

Implementasi Locally Adaptive K-Nearest Neighbor Algorithm based on Discrimination Class (DC-LAKNN) pada Kasus Deteksi Fake Account Instagram

Yosefani Kurniawan, Lily Puspa Dewi, Silvia Rostianingsih
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121- 131 Surabaya, 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-mail : yosefanik@gmail.com, lily@petra.ac.id, silvia@petra.ac.id

ABSTRAK

Instagram merupakan salah satu media sosial yang memiliki banyak pengguna. Karena kemudahan dalam membuat akun, banyak orang membuat akun palsu untuk *stalk*, percobaan *spam*, modus penipuan, pencurian foto, password bahkan serangan virus. Oleh karena itu, pengguna perlu waspada terhadap *follower* yang tidak dikenal. Dengan mendeteksi apakah akun tersebut asli atau palsu dapat membantu pengguna untuk berhati-hati menerima *follower* yang tidak dikenal. Selain itu pengguna dapat melaporkan pada pihak instagram sehingga akun tersebut dapat dinonaktifkan.

Pada skripsi ini, dirancang suatu aplikasi berbasis *website* yang dapat mendeteksi kemungkinan suatu akun Instagram merupakan akun asli atau palsu. Deteksi tersebut dilakukan dengan menggunakan metode klasifikasi *Locally Adaptive K-Nearest Neighbor algorithm based on Discrimination Class* (DC-LAKNN) yang merupakan algoritma adaptif dari algoritma *K-Nearest Neighbor*. Algoritma ini memperhatikan kelas diskriminasi sebagai dasar untuk klasifikasi. Atribut yang digunakan dalam klasifikasi berupa *user follower count*, *following count*, *biography length*, *media count*, *username digit count*, *username length*, *user has profile picture*, *user is private* yang didapat dari akun yang akan dideteksi. Hasil akhirnya adalah dengan algoritma *Locally Adaptive K-Nearest Neighbor algorithm based on Discrimination Class* (DC-LAKNN) dapat digunakan untuk klasifikasi akun Instagram dengan hasil akurasi yang didapatkan sebanyak 96.23%.

Kata Kunci: DC-LAKNN, KNN, deteksi akun Instagram, kelas diskriminasi, klasifikasi

ABSTRACT

Instagram is one of the social media that has many users. Because of the ease of creating an account, many people create a fake account for stalking, spam attempts, fraud, photo or password theft, and even attacks another account with virus. Therefore, users need to be wary of unknown followers. Detecting account, which is real or fake can help users to be careful accepting some unknown follower. In addition, users can report to Instagram so that account can be deactivated.

In this thesis, a website-based application is designed that can detect the possibility of an Instagram account being a real or fake account. The detection is carried out using the Locally Adaptive K-Nearest Neighbor algorithm classification method based on Discrimination Class (DC-LAKNN) which is an adaptive algorithm from the K-Nearest Neighbor algorithm. This algorithm pay attention at discrimination class as the basis for classification. The attributes used in the classification are user follower count, following count, biography length, media count, username digit

count, username length, user has profile picture, user is private. The end result is that the Locally Adaptive K-Nearest Neighbor algorithm based on Discrimination Class (DC-LAKNN) can be used to classify Instagram accounts with an accuracy of 96.23%.

Keywords: DC-LAKNN, KNN, account Instagram detection, discrimination class, classification.

1. PENDAHULUAN

Pada era digital sekarang ini, media sosial sangat populer di seluruh dunia, salah satunya adalah Instagram. Tidak sedikit orang yang membuat akun palsu yang dapat membahayakan karena ada kemungkinan untuk mencuri foto, *password* akun pengguna dan bahkan dapat menyerang dengan virus [3]. Sehingga perlu diwaspadai terhadap *follower* yang tidak dikenal, terlebih akun yang dimiliki tidak di-*private*.

Dengan mendeteksi suatu akun yang dicurigai sebagai akun palsu maka dapat membantu pengguna untuk berhati-hati. Selain itu, dapat melaporkan (*report*) pada pihak Instagram sehingga dapat dilakukan tindakan untuk menonaktifkan akun tersebut (Dewi, 2020). Deteksi tersebut dapat dilakukan dengan menggunakan metode klasifikasi *Locally Adaptive K-Nearest Neighbor algorithm based on Discrimination Class* (DC-LAKNN) yang merupakan algoritma adaptif dari algoritma *K-Nearest Neighbor*.

