**ANALISIS SENTIMEN MENGENAI KULIAH DARING DI TWITTER MENGGUNAKAN METODE NAÏVE BAYES DAN PEMBENTUKAN STOPWORD DENGAN TERM BASED RANDOM SAMPLING**

SKRIPSI

Untuk memenuhi sebagian persyaratan   
memperoleh gelar Sarjana Komputer

Disusun oleh:

Raditya Rinandyaswara

NIM: 175150200111047



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2020

PENGESAHAN

Judul ini digunakan untuk dokumen final yang telah direvisi/disetujui setelah kelulusan ujian.

Untuk pendaftaran semhas dan ujian skripsi, judul halaman ini adalah **PERSETUJUAN.**

ANALISIS SENTIMEN MENGENAI KULIAH DARING DI TWITTER MENGGUNAKAN METODE NAÏVE BAYES DAN PEMBENTUKAN STOPWORD DENGAN TERM BASED RANDOM SAMPLING

SKRIPSI

Bagian ini hanya digunakan jika halaman ini adalah halaman **PENGESAHAN.** Jika ini adalah halaman PERSETUJUAN,maka bagian ini tidak diperlukan.

Diajukan untuk memenuhi sebagian persyaratan

memperoleh gelar Sarjana Komputer

Disusun Oleh :

Raditya Rinandyaswara

NIM: 175150200111047

Skripsi ini telah diuji dan dinyatakan lulus pada

2 Januari 2015

Telah diperiksa dan disetujui oleh:

|  |  |
| --- | --- |
| Dosen Pembimbing I  Yuita Arum Sari, S.Kom., M.Kom.  NIK: 2016098807152001 | Dosen Pembimbing 2  Muhammad Tanzil Furqon, S.Kom., M.CompSc.  NIK: 198209302008011004 |

Mengetahui

Ketua Jurusan **Teknik Informatika**

Contoh: Ketua Jurusan **Teknik** **Informatika**

Achmad Basuki, S.T., M.MG., Ph.D.

NIP: 197411182003121002

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Januari 2015



­

Raditya Rinandyaswara

NIM: 175150200111047

PRAKATA

Bagian ini memuat pernyataan resmi untuk menyampaikan rasa terima kasih penulis kepada berbagai pihak yang telah membantu penyelesaian skripsi ini. Nama-nama penerima ucapan terima kasih sebaiknya dituliskan lengkap, termasuk gelar akademik, dan pihak-pihak yang tidak terkait dihindari untuk dituliskan. Bahasa yang digunakan seharusnya mengikuti kaidah bahasa Indonesia yang baku. Prakata boleh diakhiri dengan paragraf yang menyatakan bahwa penulis menerima kritik dan saran untuk pengembangan penelitian selanjutnya. Terakhir, prakata ditutup dengan mencantumkan kota dan tanggal penulisan prakata, lalu diikuti dengan kata “Penulis”.

Malang, 1 Januari 2015

Penulis

email@domain.com

ABSTRAK

Nama Mahasiswa, Judul Skripsi

Pembimbing: Nama Pembimbing 1 dan Nama Pembimbing 2

Bagian ini diisi dengan abstrak dalam Bahasa Indonesia. Abstrak adalah uraian singkat (umumnya 200-300 kata) yang merupakan intisari dari sebuah skripsi. Abstrak membantu pembaca untuk mendapatkan gambaran secara cepat dan akurat tentang isi dari sebuah skripsi. Melalui abstrak, pembaca juga dapat menentukan apakah akan membaca skripsi lebih lanjut. Oleh karena itu, abstrak sebaiknya memberikan gambaran yang padat tetapi tetap jelas dan akurat tentang (1) apa dan mengapa penelitian dikerjakan: sedikit latar belakang, pertanyaan atau masalah penelitan, dan/atau tujuan penelitian; (2) bagaimana penelitian dikerjakan: rancangan penelitian dan metodologi/metode dasar yang digunakan dalam penelitian; (3) hasil penting yang diperoleh: temuan utama, karakteristik artefak, atau hasil evaluasi artefak yang dibangun; (4) hasil pembahasan dan kesimpulan: hasil dari analisis dan pembahasan temuan atau evaluasi artefak yang dibangun, yang dikaitkan dengan pertanyaan/tujuan penelitian.

Yang harus dihindari dalam sebuah abstrak diantaranya (1) penjelasan latar belakang yang terlalu panjang; (2) sitasi ke pustaka lainnya; (3) kalimat yang tidak lengkap; (3) singkatan, jargon, atau istilah yang membingungkan pembaca, kecuali telah dijelaskan dengan baik; (4) gambar atau tabel; (5) angka-angka yang terlalu banyak.

Di akhir abstrak ditampilkan beberapa kata kunci (normalnya 5-7) untuk membantu pembaca memposisikan isi skripsi dengan area studi dan masalah penelitian. Kata kunci, beserta judul, nama penulis, dan abstrak biasanya dimasukkan dalam basis data perpustakaan. Kata kunci juga dapat diindeks dalam basis data sehingga dapat digunakan untuk proses pencarian tulisan ilmiah yang relevan. Oleh karena itu pemilihan kata kunci yang sesuai dengan area penelitian dan masalah penelitian cukup penting. Pemilihan kata kunci juga bisa didapatkan dari referensi yang dirujuk.

Kata kunci: abstrak, skripsi, intisari, kata kunci, artefak

ABSTRACT

Student Name, Skripsi Title

Supervisors: First Supervisor’s Name and Second Supervisor’s Name

The absract of your skripsi in English is written here.

DAFTAR ISI

[PENGESAHAN ii](#_Toc57963495)

[PERNYATAAN ORISINALITAS iii](#_Toc57963496)

[PRAKATA iv](#_Toc57963497)

[ABSTRAK v](#_Toc57963498)

[ABSTRACT vi](#_Toc57963499)

[DAFTAR ISI vii](#_Toc57963500)

[DAFTAR TABEL xi](#_Toc57963501)

[DAFTAR GAMBAR xiii](#_Toc57963502)

[DAFTAR LAMPIRAN xiv](#_Toc57963503)

[BAB 1 PENDAHULUAN 1](#_Toc57963504)

[1.1 Latar Belakang 1](#_Toc57963505)

[1.2 Rumusan Masalah 2](#_Toc57963506)

[1.3 Tujuan 2](#_Toc57963507)

[1.4 Manfaat 3](#_Toc57963508)

[1.5 Batasan Masalah 3](#_Toc57963509)

[1.6 Sistematika Pembahasan 3](#_Toc57963510)

[BAB 2 LANDASAN KEPUSTAKAAN 5](#_Toc57963511)

[2.1 Kajian Pustaka 5](#_Toc57963512)

[2.1.1 *New Normal* 6](#_Toc57963513)

[2.1.2 Kuliah Daring 6](#_Toc57963514)

[2.1.3 Twitter 7](#_Toc57963515)

[2.2 Teks *Pre-processing* 7](#_Toc57963516)

[*2.2.1 Case Folding* 7](#_Toc57963517)

[*2.2.2 Cleaning* 7](#_Toc57963518)

[2.2.3 Tokenisasi 7](#_Toc57963519)

[2.2.4 *Stopword* *Removal* 7](#_Toc57963520)

[*2.2.5 Stemming* 7](#_Toc57963521)

[2.3 *Term Based Random Sampling* 8](#_Toc57963522)

[2.4 Term Frequency – Inverse Document Frequency (TF-IDF) 9](#_Toc57963523)

[2.5 Algoritme *Naïve Bayes* 9](#_Toc57963524)

[2.6 *Confusion Matrix* 10](#_Toc57963525)

[2.7 *K-Fold Cross Validation* 11](#_Toc57963526)

[BAB 3 METODOLOGI 13](#_Toc57963527)

[3.1 Tipe Penelitian 13](#_Toc57963528)

[3.2 Strategi Penelitian 13](#_Toc57963529)

[3.3 Subjek Penelitian 13](#_Toc57963530)

[3.4 Peralatan Pendukung 13](#_Toc57963531)

[3.5 Lokasi Penelitian 14](#_Toc57963532)

[3.6 Teknik Pengumpulan Data 14](#_Toc57963533)

[3.7 Data Penelitian 14](#_Toc57963534)

[3.8 Teknik Analisis Data 14](#_Toc57963535)

[3.9 Implementasi Algoritme 14](#_Toc57963536)

[BAB 4 PERANCANGAN 16](#_Toc57963537)

[4.1 Diagram Alir Sistem 16](#_Toc57963538)

[4.1.1 Diagram Alir *Preprocessing* tanpa Filtering 17](#_Toc57963539)

[4.1.2 Diagram Alir *Term Based Random Sampling* 18](#_Toc57963540)

[4.1.2.1 Diagram Alir Kullback Leibler Divergence 22](#_Toc57963541)

[4.1.2.2 Diagram Alir Normalisasi MinMax Term Weighting 23](#_Toc57963542)

[4.1.3 Diagram Alir *Preprocessing* 24](#_Toc57963543)

[4.1.4 Diagram Alir *Term Weighting* 25](#_Toc57963544)

[4.1.4.1 Diagram Alir Raw Term Weighting 26](#_Toc57963545)

[4.1.4.2 Diagram Alir Log Term Weighting 27](#_Toc57963546)

[4.1.4.3 Diagram Alir Inverse Document Frequency 28](#_Toc57963547)

[4.1.4.4 Diagram Alir Term Frequency - Inverse Document Frequency 29](#_Toc57963548)

[4.1.5 Diagram Alir *Naïve Bayes Training* 30](#_Toc57963549)

[4.1.5.1 Diagram Alir Hitung Likelihood term tiap kelas 32](#_Toc57963550)

[4.1.5.2 Diagram Alir Hitung Prior tiap kelas 33](#_Toc57963551)

[4.1.6 Diagram Alir *Naïve Bayes Testing* 33](#_Toc57963552)

[4.2 Manualisasi 36](#_Toc57963553)

[4.2.1 Persiapan Data 36](#_Toc57963554)

[4.2.2 Manualisasi Pembuatan Daftar Stopword 37](#_Toc57963555)

[4.2.3 Preprocessing 54](#_Toc57963556)

[4.2.3.1 Case Folding 54](#_Toc57963557)

[4.2.3.2 Cleaning 56](#_Toc57963558)

[4.2.3.3 Stemming 57](#_Toc57963559)

[4.2.3.4 Tokenisasi 59](#_Toc57963560)

[4.2.3.5 Filtering 61](#_Toc57963561)

[4.2.4 Term Weighting 62](#_Toc57963562)

[4.2.4.1 Raw Term Frequeny Weighting 63](#_Toc57963563)

[4.2.4.2 Log Term Frequency Weighting 67](#_Toc57963564)

[4.2.4.3 Inverse Document Frequency 71](#_Toc57963565)

[4.2.4.4 Term Frequency - Inverse Document Frequency (TF-IDF) 79](#_Toc57963566)

[4.2.5 Manualisasi Naïve Bayes Training 83](#_Toc57963567)

[4.2.6 Manualisasi *Naïve* *Bayes* *Testing* 88](#_Toc57963568)

[4.2.7 Manualisasi Evaluasi Confusion Matrix 91](#_Toc57963569)

[4.3 Perancangan Pengujian 94](#_Toc57963570)

[4.3.1 Perancangan Pengujian Terhadap Pengaruh Parameter X, Y, dan L terhadap Hasil Evaluasi Sistem menggunakan K-fold Cross Validation. 94](#_Toc57963571)

[4.3.2 Perancangan Pengujian Pengaruh Stopword Term Based Random Sampling dalam Akurasi Sistem. 95](#_Toc57963572)

[4.3.3 Perancangan Pengujian Perbandingan Akurasi Pengunaan Stopword Tala dan Stopword Term Based Random Sampling. 96](#_Toc57963573)

[BAB 5 IMPLEMENTASI 97](#_Toc57963574)

[5.1 Implementasi Preprocessing 97](#_Toc57963575)

[5.2 Implementasi Term Based Random Sampling 100](#_Toc57963576)

[5.3 Implementasi Term Weighting 107](#_Toc57963577)

[5.4 Implementasi Naïve Bayes 110](#_Toc57963578)

[5.4.1 Implementasi Naïve Bayes Training 114](#_Toc57963579)

[5.4.2 Implementasi Naïve Bayes Testing 116](#_Toc57963580)

[5.5 Implementasi K Fold 118](#_Toc57963581)

[5.6 Implementasi Confusion Matrix 121](#_Toc57963582)

[5.7 Implementasi Main 127](#_Toc57963583)

[DAFTAR REFERENSI 133](#_Toc57963584)

DAFTAR TABEL

[Tabel 2.1 Confusion Matrix 10](#_Toc55893956)

[Tabel 3.1 Spesifikasi Hardware 13](#_Toc55893957)

[Tabel 3.2 Spesifikasi Software 13](#_Toc55893958)

[Tabel 4.1 Data Latih 35](#_Toc55893959)

[Tabel 4.2 Data Uji 36](#_Toc55893960)

[Tabel 4.3 Data Manualisasi Pembuatan Stopword yang sudah di Preprocessing 36](#_Toc55893961)

[Tabel 4.4 Term Manualisasi Pembuatan Daftar Stopword 37](#_Toc55893962)

[Tabel 4.5 Dokumen Sampel Manualisasi Pembuatan Daftar Stopword 38](#_Toc55893963)

[Tabel 4.6 Term Dokumen Sampel Manualisasi Pembuatan Daftar Stopword 39](#_Toc55893964)

[Tabel 4.7 Hasil Kullback Leibler Manualisasi 40](#_Toc55893965)

[Tabel 4.8 Hasil Normalisasi Kullback Leibler Manualisasi 42](#_Toc55893966)

[Tabel 4.9 Hasil 30 Bobot Terendah 44](#_Toc55893967)

[Tabel 4.10 Sampel Hasil Keseluruhan Bobot Tiap Iterasi 45](#_Toc55893968)

[Tabel 4.11 Hasil Rata-rata Keseluruhan Bobot 46](#_Toc55893969)

[Tabel 4.12 Hasil Rata-rata Bobot yang sudah diurutkan 49](#_Toc55893970)

[Tabel 4.13 Daftar Stopword 20 persen 52](#_Toc55893971)

[Tabel 4.14 Manualisasi Case Folding Data Latih 53](#_Toc55893972)

[Tabel 4.15 Manualisasi Case Folding Data Uji 54](#_Toc55893973)

[Tabel 4.16 Manualisasi Cleaning Data Latih 55](#_Toc55893974)

[Tabel 4.17 Manualisasi Cleaning Data Uji 56](#_Toc55893975)

[Tabel 4.18 Manualisasi Stemming Data Latih 57](#_Toc55893976)

[Tabel 4.19 Manualisasi Stemming Data Uji 57](#_Toc55893977)

[Tabel 4.20 Manualisasi Tokenisasi Data Latih 58](#_Toc55893978)

[Tabel 4.21 Manualisasi Tokenisasi Data Uji 59](#_Toc55893979)

[Tabel 4.22 Manualisasi Filtering 20 Persen Data Latih 60](#_Toc55893980)

[Tabel 4.23 Manualisasi Filtering 20 Persen Data Uji 61](#_Toc55893981)

[Tabel 4.24 Manualisasi Raw Term Frequency Weighting 62](#_Toc55893982)

[Tabel 4.25 Manualisasi Log Term Frequency Weighting 66](#_Toc55893983)

[Tabel 4.26 Manualisasi Document Frequency 70](#_Toc55893984)

[Tabel 4.27 Manualisasi Inverse Document Frequency 74](#_Toc55893985)

[Tabel 4.28 Manualisasi Term Frequency - Inverse Document Frequency 79](#_Toc55893986)

[Tabel 4.29 Manualisasi Likelihood 83](#_Toc55893987)

[Tabel 4.30 Hasil *Preprocessing* Data Uji 87](#_Toc55893988)

[Tabel 4.31 Hasil Manualisasi Data Uji 1 89](#_Toc55893989)

[Tabel 4.32 Hasil Manualisasi Data Uji 2 89](#_Toc55893990)

[Tabel 4.33 Hasil Manualisasi Data Uji 3 89](#_Toc55893991)

[Tabel 4.34 Hasil Manualisasi Data Uji 4 90](#_Toc55893992)

[Tabel 4.35 Hasil Manualisasi Data Uji 5 90](#_Toc55893993)

[Tabel 4.36 Manualisasi Confusion Matrix 90](#_Toc55893994)

[Tabel 4.37 Definisi TP, FN, FP, dan TN 91](#_Toc55893995)

[Tabel 4.38 Hasil Manualisasi TP, FN, FP, dan TN setiap kelas 91](#_Toc55893996)

[Tabel 4.39 Hasil Evaluasi Manualisasi tiap Kelas 93](#_Toc55893997)

[Tabel 4.40 Perancangan Pengujian pengaruh X, Y, L terhadap Hasil Evaluasi 93](#_Toc55893998)

[Tabel 4.41 Perancangan Pengujian Pengaruh Stopword Term Based Random Sampling dalam Akurasi Sistem 94](#_Toc55893999)

[Tabel 4.42 Perancangan Pengujian Perbandingan Akurasi Pengunaan Stopword Tala dan Stopword Term Based Random Sampling 95](#_Toc55894000)

DAFTAR GAMBAR

[Gambar 2.1 Ilustrasi *K-Fold* *Cross Validation* 12](#_Toc55256775)

[Gambar 4.1 Diagram Alir Sistem, (a) Tahap Pembuatan Stopword, (b) Tahap Pelatihan, (c) Tahap Pengujian 15](#_Toc55256776)

[Gambar 4.2 Diagram Alir Preprocessing tanpa Filtering 17](#_Toc55256777)

[Gambar 4.3 Diagram Alir *Term Based Random Sampling* 21](#_Toc55256778)

[Gambar 4.4 Diagram Alir Kullback Leibler Divergence 22](#_Toc55256779)

[Gambar 4.5 Diagram Alir Normalisasi MinMax Term Weighting 23](#_Toc55256780)

[Gambar 4.6 Diagram Alir Preprocessing 24](#_Toc55256781)

[Gambar 4.7 Diagram Alir Term Weighting 25](#_Toc55256782)

[Gambar 4.8 Diagram Alir *Raw Term Wighting* 26](#_Toc55256783)

[Gambar 4.9 Diagram Alir *Log Term Wighting* 27](#_Toc55256784)

[Gambar 4.10 Diagram Alir *Inverse Document Frequency* 28](#_Toc55256785)

[Gambar 4.11 Diagram Alir *Term Frequency - Inverse Document Frequency* 29](#_Toc55256786)

[Gambar 4.12 Diagram Alir *Naive Bayes Training* 31](#_Toc55256787)

[Gambar 4.13 Diagram Alir Hitung Likelihood term tiap kelas 31](#_Toc55256788)

[Gambar 4.14 Diagram Alir Hitung Prior tiap kelas 32](#_Toc55256789)

[Gambar 4.15 Diagram Alir Naive Bayes Testing 34](#_Toc55256790)

DAFTAR LAMPIRAN

# PENDAHULUAN

Bab ini terdiri dari hal yang melatarbelakangi dari penelitian ini dilaksanakan, rumusan masalah yang diperoleh dari latar belakang hingga tujuan dan manfaat dari penelitian ini serta batasan yang dijabarkan sesuai dengan cakupan dan kemampuan penulis , maupun sistematika yang menuliskan secara rangkum isi dari tiap bab.

## Latar Belakang

Pada saat ini kemajuan teknologi di dunia maupun di Indonesia mengalami perkembangan yang cukup pesat. Kemajuan teknologi ini salah satunya adalah ditandai dengan mudahnya proses pertukaran Informasi antara satu pengguna dan pengguna lainnya. Kemudahan proses pertukaran informasi ini ditandai dengan maraknya bermuncul sosial media di Internet. Twitter merupakan salah satu contoh dari sosial media yang banyak digunakan oleh masyarakat Indonesia yang digunakan sebagai sarana pertukaran informasi di dunia digital. Dalam pengunaannya, Twitter memberi istilah kepada pertukaran informasi tersebut dengan nama *Tweets,* yang mana *Tweets* adalah suatu teks atau kata yang dibatasi panjang nya hingga 280 karakter yang nanti akan di-*posting* dalam *platform* Twitter tersebut.

Pada akhir tahun 2019 lalu, dunia dikejutkan dengan adanya wabah yang diakibatkan oleh virus corona yang berasal dari kota Wuhan, China. Penyakit yang disebut sebagai (COVID-19) ini adalah penyakit yang menyerang sistem pernapasan virus manusia. Menurut data pemerintah China, penduduk Hubei menjadi kasus pertama Covid-19 pada 17 November 2019 (Arnani, 2020). Setelah kasus pertama Covid-19 di dunia itu terjadi peningkatan pasien tiap bulannya. Hingga saat ini Indonesia sudah melewati angka 190 ribu kasus Covid-19 yang sudah terkonfirmasi yang terhitung dari sejak pasien pertama (Ramadhan, et al., 2020). Dengan adanya pandemi Covid-19 ini, pemerintah Indonesia mengadakan sistem *New Normal* dengan tujuan untuk mempercepat penanganan Covid-19 (Putsanra, 2020). Dalam penerapannya banyak perilaku yang biasa dilakukan secara luring berubah menjadi daring. Salah satu contohnya adalah perkuliahan. Tentunya kuliah daring ini menyebabkan banyak pro dan kontra dan salah satu sarana masyarakat menuliskan opininya mengenai kuliah daring tersebut yaitu melalui Twitter. Namun twitter sendiri hanya berfungsi sebagai sosial media sehingga tidak menyediakan analisis sentimen dari *tweet* pengguna dan bagaimana masyarakat Indonesia menanggapi protokol baru tersebut.

Analisis Sentimen atau *Opinion Mining* adalah salah satu bidang studi yang menganalisis pendapat, sentimen, evaluasi, penilaian, sikap, dan emosi orang terhadap entitas seperti produk, layanan, organisasi, isu, peristiwa, topik, dan atributnya (Liu, 2012). Dengan proses analisis sentimen ini kita dapat mengetahui bagaimana pendapat orang apakah cenderung positif, negatif atau pun netral.

Pada penelitian ini, peneliti akan membangun sebuah sistem analisis sentimen terhadap kuliah daring yang dituliskan masyarakat di Twitter menggunakan metode klasifikasi *Naïve Bayes*. Dalam penelitian ini klasifikasi akan dibuat menjadi 3 kelas yaitu opini netral, positif dan negatif sesuai dari saran penelitian analisis sentimen sebelumnya (Sa'rony, et al., 2019). Dalam proses analisis sentimen diperlukan *preprocessing* data agar data dapat dihitung dan diolah*.* Salah satu tahap yang penting dalam *preprocessing* adalah *Stopword Removal*, yaitu penghapusan kata yang tidak memiliki nilai keunikan dalam suatu dokumen. Biasanya dalam proses penghapusan *stopword* ini memanfaatkan *digital library* namun ada *stopword* yang kurang tepat sehingga hal itu dapat membuat kualitas proses *stopword* removal menjadi berkurang (Dila Purnama Sari, et al., 2020). Oleh karena itu perlu adanya suatu mekanisme sistem daftar *stopword* dinamik yang dapat menghasilkan daftar *stopword* yang sesuai dengan yang diperlukan sesuai saran dari penelitian sebelumnya (Rahutomo & Ririd, 2018). Oleh karena itu pada penelitian ini, peneliti akan melakukan pembuatan daftar *stopword* menggunakan metode *Term Based Random Sampling*.

## Rumusan Masalah

Dari latar belakang yang telah diuraikan sebelumnya, berikut adalah rumusan masalah untuk penelitian ini:

1. Bagaimana pengaruh parameter X, Y dan L pada *Term Based Random Sampling* terhadap hasil pembentukan *stopword*?
2. Bagaimana hasil evaluasi dari pembentukan *stopword Term Based Random Sampling* pada analisis sentimen dengan *Naïve Bayes*?
3. Bagaimana perbandingan hasil evaluasi dari pembentukan *stopword* dengan *Term Based Random Sampling* dengan *stopword* Tala pada analisis sentimen dengan Naïve Bayes?

## Tujuan

Tujuan penelitian ini adalah:

1. Mengetahui pengaruh parameter X, Y dan L pada *Term Based Random Sampling* terhadap hasil pembentukan *stopword*.
2. Mengetahui hasil evaluasi dari pembentukan *stopword Term Based Random Sampling* pada analisis sentimen dengan Naïve Bayes.
3. Mengetahui perbandingan hasil evaluasi dari pembentukan *stopword* dengan *Term Based Random Sampling* dengan *stopword* Tala pada analisis sentimen dengan *Naïve Bayes*.

## Manfaat

Manfaat penelitian ini adalah:

1. Dapat mengetahui bagaimana pengaruh parameter X, Y dan L pada *Term Based Random Sampling* terhadap hasil pembentukan *stopword*
2. Dapat mengetahui hasil evaluasi dari pembentukan *stopword Term Based Random Sampling* pada analisis sentimen dengan Naïve Bayes.
3. Dapat mengetahui perbandingan hasil evaluasi dari pembentukan *stopword* dengan *Term Based Random Sampling* dengan *stopword* Tala pada analisis sentimen dengan Naïve Bayes.

## Batasan Masalah

Batasan masalah penelitian ini adalah:

1. Hanya menggunakan opini pengguna Twitter mengenai Kuliah Daring.
2. Algoritma yang digunakan hanya *Naïve Bayes Classifier* tidak membandingkan dengan algoritma lain.
3. Hasil klasifikasi sentimen hanya dibagi menjadi tiga kelas yaitu positif, netral, dan negatif.
4. *Tweet* yang merupakan data hanya *tweet* yang berbahasa Indonesia.
5. Jumlah data yang digunakan sebanyak 300 data.

## Sistematika Pembahasan

Berikut sistematika pembahasan dalam penelitian ini:

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan sesuai dengan aturan dalam peneliltian

BAB II LANDASAN KEPUSTAKAAN

Landasan Kepustakaan menjelaskan penelitian-penelitian sebelumnya yang serupa dengan penelitian dalam proposal ini, serta dasar-dasar teori yang akan di implementasikan dalam penelitian ini seperti *preprocessing, term weighting, Naïve Bayes, Term Based Random Sampling,* serta tabel *confusion matrix* sehingga dapat mendukung penelitian.

BAB III METODOLOGI

Pada bab ini dijelaskan tentang bagaimana menerapkan penelitian seperti untuk mengimplementasikan *Naïve Bayes* dengan pembuatan daftar *Stopword* untuk analisis sentimen pengguna Twitter terhadap layanan Kuliah Daring.

BAB IV PERANCANGAN

Bab ini menjelaskan bagaimana proses perancangan dalam sistem yang akan dibangun.

BAB V IMPLEMENTASI

Pada bab ini menjelaskan bagimana implementasi sistem yang sudah dirancang di bab sebelumnya.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan pengujian terhadap sistem yang sudah dibangun dan menganalisa hasil yang didapatkan untuk menemukan kesimpulan dari hasil pengujian.

BAB VII PENUTUP

Pada bab terakhir ini menjelaskan tentang bagaimana kesimpulan dari penelitian ini dan saran untuk penelitian berikutnya.

# LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan bagaimana penelitian-penelitian sebelumnya yang serupa dengan penelitian yang sedang diajukan, serta dasar-dasar teori yang akan diimplementasikan dalam penelitian ini seperti *preprocessing, term weighting, Naïve Bayes, Term Based Random Sampling,* serta tabel *confusion matrix* yang dapat mendukung penelitian.

## Kajian Pustaka

Pada bagian ini akan dibahasnya mengeni penelitian atau kajian pustaka yang digunakan dalam penelitian ini berasal dari beberapa penelitian yang sudah dilakukan sebelumnya dan memiliki keterkaitan dengan judul skripsi Analisis Sentimen mengenai Kuliah Daring di Twitter menggunakan metode *Naïve Bayes* dan Pembentukan *Stopword* dengan *Term Based Random Sampling*.

Contoh salah satu penelitian terkait judul skripsi yang telah disebutkan adalah penelitian yang dilakukan oleh (Septian, et al., 2019) yaitu mengenai analisis sentimen pengguna Twitter terhadap polemik persepakbolaan Indonesia menggunakan pembobotan TF-IDF dan metode klasifikasi *K-Nearest Neighbor*. Penelitian ini menggunakan kamus kata tidak baku yang dibuat oleh peneliti secara manual yang nantinya akan digunakan sebagai normalisasi kata. Hasil pengujian yang didapatkan dari pengujian silang sebanyak 10 kali dan mendapatkan hasil akurasi optimal pada nilai k-23 sejumlah 79.99%.

Selain *K-Nearest Neighbor*, salah satu metode klasifikasi umum yang digunakan adalah *Naïve Bayes*. Pada penelitian sebelumnya oleh (Devita, et al., 2018) Kinerja metode *Naïve Bayes* dibandingkan dengan *K-Nearest Neighbor* untuk Klasifikasi Teks Artikel berbahasa Indonesia. Hasil yang didapatkan menunjukan metode *Naïve Bayes* memiliki kinerja yang lebih baik dengan tingkat akurasi 70% sedangkan metode *K-Nearest Neighbor* memiliki tingkat akurasi yuang cukup rendah yaitu 40%.

Contoh selanjutnya adalah pada penelitian yang dilakukan oleh (Rahman, et al., 2017) dimana dalam penelitiannya, metode *Multinomial* *Naïve Bayes* digunakan untuk Klasifikasi Berita Online dengan menggunakan *feature* *selection* *Document Frequency Thresholding* dan menggunakan TF-IDF untuk pembobotan *term* dan menghasilkan akurasi tertinggi pada saat menggunakan TF-IDF 86,62%.

