Factor Analysis Keberlanjutan Pengguna Jala Menggunakan Klasifikasi K-Means dan Algoritma Random Forest

by Jhon Doe

Submission date: 09-Jun-2021 05:35PM (UTC+0700)
Submission ID: 1603387183
File name: Menggunakan_Klasifikasi_K-Means_dan_Algoritma_Random_Forest.pdf (867.2K)
Word count: 3346
Character count: 22094

Factor Analysis Keberlanjutan Pengguna Jala Menggunakan Klasifikasi *K-Means* dan Algoritma *Random Forest*

Keberlanjutan Pengguna Sejak Satu Bulan Pertama Menggunakan Sistem Jala

Andri Wahyu Ahmad Ruslam
Jurusan Informatika, Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta
17523221@students.uii.ac.id

Erika Ramadhani
Jurusan Informatika, Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta
115230409@ uii.ac.id

***Abstract*—Sebagai perusahaan yang berorientasi pengguna, Jala sangat peduli terhadap kenyaman pengguna dalam menggunakan produk-produk Jala, salah satunya produk software. Dari hasil pengelolaan data internal, pertumbuhan pengguna dari hari ke hari terus meningkat, akan tetapi tingkat churn juga tinggi. Oleh karena itu, penelitian mengenai *Factor Analysis* keberlanjutan pengguna diinisiasi untuk mencari tahu faktor yang mempengaruhi keberlanjutan pengguna sejak satu bulan pertama. Teknik-teknik yang digunakan adalah preprocessing menggunakan pandas, klasifikasi dengan algoritma k-means, dan modelling menggunakan random forest. Hasilnya, intensitas penggunaan fitur pakan dan sampling pada saat satu bulan pertama pengguna menggunakan Jala berpengaruh besar pada keberlanjutan penggunaan Jala pada bulan-bulan berikutnya**

Keywords—Factor Analysis, Random Forest, K-Means, Preprocessing data, Pandas

I. PENDAHULUAN

Jala merupakan salah satu perusahaan yang bergerak di bidang akuakultur khususnya dalam pembudidayaan udang. Memiliki tujuan utama memudahkan para petambak, Jala menghadirkan produk-produk berbasis teknologi agar dapat membantu petambak dalam memahami kondisi tambaknya. Produk tersebut terbagi menjadi dua, yaitu produk *hardware* dan *software*. Teknologi yang digunakan pada produk *hardware* memanfaatkan IoT (*Internet of Things*) sehingga memungkinkan para petambak untuk mengetahui kondisi dan menjalankan proses tambak secara *real-time* dan otomatis. Salah satu produk *hardware* Jala adalah baruno yang berguna untuk mengukur kualitas dan kondisi air secara *real-time*. Selain *hardware*, Jala menyediakan produk-produk *software* baik berbasis *website* maupun *mobile*. Mulai dari membantu mencatat aktivitas kulturisasi, manajemen keuangan, informasi harga udang, hingga membuat laporan budidaya. Penggunaan *software*, terutama *mobile* dapat membantu mempermudah pengawasan kondisi tambak [1].

Banyaknya produk JALA berdampak pada tingginya jumlah petambak yang ingin menggunakannya. Hingga saat ini terdapat sekitar 9000 pengguna aktif JALA yang tersebar di enam negara. Angka yang tinggi ini tentu sangat membanggakan bagi perusahaan. Akan tetapi, setelah dilakukan riset pengguna ditemukan sebuah anomali. Banyaknya pengguna yang terdaftar di sistem Jala tidak berbanding lurus dengan jumlah siklus aktif yang tercatat di sistem. Hal ini mengindikasikan terdapat pengguna yang berhenti menggunakan produk Jala sejak beberapa periode waktu tertentu menggunakannya.

Mengacu pada data internal perusahaan sejak 2018-2020 terdapat 3945 tambak yang aktif menggunakan jala. Namun demikian, hanya 851 tambak yang masih teridentifikasi aktif menggunakan sistem Jala. Sementara sisanya sudah tidak melanjutkan penggunaannya lagi. Artinya terdapat sekitar 78% pengguna yang churn. Churn merupakan istilah yang digunakan untuk menunjukkan kondisi pengguna yang berhenti menggunakan layanan perusahaan dan berpindah kepada kompetitor [2].

