

KLASIFIKASI AYAT AL-QURAN TERJEMAHAN MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE* DAN *K-NEAREST NEIGHBORS*

Aan Saputra¹, Iman Zuhri Yadi²

Fakultas Ilmu Komputer, Universitas Bina Darma

Email: aansaputra1113@gmail.com¹, ilmanzuhriyadi@binadarma.ac.id²

ABSTRAK

Artike Al-Quran merupakan kitab suci umat islam yang mengatur seluruh aspek kehidupan baik secara individu maupun bermasyarakat. Bagi umat islam, sudah seharusnya mereka mejadikan Al-Quran sebagai pedoman hidup. Minimal hal mendasar yang harus dipelajari dalam Al-Quran yaitu Larangan, perintah dan informasi yang terdapat di dalam Al-Quran. Namun, saat ini masih jarang pengelompokkan Al-Quran didalam klasifikasi tersebut, padahal Banyak metode pengklasifikasian yang dapat diterapkan pada Ayat Al-Quran terjemahan. Beberapa metode yang bisa digunakan seperti *Support vector machine* dan *k-nearest neighbors*. Pada penelitian ini, penulis menggunakan kedua metode tersebut untuk menampilkan perbandingan tingkat akurasi klasifikasi pada Ayat Al-Quran terjemahan. Didalam melakukan klasifikasi teks, sering terdapat kata-kata yang tidak penting. Sehingga mempengaruhi proses klasifikasi. Oleh karena itu, untuk mengatasi permasalahan tersebut, penulis menggunakan proses *stemming*, *tokenizing*, dan *stopword*. Hasil perbandingan dari kedua metode ini menunjukkan bahwa Metode KNN ($k=3,6,9$) memberikan hasil yang lebih baik daripada metode SVM *linear*.

Kata kunci: Al-Quran, klasifikasi, *Support vector machine*, *k-nearest neighbors*, *stemming*, *tokenizing*, dan *stopword*.

ABSTRACT

Al-Quran is the holy book of Muslims that fosters aspects of life both individually and socially. For Muslims, they should make the Koran a new life. At least the basic things that must be studied in the Al-Quran are prohibitions, commands and information contained in the Al-Quran. However, nowadays it is still rare to group Al-Quran into this classification, even though there are many classification methods that can be applied to translated Al-Quran Verses. Several methods can be used such as Support vector machine and k-nearest neighbor. In this study, the authors used these two methods to present a comparison of the level of classification in the translated Al-Quran Verse. In doing text classification, there are often words that are not important. So that affects the classification. Therefore, to solve this problem, the author uses stemming, tokenizing, and stopwords processes. The results of the comparison of these two methods show that the KNN method ($k = 3,6,9$) gives better results than the linear SVM method.

Keyword: Al-Quran, Classification, *Support vector machine*, *k-nearest neighbors*, *stemming*, *tokenizing*, dan *stopword*

1 PENDAHULUAN

Al-Quran adalah kitab suci yang diturunkan oleh ALLAH Subahanu Wa ta'ala kepada hambanya yang mulia, nabi Muhammad *Shallallahu 'alaihi wasallam* melalui malaikat jibril *alaihis*

salam, sebagai pedoman hidup bagi seluruh umat manusia dan jin. Al-Quran terdiri dari 30 juz, 114 surat, dan 6236 ayat. Setiap juz memiliki jumlah lembaran yang sama yaitu 10 lembar pada setiap juz. Adapun setiap surat memiliki jumlah ayat yang berbeda, tergantung tempat turun dan tema suratnya. Menurut Ahmad sahidin dkk, yang dilansir oleh islampos, menyatakan bahwa surat yang diturunkan di Mekkah termasuk dalam surat *makkiyah*, biasanya ditandai dengan ayat yang pendek dan membahas masalah akidah. Sedangkan surat yang diturunkan di madinah termasuk dalam surat *madaniyyah*, biasanya ditandai dengan ayat yang panjang dan membahas masalah hukum.

Berdasarkan pembagian Al-Quran dalam setiap juz, ada empat juz Al-Quran yang membahas tentang surat *makkiyah* dan *madaniyyah* yaitu juz 1 membahas tentang surat *madaniyyah* dan juz 29 membahas tentang surat *makkiyyah*. Dengan adanya pembagian surat alquran berdasarkan tempat turunnya ayat yaitu mekkah dan madinah, maka diperlukan pengklasifikasian ayat alquran, agar mempermudah umat manusia dalam mempelajari dan menjadikan al-quran sebagai pedoman hidup untuk menjadi pribadi terbaik. Karena, Rasulullah *Shallallahu 'alaihi wasallam* pernah bersabda “sebaik-baik kalian adalah orang yang belajar Al-Quran dan mengajarkannya” (HR. Bukhari)

Pesatnya perkembangan teknologi saat ini, banyak lembaga bidang AL-Quran yang menerbitkan Al-quran terjemahan dengan menambahkan klasifikasi ayat AL-Quran dibagian lampiran Al-Quran yang akan dicetak. Dalam Al-Quran terjemahan yang publikasikan oleh penerbit syaamil quran cordva, melampirkan ayat-ayat Al-Quran terdiri dari 15 klasifikasi. Selain itu, Mohan [1] juga mengumpulkan ayat-ayat Al-Quran berupa larangan dan perintah.

Berdasarkan dari klasifikasi tersebut, menegaskan bahwa pengklasifikasian Al-Quran sangat penting didalam kehidupan sehari-hari. Namun, pengujian akurasi klasifikasi ayat Al-Quran dibidang teknologi, masih sangat minim. Terutama yang membahas tiga aspek utama dalam kehidupan manusia yaitu larangan, perintah dan informasi.

Banyak cara dalam melakukan klasifikasi terhadap terjemahan ayat Al-Quran, untuk itu penting melakukan pengujian dengan beberapa metode klasifikasi. Agar bisa menampilkan metode yang tepat dalam menentukan hasil yang akurat terhadap klasifikasi ayat Al-Quran terjemahan. Oleh karena itu, penulis melakukan pengujian terhadap terhadap dua metode klasifikasi yaitu *Support Vector Mining* Dan *K-Nearest Neighbors*.

Menurut Pratiwi, dkk mengatakan bahwa metode SVM dengan fungsi kernel linier dapat memberikan kualitas hasil klasifikasi lebih baik dibandingkan dengan menggunakan metode KNN [2]. Sedangkan, (Aulianita, 2016) mengatakan bahwa k-NN adalah metode klasifikasi dengan akurasi terbaik dibandingkan dengan metode SVM [3].

