PENGELOMPOKAN DATA TWEET KECELAKAAN MENGGUNAKAN PENDEKATAN *TEXT MINING* DAN ALGORITMA BIRCH

SKRIPSI

OLEH
IFTAH NUR FADLILAH
NIM. 18610015

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022

PENGELOMPOKAN DATA TWEET KECELAKAAN MENGGUNAKAN PENDEKATAN *TEXT MINING* DAN ALGORITMA BIRCH

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
Iftah Nur Fadlilah
NIM. 18610015

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2022

PENGELOMPOKAN DATA TWEET KECELAKAAN MENGGUNAKAN PENDEKATAN *TEXT MINING* DAN ALGORITMA BIRCH

SKRIPSI

Oleh
Iftah Nur Fadlilah
NIM. 18610015

Telah Diperiksa dan Disetujui Untuk Diuji
Malang, 22 Juni 2022

Dosen Pembimbing I	Dosen Pembimbing II
Hisyam Fahmi, M.Kom.	Ari Kusumastuti, M.Pd., M.Si.
NIP. 19890727 201903 1 018	NIP. 19770521 200501 2 004

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

PENGELOMPOKAN DATA TWEET KECELAKAAN MENGGUNAKAN PENDEKATAN *TEXT MINING* DAN ALGORITMA BIRCH

SKRIPSI

Oleh
Iftah Nur Fadlilah
NIM. 18610015

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal 24 Juni 2022

Ketua Penguji	: Muhammad Khudzaifah, M.Si.	.....
Anggota Penguji 1	: Mohammad Nafie Jauhari, M.Si.	.....
Anggota Penguji 2	: Hisyam Fahmi, M.Kom.	.....
Anggota Penguji 3	: Ari Kusumastuti, M.Pd., M.Si.	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan di bawah ini:

Nama	: Iftah Nur Fadlilah
NIM	: 18610015
Program Studi	: Matematika
Fakultas	: Sains dan Teknologi
Judul Skripsi	: Pengelompokan Data Tweet Kecelakaan Menggunakan Pendekatan *Text Mining* dan Algoritma BIRCH

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan atau daftar rujukan. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 24 Juni 2021
Yang membuat pernyataan,

Iftah Nur Fadlilah
NIM. 18610015

MOTO

"Boleh jadi kamu membenci sesuatu padahal ia amat baik bagimu, dan boleh jadi pula kamu menyukai sesuatu padahal ia amat buruk bagimu, Allah mengetahui sedang kamu tidak mengetahui"

(QS. Al – Baqarah: 216)

PERSEMBAHAN

Skripsi ini penulis persembahkan untuk:

Kedua Orang tua penulis, adik penulis serta keluarga penulis yang selalu memberikan motivasi dan semangat bagi penulis dalam menuntut ilmu dan selalu mendoakan untuk kelancaran studi penulis.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur atas kehadirat Allah SWT, Dzat yang Maha Agung penguasa seluruh jagad raya yang telah melimpahkan nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi dengan judul "Pengelompokan Data Tweet Kecelakaan Menggunakan Pendekatan *Text Mining* dan Algoritma BIRCH" dengan baik.

Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi besar Muhammad SAW. yang telah menunjukkan dan membimbing kita dari gelapnya zaman jahiliah menuju zaman yang terang benderang yakni dinul islam.

Skripsi ini disusun sebagai salah satu syarat untuk mendapatkan gelar sarjana dalam bidang matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Selesainya proses penyusunan skripsi ini tidak lepas dari bimbingan dan bantuan berbagai pihak, oleh karena itu penulis ucapkan terima kasih yang sebesar-besarnya dan penghargaan setinggi-tingginya kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Elly Susanti, M.Sc, selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.

4. Hisyam Fahmi, M.Kom, selaku dosen pembimbing I yang selalu memberikan arahan, nasihat dan berbagi ilmunya kepada penulis.
5. Ari Kusumastuti, M.Pd., M.Si, selaku dosen pembimbing II yang telah memberikan arahan, nasihat serta motivasi kepada penulis.
6. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim
7. Orang tua dan seluruh keluarga yang selalu memberikan doa, dukungan, semangat serta motivasi kepada penulis sampai saat ini.
8. Seluruh mahasiswa angkatan 2018 yang bersama-sama berjuang untuk mencapai impian dalam masa studi di UIN Maulana Malik Ibrahim Malang khususnya Program Studi Matematika.
9. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu penulis dalam menyelesaikan skripsi ini.
10. *Last but not least, I wanna thank me. I wanna thank me for believing in me. I wanna thank me for all doing this hard work. I wanna thank me for having no days off. I wanna thank me for never quitting. I wanna thank me for just being me at all times.*

Akhir kata, semoga skripsi ini dapat berguna khususnya bagi penulis serta seluruh pihak yang membaca laporan ini. Aamiin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 14 Juni 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR SIMBOL

$f_{t,d}$ = Frekuensi term (t) pada dokumen (d)

D = Jumlah semua dokumen

df_j = Jumlah dokumen yang mengandung term (tj)

w_{ij} = Bobot term terhadap dokumen (di)

tf_{ij} = Jumlah munculnya term (tj)

idf_j = Hasil dari perhitungan IDF

N = Jumlah data di *cluster*

LS = Jumlah koordinat data linier pada N

SS = Jumlah koordinat kuadrat dari data

CF_{12} = Gabungan CF_1 dan CF_2

N_1 = Jumlah data di *cluster* 1

N_2 = Jumlah data di *cluster* 2

$\overrightarrow{LS_1}$ = Jumlah koordinat data linier pada N_1

$\overrightarrow{LS_2}$ = Jumlah koordinat data linier pada N_2

SS_1 = Jumlah koordinat kuadrat dari data CF_1

SS_2 = Jumlah koordinat kuadrat dari data CF_2

a_i^j = Rata-rata jarak data ke-i terhadap semua data lainnya

i = Indeks data

j = *Cluster*

x = Data

m = Banyak data dalam *cluster*

$d(x_i^j, x_r^j)$ = Jarak data ke-i dengan data ke-r dalam satu *cluster j*

b_i^j = Rata-rata jarak data ke-i terhadap semua data dari *cluster* yang lain yang tidak berada dalam *cluster* data ke-i

m_n = Banyak data dalam satu *cluster*

d = Jarak data ke-i dengan data ke-r dalam satu *cluster j*

b = Nilai minimum dari rata-rata jarak data ke-i terhadap semua data dari *cluster* yang lain (tidak dalam satu *cluster* dengan data ke-i)

$d(x_i^j, x_r^n)$ = Jarak data ke-i terhadap semua data dari *cluster* yang lain yang tidak berada dalam satu *cluster* data ke-*j*

SI_i^j = *Silhouette Index* data ke-i

a = Rata-rata jarak data ke-i terhadap semua data lainnya dalam *cluster*

b = Nilai minimum dari rata-rata jarak data ke-i terhadap semua data dari *cluster* yang lain tidak dalam satu *cluster* dengan data ke-i

$max\{a_i^j, b_i^j\}$ = Nilai maksimum dari nilai a dan b dari satu data

SI_i^j = *Silhouette Index Cluster*

M_j = Banyaknya data dalam *cluster j*

k = Jumlah *cluster*

SI_j = *Silhouette Index Cluster*

SI_g = *Silhouette Index global*

SC = *Silhouette coefficient*

SI = Nilai *Silhouette Global*

k = Jumlah *cluster*

DAFTAR LAMPIRAN

ABSTRAK

Fadlilah, Iftah Nur. 2022. **Pengelompokan Data Tweet Kecelakaan Menggunakan Pendekatan Text Mining dan Algoritma BIRCH**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (1) : Hisyam Fahmi. M.Kom., Pembimbing (2) : Ari Kusumastuti, M.Pd., M.Si.

Kata Kunci : BIRCH, *Data Mining*, Kecelakaan, *Text Mining*, Tweet.

Kecelakaan merupakan suatu peristiwa yang tidak dapat diduga dan tidak diharapkan yang dipengaruhi oleh kendaraan bermotor serta berada di jalan raya atau tempat terbuka yang digunakan untuk tempat lalu lintas sehingga berakibat terjadinya kerusakan, luka-luka, kerugian materil dan berakibat fatal yaitu kematian. Meminimalisir tingkat kecelakaan belum dilaksanakan secara maksimal sehingga diperlukan pengelompokan dengan menggunakan metode yang sesuai dengan data yang diambil. Tujuan dari penelitian ini adalah untuk mengelompokkan data tweet kecelakaan menggunakan algoritma BIRCH. Data penelitian berupa data tweet kecelakaan yang diperoleh dari Kaggle dan *crawling* dari twitter. Penelitian ini menggunakan salah satu algoritma dalam data mining yang terintegrasi yakni algoritma BIRCH yang sebelumnya melalui proses pendeatan text mining. Algoritma BIRCH merupakan algoritma yang penemuan kelompok yang bagus dengan hanya menggunakan satu kali *scan* data. Hasil yang diperoleh dari penelitian ini adalah data yang diperoleh dari kaggle menghasilkan 1545 cluster dengan *silhouette coefficient* bernilai 0.1159964638217295 sedangkan dari data hasil crawling twitter menghasilkan 487 *cluster* dengan *silhouette coefficient* bernilai 0.7262655918349612.

ABSTRACT

Fadlilah, Iftah Nur. 2022. **Accident Tweet Data Grouping Using Text Mining Approach and BIRCH Algorithm**. Thesis. Mathematics Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Supervisor (1) : Hisyam Fahmi. M.Kom., Advisor (2) : Ari Kusumastuti, M.Pd., M.Si.

Keywords : Accident, BIRCH, Data Mining, Text Mining, Tweet.

Accident is an unexpected event that is influenced by motorized vehicles and is on a highway or open area used for traffic, resulting in damage, injury, material loss and fatal, namely death. Minimizing the accident rate has not been carried out optimally so that it is necessary to group it using a method that is in accordance with the data taken. The purpose of this study was to classify accidental tweet data using the BIRCH algorithm. Research data in the form of accident tweet data obtained from Kaggle and crawling from twitter. This study uses one of the algorithms in integrated data mining, namely the BIRCH algorithm which previously went through a text mining approach. The BIRCH algorithm is an algorithm that finds good clusters using only one scan dataThe results obtained from this study are that the data obtained from kaggle produces 1545 clusters with a silhouette coefficient of 0.1159964638217295 while the data from the Twitter crawl results produces 487 clusters with a silhouette coefficient of 0.7262655918349612.

مستخلص البحث

فضيلة، إفتح نور. ٢٠٢٢. **تجميع بيانات سقسق المصادمة باستخدام مدخل استخراج النص *(Text Mining)* و ألغوريتم بيرجه *(Algoritma BIRCH)***. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) هشام فهم، الماجستير، المشرفة: (٢) أري كوسوماستوتي، الماجستير.

الكلمات الأساسية: بيرجه *(Birch)*، استخراج البيانات *(Data Mining)*، الحادثة، استخراج النص *(text mining)*، السقسق *(Tweet)*.

إن المصادمة هي الحادثة غير المفكرة وغير المرجوة التي يتأثرها الركوبات فى الشوارع أو الأماكن المفتوحة المستخدمة لمكان المرور حتى تسبب إلى حدوث الفساد، الجرح، الخسر حتى المصيبة العظيمة وهي الموت. إن عملية تقليل درجة المصادمة لم تقام بها كاملا حتى أنها تحتاج إلى التجميع باستخدام الطريقة المناسبة بالبيانات المأخوذة. يهدف هذا البحث لتجميع بيانات سقسق *(tweet)* المصادمة باستخدام ألغورتم بيرجه *(Algoritma Birch)*. فبيانات البحث نحو بيانات سقسق *(tweet)* المصادمة المأخوذة من كاغيل (*Kaggle*) ومخدر (*crawling*) من توتير *(twitter)*. استخدم هذا البحث إحدى ألغورتم فى استخراج البيانات *(data mining)* المترابطة وهي ألغورتم بيرجه *(Algoritma Birch)* مما يكون بعدها بوسيلة عملية مدخل استخراج النص *(text mining)*. فألغورتم بيرجه *(Algoritma Birch)* هو ألغورتم الذي يجد التجميع الجيد ولو باستخدام مرة واحدة من البحث. فنتائج هذا البحث هي البيانات المحصولة من كاغيل *(kaggle)* تنتج ١٥٤٥ تجمعا *(cluster)* بمعامل الخيال *(silhouette coefficient)* له قيمة ٠.١١٥٩٩٦٤٦٣٨٢١٧٢٩ غير أن نتائج البيانات المحصولة من مخدر توتير *(crawling twitter)* تنتج ٤٨٧ تجمعا *(cluster)* بمعامل الخيال *(silhouette coefficient)* له قيمة ٠.٠٧٢٦٢٦٥٥٩١٨٣٤٩٦١٢.

BAB I
PENDAHULUAN

1.1. Latar Belakang

Kecelakaan lalu lintas merupakan topik yang sering dibicarakan baik dalam berita maupun pembicaraan masyarakat Indonesia. Berdasarkan data kecelakaan, angka kecelakaan lalu lintas mengalami kenaikan setiap tahunnya. Pada 2019 tercatat telah terjadi 116.411 kecelakaan lalu lintas di Indonesia dengan 25.671 orang tewas (Badan Pusat Statistik, 2021). Hal ini menunjukkan bahwa masalah tersebut harus memperoleh perhatian dan pengendalian yang efektif berhubungan dengan kebijakan yang digunakan oleh Direktorat Lalu Lintas (Ditlantas) pada setiap Kepolisian Daerah (Polda). Di dunia sendiri, angka kecelakaan yang tercatat meningkat setiap tahunnya, diperkirakan sekitar 1,35 juta orang tewas dalam kecelakaan lalu lintas di seluruh dunia. Oleh karena itu, kecelakaan digolongkan sebagai salah satu penyumbang kasus kematian terbesar di dunia.

Kecelakaan lalu lintas yang terjadi di dunia harus mendapatkan perhatian khusus. Di samping kecelakaan lalu lintas merupakan penyumbang kasus kematian terbesar, kecelakaan lalu lintas juga merupakan faktor yang mempengaruhi berkurangnya populasi manusia karena kecelakaan fatal lebih banyak terjadi sehingga mengakibatkan korban mengalami kematian. Kecelakaan lalu lintas dapat diatasi dengan pembentukan kebijakan atau tata tertib yang mengatur jalannya lalu lintas.

Kebijakan yang diterapkan semestinya mempunyai hubungan yang sesuai dan didukung dengan pengetahuan yang bersumber dari data yang ada. Informasi yang dicatat dan dijadikan data utama diperoleh dari mengetahui di mana, kapan,

dan bagaimana kecelakaan tersebut terjadi. Kebanyakan Ditlantas mempunyai sistem atau mekanisme sendiri untuk mengumpulkan kasus kecelakaan lalu lintas dari waktu ke waktu. Dari catatan data tersebut, Ditlantas secara teratur memprediksi jumlah kecelakaan, jumlah korban (baik luka ringan, luka berat maupun yang mati) dan total kerugian materiil lalu selanjutnya dapat dianalisis untuk menentukan daerah mana yang rawan terhadap kecelakaan lalu lintas.

Dari beberapa penjabaran di atas, meminimalisir tingkat kecelakaan belum dilaksanakan secara maksimal. Hal ini disebabkan karena tidak adanya pengelompokan atau *Clustering* data kecelakaan setelah data tersebut dirangkum pertahunnya. Untuk itu dibutuhkan suatu metode pengelompokan data yang sesuai dengan data penelitian yang diambil. Data tersebut kemudian akan diproses dengan menggunakan *Data Mining Clustering* yakni algoritma *Balance Iterative Reducing Clustering Using Hierarchies* (BIRCH). Hal ini karena algoritma BIRCH merupakan suatu algoritma yang mampu melakukan pengelompokan secara optimal dengan waktu yang optimal serta data yang besar.

Berdasarkan hal di atas, maka penulis tertarik untuk mengangkat judul "Pengelompokan Data Tweet Kecelakaan Menggunakan Pendekatan *Text Mining* dan Algoritma BIRCH". Penulis terinspirasi untuk melakukan penelitian menggunakan algoritma BIRCH karena algoritma BIRCH merupakan algoritma yang dikemukakan oleh Zhang, Ramakrishnan, dan Livny pada tahun 1996 (Han et.al, 2012) dalam dunia *Data Mining* khususnya *Clustering*, algoritma BIRCH sendiri di Indonesia kurang adanya penelitian yang mengangkat topik tersebut berikut merupakan contoh pengaplikasian algoritma BIRCH yakni dalam penelitian dengan judul *"Cluster Big Data dengan Balance Iterative Reducing Clustering*

Using Hierarchies (BIRCH)" yang melakukan proses pengelompokan menggunakan data online retail dengan memodifikasi nilai threshold yang dilakukan oleh Fanny Ramadhani (2019). Pada penelitian Yongki Kusworo (2018) menyatakan bahwa algoritma BIRCH dapat digunakan untuk mengelompokan sekolah menengah atas jurusan IPS di Provinsi Daerah Istimewa Yogyakarta berdasarkan indeks integritas ujian nasional tahun ajaran 2014/2015.