Tinggi atau rendahnya *churn* akan sangat mempengaruhi dalam melakukan evaluasi efektivitas produk bahkan dapat menjadi dasar pelaksanaan teknik marketing ke depan. Ditambah lagi kondisi churn berpotensi menyebabkan kerugian perusahaan dikarenakan biaya yang sangat tinggi telah dikeluarkan untuk menjaga loyalitas pelanggan [2]. Oleh sebab itu, jala menginisiasi *project* riset keberlanjutan pengguna untuk menganalisis indikator utama yang menentukan keberlanjutan pengguna baru sejak satu bulan pertama menggunakan sistem jala. *Project* ini diberi nama *Factor Analysis* keberlanjutan pengguna.

Factor Analysis merupakan salah satu teknik dalam *data science* yang mengidentifikasi setiap fitur yang bersinggungan langsung dengan objek -dalam hal ini pengguna- dan menganalisis untuk dilihat intensitas pengaruhnya terhadap nilai *churn*. Terdapat dua data utama yang akan dianalisis, yaitu fitur pencatatan aktivitas kulturisasi dan penggunaan fitur media pada *website*. Media adalah salah satu fitur yang dibuat perusahaan untuk mempengaruhi pengguna agar dapat lanjut menggunakan jala terutama lanjut ke fitur pro (berbayar). Fitur ini berisi mengenai manfaat penggunaan jala, profil perusahaan, hingga capaian perusahaan. Harapannya dengan hasil dari inisiasi project factor analysis ini dapat dijadikan indikator awal dalam mencegah tingginya churn pengguna.

Pada publikasi ini akan menjelaskan mengenai hasil penerapan *factor analysis* keberlanjutan pengguna Jala sejak satu bulan pertama penggunaan produk. Dengan menggunakan teknik klasifikasi dan algoritma *random forest*, output yang dihasilkan akan berupa persentase nilai pengaruh setiap variabel.

II. LANDASAN TEORI

A. Ekstraksi Data

Ekstraksi data merupakan sebuah proses pengambilan data dari sumber penyimpanan / database dengan tujuan untuk menyediakan data agar dapat diolah. Data perlu diolah

menjadi informasi agar dapat mudah dipahami oleh para stakeholder yang membutuhkan informasi tersebut. Pada aplikasi perangkat lunak, data disimpan dan dikelola secara sistematis menggunakan database management system (DBMS). MySQL merupakan salah satu DBMS yang paling populer yang banyak digunakan oleh software engineer dunia. Hal ini karena DBMS bersifat open-source dan menggunakan konsep relational (tabel-kolom) sehingga mudah untuk dipahami dan dikembangkan [3].

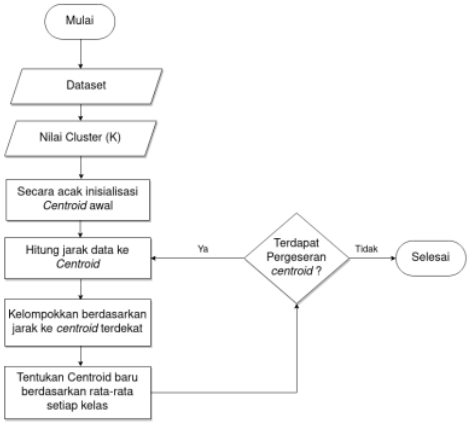
B. Data Preprocessing

Preprocessing adalah tahapan pengolahan data menjadi format tertentu dengan tujuan agar data dapat terstandarisasi dan memiliki struktur yang konsisten. Teknik ini perlu diterapkan karena pada dasarnya data yang kita miliki tidak terstruktur, banyak duplikasi data, dan inkonsisten [4]. Salah satu *tools* untuk melakukan *preprocessing* data pada python adalah Pandas. Pandas merupakan *library* python yang memiliki kemampuan pengolahan data dalam bentuk tabel (baris-kolom) dan kalkulasi statistik. Pandas dibuat untuk memenuhi kebutuhan akademis dan industri *data science* dalam *preprocessing* data menggunakan python [4].

C. Klasifikasi Data

Klasifikasi dalam *data science* merupakan suatu proses pengelompokkan data berdasarkan kriteria tertentu baik yang terlihat secara eksplisit maupun implisit [5]. Dalam proses klasifikasi, algoritma akan melakukan pencarian pola dari suatu data kemudian akan mengelompokkannya berdasarkan pola tersebut. Salah satu algoritma yang terkenal pada klasifikasi data adalah algoritma *K-Means*. *K-means* adalah algoritma yang mengelompokkan data numerik, bersifat unsupervised, dan terus melakukan perulangan hingga mencapai kondisi tertentu. *K-means* memiliki tingkat efektifitas yang tinggi dan telah terbukti dalam banyak implementasi dengan hasil klasifikasi yang memuaskan [5].

