

KLASIFIKASI KELUHAN BERBASIS APLIKASI WEB PADA UNIVERSITAS TELKOM MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBORS

*(WEB BASED COMPLAINT CLASSIFICATION APPLICATION AT TELKOM
UNIVERSITY USING K-NEAREST NEIGHBORS ALGORITHM)*

Fachri Mohammad¹, Roswan Latuconsina², M Reza Faisal³

^{1,2,3} Universitas Telkom, Bandung

fcrmhmd@student.telkomuniversity.ac.id¹, roswanlatuconsina@telkomuniversity.ac.id²,
rezafaisal@365.telkomuniversity.ac.id³

Abstrak

Saat ini ada fasilitas untuk menyampaikan keluhan dan masukan pada Universitas Telkom. Namun karena banyak pengguna aplikasi seperti mahasiswa/i yang belum mengetahui secara rinci struktur organisasi pada Universitas Telkom hal tersebut sering terjadi kesalahan sasaran dalam memasukkan keluhan dan masukan pada iGracias yang menyebabkan waktu respon dan penyelesaian menjadi terkendala dikarenakan banyak yang tidak paham akan unit yang dituju. Untuk menyelesaikan permasalahan tersebut maka, dibuat aplikasi handling complaint berbasis web menggunakan text mining untuk mendapatkan informasi dari teks yang masuk ke dalam database lalu diklasifikasikan menggunakan metode K-NEAREST NEIGHBOR untuk mengklasifikasikan keluhan yang masuk agar tepat sasaran. sehingga keluhan yang ada akan dikirim ke unit yang dituju secara langsung dan ke unit yang tepat. Dengan adanya aplikasi ini diharapkan akan membantu bagian audit dari Universitas Telkom dalam mengatasi keluhan dan masukan yang ada serta aplikasi mudah untuk digunakan oleh mahasiswa/i. K-Nearest Neighbor (KNN) adalah salah satu algoritma pembelajaran mesin paling sederhana. Algoritma ini tujuannya adalah untuk mengklasifikasikan objek ke dalam salah satu kelas yang telah ditentukan Hasil akhir dari tugas akhir ini bahwa algoritma K-Nearest Neighbor (KNN) mendapatkan hasil training model sebesar 90% dan data test sebesar 10% KNN nantinya dapat mengenali per kata yang telah berhasil di input oleh pengguna sehingga menghasilkan tingkat akurasi sebesar 78%.

Kata Kunci: Aplikasi Web, Handling Complaint, K- Nearest Neighbor, Text Mining.

Abstract

Currently there are facilities to submit complaints and inputs to Telkom University, but because of the many students who do not know in detail the organizational structure and working arrangements at Telkom University, there are often target errors in entering complaints and inputs on iGracias, which causes response and settlement time to be constrained. To solve the problem, a web-based complaint handling application is created using text mining to get information from text entered into the database and then classify using the K-NEAREST NEIGHBOR method to classify incoming complaints to be on target. So that the existing complaint will be sent to the intended unit directly and to the right unit. With this application is expected to help the audit part of Telkom University in overcoming existing complaints and inputs and applications easy to use by students. K-Nearest Neighbor (KNN) is one of the simplest machine learning algorithms. The algorithm aims to classify objects into one of the specified classes The final result of this final task is that the K-Nearest Neighbor (KNN) algorithm gets a training model result of 90% and the data test of 10% KNN can later recognize per word that has been successfully input by the user resulting in an accuracy rate of 78%.

Keywords: Handling Complaint, K- Nearest Neighbor, Text Mining, Web application.

1. Pendahuluan

Pengelolaan keluhan dan masukan yang teratur dapat menjadi bahan masukan untuk meningkatkan kualitas pelayanan suatu organisasi atau perusahaan[1]. salah satunya pada Universitas Telkom. Pihak universitas telah menyediakan fasilitas untuk menampung keluhan dan masukan pada sistem iGracias. Namun, karena banyaknya pengguna yang belum mengetahui secara rinci struktur organisasi dan tata kerja pada Universitas Telkom, mengakibatkan kesalahan sasaran dalam memasukkan keluhan dan masukan. Selain itu Universitas Telkom juga memerlukan pegawai yang bertugas untuk memantau setiap keluhan yang masuk dan melakukan perbaikan untuk setiap keluhan yang salah sasaran, sehingga menyebabkan waktu respon dan penyelesaian menjadi lebih lama. Masalah tersebut dapat diselesaikan dengan membuat aplikasi yang mudah digunakan serta menerapkan konsep kecerdasan buatan. Salah satu cara yang dapat

digunakan yaitu dengan menggunakan text mining. Analisis text mining diperlukan untuk menggali informasi yang ada di dalam teks lalu mengkategorikan atau mengklasifikasikan keluhan ke unit yang dituju. Pada penelitian ini digunakan metode K-Nearest Neighbor untuk mengklasifikasikan teks.