Namun untuk pengimplementasian algoritma BIRCH dalam data tweet khususnya tweet kecelakaan masih belum dilakukan. Oleh karena itu, perlunya untuk dilakukan penelitian tersebut sebagai bentuk pengembangan penelitian tentang algoritma BIRCH itu sendiri.

Pada penelitian ini akan menggunakan *Clustering* hirarki dalam mengelompokan data tweet kecelakaan dengan memanfaatkan algoritma BIRCH. Algoritma BIRCH merupakan salah satu Teknik *Clustering Mining*, pada penelitian akan melibatkan data tweet kecelakaan yang melalui proses *preprocessing* sebelum dilakukan proses *Clustering* algoritma BIRCH.

Pengimplementasian Algoritma BIRCH dengan data kecelakaan erat kaitannya dengan kepatuhan dalam menggunakan jalan, karena jika kita patuh mematuhi aturan pengguna jalan maka kecelakaan dapat diminimalisir serta dihindari. Dalam Islam telah dijelaskan dalam Al-Qur'an Surah Al-A'raf ayat 86 yang berbunyi:

وَلَا تَقْعُدُوا بِكُلِّ صِرَاطٍ تُوعِدُونَ وَتَصُدُّونَ عَنْ سَبِيلِ اللَّهِ مَنْ آمَنَ بِهِ وَتَبْغُونَهَا عِوَجًاوَاذْكُرُو إِذْ كُنْتُمْ قَلِيلًا فَكَثَّرَكُمْ وَانْظُرُو كَيْفَا كَانَ عَاقِبَةُالْمُفْسِدِينَ

"Dan janganlah kamu duduk di tiap-tiap jalan dengan menakut-nakuti dan menghalang-halangi orang yang beriman dari jalan Allah, dan menginginkan agar jalan Allah itu menjadi bengkok. Dan ingatlah di waktu dahulunya kamu berjumlah

sedikit, lalu Allah memperbanyak jumlah kamu. Dan perhatikanlah bagaimana kesudahan orang-orang yang berbuat kerusakan."

Dalam ayat ini menerangkan bahwa kita sebagai manusia janganlah berbuat kerusakan khususnya di jalan sehingga dapat mengganggu pengguna jalan lainnya seperti, menakut-nakuti ataupun merampas hak milik orang lain dengan cara membegal atau hal lainnya yang dari pada itu dapat mengakibatkan kecelakaan dan kerugian sampai kehilangan nyawa. Sama halnya dengan penelitian yang dilakukan oleh penulis, mengimplementasikan algoritma BIRCH dalam data kecelakaan lalu lintas yang harus dapat kita minimalisir sebaik mungkin.

1.2. Rumusan Masalah

Pada penelitian ini mempunyai rumusan masalah yakni bagaimana hasil pengelompokan data tweet kecelakaan di Indonesia menggunakan algoritma *Balanced Iterative Reducing and Clustering using Hierarchies* (BIRCH)?

1.3. Tujuan Penelitian

Pelaksanaan penelitian ini bersumber pada rumusan masalah yang ada yakni bertujuan untuk mengelompokan data tweet kecelakaan di Indonesia dengan menerapkan algoritma *Balanced Iterative Reducing and Clustering using Hierarchies* (BIRCH).

1.4. Manfaat Penelitian

Adapun manfaat dari penelitian ini yakni hasil dari penelitian ini dapat dijadikan informasi dan pengetahuan baru serta dapat digunakan sebagai sistem pendukung dalam menentukan karakteristik dari jenis kecelakaan.

1.5. Batasan Masalah

Agar analisis ini terarah, tepat sasaran dan tidak melenceng dari tujuan yang telah ditetapkan, maka penulis membatasi penelitian ini sebagai berikut:

1. Data yang dimanfaatkan adalah data tweet kecelakan di Indonesia yang diperoleh dari kaggle.
2. Data yang digunakan sebagai data 2 yakni data hasil *crawling* dari twitter.
3. Metode yang digunakan adalah metode *Clustering* dengan algoritma *Balanced Iterative Reducing and Clustering using Hierarchies* (BIRCH).
4. Data yang digunakan dikonversikan menjadi data berekstensi .csv
5. Data yang digunakan sebagai data 1 yakni data tweet kecelakaan dari April 2019 sampai April 2020.
6. Bahasa yang digunakan yakni bahasa pemrograman python.

1.6. Definisi Istilah

Agar terhindar dari kesalahpahaman dalam mendefinisikan istilah dalam penelitian ini, maka penulis memberikan Batasan-batasan pengertian sebagai berikut:

1. Kaggle merupakan *platform online* yang mempunyai kumpulan *dataset* yang digunakan baik individu maupun kelompok untuk memodelkan dan memperoleh suatu data.
2. Basis data adalah sekelompok data yang berukuran besar dengan tiap datanya berkaitan antara satu dan lainnya dan disimpan bersama yang dapat diolah menggunakan program komputer untuk mendapatkan informasi dari basis data tersebut.

3. *Record* dalam penelitian ini memiliki arti yaitu suatu bentuk penyimpanan kelompok dari sebuah *cluster*.
4. *Clustering feature* merupakan informasi yang mengandung *subcluster* dari objek data.
5. *Clustering feature tree* berguna untuk mengilustrasikan ringkasan *cluster*.
6. Data 1 merupakan data tweet kecelakaan yang diperoleh dari Kaggle.
7. Data 2 merupakan data tweet kecelakaan yang diperoleh melalui proses *crawling* data dari twitter.

BAB II
KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Data Mining

1. Definisi *Data Mining*

Data Mining merupakan susunan proses untuk mencari nilai lebih dari serangkaian data berupa pengetahuan yang sampai sekarang belum diketahui secara normal. Yang dimaksud dengan *Data Mining* adalah sebuah sistem untuk mengungkapkan serta menggali pengetahuan yang tercipta di dalam suatu dataset. *Data Mining* merupakan proses pencarian atau "menambang" sebuah pengetahuan dari sekelompok data dalam jumlah yang banyak (Han, Jiawei 2006). Sedangkan menurut ahli lainnya, *Data Mining* dapat menjadi metode yang memanfaatkan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstrak dan menguraikan informasi yang mempunyai manfaat yang terhubung ke berbagai basis data yang besar (Turban, dkk. 2005).

Berdasarkan pengertian mengenai *Data Mining* tersebut dapat ditarik kesimpulan bahwa *Data Mining* merupakan suatu proses pencarian secara otomatis untuk memperoleh pengetahuan dan informasi baru dari sejumlah data yang besar dengan memanfaatkan teknik yang tepat dengan pengolahan data yang diinginkan sehingga hasil dari penelitiannya dapat berguna bagi penggunanya.

2. Kategori *Data Mining*

Data Mining dapat diklasifikasikan menjadi dua bagian, yakni: (1). Deskripsi *Mining* yakni sebuah proses penting untuk mendapatkan karakteristik dari suatu data yang termasuk basis data, (2). Prediksi *Mining* yakni metode untuk

mendapatkan model dari suatu data dengan memanfaatkan variabel yang berbeda di masa depan.

3. Teknik *Data Mining*

Dalam suatu *database,* adanya *Data Mining* bukan hanya sebagai pelengkap saja namun mempunyai teknik penting untuk mempermudah memperoleh informasi serta meningkatkan pengetahuan bagi penggunanya maupun orang banyak. Teknik dan sifat *Data Mining* dapat dituliskan sebagai berikut (Hermawati, 2013):

a. Teknik Klasterisasi (*Unsupervised Learning*)

   Sebuah teknik untuk membagi *dataset* menjadi beberapa bagian sehingga komponen-komponen dari bagian tersebut dapat mempunyai *set property* yang dibagi secara sama, dengan tingkat kesamaan yang besar dalam suatu bagian yang kecil.

b. Teknik Regresi

   Sebuah teknik untuk menebak suatu nilai dari variabel kontinu yang didapatkan berdasarkan nilai dari variabel yang lain, dengan memperkirakan suatu model ketergantungan linier atau nonlinier.

c. Teknik Klasifikasi (*Supervised Learning*)

   Sebuah teknik untuk menentukan *record* data baru yang telah ditentukan sebelumnya ke salah satu kategori (kelas).

d. Kaidah Asosiasi (*Association rule*)

   Sebuah teknik untuk mengetahui dari atribut-atribut yang ada secara bersamaan (*co-occur*) dalam frekuensi yang sering dan membentuk beberapa aturan dari kelompok-kelompok tersebut.

4. Pengelompokan *Data Mining*

Berdasarkan tugas yang dapat dilakukan, *Data Mining* dibagi menjadi beberapa kelompok yakni:

a. Deskripsi (*Description*)

Penelitian analisis dengan cara sederhana untuk memilih cara mewujudkan pola dan kecondongan yang terletak dalam suatu data.

b. Estimasi (*Estimation*)

Model dibuat dengan memanfaatkan *record* lengkap untuk menyimpan nilai dari variabel target yang digunakan menjadi nilai prediksi. Berikutnya, perkiraan nilai dari variabel target dibentuk untuk menelusuri nilai prediksi.

c. Prediksi (*Prediction*)

Prediksi hampir mirip dengan estimasi dan klasifikasi, bedanya dalam prediksi nilai yang diperoleh terdapat di masa depan. Sejumlah metode yang ada dalam estimasi dan klasifikasi dapat juga dimanfaatkan untuk prediksi.

d. Klasifikasi (*Classification*)

Klasifikasi hampir sama dengan estimasi bedanya terdapat target variabel kategori. Jika estimasi lebih mengarah pada variabel numerik sedangkan klasifikasi lebih mengarah ke kategori.

e. Pengelompokan (*Clustering*)

Clustering menggambarkan pengelompokan *record*, pemeriksaan atau mengamati dan membangun kelas objek-objek yang mempunyai kesamaan. *Cluster* adalah kelompok *record* yang mempunyai kesamaan

satu dengan yang lainnya dan mempunyai ketidaksamaan dengan *record-record* dalam *Cluster* lain. Pengklusteran membagi atas semua data menjadi bagian-bagian yang mempunyai kesamaan (homogen), di mana kesamaan *record* dalam satu bagian yang bernilai maksimum sedangkan kesamaan dengan *record* lain menjadi bernilai minimum.

f. Asosiasi (*Association*)

Asosiasi mempunyai tugas dalam *Data Mining* yakni mendapatkan atribut yang ada dalam kurun waktu yang sama.

2.1.2 *Text Mining*

Text Mining adalah suatu konsep terapan dalam *Data Mining* untuk menemukan suatu pola dari teks. Tujuan dari *Text Mining* yakni untuk memperoleh dan memanfaatkan suatu informasi yang ada dalam teks tersebut. Suatu data teks banyak mengandung ketidakteraturan kata baik kata-kata imbuhan maupun kiasan. Oleh karena itu, untuk memperoleh data teks yang lebih baik atau terstruktur diperlukan beberapa tahapan.

Proses *Text Mining* harus melewati lima tahapan untuk mendapatkan hasil yang diharapkan. Berikut merupakan tahapan-tahapan dalam *Text Mining* (Hidayatullah, 2014):

1. *Text Preprocessing*

   Tahap ini bertujuan untuk menyiapkan data teks yang selanjutnya akan diproses pada tahap berikutnya. *Text preprocessing* memuat *case folding* yakni proses mengubah karakter huruf besar menjadi huruf kecil dalam data teks.

2. *Text Transformation*

Pada tahap ini hasil yang diperoleh dari proses text *preprocessing* akan dilanjutkan dengan proses transformasi. Pada proses ini meliputi dua tahap yakni:

a. Stopword removal

Stopword removal merupakan proses untuk mengurangi jumlah dari setiap kata seperti menghilangkan kata sambung, imbuhan dan lainnya. Proses ini dapat mengurangi beban kinerja sistem, karena kata-kata yang tidak penting sudah dihilangkan.

b. Stemming

Stemming merupakan suatu proses untuk mengubah kata-kata menjadi bentuk baku dalam data teks.

3. *Feature Selection*

Dalam *feature selection* terdapat beberapa metode yang dapat digunakan, yakni:

a. Document Frequency

Document frequency merupakan metode untuk mencari berapa banyak suatu term atau kata muncul dalam data dokumen yang akan dianalisis.

b. Term frequency

Term frequency ($tf_{t,d}$) merupakan metode yang digunakan untuk menghitung banyaknya term atau kata yang muncul dalam suatu *corpus* terhadap suatu bobot term (t) atau kata pada dokumen (d).

c. *Term frequency-Inverse Document frequency* (TF-IDF)

TF-IDF merupakan metode yang terdiri dari *Term frequency* dan *Inverse Document Frequency*. Metode ini menghitung setiap kata kunci dari masing-masing dokumen.

Rumus TF dapat dituliskan sebagai berikut:

$$TF = \begin{cases} 1 + log_{10}(f_{t,d}), f_{t,d} > 0 \\ 0, f_{t,d} = 0 \end{cases} \quad (2.1)$$

Keterangan:

$f_{t,d}$ = Frekuensi term (t) pada dokumen (d)

Selanjutnya, rumus IDF dapat dituliskan sebagai berikut:

$$IDF_j = \log\left(\frac{D}{df_j}\right) \quad (2.2)$$

Keterangan:

D = Jumlah semua dokumen

df_j = Jumlah dokumen yang mengandung term (tj)

Rumus TF-IDF merupakan gabungan dari perhitungan TF dengan IDF:

$$w_{ij} = tf_{ij}\ x\ idf_j \quad (2.3)$$

Keterangan:

w_{ij} = Bobot term terhadap dokumen (di)

tf_{ij} = Jumlah munculnya term (tj)

idf_j = Hasil dari perhitungan IDF

4. *Pattern Discovery*

   Tahap ini berguna untuk mendapatkan suatu *knowledge* atau pola dengan memanfaatkan salah satu atau beberapa Teknik dalam *Data Mining* seperti *classification* dan *Clustering*.

5. *Interpretation*

   Pada tahap akhir dilakukan proses interpretasi ke suatu bentuk yang selanjutnya dilaksanakan evaluasi.

2.1.3 *Clustering*

1. Definisi *Clustering*

   Teknik *Clustering* adalah salah satu strategi analisis data untuk membantu mengidentifikasi pengelompokan objek data dari suatu *dataset*. *Clustering* merupakan klasifikasi yang tidak melakukan pengamatan dan menggambarkan proses partisi sekelompok objek data dari satu set ke banyak kelas. Hal ini dilaksanakan dengan mengimplementasikan *Euclidean Distance* sebagai persamaan dan langkah-langkah tentang jarak algoritma (Venkateswaslu & Raju. 2013).

   Clustering dapat diartikan sebagai contoh dari teknik *Data Mining* yakni sekelompok objek yang memiliki “kesamaan” di antara elemennya dan mempunyai “ketidaksamaan” dengan objek lain pada *Cluster* lainnya, dapat dikatakan juga bahwa sebuah *Cluster* merupakan sekelompok objek yang dihimpun bersama berdasarkan kemiripan atau kedekatan jaraknya.

2. Karakteristik *Clustering*

   Berikut merupakan beberapa karakteristik dari *Clustering* yang akan diuraikan sebagai berikut (Jiawei Han dan Micheline kamber, 2001):

a. *Partitioning Clustering*.

1) Dapat dikatakan sebagai exclusive *Clustering*.

2) Data yang ada wajib berada dalam *Cluster* tertentu.

3) Setiap data yang berada pada *Cluster* tertentu pada salah satu proses dapat berubah ke *Cluster* yang lain pada proses selanjutnya. Contoh: K-Means, *residual analysis*.

b. *Hierarchical Clustering*.

1) Semua data wajib termasuk ke dalam suatu *Cluster* tertentu

2) Setiap data yang berada pada *Cluster* tertentu pada salah satu proses tidak dapat berubah ke *Cluster* yang lain pada proses selanjutnya. Contoh: *Single Linkage*, *Centroid Linkage*, *Complete Linkage*

c. *Overlapping Clustering*.

1) Memungkinkan setiap data berada di beberapa *Cluster*

2) Data dapat memiliki nilai keanggotaan pada beberapa *Cluster*. Contoh: *Fuzzy C-means*, *Gaussian Mixture*

d. *Hybrid clustering* adalah gabungan dari *partitioning*, *overlapping* dan *hierarchical*.

2.1.4 Algoritma BIRCH (*Balance Iterative Reducing and Clustering Using Hierarchies*)

1. Definisi Algoritma BIRCH

Algoritma BIRCH adalah salah satu algoritma dalam metode *Clustering* yang bersifat hirarki atau sering disebut dengan *hierarchical Clustering method*. Kelebihan dari algoritma BIRCH ini yakni penemuan

kelompok yang bagus dengan hanya menggunakan satu kali *scan data*. Selain itu, algoritma BIRCH dapat meningkatkan kualitas menjadi lebih bagus dengan melakukan *scan* tambahan. Algoritma BIRCH sangat sesuai apabila digunakan untuk memproses data besar atau yang sering dikenal dengan big data.

