Perbandingan Performa Algoritma GA-SVM dan BOA-SVM dalam Mengklasifikasi Artikel Berita Berbahasa Indonesia

Winda Ramadhanty Pratiwi[1], Ricky Eka Putra[2]

[1,2] Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya
[1]windapratiwi16051204014@mhs.unesa.ac.id
[2]rickyeka@unesa.ac.id

***Abstrak*— Kemajuan teknologi mendorong industri media massa untuk beralih dari media cetak ke media digital seperti situs berita *online*. Secara umum, situs berita *online* mengelompokkan artikel berita menjadi beberapa kategori untuk memudahkan pembaca dalam memilih artikel berita sesuai topik yang diminati. Hanya saja pengelompokkan artikel berita dilakukan secara manual dengan mempelajari isi atau kontennya terlebih dahulu. Apabila jumlah artikel berita yang ingin dikelompokkan sangat banyak, maka waktu yang dibutuhkan juga tidak sedikit. Hal ini menjadi dasar untuk membuat sistem klasifikasi otomatis dalam mengelompokkan artikel berita. Artikel berita dari IDNTimes menjadi sumber data di penelitian ini. Algoritma TF-IDF digunakan sebagai ekstraksi fitur, sedangkan SVM digunakan sebagai klasifikasi data. Penelitian ini akan membandingkan *Genetic Algorithm* (GA) dan *Bayesian Optimization Algorithm* (BOA) dalam mengoptimasi SVM untuk mengklasifikasi artikel berita. Berdasarkan hasil pengujian sistem, algoritma GA-SVM memiliki akurasi 93.80% dan waktu pemrosesan 969.015 detik, sedangkan BOA-SVM memiliki akurasi 94.79% dan waktu pemrosesan 829.921 detik. Kesimpulannya, algoritma BOA lebih baik daripada GA dalam mengoptimasi SVM untuk mengklasifikasi artikel berita.**

***Kata Kunci*— klasifikasi artikel berita, *support vector machine*, *bayesian optimization*, *genetic algorithm*.**

I. PENDAHULUAN

Kemajuan teknologi mendorong industri media massa untuk beralih dari media cetak ke media digital seperti situs berita *online*. Terhitung sejak tahun 2011-2016, jumlah pembaca berita *online* naik hingga 500%. Berdasarkan survei yang dilakukan oleh Nielsen Indonesia, konsultan pemasaran, pada tahun 2017 jumlah pembaca berita *online* di Indonesia mencapai 6 juta orang, melebihi jumlah pembaca media cetak yang hanya mencapai 4,5 juta orang [1]. Studi Nielsen pada tahun 2018 juga mengungkap bahwa konsumen Indonesia menghabiskan waktu sekitar 3 jam 14 menit per harinya untuk membaca berita *online* [2].

Secara umum, situs berita *online* mengelompokkan artikel berita menjadi beberapa kategori seperti teknologi, otomotif, olahraga, wisata, dan *lifestyle*. Pengelompokkan ini dilakukan dengan tujuan memudahkan pembaca untuk memilih artikel berita sesuai topik yang diminati. Selama ini pengelompokkan artikel berita dilakukan secara manual dengan mempelajari isi atau kontennya terlebih dahulu. Apabila jumlah artikel berita yang ingin dikelompokkan sangat banyak, maka waktu yang dibutuhkan untuk mempelajari isi atau kontennya pun juga tidak sedikit. Pengelompokkan artikel berita juga membutuhkan ketelitian yang tinggi agar tidak terjadi kesalahan dalam menentukan kategori. Oleh karena itu, sistem klasifikasi otomatis dalam mengelompokkan artikel berita diperlukan untuk mengatasi permasalahan tersebut.

Klasifikasi adalah proses pengelompokkan objek dengan ciri-ciri atau karakteristik tertentu yang sama ke dalam beberapa kategori. Salah satu objek yang dapat diklasifikasi adalah teks. Klasifikasi teks secara otomatis dapat dilakukan dengan menggunakan ciri atau fitur kata yang muncul pada dokumen latih [3]. Klasifikasi teks dapat dilakukan secara manual atau otomatis menggunakan *machine learning* seperti Support Vector Machine – SVM, pendekatan model data toleransi kasar, dan pendekatan *Rules Association* [4].

Penelitian yang dilakukan oleh Siti Nur Asiyah dan Kartika Fithriasari pada tahun 2016 menunjukkan bahwa sistem yang dapat mengklasifikasi artikel berita secara otomatis dapat dilakukan dengan algoritma *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN). Penelitian ini membandingkan kinerja SVM dan KNN dalam mengklasifikasi artikel berita dari situs detik.com ke dalam lima kategori, yaitu news, finance, hot, sport, dan otomotif. Hasil dari penelitian ini adalah SVM kernel polynomial menghasilkan akurasi lebih baik daripada kernel linier. Apabila dibandingkan dengan KNN, maka SVM lebih baik daripada KNN dengan hasil akurasi, *recall, precision dan F-Measure* sebesar 93.2%, 93.2%, 93.63% dan 93.14% [5].

Penelitian tentang klasifikasi artikel berita juga pernah dilakukan oleh Batoul Aljaddouh dan Nishith A. Kotak pada tahun 2020. Penelitian ini menggunakan algoritma TF-IDF dan SVM dalam implementasinya. Dataset yang digunakan adalah gabungan antara 2 dataset, yaitu artikel berita BBC dan dataset *20 news group*. Penelitian ini menggunakan lebih dari 3500 artikel berbeda untuk membangun pengelompokan artikel menjadi 8 kelas, yaitu *Atheism, Bussines, Car, Entertainment, Politic, Space, Sport, Technology*. Berdasarkan hasil yang diperoleh dari penelitian ini, SVM mampu menghasilkan akurasi pada proses *training* sebesar 96,40% dan 89,70% pada proses *testing* terhadap 160 artikel [6].