2. Dasar Teori

2.1 Text Mining

Text mining adalah proses menemukan informasi di dalam data tekstual dalam jumlah besar dan mengidentifikasi pola dan hubungan yang menarik dalam data tekstual secara otomatis [2]. Pada dasarnya, *text mining* adalah bidang interdisipliner yang mengacu pada pengambilan informasi, penggalian data, pembelajaran mesin, statistik, dan linguistik komputasi [3].

Text mining akan mengambil dan menganalisis indeks numerik yang penting dari sekumpulan teks, dan kemudian informasi yang terdapat dalam teks tersebut akan dapat diakses dengan menggunakan berbagai algoritma data mining. Untuk mendapatkan hasil akhir dari text mining diperlukan beberapa proses seperti text preprocessing dan pembobotan teks [4].

2.2 Text Preprocessing

Tahap *preprocessing* adalah salah satu implementasi dari *text mining*. *Preprocessing* dilakukan untuk mengubah suatu teks yang tidak terstruktur menjadi teks yang terstruktur. Berikut ini adalah tahapan *preprocessing* pada penelitian ini [5].

- *Case Folding*, adalah proses mengubah format teks dari huruf besar menjadi huruf kecil.
- *Tokenizing*, adalah proses pemotongan kalimat menjadi beberapa bagian kata.
- *Filtering*, adalah proses mengambil kata-kata yang penting atau menghapus kata, angka, dan tanda baca yang tidak diperlukan.
- *Stemming*, adalah proses mengubah kata berimbuhan menjadi kata dasar.

2.3 Pembobotan Kata

Pembobotan kata merupakan proses pemberian bobot term pada dokumen sehingga dapat diketahui kemiripan suatu term di dalam dokumen. Bobot pada term menyatakan kontribusi term terhadap suatu dokumen. Kata yang penting dalam dokumen dapat dilihat dari frekuensi kemunculan kata terhadap dokumen. Pembobotan ini digunakan nantinya oleh algoritma K-Nearest Neighbor untuk klasifikasi dokumen. Berikut merupakan jenis pembobotan kata yang digunakan pada penelitian ini.

2.3.1 Term Frequency (TF)

Term frequency merupakan frekuensi kemunculan term t pada dokumen d . Setiap term dianggap memiliki kepentingan yang sebanding dengan jumlah kemunculan term pada dokumen [6]. Ditulis dalam bentuk :

$$TF(d, t) = f(d, t) \quad (2.1)$$

Dengan TF adalah *term frequency* dan $f(d, t)$ adalah frekuensi kemunculan term t pada dokumen d .

2.3.2 Inverse Document Frequency (IDF)

Inverse document frequency memperhatikan jumlah kemunculan term pada kumpulan dokumen. Berbeda dengan *term frequency* yang memperhatikan kemunculan term di dalam dokumen. IDF berfungsi mengurangi bobot suatu term jika kemunculannya banyak tersebar di seluruh dokumen. Term yang jarang muncul pada kumpulan dokumen sangat bernilai pada pembobotan ini. Kepentingan tiap term dianggap memiliki proporsi yang berkebalikan dengan jumlah dokumen yang mengandung term [6]. Ditulis dalam bentuk :

$$IDF(t) = \log\left(\frac{N}{df(t)}\right) \quad (2.2)$$

Dengan IDF adalah *inverse document frequency*, N adalah jumlah total dokumen, dan $df(t)$ adalah jumlah dokumen yang berisi term t .

2.3.3 Term Frequency-Inverse Document Frequency (TFIDF)

TFIDF merupakan perkalian antara TF dan IDF sehingga bisa menghasilkan performa yang lebih baik.

$$TFIDF(d, t) = TF(d, t) \cdot IDF(t) \quad (2.3)$$

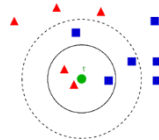
Pada TFIDF pembobotan term pada tiap dokumen dilakukan dengan memperhatikan kelangkaan term pada seluruh dokumen. Hasil akhir dari TFIDF adalah bobot term pada tiap dokumen. Term yang sering muncul pada dokumen tapi jarang muncul pada kumpulan dokumen diberikan nilai bobot yang tinggi [9].

