**PERANCANGAN SISTEM REKOMENDASI ARTIKEL ONLINE DENGAN *CONTENT BASED FILTERING***

**(Brilio.net)**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**MUHAMMAD IQBAL ROFIKURRAHMAN. NIM. 1941727019**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**JULI 2020**

HALAMAN PENGESAHAN

**PERANCANGAN SISTEM REKOMENDASI ARTIKEL ONLINE DENGAN *CONTENT BASED FILTERING***

**(Brilio.net)**

Disusun oleh:

MUHAMMAD IQBAL ROFIKURRAHMAN. NIM. 1941727019

Laporan Akhir ini telah diuji pada tanggal ... Juni 2020

Disetujui oleh:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Penguji I | : | NIP. | ........................... |
|  |  |  |  |  |
| 2. | Penguji II | : | NIP. | ........................... |
|  |  |  |  |  |
| 3. | Pembimbing I | : | Pak rudy  NIP. 197903132008121002 | ........................... |
|  |  |  |  |  |
| 4. | Pembimbing II | : | Pak yoppy  NIP. | ........................... |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Jurusan  Teknologi Informasi | Ketua Program Studi  Teknik Informatika |
| Rudy Ariyanto, S.T., M.Cs. | Imam Fahrur Rozi, S.T., M.T. |
| NIP. 19711110 199903 1 002 | |  | | --- | | NIP. 19840610 200812 1 004 | |

**PERNYATAAN**

Dengan ini saya menyatakan bahwa pada Skripsi ini tidak terdapat karya, baik seluruh maupun sebagian, yang sudah pernah diajukan untuk memperoleh gelar akademik di Perguruan Tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini serta disebutkan dalam daftar sitasi/pustaka.

|  |  |
| --- | --- |
|  | Malang, … Juni 2020  Yuanita Hendra C. |

**ABSTRAK**

**Hendra C., Yuanita**. “Klasifikasi Kalimat Mengandung Perundungan Bentuk Tubuh (*Body Shaming*) Pada *Microblogging* Menggunakan Metode *Support Vector Machine*”. **Pembimbing: (1) Arief Prasetyo, S.Kom., M.Kom., (2) Rakhmat Arianto, S.ST., M.Kom.**

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2020.**

Klasifikasi kalimat digunakan untuk mengelompokkan opini terhadap *tweet* atau cuitan yang terdapat pada *Microblogging* mengandung perundungan bentuk tubuh menurut [kelas](https://id.wikipedia.org/wiki/Kelas" \o "Kelas)-kelas. Umumnya, bentuk dan ukuran tubuh seseorang yang dijadikan sasaran berkomentar atau lebih dikenal dengan perundungan bentuk tubuh. Perundungan bentuk tubuh merupakan perbuatan mencela orang lain karena penampilan fisiknya. Dengan pesatnya perkembangan web atau media sosial, semakin banyak orang menuliskan opini mereka tentang berita yang belum tentu benar atau tidak. Salah satu sosial media yang memungkinkan penggunanya untuk mengirim dan menerima pesan berbasis teks yaitu Twitter. Namun, dari manfaat pengguna Twitter masih banyak pengguna yang belum memahami etika-etika dalam bersosialisasi pada dunia maya. Permasalahan tersebut terjadi dengan adanya tindakan perundungan siber. Klasifikasi kalimat perundungan bentuk tubuh diharapkan dapat membantu pemerintah dalam memberikan perhatian penuh terhadap pelaku perundungan bentuk tubuh. Algoritma yang akan digunakan pada penelitian ini ialah *Support Vector Machine. Dataset* berjumlah 1.000 *tweet* yang didapatkan dari *tweet* mengenai tokoh yang menjadi sorotan dalam periode tertentu. Hasil klasifikasi *tweet* terbagi menjadi 3, yaitu positif perundungan bentuk tubuh, negatif perundungan bentuk tubuh atau netral. Evaluasi menentukan tingkat akurasi dari algoritma *Support Vector Machine.*

**Kata Kunci :** Klasifikasi Kalimat, Parundungan Bentuk Tubuh, *Support Vector Machine.*

***ABSTRACT***

***Hendra C., Yuanita****. “Sentence Classification Containing Body Shape Harassment in Microblogging Using the Support Vector Machine Method.”.* ***Counseling Lecturer: (1) Arief Prasetyo, S.Kom., M.Kom., (2) Rakhmat Arianto****, S.ST.,M.Kom.*

***Thesis, Informatics Management Study Program, Department of Information Technology, State Polytechnic of Malang, 2020.***

*Sentence classifications are used to classify opinions on tweets or tweets contained in Microblogging containing harassment of body shapes according to classes. Generally, the shape and size of the body of a person being targeted is commented on or better known as harassment of body shape. Harassment of bodily form is an act of reproaching another person for his physical appearance. With the rapid development of the web or social media, more and more people are writing their opinions about news that is not necessarily true or not. One of the social media that allows users to send and receive text-based messages is Twitter. However, from the benefits of Twitter users there are still many users who do not understand the ethics of socializing in cyberspace. These problems occur because of cyber harassment. It is hoped that the clasification of sentence on the form of body harassment can help the government in giving full attention to the perpetrators of the harassment of body shape. The algorithm that will be used in this research is Support Vector Machine. Dataset totaling 1,000 tweets obtained from tweets about the characters in the spotlight in a certain period. The results of the classification of tweets are divided into 3, namely positive body shape harassment, negative body shape harassment or neutral. Evaluation determines the accuracy of the Support Vector Machine algorithm.*

***Keywords:*** *Sentence Classification, Harassment of Body Shape, Support Vector Machine*

**KATA PENGANTAR**

Puji Syukur kami panjatkan kehadirat Allah SWT/Tuhan YME atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul “PERANCANGAN SISTEM REKOMENDASI ARTIKEL ONLINE DENGAN *CONTENT BASED FILTERING*”. Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari bahwasannya dengan tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi
2. Bapak Imam Fahrur Rozi, ST., MT., selaku ketua program studi Manajemen Informatika
3. Bapak Imam Arief Prasetyo, S.Kom., M.Kom., selaku dosen pembimbing I
4. Bapak Rakhmat Arianto, S.ST., M.Kom., selaku dosen pembimbing II
5. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, … Juni 2020

Penulis

**DAFTAR ISI**

Halaman

[**BAB I. PENDAHULUAN** 1](#_Toc39694438)

[**1.1** **Latar Belakang** 1](#_Toc39694439)

[**1.2** **Rumusan Masalah** 2](#_Toc39694440)

[**1.3** **Tujuan** 2](#_Toc39694441)

[**1.4** **Batasan Masalah** 3](#_Toc39694442)

[**1.5** **Sistematika Penulisan** 3](#_Toc39694443)

[BAB II. LANDASAN TEORI 5](#_Toc39694444)

[2.1 State-of-the-Art Penelitian Terdahulu 5](#_Toc39694445)

[2.2 Twitter API 11](#_Toc39694446)

[2.3 *Body Shaming* 11](#_Toc39694447)

[2.4 *Text Mining* 11](#_Toc39694448)

[2.5 *Text Preprocessing* 12](#_Toc39694449)

[2.6 Pembobotan TF-IDF 12](#_Toc39694450)

[2.7 *Support Vector Machine* 13](#_Toc39694451)

[2.7.1 Kernel Trick dan non linear SVM 14](#_Toc39694452)

[BAB III. METODELOGI PENELITIAN 17](#_Toc39694453)

[3.1 Data 17](#_Toc39694454)

[3.2 Metode Pengambilan Data 18](#_Toc39694455)

[3.3 Metode Pengolahan Data 18](#_Toc39694456)

[3.4 Metode Pengembangan Perangkat Lunak 19](#_Toc39694457)

[3.5 Metode Pengujian Data 20](#_Toc39694458)

[BAB IV. ANALISIS DAN PERANCANGAN SISTEM 21](#_Toc39694459)

[4.1 Analisis Permasalahan 21](#_Toc39694460)

[4.2 Analisis Sistem 21](#_Toc39694461)

[4.3 Analisis Data Masukan 23](#_Toc39694462)

[4.4 Analisis Algoritma 30](#_Toc39694463)

[4.4.1 Klasifikasi dengan algoritma *Support Vector Machine* 30](#_Toc39694464)

[4.5 Analisis Kebutuhan Non Fungsional 30](#_Toc39694465)

[4.6 Analisis Kebutuhan Fungsional 30](#_Toc39694466)

[4.6.1 *Database* 31](#_Toc39694467)

[4.6.2 *Use* *Case* 31](#_Toc39694468)

[4.6.3 *Activity Diagram* 35](#_Toc39694469)

[4.6.4 *Sequence Diagram* 37](#_Toc39694470)

[BAB V. IMPLEMENTASI DAN PENGUJIAN 42](#_Toc39694471)

[5.1 Implementasi *Database* 42](#_Toc39694472)

[5.2 Implementasi Proses Klasifikasi 42](#_Toc39694473)

[5.2.1 *Load Dataset* 42](#_Toc39694474)

[5.2.2 *Preprocessing* 42](#_Toc39694475)

[5.2.3 TF IDF 42](#_Toc39694476)

[5.2.4 *Training* 42](#_Toc39694477)

[5.2.5 *Testing* 42](#_Toc39694478)

[5.2.6 Pengujian 42](#_Toc39694479)

[5.3 Implementasi *Interface* 42](#_Toc39694480)

[5.4 Pengujian Fungsional 42](#_Toc39694481)

[5.5 Pengujian Akurasi 42](#_Toc39694482)

[BAB VI. HASIL DAN PEMBAHASAN 43](#_Toc39694483)

**DAFTAR GAMBAR**

[Gambar 2. 1 Hyperplane terbaik yang memisahkan kedua kelas -1 dan +1 13](#_Toc39521175)

[Gambar 2. 2 Hyperplane terbentuk di antara kelas -1 dan +1 14](#_Toc39521176)

[Gambar 2. 3 Pemetaan input space berdimensi dua dengan pemetaan ke dimensi tinggi. 15](#_Toc39521177)

[Gambar 3. 1 Diagram Alir Pengolahan Data .........................................................19](#_Toc39521410)

[Gambar 3. 2 Diagram Alur Waterfall 19](#_Toc39521411)

[Gambar 4. 1 Flowchart Sistem Klasifikasi Kalimat Mengandung Kalimat Perundungan Bentuk Tubuh ............................................................................................................................... 22](#_Toc39619424)

[Gambar 4. 2 Sub Proses Sistem 23](#_Toc39619425)

[Gambar 4. 3 Tahap Preprocessing 24](#_Toc39619426)

[Gambar 4. 4 Flowchart Tokenizing 25](#_Toc39619427)

[Gambar 4. 5 Flowchart Filtering 25](#_Toc39619428)

[Gambar 4. 6 Flowchart Stemming 26](#_Toc39619429)

