

## ANT COLONY OPTIMIZATION UNTUK MENYELEKSI FITUR DAN KLASIFIKASI ARTIKEL

**Arief Kelik Nugroho**

Jurusan Teknik Informatika, Fakultas Teknik  
Universitas Jenderal Soedirman, Purwokerto  
Email: arief.nugroho@unsoed.ac.id

**Ipung Permadi**

Jurusan Teknik Informatika, Fakultas Teknik  
Universitas Jenderal Soedirman, Purwokerto  
Email: ipung.permadi@unsoed.ac.id

### ABSTRAK

Algoritma koloni semut dapat diterapkan dalam pengklasifikasian dengan menerapkan mekanisme perilaku semut dalam mencari sumber makanan. Semut memberikan kemungkinan hasil terbaik/optimal berdasar intensitas pheromone. Himpunan fitur pada artikel diseleksi berdasarkan topik atau jenis kelas yang diinputkan dalam sistem kemudian dievaluasi dengan mengecek kesesuaian masing-masing semut. Berdasarkan 10 percobaan yang telah dilakukan, percobaan dengan hasil terbaik didapatkan pada percobaan pertama dan terakhir yang menyeleksi/memilih 66 fitur yang artinya berhasil mengurangi fitur sebanyak 96,34% dengan akurasi pelatihan 52%.

**Kata kunci:** koloni; semut; seleksi; fitur.

### ABSTRACT

*The ant colony algorithm can be applied in classifying by applying the mechanism of ant behavior in finding food sources. Ants provide the best/optimal possibility based on the intensity of the pheromone. The set of features in article is selected based on the topic or type of class inputted in the system and then evaluated by checking the suitability of each ant. With 10 experiments that have been conducted, experiments with the best result were obtained in the first and last trials that selection/selected 66 features which mean succeeding reduce 96,34% of features and 52 % training accuracy.*

**Keywords:** colony; ant; selection; feature.

## 1. PENDAHULUAN

Klasifikasi artikel merupakan suatu kegiatan yang biasa dilakukan untuk memberi kategori terhadap suatu artikel pada suatu situs berita. Klasifikasi dapat dilakukan secara otomatis menggunakan algoritma-algoritma klasifikasi dengan mengandalkan fitur-fitur tiap artikel sebagai input (masukan) dari klasifikasi. Fitur-fitur artikel ini mencakup tiap kata unik yang terdapat pada suatu artikel. Penggunaan kata sebagai masukan dalam klasifikasi tentunya akan membuat proses klasifikasi menjadi lambat ketika terdapat banyak kata yang digunakan. Suatu artikel dapat mengandung ratusan kata di dalamnya. Kata-kata yang digunakan tidak semua memiliki pengaruh yang besar terhadap proses klasifikasi. Malah terkadang dapat mengacaukan proses klasifikasi. Maka dari itu perlu dilakukan seleksi fitur untuk memilih fitur-fitur yang penting yang memiliki pengaruh dalam proses klasifikasi. Penelitian ini bertujuan untuk menyelesaikan permasalahan tersebut yaitu menyeleksi fitur dari artikel dan melakukan klasifikasi berdasarkan fitur-fitur yang terpilih/terseleksi.

## 2. METODOLOGI PENELITIAN

Penelitian ini menggunakan algoritma *Ant Colony Optimization* (ACO) untuk proses pemilihan fitur/kata yang akan digunakan dalam proses klasifikasi dan *K-Nearest Neighbourhood Classifier* untuk klasifikasi.