Kedua penelitian sebelumnya menunjukkan bahwa algoritma SVM mampu menghasilkan akurasi tinggi untuk mengklasifikasi artikel berita. Hal itu karena SVM mampu mengklasifikasi data dengan menemukan *hyperplane* atau fungsi keputusan terbaik. *Hyperplane* tersebut dibuat dengan memperhitungkan posisi *Support Vector* atau titik-titik data

terdekat saja sehingga data berdimensi tinggi tidak akan menurunkan kinerja SVM. Algoritma SVM juga dapat dioptimasi sehingga kinerja SVM menjadi lebih baik. Algoritma yang dapat mengoptimasi SVM contohnya seperti *Grid Search, Genetic Algorithm, dan Bayesian Optimization Algorithm* [7].

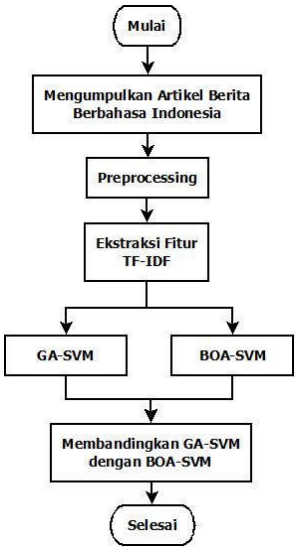
Genetic Algorithm (GA) adalah metodologi pencarian optimasi adaptif umum yang mendukung analogi seleksi alam Darwin dan sistem biologi genetika, dapat menjadi alternatif yang menjanjikan untuk pencarian heuristik standar [8]. Algoritma GA memiliki konsep serupa dengan proses evolusi dimana individu-individu terbaik akan bertahan hidup, sedangkan individu-individu yang lemah akan punah. GA bekerja dengan sekumpulan kandidat solusi yang disebut populasi. Sekumpulan kandidat tersebut akan diseleksi melalui suatu fungsi yang disebut *fitness function*. Melalui *fitness function*, akan diketahui *fitness value*-nya untuk kemudian dijadikan referensi pada proses GA berikutnya. Menurut Iwan dkk, algoritma GA dapat mengoptimasi SVM 15,9 kali lebih cepat daripada algoritma lain seperti *Grid Search* [9].

Bayesian Optimization Algorithm (BOA) adalah salah satu algoritma yang juga dapat mengoptimasi SVM selain *Grid Search* dan GA. BOA bekerja dengan menguji beberapa kombinasi *hyperparamater* SVM dalam interval nilai yang telah ditentukan. BOA memanfaatkan *Gaussian Process* sebagai model pengganti dari fungsi objektif. BOA juga memanfaatkan fungsi akuisi untuk menyeleksi titik sampel yang akan diuji di dalam proses selanjutnya. Dibandingkan dengan *Grid Search*, BOA memiliki waktu komputasi yang relatif lebih singkat dalam melakukan pencarian *hyperparameter* [10].

Berdasarkan uraian diatas, maka penelitian ini akan berfokus pada perbandingan *Genetic Algorithm* (GA) dan *Bayesian Optimization Algorithm* (BOA) dalam mengoptimasi algoritma SVM untuk mengklasifikasi artikel berita berbahasa Indonesia. Hal ini bertujuan untuk mengetahui bagaimana kinerja SVM setelah dioptimasi menggunakan GA dan BOA, sekaligus mengetahui algoritma mana yang lebih baik antara GA dengan BOA dalam mengklasifikasi artikel berita bahasa Indonesia. Artikel berita dikumpulkan dalam bentuk file txt untuk kemudian diolah dalam program *Python*. Algoritma *Term Frequency and Inverse Document Frequency* (TF-IDF) dipilih sebagai ekstrasi fitur dan *Support Vector Machine* (SVM) dipilih sebagai algoritma klasifikasi data.

II. METODOLOGI

Tujuan dari penelitian ini yaitu untuk mengetahui performa dari algoritma GA-SVM dan BOA-SVM dalam mengklasifikasi artikel berita. Artikel berita yang diklasifikasi adalah artikel berita berbahasa Indonesia dari situs IDNTimes. Sebanyak 500 artikel digunakan sebagai dataset dengan 5 kategori berita sebagai kelas data. Masing-masing kelas terdiri dari 100 artikel berita. Ekstrasi fitur yang digunakan dalam penelitian ini adalah 1000 vektor kata dari TF-IDF. Gambaran umum penelitian dapat dilihat pada Gbr 1.



Gbr 1. Alur Kerja Penelitian

A. Mengumpulkan Artikel Berita

Dataset dikumpulkan secara manual dengan membuka situs IDNTimes dan menyalin teks artikel satu persatu untuk kategori teknologi, otomotif, olahraga, wisata, dan *lifestyle* dengan mengecualikan gambar dan video. Setiap artikel yang telah disalin akan disimpan dalam format *file txt*. Setiap artikel yang berbeda kategori akan disimpan di folder yang berbeda sehingga akan ada 5 folder.