[Gambar 4. 7 Use Case Diagram 32](#_Toc39619430)

[Gambar 4. 8 Activity Diagram 36](#_Toc39619431)

[Gambar 4. 9 Sequence Diagram Load Dataset 37](#_Toc39619432)

[Gambar 4. 10 Sequence Diagram Preprocessing 37](#_Toc39619433)

[Gambar 4. 11 Sequence Diagram TF IDF 37](#_Toc39619434)

[Gambar 4. 12 Sequence Diagram Training Dataset 38](#_Toc39619435)

[Gambar 4. 13 Sequence Diagram Testing Dataset 38](#_Toc39619436)

[Gambar 4. 14 Sequence Diagram Pengujian 39](#_Toc39619437)

**DAFTAR TABEL**

[Tabel 2. 1 State-of-the-Art Penelitian Terdahulu 6](#_Toc39521646)

[Tabel 2. 2 Kernel yang Umum Digunakan 16](#_Toc39521647)

[Tabel 3. 1 Data Tweet ............................................................................................17](#_Toc39521681)

[Tabel 4. 1 Contoh Proses Processing ……………………………………………………….……26](#_Toc39694977)

[Tabel 4. 2 Perbedaan Tweet Sebelum dan Sesudah Preprocessing 27](#_Toc39694978)

[Tabel 4. 3 Contoh tweet untuk perhitungan TFIDF 27](#_Toc39694979)

[Tabel 4. 4 Daftar Kata 28](#_Toc39694980)

[Tabel 4. 5 Term Frequency 28](#_Toc39694981)

[Tabel 4. 6 Document Frequency 28](#_Toc39694982)

[Tabel 4. 7 Invers Document Frequency 29](#_Toc39694983)

[Tabel 4. 8 Perhitungan TF-IDF 29](#_Toc39694984)

[Tabel 4. 9 Kebutuhan Perangkat Lunak 30](#_Toc39694985)

[Tabel 4. 10 Kebutuhan Perangkat Keras 30](#_Toc39694986)

[Tabel 4. 11 Atribut Tabel Dataset 31](#_Toc39694987)

[Tabel 4. 12 Atribut Tabel Terms 31](#_Toc39694988)

[Tabel 4. 13 Atribut Tabel Term\_Frequency 31](#_Toc39694989)

[Tabel 4. 14 Deskripsi Use Case 32](#_Toc39694990)

[Tabel 4. 15 Skenario Use Case Melihat Dataset 33](#_Toc39694991)

[Tabel 4. 16 Skenario Use Case Melakukan Preprocessing 33](#_Toc39694992)

[Tabel 4. 17 Skenario Use Case Melakukan TF-IDF 33](#_Toc39694993)

[Tabel 4. 18 Skenario Use Case Melakukan Training Dataset 34](#_Toc39694994)

[Tabel 4. 19 Skenario Use Case Testing Dataset 34](#_Toc39694995)

[Tabel 4. 20 Skenario Use Case Melakukan Pengujian 34](#_Toc39694996)

**DAFTAR LAMPIRAN**

**BAB I. PENDAHULUAN**

* 1. **Latar Belakang**

Artikel merupakan karangan tertulis yang panjangnya tidak dapat ditentukan, dimana tujuannya untuk menyampaikan gagasan maupun fakta dengan maksud meyakinkan, mendidik, ataupun menghibur [1]. Di era modern, artikel sering ditulis di berbagai media online seperti website. Artikel online dapat dibagi menjadi berbagai kategori sesuai pembahasannya. Dalam sebuah kategori pastinya akan memiliki jumlah artikel yang sangat banyak. Sehingga pengkategorian artikel, tidak dapat digunakan sebagai acuan sarana untuk mempermudah pengguna dalam memilih artikel yang tepat. Sistem rekomendasi hadir sebagai sarana agar pengguna dapat lebih mudah memilih artikel yang relevan dengan artikel lain sesuai yang ingin pengguna akses, tanpa perlu harus melakukan pencarian di kolom pencarian website.

Sistem rekomendasi merupakan sistem yang sering diterapkan oleh *website* penyedia artikel [2], karena sistem rekomendasi dapat ini memberikan dampak yang baik terhadap statistik *website*. Dalam sistem ini tingkat relevansi terhadap artikel yang direkomendasikan harus diperhatikan, agar para pengguna dapat memberikan respon yang baik pada *website* dan artikel yang sedang diakses. Pada sistem rekomendasi terdapat berbagai teknik dalam penerapannya, yaitu terdiri dari *content based filtering, collaborative filtering,* dan *hybrid filtering*.

Brilio.net merupakan sebuah perusahaan media penyedia artikel yang sudah cukup dikenal di Indonesia. *website* ini menyediakan artikel tentang gaya hidup, berita *viral*, teknologi dan artikel-artikel jenaka yang sering diperbincangkan oleh banyak netizen di Indonesia, dan memiliki jumlah pengunjung yang banyak dengan rata-rata sepuluh ribu pengunjung setiap harinya, brilio.net belum menerapkan sistem rekomendasi (*content based filtering*) yang berdasarkan sesuai dengan kemiripan artikel yang sedang diakses oleh pengguna, hal ini dapat mengurangi ketertarikan pengguna untuk membaca artikel lainnya, karena artikel yang direkomendasikan hanya berdasarkan artikel dengan jumlah akses terbanyak saja, sehingga tidak ada peningkatan dalam statistik *website* dibagian lama pengguna mengakses.

Pada penelitian sebelumnya telah dibuat sistem rekomendasi artikel yang serupa dengan menggunakan metode yang berbeda yaitu metode *cosine similarity* dengan hasil menunjukkan bahwa artikel yang direkomendasikan adalah relevan [3], kelemahannya sistem ini beroperasi kurang cepat, dikarenakan alur sistem yang akan melakukan operasi pengecekan kemiripan artikel setiap  pengguna mengakses, sehingga semakin banyak pengguna dan artikel yang diakses, semakin lama juga proses perhitungannya.

Brilio.net membutuhkan sebuah sistem rekomendasi artikel online dengan proses perhitungan yang lebih cepat dan dapat menampilkan artikel yang relevan sesuai dengan artikel yang diakses oleh pengguna. Sebuah artikel perbandingan menyebutkan bahwa metode *dice similarity* tidak membutuhkan komputasi yang terlalu berat [4]. Oleh sebab itu dibuatlah perancangan sistem rekomendasi artikel online dengan *content based filtering*.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat diambil, yaitu :

1. Bagaimana membuat sebuah sistem rekomendasi artikel yang relevan berdasarkan artikel yang sedang diakses oleh pengguna ?
2. Bagaimana mempercepat proses pemberian rekomendasi artikel kepada pengguna, tanpa terpengaruh dari jumlah artikel dan pengguna yang mengakses ?
   1. **Tujuan**

Berdasarkan rumusan masalah yang telah dijelaskan, maka tujuan dari penelitian ini yaitu :

1. Membuat prototipe sistem dalam memberikan rekomendasi artikel yang relevan sesuai dengan artikel yang sedang diakses oleh pengguna.
2. Mempercepat proses perhitungan dalam memberikan rekomendasi artikel tanpa terpengaruh dengan banyaknya artikel yang sedang diakses.
   1. **Batasan Masalah**

Agar skripsi penulis yang berjudul “Perancangan Sistem Rekomendasi Artikel Online dengan *Content Based Filtering*” dapat berjalan sesuai dengan rencana dan tujuan awal, maka penulis memberikan batasan-batasan masalah yaitu:

1. Sistem hanya memberikan rekomendasi artikel dari website Brilio.net saja.
2. Sistem ini memberikan rekomendasi sesuai kemiripan artikel yang sedang diakses, tanpa memperhatikan perilaku pengguna sebelumnya.
3. Artikel yang direkomendasikan adalah artikel yang telah diambil dan disimpan ke penyimpanan lokal sistem sebelumnya.
4. Sistem ini memberikan rekomendasi tidak bergantung kepada kategori artikel.
   1. **Sistematika Penulisan**

Uraian dalam laporan Skripsi penulis menyusun dengan Sistematika penulisan sebagai berikut:

**BAB I : PENDAHULUAN**

Pendahuluan berisikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat.

**BAB II : LANDASAN TEORI**

Landasan teori berisikan tentang tinjauan pustaka dari aplikasi yang penulis buat.

**BAB III : METODELOGI PENELITIAN**

Berisi mengenai tahapan yang dilakukan untuk menyelesaikan masalah pada tugas akhir yang bersumber dari proses dalam perencanaa tugas akhir. Metode penelitian berisi urauian tentang metode pengmbilan data, metode pengembangan sistem, fase-fase pengembangan sistem.

**BAB IV : ANALISIS DAN PERANCANGAN**

Analisa dan Perancangan berisikan tentang analisa sistem aplikasi dan perancangannya.

**BAB V : IMPLEMENTASI DAN PENGUJIAN**

Implementasi dan pengujian berisikan penerapan/implementasi dan pengujian dari aplikasi yang telah penulis buat.

**BAB VI : HASIL DAN PEMBAHASAN**

Hasil dan Pembahasan berisikan tentang pembahasan serta analisa dari hasil proses tersebut.

**BAB VII : KESIMPULAN dan SARAN**

Kesimpulan berisikan tentang kesimpulan dari penelitian dan saran.

# BAB II. LANDASAN TEORI

## 2.1 State-of-the-Art Penelitian Terdahulu

Penelitian yang dilakukan oleh (Luqyana, Cholissodin, & Perdana, 2018) adalah melakukan Analisis Sentimen *Cyberbullying* pada Komentar Instagram dengan Metode Klasifikasi *Support Vector Machine* dengan nilai *degree kernel polynomial* sebesar 2, nilai *learning rate* sebesar 0,0001, dan jumlah iterasi maksimum yang digunakan adalah 200 kali dengan akurasi sebesar 90%. Penelitian yang dilakukan oleh (Lingga, 2019) adalah melakukan Pemodelan Deteksi *Body Shaming* Di Media Sosial Twitter Menggunakan Algoritma *Naive Bayes* dengan dilakukan enam tahap pemrosesan data yaitu *tokenize, transform cases, stemming, filter stopwords, filter tokens by length* dan *generate n-Grams* dengan akurasi sebesar 72%.

Penelitian yang dilakukan oleh (Supartini, Sukarsa, & Srinadi, 2017) adalah melakukan Analisis Diskriminan Pada Klasifikasi Desa di Kabupaten Tabanan Menggunakan Metode *K-Fold Cross Validation* dengan teknik analisis diskriminan kuadratik dengan akurasi sebesar 76%. Penelitian yang dilakukan oleh (Siregar, Sinaga, & Arianto, 2017) adalah penentuan Dosen Penguji Skripsi Menggunakan Metode TF-IDF dan *Vector Space Model* dengan menggunakan Model pengembangan perangkat lunak CRISP-DM dengan akurasi sebesar 93,22%.