Diharapkan dengan penelitian ini, dapat menentukan metode terbaik dalam melakukan klasifikasi terhadap ayat Al-quran terjemahan.

1.1 Al-Quran

Menurut Dauly Al-Quran adalah kitab suci yang diturunkan oleh ALLAH Subhaanahu Wa Ta'ala kepada Nabi Muhammad *Shallallahu 'alaihi wasallam* melalui malaikat jibril. Al-Quran tidak hanya diturunkan untuk umat manusia, tetapi juga untuk seluruh alam semesta. Sehingga, Al-Quran menjadi mukjizat terbesarnya Nabi Muhammad *Shallallahu 'alaihi wasallam* [4].

1.2 Text Mining

Menurut Rahmawati dkk. *text mining* tidak jauh berbeda dengan data mining, hanya saja dalam proses inputnya *text mining* berupa file data berupa dokumen. Dokumen *text mining* bisa dalam format apapun seperti pdf, text, excel, csv dan sebagainya. *Text mining* bertujuan untuk mengubah teks yang awalnya tidak terstruktur menjadi sumber informasi yang berharga dan bermanfaat [5].

Menurut Asiyah di era sekarang *text mining* telah dilirik diberbagai bidang yaitu akademik, media, aplikasi, pengembangan perangkat lunak, perbankan, keamanan, pemerintahan, medis, keuangan dan pemasaran. Text mining dalam studi kasus, harus dilakukan berdasarkan prosedur analisis. Adapun langkah awal sebelum analisis yaitu melakukan pre-processing text seperti

pengubahan kata dasar(*stemming*), menghilangkan kata (*stopword*) dan memisahkan kata (*tokenizing*), pembobotan kata (*term weighting*). Setelah itu, dapat dilakukan proses analisis teks [6].

Pada umumnya *Text mining* berfungsi untuk menemukan *aturan* baru melalui algoritma asosiasi, pengelompokan, peringkat, dan kategori. Diantaranya yang paling sering digunakan yaitu algoritma pengelompokan. Adapun pengelompokan terdiri dari dua metode yaitu klasifikasi dan klusterisasi.

Menurut Darujati & Gumelar klasifikasi teks adalah suatu proses yang mengelompokkan label(kelas) dari suatu dokumen berdasarkan kelas yang sudah diketahui sebelumnya atau sering disebut pengelompokan *supervised*. Sedangkan, klusterisasi teks adalah suatu proses analisis yang digunakan untuk menemukan struktur kelompok yang belum diketahui dari sekumpulan dokumen atau sering disebut pengelompokan *unsupervised* [7].

Berdasarkan pengertian diatas, dapat diketahui bahwa klasifikasi memiliki proses yang lebih mudah, dikarenakan adanya target yang akan dianalisis.

1.3 Klasifikasi

Menurut Gorunescu (dalam Khalidy, t.t.) Klasifikasi adalah suatu proses yang dilakukan dengan mengelompokkan suatu objek sesuai dengan kategori yang berkaitan dengan objek tersebut [8]. Sedangkan menurut Andriani (juga dalam Khalidy, t.t.) Klasifikasi adalah suatu model baru yang digunakan untuk membedakan suatu kelas dengan cara memprediksi kelas objek yang labelnya belum diketahui.

1.4 Pre Processing Text

Sebelum melakukan proses klasifikasi teks, ada tahapan yang kerjakan terlebih dahulu yaitu *pre processing text*. Hal ini bertujuan agar klasifikasi teks mendapatkan kinerja yang optimal. Berikut tahapan yang dilakukan pada *pre processing text*.

- Stemming* yaitu proses untuk mengembalikan kata dasar suatu kalimat, dengan cara menghilangkan imbuhan seperti awalan (men, ber, di), sisipan(el, er, em), akhiran(kan, I, nya) dan konfiks(ke-an, per-an, ber-an).
- Stopwords*, yaitu kosakata yang tidak termasuk kata unik dalam suatu dokumen atau tidak mengandung informasi [9]. Contohnya seperti kata “pada”, “yang”, “itu”, dan sebagainya.
- Tokenizing*, yaitu proses pemecahan suatu kalimat menjadi kata yang terpisah. Contohnya “segala puji bagi ALLAH” menjadi “[segala], [puji], [bagi], [ALLAH]”

1.5 Term Frequency Inverse Document Frequency(TF-IDF)

Menurut Jamhari dkk. *tf-idf* adalah pembobotan kata yang dikerjakan setelah dokumen sudah melalui tahap *pre processing text*. Proses yang digunakan untuk mencari hasil pembobotan yaitu dengan mengkalikan nilai antara *term frequency (tf)* dan *inverse document frequency (idf)* [10]. Langkah selanjutnya, untuk melihat keseluruhan hasil pembobotan yaitu kalikan nilai *tf* dengan jumlah pada nilai *idf*.n. Berikut rumus untuk menentukan nilai pembobotan :

$$w_{ij} = tf_{ij} \times idf_j$$
$$w_{IJ} = tf_{IJ} \times \log(N/df_j) \dots(1)$$

Keterangan :

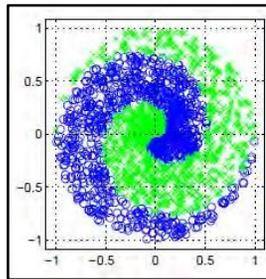
- W_{ij} = bobot dari kata i pada dokumen ke j
 tf_{ij} = jumlah kemunculan kata i pada dokumen j
 df_j = jumlah dokumen j yang mengandung kata i
 N = jumlah keseluruhan dokumen

Proses *tf-idf* berfungsi untuk mengubah kata menjadi word vector, agar dokumen bisa dianalisis menggunakan metode SVM dan KNN.

1.6 Algoritma Support Vector Machine

Menurut Pratama, dkk (dalam setiyono dan F, Ardede) tidak ada metode yang paling tepat dalam pengklasifikasiannya selain SVM. Pada awalnya, metode SVM hanya memisahkan dua kelas. Tetapi, seiring kompleksnya permasalahan menyebabkan SVM tidak bisa mengatasi masalah yang memiliki klasifikasi lebih dari dua kelas. Agar masalah tersebut bisa diatasi, maka metode SVM kemudian mengembangkan fungsi kernel yang bisa mengelompokkan data lebih dari dua kelas [11].