Error! Reference source not found. menunjukkan *flowchart K-Means.* Algoritma *K-Means* akan dieksekusi dalam dua fase. Pada fase pertama, *K-Means* akan mengambil sampel secara acak sejumlah nilai K yang ditentukan sebagai *center* dari masing-masing kelas . Pada fase selanjutnya, algoritma akan menghitung jarak masing-masing data relatif terhadap *center*. Data yang terdekat dengan *center* akan menjadi anggota kelas dari *center* tersebut.



Gambar 1 *Flowchart* algoritma *K-Means*

- *Input* yang diperlukan pada algoritma ini adalah dataset dan jumlah kelas yang diinginkan
- Algoritma kemudian akan melakukan inisialisasi secara acak *centroid* sejumlah K yang dimasukkan.
- Kemudian, setiap data akan dihitung jaraknya relatif terhadap masing-masing *centroid* menggunakan formula *Euclidean Distance*. Rumus (1) adalah formula *Euclidean Distance*.

$$d(x_i, y_i) = [\sum_{i=1}^{n}(x_i - y_i)^2]^{\frac{1}{2}} \quad (1)$$

- Data yang terdekat dengan salah satu *centroid* kelas akan menjadi anggota dari kelas tersebut
- Setelah semua data telah terklasifikasi, kemudian akan diambil *centroid* baru yang dihitung dari rata-rata nilai setiap kelas
- Apabila terdapat perubahan nilai *centroid*, algoritma akan mengulang langkah urutan ke-3 hingga 5 sampai tidak terdapat perubahan nilai *centroid* lagi

D. Random Forest

Salah satu algoritma yang dapat digunakan untuk dua kasus sekaligus, baik regresi maupun klasifikasi adalah algoritma *Random Forest* [6]. Algoritma ini memanfaatkan *decision tree* untuk melakukan pembelajaran. *Decision tree* yang digunakan memiliki jumlah yang banyak. Hal ini karena penggunaan *single decision tree* memiliki akurasi yang rendah jika mendapatkan input baru yang berbeda dari dataset training. Berikut adalah langkah-langkah saat menerapkan algoritma *random forest* [6].

- Melakukan *bootstrap* sebanyak n_{tree}. n_{tree} didapatkan berdasarkan masukkan yang diterima oleh algoritma. *Bootstrap* adalah proses *resampling* dataset dengan mengambil sejumlah baris dan kolom secara acak dengan metode *row sampling with replacement*. Artinya antara satu *bootstrap* dan *bootstrap* yang lain akan berbeda.
- Dari setiap *bootstrap* akan dikembangkan *decision tree*. Pada umumnya ketika membentuk *decision tree*, *node* diambil dari nilai variabel terbaik yang dihitung menggunakan metode tertentu, salah satunya (*Gini Impurity*) dengan mempertimbangkan seluruh variabel. Akan tetapi, pada *random forest node* diambil secara acak dengan mengambil sejumlah m_{try} variabel, bukan secara keseluruhan.
- Prediksi data dengan melakukan agregasi dari setiap *decision tree* yang telah terbentuk (pada klasifikasi menggunakan *majority vote* sementara pada kasus regresi menggunakan *average*)

Akurasi dari random forest dihitung dengan menggunakan *Out-of-Bag* (OOB). OOB merupakan data yang ada di dalam dataset tetapi tidak terdapat dalam *bootstrap*. Dengan menguji data tersebut ke setiap *decision tree* akan terlihat persentase data prediksi yang benar dengan mengakumulasi dari setiap *decision tree*. Sebaliknya, *error rate* diketahui dengan menghitung persentase prediksi yang salah dari agregasi seluruh *decision tree*.

Random forest juga dapat melihat nilai signifikansi dari setiap variabel. Nilai signifikansi ini dapat dihitung dari seberapa besar *prediction error* meningkat ketika data pada

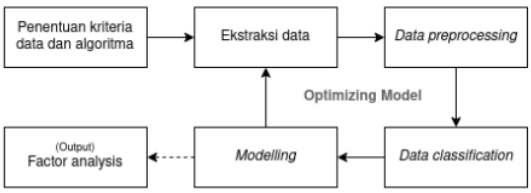
variabel tersebut diacak sementara variabel yang lain tetap sama [6].

