train = 10535 baris data dan 3 kolom data.
4. y_test = 2634 baris data dan 3 kolom data.

4.6 K-Fold Cross Validation

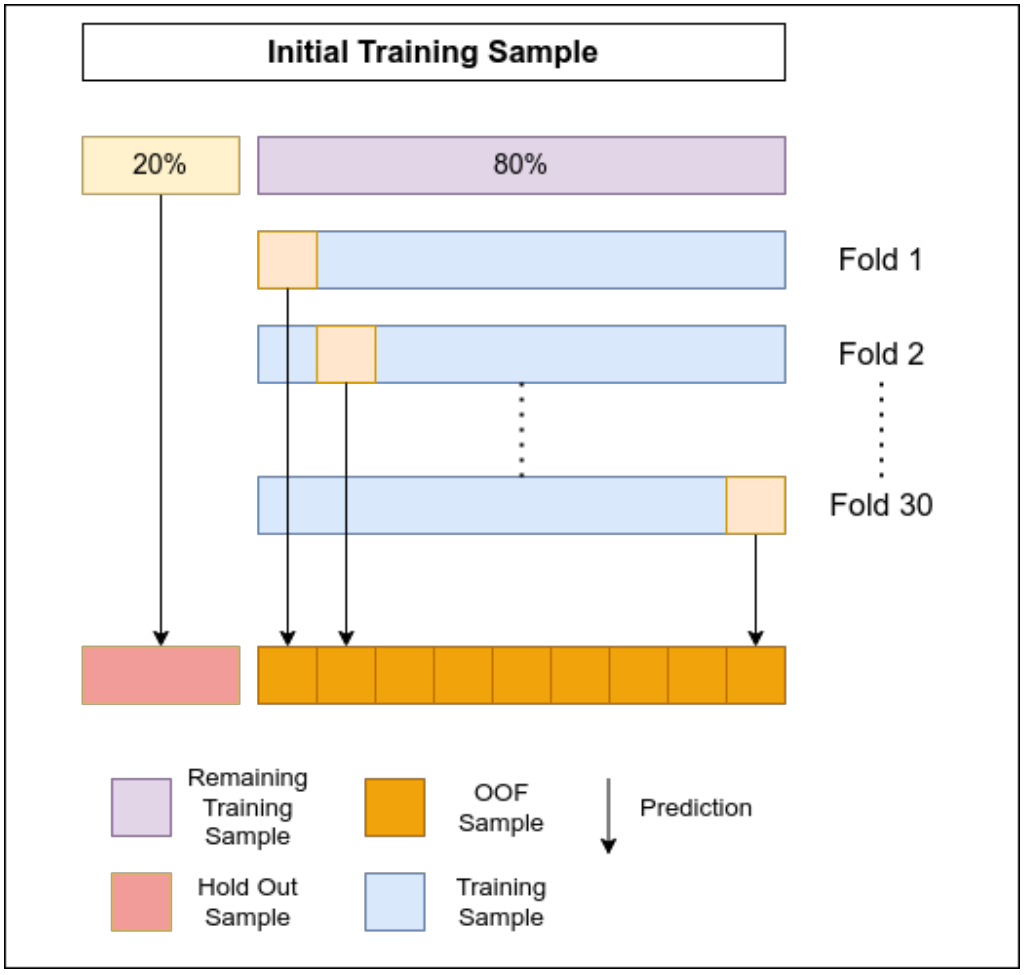
Pada pendekatan ini, setiap data digunakan dalam jumlah k yang sama untuk pelatihan dan tepat satu kali untuk pengujian. Setiap kali berjalan, satu

pecahan berperan sebagai *training set* sedangkan pecahan lainnya menjadi *training set.* Prosedur tersebut dilakukan sebanyak k kali sehingga setiap data berkesempatan menjadi data uji tepat satu kali dan menjadi data latih sebanyak k-1 kali. Total *error* didapatkan dengan menjumlahkan semua *error* yang didapatkan dari k kali proses.

Pada penelitian ini menggunakan k sebanyak 30 folds dengan random state sebanyak 42 state. Berikut kode programnya:

```
num_folds = 20
seed = 42
kfold = Kfold(n_splits=num_folds, shuffle=True, random_state=seed)
```

Buat variabel *instance* dari modul Kfold dengan nama kfold, kemudian masukan parameter n_splits dengan jumlah dari num_folds yaitu 42 folds.