2.4 K-Nearest Neighbor

K-Nearest Neighbor (KNN) adalah salah satu algoritma pembelajaran mesin malas paling sederhana. Algoritma ini tujuannya adalah untuk mengklasifikasikan objek ke dalam salah satu kelas yang telah ditentukan [7]. Idenya adalah untuk menghafal set pelatihan dan kemudian untuk memprediksi label dari

setiap contoh baru berdasarkan label dari tetangga terdekatnya di set pelatihan. Dasar pemikiran di balik metode tersebut didasarkan pada asumsi bahwa fitur yang digunakan untuk menggambarkan titik domain relevan dengan pelabelan mereka dengan cara yang membuat titik dekat 8 kemungkinan memiliki label yang sama.

Lebih jauh lagi, dalam beberapa situasi, bahkan ketika set pelatihan sangat besar, menemukan tetangga terdekat dapat dilakukan dengan sangat cepat. Metode Nearest Neighbor mencari label pada titik uji apa pun tanpa mencari prediktor dalam beberapa kelas fungsi yang telah ditentukan [8]. Pada KNN terdapat rumus yang digunakan untuk menghitung jarak yaitu Euclidean Distance yang merupakan perhitungan jarak yang sering digunakan untuk menghitung kesamaan pada dua vektor menggunakan rumus Euclidean distance [9].



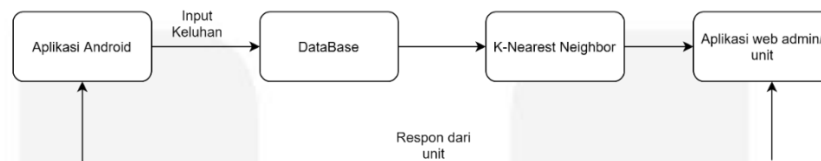
Gambar 2.1 Contoh ilustrasi KNN.

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2}$$

3. Perancangan Sistem

3.1 Gambaran Umum Sistem

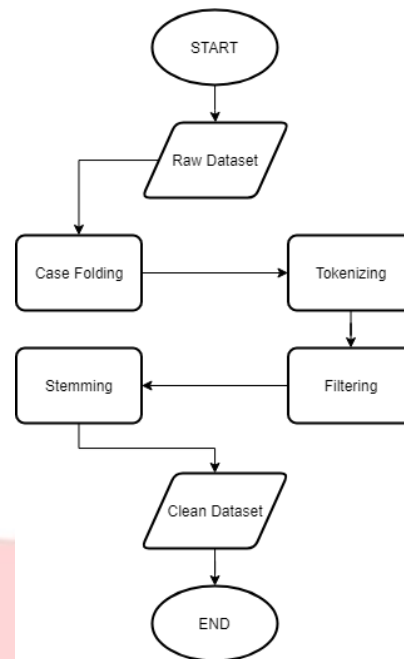
Sistem ini menggunakan aplikasi Android untuk memasukkan keluhan dari mahasiswa. Lalu data keluhan akan masuk ke database dan akan diproses oleh algoritma K-Nearest Neighbor. Algoritma yang terdiri dari tahap preprocessing, pemberian label dan pembobotan kata menggunakan TFIDF lalu tahap pemodelan klasifikasi. Setelah itu keluhan akan masuk ke unit yang dituju dan dapat dilihat pada website admin dan unit. Unit dapat memberikan tanggapan atau respon dan diteruskan langsung ke mahasiswa pengirim keluhan..



Gambar 3.1 Gambaran umum sistem handling complaint.

3.2 Mekanisme Preprocessing

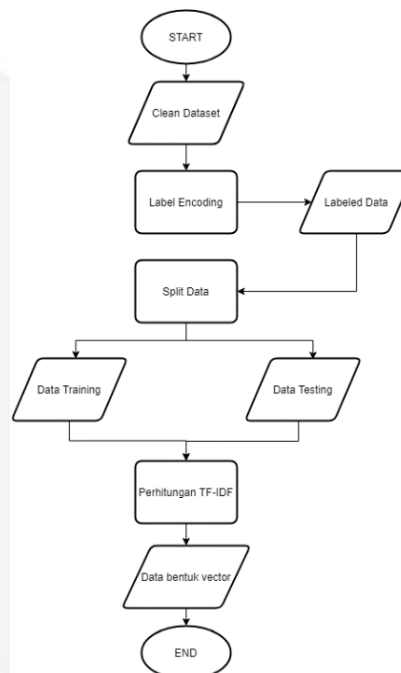
Preprocessing dilakukan untuk mentransformasikan teks yang tidak terstruktur menjadi teks yang terstruktur [6]. Berikut ini merupakan tahapan preprocessing pada penelitian ini.