Menurut Asiyah Metode *support vector machine* memiliki banyak cara dalam pengklasifikasiannya yaitu SVM pada *linearly separable data*, SVM pada *non linearly separable data*, metode kernel, dan *multi class SVM*. Metode kernel memisahkan data dengan bentuk yang tidak harus linear, misalnya bisa berbentuk spiral. Contohnya, dapat kita lihat pada gambar 1 berikut.

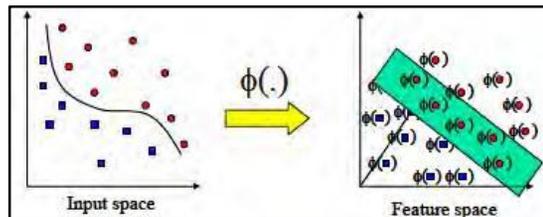


Gambar 1. Bentuk data spiral (nonlinear) [12]

Data x di yang x yang dimasukkan keruang nonlinear menggunakan fungsi kernel $x_k \rightarrow \phi(x_k)$. Sehingga $w = \sum_{i=1}^{ns} \alpha_i y_i \phi(x_i)$ output yang dihasilkan yaitu sebagai berikut :

$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i \phi(x_i) \phi(x_d) + b$$

Fungsi kernel pada kenyataannya memiliki hasil yang lebih dari nilai vektor awal. Keadaan ini akan memberikan ruang yang besar, hingga memiliki jumlah fungsi yang tak terhingga.



Gambar 2. Transformasi ruang fungsi [12]

Berdasarkan gambar 2, akan menyebabkan sulit melihat ruang fungsi yang tepat. Adapun cara mengatasi permasalahan ini yaitu dengan mengetahui nilai fungsi dari data training yang dihasilkan. Maka persamaannya bisa dilihat seperti berikut :

$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i K(x_i, x_d) + b$$

Salah satu syarat suatu fungsi menjadi fungsi kernel yaitu memenuhi teorema Mercer yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat positive semi-definite. Adapun Fungsi kernel yang yang populer pada metode SVM yaitu :

1. Kernel Linier

$$K(\mathbf{x}_i, \mathbf{x}) = \mathbf{x}_i^T \mathbf{x}$$

2. Kernel Polynomial

$$K(\mathbf{x}_i, \mathbf{x}) = (\gamma \mathbf{x}_i^T \mathbf{x} + r)^p, \gamma > 0$$

3. Kernel Radial Basis Function (RBF)

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2\right)$$

4. Sigmoid Kernel

$$K(\mathbf{x}_i, \mathbf{x}) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x} + r)$$

Dalam proses klasifikasi, hampir semua peneliti menggunakan ketiga jenis kernel pertama seperti membandingkan akurasi klasifikasi antara kernel linear dan RBF, dengan hasil kernel linear memberikan hasil yang terbaik [14].

1.7 Algoritma *K-Nearest Neighbors*(KNN)

Menurut J. Han, dkk (dalam Rivki & Bachtiar, 2017) Algoritma KNN adalah suatu metode klasifikasi yang dihasilkan berdasarkan jarak objek terdekat dan jarak terdekat ditentukan dengan nilai k . jarak k minimal lebih dari satu dan tidak boleh melebihi data *training*. Jarak terdekat pada data training bisa dihitung menggunakan cosine similarity. Karena, Cosine similarity adalah metode yang berfungsi untuk melihat kemiripan data dalam suatu dokumen [15].

Sedangkan Menurut(Prasetyo, 2012) KNN adalah metode klasifikasi yang menentukan jarak terdekat antara satu dokumen dengan dokumen yang lain.[16] Penentuan jarak terdekat pada klasifikasi KNN yang paling sering digunakan yaitu jarak Euclidean, karena menghasilkan kinerja yang sangat baik dan akurat [17].

Jarak Euclidean memiliki konsep dengan memperlakukan semua peubah dengan memberikan peran yang sama dalam setiap jarak dengan bebas. Berikut merupakan rumus pada jarak euclidean [18] :

$$d(x_i, x_j) = \sqrt{\sum_{n=1}^p (x_{ip} - x_{jp})^2}$$

Dengan :

- x_{ip} = data *testing* ke- i pada variabel ke- p
- x_{jp} = data *training* ke- j pada variabel ke- p
- $d(x_i, x_j)$ = jarak *Euclidean*
- p = dimensi data bebas variable

Ada beberapa langkah yang digunakan untuk menentukan hasil klasifikasi pada metode KNN yaitu [19] :

- 1) Menentukan nilai k (jarak terdekat)
- 2) Menghitung jarak eulid pada setiap objek terhadap sample yang dilakukan
- 3) Urutkan setiap objek kedalam kelompok jarak *Euclid*
- 4) Kelompokkan kategori Y (klasifikasi KNN)
- 5) Lihat hasil dari nilai prediksi pada *query instance*

1.8 Pengukuran Performa

Pengukuran performa berfungsi untuk menampilkan hasil dari proses klasifikasi. Banyak cara yang bisa digunakan untuk mengukur kinerja klasifikasi dan cara yang paling populer digunakan yaitu dengan menampilkan nilai akurasi total, *f1-score*, *precision* dan *recall* [20].

Total akurasi adalah presentase dari keseluruhan dokumen yang terprediksi secara akurat pada proses klasifikasi dokumen. *Precision* memprediksi nilai yang paling relevan dari dokumen yang sudah diprediksi, misalnya presentase mahasiswa yang benar-benar tidak lulus, dari presentase prediksi mahasiswa yang tidak lulus. *Recall* memprediksi dokumen berdasarkan presentase dokumen yang telah diklasifikasi, misalnya nilai presentase terhadap mahasiswa yang tidak lulus. Sedangkan *f1-score* yaitu hasil perbandingan rata-rata dari *precision* dan *recall* yang dibobotkan. Tetapi, untuk pengukuran klasifikasi test mining, biasanya peneliti sering menggunakan nilai akurasi sebagai tolak ukur utama [21]. Contohnya, Ketepatan dalam klasifikasi ditentukan untuk melihat hasil akurasi yang bisa dirunning pada *function confusion matrix*. Berikut contoh tabel 1 yang memprediksi ketepatan *confusion matrix*.

Tabel 1. *Confusion Matrix* [22]

		Prediksi		
		label 1	label 2	label 3
Asli	label 1	$F_{1,1}$	$F_{1,2}$	$F_{1,3}$
	label 2	$F_{2,1}$	$F_{2,2}$	$F_{2,3}$
	label 3	$F_{3,1}$	$F_{3,2}$	$F_{3,3}$

Berdasarkan tabel *confusion matrix* di atas, berikut rumus untuk menentukan hasil dari akurasi total, precision, recall, dan f1-score.

