

Terakreditasi SINTA Peringkat 4

Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 28/E/KPT/2019 masa berlaku mulai Vol.3 No. 1 tahun 2018 s.d Vol. 7 No. 1 tahun 2022

Terbit online pada laman web jurnal:
<http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs>



Vol. 5 No. 2 (2020) 55 - 60

JOINTECS

(Journal of Information Technology and Computer Science)

e-ISSN:2541-6448

p-ISSN:2541-3619

Klasifikasi Kinerja *Programmer* pada Aktivitas Media Sosial dengan Metode *Stochastic Gradient Descent*

Rusydi Umar¹, Imam Riadi², Purwono³

^{1,3}Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan

²Program Studi Sistem Informasi, Fakultas Sains dan Teknologi Terapan, Universitas Ahmad Dahlan

¹rusydi@mti.uad.ac.id, ²imam.riadi@is.uad.ac.id, ³purwono1907048015@webmail.uad.ac.id

Abstract

The failure of technology-based startups in Indonesia is caused by the lack of solid team performance and the number of errors in the programmer recruitment process. The rapid progress in the culture of social media can be utilized as a method to get the best programmer candidates in startup. The recruitment method used can be in the form of a process of classification of candidate media programmer social content. The classification is expected to find the pattern of performance of the candidate programmers, with good or bad results. One classification method that can be used is the Stochastic Gradient Descent (SGD). The classification results show an accuracy value of 80%, a precision value of 81% and a recall value of 80%.

Keywords: classification; stochastic gradient descent; performance; programmer; startup.

Abstrak

Kegagalan perusahaan pemula berbasis teknologi (*startup*) di Indonesia diakibatkan oleh kurang solidnya kinerja tim serta banyaknya kesalahan dalam proses rekrutmen *programmer*. Kemajuan pesat dalam budaya bermedia sosial dapat dimanfaatkan sebagai salah satu metode untuk memperoleh kandidat *programmer* terbaik dalam *startup*. Metode perekrutan yang digunakan dapat berupa melakukan proses klasifikasi konten media sosial kandidat *programmer*. Klasifikasi tersebut diharapkan dapat menemukan pola kinerja kandidat *programmer*, dengan hasil baik atau buruk. Metode klasifikasi yang dapat digunakan salah satunya adalah *Stochastic Gradient Descent* (SGD). Hasil klasifikasi menunjukkan nilai akurasi sebesar 80%, nilai *precision* 81% dan nilai *recall* 80%.

Kata kunci: klasifikasi; *stochastic gradient descent*; kinerja; *programmer*; *startup*.

© 2020 Jurnal JOINTECS

1. Pendahuluan

Kemajuan perkembangan ekonomi di Indonesia di bidang digital terus berkembang pesat. Berbagai *startup* baru terus bermunculan dengan inovasi yang mampu menyalir masyarakat secara global. Kesuksesan besar yang diraih oleh *startup* berskala *unicorn* seperti Gojek, Tokopedia, Bukalapak, Traveloka ternyata memicu lahirnya *startup-startup* di Indonesia. *Startup* dijadikan *role model* bisnis yang digunakan untuk memperoleh profit tinggi oleh para pelaku bisnis [1]. Kegagalan

startup ternyata menjadi momok mengerikan bagi pelaku industri ini, yang berbanding terbalik dengan berita kesuksesan *startup-startup* besar yang telah mendapatkan pendanaan. Tim yang bekerja dengan tidak solid serta anggotanya yang tidak kompeten rupanya menjadi salah satu penyebabnya [2].

Bagian penting dalam kelangsungan hidup *startup* adalah memiliki *programmer* dengan kinerja yang baik dan profesional. Proses rekrutmen dan seleksi kandidat profesi ini haruslah tepat dan benar. Budaya bermedia

sosial yang terus meningkat dapat dijadikan sebagai salah satu cara untuk mendapatkan kandidat profesional bagi perusahaan [3]. Konten media sosial berupa informasi pribadi serta aktivitas umum yang dibuat, dapat digunakan untuk melihat aspek keprofesionalan penggunaannya [4].