### 2.1 *Ant Colony Optimization* (Aco)

Berikut langkah-langkah dari algoritma ACO dengan (Marco dan Stutzle, 2004):

- a) Merepresentasikan *feature space* awal dalam bentuk graf. Setiap fitur yang ada dijadikan *vertex* yang masing-masing memiliki nilai *pheromone* dan nilai *heuristic*.
- b) Inisialisasi awal sistem ACO  
Tahap ini dilakukan untuk melakukan inisialisasi nilai *pheromone* dan *heuristic* tiap fitur serta inisialisasi jumlah *ant* dan jumlah iterasi/generasi. Nilai penguapan *pheromone* juga diinisialisasi pada tahap ini.
- c) Pembuatan *feature subset* dan evaluasi *ant*.  
Pada tahap ini setiap *ant* akan membuat *feature subset*-nya. Pada awalnya tiap *ant* diletakkan pada sebuah fitur secara *random*. Selanjutnya tiap *ant* memilih *vertex* baru berdasarkan peluang yang didapat dengan (1).

$$P_i^k(t) = \begin{cases} \frac{|\tau_i(t)|^\alpha \cdot |\eta_i|^\beta}{\sum_{u \in J^k} |\tau_u(t)|^\alpha \cdot |\eta_u|^\beta} & \text{jika } i \in J^k \\ 0 & \text{jika tidak} \end{cases} \quad (1)$$

Dimana  $i$  merupakan *vertex* tujuan,  $k$  merupakan nomor semut,  $J^k$  merupakan himpunan *vertex* yang masih dapat dikunjungi semut  $k$ .  $\tau$  merupakan nilai *pheromone*,  $\eta$  merupakan nilai *heuristic*.  $\alpha$  dan  $\beta$  merupakan parameter pembobotan. Peluang tiap fitur kemudian digunakan dalam pemilihan fitur selanjutnya menggunakan algoritma *Roulette Wheel*. Apabila *vertex* yang dipilih merupakan *vertex* awal *ant* yang dipilih secara *random* sebelumnya, maka *feature subset* dari *ant* tersebut didapatkan. Hasil *feature subset ant k* adalah himpunan  $J^k$ .

Setelah *feature subset* tiap *ant* didapatkan, *feature subset* tersebut dievaluasi menggunakan algoritma *knn* dengan cara mengklasifikasikan seluruh data pelatihan satu persatu. Nilai evaluasi tiap *ant* dihitung dengan menghitung jumlah klasifikasi yang benar dibagi jumlah seluruh data pelatihan.

Setelah semua *ant* dievaluasi, akan dipilih *ant* dengan nilai evaluasi terbaik sebagai *best ant*. *Best ant* pada iterasi pertama adalah *ant* dengan nilai evaluasi terbaik pada iterasi tersebut. Sementara *best ant* pada iterasi selanjutnya adalah *ant* dengan nilai evaluasi terbaik pada iterasi tersebut dibandingkan dengan *best ant* pada iterasi sebelumnya.

- d) Mengupdate *pheromone*.  
Pada tahap ini *pheromone* diperbarui dengan (Marco dan Stutzle, 2004) :

$$\tau(i) = \rho\tau(i) + \Delta\tau_b(i) + \sum_{k=1}^n \Delta\tau_k(i) \quad (2)$$

Dimana  $\rho$  merupakan nilai penguapan *pheromone*.  $\Delta\tau_b$  merupakan nilai *pheromone* yang ditinggalkan oleh *best ant* dan  $\Delta\tau_k$  merupakan nilai *pheromone* yang ditinggalkan oleh *ant k*. Nilai *pheromone* yang ditinggalkan suatu semut diperoleh dengan :

$$\Delta\tau_k(i) = \begin{cases} L_k & \text{jika vertex } i \text{ dilewati ant } k \\ 0 & \text{jika tidak} \end{cases} \quad (3)$$

Dimana  $L_k$  merupakan nilai evaluasi *ant k*.

- e) Generate populasi *ant* baru.  
Tahap selanjutnya adalah menghapus semua *ant* (kecuali *best ant*) dan membuat generasi *ant* baru apabila iterasi belum mencapai iterasi maksimal.