B. Preprocessing

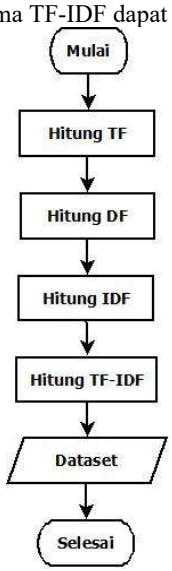
Sebelum digunakan untuk proses *training* dan testing dataset akan diolah terlebih dahulu melalui tahap *preprocessing*. Tahap *preprocessing* meliputi *case folding,* penghapusan karakter dan simbol non alfabet, filter *stopword*, *stemming,* dan tokenisasi. *Preprocessing* dilakukan dengan tujuan untuk menormalisasi dataset agar siap digunakan untuk proses *training* dan *testing*.

Case folding adalah tahap dimana seluruh teks dari artikel berita diubah menjadi huruf kecil. Kemudian karakter-karakter dan simbol-simbol pada teks artikel berita seperti emoticon dan sejenisnya dihilangkan karena tidak mengandung informasi yang penting. Beberapa kata umum yang muncul dalam jumlah besar pada teks artikel berita seperti "ke", "di", "yang" merupakan *stopwords* yang wajib dihilangkan karena kata-kata tersebut tidaklah informatif. Setelah *stopwords* dihilangkan, dilakukan proses *stemming*, yaitu proses pemotongan imbuhan pada kata sehingga kata imbuhan kembali menjadi kata dasar. Langkah terakhir yaitu tokenisasi, dimana sekumpulan kalimat pada teks artikel berita dipecah menjadi token atau kata.

C. Ekstraksi Fitur TF-IDF

Term Frequency and Inverse Document Frequency (TF-IDF) adalah algoritma yang dapat mengonversi teks menjadi representasi angka yang nantinya dapat dipelajari oleh *machine learning*. TF-IDF biasa digunakan sebagai ekstrasi fitur dalam pemrosesan teks. TF-IDF bekerja dengan menghitung bobot dengan cara integrasi antara nilai *Term Frequency* (TF) dan

Inverse Document Frequency (IDF) dari setiap kata pada seluruh dokumen [5] sehingga dapat digunakan untuk mengetahui tingkat kepentingan sebuah kata dalam dokumen. Alur kerja dari algoritma TF-IDF dapat dilihat pada Gbr 2.



Gbr 2. Alur Kerja TF-IDF

Nilai TF menyatakan frekuensi sebuah kata yang keluar di dalam sebuah dokumen. Rumus untuk menghitung nilai TF dapat dilihat pada (1). Semakin besar nilai TF sebuah kata menunjukkan bahwa kata tersebut sering keluar di dalam sebuah dokumen.

$$TF = \frac{jumlah\ kata\ t\ yang\ keluar\ pada\ dokumen}{jumlah\ seluruh\ kata\ pada\ dokumen} \quad (1)$$

Nilai IDF menyatakan seberapa unik sebuah kata di dalam dokumen. Nilai DF perlu dihitung terlebih dahulu supaya dapat menghitung nilai IDF. Rumus untuk menghitung nilai DF dapat dilihat pada persamaan (2).

$$DF = \frac{jumlah\ seluruh\ dokumen}{jumlah\ dokumen\ yang\ mengandung\ kata\ t} \quad (2)$$

Setelah diketahui nilai DF, nilai IDF dapat dihitung dengan rumus (3). Semakin tinggi nilai IDF sebuah kata menunjukkan bahwa kata tersebut jarang keluar pada sebuah dokumen atau langka.

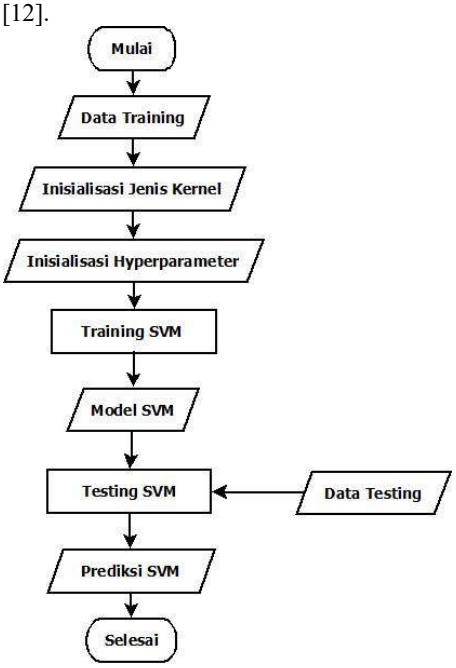
$$IDF = \log(DF) \quad (3)$$

Setelah diketahui nilai TF dan IDF, selanjutnya adalah menghitung nilai bobot TF-IDF dengan cara mengalikan nilai TF dan nilai IDF seperti persamaan (4).

$$Bobot\ TF - IDF = TF \times (IDF) \quad (4)$$

D. Support Vector Machine (SVM)

SVM merupakan salah satu metode terbaik yang bisa dipakai dalam permasalahan klasifikasi [11]. Umumnya SVM digunakan untuk memecahkan permasalahan linier. Namun SVM juga dapat memecahkan masalah non linier dengan mentransformasi data ke ruang berdimensi tinggi menggunakan fungsi kernel. Kernel yang paling sering digunakan pada SVM adalah Linier, RBF, *Polynomial*, dan *Sigmoid* [12].



Gbr 3. Alur Kerja SVM

Sebelum melakukan *training*, jenis kernel dan *hyperparameter* SVM perlu diinisialisasi terlebih dahulu [10]. Nilai *hyperparameter* SVM yang umum digunakan adalah nilai *cost*. Untuk beberapa kernel seperti kernel RBF, dibutuhkan nilai *hyperparameter* tambahan seperti nilai *gamma*. Nilai *cost* pada SVM mempengaruhi *margin* dan *error* saat klasifikasi, sedangkan nilai *gamma* mempengaruhi performa kernel RBF dalam melakukan klasifikasi. Alur kerja dari SVM dapat dilihat pada Gbr 3.