E. Factor Analysis

Factor analysis adalah teknik yang digunakan untuk mempertimbangkan setiap variabel yang berpengaruh signifikan terhadap kelas tertentu. Teknik ini telah banyak digunakan untuk menyelesaikan masalah-masalah klasifikasi dalam berbagai bidang, diantaranya yaitu segmentasi pengguna, riset penyakit kanker, riset medis, dan riset masyarakat/lingkungan [6].

III. METODOLOGI PENELITIAN

Pengerjaan *Factor Analysis* Keberlanjutan Pengguna dilakukan dengan siklus pengembangan yang berulang karena dalam prosesnya dilakukan optimasi berkali-kali agar mencapai akurasi hasil tinggi. Langkah awal yang dilakukan adalah menentukan kriteria data yang akan diolah sekaligus metode atau algoritma apa yang akan diterapkan. Setelah itu, tahap pemrograman dimulai dengan mengekstraksi data dari MySQL database menggunakan SQL. Data yang telah diekstraksi kemudian diolah untuk menghilangkan data yang tidak diperlukan dan menyelaraskan seluruhnya dengan format yang konsisten. Langkah selanjutnya adalah melakukan klasifikasi menggunakan algoritma *K-Means*. Dari hasil klasifikasi tersebut, kemudian dilakukan *modelling* menggunakan algoritma *random forest* untuk mendapatkan nilai signifikansi dari setiap variabel. Seluruh tahapan tersebut dilakukan secara berulang dalam rangka mencapai nilai optimum. Hal tersebut dicapai mulai dengan melakukan *cleaning* data, menghilangkan data yang *outlier*, hingga melakukan *tuning* model. Gambar 2 adalah gambaran dari proses pengerjaan project.



Gambar 2 Siklus pengerjaan *project*

IV. PEMBAHASAN DAN HASIL

A. Pembahasan

Seiring meningkatnya pengembangan produk yang dilakukan oleh Jala, berbanding lurus dengan peningkatan jumlah pengguna yang tertarik menggunakannya. Tentunya Jala ingin pengguna tersebut akan selalu loyal menggunakan produk-produknya dan tidak berpaling ke kompetitor yang lain. Oleh karena itu penting dilakukan sebuah riset mengenai faktor-faktor apa yang dapat membuat pengguna untuk tetap menggunakan Jala setelah mencobanya selama satu bulan pertama. Adapun faktor-faktor yang dianalisis dalam penelitian ini adalah sebagai berikut.

1) Aktivitas pengguna dalam pengisian data kegiatan kulturisasi menggunakan sistem pencatatan budidaya. Dalam data sistem ini terdapat 46 variabel yang dianalisis, beberapa di antaranya, yaitu:
   - Jumlah pengguna yang terdaftar dalam satu tambak
   - Jumlah data pakan yang diisikan selama proses kulturisasi
   - Total kolam dalam satu tambak
   - Jumlah benih tebar dalam satu tambak
   - Jumlah data kualitas air yang diisikan selama proses kulturisasi
   - Total sampling yang telah dilakukan
   - Rata-rata *survival rate* (tingkat kelulushidupan) udang
   - Rata-rata luas area kolam dalam satu tambak
   - Jenis langganan (free/pro)
   - Total jumlah hari kulturisasi yang telah dilakukan
2) Aktivitas pengguna dalam menggunakan fitur-fitur yang ada di website. Pada bagian ini, fokus pengamatan adalah menganalisis kegiatan pengguna saat berinteraksi dengan fitur-fitur yang ada, seperti jumlah klik pada tautan-tautan yang terdapat dalam website. Tujuannya adalah untuk mengetahui seberapa besar pengaruh fitur-fitur yang dibuat untuk menarik pengguna agar tetap menggunakan produk Jala. Pada bagian ini, terdapat 160 variabel, beberapa di antaranya yaitu:
   - Intensitas login pengguna dalam satu tambak
   - Intensitas jumlah klik menuju detail halaman media (fitur yang dibuat khusus mengedukasi pengguna tentang perusahaan dan produk yang ditawarkan)
   - Intensitas jumlah klik menuju halaman daftar media
   - Intensitas jumlah klik menggunakan android menuju ke halaman daftar harga udang
   - Intensitas jumlah klik submit button untuk membuat form harga udang
   - Intensitas jumlah klik submit button untuk mengedit form harga udang
   - Intensitas jumlah klik berlangganan fitur platinum
   - Intensitas jumlah action copy kupon
   - User id
   - Tahun pendaftaran user ke sistem