1) Rumus Nilai Akurasi Total

$$akurasi\ total = \frac{F_{11} + F_{22} + F_{33} + F_{44} + F_{55}}{F_{11} + F_{12} + F_{13} + F_{14} + F_{15} + \dots + F_{51} + F_{52} + F_{53} + F_{54} + F_{55}}$$

Persamaan di atas untuk melihat ketepatan klasifikasi terhadap akurasi total.

2) Rumus Dari Nilai Recall

$$recall = \frac{F_{aa}}{\sum_{b=1}^B F_{ab}}$$

Keterangan : a=1,2,3,...,A
 b=1,2,3,...,B

$$recall\ kategori\ 1 = \frac{F_{11}}{F_{11} + F_{12} + F_{13} + F_{14} + F_{15}}$$

Persamaan diatas untuk melihat ketepatan klasifikasi data aktual yang berasal dari data prediksi kelas ke-1.

3) Rumus dari Nilai Precision

$$precision = \frac{F_{aa}}{\sum_{a=1}^A F_{ab}}$$

$$precision\ kategori\ 1 = \frac{F_{11}}{F_{11} + F_{21} + F_{31} + F_{41} + F_{51}}$$

Persamaan diatas untuk melihat ketepatan klasifikasi data aktual yang berasal dari data prediksi kelas ke-1.

4) Rumus dari f1-score

$$F = \frac{2 \times recall \times precision}{recall + precision}$$

Dari rumus diatas, dapat diketahui bahwa *F1-score* merupakan perbandingan dari *recall* dan *precision* untuk mengukur performa menyeluruh dari klasifikasi [20].

1.9 Anaconda

Anaconda adalah aplikasi yang mendistribusikan bahasa pemrograman python dengan kualitas kinerja yang sangat baik. Aplikasi *Anaconda* memiliki tampilan yang menarik, dan menyediakan banyak library yang bisa digunakan untuk menjalankan pemrograman *python* seperti *jupyter notebook*, *jupyter lab*, dan *spyder*. Anaconda juga menyediakan *platform* yang *user friendly* sehingga bisa diakses di berbagai *platform* seperti *linux*, *mac*, dan *windows*. Sehingga hal tersebut sangat memanjakan pengguna [23].

1.10 Bahasa Pemrograman Python

Menurut Trisanti (dalam Perkasa dkk., 2014) Python adalah bahasa pemrograman yang perangkatnya bebas didistribusikan dengan layanan yang lengkap seperti layanan antar muka, source code, dan basis datanya. Semua itu disediakan oleh pemrograman python. Bahasa pemrograman ini berorientasi objek dengan *syntax* yang mudah dipahami, serta dengan kinerja yang sangat baik. Jadi, tidak heran program ini sekarang banyak diminati oleh para *programmer*. Bahasa python banyak digunakan sebagai analisis data seperti *data science* dan *big data*. Selain itu, pemrograman python juga sering digunakan untuk penelitian ilmiah, sehingga memudahkan untuk proses perhitungan. Contohnya pengklasifikasian algoritma SVM, navie bayes, KNN, dan lain sebagainya [24].

1.11 Jupyter Notebook

Jupyter Notebook (file yang berekstensi *ipynb*) adalah dokumen yang dihasilkan oleh *Jupyter Notebook App* yang berisikan kode komputer dan *rich text element* seperti paragraf, persamaan matematik, gambar dan tautan (*links*) [25].

2. METODOLOGI PENELITIAN

2.1. Waktu Penelitian

Adapun Waktu yang dibutuhkan dalam penelitian ini yaitu selama 4 bulan di mulai dari bulan Februari 2020 dan akan berakhir bulan Juni 2020.

2.2 Metode Pengumpulan Data

Pengumpulan data yang dilakukan pada penelitian ini yaitu studi pustaka. Pengumpulan data dengan metode ini, dengan cara mencari referensi dapat melalui media cetak ataupun elektronik seperti skripsi, jurnal, *e-book* dan sumber literatur sejenis.

Studi yang digunakan sebagai acuan, seperti jurnal tentang klasifikasi ayat Al-Qur'an terjemahan, metode penelitian yang digunakan dan permasalahan pada pengklasifikasian ayat Al-Quran terjemahan. Setelah mendapatkan referensi, penulis mencoba untuk memahami permasalahan

pada topik yang berkaitan dengan penelitian ini. Kemudian memberikan kesimpulan terhadap permasalahan tersebut.

Diharapkan dengan melakukan pengumpulan data ini, dapat memudahkan proses penelitian dan menguatkan referensi dalam memecahkan permasalahan penelitian.

2.3 Metode Penelitian

Menurut Asiyah ada beberapa langkah yang bisa digunakan untuk melakukan klasifikasi klasifikasi teks yaitu sebagai berikut [6].

- 1) Siapkan data yang akan diklasifikasi.
- 2) Lakukan tahap *pre processing text* seperti proses *stemming*, *stopword* dan *tokenizing*.
 - a. *Stemming* yaitu tahapan yang digunakan untuk mengubah kata pada setiap kalimat menjadi kata dasar.
 - b. *Stopword* yaitu tahapan yang digunakan untuk membuang kata-kata yang tidak terlalu penting dalam melakukan klasifikasi.
 - c. *Tokenizing* yaitu tahapan yang digunakan untuk mengubah sebuah kalimat menjadi kata yang terpisah.
- 3) Lakukanlah proses pembobotan kata untuk mengubah setiap kata menjadi *word vector*
- 4) Bagilah data menjadi dua bagian yaitu data *training* sebesar 80% dan data *testing* sebesar 20%.
- 5) Klasifikasi data dengan metode SVM
 - a. Menentukan fungsi kernel untuk menyusun model SVM
 - b. Mengitung hasil akurasi dari model SVM
- 6) Klasifikasi data dengan metode KNN
 - a. Menentukan nilai k pada KNN
 - b. Menghitung hasil akurasi dari setiap nilai k pada KNN
- 7) Membandingkan kualitas kinerja antara metode SVM dan KNN berdasarkan hasil akurasi dalam ketepatan klasifikasi.