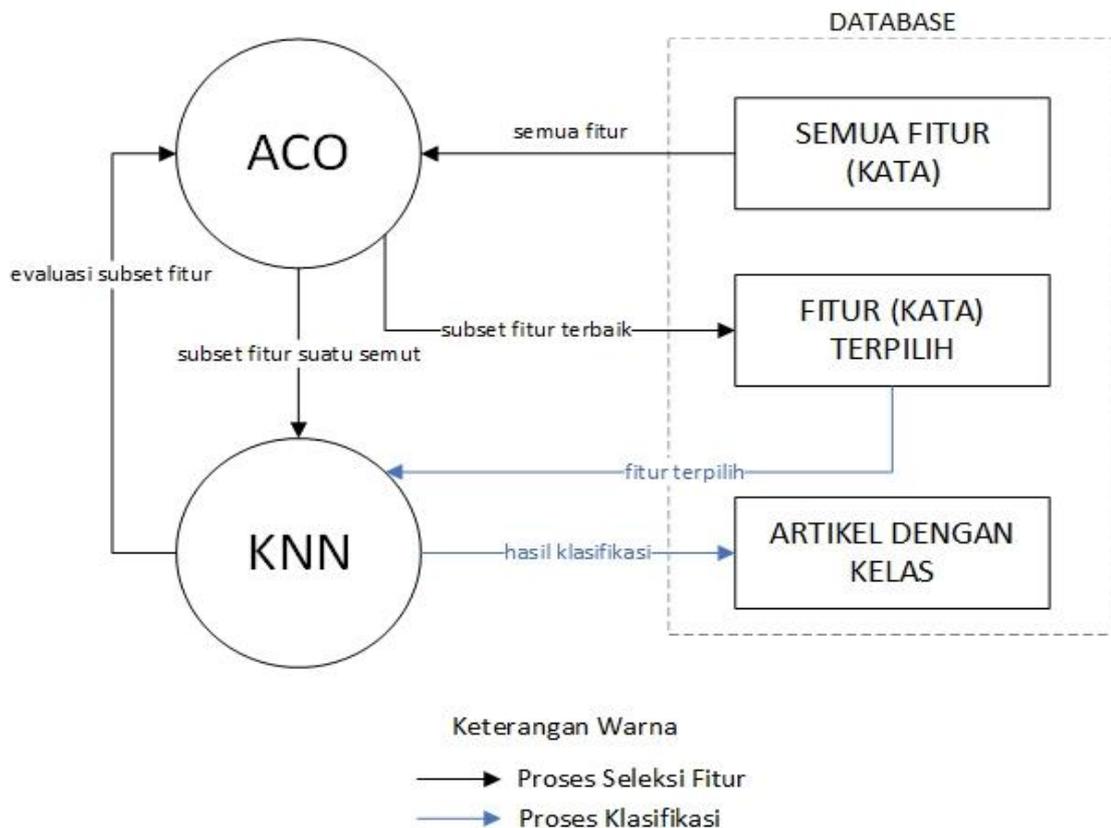
## 2.2 K-Nearest Neighbourhood Classifier

*K-Nearest Neighbourhood Classifier* merupakan satu dari algoritma yang dapat diterapkan dalam proses klasifikasi. Dasar dari penerapan metode ini adalah mengklasifikasikan suatu obyek ke dalam suatu kelas sesuai dengan kelas dari obyek yang berdekatan dengannya. Nilai  $k$  menentukan jumlah obyek tetangga yang dijadikan acuan untuk proses klasifikasi, kemudian hasil dihitung jarak obyek tersebut dengan setiap obyek lain dengan formula *euclidean distance*. Kemudian dipilih  $k$  obyek yang terdekat dengan obyek. Misalnya apabila  $k=3$ , maka kelas dari obyek hijau adalah kelas mayoritas yang dari 3

obyek terdekat. Pada Gambar1 kelas mayoritas adalah segitiga merah, maka obyek hijau masuk dalam kelas segitiga merah.

Pada penelitian ini, peneliti menggunakan nilai  $k=1$  karena terdapat lebih dari 2 kelas pada kasus yang ada pada penelitian ini. Ketika ada lebih dari 2 kelas, maka  $k$  yang banyak dapat mempersulit proses klasifikasi apabila tidak ditemukan kelas mayoritas (jumlah obyek pada suatu kelas sama dengan obyek kelas lain).

Berikut langkah-langkah proses seleksi fitur dengan algoritma *Ant Colony Optimization*. Proses yang seleksi fitur dan klasifikasi pada sistem secara garis besar dapat dilihat pada skema pada gambar 1.



