PROSIDING Seminar Nasional SISFOTEK ISSN 2597-3584 (Media Online)
Sistem Informasi dan Teknologi
website : www.seminar.iaii.or.id | email : seminar@iaii.or.id

Algoritma *Term Frequency – Inverse Document Frequency* (TF-IDF) dan K-Means *Clustering* Untuk Menentukan Kategori Dokumen

Ida Widaningrum[1], Dyah Mustikasari[2], Rizal Arifin[3], Siti Lathifah Tsaqila[4], Dwiyunia Fatmawati[5]
[1,2,5]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Ponorogo
[3]Program Studi Teknik Mesin, Fakultas Teknik, Universitas Muhammadiyah Ponorogo
[4]Program Studi Teknik Informatika, Fakultas Teknik Industri, Universitas Ahmad Dahlan
sweety.lathifahtsaqila@gmail.com

Abstract

The development of technology is speedy; one of the results is developing documents in research articles. Searching for documents in a repository will take a long time if they are not stored grouped by document category. One way to define document categories is clustering. The usefulness of document clustering, to make it easier to find documents by certain categories. The clustering process uses the Term Frequency - Inverse Document Frequency (TF-IDF) algorithm and K-Means. TF-IDF is used to find document weights, while K-Means is for the clustering process. The test documents or dataset were grouped as many as 93 documents, with various themes and document contents. The K-Means cluster quality assessment process results using the Silhouette score; the optimal number of clusters is 4 clusters. This is obtained by looking at the fluctuation in cluster size and thickness of the silhouette plot.

Keywords: document clustering, characteristics or categories, python, term frequency-inverse document frequency (tf-idf).

Abstrak

Perkembangan teknologi sangat pesat, salah satu akibatnya adalah berkembangnya dokumen berupa artikel hasil penelitian. Pencarian dokumen dalam suatu repositori, membutuhkan waktu lama apabila tidak disimpan dengan dikelompokkan berdasarkan kategori dokumen. Salah satu cara untuk menentukan kategori dokumen adalah Clustering. Kegunaan clustering atau pengelompokkan dokumen, untuk mempermudah pencarian dokumen menurut kategori tertentu. Proses clustering dalam penelitian ini menggunakan algoritma Term Frequency – Inverse Document Frequency (TF-IDF) dan K-Means. TF-IDF digunakan untuk mencari bobot dokumen, sedangkan K-Means adalah untuk proses clusteringnya. Dokumen uji yang dikelompokkan pada dataset sebanyak 93 dokumen, dengan berbagai karakteristik tema maupun isi dokumen. Hasil proses penilaian kualitas klaster K-Means menggunakan Silhouette score, jumlah klaster yang optimal adalah 4 klaster. Hal itu didapat dengan melihat fluktuasi ukuran klaster dan ketebalan plot siluet.

Kata kunci: *clustering* dokumen, karakteristik atau kategori, *k-means, phyton, term frequency-inverse document frequency (tf-idf).*

1. Pendahuluan

Perkembangan teknologi saat ini berkembang sangat pesat sehingga penyimpanan dokumen tidak harus berbentuk berkas, melainkan akan disimpan dengan bentuk file / *softcopy*. Dengan adanya penyimpanan dokumen berbetuk file ini penyebaran dokumen pada dunia teknologi sangat pesat dan setiap harinya terus mengalami perubahan dalam jumlah sangat besar. Pengolahan informasi dari banyak dokumen yang jumlahnya sangat besar itu tidak mudah. Oleh karena itu diperlukan sebuah metode yanng akan digunakan untuk mengelompokkan dokumen secara otomatis, sehingga memudahkan dalam pencarian informasi sesuai dengan kebutuhan.