Jenis kernel dan *hyperparameter* berpengaruh dalam proses *training* untuk menghasilkan model SVM yang tepat. Model tersebut nantinya akan digunakan oleh SVM untuk melakukan prediksi terhadap data baru yang belum pernah di-*training* oleh SVM sebelumnya. Adapun model SVM dapat dinyatakan dalam bentuk persamaan (5).

$$f(\emptyset(x)) = sign(w.\emptyset(x) + b) \quad (5)$$

Dimana :
$f(\emptyset(x))$ = fungsi *hyperplane*
w = bobot
$\emptyset(x)$ = fungsi kernel
b = bias

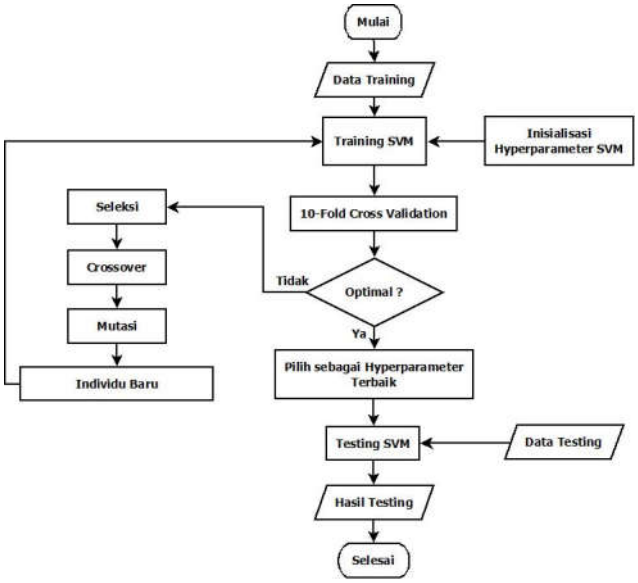
Pemilihan *hyperparameter* yang tepat akan meningkatkan performa dari SVM. Namun tidak ada patokan tertentu dalam menentukan nilai *hyperparameter* SVM. Oleh karena itu untuk mencari nilai *hyperparameter* yang tepat, dibutuhkan algoritma

tambahan untuk mengoptimasi SVM, seperti *Genetic Algorithm* (GA) dan *Bayesian Optimization Algorithm* (BOA).

E. Genetic Algorithm

Genetic Algorithm (GA) adalah metodologi pencarian optimasi adaptif umum yang mendukung analogi seleksi alam Darwin dan sistem biologi genetika, dapat menjadi alternatif yang menjanjikan untuk pencarian heuristik standar [8]. Algoritma GA memiliki konsep serupa dengan proses evolusi dimana individu-individu terbaik akan bertahan hidup, sedangkan individu-individu yang lemah akan punah. Dengan cara itu, GA dapat digunakan untuk menemukan solusi untuk menyelesaikan permasalahan dengan menyeleksi sekumpulan kandidat yang solusi yang ada.

Proses dari GA diawali dengan menginisialisasi sekumpulan individual yang disebut populasi. Setiap individual tersebut merupakan kandidat solusi yang dicari untuk menyelesaikan masalah. Masing-masing individual terdiri dari sekumpulan variabel yang disebut gen. Adapun sekumpulan dari gen disebut kromosom. Biasanya gen-gen tersebut direpresentasikan dalam bentuk *string*. *String* tersebut menggunakan bilangan biner. GA menggunakan fungsi *fitness* untuk menyeleksi populasi. Fungsi *fitness* akan menghasilkan nilai *fitness* untuk tiap individu pada populasi. Individu yang dicari oleh GA adalah individu yang menghasilkan nilai *fitness* tertinggi. Individu dengan nilai *fitness* tertinggi merupakan solusi yang ditemukan oleh GA.

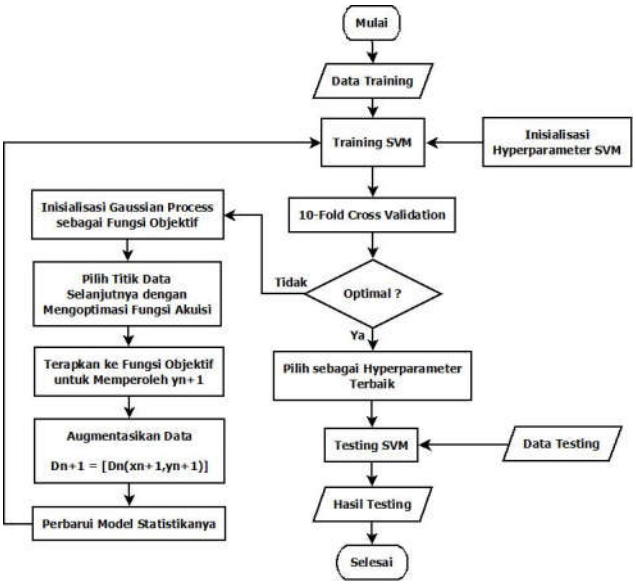


Gbr 4. Alur Kerja GA-SVM

GA menggunakan 3 operator untuk bekerja, yaitu seleksi, *crossover* dan mutasi. Konsep dari seleksi adalah menyeleksi individual yang memiliki nilai *fitness* tertinggi untuk dilanjutkan ke proses selanjutnya. Alur kerja dari GA dengan SVM sebagai fungsi *fitness*-nya dapat dilihat pada Gbr 4. GA akan memilih 2 pasang individual (kromosom) untuk melanjutkan ke proses *crossover*. *Crossover* adalah proses dimana dua pasang kromosom melakukan kawin silang dengan cara menukar gen satu sama lain sehingga menciptakan populasi baru. Proses selanjutnya yaitu mutasi, dimana gen-gen pada kromosom bermutasi untuk menjaga keberagaman antar populasi. Proses algoritma GA berhenti jika populasi telah konvergen.