Algoritma BIRCH merupakan metode *Clustering* hirarki yang terintegrasi dan digunakan untuk pemrosesan data yang besar. Pada algoritma BIRCH terdapat dua konsep yaitu, *Clustering feature* dan *Clustering feature tree* (CF-Tree) yang berguna untuk mengilustrasikan ringkasan *cluster*.

2. Proses Algoritma BIRCH (*Balance Iterative Reducing and Clustering Using Hierarchies*)

Algoritma BIRCH melakukan pemrosesan data ke dalam empat fase. Fase 1 yakni memasukkan ke dalam memori dengan membangun CF-Tree. Tahap pertama yang perlu dilakukan yakni membentuk fitur *Clustering tree* (CF Tree) dari *cluster* data di mana struktur data tree yang dibentuk memiliki tinggi yang sama rata. Pada fase ini, algoritma BIRCH menghadirkan dua konsep yakni *Clustering feature* (CF) dan *Clustering feature tree* (CF Tree) yang dipakai untuk mewujudkan ringkasan *cluster*.

a. *Clustering Feature*

Clustering feature merupakan informasi yang mengandung *subcluster* dari objek data. *Clustering feature* juga merupakan seperangkat dari tiga statistik ringkasan yang mewakili satu set titik data dalam satu *cluster*. Meringkas *cluster* menggunakan *Clustering feature* dapat menghindari penyimpanan informasi secara rinci mengenai objek data. Pada algoritma ini hanya perlu ukuran ruang yang

konstan untuk menyimpan setiap *Clustering feature* yang dibangun. Ini adalah kunci mengapa BIRCH dikatakan memiliki *space* yang efisien dan sangat efektif. *Clustering feature* memiliki rumus sebagai berikut:

$$CF = (N, LS, SS) \tag{2.4}$$

$$LS = \sum_{P_i \in N} P_i \tag{2.5}$$

$$SS = \sum_{P_i \in N} |P_i|^2 \tag{2.6}$$

Keterangan:

N = Jumlah data di *cluster*

LS = Jumlah koordinat data linier pada N

SS = Jumlah koordinat kuadrat dari data

Salah satu mekanisme dari algoritma BIRCH membutuhkan penggabungan *cluster* dalam kondisi tertentu. Aditivitas Teorema menyatakan bahwa CF untuk dua *cluster* dapat digabungkan hanya dengan menambahkan item dalam CF-Tree masing-masing. *Cluster feature* bisa digabungkan dengan *cluster* yang memiliki jarak yang dekat selama masih sesuai dengan *threshold*nya. Misalkan terdapat *cluster* $CF_1 = (N, LS, SS)$ dan $CF_2 = (N, LS, SS)$. *Cluster feature* baru dapat dibentuk dengan menggabungkan CF_1 dan CF_2 yaitu dengan rumus:

$$CF_{12} = (N_1 + N_2, \overrightarrow{LS_1} + \overrightarrow{LS_2}, SS_1 + SS_2) \tag{2.7}$$

Keterangan:

CF_{12} = Gabungan CF_1 dan CF_2

N_1 = Jumlah data di *cluster* 1

N_2 = Jumlah data di *cluster* 2

$\overrightarrow{LS_1}$ = Jumlah koordinat data linier pada N_1

$\overrightarrow{LS_2}$ = Jumlah koordinat data linier pada N_2

SS_1 = Jumlah koordinat kuadrat dari data CF_1

SS_2 = Jumlah koordinat kuadrat dari data CF_2

b. *Clustering Feature* Tree

Langkah-langkah untuk membangun CF-Tree yakni sebagai berikut:

1) Setiap *subcluster* CF yang ada, algoritma BIRCH membandingkan lokasi *record* itu dengan lokasi dari masing-masing CF di *root node* menggunakan *euclidean distance*. Algoritma BIRCH meneruskan *subcluster* Cf masuk ke *root node* CF paling dekat dengan *subcluster* CF.
2) *Record* kemudian turun ke *node child non-leaf* dari simpul CF yang dipilih pada langkah 1. Algoritma BIRCH membandingkan lokasi *record* dengan lokasi setiap CF *non-leaf*. Algoritma BIRCH meneruskan *record* yang masuk ke simpul *non-leaf* CF yang paling dekat dengan *record* yang masuk.
3) *Record* kemudian turun ke *node child* dari simpul *non-leaf* CF yang dipilih pada langkah 2. Algoritma BIRCH membandingkan lokasi *record* dengan lokasi setiap *leaf*. Algoritma BIRCH untuk sementara melewati *record* yang masuk ke *leaf* yang paling dekat dengan *subcluster* CF yang masuk.

4) Melakukan salah satu dari dua cara berikut ini:

   a) Jika jari-jari dari *leaf* yang dipilih termasuk *record* baru tidak melebihi *threshold* T, maka *subcluster* CF yang masuk ditugaskan ke *leaf* itu. *Leaf* dan semua CF induknya diperbaharui untuk memperhitungkan titik data baru.

   b) Jika jari-jari *leaf* yang dipilih termasuk *subcluster* CF melebihi *threshold* T, maka *leaf* baru terbentuk, terdiri dari *record* yang masuk saja. CF induk diperbarui ke akun untuk titik data baru.

Jika langkah 4b dilakukan dan memiliki maksimum *leaf* L dalam *leaf node*, maka *leaf node* dibagi menjadi dua *leaf node*. *Node leaf* paling jauh CF digunakan sebagai benih *leaf node*, dengan CF tersisa ditugaskan pada *leaf node* mana yang lebih dekat. Jika simpul induk penuh, pisahkan simpul induk, dan seterusnya, setiap *leaf node* CF dapat dilihat sebagai *subcluster*. Pada langkah *cluster*, *subcluster* akan digabungkan menjadi *cluster*. Untuk *cluster* yang diberikan, maka centroid *cluster* menjadi:

$$D2 = \frac{\sqrt{(N_1 SS_2) + (N_2 SS_1) - 2LS_1 LS_2}}{N_1 N_2} \tag{2.8}$$

Keterangan:

N_1 = Jumlah data di *cluster* 1

N_2 = Jumlah data di *cluster* 2

$\overrightarrow{LS_1}$ = Jumlah koordinat data linier pada N_1

$\overrightarrow{LS_2}$ = Jumlah koordinat data linier pada N_2

SS_1 = Jumlah koordinat kuadrat dari data CF_1

SS_2 = Jumlah koordinat kuadrat dari data CF_2

Dengan radiusnya:

$$R = \frac{\sqrt{SS - (LS)^2}}{n^2} \quad (2.9)$$

Keterangan:

SS = Jumlah koordinat kuadrat

LS = Jumlah koordinat data linier

Fase 2 yakni membangun CF-Tree yang lebih kecil (*optional*). Tahap kedua, algoritma akan memindah semua anggota di awal CF-Tree untuk membentuk kembali CF Tree yang lebih kecil, serta melakukan penghapusan outlier dan pengelompokan *subcluster* yang ramai menjadi *subcluster* yang lebih besar. Langkah ini ditandai dengan presentasi asli BIRCH yang opsional.

Fase 3 Global *Clustering.* Pada tahap ketiga, algoritma yang ada digunakan untuk mengelompokkan semua *entry tree* yang ada. Pada tahap ini juga algoritma *Agglomerative Hierarchical Clustering* diterapkan *subcluster* yang diwakili oleh CF *leaf node*. Oleh karena itu, memungkinkan pengguna untuk menentukan jumlah *cluster* yang dibutuhkan atau batas diameter yang dibutuhkan untuk *cluster*.

Fase 4 menghilangkan *cluster-cluster* (*optional dan offline*). Tahap keempat, centroid *cluster* yang diperoleh dari tahap ketiga dipakai sebagai *seed* dan digunakan sebagai pendistribusian ulang *cluster* data ke *seed* terdekat untuk memperoleh set *cluster* baru. Pada tahap keempat, diberikan pilihan untuk menghilangkan *outlier*.

3. Karakteristik Algoritma BIRCH

Algoritma BIRCH memiliki beberapa karakteristik yang menjadikannya algoritma yang terintegrasi daripada algoritma *Clustering* yang lainnya, yakni:

a. Algoritma BIRCH bersifat lokal, yang dimaksud dengan ini yakni algoritma BIRCH dalam setiap pengambilan keputusan untuk *Clustering* dapat dibentuk tanpa memindahkan semua *cluster* data yang ada.
b. Algoritma BIRCH menghilangkan pandangan bahwa ruang data (*data space*) yang semula tidak semuanya terpenuhi dan oleh karena itu tidak setiap *cluster* data adalah sama pentingnya untuk tujuan pengelompokan.
c. Algoritma BIRCH membutuhkan waktu yang relatif sedikit atau dapat dikatakan pemrosesan datanya cepat dalam melakukan *Clustering*.

2.1.5 *Silhouette Coefficient*

Silhouette Coefficient merupakan sebuah metode yang digunakan untuk mengetahui baik atau tidak kualitas sebuah *cluster*. *Silhouette Coefficient* dapat dimanfaatkan untuk memvalidasi *cluster* tunggal maupun semua *cluster*. Sebelum menghitung *Silhouette Coefficient*, terlebih dahulu yang harus dilakukan yakni menghitung nilai *Silhouette Index* data ke-i, dengan menghitung dua komponen yaitu a_i yang merupakan rata-rata jarak data ke-i terhadap semua data dalam satu *cluster* dan b_i merupakan rata-rata jarak data ke-i terhadap semua data dari *cluster* lain yang tidak dalam satu *cluster* dengan data ke-i, selanjutnya dipilih nilai yang paling kecil (Larose, 2015).

Rumus untuk menghitung a_i^j adalah sebagai berikut:

$$a_i^j = \frac{1}{Mj-1}\sum_{\substack{r=1 \\ r\neq 1}}^{Mj} d(x_i^j, x_r^j) \quad (2.10)$$

Keterangan:

a_i^j = Rata-rata jarak data ke-i terhadap semua data lainnya

i = Indeks data

j = *Cluster*

x = Data

m = Banyak data dalam *cluster*

$d(x_i^j, x_r^j)$ = Jarak data ke-i dengan data ke-r dalam satu *cluster j*

Dengan $d(x_i^j, x_r^j)$, diketahui sebagai berikut:

$$d = \sqrt{(x_i^j - x_r^j)^2} \quad (2.11)$$

Selanjutnya yakni rumus untuk menghitung b_i^j :

$$b_i^j = \min_{\substack{n1\ldots k \\ n\neq j}} \left(\frac{1}{Mn}\right)\sum_{\substack{r=1 \\ r\neq 1}}^{Mn} d(x_i^j, x_r^n) \quad (2.12)$$

Keterangan:

b_i^j = Rata-rata jarak data ke-i terhadap semua data dari *cluster* yang lain yang tidak berada dalam *cluster* data ke-i

m_n = Banyak data dalam satu *cluster*

d = Jarak data ke-i dengan data ke-r dalam satu *cluster j*

x = Data

i = Indeks data

j = *Cluster*

b = Nilai minimum dari rata-rata jarak data ke-i terhadap semua data dari *cluster* yang lain (tidak dalam satu *cluster* dengan data ke-i)

$d(x_i^j, x_r^n)$ = Jarak data ke-i terhadap semua data dari *cluster* yang lain yang tidak berada dalam satu *cluster* data ke-j

dengan sebelumnya $d(x_i^j, x_r^n)$ diketahui sebagai berikut:

$$d = \sqrt{(x_i^j - x_r^n)^2}$$

Selanjutnya yakni rumus untuk menghitung *Silhouette Index* data ke-i:

$$SI_i^j = \frac{b_i^j a_i^j}{max\{a_i^j, b_i^j\}} \quad (2.13)$$

Keterangan:

SI_i^j = *Silhouette Index* data ke-i

a = Rata-rata jarak data ke-i terhadap semua data lainnya

b = Nilai minimum dari rata-rata jarak data ke-i terhadap semua data dari *cluster* yang lain tidak dalam satu *cluster* dengan data ke-i

$max\{a_i^j, b_i^j\}$ = Nilai maksimum dari nilai a dan b dari satu data

Berikut merupakan rumus untuk menghitung nilai SI sebuah *cluster*:

$$SI_j = \frac{1}{M_j} \sum_{i=1}^{M_j} SI_i^j \quad (2.14)$$

Keterangan:

SI_i^j = *Silhouette Index Cluster*

i = Indeks data

j = *Cluster*

M_j = Banyaknya data dalam *cluster j*

Selanjutnya adalah rumus untuk memperoleh nilai SI global yang diperoleh dari menghitung rata-rata nilai SI dari semua *cluster*:

$$SI_g = \frac{1}{k}\sum_{j}^{k} SI_j \quad (2.15)$$

Keterangan:

j = *Cluster*

k = Jumlah *cluster*

SI_j = *Silhouette Index Cluster*

SI_g = *Silhouette Index global*

Berikut merupakan rumus untuk mengetahui hasil dari perhitungan *Silhouette Coefficient*:

$$SC = max_k\, SI_g(k) \quad (2.16)$$

Keterangan:

SC = *Silhouette coefficient*

SI = Nilai *Silhouette Global*

k = Jumlah *cluster*

Cara mengetahui baik atau tidak ukuran *Silhouette Coefficient* dengan melihat tabel berikut ini (Kauffman & Rousseeuw, 1990):

Tabel 2.1 Kriteria Subjektif Berdasarkan *Silhouette Coefficient*

Nilai *Silhouette Coefficient*	Interpretasi *Silhouette Coefficient*
0,71 – 1,00	Struktur kuat
0,51 – 0,70	Struktur baik
0,26 – 0,50	Struktur lemah
≤ 0,25	Struktur buruk

2.1.6 Kecelakaan

1. Definisi Kecelakaan

Kecelakaan adalah suatu peristiwa yang tidak diharapkan yang dapat menimbulkan kerugian bagi manusia, kerusakan pada benda, serta kekacauan terhadap proses (Heinrich, 1996). Kecelakaan lalu lintas yakni kejadian tak disangka yang terjadi pada lalu lintas, melibatkan kendaraan baik dengan atau tidak pejalan kaki atau pengguna jalan lainnya, serta berakibat adanya korban manusia atau kerugian materil. Menurut salah satu ahli, kecelakaan lalu lintas adalah kejadian yang tidak dapat diprediksi kapan dan di mana terjadinya. Kecelakaan juga tidak hanya menimbulkan trauma, cedera, maupun luka lainnya, namun juga bisa menimbulkan kematian. Kasus kecelakaan sendiri tidak mudah untuk dikurangi dan akan meningkat sesuai dengan banyaknya jalan dan mobilitas dari kendaraan (Hobbs, 1995). Berdasarkan pengertian kecelakaan lalu lintas di atas bisa diambil kesimpulan bahwa kecelakaan lalu lintas adalah suatu peristiwa yang tidak dapat diduga dan tidak diharapkan yang dipengaruhi oleh kendaraan bermotor serta berada di jalan raya atau tempat terbuka yang digunakan untuk tempat lalu lintas sehingga berakibat terjadinya kerusakan, luka-luka, kerugian materil dan berakibat fatal yaitu kematian.

2. Jenis dan Bentuk Kecelakaan

Jenis dan bentuk kecelakaan dapat digolongkan menjadi lima kategori yakni:

a. Kecelakaan bersumber pada korban kecelakaan

Akibat dari terjadinya suatu kecelakaan lalu lintas dapat dialami oleh semua atau hanya beberapa orang yang terlibat saja. Berikut merupakan

klasifikasi yang dimanfaatkan untuk membedakan korban kecelakaan lalu lintas, yakni:

1) kecelakaan Luka Fatal atau Meninggal merupakan korban kecelakaan yang ditetapkan meninggal dunia disebabkan kecelakaan lalu lintas dalam kurun waktu selambatnya 30 hari setelah terjadinya kecelakaan.
2) Kecelakaan Luka Berat merupakan korban kecelakaan yang mendapatkan luka-luka berat atau cacat tetap dan memerlukan tindakan lebih lanjut sehingga dibutuhkan perawatan di Rumah Sakit dalam kurun waktu lebih dari 30 hari dari peristiwa kecelakaan.
3) Kecelakaan Luka Ringan merupakan korban kecelakaan yang mengalami luka-luka yang tidak perlu untuk melakukan penyembuhan lebih lanjut di Rumah Sakit.

b. Kecelakaan yang bersumber pada lokasi kejadian

Kecelakaan bisa terjadi di mana dan kapan saja sepanjang ruas jalan, baik pada jalan lurus maupun tikungan, tanjakan atau turunan, di dataran tinggi atau dataran rendah, di dalam kota ataupun di luar kota (Wedasana, 2011).

c. Kecelakaan bersumber pada waktu peristiwa kecelakaan

Kecelakaan bersumber pada waktu peristiwa kecelakaan bisa dikelompokan menjadi dua yakni:

1) Berdasarkan Hari
    a) Pada hari kerja yakni Senin, Selasa, Rabu, Kamis, dan Jumat

b) Pada Hari Libur yakni Minggu serta hari libur nasional

c) Pada Akhir Pekan yaitu Sabtu

2) Berdasarkan Waktu

a) Dini Hari antara pukul 00.00 – 06.00

b) Pagi Hari antara pukul 06.00 – 12.00

c) Siang Hari antara pukul 12.00 – 18.00

d) Malam Hari antara pukul 18.00 – 24.00

d. Kecelakaan bersumber pada posisi kecelakaan

Kecelakaan berdasarkan posisi kecelakaan dapat digolongkan menjadi (Hubdat, 2006):

1) *Angle* (Ra) merupakan bentuk tabrakan antara kendaraan yang melaju dengan arah yang berbeda, namun tidak dari arah yang berlawanan,

2) *Rear-End* (Re) merupakan bentuk tabrakan antara kendaraan yang bergerak searah dan menabrak dari belakang kendaraan,

3) *Sideswipe* (Ss) merupakan kendaraan yang bergerak dengan arah yang sama, atau dengan arah yang berlawanan dan menabrak dari samping pada kendaraan lain,

4) *Head-On* (Ho) merupakan kendaraan yang bertabrakan dan berjalan dengan arah yang tidak sama atau berlawanan (*No Sideswipe*),

5) *Backing* merupakan kendaraan yang bertabrakan secara mundur.

e. Kecelakaan bersumber pada jumlah kendaraan

Kecelakaan bersumber pada jumlah kendaraan yang terlibat digolongkan menjadi:

1) Kecelakaan Tunggal merupakan kecelakan yang tidak melibatkan pejalan lain hanya kecelakaan yang dilakukan oleh satu kendaraan bermotor saja, contohnya menabrak pembatas jalan, menabrak pohon atau yang lainnya.
2) Kecelakaan Ganda yakni kecelakaan yang melibatkan dua pengguna jalan baik kendaraan bermotor maupun pejalan kaki yang mengalami kecelakaan di waktu dan tempat yang sama.
3) Kecelakaan Beruntun merupakan kecelakaan yang melibatkan lebih dari dua pengguna jalan baik kendaraan bermotor maupun pejalan kaki yang mengalami kecelakaan yang sama berdasarkan waktu dan tempat terjadinya.