**Gambar 1. Skema Proses dalam Sistem.**

Proses *dalam* sistem dibagi menjadi 2 proses utama, yaitu proses seleksi fitur (Gambar 1) menggunakan panah warna hitam) dan proses klasifikasi artikel (menggunakan panah warna biru).

### 3. HASIL DAN PEMBAHASAN

Proses seleksi fitur dengan algoritma *Ant Colony Optimization*.

#### 3.1 *Merepresentasikan Feature Space*

Dari seluruh artikel, terdapat kata-kata berbeda yang beberapa sesuai pada tabel 1.

**Tabel 1. Contoh daftar fitur**

<i>id_fitur</i>	<i>Fitur</i>
1	sudah
2	dan
3	android
4	partai
...	...

Dari berbagai fitur tersebut, dibuat matriks pheromone dan heuristic dengan cara sebagai berikut. Untuk matriks *pheromone* semua fitur diinisialisasi dengan nilai 1.

**Tabel 2. Nilai pheromone awal**

<i>id_fitur</i>	<i>nilai pheromone (<math>\tau</math>)</i>
1	1
2	1
3	1
4	1
...	...

Sementara matriks *heuristic* diisi dengan menuliskan hasil bagi dari 1 dengan berapa artikel yang memiliki kata tersebut, misalnya sebagai berikut.

**Tabel 3. Nilai heuristic awal**

<i>id_fitur</i>	<i>nilai heuristic (<math>\eta</math>)</i>
1	1/3
2	1/2
3	1/1
4	1/2
...	...

Tabel 3 menunjukkan bahwa fitur dengan id=1 terdapat pada 3 artikel berbeda. Maka nilai heuristicnya 1/3.

### 3.2 Inisialisasi Parameter

Tahap selanjutnya adalah inisialisasi parameter, berikut parameter yang perlu diinisialisasi dan nilainya.

**Tabel 4. Nilai parameter**

<i>Parameter</i>	<i>Nilai</i>
$\rho$ (penguapan)	0.07
n (jumlah semut)	10
m (maksimal iterasi)	20
$\alpha$ (pembobotan pheromone)	1
$\beta$ (pembobotan heuristic)	1

### 3.3 Pembuatan Feature Subset

Untuk setiap semut di setiap iterasi akan dicari subset fiturnya yang merupakan hasil akhir seleksi. Berikut contoh langkah-langkahnya.

- Bangkitkan bilangan random antara 1-jumlah\_fitur sebagai titik akhir semut. Misal didapatkan angka 3 maka fitur dengan id 3 akan menjadi fitur terakhir yang dipilih.
- Hitung probabilitas tiap fitur dengan formula (1) dan hitung nilai kumulatifnya. Misal didapatkan nilai sebagai berikut :

**Tabel 5. Probabilitas seleksi fitur**

<i>id_fitur</i>	<i>probabilitas</i>	<i>nilai kumulatif</i>
1	0.0002156	0.0002156
2	0.0002465	0.0004621
3	0.0003852	0.0008473
4	0.0002465	0.0010938
...	...	...

Kemudian pilih fitur untuk dimasukkan ke dalam subset dengan algoritma *Roulette Wheel* yaitu dengan membangkitkan bilangan random. Jika bilangan tersebut 0.0004243 , maka pilih fitur dengan nilai

kumulatif terbesar pertama dari bilangan random tersebut, pada kasus ini fitur 2. Masukkan 2 ke dalam subset fitur sehingga subset fitur berisi sebagai berikut : subset\_fitur = [2]  
Ulangi langkah b dan c sampai terpilih fitur terakhir yaitu 3 (berdasarkan langkah a). Misalnya didapatkan subset fitur akhir sebagai berikut:

subset\_fitur = [2, 12, 547, 65, 3]

subset fitur dari semut tersebut menunjukkan bahwa fitur dengan id 2, 12, 547, 65 dan 3 terpilih ditunjukkan pada tabel 6 di bawah ini. Misalnya fitur yang terpilih itu diantaranya sebagai berikut :