Culstering merupakan salah satu metode *data mining*. *Clustering* dokumen adalah sebuah aktifitas pengelompokan beberapa dokumen secara otomatis dalam kategori tertentu yang memiliki kemiripan tema maupun isi dokumen tersebut. Salah satu metode *clustering* untuk mengelompokkan data yaitu *TF-IDF*. *Algoritma Term Frequency-Inverse Document Frequency (TF-IDF)* merupakan algoritma yang digunakan untuk menghitung bobot kata dengan cara mempertimbangkan ada berapa banyak kata muncul (frekuensi kata) dan berapa banyak kata tersebut ditemukan dalam dokumen. Oleh karena itu IDF dapat digunakan utuk mengontrol bobot dari kata. Sehingga ketika kata tersebut selalu ada dalam setiap file atau kalimat maka bias dikaakan bahwa kata tersebut tidak penting atau kata umum [1].

Beberapa teknik yang banyak digunakan pada klasifikasi dokumen teks sudah dilakukan penelitiannya diantaranya menggunakan Naive Bayes karena fakta

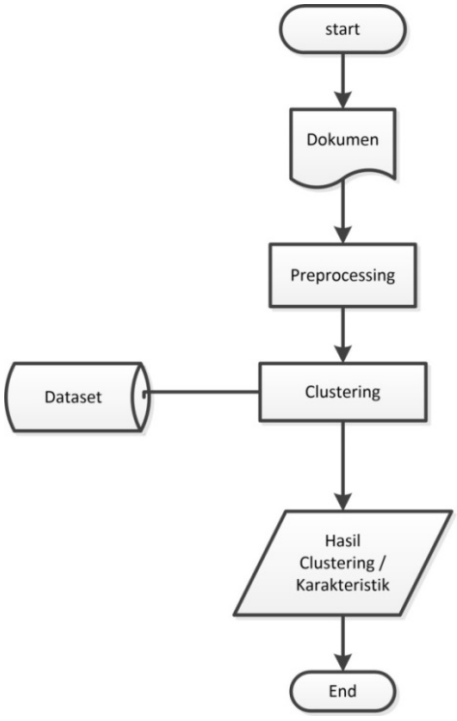
menarik, mudah diimplementasikan dan akurasinya [2], menyoroti kinerja Naive Bayes [3], dan penelitian [4]. Naive Bayes juga digunakan untuk pengkategorian dalam Bahasa Arab [5]. Penelitian di atas menggunakan *corpus* berbahasa Inggris atau bahasa asing.

Sedangkan penelitian yang menggunakan Bahasa Indonesia jumlahnya masih sangat sedikit. Beberapa diantaranya yaitu klasifikasi dokumen menggunakan Support Vector Machine (SVM) [6], teks menggunakan Naive Bayes [7], [8]. Klasifikasi berita menggunakan ontologi [9], algoritma single pass clustering [10]. [11], [12] melakukan klasifikasi artikel berita, , demikian juga [13] melakukan perbandingan antara TI-IDF dan Singular Value Decomposition untuk pemilihan fitur dan Naive Bayes dan Support Vector Machine untuk klasifikasi. Sedangkan [14] menggunakan stemmer confix stripping dan ants algorithms untuk klasifikasi.

Dengan adanya latar belakang tersebut, penulis mengambil judul yaitu "Metode Algoritma *Term Frequency-Inverse Document Frequency (TF-IDF) dan K-MeansClustering* Untuk Menentukan Kategori Dokumen". Batasan masalah yang akan digunakan dalam penulisan ini adalah, dokumen yang digunakan berupa abstrak dari jurnal atau artikel, perangkat lunak yang digunakan yaitu bahasa pemrograman *python.*

2. Metode Penelitian

Flowchart yang digunakan untuk clustering dokumen diperlihatkan pada gambar 1.



Gambar 1. *Flowchart* / diagram alur untuk *clustering* dokumen

Pada gambar1 terlihat bahwa ketika dokumen, yaitu dokumen uji yang akan diujikan untuk mengetahui bahwa dokumen tersebut masuk dalam kluster dataset mana, akan melalui tahap preprocessing. Preprocessing merupakan proses dimana dilakukannnya *converting* antara dokumen pdf ke .txt dengan penghilangan tanda baca dan proses penghilangan kata yang tidak penting.

Selanjutnya akan melalui proses clustering. Pada tahap ini dokumen akan diklusterkan, sehingga akan terlihat apakah dokumen tersebut masuk pada karakteristik mana yang ada di data set.