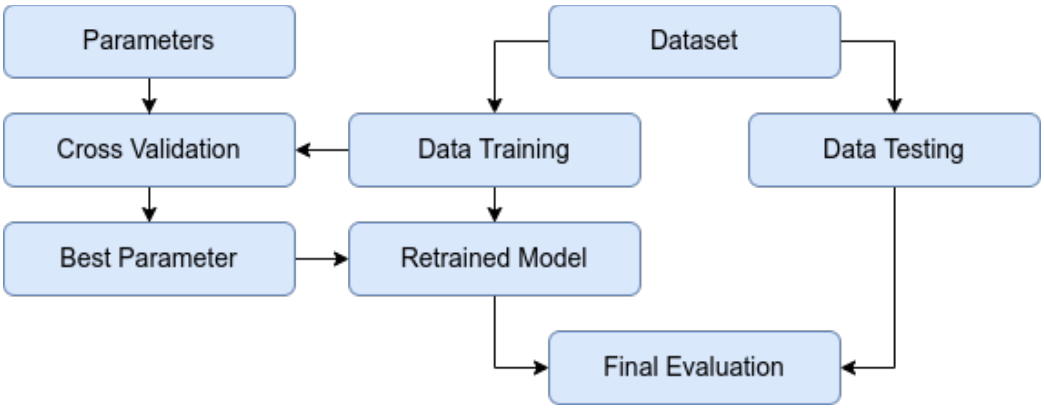


GAMBAR 4.3 Ilustrasi 30-Fold CV

4.7 *GridSearchCV*

GridSearchCV merupakan teknik pencarian parameter terbaik dalam sebuah pemodelan *machine learning*. Teknik ini akan mengevaluasi berdasarkan subset *hyperparameter* yang telah ditentukan sebelumnya secara terpenuhi akan memilih sejumlah pasangan hyperparametrik yang dipilih dari parameter yang diberikan, dan hanya menguji yang dipilih. *GridSearchCV* cenderung lebih mudah secara komputasi dan tidak memakan waktu karena tidak mengevaluasi setiap kemungkinan kombinasi hiperparametrik. Metode ini sangat menyederhanakan analisis tanpa mengorbankan optimasi secara signifikan. *GridSearchCV* seringkali merupakan pilihan yang sangat baik

untuk data dimensi tinggi karena mengembalikan kombinasi hiperparametrik yang baik dengan sangat cepat.



GAMBAR 4.3 Ilustrasi *GridSearchCV*

Dataset yang sudah dibagi dua menjadi data uji dan data latih, data latih akan masuk pada tahap cross validation, kemudian parameter yang kita tentukan akan masuk pada tahap cross validation yang kemudian akan didapat parameter terbaik dari seluruh kombinasi parameter yang ada.

List parameter yang akan diuji pada penelitian menggunakan *logistic regression* yaitu:

1. Penalty = l2,
2. C = log dari 4 sampai 20, dengan rentang angka 4
3. Solver = lbfgs, liblinear

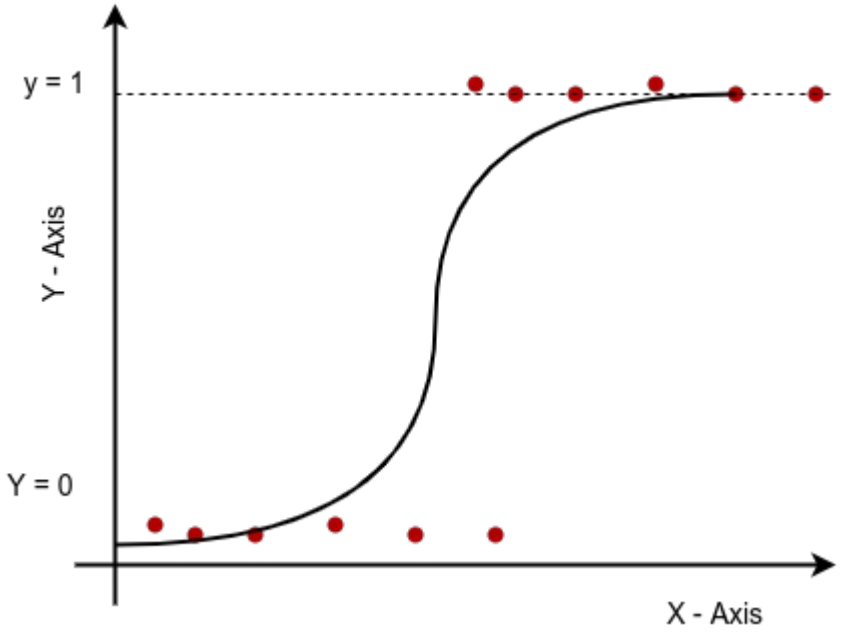
Berikut kode program pada *GridSearchCV* :

```
penalty = ['l2']
C = np.logspace(0, 2, 10)
solver = ['lbfgs', 'liblinear']

hyperparameters = dict(penalty = penalty, C = C, solver = solver)

model_tuned = GridSearchCV(model_logreg,
                           hyperparameters,
                           cv = kfold,
                           n_jobs = -1,
```

verbose = 1)