K-Nearest Neighbor (KNN) merupakan suatu metode untuk mengklasifikasikan sekumpulan data dan menggunakan algoritma *supervised* [4], dimana data yang diklasifikasikan berdasarkan mayoritas kedekatan jarak dengan data yang lain. Namun kelemahan KNN, waktu untuk komputasi yang cukup tinggi apabila data latih besar dan sangat *sensitive* dengan ciri yang redundan. Selain itu, KNN juga hanya menentukan klasifikasi berdasarkan *neighbor* mayoritas pertama dan mengabaikan kelas mayoritas kedua. Padahal jika spasial distribusi tetangga kelas mayoritas kedua lebih dekat terkait dengan *query* maka sebetulnya kelas mayoritas kedua dapat lebih diandalkan. Sehingga digunakanlah algoritma DC-LAKNN yang mempertimbangkan peran kelas mayoritas kedua atau dengan kata lain memilih data tetangga dekat yang optimal untuk klasifikasi akhir. Dengan memperhatikan kelas diskriminasi dapat meningkatkan kinerja klasifikasi [5].

Pada penelitian sebelumnya yang berjudul *a new Locally Adaptive K-Nearest Neighbor algorithm based on Discrimination Class* (DC-LAKNN) berfokus untuk memperkenalkan dan membandingkan algoritma tersebut dengan beberapa algoritma KNN yang lain. Untuk membandingkan, penelitian tersebut menggunakan 18 *dataset* dunia nyata yang kemudian dihitung

akurasi masing-masing algoritma. Jika ditinjau dari *dataset* yang digunakan untuk membandingkan masing-masing algoritma merupakan data numerik, maka dimungkinkan untuk melakukan implementasi algoritma DC-LAKNN pada klasifikasi suatu kasus yang memiliki kemiripan jenis data. Salah satu kasus yaitu penelitian pada akun media sosial seperti Instagram, karena selain memiliki dataset numerik, klasifikasi akun Instagram dengan DC-LAKNN akan memperhatikan kelas diskriminasi sehingga membuat performa yang lebih baik karena juga mempertimbangkan mayoritas kelas kedua. Sehingga kasus ini merupakan salah satu permasalahan yang dapat diambil dengan implementasi algoritma tersebut, sekaligus ingin menguji algoritma adaptif baru yaitu *Locally Adaptive K-Nearest Neighbor algorithm based on Discrimination Class* (DC-LAKNN). Akurasi dari algoritma DC-LAKNN nanti juga akan dibandingkan dengan penelitian sebelumnya yang berjudul *Instagram Fake and Automated Account Detection* [1] yang juga mendeteksi akun palsu Instagram dengan menggunakan algoritma *Support Vector Machine, Naïve Bayes, Logistic Regression, Neural Network*. Hasil tertinggi akurasi diperoleh dari algoritma *SVM dan Neural Network* sebesar 94%.

2. DASAR TEORI

2.1 Tinjauan Studi

Pada penelitian sebelumnya oleh Fatih Cagatay Akyon dan Esat Kalfaoglu[1] tentang deteksi akun palsu dan akun otomatis Instagram membuat bisnis tidak efisien, karena sistem rekomendasi tidak bekerja dengan efisien, salah menargetkan pengguna dalam membuat iklan, akses kualitas layanan dan produk yang lebih sulit serta menciptakan lingkungan media sosial yang tidak sehat. Penelitian tersebut menggunakan metode Penelitian terkait mengumpulkan dataset dan menyajikan analisis rinci deteksi akun palsu dan otomatis menggunakan SMOTE-NC. Sedangkan untuk klasifikasi digunakan algoritma *machine learning*, yaitu *Support Vector Machine, Naïve Bayes, Logistic Regression, Neural Network*. Hasilnya adalah *dataset* yang dapat digunakan untuk mendeteksi akun palsu dan akun otomatis. Selain itu, penelitian terkait menghasilkan akurasi dari masing-masing algoritma dalam deteksi akun palsu dimana SVM dan *Neural Network* mencapai skor F1 tertinggi sebesar 94%. Perbedaan penelitian sebelumnya dengan penelitian ini adalah penggunaan metode sebagai teknik klasifikasi. Penelitian terkait menggunakan metode *Support Vector Machine, Naïve Bayes, Logistic Regression, Neural Network* untuk klasifikasi dan membandingkan mana algoritma yang menghasilkan akurasi tertinggi, sedangkan penelitian ini menggunakan algoritma DC-LAKNN sebagai metode klasifikasi.

2.2 Instagram

Menurut Ronal Sukwadi, dkk Instagram adalah suatu aplikasi untuk membagikan foto, dimana pengguna dapat mengambil foto dan video serta menerapkan *filter* digital dan membagikannya ke berbagai layanan media sosial. Selain itu, pengguna juga dapat saling berkomentar, *live streaming*, memberi *like* serta membagikan unggahan pengguna lain [7]. Dengan adanya fitur seperti ini, Instagram dapat digunakan sebagai sarana untuk berinteraksi dengan orang lain ataupun untuk mengembangkan bisnis *online*.

