Penerapan Machine Learning dalam mendeteksi Fake Account pada Instagram

Hendy Gunawan, Yulia, Gregorius Satia Budhi
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto No.121-131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658
E-mail: hendy200115@gmail.com, yulia@petra.ac.id, greg@petra.ac.id

ABSTRAK

Instagram adalah media sosial keempat yang paling banyak digunakan dalam hal jumlah pengguna aktif. Saat ini, banyak masyarakat yang berusaha ingin meningkatkan jumlah *followers*-nya untuk alasan lain seperti mendapatkan ketenaran atau ingin terkenal dan dapat dipercaya oleh orang karena memiliki jumlah *followers* yang banyak. Oleh karena itu, masyarakat membuat *fake account* yang digunakan untuk meningkatkan jumlah *followers*-nya dan juga sebagai tempat untuk melakukan tindakan kejahatan seperti penipuan dan *cyberbullying*. Fleksibilitas dan penyebaran penggunaan seperti itu telah menjadikan *Instagram* sebagai *platform* yang digunakan untuk perkembangbiakan *fake account*.

Dalam penelitian ini, dirancang suatu aplikasi berbasis *website* yang dapat mendeteksi akun pada Instagram apakah tergolong *fake* atau *real account*. Deteksi tersebut dilakukan dengan menggunakan *machine learning* dengan metode *Support Vector Machine*, *Naïve Bayes*, *Random Forest* dan *Adaptive Boosting* untuk mendeteksi *fake* atau *real account* pada Instagram. Metode yang digunakan dibandingkan performanya untuk mencari metode mana yang paling sesuai dalam mendeteksi *fake* atau *real account* pada Instagram. Penggunaan *k-fold cross validation* digunakan untuk mencegah terjadinya *overfitting* pada *machine learning*. Berdasarkan pengujian yang telah dilakukan, bahwa *AdaBoost* dapat digunakan untuk klasifikasi akun pada Instagram dengan hasil akurasi yang didapatkan 92.5%, *Random Forest* sebesar 91.7%, *Support Vector Machine* sebesar 90.7% dan *Naïve Bayes* sebesar 83.6%.

Kata Kunci: *Machine learning, Support Vector Machine, Naïve Bayes, Random Forest, Adaptive Boosting, Deteksi Akun Instagram*

ABSTRACT

Instagram is the fourth most used social media in terms of the number of active users. Currently, many people are trying to increase the number of followers for other reasons such as gaining fame or wanting to be famous and trustworthy by people because they have a large number of followers. Therefore, people create fake accounts that are used to increase the number of their followers and also as a place to commit crimes such as fraud and cyberbullying. Such flexibility and spread of use has made Instagram a platform used for the proliferation of fake accounts.

In this research, a website based application was designed that can detect accounts on Instagram whether they are fake or real accounts. The detection is carried out using machine learning with the Support Vector Machine, Naïve Bayes, Random Forest and Adaptive Boosting methods to detect fake or real accounts on Instagram. The method used is compared to its performance to find which method is the most appropriate in detecting fake or real accounts on Instagram. The use of k-fold cross validation is used to prevent overfitting in machine learning. Based on the tests that have been carried out, that AdaBoost can be used for account classification on Instagram with an accuracy of 92.5%, Random Forest 91.7%, Support Vector Machine 90.7% and Naïve Bayes 83.6%.

Keywords: *Machine Learning, Support Vector Machine, Naïve Bayes, Random Forest, Adaptive Boosting, Instagram Account Detection*

1. PENDAHULUAN

Instagram adalah media sosial keempat yang paling banyak digunakan dalam hal jumlah pengguna aktif [9]. *Instagram* digunakan oleh para penggunanya untuk berbagi gambar, karya dan juga sebagai media untuk berkomunikasi. Seiring dengan bertambahnya waktu, peranan media sosial *Instagram* juga semakin mengalami perkembangan. Selain sebagai media untuk berkomunikasi, *Instagram* juga digunakan sebagai sarana untuk berbisnis dan politik. Dalam beberapa tahun terakhir, banyak selebriti telah membuat akunnya di *Instagram*. Para selebriti menggunakan *Instagram* untuk mengembangkan bisnis dan penggemarnya [4]. Selain itu, banyak selebriti lainnya menggunakan *Instagram* sebagai *platform* untuk beriklan. Ketika seseorang telah mendapatkan jumlah *follower*-nya lebih dari seratus ribu bahkan jutaan, tidak mengherankan jika menggunakan akun tersebut digunakan sebagai sumber penghasilan yang menguntungkan baginya.

Saat ini, banyak masyarakat yang berusaha ingin meningkatkan jumlah *followers*-nya untuk alasan lain seperti mendapatkan ketenaran atau ingin terkenal dan dapat dipercaya oleh orang karena memiliki jumlah *followers* yang banyak [19]. Oleh karena itu, masyarakat membuat *fake account* yang digunakan untuk meningkatkan jumlah *followers*-nya dan juga sebagai tempat untuk melakukan tindakan kejahatan seperti penipuan dan *cyberbullying* [13]. *Fake account* adalah akun yang digunakan oleh individu untuk mengekspresikan diri, memanfaatkan media sosial, dan melakukan aktivitas lain di dunia maya tanpa mengungkapkan identitas aslinya kepada orang lain [23]. *Real account* adalah pemilik akun menggunakan identitas *real*-nya supaya orang lain dapat mengenali dirinya dengan mudah. Identitas tersebut meliputi nama pendek, nama lengkap, biografi, bahkan foto profil [16].