3. Faktor-faktor Penyebab Kecelakaan Lalu Lintas

Banyak faktor yang dapat mengakibatkan kecelakaan lalu lintas terjadi, namun pada dasarnya kecelakaan terjadi karena kurangnya keefektifan dari kumpulan faktor-faktor utama yang meliputi: pemakai jalan, lingkungan jalan serta kendaraan (Harahap, 1995). Untuk lebih lanjut faktor-faktor tersebut diuraikan sebagai berikut:

a. Faktor Pengguna Jalan

Faktor penting dalam terjadinya lalu lintas yakni pengguna jalan, karena manusia adalah pengguna jalan dan pengguna jalan menjadi

faktor utama terjadinya pergerakan lalu lintas (Soesantiyo, 1985). Manusia sebagai pengguna jalan bisa dikategorikan menjadi dua yakni:

1) Pengemudi, meliputi pengemudi kendaraan tak bermotor.
2) Pejalan kaki, meliputi para pedagang asongan, pedagang kaki lima, dan lainnya.

b. Faktor Pengemudi

Faktor fisik utama bagi pengemudi agar dapat mengemudikan kendaraan dan menangani masalah lalu lintas yakni:

1) Penglihatan

   Penglihatan merupakan faktor fisik utama yang dibutuhkan pengemudi dalam melajukan kendaraannya karena hampir semua informasi yang diperoleh melalui penglihatan berguna untuk mengemudikan kendaraan.

2) Pendengaran

   Pendengaran merupakan faktor fisik utama yang dibutuhkan untuk dapat memahami himbauan-himbauan seperti contoh, bunyi klakson, peluit polisi dan lainnya. Terkadang himbauan itu dibarengi dengan isyarat yang bisa terlihat oleh mata. Oleh karena itu, keadaan fisik manusia atau *Human Physical Factor* memiliki keterkaitan erat dalam mengemudi, berikut merupakan urutan saat manusia menerima rangsangan Ketika melihat suatu tanda, yaitu:

   a) *Perception* atau dalam bahasa Indonesia disebut dengan pengamatan adalah rangsangan yang dilakukan panca indera

yaitu penglihatan kemudian disambung oleh panca indera yang lainnya.

b) *Identification* adalah penerjemahan dan pemahaman terhadap suatu rangsangan.

c) *Emotion* atau *Judgement* adalah proses pengumpulan keputusan untuk memutuskan reaksi yang sesuai.

d) *Violation* (reaksi) adalah pengambilan respons yang memerlukan kerjasama dengan kendaraan.

c. Faktor pejalan kaki

Faktor selanjutnya yang mempengaruhi kecelakaan itu terjadi adalah faktor pejalan kaki. Pejalan kaki bisa sebagai korban kecelakaan atau bisa sebagai penyebab kecelakaan itu terjadi. Perlunya pemisah antara pengemudi kendaraan bermotor dengan pejalan kaki itu sendiri seperti dibentuknya *zebra cross* maupun trotoar khusus pejalan kaki (Warpani, 2001). Akan tetapi, terkadang pejalan kaki tidak mematuhi peraturan yang telah ada sehingga menjadi penyebab suatu kecelakaan dapat terjadi.

d. Faktor Kendaraan

Kendaraan saat ini sudah didesain sedemikian hingga untuk keamanan penggunanya. Namun, kendaraan juga harus memperoleh perawatan yang baik agar bagian kendaraan juga dapat berfungsi secara optimal. Berikut merupakan faktor kendaraan yang dapat menyebabkan terjadinya suatu kecelakaan:

1) Rem tidak berfungsi, kegunaan rem pada kendaraan sangatlah penting karena apabila ada kejadian di depan kendaraan yang mendadak rem dapat digunakan jika rem tidak berfungsi maka laju kendaraan tidak dapat dihentikan.
2) *Overload* atau kelebihan kapasitas merupakan suatu tindakan yang melanggar tata tertib ketentuan muatan.
3) Desain kendaraan juga berpengaruh dalam terjadinya kecelakaan, karena apabila terdapat desain pada kendaraan yang berbahaya maka dapat memperparah apabila kecelakaan itu terjadi.
4) Lampu kendaraan, dalam hal ini lampu kendaraan memiliki peranan penting yakni agar pengemudi dapat mengetahui keadaan sekitar sehingga pengemudi dapat menyesuaikan dengan kecepatannya dan bisa melihatkan kendaraan ke pengguna jalan lain.

e. Faktor Jalan

Penyebab terjadinya kecelakaan lalu lintas juga disebabkan oleh faktor jalan. Keadaan jalan yang tidak baik atau rusak beserta rambu-rambu lalu lintas yang tidak berjalan dengan baik bisa mengakibatkan kecelakaan lalu lintas. Ahli jalan raya dan pakar lalu lintas turut bekerjasama dalam pengadaan jalan dan peraturan-peraturan dengan uraian standar yang dijalankan dengan baik dan perawatan yang baik supaya keselamatan transportasi jalan bisa terlaksana. Selain itu, efek besar yang mengakibatkan terjadinya kecelakaan yakni lebar jalan, kelengkungan, dan jarak pandang.

f. Faktor Lingkungan

Kondisi alam juga dapat menyebabkan terjadinya kecelakaan lalu lintas. Salah satu keadaan alam yang menjadi bahaya utama saat berkendara yakni faktor cuaca apalagi saat musim hujan, bila terjadi hujan deras maka akan mengakibatkan jalanan lebih licin dan memperpendek jarak pandang pengemudi saat berkendara. Selain itu, terdapat faktor alam lain seperti kabut, banjir, gempa bumi atau yang lainnya juga dapat mengakibatkan kecelakaan.

2.1.7 Crawling Data Twitter

Pengambilan data atau disebut dengan proses *crawling* data twitter menggunakan API Key Twitter dan proses pengambilan data Twitter dibantu dengan Bahasa pemrograman python. API Key Twitter adalah Application Programming Interface (API) dalam API ini suatu layanan untuk sekumpulan perintah, fungsi, komponen dan juga protokol yang disediakan untuk mempermudah penulis pada saat membangun suatu sistem perangkat lunak. API Key Twitter itu sendiri memiliki suatu *consumer keys, consumer secret, access key*, dan *access secret*.

Proses ini menggunakan library tweepy, library sys, library jsonpickle. library tweepy adalah suatu API yang disediakan oleh pihak Twitter untuk dapat mengakses dan mengambil data-data yang ada di dalam Twitter menggunakan bahasa pemrograman Phyton. Library sys adalah library yang menyediakan layanan akses ke variabel dan fungsi yang berinteraksi kuat dengan interpreter. Library jsonpickle adalah library pada Python yang berguna untuk konversi dua arah objek python kompleks dan json.

2.2 Kajian Integrasi Topik Dengan Al-Quran

Suatu Kecelakaan dapat menyebabkan kehilangan nyawa pada korban, hal ini dapat dikategorikan sebagai pembunuhan yang merupakan suatu hal yang dilarang dalam Islam. Sebagaimana firman Allah SWT dalam Q.S. Al-Maidah ayat 32 yang berbunyi:

مِنْ أَجْلِ ذٰلِكَ كَتَبْنَا عَلَىٰ بَنِي إِسْرَائِيلَ أَنَّهُ مَنْ قَتَلَ نَفْسًا نَفْسٍ أَوْ فَسَادٍ فِي الْأَرْضِ فَكَأَنَّمَا قَتَلَ النَّاسَ جَمِيعًا وَمَنْ أَحْيَاهَا فَكَأَنَّمَا أَحْيَا النَّاسَ جَمِيعًاوَ لَقَدْ جَاءَتْهُمْ رُسُلُنَابِالْبَيِّنَاتِ ثُمَّ إِنَّ كَثِيرًا مِنْهُمْ بَعْدَ ذٰلِكَ فِي الْأَرْضِ لَمُسْرِفُونَ

"Oleh karena itu Kami tetapkan (suatu hukum) bagi Bani Israil, bahwa: barangsiapa yang membunuh seorang manusia, bukan karena orang itu (membunuh) orang lain, atau bukan karena membuat kerusakan dimuka bumi, maka seakan-akan dia telah membunuh manusia seluruhnya. Dan barangsiapa yang memelihara kehidupan seorang manusia, maka seolah-olah dia telah memelihara kehidupan manusia semuanya. Dan sesungguhnya telah datang kepada mereka rasul-rasul Kami dengan (membawa) keterangan-keterangan yang jelas, kemudian banyak di antara mereka sesudah itu sungguh-sungguh melampaui batas dalam berbuat kerusakan dimuka bumi."

Menurut hukum Islam dan berdasarkan uraian tersebut pembunuhan terbagi menjadi tiga jenis yakni pembunuhan sengaja, pembunuhan semi sengaja dan pembunuhan tidak sengaja mengenai komponen-komponen dari Pembunuhan kesalahan atau sengaja yaitu terdapat tindakan yang mengakibatkan kematian, tindakan tersebut terjadi akibat kesalahan dan Terdapat hubungan antara sebab akibat dari Tindakan tersebut. Perbuatan yang mengakibatkan kematian dapat digolongkan sebagai tindakan yang tidak disengaja apabila pelaku melakukannya secara tidak sengaja atau karena kelalaiannya. Akan tetapi, tidak dengan perbuatan seperti, membakar rumah orang lain, dengan sengaja membentuk lubang di pinggir jalan, melempar batu ke jalan dan lainnya. Komponen kedua yakni, kesalahan itu adalah tindakan yang berkaitan erat antara pembunuhan kesalahan dengan

pembunuhan lainnya. Dalam melakukan kesalahan memang tidak ada sanksi secara khusus. Namun, sanksi akan diberikan, jika menyebabkan kemudharatan atau kerugian untuk orang lain. Syariat islam menjelaskan bahwa ukuran kesalahan yakni berupa kelalaian atau kurangnya hati-hati atau perasaan yang meyakini bahwa tidak akan terjadi apapun. Dengan demikian, kesalahan tersebut dapat ada disebabkan kecerobohan yang menyebabkan kerugian atau kematian orang lain. Komponen ketiga, yaitu terdapat hubungan sebab akibat antara kesalahan dengan kematian, yang berarti akibat yang ditimbulkan dari kesalahan pelaku adalah kematian korban dan dapat dikatakan bahwa sebab dari kematian korban merupakan kerusakan yang ditimbulkan oleh pelaku.

2.3 Kajian Topik Dengan Teori Pendukung

Analisis yang akan dilakukan pada penelitian ini adalah menerapkan algoritma BIRCH untuk mengelompokan *big data*. *Big data Clustering* dengan metode hirarki merupakan langkah penyelesaian yang efektif untuk menyelesaikan permasalahan dalam hal waktu eksekusi di dalam *big data*.

Proses *cluster* yang dilakukan pada penelitian ini menggunakan Teknik *Clustering* BIRCH. *Clustering* dilakukan terhadap *dataset* twitter kecelakaan yang diambil dari Kaggle dan crawling dari twitter. Data yang digunakan bertipe campuran *string* dan *numeric*. Langkah pertama yang dilakukan yakni *preprocessing*, data tweet yang akan diolah akan melalui tahapan-tahapan seperti, proses *case folding*, proses *filtering*, proses *tokenizing*, proses *stemming* dan dilanjutkan melakukan proses *feature selection* dengan menggunakan metode TF-IDF setelah tahapan ini selesai barulah masuk ke dalam proses *Clustering* hirarki.

Pada proses *Clustering* hirarki menggunakan algoritma BIRCH untuk mengetahui plot dari hasil *Clustering*, proses algoritma BIRCH disini memanfaatkan library BIRCH yang sudah ada dalam python dengan menentukan nilai *threshold*, n_*cluster* serta *branching factor*. Pada proses akhir, akan dilakukan evaluasi terhadap validitas dari hasil *Clustering* dengan menggunakan *silhouette coefficient*, semakin mendekati satu maka hasil *cluster* semakin baik.

Pada teori pendukung menjelaskan teori yang dipakai dalam penelitian ini, sebagaimana dalam penelitian ini menggunakan algoritma BIRCH dalam pemrosesan data. Dijelaskan bahwa algoritma BIRCH merupakan salah satu Teknik dalam metode *Clustering* untuk pengelompokan data dalam ukuran yang besar atau lebih sering disebut dengan big data. Dalam proses penelitian ini juga melibatkan *Text Mining* yang merupakan terapan dari *Data Mining* yang telah dijelaskan dalam teori pendukung. Untuk data yang dipakai mengenai kecelakaan yang telah diuraikan dalam teori pendukung. Dari sini dapat disimpulkan bahwa setiap teori pendukung dalam penelitian ini memiliki peranan penting untuk mendukung proses penelitian ini agar berjalan dengan baik dan lancar.

BAB III
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Pendekatan dalam penelitian ini memanfaatkan metode eksperimen. Metode eksperimen merupakan metode analisis data dari penelitian kuantitatif yang bertujuan untuk menguji pengaruh hubungan sebab akibat. Oleh karena itu, penulis memanfaatkan metode eksperimen dikarenakan sesuai dengan rumusan masalah pada penelitian ini yakni mendapatkan hasil pengelompokan data tweet kecelakaan dengan algoritma BIRCH.

3.2 Data dan Sumber Data

Jenis data yang dimanfaatkan pada penelitian ini yakni data sekunder. Data ini mencakup data tweet kecelakaan pada April 2019 sampai April 2020 dengan format file *.csv yang didapatkan dari kaggle sebagai *data 1* dan Data primer yang diperoleh langsung dengan cara *crawling* data dari twitter sebagai *data 2*.

3.3 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini yaitu:

1. Mengumpulkan literatur baik meliputi paper, buku, jurnal dan penelitian lain yang berkaitan dengan *Clustering* big data, algoritma BIRCH, *Text Mining* dan data kecelakaan.
2. Mengumpulkan *dataset* yang diperoleh secara online melalui Kaggle serta Twitter untuk digunakan dalam penelitian.
3. Validasi *dataset* yaitu melakukan validasi terhadap data yang akan dimanfaatkan dalam proses penelitian.