Tak hanya seleksi fitur, *stopword* merupakan salah satu tahapan penting dari *preprocessing*, dalam tahap *preprocessing* perlu adanya suatu mekanisme sistem daftar *stopword* dinamik yang dapat menghasilkan daftar *stopword* yang sesuai dengan yang diperlukan sesuai saran dari penelitian sebelumnya (Rahutomo & Ririd, 2018). Contoh pembuatan *stopword* dinamik ada pada penelitian selanjutnya yang dilakukan oleh (Dila Purnama Sari, et al., 2020) dimana dalam penelitiannya dilakukan pembentukan daftar *Stopword* menggunakan Zipf Law dan Pembobotan Augmented TF-Probability IDF pada klasifikasi Dokumen Ulasan Produk yang menggunakan metode Support Vector Machine dan Polynomial Kernel untuk memperoleh hasil klasifikasi. Daftar *Stopword* yang dibentuk secara dinamis dengan menggunakan metode Zipf Law dan pembobotan kata memiliki pengaruh terhadap hasil akurasi klasifikasi. Akurasi terbaik didapatkan pada saat persentase 15% untul daftar *stopword* yaitu dengan nilai *precision* 0,73, *recall* 0,7 dan *f-measure* 0,64

Lalu dilanjutkan penelitian oleh (Sa'rony, et al., 2019) analisis sentimen positif dan negatif yang dilakukan menggunakan *Multinomial* *Naïve Bayes* yang menggunakan Raw *Term* Frequency serta pembuatan *stopword* menggunakan *Term Based Random Sampling* dan berhasil mendapatkan *macroaverage* terbaik pada klasifikasi dengan stoplist 20 persen dengan *macroaverage* akurasi sebesar 0,94 *macroaverage* *precision* sebesar 0,945, *macroaverage* *recall* sebesar 0,94, dan *macroaverage* *f-measure* sebesar 0,938.

Contoh selanjutnya adalah pada penelitian (Imtiyazi, et al., 2015) dimana dilakukan perbandingan terhadap penggunaan *Multinomial* *Naïve Bayes* dengan TF-IDF dan dibandingkan terhadap *Multinomial* *Naïve Bayes* dengan TF-*Improved Gini*. Hasil yang didapatkan penggunaan TF-IDF memiliki kinerja lebih baik dibandingkan dengan TF-iGini.

Dari penelitian yang sudah disebutkan diatas, belum ada di antaranya yang melakukan Analisis Sentimen yang dibagi menjadi 3 kelas klasifikasi yaitu negatif, netral, dan positif yang menggunakan *Multinomial* *Naïve Bayes* sebagai metodenya serta pembentukan *Stopword* menggunakan Algoritma *Term Based Random Sampling* yang menggunakan TF-IDF sebagai pembobotan katanya.

### *New Normal*

Dengan adanya pandemi Covid-19 ini, pemerintah Indonesia mengadakan sistem *New Normal* yakni dengan tujun untuk mempercepat penangan COVID-19 (Putsanra, 2020).

### Kuliah Daring

Kuliah daring merupakan salah satu dari efek kebijakan sistem *New Normal* yang terjadi karena pandemi Covid-19 ini. Kuliah daring adalah metode pembelajaran yang dilakukan secara daring (*online*) dengan menggunakan berbagai fasilitas seperti platform *Zoom*, *Google Meet, Google Classroom*, situs pembelajaran universitas, dan lain- lain. Dengan adanya fasilitas- fasilitas tersebut, mahasiswa dan dosen tetap dapat berinteraksi satu sama lain layaknya kuliah secara tatap muka atau offline (Tania, 2020).

### Twitter

Twitter merupakan sosial media besutan Amerika Serikat yang diluncurkan pada tahun 2006. Twitter merupakan salah satu contoh dari sosial media yang banyak digunakan oleh masyarakat Indonesia yang digunakan sebagai sarana pertukaran informasi di dunia digital. Dalam pengunaannya, Twitter memberi istilah kepada pertukaran informasi tersebut dengan nama *Tweets,* yang mana *Tweets* adalah suatu teks atau kata yang dibatasi panjang nya hingga 280 karakter yang nanti akan di-*posting* dalam *platform* Twitter tersebut.

## Teks *Pre-processing*

Teks pre-processing merupakan langkah awal yang dilakukan dalam analisis sentimen untuk menyiapkan data yang berupa teks agar mudah untuk diproses nantinya (Gaddam, 2019). Teks pre-processing ini meliputi, case folding, cleaning, tokenisasi, *stopword* removal dan stemming.

### *Case Folding*

*Case folding* adalah suatu tahapan untuk menyeragamkan kalimat menjadi huruf kecil atau *lowercase* semua. Contohnya, jika ada kalimat “Saya suka bermain Komputer” menjadi “saya suka bermain komputer”.

### *Cleaning*

*Cleaning* adalah suatu tahapan pembersihan kalimat dari simbol-simbol, tanda baca, maupun angka. Contohnya, jika ada kalimat “Selamat pagi Adis, semoga harimu menyenangkan!” menjadi “Selamat pagi Adis semoga harimu menyenangkan”.

### Tokenisasi

Tokenisasi adalah suatu tahapan untuk memisahkan antar kata dari suatu kalimat sehingga kata-kata tersebut menjadi satu tidak tergabung dengan kata-kata lainnya (Gaddam, 2019). Contohnya, jika ada kalimat “saya sedang bermain gitar” menjadi [‘saya’, ‘sedang’, ‘bermain’, ‘gitar’].

### *Stopword* *Removal*

*Stopword* *Removal* adalah suatu tahapan untuk menghilangkan kata-kata yang kurang relevan berdasarkan kamus *stopword* yang digunakan (Gaddam, 2019). Kamus *Stopword* yang digunakan dalam penelitian ini adalah kamus *stopword* yang bersifat dinamis yang akan dibuat sesuai dengan kebutuhan sistem.

### *Stemming*

*Stemming* adalah suatu tahapan untuk mencari kata dasar dari suatu kata (Gaddam, 2019). Contohnya jika ada kata “bermain” menjadi main.

## *Term Based Random Sampling*

*Term Based Random Sampling* adalah suatu metode yang dapat digunakan untuk menghasilkan daftar *stopword* secara otomatis berdasarkan seberapa informatif kata tertentu (Lo, et al., 2005). Kita dapat mengetahui apakah kata tersebut *stopword* atau bukan dengan melihat kepentingannya, semakin tidak penting kata tersebut, maka kemungkinan menjadi *stopword*-nya lebih tinggi. Untuk mencari nilai kepentingan dari suatu *term* dapat dilakukan dengan perhitungan dengan rumus dari teori *Kullback-Leibler.* Dengan rumus tersebut kita dapat memberi bobot dari suatu term pada dokumen sampel. Berikut rumus dari *Kullback-Leibler* direpresentasikan dalam Persamaan 2.1.

|  |  |
| --- | --- |
|  | ( 2.1 ) |

Yang dimana dipresentasikan dalam Persamaan 2.2 dan dipresentasikan dalam Persamaan 2.3 adalah sebagai berikut.

|  |  |
| --- | --- |
|  | ( 2.2 ) |
|  | ( 2.3 ) |

Keterangan :

: bobot *term* t pada dokumen sampel

: frekuensi kueri *term* dalam dokumen sampel

: jumlah dari panjang dokumen sampel

: frekuensi kueri *term* dari keseluruhan dokumen

: total token dari keseluruhan dokumen

Dalam perhitungannya dilakukan pemilihan acak *term* dari keseluruhan *term*, lalu kemudian ambil dokumen yang mengandung *term* tersebut dan cari semua term dalam dokumen tersebut. Setiap term dalam dokumen tersebut akan dilakukan perhitungan bobot nya menggunakan *Kullback-Leibler*. Lalu setelah perhitungan bobotnya diambil sejumlah X *term* yang diurutkan dari bobot terendah yang dimana X adalah parameter yang dapat diubah-ubah nantinya. Dalam prosesnya pemilihan *term* acak ini dilakukan sebanyak Y kali dimana Y adalah sebuah parameter yang dapat diisi secara manual untuk mendapatkan hasil yang terbaik. Setelah melakukan proses yang dijelaskan sebelumnya dihitung rata-rata keseluruhan bobot yang didapat oleh *term* lalu diambil sejumlah L dimana L adalah parameter yang dapat diubah nantinya. L adalah parameter yang menentukan berapa jumlah daftar *stopword* yang ingin digunakan.

## Term Frequency – Inverse Document Frequency (TF-IDF)

Pada tahap ini dilakukan pembobotan kata yang mempresentasikan kata-kata tersebut untuk dilakukan perhitungan nantinya. Salah satu metode dalam *term* weighting yang sering digunakan adalah *Term Frequency – Inverse Document Frequency* (𝑡𝑓. 𝑖𝑑𝑓) (Jones, 2004). Metode TF-IDF adalah penggabungan dua metode untuk melakukan pembobotan kata. TF atau *Term Frequency* adalah frekuensi kemunculan *term* pada suatu dokumen dan IDF atau *Inverse Document Frequency* adalah perhitungan *inverse* terhadap frekuensi dokumen yang mengandung kata tersebut (Prabowo, et al., 2016).Berikut rumus yang digunakan untuk perhitungan tf-idf direpresentasikan pada Persamaan 2.4 dan Persamaan 2.5.

Berikut perhitungan nilai log tf :

|  |  |
| --- | --- |
|  | ( 2.4 ) |

Berikut perhitungan nilai idf :

|  |  |
| --- | --- |
| 𝑖𝑑𝑓(𝑡) = | ( 2.5 ) |

Keterangan :

: frekuensi *term* pada dokumen d

: nilai *Inverse Document Frequency* suatu *term* t

: total dokumen

: nilai *Document Frequency* suatu *term* t

## Algoritme *Naïve Bayes*

Algoritma *Naïve Bayes* yaitu algoritma klasifikasi *supervised* yang berbasis dengan teorema bayes dengan asumsi independensi tiap fitur (Sawla, 2018). Algoritma ini menggunakan metode probabilistik dan statistik.

Algoritma ini mencari probabilitas tertinggi untuk proses klasifikasi. Perhitungan Algoritma *Naïve Bayes* direpresentasikan pada Persamaan 2.6.

|  |  |
| --- | --- |
|  | ( 2.6 ) |

Keterangan :

: Posterior atau Probabilitas kelas diberikan dokumen

: Prior atau Probabilitas awal muncul kategori

: Likelihood

Dalam perhitungan *likelihood* atau *conditional probability* salah satu metodenya adalah menggunakan *Multinomial*. Perhitungan *conditional probability* dengan *multinomial* direpresentasikan pada Persamaan 2.7

|  |  |
| --- | --- |
|  | ( 2.7 ) |

Keterangan :

: Likelihood dalam kelas

: Jumlah kemunculan kata pada kategori

: Jumlah semua total kemunculan kata pada kategori

: Jumlah term unik atau fitur

Namun dalam penelitian kali ini digunakan *TF-IDF* sebagai pembobotan sehingga perhitungan *likelihood* atau *conditional probability* direpresentasikan pada Persamaan 2.8 (Rahman, et al., 2017).

|  |  |
| --- | --- |
|  | ( 2.8 ) |

Keterangan :

: *Likelihood* dalam kelas

: Nilai pembobotan (W) TF-IDF dari term t di kategori c

: Jumlah bobot TF-IDF seluruh term pada kelas C

: Jumlah IDF term pada seluruh dokumen.

## *Confusion Matrix*

*Confusion Matrix* adalah pengukur performa dari klasifikasi pembelajaran mesin (*Machine Learning*) (Narkhede, 2018). *Confusion Matrix* berisikan tabel untuk menampilkan hasil evaluasi yang didalamnya terdapat 2 kolom yaitu kelas hasil prediksi dan kelas sebenarnya.

Tabel 2.1 *Confusion Matrix*

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Predicted | |
| Negatif | Positif |
| Actual | Negatif | TN | FP |
| Positif | FN | TP |

Keterangan :

* *True Negative* (TN) : jumlah dokumen yang *predicted* negatif dan *actual* negatif
* *False Positive* (FP) : jumlah dokumen yang *predicted* positif namun *actual* negatif
* *False Negative* (FN) : jumlah dokumen yang *predicted* negatif namun *actual* positif
* *True Postive* (TP) : jumlah dokumen yang *predicted* positif dan *actual* positif.

Fungsi dari *confusion* *matrix* untuk mempermudah evaluasi hasil klasifikasi untuk mencari *accuracy*, *precision*, *recall* dan *f-measure*. Berikut rumus *accuracy* direpresentasikan pada Persamaan 2.8, *recall* direpresentasikan pada Persamaan 2.9, *precision* direpresentasikan pada Persamaan 2.10, *f-measure* direpresentasikan pada Persamaan 2.11.

* *Accuracy* : kesesuaian nilai prediksi dengan nilai aktual

|  |  |
| --- | --- |
|  | ( 2.9 ) |

* *Recall* : jumlah banyak atau sedikitnya kesesuaian informasi yang didapatkan berdasarkan sudut pandang kelas atau label yang digunakan

|  |  |
| --- | --- |
|  | ( 2.10 ) |

* *Precision* : tingkat ketepatan antara informasi yang diminta

|  |  |
| --- | --- |
|  | ( 2.11 ) |

* *F-measure* : bobot harmonic mean pada *recall* dan *precision*

|  |  |
| --- | --- |
|  | ( 2.12 ) |

## *K-Fold Cross Validation*

*K-Fold* *Cross Validation* adalah suatu metode yang berfungsi untuk membagi data sebanyak K dengan ukuran yang sama atau hampir sama rata. Pada implementasinya pengujian *K-Fold* ini dilakukan dengan iterasi sebanyak K dimana pada setiap iterasinya data dibagi menjadi 2 tipe yaitu data latih dan data uji (Singh & Shukla, 2016). Berikut contoh illustrasi dari *K-Fold* *Cross Validation* yang ditunjukkan pada Gambar 2.1.



Gambar 2.1 Ilustrasi *K-Fold* *Cross Validation*

Sumber : <https://www.researchgate.net/figure/The-technique-of-KFold-cross-validation-illustrated-here-for-the-case-K-4-involves_fig10_278826818> (2015)

Berdasarkan Gambar 2.1 ditunjukkan bahwa tiap iterasi dibagi menjadi 2 tipe data yaitu yang berwarna hijau adalah data latih dan yang berwarna biru adalah data uji. Untuk menghitung nilai evaluasi akhir maka dihitung rata-rata dari evaluasi tiap iterasi (Neale, et al., 2019).

# METODOLOGI

Pada bab ini akan dijelaskan metodologi yang digunakan pada penelitian ini. Metodologi yang digunakan berupa tipe penelitian, strategi penelitian, subjek penelitian, lokasi penelitian, teknik pengumpulan data, peralatan pendukung, implementasi algoritme.

## Tipe Penelitian

Tipe penelitian yang dilakukan adalah bersifat non-implementatif dengan menggunakan pendekatan analitik. Penelitian bertipe non-implementatif adalah penelitian yang menguji hubungan terhadap suatu kejadian yang kemudian akan di analisis. Sedangkan pendekatan analitik memiliki fungsi untuk menjelaskan hubungan suatu kejadian dengan suatu objek penelitian yang sedang diteliti.

## Strategi Penelitian

Strategi penelitian ini menggunakan studi kasus analisis sentimen masyarakat terhadap kuliah daring yang didapat dari Twitter. Data tersebut dilabeli manual oleh pakar lalu dibagi menjadi data latih dan data uji. Studi eksperimen berfokus kepada pengujian pada parameter X, Y, dan L pada *Term Based Random Sampling.*

## Subjek Penelitian

Subjek penelitian yang digunakan pada penelitian ini adalah pengguna Twitter yang membahas mengenai kuliah daring.

## Peralatan Pendukung

Peralatan pendukung yang digunakan pada penelitian ini adalah:

Tabel 3.1 Spesifikasi Hardware

|  |  |
| --- | --- |
| Spesifikasi | Keterangan |
| Laptop | Dell XPS 15 9575 |
| CPU | Core i7-8750G |
| GPU | NVIDIA GeForce GTX 1050 (4GB GDDR5) |
| RAM | 16 GB |
| Tipe Memori | DDR4 |
| SSD | 512GB SSD PCIe NVMe |

Tabel 3.2 Spesifikasi Software

|  |  |
| --- | --- |
| Jenis | Keterangan |
| *Operating System* | MacOS Catalina 10.15.4 |
| Bahasa Pemrograman | Python 3.7.7 |
| IDE | Visual Studio Code |
| Library | Sastrawi, Pandas, Numpy, Re, Math |

## Lokasi Penelitian

Lokasi penelitian ini bertempat di Laboratorium Komputasi Cerdas, Fakultas Ilmu Komputer Universitas Brawijaya.

## Teknik Pengumpulan Data

Teknik yang digunakan untuk pengumpulan data pada penelitian ini berasal dari Pengguna Twitter. Data diambil menggunakan *library* *Twint* yang berfungsi sebagai *data scrapper* Twittter untuk *Python*. Kata kunci yang digunakan pada saat pengumpulan data adalah “Kuliah Daring” dan “Kuliah Online”. Pengumpulan data dilakukan dalam 7 bulan terhitung sejak April 2020 hingga Oktober 2020. Data yang dikumpulkan dilakukan proses normalisasi secara manual terlebih dahulu, kata yang dinormalisasi seperti berupa kata singkatan, kata tidak baku, dan kata-kata yang memiliki kesalahan penulisan.

## Data Penelitian

Pada penelitian kali ini, data yang digunakan adalah *tweet* berbahasa Indonesia. Total dokumen yang akan diambil dari Twitter adalah 300 dokumen dimana dari 300 dokumen akan dibagi menjadi 240 data latih, dan 60 data uji. Proses klasifikasi akan dibagi menjadi 3 yaitu positif, netral dan negatif.

## Teknik Analisis Data

Teknik Analisis Data pada penelitian ini ditujukan untuk mengetahui kinerja dari sistem yang telah dibuat sesuai algoritma yang diajukan oleh peneliti. Tingkat kinerja sistem diperoleh dengan menggunakan *Confusion Matrix* dan nantinya hasil yang diterima akan dimasukan ke dalam tabel *Confusion Matrix* dan dicari nilai *precision, recall, accuracy, dan f-measure* pada tiap iterasi *fold* dalam *K-fold cross validation*.

## Implementasi Algoritme

Implementasi Algoritme diawali dengan melakukan pembuatan daftar *stopword* yang proses nya diawali dengan *preprocessing* data latih dengan tujuan untuk merubah data berbentuk kumpulan dokumen menjadi *term* untuk dilakukan perhitungan algoritma *Term Based Random Sampling*. Setelah daftar *stopword* didapatkan, data latih akan melalui tahap *preprocessing* yang dimana didalamnya terdapat tahap-tahapan seperti *case folding, cleaning,* tokenisasi, *stopword removal* dengan menggunakan daftar *stopword* yang telah dibuat, lalu diakhiri dengan *stemming*. Lalu dilanjutkan dengan pembobotan kata menggunakan 𝑡𝑓. 𝑖𝑑𝑓 untuk merubah kata tersebut menjadi suatu nilai yang nantinya dapat diproses oleh sistem untuk dilatih dan diklasifikasi menggunakan metode *Multinomial* *Naïve Bayes*.

# PERANCANGAN

Pada bab ini akan dijelaskan perancangan dengan diagram alir dari metode-metode yang digunakan pada penelitian ini serta manualisasi sistem klasifikasi dengan Naïve Bayes serta *Term Based Random Sampling* sebagai metode pembentuk daftar *stopword*.

## Diagram Alir Sistem

Pada diagram ini akan dijelaskan bagaimana tahapan-tahapan dari sistem. Tahapan-tahapan tersebut dijelaskan pada Gambar 4.1.

Mulai

*Preprocessing*

Data Uji, term, stopword, likelihood, prior

Mulai

*Preprocessing*

Data Latih, stopword

Mulai

Preprocessing tanpa Filtering

Data Latih

Term Based Random Sampling

*Term Weighting*

*Naïve Bayes Testing*

Naïve Bayes Training

Term, Likelihood, Prior

Hasil Klasifikasi  
Data Uji

Selesai

Selesai

Selesai

Daftar Stopword

|  |  |  |
| --- | --- | --- |
| (a) | (b) | (c) |

Gambar 4.1 Diagram Alir Sistem, (a) Tahap Pembuatan Stopword, (b) Tahap Pelatihan, (c) Tahap Pengujian

Berdasarkan pada Gambar 4.1 terdapat 3 diagram alir yaitu diagram tahap pembuatan *stopword*, tahap pelatihan, dan tahap pengujian. Pada tahap pembuatan *stopword*, masukan berupa data latih dan diawali dengan preprocessing data latih namun tidak menggunakan tahap *filtering*. Selanjutnya dilanjutkan oleh proses pembuatan daftar *stopword* menggunakan *Term Based Random Sampling* yang memiliki hasil berupa daftar *stopword*. Daftar *stopword* ini akan digunakan dalam tahap pelatihan dan tahap pengujian.

Pada tahap pelatihan, masukan berupa data latih dan *stopword* yang sudah dibuat sebelumnya dan diawali dengan *preprocessing* yang menggunakan *stopword* yang didapatkan sebagai masukan untuk proses *filtering*. Selanjutnya dilanjutkan oleh proses *Term Weighting* menggunakan metode *term frequency – inverse document frequency*. Setelah bobot didapatkan akan dilanjutkan proses pelatihan Naïve Bayes yang akan menghasilkan *likelihood* serta *prior*. Pada tahapan ini akan menghasilkan *term*, *likelihood*, *prior* yang akan digunakan pada tahap pelatihan.

Pada tahap pengujian, masukan berupa data uji, *term*, *stopword*, *likelihood*, dan juga *prior*. Tahapan ini diawali dengan *preprocessing* data uji menggunakan stopword yang sudah dibuat untuk proses *filtering*. Selanjutnya adalah *Naïve Bayes* *Testing* dimana didalam proses ini akan terjadi perhitungan posterior untuk masing-masing kategori. Sehingga akan ditentukan kelas klasifikasi dengan melihat *posterior* tertinggi.

### Diagram Alir *Preprocessing* tanpa Filtering

Pada tahapan *Preprocessing* tanpa *Filtering* ini terdapat beberapa tahapan yaitu *case folding, cleaning, tokenizing, stemming.* Preprocessing ini memiliki perbedaan dengan *preprocessing* pada umumnya karena tidak adanya *filtering* karena tujuan tahapan ini adalah proses menghasilkan suatu daftar *stopword.* Tahapan ini akan dijelaskan pada Gambar 4.2.

­

Mulai

Preprocessing tanpa Filtering

Data Latih

cleaned\_data =[],

terms=[]

i=0,

I < jumlah data latih

Case folding

Cleaning

Stemming

A

C

B

A

C

B

Tokenizing

Perulangan kata dalam tokenizing

Jika kata belum ada di terms

Kata masuk ke terms

Hasil Stemming ditambahkan ke cleaned\_data

Selesai

Keluaran

cleaned\_data, terms

Gambar 4.2 Diagram Alir Preprocessing tanpa Filtering

### Diagram Alir *Term Based Random Sampling*

Pada tahapan *Term Based Random Sampling* ini terdapat beberapa tahapan-tahapan untuk mendapatkan stopword berdasarkan dokumen tertentu. Tahapan ini akan dijelaskan pada Gambar 4.3.

Mulai

Term Based Random Sampling

Cleaned\_data,

terms, X, Y, L

A

token\_used = []

Tidak

A

i=0, i<Y

token\_w = {},  
normalized\_term\_w = {},  
sorted\_term\_w = {}

Ambil kata acak dari terms

Ambil dokumen dari data latih yang mengandung kata acak

Cari terms dari sampled\_documents

Perulangan word dalam term\_sampled\_documents

Kullback-Leibler Divergence

Jika word belum ada di token\_used

Hasil Kullback-Leibler Divergence dimasukan kedalam token\_w[word]

A

C

D

Ya

B

E

word dimasukan ke dalam token\_used

Ambil bobot maksimum dalam token\_w

Ambil bobot minimum dalam token\_w

Normalisasi MinMax Term Weighting

A

C

D

B

E

Urutkan normalized\_term\_weight dari terendah

Jika index lebih kecil dari X

Perulangan term\_weight dalam sort\_term\_weight

Term\_weight dimasukan ke sorted\_term\_weight

sorted\_term\_weight di tambahkan ke token\_weight

A

weighted\_token = {}

Ya

Tidak

temp = []

A

Perulangan token dalam token\_used

Perulangan tok\_w dalam token\_weight

Jika used\_token ada didalam tok\_w

tok\_w dengan key used\_tok akan ditambahkan ke temp

weighted\_token dengan key used\_tok diinsialisasi dengan temp

Ya

Tidak

merged\_weighted\_token = {}

Perulangan k,v dalam setiap item weighted\_token

Jika panjang v tidak sama dengan 0

merged\_weighted\_token dengan key k diinisialisasi dengan mean v

Urutkan merged\_weighted\_token dari rendah dan dimasukan kedalam variabel sorted\_merged\_weighted\_token

A

Ya

Tidak

Ambil term sejumlah L teratas dari sorted\_merged\_weighted\_token dan masukan ke dalam variabel sorted\_final\_weight

A

Perulangan k, v dalam setiap item sorted\_final\_weight

tok\_w dengan key used\_tok akan ditambahkan ke temp

stopwords = []

Keluaran stopwords

Selesai

Gambar 4.3 Diagram Alir *Term Based Random Sampling*

Pada tahapan Term Based Random Sampling ini terdapat beberapa tahapan yaitu diawali dengan pilih term acak dari keseluruhan term, ambil dokumen yang mengandung dokumen tersebut, hitung bobot tiap term menggunakan Kullback Leibler, normalisasi bobot dengan MinMax, ambil sejumlah X term yang diurutkan dari bobot terendah, lakukan proses sebelumnya sebanyak Y kali, hitung rata-rata keseluruhan bobot tiap term, dan yang terakhir ambil sejumlah L term.

#### Diagram Alir Kullback Leibler Divergence

Pada tahapan Kullback Leibler Divergence ini terdapat perhitungan pemberian bobot term untuk mendapatkan stopword berdasarkan dokumen tertentu. Tahapan ini akan dijelaskan pada Gambar 4.4.

Mulai

Kullback Leibler Divergence

word, sampled documents, cleaned\_data, terms

A

A

Hitung jumlah word dalam sampled\_documents dimasukan kedalam tf\_x

Hitung jumlah word dalam sampled\_documents

dimasukan ke dalam l\_x

Hitung jumlah word dalam cleaned\_data

dimasukan ke dalam F

Hitung jumlah token dalam keseluruhan dokumen dimasukan ke dalam token\_c

Keluaran w(t)

Mulai

Gambar 4.4 Diagram Alir Kullback Leibler Divergence

#### Diagram Alir Normalisasi MinMax Term Weighting

Pada tahapan ini terdapat perhitungan normalisasi bobot term dengan MinMax agar bobot dalam angka 0 hingga 1. Tahapan ini akan dijelaskan pada Gambar 4.5.

Mulai

Normalisasi MinMax Term Weighting

A

token\_w, max\_weight\_term, min\_weight\_term

A

normalized\_term\_weight = {}

Perulangan k,v dalam setiap item token\_w

Keluaran normalized\_term\_weight

Selesai

Hasil normalisasi akan dimasukan ke normalized\_term\_weight dengan key k

Gambar 4.5 Diagram Alir Normalisasi MinMax Term Weighting

### Diagram Alir *Preprocessing*

Tahapan *preprocessing* ini terdapat di 2 tahap yaitu tahap pelatihan dan tahap pengujian. Adapun tahap dalam preprocessing yaitu yaitu *case folding, cleaning, tokenizing, filtering, stemming.* Tahapan ini akan dijelaskan pada Gambar 4.6.

Mulai

Preprocessing

Data Latih, stopwords

cleaned\_data =[],

terms=[]

i=0,

I < jumlah data latih

A

C

B

A

C

B

*Tokenizing*

Perulangan kata dalam tokenizing

Jika kata belum ada di terms

Kata masuk ke terms

Hasil Stemming ditambahkan ke cleaned\_data

Selesai

Keluaran

cleaned\_data, terms

*Case folding*

*Cleaning*

*Filtering*

*Stemming*

Ya

Tidak

Gambar 4.6 Diagram Alir Preprocessing

### Diagram Alir *Term Weighting*

Tahapan *Term Weighting* ini terdapat beberapa tahapan-tahapan yaitu diawali dengan menghitung *raw term frequency*, *log term frequency, inverse document frequency,* dan *term frequency – inverse document frequency*. Tahapan ini akan dijelaskan pada Gambar 4.7.

Mulai

cleaned\_data, terms

*Raw Term Frequency*

*Log Term Frequency*

*Inverse Document Frequency*

*Term Frequency – Inverse Document Frequency*

Keluaran tf\_idf

Selesai

Term

Weighting

Gambar 4.7 Diagram Alir Term Weighting

#### Diagram Alir Raw Term Weighting

Tahapan *Raw Term Weighting* ini bertujuan untuk menghitung frekuensi setiap term yang terdapat dalam dokumen. Tahapan ini akan dijelaskan pada Gambar 4.8.

Mulai

*Raw Term Weighting*

cleaned\_data, terms

A

C

B

Perulangan term didalam terms

raw\_tf = {}

Keluaran raw\_tf

Selesai

A

C

B

temp = []

Perulangan data didalam cleaned\_data

Menghitung kemunculan term pada cleaned\_data dan dimasukan ke temp

raw\_tf dengan key term diinisialisasi dengan temp

Gambar 4.8 Diagram Alir *Raw Term Wighting*

#### Diagram Alir Log Term Weighting

Tahapan *Log Term Weighting* ini bertujuan untuk menghitung frekuensi setiap term yang terdapat dalam dokumen lalu di logaritma. Tahapan ini akan dijelaskan pada Gambar 4.9.

Mulai

*Log Term Weighting*

raw\_tf, terms

A

C

B

Perulangan term didalam terms

log\_tf = {}

Keluaran log\_tf

Selesai

A

C

B

temp = []

i = 0, i < panjang raw\_tf dengan key term

logtf dimasukkan ke temp

log\_tf[term] = temp

Jika raw\_tf key term dengan index i tidak sama dengan 0

Ya

Tidak

Gambar 4.9 Diagram Alir *Log Term Wighting*

#### Diagram Alir Inverse Document Frequency

Tahapan *Inverse Document Frequency* ini bertujuan untuk perhitungan inverse terhadap frekuensi dokumen yang mengandung kata tersebut. Tahapan ini akan dijelaskan pada Gambar 4.10.