Cara yang dapat digunakan untuk melihat aspek profesionalitas dari media sosial yaitu dengan melakukan klasifikasi setiap postingan yang telah dibuat. Klasifikasi ini digunakan sebagai alat prediksi masing-masing kandidat *programmer* dengan hasil kinerja baik atau buruk. Indikator yang dapat digunakan untuk mengukurnya yaitu *attitude*, komunitas, promosi, *portfolio*, *share knowledge & experiments*, *mentoring* dan opini diskusi [5].

Penelitian yang dilakukan oleh Tu [4] dengan judul “PRISM: *Profession identification in social media*” menghasilkan data bahwa kepribadian seseorang dari jejaring sosial dapat diprediksi dengan akurasi mencapai nilai 78,6%. Penelitian yang dilakukan oleh Hoang [6] dengan judul “*A stochastic gradient descent logistic regression software program for civil engineering data classification developed in .NET framework*” menghasilkan akurasi klasifikasi SGD hingga 84%. Penelitian yang dilakukan oleh Oktanisa [7] dengan judul “*Perbandingan Teknik Klasifikasi Dalam Data Mining Untuk Bank Direct Marketing*” menghasilkan akurasi klasifikasi SGD hingga 97,2%.

Algoritma *Stochastic Gradient Descent* (SGD) akan digunakan sebagai metode pemodelan klasifikasi kinerja *programmer* pada aktivitas media sosial.

2. Metode Penelitian

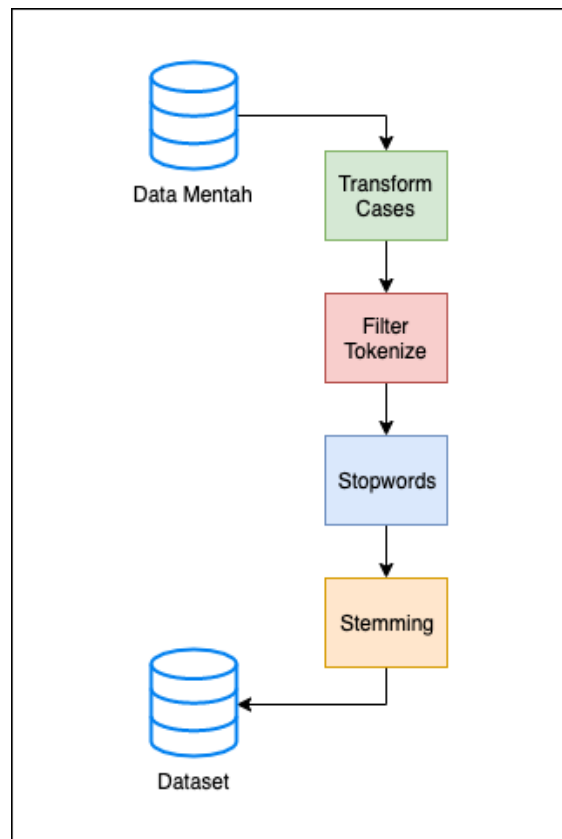
Tahapan-tahapan yang digunakan untuk mencapai tujuan penelitian meliputi beberapa langkah antara lain, (1) Pengumpulan data, (2) Pemrosesan teks, (3) TF-IDF, (4) Seleksi Fitur, (5) Pembuatan model klasifikasi, (6) Pengujian dan validasi hasil dan (7) Kesimpulan.

2.1. Pengumpulan data

Metode *survey*, *observasi*, studi pustaka dan studi literatur sejenis digunakan untuk menghasilkan informasi berharga dalam penelitian ini.

2.2. Pemrosesan Teks

Data yang dihasilkan masih perlu dilakukan penyempurnaan agar memiliki nilai yang berarti. Penyempurnaan dilakukan dengan cara pemilahan data dan pembersihan data. Pemrosesan teks (*Text Processing*) dilakukan sebagai upaya normalisasi data teks, pembersihan dari simbol-simbol, tanda baca, serta karakter-karakter yang tidak digunakan sehingga menghasilkan nilai informasi yang berharga [8]. Tahapan utama dalam pemrosesan teks dapat dilihat pada Gambar 1.