Fake account yang dibuat atas nama orang atau organisasi dapat berbahaya dan merusak reputasi orang dan bisnis [6]. Hal ini dapat menyebabkan penurunan jumlah like dan *followers* asli mereka. Semua jenis *fake account* memiliki efek buruk pada keuntungan media sosial untuk bisnis pemasaran dan bisnis periklanan. Fleksibilitas dan penyebaran penggunaan seperti itu telah menjadikan *Instagram* sebagai *platform* yang digunakan untuk perkembangbiakan *fake account*. Hal seperti inilah yang membuat jumlah *fake account* di *Instagram* terus meningkat. Karena banyaknya *fake account* yang bermunculan maka perlu adanya

suatu sistem untuk mendeteksi apakah *account Instagram* ini *fake account* atau *real account* sehingga dengan demikian maka dapat memberikan kenyamanan dan keamanan kepada pengguna *Instagram* dalam bersosial media khususnya *Instagram.*

Pada penelitian sebelumnya *yaitu* oleh Albayati & Altamimi, pada tahun 2019 [1]. *Identifying Fake Facebook Profiles Using Data Mining Techniques.* Tujuan dari penelitian ini adalah untuk mengatasi masalah ini dengan memanfaatkan teknik data *mining* untuk mendeteksi profil palsu di *Facebook.* Metode yang diusulkan dari penelitian ini adalah 3 algoritma *supervised learning* (K-NN, SVM, dan ID3) dan 2 algoritma *unsupervised learning* (k-Means, dan k-medoids). Penelitian ini menyatakan bahwa akurasi dari *Decision Tree* 97.76%, SVM 95.72% dan K-NN dengan k=3 sebesar 91.45%. Untuk algoritma *unsupervised learning,* hasil yang didapatkan K-*means* sebesar 67.31% dan K-*medoids* sebesar 67.01%. Algoritma *supervised learning* yang unggul daripada algoritma *unsupervised learning*. Penelitian ini belum menggunakan *cross validation*.

Kemudian penelitian oleh Sheikhi, pada tahun 2020 [19]. *An Efficient Method for Detection of Fake Accounts on the Instagram Platform.* Penelitian ini bertujuan untuk mencari metode yang mana yang paling efisien dalam mendeteksi *fake account* pada *Instagram*. Algoritma yang digunakan cukup banyak yaitu *Hoeffding tree, Random Forest, Support Vector Machine, Naïve Bayes, Multilayer Perceptron,* dan *Bagged Decision Tree*. Penelitian ini menyatakan bahwa akurasi dari *Bagged Decision Tree* ini mencapai 98.45%, *Random Forest* mencapai 97.2%, *Naïve Bayes* mencapai 94.58% dan *Support Vector Machine* berada diurutan akhir yaitu 68.68%. Percobaan ini dilakukan dengan menggunakan *cross validation* 10-*fold.* Dari penelitian ini bisa dilihat bahwa penggunaan *bagging method* dapat mendeteksi *fake account* dengan lebih akurat.

Kemudian penelitian oleh Purba et al, pada tahun 2020 [12]. tentang *Classification of instagram fake users using Supervised Machine Learning Algorithms.* Penelitian ini bertujuan untuk mengklasifikasikan *fake users* menggunakan algoritma *Supervised Learning*. Algoritma yang digunakan cukup banyak yaitu *Random Forest, Multilayer Perceptron, Logistic Regression, Naïve Bayes* dan *J48 Decision Tree.* Percobaan dilakukan dengan variasi klasifikasi 2-*classes* (*fake or authentic users*) dan 4-*classes* (*authentic users, active fake users, inactive fake users* dan *spammers*). Pada 2-*classes* algoritma *Random Forest* mendapatkan akurasi tertinggi yaitu 90.09% dan urutan ke 2 yaitu *J48 Decision Tree* akurasi yang dicapai yaitu 88.34%. Pada 4-*classes* algoritma *Random Forest* mendapatkan akurasi tertinggi yaitu 91.76% dan urutan ke 2 yaitu *J48 Decision Tree* akurasi yang dicapai yaitu 88.28%. Percobaan ini dilakukan dengan menggunakan *cross validation* 10-*fold.*

Kemudian penelitian oleh Sutter et al, pada tahun 2021 [21]. tentang *Predicting Psychological Distress from Ecological Factors: A Machine Learning Approach.* Penelitian ini bertujuan untuk memprediksi tekanan psikologis seseorang dari faktor ekologis. Algoritma yang digunakan cukup banyak pada penelitian ini yaitu *Logistic Regression*, SVM, ANN, *Naïve Bayes*, *Decision Tree* dan 3 *ensemble* algoritma yaitu *Random Forest, Adaptive Boosting,* dan *Gradient Boosting*. Percobaan dilakukan dengan melakukan *cross validation* 10-*fold.* Penelitian ini menyatakan bahwa *Logistic Regression* mendapatkan akurasi tertinggi dari semua algoritma yang diujikan yaitu 81.1% dan algoritma *ensemble* cukup baik seperti *Random Forest* mendapatkan akurasi 79.5%, *Adaptive Boosting* mendapatkan akurasi 81% dan *Gradient Boosting* mendapatkan akurasi 81%.

Pada penelitian [1] menyatakan bahwa SVM cukup baik mampu mendapatkan akurasi 95.72%. Penelitian [12] menyatakan *Random Forest* mampu mendapatkan akurasi 90.09% dan pada penelitian [19] juga menyatakan *Random Forest* mendapatkan akurasi 97.2%, *Naïve Bayes* 94.58% dan SVM 68.68%. Pada penelitian [21] menyatakan algoritma *ensemble* cukup baik seperti *Random Forest* mendapatkan akurasi 79.5%, *Adaptive Boosting* mendapatkan akurasi 81% dan *Gradient Boosting* mendapatkan akurasi 81%. Penggunaan k-*fold cross validation* merupakan salah satu metode pengambilan sampel ulang data yang paling banyak digunakan untuk menilai kemampuan generalisasi model prediktif dan untuk mencegah *overfitting* [3].

