

HASIL CEK_60020397_Point- C69-IRD-850GB-Klasifikasi Kinerja Programmer pada Aktivitas Media Sosial dengan Metode Support Vector Machines

by Imam Riadi 60020397

Submission date: 11-Dec-2020 11:03AM (UTC+0700)

Submission ID: 1471715133

File name: Aktivitas_Media_Sosial_dengan_Metode_Support_Vector_Machines.pdf (398.8K)

Word count: 3574

Character count: 21389

3 Klasifikasi Kinerja Programmer pada Aktivitas Media Sosial dengan Metode Support Vector Machines

Rusydi Umar¹⁾, Imam Riadi²⁾, Purwono³⁾

Universitas Ahmad Dahlan Yogyakarta^{1,2,3)}

Jl. Prof. DR. Soepomo, S.H, Warungboto, Kec. Umbulharjo, Yogyakarta^{1,2,3)}

rusydi@mti.uad.ac.id¹⁾, imam.riadi@is.uad.ac.id²⁾, purwono1907048015@webmail.uad.ac.id³⁾

Abstrak

Peningkatan kebutuhan software di berbagai perusahaan terus meningkat. Kualitas software yang dibutuhkan harus memiliki nilai manfaat yang tinggi bagi perusahaan. Kesalahan dalam perekrutan programmer berakibat fatal terhadap kualitas software yang dihasilkan. Perekrutan profesi ini harus dilakukan dengan prosedur yang benar. Meningkatnya penggunaan media sosial dapat dijadikan sebagai salah satu metode perekrutan kandidat profesional karena sebelumnya belum banyak menyentuh aspek ini. Media sosial telah merekam aktivitas digital yang dilakukan oleh kandidat programmer dalam bentuk data posting sebagai pola kinerja mereka. Tujuan dari penelitian ini adalah menemukan pola kinerja baik atau buruknya programmer dengan cara melakukan klasifikasi berdasarkan postingan media sosial. Penelitian ini berupa implementasi dari metode SVM yang digunakan sebagai sistem klasifikasi otomatis. Hasil penelitian menunjukkan persentase akurasi dengan $k=10$ cross-validation untuk algoritma SVM mencapai angka tertinggi 85,1%, sehingga metode ini dianggap cukup baik dalam melakukan klasifikasi.

Kata kunci: support vector machines, klasifikasi, kinerja, media sosial, programmer

Abstract

Increasing software requirements in various companies continue to increase. The quality of the software needed must have a high value for the company. Error in recruiting programmers is fatal to the quality of the software produced. Recruitment of this profession must be carried out with the right procedures. The increasing use of social media can be used as a method of recruiting professional candidates because previously it has not touched this aspect. Social media has recorded digital activities carried out by prospective programmers in the form of posting data as a pattern of their performance. The purpose of this research is to find the pattern of good or bad performance of the programmer by classification based on social media posts. This research is an implementation of the SVM method which is used as an automatic classification system. The results showed the percentage of accuracy with $k = 10$ cross-validation for the SVM algorithm reached the highest rate of 85.1%, so this method is considered quite good in doing classification.

Keywords: support vector machines, classification, performance, social media, programmer

1. Pendahuluan

Perkembangan teknologi informasi yang begitu pesat menjadikan kebutuhan software menjadi semakin tinggi. Kebutuhan ini digunakan oleh perusahaan agar dapat bersaing di pasar global serta menunjang adaptasi proses bisnis [1]. Kebutuhan software yang meningkat dibanyak perusahaan membuat profesi programmer semakin diperlukan. Kualitas software yang baik ditentukan oleh kinerja programmer yang baik, sehingga proses rekrutmen harus dilakukan dengan prosedur yang benar.