Dalam pengerjaan project ini, terdapat dua peran, yaitu lead project dan data scientist. Lead project bertugas untuk menentukan dan membuat PRD (Project Requirements Document) yang berfungsi untuk mendokumentasi seluruh kebutuhan sistem serta menjadi dokumen pertanggungjawaban kepada CTO dan CEO berkaitan dengan project yang sedang dikerjakan. Lead project juga bertugas untuk menentukan hasil akhir dari project serta memastikan bahwa pengerjaan project sesuai target serta berada dalam rentang waktu yang sesuai dengan PRD yang telah dibuat. Data scientist bertugas untuk mengeksekusi seluruh tugas yang telah didefinisikan di dalam PRD. Mulai dari ekstraksi data, preprocessing, klasifikasi, hingga modelling dan optimasi model. Komunikasi antara data scientist dan lead project berlangsung dua arah agar memastikan bahwa seluruh kualifikasi project dapat terpenuhi dengan benar.

Project ini dikerjakan dalam periode satu bulan. TABEL I adalah timeline pengerjaan project.

TABEL I *TIMELINE* PENGERJAAN PROJECT

No	Aktivitas	Durasi
1.	Ekstraksi data dan *data preprocessing*. Ekstraksi dilakukan dengan mengambil data yang berada di database mulai dari data ukuran tambak, hingga aktivitas kulturisasi. Ekstraksi juga dilakukan dengan mengambil seluruh aktivitas *user* dalam website. Setelah itu, data yang sudah diekstraksi kemudian dibersihkan dari *missing value/ outlier data* lalu distandarisasi dengan format yang sama. Proses ekstraksi dan *preprocessing* dilakukan secara berulang agar dapat menghilangkan data *outlier* yang tidak terdeteksi sebelumnya	2 Minggu
2.	Klasifikasi data. Pada pengerjaan proses klasifikasi data, banyak melakukan riset dan uji coba metode dan jumlah cluster yang tepat. Terdapat dua metode yang diuji coba, menggunakan *cohort* (bulan dan tahun pendaftaran tambak) dan menggunakan k-means. Setelah dilakukan uji coba, k-means menjadi pilihan sebagai algoritma klasifikasi data.	5 hari
3.	*Modelling* dan *optimizing model*. *Modelling* dilakukan dengan menerapkan algoritma *random forest*. Proses penerapan model banyak berfokus pada *tuning* parameter agar mendapatkan akurasi model yang maksimal dan menghindari *overfitting* ataupun *underfitting*	9 hari

B. Hasil

1) Ekstraksi Data

Ekstraksi data diambil dari database sistem yang menggunakan MySQL. Gambar 3 menunjukkan query SQL untuk *retrieve* data dari MySQL. Secara umum untuk mengekstraksi data, sebagian besar menggunakan *query* join karena harus menggabungkan beberapa tabel sekaligus dengan struktur yang kompleks.

```
with main_table as (
    select
        timestamp (samplings.sampled_at) as date,
        samplings.average_weight,
        samplings.survival_rate,
        cycles.id as cycle_id,
        cycles.pond_id,
        ponds.farm_id,
        cycles.initial_age,
        cycles.started_at,
        cycles.finished_at
    from cycles
    inner join samplings
        on cycles.id = samplings.cycle_id
    inner join ponds
        on cycles.pond_id  = ponds.id
)
select
        specific_date.cycle_id,
        specific_date.pond_id,
        date_format(specific_date.date, '%Y-%m-%d') as date,
```

Gambar 3 *Query* MySQL

Selain menggunakan join, *with clause* juga banyak digunakan. *With clause* berguna untuk melakukan *subquery* sehingga dapat membantu penggabungan banyak data sekaligus dalam satu query SQL.

