

# Implementation of Cosine Similarity and Time Interval Entropy Method to Identify Bot Spammer Account on Twitter

Sisca Dewi Priyani  
Teknik Informatika, FST  
Universitas Al Azhar Indonesia  
Jl. Sisingamangaraja, Kebayoran  
Baru, Jakarta Selatan, 12110  
sisca\_priyani@if.uai.ac.id

Endang Ripmiatin  
Teknik Informatika, FST  
Universitas Al Azhar Indonesia  
Jl. Sisingamangaraja, Kebayoran  
Baru, Jakarta Selatan, 12110  
endang\_nizar@uai.ac.id

Solechoel Arifin  
Senior Data Analyst,  
PT.Dua Empat Tujuh  
Jl. Prof.Dr. Satrio Kav. 6, Karet  
Kuningan, Jakarta Selatan, 12940  
solechoel@solusi247.com

## ABSTRACT

*Twitter is one of the social media that has many users. However, the popularity of twitter is also followed by the large number of spam sent by automated programs called bot spammers. Bot spammer produces tweets that have similar content with previous tweets, and it may post in regular interval since it was automatically posted based on a scheduler system. This research uses HGrid247 tool and supervised classification method to classify bot spammer account and legitimate user account based on tweet similarity rates and regularity of time interval between tweets. Cosine similarity method used to observe tweet similarity rates while time interval is used to observe regular behavior in posting a tweet. Based on the result of performance evaluation, the proposed method can produce accuracy by 90%. This proves that the combination of cosine similarity and time interval entropy methods can optimize the identification process of bot spammers in twitter.*

Twitter merupakan salah satu media sosial yang memiliki banyak penggunanya. Namun popularitas twitter juga diikuti oleh banyaknya penyebaran spam yang dilakukan oleh program otomatis yang disebut dengan bot spammer. Bot spammer menghasilkan tweet dengan konten yang mirip dengan tweet-tweet sebelumnya, dan diunggah dengan interval waktu yang teratur karena tweet diunggah berdasarkan sistem penjadwalan. Penelitian ini menggunakan tool HGrid247 serta metode klasifikasi supervised untuk mengklasifikasi akun bot spammer dan akun pengguna sah berdasarkan tingkat kemiripan konten antar tweet serta keteraturan interval waktu antar tweet. Metode cosine similarity digunakan untuk mengamati tingkat kemiripan antar tweet sedangkan time interval entropy digunakan untuk mengamati adanya keteraturan waktu dalam mengunggah suatu tweet. Berdasarkan hasil evaluasi kinerja, metode yang diusulkan menghasilkan akurasi sebesar 90%. Hal ini membuktikan bahwa perpaduan metode cosine similarity dan time interval entropy dapat mengoptimalkan proses identifikasi akun bot spammer pada twitter.

## Keywords

*Bot spammer, classification, cosine similarity, time interval entropy, HGrid247.*

## 1. PENDAHULUAN

Twitter merupakan salah satu media sosial yang memiliki banyak penggunanya. Menurut situs Sysomos [1], pengguna twitter dari

negara asia mencapai 7.74% dari total pengguna twitter di seluruh dunia. Peringkat pertama diduduki oleh Indonesia dengan 2.41%, diikuti oleh India 1.27% dan Jepang 1.22%. Namun, popularitas twitter juga diikuti oleh banyaknya jumlah penyalahgunaan twitter. Salah satunya penyalahgunaan twitter adalah penyebaran spam. Spam merupakan penyalahgunaan pengiriman pesan secara berkali-kali dan tidak dikehendaki oleh penerimanya. Seseorang yang melakukan aktivitas spam disebut spammer. Spammer mengeksploitasi media sosial untuk kepentingan pribadi seperti menyebarkan iklan, mempromosikan diri, hingga penipuan. Jumlah aktivitas spam semakin meningkat dengan adanya bot spammer di twitter.

Bot spammer dapat secara otomatis menghasilkan pesan spam pada waktu tertentu menggunakan sistem penjadwalan [2]. Selain itu, bot spammer juga mengunggah tweet secara terus-menerus dengan konten tweet yang sama. Meningkatnya jumlah bot spammer di twitter membuat pengguna twitter merasa terganggu. Dengan banyaknya pengguna twitter yang merasa terganggu, otomatis kredibilitas twitter menjadi menurun. Twitter sudah menyediakan fitur report as spam yang berfungsi untuk melaporkan sebuah akun yang dianggap sebagai spammer. Namun banyak pengguna twitter yang tidak memanfaatkan fitur tersebut untuk melaporkan akun spammer ke pihak twitter. Hal ini yang membuat pihak twitter sulit mengidentifikasi akun bot spammer pada twitter sehingga akun bot spammer di twitter tidaklah hilang.

Terdapat beberapa penelitian yang telah dilakukan sebelumnya untuk mengidentifikasi akun bot spammer pada twitter, diantaranya penelitian [2] [3] [4]. Untuk mengetahui metode yang paling baik untuk digunakan dalam mengidentifikasi bot spammer pada twitter, penulis melakukan penelitian untuk mengidentifikasi akun bot spammer pada twitter menggunakan metode cosine similarity dan time interval entropy. Metode cosine similarity digunakan karena menurut Strehl, metode cosine similarity merupakan metode yang paling baik digunakan untuk menganalisis tingkat kemiripan antar dokumen. Sedangkan metode time interval entropy digunakan untuk mengamati tingkat keteraturan interval waktu tweet karena berdasarkan penelitian [2] [3], metode ini menghasilkan akurasi yang tinggi. Selain melakukan klasifikasi akun menggunakan penggabungan metode cosine similarity dan time interval entropy, penulis juga melakukan klasifikasi berdasarkan metode unigram matching based similarity dan time interval entropy untuk membandingkan akurasi metode yang diusulkan dengan metode pada penelitian

[2]. Untuk melakukan penelitian ini, penulis menggunakan *tool* HGrid247 karena memiliki banyak fitur pengolahan data.