2.3 Klasifikasi

Klasifikasi adalah suatu model untuk membedakan dan menggambarkan kelas data atau konsep untuk memprediksi kelas dari objek yang masih tidak diketahui label kelasnya. Algoritma klasifikasi contohnya adalah *K-Nearest Neighbor, Rule Based, Naïve bayes, Decision trees, Neural Network, Support Vector*

Machines (SVM), Rough sets, analisa statistik. Terdapat 2 proses dalam klasifikasi, pertama adalah fase *training* yaitu algoritma klasifikasi dibuat untuk menganalisa suatu data lalu direpresentasikan dalam bentuk *rule* klasifikasi. Proses yang kedua adalah klasifikasi, dimana data *test* digunakan untuk memperkirakan akurasi dari *rule* klasifikasi [2].

2.4 K-Nearest Neighbor

Klasifikasi *K-Nearest Neighbor* adalah pengklasifikasi yang telah digunakan sebagai pengklasifikasian dasar dalam banyak permasalahan. Hal ini didasarkan pada pengukuran jarak antara data pengujian dan setiap data *training* untuk menentukan hasil klasifikasi akhir (Hu et al., 2016). Pengklasifikasian pada objek berdasarkan *k* buah data latih yang jaraknya paling dekat dengan objek tersebut. Dengan syarat nilai *k* tidak boleh lebih besar dari jumlah data *training* dan bernilai ganjil serta lebih dari satu. Jarak tersebut dapat dihitung dengan menggunakan metode *cosine similarity* (J et all., 2012). Namun secara umum, dapat menggunakan persamaan *Euclidean Distance* untuk mendefinisikan jarak antara 2 objek *x* dan *y*. Untuk menghitung jarak pada algoritma *K-Nearest Neighbor* dapat dilihat pada persamaan 2.1.

$$d_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

2.5 Min-Max normalization

Min-Max normalization merupakan salah satu metode normalisasi, dimana metode ini dilakukan dengan standarisasi suatu *value*. Standarisasi tersebut diterapkan pada data dengan mengubah nilai dalam *range* 0 hingga 1. Nilai 0 merupakan nilai terkecil dan 1 merupakan nilai terbesar. Dengan rentang nilai yang tidak jauh seperti ini, membuat data dapat diolah dengan lebih mudah dan efektif. Adapun rumus perhitungan metode *min-max normalization* adalah sebagai berikut [8]:

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{range.max} - \text{range.min}) + \text{range.min} \quad (2)$$

2.6 New Locally Adaptive K-Nearest Neighbor algorithm based on Discrimination Class (DC-LAKNN)

Algoritma DC-LAKNN merupakan algoritma yang dibuat untuk mengatasi permasalahan pada algoritma KNN yaitu pengambilan langsung label kelas mayoritas sebagai hasil keputusan klasifikasi dan irasionalitas penggunaan *single value* dan *fixed value* untuk semua *query*. Algoritma DC-LAKNN mempertimbangkan peran kelas mayoritas kedua dalam klasifikasi dengan memperkenalkan konsep kelas diskriminasi berdasarkan informasi yang diterima. Dengan kata lain, algoritma DC-LAKNN memilih nilai objek paling optimal yang dekat dengan *query* untuk klasifikasi akhir. Berikut langkah-langkah untuk menentukan kelas diskriminasi pada *query* tertentu [5]:

1. Memilih *k-nearest neighbors* pada data *training* berdasarkan urutan *ascending* dari *Euclidean distance* antara *query* dan semua *dataset training*.
2. Menentukan kelas mayoritas pertama dan kelas mayoritas kedua.
3. Menghitung *centroid* dari tetangga terdekat pada kelas mayoritas pertama dan kedua yang berbeda.

4. Menentukan kelas diskriminasi dari setiap kelas mayoritas.
5. Mencatat jumlah dan *centroid* tetangga terdekat dalam kelas diskriminasi.

2.7 Confusion Matrix

Metode *Confusion Matrix* digunakan jika *dataset* hanya terdiri dari 2 kelas, yaitu kelas positif dan negatif [2] yang dapat dilihat pada Tabel 1.