Penelitian yang dilakukan oleh (Purnamasari, Fauzi, Indriarti, & Dewi, 2018) adalah Identifikasi *Tweet Cyberbullying* pada Aplikasi Twitter menggunakan Metode *Support Vector Machine* (SVM) dan *Information Gain* (IG) sebagai Seleksi Fitur dengan akurasi sebesar 75%. Penelitian yang dilakukan oleh (Munir, Fauzi, & Perdana, 2018) adalah Implementasi Metode *Backpropagation Neural Network* berbasis *Lexicon Based Features* dan *Bag of Words* Untuk Identifikasi Ujaran Kebencian Pada Twitter dengan pembaruan fitur menggunakan *Lexicon Based Features* yang dikombinasikan dengan *Bag of Words* dengan akurasi sebesar ketika menggunakan *Lexicon Based Features* nilai rata-rata *f-measure* sebesar 0%, lebih buruk dibandingkan dengan menggunakan *Bag of Words* yang nilai rata-rata *f-measure* sebesar 76,638%, sedangkan ketika *Lexicon Based Features* dikombinasikan dengan *Bag of Words* mendapat nilai rata-rata terbaik diantara fitur sebelumnya dengan *f-measure* sebesar 78,081%.

Penelitian yang dilakukan oleh (Sidiq et al., 2019) adalah Mendeteksi *Cyberhate* pada Twitter Menggunakan *Text Classification* dan *Crowdsourced Labeling* dengan akurasi sebesar 97%. Selain itu, juga diterapkan *crowdsourced labeling* dalam pelabelan dataset yang digunakan, yang menghasilkan validitas data sebesar 98% data valid dengan tujuh data yang salah. Penelitian yang dilakukan oleh (Van Hee et al., 2018) adalah *Automatic detection of cyberbullying in social media text* dengan menggunakan *linear support vector machines* dengan akurasi sebesar 64%.

Penelitian yang dilakukan oleh (Maulina, Sagara, Komputer, & Utara, 2018) adalah melakukan Klasifikasi Artikel Hoax Menggunakan *Support Vector Machine* Linear dengan Pembobotan *Term Frequency – Inverse Document Frequency* dengan akurasi sebesar 95.8333%. Penelitian yang dilakukan oleh (Syarif, Herdiani, Astuti, & Kom, 2019) adalah Identifikasi *Cyberbullying* pada Komentar Instagram menggunakan Metode *Lexicon-Based* dan *Naïve Bayes Classifier* dengan akurasi dari metode *LexiconBased* menghasilkan akurasi sebesar 58%, presisi 52%, recall 75% dan F-score 61%. Sedangkan *naïve bayes classifier* didapatkan akurasi 97%, presisi 94%, recall 100%, dan F1-score 97%.

Tabel 2. 1 State-of-the-Art Penelitian Terdahulu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Judul | Penulis/Jurnal | Univ/Tahun | Permasalahan | Classifier | Hasil |
| 1. | Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine | Wanda Athira Luqyana, Imam Cholissodin, Rizal Setya Perdana  Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer | Universitas Brawijaya  2018 | Mengetahui sentimen dari setiap komentar.pada sosial media instagram | Support Vector Machine | Hasil akurasi 90% |
| 2. | Pemodelan Deteksi Body Shaming Di Media Sosial Twitter Menggunakan Algoritma Naive Bayes | Imelda Yohana Uli Rastra Lingga  Jurnal Teknosains: The Journal of Science and Technology | Universitas Marcu Buana  2019 | Mendeteksi body shaming pada sosial media Twitter | Naive Bayes | Hasil akurasi 72% |
| 3. | Analisis Diskriminan Pada Klasifikasi Desa di Kabupaten Tabanan Menggunakan Metode K-Fold Cross Validation | Ida Ayu Made Supartini, I Komang Gde Sukarsa, I Gusti Ayu Made Srinadi  E-Jurnal Matematika | Universitas Udayana  2017 | Estimasi model klasifikasi desa di Kabupaten Tabanan dengan teknik analisis diskriminan | K-Fold Cross Validation | Hasil akurasi 76% |
| 4. | Aplikasi Penentuan Dosen Penguji Skripsi Menggunakan Metode TF-IDF dan Vector Space Model | Riki Ruli A. Siregar, Fera Amelia Sinaga, Rakhmat Arianto  Computatio: Journal of Computer Science and Information Systems | Sekolah Tinggi Teknik PLN Jakarta  2017 | Menentukan dosen penguji skripsi | Vector Space Model | Hasil akurasi 93,22% |
| 5. | Identifikasi Tweet Cyberbullying pada Aplikasi Twitter menggunakan Metode Support Vector Machine (SVM) dan Information Gain (IG) sebagai Seleksi Fitur | Ni Made Gita Dwi Purnamasari, M. Ali Fauzi, Indriati, Liana Shinta Dewi  Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer | Universitas Brawijaya  2018 | Mengklasifikasikan tweet yang mengandung konten bullying | Support Vector Machine (SVM) | Hasil akurasi 75% |
| 6. | Implementasi Metode Backpropagation Neural Network berbasis Lexicon Based Features dan Bag of Words Untuk Identifikasi Ujaran Kebencian Pada Twitter | Muhammad Mishbahul Munir, Mochammad Ali Fauzi, Rizal Setya Perdana  Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer | Universitas Brawijaya  2018 | Mendeteksi ujaran yang bermaksud untuk menghina atau mempermalukan dan medianya bisa terdapat dimana saja, salah satunya Twitter. | Backpropagation Neural Network | Hasil akurasi 78,081%. |
| 7. | Mendeteksi Cyberhate pada Twitter Menggunakan Text Classification dan Crowdsourced Labeling | Hadi Kurniawan Sidiq, Dana Sulistyo Kusumo, Indra Lukmana Sardi  JNTETI | -  2019 | Mendeteksi cyberhate yang menyebabkan perselisihan antar kelompok pendukung kedua pasangan calon presiden. | Support Vector Machine | Hasil akurasi 97% |
| 8. | Automatic detection of cyberbullying in social media text | Cynthia Van Hee, Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, Ve ´ronique Hoste  PLOS ONE | University of Antwerp  2018 | Detection in social media text by modelling posts written by bullies, victims, and bystanders of online bullying | Linear Support Vector Machines | Hasil akurasi 64% |
| 9. | Klasifikasi Artikel Hoax Menggunakan Support Vector Machine Linear dengan Pembobotan Term Frequency – Inverse Document Frequency | Dina Maulina, Rofie Sagara  Jurnal Mantik Penusa | Universitas AMIKOM Yogyakarta  2018 | Cara merancang sistem yang dapat melakukan klasifikasi informasi dengan melewati bidang keilmuan Text Mining | Support Vector Machine Linear | Hasil akurasi 95.8333% |
| 10. | Identifikasi Cyberbullying pada Komentar Instagram menggunakan Metode Lexicon-Based dan Naïve Bayes Classifier | Rizky Dhian Syarif, Anisa Herdiani, S.T., M.T., Widi Astuti, S.T., M.Kom.  e-Proceeding of Engineering | Universitas Telkom, Bandung  2019 | Komentar yang mengandung cyberbullying atau non cyberbullying | Lexicon-Based dan Naïve Bayes | LexiconBased menghasilkan akurasi sebesar 58%.  Naïve bayes classifier didapatkan akurasi 97%, |

## 2.2 Twitter API

Twitter merupakan salah satu layanan jejaring sosial yang masuk ke dalam *Microblogging* secara singkat dalam satu paragraf dengan maksimal 280 huruf(Purnamasari et al., 2018). Twitter API adalah salah satu layanan yang disediakan oleh Twitter untuk pengembang. Twitter API dapat digunakan untuk mengembangkan aplikasi yang membutuhkan data dari Twitter.

## 2.3 *Body Shaming*

*Body shaming* atau bisa disebut dengan perundungan bentuk tubuh merupakan tindakan mengejek atau berkomentar negatif terhadap keadaan fisik atau tubuh seseorang. Umumnya, bentuk dan ukuran tubuh seseorang yang dijadikan sasaran berkomentar. *Body shaming* merupakan kriteria dari tindakan perundungan siber. Perundungan siber atau *cyberbullying* merupakan tindakan perundungan yang dilakukan pada dunia *cyber*(Luqyana et al., 2018). Misalnya wanita yang cantik adalah wanita yang berkulit putih, berhidung mancung, bertubuh langsing atau memiliki rambut hitam dan lurus. Sedangkan wanita yang berkulit gelap, berhidung pesek, bertubuh gemuk atau memiliki rambut keriting selalu dianggap tidak menarik atau jelek. Pemikiran seperti ini membawa pengaruh besar bagi setiap persepsi orang terhadap bentuk tubuhnya(Lingga, 2019).

## 2.4 *Text Mining*

*Text mining* adalah proses ekstraksi informasi dari data sumber yang belum terstruktur(Fidyawan, 2017). *Text mining* memiliki tujuan untuk memproses teks agar menjadi informasi yang diperoleh dari peramalan pola dan kecenderungan melalui pola statistik(Luqyana et al., 2018). Input untuk *text mining* adalah data yang tidak (atau kurang) terstruktur, seperti dokumen word, PDF, kutipan teks. *Text mining* digunakan untuk menangani masalah *classification, clustering, information extraction* dan *information retrival.*

## 2.5 *Text Preprocessing*

*Preprocessing text* merupakan tindakan menghilangkan karakter-karakter tertentu yang terkandung dalam dokumen, seperti koma, tanda petik dan lain-lain serta mengubah semua huruf kapital menjadi huruf kecil. Selain itu, dalam tahap *text preprocessing* ini dilakukan *tokenization*. *Text mining* dalam prakteknya mencari pola-pola tertentu, mengasosiasikan suatu bagian teks dengan yang lain berdasarkan aturan-aturan tertentu, katakata yang dapat mewakili sehingga dapat dilakukan analisa keterhubungan antar satu dengan yang lain(Siregar et al., 2017). Berikut tahapan-tahapan proses didalam *text mining*:

1. *Tokenizing*

Tahap *Tokenizing* adalah tahap pemotongan tiap kata dalam kalimat atau parsing dengan menggunakan spasi sebagai delimiter yang akan menghasilkan token berupa kata. Pada tokenizing terdapat beberapa proses yang harus dilakukan yaitu merubah semua hruf besar menjadi kecil (*text* *to* *lowercase*). Proses selanjutnya adalah penguraian, proses penguraian yang dimaksud adalah membagi *text* menjadi kumpulan kata tanpa memperhatikan keterhubungan antara kata satu dengan kata lain serta peran dan posisinya pada kalimat.