**Tabel 6. Subset fitur**

<i>id_fitur</i>	<i>fitur</i>
2	dan
12	lalu
547	ini
65	televisi
3	android

Tabel 7 menunjukkan hasil inisialisasi nilai evaluasi\_benar dengan 0 lalu hitung kemunculan tiap fitur pada subset fitur pada tiap artikel. Misalnya pada artikel 1

**Tabel 7. Kemunculan fitur pada artikel 1**

<i>id_fitur</i>	<i>jumlah kemunculan</i>
2	5
12	2
547	4
65	1
3	4

Tabel 8 merupakan jumlah kemunculan fitur pada artikel ke 2

**Tabel 8. Kemunculan fitur pada artikel 2**

<i>id_fitur</i>	<i>jumlah kemunculan</i>
2	2
12	5
547	6
65	7
3	0

Hitung jarak artikel pelatihan (artikel 1) terhadap semua artikel pelatihan lainnya dengan *Euclidean Distance*.

$$d(a,b) = \sqrt{(x_{a(1)} - x_{b(1)})^2 + (x_{a(2)} - x_{b(2)})^2 + \dots + (x_{a(n)} - x_{b(n)})^2}$$

Sehingga jarak antara artikel 1 dan 2 dapat dihitung sebagai berikut :

$$d(1,2) = \sqrt{(5 - 2)^2 + (2 - 5)^2 + \dots + (4 - 0)^2} = d(1,2) = 8.60$$

Sehingga didapatkan jarak euclid artikel 1 dan 2 adalah 8.60. Hitung juga jarak artikel 1 dengan artikel-artikel lainnya.

- c) Setelah didapatkan jarak suatu artikel (artikel x) ke seluruh artikel pelatihan lainnya. Lalu tentukan kelas artikel x dengan artikel lainnya yang memiliki jarak euclid terdekat. Apabila kelas artikel x sesuai dengan target kelasnya maka nilai evaluasi\_benar ditambah 1.
- d) Ulangi langkah f dan g untuk seluruh artikel lalu hitung nilai\_akurasi dengan formula sebagai berikut.

$$akurasi = \frac{evaluasi\_benar}{jumlah\_artikel\_pelatihan}$$

Percobaan dari 30 artikel pelatihan, didapatkan 15 artikel yang kelasnya sesuai dengan target maka

$$akurasi = \frac{15}{30} = 0.5$$

Nilai tersebut merupakan nilai akurasi dari subset fitur yang didapatkan pada langkah d.

- e) Ulangi langkah a sampai h untuk semut-semut selanjutnya sehingga ditemukan subset fitur dan nilai akurasi untuk setiap semutnya.
- f) Pilih semut terbaik dengan memilih semut dengan nilai akurasi terbesar. Apabila nilai akurasinya sama, pilih semut dengan panjang subset fitur terpendek (elemennya sedikit).

### 3.4 Update Pheromone

Update nilai pheromone tiap fitur dengan formula (2) dan (3), misalnya fitur dengan id2 dilewati semut terbaik dan 3 semut yang masing-masing memiliki nilai akurasi sebagai berikut:

**Tabel 9. Nilai setiap semut**

<i>Semut</i>	<i>Nilai Akurasi</i>
semut terbaik	0.9
semut 1	0.5
semut 2	0.9
semut 3	0.2

Maka nilai pheromone milik fitur dengan id 2 adalah

$$\tau(2) = 0.07 \times 1 + 0.9 + 0.5 + 0.9 + 0.2 = 2.57$$

### 3.5 Generate Populasi

Ulangi langkah 3 dan 4 untuk membuat generasi baru untuk iterasi selanjutnya. Perulangan ini akan selesai apabila jumlah iterasi telah mencapai iterasi maksimum.

### 3.6 Hasil Akhir

Setelah iterasi mencapai maksimum, subset fitur milik semut terbaik merupakan output dari *ant colony optimization*. Misalnya semut terbaik memiliki subset fitur sebagai berikut.

$$\text{subset\_fitur} = [2, 12, 547, 65, 3]$$

Maka kelima fitur pada subset fitur itulah yang nantinya akan digunakan pada proses klasifikasi. Proses klasifikasi artikel dilakukan dengan menjalankan algoritma KNN. Proses ini ditujukan terhadap artikel yang belum memiliki kelas.