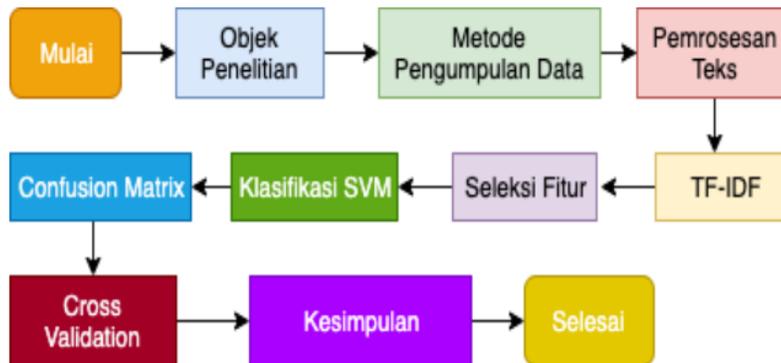
Banyak waktu yang dihabiskan oleh *programmer* untuk berselancar di dunia maya atau bermedia sosial seperti berdiskusi teknik pemrograman atau hanya sebatas meluapkan ekspresi kehidupan sehari-hari. Hal tersebut rupanya dapat dijadikan data pola tertentu seperti metode prediksi kepribadian [2]. Media sosial berkembang begitu pesat sehingga dapat dimanfaatkan sebagai salah satu metode strategis untuk merekrut kandidat potensial suatu perusahaan [3]. Metode ini digunakan sebagai salah satu fitur baru dalam proses perekrutan, karena sebelumnya belum banyak menyentuh aspek ini. Metode ini berfungsi untuk mengurangi resiko kesalahan dalam pembuatan *software* yang diakibatkan oleh kinerja *programmer* yang buruk. Indikator kinerja yang dapat digunakan yaitu *attitude*, komunitas, promosi, *portfolio*, *share knowledge & experiments*, *mentoring* dan opini diskusi [4].

Penelitian serupa telah dilakukan oleh para peneliti antara lain dilakukan oleh Tadesse [5] menunjukkan bahwa fitur jejaring sosial dapat digunakan untuk memprediksi kepribadian seseorang dengan akurasi hingga 78,6 % pada model kepribadian Big 5 yaitu *openness*, *conscientiousness*, *extraversion*, *agreeableness* dan *neuroticism*. Penelitian yang dilakukan oleh Mustika [6] menunjukan bahwa metode SVM dapat digunakan untuk klasifikasi *tweet sentiment public figure* dengan *kernel RBF* dan *polynomial* dengan keakuratan yang cukup baik yaitu 72,5%. Penelitian yang dilakukan oleh Maulina [7] menunjukan hasil bahwa metode SVM memiliki akurasi yang baik dalam melakukan klasifikasi artikel *hoax* dengan nilai akurasi sebesar 95,8% dengan $k=10$ *cross-validation*. Penelitian yang dilakukan oleh Asiyah [8] menunjukan hasil bahwa metode SVM dapat digunakan sebagai sistem klasifikasi berita online dengan akurasi total sebesar 93,2%. Penelitian yang dilakukan oleh Wulandari [9] metode SVM digunakan untuk klasifikasi keluhan pelanggan berbasis twitter dengan nilai rata-rata akurasi yang cukup baik yaitu 81,26%.

Algoritma *Support Vector Machines* (SVM) dapat digunakan sebagai metode strategis yang dikembangkan sebagai sebuah sistem klasifikasi otomatis postingan media sosial dari setiap kandidat *programmer*. Kandidat *programmer* yang akan direkrut akan diklasifikasi postingan media sosialnya. Hasil klasifikasi ialah apakah postingan kandidat tersebut masuk pada tujuh indikator kinerja *programmer* atau tidak. Jumlah postingan yang diklasifikasi ditentukan oleh pihak *rekrutmen*, semakin banyak indikator yang masuk maka kinerjanya dianggap semakin baik. Kandidat *programmer* satu dengan yang lainnya akan memiliki predikat kinerja yang berbeda berdasarkan banyaknya indikator yang masuk setelah diklasifikasi. Pola predikat yang dihasilkan dapat digunakan sebagai sistem penunjang keputusan dalam proses *rekrutmen* di mana apakah kandidat dapat diterima atau ditolak.