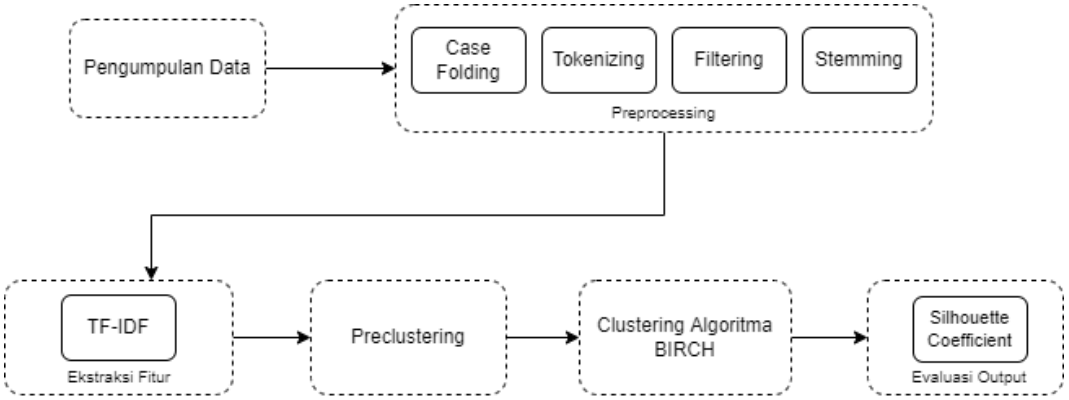
3.4 Instrumen Penelitian

Instrumen penelitian adalah alat untuk mengumpulkan data yang berguna untuk mengukur fenomena alam maupun sosial yang sedang diselidiki (Sugiyono, 2014). Pada penelitian ini menggunakan instrumen penelitian berupa dokumentasi yang diperoleh dari salah satu platform digital kumpulan *dataset* yakni Kaggle. Untuk memperlancar dan memudahkan proses penelitian ini dibantu dengan menggunakan komputer dengan spesifikasi sebagai berikut:

1. Spesifikasi Hardware
    a. Processor Intel Core i7
    b. RAM 16 GB
    c. Resolusi monitor 1920 x 1080 pixel
    d. Disk 476 GB
2. Spesifikasi Software
    a. Sistem operasi windows 11
    b. Python dengan software interaktif yakni jupyter notebook

3.5 Teknik Analisis Data

Pada penelitian ini memanfaatkan algoritma BIRCH untuk pemrosesan data yang dilaksanakan sesuai langkah-langkah berikut:



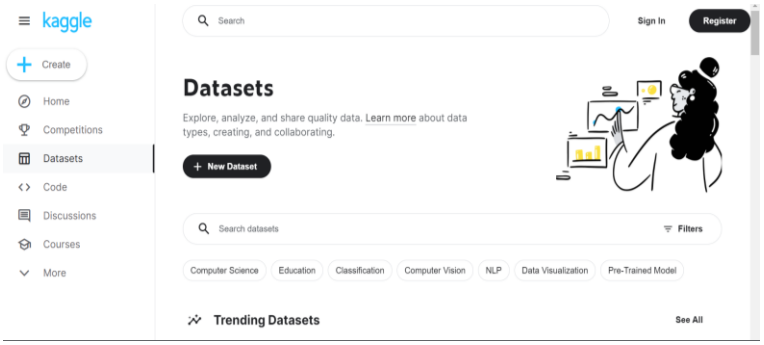
Gambar 3.1 Alur Teknik Analisis Data

Langkah pertama yang dilakukan dalam penelitian ini adalah pengumpulan data dari *platform online* yaitu Kaggle. Langkah kedua, melakukan *preprocessing* yang terdiri dari empat tahapan yaitu *Case folding, Tokenizing, Filtering,* dan *Stemming*. Langkah ketiga yakni melakukan ekstraksi fitur dengan menggunakan metode TF-IDF untuk mempermudah dalam proses *clustering*. Langkah keempat, melakukan proses *pre-clustering* yaitu menentukan nilai *Branching Factor*, *n-cluster*, dan *Threshold*. Selanjutnya, proses yang dilakukan adalah pengelompokan data menggunakan algoritma BIRCH. Setelah proses *clustering* akan menghasilkan output yang kemudian dievaluasi untuk mengetahui tingkat akurasi output yang dihasilkan menggunakan metode yang digunakan. Berikut merupakan uraian untuk menjelaskan lebih dalam tahapan-tahapan yang dilakukan dalam penelitian.

3.5.1 Pengumpulan Data

Penelitian ini menggunakan data postingan pengguna Twitter tentang kecelakaaan yang diperoleh dari situs kumpulan dataset yakni Kaggle dalam kurun waktu April 2019 sampai dengan April 2020 sebagai data 1 dan crawling data dari twitter dengan *query* “kecelakaan” dengan bantuan Bahasa pemrograman python.

Data yang diperoleh berupa data teks yang kemudian diolah terlebih dahulu melalui tahapan-tahapan sebelum melakukan proses *clustering*.



Gambar 3.2 Halaman Awal Kaggle

3.5.2 Pre-Processing

Tahapan yang perlu dilakukan sebelum dilakukannya proses *Clustering* yakni praproses agar data berubah menjadi bentuk matriks dan memudahkan untuk proses *Clustering*. Berdasarkan Gambar 3.3 data yang akan diproses harus melalui tahapan pra-proses sebagai berikut:

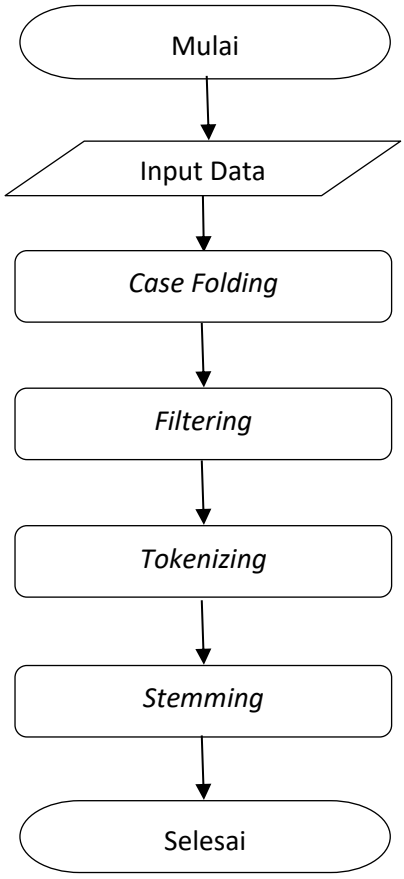
1. Proses *Case folding*

   Text twitter terdiri dari banyak karakter seperti isi tweet itu sendiri, *emoticon*, @, #, dan tag lainnya. Karakter tersebut tidak diperlukan dalam proses *Clustering*.

   Oleh karena itu, diperlukan proses *case folding* untuk merubah text menjadi *lower text* beserta karakter-karakter yang tidak diperlukan dalam proses *tokenizing* sehingga, hanya diperoleh isi dari tweet tersebut untuk digunakan pada proses selanjutnya.

2. Proses *Tokenizing*

Pada proses ini, teks tweet hasil proses *case folding* kemudian dilakukan proses *tokenizing* yakni proses pemotongan atau pemisahan *string* yang dimasukkan berdasarkan tiap kata penyusunnya



Gambar 3.3 Flowchart *Preprocessing*

2. Proses *Filtering*

Proses *filtering* adalah proses pengumpulan kata penting dari hasil proses *tokenizing*. Proses ini memanfaatkan algoritma *stoplist* atau *stopword* yaitu membuang kata yang dirasa kurang penting atau dapat menggunakan *wordlist* yaitu menyimpan kata penting.

Pada proses penelitian ini algoritma yang digunakan dalam proses *filtering* yakni *stopword* untuk membuang kata-kata yang tidak deskriptif seperti “yang”, “di”, “dan”, “dari” dan seterusnya.

3. Proses *Stemming*

Pada proses ini, teks tweet yang telah melalui proses *filtering* kemudian dicari akar katanya. Tahapan ini merupakan proses pengumpulan berbagai macam kata ke dalam suatu representasi yang sama.

Proses *stemming* kebanyakan digunakan dalam teks berbahasa inggris dan lebih sulit untuk diimplementasikan pada teks berbahasa Indonesia. Hal ini disebabkan karena Bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen. Namun, karena data tweet yang kita pakai tidak hanya berbahasa Indonesia maka harus proses *stemming* ini dilakukan.

3.5.3 Ekstraksi Fitur

Pada penelitian ini menggunakan salah satu metode dari ekstraksi fitur yakni TF-IDF yang merupakan metode yang terdiri dari *Term frequency* dan *Inverse Document frequency* dengan menghitung bobot berdasarkan persamaan (2.3). Proses ini dilakukan setelah proses *preprocessing* dilewati, pada tahap ini digunakan untuk mempermudah proses pengelompokan data tweet tersebut dengan memberikan bobot dari setiap kata pada semua kalimat yang telah melalui proses *preprocessing*. Setelah proses pemberian nilai telah dilakukan maka *dataset* dapat digunakan untuk proses selanjutnya.

3.5.4 *Pre-Clustering*

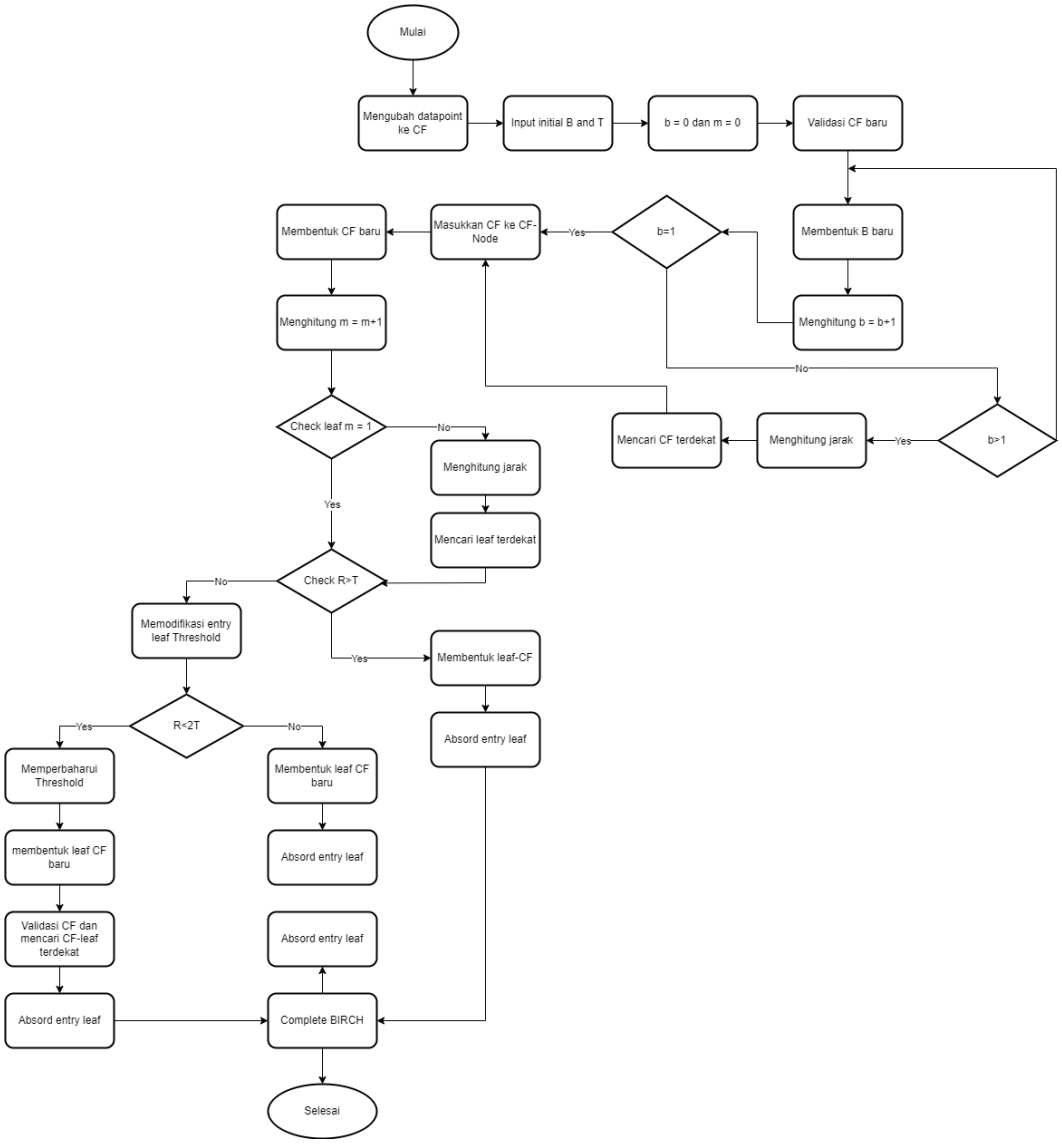
Proses yang dilakukan yakni *pre-Clustering* untuk mengubah setiap data ke format CF yang kemudian digunakan untuk setiap proses komputasi. Kemudian, menentukan parameter yang akan digunakan dalam pembentukan CF-Tree dan dilanjutkan dengan membentuk CF-Tree inisialisasi yang merupakan kumpulan *subcluster* yang tergabung secara kolektif yang sesuai kedekatan jarak. Setiap *subcluster* berisi informasi yang telah diringkas dalam bentuk CF.

Berikut merupakan parameter input yang digunakan dalam Algoritma BIRCH:

1. *Branching Factor* (B), yaitu jumlah maksimal *child* pada *node non leaf*.
2. *Threshold* (T), yaitu batas atas yang menentukan suatu data point dimasukkan ke dalam CF-Tree.

3.5.5 Proses Clustering

Pada proses ini menggunakan salah satu metode *clustering* yakni algoritma BIRCH. Berikut merupakan flowchart dari algoritma BIRCH (Fanny Ramadhani, 2019):



Gambar 3.4 Flowchart Algoritma BIRCH

Berikut merupakan uraian dari flowchart algoritma BIRCH:

1. Mengubah data point ke dalam bentuk CF menggunakan rumus CF = (N, LS, SS).
2. Kemudian, data yang telah diubah ke dalam bentuk CF akan digabungkan dengan menggunakan CF-Tree. Pada proses ini juga menentukan jumlah B (*Branching*) yang harus dimasukkan.

3. Untuk membantu perhitungan, dibutuhkan dua variabel tambahan yaitu *m* dan *b*. Variabel *b* digunakan untuk menghitung jumlah cabang pada *leaf* yang sudah terbentuk.
4. Untuk setiap *subcluster* CF yang masuk, BIRCH akan membandingkan lokasi dari tiap CF yang sudah terbentuk di root node, menggunakan *euclidean distance*. BIRCH meneruskan *subcluster* CF masuk ke *root node* CF paling dekat dengan *subcluster* CF.
5. *Subcluster* kemudian turun ke *node child non-leaf* dari simpul CF dipilih. BIRCH membandingkan lokasi *subcluster* dengan *subcluster* setian *leaf*. BIRCH untuk sementara melewati *subcluster* yang masuk ke *leaf* yang paling dekat dengan *subcluster* CF yang masuk.
6. Setelah menemukan *leaf* terdekat, maka *subcluster* akan mengecek jika radius dari *leaf* yang dipilih termasuk *subcluster* yang baru tidak melebihi *Threshold* T, maka *subcluster* akan masuk ke *leaf* (*leaf*-CF(modif)) tersebut.
7. Tetapi bila perubahan nilai *threshold* T tetap membuat nilai radius melebihi *threshold* T, maka *leaf* (*leaf*-CF(modif)) baru terbentuk, terdiri dari *subcluster* yang masuk saja. CF induk diperbaharui ke root untuk titik data baru.
8. Kemudian BIRCH akan melakukan fase 3, yaitu global *clustering*.