1. *Filtering*

Tahap Filtering adalah tahap penyaringan kata yang didapat dari *Tokenizing* yang dianggap tidak penting atau tidak memiliki makna dalam proses *Text* *mining* yang disebut *stopword*. *Stopword* berisi katakata umum yang sering muncul dalam sebuah dokumen dalam jumlah banyak namun tidak memiliki kaitan dengan tema tertentu. Contoh stopwords adalah “yang”, “di”, “dan”, dll.

1. *Stemming*

Tahap stemming adalah tahap mengembalikan kata-kata yang diperoleh dari hasil Filtering ke bentuk dasarnya, menghilangkan imbuhan awal (prefix) dan imbuhan akhir (sufix) sehingga didapat kata dasar.

## 2.6 Pembobotan TF-IDF

*Term Frequency-Inverse Document Frequency* (TF-IDF) adalah metode yang digunakan untuk menghitung bobotan setiap kata yang telah diekstrak. Penggunaan metode ini umumnya dilakukan untuk mengihitung kata umum yang ada pada *information retrieval*. Model pembobotan TF-IDF merupakan metode yang mengintegrasikan model *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). *Term frequency* merupakan proses untuk menghitung jumlah kemunculan term dalam satu dokumen dan *inverse* *document* *frequency* digunakan untuk menghitung term yang muncul di berbagai dokumen (komentar) yang dianggap sebagai term umum, yang dinilai tidak penting (Akbari, et al., 2012). Tahapan pembobotan dengan TF-IDF adalah:

𝑊𝑑𝑡 = 𝑡𝑓𝑑𝑡 ∗ 𝐼𝐷𝐹𝑡 (1)

Keterangan

d : Dokumen ke-d

t : Kata ke-t dari kata kunci

W : Bobot dokumen ke-d terhadap kata ke-t

tf : Banyaknya kata yang dicari pada sebuah dokumen

IDF : Inverse Document Frequency

Nilai IDF didapat dengan cara sebagai berikut.

𝐼𝐷𝐹 = 𝑙𝑜𝑔(𝐷/𝑑𝑓) (2)

Keterangan

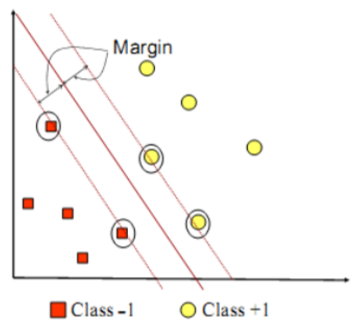
D : Total dokumen

df : Banyak dokumen yang mengandung kata kunci

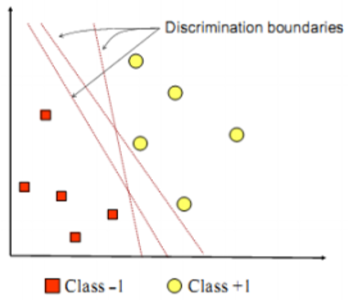
## 2.7 *Support Vector Machine*

*Support Vector Machine* pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep- konsep unggulan dalam bidang *pattern recognition*(Fidyawan, 2017). SVM adalah algoritma *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*.

Gambar 2.1 memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* : +1 dan –1. *Pattern* yang tergabung pada *class* –1 disimbolkan dengan warna merah (kotak), sedangkan pattern pada class +1, disimbolkan dengan warna kuning (lingkaran). Masalah klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut(Fidyawan, 2017).



Gambar 2. 1 Hyperplane terbaik yang memisahkan kedua kelas -1 dan +1



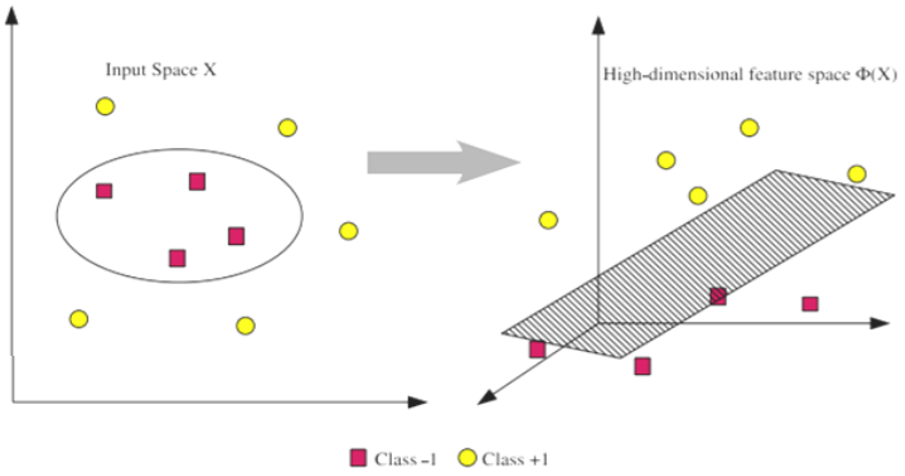
Gambar 2. 2 Hyperplane terbentuk di antara kelas -1 dan +1

(Sumber : (Fidyawan, 2017))

*Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur margin *hyperplane* tersebut. dan mencari titik maksimalnya. Margin adalah jarak antara hyperplane tersebut dengan pattern terdekat dari masing- masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support* *vector*. Garis *solid* pada gambar menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support* *vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM.

### 2.7.1 Kernel Trick dan non linear SVM

Pada umumnya masalah dalam domain dunia nyata (*real world problem*) jarang yang bersifat linear *separable*, kebanyakan bersifat non linear. Untuk menyelesaikan *problem* non linear, SVM dimodifikasi dengan memasukkan fungsi Kernel. Dalam non linear SVM, pertama- tama data x dipetakan oleh fungsi 𝜙(𝑥) ke ruang vektor yang berdimensi lebih tinggi. Pada ruang vektor yang baru ini, *hyperplane* yang memisahkan kedua class tersebut dapat dikonstruksikan. Hal ini sejalan dengan teori Cover yang menyatakan “Jika suatu transformasi bersifat non linear dan dimensi dari *feature* *space* cukup tinggi, maka data pada *input* *space* dapat dipetakan ke feature space yang baru, dimana patternpattern tersebut pada probabilitas tinggi dapat dipisahkan secara linear”.



Gambar 2. 3 Pemetaan input space berdimensi dua dengan pemetaan ke dimensi tinggi.

(Sumber :(Fidyawan, 2017))

Ilustrasi dari konsep ini dapat dilihat pada gambar. Pada gambar di atas sisi kiri diperlihatkan data pada *class* kuning dan data pada *class* merah yang berada pada *input* *space* berdimensi dua tidak dapat dipisahkan secara linear. Selanjutnya gambar menunjukkan bahwa fungsi Φ memetakan tiap data pada *input* *space* tersebut ke ruang vektor baru yang berdimensi lebih tinggi (dimensi 3), dimana kedua class dapat dipisahkan secara linear oleh sebuah *hyperplane*. Notasi matematika dari *mapping* ini adalah sebagai berikut. 𝛷 ∶ ℜ𝑑 → ℜ𝑑 𝑑 < 𝑞 (3) Pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada input space akan berjarak dekat juga pada *feature* *space*, sebaliknya dua data yang berjarak jauh pada *input* *space* akan juga berjarak jauh pada *feature* *space*.

Proses pembelajaran pada SVM dalam menemukan titik- titik *support* *vector*, hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang baru yang berdimensi lebih tinggi, yaitu : 𝛷(𝑥𝑖).𝛷(𝑥𝑗) (4)

Karena umumnya transformasi Φ ini tidak diketahui, dan sangat sulit untuk dipahami secara mudah, maka perhitungan dot product tersebut sesuai teori Mercer dapat digantikan dengan fungsi kernel yang mendefinisikan secara implicit transformasi Φ. Hal ini disebut sebagai Kernel Trick, yang dirumuskan: 𝐾(𝑥𝑖,𝑥𝑗) = 𝛷(𝑥𝑖).𝛷(𝑥𝑗) (5)

Kernel trick memberikan berbagai kemudahan, karena dalam proses pembelajaran SVM, untuk menentukan *support vector*, kita hanya cukup mengetahui fungsi kernel yang dipakai, dan tidak perlu mengetahui wujud dari fungsi non linear Φ. Berbagai jenis fungsi kernel dikenal, sebagaimana dirangkumkan pada tabel 2.1.

Tabel 2. 2 Kernel yang Umum Digunakan

|  |  |
| --- | --- |
| **Jenis Kernel** | **Definisi** |
| *Polynomial* | *K*(*xi,xj*) *=*  (6) |
| *Gaussian* *RBF* | *K*(*xi,xj*) = exp) (7) |
| *Signoid* | *K*(*xi,xj*) = tanh (8) |
| *Linear* | *K*(*xi,xj*) = (9) |

Sumber : (Fidyawan, 2017)

Selanjutnya hasil klasifikasi dari data xr diperoleh dari persamaan berikut. 𝑓(𝛷(𝑥)) = 𝑤.𝛷(𝑥) + 𝑏 (10)

= ∑ 𝛼𝑖𝑦𝑖 𝛷(𝑥).𝛷(𝑥) + 𝑏 𝑛 𝑖=1.𝑥,∈𝑆𝑉 (11)

= ∑ 𝛼𝑖𝑦𝑖 𝐾(𝑥,𝑥𝑖) + 𝑏 𝑛 𝑖=1.𝑥,∈𝑆𝑉 (12)

*Support vector* pada persamaan di atas dimaksudkan dengan subset dari training set yang terpilih sebagai *support vector,* dengan kata lain data 𝑥𝑖 yang berkorespondensi pada 𝛼𝑖 ≥ 0.

# BAB III. METODELOGI PENELITIAN

## 3.1 Data

Data yang digunakan dalam penelitian ini adalah *tweet* yang tercantum tokoh yang menjadi sorotan dalam periode tertentu pada sosial media Twitter. Berikut beberapa contoh data *tweet* yang akan diolah.