2.1. *Data Mining*

Data mining atau *knowledge discovery* adalah merupakan proses pengambilan informasi yang berasal dari sekumpulan data besar, informasi penting yang diambil tergantung dari kepentingannya. Pegambilan data dilakukan dengan berbagai cara dan metode, dan ini mrupakan kolaborasi dari berbagai metode yang berasal dari matematika, statistika, atau bahkan program computer (Artificial Intelligence) [15].

2.2. *Clustering*

clustering adalah metode pengelompokan yang mengatur dokumen teks tidak berurutan dalam jumlah besar menjadi sejumlah kecil kluster yang bermakna dan koheren, sehingga memberikan dasar untuk navigasi dan mekanisme penelusuran yang intuitif dan informative [16]–[18].

2.3. *Algoritma Term Frequency – Inverse Document Frequency (TF-IDF) Algoritma Term Frequency – Inverse Document Frequency (TF-IDF)*

Algoritma *Term Frequency – Inverse Document Frequency (TF-IDF)* merupakan bidang information retrieval [1], [19]–[21], menghitung seberapa relevan sebuah kata dalam rangkaian korpus dengan sebuah teks. Tf-idf adalah salah satu metrik terbaik untuk menentukan seberapa signifikan suatu istilah terhadap teks dalam rangkaian atau korpus. tf-idf adalah sistem pembobotan yang memberikan bobot pada setiap kata dalam dokumen berdasarkan frekuensi term (tf) dan frekuensi resiprokal dokumen (tf) (idf). Kata-kata dengan skor bobot yang lebih tinggi dianggap lebih signifikan:

$$tf - idf(t,d) = tf(t,d) * idf(t) \quad (1)$$

Term Frequency (TF) menyatakan jumlah kata t yang muncul dalam sebuah dokumen d. Pendekatan paling sederhana dari konsep ini adalah dengan menyatakan bobot suatu kata t sebagai jumlah kemunculannya pada dokumen d.

$$tf(t,d) = jumlah\ t\ dalam\ d\ /jumlah\ kata\ pada\ d \quad (2)$$

Document Frequency menyatakan jumlah dokumen yang memuat kata t.

$$df(t) = \text{jumlah dokumen yang memuat } term \text{ t}$$

Inverse Document Frequency (IDF) untuk menguji seberapa relevan kata tersebut. Tujuannya adalah untuk menemukan proporsi dokumen dalam korpus yang mengandung istilah atau term tersebut. Kata yang unik akan menadapatkan nilai kepentingan yang lebih tinggi daripada kata yang umum.

$$idf(t) = log\left({}^{N}/_{df(t)}\right) \quad (3)$$

Keterangan:
$idf(t)$ = *inverse document frequency*
N = jumlah keseluruhan dokumen

2.4. *K-Means Clustering*

K-Means adalah algoritma pengelompokan yang bertujuan untuk meminimalkan jarak rata-rata kuadrat Euclidean dokumen dari pusat kalster, dimana pusat klaster didefinisikan sebagai mean atau centroid [22]. K-Means dilakukan dengan cara penentuan jumlah klaster yang ingin dibentuk, membangkitkan secara acak titik pusat klaster (centroid) awal, menghitung jarak dari setiap dokumen / data ke masing-masing centroid yang dibuat tadi menggunakan Euclidean Distance, mengelompokkan data berdasar jarak terdekat ke pusat centroid, menentukan centroid baru dengan menghitung nilai rata-rata jarak data pada centroid tadi, demikian seterusnya sampai didapat centroid yang benar-benar pas [23], [24].

$$Centroid\ \vec{\mu}(\omega) = \frac{1}{|\omega|}\sum_{\vec{x}\epsilon\omega}\vec{x} \cdots\cdots\cdots \quad (4)$$

Dimana ω adalah klaster, dan x adalah dokumen.