F. Bayesian Optimization Algorithm

Bayesian Optimization Algorithm (BOA) merupakan algoritma yang menggunakan pendekatan Teorema *Bayes* untuk melakukan pencarian guna menemukan fungsi tujuan maksimum atau minimum. Pendekatan ini bermanfaat untuk menyelesaikan fungsi objektif yang kompleks, rumit, dan mahal untuk dievaluasi. BOA menggunakan fungsi pengganti (*surrogate function*) dapat memperkirakan fungsi tujuan sehingga dapat mengarahkan pengambilan sampel di masa mendatang. Pengambilan sampel melibatkan penggunaan *posterior* secara hati-hati dalam fungsi yang dikenal sebagai fungsi akuisisi (*acquition function*), misalnya untuk memperoleh lebih banyak sampel. Fungsi akuisi adalah teknik di mana *posterior* digunakan untuk memilih sampel berikutnya dalam ruang pencarian. Setelah sampel tambahan dan evaluasinya dikumpulkan melalui fungsi tujuan, sampel tersebut ditambahkan ke data D kemudian *posterior* diperbarui. Proses ini terus diulangi sampai hasil maksimum atau minimum dari fungsi tujuan ditemukan, atau sampai *resource* habis .



Gbr 5. Alur Kerja BOA-SVM

BOA menjadi salah satu algoritma yang juga dapat mengoptimasi SVM selain *Grid Search* dan GA. BOA bekerja dengan menguji beberapa kombinasi *hyperparamater* SVM dalam interval nilai yang telah ditentukan. BOA memanfaatkan *Gaussian Process* sebagai model pengganti dari fungsi objektif. BOA juga memanfaatkan fungsi akuisi untuk menyeleksi titik sampel yang akan diuji di dalam proses selanjutnya. Gambaran dari proses yang terjadi pada BOA dapat dilihat pada Gbr 5. Dibandingkan dengan *Grid Search*, BOA memiliki waktu

komputasi yang relatif lebih singkat dalam melakukan pencarian *hyperparameter* [10].

G. *Perhitungan Akurasi*

Nilai akurasi digunakan sebagai tolok ukur kedekatan antara nilai sebenarnya dengan nilai prediksi dari sistem yang dibangun pada penelitian. Akurasi menggambarkan ketepatan sasaran dari penelitian dalam menyelesaikan masalah. Rumus menghitung akurasi adalah sebagai berikut:

$$Akurasi = \frac{Jumlah\ data\ benar}{Jumlah\ keseluruhan\ data}\ x\ 100\% \qquad (6)$$

Untuk menguji keberhasilan dari sistem yang dibangun pada penelitian ini, dilakukan perhitungan akurasi seperti pada persamaan (6). Nilai akurasi dinyatakan dalam bentuk persen (%). Semakin tinggi nilai persentase akurasi, semakin baik pula performa dari sistem yang dibangun.

III. HASIL PEMBAHASAN

Artikel berita dari IDN Times digunakan sebagai dataset pada penelitian ini. Penelitian ini akan membangun sistem yang dapat mengklasifikasi artikel berita dari IDN Times menjadi beberapa kategori, yaitu teknologi, otomotif, olahraga, wisata, dan *lifestyle*. Sistem yang dibangun pada penelitian ini menggunakan bahasa pemrograman *Python*. Penelitian ini akan berfokus pada perbandingan performa algoritma GA- SVM dengan BOA-SVM dalam mengklasifikasi artikel berita.

A. *Pengumpulan Dataset*

Pengumpulan dataset dilakukan dengan menyalin artikel berita dari situs IDN Times yang beralamat https://www.idntimes.com untuk kemudian disimpan ke dalam file berformat *txt* artikel berita yang disalin adalah artikel berita yang sudah dilabeli dengan kategori teknologi, otomotif, olahraga, wisata, dan *lifestyle* oleh IDN Times. File artikel berita kemudian disimpan dalam folder yang sesuai dengan kategorinya sehingga akan ada 5 folder untuk 5 kategori file artikel berita.

File artikel berita yang digunakan sebagai dataset pada penelitian ini memiliki cara penamaan yang khusus. Untuk file artikel berita dengan kategori teknologi akan diberi nama *Teknologi_angka* (dimana angka yang dimaksud adalah angka 1 sampai dengan 100). Untuk file artikel berita dengan kategori otomotif akan diberi nama *Otomotif_angka*. Untuk file artikel berita dengan kategori olahraga akan diberi nama *Olahraga_angka*. Untuk file artikel berita dengan kategori wisata akan diberi nama *Wisata_angka*. Untuk file artikel berita dengan kategori *lifestyle* akan diberi nama *Lifestyle_angka* hal ini dilakukan agar memudahkan program *Python* dalam membaca file-file tersebut sebagai dataset.

Tabel 1 merupakan rincian jumlah dataset yang digunakan pada penelitian ini.

Tabel 1
Rincian Jumlah Dataset Artikel Berita IDN Times

Kategori	Jumlah
Teknologi	100
Otomotif	100
Olahraga	100
Wisata	100
Lifestyle	100
Total	500

B. *Preprocessing*

Dataset yang telah terkumpul akan melalui tahap *preprocessing* terlebih dahulu sebelum siap digunakan. Metode yang digunakan sebagai *preprocessing* adalah ekstraksi fitur menggunakan algoritma TF-IDF. Algoritma TF-IDF akan mengonversi teks artikel berita menjadi representasi angka yang nantinya dapat dipelajari oleh algoritma *machine learning* seperti *Support Vector Machine* (SVM).