3. HASIL DAN PEMBAHASAN

Langkah awal dalam menentukan hasil dan pembahasan pada bab ini, yaitu *pre processing*. Setelah itu membagi *data training* dan *testing* untuk melakukan analisis. Pada proses ini, SVM menggunakan *Kernel Linear* dan KNN menggunakan $k = 3, 6$ dan 9 .

3.1 Pre Processing Text

Data ayat Al-Quran terjemahan yang telah disiapkan, lalu diproses dengan tahapan berikut yaitu menghilangkan imbuhan kata (*stemming*), menghilangkan kata penghubung (*stopword*), pemisahan kata (*tokenizing*) dan pembobotan kata (*tf-idf*).

3.1.1 Stemming

Proses *stemming* ini digunakan untuk mengubah ayat Al-Quran terjemahan menjadi kata dasar.

Tabel 2. Proses *Stemming*

Ayat Al-Quran Terjemahan	Sebelum	Sesudah
Al-Baqoroh (2 :38)	'Kami berfirman: "Turunlah kamu semuanya dari surga itu! Kemudian jika datang petunjuk-Ku kepadamu, maka barang siapa yang mengikuti petunjuk-Ku, niscaya tidak	'kami firman turun kamu semua dari surga itu kemudian jika datang tunjuk kepada maka barang siapa yang ikut tunjuk

	ada kekhawatiran atas mereka, dan tidak (pula) mereka bersedih hati".’	niscaya tidak ada khawatir atas mereka dan tidak pula mereka sedih hati’
Al-Baqoroh (2 :39)	‘Adapun orang-orang yang kafir dan mendustakan ayat-ayat Kami, mereka itu penghuni neraka; mereka kekal di dalamnya.’	‘ adapun orang yang kafir dan dusta ayat kami mereka itu huni neraka mereka kekal di dalam’

Berdasarkan tabel 2 dapat diketahui bahwa banyak kata yang di *stemming* berubah menjadi kata dasar. Contohnya seperti di surah Al-Baqoroh (2:38) “Turunlah” menjadi “turun” dan di surah Al-Baqoroh (2:39) “mendustakan” menjadi “dusta”.

3.1.2 Stopword

Setelah melakukan proses stemmer, langkah selanjutnya yaitu proses stopword yang berfungsi untuk menghilangkan kata-kata yang tidak terlalu penting. Sehingga meningkatkan kinerja pengklasifikasian teks. Hasil proses stopword ditampilkan dalam tabel 3 berikut.

Tabel 3. Proses *Stopword*

Ayat Al-Quran Terjemahan	Sebelum	Sesudah
Al-Baqoroh (2 : 38)	kami firman turun kamu semua dari surga itu kemudian jika datang tunjuk kepada maka barang siapa yang ikut tunjuk niscaya tidak ada khawatir atas mereka dan tidak pula mereka sedih hati	firman turun kamu semua surga kemudian datang tunjuk barang siapa ikut tunjuk niscaya khawatir atas sedih hati
Al-Baqoroh (2 : 39)	adapun orang yang kafir dan dusta ayat kami mereka itu huni neraka mereka kekal di dalam	adapun orang kafir dusta ayat huni neraka kekal

Dari tabel 3 kita melihat penghapusan kata dalam surah Al-Baqoroh (2 : 38) “kami”, “dari”, “itu”. Begitu juga dalam surah Al-Baqoroh (2 : 39) adapun kata yang hilang seperti “yang”, “dan”, “kami”.

3.1.3 Tokenizing

Tokenizing berfungsi untuk mengubah kalimat menjadi word vector. Agar memudahkan proses klasifikasi teks. Hasil *tokenizing* ditampilkan pada tabel 4 berikut.

Tabel 4. Proses *Tokenizing*

Ayat Al-Quran Terjemahan	Sebelum	Sesudah
Al-Baqoroh (2 :38)	firman turun kamu semua surga kemudian datang tunjuk barang siapa ikut tunjuk niscaya khawatir atas sedih hati	['firman', 'turun', 'kamu', 'semua', 'surga', 'kemudian', 'datang', 'tunjuk', 'barang', 'siapa', 'ikut', 'tunjuk', 'niscaya', 'khawatir', 'atas', 'sedih', 'hati']

Al-Baqoroh (2 :39)	adapun orang kafir dusta ayat huni neraka kekal	['adapun', 'orang', 'kafir', 'dusta', 'ayat', 'huni', 'neraka', 'kekal']
--------------------	--	---

Dari tabel 4 diatas, setiap kalimat pada surah Al-Baqoroh di ubah menjadi kata yang dipisahkan dengan tanda kutip (' ') dan koma(,).

3.1.4 Pembobotan Kata (*Term Weighting TF-IDF*)

Pembobotan kata akan menampilkan rank tertinggi dalam setiap *word vector*.

term	rank
kamu	0.981412
orang	0.848000
jadi	0.627808
allah	0.585550
sungguh	0.946314
kata	0.680763
atas	0.546848
ingat	0.511233
aku	0.639988
tahu	0.719190
diri	0.482094

Gambar 3. Rank Term

Dari gambar 3 ditampilkan bahwa kata tertinggi yang sering muncul adalah kata “kamu”, “orang”, “jadi”, “ALLAH”, “sungguh”, “kata”, “atas”, “ingat”, “aku”, “tahu” dan “diri”.

3.2 Metode SVM pada Text mining

Proses klasifikasi teks pada ayat Al-Quran terjemahan dengan metode SVM ini menggunakan fungsi kernel linear.

3.2.1 SVM menggunakan Kernel Linear pada confusion matrix

Kernel linear ini berfungsi untuk melihat ketepatan hasil dari word vector dalam setiap label. Analisis yang dijalankan untuk mendapatkan ketepatan hasil ini menggunakan *confusion matrix* pada *pandas*. Adapun analisis yang diperoleh dalam metode SVM terhadap *confusion matrix* sebagai berikut.

```
In [1016]: print(confusion_matrix(y_test, y_pred))  
[[0 0 1]  
 [0 0 2]  
 [0 0 2]]
```

Gambar 4. Confusion Matrix SVM

Dari analisis *confusion matrix* diatas, maka hasilnya dapat dilihat pada Tabel.4 berikut.

Tabel 4. Hasil *Confusion Matrix* pada SVM

	Label	Label prediksi		
		a	b	c
a	Larangan	0	0	1
b	Perintah	0	0	2
c	Informasi	0	0	2

Dari Tabel 4 terlihat bahwa hanya label informasi yang terklasifikasi dengan benar. Sedangkan kelas larangan dan perintah tidak masuk dalam klasifikasi yang benar.