Gambar 3.2 Flowchart tahapan preprocessing.

3.3 Mekanisme Pembobotan Kata

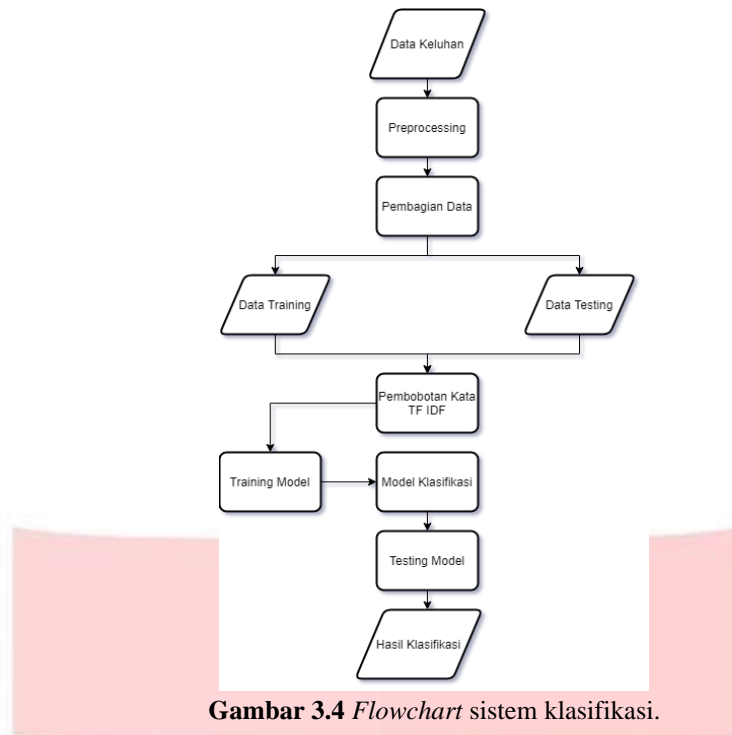
Pembobotan kata merupakan proses pemberian bobot term pada dokumen sehingga dapat diketahui kemiripan suatu term di dalam dokumen. Bobot pada term menyatakan kontribusi term terhadap suatu dokumen TFIDF.



Gambar 3.3 Flowchart TFIDF.

3.4 Mekanisme K-Nearest Neighbor

Sistem ini menggunakan metode K-Nearest Neighbor untuk mengklasifikasikan keluhan ke unit yang dituju. proses ini terdiri dari 4 tahap, yaitu tahap preprocessing, tahap pembagian data, tahap pembobotan kata, dan tahap pengujian menggunakan model klasifikasi.



Gambar 3.4 Flowchart sistem klasifikasi.

Setelah tahap pembobotan kata selesai, maka dokumen telah direpresentasikan sebagai vektor dan dapat dijadikan input untuk perhitungan algoritma. Di bawah ini merupakan perhitungan vektor KNN pada algoritma K-Nearest Neighbor. Sebelumnya pada perhitungan ini, akan diambil contoh teks dari 3 kelas yang ada di dataset.

Tabel 3.1 Dataset buatan

No.	Keluhan	Bagian
0	bayar eprt status igracias update bantu	Pusat Bahasa (2)
1	adakan eprt online mudah proses ketes eprt	Pusat Bahasa (2)
2	coba bayar bpp atm teller ganggu rek telkomnya rek mandiri	Akuntansi (0)
3	perihal alih uang bpp semester semester	Akuntansi (0)
4	koleksi buku seni	Open Library (1)
5	stop kontak unjung stop kontak fungsi	Open Library (1)
6	alami sulit bayar eprt mbanking mandiri bantu	Pusat Bahasa (2)

Data pada nomor 6 merupakan data uji yang akan dihitung fungsi kosinus dalam kesamaan vektornya dengan vektor kelas yang sesuai. Dan data nomor 0 sampai 5 merupakan data latih. Data di atas akan diproses terlebih dahulu dengan preprocessing sehingga mendapatkan data di bawah ini.