Teks artikel berita akan melalui *case folding*, *filtering*, dan tokenisasi. *Case folding* adalah proses penyeragaman karakter, dalam hal ini mengonversi semua huruf menjadi huruf kecil. Kemudian *filtering* yaitu menghapus karakter non-alfabet dan *stopwords* dari teks artikel berita. Terakhir adalah tokenisasi yaitu memecah teks menjadi kata per kata (token). Token tersebut nantinya diproses oleh TF-IDF.

TF-IDF menentukan tingkat kepentingan sebuah kata dalam dokumen dengan langkah-langkah seperti pada Gbr 2. Skor TF-IDF akan digunakan sebagai fitur yang dipelajari oleh SVM. Contoh hasil perhitungan TF-IDF dapat dilihat pada Tabel 2.

Tabel 2
Contoh Hasil Perhitungan TF-IDF

Term	Data							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Hubungan	0.19	0.10	0.00	0.00	0.00	0.00	0.00	0.00
Belajar	0.02	0.00	0.00	0.00	0.00	0.00	0.09	0.00
Kondisi	0.02	0.00	0.00	0.00	0.00	0.02	0.03	0.00
Masalah	0.11	0.02	0.05	0.00	0.00	0.00	0.00	0.39
Seru	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00

C. *Pengujian Sistem Menggunakan GA-SVM*

Pengujian sistem yang dilakukan pertama kali adalah pengujian menggunakan algoritma GA-SVM. Algoritma GA bekerja sebagai algoritma yang mengoptimasi SVM dengan mencari nilai *hyperparameter* yang tepat agar menghasilkan akurasi yang lebih baik. Nilai *hyperparameter* yang dicari adalah nilai *cost* dan *gamma.*

Interval nilai *cost* dan *gamma* yang dicari oleh GA adalah 0.01 sampai 100. Kernel SVM yang digunakan adalah Kernel RBF. Penelitian ini menggunakan *10-Fold Cross Validation* untuk membagi dataset menjadi data *training* dan data *testing* secara acak. Parameter GA yang digunakan pada penelitian ini antara lain *population_size=50*, *gene_mutation_prob=0.10, gene_crossover_prob=0.5, tournament_size=2, generations_number=10*.

```
Lenovo@Rama-Comp MINGW64 /e/Project/Python/Klasifikasi Topik Berita
$ python main.py
Types [2, 2] and maxint [24, 24] detected
--- Evolve in 625 possible combinations ---
gen     nevals  avg     min     max     std
0       50      0.25768 0.214   0.938   0.171888
1       32      0.3588  0.214   0.938   0.2896
2       31      0.46064 0.214   0.938   0.342629
3       27      0.562   0.214   0.938   0.361256
4       29      0.69256 0.214   0.938   0.341972
5       29      0.79392 0.214   0.938   0.28817
6       29      0.89456 0.214   0.938   0.17194
7       35      0.92352 0.214   0.938   0.10136
8       32      0.938   0.938   0.938   0
9       31      0.938   0.938   0.938   0
10      39      0.92352 0.214   0.938   0.10136
Best individual is: {'C': 3.208115532823594e+37, 'gamma': 1.023292992280754}
with fitness: 0.938
None
0.938 {'C': 3.208115532823594e+37, 'gamma': 1.023292992280754}
Waktu Pemrosesan =  969.015625 detik
```

Gbr 6. Contoh Hasil Pengujian Sistem Menggunakan GA-SVM

Berdasarkan percobaan yang telah dilakukan, algoritma GA dapat mengoptimasi SVM sehingga menghasilkan akurasi 93.80% untuk nilai *cost* 3.208 dan *gamma* 1.023. Hasil dari pengujian sistem menggunakan GA-SVM dapat dilihat pada Gbr 6 dan Tabel 3.

Tabel 3
Contoh Hasil Pengujian Sistem Menggunakan GA-SVM

Generation	Nevals	Average	Min	Max	Std
0	50	0.25768	0.214	0.938	0.171888
1	32	0.3588	0.214	0.938	0.2896
2	31	0.46064	0.214	0.938	0.342629
3	27	0.562	0.214	0.938	0.361256
4	29	0.69256	0.214	0.938	0.341972
5	29	0.79392	0.214	0.938	0.28817
6	29	0.89456	0.214	0.938	0.17194
7	35	0.92352	0.214	0.938	0.10136
8	32	0.938	0.938	0.938	0
9	31	0.938	0.938	0.938	0
10	39	0.92352	0.214	0.938	0.10136

D. *Pengujian Sistem Menggunakan BOA-SVM*

Pengujian sistem yang dilakukan selanjutnya adalah pengujian menggunakan algoritma BOA-SVM. Algoritma BOA digunakan sebagai algoritma yang mengoptimasi SVM seperti GA. BOA akan mencari nilai *hyperparameter* SVM yang tepat agar menghasilkan akurasi yang lebih baik. Nilai *hyperparameter* yang dicari adalah nilai *cost* dan *gamma*.

Interval nilai *cost* dan *gamma* yang dicari oleh BOA adalah 0.01 sampai 100. Kernel SVM yang digunakan adalah Kernel RBF. Penelitian ini menggunakan *10-Fold Cross Validation* untuk membagi dataset menjadi data *training* dan data *testing* secara acak. Parameter BOA yang digunakan pada penelitian ini antara lain *init_points=50, n_iter=50*.