Tabel 3. 1 Data Tweet

|  |  |
| --- | --- |
| ID | *Tweet* |
| 1233516761238110000 | Dua tiga tutup botol Jangan kaya mas fatah sukanya pake tramadol   #bacotsantuy  #LucintaLuna |
| 1233425287905460000 | .  Thank you Sparta 101 Jak FM . Jangan lupa Like,comment,subscribe & Share KANG OBAY CHANNEL  http:// Youtube.com/channel/UCX5Lk jReV5BTBBZ\_GRhLNhQ Â  â€¦  ================================= #jakfm #lucintaluna â€¦ https://www. instagram.com/p/B9HdsgNluXl/ ?igshid=1wi5p60o551ii Â  â€¦ |
| 1233423629578200000 | p. @mimiperi @lucintaluna https:// twitter.com/yonghocn/statu s/1233363513520689153 Â  â€¦ |
| 1233405694021450000 | Wehhhh lubang jareee...  I hate my mind |
| 1233404637752900000 | Pen macarin lucintaluna |
| 1233379027781500000 | Ya ga salah jg sih ler.  Dia punya pisang, aku punya lubang |
| 1233378651225360000 | Aku bntu tag orngnya..  @LUCINTALUNA |
| 1233375971056780000 | Kalau bidannya lucintaluna mau gak? Kan cantik |
| 1233368948453690000 | Hahhha jang halu halu ntar kayaa lucintaluna wkwk |

## 3.2 Metode Pengambilan Data

Metode pengambilan data digunakan untuk mengumpulkan data pendukung yang dibutuhkan dalam pembuatan aplikasi. Beberapa cara yang dapat digunakan adalah melalui studi literatur, yaitu dengan mengumpulkan dan mempelajari beberapa referensi dari berbagai sumber yang berkaitan dengan judul penelitian yang dilakukan. Selain itu, untuk dataset yang digunakan diambil melalui Twitter API dengan proses *crawling* data tokoh yang menjadi sorotan dalam periode tertentu. Data yang telah terkumpul nantinya dibagi menjadi 2 yaitu data *training* dan data *testing*.

## 3.3 Metode Pengolahan Data

Data yang didapat dari *crawling* postingan *tweet* yang tercantum nama tokoh yang menjadi sorotan pada sosial media Twitter dalam periode tertentu. Data yang akan diambil berjumlah 1000 data *tweet.* Dengan pembagian 90% data latih dan 10% data uji. Data tersebut perlu dilakukan pengolahan agar menjadi data yang mudah digunakan dalam proses identifikasi.

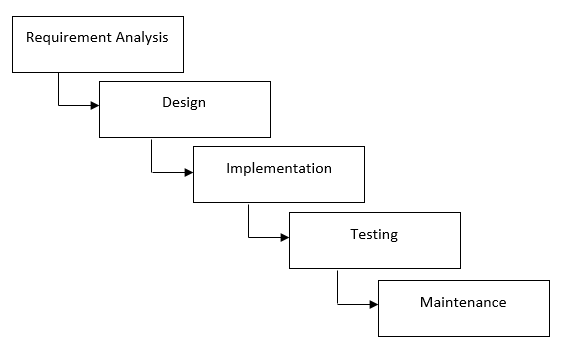
Sebelum data diolah untuk mempermudah proses identifikasi, data akan diberi label dengan kategori positif, negatif atau netral. Proses pelabelan dilakukan oleh pendapat beberapa orang dan dibantu oleh ahli psikolog untuk memvalidasi data. *Tweet* akan mengalami penyeleksian kata-kata sehingga *tweet* menjadi lebih ringkas. Beberapa komponen dari *tweet* akan dihapus untuk menyeleksi *tweet*. Proses ini dapat disebut dengan *preprocessing*. Setelah melalui *preprocessing*, data yang berupa teks akan diubah ke dalam bentuk angka melalui perhitungan *Term Frequency-Inverse Document Frequency* (TF-IDF). Nilai TF-IDF ini yang akan menjadi masukan untuk algoritma *Support Vector Machine* dalam melakukan pengklasifikasian *tweet.*



Gambar 3. 1 Diagram Alir Pengolahan Data

## 3.4 Metode Pengembangan Perangkat Lunak

Metode dalam penelitian ini menggunakan metode waterfall. Dalam metode ini terdapat 6 tahap, langkah-langkahnya adalah sebagai berikut:



Gambar 3. 2 Diagram Alur Waterfall

1. *Requirement Analysis*

Pada tahapan ini dilakukan pengumpulan bahan-bahan yang dibutuhkan untuk mengelola sistem yang akan dibuat dan didapat dari hasil observasi dan studi literatur. Seperti fungsi-fungsi yang akan digunakan, metode yang akan digunakan, kinerja sistem tersebut, dan juga output yang dihasilkan. Setelah itu, akan didefinisikan kebutuhan tentang sistem yang akan dibuat.

1. *Design*

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Mempelajari kebutuhan dari tahap sebelumnya, yang kemudian menyiapkan design dari sistem untuk membantu mendefinisikan arsitektur secara keseluruhan

1. *Implementation*

Dalam tahap ini dilakukan pembuatan aplikasi. Pembuatan perangkat lunak dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan*.*

1. *Testing*

Di tahap ini dilakukan uji coba pada aplikasi yang telah dibuat untuk mengetahui apakah perangkat lunak yang dibuat telah sesuai dengan desainnya dan apakah masih terdapat kesalahan

1. *Maintenance*

Tahap ini dilakukan jika perangkat lunak sudah jadi danmengalami kerusakan. Maka akandilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

## 3.5 Metode Pengujian Data

Pengujian yang dilakukan untuk menguji sistem dilakukan dengan 2 langkah, yaitu pengujian fungsionalitas dan pengujian performa sistem. Pengujian bertujuan untuk memastikan bahwa sistem yang dibangun telah berjalan sesuai dengan yang diharapkan.

1. Pengujian Fungsionalitas

Uji coba fungsionalitas dikatakan berhasil apabila fungsi yang ada pada sistem sesuai dengan yang diharapkan. Uji fungsionalitas yang diterapkan pada penelitian ini ialah dengan mencoba menjalankan setiap fitur yang ada dan yang disediakan pada sistem ini

1. Pengujian Akurasi Sistem

Dalam pengujian ini, dilakukan untuk mengetahui keakuratan aplikasi dalam menampilkan data hasil pencarian oleh pengguna.

# BAB IV. ANALISIS DAN PERANCANGAN SISTEM

## 4.1 Analisis Permasalahan

Pengguna akan memberikan opini kepada tokoh yang menjadi sorotan dalam periode tertentu melalui Twitter. Komentar yang diberikan beraneka ragam, ada yang berupa pujian, hujatan atau sebuah kabar berita. Dalam memberikan komentar banyak pengguna yang kurang memperhatikan etika dalam bersosialisasi pada dunia maya yang dapat mengakibatkan terjadinya tindakan perundungan siber. Umumnya bentuk dan ukuran tubuh seseorang yang akan dijadikan sasaran dalam berkomentar.

Permasalahan yang terjadi adalah mendeteksi opini yang diberikan oleh pengguna, apakah opini tersebut mengandung perundungan bentuk tubuh atau tidak. Sehingga diperlukan klasifikasi untuk memberikan kemudahan dalam mendeteksi opini yang bernilai positif perundungan bentuk tubuh, negatif perundungan bentuk tubuh atau netral.

## 4.2 Analisis Sistem

Sistem yang akan dibangun diharapkan dapat mengklasifikasikan *tweet* yang telah didapatkan dari proses *crawling* menggunakan *TweetScraper* ke dalam label positif, negatif atau netral. Sistem ini akan memiliki beberapa sub proses yang akan dilakukan untuk menghasilkan satu proses yang utuh.

1. Tahap Awal

Tahapan awal yaitu *input* dataset yang diperoleh dari proses *crawling* data *tweet* dari Twitter yang berkaitan dengan tokoh yang menjadi sorotan dalam periode tertentu yang telah diberi label kategori *tweet* positif, negatif atau netral. Selanjutnya proses pre-*processing* atau praproses pada dataset. Dalam tahap *pre-processing* meliputi *itokenizing, filtering* dan *stemming.* Setelah melakukan proses pre-*processing* terhadap dataset, selanjutnya ke tahap perhitungan TF-IDF yang menghitung probabilitas kata setiap *tweet* sehingga dapat menjadi masukan untuk perhitungan ke dalam SVM. Tahap selanjutnya masuk ke dalam metode SVM yang sebelumnya menghitung TF-IDF menghasilkan pembobotan sehingga dapat melakukan perhitungan metode dan mendapatkan hasil perhitungan tiap data *training* maupun *testing*. Data *training* dan data *testing* diklasifikasian setelah mendapatkan nilai masing masing pada proses sebelumnya dan menjadi acuan untuk hasil klasifikasi pada sistem.



Gambar 4. 1 Flowchart Sistem Klasifikasi Kalimat Mengandung Kalimat Perundungan Bentuk Tubuh

1. *Training Dataset*

Proses *training dataset* digunakan untuk melatih algoritma *Support Vector Machine* untuk proses klasifikasi. Dataset yang digunakan sebagai data masukan ialah hasil dari proses perhitungan TF-IDF sebelumnya. Data yang digunakan dalam proses training adalah 90 persen dari total dataset keseluruhan. Pengambilan data diambil secara acak dari seluruh *tweet* yang ada.

1. *Testing Dataset*

Proses *testing dataset* dilakukan untuk menghasilkan klasifikasi *tweet* berdasarkan dari apa yang telah dilakukan dari proses *training dataset*. Dataset yang digunakan sebagai data masukan adalah nilai dari TF-IDF yang telah dihitung pada tahap awal. Data yang digunakan dalam proses testing adalah 10 persen dari total dataset keseluruhan. Data diambil secara acak dari seluruh *tweet* yang ada. Hasil dari proses *testing* ialah hasil klasifikasi berupa positif, negatif dan netral.

1. Pengujian Sistem

Proses pengujian sistem dilakukan dengan cara menghitung tingkat akurasi sistem. Tingkat akurasi didapatkan dengan membandingkan label dan hasil dari kalsifikasi *tweet* pada proses *testing*. Banyaknya kesamaan antara label dan klasifikasi akan meningkatkan tingkat akurasi dari sistem yang dihasilkan.



Gambar 4. 2 Sub Proses Sistem

## 4.3 Analisis Data Masukan

Data yang digunakan adalah data yang diperoleh dari Twitter *Search* API. Twitter *Search* API merupakan fasilitas yang disediakan oleh Twitter untuk pengembang perangkat lunak agar dapat mengambil data yang diperlukan pada Twitter. Data berupa *tweet* yang berdasarkan kata kunci dari tokoh yang menjadi sorotan dalam periode tertentu. Tokoh sorotan yang digunakan sebagai kata kunci adalah Lucinta Luna. *Tweet* yang didapatkan masih berupa *tweet* dengan berbagai komponen. *Tweet* tersebut akan mengalami *preprocessing* untuk mendapatkan dataset yang baik.

*Preprocessing* memiliki beberapa tahapan agar menghasilkan dataset yang baik. Tahapan yang ada pada *preprocessing* sebagai berikut.



Gambar 4. 3 Tahap Preprocessing

Pada Gambar 4.2 dapat dilihat tahapan pertama praproses atau *preprocessing* adalah *input* dataset kemudian akan diproses pada tahap *tokenizing,* yakni tahap pemotongan tiap kata dalam kalimat atau *parsing* dengan menggunakan spasi sebagai delimiter yang akan menghasilkan token berupa kata. Tahap kedua *filtering* adalah penyaringan kata yang didapat dari *Tokenizing* yang dianggap tidak penting atau tidak memiliki makna. Setelah itu masuk ke tahap *Stemming,* yaitu mengembalikan kata-kata yang diperoleh dari hasil *Filtering* ke bentuk dasarnya, menghilangkan imbuhan awal (prefix) dan imbuhan akhir (sufix) sehingga didapat kata dasar.