**Gambar 2. Halaman Artikel**

Artikel yang belum memiliki kelas ditunjukkan dengan warna latar belakang putih pada gambar 2. Langkah-langkah untuk melakukan klasifikasi artikel adalah sebagai berikut.

- a) Untuk subset fitur yang diperoleh dari seleksi fitur adalah sebagai berikut.

`subset_fitur = [2, 12, 547, 65, 3]`

- b) Hitung kemunculan tiap fitur pada subset fitur pada tiap artikel pelatihan dan artikel yang ingin diklasifikasikan (artikel x). Misalnya pada artikel x

**Tabel 10. Kemunculan fitur pada artikel**

<i>id_fitur</i>	<i>jumlah kemunculan</i>
2	3
12	5
547	1
65	6
3	2

Pada artikel pelatihan 1 diperoleh hasil yang ditunjukkan pada tabel 1.

**Tabel 11. Kemunculan fitur pada pelatihan 1**

<i>id_fitur</i>	<i>jumlah kemunculan</i>
2	1
12	3
547	8
65	5
3	1

- c) Hitung jarak artikel x terhadap semua artikel pelatihan dengan *Euclidean Distance*. Sehingga jarak antara artikel x dan artikel pelatihan 1 dapat dihitung sebagai berikut :

$$d(x, 1) = \sqrt{(3 - 1)^2 + (5 - 3)^2 + \dots + (2 - 1)^2} = 7.68$$

Sehingga didapatkan jarak euclid artikel x dengan artikel pelatihan 1 adalah 7.68. Hitung juga jarak artikel 1 dengan artikel-artikel lainnya.

- d) Setelah didapatkan jarak artikel x ke seluruh artikel pelatihan. Tentukan kelas artikel x sama dengan artikel lainnya yang memiliki jarak euclid terdekat.

Sistem pada penelitian ini memiliki 3 halaman yaitu halaman artikel, halaman fitur dan halaman kelas. Pada gambar 2 yang merupakan tangkapan layar halaman artikel tampak daftar artikel yang tiap artikelnya dapat memiliki warna yang berbeda. Artikel dengan warna latar biru merupakan artikel yang digunakan sebagai data pelatihan (sudah memiliki kelas), sedangkan warna latar putih adalah artikel yang belum memiliki dan akan ditentukan kelasnya. Untuk menambahkan artikel, cukup salin url artikel dari [www.detik.com](http://www.detik.com) ke dalam form yang ada di kiri bawah, pilih kelas, lalu klik tambah. Kemudian untuk melakukan klasifikasi atau memberi kelas kepada artikel dengan warna latar putih cukup klik tombol klasifikasi yang terdapat di kanan bawah.



Gambar 3. Halaman Fitur

Pada gambar 3 terdapat daftar fitur/kata yang didapatkan dari seluruh kata unik pada seluruh artikel. Seperti halnya daftar artikel, terdapat perbedaan warna latar juga pada beberapa kata. Kata dengan warna latar hijau adalah kata yang terpilih/terseleksi ACO. Sementara kata dengan warna latar putih adalah kata yang tidak terpilih. Di atas tabel daftar fitur terdapat keterangan jumlah kata yang terpilih dan jumlah keseluruhan kata pada basis data. Proses seleksi fitur dapat dilakukan dengan menekan tombol seleksi fitur yang terletak di bagian kiri bawah.



Gambar 4. Halaman Kelas

Gambar 4 menampilkan halaman kelas yang di dalamnya terdapat daftar kelas. Kelas dapat ditambahkan dengan menuliskan nama kelas pada form di bagian kiri bawah kemudian menekan tombol simpan. Setelah dilakukan percobaan (menekan tombol seleksi fitur 10 kali) dengan tiap percobaan menggunakan 10 semut dan iterasi maksimal 20 menghasilkan hasil seleksi seperti pada tabel 12.