Tabel 1. Confusion Matrix

Kelas yang benar	Kelas hasil prediksi	
	+	-
+	<i>True positive (TP)</i>	<i>False negatives (FN)</i>
-	<i>False positive (FP)</i>	<i>True negatives (TN)</i>

Setelah mengetahui 2 tipe *error* dalam *confusion matrix* yaitu *False negatives* dan *False Positive* maka akan dilanjutkan dengan menghitung *Precision*, *Recall* dan *F1-Score* yang secara rumus dituliskan seperti:

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

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{precision} + \frac{1}{recall} \right) \quad (5)$$

Precision adalah perbandingan antara data dengan nilai *True Positive* dengan jumlah data yang diprediksi positif. Sedangkan *Recall* adalah perbandingan antara data dengan nilai *True Positive* dengan banyaknya data yang sebenarnya positif. Untuk mengetahui indikasi apakah model klasifikasi *precision* dan *recall* yang baik dapat menggunakan penilaian *F1-Score* dengan nilai minimal 0 dan nilai maksimal 1. Semakin tinggi nilai *F1-Score* maka model klasifikasi yang dilakukan mempunyai nilai *precision* dan *recall* yang baik [6].

3. DESAIN SISTEM

3.1 Analisa Permasalahan

Akun Instagram sangat mudah dibuat karena hanya memerlukan *email* atau nomor *handphone* dan melengkapi data seperti nama lengkap, *username* dan *password*. Kemudahan dalam membuat akun tersebut membuat orang yang tidak bertanggung jawab menyalahgunakan akun tersebut dengan membuat akun palsu. Akun palsu tersebut digunakan untuk menipu atau sekedar untuk *stalk* aktivitas pengguna lain. Kemungkinan modus kejahatan yang dapat terjadi adalah pencurian foto, *password* akun pengguna sehingga perlu waspada terhadap *follower* yang tidak dikenal. Modus kejahatan lain diungkapkan oleh Justin Zadorsky, seorang penulis berita pada *website CTV News London* bahwa pengguna Instagram perlu waspada terhadap akun palsu. Hal ini disebabkan karena telah terjadi kasus akun palsu yang meminta para remaja untuk mengirimkan foto eksplisit dan telah berinteraksi dengan lebih dari 300 akun lain[9]. Dengan mendeteksi kemungkinan akun palsu dapat membuat pengguna Instagram lebih berhati-hati dan dapat melaporkan akun palsu tersebut pada Instagram sehingga akun tersebut dapat dihapus. Deteksi *fake account* dapat dilakukan dengan menggunakan algoritma *Locally Adaptive K-Nearest Neighbor algorithm based on Discrimination Class (DC-LAKNN)* sebagai metode klasifikasinya.

3.2 Analisa Data

Analisa data membahas tentang pengumpulan data (*pengambilan dataset*) dan pengolahan data. Selain itu, analisis sistem yang dilakukan meliputi analisa klasifikasi DC-LAKNN dan desain sistem secara keseluruhan. Analisa data diperlukan untuk memperoleh informasi dari *dataset* yang akan diproses terlebih dahulu sebelum digunakan.

3.2.1 Pengambilan Dataset

Dataset berupa akun Instagram yang diambil dari *paper* yang berjudul *Instagram Fake and Automated Account Detection* yang ditulis oleh Fatih Cagatay Akyon dan Esat Kalfaoglu [1]. *Dataset* yang disediakan oleh penulis terdiri atas *dataset fake account* dan *automated account* yang masing-masing sudah diberi label (*fake* atau tidak). Namun yang dipakai pada penelitian ini hanya *dataset fake account* saja. *Dataset* tersebut berjumlah 994 data *real account*, dan 200 data *fake account*. Data tersebut dikumpulkan oleh penulis secara *manual labeling* dan menerapkan implementasi algoritma SMOTE-NC sebagai teknik *oversampling*-nya.

3.2.2 Data Preprocessing

Data *preprocessing* dalam skripsi ini diperlukan karena *dataset* yang digunakan memiliki value dengan rentang yang cukup jauh. Selain itu, *dataset* yang digunakan perlu dipisah menjadi 2 *subset* yaitu data *training* dan data *testing*. Oleh karena itu sebelum digunakan, *dataset* perlu diolah terlebih dahulu. Proses yang dilakukan dalam sub bab ini yaitu data *normalization* dan data *splitting*.

1. Data Normalization

Normalisasi pada *dataset* ini diperlukan karena terdapat rentang nilai yang cukup jauh pada value atribut yang ada. Contohnya pada kolom *userFollowingCount*, terdapat value tertinggi dengan jumlah 7497 dan value terendah adalah 0. Karena rentang yang jauh tersebut diperlukan normalisasi dengan *min-max normalization*. Dengan melakukan normalisasi, klasifikasi akan lebih efisien karena data sudah diolah lebih baik.