## 2. PEMBAHASAN STUDI LITERATUR

Penulis melakukan studi literatur terhadap beberapa penelitian yang berkaitan dengan identifikasi akun *bot spammer* pada *twitter*. Perdana dkk. [2] dapat mendeteksi akun *bot spammer* pada *twitter* dengan mengamati konten *tweet* dan interval waktu antar *tweet*. Metode *unigram matching based similarity* digunakan untuk mengetahui tingkat kemiripan antar konten *tweet* dan metode *time interval entropy* digunakan untuk mengamati adanya keteraturan waktu antar *tweet*. Penggabungan metode *unigram matching based similarity* dan *time interval entropy* menghasilkan akurasi sebesar 85,71%. Sedangkan Penelitian Chu dkk. [3] melakukan klasifikasi pengguna *twitter* ke dalam tiga kategori yaitu *human*, *bot* dan *cyborg* (perpaduan *bot* dan manusia). Proses klasifikasi akun pengguna *twitter* dilakukan dengan mengamati komponen entropi, deteksi konten *tweet*, serta rasio jumlah *follower* dan *following*. Algoritma klasifikasi yang digunakan yaitu *decision tree*. Berdasarkan hasil penelitian yang dilakukan, pengklasifikasian pada komponen entropi menghasilkan akurasi tertinggi yaitu sebesar 82,8%. Penelitian Zhang dan Paxson [4] mengidentifikasi adanya aktivitas otomatisasi pada *twitter* menggunakan algoritma Pearson  $\chi^2$ . Algoritma tersebut digunakan untuk mengamati pola persebaran *timestamp tweet* dari suatu akun. Adanya otomatisasi menunjukkan adanya pola-pola tertentu. Sedangkan pengguna sah akan menghasilkan persebaran acak sehingga tidak membentuk pola.

## 3. METODOLOGI PENELITIAN

### 3.1 Pengumpulan Data

Pengumpulan data *tweet* diambil dari 500 akun *twitter* yang terdiri dari 250 akun *bot* dan 250 akun pengguna sah *twitter*. Setiap akun diambil sebanyak 1000 *tweet*. Pencarian akun *bot* dan akun pengguna sah *twitter* dilakukan secara manual oleh penulis. Selanjutnya penulis memperhatikan konten *tweet* dan *timestamp tweet* dari suatu akun untuk mengkategorikan akun tersebut sebagai akun *bot* atau akun pengguna sah *twitter*. Untuk mengambil data *tweet* dari suatu akun *twitter*, penulis menggunakan script Phyton dan REST API *Twitter*. Proses penarikan data *twitter* dilakukan dengan menjalankan script Phyton menggunakan *tool* Spyder. Data *tweet* dihasilkan dalam format csv yang terdiri dari tiga kolom yaitu *id tweet*, konten *tweet* dan *timestamp tweet*.

### 3.2 Data Preprocessing

Data *tweet* yang telah dikumpulkan akan dilanjutkan ke tahap *data preprocessing*. *Data preprocessing* digunakan untuk membantu dalam melakukan pembobotan term serta menghasilkan data yang lebih akurat dalam perhitungan *cosine similarity* karena kata-kata yang tidak dianggap penting akan dibuang pada tahapan ini. Pada tahapan *data preprocessing*, penulis menggunakan fitur yang terdapat pada *tool* HGrid247.

#### 3.2.1 Data Cleansing

*Data cleansing* bertujuan untuk membersihkan data *tweet* dengan menghilangkan URL, @ (*mention*), RT (*retweet*), # (*hashtag*), *emoticon* serta tanda baca dan angka. Proses *data cleansing* dilakukan dengan menggunakan fitur *ReplaceTextMatchPattern* pada *tool* HGrid247. Fitur *ReplaceTextMatchPattern* merupakan bagian dari fitur *transformator* yang berfungsi untuk mengganti bagian input teks dengan teks lain, berdasarkan *regular expression matching pattern*.

#### 3.2.2 Case Folding

Tahapan ini bertujuan untuk mengubah semua huruf menjadi huruf kecil. Untuk mengubah semua huruf menjadi huruf kecil digunakan fitur *lower* yang ada pada fitur *transformator*.

#### 3.2.3 Filtering

*Filtering* merupakan tahap mengambil kata-kata penting dan membuang kata-kata yang tidak penting. Proses *filtering* dapat menggunakan metode *removal stopwords*. *Stopword* yaitu kata yang kurang penting atau tidak bermakna. Contoh kata yang tidak bermakna antara lain ini, itu, yang, akan, bahwa, yaitu, oleh, karena, dan lainnya. Untuk melakukan tahap *filtering*, diperlukan kamus *stopword* yang berisi kata-kata tidak bermakna. *Tweet* dari pengguna *twitter* akan dibandingkan dengan kamus *stopword*, jika pada *tweet* tersebut mengandung kata yang ada pada kamus *stopword* maka kata tersebut akan dihilangkan dari *tweet* pengguna *twitter*. Proses *removal stopwords* ini menggunakan fitur *ReferenceReplacement* untuk mengganti konten teks berdasarkan *file reference*.