Pada proses ini seluruh *leaf* di *cluster* kan dengan menggunakan algoritma *Clustering* hirarki. Penggabungan *subcluster* dibuat untuk memperoleh *cluster* yang lebih besar dan setiap penggabungan diiringi dengan perubahan informasi CF. Secara umum, algoritma dari *Clustering* hirarki yaitu sebagai berikut:

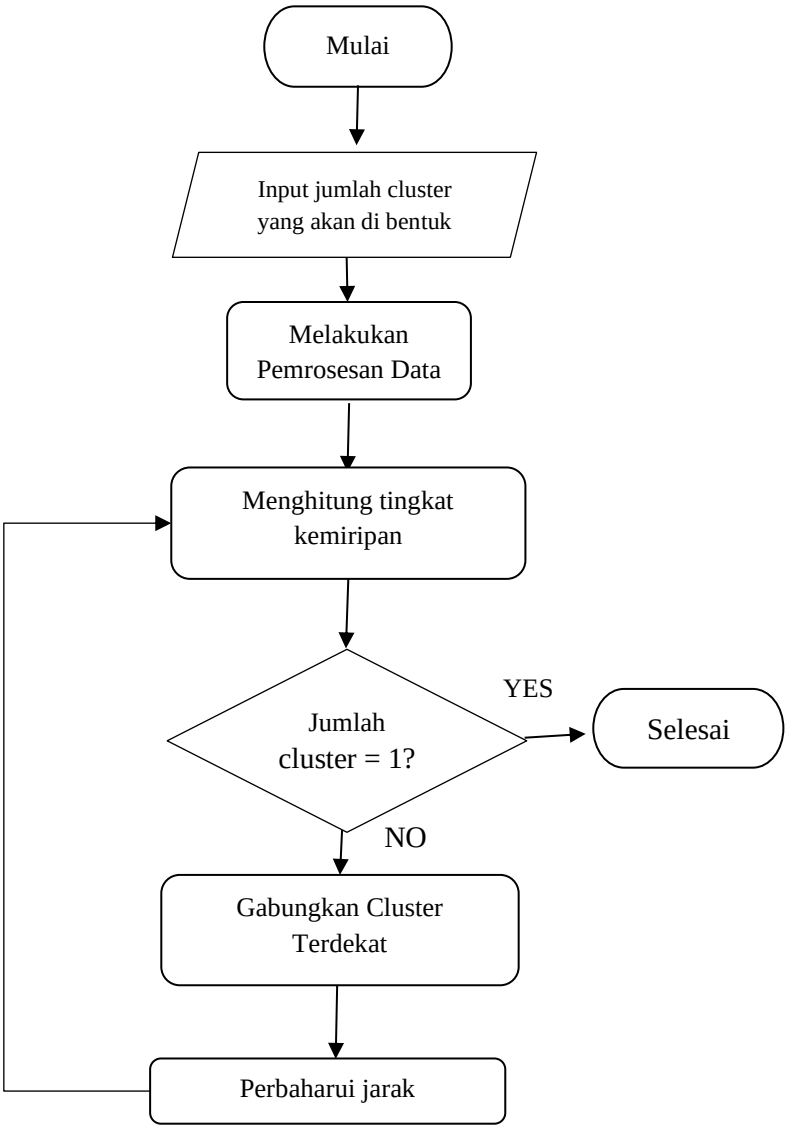
1. Menentukan nilai k dan B sebagai jumlah *cluster* yang akan dibentuk dan ini merupakan salah satu parameter yang diinput oleh user.
2. Menginputkan semua *subcluster* yang diperoleh dari hasil proses pre-*Clustering*. Setiap *subcluster* dianggap sebagai *cluster*. Jika N = jumlah *subcluster* dan n = jumlah *cluster*, maka $n = N$.
3. Perhitungan jarak antar *cluster*. Pada penelitian ini akan digunakan persamaan jarak *average inter cluster distance* (D2) pada persamaan (2.8). Pemilihan persamaan jarak ini berdasarkan penelitian yang telah dilakukan dan membuktikan bahwa penggunaan jarak inter *cluster* mampu menghasilkan *cluster* dengan kualitas yang lebih baik daripada dengan penetapan jarak yang lainnya (Tian Zhang, 1996).
4. Mencari 2 *cluster* yang memiliki jarak antar *cluster* yang paling minimum kemudian digabungkan.
5. Mengecek radius ketika digabungkan dengan *leaf*, jika radius tidak melebihi nilai *threshold*, *subcluster* dapat bergabung dengan *Leaf* tersebut. Jika tidak, akan dilakukan modifikasi nilai *threshold* menjadi dinamis, bila setelah dimodifikasi nilai radius pada *subcluster* tidak melebihi ambang maka akan diperbaharui nilai *threshold* dan *subcluster* bergabung dengan *leaf* tersebut. Bila tidak, *leaf* baru akan terbentuk.
6. Jika $n > b$, kembali ke langkah 3.
7. Jika $n = b$, maka proses akan diselesaikan.

3.3.6 Validasi Output

Pada tahap ini akan dilakukan evaluasi dari output yang diperoleh pada proses *clustering* algoritma BIRCH. Evaluasi ini bertujuan untuk mengetahui baik

atau tidaknya *cluster* yang dihasilkan dalam penelitian ini menggunakan *silhouette coefficient* dengan menghitung persamaan (2.16) untuk mengetahui tingkat akurasinya. Semakin hasil *Silhouette Coefficient* lebih mendekati 1,00 maka tingkat akurasinya semakin baik.

Berikut merupakan proses *clustering* pada sistem yang disajikan ke dalam flowchart:



Gambar 3.5 Flowchart Umum *Clustering*

BAB IV
HASIL DAN PEMBAHASAN

4.1 Proses Pengumpulan Data

4.1.1 Pengumpulan Data dari Kaggle

Dataset yang digunakan adalah data tweet kecelakaan yang diperoleh secara online melalui Kaggle dengan situs https://www.kaggle.com/datasets yang disediakan oleh Kaggle. Dataset ini terdiri dari 1850 data dengan 8 atribut yang terdiri dari *tweet_id, full_text, username, is_accident, created_at, updated_at, status,* dan *index* yang berisi tweet tentang kecelakaan yang terjadi antara April 2019 sampai dengan April 2020. Berikut merupakan tampilan dari data tweet kecelakaan yang diperoleh dari Kaggle.

	tweet_id	full_text	username	is_accident	created_at	updated_at	status	index
0	1115425444663181313	WNI Korban Tewas Kecelakaan Bus di Malaysia Be...	bestprofitsby	1	21/9/2020 12:31:14	21/9/2020 05:43:40	1	1
1	1115609216125816833	20.35 WIB #Tol_Japek Karawang Timur KM 51 - KM...	PTJASAMARGA	1	21/9/2020 12:31:14	21/9/2020 05:43:48	1	2
2	1306782023877570560	Help RT #zonaba #zonauang \nGua temen ba nya c...	Cibbbl	1	20/9/2020 17:06:08	21/9/2020 05:44:07	1	3
3	1116268160003428352	Kecelakaan Maut Bus Eka di Ring Road Sragen ht...	terasid	1	21/9/2020 12:31:14	21/9/2020 05:46:39	1	4
4	1114010329313206272	Tewaskan 346 Orang dalam 2 Kecelakaan, Boss Bo...	PanritaNews	1	21/9/2020 12:31:14	21/9/2020 05:41:42	1	5

Gambar 4.1 Data Tweet Kecelakaan Kaggle

4.1.2 *Crawling* Data Twitter

Crawling data twitter digunakan dengan membutuhkan API Key yang terdiri dari *Consumer keys*, *consumer secret*, *access key*, dan *access secret* tersebut digunakan untuk mengakses data Twitter yang dibutuhkan oleh penulis.

Sebelum masuk ke proses berikutnya terlebih dahulu dilakukan proses autentifikasi untuk menunjukkan bahwa API key yang kita masukkan sudah benar. Seperti ditunjukkan pada gambar di bawah ini.

```
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)

api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)

try:
    api.verify_credentials()
    print("Authentication OK")
except:
    print("Error during authentication")
```

```
Authentication OK
```

Gambar 4.2 Proses Autentifikasi API Key

Proses selanjutnya yakni memasukkan kode proses untuk menentukan kata kunci. Pada proses ini yang dilakukan yakni memasukkan variabel – variabel yang diperlukan dalam proses *crawling* seperti qry yakni untuk memasukkan kata kunci yang akan dicari. fName yakni untuk memasukkan nama file hasil *crawling* data. *maxTweets* yakni untuk memasukkan jumlah tweets yang dibutuhkan. *tweetsPerQry* yakni ketentuan maksimal tweets setiap qry yang telah ditentukan oleh twitter itu sendiri. Selanjutnya penulis melakukan proses *crawling* data dari twitter dengan menggunakan kode program pada jupyter notebook.

Setelah proses tersebut dilakukan dan berhasil maka akan diperoleh file json untuk diolah dalam proses berikutnya. Karena data yang kita gunakan dalam penelitian ini harus berekstensi *.csv maka langkah selanjutnya yakni mengconvert file json yang diperoleh ke dalam file csv.

Proses ini menggunakan library xlrd, library xlwt, library json, dan library pandas. Pada proses ini dilakukan convert data dari file json ke dalam bentuk file yang diperlukan. Untuk proses kali ini akan mengubah file json menjadi file berekstensi csv untuk mempermudah proses berikutnya. Setelah proses tersebut dilakukan dan berhasil maka akan diperoleh file csv untuk diolah dalam proses berikutnya. Hasil convert data kecelakaan ditampilkan seperti gambar berikut ini.

	Unnamed: 0	created_at	id	id_str	text	truncated	entities	metadata	
0	0	2022-05-19 13:57:10+00:00	1527287250207182849	1527287250207182848	RT @jawapos: Mayoritas Kecelakaan Lalu Lintas ...	False	{'hashtags': [], 'symbols': [], 'user_mentions...	{'iso_language_code': 'in', 'result_type': 're...	href="http://twitte
1	1	2022-05-19 13:57:08+00:00	1527287242125103104	1527287242125103104	RT @yohansnuna_: Kim saeron....\nIni keras ban...	False	{'hashtags': [], 'symbols': [], 'user_mentions...	{'iso_language_code': 'in', 'result_type': 're...	href="http://twitte
2	2	2022-05-19 13:56:55+00:00	1527287185560698880	1527287185560698880	RT @wifenyaYoongi: Kenapa kalau ada kasus kece...	False	{'hashtags': [], 'symbols': [], 'user_mentions...	{'iso_language_code': 'in', 'result_type': 're...	href="http://twitte
3	3	2022-05-19 13:56:45+00:00	1527287143508238337	1527287143508238336	RT @yohansnuna_: Kim saeron....\nIni keras ban...	False	{'hashtags': [], 'symbols': [], 'user_mentions...	{'iso_language_code': 'in', 'result_type': 're...	href="http://twitte
4	4	2022-05-19 13:56:02+00:00	1527286964466368512	1527286964466368512	Kecelakaan Bus Pariwisata Tewaskan 14 Orang ht...	False	{'hashtags': [{'text': 'TimnasDay', 'indices':...	{'iso_language_code': 'in', 'result_type': 're...	<a href="h

Gambar 4.3 Hasil Crawling Dari Twitter

4.2 *Preprocessing*

Tahap *preprocessing* ini terdiri dari beberapa tahapan untuk menjadikan kalimat pada tweet menjadi Bahasa yang baku, karena tidak sepenuhnya tweet hasil dari *crawling* menggunakan kata baku. Selain itu proses ini berguna untuk menghilangkan beberapa bagian dari kalimat yang tidak berguna. Proses *preprocessing* dikerjakan dengan menggunakan bantuan dari library pada Bahasa pemrograman Python 3. Untuk mengerjakan proses *preprocessing* terdapat 4 tahapan proses untuk memperoleh hasil yang maksimal, yakni:

1. *Case folding*

   Proses *case folding* digunakan untuk mengurangi atau membersihkan data tweet dari kata atau kalimat yang tidak diperlukan seperti tanda baca, Unicode, dan lain-lain. Pada proses *case folding* terdapat tiga tahapan yang akan dilakukan oleh system untuk mendapatkan hasil yang optimal, sebagai berikut:

   a. Merubah huruf besar menjadi huruf kecil semua

   b. Menghilangkan kelebihan spasi

c. Menghilangkan tanda baca

Berikut merupakan kode program untuk melakukan proses *case folding* yang ditunjukkan sebagai berikut.

```
#Proses case folding

import re
def casefolding(full_text):
    full_text = full_text.lower()
    full_text = full_text.strip(" ")
    full_text = re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",full_text)
    return full_text
data['full_text'] = data['full_text'].apply(casefolding)
data.head()
```

Gambar 4.4 Proses *Case Folding*

Pada proses ini menggunakan library `re` yaitu untuk menentukan atau memeriksa string tertentu cocok dengan ekspreksi reguler yang diberikan. Setelah kode program dijalankan maka diperoleh hasil *case folding* seperti di bawah ini.

Tabel 4.1 Contoh Data Hasil *Case folding*

Tweet Sebelum *Case folding*	**Tweet Sesudah *Case folding***
Kecelakaan Maut Bus Eka di Ring Road Sragen https://t.co/ZgJGXEmYPD	kecelakaan maut bus eka di ring road sragen
Tewaskan 346 Orang dalam 2 Kecelakaan, Boss Boeing Minta Maaf https://t.co/wLRhFy8oYE	tewaskan 346 orang dalam 2 kecelakaan boss boeing minta maaf
Rasa takut akan kematian adalah yang paling dibenarkan dari semua ketakutan, karena tidak ada risiko	rasa takut akan kematian adalah yang paling dibenarkan dari semua ketakutan karena tidak ada risiko

kecelakaan bagi seseorang yang sudah mati.	kecelakaan bagi seseorang yang sudah mati

2. *Tokenizing*

Pada proses *tokenizing* berguna untuk pemotongan atau pemisahan string yang dimasukkan berdasarkan tiap kata penyusunnya. Berikut merupakan proses pengimplementasian pada kode program seperti ditunjukkan pada gambar berikut ini.

```
#Proses tokenizing

def token(full_text):
    nstr = full_text.split(' ')
    dat = []
    a = -1
    for hu in nstr:
        a = a + 1
        if hu == '':
            dat.append(a)
    p = 0
    b = 0
    for q in dat:
        b = q - p
        del nstr[b]
        p = p +1
    return nstr
data['full_text'] = data['full_text'].apply(token)
data.head()
```

Gambar 4.5 Proses *Tokenizing*

Setelah kode program dijalankan dan berhasil maka diperoleh hasil dari proses penerapan *tokenizing* ke dalam data tweet kecelakaan seperti ditunjukkan pada tabel dibawah ini.

Tabel 4.2 Contoh Data Hasil *Tokenizing*

Tweet Sebelum *Tokenizing*	**Tweet Sesudah *Tokenizing***
kecelakaan maut bus eka di ring road sragen	[kecelakaan, maut, bus, eka, di, ring, road, sragen]

tewaskan 346 orang dalam 2 kecelakaan boss boeing minta maaf	[tewaskan, 346, orang, dalam, 2, kecelakaan, boss, boeing, minta, maaf]
rasa takut akan kematian adalah yang paling dibenarkan dari semua ketakutan karena tidak ada risiko kecelakaan bagi seseorang yang sudah mati	[rasa, takut, akan, kematian, adalah, yang, paling, dibenarkan, dari, semua, ketakutan, karena, tidak, ada, risiko, kecelakaan, bagi seseorang, yang, sudah mati]

3. *Filtering*

Proses *filtering* digunakan untuk membuang kata yang dirasa kurang penting atau biasa disebut dengan kata penghubung seperti “yang”, “di”, “dan”, “dari” dan seterusnya. Pada proses *filtering* menggunakan algoritma stoplist atau stopword untuk penghapusan kata-kata yang tidak deskriptif.

Dalam proses *filtering*, Langkah pertama yang harus dilakukan yakni mendefinisikan kata-kata yang akan terhapus ketika proses ini dijalankan. Seluruh kata-kata yang sudah didefinisikan tersimpan dalam sebuah file dengan nama full_text. Pada proses ini dibantu dengan library `nltk` yang terdapat pada Bahasa pemrograman python 3 yang terlebih dahulu dilakukan proses install library `nltk` menggunakan pip.

Setelah proses install berhasil, Langkah selanjutnya yakni mendeklarasikan library nltk sebelum masuk ke dalam proses *filtering*. Langkah berikutnya yakni proses *filtering* dengan mengimplementasikan algoritma stopword pada kode program seperti gambar di bawah ini.

```
def stopword_removal(full_text):
    filtering = stopwords.words('indonesian','english')
    x = []
    data = []
    def myFunc(x):
        if x in filtering:
            return False
        else:
            return True
    fit = filter(myFunc, full_text)
    for x in fit:
        data.append(x)
    return data
data['full_text'] = data['full_text'].apply(stopword_removal)
data.head()
```

Gambar 4.6 Proses *Filtering*

Contoh proses penerapan *filtering* dengan menggunakan kode program yang telah dilakukan dapat dilihat pada tabel di bawah ini.

Tabel 4.3 Contoh Data Hasil *Filtering*

Tweet Sebelum *Filtering*	**Tweet Sesudah *Filtering***
[kecelakaan, maut, bus, eka, di, ring, road, sragen]	[kecelakaan, maut, bus, eka, ring, road, sragen]
[tewaskan, 346, orang, dalam, 2, kecelakaan, boss, boeing, minta, maaf]	[tewaskan, 346, orang, 2, kecelakaan, boss, boeing, minta, maaf]
[rasa, takut, akan, kematian, adalah, yang, paling, dibenarkan, dari, semua, ketakutan, karena, tidak, ada, risiko, kecelakaan, bagi, seseorang, yang, sudah, mati]	[takut, kematian, dibenarkan, ketakutan, risiko, kecelakaan, mati]

4. *Stemming*

Stemming merupakan proses pengambilan berbagai bentukan kata ke dalam suatu representasi yang sama atau dapat diartikan sebagai proses

penghapusan kata imbuhan dari setiap kata, baik kata imbuhan yang berada di depan maupun di belakang kata. Pada proses ini dibantu dengan library sastrawi yang terdapat pada Bahasa pemrograman python 3 yang terlebih dahulu dilakukan proses install library `sastrawi` menggunakan perintah pada *command prompt*.

Setelah proses instalasi selesai, langkah selanjutnya yakni mendeklarasikan library sastrawi yang digunakan dalam proses *stemming*. Selanjutnya proses pengimplementasian dari tahapan stemming pada kode program seperti ditunjukkan pada gambar berikut ini.

```
def stemming(full_text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    do = []
    for w in full_text:
        dt = stemmer.stem(w)
        do.append(dt)
    d_clean = []
    d_clean = " ".join(do)
    print(d_clean)
    return d_clean
data['full_text'] = data['full_text'].apply(stemming)

data.to_csv('data_twitter.csv', index = False)
data_clean = pd.read_csv('data_twitter.csv', encoding='latin1')
data.head()
```

Gambar 4.7 Proses *Stemming*

Setelah proses *stemming* dilakukan, maka hasil yang diperoleh dapat ditunjukkan pada tabel di bawah ini.