2. Metode Penelitian

Tahapan-tahapan dalam penelitian yang digunakan untuk mencapai tujuan dari penelitian ini yaitu, (1) Objek Penelitian, (2) Metode Pengumpulan Data, (3) Pemrosesan Teks, (4) TF-IDF, (5) Seleksi Fitur, (6) Klasifikasi SVM, (7) *Confusion Matrix*, (8) *Cross Validation*, (9) Penarikan Kesimpulan. Tahapan-tahapn tersebut dapat dilihat pada diagram alir yang terdapat pada Gambar 1.



Gambar 1. Diagram alir penelitian

2.1. Objek Penelitian

Objek penelitian ialah anggota group *facebook* PHP Indonesia. Group ini adalah salah satu group yang membahas seputar teknologi informasi dan dunia *programming* dengan anggota lebih dari 153.000 orang.

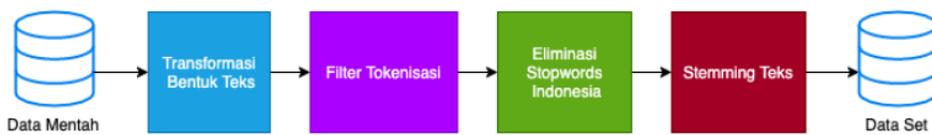
2.2. Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan yaitu observasi, studi pustaka dan studi literatur sejenis. Data yang dihasilkan adalah postingan *facebook* dari masing-masing kandidat *programmer* yang selanjutnya akan disimpan dalam format Microsoft Excel (.xls)

2.3. Pemrosesan Teks

Data yang diperoleh pada tahap sebelumnya perlu dilakukan normalisasi, pembersihan data yang tidak memiliki makna serta menghapus data yang tidak diperlukan karena data tersebut dianggap masih memiliki banyak kesalahan nilai (*missing value*) [10].

Diagram alir tahapan-tahapan dalam pemrosesan teks dapat dilihat pada Gambar 2.



Gambar 2. Tahap pemrosesan teks

Data mentah terlebih dahulu dilakukan transformasi bentuk teks ke dalam bentuk huruf kecil (*lower case*) atau besar (*capital case*) secara keseluruhan (*transform cases*) [11]. Perubahan ke dalam bentuk *lower case* digunakan dalam penelitian ini. *Filter tokenisasi* dilakukan dengan cara menghilangkan berbagai karakter, simbol serta tanda baca yang dianggap tidak penting dengan cara melakukan penyaringan berdasarkan panjang suatu teks [11]. *Stopword elimination* dijadikan langkah-langkah pembersihan kata-kata yang dianggap tidak relevan dijadikan sebagai indeks dokumen [12]. Stopwords yang digunakan adalah format bahasa Indonesia seperti “yang”, “di”, “ke”, “dan”, “atau” [13]. Tahapan akhir sebelum menghasilkan *dataset* yang sudah memiliki nilai ialah *stemming* yaitu dilakukan perubahan bentuk kata imbuhan menjadi kata dasar [14]. Tabel 1 merupakan contoh sebelum dan sesudah postingan dilakukan pemrosesan teks.

Tabel 1. Contoh pemrosesan teks

Data Mentah	Dataset
Video ini adalah progress pengembangan software automation di hari kedua, setelah kemarin gagal memiliki waktu untuk coding. Terdapat POC yang berhasil saya selesaikan. Kita dapat membuat program yang dapat memproduksi bahasa pemrograman lainnya untuk keperluan otomasi rapid prototyping.	video progress kembang software automation hari dua gagal milik waktu coding poc hasil selesai buat program produk bahasa program lain perlu otomasi rapid prototyping

2.4. TF-IDF

Metode TF-IDF digunakan untuk menentukan seberapa kuat hubungan kata (*term*) dengan dokumen dengan cara memberikan pembobotan pada setiap kata [4, 5]. Metode TF-IDF mengkombinasikan dua buah konsep yaitu banyaknya kemunculan kata dan *invers* frekuensi dokumen yang mengandung kata tersebut. Metode ini digunakan untuk menghitung nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap teks postingan *facebook* [16]. Rumus yang digunakan yaitu:

1. Rumus pembobotan TF-IDF

$$W = tf * idf \quad (1)$$

$$W = tf * \log \left(\frac{N}{df} \right) \quad (2)$$

Keterangan:

tf = term frequency
 idf = inverse document frequency
 N = jumlah document

2. Berdasarkan persamaan (2), berapapun besarnya nilai tf , jika $N = df$ dimana sebuah kata muncul di semua dokumen, maka akan didapatkan hasil 0 (nol) untuk perhitungan idf , sehingga perhitungan bobotnya diubah menjadi sebagai berikut:

$$W = tf * \left(\log \left(\frac{N}{df} \right) + 1 \right) \quad (3)$$

3. Persamaan (3) dapat dinormalisasi dengan persamaan (4) yang bertujuan untuk menstandarisasi nilai bobot (wtd) ke dalam interval 0 sampai dengan 1:

$$W = \frac{tf * \left(\log \left(\frac{N}{df} \right) + 1 \right)}{\sqrt{\sum_{k=1}^t (tf)^2 + \left(\log \left(\frac{N}{df} \right) + 1 \right)^2}} \quad (4)$$

Keterangan:

t = term
 df = document frequency
 k = kelipatan document

2.5. Seleksi Fitur

Seleksi fitur dilakukan setelah melakukan pembobotan kata dengan metode TF-IDF. Nilai bobot dihasilkan dari setiap paragraf yang ada pada suatu dokumen. Kata-kata yang memiliki bobot tertinggi dari masing-masing paragraf kemudian diseleksi dan dipilih dan digabungkan mejadi sebuah ringkasan [17]. Metode seleksi fitur yang digunakan adalah *information gain*. Metode ini digunakan untuk memilih fitur terbaik yang digunakan untuk memberikan *rangking* terhadap kata-kata penting hasil pembobotan kata, sehingga proses klasifikasi menjadi lebih efektif dan optimal [18].

2.6. Support Vector Machines

Algoritma SVM banyak digunakan untuk melakukan klasifikasi secara otomatis seperti klasifikasi text, pengenalan citra, analisis medik ataupun prediksi. Klasifikasi adalah pengumpulan objek yang memiliki karakteristik yang sama ke dalam beberapa kelas. SVM dapat digunakan sebagai parameter untuk membantu dalam pembagian klasifikasi data [19]. SVM bekerja dengan cara membuat garis pembatas atau *hyperplane* dari kedua data dengan dua kelas yang berbeda dengan memanfaatkan pola untuk membedakan kelas pembatas dengan nilai kelas 1 dan nilai kelas -1 [20]. *Linear SVM* merupakan salah satu *kernel* berbentuk lurus yang membagi dua kelas. Perhitungan SVM *Linear* seperti pada Rumus (5) [7]:

$$w * x - b = 0 \quad (5)$$

Keterangan:

w = parameter *hyperplane* yang dicari
 x = data input SVM
 b = parameter *hyperplane* yang dicari (bias)

Pencarian *hyperplane* paling optimum pada klasifikasi linier dapat dilakukan dengan persamaan (6) [22].

$$\min \frac{1}{2} \|\omega\|^2 \quad (6)$$

$$y_i(wx_i + b) \geq 1, i = 1, \dots, \lambda \quad (7)$$

3

Klasifikasi Kinerja Programmer pada Aktivitas Media Sosial dengan Metode Support Vector Machines (Rusydi Umar, dkk)

Keterangan:

x_i = Data input

w, b = Parameter yang akan dicari nilainya

y_i = Keluaran dari data x_i

ω = Parameter model

Persamaan (7) digunakan untuk meminimalkan fungsi tujuan $\frac{1}{2} \|\omega\|^2$ atau memaksimalkan kuantitas $\|\omega\|^2$ dengan cara memperhatikan pembatas $y_i(w x_i + b) \geq 1$. Apabila *output* data $y_i = +1$, maka pembatas menjadi $(w x_i + b) \geq 1$ dan bila $y_i = -1$, pembatas menjadi $(w x_i + b) \geq -1$.