### 3.3 Pembobotan Term

Tahap pembobotan term dilakukan menggunakan metode *Term frequency-Invers Document frequency* (TF-IDF). Fungsi dari TF-IDF yaitu merepresentasikan *tweet* ke dalam sebuah vektor. Vektor tersebut beranggotakan term yang dihitung berdasarkan *Term frequency-Invers Document frequency* (TF-IDF). Metode TF-IDF merupakan perpaduan antara metode *Term Frequency* dan *Invers Document Frequency*. *Term frequency* adalah frekuensi dari kemunculan sebuah term dalam *tweet* yang bersangkutan. Nilai jumlah kemunculan suatu kata (*term frequency*) diperhitungkan dalam pemberian bobot terhadap suatu kata.

$$tf(t) = \text{jumlah term } t \text{ pada suatu tweet} \quad (1)$$

*Invers document frequency* menunjukkan hubungan sebuah term dalam sebuah *tweet*. Semakin sedikit jumlah *tweet* yang mengandung term yang dimaksud, maka nilai *idf* semakin besar. Nilai *idf* sebuah term dirumuskan dalam persamaan berikut:

$$idf(t) = \log\left(\frac{\text{jumlah tweet}}{\text{jumlah tweet yang mengandung term } t}\right) \quad (2)$$

Bobot dari term dihitung menggunakan ukuran *tf-idf* dalam persamaan berikut:

$$w(t) = tf(t) \times idf(t) \quad (3)$$

Notasi *tf* merupakan frekuensi kemunculan term dari setiap *tweet*, dan *w(t)* merupakan bobot *tweet* terhadap kata di dalamnya. Pada penelitian ini, penggunaan metode TF-IDF akan digunakan untuk membobotkan term atau kata yang ada pada setiap *tweet* sebelum dihitung kemiripan *tweet*-nya menggunakan metode *cosine similarity*.

### 3.4 Cosine Similarity

Untuk mengetahui adanya kemiripan konten antar *tweet* dari suatu *user*, dibutuhkan perhitungan untuk mengukur tingkat kemiripan antar *tweet*. *Cosine similarity* merupakan metode yang sering digunakan untuk mengukur tingkat kemiripan antar dua objek. Metode *cosine similarity* memanfaatkan pembobotan TF-IDF untuk membentuk vektor. *Cosine similarity* merupakan *similarity measurement* atau pengukuran kemiripan dengan menghitung jarak antara vektor A dan B yang menghasilkan sudut *cosine x*. Nilai sudut kosinus antara dua vektor menentukan kesamaan dua buah objek yang dibandingkan dengan nilai terkecil adalah 0 dan

nilai terbesar yaitu 1. Metode *cosine similarity* digunakan pada penelitian ini karena menurut Strehl, metode *cosine similarity* merupakan metode yang paling baik untuk mencari kemiripan antar dokumen. Persamaan perhitungan *cosine similarity* sebagai berikut [5]:

$$\text{similarity} = \cos \theta = \frac{A \cdot B}{|A||B|} \quad (4)$$

Variabel A dan B menunjukkan *tweet* A dan *tweet* B dari pengguna *twitter*. Nilai  $A \cdot B$  merupakan perkalian skalar nilai tf-idf dari *tweet* A dan *tweet* B. Sedangkan  $|A||B|$  merupakan panjang vektor dari *tweet* A dan *tweet* B. Perkalian skalar dua buah vektor dan panjang suatu vektor dapat dihitung dengan rumus sebagai berikut:

$$A \cdot B = A_1 B_1 + \dots + A_n B_n \quad (5)$$

$$|A||B| = \sqrt{A_1^2 + \dots + A_n^2} \sqrt{B_1^2 + \dots + B_n^2} \quad (6)$$

Paramater yang digunakan adalah jumlah kata atau term pada *tweet* yang akan dibandingkan. Batasan nilai yang dihasilkan mulai dari 0 sampai dengan 1. Nilai 0 menunjukkan ketidakmiripan konten antar *tweet* sedangkan nilai 1 menunjukkan tingkat kemiripan antar *tweet* yang tinggi.

Untuk mencari nilai *cosine similarity* dapat dilakukan dengan cara membagi hasil perkalian skalar bobot antar *tweet* dengan panjang vektor masing-masing *tweet*. Perkalian skalar didapat dari hasil perkalian bobot term antar *tweet* sedangkan panjang vektor dari setiap *tweet* didapat dari hasil kuadrat bobot setiap term dalam setiap *tweet*, lalu jumlahkan nilai kuadrat dan terakhir akarakan. Rata-rata nilai *cosine* dari sejumlah N *tweet* dapat dihitung dengan menjumlahkan seluruh nilai *cosine* antar *tweet* dibagi dengan  $((N \times (N-1))/2)$ . Akun yang memiliki nilai *cosine similarity* di atas *threshold* akan dikasifikasikan sebagai akun *bot spammer*.

### 3.5 Unigram Matching Based Similarity

*Unigram matching based similarity* merupakan merupakan metode pengukuran kemiripan kalimat dengan membandingkan jumlah kata yang sama terhadap jumlah keseluruhan kata. *Unigram matching based similarity* lebih dikenal dengan sebutan *dice coefficient*. Algoritma ini ditemukan oleh Throvald Sorensendan Lee Raymond Dice [6]. Algoritma ini digunakan pada penelitian sebelumnya untuk menganalisis tingkat kemiripan antar *tweet*. Metode *unigram matching based similarity* tidak menggunakan proses pembobotan term karena pada metode ini yang diperhitungkan adalah kata-kata yang sama pada kalimat-kalimat yang dihitung nilai kemiripannya. *Unigram matching based similarity* dapat dihitung dengan, menggunakan rumus di bawah ini [2]:

$$\text{sim}(t_i, t_j) = \frac{(2 * |t_i \cap t_j|)}{(|t_i| + |t_j|)} \quad (9)$$

Notasi  $|t_i \cap t_j|$  menunjukkan jumlah kata yang sama antar *tweet* ke-i dan *tweet* ke-j.  $|t_i|$  menunjukkan jumlah kata pada *tweet* ke-i dan  $|t_j|$  merupakan jumlah kata pada *tweet* ke-j.