2.5. *Silhouette score*

Silhouette score atau *Silhouette coefficient* adalah metric yang dibangun untuk menilai atau mengukur akurasi dari suatu teknik pengelompokkan data [25], [26]. Metrik ini dinyatakan dalam bentuk tampilan grafis. Setiap kalster dinyatakan dalam siluet, yang didasarkan pada perbandingan kerapatan dan pemisahnya. Siluet ini menunjukkan keberadaan objek didalam clusterr. Klastering dinyatakan dengan plot yang terdiri dari beberapa siluet. Plot ini menampilkan apresiasi kualitas relatif kluster dan gambaran umum konfigurasi data. Lebar siluet rata-rata memberikan evaluasi validitas pengelompokan, dan dapat digunakan untuk memilih jumlah kluster yang sesuai [25].

2.6. Bahasa Pemrograman Python

Bahasa pemrograman Python adalah memantapkan dirinya sebagai salah satu bahasa yang paling populer untuk komputasi ilmiah. Berkat sifat interaktif tingkat tinggi dan ekosistem perpustakaan ilmiah yang semakin matang, ini merupakan pilihan yang menarik untuk pengembangan algoritmik dan analisis data eksplorasi [27], [28].

Python merupakan merupaka bahasa pemrograman yang sedang booming saat ini. Banyak kalangan mempelajarinya, karena kemampuan dan kemudahan yang ditawarkannya. Pytjon bias digunakan dalam hamper semua platform, dan juga mempunyai library yang berisi modul yang siap bias digunakan. Selain itu juga python bias mengakomodasi bahasa pemrograman lainnya [29], [30].

3. Hasil dan Pembahasan

Langkah-langkah pengelompokan dokumen menggunakan bahasa pemrograman python yaitu setiap dokumen uji (dataset) akan di kelompokkan sesuai karakteristik pada kluster data set yang dibuat. Setiap dokumen yang diuji akan melewati proses *converting* terdahulu. Proses dilakukan dengan memasukkan dataset (diperlihatkan pada Gambar 2), case folding, tokenisasi, dan stemming dan menghilangkan kata tidak berarti dengan stopword (diperlihatkan pada Gambar 3), proses vectorizer, baru kemudian masuk K-Means. Selanjutnya proses penentuan klasternya dan centroid (Gambar 4). Dari proses, bisa juga dilihat sebaran datasetnya seperti diperlihatkan pada gambar 5.

```
df['Abstrak']

0     Pada era teknologi informasi seperti saat ini,...
1     Penelitian atau tugas akhir merupakan syarat k...
2     Pemerintahan yang baik adalah pemerintahan yan...
3     Universitas medan area memiliki dosen dengan j...
4     Volume berita elektronik berbahasa Indonesia y...
                            ...
93    Tulisan ini menganalisis tentang implementasi ...
94    Penelitian ini menganalisis cerpen berjudul Ja...
95    Latar belakang penelitian ini adalah kesuksesa...
96    Perekrutan perangkat desa secara selektif dan ...
97    Korupsi dan demokrasi merupakan dua hal yang t...
Name: Abstrak, Length: 98, dtype: object
```

Gambar 2. Dataset

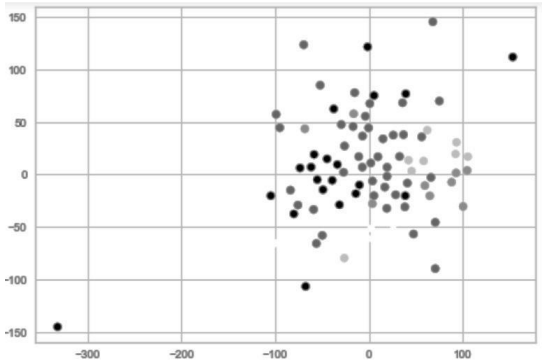
```
print(filtering2) #mencetak hasl filtering

['era', 'teknologi', 'informasi', 'dokumen', 'tek
jurnal', 'cenderung', 'tersimpan', 'format', 'dig
', 'penyimpanan', 'terlalu', 'banyaknya', 'dokume
', 'komputer', 'membuat', 'pencarian', 'informasi
'kesulitan', 'pencarian', 'informasi', 'sesuai',
li', 'menjadi', 'penghambat', 'proses', 'pembelaj
', 'data', 'clustering', 'menjadi', 'salah', 'sat
anisasi', 'dokumen-dokumen', 'teks', 'digital', '
metode', 'k-means', 'clustering', 'dipadukan', 'p
```

```
data : data
terlalu : terlalu
banyak : banyak
penelitian : teliti
tugas : tugas
akhir : akhir
merupakan : rupa
syarat : syarat
kelulusan : kelulus
mahasiswa : mahasiswa
tahun : tahun
penelitian : teliti
menjadi : jadi
bertambah : tambah
memungkinkan : mungkin
mahasiswa : mahasiswa
mengambil : ambil
```