4.8 *Logistic Regression*

Pembelajaran pada model *logistic regression* dilakukan menggunakan data latih, pada penelitian ini digunakan regresi logistik multinomial merupakan regresi logistik yang digunakan saat variabel dependen mempunyai skala yang bersifat *polichotomous* atau multinomial. Skala multinomial adalah suatu pengukuran yang dikategorikan menjadi lebih dari dua kategori. Metode yang digunakan dalam penelitian ini adalah regresi logistik dengan variabel dependen berskala nominal dengan 3 kategori, yaitu kategori *hate speech, abusive,* dan *neutral.*



GAMBAR 4.4 Ilustrasi dari *Logistic Regression*

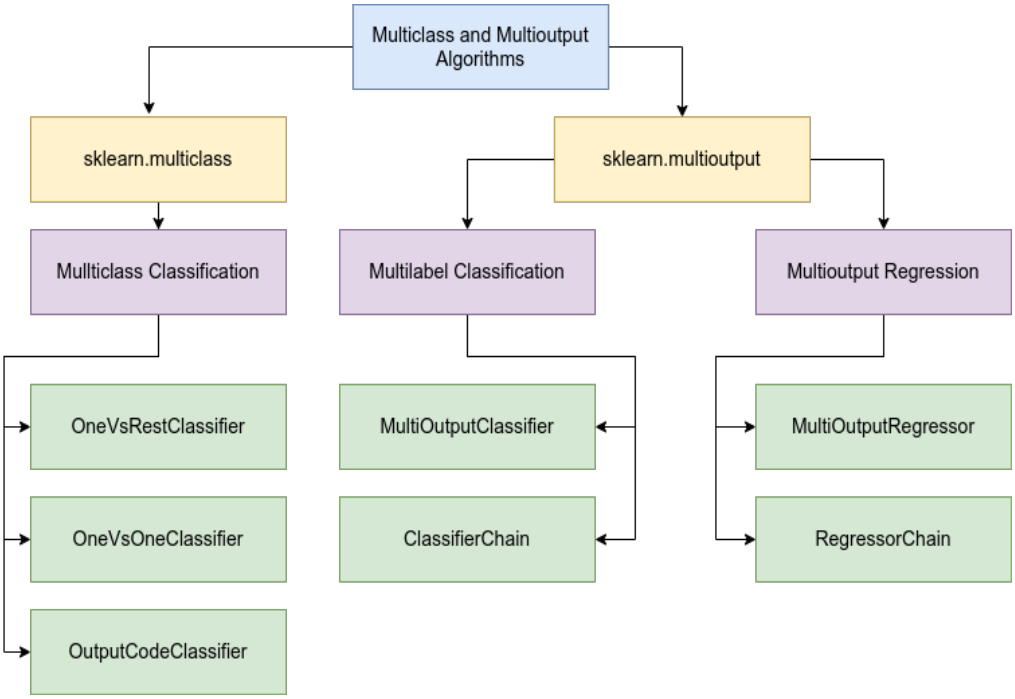
Pada penelitian ini penulis menggunakan bahasa pemrograman python untuk mengimplementasikan algoritma *logistic regression* pada dataset.

Pertama buat variabel untuk menampung *class* dari modul Scikit-learn dengan nama model_logreg, Berikut kode program untuk memanggil algoritma *logistic regression:*

```
model_logreg = LogisticRegression(max_iter=5000)
```

4.9 *MultiOutput Classifier*

Dalam *machine learning*, *multilabel classification* adalah masalah dalam contoh klasifikasi ke dalam satu dari tiga atau banyak kelas (Klasifikasi *instance* ke dalam salah satu dari dua kelas disebut klasifikasi biner). Walaupun beberapa algoritma klasifikasi biasanya mengizinkan penggunaan lebih dari dua kelas (ada yang menggunakan algoritma biner), namun hal ini dapat diubah menjadi *classifier multinomial* dengan berbagai strategi.