1. *Tokenizing*

Pada proses *tokenizing* dilakukan pembacaan dokumen yang dimiliki dan memisahkan deretan kata didalam kalimat, paragraph atau dokumen menjadi token atau potongan kata tunggal. Dalam proses *tokenising* dapat dilakukan dengan menghilangkan URL, menghilangkan angka, menghilangkan karakter-karakter tertentu, mengubah kata dalam huruf kecil dan memisahkannya per spasi.. Alur pada *tokenizing* dapat dilihat dari *flowchart* dibawah ini.



Gambar 4. 4 Flowchart Tokenizing

1. *Filtering*

Pada tahapan ini dilakukan pengambilan kata-kata penting dari hasil *tokenizing* dengan membuang *stopwords* (kata-kata yang kurang penting). *Stopword* dapat berupa subjek atau kata penghubung. Filtering dilakukan dengan menentukan term mana yang akan digunakan untuk merepresentasikan dokumen. Alur dari *Filtering* dapat dilihat pada *flowchart* dibawah ini:



Gambar 4. 5 Flowchart Filtering

1. *Stemming*

*Stemming* merupakan suatu proses yang terdapat dalam sistem IR yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root* *word*) dengan menggunakan aturan-aturan tertentu. Pada tahap ini akan dicari *root* kata dari tiap kata hasil *Filtering*.



Gambar 4. 6 Flowchart Stemming

Contoh data yang mengalami *preprocessing* dapat dilihat di tabel 4.1 dan 4.2.

Tabel 4. 1 Contoh Proses Processing

|  |  |  |
| --- | --- | --- |
| **No**. | ***Preprocessing*** | **Hasil** |
| 1. | *Tweet* Awal | "Fakta Baru Lucinta Luna Terjerat Kasus Narkotika, Pemasok Narkoba Ternyata Teman Sendiri, Artis? https:// kaltim.tribunnews.com/2020/02/13/fak ta-baru-lucinta-luna-terjerat-kasus-narkotika-pemasok-narkoba-ternyata-teman-sendiri-artis Â  â€¦ via @tribunkaltim #LucintaLuna #Narko" |
| 2. | Tokenizing | ['fakta', 'baru', 'lucinta', 'luna', 'terjerat', 'kasus', 'narkotika', 'pemasok', 'narkoba', 'ternyata', 'teman', 'sendiri', 'artis', 'https', 'ta', 'baru', 'lucinta', 'luna', 'terjerat', 'kasus', 'narkotika', 'pemasok', 'narkoba', 'ternyata', 'teman', 'sendiri', 'artis', 'â', 'â', 'via', 'tribunkaltim', 'lucintaluna', 'narko'] |
| 3. | Filtering | fakta terjerat narkotika pemasok narkoba teman artis ta terjerat narkotika pemasok narkoba teman artis via tribunkaltim narko |
| 4. | Stemming | fakta jerat narkotika pasok narkoba teman artis ta jerat narkotika pasok narkoba teman artis via tribunkaltim narko |

Dari proses *preprocessing* yang telah dilakukan, *tweet* awal akan mengalami perbedaan dengan *tweet* yang telah di *preprocessing*. *Tweet* yang telah melalui proses *preprocessing* memiliki komponen *tweet* yang lebih ringkas. Perbedaan dari kedua *tweet* tesebut dapat dilihat pada tabel 4.2

Tabel 4. 2 Perbedaan Tweet Sebelum dan Sesudah Preprocessing

|  |  |
| --- | --- |
| ***Tweet* Sebelum *Preprocessing*** | ***Tweet* Sesudah *Preprocessing*** |
| "Fakta Baru Lucinta Luna Terjerat Kasus Narkotika, Pemasok Narkoba Ternyata Teman Sendiri, Artis? https:// kaltim.tribunnews.com/2020/02/13/fak ta-baru-lucinta-luna-terjerat-kasus-narkotika-pemasok-narkoba-ternyata-teman-sendiri-artis Â  â€¦ via @tribunkaltim #LucintaLuna #Narko" | fakta jerat narkotika pasok narkoba teman artis ta jerat narkotika pasok narkoba teman artis via tribunkaltim narko |

Dataset *tweet* yang telah mengalami *preprocessing* akan masuk ke proses selanjutnya, yaitu perhitungan TF IDF. Proses ini akan sangat menentukan *training* dan *testing* yang menjadi proses selanjutnya karena training dan testing akan menggunakan dataset yang telah diubah ke dalam TF IDF, yaitu data yang berupa angka dari ektraksi kalimat dalam *tweet*.

TF IDF merupakan salah satu cara yang digunakan untuk menghitung bobot setiap kata. Perhitungan ini dilakukan dengan cara menghitung nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap kata dalam setiap *tweet*.

Contoh perhitungan manualisasi TF-IDF

Tabel 4.3 ialah contoh dari dataset yang akan diolah, yang telah melalui proses *preprocessing* dan telah di labeli. Proses pelabelan data akan dibagi menjadi beberapa kategori, yaitu positif, negatif dan netral. Data akan diberi label menurut pendapat beberapa orang dan dibantu oleh ahli psikolog untuk memvalidasi data.

Tabel 4. 3 Contoh tweet untuk perhitungan TFIDF

|  |  |  |
| --- | --- | --- |
| No. | *Tweet* | Label |
| 1. | krna kmu bau lucintaluna | Positif |
| 2. | lucintaluna cantik putih insesyur | Negatif |
| 3. | ria ricis nilai sosok luna tangkap polisi penyalahgunaan narkoba lucintaluna ria ricis sosok luna | Netral |

1. *Tweet* pada tabel 4.3 akan mengalami *tokenizing* atau *tweet* akan dipisah berdasarkan kata per kata.

Tabel 4. 4 Daftar Kata

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kata(Term)** | **No.** | **Kata(Term)** |
| 1. | krna | 9. | ricis |
| 2. | kmu | 10. | nilai |
| 3. | bau | 11. | sosok |
| 4. | lucintaluna | 12. | luna |
| 5. | cantik | 13. | tangkap |
| 6. | putih | 14. | polisi |
| 7. | insesyur | 15. | penyalahgunaan |
| 8. | ria | 16. | narkoba |

1. Melakukan perhitungan TF, yaitu menghitung frekuensi kemunculan kata pada dokumen.

Tabel 4. 5 Term Frequency

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Kata(Term)** | ***Term Frequency* (TF)** | | |
| **D1** | **D2** | **D3** |
| 1. | krna | 1 | 0 | 0 |
| 2. | kmu | 1 | 0 | 0 |
| 3. | bau | 1 | 0 | 0 |
| 4. | lucintaluna | 1 | 1 | 1 |
| 5. | cantik | 0 | 1 | 0 |
| 6. | putih | 0 | 1 | 0 |
| 7. | insesyur | 0 | 1 | 0 |
| 8. | ria | 0 | 0 | 2 |
| 9. | ricis | 0 | 0 | 2 |
| 10. | nilai | 0 | 0 | 1 |
| 11. | sosok | 0 | 0 | 2 |
| 12. | luna | 0 | 0 | 2 |
| 13. | tangkap | 0 | 0 | 1 |
| 14. | polisi | 0 | 0 | 1 |
| 15. | penyalahgunaan | 0 | 0 | 1 |
| 16. | narkoba | 0 | 0 | 1 |

1. Melakukan perhitungan DF, yaitu menghitung frekuensi kemunculan kata pada keseluruhan dokumen.

Tabel 4. 6 Document Frequency

|  |  |  |
| --- | --- | --- |
| No. | Kata(*Term*) | *Document* *Frequency* (DF) |
| 1. | krna | 1 |
| 2. | kmu | 1 |
| 3. | bau | 1 |
| 4. | lucintaluna | 3 |
| 5. | cantik | 1 |
| 6. | putih | 1 |
| 7. | insesyur | 1 |
| 8. | ria | 2 |
| 9. | ricis | 2 |
| 10. | nilai | 1 |
| 11. | sosok | 2 |
| 12. | luna | 2 |
| 13. | tangkap | 1 |
| 14. | polisi | 1 |
| 15. | penyalahgunaan | 1 |
| 16. | narkoba | 1 |

1. Melakukan perhitungan IDF

Tabel 4. 7 Invers Document Frequency

|  |  |  |
| --- | --- | --- |
| No. | Kata(*Term*) | *Invers Document* *Frequency* (IDF) |
| 1. | krna | 0,477121 |
| 2. | kmu | 0,477121 |
| 3. | bau | 0,477121 |
| 4. | lucintaluna | 0 |
| 5. | cantik | 0,477121 |
| 6. | putih | 0,477121 |
| 7. | insesyur | 0,477121 |
| 8. | ria | 0,176091 |
| 9. | ricis | 0,176091 |
| 10. | nilai | 0,477121 |
| 11. | sosok | 0,176091 |
| 12. | luna | 0,176091 |
| 13. | tangkap | 0,477121 |
| 14. | polisi | 0,477121 |
| 15. | penyalahgunaan | 0,477121 |
| 16. | narkoba | 0,477121 |

1. Melakukan peritungan TF-IDF dari hasil yang telah didapatkan.

Tabel 4. 8 Perhitungan TF-IDF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Kata(*Term*) | TF-IDF | | |
| D1 | D2 | D3 |
| 1. | krna | 0,477121 | 0 | 0 |
| 2. | kmu | 0,477121 | 0 | 0 |
| 3. | bau | 0,477121 | 0 | 0 |
| 4. | lucintaluna | 0 | 0 | 0 |
| 5. | cantik | 0 | 0,477121 | 0 |
| 6. | putih | 0 | 0,477121 | 0 |
| 7. | insesyur | 0 | 0,477121 | 0 |
| 8. | ria | 0 | 0 | 0,352183 |
| 9. | ricis | 0 | 0 | 0,352183 |
| 10. | nilai | 0 | 0 | 0,477121 |
| 11. | sosok | 0 | 0 | 0,352183 |
| 12. | luna | 0 | 0 | 0,352183 |
| 13. | tangkap | 0 | 0 | 0,477121 |
| 14. | polisi | 0 | 0 | 0,477121 |
| 15. | penyalahgunaan | 0 | 0 | 0,477121 |
| 16. | narkoba | 0 | 0 | 0,477121 |

## 4.4 Analisis Algoritma

### 4.4.1 Klasifikasi dengan algoritma *Support Vector Machine*

## 4.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis untuk mengetahui spesifikasi kebutuhan sistem yang dibangun. Spesifikasi kebutuhan,yaitu analisis perangkat lunak dan perangkat keras.