### 3.6 Time Interval Entropy

Dalam ilmu fisika, entropi merupakan besaran termodinamika yang menyatakan derajat ketidakteraturan suatu partikel. Entropi

menggambarkan ukuran keacakan atau turbulensi suatu sistem [7]. Kompleksitas proses acak dapat digambarkan menggunakan nilai entropi, nilai entropi mendekati nol menunjukkan suatu proses dengan pengulangan secara periodik dan nilai entropi tinggi menunjukkan proses dengan perilaku acak [3]. Pada penelitian ini, penggunaan entropi digunakan untuk menganalisis keacakan waktu dalam *mem-posting tweet* dari suatu akun dengan cara menghitung *time interval entropy*. *Time interval entropy* dihitung dengan menggunakan persamaan (7) dan (8) [2]:

$$P_{\Delta T}(\Delta t_i) = \frac{n_{\Delta t_i}}{\sum_{k=1}^{nT} n_{\Delta t_k}} \quad (7)$$

$$H_{\Delta T}(T_i) = \sum_{i=1}^{nT} P_{\Delta T}(\Delta t_i) \log(P_{\Delta T}(\Delta t_i)) \quad (8)$$

Untuk mengetahui nilai *time interval entropy* pada suatu *user*, terlebih dahulu hitung peluang kemunculan suatu interval waktu yang dinotasikan dengan  $P_{\Delta T}(\Delta t_1)$  dengan cara membagi jumlah kemunculan interval waktu t yang dilambangkan  $n\Delta t_i$  dengan banyaknya interval waktu pada suatu *user*. Setelah mendapatkan peluang kemunculan interval waktu  $t_i$ , hitung *time interval entropy* yang dilambangkan dengan  $H_{\Delta T}(T_i)$ .

Metode *time interval entropy* dapat digunakan untuk menangkap pola keteraturan waktu *posting tweet*. Nilai entropi rendah menunjukkan adanya perilaku keteraturan, sedangkan nilai entropi tinggi menunjukkan perilaku acak. Sehingga pengguna *twitter* yang memiliki entropi lebih rendah dari *threshold* akan diklasifikasikan sebagai *bot spammer* karena nilai entropi rendah di bawah *threshold* menunjukkan perilaku yang teratur. [2]

## 4. IMPLEMENTASI DAN PENGUJIAN

### 4.1 Pembuatan Training Set

Untuk melakukan klasifikasi menggunakan metode *supervised, data set* yang telah dikumpulkan dibagi menjadi dua yaitu *training set* dan *testing set*. *Training set* digunakan untuk melatih atau membangun sebuah model, sedangkan *testing set* digunakan untuk menguji keakuratan sebuah model. Aturan standar yang digunakan untuk membagi *data set* yaitu 60% dari data set digunakan sebagai *training set* dan 40% digunakan untuk *testing set*. [8]

Tabel 1. Data Set

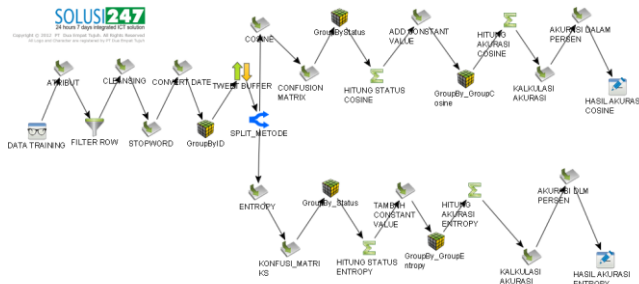
Data Set	Label	Jumlah Akun
Training	Bot spammer	150
	Pengguna sah	150
	Total	300
Testing	Bot spammer	150
	Pengguna sah	100
	Total	200

Data yang dijadikan sebagai *training set* atau data latih sebanyak 300 akun. Data ini telah dilabelkan secara manual ke dalam dua kategori yaitu akun *bot spammer* dan akun pengguna sah *twitter*. Pengkategorian akun dilakukan dengan mengamati konten *tweet* dan *timestamp tweet* dari akun tersebut. Akun yang *mem-posting tweet* dengan konten yang mirip dengan konten *tweet* sebelum-sebelumnya dan *mem-posting tweet* dengan interval waktu yang tidak bervariasi maka akun tersebut dikategorikan sebagai akun *bot spammer*. Sebaliknya jika akun tersebut memiliki konten antar *tweet*-nya bervariasi dan tidak adanya keteraturan waktu dalam

mem-posting suatu *tweet* maka akun tersebut akan diklasifikasikan sebagai akun pengguna sah *twitter*. Proses pelabelan dilakukan dengan mencatat *username* pengguna *twitter* beserta kategorinya yang kemudian disimpan ke dalam file txt.