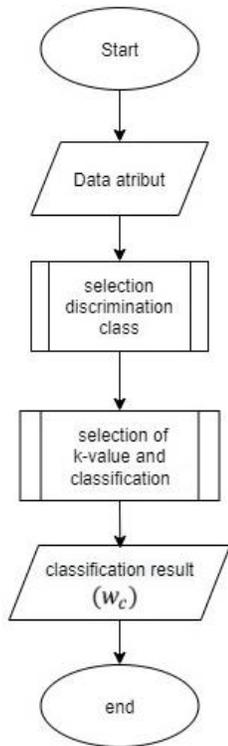
2. Data Splitting

Pada bagian ini dilakukan pemisahan *dataset* menjadi 2 yaitu data *training* dan data *testing*. Data *training* digunakan untuk melatih *dataset* yang telah diterapkan algoritma sehingga data *training* tersebut dapat membuat suatu prediksi dari klasifikasi algoritma tersebut. Data *training* yang digunakan sejumlah 1194 data, yang dibagi atas 2 data yaitu sejumlah 795 data *real account* dan 160 data *fake account*. Sedangkan data *testing* digunakan untuk menguji performa algoritma yang telah dilatih dalam data *testing* sebelumnya pada data baru. Data *testing* yang digunakan sejumlah 20% dari masing-masing data yaitu sejumlah 199 data *real account* dan 40 data *fake account* yang dipilih secara acak.

3.3 Algoritma Klasifikasi DC-LAKNN

DC-LAKNN merupakan algoritma yang memperhatikan kelas diskriminasi dalam melakukan klasifikasi. Algoritma ini dibuat untuk mengatasi permasalahan pada algoritma KNN, dimana hasil klasifikasi diambil secara langsung dari label kelas mayoritas untuk semua *query*. Algoritma DC-LAKNN dalam klasifikasinya mempertimbangkan peran kelas mayoritas kedua berdasarkan informasi yang diterima sebelumnya. DC-LAKNN dapat terdiri dari 2 *stage* yang harus dilakukan, yaitu seleksi kelas diskriminasi dan *k value* dan prosedur klasifikasi. Alur program dalam klasifikasi DC-LAKNN adalah melakukan *input* data berupa atribut klasifikasi, diantaranya adalah *user follower count*, *user following count*, *user biography*, *user media count*, *user has*

private, *username digit count* dan *username length*. Setelah memasukkan atribut, data tersebut akan diolah pada proses seleksi kelas diskriminasi dimana dalam tahapan ini data tersebut dihitung jarak terdekat dengan kelas klasifikasi. Sehingga diperoleh hasil klasifikasi sesuai dengan jarak terdekat dengan kelas yang juga memperhatikan kelas diskriminasi. Untuk langkah-langkah secara garis besar program dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Program Keseluruhan

3.3.1 Selection Discrimination Class

Dalam klasifikasi KNN, pertimbangan klasifikasi yang dilakukan adalah menggunakan kelas utama (*majority class*) dan mengabaikan kelas lainnya (*discrimination class*). Padahal terdapat kemungkinan dimana kelas diskriminasi tersebut memiliki jarak yang lebih dekat dengan *query*, walaupun memiliki jumlah tetangga yang lebih sedikit dibandingkan dengan kelas utama. Oleh karena itu, jarak kelompok kelas diskriminasi yang lebih dekat justru cenderung memiliki kemiripan sehingga bisa saja *query* tersebut masuk kelas yang sama dengan kelas diskriminasi. Pada proses pemilihan kelas diskriminasi memerlukan *input* yaitu *query*, data *training*, dan nilai *kmax*. Kelas mayoritas yang digunakan sebagai pembanding adalah kelas mayoritas kedua dengan jumlah terbesar dari tetangga terdekat dalam lingkungan suatu *query*. Jadi, tidak hanya jumlah tetangga terdekat yang diperhatikan namun juga jarak distribusi suatu *query*.

Berdasarkan algoritma DC-LAKNN ini, langkah pertama yang dilakukan pada proses ini adalah menentukan *k-nearest neighbors* pada data *training* dengan memperhitungkan *euclidean distances* antara data input (*query*) dengan data *training*. Selama nilai *k* kurang dari nilai *kmax* maka proses selanjutnya adalah menentukan kelas mayoritas pertama dan kedua. Masing-masing kelas ditentukan berdasarkan jumlah tetangga terdekat, kelas mayoritas pertama diambil dari jumlah kelas tetangga terdekat yang paling banyak. Setelah menentukan kelas mayoritas pertama dan kedua maka langkah selanjutnya adalah menentukan *centroid* atau titik tengah dari masing-masing kelas. Kelas diskriminasi diperoleh dari

jarak *input* dengan titik data *training*, misalnya apabila titik data *input* lebih dekat dengan kelas mayoritas namun jumlah tetangga lebih sedikit dibandingkan dengan kelas mayoritas kedua maka kelas diskriminasi adalah kelas mayoritas kedua tersebut. Walaupun jarak kelas mayoritas kedua lebih jauh dari titik *input* namun jumlah kelas mayoritas kedua lebih banyak dibandingkan dengan jumlah kelas mayoritas pertama sehingga kemiripan data *input* dianggap sesuai dengan kelas diskriminasi tersebut. Nilai kelas diskriminasi itulah yang digunakan sebagai pertimbangan klasifikasi selain mempertimbangkan jarak kelas dengan titik *input*.

