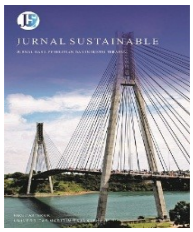


Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan
Vol. 09, No. 01, hal. 31- 37, Mei 2020
Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan
ISSN 2615-6334 (Online)
ISSN 2087-5347 (Print)

Deteksi Twitter Bot Menggunakan Klasifikasi Decision Tree

Hendra Kurniawan*
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji
Jl. Politeknik Senggarang, Tanjungpinang 29111
*Corresponding Author: hendra@umrah.ac.id

Abstract— *Advance development of social media application has affected to human lifestyle. Everyone can obtain information from social media easyly. Its become easier to communicate each other using social media. Twitter is one of the fastest growing social media application. Deliver good and hoax information from one user to another. Event there are alot of fake account (bot) in Twitter. This objection of this study is to detect Twitter Bot accounts on Twitter social media by using the Decission Tree classification. Experiment results show the accuracy performance of the Decision Tree model reached 88.84% and UC curve by 0.965. Its shows that the Decision Tree classification is excellent in detecting Twitter Bot accounts.*
Keywords—*Decision Tree, Classification, Machine Learning, Twitter Bot, Detection System*.

Intisari— Perkembangan media sosial menjadikannya sebagai kebutuhan manusia dalam berkomunikasi pada saat ini. Berbagai informasi dapat diperoleh pengguna dari media sosial. Informasi tersebut ada yang bersifat benar, ada juga yang bersifat salah. Salah satu media sosial yang sangat digemari adalah *Twitter*. Namun tidak jarang, akun yang berada pada aplikasi *Twitter* merupakan akun palsu (*Bot*). Penelitian ini bertujuan untuk melakukan pendeteksian akun *Twitter Bot* pada media sosial *Twitter* dengan menggunakan klasifikasi *Decission Tree*. Hasil pengukuran menunjukkan performa *accuracy* model *Decision Tree* mencapai 88.84% dan perhitungan kurva AUC dengan nilai 0.965. Hal tersebut menunjukkan bahwa klasifikasi *Decision Tree* sangat sesuai dalam pendeteksian akun *Twitter Bot*.
Kata kunci—*Decision Tree, Klasifikasi, Machine Learning, Twitter Bot, Sistem Deteksi*.

I. PENDAHULUAN

Peranan sosial media merupakan bagian yang tidak terpisahkan dalam kehidupan manusaia saat ini. Berbagai *platform* media sosial seperti *Twitter, Facebook, Instagram, Google+* dan *LinkedIn* sudah menjadi konsumsi harian masyarakat. Media sosial digunakan sebagai perangkat komunikasi yang memberikan kenyamanan bagi manusia dalam berinteraksi. Studi yang dilakukan oleh [1] memaparkan bahwa media sosial digunakan sebagai media promosi/iklan, 75.8% dari iklan tersebut dilihat oleh pengguna media sosial dan 59.2% dari pengguna tersebut melakukan pemesanan dari produk yang diiklankan.

Aplikasi *Twitter* menempati urutan ketiga dalam kategori media sosial yang paling banyak digunakan dan jumlah akses pengguna dalam satu bulan [17]. *Facebook* menempati urutan pertama dengan 1,5 miliyar pengguna perbulan, *YouTube* 1,49 miliyar pengguna perbulan dan Twitter 400 juta pengguna perbulan. Penggunaan media sosial yang semakin populer dan dalam skala besar membuat berbagai pihak memanfaatkan hal tersebut, termasuk pengguna yang tidak bertanggung jawab dengan menyebarkan *malware* didalam unggahan konten media sosial. Para pengguna media sosial harus berhati-hati terhadap konten yang diunggah oleh suatu akun. Lebih lanjut, para pengguna harus berhati-hati terhadap keaslian suatu akun.