Pada awalnya SVM hanya digunakan untuk mengklasifikasi dua buah kelas (*binary classifier*). Penelitian terus berlanjut hingga menjadikan SVM bersifat multi kelas (*multi classifier*) dengan kemampuan mengklasifikasi lebih dari dua kelas. Untuk klasifikasi data dalam k kelas maka diharuskan membuat $\frac{k(k-1)}{2}$ model SVM *Biner*. Sebagai contoh untuk membuat klasifikasi 4 kelas maka, harus membangun $\frac{4(4-1)}{2} = 6$ buah SVM *Biner*. SVM *Biner* pertama dilatih dengan data latih dari kelas pertama dan kelas kedua untuk mengklasifikasikan data ke dalam C1 atau C2. SVM *Biner* kedua dilatih dengan data latih dari kelas C2 atau C3, dan seterusnya. Setiap kelas harus dibandingkan dengan tiga kelas lainnya. Cara *voting* dilakukan untuk menghasilkan kelas keputusan. Kelas yang paling sering menang adalah kelas keputusan [23].

2.7. Confusion Matrix

Algoritma SVM dapat dihitung akurasi kerjanya dengan menggunakan metode *Confusion Matrix* [24]. *Confusion matrix* sangat berguna untuk menganalisis kualitas model klasifikasi dalam mengenali kelas tuple-tuple yang ada [21]. Metode ini akan menghasilkan *F1 score*, *accuracy*, *precision* dan *recall*. *Accuracy* merupakan tingkat identifikasi yang menghasilkan persentase dari jumlah data uji yang diklasifikasikan secara benar oleh model klasifikasi [24]. Perhitungan *accuracy* dapat dilakukan dengan persamaan (8).

$$Accuracy = (TP + TN)/TD \quad (8)$$

1
Keterangan:

- TP (*True Positive*) yaitu jumlah tuple positif dilabeli dengan benar oleh model klasifikasi.
- TN (*True Negative*) yaitu jumlah tuple negatif yang dilabeli dengan benar oleh model klasifikasi.
- TD yaitu Total Data

Menurut [25] *Precision* digunakan untuk mengukur kepastian yaitu berapa persentase list data yang diberikan label sebagai data positif maka pada kenyataannya juga benar. Perhitungan *precision* dapat dilakukan dengan persamaan (9).

$$Precision = TP/(TP + FP) \quad (9)$$

FP (*False Positive*) yaitu jumlah tuple negatif yang salah dilabeli oleh model klasifikasi.

Recall digunakan sebagai ukuran kelengkapan, yaitu berapa persentase list data positif yang diberikan label sebagai data positif atau bisa disebut juga dengan *sensitivity*. Perhitungan *recall* dapat dilakukan dengan persamaan (10).

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

FN (*False Negative*) yaitu jumlah tuple positif yang dilabeli salah oleh model klasifikasi.

F1 score merupakan rata-rata harmonik dari *precision* dan *recall* [26]. Perhitungan *f1 score* dapat dilakukan dengan persamaan (11).

$$F1 = 2 \frac{Precision * Recall}{Precision + Recall} \quad (11)$$

2.8. Cross Validation

Validasi dilakukan dengan metode *Cross Validation*. *Cross Validation* adalah sebuah metode untuk memprediksi keakuratan data pengujian [27]. Metode *Cross Validation* yang digunakan adalah *K-fold cross validation* yaitu, sebuah teknik yang dapat melakukan pengulangan data latih (*training*) dan data uji (*testing*) dengan sebanyak *k* pengulangan dan pembagian $1/k$ dari dataset, dimana $1/k$ tersebut akan digunakan sebagai data uji [8]. Sebagai analogi misalkan kita memiliki 500 data. Kita asumsikan $k = 5$ maka, keseluruhan data akan dibagi menjadi lima lipatan sehingga masing-masing memiliki 100 data. Setiap lipatan yang memiliki 100 data terlebih dahulu ditentukan mana yang sebagai data latih dan uji. Perbandingan data latih dan uji adalah 75%: 25% sehingga, 75 data latih dan 25 data uji. Berdasarkan kelima lipatan maka, terdapat 4 lipatan yaitu $4 \times 100 = 400$ data *training* dan, sisanya yaitu 100 sebagai lipatan data uji.