1. Analisis Kebutuhan Perangkat Lunak

Tabel 4. 9 Kebutuhan Perangkat Lunak

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat** | **Keterangan** |
| 1. | *Web Browser* | *Google Chrome* |
| 2. | *Web Server* | *Apache* |
| 3. | *DBMS* | *PhpMyAdmin* |
| 4. | *Text Editor* | *Visual Studio Code* |
| 5. | *API* | *Twitter API* |
| 6. | *Framework* | *Python Flask* |

1. Analisis Kebutuhan Perangkat Keras

Tabel 4. 10 Kebutuhan Perangkat Keras

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat** | **Keterangan** |
| 1. | *Processor* | *AMD Ryzen 5* |
| 2. | *Memory* | *4GB* |
| 3. | *Hardisk* | *1TB* |

## 4.6 Analisis Kebutuhan Fungsional

Analisa kebutuhan fungsional berisikan proses-proses yang dilakukan oleh sistem. Dalam hal ini user sebagai pengguna dapat menggunakan layanan-layanan pada sistem, antara lain :

1. Sistem dapat melakukan proses *import* dataset kedalam *database.*
2. Sistem dapat melakukan proses *load* dataset dari *database.*
3. Sistem dapat melakukan proses *preprocessing* pada dataset.
4. Sistem dapat melakukan proses perhitungan TF-IDF.
5. Sistem dapat melakukan klasifikasi *tweet* ke dalam klasifikasi positif, negatif maupun netral.
6. Sistem dapat menampilkan grafik dari *tweet* yang telah diklasifikasi.
7. Sistem dapat melakukan pengujian dari hasil klasifikasi.

### 4.6.1 *Database*

*Database* digunakan untuk menyimpan dataset *tweet* yang didapatkan dari Twitter API. Database yang digunakan memiliki nama perundungan dengan jumlah tabel 3, yaitu tabel dataset, tabel terms, tabel term\_frequency. Atribut yang ada pada masing-masing tabel sebagai berikut.

Tabel 4. 11 Atribut Tabel Dataset

|  |  |  |
| --- | --- | --- |
| **No.** | **Atribut** | **Tipe Data** |
| 1. | *id* | *bigint(50)* |
| 2. | *teks* | *text* |
| 3. | *label* | *text* |
| 4. | *prepro* | *text* |

Tabel 4. 12 Atribut Tabel Terms

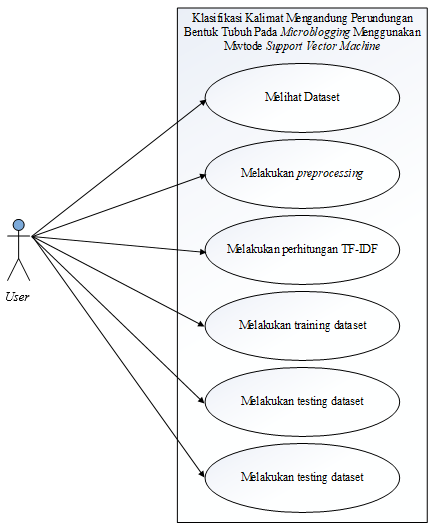
|  |  |  |
| --- | --- | --- |
| **No.** | **Atribut** | **Tipe Data** |
| 1. | *id* | *int(11)* |
| 2. | *term* | *varchar(255)* |

Tabel 4. 13 Atribut Tabel Term\_Frequency

|  |  |  |
| --- | --- | --- |
| **No.** | **Atribut** | **Tipe Data** |
| 1. | *id* | *int(11)* |
| 2. | *id\_term* | *int(11)* |
| 3. | *id\_tweet* | *bigint(50)* |
| 4. | *count* | *int(11)* |

### 4.6.2 *Use* *Case*

*Use* *case* merupakan diagram yang terdiri dari aktor dan proses yang dapat dilakukan oleh aktor tersebut. *Use* *case* dibuat berdasarkan layanan-layanan yang dimiliki oleh sistem. *Use case* memiliki hak akses untuk melakukan beberapa layanan, yaitu melihat dataset, melakukan preprocessing, melakukan TF IDF, melakukan training dataset, melakukan testing dataset, melakukan pengujian.



Gambar 4. 7 Use Case Diagram

1. Deskripsi *Use Case*

Tabel 4. 14 Deskripsi Use Case

|  |  |  |
| --- | --- | --- |
| **No.** | **Use Case** | **Deskripsi** |
| 1. | *Dataset* | Data yang didapatkan dari Twitter sebagai data masukan untuk klasifikasi |
| 2. | *Preprocessing* | Proses untuk memperbaiki kalimat dari *tweet* agar dapat digunakan pada proses selanjutnya |
| 3. | *TF-IDF* | Proses perhitungan untuk mendapatkan nilai dari *dataset* |
| 4. | *Training Dataset* | Proses untuk melatih algoritma SVM dan menghasilkan model classifier untuk proses *testing* |
| 5. | *Testing Dataset* | Proses untuk klasifikasi *dataset* ke dalam positif, negatif atau netral |
| 6. | Pengujian | Membandingkan hasil klasifikasi dengan label |

1. Skenario *Use Case*

Skenario use case menggunakan alur penggunaan sistem. Setiap skenario digambarkan dari sudut pandang aktor yang berinteraksi dengan perangkat lunak dalam berbagai cara.

1. Melihat *Dataset*

Tabel 4. 15 Skenario Use Case Melihat Dataset

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Melihat Dataset |
| **Tujuan** | Menampilkan kepada user tentang dataset yang digunakan untuk klasifikasi |
| **Aktor** | User |
| **Skenario Utama** | |
| **Kondisi Awal** | User ditampilkan beberapa menu pilihan |
| **Aksi Aktor** | Reaksi Sistem |
| **Memilih menu dataset** | Melakukan load dataset dari database |
| **Kondisi Akhir** | User dapat melihat dataset yang digunakan untuk klasifikasi |

1. Melakukan *Preprocessing*

Tabel 4. 16 Skenario Use Case Melakukan Preprocessing

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 2 |
| **Nama** | Melakukan preprocessing |
| **Tujuan** | Menampilkan kepada user hasil dari dataset yang telah mengalami preprocessing |
| **Aktor** | User |
| **Skenario Utama** | |
| **Kondisi Awal** | User ditampilkan beberapa menu pilihan |
| **Aksi Aktor** | Reaksi Sistem |
| **Memilih menu preprocessing** | Melakukan proses preprocessing pada dataset |
| **Kondisi Akhir** | User dapat melihat dataset yang telah mengalami preprocessing |

1. Melakukan TF-IDF

Tabel 4. 17 Skenario Use Case Melakukan TF-IDF

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 3 |
| **Nama** | Melakukan Perhitungan TF-IDF |
| **Tujuan** | Menampilkan kepada user hasil perhitungan TF-IDF untuk proses training dan testing |
| **Aktor** | User |
| **Skenario Utama** | |
| **Kondisi Awal** | User ditampilkan beberapa menu pilihan |
| **Aksi Aktor** | Reaksi Sistem |
| **Memilih menu TF-IDF** | Melakukan perhitungan TF-IDF dan dataset yang telah mengalami preprocessing |
| **Kondisi Akhir** | User dapat melihat hasil perhitungan TF-IDF untuk proses training dan testing |

1. Melakukan *training dataset*

Tabel 4. 18 Skenario Use Case Melakukan Training Dataset

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 4 |
| **Nama** | Melakukan Training Dataset |
| **Tujuan** | Melakukan training pada dataset |
| **Aktor** |  |
| **Skenario Utama** | |
| **Kondisi Awal** | User ditampilkan beberapa menu pilihan |
| **Aksi Aktor** | Reaksi Sistem |
| **Memilih menu training** | Melakukan proses training pada dataset yang dipilih |
| **Kondisi Akhir** | User dapat melihat dataset yang digunakan untuk proses training |

1. Melakukan *testing dataset*

Tabel 4. 19 Skenario Use Case Testing Dataset

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 5 |
| **Nama** | Melakukan Testing Dataset |
| **Tujuan** | Menampilkan kepada user hasil klasifikasi pada proses testing |
| **Aktor** |  |
| **Skenario Utama** | |
| **Kondisi Awal** | User ditampilkan beberapa menu pilihan |
| **Aksi Aktor** | Reaksi Sistem |
| **Memilih menu testing** | Melakukan proses testing pada dataset yang dipilih |
| **Kondisi Akhir** | User dapat melihat hasil klasifikasi |

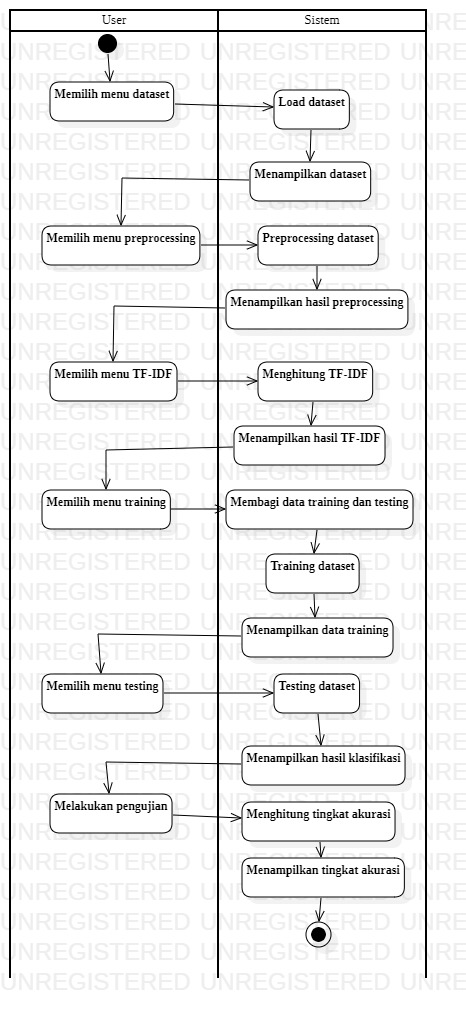
1. Melakukan pengujian

Tabel 4. 20 Skenario Use Case Melakukan Pengujian

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 6 |
| **Nama** | Melakukan Pengujian |
| **Tujuan** | Menampilkan kepada user tentang tingkat akurasi klasifikasi |
| **Aktor** |  |
| **Skenario Utama** | |
| **Kondisi Awal** | User ditampilkan beberapa menu pilihan |
| **Aksi Aktor** | Reaksi Sistem |
| **Memilih menu dataset** | Melakukan analisis tingkat akurasi dari klasifikasi |
| **Kondisi Akhir** | User dapat melihat tingkat akurasu klasifikasi |

### 4.6.3 *Activity Diagram*

*Activity diagram* menggambarkan aktivitas dari sebuah sistem atau proses. Dalam *activity diagram* yang terlibat adalah antara user dan sistem. Terdapat beberapa aktivitas yang digambarkan sesuai dengan kondisi pada klasifikasi kalimat mengandung perundungan bentuk tubuh.