Beberapa faktor keamanan keamanan termasuk privasi pengguna sangat mungkin dilanggar oleh akun palsu tersebut. Akun palsu yang melakukan unggakan konten *spam* dapat memberikan dampak buruk bagi pengguna media sosial, termasuk pengguna *Twitter*. *Spam* memungkinkan interaksi antara akun palsu dengan pengguna asli media sosial, memiliki peluang dalam memasukkan konten yang bersifat nonobjektif, *hoax*, negatif dan berbahaya bagi penggunaa *Twitter* [2]–[4].

Dengan perkembangan *Twitter* yang merupakan salah satu platform media sosial yang banyak digunakan oleh manusia, dengan layanan *micro blogging* yang dapat menyampaikan pesan singkat secara publik maupun pribadi dari pengguna. Dengan berkembangnya *platform Twitter*, hal tersebut juga dimanfaatkan oleh pihak-pihak yang tidak bertanggung jawab dengan membuat akun pengguna yang palsu. Akun tersebut dapat memberikan informasi palsu atau memberikan link pada laman yang membayakan pengguna. Akun palsu tersebut bekerja dengan mengirimkan permintaan pertemanan, mengirimkan pesan secara personal, dan menyampaikan informasi palsu melalui akunnya dalam proses yang sangat cepat dan secara otomatis [5]. Akun palsu tersebut bertindak mirip seperti pengguna manusia pada umumnya, melakukan komunikasi pada akun pengguna yang menerima permintaan pertemanan yang telah dikirimkan. Bahkan dapat melakukan analisis prilaku pengguna yang telah masuk kedalam jejaring akun palsu tersebut untuk mencari kelemahan dan kekurangan dari akun pengguna asli sebagai target [6].

Pada penelitian ini difokuskan pendeteksian akun palsu (*Twitter Bot*) dengan pendekatan Data *Mining* pada *Dataset* publik menggunakan klasisfikasi *Decision Tree*. Pada bagian kedua akan dibahas penelitian terdahulu yang telah dilakukan oleh peneliti lainnya terkait model klasifikasi *Decision Tree*, bagian ketiga akan dijelaskan metode penelitian, bagian keempat hasil dan pembahasan dari metode yang diusulkan dan bagian terakhir akan ditarik kesimpulan.

II. PENELITIAN TERDAHULU

Pada bagian ini akan dipaparkan penelitan terdahulu yang telah dilakukan terkait pendeteksian *Twitter Bot*. Penelitian yang dilakukan oleh [7] menjelaskan tentang metode deteksi anomali pada *Twitter*. Anomali dikenal sebagai suatu proses yang berbeda dari proses yang ada secara umum. Proses disini dapat diartikan sebagai prilaku akun pengguna *Twitter*, sehingga akun tersebut dapat dideteksi sebagai akun palsu jika memiliki prilaku diluar proses normal yang ada. Dapat mengetahui dan menemukan aksi spam yang belum pernah diketahui sebelumnya merupakan kelebihan dari metode deteksi anomali [8]. Metode deteksi anomali tidak hanya dapat mencari profil perilaku dengan sangat tepat, tetapi juga dapat menggunakan informasi serangan yang diketahui secara tidak langsung di *Twitter*.

Pendekatan lainnya yang telah dilakukan dalam penelitian terdahulu adalah dengan memanfaatkan model Pembelajaran Mesin (*Machine Learning*). Tiga kategori dari pembelajaran mesin menjadi bagian utama dalam pendeteksian akun palsu yaitu : Pembelajaran yang memiliki supervisi (*Supervised Machine Learning*), Pembelajaran dengan semi-supervisi (*Semi-supervised Machine Learning*) dan pembelajaran tanpa supervisi (*Unsupervised Machine Learning*). Ketiganya memiliki karakteristik yang berbeda berdasarkan pendekatan pola himpunan data (*Dataset*) yang digunakan, ditandai dengan label dari setiap data yang ada. [9] telah melakukan penelitian dan publikasi dalam pendeteksian akun palsu dengan model supervised learning menggunakan *Support Vector Machines (SVMs), Decision Tree, Naive Bayes* dan *Neural Networks*. Pendeteksian akun palsu dengan model *semi-supervised learning* telah dilakukan oleh [10]. Model *semi-supervised learning* membutuhkan pelabelan yang jelas antara setiap kelas dalam melakukan pendeteksian. Model *unsupervised learning* telah berhasil diterapkan oleh [11]. Penelitian tersebut menunjukkan bagaimana *clustering*, yang merupakan metode *unsupervised learning*, dapat digunakan untuk mendeteksi akun palsu. Pada model *unsupervised learning* data tidak memiliki label