Berdasarkan latar belakang masalah yang diuraikan, maka masalah yang dapat dirumuskan dalam skripsi ini adalah mengidentifikasi mana algoritma *machine learning* (*support vector machine, naïve bayes classifier,* dan 2 algoritma *ensemble* yaitu *random forest* dan *adaptive boosting*) yang paling sesuai untuk mendeteksi *fake account* pada Instagram diukur dari hasil uji akurasi, presisi, *recall*, dan *f1-score*. Apakah *machine learning* mengalami *overfitting* dalam data ini ? dan apa saja kriteria penentu *fake account* pada media sosial Instagram ?

Dibandingkan dengan penelitian yang sudah dilakukan sebelumnya, pada skripsi ini akan diuji algoritma *Support Vector Machine*, *Naïve Bayes*, *Random Forest* dan *Adaptive Boosting* untuk mengidentifikasi mana metode yang mampu melakukan identifikasi yang paling sesuai untuk mendeteksi *fake account* pada *Instagram.* Performa algoritma yang diuji menggunakan pengukuran seperti: akurasi, *recall*, presisi dan *f1-score* serta pengujian *confusion matrix* dan penggunaan k-*fold cross validation*. Sehingga dapat mengetahui performa yang stabil dari tiap algoritma yang diujikan.

2. LANDASAN TEORI

2.1. *Fake Account*

Fake account adalah akun yang digunakan oleh individu untuk mengekspresikan diri, memanfaatkan media sosial, dan melakukan aktivitas lain di dunia maya tanpa mengungkapkan identitas aslinya kepada orang lain [23].

2.2. *Web Scraping*

Web Scraping adalah proses pengambilan data dari sebuah *website*. Data tersebut umumnya disimpan dalam sebuah format tertentu. Keuntungan dari *web scrapping* adalah dapat dengan cepat mengumpulkan sebuah data. *Web scrapping* dapat dilakukan dengan menggunakan *web scraper*. Teknik – teknik yang digunakan dalam *web scrapping* antara lain: *Parsing HTML, Parsing DOM* dan *XPath* [11]. Data inputan dari *user* berupa *username Instagram* diambil dengan menggunakan *instagram profile scrapers* [18].

2.3. *Dataset Instagram Fake Spammer Genuine Accounts*

Dataset yang digunakan pada skripsi ini diambil dari sebuah *website kaggle.com* yang berjudul *Instagram Fake Spammer Genuine Accounts* [2]. Jumlah data yang ada sebanyak 696 (enam ratus sembilan puluh enam). Dari data tersebut, sebanyak 348 (tiga ratus empat puluh delapan) data dengan label akun *fake* atau *spammer* dan 348 (tiga ratus empat puluh delapan) data dengan label akun yang *genuine* atau *real*. *Dataset* ini terdiri dari 12 *attribute* yang telah dibuat oleh pembuatnya. Untuk lebih jelasnya akan dijabarkan *attribute* data sebagai berikut: ada atau tidak adanya *profile pic*, rasio angka pada *username*, jumlah kata pada *fullname*, rasio angka pada *fullname*, sama atau tidaknya *username*

pada *fullname*, panjang deskripsi, ada atau tidak adanya *external url*, tipe akun *public* atau *private*, jumlah *post*, jumlah *followers*, jumlah *followings* dan label *fake* atau *genuine*.

2.4. *Support Vector Machine*

Support Vector Machine adalah algoritma *machine learning* yang dapat digunakan untuk klasifikasi dengan memprediksi *label*/*group* dari objek. *SVM* dapat melakukan klasifikasi dengan mencari *hyperplane* yang membedakan antar kelas pada plot ruang n-dimensi [14]. Tujuan dari model ini adalah menemukan *hyperplane* terbaik yang memisahkan dua buah kelas pada sebuah tempat sehingga terbentuknya *classifier*. *Hyperplane* adalah sebuah fungsi yang digunakan untuk menjadi pemisah antar kelas. *Hyperplane* yang terbaik atau optimal dapat dilakukan dengan cara mengukur nilai *margin* dari *hyperplane* tersebut dan menemukan titik maksimalnya. Objek terluar atau yang paling dekat dengan *hyperplane* ini yang disebut sebagai *support vector*.

2.5. *Naïve Bayes*

Naïve Bayes Classifier adalah metode *machine learning* yang melakukan klasifikasi berdasarkan probabilitas dari objek. Teorema *Bayes* mengasumsikan semua atribut independen atau tidak saling ketergantungan nilai pada variabel tiap kelas objek [17]. Rumus untuk *Naïve Bayes* menjadi seperti :

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad (1)$$

Dimana:

P(A|B) = Peluang terjadinya A jika B terjadi
P(B|A) = Peluang B terjadi saat A muncul
P(A) = Peluang terjadinya A
P(B) = Peluang terjadinya B

2.6. *Random Forest*

Random Forest adalah salah satu algoritma *ensemble machine learning* yang dapat digunakan baik dalam klasifikasi ataupun regresi. *Random forest* terdiri dari banyak *decision tree* yang beroperasi secara *ensemble* [24]. Cara kerja dari algoritma ini adalah kumpulan dari *decision tree* yang masing - masing akan menghasilkan suatu keputusan atau *decision*. Dari kumpulan keputusan tersebut akan diambil satu keputusan mayoritas. Untuk membentuk *decision tree*, akan diambil beberapa *feature* dari *training* data. Setelah sejumlah besar *decision tree* dihasilkan, langkah akhir yaitu memilih kelas yang paling populer atau *majoring vote*. Konsep inilah yang dinamakan *random forest* [5].