3. Hasil dan Pembahasan

3.1. Dataset

Dataset yang dihasilkan berupa postingan *programmer* terpilih melalui *survey* dan observasi pada group *facebook* PHP Indonesia. Hasil dataset berupa 2178 postingan teks terkait dunia *programming* dan teknologi informasi. Untuk kelas klasifikasi yang digunakan adalah indikator kinerja *programmer* berdasarkan pada aktivitas media sosial [4] yaitu *Attitude*, *Komunitas*, *Promosi*, *Portfolio*, *Share Knowledge & Experiments*, *Mentoring* dan *Opini Diskusi*. Format *dataset* disimpan dalam bentuk (.csv) menggunakan Microsoft Excel. *Dataset* dapat diakses secara *public* pada situs dengan alamat <https://www.kaggle.com/purwonopurwono/dataset-postingan-programmer-facebook>.

3.2. Performa Algoritma SVM

Pemodelan metode klasifikasi SVM dilakukan dengan cara membagi *dataset* menjadi data latih (*training*) dan data uji (*test*) dengan persentase 70% berbanding 30%. Contoh *dataset training* dapat dilihat pada Gambar 3 dan *dataset testing* dapat dilihat pada Gambar 4.

```
x_train.head()

1419  apa lihat dengar baca ajar perlu pikir banyak ...
261   selamat hari lahir brother panjang umur sukses...
2114  jalan akhir nih mysql yosemite bikin puyeng aj...
1291          tips guna emmet visual studio code
120   video workshop pandu tangan kasus missing valu...
Name: Posting, dtype: object
```

Gambar 3. Contoh dataset *training*

```
x_test.head()

1191  latih buat aplikasi video chat stack modern ja...
162   banyak ajar satu minggu akhir mulai rasa press...
377          minat klik
1888          beberapa tips kembang guna vue js
1816  bagaimana menambahkan script mereseize gambar j...
Name: Posting, dtype: object
```

Gambar 4. Contoh dataset *testing*

Algoritma SVM yang digunakan adalah Linear SVM. Hasil *precision*, *recall* dan *f1-score* dari metode SVM dapat dilihat pada Tabel 2.

Table 2. Performa model klasifikasi SVM

Indikator	Precision	Recall	F1-Score	Accuracy
Attitude	0,90	0,80	0,85	0,8
Komunitas	0,82	0,89	0,85	0.88652482

3

Klasifikasi Kinerja Programmer pada Aktivitas Media Sosial dengan Metode Support Vector Machines (Rusydi Umar, dkk)

Promosi	0,74	0,78	0,76	0.78181818
Portfolio	0,84	0,84	0,84	0.84210526
Share Knowledge & Experiments	0,85	0,64	0,73	0.63934426
Mentoring	0,00	0,00	0,00	0,0
Opini Diskusi	0,87	0,77	0,83	0.77419355

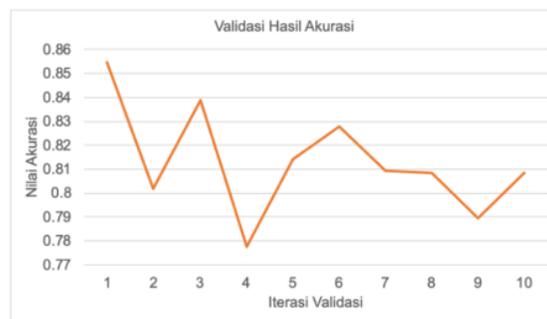
3.3. Validasi Hasil

Pengujian validasi akurasi algoritma SVM dengan menggunakan $k=10$ *cross validation*, yang bertujuan memastikan bahwa model tersebut sesuai atau tidak dengan sistem yang akan dibuat. Hasil uji validasi dapat dilihat nilai akurasinya pada Tabel 2.