ISSN 2615-6334 (Online)
ISSN 2087-5347 (Print)

dan data dikelompokkan berdasarkan kesamaan [12]. Hasil *clustering* tersebut berfungsi dengan baik untuk mendeteksi akun palsu karena akun tersebut biasanya memiliki karakteristik yang sama dan memiliki tujuan yang sama.

Penelitian ini akan difokuskan pada model *supervised learning* dengan metode klasifikasi *Decision Tree*. Penggunaan dataset yang sesuai menjadi kunci dalam keberhasilan metode klasifikasi *Decision Tree*. Dataset yang telah memiliki label sebagai data normal dan data palsu telah dipublikasikan oleh [13] dengan memanfaatkan kerangka kerja (*Framework*) dalam pendeteksian akun palsu pada *Twitter* yang memperhatikan skalibilitas dari *streaming* data dan generalisasi dari *dataset* yang berbeda dari sebelumnya.

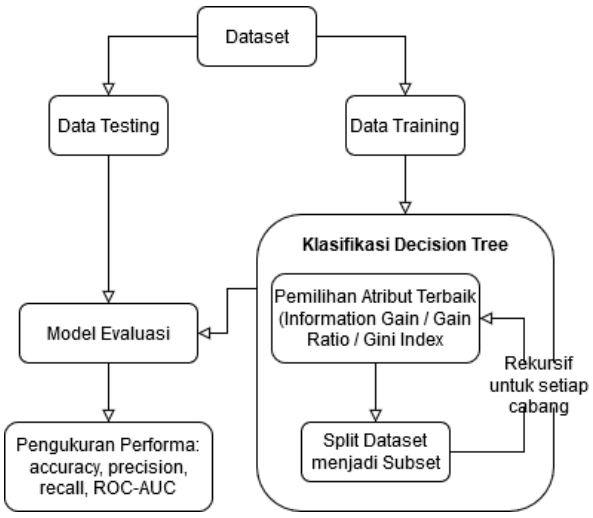
Fokus penelitian tersebut pada pendeteksian akun palsu melalui profil pengguna yang dapat diakses dengan mudah. Dua jenis dataset digunakan pada penelitian tersebut yaitu *verfied-2019* dan *botwiki-2019*. Kedua dataset tersebut sangat sesuai untuk *supervised learning*, karena telah diberikan label secara jelas. Penelitian [14] menghasilkan teknik terbaru dalam pendeteksian akun palsu *Twitter* yang membutuhkan *timestamp* dari fitur *retweet* untuk setiap akun yang dianalisis. Analisis tersebut dibandingkan dengan pola *retweet* dari kelompok pengguna secara umum. Pada penelitian tersebut digunakan dataset *cresci-rtbust-2019* yang melakukan pelabelan data akun palsu dan asli secara manual. Dataset berikutnya *botometer-feedback-2019* yang digunakan oleh [13]. Dataset tersebut hasil *feedback* dari *website* Botometer [*citation*] yang dilakukan pelabelan secara manual oleh penulis.

III. METODE

Bagian ini menjelaskan tetang model klasifikasi *Decision Tree*, *Dataset* yang digunakan dan model evaluasi serta pengukuran performa dari hasil klasifikasi.