Mulai

*Inverse Document Frequency*

A

raw\_tf, terms, cleaned\_data

idf = []

A

Perulangan term dalam terms

Hitung jumlah frekuensi dokumen yang memiliki term

Keluaran idf

Selesai

Hitung 𝑖𝑑𝑓(𝑡) =

Masukan hasil idf(t) ke list idf

Gambar 4.10 Diagram Alir *Inverse Document Frequency*

#### Diagram Alir Term Frequency - Inverse Document Frequency

Tahapan *Term Frequency - Inverse Document Frequency* ini bertujuan untuk mengkalikan *log term frequency* dengan *inverse document frequency*. Tahapan ini akan dijelaskan pada Gambar 4.11.

Mulai

*Term Frequency – Inverse Document Frequency*

terms, log\_tf, idf

Perulangan term didalam terms

tf\_idf = {}, count = 0

temp = []

i = 0, I < panjang dokumen

A

C

D

B

E

Hasil

log\_tf[term][i] \* idf[count] dimasukan ke temp

A

C

D

B

E

tf\_idf[term] = temp,

count+=1

Keluaran tf\_idf

Selesai

Gambar 4.11 Diagram Alir *Term Frequency - Inverse Document Frequency*

### Diagram Alir *Naïve Bayes Training*

Pada tahapan Naïve Bayes Training ini terdapat beberapa tahapan yaitu mencari likelihood setiap kelas serta mencari prior tiap kelasnya. Tahapan ini akan dijelaskan pada Gambar 4.12.

Mulai

*Term Frequency – Inverse Document Frequency*

cleaned\_data, terms, target, stopword, tf\_idf

i = 0, i < panjang cleaned\_data

total = []

total\_word = 0

Perulangan term didalam terms

total\_word += tf\_idf[term][i]

A

C

B

A

C

B

total\_word tambahkan ke total

con\_prob\_negative = [], con\_prob\_neutral = [],

con\_prob\_positive = []

Perulangan term didalam terms

Hitung Likelihood term tiap kelas

Likelihood\_negatif dimasukan ke con\_prob\_negatif

Likelihood\_ neutral dimasukan ke con\_prob\_neutral

Likelihood\_ positive dimasukan ke con\_prob\_positive

terms\_con\_prob = {},

i = 0

Perulangan term didalam terms

con\_prob\_negatif index i

ditambahkan ke temp

temp = []

con\_prob\_neutral index i

ditambahkan ke temp

con\_prob\_positive index i

ditambahkan ke temp

i +=1

A

A

Hitung Prior tiap kelas

Keluaran likelihood, prior, term

Selesai

Gambar 4.12 Diagram Alir *Naive Bayes Training*

#### Diagram Alir Hitung Likelihood term tiap kelas

Pada tahapan Hitung *Likelihood term* tiap Kelas ini yaitu mencari likelihood term tertentu setiap kelasnya. Tahapan ini akan dijelaskan pada Gambar 4.13.

Mulai

Hitung Likelihood term tiap kelas

kata, kategori

total = []

Hitung jumlah word didalam kategori dimasukan kedalam count\_w\_c

Hitung jumlah semua term didalam kategori dimasukan ke dalam count\_c

Hitung jumlah semua term dimasukan ke dalam V

Keluaran P(w|c)

Selesai

Gambar 4.13 Diagram Alir Hitung Likelihood term tiap kelas

#### Diagram Alir Hitung Prior tiap kelas

Pada tahapan Hitung *Prior* tiap Kelas ini yaitu mencari prior setiap kelasnya. Tahapan ini akan dijelaskan pada Gambar 4.14.

Mulai

Hitung Likelihood term tiap kelas

kelas

prior\_negatif, prior\_neutral, prior\_positive

Hitung total dokumen dikelas negatif dibagi total keseluruhan dokumen dan hasil dimasukan ke dalam prior\_negatif

Keluaran prior\_negatif, prior\_neutral, prior\_positive

Selesai

Hitung total dokumen dikelas negatif dibagi total keseluruhan dokumen dan hasil dimasukan ke dalam prior\_negatif

Hitung total dokumen dikelas negatif dibagi total keseluruhan dokumen dan hasil dimasukan ke dalam prior\_negatif

Gambar 4.14 Diagram Alir Hitung Prior tiap kelas

### Diagram Alir *Naïve Bayes Testing*

Pada tahapan Naïve Bayes Testing ini berfungsi untuk menghitung posterior setiap kelasnya dari data uji. Tahapan ini akan dijelaskan pada Gambar 4.15.

Mulai

*Naïve Bayes Testing*

terms, cleaned\_data\_test, terms\_test

A

A

used\_terms = []

Perulangan term dalam terms\_test

Jika term ada di terms

term ditambahkan ke used\_terms

used\_terms\_with\_likelihood = {}

Perulangan term dalam used\_terms

temp = []

likelihood[term][0]

dimasukkan ke temp

likelihood[term][1]

dimasukkan ke temp

likelihood[term][2]

dimasukkan ke temp

temp dimasukkan ke

used\_terms\_with\_likelihood[term]

negatif = 1,

netral = 1,

positif = 1

A

A

Perulangan term dalam used\_terms

negatif\*= used\_terms\_with\_likelihood[term][0]

netral\*= used\_terms\_with\_likelihood[term][1]

positif\*= used\_terms\_with\_likelihood[term][2]

negatif = negatif \* prior\_negative

netral = netral \* prior\_ neutral

positif = positif \* prior\_positive

Jika negatif lebih besar dari netral dan positif

Selain itu

Jika positif lebih besar dari negatif dan netral

final\_result = “Positif”

final\_result = “Netral”

final\_result = “Negatif”

Keluaran final\_result

Selesai

Ya

Ya

Ya

Tidak

Tidak

Gambar 4.15 Diagram Alir Naive Bayes Testing

## Manualisasi

Pada perhitungan manual ini akan diawali dengan persiapan data dan tahapan manualisasi akan dibagi menjadi 3 tahapan yaitu pembuatan daftar stopword, pelatihan, dan pengujian.

### Persiapan Data

Data yang digunakan berupa tweet dari pengguna Twitter yang memiliki kuliah daring atau kuliah online sebagai kata kuncinya. Dalam proses perhitungan manualisasi ini akan digunakan 9 dokumen data latih dan 1 dokumen sebagai data uji. Berikut adalah sampel data yang digunakan yang ditunjukkan pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Data Latih

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | Aku selama kuliah online benar-benar tidak belajar sama sekali. Ujian selalu tidak jujur, tugas tinggal memindahkan dari internet, dosen hanya memberi tugas, tidak pernah ada penjelasan materi. Ditambah semester 5 mau tetap daring, mau jadi apa Aku | Negatif |
| 2. | Rasanya mau berhenti kuliah saja kalau daring begini, seperti bayar cuma cuma, materi dikasih secara online, disuruh baca sendiri tanpa ada yang menjelaskan, berasa otodidak :" | Negatif |
| 3. | Maaf, aku kuliah daring semakin malas. Kelas online saja ketiduran. Baik darimananya coba? Nilai sempurna bukan karena kita yang cerdas tapi karena dosennya yang kasihan sama kitanya. Tapi secara pemahaman, kosong sekali otak ini. Terima kasih | Negatif |
| 4. | Sejujurnya aku oke-oke saja dengan kuliah daring. Cuma ya itu, kangen sama suasana kelas. Kalau corona sudah selesai, perpaduan offline-online sepertinya asik.... | Netral |
| 5. | Ada yang mempeributkan masalah kuliah online/daring, sebagian ada yang menyalahkan dosen ada juga yang menyalahkan diri sendiri. Mau kuliah online atau tidak semua tergantung pribadi masing-masing dalam memahami materi yang dikasih dosen:) | Netral |
| 6. | Pak ini gimana anak sekolahan offline untuk beberapa zona, tapi kenapa mahasiswa tetap melaksanakan kuliah secara online / daring, justru mahasiswa lebih bisa beradaptasi dengan new normal dibandingkan dengan anak-anak yang masih sangat rentan, mohon dikaji lagi pak | Netral |
| 7. | Saya berdoa kuliah tetap daring saja, kampus mau offline padahal tempat masih zona merah dan kerabat aku yang kerjanya dokter saja suka bilang lagi kerja keras karena pasien corona. Lebih nyaman online, tetap dirumah adalah jalanku | Positif |
| 8. | Nilai positif saja yang diambil buang yang negatif. Positifnya (mungkin) ada beberapa mahasiswa yang tidak berani bertanya di kelas jadi lebih aktif bertanya di kuliah online (daring) | Positif |
| 9. | Benar juga ya lama lama kuliah online jadi new normal sampai masa pandemi ini selesai juga bisa jadi online, kalau bisa daring kenapa harus kuliah offline ðÿ˜œ | Positif |

Tabel 4.2 Data Uji

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | Apa saya saja yang merasa kalau selama kuliah daring  nyaman banget sampai saya tidak ingin masuk kuliah karena  takut panik | ? |
| 2. | Aku merasa lebih leluasa dengan kuliah daring, tidak capek harus siap-siap berangkat. Hanya tinggal makan, beres didepan komputer sudah siap nyimak. Buat materi, selama online emang tidak pernah mengandalkan dosen atau teman. Jadi lebih banyak waktu buat searching sama buka text book. | ? |
| 3. | Jujur tidak ada senang-senangnya kuliah daring. Aku butuh praktik lapangan. Apalagi semester depan magang. Apa magang online juga? Bisa stres gara-gara banyak deadline | ? |
| 4. | Tatap langsung aja kadang tidak paham, apalagi kuliah daring, belum lagi jaringan lambat­ ditambah beberapa dosen yang jarang memberi kuliah online, atau cuma memberi tugas saja... Fix kampus ku belum siap menerapkan kuliah daring! pic.twitter.com/UHdReyLgh8 | ? |
| 5. | Orang lain pada ribut sama keadaan kosan yang sudah ditinggal berbulan-bulan terus ribut gimana caranya balik ke kosan. Aku anteng-anteng saja jadi penghuni kos dari awal pemerintah nyuruh dirumah saja dan kuliah jadi daring | ? |

### Manualisasi Pembuatan Daftar Stopword

Pada tahapan ini dilakukan manualisasi pembuatan daftar stopword dengan menggunakan *Term Based Random Sampling.*

Berikut adalah data yang digunakan untuk pembuatan stopword yang sudah melalui proses case folding, cleaning, tokenisasi yang dapat dilihat pada Tabel 4.3.

Tabel 4.3 Data Manualisasi Pembuatan Stopword yang sudah di Preprocessing

|  |  |
| --- | --- |
| No | Tweet |
| 1. | aku lama kuliah online benar benar tidak ajar sama sekali uji selalu tidak jujur tugas tinggal pindah dari internet dosen hanya beri tugas tidak pernah ada jelas materi tambah semester mau tetap daring mau jadi apa aku |
| 2. | rasa mau henti kuliah saja kalau daring begini seperti bayar cuma cuma materi kasih cara online suruh baca sendiri tanpa ada yang jelas asa otodidak |
| 3. | maaf aku kuliah daring makin malas kelas online saja tidur baik darimananya coba nilai sempurna bukan karena kita yang cerdas tapi karena dosen yang kasihan sama kita tapi cara paham kosong sekali otak ini terima kasih |
| 4. | jujur aku oke oke saja dengan kuliah daring cuma ya itu kangen sama suasana kelas kalau corona sudah selesai padu offline online seperti asik |
| 5. | ada yang ribut masalah kuliah online daring bagi ada yang salah dosen ada juga yang salah diri sendiri mau kuliah online atau tidak semua gantung pribadi masing masing dalam paham materi yang kasih dosen |
| 6. | pak ini gimana anak sekolah offline untuk beberapa zona tapi kenapa mahasiswa tetap laksana kuliah cara online daring justru mahasiswa lebih bisa adaptasi dengan new normal banding dengan anak anak yang masih sangat rentan mohon kaji lagi pak |
| 7. | saya doa kuliah tetap daring saja kampus mau offline padahal tempat masih zona merah dan kerabat aku yang kerja dokter saja suka bilang lagi kerja keras karena pasien corona lebih nyaman online tetap rumah adalah jalan |
| 8. | nilai positif saja yang ambil buang yang negatif positif mungkin ada beberapa mahasiswa yang tidak berani tanya di kelas jadi lebih aktif tanya di kuliah online daring |
| 9. | benar juga ya lama lama kuliah online jadi new normal sampai masa pandemi ini selesai juga bisa jadi online kalau bisa daring kenapa harus kuliah offline |

Dan setelah preprocessing tersebut didapatkan sejumlah term sebagai berikut yang dapat dilihat di Tabel 4.4.

Tabel 4.4 Term Manualisasi Pembuatan Daftar Stopword

|  |
| --- |
| Term |
| 'aku', 'lama', 'kuliah', 'online', 'benar', 'tidak', 'ajar', 'sama', 'sekali', 'uji', 'selalu', 'jujur', 'tugas', 'tinggal', 'pindah', 'dari', 'internet', 'dosen', 'hanya', 'beri', 'pernah', 'ada', 'jelas', 'materi', 'tambah', 'semester', 'mau', 'tetap', 'daring', 'jadi', 'apa', 'rasa', 'henti', 'saja', 'kalau', 'begini', 'seperti', 'bayar', 'cuma', 'kasih', 'cara', 'suruh', 'baca', 'sendiri', 'tanpa', 'yang', 'asa', 'otodidak', 'maaf', 'makin', 'malas', 'kelas', 'tidur', 'baik', 'darimananya', 'coba', 'nilai', 'sempurna', 'bukan', 'karena', 'kita', 'cerdas', 'tapi', 'kasihan', 'paham', 'kosong', 'otak', 'ini', 'terima', 'oke', 'dengan', 'ya', 'itu', 'kangen', 'suasana', 'corona', 'sudah', 'selesai', 'padu', 'offline', 'asik', 'ribut', 'masalah', 'bagi', 'salah', 'juga', 'diri', 'atau', 'semua', 'gantung', 'pribadi', 'masing', 'dalam', 'pak', 'gimana', 'anak', 'sekolah', 'untuk', 'beberapa', 'zona', 'kenapa', 'mahasiswa', 'laksana', 'justru', 'lebih', 'bisa', 'adaptasi', 'new', 'normal', 'banding', 'masih', 'sangat', 'rentan', 'mohon', 'kaji', 'lagi', 'saya', 'doa', 'kampus', 'padahal', 'tempat', 'merah', 'dan', 'kerabat', 'kerja', 'dokter', 'suka', 'bilang', 'keras', 'pasien', 'nyaman', 'rumah', 'adalah', 'jalan', 'positif', 'ambil', 'buang', 'negatif', 'mungkin', 'berani', 'tanya', 'di', 'aktif', 'sampai', 'masa', 'pandemi', 'harus' |

Setelah term didapatkan langkah selanjutnya adalah proses pembuatan stopword dengan Term Based Random Sampling. Dalam algoritma *Term Based Random Sampling* ini memiliki beberapa parameter yang harus ditentukan. X sebagai jumlah angka yang diambil dari urutan tertinggi dari tiap perulangan, Y adalah jumlah perulangan pemilihan kata acak, dan L adalah jumlah stopword yang ingin dibuat.

Dalam proses manualisasi ini akan digunakan Y = 50, X = 30, dan L akan diambil 20 persen dari keseluruhan. Adapun langkah-langkahnya akan dijelaskan sebagai berikut:

1. Pilih term acak dari keseluruhan term
2. Ambil dokumen yang mengandung term tersebut dan dokumen tersebut akan menjadi dokumen sampel. Dokumen sampel yang diambil akan ditampilkan pada Tabel 4.5.

Tabel 4.5 Dokumen Sampel Manualisasi Pembuatan Daftar Stopword

|  |  |
| --- | --- |
| No | Tweet |
| 1. | aku lama kuliah online benar benar tidak ajar sama sekali uji selalu tidak jujur tugas tinggal pindah dari internet dosen hanya beri tugas tidak pernah ada jelas materi tambah semester mau tetap daring mau jadi apa aku |
| 2. | rasa mau henti kuliah saja kalau daring begini seperti bayar cuma cuma materi kasih cara online suruh baca sendiri tanpa ada yang jelas asa otodidak |
| 3. | ada yang ribut masalah kuliah online daring bagi ada yang salah dosen ada juga yang salah diri sendiri mau kuliah online atau tidak semua gantung pribadi masing masing dalam paham materi yang kasih dosen |

1. Cari term dari dokumen sampel atau dokumen yang diambil. Term yang diambil dari dokumen sampel akan ditampilkan pada Tabel 4.6.

Tabel 4.6 Term Dokumen Sampel Manualisasi Pembuatan Daftar Stopword

|  |
| --- |
| Term dari dokumen sampel |
| 'aku', 'lama', 'kuliah', 'online', 'benar', 'tidak', 'ajar', 'sama', 'sekali', 'uji', 'selalu', 'jujur', 'tugas', 'tinggal', 'pindah', 'dari', 'internet', 'dosen', 'hanya', 'beri', 'pernah', 'ada', 'jelas', 'materi', 'tambah', 'semester', 'mau', 'tetap', 'daring', 'jadi', 'apa', 'rasa', 'henti', 'saja', 'kalau', 'begini', 'seperti', 'bayar', 'cuma', 'kasih', 'cara', 'suruh', 'baca', 'sendiri', 'tanpa', 'yang', 'asa', 'otodidak', 'ribut', 'masalah', 'bagi', 'salah', 'juga', 'diri', 'atau', 'semua', 'gantung', 'pribadi', 'masing', 'dalam', 'paham' |

1. Hitung bobot tiap term menggunakan Kullback-Leibler

Berikut adalah beberapa contoh kata dalam perhitungan Kullback Leibler.

* 1. Kata “aku”

Setelah perhitungan diatas, kata “aku” mendapatkan nilai bobot sebesar -0.01473.

* 1. Kata “lama”

Setelah perhitungan diatas, kata “lama” mendapatkan nilai bobot sebesar -0.01010.

* 1. Kata “kuliah”

Setelah perhitungan diatas, kata “kuliah” mendapatkan nilai bobot sebesar -0.03519.

Dan dalam proses iterasi pertama ini, didapatkan nilai sebagai berikut yang akan ditampilkan dalam Tabel 4.7.

Tabel 4.7 Hasil Kullback Leibler Manualisasi

|  |  |
| --- | --- |
| **Term** | **Bobot Kullback Leibler** |
| aku | -0.014734 |
| lama | -0.010107 |
| kuliah | -0.035197 |
| online | -0.035197 |
| benar | 0.000620 |
| tidak | 0.012199 |
| ajar | 0.006403 |
| sama | -0.010107 |
| sekali | -0.004013 |
| uji | 0.006403 |
| selalu | 0.006403 |
| jujur | -0.004013 |
| tugas | 0.012806 |
| tinggal | 0.006403 |
| pindah | 0.006403 |
| dari | 0.006403 |
| internet | 0.006403 |
| dosen | 0.006240 |
| hanya | 0.006403 |
| beri | 0.006403 |
| pernah | 0.006403 |
| ada | 0.018316 |
| jelas | 0.012806 |
| materi | 0.019210 |
| tambah | 0.006403 |
| semester | 0.006403 |
| mau | 0.012199 |
| tetap | -0.014430 |
| daring | -0.030320 |
| jadi | -0.014430 |
| apa | 0.006403 |
| rasa | 0.006403 |
| henti | 0.006403 |
| saja | -0.020523 |
| kalau | -0.010107 |
| begini | 0.006403 |
| seperti | -0.004013 |
| bayar | 0.006403 |
| cuma | 0.000620 |
| kasih | 0.000620 |
| cara | -0.010107 |
| suruh | 0.006403 |
| baca | 0.006403 |
| sendiri | 0.012806 |
| tanpa | 0.006403 |
| yang | -0.033767 |
| asa | 0.006403 |
| otodidak | 0.006403 |
| ribut | 0.006403 |
| masalah | 0.006403 |
| bagi | 0.006403 |
| salah | 0.012806 |
| juga | -0.010107 |
| diri | 0.006403 |
| atau | 0.006403 |
| semua | 0.006403 |
| gantung | 0.006403 |
| pribadi | 0.006403 |
| masing | 0.012806 |
| dalam | 0.006403 |
| paham | -0.004013 |

1. Normalisasi bobot menggunakan MinMax agar didalam range 0 hingga 1.

Dalam perhitungan normalisasi MinMax, diperlukan untuk mencari nilai maksimum dan nilai minimum dari bobot yang sudah kita hitung sebelumnya yaitu:

Setelah minimum dan maximum didapatkan, proses normalisasi dapat dilakukan dengan cara sebagai berikut.

Berikut adalah contoh perhitungan normalisasi MinMax menggunakan data sebelumnya.

* 1. Bobot “aku” = -0.01473

Sehingga bobot “aku” yang sebelumnya adalah -0.01473 telah dinormalisasi menjadi 0.37611.

* 1. Bobot “lama” = -0.01010

Sehingga bobot “lama” yang sebelumnya adalah -0.01010 telah dinormalisasi menjadi 0.46115.

* 1. Bobot “kuliah” = -0.03519

Sehingga bobot “kuliah” yang sebelumnya adalah -0.03519 telah dinormalisasi menjadi 0.0.

Sehingga setelah melalui proses normalisasi, didapatkan nilai bobot sebagai berikut yang akan ditampilkan dalam Tabel 4.8.

Tabel 4.8 Hasil Normalisasi Kullback Leibler Manualisasi

|  |  |
| --- | --- |
| **Term** | **Bobot Kullback Leibler** |
| aku | 0.376114 |
| lama | 0.461158 |
| kuliah | 0.000000 |
| online | 0.000000 |
| benar | 0.658313 |
| tidak | 0.871147 |
| ajar | 0.764615 |
| sama | 0.461158 |
| sekali | 0.573155 |
| uji | 0.764615 |
| selalu | 0.764615 |
| jujur | 0.573155 |
| tugas | 0.882308 |
| tinggal | 0.764615 |
| pindah | 0.764615 |
| dari | 0.764615 |
| internet | 0.764615 |
| dosen | 0.761610 |
| hanya | 0.764615 |
| beri | 0.764615 |
| pernah | 0.764615 |
| ada | 0.983582 |
| jelas | 0.882308 |
| materi | 1.000000 |
| tambah | 0.764615 |
| semester | 0.764615 |
| mau | 0.871147 |
| tetap | 0.381695 |
| daring | 0.089628 |
| jadi | 0.381695 |
| apa | 0.764615 |
| rasa | 0.764615 |
| henti | 0.764615 |
| saja | 0.269697 |
| kalau | 0.461158 |
| begini | 0.764615 |
| seperti | 0.573155 |
| bayar | 0.764615 |
| cuma | 0.658313 |
| kasih | 0.658313 |
| cara | 0.461158 |
| suruh | 0.764615 |
| baca | 0.764615 |
| sendiri | 0.882308 |
| tanpa | 0.764615 |
| yang | 0.026281 |
| asa | 0.764615 |
| otodidak | 0.764615 |
| ribut | 0.764615 |
| masalah | 0.764615 |
| bagi | 0.764615 |
| salah | 0.882308 |
| juga | 0.461158 |
| diri | 0.764615 |
| atau | 0.764615 |
| semua | 0.764615 |
| gantung | 0.764615 |
| pribadi | 0.764615 |
| masing | 0.882308 |
| dalam | 0.764615 |
| paham | 0.573155 |

1. Ambil sejumlah X term (dimana X adalah parameter) yang diurutkan dari bobot terendah. Dalam manualisasi nilai X yang digunakan adalah 30. Berikut adalah 30 data terendah yang ditampilkan dalam Tabel 4.9.

Tabel 4.9 Hasil 30 Bobot Terendah

|  |  |
| --- | --- |
| **Term** | **Bobot Kullback Leibler** |
| kuliah | 0.000000 |
| online | 0.000000 |
| yang | 0.026281 |
| daring | 0.089628 |
| saja | 0.269697 |
| aku | 0.376114 |
| tetap | 0.381695 |
| jadi | 0.381695 |
| lama | 0.461158 |
| sama | 0.461158 |
| kalau | 0.461158 |
| cara | 0.461158 |
| juga | 0.461158 |
| sekali | 0.573155 |
| jujur | 0.573155 |
| seperti | 0.573155 |
| paham | 0.573155 |
| benar | 0.658313 |
| cuma | 0.658313 |
| kasih | 0.658313 |
| dosen | 0.761610 |
| ajar | 0.764615 |
| uji | 0.764615 |
| selalu | 0.764615 |
| tinggal | 0.764615 |
| pindah | 0.764615 |
| dari | 0.764615 |
| internet | 0.764615 |
| hanya | 0.764615 |
| beri | 0.764615 |

Setelah bobot normalisasi didapatkan, simpan bobot tersebut dan kumpulkan dalam suatu *dictionary* dengan kata kunci term tersebut untuk dicari rata-ratanya ditahapan berikutnya.

1. Lakukan proses 1 hingga 7 sebanyak Y kali (dimana Y adalah parameter). Dimana dalam proses manualisasi ini Y adalah 50. Sehingga nanti tiap masing-masing term memiliki sejumlah bobot berbeda-beda yang didapatkan tiap perulangan sebanyak Y kali. Berikut adalah contoh sampel term yang memiliki beberapa sampel bobot yang didapatkan di setiap perulangannya disaat perulangan sudah selesai yang akan ditampilkan pada Tabel 4.10.

Tabel 4.10 Sampel Hasil Keseluruhan Bobot Tiap Iterasi

|  |  |
| --- | --- |
| **Term** | **Kumpulan Bobot** |
| aku | 0.37611, 0.22853, 0.51485, 0.20722 |
| lama | 0.46115, 0.61264 |
| kuliah | 0.00000, 0.02525, 0.11938 |

1. Setelah itu hitung rata-rata keseluruhan bobot yang didapatkan setiap term
   1. Kata “aku”
   2. Kata “lama”
   3. Kata “kuliah”

Setelah tiap term dicari rata-ratanya, maka dihasilkan hasil akhir bobot setiap term yang akan ditampilkan pada Tabel 4.11.

Tabel 4.11 Hasil Rata-rata Keseluruhan Bobot

|  |  |
| --- | --- |
| **Term** | **Bobot Rata-rata** |
| aku | 0.327350 |
| lama | 0.493203 |
| kuliah | 0.024470 |
| online | 0.024470 |
| benar | 0.656800 |
| tidak | 0.391615 |
| ajar | 0.688190 |
| sama | 0.424298 |
| sekali | 0.527057 |
| uji | 0.688190 |
| selalu | 0.688190 |
| jujur | 0.520274 |
| tinggal | 0.688190 |
| pindah | 0.688190 |
| dari | 0.688190 |
| internet | 0.688190 |
| dosen | 0.516360 |
| hanya | 0.688190 |
| beri | 0.680499 |
| pernah | 0.634799 |
| ada | 0.453460 |
| jelas | 0.541732 |
| materi | 0.409377 |
| tambah | 0.634799 |
| semester | 0.634799 |
| mau | 0.299504 |
| tetap | 0.371090 |
| daring | 0.070763 |
| jadi | 0.399814 |
| apa | 0.634799 |
| rasa | 0.728583 |
| henti | 0.728583 |
| saja | 0.286343 |
| kalau | 0.480921 |
| begini | 0.728583 |
| seperti | 0.542063 |
| bayar | 0.728583 |
| cuma | 0.659504 |
| kasih | 0.449530 |
| cara | 0.441262 |
| suruh | 0.728583 |
| baca | 0.728583 |
| sendiri | 0.476301 |
| tanpa | 0.728583 |
| yang | 0.171075 |
| asa | 0.721197 |
| otodidak | 0.721197 |
| ribut | 0.600122 |
| masalah | 0.600122 |
| bagi | 0.600122 |
| salah | 1.000.000 |
| juga | 0.405606 |
| diri | 0.600122 |
| atau | 0.600122 |
| semua | 0.600122 |
| gantung | 0.600122 |
| pribadi | 0.600122 |
| masing | 1.000.000 |
| dalam | 0.600122 |
| paham | 0.505293 |
| maaf | 0.630228 |
| makin | 0.630228 |
| malas | 0.630228 |
| kelas | 0.398823 |
| tidur | 0.630228 |
| baik | 0.630228 |
| darimananya | 0.630228 |
| coba | 0.630228 |
| nilai | 0.477369 |
| sempurna | 0.630228 |
| bukan | 0.630228 |
| karena | 0.539828 |
| cerdas | 0.630228 |
| tapi | 0.477876 |
| kasihan | 0.630228 |
| kosong | 0.630228 |
| otak | 0.630228 |
| ini | 0.405482 |
| terima | 0.630228 |
| oke | 1.000.000 |
| dengan | 0.522447 |
| ya | 0.534539 |
| itu | 0.629438 |
| kangen | 0.629438 |
| suasana | 0.629438 |
| corona | 0.482842 |
| sudah | 0.629438 |
| selesai | 0.534539 |
| padu | 0.629438 |
| offline | 0.379735 |
| asik | 0.629438 |
| saya | 0.652592 |
| doa | 0.638495 |
| kampus | 0.638495 |
| padahal | 0.638495 |
| tempat | 0.638495 |
| masih | 0.493741 |
| zona | 0.448584 |
| merah | 0.638495 |
| dan | 0.638495 |
| kerabat | 0.638495 |
| dokter | 0.638495 |
| suka | 0.638495 |
| bilang | 0.638495 |
| lagi | 0.448584 |
| keras | 0.638495 |
| pasien | 0.638495 |
| lebih | 0.427917 |
| nyaman | 0.638495 |
| rumah | 0.638495 |
| adalah | 0.638495 |
| positif | 1.000.000 |
| ambil | 0.585926 |
| buang | 0.585926 |
| negatif | 0.585926 |
| mungkin | 0.585926 |
| beberapa | 0.410118 |
| mahasiswa | 0.536879 |
| berani | 0.585926 |
| tanya | 1.000.000 |
| di | 1.000.000 |
| aktif | 0.585926 |
| new | 0.401453 |
| normal | 0.401453 |
| sampai | 0.620283 |
| masa | 0.641505 |
| pandemi | 0.641505 |
| bisa | 0.318457 |
| kenapa | 0.401453 |
| harus | 0.641505 |
| gimana | 0.553001 |
| anak | 0.818182 |
| sekolah | 0.553001 |
| untuk | 0.553001 |
| laksana | 0.553001 |
| justru | 0.553001 |
| adaptasi | 0.553001 |
| banding | 0.553001 |
| sangat | 0.553001 |
| rentan | 0.553001 |
| mohon | 0.502798 |
| kaji | 0.502798 |

1. Urutkan bobot secara meningkat dari bobot terendah hingga bobot tertinggi. Hasil rata-rata bobot yang sudah diurutkan dapat dilihat di Tabel 4.12.