Berdasarkan percobaan yang telah dilakukan, algoritma BOA dapat mengoptimasi SVM sehingga menghasilkan akurasi 94.79% untuk nilai *cost* 100 dan *gamma* 0.010. Hasil dari pengujian sistem menggunakan BOA-SVM dapat dilihat pada Tabel 4.

Tabel 4
Contoh Hasil Pengujian Sistem Menggunakan BOA-SVM

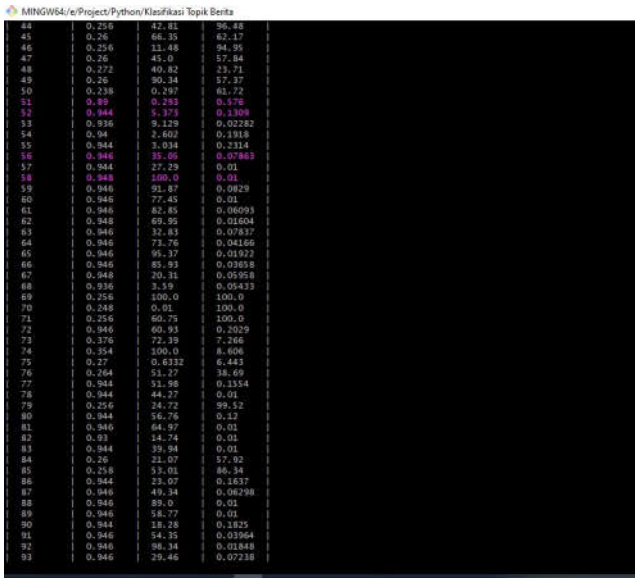
Iterasi	Target	Cost	Gamma
1	0.258	41.71	72.04
4	0.264	18.63	34.56
10	0.282	14.05	19.82
14	0.688	8.51	3.91
25	0.3	28.78	13.01
28	0.476	49.16	5.34
51	0.89	0.293	0.576
54	0.94	2.602	0.1918
58	0.948	100	0.010
83	0.944	39.94	0.010

MINGW64:/e/Project/Python/Klasifikasi Topik Berita

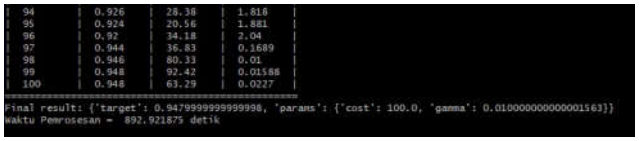
```
Lenovo@Rama-Comp MINGW64 /e/Project/Python/Klasifikasi Topik Berita
$ python main.py

--- Optimizing SVM ---
|   iter    |  target   |   cost    |   gamma   |
-------------------------------------------------
|  1        |  0.258    |  41.71    |  72.04    |
|  2        |  0.246    |  0.02144  |  30.24    |
|  3        |  0.342    |  14.68    |  9.243    |
|  4        |  0.264    |  18.63    |  34.56    |
|  5        |  0.26     |  39.68    |  53.89    |
|  6        |  0.258    |  41.93    |  68.53    |
|  7        |  0.256    |  20.45    |  87.81    |
|  8        |  0.26     |  2.748    |  67.05    |
|  9        |  0.26     |  41.74    |  55.87    |
|  10       |  0.282    |  14.05    |  19.82    |
|  11       |  0.256    |  80.08    |  96.83    |
|  12       |  0.258    |  31.35    |  69.24    |
|  13       |  0.256    |  87.64    |  89.46    |
|  14       |  0.688    |  8.514    |  3.915    |
|  15       |  0.256    |  16.99    |  87.82    |
|  16       |  0.264    |  9.844    |  42.12    |
|  17       |  0.26     |  95.79    |  53.32    |
|  18       |  0.266    |  69.19    |  31.56    |
|  19       |  0.258    |  68.65    |  83.46    |
|  20       |  0.258    |  1.839    |  75.02    |
|  21       |  0.258    |  98.89    |  74.82    |
|  22       |  0.258    |  28.05    |  78.93    |
|  23       |  0.262    |  10.33    |  44.79    |
|  24       |  0.268    |  90.86    |  29.37    |
|  25       |  0.3      |  28.78    |  13.01    |
|  26       |  0.258    |  1.947    |  67.89    |
|  27       |  0.268    |  21.17    |  26.56    |
|  28       |  0.476    |  49.16    |  5.346    |
|  29       |  0.294    |  57.42    |  14.68    |
|  30       |  0.258    |  58.93    |  69.98    |
|  31       |  0.264    |  10.24    |  41.41    |
|  32       |  0.264    |  69.44    |  41.42    |
|  33       |  0.26     |  5.005    |  53.59    |
|  34       |  0.26     |  66.38    |  51.49    |
|  35       |  0.26     |  94.46    |  58.66    |
|  36       |  0.296    |  90.34    |  13.76    |
|  37       |  0.258    |  13.94    |  80.74    |
|  38       |  0.292    |  39.77    |  16.54    |
|  39       |  0.264    |  92.75    |  34.78    |
|  40       |  0.258    |  75.08    |  72.6     |
|  41       |  0.26     |  88.33    |  62.37    |
|  42       |  0.264    |  75.1     |  34.9     |
|  43       |  0.256    |  27.0     |  89.59    |
```

Gbr 7. Contoh Hasil Pengujian Sistem Menggunakan BOA-SVM (1)



Gbr 8. Contoh Hasil Pengujian Sistem Menggunakan BOA-SVM (2)



Gbr 9. Contoh Hasil Pengujian Sistem Menggunakan BOA-SVM (3)

E. *Perbandingan Performa GA-SVM dan BOA-SVM*

Penelitian ini akan membandingkan performa algoritma GA-SVM dan BOA-SVM. Algoritma GA-SVM menghasilkan akurasi 93.80% untuk nilai *cost* 3.208 dan *gamma* 1.023, sedangkan algoritma BOA-SVM menghasilkan akurasi 94.79% untuk nilai *cost* 100 dan *gamma* 0.010. Untuk mengetahui perbedaan antara SVM setelah dioptimasi dan sebelum dioptimasi, dilakukan juga pengujian sistem menggunakan algoritma SVM (tanpa optimasi). Tabel 5 ini menunjukkan perbandingan performa antara SVM, GA-SVM, dan BOA-SVM.