3.3.2 Selection of K Value and Classification

Procedure

Setelah menyeleksi kelas diskriminasi maka langkah selanjutnya adalah menyeleksi nilai *k*. Hal ini dilakukan untuk mengatasi batasan pada nilai *k* yang diperoleh dari distribusi informasi tentang nilai tetangga terdekat pada kelas diskriminasi [5]. Dengan kata lain, proses ini dilakukan untuk mencari nilai *k* optimal dengan akurasi tertinggi dari model klasifikasi.

4. ANALISA DAN PENGUJIAN SISTEM

4.1 Perangkat Lunak yang Digunakan

Perangkat lunak yang digunakan dalam proses pembuatan algoritma dan website pada skripsi ini adalah Jupyter Notebook dan SublimeText3. Jupyter digunakan untuk membuat *coding* algoritma, pengolahan data *training* dengan menggunakan beberapa *library* seperti *Pandas library* dan *Numpy Library*. SublimeText3 digunakan untuk konfigurasi atau edit *user interface* dan menghubungkan *website* dengan algoritma klasifikasi yang telah dibuat di *python*.

4.2 Pengujian Aplikasi

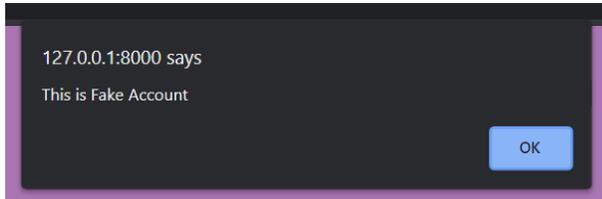
Pada saat pertama kali program dijalankan akan menampilkan *form input* yang digunakan *user* untuk memasukkan data atau parameter terkait untuk klasifikasi. Tampilan utama dari aplikasi berbasis website ini adalah seperti pada Gambar 2.

The screenshot shows a web form titled "Instagram Account Detection" with the following fields and options:

- User Follower Count: 0
- User Following Count: 726
- User Biography Length: 0
- User Media Count: 0
- Username Digit Count: 4
- Username Length: 5
- User Has Profile Picture?: Yes No
- User Is Private?: Yes No
- Check Account button

Gambar 2. Tampilan Utama Website

User memasukkan parameter seperti *user follower count*, *following count*, *biography length*, *media count*, *username digit count*, *username length*, *user has profile picture*, *user is private*. Setelah selesai memasukkan semua parameter maka *user* dapat menekan tombol *Check Account* untuk melihat apakah akun yang dimasukkan merupakan akun palsu atau asli. Setelah menekan tombol *Check Account*, *user* akan ditunjukkan tampilan seperti pada Gambar 3 yang menampilkan hasil dari klasifikasi atribut akun Instagram yang dimasukkan. Hasil yang diperoleh *user* adalah klasifikasi apakah akun Instagram tersebut merupakan *fake account* atau *real account*. Setelah *user* mengetahui klasifikasi akun tersebut dapat dilanjutkan dengan menekan tombol *Ok* untuk keluar dari tampilan pada Gambar 3 dan kembali ke tampilan utama.



Gambar 3. Tampilan Hasil Deteksi

4.3 Hasil Pengujian

Pengujian pertama dilakukan dengan membagi jumlah data *training* dan data *testing* dalam sebuah perbandingan. Dalam penelitian ini dilakukan 4 kali pengujian dengan pembagian data *training* dan data *testing* sebanyak 80%:20%, 75%:25%, 70%:30%, 60%:40%. Hal ini dilakukan untuk melihat bagaimana hasil perhitungan akurasi apabila jumlah data *training* dan data *testing* dikelompokkan dengan ukuran yang berbeda-beda. Dari 4 kali perhitungan tersebut dilihat berapa perbandingan yang menghasilkan akurasi yang paling tinggi.