Tabel 4.12 Hasil Rata-rata Bobot yang sudah diurutkan

|  |  |
| --- | --- |
| **Term** | **Bobot Rata-rata** |
| kuliah | 0.024470 |
| online | 0.024470 |
| daring | 0.070763 |
| yang | 0.171075 |
| saja | 0.286343 |
| mau | 0.299504 |
| bisa | 0.318457 |
| aku | 0.327350 |
| tetap | 0.371090 |
| offline | 0.379735 |
| tidak | 0.391615 |
| kelas | 0.398823 |
| jadi | 0.399814 |
| new | 0.401453 |
| normal | 0.401453 |
| kenapa | 0.401453 |
| ini | 0.405482 |
| juga | 0.405606 |
| materi | 0.409377 |
| beberapa | 0.410118 |
| sama | 0.424298 |
| lebih | 0.427917 |
| cara | 0.441262 |
| zona | 0.448584 |
| lagi | 0.448584 |
| kasih | 0.449530 |
| ada | 0.453460 |
| sendiri | 0.476301 |
| nilai | 0.477369 |
| tapi | 0.477876 |
| kalau | 0.480921 |
| corona | 0.482842 |
| lama | 0.493203 |
| masih | 0.493741 |
| mohon | 0.502798 |
| kaji | 0.502798 |
| paham | 0.505293 |
| dosen | 0.516360 |
| jujur | 0.520274 |
| dengan | 0.522447 |
| sekali | 0.527057 |
| ya | 0.534539 |
| selesai | 0.534539 |
| mahasiswa | 0.536879 |
| karena | 0.539828 |
| jelas | 0.541732 |
| seperti | 0.542063 |
| gimana | 0.553001 |
| sekolah | 0.553001 |
| untuk | 0.553001 |
| laksana | 0.553001 |
| justru | 0.553001 |
| adaptasi | 0.553001 |
| banding | 0.553001 |
| sangat | 0.553001 |
| rentan | 0.553001 |
| ambil | 0.585926 |
| buang | 0.585926 |
| negatif | 0.585926 |
| mungkin | 0.585926 |
| berani | 0.585926 |
| aktif | 0.585926 |
| ribut | 0.600122 |
| masalah | 0.600122 |
| bagi | 0.600122 |
| diri | 0.600122 |
| atau | 0.600122 |
| semua | 0.600122 |
| gantung | 0.600122 |
| pribadi | 0.600122 |
| dalam | 0.600122 |
| sampai | 0.620283 |
| itu | 0.629438 |
| kangen | 0.629438 |
| suasana | 0.629438 |
| sudah | 0.629438 |
| padu | 0.629438 |
| asik | 0.629438 |
| maaf | 0.630228 |
| makin | 0.630228 |
| malas | 0.630228 |
| tidur | 0.630228 |
| baik | 0.630228 |
| darimananya | 0.630228 |
| coba | 0.630228 |
| sempurna | 0.630228 |
| bukan | 0.630228 |
| cerdas | 0.630228 |
| kasihan | 0.630228 |
| kosong | 0.630228 |
| otak | 0.630228 |
| terima | 0.630228 |
| pernah | 0.634799 |
| tambah | 0.634799 |
| semester | 0.634799 |
| apa | 0.634799 |
| doa | 0.638495 |
| kampus | 0.638495 |
| padahal | 0.638495 |
| tempat | 0.638495 |
| merah | 0.638495 |
| dan | 0.638495 |
| kerabat | 0.638495 |
| dokter | 0.638495 |
| suka | 0.638495 |
| bilang | 0.638495 |
| keras | 0.638495 |
| pasien | 0.638495 |
| nyaman | 0.638495 |
| rumah | 0.638495 |
| adalah | 0.638495 |
| masa | 0.641505 |
| pandemi | 0.641505 |
| harus | 0.641505 |
| saya | 0.652592 |
| benar | 0.656800 |
| cuma | 0.659504 |
| beri | 0.680499 |
| ajar | 0.688190 |
| uji | 0.688190 |
| selalu | 0.688190 |
| tinggal | 0.688190 |
| pindah | 0.688190 |
| dari | 0.688190 |
| internet | 0.688190 |
| hanya | 0.688190 |
| asa | 0.721197 |
| otodidak | 0.721197 |
| rasa | 0.728583 |
| henti | 0.728583 |
| begini | 0.728583 |
| bayar | 0.728583 |
| suruh | 0.728583 |
| baca | 0.728583 |
| tanpa | 0.728583 |
| anak | 0.818182 |
| salah | 1.000000 |
| masing | 1.000000 |
| oke | 1.000000 |
| positif | 1.000000 |
| tanya | 1.000000 |
| di | 1.000000 |

1. Ambil sejumlah L term (dimana L adalah parameter yang menentukan jumlah stopword yang ingin digunakan). Dalam manualisasi ini kita akan mencoba untuk menggunakan varian parameter L dengan nilai 20 persen.

Berikut adalah daftar stopword jika parameter L yang digunakan adalah 20 persen yang akan ditampilkan pada Tabel 4.14.

Tabel 4.13 Daftar Stopword 20 persen

|  |  |
| --- | --- |
| **Term** | **Bobot Rata-rata** |
| kuliah | 0.024470 |
| online | 0.024470 |
| daring | 0.070763 |
| yang | 0.171075 |
| saja | 0.286343 |
| mau | 0.299504 |
| bisa | 0.318457 |
| aku | 0.327350 |
| tetap | 0.371090 |
| offline | 0.379735 |
| tidak | 0.391615 |
| kelas | 0.398823 |
| jadi | 0.399814 |
| new | 0.401453 |
| normal | 0.401453 |
| kenapa | 0.401453 |
| ini | 0.405482 |
| juga | 0.405606 |
| materi | 0.409377 |
| beberapa | 0.410118 |
| sama | 0.424298 |
| lebih | 0.427917 |
| cara | 0.441262 |
| zona | 0.448584 |
| lagi | 0.448584 |
| kasih | 0.449530 |
| ada | 0.453460 |
| sendiri | 0.476301 |

### Preprocessing

Tahapan preprocessing ini adalah tahapan untuk menyiapkan data yang akan digunakan dengan mengubah data yang tidak terstruktur menjadi suatu data yang terstruktur agar dapat diolah oleh sistem. Tahapan preprocessing ini terdiri dari tahapan, case folding, cleaning, stemming, tokenizing.

#### Case Folding

Pada tahapan ini terjadi perubahan huruf kapital didalam dokumen menjadi huruf kecil. Tahapan ini dapat dilihat di Tabel 4.3 dan 4.4

Tabel 4.14 Manualisasi Case Folding Data Latih

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | aku selama kuliah online benar-benar tidak belajar sama sekali. ujian selalu tidak jujur, tugas tinggal memindahkan dari internet, dosen hanya memberi tugas, tidak pernah ada penjelasan materi. ditambah semester 5 mau tetap daring, mau jadi apa aku | Negatif |
| 2. | rasanya mau berhenti kuliah saja kalau daring begini, seperti bayar cuma cuma, materi dikasih secara online, disuruh baca sendiri tanpa ada yang menjelaskan, berasa otodidak :" | Negatif |
| 3. | maaf, aku kuliah daring semakin malas. kelas online saja ketiduran. baik darimananya coba? nilai sempurna bukan karena kita yang cerdas tapi karena dosennya yang kasihan sama kitanya. tapi secara pemahaman, kosong sekali otak ini. terima kasih | Negatif |
| 4. | sejujurnya aku oke-oke saja dengan kuliah daring. cuma ya itu, kangen sama suasana kelas. kalau corona sudah selesai, perpaduan offline-online sepertinya asik.... | Netral |
| 5. | ada yang mempeributkan masalah kuliah online/daring, sebagian ada yang menyalahkan dosen ada juga yang menyalahkan diri sendiri. mau kuliah online atau tidak semua tergantung pribadi masing-masing dalam memahami materi yang dikasih dosen:) | Netral |
| 6. | pak ini gimana anak sekolahan offline untuk beberapa zona, tapi kenapa mahasiswa tetap melaksanakan kuliah secara online / daring, justru mahasiswa lebih bisa beradaptasi dengan new normal dibandingkan dengan anak-anak yang masih sangat rentan, mohon dikaji lagi pak | Netral |
| 7. | saya berdoa kuliah tetap daring saja, kampus mau offline padahal tempat masih zona merah dan kerabat aku yang kerjanya dokter saja suka bilang lagi kerja keras karena pasien corona. lebih nyaman online, tetap dirumah adalah jalanku | Positif |
| 8. | nilai positif saja yang diambil buang yang negatif. positifnya (mungkin) ada beberapa mahasiswa yang tidak berani bertanya di kelas jadi lebih aktif bertanya di kuliah online (daring) | Positif |
| 9. | benar juga ya lama lama kuliah online jadi new normal sampai masa pandemi ini selesai juga bisa jadi online, kalau bisa daring kenapa harus kuliah offline ðÿ˜œ | Positif |

Tabel 4.15 Manualisasi Case Folding Data Uji

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | apa saya saja yang merasa kalau selama kuliah daring  nyaman banget sampai saya tidak ingin masuk kuliah karena  takut panik | ? |
| 2. | aku merasa lebih leluasa dengan kuliah daring, tidak capek harus siap-siap berangkat. hanya tinggal makan, beres didepan komputer sudah siap nyimak. buat materi, selama online emang tidak pernah mengandalkan dosen atau temen. jadi lebih banyak waktu buat searching sama buka textbook | ? |
| 3. | jujur tidak ada senang-senangnya kuliah daring. aku butuh praktik lapangan. apalagi semester depan magang. apa magang online juga? bisa stres gara-gara banyak deadline | ? |
| 4. | tatap langsung aja kadang tidak paham, apalagi kuliah daring, belum lagi jaringan lambat­ ditambah beberapa dosen yang jarang memberi kuliah online, atau cuma memberi tugas saja... fix kampus ku belum siap menerapkan kuliah daring! pic.twitter.com/uhdreylgh8 | ? |
| 5. | orang lain pada ribut sama keadaan kosan yang sudah ditinggal berbulan-bulan terus ribut gimana caranya balik ke kosan. aku anteng-anteng saja jadi penghuni kos dari awal pemerintah nyuruh dirumah saja dan kuliah jadi daring | ? |

#### Cleaning

Pada tahapan ini terjadi penghapusan simbol dan non karakter dalam dokumen. Tahapan ini dapat dilihat di Tabel 4.5 dan 4.6

Tabel 4.16 Manualisasi Cleaning Data Latih

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | aku selama kuliah online benar benar tidak belajar sama sekali ujian selalu tidak jujur tugas tinggal memindahkan dari internet dosen hanya memberi tugas tidak pernah ada penjelasan materi ditambah semester mau tetap daring mau jadi apa aku | Negatif |
| 2. | rasanya mau berhenti kuliah saja kalau daring begini seperti bayar cuma cuma materi dikasih secara online disuruh baca sendiri tanpa ada yang menjelaskan berasa otodidak | Negatif |
| 3. | maaf aku kuliah daring semakin malas kelas online saja ketiduran baik darimananya coba nilai sempurna bukan karena kita yang cerdas tapi karena dosennya yang kasihan sama kitanya tapi secara pemahaman kosong sekali otak ini terima kasih | Negatif |
| 4. | sejujurnya aku oke oke saja dengan kuliah daring cuma ya itu kangen sama suasana kelas kalau corona sudah selesai perpaduan offline online sepertinya asik | Netral |
| 5. | ada yang mempeributkan masalah kuliah online daring sebagian ada yang menyalahkan dosen ada juga yang menyalahkan diri sendiri mau kuliah online atau tidak semua tergantung pribadi masing masing dalam memahami materi yang dikasih dosen | Netral |
| 6. | pak ini gimana anak sekolahan offline untuk beberapa zona tapi kenapa mahasiswa tetap melaksanakan kuliah secara online daring justru mahasiswa lebih bisa beradaptasi dengan new normal dibandingkan dengan anak anak yang masih sangat rentan mohon dikaji lagi pak | Netral |
| 7. | saya berdoa kuliah tetap daring saja kampus mau offline padahal tempat masih zona merah dan kerabat aku yang kerjanya dokter saja suka bilang lagi kerja keras karena pasien corona lebih nyaman online tetap dirumah adalah jalanku | Positif |
| 8. | nilai positif saja yang diambil buang yang negatif positifnya mungkin ada beberapa mahasiswa yang tidak berani bertanya di kelas jadi lebih aktif bertanya di kuliah online daring | Positif |
| 9. | benar juga ya lama lama kuliah online jadi new normal sampai masa pandemi ini selesai juga bisa jadi online kalau bisa daring kenapa harus kuliah offline | Positif |

Tabel 4.17 Manualisasi Cleaning Data Uji

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | apa saya saja yang merasa kalau selama kuliah daring nyaman banget sampai saya tidak ingin masuk kuliah karena takut panik | ? |
| 2. | aku merasa lebih leluasa dengan kuliah daring tidak capek harus siap siap berangkat hanya tinggal makan beres didepan komputer sudah siap nyimak buat materi selama online emang tidak pernah mengandalkan dosen atau temen jadi lebih banyak waktu buat searching sama buka textbook | ? |
| 3. | jujur tidak ada senang senangnya kuliah daring aku butuh praktik lapangan apalagi semester depan magang apa magang online juga bisa stres gara gara banyak deadline | ? |
| 4. | tatap langsung aja kadang tidak paham apalagi kuliah daring belum lagi jaringan lambat ditambah beberapa dosen yang jarang memberi kuliah online atau cuma memberi tugas saja fix kampus ku belum siap menerapkan kuliah daring pic twitter com uhdreylgh | ? |
| 5. | orang lain pada ribut sama keadaan kosan yang sudah ditinggal berbulan bulan terus ribut gimana caranya balik ke kosan aku anteng anteng saja jadi penghuni kos dari awal pemerintah nyuruh dirumah saja dan kuliah jadi daring | ? |

#### Stemming

Pada tahapan ini terjadi perubahan kata menjadi kata dasar. Tahapan ini dibantu oleh *library* Sastrawi untuk proses stemming. Tahapan ini dapat dilihat di Tabel 4.7 dan 4.8

Tabel 4.18 Manualisasi Stemming Data Latih

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | aku lama kuliah online benar benar tidak ajar sama sekali uji selalu tidak jujur tugas tinggal pindah dari internet dosen hanya beri tugas tidak pernah ada jelas materi tambah semester mau tetap daring mau jadi apa aku | Negatif |
| 2. | rasa mau henti kuliah saja kalau daring begini seperti bayar cuma cuma materi kasih cara online suruh baca sendiri tanpa ada yang jelas asa otodidak | Negatif |
| 3. | maaf aku kuliah daring makin malas kelas online saja tidur baik darimananya coba nilai sempurna bukan karena kita yang cerdas tapi karena dosen yang kasihan sama kita tapi cara paham kosong sekali otak ini terima kasih | Negatif |
| 4. | jujur aku oke oke saja dengan kuliah daring cuma ya itu kangen sama suasana kelas kalau corona sudah selesai padu offline online seperti asik | Netral |
| 5. | ada yang ribut masalah kuliah online daring bagi ada yang salah dosen ada juga yang salah diri sendiri mau kuliah online atau tidak semua gantung pribadi masing masing dalam paham materi yang kasih dosen | Netral |
| 6. | pak ini gimana anak sekolahan offline untuk beberapa zona tapi kenapa mahasiswa tetap melaksanakan kuliah secara online daring justru mahasiswa lebih bisa beradaptasi dengan new normal dibandingkan dengan anak anak yang masih sangat rentan mohon dikaji lagi pak | Netral |
| 7. | saya doa kuliah tetap daring saja kampus mau offline padahal tempat masih zona merah dan kerabat aku yang kerja dokter saja suka bilang lagi kerja keras karena pasien corona lebih nyaman online tetap rumah adalah jalan | Positif |
| 8. | nilai positif saja yang ambil buang yang negatif positif mungkin ada beberapa mahasiswa yang tidak berani tanya di kelas jadi lebih aktif tanya di kuliah online daring | Positif |
| 9. | benar juga ya lama lama kuliah online jadi new normal sampai masa pandemi ini selesai juga bisa jadi online kalau bisa daring kenapa harus kuliah offline | Positif |

Tabel 4.19 Manualisasi Stemming Data Uji

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | apa saya saja yang rasa kalau lama kuliah daring nyaman banget sampai saya tidak ingin masuk kuliah karena takut panik | ? |
| 2. | aku rasa lebih leluasa dengan kuliah daring tidak capek harus siap siap berangkat hanya tinggal makan beres depan komputer sudah siap nyimak buat materi lama online emang tidak pernah andal dosen atau temen jadi lebih banyak waktu buat searching sama buka textbook | ? |
| 3. | jujur tidak ada senang senang kuliah daring aku butuh praktik lapang apalagi semester depan magang apa magang online juga bisa stres gara gara banyak deadline | ? |
| 4. | tatap langsung aja kadang tidak paham apalagi kuliah daring belum lagi jaring lambat tambah beberapa dosen yang jarang beri kuliah online atau cuma beri tugas saja fix kampus ku belum siap terap kuliah daring pic twitter com uhdreylgh | ? |
| 5. | orang lain pada ribut sama ada kosan yang sudah tinggal bulan bulan terus ribut gimana cara balik ke kosan aku anteng anteng saja jadi huni kos dari awal perintah nyuruh rumah saja dan kuliah jadi daring | ? |

#### Tokenisasi

Pada tahapan ini terjadi perubahan kalimat menjadi kesatuan kata yang terpisah. Tahapan ini dapat dilihat di Tabel 4.20 dan 4.21

Tabel 4.20 Manualisasi Tokenisasi Data Latih

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | ['aku', 'lama', 'kuliah', 'online', 'benar', 'benar', 'tidak', 'ajar', 'sama', 'sekali', 'uji', 'selalu', 'tidak', 'jujur', 'tugas', 'tinggal', 'pindah', 'dari', 'internet', 'dosen', 'hanya', 'beri', 'tugas', 'tidak', 'pernah', 'ada', 'jelas', 'materi', 'tambah', 'semester', 'mau', 'tetap', 'daring', 'mau', 'jadi', 'apa', 'aku'] | Negatif |
| 2. | ['rasa', 'mau', 'henti', 'kuliah', 'saja', 'kalau', 'daring', 'begini', 'seperti', 'bayar', 'cuma', 'cuma', 'materi', 'kasih', 'cara', 'online', 'suruh', 'baca', 'sendiri', 'tanpa', 'ada', 'yang', 'jelas', 'asa', 'otodidak'] | Negatif |
| 3. | ['maaf', 'aku', 'kuliah', 'daring', 'makin', 'malas', 'kelas', 'online', 'saja', 'tidur', 'baik', 'darimananya', 'coba', 'nilai', 'sempurna', 'bukan', 'karena', 'kita', 'yang', 'cerdas', 'tapi', 'karena', 'dosen', 'yang', 'kasihan', 'sama', 'kita', 'tapi', 'cara', 'paham', 'kosong', 'sekali', 'otak', 'ini', 'terima', 'kasih'] | Negatif |
| 4. | ['jujur', 'aku', 'oke', 'oke', 'saja', 'dengan', 'kuliah', 'daring', 'cuma', 'ya', 'itu', 'kangen', 'sama', 'suasana', 'kelas', 'kalau', 'corona', 'sudah', 'selesai', 'padu', 'offline', 'online', 'seperti', 'asik'] | Netral |
| 5. | ['ada', 'yang', 'ribut', 'masalah', 'kuliah', 'online', 'daring', 'bagi', 'ada', 'yang', 'salah', 'dosen', 'ada', 'juga', 'yang', 'salah', 'diri', 'sendiri', 'mau', 'kuliah', 'online', 'atau', 'tidak', 'semua', 'gantung', 'pribadi', 'masing', 'masing', 'dalam', 'paham', 'materi', 'yang', 'kasih', 'dosen'] | Netral |
| 6. | ['pak', 'ini', 'gimana', 'anak', 'sekolah', 'offline', 'untuk', 'beberapa', 'zona', 'tapi', 'kenapa', 'mahasiswa', 'tetap', 'laksana', 'kuliah', 'cara', 'online', 'daring', 'justru', 'mahasiswa', 'lebih', 'bisa', 'adaptasi', 'dengan', 'new', 'normal', 'banding', 'dengan', 'anak', 'anak', 'yang', 'masih', 'sangat', 'rentan', 'mohon', 'kaji', 'lagi', 'pak'] | Netral |
| 7. | ['saya', 'doa', 'kuliah', 'tetap', 'daring', 'saja', 'kampus', 'mau', 'offline', 'padahal', 'tempat', 'masih', 'zona', 'merah', 'dan', 'kerabat', 'aku', 'yang', 'kerja', 'dokter', 'saja', 'suka', 'bilang', 'lagi', 'kerja', 'keras', 'karena', 'pasien', 'corona', 'lebih', 'nyaman', 'online', 'tetap', 'rumah', 'adalah', 'jalan'] | Positif |
| 8. | ['nilai', 'positif', 'saja', 'yang', 'ambil', 'buang', 'yang', 'negatif', 'positif', 'mungkin', 'ada', 'beberapa', 'mahasiswa', 'yang', 'tidak', 'berani', 'tanya', 'di', 'kelas', 'jadi', 'lebih', 'aktif', 'tanya', 'di', 'kuliah', 'online', 'daring'] | Positif |
| 9. | ['benar', 'juga', 'ya', 'lama', 'lama', 'kuliah', 'online', 'jadi', 'new', 'normal', 'sampai', 'masa', 'pandemi', 'ini', 'selesai', 'juga', 'bisa', 'jadi', 'online', 'kalau', 'bisa', 'daring', 'kenapa', 'harus', 'kuliah', 'offline'] | Positif |

Tabel 4.21 Manualisasi Tokenisasi Data Uji

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | ['apa', 'saya', 'saja', 'yang', 'rasa', 'kalau', 'lama', 'kuliah', 'daring', 'nyaman', 'banget', 'sampai', 'saya', 'tidak', 'ingin', 'masuk', 'kuliah', 'karena', 'takut', 'panik'] | ? |
| 2. | ['aku', 'rasa', 'lebih', 'leluasa', 'dengan', 'kuliah', 'daring', 'tidak', 'capek', 'harus', 'siap', 'siap', 'berangkat', 'hanya', 'tinggal', 'makan', 'beres', 'depan', 'komputer', 'sudah', 'siap', 'nyimak', 'buat', 'materi', 'lama', 'online', 'emang', 'tidak', 'pernah', 'andal', 'dosen', 'atau', 'temen', 'jadi', 'lebih', 'banyak', 'waktu', 'buat', 'searching', 'sama', 'buka', 'textbook'] | ? |
| 3. | ['jujur', 'tidak', 'ada', 'senang', 'senang', 'kuliah', 'daring', 'aku', 'butuh', 'praktik', 'lapang', 'apalagi', 'semester', 'depan', 'magang', 'apa', 'magang', 'online', 'juga', 'bisa', 'stres', 'gara', 'gara', 'banyak', 'deadline'] | ? |
| 4. | ['tatap', 'langsung', 'aja', 'kadang', 'tidak', 'paham', 'apalagi', 'kuliah', 'daring', 'belum', 'lagi', 'jaring', 'lambat', 'tambah', 'beberapa', 'dosen', 'yang', 'jarang', 'beri', 'kuliah', 'online', 'atau', 'cuma', 'beri', 'tugas', 'saja', 'fix', 'kampus', 'ku', 'belum', 'siap', 'terap', 'kuliah', 'daring', 'pic', 'twitter', 'com', 'uhdreylgh'] | ? |
| 5. | ['orang', 'lain', 'pada', 'ribut', 'sama', 'ada', 'kosan', 'yang', 'sudah', 'tinggal', 'bulan', 'bulan', 'terus', 'ribut', 'gimana', 'cara', 'balik', 'ke', 'kosan', 'aku', 'anteng', 'anteng', 'saja', 'jadi', 'huni', 'kos', 'dari', 'awal', 'perintah', 'nyuruh', 'rumah', 'saja', 'dan', 'kuliah', 'jadi', 'daring'] | ? |

#### Filtering

Pada tahapan ini terjadi penghapusan kata yang terdapat dialam daftar stopword. Stopword yang digunakan pada tahapan ini adalah stopword yang dibuat dengan algoritma Term Based Random Sampling dengan Y senilai 50, X senilai 30 dan L senilai 20 persen. Berikut adalah hasil filtering yang ditampilkan pada Tabel 4.26 hingga Tabel 4.37.

Tabel 4.22 Manualisasi Filtering 20 Persen Data Latih

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | ['lama', 'benar', 'benar', 'ajar', 'sekali', 'uji', 'selalu', 'jujur', 'tugas', 'tinggal', 'pindah', 'dari', 'internet', 'dosen', 'hanya', 'beri', 'tugas', 'pernah', 'jelas', 'tambah', 'semester', 'apa'] | Negatif |
| 2. | ['rasa', 'henti', 'kalau', 'begini', 'seperti', 'bayar', 'cuma', 'cuma', 'suruh', 'baca', 'tanpa', 'jelas', 'asa', 'otodidak'] | Negatif |
| 3. | ['maaf', 'makin', 'malas', 'tidur', 'baik', 'darimananya', 'coba', 'nilai', 'sempurna', 'bukan', 'karena', 'kita', 'cerdas', 'tapi', 'karena', 'dosen', 'kasihan', 'kita', 'tapi', 'paham', 'kosong', 'sekali', 'otak', 'terima'] | Negatif |
| 4. | ['jujur', 'oke', 'oke', 'dengan', 'cuma', 'ya', 'itu', 'kangen', 'suasana', 'kalau', 'corona', 'sudah', 'selesai', 'padu', 'seperti', 'asik'] | Netral |
| 5. | ['ribut', 'masalah', 'bagi', 'salah', 'dosen', 'salah', 'diri', 'atau', 'semua', 'gantung', 'pribadi', 'masing', 'masing', 'dalam', 'paham', 'dosen'] | Netral |
| 6. | ['pak', 'gimana', 'anak', 'sekolah', 'untuk', 'tapi', 'mahasiswa', 'laksana', 'justru', 'mahasiswa', 'adaptasi', 'dengan', 'banding', 'dengan', 'anak', 'anak', 'masih', 'sangat', 'rentan', 'mohon', 'kaji', 'pak'] | Netral |
| 7. | ['saya', 'doa', 'kampus', 'padahal', 'tempat', 'masih', 'merah', 'dan', 'kerabat', 'kerja', 'dokter', 'suka', 'bilang', 'kerja', 'keras', 'karena', 'pasien', 'corona', 'nyaman', 'rumah', 'adalah', 'jalan'] | Positif |
| 8. | ['nilai', 'positif', 'ambil', 'buang', 'negatif', 'positif', 'mungkin', 'mahasiswa', 'berani', 'tanya', 'di', 'aktif', 'tanya', 'di'] | Positif |
| 9. | ['benar', 'ya', 'lama', 'lama', 'sampai', 'masa', 'pandemi', 'selesai', 'kalau', 'harus'] | Positif |

Tabel 4.23 Manualisasi Filtering 20 Persen Data Uji

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | ['apa', 'saya', 'rasa', 'kalau', 'lama', 'nyaman', 'banget', 'sampai', 'saya', 'ingin', 'masuk', 'karena', 'takut', 'panik'] | ? |
| 2. | ['rasa', 'leluasa', 'dengan', 'capek', 'harus', 'siap', 'siap', 'berangkat', 'hanya', 'tinggal', 'makan', 'beres', 'depan', 'komputer', 'sudah', 'siap', 'nyimak', 'buat', 'lama', 'emang', 'pernah', 'andal', 'dosen', 'atau', 'temen', 'banyak', 'waktu', 'buat', 'searching', 'buka', 'textbook'] | ? |
| 3. | ['jujur', 'senang', 'senang', 'butuh', 'praktik', 'lapang', 'apalagi', 'semester', 'depan', 'magang', 'apa', 'magang', 'stres', 'gara', 'gara', 'banyak', 'deadline'] | ? |
| 4. | ['tatap', 'langsung', 'aja', 'kadang', 'paham', 'apalagi', 'belum', 'jaring', 'lambat', 'tambah', 'dosen', 'jarang', 'beri', 'atau', 'cuma', 'beri', 'tugas', 'fix', 'kampus', 'ku', 'belum', 'siap', 'terap', 'pic', 'twitter', 'com', 'uhdreylgh'] | ? |
| 5. | ['orang', 'lain', 'pada', 'ribut', 'kosan', 'sudah', 'tinggal', 'bulan', 'bulan', 'terus', 'ribut', 'gimana', 'balik', 'ke', 'kosan', 'anteng', 'anteng', 'huni', 'kos', 'dari', 'awal', 'perintah', 'nyuruh', 'rumah', 'dan'] | ? |

Sehingga didapatkan term data latih sebagai berikut.

|  |
| --- |
| **Term** |
| ['lama', 'benar', 'ajar', 'sekali', 'uji', 'selalu', 'jujur', 'tugas', 'tinggal', 'pindah', 'dari', 'internet', 'dosen', 'hanya', 'beri', 'pernah', 'jelas', 'tambah', 'semester', 'apa', 'rasa', 'henti', 'kalau', 'begini', 'seperti', 'bayar', 'cuma', 'suruh', 'baca', 'tanpa', 'asa', 'otodidak', 'maaf', 'makin', 'malas', 'tidur', 'baik', 'darimananya', 'coba', 'nilai', 'sempurna', 'bukan', 'karena', 'kita', 'cerdas', 'tapi', 'kasihan', 'paham', 'kosong', 'otak', 'terima', 'oke', 'dengan', 'ya', 'itu', 'kangen', 'suasana', 'corona', 'sudah', 'selesai', 'padu', 'asik', 'ribut', 'masalah', 'bagi', 'salah', 'diri', 'atau', 'semua', 'gantung', 'pribadi', 'masing', 'dalam', 'pak', 'gimana', 'anak', 'sekolah', 'untuk', 'mahasiswa', 'laksana', 'justru', 'adaptasi', 'banding', 'masih', 'sangat', 'rentan', 'mohon', 'kaji', 'saya', 'doa', 'kampus', 'padahal', 'tempat', 'merah', 'dan', 'kerabat', 'kerja', 'dokter', 'suka', 'bilang', 'keras', 'pasien', 'nyaman', 'rumah', 'adalah', 'jalan', 'positif', 'ambil', 'buang', 'negatif', 'mungkin', 'berani', 'tanya', 'di', 'aktif', 'sampai', 'masa', 'pandemi', 'harus'] |

### Term Weighting

Tahapan term weighting ini adalah tahapan untuk memberi bobot setiap term sesuai dengan ciri dari masing-masing term tersebut. Adapun tahap-tahap term weighting yang digunakan adalah *raw term frequency, log term frequency, inverse document frequency,* dan *term frequency-inverse document (tf-idf).*

#### Raw Term Frequeny Weighting

Pada *raw term frequency weignting,* setiap term diberi bobot berdasarkan frekuensi kemunculan tersebut dalam suatu dokumen. Berikut adalah hasil *raw term frequency* yang dapat dilihat dalam Tabel 4.38.