Tabel 4.4 Contoh Data Hasil *Stemming*

Tweet Sebelum *Stemming*	**Tweet Sesudah *Stemming***
[kecelakaan, maut, bus, eka, ring, road, sragen]	Celaka maut bus eka ring road sragen
[tewaskan, 346, orang, 2, kecelakaan, boss, boeing, maaf]	Tewas 346 orang 2 celaka boss boeing maaf

[takut, kematian, dibenarkan, ketakutan, risiko, kecelakaan, mati]	Takut mati benar takut risiko celaka mati

Setelah semua proses *preprocessing* dilakukan, maka hasil yang diperoleh disimpan ke dalam bentuk file baru yang kemudian digunakan sebagai dataset dalam proses pengelompokkan

4.3 Ekstraksi Fitur

Pada tahapan ini, proses pertama yang dilakukan yakni mengubah dataset menjadi suatu representasi vector dengan menggunakan library yang sudah disediakan oleh python yaitu library Vectorizer.

```
#Proses TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer

# banyaknya term yang akan digunakan,
# di pilih berdasarkan top max_features
# yang diurutkan berdasarkan term frequency seluruh corpus
max_features = 1000

# Feature Engineering
print ("------- TF-IDF on Tweet data -------")

tf_idf = TfidfVectorizer(max_features=max_features, binary=True)
tfidf_mat = tf_idf.fit_transform(data['full_text']).toarray()

print("TF-IDF ", type(tfidf_mat), tfidf_mat.shape)
```

Gambar 4.8 Proses Ekstraksi Fitur

Langkah pertama yang dilakukan yakni mendeklarasikan library yang akan digunakan dalam proses ekstraksi fitur. Kemudian menentukan max features untuk menentukan banyaknya term yang digunakan barulah masuk ke proses perhitungan atau pembentukan vector dengan menggunakan TF-IDF. Dari proses TF-IDF diperoleh vector dengan ukuran *class numpy.ndarray* (1855, 1000) untuk data tweet kecelakaan dari Kaggle sedangkan untuk *data 2* yang dipakai yakni (2000, 1000). Berikut merupakan output hasil teratas dari proses TF-IDF menggunakan python.

Out[10]:

	term	rank
224	celaka	149.079965
916	tol	71.978699
427	jl	55.135754
997	yg	49.659096
399	jalan	48.520107
...	...	...
104	ancol	1.475858
247	covid	1.444380
990	wiyoto	1.377579
989	wiyono	1.377579
387	ir	1.377579

1000 rows × 2 columns

Gambar 4. 9 Hasil Term Teratas Proses TF-IDF

Untuk mengetahui bagaimana TF-IDF berjalan, berikut merupakan contoh perhitungan TF-IDF secara matematik. Sebagai contoh penelitian menggunakan 3 komentar, sebagai berikut:

1. “celaka maut bus eka ring road sragen”
2. “tewas 346 orang 2 celaka boss boeing maaf”
3. “takut mati benar takut risiko celaka mati”

Kemudian dilakukan *preprocessing* untuk memperoleh jumlah kata baku dari 3 kalimat tersebut. Langkah selanjutnya yakni dari setiap kalimat akan ditampilkan menjadi sebuah vektor dengan elemen, sebagai contoh di bawah ini.

Tabel 4.5 Pembuatan *Word Vector*

	celaka	**maut**	**bus**	**eka**	**tewas**	**orang**	**maaf**	**takut**	**mati**	**benar**	**risiko**
1	1	1	1	1	0	0	0	0	0	0	0
2	1	0	0	0	1	1	1	0	0	0	0

3	1	0	0	0	0	0	0	2	2	1	1

Setelah kalimat diubah menjadi word vector, Langkah selanjutnya yakni menghitung menggunakan rumus TF-IDF untuk mendapatkan nilai yang berbobot. Proses yang dilakukan yakni dengan menghitung TF atau *Term frequency* terlebih dahulu.

Tabel 4.6 Proses Perhitungan TF (*Term frequency*)

	Doc1	**Doc2**	**Doc3**
Celaka	1	1	1
Maut	1	0	0
Bus	1	0	0
Eka	1	0	0
Tewas	0	1	0
Orang	0	1	0
Maaf	0	1	0
Takut	0	0	2
Mati	0	0	2
Benar	0	0	1
Risiko	0	0	1

Setelah melakukan perhitungan TF, Langkah selanjutya yakni melakukan proses DF atau *Document frequency* yang memperoleh hasil sebagai berikut.

Tabel 4.7 Proses Perhitungan DF (Document Frequency)

T(Term)	**DF (Document Frequency)**
Celaka	3

Maut	1
Bus	1
Eka	1
Tewas	1
Orang	1
Maaf	1
Takut	2
Mati	2
Benar	1
Risiko	1

Setelah proses TF dan DF selesai, langkah selanjutnya menghitung nilai IDF (*Inverse Document Frequency*) yakni menghitung nilai log hasil D atau jumlah dokumen kemudian dibagi dengan nilai DF (*Document Frequency*). Sehingga dihasilkan nilai perhitungan di bawah ini.

Tabel 4.8 Proses IDF (*Inverse Document Frequency*)

T(Term)	DF (Document Frequency)	D/DF	IDF (Inverse Document Frequency)
Celaka	3	1	log 1 = 0
Maut	1	3	log 3 = 0,477
Bus	1	3	log 3 = 0,477
Eka	1	3	log 3 = 0,477
Tewas	1	3	log 3 = 0,477
Orang	1	3	log 3 = 0,477
Maaf	1	3	log 3 = 0,477

Takut	2	1,5	log 1,5 = 0,176
Mati	2	1,5	log 1,5 = 0,176
Benar	1	3	log 3 = 0,477
Risiko	1	3	log 3 = 0,477

Setelah memperoleh nilai IDF (*Inverse Document Frequency*), Langkah selanjutnya yakni menghitung nilai TF-IDF. Seperti berikut:

Tabel 4.9 Proses Perhitungan TF-IDF

Q	TF			DF	D/DF	IDF	IDF+1	W=TF*(IDF+1)		
	Doc1	Doc2	Doc3					Doc1	Doc2	Doc3
Celaka	1	1	1	3	1	0	1	1	1	1
Maut	1	0	0	1	3	0,477	1,477	1,477	0	0
Bus	1	0	0	1	3	0,477	1,477	1,477	0	0
Eka	1	0	0	1	3	0,477	1,477	1,477	0	0
Tewas	0	1	0	1	3	0,477	1,477	0	1,477	0
Orang	0	1	0	1	3	0,477	1,477	0	1,477	0
Maaf	0	1	0	1	3	0,477	1,477	0	1,477	0
Takut	0	0	2	2	1,5	0,176	1,176	0	0	2,352
Mati	0	0	2	2	1,5	0,176	1,176	0	0	2,352
Benar	0	0	1	1	3	0,477	1,477	0	0	1,477
Risiko	0	0	1	1	3	0,477	1,477	0	0	1,477
Jumlah								4,431	4,431	7,658

Pada tabel dibawah ini menunjukkan hasil dari word vector yang sudah mendapatkan bobot.

Tabel 4.10 *Word Vector* Yang Sudah Di Bobotkan

	celaka	maut	bus	eka	tewas	orang	maaf	takut	mati	benar	risiko
1	1	1,477	1,477	1,477	0	0	0	0	0	0	0
2	1	0	0	0	1,477	1,477	1,477	0	0	0	0
3	1	0	0	0	0	0	0	2,352	2,352	1,477	1,477

4.4 Implementasi Algoritma BIRCH

Pada proses ini dilakukan implementasi algoritma BIRCH untuk pengelompokkan data tweet kecelakaan. Proses ini berjalan dengan menggunakan Bahasa pemrograman Python 3 dengan dibantu oleh library scikit-learn dalam proses pengklasteran, selain itu juga terdapat library matplotlib sebagai bantuan untuk memvisualisasikan data. Berikut merupakan kode program yang digunakan dalam pemrosesan data menggunakan algoritma BIRCH.

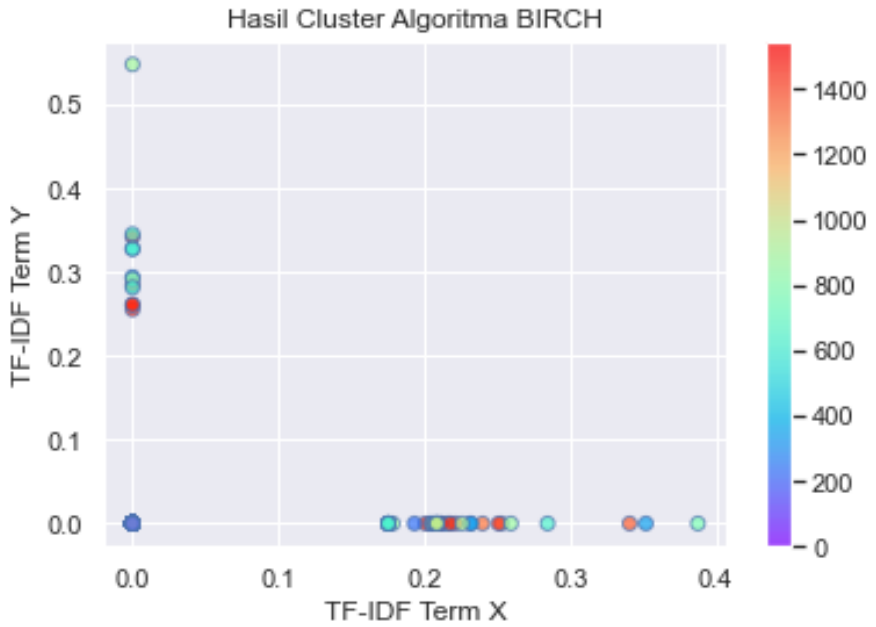
```
#Implementasi Algoritma BIRCH

from sklearn.cluster import Birch
import matplotlib.pyplot as plt

model = Birch(branching_factor = 50, n_clusters = None, threshold = 0.5)
model.fit(tfidf_mat)
labels = model.predict(tfidf_mat)
pred =model.predict(tfidf_mat )
plt.scatter(tfidf_mat [:, 0],tfidf_mat [:, 1], c = pred, cmap='rainbow', alpha=0.7, edgecolors='b')
plt.title('Hasil Cluster Algoritma BIRCH', size=12)
plt.xlabel('Sumbu X', size=12)
plt.ylabel('Sumbu Y', size=12)
plt.show()
```

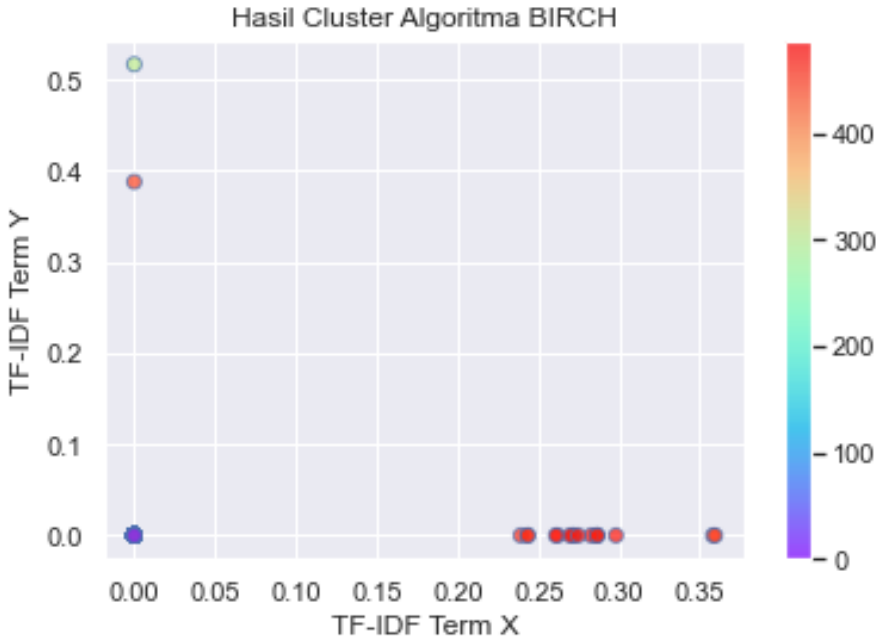
Gambar 4.10 Proses *Clustering* Algoritma BIRCH

Sebelum pemrosesan dijalankan perlunya untuk menentukan branching factor, threshold dan n clusters. Pada proses ini penulis menggunakan branching factor = 50, n_clusters = None, dan nilai threshold = 0,5. Library yang digunakan pada proses ini yakni library Birch serta matplotlib untuk memvisualisasikan cluster dalam bentuk plot. Dari hasil proses tersebut diperoleh cluster seperti ditunjukkan dalam gambar di bawah ini.



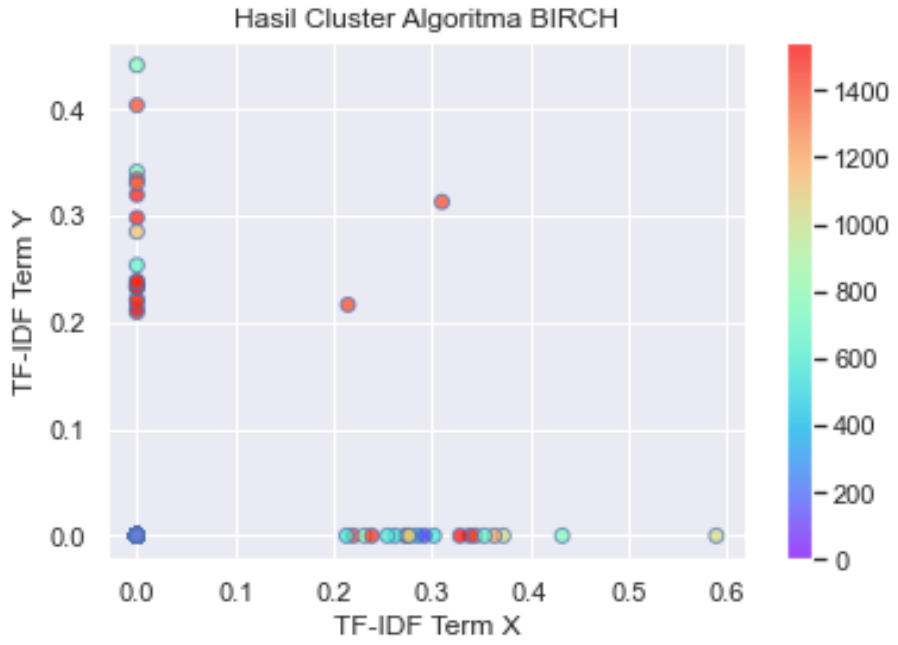
Gambar 4.11 Hasil Cluster *Data 1*

Dari gambar plot tersebut dihasilkan sebanyak 1545 cluster dari proses *clustering* algoritma BIRCH menggunakan data 1. Sedangkan, dari cluster yang dihasilkan dalam proses *clustering* algoritma BIRCH menggunakan data 2 dihasilkan sebanyak 487 cluster yang tersebar seperti ditunjukkan dalam gambar berikut.

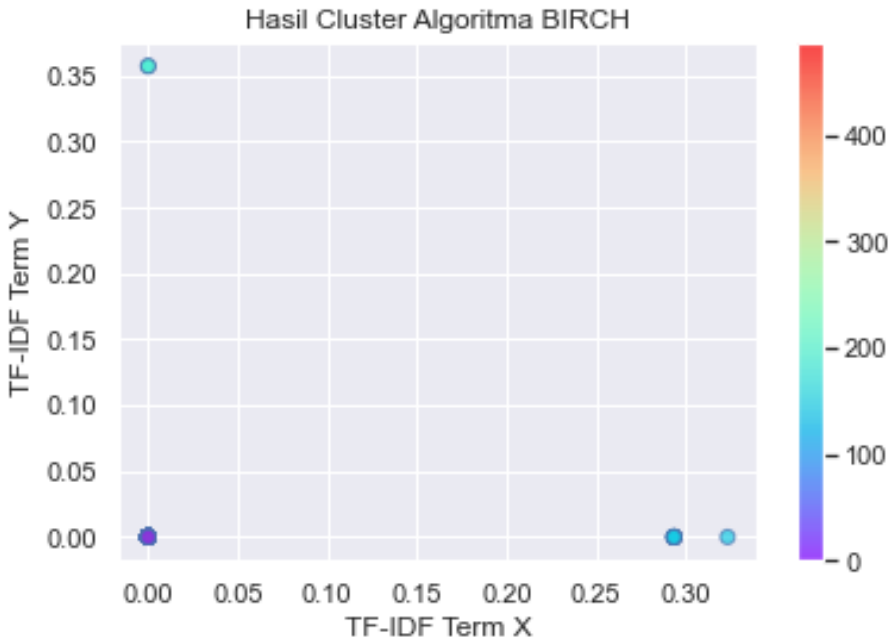


Gambar 4.12 Hasil Cluster Data 2

Pada gambar 4.11 dan 4.12 merupakan output cluster dengan mengunakan term 1 dan 2 sedangkan jika kita ingin mengetahui lebih lanjut mengenai cluster term lainnya maka pada kode `plt.scatter` masukkan 0 dan 1 dapat diganti dengan term yang kita inginkan. Contoh kita masukkan 20 dan 23 maka akan menghasilkan output plot sebagai berikut.