Tabel 2. Perhitungan Akurasi Algoritma DC-LAKNN (80%:20%)

K	TP	TN	FP	FN	Accu racy	Recall	Precisi on	F1-Score
1	37	187	12	3	0.937	0.925	0.75	0.831
2	37	189	10	3	0.955	0.925	0.787	0.851
3	36	194	5	4	0.962	0.9	0.878	0.888
4	36	192	7	4	0.952	0.9	0.837	0.867
5	37	193	6	3	0.962	0.925	0.860	0.891
6	35	192	7	5	0.949	0.875	0.833	0.853
7	36	191	8	4	0.949	0.9	0.818	0.857
8	37	192	7	3	0.958	0.925	0.841	0.881
9	37	192	7	3	0.958	0.925	0.841	0.881
10	37	191	8	3	0.954	0.925	0.822	0.870
11	37	191	8	3	0.954	0.925	0.822	0.870
12	37	191	8	3	0.954	0.925	0.822	0.870

Tabel 2 merupakan hasil perhitungan akurasi dengan menggunakan perbandingan data *training* dan data *testing* sebanyak 80%:20%. Dimana data *training* sejumlah 955 data dan data *testing* sejumlah 239 data. Dari hasil perhitungan dengan menggunakan *confusion matrix* dapat dilihat bahwa akurasi tertinggi diperoleh ketika nilai $k=3$ dan nilai $k=5$, dimana akurasi keduanya mencapai 96.2%. Dengan *F1-Score* yang paling tinggi diantara keduanya yaitu 89.1% saat nilai $k=5$. Dan nilai akurasi terendah diperoleh sebanyak 93.7% saat nilai $k=1$.

Tabel 3. Data Akun Instagram

Account	Attribute	Label
1	[0,469,0,0,0,1,0,10]	1
2	[0,14,0,0,0,1,4,17]	1
3	[0,731,0,0,0,1,4,5]	1
4	[149,148,30,0,0,1,0,7]	0
5	[99,251,146,4,1,0,2,7]	0
6	[7,159,0,0,0,0,3,13]	1
7	[22,25,0,1,1,0,4,7]	0
8	[733,609,38,11,1,1,0,10]	0
9	[24,126,5,6,1,1,0,10]	1
10	[591,564,0,6,1,1,3,10]	0
11	[6,107,0,0,1,1,5,7]	1
12	[4,20,78,1,1,0,0,9]	0
13	[133,178,119,19,1,0,3,15]	0
14	[225,399,30,79,1,0,0,8]	0
15	[4,26,21,198,1,1,0,16]	1
16	[638,835,0,73,1,1,6,11]	0
17	[15,29,0,0,1,0,0,7]	1
18	[1654,1107,31,23,1,1,2,9]	0
19	[6544,2888,137,115,1,0,0,8]	0
20	[1299,490,146,34,1,1,0,12]	0
21	[1400,1188,33,3,1,1,0,7]	0
22	[238,429,85,1,1,1,0,11]	0
23	[10,69,22,15,1,1,1,8]	1
24	[135,152,0,0,1,1,0,11]	0
25	[137,136,27,2,1,1,8,9]	0
26	[77,114,20,54,1,1,0,9]	0
27	[0,85,0,0,0,1,2,9]	1
28	[65,51,0,8,1,1,2,9]	0
29	[108,105,122,117,1,0,0,6]	0
30	[320,265,9,30,1,0,0,7]	0
31	[298,280,10,2,0,0,0,9]	0
32	[564,545,0,21,0,1,2,14]	0
33	[51,7160,7,2,1,0,0,10]	1
34	[562,502,0,11,1,0,0,16]	0
35	[0,3,0,0,0,0,0,9]	1

Pada Tabel 3, label 0 mengidentifikasi bahwa akun tersebut merupakan *real account* dan label 1 merupakan *fake account*. Dari data yang telah dikumpulkan, kemudian diprediksi apakah data tersebut dicurigai merupakan *real account* atau *fake account*. Untuk melihat hasil akurasi dari prediksi data akun yang telah dikumpulkan dapat dilihat pada Tabel 3.

Data yang terkumpul kemudian diprediksi dan dihitung berapa akurasi yang diperoleh dengan menggunakan *confusion matrix* dengan nilai $k=5$. Dari prediksi menghasilkan label prediksi yaitu [1,1,1,0,1,1,0,0,1,0,1,0,1,1,0,1,1,0,0,0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,1]. Hasil akurasi mencapai 84.84% dan *F1-Score* sebanyak 75.8%. Dari hasil prediksi jumlah *False Positive* sebanyak 6 yang

diperoleh dari akun yang merupakan *real account* namun dideteksi sebagai *fake account* yaitu data ke-5, data ke-13, data ke-14, data ke-15, data ke-22 dan data ke-26. Sedangkan jumlah *False Negative* sebanyak 1 diperoleh dari akun yang merupakan *fake account* namun dideteksi sebagai *real account* yaitu data ke-15.