#### 4.1.1 Proses Training Metode Cosine Similarity dan Time Interval Entropy

Data yang telah dilabelkan akan dijadikan sebagai *data training*. Proses *training* bertujuan untuk melatih model yang nantinya model tersebut akan digunakan pada proses *testing*. Konten *tweet* pada *data training* akan masuk ke tahap *text preprocessing* dan dibobatkan menggunakan metode TF-IDF. *Tweet* yang telah dibobatkan akan dihitung nilai *cosine similarity*-nya, jika akun pengguna *twitter* memiliki rata-rata *cosine similarity* diatas *threshold* maka akun tersebut akan diklasifikasikan sebagai akun *bot spammer* namun jika tidak akan diklasifikasikan sebagai akun pengguna *twitter*. Setelah diklasifikasikan, hitung akurasi dengan menghitung jumlah akun yang diklasifikasikan sesuai dengan label kelas datanya dibagi dengan jumlah total akun pada *data training*. Sedangkan untuk *timestamp*-nya akan dihitung nilai *time interval entropy*-nya, akun yang memiliki nilai *time interval entropy* lebih kecil dari *threshold* maka akun tersebut akan diklasifikasikan sebagai akun *bot spammer*, jika tidak maka akan diklasifikasikan sebagai akun pengguna sah dan kemudian hitung akurasinya. Berikut merupakan *workflow* proses *training* pada tool HGrid247:



**Gambar 1. Workflow Training Metode Cosine dan Time Interval Entropy pada Tool HGrid247**

Proses utama pada proses *training* yaitu mencari nilai *threshold* yang menghasilkan akurasi tertinggi pada masing-masing metode. Nilai *threshold* dengan akurasi terbaik akan dijadikan *threshold* pada proses klasifikasi pada *data testing*. Penentuan nilai *threshold* dilakukan setelah melakukan beberapa kali percobaan. Berikut tabel percobaan penentuan nilai *threshold* untuk metode *cosine similarity*:

**Tabel 2. Penentuan Threshold Metode Cosine Similarity**

Percobaan ke-	Threshold	Akurasi
1	0.030	87.67%
2	0.033	88.34%
3	0.036	89.67%
4	0.039	90.34%
5	0.040	89.67%
6	0.045	89.67%
7	0.050	89.00%
8	0.055	89.00%
9	0.060	88.00%
10	0.065	88.00%

Dari tabel 9 terlihat bahwa nilai *threshold* dengan akurasi terbaik didapatkan pada nilai 0.039. Nilai *threshold* ini menunjukkan akurasi sebesar 90.34%. Nilai *threshold* 0.039 akan digunakan sebagai *threshold* metode *cosine similarity* pada proses klasifikasi pada proses *testing*.

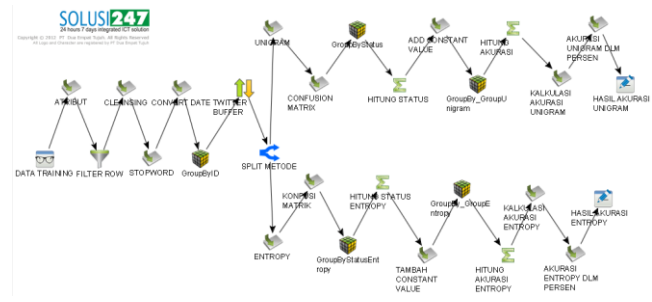
Setelah mendapatkan nilai *threshold* terbaik untuk metode *cosine similarity*, dilakukan pencarian nilai *threshold* terbaik untuk *time interval entropy*. Pencarian *threshold time interval entropy* juga melalui beberapa kali percobaan untuk mendapatkan nilai *threshold* dengan akurasi terbaik. Berikut tabel percobaan penentuan nilai *threshold* untuk metode *time interval entropy*:

Percobaan ke-	Threshold	Akurasi
1	1.00	81.67%
2	1.10	84.34%
3	1.20	84.67%
4	1.25	84.34%
5	1.28	84.67%
6	1.30	85.33%
7	1.33	84.00%
8	1.35	83.34%
9	1.40	82.67%
10	1.45	82.67%

Berdasarkan tabel diatas, nilai *threshold* 1.30 merupakan *threshold* dengan akurasi terbaik yaitu sebesar 85.33%. Nilai *threshold* tersebut akan digunakan sebagai *threshold* metode *time interval entropy* dalam proses klasifikasi pada proses *testing*.

#### 4.1.2 Proses Training Metode Unigram Matching Based Similarity dan Time Interval Entropy

Proses *training* untuk metode *unigram matching based similarity* dan *time interval entropy* mirip dengan proses *training* metode *cosine similarity* dan *time interval entropy*, hanya saja *tweet* pada proses *training* metode *unigram matching based similarity* dan *time interval entropy* tidak perlu dibobatkan menggunakan metode TF-IDF. Konten *tweet* yang telah memasuki tahapan *text preprocessing* akan dihitung nilai *unigram similarity*-nya. Akun yang memiliki nilai *unigram similarity* diatas *threshold* akan diklasifikasikan sebagai akun *bot spammer*, sedangkan akun yang memiliki nilai *unigram similarity* akan diklasifikasikan sebagai akun pengguna sah *twitter*. Berikut merupakan *workflow* proses *training* metode *unigram matching based similarity* dan *time interval entropy* pada tool HGrid247:



**Gambar 2. Workflow Training Metode Unigram Matching Based Similarity dan Time Interval Entropy**

Penentuan *threshold* metode *unigram matching based similarity* juga dilakukan dengan melakukan beberapa kali percobaan hingga mendapatkan *threshold* dengan akurasi tertinggi. Sedangkan

untuk *threshold* metode *time interval entropy*, *threshold* yang digunakan sama dengan *threshold time interval entropy* yang didapat pada proses *training* metode *cosine similarity* dan *time interval entropy* yaitu 1.3. Berikut merupakan tabel percobaan untuk menentukan *threshold unigram matching based similarity*:

**Tabel 3. Penentuan *Threshold* Metode *Unigram Matching Based Similarity***

Percobaan ke-	<i>Threshold</i>	Akurasi
1	0.10	81.33%
2	0.17	94.34%
3	0.18	94.67%
4	0.20	95.34%
5	0.25	94.34%
6	0.28	93.34%
7	0.30	92.30%
8	0.32	91.00%
9	0.40	81.66%
10	0.50	74.00%

Berdasarkan Tabel 3, *threshold* terbaik untuk metode *unigram matching based similarity* yaitu 0.20 karena menghasilkan akurasi tertinggi yaitu sebesar 95.34%.

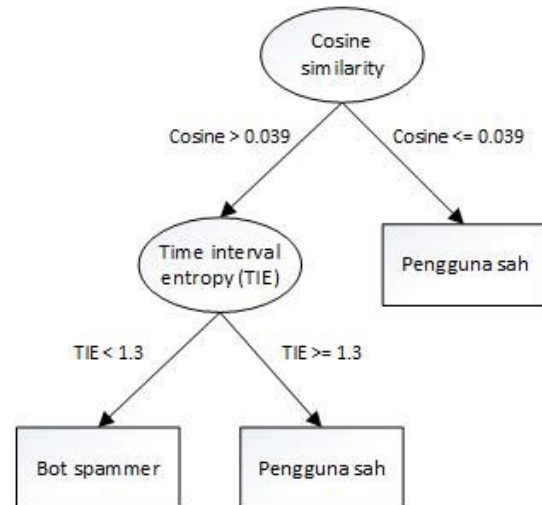
#### 4.2 Pembuatan *Testing Set*

Proses *testing* dilakukan setelah mendapatkan *threshold* terbaik untuk masing-masing metode pada proses *training*. Proses *testing* merupakan proses yang bertujuan untuk mengetahui keakuratan dari model yang dibuat pada proses *training*. Output yang dihasilkan pada proses *testing* yaitu nilai akurasi dari model yang dibuat pada proses *training* yang diuji menggunakan data baru yang disebut *data testing*. *Data testing* yang digunakan yaitu sebanyak 200 akun.

##### 4.2.1 Proses *Testing* Metode *Cosine Similarity* dan *Time Interval Entropy*

*Data testing* yang digunakan meliputi konten *tweet* dan *timestamp tweet* dari suatu akun pengguna *twitter*. Konten *tweet* kemudian akan memasuki tahapan *text preprocessing* untuk menghilangkan noise dan kata-kata yang dianggap tidak penting. Setelah memasuki tahapan *text preprocessing*, term akan dibobotkan menggunakan TF-IDF. Term *tweet* yang telah dibobotkan kemudian akan dihitung nilai *cosine similarity*-nya. Sedangkan untuk *timestamp*-nya akan dihitung *time interval entropy*-nya. Akun yang memiliki nilai *cosine similarity* di atas *threshold* dan nilai *time interval entropy* di bawah *threshold* maka akan diklasifikasikan sebagai akun *bot spammer*, selain itu akan diklasifikasikan sebagai akun pengguna sah.

Pada tahap ini, proses klasifikasi dilakukan dengan menggabungkan metode *cosine similarity* dan *time interval entropy*. Akun yang memiliki nilai *cosine similarity* di atas 0.039 dan nilai *time interval entropy* di bawah nilai *threshold* yaitu 1.30 akan diklasifikasikan sebagai akun *bot spammer*. Berikut merupakan *decision tree* pada proses klasifikasi metode *cosine similarity* dan *time interval entropy*:

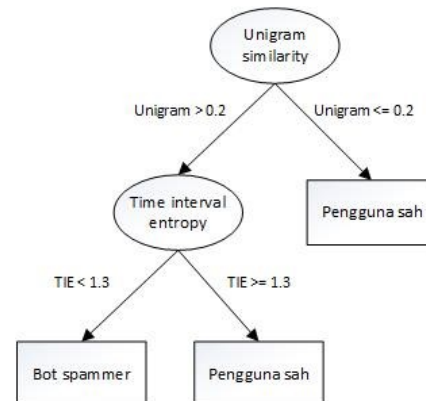


**Gambar 3. *Decision Tree* Klasifikasi Metode *Cosine Similarity* dan *Time interval entropy***

##### 4.2.2 Proses *Testing* Metode *Unigram Matching Based Similarity* dan *Time Interval Entropy*

Untuk membandingkan akurasi metode yang penulis usulkan yaitu metode *cosine similarity* dan *time interval entropy* dengan metode yang digunakan pada penelitian sebelumnya yaitu metode *unigram matching based similarity*, penulis juga melakukan klasifikasi menggunakan metode *unigram matching based similarity* dan *time interval entropy*. *Unigram matching based similarity* merupakan metode pengukuran kemiripan kalimat yang dengan membandingkan jumlah kata yang sama terhadap jumlah keseluruhan kata.