Tabel 4.24 Manualisasi Raw Term Frequency Weighting

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Term** | **D1** | **D2** | **D3** | **D4** | **D5** | **D6** | **D7** | **D8** | **D9** |
| lama | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| benar | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ajar | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| sekali | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| uji | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| selalu | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| jujur | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| tugas | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| tinggal | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pindah | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dari | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| internet | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dosen | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 |
| hanya | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| beri | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pernah | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| jelas | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| tambah | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| semester | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| apa | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| rasa | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| henti | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| kalau | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| begini | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| seperti | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| bayar | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| cuma | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| suruh | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| baca | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| tanpa | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| asa | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| otodidak | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| maaf | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| makin | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| malas | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| tidur | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| baik | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| darimananya | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| coba | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| nilai | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| sempurna | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| bukan | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| karena | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| kita | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| cerdas | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| tapi | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| kasihan | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| paham | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| kosong | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| otak | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| terima | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| oke | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| dengan | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 |
| ya | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| itu | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| kangen | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| suasana | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| corona | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| sudah | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| selesai | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| padu | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| asik | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| ribut | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| masalah | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| bagi | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| salah | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| diri | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| atau | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| semua | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| gantung | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| pribadi | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| masing | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| dalam | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| pak | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| gimana | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| anak | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| sekolah | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| untuk | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| mahasiswa | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 0 |
| laksana | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| justru | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| adaptasi | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| banding | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| masih | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| sangat | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| rentan | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| mohon | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| kaji | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| saya | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| doa | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| kampus | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| padahal | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| tempat | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| merah | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| dan | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| kerabat | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| kerja | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| dokter | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| suka | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| bilang | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| keras | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| pasien | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| nyaman | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| rumah | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| adalah | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| jalan | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| positif | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| ambil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| buang | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| negatif | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| mungkin | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| berani | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| tanya | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| di | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| aktif | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| sampai | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| masa | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| pandemi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| harus | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

#### Log Term Frequency Weighting

Pada *log term frequency weighting,* setiap term yang sudah dilakukan proses *raw term frequency weighting* akan dihitung logaritmanya. Berikut adalah contoh perhitungan dari *log term frequency weighting.*

Misalkan kita menggunakan kata “benar” dan berikut adalah hasil yang didapatkan setelah proses *raw term frequency weighting.*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Term** | **D1** | **D2** | **D3** | **D4** | **D5** | **D6** | **D7** | **D8** | **D9** |
| benar | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Berikut adalah rumus yang digunakan dalam perhitungan *log term frequency weighting.*

Sehingga berikut adalah contoh perhitungan dari term “aku” di D1, D2, dan D3.

Setelah melalui proses perhitungan tersebut, berikut adalah hasil *log term frequency* yang dapat dilihat dalam Tabel 4.39.

Tabel 4.25 Manualisasi Log Term Frequency Weighting

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Term** | **D1** | **D2** | **D3** | **D4** | **D5** | **D6** | **D7** | **D8** | **D9** |
| lama | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.301 |
| benar | 1.301 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 |
| ajar | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| sekali | 1.0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| uji | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| selalu | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| jujur | 1.0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| tugas | 1.301 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| tinggal | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pindah | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dari | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| internet | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dosen | 1.0 | 0 | 1.0 | 0 | 1.301 | 0 | 0 | 0 | 0 |
| hanya | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| beri | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| pernah | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| jelas | 1.0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| tambah | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| semester | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| apa | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| rasa | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| henti | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| kalau | 0 | 1.0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 1.0 |
| begini | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| seperti | 0 | 1.0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| bayar | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| cuma | 0 | 1.301 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| suruh | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| baca | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| tanpa | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| asa | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| otodidak | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| maaf | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| makin | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| malas | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| tidur | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| baik | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| darimananya | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| coba | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nilai | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 1.0 | 0 |
| sempurna | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| bukan | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| karena | 0 | 0 | 1.301 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| kita | 0 | 0 | 1.301 | 0 | 0 | 0 | 0 | 0 | 0 |
| cerdas | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| tapi | 0 | 0 | 1.301 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| kasihan | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| paham | 0 | 0 | 1.0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| kosong | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| otak | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| terima | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| oke | 0 | 0 | 0 | 1.301 | 0 | 0 | 0 | 0 | 0 |
| dengan | 0 | 0 | 0 | 1.0 | 0 | 1.301 | 0 | 0 | 0 |
| ya | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 1.0 |
| itu | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| kangen | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| suasana | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| corona | 0 | 0 | 0 | 1.0 | 0 | 0 | 1.0 | 0 | 0 |
| sudah | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| selesai | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 1.0 |
| padu | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| asik | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| ribut | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| masalah | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| bagi | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| salah | 0 | 0 | 0 | 0 | 1.301 | 0 | 0 | 0 | 0 |
| diri | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| atau | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| semua | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| gantung | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| pribadi | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| masing | 0 | 0 | 0 | 0 | 1.301 | 0 | 0 | 0 | 0 |
| dalam | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| pak | 0 | 0 | 0 | 0 | 0 | 1.301 | 0 | 0 | 0 |
| gimana | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| anak | 0 | 0 | 0 | 0 | 0 | 1.477 | 0 | 0 | 0 |
| sekolah | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| untuk | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| mahasiswa | 0 | 0 | 0 | 0 | 0 | 1.301 | 0 | 1.0 | 0 |
| laksana | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| justru | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| adaptasi | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| banding | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| masih | 0 | 0 | 0 | 0 | 0 | 1.0 | 1.0 | 0 | 0 |
| sangat | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| rentan | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| mohon | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| kaji | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 | 0 |
| saya | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| doa | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| kampus | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| padahal | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| tempat | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| merah | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| dan | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| kerabat | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| kerja | 0 | 0 | 0 | 0 | 0 | 0 | 1.301 | 0 | 0 |
| dokter | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| suka | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| bilang | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| keras | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| pasien | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| nyaman | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| rumah | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| adalah | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| jalan | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 | 0 |
| positif | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.301 | 0 |
| ambil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 |
| buang | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 |
| negatif | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 |
| mungkin | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 |
| berani | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 |
| tanya | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.301 | 0 |
| di | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.301 | 0 |
| aktif | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 | 0 |
| sampai | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 |
| masa | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 |
| pandemi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 |
| harus | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.0 |

#### Inverse Document Frequency

Pada *inverse document frequency*, akan dilakukan proses perhitungan jumlah dokumen yang mengandung suatu term dan jumlah tersebut akan dilakukan proses *inverse.* Tahapan awal yang harus dilakukan adalah menghitung *document frequency* atau frekuensi dokumen yang mengandung suatu term. Berikut adalah hasil perhitungan *document frequency* yang ditampilkan pada Tabel 4.40.

Tabel 4.26 Manualisasi Document Frequency

|  |  |
| --- | --- |
| **Term** | **DF** |
| lama | 2 |
| benar | 2 |
| ajar | 1 |
| sekali | 2 |
| uji | 1 |
| selalu | 1 |
| jujur | 2 |
| tugas | 1 |
| tinggal | 1 |
| pindah | 1 |
| dari | 1 |
| internet | 1 |
| dosen | 3 |
| hanya | 1 |
| beri | 1 |
| pernah | 1 |
| jelas | 2 |
| tambah | 1 |
| semester | 1 |
| apa | 1 |
| rasa | 1 |
| henti | 1 |
| kalau | 3 |
| begini | 1 |
| seperti | 2 |
| bayar | 1 |
| cuma | 2 |
| suruh | 1 |
| baca | 1 |
| tanpa | 1 |
| asa | 1 |
| otodidak | 1 |
| maaf | 1 |
| makin | 1 |
| malas | 1 |
| tidur | 1 |
| baik | 1 |
| darimananya | 1 |
| coba | 1 |
| nilai | 2 |
| sempurna | 1 |
| bukan | 1 |
| karena | 2 |
| kita | 1 |
| cerdas | 1 |
| tapi | 2 |
| kasihan | 1 |
| paham | 2 |
| kosong | 1 |
| otak | 1 |
| terima | 1 |
| oke | 1 |
| dengan | 2 |
| ya | 2 |
| itu | 1 |
| kangen | 1 |
| suasana | 1 |
| corona | 2 |
| sudah | 1 |
| selesai | 2 |
| padu | 1 |
| asik | 1 |
| ribut | 1 |
| masalah | 1 |
| bagi | 1 |
| salah | 1 |
| diri | 1 |
| atau | 1 |
| semua | 1 |
| gantung | 1 |
| pribadi | 1 |
| masing | 1 |
| dalam | 1 |
| pak | 1 |
| gimana | 1 |
| anak | 1 |
| sekolah | 1 |
| untuk | 1 |
| mahasiswa | 2 |
| laksana | 1 |
| justru | 1 |
| adaptasi | 1 |
| banding | 1 |
| masih | 2 |
| sangat | 1 |
| rentan | 1 |
| mohon | 1 |
| kaji | 1 |
| saya | 1 |
| doa | 1 |
| kampus | 1 |
| padahal | 1 |
| tempat | 1 |
| merah | 1 |
| dan | 1 |
| kerabat | 1 |
| kerja | 1 |
| dokter | 1 |
| suka | 1 |
| bilang | 1 |
| keras | 1 |
| pasien | 1 |
| nyaman | 1 |
| rumah | 1 |
| adalah | 1 |
| jalan | 1 |
| positif | 1 |
| ambil | 1 |
| buang | 1 |
| negatif | 1 |
| mungkin | 1 |
| berani | 1 |
| tanya | 1 |
| di | 1 |
| aktif | 1 |
| sampai | 1 |
| masa | 1 |
| pandemi | 1 |
| harus | 1 |

Setelah *document frequency* tiap termnya didapatkan, langkah selanjutnya adalah menghitung *inverse document frequency* dengan persamaan sebagai berikut.

Sehingga berikut adalah contoh perhitungan dari term “lama”, “benar”, “ajar”.

Setelah melalui proses perhitungan tersebut, berikut adalah proses perhitungan *inverse document frequency* keseluruhan term yang dapat dilihat dalam Tabel 4.41.

Tabel 4.27 Manualisasi Inverse Document Frequency

|  |  |
| --- | --- |
| **Term** | **IDF** |
| lama | 0.653 |
| benar | 0.653 |
| ajar | 0.954 |
| sekali | 0.653 |
| uji | 0.954 |
| selalu | 0.954 |
| jujur | 0.653 |
| tugas | 0.954 |
| tinggal | 0.954 |
| pindah | 0.954 |
| dari | 0.954 |
| internet | 0.954 |
| dosen | 0.477 |
| hanya | 0.954 |
| beri | 0.954 |
| pernah | 0.954 |
| jelas | 0.653 |
| tambah | 0.954 |
| semester | 0.954 |
| apa | 0.954 |
| rasa | 0.954 |
| henti | 0.954 |
| kalau | 0.477 |
| begini | 0.954 |
| seperti | 0.653 |
| bayar | 0.954 |
| cuma | 0.653 |
| suruh | 0.954 |
| baca | 0.954 |
| tanpa | 0.954 |
| asa | 0.954 |
| otodidak | 0.954 |
| maaf | 0.954 |
| makin | 0.954 |
| malas | 0.954 |
| tidur | 0.954 |
| baik | 0.954 |
| darimananya | 0.954 |
| coba | 0.954 |
| nilai | 0.653 |
| sempurna | 0.954 |
| bukan | 0.954 |
| karena | 0.653 |
| kita | 0.954 |
| cerdas | 0.954 |
| tapi | 0.653 |
| kasihan | 0.954 |
| paham | 0.653 |
| kosong | 0.954 |
| otak | 0.954 |
| terima | 0.954 |
| oke | 0.954 |
| dengan | 0.653 |
| ya | 0.653 |
| itu | 0.954 |
| kangen | 0.954 |
| suasana | 0.954 |
| corona | 0.653 |
| sudah | 0.954 |
| selesai | 0.653 |
| padu | 0.954 |
| asik | 0.954 |
| ribut | 0.954 |
| masalah | 0.954 |
| bagi | 0.954 |
| salah | 0.954 |
| diri | 0.954 |
| atau | 0.954 |
| semua | 0.954 |
| gantung | 0.954 |
| pribadi | 0.954 |
| masing | 0.954 |
| dalam | 0.954 |
| pak | 0.954 |
| gimana | 0.954 |
| anak | 0.954 |
| sekolah | 0.954 |
| untuk | 0.954 |
| mahasiswa | 0.653 |
| laksana | 0.954 |
| justru | 0.954 |
| adaptasi | 0.954 |
| banding | 0.954 |
| masih | 0.653 |
| sangat | 0.954 |
| rentan | 0.954 |
| mohon | 0.954 |
| kaji | 0.954 |
| saya | 0.954 |
| doa | 0.954 |
| kampus | 0.954 |
| padahal | 0.954 |
| tempat | 0.954 |
| merah | 0.954 |
| dan | 0.954 |
| kerabat | 0.954 |
| kerja | 0.954 |
| dokter | 0.954 |
| suka | 0.954 |
| bilang | 0.954 |
| keras | 0.954 |
| pasien | 0.954 |
| nyaman | 0.954 |
| rumah | 0.954 |
| adalah | 0.954 |
| jalan | 0.954 |
| positif | 0.954 |
| ambil | 0.954 |
| buang | 0.954 |
| negatif | 0.954 |
| mungkin | 0.954 |
| berani | 0.954 |
| tanya | 0.954 |
| di | 0.954 |
| aktif | 0.954 |
| sampai | 0.954 |
| masa | 0.954 |
| pandemi | 0.954 |
| harus | 0.954 |

#### Term Frequency - Inverse Document Frequency (TF-IDF)

Pada term frequency - *inverse document frequency*, akan dilakukan proses perhitungan perkalian antara *log term frequency* dikali dengan *inverse document frequency.* Berikut adalah hasil perhitungan perkalian antara *log term frequency* dikali dengan *inverse document frequency* yang ditampilkan pada Tabel 4.42*.*

Tabel 4.28 Manualisasi Term Frequency - Inverse Document Frequency

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Term** | **D1** | **D2** | **D3** | **D4** | **D5** | **D6** | **D7** | **D8** | **D9** |
| lama | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.850 |
| benar | 0.850 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.653 |
| ajar | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| sekali | 0.653 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| uji | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| selalu | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| jujur | 0.653 | 0.0 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| tugas | 1.241 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| tinggal | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| pindah | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| dari | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| internet | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| dosen | 0.477 | 0.0 | 0.477 | 0.0 | 0.621 | 0.0 | 0.0 | 0.0 | 0.0 |
| hanya | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| beri | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| pernah | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| jelas | 0.653 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| tambah | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| semester | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| apa | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| rasa | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| henti | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| kalau | 0.0 | 0.477 | 0.0 | 0.477 | 0.0 | 0.0 | 0.0 | 0.0 | 0.477 |
| begini | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| seperti | 0.0 | 0.653 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| bayar | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| cuma | 0.0 | 0.850 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| suruh | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| baca | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| tanpa | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| asa | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| otodidak | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| maaf | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| makin | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| malas | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| tidur | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| baik | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| darimananya | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| coba | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| nilai | 0.0 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.653 | 0.0 |
| sempurna | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| bukan | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| karena | 0.0 | 0.0 | 0.850 | 0.0 | 0.0 | 0.0 | 0.653 | 0.0 | 0.0 |
| kita | 0.0 | 0.0 | 1.241 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| cerdas | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| tapi | 0.0 | 0.0 | 0.850 | 0.0 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 |
| kasihan | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| paham | 0.0 | 0.0 | 0.653 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 |
| kosong | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| otak | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| terima | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| oke | 0.0 | 0.0 | 0.0 | 1.241 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| dengan | 0.0 | 0.0 | 0.0 | 0.653 | 0.0 | 0.850 | 0.0 | 0.0 | 0.0 |
| ya | 0.0 | 0.0 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.653 |
| itu | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| kangen | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| suasana | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| corona | 0.0 | 0.0 | 0.0 | 0.653 | 0.0 | 0.0 | 0.653 | 0.0 | 0.0 |
| sudah | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| selesai | 0.0 | 0.0 | 0.0 | 0.653 | 0.0 | 0.0 | 0.0 | 0.0 | 0.653 |
| padu | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| asik | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ribut | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| masalah | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| bagi | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| salah | 0.0 | 0.0 | 0.0 | 0.0 | 1.241 | 0.0 | 0.0 | 0.0 | 0.0 |
| diri | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| atau | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| semua | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| gantung | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| pribadi | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| masing | 0.0 | 0.0 | 0.0 | 0.0 | 1.241 | 0.0 | 0.0 | 0.0 | 0.0 |
| dalam | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 | 0.0 |
| pak | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.241 | 0.0 | 0.0 | 0.0 |
| gimana | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| anak | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.410 | 0.0 | 0.0 | 0.0 |
| sekolah | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| untuk | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| mahasiswa | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.850 | 0.0 | 0.653 | 0.0 |
| laksana | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| justru | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| adaptasi | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| banding | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| masih | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.653 | 0.653 | 0.0 | 0.0 |
| sangat | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| rentan | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| mohon | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| kaji | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 | 0.0 |
| saya | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| doa | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| kampus | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| padahal | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| tempat | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| merah | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| dan | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| kerabat | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| kerja | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.241 | 0.0 | 0.0 |
| dokter | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| suka | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| bilang | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| keras | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| pasien | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| nyaman | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| rumah | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| adalah | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| jalan | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 | 0.0 |
| positif | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.241 | 0.0 |
| ambil | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 |
| buang | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 |
| negatif | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 |
| mungkin | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 |
| berani | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 |
| tanya | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.241 | 0.0 |
| di | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.241 | 0.0 |
| aktif | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 | 0.0 |
| sampai | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 |
| masa | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 |
| pandemi | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 |
| harus | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.954 |

### Manualisasi Naïve Bayes Training

Pada tahapan ini terjadi pelatihan Naïve Bayes Multinomial untuk mendapatkan nilai *prior* tiap kelasnya dan *likelihood* tiap kata pada tiap kelasnya. Pada tahap manualisasi pelatihan Naïve Bayes ini, peneliti menggunakan contoh hasil pembobotan dengan stoplist 20 persen.

Tahapan ini diawali dengan pencarian prior untuk tiap kelasnya. Perhitungan prior dapat menggunakan persamaan sebagai berikut.

Untuk mendapatkan prior dari kelas tertentu, jumlah dokumen dikelas tertentu harus dibagi dengan jumlah keseluruhan dokumen. Sehingga dapat ditentukan prior dari tiap kelasnya adalah sebagai berikut.

Berikut adalah persamaan yang digunakan untuk menghitung likelihood dalam Naïve Bayes Multinomial jika pembobotan menggunakan TF-IDF.

Dalam persamaan tersebut kita memasukan bobot term, jumlah bobot tiap kelas, serta jumlah idf yang sudah didapatkan diperhitungan sebelumnya. Berikut adalah contoh perhitungan dari term “berani”.

Setelah dilakukan dengan semua term dalam keseluruhan dokumen setiap kelasnya maka didapatkan likelihood yang akan ditampilkan pada Tabel 4.43.

Tabel 4.29 Manualisasi *Likelihood*

|  |  |  |  |
| --- | --- | --- | --- |
| **Term** | **Negatif** | **Netral** | **Positif** |
| lama | 0.010664 | 0.006757 | 0.012778 |
| benar | 0.011932 | 0.006757 | 0.011420 |
| ajar | 0.012606 | 0.006757 | 0.006908 |
| sekali | 0.014877 | 0.006757 | 0.006908 |
| uji | 0.012606 | 0.006757 | 0.006908 |
| selalu | 0.012606 | 0.006757 | 0.006908 |
| jujur | 0.010664 | 0.011170 | 0.006908 |
| tugas | 0.014459 | 0.006757 | 0.006908 |
| tinggal | 0.012606 | 0.006757 | 0.006908 |
| pindah | 0.012606 | 0.006757 | 0.006908 |
| dari | 0.012606 | 0.006757 | 0.006908 |
| internet | 0.012606 | 0.006757 | 0.006908 |
| dosen | 0.012606 | 0.010951 | 0.006908 |
| hanya | 0.012606 | 0.006757 | 0.006908 |
| beri | 0.012606 | 0.006757 | 0.006908 |
| pernah | 0.012606 | 0.006757 | 0.006908 |
| jelas | 0.014877 | 0.006757 | 0.006908 |
| tambah | 0.012606 | 0.006757 | 0.006908 |
| semester | 0.012606 | 0.006757 | 0.006908 |
| apa | 0.012606 | 0.006757 | 0.006908 |
| rasa | 0.012606 | 0.006757 | 0.006908 |
| henti | 0.012606 | 0.006757 | 0.006908 |
| kalau | 0.009528 | 0.009981 | 0.010203 |
| begini | 0.012606 | 0.006757 | 0.006908 |
| seperti | 0.010664 | 0.011170 | 0.006908 |
| bayar | 0.012606 | 0.006757 | 0.006908 |
| cuma | 0.011932 | 0.011170 | 0.006908 |
| suruh | 0.012606 | 0.006757 | 0.006908 |
| baca | 0.012606 | 0.006757 | 0.006908 |
| tanpa | 0.012606 | 0.006757 | 0.006908 |
| asa | 0.012606 | 0.006757 | 0.006908 |
| otodidak | 0.012606 | 0.006757 | 0.006908 |
| maaf | 0.012606 | 0.006757 | 0.006908 |
| makin | 0.012606 | 0.006757 | 0.006908 |
| malas | 0.012606 | 0.006757 | 0.006908 |
| tidur | 0.012606 | 0.006757 | 0.006908 |
| baik | 0.012606 | 0.006757 | 0.006908 |
| darimananya | 0.012606 | 0.006757 | 0.006908 |
| coba | 0.012606 | 0.006757 | 0.006908 |
| nilai | 0.010664 | 0.006757 | 0.011420 |
| sempurna | 0.012606 | 0.006757 | 0.006908 |
| bukan | 0.012606 | 0.006757 | 0.006908 |
| karena | 0.011932 | 0.006757 | 0.011420 |
| kita | 0.014459 | 0.006757 | 0.006908 |
| cerdas | 0.012606 | 0.006757 | 0.006908 |
| tapi | 0.011932 | 0.011170 | 0.006908 |
| kasihan | 0.012606 | 0.006757 | 0.006908 |
| paham | 0.010664 | 0.011170 | 0.006908 |
| kosong | 0.012606 | 0.006757 | 0.006908 |
| otak | 0.012606 | 0.006757 | 0.006908 |
| terima | 0.012606 | 0.006757 | 0.006908 |
| oke | 0.006450 | 0.015145 | 0.006908 |
| dengan | 0.006450 | 0.016913 | 0.006908 |
| ya | 0.006450 | 0.011170 | 0.011420 |
| itu | 0.006450 | 0.013204 | 0.006908 |
| kangen | 0.006450 | 0.013204 | 0.006908 |
| suasana | 0.006450 | 0.013204 | 0.006908 |
| corona | 0.006450 | 0.011170 | 0.011420 |
| sudah | 0.006450 | 0.013204 | 0.006908 |
| selesai | 0.006450 | 0.011170 | 0.011420 |
| padu | 0.006450 | 0.013204 | 0.006908 |
| asik | 0.006450 | 0.013204 | 0.006908 |
| ribut | 0.006450 | 0.013204 | 0.006908 |
| masalah | 0.006450 | 0.013204 | 0.006908 |
| bagi | 0.006450 | 0.013204 | 0.006908 |
| salah | 0.006450 | 0.015145 | 0.006908 |
| diri | 0.006450 | 0.013204 | 0.006908 |
| atau | 0.006450 | 0.013204 | 0.006908 |
| semua | 0.006450 | 0.013204 | 0.006908 |
| gantung | 0.006450 | 0.013204 | 0.006908 |
| pribadi | 0.006450 | 0.013204 | 0.006908 |
| masing | 0.006450 | 0.015145 | 0.006908 |
| dalam | 0.006450 | 0.013204 | 0.006908 |
| pak | 0.006450 | 0.015145 | 0.006908 |
| gimana | 0.006450 | 0.013204 | 0.006908 |
| anak | 0.006450 | 0.016281 | 0.006908 |
| sekolah | 0.006450 | 0.013204 | 0.006908 |
| untuk | 0.006450 | 0.013204 | 0.006908 |
| mahasiswa | 0.006450 | 0.012499 | 0.011420 |
| laksana | 0.006450 | 0.013204 | 0.006908 |
| justru | 0.006450 | 0.013204 | 0.006908 |
| adaptasi | 0.006450 | 0.013204 | 0.006908 |
| banding | 0.006450 | 0.013204 | 0.006908 |
| masih | 0.006450 | 0.011170 | 0.011420 |
| sangat | 0.006450 | 0.013204 | 0.006908 |
| rentan | 0.006450 | 0.013204 | 0.006908 |
| mohon | 0.006450 | 0.013204 | 0.006908 |
| kaji | 0.006450 | 0.013204 | 0.006908 |
| saya | 0.006450 | 0.006757 | 0.013499 |
| doa | 0.006450 | 0.006757 | 0.013499 |
| kampus | 0.006450 | 0.006757 | 0.013499 |
| padahal | 0.006450 | 0.006757 | 0.013499 |
| tempat | 0.006450 | 0.006757 | 0.013499 |
| merah | 0.006450 | 0.006757 | 0.013499 |
| dan | 0.006450 | 0.006757 | 0.013499 |
| kerabat | 0.006450 | 0.006757 | 0.013499 |
| kerja | 0.006450 | 0.006757 | 0.015484 |
| dokter | 0.006450 | 0.006757 | 0.013499 |
| suka | 0.006450 | 0.006757 | 0.013499 |
| bilang | 0.006450 | 0.006757 | 0.013499 |
| keras | 0.006450 | 0.006757 | 0.013499 |
| pasien | 0.006450 | 0.006757 | 0.013499 |
| nyaman | 0.006450 | 0.006757 | 0.013499 |
| rumah | 0.006450 | 0.006757 | 0.013499 |
| adalah | 0.006450 | 0.006757 | 0.013499 |
| jalan | 0.006450 | 0.006757 | 0.013499 |
| positif | 0.006450 | 0.006757 | 0.015484 |
| ambil | 0.006450 | 0.006757 | 0.013499 |
| buang | 0.006450 | 0.006757 | 0.013499 |
| negatif | 0.006450 | 0.006757 | 0.013499 |
| mungkin | 0.006450 | 0.006757 | 0.013499 |
| berani | 0.006450 | 0.006757 | 0.013499 |
| tanya | 0.006450 | 0.006757 | 0.015484 |
| di | 0.006450 | 0.006757 | 0.015484 |
| aktif | 0.006450 | 0.006757 | 0.013499 |
| sampai | 0.006450 | 0.006757 | 0.013499 |
| masa | 0.006450 | 0.006757 | 0.013499 |
| pandemi | 0.006450 | 0.006757 | 0.013499 |
| harus | 0.006450 | 0.006757 | 0.013499 |

### Manualisasi *Naïve* *Bayes* *Testing*

Setelah *prior* dan *likelihood* didapatkan dalam proses *training.* Selanjutnya adalah tahapan testing yang dimana didalamnya terjadi perkalian nilai *prior* tiap kelasnya dan *likelihood* tiap kata pada tiap kelasnya. Pada tahap manualisasi pengujian *Naïve* *Bayes* ini, peneliti menggunakan contoh hasil pembobotan dengan stoplist 20 persen.

Dalam tahapan ini, akan dicari probabilitas tertinggi masing-masing kelas untuk proses klasifikasi. Perhitungan probabilitas Algoritma *Naïve Bayes* adalah sebagai berikut.

Setelah dilakukan *preprocessing* pada pembahasan sebelumnya, berikut adalah hasil akhir dari tahap *preprocessing* dari data uji menggunakan *stopword* 20 persen yang ditampilkan pada Tabel 4.30.

Tabel 4.30 Hasil *Preprocessing* Data Uji

|  |  |  |
| --- | --- | --- |
| No | Tweet | Kelas |
| 1. | ['apa', 'rasa', 'lama', 'nyaman', 'banget', 'ingin', 'masuk', 'takut', 'panik'] | ? |

Sehingga dapat dihitung posterior dari tiap kelasnya sebagai berikut.

Sehingga didapatkan bahwa probabilitas data uji diklasifikasikan kelas negatif adalah .

Sehingga didapatkan bahwa probabilitas data uji diklasifikasikan kelas netral adalah .

Sehingga didapatkan bahwa probabilitas data uji diklasifikasikan kelas positif adalah .

|  |  |
| --- | --- |
| **Klasifikasi** | **Posterior** |
| Negatif | 1.1117643466553663e-19 |
| Netral | 1.4453587316155e-20 |
| Positif | 7.864406782717774e-19 |

Kelas Positif memiliki nilai posterior tertinggi dibanding dengan posterior kelas lainnya. Oleh karena itu data uji dapat diklasifikasikan sebagai kelas Positif.

Data uji lainnya mengikuti proses yang sama seperti perhitungan sebelumnya, sehingga didapatkan hasil klasifikasi serta posteriornya yang ditampilkan pada Tabel 4.31, Tabel 4.32, Tabel 4.33, Tabel 4.34, dan Tabel 4.35.