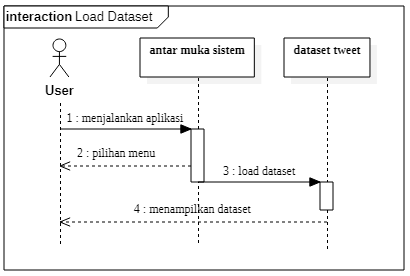
Gambar 4. 8 Activity Diagram

### 4.6.4 *Sequence Diagram*

*Sequence diagram* menggambarkan proses objek pada *use case* dengan mendeskripsikan alur yang dilakukan oleh aktor terhadap respon dari setiap objek. Terdapat enam diagram yang menggambarkan *sequence diagram* berdasarkan proses yang dilakukan sistem.

1. *Load Dataset*

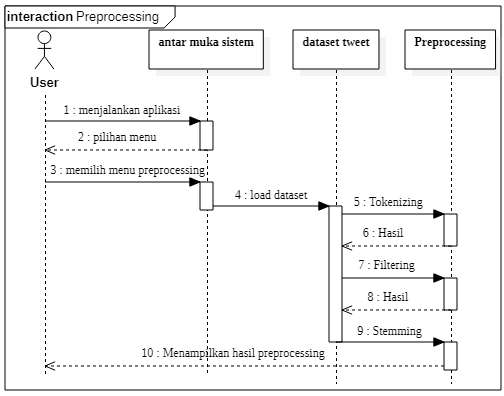
*Load dataset* diawali dengan *user* menjalankan aplikasi. Pada antarmuka sistem terdapat pilihan menu. *User* memilih menu dataset, kemudian sistem akan memuat dataset dari tabel dataset yang ada di *database*. Sistem akan menampilkan *dataset* kepada *user*.



Gambar 4. 9 Sequence Diagram Load Dataset

1. *Preprocessing*

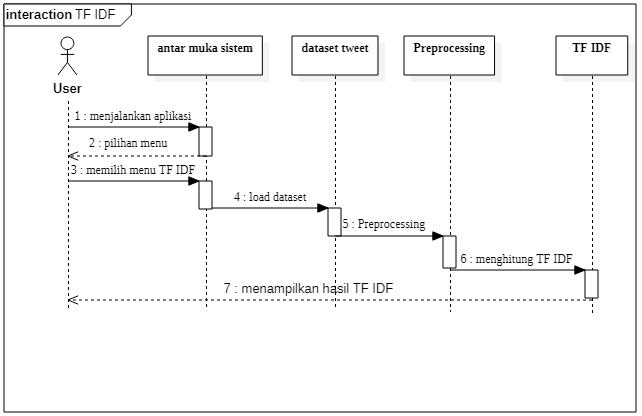
*Preprocessing* diawali dengan *user* menjalankan aplikasi. Pada antarmuka sistem terdapat pilihan menu. *User* memilih menu *preprocessing*, kemudian sistem akan memuat *dataset* dari tabel *dataset* yang ada di *database*. *Dataset* tersebut akan diolah oleh sistem mulai dari mengubah semua huruf ke huruf kecil, menghapus URL, menghapus angka dan karakter yang kurang penting, menghapus stopword, dan menghilangkan imbuhan awal dan imbuhan akhir. Sistem akan menampilkan hasil ke *user*.



Gambar 4. 10 Sequence Diagram Preprocessing

1. TF-IDF

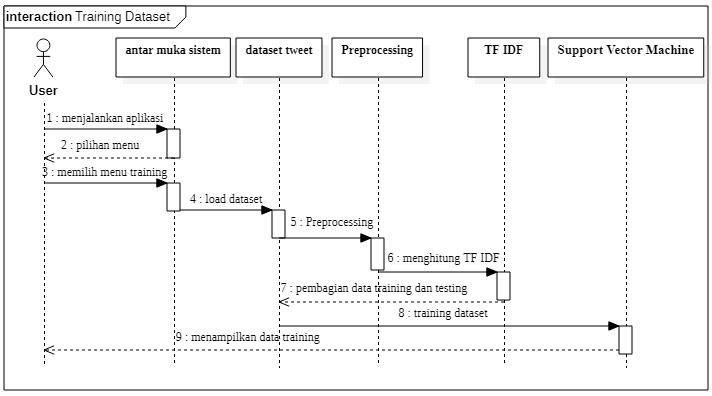
TF IDF diawali dengan *user* menjalankan aplikasi. Pada antarmuka sistem terdapat pilihan menu. *User* memilih menu TF IDF, kemudian sistem akan memuat *dataset* dari tabel *dataset* yang ada di *database*. Sistem akan melakukan *preprocessing*, lalu menghitung nilai TF IDF. Sistem akan menampilkan hasil ke *user*.



Gambar 4. 11 Sequence Diagram TF IDF

1. *Training Dataset*

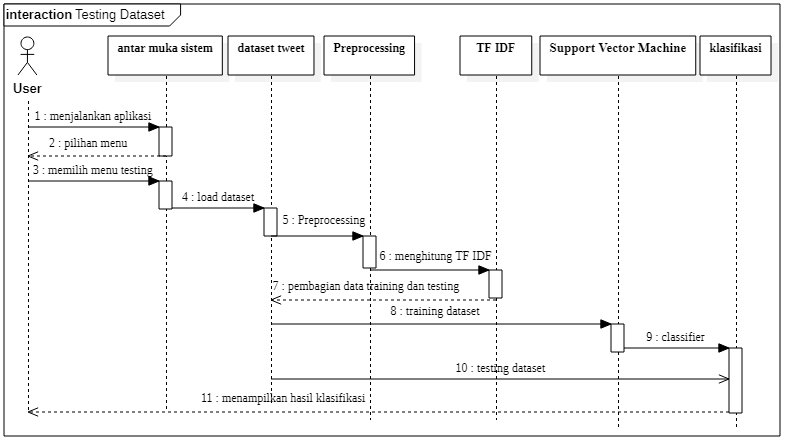
*Training* *dataset* diawali dengan *user* menjalankan aplikasi. Pada antarmuka sistem terdapat pilihan menu. *User* memilih menu *training*, kemudian sistem akan memuat *dataset* dari tabel *dataset* yang ada di *database*. Sistem akan melakukan *preprocessing*, lalu menghitung nilai TF IDF. Kemudian sistem akan membagi *dataset* untuk *training* dan *testing* secara acak. Sistem akan mengambil data *training* kemudian masuk ke dalam proses *training* *dataset* menggunakan algoritma *Support Vector Machine*. Sistem akan menampilkan data *training.*



Gambar 4. 12 Sequence Diagram Training Dataset

1. *Testing Dataset*

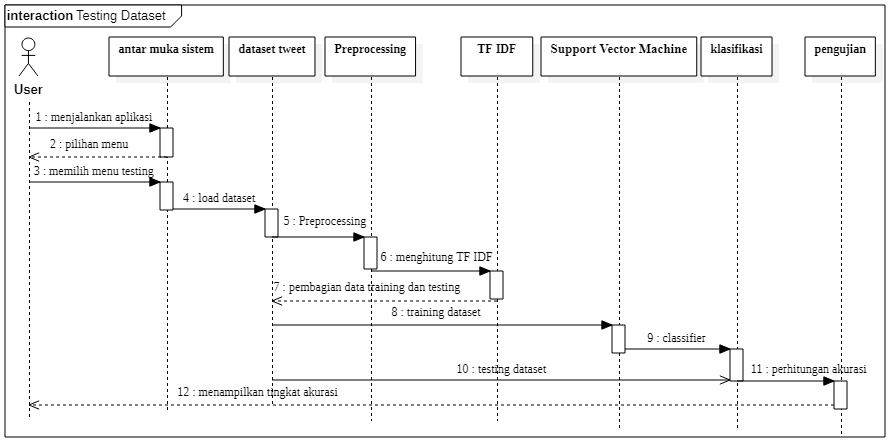
*Testing* *dataset* diawali dengan *user* menjalankan aplikasi. Pada antarmuka sistem terdapat pilihan menu. *User* memilih menu *testing*, kemudian sistem akan memuat *dataset* dari tabel *dataset* yang ada di *database*. Sistem akan melakukan *preprocessing*, lalu menghitung nilai TF IDF. Kemudian sistem akan membagi *dataset* untuk *training* dan *testing* secara acak. Sistem akan mengambil data *training* kemudian masuk ke dalam proses *training* dataset menggunakan algoritma *Support Vector Machine*. *Training* tersebut akan menghasilkan model *classifier* untuk klasifikasi pada proses *testing*. Sistem mengambil data *testing* dan melakukan *testing* menggunakan model *classifier* dari proses *training*. Hasil klasifikasi akan ditampilkan ke *user*.



Gambar 4. 13 Sequence Diagram Testing Dataset

1. Pengujian

Pengujian tingkat akurasi diawali dengan *user* menjalankan aplikasi. Pada antarmuka sistem terdapat pilihan menu. *User* memilih menu *testing*, kemudian sistem akan memuat *dataset* dari tabel dataset yang ada di database. Sistem akan melakukan *preprocessing*, lalu menghitung nilai TF IDF. Kemudian sistem akan membagi dataset untuk *training* dan *testing* secara acak. Sistem akan mengambil data *training* kemudian masuk ke dalam proses *training* *dataset* menggunakan algoritma *Support Vector Machine*. *Training* tersebut akan menghasilkan model *classifier* untuk klasifikasi pada proses *testing*. Sistem mengambil data *testing* dan melakukan *testing* menggunakan model *classifier* dari proses *training*. Dari hasil klasifikasi yang didapatkan akan dicari tingkat akurasi menggunakan *precision* dan *accuracy*. Sistem akan menghitung tingkat akurasi dan menampilkan hasil ke *user*.



Gambar 4. 14 Sequence Diagram Pengujian

# BAB V. IMPLEMENTASI DAN PENGUJIAN

## 5.1 Implementasi *Database*

Implementasi *database* dengan nama perundungan yang memiliki tiga tabel, yaitu tabel dataset, term dan term\_frequency sesuai dengan analisis dan perancangan sebagai berikut**.**

Gambar 5. 1 Tabel Dataset

## 5.2 Implementasi Proses Klasifikasi

Implementasi proses klasifikasi berdasarkan pada analisis dan perancangan terdiri dari beberapa proses sebagai berikut.

## 5.2.1 *Load Dataset*

### 5.2.2 *Preprocessing*

### 5.2.3 TF IDF

### 5.2.4 *Training*

### 5.2.5 *Testing*

### 5.2.6 Pengujian

## 5.3 Implementasi *Interface*

## 5.4 Pengujian Fungsional

## 5.5 Pengujian Akurasi

# BAB VI. HASIL DAN PEMBAHASAN