A. Klasifikasi Decision Tree

Klasifikasi *Decision Tree* merupakan bagian dari model *supervised machine learning*. Penggunaan *Decision Tree* sangat bergantung pada *dataset* yang telah ditentukan kelas atau label dari setiap data. [15] menjelaskan model klasifikasi *Decision Tree* secara detail. *Decision Tree* memiliki keunggulan dalam hal visualisasi dan mudah dalam implementasi sesuai dengan kasus yang dihadapi seperti pola data non-linear, tidak membutuhkan data *pre-processing* yang terlalu banyak dari pengguna (seperti normalisasi artibut), sangat sesuai untuk seleksi variabel dan memprediksi nilai-nilai yang hilang (*missing values*). Namun juga memiliki kelemahan dalam hal akurasi yang tidak terlalu baik pada *noisy* data (*overfit problem*), *Decision Tree* mengalami bias pada *dataset* yang tidak seimbang, sehingga harus dilakukan penyeimbangan data sebelum menggunakan *Decision Tree*. Algoritma *Decision Tree* bekerja seperti struktur pohon yang memiliki tiga simpul: simpul akar (*root*), simpul cabang (*branch*) dan simpul daun (*leaf*). Simpul *root* merupakan bagian paling atas, simpul *branch* merupakan aturan pengambil keputusan dan simpul *leaf* merupakan hasil dari keputusan.



Gambar 1. Model Klasifikasi Decision Tree

DT dibangun dalam suatu metode rekursif secara *topdown divide-and-conquer*. Pada awalnya semua data *sampel* merupakan *root*, lalu dilakukan pengujian. Proses dilakukan mencabang ke jalur yang tepat berdasarkan hasil pengujian. Akan diperiksa apakah simpul *leaf* ditemukan? jika YA, masukkan data *sample* tersebut kedalam kelas target, jika TIDAK maka kembali kepengujian awal. Atribut data berada dalam suatu kategori (jika data bernilai kontinu, maka nilai tersebut harus dijadikan nilai disktrit terlebih dahulu). Data *sample* dipartisi secara

rekursif berdasarkan atribut terpilih. Atribut yang diuji akan dipilih secara pendekatan heuristik atau pendekatan statistik (Contoh: *Information Gain, Gain Ratio, Gini Index*). Proses rekursif *Decision Tree* akan berhenti bila kondisi terpenuhi, yaitu: semua data *sample* berada dalam kelas yang sama, tidak ada lagi atribut yang akan dilakukan partisi, atau tidak ada data *sample* lagi yang akan diuji. Gambar 1 menjelaskan proses klasifikasi *Decision Tree* dengan menggunakan *dataset* yang terbagi dalam Data *Training* dan Data *Testing*. Data *training* digunakan untuk mencari model yang sesuai dari algoritma *Decision Tree*, sedangkan data *testing* digunakan untuk mengevaluasi model yang telah didapatkan.

Pengukuran dalam seleksi atribut dibutuhkan dalam model klasifikasi *Decision Tree*, seperti yang tertera pada gambar 1, terdapat beberapa metode yang sering digunakan dalam seleksi atribut : *Information Gain, Gain Ratio*, dan *Gini Index*. Tujuan pemilihan atribut untuk memperoleh DT yang memiliki ukuran paling kecil. Semakin bersih (*pure*) simpul *leaf* dari suatu cabang, maka hasilnya akan semakin baik.

1) Information Gain

Pada awalnya akan dilakukan perhitungan *Entropy* yang mengukur ketidakmurnian input dalam suatu himpunan, atau juga dikenal dengan istilah keacakan atau ketidakmurnian sistem dalam bidang fisika dan matematika. *Information Gain* menggunakan metode *entropy* menurun, algoritma yang digunakan adalah ID3 (*Iterative Dichotomiser*).

$$Info(D) = \sum_{i=1}^{m} pi\, log_2(pi) \quad (1)$$

Dimana, pi adalah probabilitas bahwa tuple arbitrer di D pada kelas Ci.

$$Info_A(D) = \sum_{j=1}^{V} \frac{|Dj|}{|D|} \times Info\,(D_j) \quad (2)$$

$$Gain(A) = Info(D) - Info_A(D) \quad (3)$$

Dimana, $Info(D)$ merupakan nilai entropy yang dibutuhkan untuk klasifikasi suatu tuple pada D, dan $Gain(A)$ merupakan Information Gain dari atribut A. Atribut A yang memiliki nilai *Information Gain* terbesar akan dipilih sebagai *split* atribut pada node N.