Proses klasifikasi metode *unigram matching based similarity* menggunakan *threshold* 0.20. Nilai *threshold* 0.20 dipilih karena menghasilkan akurasi tertinggi pada proses *training* yaitu sebesar 95.33%. Akun yang memiliki nilai *unigram matching based similarity* di atas 0.20 dan nilai *time interval entropy* dibawah 1.30 akan diklasifikasikan sebagai akun *bot spammer*. Berikut merupakan *decision tree* pada proses klasifikasi metode *unigram matching based similarity* dan *time interval entropy*:



**Gambar 4. *Decision Tree* Klasifikasi Metode *Unigram Matching Based Similarity* dan *Time Interval Entropy***

### 4.3 Evaluasi Hasil

Evaluasi kinerja suatu sistem klasifikasi merupakan hal yang penting. Evaluasi hasil klasifikasi menggambarkan seberapa baik metode yang digunakan dalam mengklasifikasikan data. Pada penelitian ini, penulis menggunakan *confusion matrix* untuk mengevaluasi hasil klasifikasi. *Confusion matrix* digunakan untuk membandingkan data label akun pengguna *twitter* hasil klasifikasi dengan data label yang sebenarnya.

Tabel 4. *Confusion Matrix*

Hasil Observasi (Actual Class)	Hasil Klasifikasi (Classified Class)	
	Pengguna sah	Bot spammer
Pengguna sah	True Negative	False Negative
Bot spammer	False Positive	True Negative

Dalam penelitian ini, *True Negative* (TN) mengacu pada jumlah akun yang diklasifikasikan benar sebagai akun pengguna sah *twitter*. *False Negative* (FN) merupakan akun pengguna sah yang diklasifikasikan tidak benar sebagai *bot spammer*. *False Positive* (FP) merupakan jumlah akun *bot spammer* yang tidak tepat diklasifikasikan sebagai akun pengguna sah. Sedangkan *True Positive* (TP) adalah jumlah akun *bot spammer* yang diklasifikasikan benar sebagai *bot spammer*.

Dari *confusion matrix* dapat diketahui nilai akurasi dari hasil klasifikasi. Nilai akurasi menggambarkan seberapa akurat metode yang digunakan dapat mengklasifikasikan data secara benar. Semakin tinggi nilai akurasi, semakin baik pula metode yang digunakan. Berikut merupakan rumus perhitungan nilai akurasi:

$$\text{akurasi} = \frac{TN + TP}{TN + FN + TP + FP} \times 100\% \quad (10)$$

Pada proses *testing*, *data testing* yang digunakan sebanyak 200. Akun yang terdiri dari 100 akun *bot spammer* dan 100 akun pengguna sah. Hasil klasifikasi berdasarkan metode *cosine similarity* dan *time interval entropy* dapat dilihat pada tabel di bawah ini.

Tabel 5. Hasil Klasifikasi Metode *Cosine Similarity* dan *Time Interval Entropy*

Hasil Observasi (Actual Class)	Hasil Klasifikasi (Classified Class)	
	Pengguna sah	Bot spammer
Pengguna sah	97	3
Bot spammer	17	83

Berdasarkan tabel di atas, terdapat 97 akun pengguna sah yang diklasifikasikan dengan benar sebagai pengguna sah namun terdapat 3 akun yang diklasifikasikan tidak benar sebagai akun *bot spammer*. Hal ini mungkin terjadi karena akun tersebut memiliki karakteristik seperti *bot* yaitu mem-posting konten *tweet* yang mirip dengan interval waktu yang cenderung teratur. Selain itu, terdapat 83 akun *bot spammer* yang diklasifikasikan benar sebagai akun *bot spammer* dan terdapat 17 akun yang diklasifikasikan oleh sistem secara tidak tepat sebagai akun pengguna sah, hal ini disebabkan karena karakteristik akun *bot spammer* tersebut cukup mirip dengan akun pengguna sah. Sedangkan hasil klasifikasi berdasarkan metode *unigram matching based similarity* dan *time interval entropy* yaitu sebagai berikut:

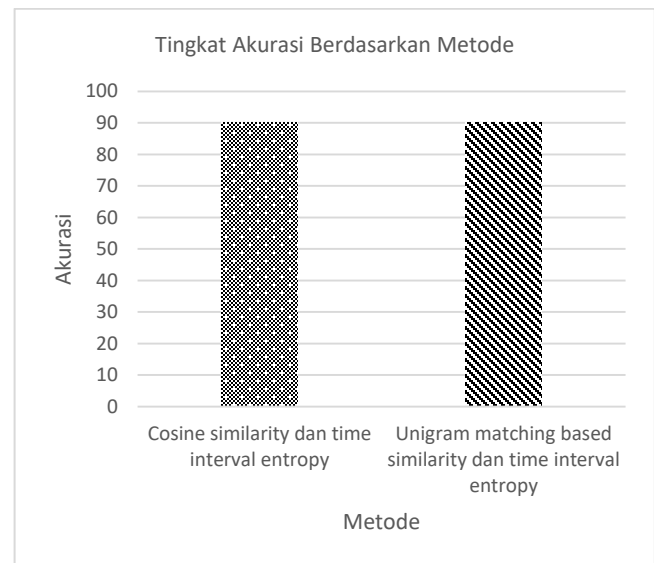
Tabel 6. Hasil Klasifikasi Metode *Unigram Matching Based Similarity* dan *Time Interval Entropy*

Hasil Observasi (Actual Class)	Hasil Klasifikasi (Classified Class)	
	Pengguna sah	Bot spammer

Pengguna sah	100	0
Bot spammer	20	80

Berdasarkan Tabel 6, semua akun pengguna sah diklasifikasikan dengan tepat oleh sistem sebagai akun pengguna sah. Namun terdapat 20 akun *bot spammer* yang diklasifikasikan tidak benar sebagai akun pengguna sah dan terdapat 80 akun *bot spammer* yang benar diklasifikasikan sebagai akun *bot spammer*.