2.7. *Adaptive Boosting*

Adaptive Boosting adalah teknik *boosting* yang digunakan sebagai metode *ensemble* dalam *machine learning.* Disebut *adaptive boosting* karena bobot atau *weights* ditetapkan kembali ke setiap *instance*, dengan bobot yang lebih tinggi ditetapkan ke *instance* yang diklasifikasikan secara salah [8]. Algoritma *adaptive boosting* secara iteratif menggabungkan beberapa *weak classifiers*. Proses ini dimulai dengan bobot yang sama untuk semua data *training*. Ketika data *training* salah diklasifikasikan, maka bobot atau *weights* pada data ini di *boosting*, maka pengklasifikasi baru dibuat menggunakan bobot baru yang tidak sama. Proses ini diulang untuk satu set *classifiers* [7]. Ini akan menjaga model *training* sampai kesalahan *error*-nya rendah.

2.8. *Confusion Matrix*

Confusion Matix adalah cara pengukuran performa untuk klasifikasi *machine learning* [10]. Matrix ini menunjukan nilai *true class* (nilai sebenarnya) dan *predicted class* (nilai hasil prediksi) oleh *machine learning*. Ada dua kelas yaitu positif dan negatif untuk masing – masing *true class* dan *predicted class*.

Berikut adalah cara pengukuran performa *confusion matrix.*

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$F1 - Score = \frac{2TP}{2TP+FP+FN} \quad (4)$$

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

2.9. *K-Fold Cross Validation*

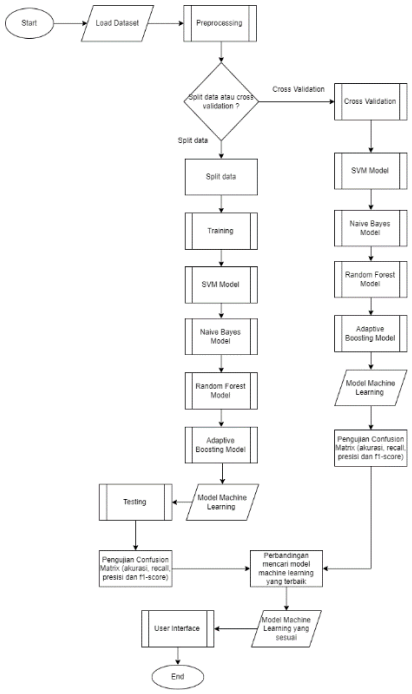
K-*Fold Cross Validation* merupakan proses validasi model yang dilakukan dengan cara membagi *dataset* menjadi k bagian atau *fold*, dan dilakukan iterasi sebanyak k kali. Pada setiap iterasi, setiap bagian atau *fold* digunakan sebagai *testing dataset* sebanyak satu kali secara bergantian. Bagian yang lainnya k-1*fold* digunakan sebagai *training dataset*. Hal ini bertujuan untuk melakukan *testing* terhadap model menggunakan data yang belum pernah dilihat sebelumnya. Untuk penggunaan jumlah *fold* terbaik untuk uji validitas, dianjurkan menggunakan 10-*fold cross validation* dalam model *machine learning* [3].

2.10. *Overfitting*

Overfitting adalah kesalahan pemodelan dalam statistik yang terjadi saat fungsi terlalu dekat dengan kumpulan titik data yang terbatas [22]. Akibatnya, model data ini berguna dalam referensi hanya ke data awalnya, dan tidak untuk kumpulan data lainnya atau data yang baru. *Overfitting* bisa diatasi dengan cara melakukan *cross-validation* dan menyisihkan sebagian data untuk dilakukan *testing* pada model data.

3. DESAIN SISTEM

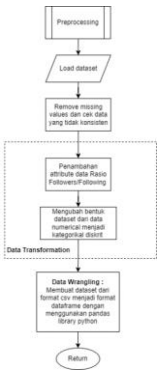
Desain sistem membahas alur terbentuknya *input* data, proses dan hasil *output* yang diharapkan dan disertakan dengan *user interface* berbentuk *website* deteksi *fake* dan *real account*. Alur dari penelitian ini secara garis besar tergambar pada Gambar 1 berikut.



Gambar 1. Alur Sistem

3.1. *Preprocessing Dataset*

Tujuan dari *preprocessing* data adalah untuk memastikan data yang akan digunakan pada model *machine learning* sudah berkulitas dan tidak ada lagi data *noise* untuk proses klasifikasi. Proses preprocessing data dapat dilihat pada Gambar 2 berikut.



Gambar 2. *Preprocessing dataset*

3.2. Data *Transformation*

Data *transformation* adalah tahap perubahan data dengan merubah format tertentu pada *dataset*. Tahap ini dilakukan dengan tujuan agar data tersebut dapat lebih sesuai dalam proses klasifikasi nantinya. Tranformasi data yang dilakukan pada tahap ini ada dua yaitu: transformasi bentuk dan transformasi nilai yang terdapat pada Gambar 2. Untuk lebih jelasnya proses data transformasi dapat dilihat pada proses berikut:

1. Transformasi nilai adalah transformasi yang dilakukan pada *dataset* untuk menambahkan *attribute* data baru hasil dari perhitungan dari *attribute* data sebelumnya. Transformasi nilai yang dilakukan adalah penambahan *attribute* rasio *followers*/*followings.* Tujuannya adalah untuk mengetahui sebuah akun seberapa sering melakukan *follows* kepada akun orang lain dan di *follow* oleh orang tersebut. Semakin tinggi nilai rasionya, maka kualitas akun tersebut semakin bagus. Persamaan rasio ini dapat dijabarkan pada persamaan 6.

$$Rasio\ Followers\ Followings = \frac{Jumlah\ Followers}{Jumlah\ Followings} \quad (6)$$

2. Transformasi bentuk pada attribut *description length*, *nums/length username*, *nums/length fullname*, *fullname words*, *post*, *followers*, *following*, dan Rasio *followers* per *followings*. Persamaan 7 hingga persamaan 30 adalah persamaan yang membahas mengenai transformasi bentuk pada tiap atribut yang ada.