2) Gain Ratio

Information Gain memiliki bias untuk atribut dengan banyak *outcome*, sehingga hanya sesuai untuk atribut dengan nilai yang berbeda. Dalam beberapa kasus, hal tersebut membuat partisi data *sample* tidak berguna. *Gain Ratio* menerapkan pola yang berbeda dengan algoritma C.45 (suksesor dari algoritma ID3) untuk mengatasi masalah normalisasi pada *Information Gain*.

$$SplitInfo_A(D) = -\sum_{j=1}^{V} \frac{|Dj|}{|D|} \times log_2(\frac{|Dj|}{|D|}) \quad (4)$$

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)} \quad (5)$$

Dimana, $\frac{|Dj|}{|D|}$ merupakan nilai dari partisi ke-j dan v adalah nilai diskrit dari atribut A. Atribut yang memiliki *gain ratio* terbesar akan dijadikan sebagai split atribut.

3) Gini Index

Algoritma lainnya yang digunakan untuk seleksi atribut pada *Decision Tree* adalah CART (*Classification and Regression Tree*) yang menggunakan metode *Gini Index* untuk menentukan *split* poin. *Gini Index* mempertimbangkan biner *split* dari setiap atribut, dengan melakukan penjumlahan *impurity* dari setiap atribut. Sebagai contoh jika suatu *dataset* D dilakukan split menjadi 2 sub-data $D1$ dan $D2$. *Gini index* dari D dapat ditentukan melalui persamaan (7).

$$Gini(D) = 1 - \sum_{i=1}^{m} pi^2 \quad (6)$$

$$Gini_A(D) = \frac{|D1|}{|D|}\, Gini(D1) + \frac{|D2|}{|D|}\, Gini(D2) \quad (7)$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D) \quad (8)$$

Pada kasus atribut dengan nilai diskrit, *subset* yang memiliki nilai gini index minimum akan bertindak sebagai *split* atribut. Pada kasus dengan nilai kontinu, strateginya adalah untuk memilih setiap pasangan nilai yang berdekatan dengan *split* poin dan poin dengan nilai *gini index* terkecil yang dipilih sebagai *split* poin.

ISSN 2615-6334 (Online)
ISSN 2087-5347 (Print)

B. Dataset

Empat kombinasi dataset digunakan pada penelitian ini *verified-2019, botwiki-2019, cresci-rtbust-2019*, dan *botometer-feedback-2019* berdasarkan [13], [14]. Setiap *dataset* telah ditentukan label/kelas yang dibagi dalam dua kriteria: *Human* atau *Bot*. Terdapat 3725 data *record* dengan jumlah akun berlabel *Human* 2625 dan akun berlabel *Bot* sebanyak 1100 data.

C. Model Evaluasi

Empat model evaluasi digunakan pada penelitian ini untuk menentukan performa klasifikasi *Decision Tree* dalam menentukan akun palsu pada Twitter. *Accuracy*, *Precision*, *Recall* dan kurva ROC-AUC [16]. *Accuracy* dihitung dengan menggunakan *Confusion Matrix* pada Tabel 1 dengan membandingkan penjumlahan *True Positif* (TP) dan *True Negatif* (TN) dengan keseluruhan data sesuai dengan persamaan (9). *Precision* mengukur tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem sesuai persamaan (10), sedangkan *recall* mengukur tingkat keberhasilan sistem untuk menemukan kembali sebuah informasi melalui persamaan (11).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