GAMBAR 4.5 Ilustrasi *MultiOutputClassifier*

Ada teknik khusus dalam menangani *multilabel Classification,* yaitu dengan menggunakan modul scikit-learn. Berikut kode programnya:

```
clf = MultiOutputClassifier(grid).fit(X_train, y_train)
```

Panggil *class* MultiOutputClassifier kemudian masuka parameter dari GridSearchCV yang sudah ditentukan. *Classifier* akan melakukan proses *training* model.

4.10 Evaluasi Model

Proses klasifikasi data ujaran kebencian dan bahasa kasar berbahasa Indonesia menggunakan pendeketan *machine learning*. Algoritma yang digunakan adalah *Logistic Regression*. Untuk memudahkan percobaan, penulis menggunakan library scikit-learn dengan bahasa pemrograman Python. Evaluasi algoritma menggunakan metode *10-fold cross validation* dan mencari nilai akurasi dari hasil evaluasi pada penelitian ini.

Berdasarkan hasil pada pengujian model klasifikasi *Logistic Regression* menghasilkan nilai akurasi pada keseluruhan sistem dapat dihitung sebesar 73%. Untuk menghitung nilai akurasi dapat dilihat pada kode program dibawah berikut:

```
score_lr = accuracy_score(y_pred, y_test)
print(score_lr)
```

.

Akurasi menggambarkan seberapa besar tingkat akurat model yang telah dibuat dapat mengklasifikasi data dengan benar. Akurasi didapatkan dari perhitungan rasio prediksi benar dengan keseluruhan data. Dengan mengetahui besarnya nilai akurasi pada kinerja keseluruhan sistem dapat dinyatakan tingkat kemampuan sistem dalam mencari ketepatan antara informasi yang diinginkan pengguna dengan jawaban yang diberikan sistem. Tingkat keberhasilan sistem dalam menemukan sebuah informasi dalam penelitian ini sebesar 73%. Selanjutnya untuk melihat performa klasifikasi dari setiap kelas dapat diketahui

melalui nilai *precision*, *recall* dan f1 score pada setiap kelas klasifikasi. *Precision* menggambarkan tingkat keakuratan data yang diminta dengan hasil yang diberikan oleh model. *Precision* didapatkan dari perhitungan rasio prediksi benar dibandingkan dengan keseluruhan hasil yang diprediksi positif. *Recall* menggambarkan keberhasilan model dalam menemukan kembali informasi yang dimasukkan dalam pengujian. *Recall* didapatkan dari hasil perhitungan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. F1-Score merupakan parameter tunggal ukuran keberhasilan *retrieval* yang menggabungkan *recall* dan *precision*. Hasil nilai *precision*, *recall*, dan f1-score memiliki nilai sebesar 0-1. Semakin tinggi nilai maka semakin baik hasil model yang dibuat. Nilai akurasi yang tinggi didapat ketika banyak data yang berhasil diklasifikasi dengan benar sesuai kelas sentimennya. Dapat diketahui juga nilai *Precision* dan *Recall*. Nilai *Precision* mengikuti nilai akurasi, semakin tinggi nilai akurasi maka akan diikuti nilai *Precision* yang tinggi juga, begitu sebaliknya. Nilai *Precision* adalah jumlah data positif yang benar diklasifikasi sebagai data positif dibagi total data yang diklasifikasi sebagai data positif. Sedangkan Nilai *recall* adalah jumlah data positif yang benar diklasifikasi sebagai data positif dibagi jumlah data positif sebenarnya. Perbandingan nilai presisi,recall dan F1-Score dapat dilihat pada Tabel 4.11.

TABEL 4.11 Nilai *Precision, Recall, dan F-1 Score* Evaluasi Model

Jenis Klasifikasi	Precision	Recall	F1-Score
Hate Speech	0.86	0.71	0.78
Abusive	0.91	0,81	0.86
Neutral	0.87	0.83	0.85

Hasil dari evaluasi model dapat dilihat bahwa nilai *Precision* dan *recall* disetiap kelas dapat dikatakan memiliki tingkat kemampuan yang tinggi dalam mencari ketepatan antara informasi yang diminta oleh pengguna. Nilai *precision* untuk kelas Hate Speech sebesar 86%, untuk kelas Neutral sebesar

87%, untuk kelas Abusive sebesar 91%. Angka ini dapat diartikan bahwa proporsi label yang diprediksi dengan benar dari total prediksi cukup tinggi untuk semua kelas. Sedangkan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi untuk kelas Hate Speech sebesar 71%, untuk kelas Neutral sebesar 81% dan kelas Abusive sebesar 83%. Hal ini berarti kinerja keberhasilan sistem dalam menemukan kembali informasi yang bernilai Hate Speech dalam dokumen rendah dibandingkan dengan menemukan informasi kembali yang bernilai Abusive dan Neutral.