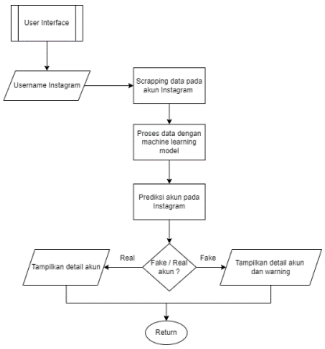
$Description\ length < 50$, *Description length = Low* (7)
$50 \leq Description\ length < 100$, *Description length = Middle* (8)
$100 \leq Description\ length \leq 150$, *Description length = High* (9)
$Nums/length\ username < 0.30$, *Nums/length username = Low* (10)
$0.30 \leq Nums/length\ username < 0.60$, *Nums/length username = Middle* (11)
$0.60 \leq Nums/length\ username \leq 1.0$, *Nums/length username = High* (12)
$Nums/length\ fullname < 0.30$, *Nums/length fullname = Low* (13)
$0.30 \leq Nums/length\ fullname < 0.60$, *Nums/length fullname = Middle* (14)
$0.60 \leq Nums/length\ fullname \leq 1.0$, *Nums/length fullname = High* (15)
$Fullname\ words < 4$, *Fullname words = Low* (16)
$4 \leq Fullname\ words < 8$, *Fullname words = Middle* (17)
$8 \leq Fullname\ words \leq 12$, *Fullname words = High* (18)
$Post < 50$, *Post = Low* (19)
$50 \leq Post < 100$, *Post = Middle* (20)
$Post \geq 100$, *Post = High* (21)
$Followers \geq 300$, *Followers = High* (22)
$Followers < 300$, *Followers = Low* (23)
$Followings \geq 500$, *Followings = High* (24)
$Followings < 500$, *Followings = Low* (25)
Rasio < 0.5, Rasio = *Spammer* (26)
$0.5 \leq Rasio < 1$, Rasio = *Suspicious* (27)
$1 \leq Rasio < 2$, Rasio = Normal (28)
$2 \leq Rasio < 10$, Rasio = *Micro Influencer* (29)
Rasio ≥ 10 Rasio = *Influencer* (30)

3.3. Training dan Testing Model

Tahap awal proses ini adalah *dataset* tersebut dibagi menjadi *training* dan *testing*. Selanjutnya data yang di *training* akan diproses menggunakan keempat metode. Setelah data di *training*, maka akan mendapatkan model *machine learning* dari data yang sudah di *trained* tadi. Setelah model terbentuk, maka model akan diuji dengan menggunakan data *testing* dan pengujian *confusion matrix* berupa akurasi, *recall*, *precision*, dan f1-*score.* Pada pengujian *split dataset*, dilakukan dengan 80 % *training* dan 20 % *testing* sedangkan dengan pengujian pada *cross validation* menggunakan 10-*fold cross validation.* Setelah semua model selesai di *training*, maka tahap selanjutnya yaitu melakukan uji *overfitting* pada tiap metode.

3.4. Implementasi Program

Implementasi program yang dibuat dalam bentuk *website.* Terdapat bagian *homepage* merupakan halaman tampilan awal dari *website*. Terdapat sebuah kolom *search bar* dimana pengguna bisa memasukan inputan *username* Instagram yang akan deteksi akun Instagram tersebut apakah *fake* atau *real account*. Model *machine learning* yang telah dibangun akan di *load* modelnya untuk memprediksi akun pada Instagram. *Output* dari model *machine learning* ini berupa hasil prediksinya. Alur deteksi akun pada Instagram dapat dilihat pada Gambar 3.



Gambar 3. Deteksi Akun pada Instagram

4. PENGUJIAN SISTEM

Pengujian dilakukan untuk mendapatkan metode mana yang terbaik atau paling sesuai dari beberapa metode yang telah diusulkan, sehingga dapat menghasilkan performa klasifikasi yang tinggi. Metode dengan hasil terbaik akan diterapkan dalam pembuatan *website* deteksi akun *Instagram*.

4.1. Pengujian Dengan Perbandingan 80:20

Pada pengujian ini dilakukan dengan pembagian *dataset* menjadi 2 bagian menggunakan *split dataset* dengan perbandingan 80:20 antara data *training* dan data *testing.*

Tabel 1. Hasil Pengujian tiap Metode

	SVM	*Naïve Bayes*	*Random Forest*	*AdaBoost*
Std Deviasi	±0.022	±0.034	±0.034	±0.018
Accuracy	0.901	0.829	0.913	0.920
Precision	0.921	0.767	0.923	0.931
Recall	0.874	0.948	0.904	0.907
F1-Score	0.896	0.847	0.913	0.919
Training Time	0.797 ms	0.558 ms	14.681 ms	5.418 ms
Testing Time	0.558 ms	0.477 ms	0.558 ms	0.624 ms
Memory Usage	2.078 MB	0.615 MB	0.584 MB	0.602 MB

Hasil pengujian pada Tabel 1 dapat dianalisa hasilnya sebagai berikut. Dari keempat metode yang diujikan, penggunaan metode *ensemble learning* seperti *Random Forest* dan *AdaBoost* lebih unggul daripada metode *single classifier* seperti *SVM* dan *Naïve Bayes.* Perbedaan performa pada tiap metode tidak terlalu jauh perbedaannya. Hasil dari keempat metode yang telah diujikan bahwa *AdaBoost* metode yang memiliki performa yang paling sesuai baik dari akurasi, *precision, recall, f1-score* dan juga waktu *training* dan *testing* yang tidak terlalu lama dan juga tidak terlalu cepat. Serta penggunaan *memory* yang tidak terlalu besar. Sehingga metode *AdaBoost* akan diterapkan kedalam *website* untuk deteksi akun pada *Instagram.*