$$precision = \frac{TP}{TP+FP} \quad (10)$$

$$recall = \frac{TP}{TP+FN} \quad (11)$$

Tabel 1. *Confusion Matrix*

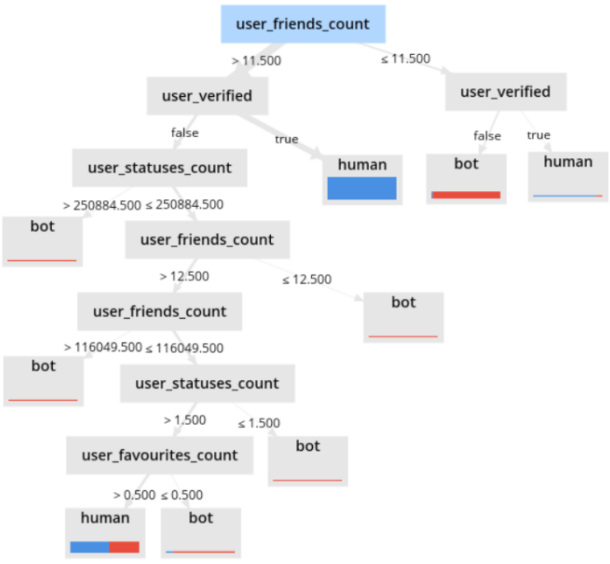
		Nilai Sebenarnya	
		TRUE	**FALSE**
Nilai Prediksi	**TRUE**	True Positive (TP)	False Positive (FP)
	FALSE	False Negative (FN)	True Negative (TN)

Model ROC (*Receiver Operating Characteristics*) sering digunakan untuk menentukan *threshold* dari model klasifikasi *Decision Tree*, sedangkan AUC adalah luas area di bawah kurva ROC, atau integral dari fungsi ROC. Perbandingan AUC (*Area Under Curve*) membuat pengguna mudah dalam membandingkan performa model satu dengan yang lainnya secara visual.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan hasil klasifikasi *Decision Tree* dalam pendeteksian *Twitter Bot* dengan membandingkan beberapa model seleksi atribut. Proses klasifikasi *Decision Tree* dilakukan sesuai diagram pada gambar 1, dimana *dataset* yang memiliki label *Human* dan *Bot* dibagi dalam 2 sub-*dataset*: data *training* dan data *testing*. Pengujian dilakukan dengan membedakan rasio antara data *training* dan data *testing* yang diterapkan pada model seleksi atribut yang berbeda. Data *training* akan digunakan untuk menentukan model *Decision Tree* dengan menggunakan seleksi atribut *Information Gain, Gain Ratio*, dan *Gini Index*.

Klasifikasi *Decision Tree* akan selesai setelah kondisi akhir terpenuhi. Didapat hasil visualisasi klasifikasi *Decision Tree* pada gambar 3. Gambar 3 memaparkan visualisasi grap hasil klasifikasi *Decision Tree* dengan kedalaman 8 cabang. Terlihat atribut yang paling dominan adalah *user_friends_count,* diikuti oleh *user_virified,user_status_count,user_friends_count*.



Gambar 2. Visualisasi Grap Hasil Klasifikasi *Decision Tree*

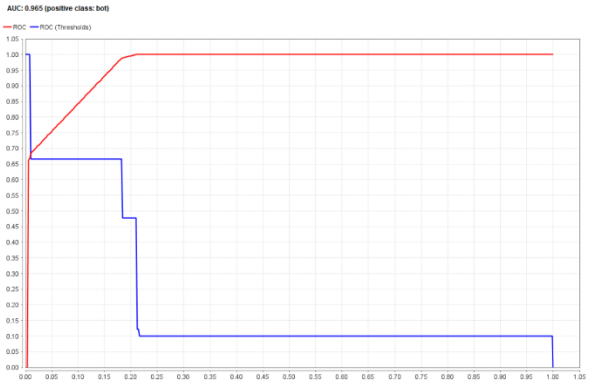
Model yang telah didapat pada Gambar 2 akan dilakukan evaluasi dengan memasukkan data *testing*. Data *testing* didapat dari *dataset* total yang dilakukan pembagian sesuai dengan rasio 0.9/0.1, 0.8/0.2, 0.7/0.3 dimana jumlah terbesar digunakan untuk data *training*, dan rasio terkecil untuk data *testing*. Setelah dilakukan

evaluasi dilakukan pengukuran performa model klasifikasi *Decision Tree* dengan menggunakan empat kriteria: *accuracy*, *precision*, *recall* dan kurva ROC-AUC.