3.2.2 SVM fungsi *Kernel Linear* pada Performensi Klasifikasi

Dalam melakukan klasifikasi, tentu hal yang dicari yaitu performansi klasifikasi terhadap ayat Al-Quran terjemahan. Berikut hasil performansi klasifikasi dalam menggunakan metode SVM.

Tabel 5. Performansi Klasifikasi

Kategori	<i>Recall</i>	<i>Precision</i>	<i>F1-score</i>
Larangan	0	0	0
Perintah	0	0	0
Informasi	40%	100%	57%
rata-rata makro	13%	33%	19%
rata-rata tertimbang	16%	40%	23%

Berdasarkan tabel di atas, performansi terbaik dalam klasifikasi yaitu label informasi. Pada bagian *recall* 40%, *precision* 100%, dan *f1-score* 57%. Sedangkan untuk label larang dan perintah, hasilnya 0%.

3.2.3 Metode SVM menggunakan *Kernel Linear* pada tingkat akurasi

Tingkat akurasi pada SVM berfungsi untuk melihat klasifikasi yang akurat pada setiap label ayat Al-Quran terjemahan. Adapun proses analisis pada library jupyter notebook dapat dilihat pada gambar berikut.

```
In [38]: print(accuracy_score(y_test, y_pred))  
0.4
```

Gambar 5. SVM

Berdasarkan analisis akurasi di atas, dapat dilihat bahwa nilai akurasi dengan metode SVM hanya sebesar 40%.

3.3 Metode KNN pada Text Mining

Klasifikasi KNN pada ayat Al-Quran terjemahan ini ditentukan dengan nilai k yaitu 3,6, dan 9. Nilai k ini berfungsi untuk melihat hasil terbaik dalam metode KNN. Karena prinsip kerja metode KNN yaitu menentukan hasil klasifikasi dari jarak terdekat. Proses yang digunakan dalam analisis yaitu *confusion matrix* dan performansi klasifikasi. Kedua hasil tersebut dijalankan menggunakan *pandas*.

3.3.1 KNN pada *confusion matrix*

a. *Confusion matrix* k=3

Confusion matrix k=3 memprediksi ketepatan Ayat Al-Quran terjemahan sesuai dengan klasifikasi berdasarkan nilai jarak Euclidean yaitu k=3. Adapun analisis yang diperoleh dalam metode KNN terhadap *confusion matrix* sebagai berikut.

```
In [1037]: print(confusion_matrix(y_test, y_pred))  
  
[[0 1 1]  
 [0 0 1]  
 [0 1 1]]
```

Gambar 6. analisis *confusion matrix* k=3

Dari analisis *confusion matrix* k=3 diatas, maka hasilnya dapat dilihat pada Tabel berikut.

Tabel 6. *Confusion Matrix* k=3

Label		Label Prediksi		
		a	b	c
a	Larangan	0	1	1
b	Perintah	0	0	1
c	Informasi	0	1	1

Dari Tabel 6 terlihat bahwa hanya tabel informasi yang terklasifikasi dengan benar. Sedangkan kelas larangan dan perintah tidak masuk dalam kategori yang benar.

b. *Confusion matrix* k=6

Confusion matrix k=6 memprediksi ketepatan Ayat Al-Quran terjemahan sesuai dengan klasifikasi berdasarkan nilai jarak Euclidean yaitu k=6. Adapun analisis yang diperoleh dalam metode KNN terhadap *confusion matrix* sebagai berikut.

```
In [1053]: print(confusion_matrix(y_test, y_pred))  
  
[[0 1 1]  
 [0 0 0]  
 [0 0 3]]
```

Gambar 7. Analisis *Confusion Matrix* k=6

Dari analisis *confusion matrix* k=6 diatas, maka hasilnya dapat dilihat pada Tabel berikut.

Tabel 7. *Confusion Matrix* k=6

Label		Label prediksi		
		a	b	c
a	Larangan	0	1	1

b	Perintah	0	0	0
c	Informasi	0	0	3

Dari tabel 7 terlihat bahwa hanya tabel informasi yang terklasifikasi dengan benar. Sedangkan kelas larangan dan perintah tidak masuk dalam kategori yang benar.

c. Confusion matrix k=9

Confusion matrix k=9 memprediksi ketepatan Ayat Al-Quran terjemahan sesuai dengan klasifikasi berdasarkan nilai jarak Euclidean yaitu k=9. Adapun analisis yang diperoleh dalam metode KNN terhadap confusion matrix sebagai berikut.

```
In [1069]: print(confusion_matrix(y_test, y_pred))
          [[0 1 0]
           [0 0 1]
           [0 1 2]]
```

Gambar 8. Analisis Confusion Matrix k=9

Dari analisis confusion matrix k=9 di atas, maka hasilnya dapat dilihat pada Tabel berikut.

Tabel 8. Hasil Confusion Matrix k=9

Kelas Asli		Kelas Prediksi		
		a	b	c
a	Larangan	0	1	0
b	Perintah	0	0	1
c	Informasi	0	1	2

Dari Tabel 8 terlihat bahwa hanya tabel informasi yang terklasifikasi dengan benar. Sedangkan kelas larangan dan perintah tidak masuk dalam kategori yang benar.

3.3.2 Hasil KNN menggunakan k=3,6,9 pada performansi klasifikasi

a. Performansi Klasifikasi k=3

KNN untuk performansi klasifikasi k=3 bertujuan untuk melihat ketepatan presentase recall, precision dan f1-score pada jarak Euclidean yang diberikan nilai 3. Hasil performansi k=3 dapat dilihat pada tabel berikut.

Tabel 9. Performansi Klasifikasi k=3

Kategori	recall	precision	f1-score
Larangan	0	0	0
Perintah	0	0	0
Informasi	33%	50%	40%
rata-rata makro	11%	17%	13%
rata-rata tertimbang	13%	20%	16%

Berdasarkan tabel 9 dapat dilihat bahwa ketepatan pada kategori larangan semuanya 0%, perintah semuanya 0%, dan informasi presentase *recall* 33%, *precision* 50%, *f1-score* 40%.

Adapun nilai rata-rata makro pada *recall*(11%), *precision*(17%), dan *f-measur*(13%). Sedangkan nilai rata-rata tertimbang pada *recall*(13%), *precision*(20%) dan *f1-score*(16%).

b. Performansi Klasifikasi k=6

KNN untuk performansi klasifikasi k=3 bertujuan untuk melihat ketepatan presentase *recall*, *precision* dan *f1-score* pada jarak Euclidean yang diberikan nilai 3. Hasil performansi k=3 dapat dilihat pada tabel berikut.