Tabel 4. Perbandingan Akurasi dengan Metode lain

<i>Method</i>	<i>F1-Score</i>
Support Vector Machine	88.2%
Naive Bayes	54.2%
Logistic Regression	87.8%
Neural Network	87.8%
DC-LAKNN	89.1%

Penelitian sebelumnya yang berjudul *Instagram Fake and Automated Account Detection* juga mendeteksi akun palsu Instagram dengan menggunakan algoritma *Support Vector Machine, Naive Bayes, Logistic Regression, Neural Network* [1]. Dari akurasi penelitian tersebut dibandingkan dengan akurasi dari algoritma DC-LAKNN. Dari akurasi beberapa metode tersebut dapat dilihat bahwa nilai *F1-Score* tertinggi untuk klasifikasi *fake account dataset* adalah 89.1% dengan menggunakan metode DC-LAKNN. Untuk lebih detail dapat dilihat pada Tabel 4.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian data *training* dan *testing* dapat disimpulkan beberapa hal, yaitu:

- Akurasi metode DC-LAKNN bergantung kepada jumlah tetangga sekitar data yang dimasukkan. Pada skripsi ini akurasi terbaik ditemukan pada nilai *k* sebesar 5, dengan akurasi sebesar 96.23 % dan *F1-Score* sebesar 89%.
- Akurasi metode DC-LAKNN dapat dipengaruhi oleh jumlah perbandingan data *training* dan data *testing*. Perbandingan data *training* dan data *testing* yang menghasilkan akurasi deteksi tertinggi adalah pengelompokan data sebanyak 80%:20%.
- Dari hasil percobaan pengujian akun Instagram dari beberapa *sample* didapatkan hasil akurasi mencapai 84.84% dan *F1-Score* sebanyak 75.8% yang dianggap bahwa klasifikasi dengan menggunakan algoritma DC-LAKNN dalam mengidentifikasi suatu akun Instagram telah berhasil.

5.2 Saran

Saran yang dapat digunakan untuk pengembangan penelitian selanjutnya yaitu:

- Pengembangan aplikasi dalam bentuk lain selain *website* atau dengan metode yang berbeda.

- Sistem dapat langsung mengakses atribut klasifikasi dengan *input* berupa *username* atau *link* Instagram.

6. DAFTAR PUSTAKA

- [1] Akyon, F. C., & Esat Kalfaoglu, M. (2019). Instagram Fake and Automated Account Detection. In *Proceedings - 2019 Innovations in Intelligent Systems and Applications Conference, ASYU 2019*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ASYU48272.2019.8946437>
- [2] Leidiyana, H. (2013). PENERAPAN ALGORITMA K-NEAREST NEIGHBOR UNTUK PENENTUAN RESIKO KREDIT KEPEMILIKAN KENDARAAN BEMOTOR. *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, 1(1), 65-76. <https://media.neliti.com/media/publications/155541-ID-penerapan-algoritma-k-nearest-neighbor-u.pdf>
- [3] kumparan. (2018, 07 17). *Waspada, Punya Banyak Follower Palsu di Instagram Bisa Berbahaya*. kumparanTECH. Retrieved 12 4, 2020, from <https://kumparan.com/kumparantech/waspada-punya-banyak-follower-palsu-di-instagram-bisa-berbahaya-27431110790547314/full>
- [4] Mustakim, & F, G. O. (2016, Juni). Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa. *Jurnal Sains, Teknologi dan Industri*, 13(2), 195-202. <http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/1688>
- [5] Pan, Z., Wang, Y., & Pan, Y. (2020, 06 26). A new locally adaptive k-nearest neighbor algorithm based on discrimination class. *Knowledge-Based System*, 204. <https://doi.org/10.1016/j.knosys.2020.106185>
- [6] Setiawan, S. (2020, July 12). *Membicarakan Precision, Recall, dan F1-Score*. <https://stevkarta.medium.com/membicarakan-precision-recall-dan-f1-score-e96d81910354>
- [7] Sukwadi, R., Inderawati, M. W., & Indah, M. Y. (2016). Perilaku Konsumen dalam Pemilihan Online Shop Instagram. *Jurnal Metris*, 17(02), 123 – 132. <http://ojs.atmajaya.ac.id/index.php/metris/article/view/476>
- [8] T. T. Hanifa, S. Al-faraby, F. Informatika, and U. Telkom, “ Analisis Churn Prediction pada Data Pelanggan PT . Telekomunikasi dengan Logistic Regression dan Underbagging.” vol. 4, no. 2, pp. 3210–3225, 2017.
- [9] Zadorsky, J. (2020, December 10). *Fake Instagram account asking youth for explicit photos interacted with 300 accounts: police*. CTV NEWS. <https://london.ctvnews.ca/fake-instagram-account-asking-youth-for-explicit-photos-interacted-with-300-accounts-police-1.522527>