Gambar 1. Tahap Pemrosesan Teks

Data mentah yang dihasilkan kemudian dilakukan *transform cases* yaitu merubah deretan teks ke dalam bentuk kecil (*lower case*) atau kapital (*capitalize case*) [9]. Karakter, tanda baca serta simbol yang dianggap tidak berarti kemudian dihilangkan menggunakan *tokenize* melalui proses penyaringan (*filter*) dari panjang teks yang ada. Filter tokenisasi juga dapat menyaring kata-kata yang dianggap alay atau tidak sesuai dengan standar bahasa Indonesia [10]. Kata-kata umum yang sering muncul serta tidak berpengaruh dalam arti kalimat secara keseluruhan harus dihilangkan menggunakan proses *stop words*. Dalam bahasa Indonesia kata-kata umum itu seperti “yang”, “di”, “ke”, “nya” [11]. Hasil dari proses *stop words* harus disempurnakan kembali dengan proses *stemming* dengan metode pendekatan model *dictionary base stemming* yang dapat memberikan solusi untuk *stemm* kata berimbuhan dalam bahasa Indonesia, karena menggunakan struktur morfologi ketika mengekstrak kata berimbuhan menjadi kata dasar [12], sehingga dapat diolah ke tahap selanjutnya.

2.3. TF-IDF

TF-IDF digunakan dengan cara memberikan nilai bobot pada setiap kata yang paling umum digunakan agar dapat diketahui seberapa kuat hubungan kata (*term*) dengan dokumennya [13]. Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) dilakukan agar perhitungan bobot setiap kata dapat dilakukan dengan efisien, mudah dan akurat [14]. TF-IDF ialah sebuah

ukuran statistik pada proses evaluasi seberapa penting sebuah kata dalam suatu dokumen [15].

Metode TF-IDF dapat dihitung dengan menggunakan rumus pembobotan pada rumus (1) [14] yaitu:

$$Wdt = tfdt * Idft \tag{1}$$

Keterangan:

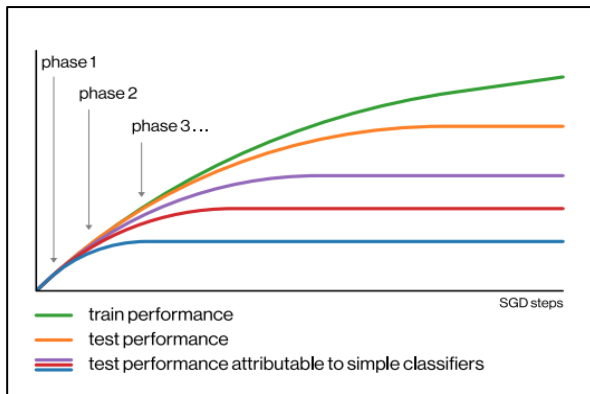
- Wdt = bobot dokumen ke-d terhadap kata ke-t
- $tfdt$ = banyaknya kata yang dicari
- $Idft$ = *Inverse Document Frequency*
- N = total dokumen
- df = jumlah N mengandung kata yang dicari

2.4. Seleksi Fitur

Hasil pembobotan dari metode TF-IDF dilakukan proses seleksi, dipilih dan digabungkan sehingga terbentuk ringkasan inti dari dokumen tersebut [16].

2.5. Stochastic Gradient Descent

Stochastic Gradient Descent ialah pendekatan sederhana dan efisien dalam melakukan klasifikasi linier dengan pembelajaran diskriminatif [5]. Proses klasifikasi menggunakan SGD dapat dilihat pada Gambar 2 [17]



Gambar 2. SGD Classification

Metode SGD merupakan algoritma optimasi iteratif untuk mencari titik fungsi minimum yang dapat diturunkan. Algoritmanya dimulai dengan cara melakukan penebakan di awal proses. Kesalahan penebakan-penebakan kemudian diperbaiki seiring adanya perulangan tebakan yang menggunakan aturan gradient (turunan) dari fungsi yang hendak diminimalkan. Penurunan fungsi minimum digunakan secara khusus dengan rumus (2) [18].