Tabel 10. Performansi Klasifikasi k=6

Kategori	<i>recall</i>	<i>precision</i>	<i>f1-score</i>
Larangan	0	0	0
Perintah	0	0	0
Informasi	75%	100%	86%
rata-rata makro	25%	33%	29%
rata-rata tertimbang	45%	60%	51%

Berdasarkan tabel 10 dapat dilihat bahwa ketepatan pada kategori larangan semuanya 0%, perintah semuanya 0%, dan informasi presentase *recall* 75%, *precision* 100%, *f1-score* 86%.

Adapun nilai rata-rata makro pada *recall*(25%), *precision* (33%), dan *f1-score* (29%). Sedangkan nilai rata-rata tertimbang pada *recall* (45%), *precision* n(60%) dan *f1-score* (16%).

c. Performansi Klasifikasi k=9

KNN untuk performansi klasifikasi k=3 bertujuan untuk melihat ketepatan presentase *recall*, *precision* dan *f1-score* pada jarak Euclidean yang diberikan nilai 3. Hasil performansi k=3 dapat dilihat pada tabel berikut.

Tabel 11. Performansi Klasifikasi k=9

Kategori	<i>recall</i>	<i>precision</i>	<i>f1-score</i>
Larangan	0	0	0
Perintah	0	0	0
Informasi	67%	67%	67%
rata-rata makro	22%	22%	22%
rata-rata tertimbang	40%	40%	40%

Berdasarkan tabel 11 dapat dilihat bahwa ketepatan pada kategori larangan semuanya 0%, perintah semuanya 0%, dan informasi presentase *recall* 33%, *precision* 50%, *f1-score* 40%.

Adapun nilai rata-rata makro pada *recall* (11%), *precision* (17%), dan *f1-score* (13%). Sedangkan nilai rata-rata tertimbang pada *recall* (13%), *precision* (20%) dan *f1-score* (16%).

3.3.3 Hasil KNN Menggunakan k=3,6,9 pada Tingkat Akurasi

a. Tingkat akurasi k=3

Tingkat akurasi pada KNN (k=3) berfungsi untuk melihat klasifikasi yang akurat pada setiap label ayat Al-Quran terjemahan menggunakan jarak *Euclidean* k=3. Adapun proses analisis pada library jupyter notebook dapat dilihat pada gambar berikut.

```
In [27]: print(accuracy_score(y_test, y_pred))  
0.2
```

Gambar 9. Analisis Tingkat Akurasi k=3

Berdasarkan analisis akurasi diatas, dapat dilihat bahwa nilai akurasi dengan metode KNN hanya sebesar 20%.

b. Tingkat akurasi k=6

Tingkat akurasi pada KNN (k=6) berfungsi untuk melihat klasifikasi yang akurat pada setiap label ayat Al-Quran terjemahan menggunakan jarak *Euclidean* k=6. Adapun proses analisis pada library jupyter notebook dapat dilihat pada gambar berikut.

```
In [38]: print(accuracy_score(y_test, y_pred))  
0.4
```

Gambar 10. Analisis Tingkat Akurasi k=6

Berdasarkan analisis akurasi di atas, dapat dilihat bahwa nilai akurasi dengan metode KNN hanya sebesar 40%.

c. Tingkat akurasi k=9

Tingkat akurasi pada KNN (k=9) berfungsi untuk melihat klasifikasi yang akurat pada setiap label ayat Al-Quran terjemahan menggunakan jarak *Euclidean* k=9. Adapun proses analisis pada library jupyter notebook dapat dilihat pada gambar berikut.

```
In [20]: print(accuracy_score(y_test, y_pred))  
0.6
```

Gambar 11. Analisis Tingkat Akurasi k=9

Berdasarkan analisis akurasi di atas, dapat dilihat bahwa nilai akurasi dengan metode KNN hanya sebesar 60%.

d. Rata-rata Nilai k=3,6,9 pada KNN

Berikut merupakan hasil rata-rata nilai k=3,6,9 pada metode KNN.

Tabel 12. Rata-rata Nilai $k=3,6,9$

Metode KNN	Akurasi Total	Recall	Precision	F1-score
K=3	20%	13%	20%	16%
K=6	40%	45%	60%	51%
K=9	60%	40%	40%	40%
Rata-rata	40%	33%	40%	36%

Berdasarkan tabel di atas, rata rata nilai klasifikasi ayat Al-Quran terjemahan berdasarkan $k=3,6,9$ yaitu akurasi total 40%, *recall* 33%, *precision* 40% dan *f1-score* 36%.

3.4 Perbandingan Hasil antara Metode SVM dan KNN

Setelah mendapatkan hasil klasifikasi ayat Al-Quran terjemahan dari Metode SVM dan KNN. Langkah terakhir yaitu melakukan perbandingan hasil yang terdiri dari nilai akurasi total, *recall*, *precision*, dan *f1-score*.

Tabel 13. Perbandingan Hasil

Metode	Akurasi Total	Recall	Precision	F1-score
SVM	40%	16%	40%	23%
KNN	40%	33%	40%	36%

Dari Tabel 13 didapatkan hasil bahwa nilai akurasi total dan *precision* antara metode KNN ($k=3,6,9$) dengan SVM kernel memiliki hasil yang sama. Sedangkan hasil *recall* dan *f1-score* menunjukkan bahwa metode KNN lebih baik daripada SVM.

4. KESIMPULAN

Berdasarkan hasil dan uraian dari pembahasan sebelumnya maka dapat disimpulkan sebagai berikut:

- 1) Diketahui metode KNN mendapatkan hasil akurasi terbaik saat nilai jarak $k=9$ yaitu 60%.
- 2) Diketahui metode KNN mendapatkan hasil *recall*, *precision* dan *f1-score* terbaik saat nilai jarak $k=6$ yaitu 45%, 60% dan 51%
- 3) Diketahui nilai rata-rata KNN saat jarak $k=3,6,9$ yaitu akurasi total 40%, *recall* 33%, *precision* 40% dan *f1-score* 36%
- 4) Diketahui bahwa metode KNN ($k=3,6,9$) memberikan hasil yang lebih baik daripada metode SVM linear

DAFTAR PUSTAKA

- [1] "Kumpulan Ayat-Ayat Perintah Dan Larangan Dalam Al-Quran_By Mohan." <https://www.scribd.com/doc/237294546/Kumpulan-Ayat-Ayat-Perintah-Dan-Larangan-Dalam-Al-Quran-By-Mohan> (diakses Des 12, 2019).
- [2] S. N. D. Pratiwi, B. S. S. Ulama, dan J. A. R. Hakim, "Klasifikasi Email Spam dengan Menggunakan Metode Support Vector Machine dan k-Nearest Neighbor," vol. 5, no. 2, hlm. 6, 2016.
- [3] R. Aulianita, "Komparasi Metode K-Nearest Neighbors dan Support Vector Machine Pada Sentiment Analysis Review Kamera," vol. 8, no. 3, hlm. 7, 2016.