Tabel 4.31 Hasil Manualisasi Data Uji 1

|  |  |  |
| --- | --- | --- |
| **Data Uji 1** | | |
| Apa saya saja yang merasa kalau selama kuliah daring  nyaman banget sampai saya tidak ingin masuk kuliah karena  takut panik | | |
| **Negatif** | **Netral** | **Positif** |
| 1.1117643466553663e-19 | 1.4453587316155e-20 | 7.864406782717774e-19 |
| **Actual: Positif** | | |
| **Prediction: Positif** | | |

Tabel 4.32 Hasil Manualisasi Data Uji 2

|  |  |  |
| --- | --- | --- |
| **Data Uji 2** | | |
| Aku merasa lebih leluasa dengan kuliah daring, tidak capek harus siap-siap berangkat. Hanya tinggal makan, beres didepan komputer sudah siap nyimak. Buat materi, selama online emang tidak pernah mengandalkan dosen atau teman. Jadi lebih banyak waktu buat searching sama buka text book. | | |
| **Negatif** | **Netral** | **Positif** |
| 1.9587729876808433e-21 | 1.0243444918164579e-21 | 2.9806998411846044e-22 |
| **Actual: Positif** | | |
| **Prediction: Negatif** | | |

Tabel 4.33 Hasil Manualisasi Data Uji 3

|  |  |  |
| --- | --- | --- |
| **Data Uji 3** | | |
| Jujur tidak ada senang-senangnya kuliah daring. Aku butuh praktik lapangan. Apalagi semester depan magang. Apa magang online juga? Bisa stres gara-gara banyak deadline | | |
| **Negatif** | **Netral** | **Positif** |
| 5.64845483313626e-07 | 1.699942956949124e-07 | 1.0986925237206374e-07 |
| **Actual: Negatif** | | |
| **Prediction: Negatif** | | |

Tabel 4.34 Hasil Manualisasi Data Uji 4

|  |  |  |
| --- | --- | --- |
| **Data Uji 4** | | |
| Tatap langsung aja kadang tidak paham, apalagi kuliah daring, belum lagi jaringan lambat­ ditambah beberapa dosen yang jarang memberi kuliah online, atau cuma memberi tugas saja... Fix kampus ku belum siap menerapkan kuliah daring! pic.twitter.com/UHdReyLgh8 | | |
| **Negatif** | **Netral** | **Positif** |
| 6.443055725811558e-19 | 8.470540171181903e-20 | 2.332649424457759e-20 |
| **Actual: Negatif** | | |
| **Prediction: Negatif** | | |

Tabel 4.35 Hasil Manualisasi Data Uji 5

|  |  |  |
| --- | --- | --- |
| **Data Uji 5** | | |
| Orang lain pada ribut sama keadaan kosan yang sudah ditinggal berbulan-bulan terus ribut gimana caranya balik ke kosan. Aku anteng-anteng saja jadi penghuni kos dari awal pemerintah nyuruh dirumah saja dan kuliah jadi daring | | |
| **Negatif** | **Netral** | **Positif** |
| 3.815427230622958e-18 | 2.1121804600458974e-17 | 6.5992632732260386e-18 |
| **Actual: Netral** | | |
| **Prediction: Netral** | | |

### Manualisasi Evaluasi Confusion Matrix

Pada tahapan ini akan dijelaskan hasil evaluasi yang didapatkan dari pengujian yang sudah dijelaskan sebelumnya. Berdasarkan pengujian sebelumnya berikut adalah hasil klasifikasi yang didapatkan yang akan ditampilkan pada Tabel 4.36.

Tabel 4.36 Manualisasi Confusion Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted** | | | |
| **Actual** |  | **Negatif** | **Netral** | **Positif** |
| **Negatif** | 2 | 0 | 0 |
| **Netral** | 0 | 1 | 0 |
| **Positif** | 1 | 0 | 1 |

Setelah tabel *confusion matrix* dibuat, langkah selanjutnya adalah menghitung *accuracy* keseluruhan dan *accuracy, precision, recall,* dan *f-measure* tiap kelasnya*.* Untuk *accuracy* keseluruhan dapat dihitung sebagai berikut.

Sehingga untuk *accuracy* keseluruhan didapatkan 0.8 atau jika diubah dalam bentuk persentase menjadi **80%.**

Untuk melakukan perhitungan *accuracy, precision, recall,* dan *f-measure* tiap kelasnya diperlukan pencarian TP, FN, FP, TN terlebih dahulu yang memiliki tiap-tiap istilah tersebut memiliki definisi sebagai berikut yang akan ditampilkan pada Tabel 4.37.

Tabel 4.37 Definisi TP, FN, FP, dan TN

|  |  |
| --- | --- |
| **Note** | |
| **TP** | Jumlah benar kelas tersebut |
| **FN** | Jumlah baris kelas tersebut tanpa TP |
| **FP** | Jumlah kolom kelas tersebut tanpa TP |
| **TN** | Jumlah semua baris dan kolom kecuali baris dan kolom kelas tersebut |

Sehingga dapat ditentukan TP, FN, FP, TN untuk setiap kelasnya sebagaimana yang akan ditampilkan pada Tabel 4.38.

Tabel 4.38 Hasil Manualisasi TP, FN, FP, dan TN setiap kelas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Negatif** | **TP** | **FN** | **FP** | **TN** |
| 2 | 0 | 1 | 2 |
| **Netral** | **TP** | **FN** | **FP** | **TN** |
| 1 | 0 | 0 | 4 |
| **Positif** | **TP** | **FN** | **FP** | **TN** |
| 1 | 1 | 0 | 3 |

Setelah semua didapatkan, kita bisa langsung menghitung *accuracy, precision, recall,* serta *f-measure* tiap kelasnya. Berikut adalah perhitungan *accuracy*, *precision, recall, f-measure* untuk masing-masing kelasnya.

Berikut adalah perhitungan untuk kelas Negatif.

Berikut adalah perhitungan untuk kelas Netral.

Berikut adalah perhitungan untuk kelas Positif.

Setelah setiap kelas melalui proses perhitungan tersebut, berikut adalah hasil evaluasi dari masing-masing kelas yang akan ditampilkan pada Tabel 4.39.

Tabel 4.39 Hasil Evaluasi Manualisasi tiap Kelas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Negatif** | **Accuracy** | **Precision** | **Recall** | **F-Measure** |
| 0.8 | 0.67 | 1 | 0.802 |
| **Netral** | **Accuracy** | **Precision** | **Recall** | **F-Measure** |
| 1 | 1 | 1 | 1 |
| **Positif** | **Accuracy** | **Precision** | **Recall** | **F-Measure** |
| 0.8 | 1 | 0.5 | 0.67 |

## Perancangan Pengujian

Pada tahapan ini akan dijelaskan perancangan pengujian mengenai hasil klasifikasi sentimen yang terdiri dari kelas negatif, netral, dan positif dengan menggunakan metode *Naïve Bayes Multinomial* serta pembuatan daftar *stopword* dengan *Term Based Random Sampling*. Pada bagian ini ada 3 skenario perancangan pengujian yang akan dilakukan.

### Perancangan Pengujian Terhadap Pengaruh Parameter X, Y, dan L terhadap Hasil Evaluasi Sistem menggunakan K-fold Cross Validation.

Dalam melakukan perancangan pengujian terhadap pengaruh parameter X, Y, dan L terhadap hasil evaluasi diperlukan tabel pengujian yang dapat dilihat dalam Tabel 4.40.

Tabel 4.40 Perancangan Pengujian pengaruh X, Y, L terhadap Hasil Evaluasi

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Parameter** | | | **Hasil Evaluasi** | | | | | | | | | | | | | | |
| **X** | **Y** | **L** | **Acc per Fold** | **Prec per Fold** | | | **Recall per Fold** | | | | | **F-Measure per Fold** | | | | | **Final Acc** |
| **Neg** | **Net** | **Pos** | **Neg** | | **Net** | | **Pos** | **Neg** | | **Net** | | **Pos** |
| 10 | 10 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 20 | 10 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 30 | 10 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 40 | 10 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 50 | 10 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 10 | 10 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 10 | 20 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 10 | 30 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 10 | 40 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| 10 | 50 | 10 |  |  |  |  |  |  | |  | |  |  | |  | |  |
| … | … | … |  |  |  |  |  |  | |  | |  |  | |  | |  |

### Perancangan Pengujian Pengaruh Stopword Term Based Random Sampling dalam Akurasi Sistem.

Dalam melakukan perancangan pengujian Pengaruh *Stopword* *Term Based Random Sampling* dalam Akurasi Sistem diperlukan tabel pengujian yang dapat dilihat dalam Tabel 4.41.

Tabel 4.41 Perancangan Pengujian Pengaruh *Stopword Term Based Random Sampling* dalam Akurasi Sistem

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **k-fold** | **Stopword** | **Hasil Evaluasi** | | | | | | | | | |
| **Accuracy** | **Precision** | | | **Recall** | | | **F-Measure** | | |
| **Neg** | **Net** | **Pos** | **Neg** | **Net** | **Pos** | **Neg** | **Net** | **Pos** |
| 1 | Tanpa Stopword |  |  |  |  |  |  |  |  |  |  |
| TBRS |  |  |  |  |  |  |  |  |  |  |
| 2 | Tanpa Stopword |  |  |  |  |  |  |  |  |  |  |
| TBRS |  |  |  |  |  |  |  |  |  |  |
| Dst. | ….. |  |  |  |  |  |  |  |  |  |  |

### Perancangan Pengujian Perbandingan Akurasi Pengunaan Stopword Tala dan Stopword Term Based Random Sampling.

Dalam melakukan perancangan pengujian terhadap Akurasi Pengunaan Stopword Tala dan Stopword Term Based Random Sampling diperlukan tabel pengujian yang dapat dilihat dalam Tabel 4.42.

Tabel 4.42 Perancangan Pengujian Perbandingan Akurasi Pengunaan Stopword Tala dan Stopword Term Based Random Sampling

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **k-fold** | **Stopword** | **Hasil Evaluasi** | | | | | | | | | | | | | | | |
| **Accuracy** | **Precision** | | | | | **Recall** | | | | | **F-Measure** | | | | |
| **Neg** | | **Net** | | **Pos** | **Neg** | | **Net** | | **Pos** | **Neg** | | **Net** | | **Pos** |
| 1 | Tala |  |  |  | |  | |  |  | |  | |  |  | |  | |
| TBRS |  |  |  | |  | |  |  | |  | |  |  | |  | |
| 2 | Tala |  |  |  | |  | |  |  | |  | |  |  | |  | |
| TBRS |  |  |  | |  | |  |  | |  | |  |  | |  | |
| Dst. | ….. |  |  |  | |  | |  |  | |  | |  |  | |  | |

# IMPLEMENTASI

Pada bab ini akan dijelaskan implementasi kode program berdasarkan yang sudah dirancang serta dilakukan penjelasan tiap baris pada kode program.

## Implementasi Preprocessing

Pada bagian ini preprocessing ini akan meliputi proses *case folding, cleaning, stemming, stopword removal* (opsional), serta *tokenizing*. Tahapan *preprocessing* ini diawali dengan pemanggilan *library* yang diperlukan untuk prosesnya seperti pada Kode Program 5.1.

|  |  |
| --- | --- |
| Import library | |
| 1  2 | import re  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory |

Kode Program 5.1 Import Library

Penjelasan Kode Program 5.1:

|  |  |
| --- | --- |
| Baris 1 | Digunakan untuk *import library re* |
| Baris 2 | Digunakan untuk *import library StemmerFactory dari Sastrawi* |

Setelah tahapan *import library*, tahapan selanjutnya adalah tahapan *setup library* yang ditunjukkan pada Kode Program 5.2.

|  |  |
| --- | --- |
| Setup library | |
| 1  2 | stemmerFactory = StemmerFactory()  self.stemmer = stemmerFactory.create\_stemmer() |

Kode Program 5.2 Setup Library

Penjelasan Kode Program 5.2:

|  |  |
| --- | --- |
| Baris 1-2 | Digunakan untuk pembuatan fungsi stemmer dari objek instansiasi kelas *StemmerFactory* yang memanggil fungsi create\_stemmer |

Setelah tahapan *setup library*, tahapan selanjutnya adalah tahapan *preprocessing* yang ditunjukkan pada Kode Program 5.3.

|  |  |
| --- | --- |
| Algoritma Preprocessing | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | def preprocessing(self,data,stopwords=None):      for i in range(len(data)):          case\_folding = data[i].lower()          remove\_newline = case\_folding.replace("\n"," ")          cleaning = re.sub(r'[^a-zA-Z]', " ",remove\_newline)          if stopwords != None:              stemming = self.stemmer.stem(cleaning)              filtered\_words = [word for word in stemming.split() if word not in stopwords]              self.cleaned\_data.append(" ".join(filtered\_words))              tokenizing = [word for word in filtered\_words if word.isalpha()]          else:              stemming = self.stemmer.stem(cleaning)              self.cleaned\_data.append(stemming)              tokenizing = [word for word in stemming.split() if word.isalpha()]          for word in tokenizing:              self.token.append(word)              if word not in self.terms:                  self.terms.append(word)      return self.cleaned\_data, self.terms |

Kode Program 5.3 Preprocessing

Penjelasan Kode Program 5.3:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method preprocessing* dengan parameter *data,* dan *stopwords* yang memiliki *default value None*. |
| Baris 2 | Melakukan proses perulangan I hingga sepanjang variabel data |
| Baris 3 | Mengecilkan huruf didalam data index i dan disimpan dalam case\_folding |
| Baris 4 | Menghilangkan *newline* dari dalam kalimat dan menggantinya dengan spasi dan disimpan pada remove\_newline |
| Baris 5 | Menghilangkan *non-alphabet* dari dalam kalimat dan disimpan pada cleaning |
| Baris 6 | Melakukan seleksi jika *stopwords* tidak sama dengan *None* |
| Baris 7 | Melakukan *stemming* variabel *cleaning* dan disimpan ke dalam *stemming* |
| Baris 8 | Melakukan proses penghapusan kata *stopwords* didalam variabel *stemming* dan disimpan dalam *filtered\_words* |
| Baris 10 | Menambahkan *filtered\_words* yang sudah di *join* dengan spasikedalam *cleaned\_data.* |
| Baris 12 | Melakukan tokenisasi kata didalam *filtered\_words* |
| Baris 14 | Selain itu |
| Baris 15 | Melakukan *stemming* variabel *cleaning* dan disimpan ke dalam *stemming* |
| Baris 16 | Menambahkan *stemming* kedalam *cleaned\_data.* |
| Baris 17 | Melakukan tokenisasi kata didalam *stemming* |
| Baris 19-22 | Melakukan perulangan kata variabel *tokenizing* untuk memasukan setiap kata ke dalam *token*, serta melakukan seleksi jika kata belum ada di *terms* maka kata tersebut dimasukkan ke dalam variabel *terms* |
| Baris 23 | Mengembalikan *self.cleaned\_data*, dan *self.terms* |

Setelah tahapan *preprocessing*, tahapan selanjutnya adalah tahapan *get token* yang ditunjukkan pada Kode Program 5.4.

|  |  |
| --- | --- |
| Get Token | |
| 1  2 | def get\_token(self):      return self.token |

Kode Program 5.4 *Get Token*

Penjelasan Kode Program 5.4:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method get\_token*. |
| Baris 2 | Mengembalikan *self.token* |

Setelah tahapan *get token*, tahapan selanjutnya adalah tahapan *remove stopword* yang digunakan untuk menghapus *stopword* jika tahap *preprocessing* awal dilakukan tanpa penghapusan *stopword*. Proses ini ditunjukkan pada Kode Program 5.5.

|  |  |
| --- | --- |
| Remove Stopword | |
| 1  2  3  4  5  6  7  8  9  10  1112 | def remove\_stopword(self,cleaned\_data,stopwords):      new\_cleaned\_data = []      new\_terms = []      for data in cleaned\_data:          filtered\_words = [word for word in data.split() if word not in stopwords]          new\_cleaned\_data.append(" ".join(filtered\_words))          for word in filtered\_words:  self.token.append(word)              if word not in new\_terms:                  new\_terms.append(word)      return new\_cleaned\_data, new\_terms |

Kode Program 5.5 Remove Stopword

Penjelasan Kode Program 5.5:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method remove\_stopword* dengan parameter *cleaned\_data,* dan *stopwords*. |
| Baris 2-3 | Mendefinisikan *new\_cleaned\_data* dan *new\_terms* sebagai list |
| Baris 4 | Melakukan perulangan data dalam *cleaned\_data* |
| Baris 5 | Melakukan proses penghapusan kata *stopwords* didalam variabel *data* dan disimpan dalam *filtered\_words* |
| Baris 7 | Menambahkan *filtered\_words* yang sudah di *join* dengan spasikedalam new\_*cleaned\_data.* |
| Baris 8-11 | Melakukan perulangan kata variabel *filtered\_words* untuk memasukan setiap kata ke dalam *token*, serta melakukan seleksi jika kata belum ada di *new\_terms* maka kata tersebut dimasukkan ke dalam variabel *new*\_*terms* |
| Baris 12 | Mengembalikan *new\_cleaned\_data*, dan *new\_terms* |

## Implementasi Term Based Random Sampling

Pada bagian ini preprocessing ini akan meliputi proses *case folding, cleaning, stemming, stopword removal* (opsional), serta *tokenizing*. Tahapan *preprocessing* ini diawali dengan pemanggilan *library* yang diperlukan untuk prosesnya seperti pada Kode Program 5.6.

|  |  |
| --- | --- |
| Import library | |
| 1  2  3  4 | import numpy as np  import re  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory  from random import randint |

Kode Program 5.6 Import Library

Penjelasan Kode Program 5.6:

|  |  |
| --- | --- |
| Baris 1 | Digunakan untuk *import library numpy* sebagai *np* |
| Baris 2 | Digunakan untuk *import library re* |
| Baris 3 | Digunakan untuk *import library StemmerFactory dari Sastrawi* |
| Baris 4 | Digunakan untuk *import library randint* dari *random* |

Setelah tahapan *import library*, tahapan selanjutnya adalah tahapan *setup library* yang ditunjukkan pada Kode Program 5.7.

|  |  |
| --- | --- |
| Setup library | |
| 1  2  3 | def setup\_library(self):      stemmerFactory = StemmerFactory()      self.stemmer = stemmerFactory.create\_stemmer() |

Kode Program 5.7 Setup Library

Penjelasan Kode Program 5.7:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method setup\_library* |
| Baris 2-3 | Digunakan untuk pembuatan fungsi stemmer dari objek instansiasi kelas *StemmerFactory* yang memanggil fungsi create\_stemmer |

Setelah tahapan *setup library*, tahapan selanjutnya adalah tahapan persiapan *method* yang digunakan dalam proses pembuatan *stopwords.* Proses diawali dengan *method generate random words* yang ditunjukkan pada Kode Program 5.8.

|  |  |
| --- | --- |
| Algoritme Generate Random Words | |
| 1  2 | def generate\_random\_words(self,token):      return token[randint(0,len(token)-1)] |

Kode Program 5.8 Generate Random Words

Penjelasan Kode Program 5.8:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method generate\_random\_words* dengan parameter *token* |
| Baris 2 | Mengembalikan token dengan index yang ditentukan secara random dari 0 hingga panjang token dikurang 1 |

Setelah tahapan *generate random words*, tahapan selanjutnya adalah tahapan *get documents contains words* yang ditunjukkan pada Kode Program 5.9

|  |  |
| --- | --- |
| Algoritme Get Documents Contains Words | |
| 1  2  3  4  5  6 | def get\_documents\_contains\_words(self,words,documents):      sampled\_documents = []      for tweet in documents:          if words in tweet.split():              sampled\_documents.append(tweet)      return sampled\_documents |

Kode Program 5.9 Get Documents Contains Words

Penjelasan Kode Program 5.9:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method get\_documents\_contains\_words* dengan parameter *words* dan *documents* |
| Baris 2 | Mendefinisikan *sampled\_documents* sebagai list |
| Baris 3 | Melakukan perulangan tweet dalam documents |
| Baris 4 - 5 | Jika *words* ada didalam *list* *tweet* maka *tweet* dimasukkan ke *sampled\_documents*. |
| Baris 6 | Mengembalikan *sampled\_documents* |

Setelah tahapan *get documents contains words*, tahapan selanjutnya adalah tahapan *count words* yang ditunjukkan pada Kode Program 5.10.

|  |  |
| --- | --- |
| Algortime Count Words | |
| 1  2  3  4  5  6  7 | def count\_words(self,word, documents):      count = 0      for tweet in documents:          for w in tweet.split():              if word == w:                  count+=1      return float(count) |

Kode Program 5.10 Count Words

Penjelasan Kode Program 5.10:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method count\_words* dengan parameter *word* dan *documents* |
| Baris 2 | Mendefinisikan *count dan diinisialisasi dengan 0* |
| Baris 3 | Melakukan perulangan *tweet* dalam *documents* |
| Baris 4 | Melakukan perulangan w didalam *tweet* yang di *split* |
| Baris 5 - 6 | Jika *word* sama dengan w maka *count* ditambah 1. |
| Baris 7 | Mengembalikan *count* yang diubah menjadi *float* |

Setelah tahapan *count words*, tahapan selanjutnya adalah tahapan *get sum of the length document* yang ditunjukkan pada Kode Program 5.11.

|  |  |
| --- | --- |
| Algoritme Get Sum of the length document | |
| 1  2  3  4  5 | def get\_sum\_of\_the\_length\_document(self,documents):      sum = 0      for tweet in documents:          sum+=len(tweet.split())      return float(sum) |

Kode Program 5.11 Get Sum of the length document

Penjelasan Kode Program 5.11:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method get\_sum\_of\_the\_length\_document* dengan parameter *documents* |
| Baris 2 | Mendefinisikan *sum dan diinisialisasi dengan 0* |
| Baris 3 | Melakukan perulangan *tweet* dalam *documents* |
| Baris 4 | Variabel *sum* ditambahkan dengan panjang *tweet* yang di *split* |
| Baris 5 | Mengembalikan *sum* yang diubah menjadi *float* |

Setelah tahapan *get sum of the length document*, tahapan selanjutnya adalah tahapan *get term* yang ditunjukkan pada Kode Program 5.12.

|  |  |
| --- | --- |
| Algoritme Get Term | |
| 1  2  3  4  5  6  7 | def get\_term(self,documents):      term = []      for tweet in documents:          for word in tweet.split():              if word not in term:                  term.append(word)      return term |

Kode Program 5.12 Get Term

Penjelasan Kode Program 5.12:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method get\_term* dengan parameter *documents* |
| Baris 2 | Mendefinisikan *term* sebagai *list* |
| Baris 3 | Melakukan perulangan *tweet* dalam *documents* |
| Baris 4 | Melakukan perulangan *word* dalam *tweet* yang di *split* |
| Baris 5 | Jika *word* tidak ada di dalam term |
| Baris 6 | *Word* dimasukkan kedalam *term* |
| Baris 7 | Mengembalikan *term* |

Setelah tahapan *get term*, tahapan selanjutnya adalah tahapan *get total token* yang ditunjukkan pada Kode Program 5.13

|  |  |
| --- | --- |
| Algoritme Get Total Token | |
| 1  2 | def get\_total\_token(self,token):      return float(len(token)) |

Kode Program 5.13 Get Total Token

Penjelasan Kode Program 5.13:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method get\_total\_token* dengan parameter *token* |
| Baris 2 | Mengembalikan panjang *token* dalam bentuk float |

Setelah tahapan *get total token*, tahapan selanjutnya adalah tahapan perhitungan *Kullback Leibler Divergence* yang ditunjukkan pada Kode Program 5.14.

|  |  |
| --- | --- |
| Algoritme Hitung KL Div | |
| 1  2  3  4  5  6  7  8  9  10 | def kl\_div(self,word,sampled\_documents):      tf\_x = self.count\_words(word,sampled\_documents)      l\_x = self.get\_sum\_of\_the\_length\_document(sampled\_documents)      p\_x = tf\_x / l\_x      F = self.count\_words(word,self.cleaned\_data)      token\_c = self.get\_total\_token(self.terms)      p\_c = F / token\_c      w\_t = p\_x \* np.log2(p\_x/p\_c)      return w\_t |

Kode Program 5.14 Hitung KL Div

Penjelasan Kode Program 5.14:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method kl\_div* dengan parameter *word* dan *documents* |
| Baris 2 | Menghitung nilai *tf\_x* dengan memanggil fungsi *count\_words* dengan *parameter word* dan *sampled\_documents.* |
| Baris 3 | Menghitung nilai l*\_x* dengan memanggil fungsi *get\_sum\_of\_the\_length\_document* dengan *parameter sampled\_documents.* |
| Baris 5 | Menghitung nilai *p\_x* dengan cara *tf\_x* dibagi dengan *l\_x* |
| Baris 6 | Menghitung nilai *F* dengan memanggil *method* *count\_words* dengan parameter *word* dan *self.cleaned\_data* |
| Baris 7 | Menghitung nilai *token\_c* dengan memanggil *method get\_total\_token* dengan parameter *self.terms* |
| Baris 8 | Menghitung nilai *p\_c* dengan cara *F* dibagi dengan *token\_c* |
| Baris 9 | Menghitung w\_t dengan cara *p\_x* dikali dengan hasil log2 dari *p\_x* dibagi dengan *p\_c* |
| Baris 10 | Mengembalikan bobot |

Setelah tahapan perhitungan *Kullback Leibler Divergence,* tahapan selanjutnya adalah tahapan perhitungan *create stopwords* yang memanggil *method-method* yang sudah dibuat sebelumnyayang ditunjukkan pada Kode Program 5.15.

|  |  |
| --- | --- |
| Algoritma Create Stopwords | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | def create\_stopwords(self,cleaned\_data,terms):      self.cleaned\_data = cleaned\_data      self.terms = terms      for i in range(self.Y):          w\_random = self.generate\_random\_words(self.terms)          sampled\_documents = self.get\_documents\_contains\_words(w\_random, self.cleaned\_data)          term\_sampled\_documents = self.get\_term(sampled\_documents)          token\_w = {}          for word in term\_sampled\_documents:              token\_w[word] = self.kl\_div(word,sampled\_documents)              if word not in self.token\_used:                  self.token\_used.append(word)          maximum = max(token\_w, key=token\_w.get)          minimum = min(token\_w, key=token\_w.get)          max\_weight\_term = token\_w[maximum]          min\_weight\_term = token\_w[minimum]          normalized\_term\_weight = {}          for k,v in token\_w.items():              normalized\_term\_weight[k] = ( v - min\_weight\_term) / (max\_weight\_term - min\_weight\_term)          sort\_term\_weight = sorted(normalized\_term\_weight.items(), key=lambda x: x[1])          sorted\_term\_weight = {}          count = 0          for i in sort\_term\_weight:              if count < self.X:                  sorted\_term\_weight[i[0]] = i[1]              else:                  break              count+=1          self.token\_weight.append(sorted\_term\_weight)      weighted\_token = {}      for used\_tok in self.token\_used:          temp = []          for tok\_w in self.token\_weight:              if used\_tok in tok\_w:                  temp.append(tok\_w[used\_tok])          weighted\_token[used\_tok] = temp      merged\_weighted\_token = {}      for k,v in weighted\_token.items():          if len(v) != 0:              merged\_weighted\_token[k] = np.mean(v)        sorted\_merged\_weighted\_token = sorted(merged\_weighted\_token.items(), key=lambda x: x[1])      sorted\_final\_weight = {}      count = 0      l\_value = int(len(sorted\_merged\_weighted\_token) \* ( self.L / 100))      for i in sorted\_merged\_weighted\_token:          if count < l\_value:              sorted\_final\_weight[i[0]] = i[1]          else:              break          count+=1      stopwords = []      for k, v in sorted\_final\_weight.items():          stopwords.append(k)      return stopwords |

Kode Program 5.15 Create Stopwords

Penjelasan Kode Program 5.15:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan *method create\_stopwords* dengan parameter *cleaned\_data* dan *terms* |
| Baris 2-3 | Memasukan cleaned\_data dan terms kedalam variabel kelas ini |
| Baris 5 | Melakukan perulangan index i sepanjang Y |
| Baris 6 | Memanggil fungsi generate\_random\_words dengan parameter terms lalu dimasukan ke w\_random |
| Baris 7 | Memanggil fungsi get\_documents\_contains\_words dengan parameter w\_random dan cleaned\_data lalu dimasukan ke dalam sampled\_documents |
| Baris 9 | Memanggil fungsi get\_term dengan parameter sampled\_documents lalu dimasukan ke dalam term\_sampled\_documents. |
| Baris 11 | Mendefinisikan token\_w sebagai dict |
| Baris 12 | Melakukan perulangan word didalam term\_sampled\_documents |
| Baris 13 | Memanggil fungsi kl\_div dengan parameter word dan sampled\_documents lalu dimasukan ke dalam token\_w dengan index word |
| Baris 15 | Jika word tidak ada di dalam token\_used maka word ditambahkan ke dalam list token\_used |
| Baris 17-20 | Mencari nilai bobot maximum dan minimum dari token\_w yang di masukkan ke dalam max\_weight\_term dan min\_weight\_term |
| Baris 21 | Mendefinisikan normalized\_term\_weight sebagai dict |
| Baris 22 | Melakukan perulangan k dan v dari item token\_w |
| Baris 23-24 | Melakukan perhitungan normalisasi bobot dengan min max dengan cara v dikurangi min\_weight\_term lalu hasilnya dibagi dengan max\_weight\_term dikurangi min\_weight\_term lalu dimasukkan kedalam normalized\_term\_weight |
| Baris 25 | Mengurutkan normalized\_term\_weight dan hasilnya dimasukkan ke dalam sort\_term\_weight |
| Baris 27 | Mendefinisikan sorted\_term\_weight sebagai dict |
| Baris 28 | Mendefinisikan count dan diinisialisasi dengan 0 |
| Baris 29 | Melakukan perulangan i dalam sort\_term\_weight |
| Baris 30 – 31 | Jika count kurang dari self.X maka i index 1 dimasukkan kedalam sorted\_term\_weight index i index 0 |
| Baris 32 – 33 | Selain itu berhentikan loop |
| Baris 34 | Count diincrement 1 |
| Baris 35 | Variabel sorted\_term\_weight dimasukkan ke dalam token\_weight |
| Baris 37 | Mendefinisikan weighted\_token sebagai dict |
| Baris 38 | Melakukan perulangan used\_tok dalam token\_used |
| Baris 39 | Mendefinisikan temp sebagai list |
| Baris 40 | Melakukan perulangan tok\_w dalam token\_weight |
| Baris 41-42 | Jika used\_tok ada didalam tok\_w maka tok\_w index used\_tok akan dimasukkan kedalam temp |
| Baris 43 | Temp dimasukkan kedalam weighted\_token dengan index used\_tok |
| Baris 45 | Mendefinisikan merged\_weighted\_token sebagai dict |
| Baris 46 | Melakukan perulangan k dan v dalam item weighted\_token |
| Baris 47 – 48 | Jika panjang v tidak sama dengan 0 maka v akan dicari rata-rata dan dimasukkan ke dalam merged\_weighted\_token dengan index k |
| Baris 50 | Mengurutkan merged\_weighted\_token dan dimasukkan ke dalam sorted\_merged\_weighted\_token. |
| Baris 52 | Mendefinisikan sorted\_final\_weight sebagai dict |
| Baris 53 | Mendefinisikan count dan diinisialisasi dengan 0 |
| Baris 54 | L\_value diinisialisasi dengan panjang sorted\_merged\_weighted\_token yang dikalikan dengan L dibagi 100 |
| Baris 56 | Melakukan perulangan i dalam sorted\_merged\_weighted\_token |
| Baris 57 – 58 | Jika count kurang dari l\_value maka sorted\_final\_weight indeks i indeks 0 diinisialisasi dengan i indeks 1 |
| Baris 59 – 60 | Selain itu maka perulangan diberhentikan |
| Baris 61 | Count diincrement 1 |
| Baris 63 | Mendefinisikan stopword sebagai list |
| Baris 64 - 65 | Melakukan perulangan k dan v dalam item sorted\_final\_weight dan setiap perulangannya k ditambahkan ke dalam stopwords |
| Baris 67 | Mengembalikan nilai stopwords |