$$\omega_i + 1 = \omega_i - \eta \nabla_{\omega_i} L(\omega_i) \tag{2}$$

Keterangan:

- $\omega_i + 1$ = Model parameter prediksi
- ω_i = Parameter model pada iterasi sebelumnya
- η = Tingkat pembelajaran
- L = *loss cost function*

SGD memiliki sifat yang lebih label dan lebih cepat ketika dilakukan pelatihan klasifikasi serta tidak

terbatas pada waktu dalam pelaksanaannya berdasarkan ukuran dataset latih. Metode SGD memiliki kemampuan belajar yang lebih cepat. Fungsi *hinge loss* yang digunakan sebagai pelatihan *classifier* dapat dijelaskan dengan rumus (3) [18].

$$L(x_j, y_j) = \max(0, 1 - y_j \cdot (\omega x_j + b)) \tag{3}$$

Keterangan:

- ω dan b = Model parameter untuk prediksi
- x_j = Contoh input
- y_j = Target kelas

Persamaan (3) merupakan fungsi metrik klasifikasi sebagai alat ukur kemampuan model linier yang telah diprediksi menggunakan metode SGD pada setiap perulangan fase pembelajaran. Persamaan (3) kemudian dimodifikasi parameter (ω, b) sehingga menghasilkan persamaan (4). Klasifikasi menggunakan SGD, nilai ω sesuai dengan bobot yang telah diatur untuk fitur hamburan balik pada fungsi keputusan, dan b ialah intersepnya. Bagian menarik dari fungsi *hinge loss* ialah proses penghukuman terhadap sampel yang salah dalam klasifikasi, namun tetap diberikan kepercayaan yang rendah sebagai pembatas antar kelas.

Loss function juga bekerja dengan regularisasi yang bertujuan untuk membantu model yang diprediksi serta menggeneralisasi data yang tidak memiliki label. Regularisasi bertugas sebagai protokol untuk menghukum model kompleks yang lebih dominan terjadi *overfitting*, yang ditandai dengan nilai yang lebih besar untuk parameter ω_i . Regularisasi dapat dilihat pada rumus (4) dan (5).

$$L1 = \sum_{i=1}^m |\omega_i| \tag{4}$$

$$L2 = \sum_{i=1}^m \omega_i^2 \tag{5}$$

Keterangan:

- m = Prediktor variabel
- ω = Model parameter untuk prediksi

2.6. Confusion Matrix

Data dalam klasifikasi dibagi menjadi data training dan data testing dengan skala perbandingan 75% dan 25%. Hasil klasifikasi menghasilkan *confusion matrix* dalam bentuk tabel yang terdiri dari kelas prediksi dan kelas aktual. Model *confusion matrix* 3x3 ditunjukkan pada Tabel 1 [19].

Tabel 1. Model Confusion Matrix

Keterangan		Kelas Prediksi		
		Kelas A	Kelas B	Kelas C
Kelas Aktual	Kelas A	AA	AB	AC
	Kelas B	BA	BB	BC
	Kelas C	CA	CB	CC

Evaluasi model klasifikasi dapat dihitung nilai akurasi, *presisi* dan *recall*-nya [19]. Akurasi klasifikasi dapat dihitung dengan rumus (6) yaitu:

$$Ak = \frac{AA + BB + CC}{AA + AB + AC + BA + BB + BC + CA + CB + CC} \quad (6)$$

Keterangan:

Hasil akurasi klasifikasi (Ak) merupakan jumlah data yang tepat ($AA+AB+CC$) dibagi dengan total data ($AA+AB+AC+BA+BB+BC+CA+CB+CC$).