**Tabel 12. Hasil seluruh percobaan**

No	Akurasi (%)	Banyak Fitur Terpilih	Pengurangan Fitur (%)
1	52	66	96,34
2	48	81	95,50
3	48	78	95,67
4	52	67	96,28
5	48	81	95,50
6	48	77	95,72
7	48	88	95,11
8	52	67	95,28
9	48	79	95,61
10	52	66	96,34

Kolom akurasi dihitung dengan membagi jumlah klasifikasi yang benar dengan jumlah klasifikasi data pelatihan. Banyak fitur terpilih adalah banyak fitur/kata yang dipilih oleh semut terbaik pada percobaan tersebut. Sementara pengurangan fitur dihitung dengan membagikan banyak fitur yang tidak dipilih dengan banyak semua fitur.

Berdasarkan 10 percobaan yang telah dilakukan, percobaan dengan hasil terbaik didapatkan pada percobaan pertama dan terakhir yang menyeleksi/memilih 66 fitur yang artinya berhasil mengurangi fitur sebanyak 96,34% dengan akurasi pelatihan 52%.

#### 4. KESIMPULAN

##### 4.1 Simpulan

Berdasarkan hasil dan pembahasan pada poin sebelumnya, peneliti dapat menyimpulkan bahwa sistem yang dibuat dapat digunakan untuk melakukan seleksi fitur untuk klasifikasi artikel detik.com. Hasil terbaik yang didapatkan pada pelatihan dengan 25 data dan 5 kelas yang dilakukan 10 kali menghasilkan akurasi sebesar 52% dengan memilih 66 fitur dari 1801 fitur yang ada. Hasil seleksi fitur tidak stabil karena terdapat algoritma *roulette wheel* dimana pengambilan nilai dilakukan secara random.

##### 4.2 Saran

Hasil pada penelitian ini masih banyak kekurangan terutama dalam penentuan kata, sehingga terdapat kata dengan hasil klasifikasi tidak maksimal, untuk itu perlu diperbanyak dan pilih artikel secara teliti (pilih artikel dengan kata-kata yang relevan) dan mengeliminasi kata-kata yang pasti tidak relevan dalam pengklasifikasian artikel seperti.

#### DAFTAR PUSTAKA

- [1] Sarac C, Ozel S A. 2014. “ An Ant Colony Optimization Based Feature Selection For Web Classification”. *The Scientific World Journal*. Vol.2014
- [2] Aghadam M H, Aghaee N G, Basiri M E .Text feature selection using ant colony optimization. *Elsevier*.2008.
- [3] Bali M R, Gore D.”Text Document Classification using Ant Colony Optimization and Genetic Algorithm”: *International Journal of Innovative Research in Computer and Communication Engineering*; Vol. 3, Issue 12, December,2015.
- [4] Dorigo M, Stützle T. *Ant Colony Optimization*. Cambridge. Massachusetts; The MIT Press.2004.
- [5] Nugroho A K, Iskandar D.2015.“Algoritma Iterative Dichotomizer 3 (ID3) Pengambilan Keputusan”. *Jurnal Dinamika Rekayasa. (DINAREK)*.
- [6] Putra Y A , Faticah C. “Implementasi Metode Ant Colony Optimization untuk Pemilihan Fitur pada Kategorisasi Dokumen Teks”. Surabaya: *ITS Undergraduate*.
- [7] Sarac E, Ozel S A. An Ant Colony Optimization Based Feature Selection for Web Page Classification.;*The Scientific World Journal*;vol. 2014, 2014.
- [8] Suyanto.2010. *Algoritma Optimasi dan Probabilistik*. Yogyakarta: Graha Ilmu
- [9] Satzinger, John, W., et al. 2010. *System Analysis and Design in Changing World, Fifth Edition*.Boston: Course Technology.

- [10] Nurlaelasari E, Supriyadi, Lenggana U.T., 2018. "Penerapan Algoritma Ant Colony Optimization Menentukan Nilai Optimal Dalam Memilih Objek Wisata Berbasis Android". *Jurnal SIMETRIS*. Vol. 9 No. 1.