Table 3. Validasi hasil

Pengujian Ke	Akurasi SVM
1	0,854545455
2	0,801843318
3	0,838709677
4	0,777777778
5	0,813953488
6	0,827906977
7	0,809302326
8	0,808411215
9	0,789719626
10	0,808411215

Berdasarkan Tabel 2. Akurasi tertinggi yang dimiliki oleh algoritma SVM sebesar 85,45% pada pengujian ke-1 dan nilai akurasi terendah pada pengujian ke-4 dengan nilai sebesar 77,7%. Algoritma SVM menghasilkan rata-rata tingkat akurasi dengan $k=10$ *cross validation* sebesar 81,3%, hasil akurasi rupanya lebih tinggi dibandingkan penelitian yang telah dilakukan oleh Mustika [6] yaitu sebesar 72,5% dan Wulandari [9] 81,26%, namun memiliki nilai akurasi yang lebih kecil dibandingkan dengan penelitian yang dilakukan Maulina [7] 95,8% dan Asiyah [8] 93,2%. Untuk mempermudah pengamatan, nilai iterasi akurasi metode SVM dengan $k=10$ *cross-validation* ditampilkan dalam sebuah grafik *line* yang dapat dilihat pada Gambar 5.



Gambar. 5 Grafik iterasi akurasi SVM dengan *cross-validation*

4. Kesimpulan

Berdasarkan hasil penelitian didapatkan bahwa performa dari metode SVM dalam klasifikasi kinerja *programmer* pada aktivitas media sosial memiliki tingkat akurasi yang cukup baik. Algoritma SVM menghasilkan rata-rata tingkat akurasi dengan $k=10$ *cross validation* sebesar 81,3%. Akurasi tertinggi yang dimiliki oleh algoritma SVM sebesar 85,45% pada pengujian ke-1 dan nilai akurasi terendah pada pengujian ke-4 dengan nilai sebesar 77,7%.

Kontribusi penelitian ini adalah menghasilkan fitur baru dalam proses *rekrutmen* yang memanfaatkan media sosial dengan cara melakukan klasifikasi postingan setiap kandidat *programmer* dengan Algoritma SVM. Penelitian selanjutnya ialah membuat pemodelan klasifikasi dengan objek citra atau video dari konten media sosial. Hasil akhir penelitian ini berupa sistem berbasis mobile atau website yang dapat membantu pekerjaan HRD dalam melakukan rekrutmen kandidat *programmer* terbaik.