## Implementasi Term Weighting

Pada bagian term weighting ini akan meliputi proses *raw term frequency, log term frequency, inverse document frequency,* dan *term frequency-inverse document frequency*. Tahapan *term weighitng* ini diawali dengan pemanggilan *library* yang diperlukan untuk prosesnya seperti pada Kode Program 5.17.

|  |  |
| --- | --- |
| Import library | |
| 1 | Import math |

Kode Program 5.16 Import Library

Penjelasan Kode Program 5.16:

|  |  |
| --- | --- |
| Baris 1 | Digunakan untuk *import library math* |

Setelah tahapan *import library*, tahapan selanjutnya adalah tahapan *count word* yang ditunjukkan pada Kode Program 5.17.

|  |  |
| --- | --- |
| Count Word | |
| 1  2  3  4  5  6 | def count\_word(self,term, document):      count = 0      for word in document.split():          if term == word:              count+=1      return count |

Kode Program 5.17 Count Word

Penjelasan Kode Program 5.17:

|  |  |
| --- | --- |
| Baris 1 | Mendefisinikan method count\_word dengan parameter term dan document |
| Baris 2 | Count diinisialisasikan dengan 0 |
| Baris 3 | Melakukan perulangan word dalam document yang di split |
| Baris 4 – 5 | Jika term sama dengan word maka count ditambah 1 |
| Baris 6 | Mengembalikan nilai count |

Setelah tahapan *count word*, tahapan selanjutnya adalah tahapan *count dft* yang ditunjukkan pada Kode Program 5.18.

|  |  |
| --- | --- |
| Count DFT | |
| 1  2  3  4  5  6 | def count\_dft(self,numbers):      count = len(self.data)      for number in numbers:          if number == 0:              count-=1      return count |

Kode Program 5.18 Count DFT

Penjelasan Kode Program 5.18:

|  |  |
| --- | --- |
| Baris 1 | Mendefisinikan method count\_dft dengan parameter numbers |
| Baris 2 | Count diinisialisasikan dengan panjang self.data |
| Baris 3 | Melakukan perulangan number dalam numbers |
| Baris 4 – 5 | Jika number sama dengan 0 maka count dikurang 1 |
| Baris 6 | Mengembalikan nilai count |

Setelah tahapan *count dft*, tahapan selanjutnya adalah tahapan *raw tf weighting* yang ditunjukkan pada Kode Program 5.19.

|  |  |
| --- | --- |
| Get Raw TF Weighting | |
| 1  2  3  4  5  6  7 | def get\_raw\_tf\_weighting(self):      for term in self.terms:          temp = []          for data in self.data:              temp.append(self.count\_word(term, data))          self.raw\_tf[term] = temp      return self.raw\_tf |

Kode Program 5.19 Get Raw TF Weighting

Penjelasan Kode Program 5.19:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_raw\_tf\_weighting |
| Baris 2 | Melakukan perulangan term dalam self.terms |
| Baris 3 | Term didefinisikan sebagai list |
| Baris 4 | Melakukan perulangan data dalam self.data |
| Baris 5 | Variabel temp ditambahkan dari hasil pemanggilan method count\_word dengan parameter term dan data |
| Baris 6 | Nilai temp dimasukkan ke dalam self.raw\_tf index term |
| Baris 7 | Mengembalikan nilai self.raw\_tf |

Setelah tahapan *raw tf weighting*, tahapan selanjutnya adalah tahapan *log tf weighting* yang ditunjukkan pada Kode Program 5.20.

|  |  |
| --- | --- |
| Get Log TF Weighting | |
| 1  2  3  4  5  6  7  8  9  10 | def get\_log\_tf\_weighting(self):      self.get\_raw\_tf\_weighting()      for term in self.terms:          temp = []          for i in range(len(self.raw\_tf[term])):              tf = 0 if (self.raw\_tf[term])[i] == 0 else 1+math.log((self.raw\_tf[term])[i],10)              temp.append(tf)          self.log\_tf[term] = temp      return self.log\_tf |

Kode Program 5.20 Get Log TF Weighting

Penjelasan Kode Program 5.20:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_log\_tf\_weighting |
| Baris 2 | Memanggil method get\_raw\_tf\_weighting |
| Baris 3 | Melakukan perulangan term dalam self.terms |
| Baris 4 | Mendefinisikan temp sebagai dict |
| Baris 5 | Melakukan perulangan i hingga panjang raw\_tf index term |
| Baris 6 – 7 | Menghitung nilai tf akan 0 jika raw\_tf index term index i sama dengan 0, selain itu nilai tf akan diisi dengan perhitungan 1 ditambah log 10 dari nilai raw\_tf term index i. |
| Baris 8 | Nilai tf dimasukkan kedalam temp |
| Baris 9 | Nilai temp dimasukkan kedalam self.log\_tf |
| Baris 10 | Mengembalikan nilai self.log\_tf |

Setelah tahapan log tf weighting, tahapan selanjutnya adalah tahapan *calculate idf* yang ditunjukkan pada Kode Program 5.21

|  |  |
| --- | --- |
| Calculate IDF | |
| 1  2  3  4  5  6 | def calculate\_idf(self):      for term in self.terms:          df = self.count\_dft(self.raw\_tf[term])          idf\_value = math.log(len(self.data)/df,10)          self.idf.append(idf\_value)      return self.idf |

Kode Program 5.21 Calculate IDF

Penjelasan Kode Program 5.21:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method calculate\_idf |
| Baris 2 | Melakukan perulangan term dalam self.terms |
| Baris 3 | Menghitung df dengan memanggil method count\_dft dengan parameter self.raw\_tf index term |
| Baris 4 | Menghitung idf value dengan menghitung log 10 dari hasil pembagian panjang self.data dan df |
| Baris 5 | Nilai idf value dimasukkan kedalam self.idf |
| Baris 6 | Mengembalikan nilai self.idf |

Setelah tahapan calculate idf, tahapan selanjutnya adalah tahapan *get idf* yang ditunjukkan pada Kode Program 5.22

|  |  |
| --- | --- |
| Get IDF | |
| 1  2 | def get\_idf(self):      return self.idf |

Kode Program 5.22 Get IDF

Penjelasan Kode Program 5.21:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_idf |
| Baris 2 | Mengembalikan nilai self.idf |

Setelah tahapan get idf, tahapan selanjutnya adalah tahapan *get tf idf weighting* yang ditunjukkan pada Kode Program 5.22

|  |  |
| --- | --- |
| Get TF IDF Weighting | |
| 1  2  3  4  5  6  7  8  9  10  11  12 | def get\_tf\_idf\_weighting(self):      self.get\_log\_tf\_weighting()      self.calculate\_idf()      count = 0      for term in self.terms:          temp = []          for i in range(len(self.data)):              tfidf\_value = self.log\_tf[term][i]\*self.idf[count]              temp.append(tfidf\_value)          self.tf\_idf[term] = temp          count+=1      return self.tf\_idf |

Kode Program 5.23 Get TF IDF Weighting

Penjelasan Kode Program 5.23:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_tf\_idf\_weighting |
| Baris 2 | Memanggil method get\_log\_tf\_weighting |
| Baris 3 | Memanggil method calculate\_idf |
| Baris 4 | Count diinisialisasikan dengan 0 |
| Baris 5 | Melakukan perulangan term dalam self.terms |
| Baris 6 | Mendefinisikan term sebagai list |
| Baris 7 | Melakukan perulangan i hingga sepanjang self.data |
| Baris 8 | Menghitung tfidf value dengan perkalian self.log\_tf index term index i dengan self.idf index count |
| Baris 9 | Niai tfidf value dimasukkan kedalam temp |
| Baris 10 | Nilai temp dimasukkan kedalam self.tf\_idf index term |
| Baris 11 | Nilai count ditambahkan 1 |
| Baris 12 | Mengembalikan nilai self.tf\_idf |

## Implementasi Naïve Bayes

Pada bagian Naïve Bayes ini akan dibagi menjadi 2 bagian yaitu bagian training atau pelatihan dan bagian testing atau pengujian. Sebelum masuk ke dalam bagian training maupun testing. Tahapan *naïve bayes* ini diawali dengan beberapa fungsi yang diperlukan dalam perhitungan pelatihan maupun pengujian. Tahapan awal tahap ini diawali dengan pemanggilan *library* dan kelasyang diperlukan seperti yang ditunjukkan pada Kode Program 5.24.

|  |  |
| --- | --- |
| Import library dan kelas | |
| 1  2  3  4 | import re  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory  from preprocessing import Preprocessing  from weighting import Weighting |

Kode Program 5.24 Import Library

Penjelasan Kode Program 5.24:

|  |  |
| --- | --- |
| Baris 1 | Digunakan untuk *import library math* |
| Baris 2 | Digunakan untuk *import library StemmerFactory* dari Sastrawi |
| Baris 3 | Digunakan untuk *import* kelas *Preprocessing* dari *file preprocessing* |
| Baris 4 | Digunakan untuk *import* kelas *Weighting* dari *file weighting* |

Setelah tahapan *import library* dan kelas, tahapan selanjutnya adalah tahapan *count word* yang ditunjukkan pada Kode Program 5.25.

|  |  |
| --- | --- |
| Count Word | |
| 1  2  3  4  5  6 | def count\_word(self,term,document):      count = 0      for word in document.split():          if term == word:              count += 1      return count |

Kode Program 5.25 Count Word

Penjelasan Kode Program 5.25:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method count\_word dengan parameter term dan document |
| Baris 2 | Count diinisialisasi dengan 0 |
| Baris 3 | Melakukan perulangan word dalam document yang di split |
| Baris 4 | Jika term sama dengan word maka count ditambah 1 |
| Baris 5 | Mengembalikan nilai count |

Setelah tahapan *count word* dan kelas, tahapan selanjutnya adalah tahapan *count specific word in category* yang ditunjukkan pada Kode Program 5.26.

|  |  |
| --- | --- |
| Count Specific Word in Category | |
| 1  2  3  4  5  6  7  8 | def count\_specific\_word\_in\_category(self,word,category):      wct = 0      indexDocument = 0      for wt in self.weighted\_terms[word]:          if self.target[indexDocument]==category:              wct = wct + wt          indexDocument += 1      return wct |

Kode Program 5.26 Count Specific Word in Category

Penjelasan Kode Program 5.26:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method count\_specific\_word\_in\_category dengan parameter word dan category |
| Baris 2 – 3 | Wct dan indexDocument diinisialisasi dengan 0 |
| Baris 4 | Melakukan perulangan wt dalam self.weighted\_terms index word |
| Baris 5 - 6 | Jika self.target index indexDocument sama dengan category maka wct ditambah wt |
| Baris 7 | Nilai indexDocument ditambah 1 |
| Baris 8 | Mengembalikan nilai wct |

Setelah tahapan *count specific word in category*, tahapan selanjutnya adalah tahapan *count all word in category* yang ditunjukkan pada Kode Program 5.27.

|  |  |
| --- | --- |
| Count All Word in Category | |
| 1  2  3  4  5  6  7  8 | def count\_all\_word\_in\_category(self,category):      counter = 0      indexDocument = 0      for totalTiapDokumen in self.total:          if self.target[indexDocument]==category:              counter = counter + totalTiapDokumen          indexDocument += 1      return counter |

Kode Program 5.27 Count All Word in Category

Penjelasan Kode Program 5.27:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method count\_all\_word\_in\_category dengan parameter category |
| Baris 2 – 3 | Counter dan indexDocument diinisialisasi dengan 0 |
| Baris 4 | Melakukan perulangan totalTiapDokumen dalam self.total |
| Baris 5 - 6 | Jika self.target index indexDocument sama dengan category maka counter ditambah totalTiapDokumen |
| Baris 7 | Nilai indexDocument ditambah 1 |
| Baris 8 | Mengembalikan nilai counter |

Setelah tahapan *count all word in category*, tahapan selanjutnya adalah tahapan *count all word in category* yang ditunjukkan pada Kode Program 5.28.

|  |  |
| --- | --- |
| Get Total IDF | |
| 1  2  3  4  5 | def get\_total\_idf(self):      idf\_total = 0      for idf\_item in self.idf:          idf\_total+=idf\_item      return idf\_total |

Kode Program 5.28 Get Total IDF

Penjelasan Kode Program 5.28:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_total\_idf |
| Baris 2 | Idf\_total diinisialisasi dengan 0 |
| Baris 3 | Melakukan perulangan idf\_item dalam self.idf |
| Baris 4 | Idf\_total ditambah dengan idf\_iem |
| Baris 5 | Mengembalikan idf\_total |

Setelah tahapan *get total idf,* tahapan selanjutnya adalah tahapan *calculate probability multinomial* yang ditunjukkan pada Kode Program 5.29.

|  |  |
| --- | --- |
| Calcaulte Probability Multinomial | |
| 1  2  3  4 | def calculate\_probability\_multinomial(self,word, category):      return (self.count\_specific\_word\_in\_category(word, category) + 1) / (self.count\_all\_word\_in\_category(category) + self.get\_total\_idf()) |

Kode Program 5.29 Calculate Probability Multinomial

Penjelasan Kode Program 5.29:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method calculate\_probability\_multinomial dengan parameter word dan category |
| Baris 2 | Mengembalikan nilai perhitungan dari pemanggilan fungsi count\_specific\_word\_in\_category dengan parameter word dan category ditambah 1 dan dibagi dengan hasil pertambahan dari pemanggilan fungsi count\_all\_word\_in\_category parameter category dengan get\_total\_idf |

Setelah tahapan *calculate probability multinomial*, tahapan selanjutnya adalah tahapan *get total document* yang ditunjukkan pada Kode Program 5.30.

|  |  |
| --- | --- |
| Get Total Document | |
| 1  2 | def get\_total\_document(self):      return len(self.cleaned\_data) |

Kode Program 5.30 Get Total Document

Penjelasan Kode Program 5.30:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_total\_document |
| Baris 2 | Mengembalikan nilai panjang dari self.cleaned\_data |

Setelah tahapan *get total document*, tahapan selanjutnya adalah tahapan *get total document with specific category* yang ditunjukkan pada Kode Program 5.31.

|  |  |
| --- | --- |
| Get Total Document With Specific Category | |
| 1  2 | def get\_total\_document\_with\_specific\_category(self,category):      return len([tgt for tgt in self.target if tgt == category]) |

Kode Program 5.31 Get Total Document With Specific Category

Penjelasan Kode Program 5.31:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method calculate\_probability\_multinomial dengan parameter word dan category |
| Baris 2 | Mengembalikan nilai perhitungan dari pemanggilan fungsi count\_specific\_word\_in\_category dengan parameter word dan category ditambah 1 dan dibagi dengan hasil pertambahan dari pemanggilan fungsi count\_all\_word\_in\_category parameter category dengan get\_total\_idf |

Setelah tahapan *get total document with specific category*, tahapan selanjutnya adalah tahapan pelatihan dan pengujian.

### Implementasi Naïve Bayes Training

Pada bagian Naïve Bayes Training ini akan dijelaskan tahapan-tahapan pelatihan Naïve Bayes yaitu mencari prior serta likelihood untuk tiap kelas menggunakan method-method yang sudah ditunjukkan sebelumnya. Tahapan ini akan ditunjukkan pada Kode Program 5.32.

|  |  |
| --- | --- |
| Fit | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  2829  30  3132  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | def fit(self, cleaned\_data, terms, target, stopwords, idf, weight = None):      self.cleaned\_data = cleaned\_data      self.terms = terms      self.target = target      if weight == None:          weighting = Weighting(self.cleaned\_data, self.terms)          self.weighted\_terms = weighting.get\_tf\_idf\_weighting()      else:          self.weighted\_terms = weight          self.idf = idf      self.stopwords = stopwords      for i in range(len(self.cleaned\_data)):          total\_word = 0          for term in self.terms:              total\_word += self.weighted\_terms[term][i]          self.total.append(total\_word)        for term in self.terms:          self.con\_prob\_negative.append(self.calculate\_probability\_multinomial(term, 'Negatif'))          self.con\_prob\_neutral.append(self.calculate\_probability\_multinomial(term, 'Netral'))          self.con\_prob\_positive.append(self.calculate\_probability\_multinomial(term, 'Positif'))        self.likelihood = {}      indexKomentar = 0      for term in self.terms:          temp = []          temp.append(self.con\_prob\_negative[indexKomentar])          temp.append(self.con\_prob\_neutral[indexKomentar])          temp.append(self.con\_prob\_positive[indexKomentar])          self.likelihood[term] = temp          indexKomentar += 1      self.prior\_negative = self.get\_total\_document\_with\_specific\_category(          'Negatif') / self.get\_total\_document()      self.prior\_neutral = self.get\_total\_document\_with\_specific\_category(          'Netral') / self.get\_total\_document()      self.prior\_positive = self.get\_total\_document\_with\_specific\_category(          'Positif') / self.get\_total\_document() |

Kode Program 5.32 Naive Bayes Training

Penjelasan Kode Program 5.32:

|  |  |
| --- | --- |
| Baris 1 – 2 | Mendefinisikan method *fit* dengan parameter cleaned\_data, terms, target, stopwords, idf, dan weight yang memiliki default value None |
| Baris 3 – 5 | Memasukan cleaned\_data, terms, dan target ke dalam variabel tersebut didalam kelas ini |
| Baris 6 - 8 | Jika weight sama dengan None maka dilakukan proses perhitungan weighting dengan memanggil kelas Weighting dan memanggil fungsi get\_tf\_idf\_weighting dan memasukkannya ke dalam variabel self.weighted\_terms |
| Baris 9 - 11 | Selain itu weight dimasukkan ke dalam self.weighted\_terms dan idf dimasukkan ke dalam self.idf |
| Baris 12 | Stopwords dimasukkan kedalam self.stopwords |
| Baris 14 | Melakukan perulangan i hingga sepanjang cleaned\_data |
| Baris 15 | Total\_word diinisialisasi dengan 0 |
| Baris 16 | Melakukan perulangan term dalam self.terms |
| Baris 17 | Total\_word ditambah dengan self.weighted\_terms index term index i |
| Baris 18 | Total\_word dimasukkan kedalam self.total |
| Baris 20 | Melakukan perulangan term dalam self.terms |
| Baris 21 - 26 | Memanggil calculate\_probability\_multinomial dengan parameter term dan setiap kategorinya dan dimasukkan masing-masing kedalam self.con\_prob\_negative, self.con\_prob\_netral, self.con\_prob\_positive |
| Baris 28 | Mendefinisikan likelihood sebagai dict |
| Baris 29 | indexKomentar diinisialisasi dengan 0 |
| Baris 30 | Melakukan perulangan term dalam self.terms |
| Baris 31 | Temp didefinisikan sebagai list |
| Baris 32 - 34 | Temp dimasukkan self.con\_prob\_negative, self.con\_prob\_netral, self.con\_prob\_positive masing-masing index indexKomentar |
| Baris 35 - 36 | Temp dimasukkan kedalam self.likelihood index term dan indexKomentar ditambah 1 |
| Baris 38 - 46 | Menghitung prior setiap kelas dengan memanggil fungsi get\_total\_document\_with\_specific\_category dengan parameter kelas yang dibagi dengan get\_total\_document |

### Implementasi Naïve Bayes Testing

Pada bagian Naïve Bayes Testing ini akan dijelaskan tahapan-tahapan pelatihan Naïve Bayes yaitu mencari posterior untuk tiap kelas menggunakan method-method yang sudah ditunjukkan sebelumnya. Tahapan ini akan ditunjukkan pada Kode Program 5.33.

|  |  |
| --- | --- |
| Predict | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  2829  30  3132  33  34  35  36  37 | def predict(self,data\_test,expected\_result):      self.used\_terms = []      prepro = Preprocessing()      cleaned\_data\_test, terms\_test = prepro.preprocessing([data\_test],self.stopwords)      terms\_test = prepro.get\_token()      for term in terms\_test:          if term in self.terms:              self.used\_terms.append(term)      for term in self.used\_terms:          temp = []          temp.append(self.likelihood[term][0])          temp.append(self.likelihood[term][1])          temp.append(self.likelihood[term][2])          self.used\_terms\_with\_likelihood[term] = temp      negatif = 1      netral = 1      positif = 1      for term in self.used\_terms:          negatif \*= self.used\_terms\_with\_likelihood[term][0]          netral \*= self.used\_terms\_with\_likelihood[term][1]          positif \*= self.used\_terms\_with\_likelihood[term][2]      negatif = negatif \* self.prior\_negative      netral = netral \* self.prior\_neutral      positif = positif \* self.prior\_positive      finalResult = ""      if (positif > negatif and positif > netral):          finalResult = "Positif"      elif negatif > positif and negatif > netral:          finalResult = "Negatif"      elif netral > positif and netral > negatif:          finalResult = "Netral"      return finalResult |

Kode Program 5.33 Naive Bayes Testing

Penjelasan Kode Program 5.33:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method *predict* dengan parameter data\_test dan expected\_result |
| Baris 2 | Mendefinisikan self.used\_terms sebagai list |
| Baris 3 - 5 | Membuat objek dari kelas Preprocessing dan memanggil method preprocessing dengan parameter data\_test dan self.stopwords dan dimasukkan ke dalam cleaned\_data\_test dan terms\_test |
| Baris 6 | Terms\_test dimasukkan dengan get\_token dari objek prepro |
| Baris 7 | Melakukan perulangan term dalam self.terms |
| Baris 8 - 9 | Jika term ada didalam self.terms maka term dimasukkan kedalam self.used\_term |
| Baris 11 | Melakukan perulangan term dalam self.used\_terms |
| Baris 12 | Mendefinisikan temp sebagai list |
| Baris 13 | Temp dimasukkan self.likelihood index term index 0 hingga 2 |
| Baris 16 | Temp dimasukkan ke self.used\_terms\_with\_likelihood index term |
| Baris 18 - 20 | Negatif, netral, positif diinisialisasi dengan 0 |
| Baris 21 | Melakukan perulangan term dalam self.used\_terms |
| Baris 22 | Negatif dikali sama dengan self.used\_terms\_with\_likelihood index term index 0 |
| Baris 23 | Netral dikali sama dengan self.used\_terms\_with\_likelihood index term index 1 |
| Baris 24 | Positif dikali sama dengan self.used\_terms\_with\_likelihood index term index 2 |
| Baris 26-28 | Negatif, netral, dan positif masing-masing dikali dengan prior yang sudah dihitung pada training |
| Baris 29 | Mendefinisikan finalResult sebagai String |
| Baris 30 | Jika positif lebih dari negatif dan netral maka finalResult diinisialisasi dengan Positif, selain itu jika negatif lebih dari positif dan netral maka finalResult diinisialisasi dengan Netral, dan jika netral lebih dari positif dan negatif maka finalResult diinisialisasi dengan Netral |
| Baris 37 | Mengembalikan nilai finalResult |

## Implementasi K Fold

Pada bagian implementasi k-fold cross validation ini akan diawali dengan tahapan persiapan data yang diperlukan seperti yang ditunjukkan pada Kode Program 5.34.

|  |  |
| --- | --- |
| Prepare Data | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | def \_prepare\_data(self):      self.data\_negative = []      self.data\_netral = []      self.data\_positive = []      for i in range(len(self.data)):          if self.target[i] == self.NEGATIVE:              self.data\_negative.append(self.data[i])          elif self.target[i] == self.NETRAL:              self.data\_netral.append(self.data[i])          elif self.target[i] == self.POSITIVE:              self.data\_positive.append(self.data[i])          else:              return None |

Kode Program 5.34 Persiapan Data

Penjelasan Kode Program 5.34:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method prepare\_data bersifat private |
| Baris 2 – 4 | Mendefinisikan data\_negative, data\_netral, dan data\_positive sebagai list |
| Baris 6 - 14 | Melakukan perulangan i hingga sepanjang data dan dilakukan seleksi data berdasarkan target dan dimasukkan kedalam variabel terkait. |

Setelah tahapan persiapan data, tahapan selanjutnya adalah tahapan *get data sequence* yang ditunjukkan pada Kode Program 5.35.

|  |  |
| --- | --- |
| Get Data Sequence | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108 | def get\_data\_sequence(self):      data\_test\_size = len(self.data\_negative) / self.fold      self.test\_size = len(self.data\_negative) \* self.test\_size      data\_test\_start\_index = 0      data\_test\_end\_index = self.test\_size      data\_train = []      data\_test = []      for i in range(self.fold):          data\_neg\_test = []          data\_net\_test = []          data\_pos\_test = []          data\_neg\_train = []          data\_net\_train = []          data\_pos\_train = []            check = False          for j in range(len(self.data\_negative)):              if j >= data\_test\_start\_index and j < data\_test\_end\_index:                  temp = []                  temp.append(self.data\_negative[j])                  temp.append(self.NEGATIVE)                  data\_neg\_test.append(temp)                  temp = []                  temp.append(self.data\_netral[j])                  temp.append(self.NETRAL)                  data\_net\_test.append(temp)                  temp = []                  temp.append(self.data\_positive[j])                  temp.append(self.POSITIVE)                  data\_pos\_test.append(temp)                  if j == 99 and data\_test\_start\_index == 90:                      for k in range(10):                          temp = []                          temp.append(self.data\_negative[k])                          temp.append(self.NEGATIVE)                          data\_neg\_test.append(temp)                          temp = []                          temp.append(self.data\_netral[k])                          temp.append(self.NETRAL)                          data\_net\_test.append(temp)                          temp = []                          temp.append(self.data\_positive[k])                          temp.append(self.POSITIVE)                          data\_pos\_test.append(temp)              else:                  if i!=9:                      temp = []                      temp.append(self.data\_negative[j])                      temp.append(self.NEGATIVE)                      data\_neg\_train.append(temp)                      temp = []                      temp.append(self.data\_netral[j])                      temp.append(self.NETRAL)                      data\_net\_train.append(temp)                      temp = []                      temp.append(self.data\_positive[j])                      temp.append(self.POSITIVE)                      data\_pos\_train.append(temp)                  else:                      if j > 9:                          temp = []                          temp.append(self.data\_negative[j])                          temp.append(self.NEGATIVE)                          data\_neg\_train.append(temp)                          temp = []                          temp.append(self.data\_netral[j])                          temp.append(self.NETRAL)                          data\_net\_train.append(temp)                          temp = []                          temp.append(self.data\_positive[j])                          temp.append(self.POSITIVE)                          data\_pos\_train.append(temp)          data\_combine\_test = data\_neg\_test + data\_net\_test + data\_pos\_test          data\_combine\_train = data\_neg\_train + data\_net\_train + data\_pos\_train          data\_combine\_test\_tweet = [data[0] for data in data\_combine\_test]          data\_combine\_test\_target = [data[1] for data in data\_combine\_test]          data\_combine\_train\_tweet = [data[0] for data in data\_combine\_train]          data\_combine\_train\_target = [data[1]                                          for data in data\_combine\_train]          data\_dict\_test = {}          data\_dict\_train = {}          data\_dict\_test["tweet"] = data\_combine\_test\_tweet          data\_dict\_test["target"] = data\_combine\_test\_target          data\_dict\_train["tweet"] = data\_combine\_train\_tweet          data\_dict\_train["target"] = data\_combine\_train\_target          data\_train.append(data\_dict\_train)          data\_test.append(data\_dict\_test)          data\_test\_start\_index += data\_test\_size          data\_test\_end\_index += data\_test\_size      return data\_train, data\_test |

Kode Program 5.35 Get Data Sequence

Penjelasan Kode Program 5.34:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_data\_sequence |
| Baris 2 – 6 | Menghitung ukuran data test, dan menentukan index awal serta akhir index data test |
| Baris 8 – 9 | Mendefinisikan data\_train dan data\_test sebagai list |
| Baris 10 | Melakukan perulangan i hingga sepanjang fold |
| Baris 11 – 18 | Mendefinisikan variabel data test dan data train tiap kelas, check didefinisikan false |
| Baris 20 | Melakukan perulangan j hingga sepanjang data\_negative |
| Baris 21 – 34 | Melakukan seleksi jika index j berada didalam jangka start dan end index maka akan dimasukkan kedalam data\_test masing-masing kelasnya |
| Baris 35 - 48 | Jika index j sama dengan 99 dan start 90 melakukan perulangan k hingga 10 untuk memasukkan ke dalam data\_test masing-masing kelasnya |
| Baris 49 - 76 | Jika tidak, jika i tidak sama dengan 9 maka akan dimasukkan kedalam data\_train masing-masing kelasnya, selain itu jika j lebih dari 9 maka akan dimasukkan kedalam data\_train masing-masing kelasnya. |
| Baris 78 - 81 | Menggabungkan data\_test tiap kelas, dan data\_train tiap kelas |
| Baris 83 - 92 | Memisahkan tweet dengan target dari data test dan data train yang sudah di combine |
| Baris 94 - 101 | Membuat dictionary test dan train dengan kata key tweet untuk data\_combine\_test\_tweet dan key target untuk data\_combine\_test\_target dan dictionary train dengan kata key tweet untuk data\_combine\_train\_tweet dan key target untuk data\_combine\_train\_target |
| Baris 103 – 104 | Dictionary train dan test dimasukkan kedalam data\_train dan data\_test |
| Baris 105 - 106 | Start index dan end index ditambahkan data\_test\_size |
| Baris 108 | Mengembalikan data\_train dan data\_test |