4.11 Hasil Dan Visualisasi Klasifikasi Sentimen

Setelah dilakukan proses pembersihan data dan klasifikasi selanjutnya didapatkan hasil sentimen. Wordcloud adalah bentuk visualisasi dari data teks yang menggambarkan kumpulan kata yang banyak terdapat dalam sebuah analisis teks. Wordcloud dibuat dengan menggunakan library Wordcloud dan PIL (Python Imaging Library) yaitu pustaka tambahan gratis dan sumber terbuka untuk bahasa pemrograman Python yang menambahkan dukungan untuk membuka, memanipulasi, dan menyimpan banyak format file gambar yang berbeda. Script Python untuk membuat wordcloud neutral dapat dilihat pada kode program sebagai berikut:

```
# wordcloud neutral

wordcloud_neutral = WordCloud(width=800,
                              height=500,
                              background_color="white",
                              colormap="Dark2",
                              random_state=21,
                              max_font_size=110).generate(neutral)
plt.figure(figsize=(14, 12))
plt.imshow(wordcloud_neutral, interpolation="bilinear")
plt.axis('off')
plt.show()
```

Gambar 4.6 menunjukkan tampilan *wordcloud* yang terdiri dari kata yang termasuk dalam label neutral yang banyak muncul dalam data penelitian.

GAMBAR 4.6 Wordcloud Label Neutral

Terlihat pada Gambar 4.6 kata kepala daerah, indonesia, dan presiden merupakan kata yang ukurannya paling besar diantara yang lain, itu berarti kata tersebut sering muncul dalam *tweet* yang tidak mengandung kalimat ujaran kebencian maupun kasar.

Script Python untuk membuat wordcloud abusive dapat dilihat pada kode program sebagai berikut:

```
# wordcloud kalimat abusive

wordcloud_abusive = WordCloud(width=800,
                              height=500,
                              background_color="white",
                              random_state=21,
                              max_font_size=110).generate(abusive)
plt.figure(figsize=(14, 12))
plt.imshow(wordcloud_abusive, interpolation="bilinear")
plt.axis('off')
plt.show()
```

Gambar 4.7 menunjukkan tampilan *wordcloud* yang terdiri dari kata yang termasuk dalam label abusive yang banyak muncul dalam data penelitian.

GAMBAR 4.6 Wordcloud Label Abusive

Script Python untuk membuat wordcloud hate speech dapat dilihat pada kode program sebagai berikut:

```
# wordcloud hate speech

wordcloud_hatespeech = WordCloud(width=800,
                    height=500,
                    colormap="Dark2",
                    background_color="white",
                    random_state=21,
                    max_font_size=110).generate(hate_speech)
plt.figure(figsize=(14, 12))
plt.imshow(wordcloud_hatespeech, interpolation="bilinear")
plt.axis('off')
plt.show()
```

Gambar 4.8 menunjukkan tampilan *wordcloud* yang terdiri dari kata yang termasuk dalam label abusive yang banyak muncul dalam data penelitian.

GAMBAR 4.6 Wordcloud Label Hate Speech

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Algoritma *Logistic Regression* dapat diterapkan untuk klasifikasi *multilabel hatespeech* dan *abusive* pada twitter Bahasa Indonesia.
2. Kombinasi fitur terbaik yang diperoleh adalah L1-norm + Stopwords + Case folding dengan akurasi sebesar 76,07%.
3. Kombinasi parameter terbaik adalah pada feature selection CO-L1 dengan nilai n_estimators=100, max_depth=10, criterion=”entropy”, min_samples_split=10, dan max_features=”auto”. Dengan hasil akurasi sebesar 76,20%.
4. Hasil akurasi model Random Forest dengan menggunakan fitur terbaik dan parameter terbaik sebesar 78%.

5.2 Saran

Saran yang dapat diberikan untuk penelitian lanjutan terkait dengan nelitian ini adalah :

1. Klasifikasi dapat dilakukan dengan menggunakan teknik *word embedding* lainnya seperti Word2Vec dan FastText, untuk menemukan teknik *word embedding* yang terbaik.
2. Pada penelitian selanjutnya dapat menggunakan metode klasifikasi lainnya untuk dapat melihat hasil perbandingan akurasinya seperti Naïve Bayes Classifier, Support Vector Machine, atau bahkan metode deep learning seperti Convolutional Neural Network.

DAFTAR PUSTAKA