Presisi merupakan nilai probabilitas dari sebuah item yang terpilih adalah relevan. Rumus perhitungan presisi dapat dilihat pada rumus (7) yaitu:

$$Presisi\ i = \frac{Ai}{Ai + Bi + Ci} \quad (7)$$

Keterangan:

Ai = item kelas A
 Bi = item kelas B
 Ci = item kelas C

Recall adalah rasio item relevan yang dipilih berdasarkan jumlah item yang relevan. Rumus perhitungan *recall* dapat dilihat pada rumus (8) yaitu:

$$Recall\ i = \frac{iA}{iA + iB + iC} \quad (8)$$

Keterangan:

iA = item relevan kelas A
 iB = item relevan kelas B
 iC = item relevan kelas C

2.7. Validasi Hasil

Metode *cross validation* digunakan sebagai alat validasi hasil klasifikasi. *Cross validation* merupakan metode untuk memprediksi keakuratan data pengujian [20]. Model *cross validation* yang digunakan adalah *K-fold cross validation* yaitu sebuah teknik proses iterasi data *training* dan testing sebanyak k iterasi dan pembagian $1/k$ dari dataset, dengan catatan bahwa $1/k$ tersebut digunakan sebagai data testing [21]. Analogi sederhana untuk menjelaskan metode ini adalah ketika kita memiliki 600 data dengan $k=6$, maka keseluruhan data akan terbagi menjadi 6 bagian dengan isi 100 data. Setiap bagian kelompok data akan dipisahkan dengan menentukan persentase data *training* dan *testing*. Pembagian *persentase* paling umum adalah 70% berbanding 30% [5].

3. Hasil dan Pembahasan

3.1. Dataset

Dataset dihasilkan melalui tahap pengumpulan data yaitu proses observasi dan survey dari group *facebook* PHP Indonesia. *Dataset* yang terkumpul sebanyak 2178 data posting *facebook* dan disimpan dengan format (.csv) menggunakan Microsoft Excel. Format dataset

ini adalah membagi ke dalam dua kolom yaitu Posting dan Kategori. Kategori merupakan kelas klasifikasi yang digunakan merujuk pada indikator kinerja programmer dari aktivitas media sosial [5]. *Dataset* posting *programmer* ini dapat diakses secara *public* pada alamat <https://www.kaggle.com/purwonopurwon/o/dataset-postingan-programmer-facebook>.

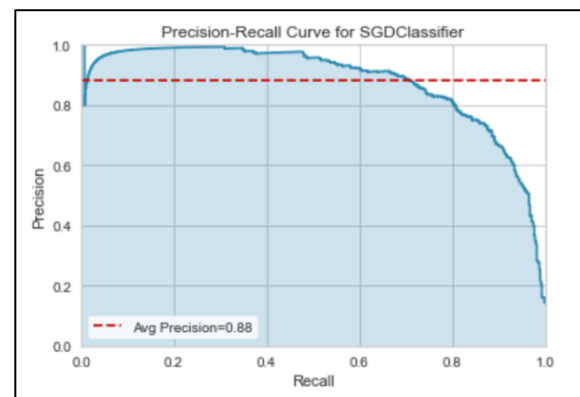
3.2. Performa Stochastic Gradient Descent

Pembagian data *training* dan *testing* dari 2178 posting *facebook* pada model klasifikasi ini adalah 70% berbanding 30%. Parameter SGD yang digunakan yaitu *loss function*, *penalty*, *random state*, *max_iter* dan *tol*. *Loss function* berfungsi sebagai penghitung kesalahan model optimasi klasifikasi, *penalty* sebagai ukuran kesalahan prediksi *random_state* sebagai pembangkit bilangan acak iterasi, *max_iter* sebagai pembatas data *training* dan *tol* sebagai penanda kapan kriteria akan berhenti. Parameter model yang digunakan adalah *loss* = 'hinge', *penalty* = 'l2', *alpha*=1e-3, *random_state*=42, *max_iter*=5 dan *tol*=none. Hasil klasifikasi dengan pemodelan SGD dapat dilihat pada Tabel 2.

Tabel 2. Evaluasi Hasil Klasifikasi Model SGD

Indikator	Precision	Recall	Accuracy
Attitude	0,94	0,80	0,8
Komunitas	0,82	0,90	0,89716312
Promosi	0,73	0,77	0,76969697
Portfolio	0,84	0,84	0,84210526
Share Knowledge & Experiments	0,85	0,56	0,55737705
Mentoring	0,00	0,00	0,0
Opini & Diskusi	0,83	0,79	0,79032258

Grafik hasil evaluasi menggunakan *precision* dan *recall* untuk klasifikasi dengan pemodelan SGD dapat dilihat pada Gambar 3.