- [4] M. R. Daulay, "STUDI PENDEKATAN ALQURAN," *Thariqah Ilm.*, vol. 1, no. 01, 2014, Diakses: Des 04, 2019. [Daring]. Tersedia pada: <http://jurnal.iain-padangsidempuan.ac.id/index.php/TI/article/view/254>.
- [5] L. Rahmawati, S. W. Sihwi, dan E. Suryani, "ANALISA CLUSTERING MENGGUNAKAN METODE K-MEANS DAN HIERARCHICAL CLUSTERING (STUDI KASUS: DOKUMEN SKRIPSI JURUSAN KIMIA, FMIPA, UNIVERSITAS SEBELAS MARET)," 2016, doi: 10.20961/itsmart.v3i2.654.
- [6] S. N. Asiyah, "KLASIFIKASI BERITA ONLINE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE DAN K-NEAREST NEIGHBOR," hlm. 81.
- [7] C. Darujati dan A. B. Gumelar, "PEMANFAATAN TEKNIK SUPERVISED UNTUK KLASIFIKASI TEKS BAHASA INDONESIA," hlm. 9.
- [8] W. Khalidy, "PENERAPAN ALGORITMA C4.5 PADA PROGRAM KLASIFIKASI MAHASISWA DROPOUT," Diakses: Des 13, 2019. [Daring]. Tersedia pada: https://www.academia.edu/6348045/PENERAPAN_ALGORITMA_C4.5_PADA_PROGRAM_KLASIFIKASI_MAHASISWA_DROPOUT.
- [9] E. Dragut, F. Fang, P. Sistla, C. Yu, dan W. Meng, "Stop word and related problems in web interface integration," *Proc. VLDB Endow.*, vol. 2, no. 1, hlm. 349–360, Agu 2009, doi: 10.14778/1687627.1687667.
- [10] M. Jamhari, E. Noersasongko, dan H. Subagyo, "Pengklusteran Dokumen Teks Hasil Peringkasan Dokumen Otomatis yang Menggunakan Metode Seleksi Fitur dan Latent Semantic Analysis (LSA)," *J. Teknol. Inf. Univ. Dian Nuswantoro*.
- [11] "KLASIFIKASI SMS SPAM MENGGUNAKAN SUPPORT VECTOR MACHINE | Lembaga Ilmu Pengetahuan Indonesia." <http://lipi.go.id/publikasi/klasifikasi-sms-spam-menggunakan-support-vector-machine/29729> (diakses Des 13, 2019).
- [12] S. If dan K. Sembiring, "Penerapan Teknik Support Vector Machine untuk Pendeteksian Intrusi pada Jaringan," hlm. 35.
- [13] S.-F. Liang, H.-C. Wang, dan W.-L. Chang, "Combination of EEG Complexity and Spectral Analysis for Epilepsy Diagnosis and Seizure Detection," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, hlm. 853434, Des 2010, doi: 10.1155/2010/853434.
- [14] N. Guduru, "TEXT MINING WITH SUPPORT VECTOR MACHINES AND NON-NEGATIVE MATRIX FACTORIZATION ALGORITHMS," hlm. 67.
- [15] M. Rivki dan A. Bachtiar, "IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR DALAM PENGKLASIFIKASIAN FOLLOWER TWITTER YANG MENGGUNAKAN BAHASA INDONESIA," *J. Sist. Inf.*, vol. 13, hlm. 31, Mei 2017, doi: 10.21609/jsi.v13i1.500.
- [16] E. Prasetyo, "Data mining konsep dan aplikasi menggunakan matlab," *Yogyak. Andi*, 2012.
- [17] B. V. Dasarathy, "Nearest neighbor (NN) norms: NN pattern classification techniques," *IEEE Comput. Soc. Tutor.*, 1991.
- [18] Y. Yang dan X. Liu, "A re-examination of text categorization methods," dalam *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*, Berkeley, California, United States, 1999, hlm. 42–49, doi: 10.1145/312624.312647.
- [19] A. J. T. D. Yanosma, dan K. Anggriani, "IMPLEMENTASI METODE K-NEAREST NEIGHBOR (KNN) DAN SIMPLE ADDITIVE WEIGHTING (SAW) DALAM PENGAMBILAN KEPUTUSAN SELEKSI PENERIMAAN ANGGOTA PASKIBRAKA," *Pseudocode*, vol. 3, no. 2, hlm. 98–112, Jan 2017, doi: 10.33369/pseudocode.3.2.98-112.
- [20] A. Hotho, A. Nurnberger, G. Paaß, dan S. Augustin, "A Brief Survey of Text Mining," hlm. 37.
- [21] M. H. Dunham, *Data Mining Introductory and Advanced Topics*. Prentice Hall/Pearson Education, 2003.
- [22] B. S. Arifin, *Psikologi sosial*. Pustaka Setia, 2015.

- [23] D. A. Nugroho, T. Ahmad, dan H. T. Ciptaningtyas, "Pengembangan Metode Quad Smoothness Reduced Difference Expansion Untuk Steganography Pada Audio," 2018, doi: 10.12962/j23373539.v6i2.23501.
- [24] T. R. Perkasa, H. Widyantara, dan P. Susanto, "RANCANG BANGUN PENDETEKSI GERAK MENGGUNAKAN METODE IMAGE SUBTRACTION PADA SINGLE BOARD COMPUTER (SBC)," hlm. 8.
- [25] D. Setiabudidaya, "PENGUNAAN PIRANTI LUNAK JUPYTER NOTEBOOK DALAM UPAYA MENSOSIALISASIKAN OPEN SCIENCE," INA-Rxiv, preprint, Feb 2018. doi: 10.31227/osf.io/2h7q4.