4.2. Pengujian Dengan 10-*Fold Cross Validation*

Pada pengujian ini dilakukan dengan membagi *dataset* menjadi 10 bagian *subset* data. Setiap *subset* data berisi 69 (enam puluh sembilan) data. Dilakukan pengulangan iterasi sebanyak 10 kali untuk *training* dan *testing*. Pada setiap iterasi yang dilakukan disisakan satu *subset* untuk *testing* dan *subset* yang lain digunakan untuk *training.*

Tabel 2. Hasil Pengujian 10-Fold Cross Validation

	SVM	*Naïve Bayes*	*Random Forest*	*AdaBoost*
Std Deviasi	±0.024	±0.026	±0.031	±0.032
Accuracy	0.907	0.836	0.917	0.925
Precision	0.927	0.768	0.929	0.939
Recall	0.881	0.963	0.905	0.905
F1-Score	0.903	0.854	0.916	0.921
Training Time	1.196 ms	1.562 ms	37.503 ms	13.851 ms
Testing Time	0.598 ms	1.562 ms	1.795 ms	1.561 ms
Memory Usage	0.576 MB	0.610 MB	0.565 MB	0.602 MB

Hasil pengujian pada Tabel 2 dapat dianalisa hasilnya sebagai berikut. Pada pengujian tiap metode menggunakan 10-*Fold Cross Validation* dengan tujuan dilakukannya iterasi sebanyak 10-*Fold* adalah membantu untuk menghindari terjadinya *overfitting*. Selain itu *Cross Validation* digunakan untuk memperkirakan keterampilan model pada *machine learning* untuk menggunakan data yang belum pernah dilihat sebelumnya. Penggunaan 10-*Fold Cross Validation* ini dapat menambah performa tiap metode sekitar 1% dan tidak terlalu jauh berbeda. Waktu komputasi keempat metode menggunakan 10-*Fold Cross Validation* lebih lama daripada menggunakan *split dataset*.

4.3. Pengujian *Overfitting*

Dari pengujian yang telah dilakukan menggunakan *split dataset* dengan perbandingan 80:20 dan menggunakan 10-*Fold Cross Validation*. Didapatkan hasil *training* dan *testing score* serta *MSE training* dan *testing* pada tiap metode. Hasil pengujian *Overfitting* pada tiap metode dapat dilihat pada Tabel 3 dan 4.

Tabel 3. Tabel Pengujian *Overfitting* tiap Metode tanpa 10-*Fold Cross Validation*

Metode	Training Score	Testing Score	MSE Training	MSE Testing
SVM	0.912	0.907	0.088	0.093
Naïve Bayes	0.831	0.843	0.169	0.157
Random Forest	0.955	0.914	0.045	0.086
AdaBoost	0.926	0.921	0.074	0.079

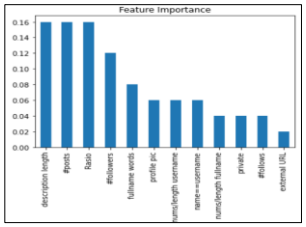
Tabel 4. Tabel Pengujian *Overfitting* tiap Metode dengan 10-*Fold Cross Validation*

Metode	Training Score	Testing Score	MSE Training	MSE Testing
SVM	0.911	0.899	0.089	0.101
Naïve Bayes	0.837	0.855	0.163	0.145
Random Forest	0.954	0.913	0.046	0.087
AdaBoost	0.931	0.913	0.069	0.087

Untuk mengetahui seberapa jauh model *machine learning* dalam bekerja, dapat dilihat dari nilai *training* dan *testing score* serta nilai *MSE training* dan *testing* pada metode tersebut. Jika *error rates* pada *training dataset* rendah, dan *error rates* pada *testing dataset* tinggi, ini berarti *overfitting*. Jika *error rates* pada *training dataset* dan *testing dataset* tinggi, ini berarti *underfitting*, model tidak menangkap *trend* dengan baik. Sebuah model yang tidak *overfitting* maupun *underfitting* bisa dikatakan sebagai *Goodfit* [15]. Dari hasil pengujian *overfitting* yang dilakukan pada tiap metode menunjukan bahwa nilai *training* dan *testing score* yang sangat kecil perbedaannya serta nilai *MSE training* dan *testing* yang sangat kecil. Perbedaan nilai *training* dan *testing score* serta nilai *MSE training* dan *testing* pada metode tersebut tanpa menggunakan *Cross Validation* ataupun menggunakan *Cross Validation* perbedaannya tidak terlalu berbeda. Sehingga dapat disimpulkan bahwa keempat metode ini tidak mengalami *overfitting* pada data ini. Model ini mampu mempelajari data yang ada untuk menentukan pola modelnya.

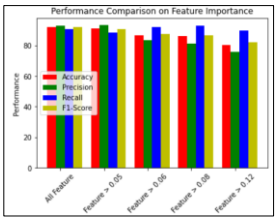
4.4. Kriteria Penentu *Fake Account*

Dalam menentukan kriteria penentu *fake account* pada *Instagram* dapat dilakukan dengan cara menguji dan menganalisa *feature importance* pada metode *AdaBoost*. Metode *AdaBoost* digunakan karena metode ini memiliki performa yang paling sesuai atau bagus dari ketiga metode lain yang diujikan. *Feature importance* mengacu pada teknik yang menghitung skor untuk semua *feature* masukan untuk model. Skor yang lebih tinggi berarti bahwa *feature* tertentu akan memiliki efek yang lebih besar pada model yang digunakan untuk memprediksi variabel tertentu [20]. *Feature importance* pada *AdaBoost* dapat dilihat pada Gambar 4.



Gambar 4. *Feature Importance* pada *AdaBoost*

Pada Gambar 4 dapat dilihat *feature importance* pada *AdaBoost*. *Features* yang memiliki skor tertinggi ada 4 *features* yaitu *description length* dengan skor 0.16, *#post* dengan skor 0.16, rasio dengan skor 0.16 dan #*followers* dengan skor 0.12. Untuk menguji *feature importance* ini seberapa pengaruh terhadap proses klasifikasinya akan diuji beberapa pengujian dengan *feature* dengan skor diatas 0.05, *feature* dengan skor diatas 0.06, *feature* dengan skor diatas 0.08 dan diatas 0.012. Perbandingan hasil performa *AdaBoost* pada pengujian *feature importance* dapat dilihat pada Gambar 5.