Tabel 2. Hasil pengukuran performa *accuracy*, *precision*, dan *recall*

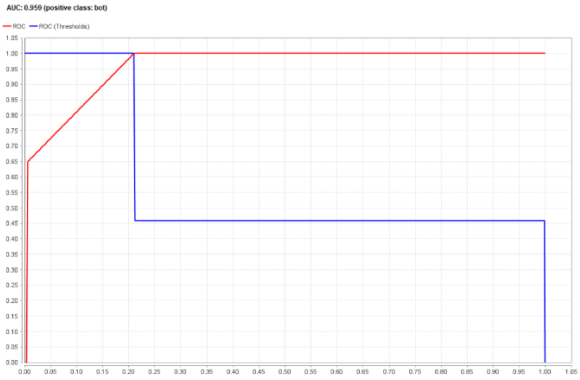
Model Evaluasi DT	Data Training / Data Testing (Rasio)			Rata-Rata (%)
	0.9/0.1	0.8/0.2	0.7/0.3	
Information Gain (accuracy %)	89.63	87.98	88.31	88.64
Gain Ratio (accuracy %)	90.30	87.81	88.42	88.84
Gini Index (accuracy %)	89.63	87.98	88.53	88.71
Information Gain (precision %)	98.21	99.00	99.34	98.85
Gain Ratio (precision %)	96.67	96.19	98.71	97.19
Gini Index (precision %)	98.21	98.04	99.35	98.53
Information Gain (recall %)	64.71	58.24	59.22	60.72
Gain Ratio (recall %)	68.24	59.41	60.00	62.55
Gini Index (recall %)	64.71	58.82	60.00	61.18

Hasil pengukuran performa model klasifikasi *Decision Tree* didapat pada Tabel 2 dengan membandingkan setiap metode seleksi atribut dan rasio data *training* dan data *testing* yang digunakan. Berdasarkan rata-rata *accuracy*, metode *Gain Ratio* memiliki nilai tertinggi dengan 88.84%, namun tidak memiliki perbedaan yang signifikan dengan metode seleksi atribut lainnya.

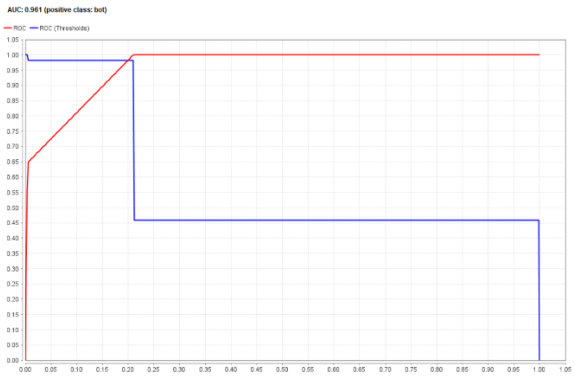


Gambar 3. Kurva ROC-AUC *Gain Ratio*

Untuk perhitungan *precision,* model *Information Gain* memiliki nilai tertinggi dengan 98.85% dan perhitungan *recall* pada model *Information Gain* memiliki nilai terendah dengan 60.72%. Berdasarkan perbandingan rasio data *training* dan data *testing* didapatkan *accuracy* tertinggi pada rasio 0.9/0.1, namun perbedaan tidak terlalu signifikan dengan rasio lainnya.



Gambar 4. Kurva ROC-AUC *Information Gain*



Gambar 5. Kurva ROC-AUC *Gini Index*

Pengukuran performa berikutnya menggunakan kurva ROC-AUC, pada Gambar 3 didapatkan hasil pengukuran AUC sebesar 0.965 untuk metode seleksi atribut *Gain Ratio*. Gambar 4 merupakan hasil pengukuran AUC dari metode seleksi atribut *Information Gain* dengan nilai 0.959 dan nilai 0.961 merupakan perhitungan AUC dari metode *Gini Index* pada Gambar 5. Bersarkan [16] hasil perhitungan AUC untuk model klasifikasi *Decision Tree* dengan menggunakan ketiga metode seleksi atribut tersebut masuk kedalam kategori sangat baik dalam proses klasifikasi untuk mendeteksi *twitter bot*.