Daftar Pustaka

- [1] I. Lewenusa, "Rekayasa Kebutuhan Perangkat Lunak Pada Perusahaan Skala Kecil Dan Menengah Dengan Pendekatan Soft System Methodology (Ssm) – Studi Kasus Pt Xyz," *Comput. J. Comput. Sci. Inf. Syst.*, vol. 1, no. 1, p. 49, 2017.
- [2] A. Z. Fauzia, S. Maslihah, and H. Ihsan, "Pengaruh Tipe Kepribadian Terhadap Self-Disclosure pada Dewasa Awal Pengguna Media Sosial Instagram di Kota Bandung," *Jakarta*, vol. 3, no. 3, pp. 151–160, 2019.
- [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] 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.
- [5] M. M. Tadesse, H. Lin, B. Xu, and L. Yang, "Personality Predictions Based on User Behavior on the Facebook Social Media Platform," *IEEE Access*, vol. 6, pp. 61959–61969, 2018.
- [6] A. Mustika and M. Affandes, "Penerapan Metode Support Vector Machine Dalam Klasifikasi Sentimen Tweet Public Figure," *Sentra*, pp. 978–979, 2015.
- [7] D. Maulina and R. Sagara, "Klasifikasi Artikel Hoax Menggunakan Support Vector Machine Linear Dengan Pembobotan Term Frequency – Inverse Document Frequency," *Mantik Penusa*, vol. 2, no. 1, pp. 35–40, 2018.
- [8] 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.
- [9] Y. Wulandari, Wiranto, and Wiharto, "Klasifikasi Keluhan Pelanggan Berbasis Twitter Menggunakan Metode Support Vector Machine (Svm) (Studi Kasus Pos Indonesia)," *Pus. Dokumentasi dan Inf. Ilm. UNS*, 2017.
- [10] 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.
- [11] A. T. J. Harjanta, "Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining," *Inform. UPGRIS*, vol. 1, pp. 1–9, 2015.
- [12] F. Rahutomo and A. R. T. H. Ririd, "Evaluasi Daftar Stopword Bahasa Indonesia," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 1, p. 41, 2019.
- [13] A. Alajmi and E. mostafa Saad, "Toward an ARABIC Stop-Words List Generation Toward an ARABIC Stop-Words List Generation," no. January 2012, 2018.
- [14] M. A. Fauzi, "Text Pre-Processing," 2016.
- [15] B. Herwijayanti, D. E. Ratnawati, and L. Muflikhah, "Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 306–312, 2018.
- [16] A. Taufik, "Komparasi Algoritma Text Mining Untuk Klasifikasi Review Hotel," *J. Tek. Komput. AMIK BSI*, vol. IV, no. 2, pp. 69–74, 2018.
- [17] 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.
- [18] I. Maulida, A. Suyatno, H. Rahmania Hatta, and U. Mulawarman, "Seleksi Fitur Pada Dokumen Abstrak Teks Bahasa Indonesia Menggunakan Metode Information Gain," *Oktober 2016 Ijccs*, vol. 17, no. 2, pp. 1–5, 2016.
- [19] I. Riadi, R. Umar, and F. D. Aini, "Analisis Perbandingan Detection Traffic Anomaly Dengan Metode Naive Bayes Dan Support Vector Machine (Svm)," *Ilk. J. Ilm.*, vol. 11, no. 1, p. 17, 2019.
- [20] M. Y. Ramadan, D. Syauby, and Tibyani, "Implementasi Metode Klasifikasi Support Vector Machine (SVM) Terhadap Pemakaian Minyak Goreng," vol. 3, no. 2, pp. 1669–1677, 2019.
- [21] Suyanto, *Machine Learning Tingkat Dasar dan Lanjut*. Bandung: Informatika, 2018.
- [22] A. S. Ritonga and E. S. Purwaningsih, "Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding)," *Ilm.*

- Educic*, vol. 5, no. 1, pp. 17–25, 2018.
- [23] N. I. Widiastuti, E. Rainarli, and K. E. Dewi, "Peringkasan dan Support Vector Machine pada Klasifikasi Dokumen," *J. Infotel*, vol. 9, no. 4, p. 416, 2017.
- [24] M. F. Rahman, D. Alamsah, M. I. Darmawidjadja, and I. Nurma, "Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN)," *J. Inform.*, vol. 11, no. 1, p. 36, 2017.
- [25] I. Saputra and D. Rosiyadi, "Perbandingan Kinerja Algoritma K-Nearest Neighbor , Naïve Bayes Classifier dan Support Vector Machine dalam Klasifikasi Tingkah Laku Bully pada Aplikasi Whatsapp," *Fakt. Exacta*, vol. 12, no. 2, pp. 101–111, 2019.
- [26] G. Hackeling, *Mastering Machine Learning with scikit-learn*. Birmingham: Packt Publishing, 2014.
- [27] 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.

HASIL CEK_60020397_Point-C69-IRD-850GB-Klasifikasi Kinerja Programmer pada Aktivitas Media Sosial dengan Metode Support Vector Machines

ORIGINALITY REPORT

9%

SIMILARITY INDEX

7%

INTERNET SOURCES

0%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Brawijaya Student Paper	2%
2	repository.bsi.ac.id Internet Source	2%
3	garuda.ristekbrin.go.id Internet Source	2%
4	repository.uin-suska.ac.id Internet Source	1%
5	Submitted to Universitas Amikom Student Paper	1%
6	media.neliti.com Internet Source	1%
7	www.scribd.com Internet Source	1%
8	zombiedoc.com Internet Source	1%

Exclude quotes On

Exclude bibliography On

Exclude matches < 1%