Gambar 4. 13 Hasil Output Algoritma BIRCH Berdasarkan Term Data 1



Gambar 4. 14 Hasil Output Algoritma BIRCH Berdasarkan Term Data 2

Dari gambar diatas maka dapat ditarik kesimpulan bahwa cluster yang dihasilkan tidak selalu berada dalam sumbu yang sama akan tetapi dapat berada pada titik yang berbeda sesuai dengan nilai bobot kesamaannya.

Hasil implementasi algoritma BIRCH ini maka diperoleh suatu cluster yang memiliki kesamaan dalam setiap anggota dalam satu cluster tersebut. Berikut merupakan contoh hasil implementasi algoritma BIRCH dalam satu cluster.

Tabel 4.11 Tampilan Data Dalam Satu Cluster

	Tweet	Datapoint	Cluster
Doc 1	"celaka maut bus eka ring road sragen"	CF_1 = [1; (1; 1,48; 1,48; 1,48);(1; 2,19; 2,19; 2,19)]	CF1
Doc 2	"tewas 346 orang 2 celaka boss boeing maaf"	CF_2 = [1; (1; 1,48; 1,48; 1,48);(1; 2,19; 2,19; 2,19)]	

Doc 3	“takut mati benar takut risiko celaka mati”	CF3 = [1; (1; 2,35; 2,35; 1,48; 1,48); (1; 5,52; 5,52; 2,19; 2,19)]	

4.5 Validasi Output

Proses validasi output dilakukan setelah memperoleh output dari hasil *clustering* algoritma BIRCH. Validasi output digunakan untuk mengetahui baik atau tidaknya cluster yang dihasilkan dengan menggunakan silhouette coefficient. Berikut merupakan kode program yang digunakan untuk mengetahui nilai silhouette coefficient dari cluster yang dihasilkan.

```
#Sillhouette Coefficient

#Untuk mengevaluasi apakah output yg dikeluarkan sudah baik
#Menghitung Silhoutte Coefficient

from sklearn.metrics import silhouette_score
silhouette_score(tfidf_mat, labels)
```

Gambar 4.15 Kode Program Silhouette Coefficient

Pada kode program menggunakan library python untuk memudahkan dalam menghitung nilai silhouette coefficient dengan mengimport library silhouette_score. Setelah proses tersebut dijalankan maka diperoleh nilai silhouette coefficient dari data 1 yakni 0.1159964638217295. Sedangkan untuk data 2 diperoleh nilai silhouette coefficient yakni 0.7262655918349612.

Untuk mengetahui lebih lanjut tentang nilai silhouette coefficient yang dihasilkan maka penulis mengubah *max_features* atau *maximal term* yang digunakan sebagai berikut.

Tabel 4.12 Perbandingan Hasil Silhouette Coefficient

Max_Features	**Data 1**	**Data 2**
1000	0.1159964638217295	0.7262655918349612
1500	0.10845864278605032	0.7146335164265296

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan dari tahapan proses yang dijelaskan pada bab sebelumnya maka diperoleh hasil pengujian algoritma BIRCH menggunakan data tweet kecelakaan dengan pendekatan *text mining* yang telah dilakukan yakni algoritma BIRCH dapat melakukan pengelompokkan data yang berupa teks dengan sebelumnya melalui tahap *preprocessing* data yakni mencakup *case folding*, *tokenizing*, *filtering* dan *stemming* serta terlebih dahulu menghitung nilai TF-IDF untuk selanjutnya dilakukan proses *clustering* algoritma BIRCH dengan menentukan *branching factor* bernilai 50, banyaknya *cluster* dan *threshold* bernilai 0,5. Dalam penelitian ini menggunakan *cluster* sebanyak None untuk mengetahui berapa banyak cluster yang dihasilkan secara otomatis melalui proses *clustering* algoritma BIRCH. Dari hasil penelitian ini juga diperoleh bahwa hasil cluster juga dipengaruhi oleh banyaknya *max_features* atau *maximal term* yang kita tentukan. oleh sebab itu, dari hasil *cluster* yang dihasilkan dalam proses *clustering* dengan algoritma BIRCH diketahui bahwa penentuan nilai *cluster* juga berpengaruh dalam proses eksekusi dan nilai *sillhouette coefficient* yang dihasilkan. Semakin banyak *cluster* yang dihasilkan maka nilai *silhouette coefficient* semakin baik ditandai dengan nilai bobot *silhouette coefficient* itu sendiri.

5.2 Saran

Dari hasil penelitian yang dilakukan dalam kasus ini masih mempunyai banyak kekurangan dalam algoritma BIRCH untuk mengelompokkan kasus data

kecelakaan. Diharapkan dalam penelitian berikutnya proses pengerjaannya dapat menggunakan metode atau algoritma pengelompokkan yang lain supaya berguna sebagai pembanding hasil pengelompokkan yang dipergunakan untuk mencari algoritma *clustering* terbaik.

DAFTAR PUSTAKA

Darat, D. P. (2006). *Buku Petunjuk Tata Cara Bersepeda Motor di Indonesia.* Jakarta: Departemen Perhubungan RI.

Harahap, G. (1995). *Masalah Lalu lintas dan Pengembangan Jalan.* Bandung.

Han, J., Kamber, M, Pei, J. (2012). *Data Mining Concepts and Techniques Third Edition.* USA: Elsevier

Heinrich, H. e. (1996). *A safety Management Approach in Industrial Accident Prevention.* New York: Mc. Grow Hill Book Company.

Hermawati, F. A. (2013). *Data Mining.* Yogyakarta: Penerbit Andi.

Hidayatullah, A. F. (2014). *Analisis Sentiment dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Data Twitter Menggunakan Naive Bayes Classifier*. Yogyakarta: Universitas Islam Indonesia.

Hobbs, F. (1995). *Perencanaan dan Teknik Lalu Lintas.* Yogyakarta: Edisi Kedua, Gajahmada University Press.

Indonesia, B. P. (2021). *Kependudukan*. Diambil kembali dari Bps.go.id: https://www.bps.go.id/site/resultTab

Kamber, H. J. (2001). *Data Mining: Concepts and Techniques.* Simon Fraser University. USA: Morgan Kaufman Publisher.

Kamber, H. J. (2006). *Data Mining: Concepts and Techniques second Edition.* Simon Fraser University. USA: Morgan Kaufman Publisher.

Kaufmann L, & Rousseeuw PJ. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley

Kusworo, Y. (2018). *Pengelompokan Sekolah Menengah Atas jurusan IPS di Provinsi DIY berdasarkan data indeks integritas ujian nasional dengan algoritma Balanced Iterative Reducing and Clustering Using Hierarchies.* Yogyakarta: Sanata Dharma University.

Larose, D. (2015). *Discovering Knowledge in Data An Introduction to Data Mining.* New Jersey: John Wiley & Sons, inc.

Ramadhani, F. (2019). Cluster Big Data Dengan Balance Itterative. *Universitas Sumatera Utara*.

RI, K. A. (2014). *Al-Qur'an Terjemah & Tajwid.* Bandung: PT Sygma Examedia Arkanleema.

Soesantiyo. (1985). *Teknik Lalu Lintas, Traffic Engineering Jilid I.* Jakarta.

Sugiyono. (2014). *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif, dan R&D.* Bandung: Alfabeta.

Turban, E. (2005). *Decision Support System and Intelligent System* . Yogyakarta: Andi Offset.

Venkateswarlu, B. &. (2013). Mine Blood Donors Information through Improved K-Means Clustering. *International Journal of Computational Science and Information Technology, 1(3)*, 9-15.

Warpani, S. (2001). *Rekayasa Lalu Lintas.* Jakarta: Bharatara.

Wedasana, S. A. (2011). *Analisis Daerah Rawan Kecelakaan dan Penyusunan Database Berbasis Sistem Informasi Geografis (Studi Kasus Kota Denpasar).* Bali: Tesis, Jurusan Teknik Sipil, Universitas Denpasar.

LAMPIRAN

Lampiran 1 Kode Program *Crawling* Data Twitter

```
import tweepy,sys,jsonpickle
----
----
CONSUMER_KEY = " "
CONSUMER_SECRET = " "
ACCESS_KEY = " "
ACCESS_SECRET = " "
----
auth = tweepy.OAuthHandler(CONSUMER_KEY,
CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
api = tweepy.API(auth, wait_on_rate_limit=True,
wait_on_rate_limit_notify=True)
try:
    api.verify_credentials()
    print("Authentication OK")
except:
    print("Error during authentication")
----
#Memasukkan variabel-variabel yang diperlukan dalam
proses crawling
qry = input("Masukkan Query yang akan anda cari :")
#input query yang akan dicari
fName = input("Nama File Hasil Crawling :") #input
nama file hasil pencarian
maxTweets = 2000 # Isi sembarang nilai sesuai
kebutuhan
tweetsPerQry = 100  # Jangan isi lebih dari 100, tidak
boleh oleh Twitter
----
sinceId, max_id, tweetCount = None, -1, 0
----
#Proses Crawling Data From Twitter
print("Mulai mengunduh maksimum {0}
tweets".format(maxTweets))
with open(fName,'w') as f:
    while tweetCount < maxTweets:
        try:
            if (max_id <= 0):
                if (not sinceId):
```

```
new_tweets=api.search(q=qry,count=tweetsPerQry)
                else:

new_tweets=api.search(q=qry,count=tweetsPerQry,since
_id=sinceId)
            else:
                if (not sinceId):

new_tweets=api.search(q=qry,count=tweetsPerQry,max_i
d=str(max_id - 1))
                else:
new_tweets=api.search(q=qry,count=tweetsPerQry,max_i
d=str(max_id - 1),since_id=sinceId)
            if not new_tweets:
                print('Tidak ada lagi Tweet
ditemukan dengan Query="{0}"'.format(qry));break
            for tweet in new_tweets:

f.write(jsonpickle.encode(tweet._json,unpicklable=Fa
lse)+'\n')
            tweetCount+=len(new_tweets)
sys.stdout.write("\r");sys.stdout.write("Jumlah
Tweets telah tersimpan: %.0f"
%tweetCount);sys.stdout.flush()
            max_id=new_tweets[-1].id
        except tweepy.TweepError as e:
            print("some error : " + str(e));break #
Aya error, keluar
print ('\nSelesai! {0} tweets tersimpan di
"{1}"'.format(tweetCount,fName))
----
```

Lampiran 2 Kode Program Convert Json ke CSV

```
import xlrd
import xlwt
import json
import pandas as pd

pd.read_json('kecelaakaan_twitter', lines=True)
----

----
xl = pd.read_json('kecelaakaan_twitter',
lines=True)
xl.to_csv('output_kecelakaan.csv', encoding="utf8")
----

----
df = pd.read_csv('output_kecelakaan.csv')
df.head()
----
```

Lampiran 3 Kode Program *Clustering* Algoritma BIRCH

```
from sklearn.pipeline import Pipeline
import pandas as pd
import numpy as np
import seaborn as sns
sns.set()

df_dataset = pd.read_csv('process.csv')
df_dataset.head()

data = df_dataset[["full_text", "is_accident"]]
data.head()


#Proses case folding

import re
def casefolding(full_text):
    full_text = full_text.lower()
```

```
full_text = full_text.strip(" ")
    full_text = re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z
\t])|(\w+:\/\/\S+)"," ",full_text)
    return full_text
data['full_text'] =
data['full_text'].apply(casefolding)
data.head()

#Proses tokenizing

def token(full_text):
    nstr = full_text.split(' ')
    dat = []
    a = -1
    for hu in nstr:
        a = a + 1
        if hu == '':
            dat.append(a)
    p = 0
    b = 0
    for q in dat:
        b = q - p
        del nstr[b]
        p = p +1
    return nstr
data['full_text'] = data['full_text'].apply(token)
data.head()


#Proses Filtering

import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

def stopword_removal(full_text):
    filtering =
stopwords.words('indonesian','english')
    x = []
    data = []
    def myFunc(x):
```

```
        if x in filtering:
            return False
        else:
            return True
    fit = filter(myFunc, full_text)
    for x in fit:
        data.append(x)
    return data
data['full_text'] =
data['full_text'].apply(stopword_removal)
data.head()

# proses stemming

from sklearn.pipeline import Pipeline
from Sastrawi.Stemmer.StemmerFactory import
StemmerFactory

def stemming(full_text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    do = []
    for w in full_text:
        dt = stemmer.stem(w)
        do.append(dt)
    d_clean = []
    d_clean = " ".join(do)
    print(d_clean)
    return d_clean
data['full_text'] =
data['full_text'].apply(stemming)

data.to_csv('data_twitter.csv', index = False)
data_clean = pd.read_csv('data_twitter.csv',
encoding='latin1')
data.head()

data_clean = data_clean.astype({'is_accident' :
'category'})
data_clean = data_clean.astype({'full_text' :
'string'})
data_clean.dtypes
```

```
#Proses TF-IDF
From sklearn.feature_extraction.text import
TfidfVectorizer

# banyaknya term yang akan digunakan,
max_features = 1000

# Feature Engineering
print ("------- TF-IDF on Tweet data -------")

tf_idf = TfidfVectorizer(max_features=max_features,
binary=True)
tfidf_mat =
tf_idf.fit_transform(data['full_text']).toarray()

print("TF-IDF ", type(tfidf_mat), tfidf_mat.shape)


#Implementasi Algoritma BIRCH

from sklearn.cluster import Birch
import matplotlib.pyplot as plt

model = Birch(branching_factor = 50, n_clusters =
None, threshold = 0.5)
model.fit(tfidf_mat)
labels = model.predict(tfidf_mat)
pred =model.predict(tfidf_mat )
plt.scatter(tfidf_mat [:, 0],tfidf_mat [:, 1], c =
pred, cmap='rainbow', alpha=0.7, edgecolors='b')
plt.title('Hasil Cluster Algoritma BIRCH', size=12)
plt.xlabel('Sumbu X', size=12)
plt.ylabel('Sumbu Y', size=12)
plt.show()

np.unique(model.labels_)
```

```
#Sillhouette Coefficient

#Untuk mengevaluasi apakah output yg dikeluarkan
sudah baik
#Menghitung Silhoutte Coefficient

from sklearn.metrics import silhouette_score
silhouette_score(tfidf_mat, labels)
```

RIWAYAT HIDUP

Iftah Nur Fadlilah lahir di Lamongan pada 8 Maret 2000. Memiliki nama panggilan Iftah, bertempat tinggal di Dusun Sawo, RT 05 RW 02, Des. Payaman, Kec. Solokuro, Kab. Lamongan. Anak pertama dari Bapak Noer Cholis dan Ibu Mashudah. Pendidikan yang pernah ditempuh yaitu Pendidikan Sekolah Dasar di MI Muhammadiyah 01 Payaman dan lulus tahun 2012. Setelah itu melanjutkan Pendidikan di SMP Muhammadiyah 12 Paciran dan lulus tahun 2015, untuk Pendidikan selanjutnya ditempuh di MA Al-Ishlah Sendangagung, Paciran dan lulus tahun 2018. Pada tahun 2018 melanjutkan studi ke jenjang Pendidikan Strata-1 di Universitas Islam Negeri Maulana Malik Ibrahim Malang dan mengambil Program Studi Matematika Fakultas Sains dan Teknologi.

Selama menjadi mahasiswa, kegiatan-kegiatan yang pernah diikuti yaitu menjadi bagian dari Tim Kreatif di Scholars Jawa Timur Bacth 2 dan Batch 3, Kuliah Kerja Mahasiswa (KKM) UIN Malang mengabdi tahun 2021, Praktek Kerja Lapagan (PKL) di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan tahun 2021 dan menjadi Student Ambassador di Halolearn Batch 1 pada tahun 2021.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Iftah Nur Fadlilah
NIM 18610015
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi :Pengelompokan Data Tweet Kecelakaan Menggunakan Pendekatan *Text Mining* dan Algoritma BIRCH
Pembimbing I : Hisyam Fahmi, M.Kom
Pembimbing II : Ari Kusumastuti, M.Pd, M.Si

No	Tanggal	Hal	Tanda Tangan
1.	3 Februari 2022	Konsultasi Bab 1,2,3	1.
2.	21 Maret 2022	Konsultasi Bab 1,2,3	2.
3.	25 Maret 2022	Konsultasi Kajian Agama	3.
4.	28 Maret 2022	Konsultasi Bab 1,2,3	4.
5.	29 Maret 2022	Konsultasi Kajian Agama	5.
6.	30 Maret 2022	Acc Bab 1,2,3	6.
7.	14 April 2022	Konsultasi Bab 4	7.
8.	24 Mei 2022	Konsultasi Bab 4,5	8.
9.	25 Mei 2022	Konsultasi Bab 4,5	9.
10.	27 Mei 2022	Konsultasi Kajian Agama	10.
11.	30 Mei 2022	Acc Bab 4,5	11.
12.	14 Juni 2022	Konsultasi Keseluruhan	12.
13.	16 Juni 2022	Konsultasi Kajian Agama	13.
14.	17 Juni 2022	Acc Keseluruhan	14.

Malang, 24 Juni 2022
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP.19741129 200012 2 005