Tabel 3.2 Hasil perhitungan vektor menggunakan TFIDF.

Term	D0	D1	D2	D3	D4	D5	D6
absensi	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
acara	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
account	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
adakan	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
adalah	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
yang	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
yang sudah	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
yang tersedia	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
sulit	0.000000	0.0	0.000000	0.000000	0.0	0.000000	0.543530
stop	0.000000	0.0	0.000000	0.000000	0.0	0.707107	0.000000
sulit	0.000000	0.0	0.000000	0.000000	0.0	0.000000	0.543530

Nilai bobot pada term di atas merupakan nilai yang sudah dinormalisasi. Setelah didapat bobot term pada setiap dokumen, akan dilakukan perhitungan nilai KNN vektor pada masing-masing kelas dengan cara menghitung bobot term yang ada di data latih pada masing-masing kelas.

4. Hasil dan Analisis

4.1 Pengujian Model Aplikasi Web Black Box

Pengujian pada aplikasi web ini menggunakan metode black box testing. Metode pengujian ini merupakan pengujian terhadap fungsionalitas sebuah aplikasi untuk menemukan bug atau gangguan pada aplikasi. Pengujian ini dilakukan dengan cara menguji semua fitur yang ada di dalam aplikasi.

Tabel 4.2 Pengujian aplikasi dengan metode *black box testing*.

Fitur	Kasus Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Login	Memasukkan username dan password, klik tombol login	Berhasil login dan masuk ke halaman utama	Sesuai	Normal
Register	Memasukkan data register, klik tombol register	Berhasil register akun	Sesuai	Normal
Masuk halaman input keluhan	Klik tombol <i>add complaint</i>	Berhasil masuk ke halaman input keluhan	Sesuai	Normal
Input keluhan	Memasukkan data keluhan, klik tombol tambah	Berhasil input keluhan, dan kembali ke halaman utama	Sesuai	Normal
Melihat riwayat	Klik tombol <i>history</i>	Berhasil masuk ke halaman riwayat	Sesuai	Normal
Memberi feedback	Memasukkan data feedback, klik tombol update	Berhasil memberi feedback	Sesuai	Normal

4.2 Analisis Akurasi

Analisis akurasi dilakukan untuk mengetahui performa dari model K-Nearest Neighbor. Hasil akurasi dari model ini didapat dari proses pengujian menggunakan model klasifikasi terhadap data uji. Data uji yang dipakai yaitu 10% dari keseluruhan dataset. Perhitungan nilai akurasi dapat dihitung dengan menggunakan persamaan di bawah ini..

$$akurasi = \frac{\text{jumlah klasifikasi yang benar}}{\text{jumlah data testing}} \times 100\% = akurasi\% \quad (4.1)$$

$$akurasi = \frac{121,68}{156} \times 100\% = 78 \%$$

Nilai akurasi yang didapat pada pengujian ini adalah 78% dengan jumlah data uji sebanyak 156 dokumen. Jumlah dataset dan keakuratan dataset sangat berpengaruh pada nilai akurasi suatu model. Semakin banyak dan akurat suatu dataset semakin tinggi nilai akurasi yang akan didapatkan.

4.3 Analisis *F1-Score*

Performa sebuah model *multiclass classification* tidak cukup dilihat dari nilai akurasi saja, namun perlu juga dilihat nilai *precision*, *recall*, dan *f1-score*. *F1-score* merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. Perhitungan *f1-score* dapat dihitung menggunakan persamaan di bawah ini.

$$F1 = \frac{2(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \quad (4.2)$$

Precision menggambarkan perbandingan antara rasio prediksi positif dengan keseluruhan data yang diprediksi positif. Sedangkan untuk *recall* menggambarkan perbandingan antara rasio prediksi *true* positif dengan keseluruhan data yang sebenarnya positif [10]. Hasil perhitungan dapat dilihat pada data di bawah ini.

Tabel 4.2 *Classification report* pada model klasifikasi.