Tabel 5
Perbandingan Performa SVM, GA-SVM, dan BOA-SVM.

Algoritma	Akurasi	Durasi (detik)
SVM	93.20%	8.578
GA-SVM	93.80%	969.015
BOA-SVM	94.79%	892.921

Berdasarkan Tabel 5, algoritma GA maupun BOA mampu meningkatkan akurasi dari SVM. Di antara algoritma GA dan BOA, BOA-SVM menghasilkan akurasi lebih tinggi dari GA-SVM. Sedangkan untuk durasi pemrosesan, BOA-SVM membutuhkan waktu lebih sedikit daripada GA-SVM. Jadi performa algoritma BOA-SVM dalam mengklasifikasi artikel berita berbahasa Indonesia lebih baik daripada GA-SVM.

IV. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, diperoleh kesimpulan bahwa artikel berita berbahasa indonesia dapat diklasifikasi berdasarkan kategorinya menggunakan algoritma *Support Vector Machine* (SVM). Akurasi dari algoritma SVM dapat meningkat bila dioptimasi menggunakan *Bayesian Optimization Algorithm* (BOA) dan *Genetic Algorithm* (GA). Klasifikasi artikel berita berdasarkan kategorinya menghasilkan akurasi 93.20% jika menggunakan SVM 93.80% jika menggunakan GA-SVM, 94.79% jika menggunakan BOA-SVM. Algoritma BOA dapat mengoptimasi SVM lebih baik daripada algoritma GA dilihat dari tingkat akurasi maupun waktu pemrosesan. Saran untuk penelitian selanjutnya adalah dapat melakukan perbandingan menggunakan algoritma yang lain.

REFERENSI

[1] L. Artama, ”Pemred Senayan Post : Pembaca Media Online Naik 500 Persen,” Suara Indonesia, 30 Maret 2019. [Online]. Available: https://www.suaraindonesia.co.id/read/5238/20190330/153619/pemred-senayan-post--pembaca-media-online-naik-500-persen.

[2] T. Fajar, ”Studi Nielsen: Pemirsa Indonesia Habiskan 5 Jam Nonton TV, 3 Jam Berselancar di Internet,” 3 Maret 2019. [Online]. Available: https://economy.okezone.com/read/2019/03/05/320/2025987/studi-nielsen-pemirsa-indonesia-habiskan-5-jam-nonton-tv-3-jam-berselancar-di-internet.

[3] E. R. K. E. D. Nelly Indriani, ”Peringkasan dan Support Vector Machine pada Klasifikasi Dokumen,” *JURNAL INFOTEL,* vol. 9, nr 4, pp. 416 - 421, 2017.

[4] Q. D. Truong, H. X. Huynh och C. N. Nguyen, ”An Abstract-Based Approach for Text Classification,” *International Conference on Nature of Computation and Communication,* pp. 237-245, 2016.

[5] S. N. Asiyah och K. Fithriasari, ”Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan K- Nearest Neighbor,” *JURNAL SAINS DAN SENI ITS,* vol. 5, nr 2, pp. 317-322, 2016.

[6] B. Aljaddouh och N. A. Kotak, ”Document Text Classification Using Support Vector Machine,” *IJEDR,* vol. 8, nr 1, pp. 138-142, 2020.

[7] G. Y. N. Adi, M. H. Tandio, V. Ong och D. Suhartono, ”Optimization for Automatic Personality Recognition on Twitter in Bahasa Indonesia,” *3rd International Conference on Computer Science and Computational Intelligence,* p. 473–480, 2018.

[8] Mansoori, T. Khan, A. Suman, Mishra och D. S. K., ”Feature Selection by Genetic Algorithm and SVM Classification for Cancer Detection,” *International Journal of Advanced Research in Computer Science and Software Engineering,* pp. 357-365, 2014.

[9] I. Syarif, A. Prugel-Bennett och G. Wills, ”SVM Parameter Optimization Using Grid Search and Genetic Algorithm to Improve Classification Performance,” *TELKOMNIKA,* vol. 14, nr 4, pp. 1502-1509, 2016.

[10] M. Ramadhan och R. E. Putra, ”Prediksi Kepribadian Pengguna Instagram Berdasarkan Model Big Five Personality Menggunakan Algoritma SVM,” *Journal of Informatics and Computer Science,* vol. 1, nr 4, pp. 179-187, 2020.

[11] R. C. W. D. E. R. Arif Pratama, ”Implementasi Algoritme Support Vector Machine (SVM) untuk Prediksi Ketepatan Waktu Kelulusan Mahasiswa,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 2, nr 4, pp. 1704-1708, 2018.

[12] Hefinioanrhys, ”Support Vector Machines with the MLR Package,” R-Bloggers, 10 Oktober 2019. [Online]. Available: https://www.r-bloggers.com/support-vector-machines-with-the-mlr-package/.