Gambar 5. Perbandingan Hasil Performa *AdaBoost* pada Pengujian *Feature Importance*

Pada Gambar 5 dapat dilihat hasil performa *AdaBoost* pada pengujian *Feature Importance.* Bahwa semakin banyak *feature* yang digunakan maka hasil performanya semakin bagus, apabila *feature* tersebut dihilangkan maka hasil performa (akurasi, *precision*, *f1-score*) dari metode *AdaBoost* akan semakin mengalami penurunan. Artinya bahwa *feature* - *feature* yang dihilangkan tersebut memiliki pengaruh dalam klasifikasinya.

Dari hasil Gambar 5 tersebut dapat disimpulkan bahwa nilai *recall* atau sensitivitas dapat menjadi acuan karena *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Artinya klasifikasi *fake account* berhasil dideteksi sebagai positif oleh model. Semakin tinggi skor *feature* yang di *training* maka nilai *recall* semakin meningkat. Dimana pada Gambar 5 dapat dilihat bahwa nilai *recall* mengalami peningkatan. Meskipun pada *feature* di atas skor 0.12 mengalami penurunan karena hanya 3 *feature* penting saja yang di *training* dimana seharusnya ada 4 *feature* yang paling berpengaruh.

Untuk menentukan kriteria pada penentu *fake account* dapat dilihat pada Gambar 4 dimana *feature importance* tersebut adalah hasil pengujian dengan menggunakan semua *feature* yang di *training* menggunakan *AdaBoost*. Hasil pada Gambar 4 tersebut bahwa ada 4 *Feature* dengan skor tertinggi yaitu *description length* dengan skor 0.16, *#post* dengan skor 0.16, rasio dengan skor 0.16 dan *#followers* dengan skor 0.12. Dimana rasio ini merupakan hasil transformasi nilai baru dari *feature followers* dan *following* pada *dataset*. Artinya keempat *feature* atau *attribute* ini sangat berpengaruh dalam menentukan *fake account* pada Instagram.

4.5. Pengujian Aplikasi

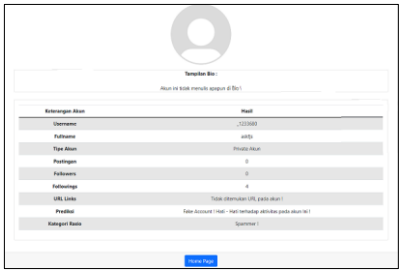
Pada saat pertama kali aplikasi dijalankan akan menampilkan *form input* berupa *username* Instagram *user*. Tampilan halaman utama pada *website* ini bisa dilihat pada Gambar 6. *User* memasukan *username* Instagram yang akan dideteksi akun tersebut.

Gambar 6. Tampilan Utama *Website*

Setelah *user* memasukan *username* yang akan dideteksi maka *user* bisa menekan tombol *check* pada *website* untuk melakukan eksekusi program. Program akan dijalankan dan *user* akan ditujukan kehalaman *resultpage* seperti pada Gambar 8 untuk melihat hasil deteksi. Apabila hasil prediksi dari *machine learning fake account*, maka akan muncul peringatan seperti pada Gambar 7.

Gambar 7. Peringatan *Fake Account*

Pada Gambar 8 *user* bisa melihat hasil deteksi dan isi detail – detail dari akun tersebut. Meliputi foto profil, tampilan bio, *username*, *fullname*, jumlah postingan, jumlah *followers*, jumlah *followings* dan lain - lain serta hasil prediksi dari *machine learning*. Apabila *user* ingin mendeteksi akun lagi, dapat dilanjutkan dengan menekan tombol *homepage* pada *button* untuk kembali ke halaman utama.



Gambar 8. Tampilan Hasil Deteksi

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut. Dari keempat metode yang diujikan, bahwa metode yang memiliki performa yang paling sesuai atau bagus. Diurutan pertama mulai dari *AdaBoost*, *Random Forest*, *SVM* dan *Naïve Bayes*. Hasil performa dari metode *AdaBoost* akurasi didapatkan sebesar 92.5%, *precision* didapatkan sebesar 93.9%, *recall* didapatkan sebesar 90.5% dan *f1-score* didapatkan sebesar 92.1%. Dari keempat metode yang diujikan, bahwa penggunaan metode *ensemble learning* seperti *Random Forest* dan *AdaBoost* lebih unggul daripada metode *single classifier* seperti *SVM* dan *Naïve Bayes*. Berdasarkan hasil pengujian *overfitting* pada tiap metode, baik menggunakan 10-*fold cross validation* maupun tidak menggunakan 10-*fold cross validation* keempat metode ini tidak mengalami *overfitting* pada data ini. Perbedaannya hasilnya tidak terlalu jauh berbeda. Namun hasil dari penggunaan 10-*Fold Cross Validation* menunjukan performa keempat metode yang diujikan naik sebesar 1%. Berdasarkan pengujian *feature importance* hasil klasifikasi *AdaBoost*. Telah ditemukan kriteria penentu *fake account* pada Instagram. Kriteria

tersebut adalah *description length* atau jumlah karakter pada *description*, *#post* atau jumlah postingan, rasio *followers* terhadap *followings* dan #*followers* yaitu jumlah *followers*.

5.2. Saran

Saran yang dapat digunakan untuk pengembangan dan penelitian lebih lanjut diantaranya sebagai berikut: Pengembangan implementasi sistem dalam bentuk lain selain *website*. Menggunakan metode *machine learning* lainya yang berbeda.