Gambar 3. Precision recall Curve SGD

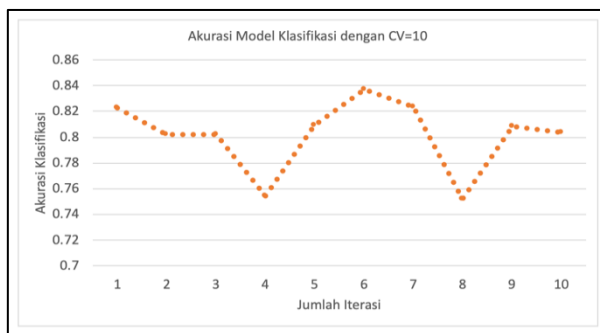
3.3. Validasi Hasil

Validasi dilakukan dengan melakukan iterasi sebanyak 10 kali ($k=10$) pengujian model. Hasil iterasi validasi model SGD dapat dilihat pada Tabel 3.

Tabel 3. Akurasi Model Klasifikasi dengan CV=10

Iterasi ke-i	Akurasi
1	0,822727273
2	0,801843318
3	0,801843318
4	0,75462963
5	0,809302326
6	0,837209302
7	0,823255814
8	0,752336449
9	0,808411215
10	0,803738318

Berdasarkan Tabel 3, nilai akurasi tertinggi dari model klasifikasi SGD sebesar 83,7% pada iterasi pengujian ke-6 dan nilai akurasi terendah pada iterasi pengujian ke-8 yaitu 75,2%. Hasil akurasi tersebut rupanya belum bisa melampaui akurasi yang sudah diteliti oleh Hoang [6] yaitu sebesar 84% dan Oktanisa [7] yaitu sebesar 97,2%. Rata-rata akurasi yang dihasilkan dari model klasifikasi ini adalah 80,1%. Hasil iterasi pengujian akurasi klasifikasi dengan $k=10$ *cross validation* dapat dilihat lebih jelas menggunakan *grafik line* pada Gambar 4.



Gambar 4. Iterasi Pengujian Model SGD

4. Kesimpulan

Berdasarkan hasil penelitian, diperoleh data bahwa metode SGD dengan pengujian $k=10$ *cross validation* dalam melakukan klasifikasi kinerja *programmer* menghasilkan nilai akurasi yang cukup baik yaitu 80,1%. Kontribusi penelitian ini menghasilkan model klasifikasi otomatis hasil posting dari media sosial yang dikhususkan untuk kandidat *programmer* mungkin dapat juga diterapkan pada profesi lain.

Penelitian selanjutnya adalah melakukan optimasi akurasi agar dapat menghasilkan data prediksi yang lebih baik. Peneliti juga berharap dapat melakukan klasifikasi kinerja tidak terbatas pada konten media sosial berbentuk teks saja, namun mampu melakukan klasifikasi juga pada konten berbentuk citra.