Document	Precision	Recall	F1-score	Support
0	0.75	0.67	0.71	9
1	0.33	0.33	0.33	9
2	0.75	1.00	0.86	9
3	0.67	0.93	0.78	15
4	0.30	0.75	0.43	4
5	0.80	0.50	0.62	8
6	0.67	1.00	0.80	4
7	0.86	0.55	0.67	11
8	0.78	0.64	0.70	11
9	0.64	0.50	0.56	14
10	0.80	0.67	0.73	6
11	0.70	0.78	0.74	9
12	0.83	0.62	0.71	16
Macro avg	0.68	0.69	0.66	125

Precision, *recall*, dan *f1-score* didapat dari masing-masing kelas lalu dirata-ratakan sehingga mendapatkan hasil pada *macro-averaged*. Kolom *support* merupakan jumlah data uji pada masing-masing kelas. Secara representasi, jika *f1-score* mempunyai skor yang baik mengindikasikan bahwa model klasifikasi mempunyai *precision* dan *recall* yang baik. Hasil *f1-score* pada model ini adalah 67%. Pada tugas akhir ini digunakan *K* yang berlipat dari *K:3* sampai dengan *K:17* disini terjadi permainan dari nilai *K* agar mendapatkan tingkat akurasi yang lebih baik.

5. Kesimpulan

Berdasarkan hasil dari perancangan dan analisis yang telah dilakukan pada penelitian ini, dapat diambil kesimpulan yaitu :

1. Dengan menggunakan metode *K-Nearest Neighbor*, dapat meningkatkan efektifitas dalam penggunaan aplikasi web handling complaint..
2. Metode *K-Nearest Neighbor* mampu mengklasifikasikan keluhan sesuai dengan unit yang dituju, sehingga dapat mempercepat proses penanganan keluhan oleh staf unit bagian pada Telkom University.
3. Hasil performa yang didapat dari analisis akurasi mendapatkan nilai akurasi sebesar 65%. Dan hasil performa dari nilai *precision*, *recall*, dan *f1-score* mendapatkan hasil yang tidak jauh berbeda dengan nilai akurasi yaitu sebesar 78%. Dengan perbandingan antara data latih dan data uji adalah 90 : 10.

REFERENSI

- [1] S. Indriyani, S. Mardiana. "Pengaruh Penanganan Keluhan (Complaint Handling) Terhadap Kepercayaan Dan Komitmen Mahasiswa Pada Perguruan Tinggi Swasta Di Bandar Lampung". Jurnal Bisnis Darmajaya, Vol.2 No.01, 2016
- [2] S. Pongpaichet, P. Kormpho, P. Liawsomboon, N. Phongoen. "Smart Complaint Management System". Nakhon Pathom, Thailand: Mahidol University, 2018.
- [3] R. Fieldman, J. Sanger, The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, 2006.
- [4] J. Han, M. Kamber, J. Pei, Data Mining: Concepts And Techniques Third Edition. Morgan Kaufmann Publisher, 2012.
- [5] I. Surjandari, C. Megawati, A. Dhini, I.B.N.S. Hardaya, "Application of Text Mining for Classification of Textual Reports : A Study of Indonesia's National Complaint Handling System," presented at International Conference on Industrial Engineering and Operations Management Kuala Lumpur, Malaysia, March 8-10, 2016.
- [6] Tokunaga, Takenobu, Makoto, Iwayama, "Text Categorization based on Weighted Inverse Document Frequency," Advanced Research Laboratory, Hitachi Ltd, 1994.
- [7] 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013 KNN with TF-IDF Based Framework for Text Categorization Bruno Trstenjaka, Sasa Mikac, Dzenana Donkoc.
- [8] G. Panchal dan K. N. Pushpalatha, "A Local Binary Pattern Based Facial Expression Recognition using K- Nearest Neighbor (KNN) Search," International Journal of Engineering Research & Technology (IJERT), vol.6, no. 5, pp. 525-530, 2017.
- [9] A. Ismail, F. Sumarsono, Nuryana, T. Kurniawan, " Perancangan Website Data Karyawan Dengan Menggunakan PHP dan MySQL (Studi Kasus : STMIK Sumedang)," Jurnal Sistem Basis Data, 2019
- [10] L. Setiyani, S.T, M.Kom, Rekasaya Perangkat Lunak [Software Engineering]. Karawang: Jatayu Catra Internusa, 2018
- [11] Tri. A. Kurniawan, "Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik," Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), vol. 5.1 , p. 77-86, 2018
- [12] D. Putra, R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," Jurnal TEKNOIF, vol. 7.1, 2019
- [13] Tokunaga, Takenobu, Makoto, Iwayama, "Text Categorization based on Weighted Inverse Document
- [14] D. Putra, R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," Jurnal TEKNOIF, vol. 7.1, 2019