## Implementasi Confusion Matrix

Pada bagian implementasi confusion matrix ini akan diawali dengan tahapan *create confusion matrix* seperti yang ditunjukkan pada Kode Program 5.36.

|  |  |
| --- | --- |
| *Create Confusion Matrix* | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | def create\_confusion\_matrix(self,actual,predicted):      self.cm = pd.DataFrame(np.zeros((3, 3), dtype=int), index=['Actually Negatif', 'Actually Netral', 'Actually Positif'], columns=[                              'Predicted Negatif', 'Predicted Netral', 'Predicted Positif'])      for i in range(len(actual)):          if actual[i] == self.NEGATIVE:              if predicted[i] == self.NEGATIVE:                  self.cm.loc["Actually " + self.NEGATIVE,                              "Predicted " + self.NEGATIVE] += 1              elif predicted[i] == self.NETRAL:                  self.cm.loc["Actually " + self.NEGATIVE,                              "Predicted " + self.NETRAL] += 1              elif predicted[i] == self.POSITIVE:                  self.cm.loc["Actually " + self.NEGATIVE,                              "Predicted " + self.POSITIVE] += 1          elif actual[i] == self.NETRAL:              if predicted[i] == self.NEGATIVE:                  self.cm.loc["Actually " + self.NETRAL,                              "Predicted " + self.NEGATIVE] += 1              elif predicted[i] == self.NETRAL:                  self.cm.loc["Actually " + self.NETRAL,                              "Predicted " + self.NETRAL] += 1              elif predicted[i] == self.POSITIVE:                  self.cm.loc["Actually " + self.NETRAL,                              "Predicted " + self.POSITIVE] += 1          elif actual[i] == self.POSITIVE:              if predicted[i] == self.NEGATIVE:                  self.cm.loc["Actually " + self.POSITIVE,                              "Predicted " + self.NEGATIVE] += 1              elif predicted[i] == self.NETRAL:                  self.cm.loc["Actually " + self.POSITIVE,                              "Predicted " + self.NETRAL] += 1              elif predicted[i] == self.POSITIVE:                  self.cm.loc["Actually " + self.POSITIVE,                              "Predicted " + self.POSITIVE] += 1 |

Kode Program 5.36 *Create Confusion Matrix*

Penjelasan Kode Program 5.36:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method create\_confusion\_matrix dengan parameter actual dan predicted |
| Baris 2 – 6 | Membuat DataFrame confusion matrix dengan index sebagai actual dan kolom sebagai predicted tiap kelas dan diinisialisasi dengan 0 |
| Baris 7 – 8 | Melakukan perulangan i sepanjang actual, jika actual index i negatif |
| Baris 9 - 12 | Jika predicted negatif maka dataframe dengan index actual negatif dan predicted negatif ditambah 1 |
| Baris 13 – 15 | Jika predicted netral maka dataframe dengan index actual negatif dan predicted netral ditambah 1 |
| Baris 16 – 19 | Jika predicted positif maka dataframe dengan index actual negatif dan predicted positif ditambah 1 |
| Baris 20 | Selain itu jika actual index i sama dengan netral |
| Baris 21 – 24 | Jika predicted negatif maka dataframe dengan index actual netral dan predicted negatif ditambah 1 |
| Baris 25 – 27 | Jika predicted netral maka dataframe dengan index actual netral dan predicted netral ditambah 1 |
| Baris 28 – 31 | Jika predicted positif maka dataframe dengan index actual netral dan predicted positif ditambah 1 |
| Baris 32 | Selain itu jika actual index i sama dengan positif |
| Baris 33 – 36 | Jika predicted negatif maka dataframe dengan index actual positif dan predicted negatif ditambah 1 |
| Baris 37 – 39 | Jika predicted netral maka dataframe dengan index actual positif dan predicted netral ditambah 1 |
| Baris 40 - 43 | Jika predicted positif maka dataframe dengan index actual positif dan predicted positif ditambah 1 |

Setelah tahapan *create confusion matrix*, tahapan selanjutnya adalah tahapan mencari tp, tn, fp, dan fn yang ditunjukkan pada Kode Program 5.37.

|  |  |
| --- | --- |
| Find TP FN FP TN | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | def find\_tp\_fn\_fp\_tn(self):      self.tp\_negatif = self.cm.loc["Actually " +                                      self.NEGATIVE, "Predicted " + self.NEGATIVE]      self.tp\_netral = self.cm.loc["Actually " +                                      self.NETRAL, "Predicted " + self.NETRAL]      self.tp\_positif = self.cm.loc["Actually " +                                      self.POSITIVE, "Predicted " + self.POSITIVE]      temp = self.cm.copy()      temp.loc["Actually " + self.NEGATIVE, "Predicted " + self.NEGATIVE] = 0      self.fn\_negatif = sum(temp.loc["Actually " + self.NEGATIVE, :])      self.fp\_negatif = sum(temp.loc[:, "Predicted " + self.NEGATIVE])      temp = self.cm.copy()      temp.loc["Actually " + self.NETRAL, "Predicted " + self.NETRAL] = 0      self.fn\_netral = sum(temp.loc["Actually " + self.NETRAL, :])      self.fp\_netral = sum(temp.loc[:, "Predicted " + self.NETRAL])      temp = self.cm.copy()      temp.loc["Actually " + self.POSITIVE, "Predicted " + self.POSITIVE] = 0      self.fn\_positif = sum(temp.loc["Actually " + self.POSITIVE, :])      self.fp\_positif = sum(temp.loc[:, "Predicted " + self.POSITIVE])      temp = self.cm.copy()      temp = temp.drop("Actually " + self.NEGATIVE, axis = 0).drop("Predicted " + self.NEGATIVE, axis = 1)      self.tn\_negatif = sum(temp.sum())        temp = self.cm.copy()      temp = temp.drop("Actually " + self.NETRAL, axis = 0).drop("Predicted " + self.NETRAL, axis = 1)      self.tn\_netral = sum(temp.sum())        temp = self.cm.copy()      temp = temp.drop("Actually " + self.POSITIVE, axis = 0).drop("Predicted " + self.POSITIVE, axis = 1)      self.tn\_positif = sum(temp.sum()) |

Kode Program 5.37 Find TP, FP, FN, TN

Penjelasan Kode Program 5.37:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method find\_tp\_fn\_fp\_tn |
| Baris 2 – 10 | Menghitung tp tiap kelas dengan mengambil confusion matrix dengan index actually kelas tersebut dan predicted kelas tersebut |
| Baris 12 - 34 | Menghitung fn dan fp tiap kelas dengan diawali mengubah nilai tp menjadi 0 lalu untutk fn dilakukan penjumlahan actually kelas tersebut dengan semua kolom, lalu untuk fp dilakukan penjumlahan semua index dengan hanya kolom predicted kelas tersebut. |
| Baris 36 - 49 | Menghitung tn tiap kelas dengan diawali menghapus index actually kelas tersebut dan menghapus kolom predicted kelas tersebut lalu sisa confusion matrix yang ada dijumlahkan |

Setelah tahapan mencari tp, fn, fp, dan tn, tahapan selanjutnya adalah tahapan mencari akurasi yang ditunjukkan pada Kode Program 5.38.

|  |  |
| --- | --- |
| Get Accuracy | |
| 1  2  3  4 | def get\_accuracy(self):      acc = float(self.tp\_negatif + self.tp\_netral + self.tp\_positif) / len(self.actual)      return acc |

Kode Program 5.38 *Get Accuracy*

Penjelasan Kode Program 5.38:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_accuracy |
| Baris 2 – 4 | Menghitung hasil dari penjumlahan tp tiap kelas lalu dibagi sepanjang actual dan mengembalikan nilai nya |

Setelah tahapan mencari akurasi, tahapan selanjutnya adalah tahapan mencari akurasi tiap kelasnya yang ditunjukkan pada Kode Program 5.39.

|  |  |
| --- | --- |
| Get Accuracy Each Clas | |
| 1  2  3  4  5  6  7  8  9  10  11  12 | def get\_accuracy\_each\_class(self):      accuracy\_each\_class = []      accuracy\_each\_class.append((self.tn\_negatif + self.tp\_negatif)/(self.tn\_negatif + self.tp\_negatif+ self.fn\_negatif + self.fp\_negatif))      accuracy\_each\_class.append((self.tn\_netral + self.tp\_netral)/(self.tn\_netral + self.tp\_netral+ self.fn\_netral + self.fp\_netral))      accuracy\_each\_class.append((self.tn\_positif + self.tp\_positif)/(self.tn\_positif + self.tp\_positif+ self.fn\_positif + self.fp\_positif))      return accuracy\_each\_class |

Kode Program 5.39 *Get Accuracy Each Class*

Penjelasan Kode Program 5.39:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_accuracy\_each\_class |
| Baris 2 – 11 | Menghitung hasil akurasi setiap kelas dengan cara menambahkan tn kelas tersebut dan tp kelas tersebut lalu hasilnya dibagi dengan hasil penjumlahan dari tp, fn, fp, dan tn kelas tersebut |
| Baris 12 | Mengembalikan accuracy\_each\_class |

Setelah tahapan mencari akurasi tiap kelas, tahapan selanjutnya adalah tahapan mencari *precision* tiap kelasnya yang ditunjukkan pada Kode Program 5.40.

|  |  |
| --- | --- |
| Get Precision Each Clas | |
| 1  2  3  4  5  6  7  8  9 | def get\_precision\_each\_class(self):      precision\_each\_class = []      precision\_each\_class.append((self.tp\_negatif)/(self.tp\_negatif+ self.fp\_negatif))      precision\_each\_class.append((self.tp\_netral)/(self.tp\_netral+ self.fp\_netral))      precision\_each\_class.append((self.tp\_positif)/(self.tp\_positif+ self.fp\_positif))      return precision\_each\_class |

Kode Program 5.40 *Get Precision Each Class*

Penjelasan Kode Program 5.40:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_precision\_each\_class |
| Baris 2 – 8 | Menghitung hasil *precision* setiap kelas dengan cara tp kelas tersebut dibagi dengan hasil penjumlahan dari tp dan fp kelas tersebut |
| Baris 9 | Mengembalikan precision\_each\_class |

Setelah tahapan mencari *precision* tiap kelas, tahapan selanjutnya adalah tahapan mencari *recall* tiap kelasnya yang ditunjukkan pada Kode Program 5.41.

|  |  |
| --- | --- |
| Get Recall Each Class | |
| 1  2  3  4  5  6  7  8  9 | def get\_recall\_each\_class(self):      recall\_each\_class = []      recall\_each\_class.append((self.tp\_negatif)/(self.tp\_negatif+ self.fn\_negatif))      recall\_each\_class.append((self.tp\_netral)/(self.tp\_netral+ self.fn\_netral))      recall\_each\_class.append((self.tp\_positif)/(self.tp\_positif+ self.fn\_positif))      return recall\_each\_class |

Kode Program 5.41 *Get Recall Each Class*

Penjelasan Kode Program 5.41:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_recall\_each\_class |
| Baris 2 – 8 | Menghitung hasil *recall* setiap kelas dengan cara tp kelas tersebut dibagi dengan hasil penjumlahan dari tp dan fn kelas tersebut |
| Baris 9 | Mengembalikan recall\_each\_class |

Setelah tahapan mencari *recall* tiap kelas, tahapan selanjutnya adalah tahapan mencari *fmeasure* tiap kelasnya yang ditunjukkan pada Kode Program 5.42.

|  |  |
| --- | --- |
| Get F-Measure Each Clas | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | def get\_fmeasure\_each\_class(self):      precision\_each\_class = self.get\_precision\_each\_class()      recall\_each\_class = self.get\_recall\_each\_class()      fmeasure\_each\_class = []      fmeasure\_each\_class.append((2 \* precision\_each\_class[0] \* recall\_each\_class[0])/(precision\_each\_class[0]+recall\_each\_class[0]))      fmeasure\_each\_class.append((2 \* precision\_each\_class[1] \* recall\_each\_class[1])/(precision\_each\_class[1]+recall\_each\_class[1]))      fmeasure\_each\_class.append((2 \* precision\_each\_class[2] \* recall\_each\_class[2])/(precision\_each\_class[2]+recall\_each\_class[2]))      return fmeasure\_each\_class |

Kode Program 5.42 *Get F-Measure Each Class*

Penjelasan Kode Program 5.42:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method get\_fmeasure\_each\_class |
| Baris 2 – 3 | Memanggil method get\_precision\_each\_class dan get\_recall\_each\_class dan memasukannya kedalam variabel |
| Baris 4 - 13 | Menghitung fmeasure tiap kelas dengan cara hasil perkalian 2, precision, dan recall dibagi dengan hasil penjumlahan dari precision ditambah recall |
| Baris 14 | Mengembalikan fmeasure\_each\_class |

Setelah tahapan mencari *fmeasure* tiap kelas, tahapan selanjutnya adalah tahapan membuat method *score* yang berfungsi untuk memanggil method-method yang sudah dibaut sebelumnya yang ditunjukkan pada Kode Program 5.43.

|  |  |
| --- | --- |
| Score | |
| 1  2  3  4  5  6  7 | def score(self, actual, predicted):      self.actual = actual      self.create\_confusion\_matrix(actual,predicted)      self.find\_tp\_fn\_fp\_tn()      return self.get\_accuracy(),self.get\_accuracy\_each\_class(),self.get\_precision\_each\_class(),self.get\_recall\_each\_class(),self.get\_fmeasure\_each\_class() |

Kode Program 5.43 *Score*

Penjelasan Kode Program 5.43:

|  |  |
| --- | --- |
| Baris 1 | Mendefinisikan method score dengan parameter actual dan predicted |
| Baris 2 | Actual dimasukkn kedalam ke variabel actual dalam kelas |
| Baris 3 – 4 | Memanggil method create\_confusion\_matrix dengan parameter actual dan predicted, dan memanggil method find\_tp\_tn\_fp\_tn |
| Baris 5 - 7 | Mengembalikan nilai dengan memanggil method get\_accuracy, get\_accuracy\_each\_class, get\_precision\_each\_class, get\_recall\_each\_class, get\_fmeasure\_each\_class. |

## Implementasi Main

Pada bagian implementasi main ini akan dilakukan pemanggilan metode dan kelas-kelas yang sudah dibuat sebelumnya dan tahap ini akan diawali dengan *import library* dan kelas yang sudah dibuat sebelumnya yang ditunjukkan pada Kode Program 5.44.

|  |  |
| --- | --- |
| Import library dan kelas | |
| 1  2  3  4  5  6  7 | import pandas as pd  from tbrs import TermBasedRandomSampling  from preprocessing import Preprocessing  from naivebayes import NBMultinomial  from weighting import Weighting  from kfold import KFold  from confusionmatrix import ConfusionMatrix |

Kode Program 5.44 Import Library Main

Penjelasan Kode Program 5.44:

|  |  |
| --- | --- |
| Baris 1 | Mengimpor library pandas |
| Baris 2 | Mengimpor kelas TermBasedRandomSampling |
| Baris 3 | Mengimpor kelas Preprocessing |
| Baris 4 | Mengimpor kelas NBMultinomial |
| Baris 5 | Mengimpor kelas Weighting |
| Baris 6 | Mengimpor kelas KFold |
| Baris 7 | Mengimpor kelas ConfusionMatrix |

Setelah tahapan mengimpor library dan kelas yang diperlukan, tahapan selanjutnya adalah tahapan utama yang memanggil kelas yang sudah diimpor sebelumnya yang ditunjukkan pada Kode Program 5.45.

|  |  |
| --- | --- |
| Main | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109 | data = pd.read\_excel(      r'Skripsi.xlsx',"Data Coding")  data\_tweet = data['Tweet']  data\_target = data['Label']  kfold = KFold(data\_tweet,data\_target,10)  data\_train, data\_test = kfold.get\_data\_sequence()  x\_array = []  y\_array = []  l\_array = []  accuracy\_per\_fold\_per\_xyl\_combination = []  accuracy\_each\_class\_per\_fold\_per\_xyl\_combination = []  accuracy\_total\_per\_xyl\_combination = []  precision\_per\_fold\_per\_xyl\_combination = []  recall\_per\_fold\_per\_xyl\_combination = []  fmeasure\_per\_fold\_per\_xyl\_combination = []  count=1  for l in range(10,60,10):      for y in range (10,60,10):          for x in range (10,60,10):              print("PERULANGAN " + str(count))              count+=1              print('X={}, Y={}, L={}'.format(x,y,l))              x\_array.append(x)              y\_array.append(y)              l\_array.append(l)              accuracy\_total\_accumulation = 0              accuracy\_per\_fold = []              precision\_per\_fold = []              recall\_per\_fold = []              fmeasure\_per\_fold = []              accuracy\_each\_class\_per\_fold = []              for i in range(len(data\_train)):                  y\_test = []                  y\_pred = []                  prepro = Preprocessing()                  cleaned\_data, terms = prepro.preprocessing(data\_train[i]["tweet"])                    tbrs = TermBasedRandomSampling(X=x, Y=y, L=l)                  stopwords = tbrs.create\_stopwords(cleaned\_data,terms)                  prepro2 = Preprocessing()                  new\_cleaned\_data, new\_terms = prepro2.remove\_stopword(cleaned\_data, stopwords)                  weight = Weighting(new\_cleaned\_data, new\_terms)                  tfidf = weight.get\_tf\_idf\_weighting()                  idf = weight.get\_idf()                  nb = NBMultinomial()                  nb.fit(new\_cleaned\_data,new\_terms,data\_train[i]["target"],stopwords,idf,tfidf)                    for j in range(len(data\_test[i]["tweet"])):                      prediction = nb.predict(data\_test[i]["tweet"][j],data\_test[i]["target"][j])                      y\_test.append(data\_test[i]["target"][j])                      y\_pred.append(prediction)                  cm = ConfusionMatrix()                  accuracy, accuracy\_each\_class, precision\_each\_class, recall\_each\_class, fmeasure\_each\_class = cm.score(y\_test, y\_pred)                    accuracy = float(accuracy\*100)                  accuracy\_per\_fold.append(accuracy)                  accuracy\_total\_accumulation+=accuracy                  accuracy\_each\_class\_per\_fold.append(accuracy\_each\_class)                  precision\_per\_fold.append(precision\_each\_class)                  recall\_per\_fold.append(recall\_each\_class)                  fmeasure\_per\_fold.append(fmeasure\_each\_class)                accuracy\_per\_fold\_per\_xyl\_combination.append(accuracy\_per\_fold)              accuracy\_each\_class\_per\_fold\_per\_xyl\_combination.append(accuracy\_each\_class\_per\_fold)              precision\_per\_fold\_per\_xyl\_combination.append(precision\_per\_fold)              recall\_per\_fold\_per\_xyl\_combination.append(recall\_per\_fold)              fmeasure\_per\_fold\_per\_xyl\_combination.append(fmeasure\_per\_fold)              accuracy\_total = float(accuracy\_total\_accumulation/len(accuracy\_per\_fold))              accuracy\_total\_per\_xyl\_combination.append(accuracy\_total)  df = pd.DataFrame({'X':x\_array,'Y':y\_array,'L':l\_array,'Accuracy per Fold':accuracy\_per\_fold\_per\_xyl\_combination,'Accuracy each class per fold':accuracy\_each\_class\_per\_fold\_per\_xyl\_combination,'Precision each class per fold':precision\_per\_fold\_per\_xyl\_combination,'Recall each class per Fold':recall\_per\_fold\_per\_xyl\_combination,'F-Measure each class per Fold':fmeasure\_per\_fold\_per\_xyl\_combination,'Accuracy':accuracy\_total\_per\_xyl\_combination})  print(df)  df.to\_excel(r'output.xlsx', index = False, header=True) |

Kode Program 5.45 *Main*

Penjelasan Kode Program 5.45:

|  |  |
| --- | --- |
| Baris 1 – 2 | Mengakses data Skripsi dengan sheet Data Coding |
| Baris 3 – 4 | Mengambil kolom Tweet dan kolom Label dan dimasukkan kedalam data\_tweet dan data\_target |
| Baris 6 – 7 | Membuat objek Kfold dengan parameter data\_tweet, data\_target, dan 10 sebagai jumlah kfold, memanggil fungsi get\_data\_sequence dan memasukkan hasilnya kedalam data\_train, dan data\_test |
| Baris 9 – 19 | Mendefinisikan x\_array, y\_array, l\_array, accuracy\_per\_fold\_per\_xyl\_combination, accuracy\_each\_class\_per\_fold\_per\_xyl\_combination, accuracy\_total\_per\_xyl\_combination, precision\_per\_fold\_per\_xyl\_combination, recall\_per\_fold\_per\_xyl\_combination, fmeasure\_per\_fold\_per\_xyl\_combination sebagai list dan count diinisialisasi dengan 1 |
| Baris 20 – 22 | Melakukan perulangan l dari range 10 hingga 60 dengan langkah 10, melakukan perulangan y dari range 10 hingga 60 dengan langkah 10, melakukan perulangan x dari range 10 hingga 60 dengan langkah 10. |
| Baris 23 – 28 | Mencetak angka perulangan, count ditambah 1, mencetak kombinasi x y l yang digunakan, dan memasukan x, y, dan l ke dalam masing-masing list x\_array, y\_array, l\_array |
| Baris 30 – 35 | Accuracy\_total\_accumulation diinisialisasi dengan 0, accuracy\_per\_fold, precision\_per\_fold, recall\_per\_fold, fmeasure\_per\_fold, accuracy\_each\_class\_per\_fold didefinisikan sebagai list |
| Baris 37 | Melakukan perulangan i hingga sepanjang data\_train |
| Baris 38 – 39 | Mendefinisikan y\_test, y\_pred sebagai list |
| Baris 41 – 43 | Membuat objek Preprocessing dan memanggil method preprocessing dengan parameter data\_train index i index tweet dan nilai kembalian akan dimasukkan ke dalam cleaned\_data dan terms |
| Baris 45 – 48 | Membuat objek TermBasedRandomSampling dengan parameter x, y, dan l, dan memanggil method create\_stopwords dengan parameter cleaned\_data, terms dan hasilnya akan dimasukkan kedalam stopwords |
| Baris 50 – 52 | Membuat objek Preprocessing ke 2 untuk menghapus stopword yang sudah dihasilkan sebelumnya dengan cara memanggil fungsi remove\_stopword dengan cleaned\_data dan terms sebagai parameter dan dimasukkan hasilnya ke dalam new\_cleaned\_data dan new\_terms |
| Baris 54 – 57 | Membuat objek Weighting dengan parameter new\_cleaned\_data dan new\_terms dan memanggil method get\_tf\_idf\_weighting dan get\_idf yang masing-masing dimasukkan kedalam tfidf dan idf |
| Baris 59 – 61 | Membuat objek NBMultinomial dan memanggil method fit dengan parameter new\_cleaned\_data, new\_terms, data\_train index i index target, stopwords, idf, dan tfidf |
| Baris 63 | Melakukan perulangan j sepanjang data\_test index i index tweet |
| Baris 64 – 65 | Memanggil method predict dari kelas NBMultinomial dengan parameter data\_test index i index tweet index ke j, data\_test index i index target index ke j |
| Baris 66 – 67 | Memasukkan target ke dalam y\_test dan hasil prediksi ke y\_pred |
| Baris 69 – 72 | Membuat objek ConfusionMatrix dan memanggil method score dengan parameter y\_test, y\_pred dan hasilnya dimasukkan ke dalam accuracy, accuracy\_each\_class, precision\_each\_class, recall\_each\_class, fmeasure\_each\_class |
| Baris 74 – 76 | Hasil acccuracy dikalikan dengan 100 dan hasilnya dimasukkan kedalam accuracy\_per\_fold dan dijumlahkan dengan accuracy\_total\_accumulation |
| Baris 77 – 83 | Accuracy\_each\_class dimasukkan ke dalam accuracy\_each\_class\_per\_fold, precision\_each\_class dimasukkan ke dalam precision\_per\_fold, recall\_each\_class dimasukkan ke dalam recall\_per\_fold, fmeasure\_each\_class dimasukkan ke dalam fmeasure\_per\_fold |
| Baris 85 – 94 | Accuracy\_per\_fold dimasukkan ke dalam accuracy\_per\_fold\_per\_xyl\_combination, accuracy\_each\_class\_per\_fold dimasukkan ke dalam accuracy\_each\_class\_per\_fold\_per\_xyl\_combination, precision\_per\_fold dimasukkan ke dalam precision\_per\_fold\_per\_xyl\_combination, recall\_per\_fold dimasukkan ke dalam recall\_per\_fold\_per\_xyl\_combination, fmeasure\_per\_fold dimasukkan ke dalam fmeasure\_per\_fold\_per\_xyl\_combination |
| Baris 95 – 98 | Menghitung akumulasi akurasi total dengan cara accuracy\_total\_accumulation dibagi dengan panjang accuracy\_per\_fold lalu hasilnya dimasukkan ke accuracy\_total\_per\_xyl\_combination |
| Baris 100 - 107 | Membuat dataframe dan memasukkan variabel x\_array, y\_array, l\_array, accuracy\_per\_fold\_per\_xyl\_combination, accuracy\_each\_class\_per\_fold\_per\_xyl\_combination, precision\_per\_fold\_per\_xyl\_combination, recall\_per\_fold\_per\_xyl\_combination, fmeasure\_per\_fold\_per\_xyl\_combination, accuracy\_total\_per\_xyl\_combination |
| Baris 108 - 109 | Mencetak dan menyimpan DataFrame dalam bentuk excel |

# DAFTAR REFERENSI

Arnani, M., 2020. *KOMPAS.* [Online]   
Available at: https://www.kompas.com/tren/read/2020/03/13/111245765/kasus-pertama-virus-corona-di-china-dilacak-hingga-17-november-2019

Ramadhan, A., Nugraheny, D. E. & Maharani, T., 2020. *KOMPAS.* [Online]   
Available at: https://nasional.kompas.com/read/2020/09/05/15204581/update-kembali-bertambah-di-atas-3000-kasus-covid-19-lewati-190000?page=all

Putsanra, D. V., 2020. *tirto.* [Online]   
Available at: https://tirto.id/apa-itu-new-normal-dan-bagaimana-penerapannya-saat-pandemi-corona-fCSg

Liu, B., 2012. *Sentiment Analysis and Opinion Mining.* Chicago: Morgan & Claypool.

Sa'rony, A., Adikara, P. P. & Wihandika, R. C., 2019. Analisis Sentimen Kebijakan Pemindahan Ibukota Republik Indonesia dengan Menggunakan Algoritme Term-Based Random Samplingdan Metode Klasifikasi Naïve Bayes. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* pp. 10086-10094.

Dila Purnama Sari, D. E., Sari, Y. A. & Furqon, M. T., 2020. Pembentukan Daftar Stopword menggunakan Zipf Law dan Pembobotan Augmented TF - Probability IDF pada Klasifikasi Dokumen Ulasan Produk. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* pp. 406-412.

Rahutomo, F. & Ririd, A. R. T. H., 2018. EVALUASI DAFTAR STOPWORD BAHASA INDONESIA. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK),* pp. 41-48.

Septian, J. A., Fahrudin, T. M. & Nugroho, A., 2019. Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor. *JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTATION.*

Devita, R. N., Herwanto, H. W. & Wibawa, A. P., 2018. ERBANDINGANKINERJAMETODE NAIVE BAYES DAN K-NEAREST NEIGHBOR UNTUK KLASIFIKASI ARTIKEL BERBAHASA INDONESIA. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK),* 5(4), pp. 427-434.

Rahman, A., Wiranto & Doewes, A., 2017. Online News Classification Using Multinomial Naive Bayes. *ITSMART: Jurnal Ilmiah Teknologi dan Informasi.*

Imtiyazi, M. A., S. & Bijaksana, M. A., 2015. Sentiment Analysis Berbahasa Indonesia Menggunakan Improved Multinomial Naive Bayes. *e-Proceeding of Engineering,* 2(2), p. 6331.

Tania, A., 2020. *Muda Kompas.* [Online]   
Available at: https://muda.kompas.id/baca/2020/05/13/perlu-kerjasama-dosen-dan-mahasiswa-dalam-kuliah-daring/  
[Diakses 22 September 2020].

Gaddam, S. H. R., 2019. *Text Preprocessing in Natural Language Processing.* [Online]   
Available at: https://towardsdatascience.com/text-preprocessing-in-natural-language-processing-using-python-6113ff5decd8

Lo, R. T.-W., He, B. & Ounis, I., 2005. *Automatically Building a Stopword List for an Information Retrieval System,* Glasgow, UK: Department of Computing Science.

Jones, S., 2004. A Statistical Interpretation of Term Specificity and Its Retrieval. *Journal Of Documentation,* 60(5), pp. 11-21.

Prabowo, D. A., Fhadli, M., Najib, M. A. & Fauzi, H. A., 2016. TF-IDF-Enhanced Genetic Algorithm Untuk Extractive Automatic Text Summarization. *Jurnal Teknologi Informasi dan Ilmu Komputer,* 3(3), pp. 208-215.

Sawla, S., 2018. *Introduction to Naive Bayes for Classification.* [Online]   
Available at: https://medium.com/@srishtisawla/introduction-to-naive-bayes-for-classification-baefefb43a2d

Narkhede, S., 2018. *Understanding Confusion Matrix.* [Online]   
Available at: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Singh, S. & Shukla, S., 2016. *Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification.* Bhimavaram, IEEE.

Neale, C., Workman, D. & Dommalapati, A., 2019. *Cross Validation: A Beginner’s Guide.* [Online]   
Available at: https://towardsdatascience.com/cross-validation-a-beginners-guide-5b8ca04962cd  
[Diakses 23 September 2020].