Daftar Pustaka

- [1] M. Arya, R. Ferdiana, and S. Fauziyati, "Analisis Faktor Keberhasilan Startup Digital di Yogyakarta," in *Jurnal.Umk.Ac.Id*, 2017, vol. 4, no. 1, pp. 167–173.
- [2] M. D. K. Perdani, Widyawan, and P. I. Santoso, "Faktor-faktor yang mempengaruhi pertumbuhan startup di yogyakarta," in *Seminar Nasional Teknologi Informasi dan Komunikasi 2018*, 2018, vol. 2018, no. Sentika, pp. 23–24.
- [3] T. Koch, C. Gerber, and J. J. De Klerk, "The impact of social media on recruitment: Are you LinkedIn?," *SA J. Hum. Resour. Manag.*, vol. 16, pp. 1–14, 2018.
- [4] C. Tu, Z. Liu, H. Luan, and M. Sun, "PRISM: Profession identification in social media," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 6, 2017.
- [5] R. Umar, I. Riadi, and Purwono, "Perbandingan Metode SVM, RF dan SGD untuk Penentuan Model Klasifikasi Kinerja Programmer pada Aktivitas Media Sosial," *RESTI*, vol. 4, no. 2, pp. 329–335, 2020.
- [6] N. D. Hoang and H. T. Nguyen, "A stochastic gradient descent logistic regression software program for civil engineering data classification developed in .NET framework," *DTU J. Sci. Technol.*, no. June, 2019.
- [7] I. Oktanisa and A. A. Supianto, "Perbandingan Teknik Klasifikasi Dalam Data Mining Untuk Bank Direct Marketing," *Tekno. Inf. dan Ilmu Komput.*, vol. 5, no. 5, pp. 567–576, 2018.
- [8] A. Handayanto, K. Latifa, N. D. Saputro, and R. R. Waliyansyah, "Analisis dan Penerapan Algoritma Support Vector Machine (SVM) dalam Data Mining untuk Menunjang Strategi Promosi (Analysis and Application of Algorithm Support Vector Machine (SVM) in Data Mining to Support Promotional Strategies)," *JUITA J. Inform.*, vol. 7, no. November, pp. 71–79, 2019.
- [9] A. T. J. Harjanta, "Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining," *Inform. UPGRIS*, vol. 1, pp. 1–9, 2015.
- [10] A. T. Jaka, "Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining," *Inform. UPGRIS*, vol. 1, pp. 1–9, 2015.
- [11] A. Alajmi and E. mostafa Saad, "Toward an ARABIC Stop-Words List Generation Toward an ARABIC Stop-Words List Generation," no. January 2012, 2018.
- [12] A. F. Zulfikar, "Pengembangan Algoritma Stemming Bahasa Indonesia dengan Pendekatan Dictionary Base Stemming untuk Menentukan Kata Dasar dari Kata yang Berimbuhan," *J. Inform. Univ. Pamulang*, vol. 2, no. 3, p. 143, 2017.
- [13] B. Herwijayanti, D. E. Ratnawati, and L. Muflikhah, "Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity," *Pengemb. Tekno. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 306–312, 2018.

- [14] R. Melita *et al.*, "(Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus : Syarah Umdatil Ahkam)," vol. 11, no. 2, 2018.
- [15] A. A. A. Putra, "Implementasi Text Summarization Menggunakan Metode Vector Space Model Pada Artikel Berita Berbahasa Indonesia," Universitas Komputer Indonesia, 2018.
- [16] N. K. Widyasanti, I. K. G. D. Putra, and N. K. D. Rusjyanthi, "Seleksi Fitur Bobot Kata dengan Metode TF-IDF untuk Ringkasan Bahasa Indonesia," *Merpati*, vol. 6, no. 2, pp. 119–126, 2018.
- [17] P. Nakkiran *et al.*, "SGD on Neural Networks Learns Functions of Increasing Complexity," 2019, no. NeurIPS, pp. 1–11.
- [18] A. S. Ritonga and E. S. Purwaningsih, "Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding)," *Ilm. Edutic*, vol. 5, no. 1, pp. 17–25, 2018.
- [19] D. Iskandar and Y. K. Suprpto, "Perbandingan Akurasi Klasifikasi Tingkat Kemiskinan Antara Algoritma C 4.5 dan Naive Bayes," *J. Ilm. NERO*, vol. 2, no. 1, pp. 37–43, 2015.
- [20] T. H. Apandi, C. A. Sugianto, and C. R. Service, "Algoritma Naive Bayes untuk Prediksi Kepuasan Pelayanan Perekaman e-KTP (Naive Bayes Algorithm for Satisfaction Prediction of e-ID," *JUITA J. Inform.*, vol. 7, no. November, pp. 125–128, 2019.
- [21] S. Asiyah and K. Fithriasari, "Klasifikasi Berita Online Menggunakan Metode Support Vector Machine Dan K-Nearest Neighbor," *J. Sains dan Seni ITS*, vol. 5, no. 2, 2016.