Gambar 3. Proses tokenize dan stemming

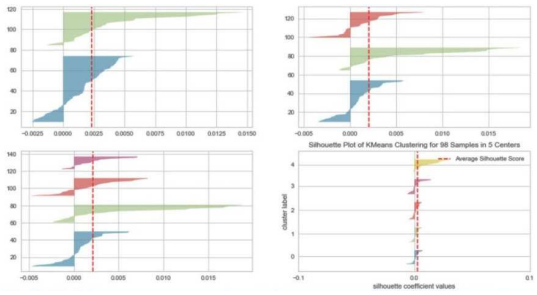
```
centroids = km.cluster_centers_
centroids

array([[0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.00388374, 0.00226168, 0.00226168, ..., 0.00171425, 0.00171425,
        0.00171425],
       ...,
       [0.0031044 , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ]])
```

Gambar 3. Centroid yang terbentuk



Gambar 4. Sebaran dataset

Penghitungan Akurasi



Gambar 5. Silhouete plot untuk klastering K-Means

Akurasi dihitung dengan menggunakan Silhouete score dengan hasil sebesar 0,00191896. Dari hasil score ini, dapat dilihat bahwa dokumen-dokumen uji (dataset) temanya hampir bersinggungan.

4. Kesimpulan

Kesimpulan yang didapat dari analisa *clustering* dokumen ini yaitu dokumen uji yang dikelompokkan menggunakan bahasa pemrograman python ini akan mengelompok pada data set sesuai karakteristik tema maupun isi dokumen. Dataset yang digunakan sebanya 98 dan setelah diuji menggunakan K-Means, didapat hasilnya adalah klaster yang terbentuk sebanyak 4 klaster. Hasil ini bias dilihat pada grafik Silhouete plot yang menyatakan siluet dari data berdasarkan kerapatan dan jarak antaranya.

Daftar Rujukan

[1] S. Andayani and A. Ryansyah, "Implementasi Algoritma TF-IDF Pada Pengukuran Kesamaan Dokumen," *JuSiTik J. Sist. dan Teknol. Inf. Komun.*, vol. 1, no. 1, p. 53, 2017, doi: 10.32524/jusitik.v1i1.218.

[2] K. A. Vidhya and G. Aghila, "A Survey of Naive Bayes Machine Learning approach in Text Document Classification," *Int. J. Comput. Sci. Inf. Secur.*, vol. 7, no. 2, pp. 206–211, 2010.

[3] S. L. Ting, W. H. Ip, and A. H. C. Tsang, "Is Naive Bayes a good classifier for document classification," *Int. J. Softw. Eng. Its Appl.*, vol. 5, no. 3, pp. 37–46, 2011.

[4] E. Frank and R. R. Bouckaert, "Naive bayes for text classification with unbalanced classes," in *European Conference on Principles of Data Mining and Knowledge Discovery*, 2006, pp. 503–510.

[5] M. El Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic document categorization based on the Naïve Bayes algorithm," in *proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, 2004, pp. 51–58.

[6] N. I. Widiastuti, E. Rainarli, and K. E. Dewi, "Peringkasan dan Support Vector Machine pada Klasifikasi Dokumen," *J. Infotel*, vol. 9, no. 4, pp. 416–421, 2017.