6. DAFTAR PUSTAKA

[1] Albayati, M., & Altamimi, A. (2019). Identifying Fake Facebook Profiles Using Data Mining Techniques. *Journal Of ICT Research And Applications*, *13*(2), 107-117. https://doi.org/10.5614/itbj.ict.res.appl.2019.13.2.2

[2] Bakhshandeh, B. (2019). *Instagram fake spammer genuine accounts*. Kaggle.com. Retrieved 3 January 2022, from https://www.kaggle.com/datasets/free4ever1/instagram-fake-spammer-genuine-accounts?select=train.csv.

[3] Berrar, D. (2019). Cross-Validation. *Encyclopedia Of Bioinformatics And Computational Biology*, *1*, 542-545. https://doi.org/10.1016/b978-0-12-809633-8.20349-x

[4] Boerman, S. (2020). The effects of the standardized instagram disclosure for micro- and meso-influencers. *Computers In Human Behavior*, *103*, 199-207. https://doi.org/10.1016/j.chb.2019.09.015

[5] Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/10.1023/a:1010933404324

[6] Jiang, X., Li, Q., Ma, Z., Dong, M., Wu, J., & Guo, D. (2018). QuickSquad: A new single-machine graph computing framework for detecting fake accounts in large-scale social networks. *Peer-To-Peer Networking And Applications*, *12*(5), 1385-1402. https://doi.org/10.1007/s12083-018-0697-2

[7] Hastie, T., Rosset, S., Zhu, J., & Zou, H. (2009). Multi-class AdaBoost. *Statistics And Its Interface*, *2*(3), 349-360. https://doi.org/10.4310/sii.2009.v2.n3.a8

[8] Kumar, A. (2020). *The Ultimate Guide to AdaBoost Algorithm | What is AdaBoost Algorithm?*. GreatLearning Blog: Free Resources what Matters to shape your Career!. Retrieved 7 January 2022, from https://www.mygreatlearning.com/blog/adaboost-algorithm/#How%20Does%20AdaBoost%20Work?.

[9] *Most used social media 2021 | Statista*. Statista. (2022). Retrieved 11 December 2021, from https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

[10] Narkhede, S. (2018). *Understanding Confusion Matrix*. Medium. Retrieved 5 January 2022, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[11] Pradana, G. (2021). *Web Scraping Pengertian, Teknik, Manfaat dan Kendala adalah*. Ngalup Collaborative Network. Retrieved 5 January 2022, from https://ngalup.co/articles/pengertian-teknik-manfaat-kendala-web-scraping/.

[12] Purba, K., Asirvatham, D., & Murugesan, R. (2020). Classification of instagram fake users using supervised machine learning algorithms. *International Journal Of Electrical And Computer Engineering (IJECE)*, *10*(3), 2763. https://doi.org/10.11591/ijece.v10i3.pp2763-2772

[13] Ramalingam, D., & Chinnaiah, V. (2018). Fake profile detection techniques in large-scale online social networks: A comprehensive review. *Computers & Electrical Engineering*, *65*, 165-177. https://doi.org/10.1016/j.compeleceng.2017.05.020

[14] Reddy, V. (2018). *Sentiment Analysis using SVM*. Medium. Retrieved 3 January 2022, from https://medium.com/@vasista/sentiment-analysis-using-svm-338d418e3ff1.

[15] Ruslidiantoro, A. (2021). *Overfitting dan Underfitting*. Medium. Retrieved 31 April 2022, from https://ariprusli.medium.com/overfitting-dan-underfitting-7f9e686aa97d.

[16] Pamungkas, R, I., & Lailiyah, N. (2019). PRESENTASI DIRI PEMILIK DUA AKUN INSTAGRAM DI AKUN UTAMA DAN AKUN ALTER. Interaksi Online, 7(4), 371-376. Retrieved from https://ejournal3.undip.ac.id/index.php/interaksi-online/article/view/24960

[17] Rish, I. (2001). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).

[18] Shaikh, S. (2021). *GitHub - shaikhsajid1111/social-media-profile-scrapers: Fetch user's data across social media*. GitHub. Retrieved 1 March 2022, from https://github.com/shaikhsajid1111/social-media-profile-scrapers.

[19] Sheikhi, S. (2020). An Efficient Method for Detection of Fake Accounts on the Instagram Platform. *Revue D'intelligence Artificielle*, *34*(4), 429-436. https://doi.org/10.18280/ria.340407

[20] Shin, T. (2021). *Understanding Feature Importance and How to Implement it in Python*. Medium. Retrieved 11 May 2022, from https://towardsdatascience.com/understanding-feature-importance-and-how-to-implement-it-in-python-ff0287b20285#:~:text=Feature%20Importance%20refers%20to%20techniques,to%20predict%20a%20certain%20variable.

[21] Sutter, B., Chiong, R., Budhi, G., & Dhakal, S. (2021). Predicting Psychological Distress from Ecological Factors: A Machine Learning Approach. *Advances And Trends In Artificial Intelligence. Artificial Intelligence Practices*, 341-352. https://doi.org/10.1007/978-3-030-79457-6_30

[22] Twin, A. (2021). *How Overfitting Works*. Investopedia. Retrieved May 14, 2021, from https://www.investopedia.com/terms/o/overfitting.asp.

[23] Wanda, P., Hiswati, M., Diqi, M., & Herlinda, R. (2021). Re-Fake: Klasifikasi Akun Palsu di Sosial Media Online menggunakan Algoritma RNN. *Prosiding Seminar Nasional Sains Teknologi Dan Inovasi Indonesia (SENASTINDO)*, *3*, 191-200. https://doi.org/10.54706/senastindo.v3.2021.139

[24] Yiu, T. (2019). *Understanding Random Forest*. Retrieved May 16, 2021, from https://towardsdatascience.com/understanding-random-forest-58381e0602d2.