Dari hasil klasifikasi berdasarkan metode yang diusulkan penulis yaitu metode *cosine similarity* dan *time interval entropy* serta hasil klasifikasi berdasarkan metode penelitian sebelumnya yaitu metode *unigram matching based similarity* dan *time interval entropy*, dapat dilihat perbandingan akurasi dari hasil klasifikasi kedua metode dengan grafik di bawah ini:



Gambar 5. Grafik Perbandingan Akurasi Metode *Cosine Similarity* dan *Time Interval Entropy* dengan Metode *Unigram Matching Based Similarity* dan *Time Interval Entropy*

Dari grafik di atas, dapat disimpulkan bahwa hasil klasifikasi metode *cosine similarity* dan *time interval entropy* memiliki tingkat akurasi yang sama dengan metode *unigram matching based similarity* dan *time interval entropy* yaitu sebesar 90%. Hal ini membuktikan bahwa kedua metode memiliki akurasi yang cukup tinggi sehingga kedua metode cocok digunakan untuk mengidentifikasi akun *bot spammer* pada *twitter*.

### 4.4 Evaluasi Metode

Dari percobaan yang telah dilakukan, penulis membuat tabel perbandingan untuk mengevaluasi metode yang diusulkan penulis dengan metode yang digunakan pada penelitian sebelumnya. Perbandingan ini meliputi aspek efisiensi, akurasi serta waktu pemrosesan data. Berikut tabel evaluasi metode:

Tabel 7. Evaluasi Metode

Kriteria	Metode	
	<i>Cosine Similarity</i> dan <i>Time Interval Entropy</i>	<i>Unigram Matching Based Similarity</i> dan <i>Time Interval Entropy</i>
Efisiensi	<i>Tweet</i> dibobotkan terlebih dahulu	<i>Tweet</i> tidak perlu dibobotkan sebelum

	sebelum dihitung menggunakan <i>cosine similarity</i>	dihitung menggunakan <i>unigram matching based similarity</i>
Akurasi	Menghasilkan akurasi sebesar 90%	Menghasilkan akurasi sebesar 90%
Waktu pemrosesan data	Membutuhkan waktu sebanyak 19 menit 23 detik	Membutuhkan waktu sebanyak 18 menit 27 detik

Dari tabel di atas, dapat disimpulkan bahwa penggabungan metode *unigram matching based similarity* dan *time interval entropy* memiliki keunggulan dalam segi efisiensi serta waktu pemrosesan data lebih singkat dibanding menggunakan metode *cosine similarity* dan *time interval entropy*. Namun dari segi akurasi, penggabungan metode *unigram matching based similarity* dan *time interval entropy* serta metode *cosine similarity* dan *time interval entropy* menghasilkan akurasi yang sama yaitu sebesar 90 %.

## 5. KESIMPULAN

Berdasarkan penelitian yang dilakukan penulis, dapat ditarik beberapa kesimpulan antara lain:

- Berdasarkan proses *training* yang telah dilakukan, nilai *threshold* terbaik untuk metode *cosine similarity*, *time interval entropy*, *unigram matching based similarity* secara berturut-turut yaitu 0,039, 1,3, dan 0,2.
- Berdasarkan hasil klasifikasi, metode yang digunakan peneliti sebelumnya yaitu metode *unigram matching based similarity* dan *time interval entropy* dengan metode yang penulis gunakan yaitu metode *cosine similarity* dan *time interval entropy* terbukti bahwa kedua metode tersebut merupakan metode yang baik digunakan untuk mengidentifikasi akun *bot spammer* pada *twitter* karena kedua metode menghasilkan akurasi yang sama dan tinggi yaitu sebesar 90%.
- Dari hasil evaluasi metode, penggabungan metode *unigram matching based similarity* dan *time interval entropy* lebih efisien dan lebih singkat dalam memproses data dibandingkan menggunakan metode *cosine similarity* dan *time interval entropy*.

## REFERENCES

- [1] "Exploring the Use of *Twitter* Around the World," 04 Januari 2010. [Online]. Available: <https://sysomos.com/2010/01/14/exploring-the-use-of-twitter-around-the-world/>.
- [2] R. S. Perdana, T. H. Muliawati and R. Alexandro, "*Bot Spammer Detection in Twitter Using Similarity and Time interval entropy*," *Journal of Computer Science and Information UI*, vol. 8, no. 1, 2015.
- [3] Z. Chu, S. Gianvecchio, H. Wang and S. Jajodia, "Detecting Automation of *Twitter* Accounts: Are You a Human, *Bot*, or *Cyborg?*," *IEEE*, vol. 9, no. 6, pp. 811-824, 2012.
- [4] C. M. Zhang and V. Paxson, "Detecting and Analyzing Automated Activity on *Twitter*," in *PAM'11 Proceedings of the 12th international conference on Passive and active measurement*, Atlanta, 2011.
- [5] A. Firdaus and A. Vatesia, "Aplikasi Pendeteksi Kemiripan pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani dan Metode *Cosine Similarity*," *Jurnal Teknologi Informasi*, vol. 10, 2014.
- [6] E. A. Lisangan, "Implementasi n-gram Technique dalam Deteksi Plagiarisme pada Tugas Mahasiswa," *Jurnal Tematika*, vol. 1, no. 1, 2013.
- [7] C. Soekardi, *Termodinamika Dasar Mesin Konversi Energi*, ANDI, 2015.
- [8] V. Kotu and B. Deshpande, *Predictive Analytics and Data Mining*, Elsevier, 2015.