V. KESIMPULAN

Pada penelitian ini telah dilakukan model klasifikasi *Decision Tree* untuk menentukan *Twitter bot* dengan memanfaatkan dataset pengguna *Twitter* pada tahun 2019. Hasil

ISSN 2615-6334 (Online)
ISSN 2087-5347 (Print)

pengukuran *accuracy* menunjukkan klasifikasi *Decision Tree* dengan metode seleksi *Information Gain*, *Gain Ratio*, *Gini Index* dengan nilai rata-rata 88.64%, 88.884%, 88.71% secara berurutan. Pengukuran performa AUC dilakukan pada model klasifikasi *Decision Tree* dengan nilai 0.965, 0.959, 0.961 untuk metode seleksi atribut *Gain Ratio, Information Gain*, dan *Gini Index* secara berurutan. Hal tersebut menunjukkan model klasifikasi *Decision Tree* sangat sesuai untuk menentukan *Twitter Bot* dengan memanfaatkan dataset pengguna *Twitter* yang telah ada.

REFERENSI

[1] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the 26th Annual Computer Security Applications Conference on - ACSAC '10*, 2010, p. 1.

[2] S. Gurajala, J. S. White, B. Hudson, B. R. Voter, and J. N. Matthews, "Profile characteristics of fake Twitter accounts," *Big Data Soc.*, vol. 3, no. 2, p. 205395171667423, Dec. 2016.

[3] B. Viswanath *et al.*, "Towards detecting anomalous user behavior in online social networks," *Proc. 23rd USENIX Secur. Symp.*, pp. 223–238, 2014.

[4] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna, "COMPA: Detecting Compromised Accounts on Social Networks," *ISOC Netw. Distrib. Syst. Symp.*, 2013.

[5] D. J. Watts and P. S. Dodds, "Influentials, Networks, and Public Opinion Formation," *J. Consum. Res.*, vol. 34, no. 4, pp. 441–458, Dec. 2007.

[6] M. Mateen, M. A. Iqbal, M. Aleem, and M. A. Islam, "A hybrid approach for spam detection for Twitter," in *2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2017, pp. 466–471.

[7] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "An effective unsupervised network anomaly detection method," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics - ICACCI '12*, 2012, p. 533.

[8] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-Based Anomaly Detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 1–39, Mar. 2012.

[9] P. Sharma and U. Bhardwaj, "Machine Learning based Spam E-Mail Detection," *Int. J. Intell. Eng. Syst.*, vol. 11, no. 3, pp. 1–10, Jun. 2018.

[10] M. Ebrahimi, C. Suen, O. Ormandjieva, and A. Krzyzak, "Recognizing Predatory Chat Documents using Semi-supervised Anomaly Detection," *Electron. Imaging*, vol. 2016, no. 17, pp. 1–9, Feb. 2016.

[11] W. Wu, J. Alvarez, C. Liu, and H.-M. Sun, "Bot detection using unsupervised machine learning," *Microsyst. Technol.*, vol. 24, no. 1, pp. 209–217, Jan. 2018.

[12] S. Miller and C. Busby-Earle, "The Impact of Different Botnet Flow Feature Subsets on Prediction Accuracy Using Supervised and Unsupervised Learning Methods," *J. Internet Technol. Secur. Trans.*, vol. 5, no. 2, Jun. 2016.

[13] K. Yang, O. Varol, P. Hui, and F. Menczer, "Scalable and Generalizable Social Bot Detection through Data Selection," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, pp. 1096–1103, Apr. 2020.

[14] M. Mazza, S. Cresci, M. Avvenuti, and M. Tesconi, "RTbust : Exploiting Temporal Patterns for Botnet Detection on Twitter," 2019.

[15] W. Du, W. Du, Z. Zhan, and Z. Zhan, "Building decision tree classifier on private data," *Proc. IEEE Int. Conf. Privacy, Secur. data mining-Volume 14*, pp. 1–8, 2002.

[16] Gorunescu, Florin, "Data Mining Concept Model and Techniques", 2011.

[17] Top 15 Most Popular Social Networking Sites 2019, https://www.webcitation.org/78VNU1R4J, accessed 20 Maret 2020.