[7] J. Samodra, S. Sumpeno, and M. Hariadi, "Klasifikasi Dokumen Teks Berbahasa Indonesia dengan Menggunakan Naïve Bayes," *Semin. Nas. Electr. INFORMATICS, IT'S Educ.*, pp. 1–4, 2009.

[8] N. M. A. Lestari, I. K. G. D. Putra, and A. A. K. A. Cahyawan, "Personality types classification for indonesian text in partners searching website using naïve bayes methods," *Int. J. Comput. Sci. Issues*, vol. 10, no. 1, p. 1, 2013.

[9] H. Februariyanti and E. Zuliarso, "Klasifikasi dokumen berita teks bahasa indonesia menggunakan ontologi," *Dinamik*, vol. 17, no. 1, 2012.

[10] A. Z. Arifin and A. N. Setiono, "Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering," in *Prosiding Seminar on Intelligent Technology and its Applications (SITIA), Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya*, 2002, pp. 29–39, doi: 10.1109/ICODSE.2014.7062678.

[11] A. A. Hakim, A. Erwin, K. I. Eng, M. Galinium, and W. Muliady, "Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach," in *2014 6th international conference on information technology and electrical engineering (ICITEE)*, 2014, pp. 1–4.

[12] A. D. Asy'arie and A. W. Pribadi, "Automatic news articles classification in indonesian language by using naive bayes classifier method," in *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, 2009, pp. 658–662.

[13] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, and O. Rusli, "News article text classification in Indonesian language," *Procedia Comput. Sci.*, vol. 116, pp. 137–143, 2017.

[14] A. Z. Arifin, I. Mahendra, and H. T. Ciptaningtyas, "Enhanced confix stripping stemmer and ants algorithm for classifying news document in indonesian language," in *The International Conference on Information & Communication Technology and Systems*, 2009, vol. 5, pp. 149–158.

[15] D. J. Hand, H. Mannila, and P. Smyth, "Principles of data mining (adaptive computation and machine learning)," *Publ. A Bradford Book, 2001.–584 p*, 2001.

[16] A. Huang, "Similarity measures for text document clustering," in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, 2008, vol. 4, pp. 9–56.

[17] P. V. Amoli and O. S. Sh, "Scientific documents clustering based on text summarization," *Int. J. Electr. Comput. Eng.*, vol. 5, no. 4, p. 782, 2015.

[18] J. L. Neto, A. D. Santos, C. A. A. Kaestner, N. Alexandre, and D. Santos, "Document clustering and text summarization," 2000.

[19] L. Havrlant and V. Kreinovich, "A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation)," *Int. J. Gen. Syst.*, vol. 46, no. 1, pp. 27–36, 2017.

[20] T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.," Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[21] G. Salton and C. Buckley, "Term-weighting approaches in

automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988.
[22] C. D. Manning, P. Raghavan, and H. Schultze, *Introduction to Information Retrieval*. Cambridge University Press, 2009.
[23] R. Handoyo, R. Mangkudjaja, and S. M. Nasution, "Perbandingan metode clustering menggunakan metode Single Linkage dan K-means pada Pengelompokan Dokumen," *J. Sifo Mikroskil*, vol. 15, no. 2, pp. 73–82, 2014.
[24] R. Muliono and Z. Sembiring, "Data Mining Clustering Menggunakan Algoritma K-Means Untuk Klasterisasi Tingkat Tridarma Pengajaran Dosen," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 4, no. 2, pp. 272–279, 2019.
[25] R. Lletı, M. C. Ortiz, L. A. Sarabia, and M. S. Sánchez, "Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes," *Anal. Chim. Acta*, vol. 515, no. 1, pp. 87–100, 2004.
[26] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.
[27] R. Garreta and G. Moncecchi, *Learning scikit-learn: machine learning in python*. Packt Publishing Ltd, 2013.
[28] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[29] D. Rosmala and G. D. L, "Pembangunan Website Content Monitoring System Menggunakan Difflib Python," *J. Inform.*, vol. 4, no. 1, pp. 57–68, 2012.
[30] L. Buitinck *et al.*, "API design for machine learning software: experiences from the scikit-learn project," *arXiv Prepr. arXiv1309.0238*, 2013.