2) Data Preprocessing

Data *preprocessing* dilakukan dengan menggunakan bahasa pemrograman python versi 3.8.5. Untuk membantu mempermudah pengelolaan data, digunakan salah satu *library* python, yaitu Pandas. Pandas akan mengubah format csv menjadi dataframe (baris-kolom) untuk mempermudah pengelolaan data. Banyak tahapan yang dilakukan dalam *preprocessing*, beberapa diantaranya adalah sebagai berikut.

a) Memuat dan menghapus data

Gambar 4 menunjukkan pemuatan data ke python menggunakan pandas dan menghapus kolom-kolom yang tidak diperlukan.

```
# farms_data = pd.read_csv(path+'groupbyfarms.csv')
farms_data = pd.read_csv('groupbyfarms.csv')
```

```
farms_data_process = farms_data.drop(['farm_id', 'farm_name', 'register_year', 'register_month',
                                      'province', 'district', 'is_use_jala'], axis=1)
```

Gambar 4 Memuat dan menghapus data

b) Menyaring outlier data dan mengisi missing value

Dalam *preprocessing*, proses penggantian *missing value* dengan nilai tertentu sangat penting agar data kosong tidak terbuang. Salah satu metode statistik yang dapat digunakan untuk mengisi *missing value* adalah metode rata-rata. Gambar 5 menunjukkan *method* untuk menyaring data *outlier* dan mengisi *missing value*.

```
def filtering_quantile(df, lc) :
    df_temp = pd.DataFrame()

    for column_name in lc :
        cond = df.loc[:, column_name].between(df.loc[:,column_name].quantile(0.05),
                                              df.loc[:,column_name].quantile(0.95))
        val_temp = df.loc[cond, column_name].copy()
        df_temp = pd.concat([df_temp, val_temp], ignore_index=True, axis=1)

    return df_temp
```

```
def fill_empty_after_filtering(df, lc) :
    df.fillna(df.mean(), inplace=True)

    return df
```

Gambar 5 Menyaring data *outlier* dan mengisi *missing value*

c) Menggabungkan data

Masing-masing data hasil ekstraksi yang telah melalui tahap *preprocessing* akan digabungkan menjadi satu kesatuan data. Gambar 6 menunjukkan salah satu cara untuk menggabungkan dua *dataframe* atau lebih menjadi satu *dataframe* pada pandas.

```
dfs_farm = [farms_sum_data, farms_mean_data, total_farms]
dfs_farm = [df_farm.set_index('cohort_date') for df_farm in dfs_farm]
farms_cohort = dfs_farm[0].join(dfs_farm[1:]).reset_index()
```

Gambar 6 Menggabungkan data pada *pandas*

3) Klasifikasi Data

Dalam proses klasifikasi data, terdapat beberapa tahapan yang dilakukan, berikut adalah tahapan-tahapan tersebut.

a) Standarisasi nilai setiap variabel

Setiap variabel memiliki *range* nilai yang berbeda-beda. Hal ini akan sangat sulit jika langsung dilakukan klasifikasi. Oleh karena itu diperlukan standarisasi nilai dengan metode statistik tertentu untuk mengubah *range* nilai yang berbeda-beda menjadi satu *range* nilai yang sama. Gambar 7 menunjukkan program python untuk melakukan standarisasi nilai variabel.

```
from sklearn.preprocessing import StandardScaler
farm_std = StandardScaler().fit_transform(farm_result)
```

Gambar 7 Standarisasi nilai variabel

b) Penerapan PCA (Principal Component Analysis)

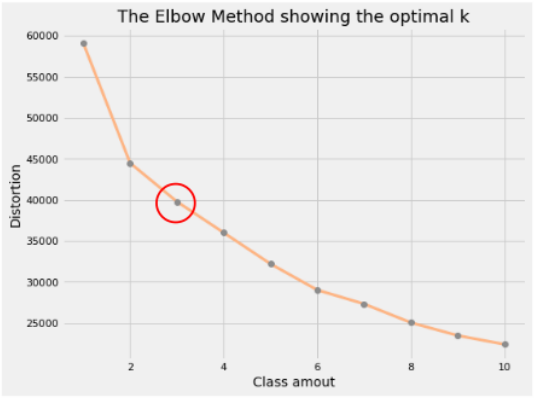
Metode ini berguna untuk mengubah dimensi dari suatu data. Hal ini bertujuan untuk meminimalkan dimensi dari data agar dapat lebih mudah dan cepat untuk diolah, Gambar 8 menunjukkan penerapan PCA.

```
from sklearn.decomposition import PCA
pca=PCA(n_components=11)
pca_farm = pca.fit_transform(farm_std)
```

Gambar 8 Penerapan PCA

c) K-means dan Elbow Method

Elbow method berfungsi untuk membantu menentukan berapakah jumlah K yang optimal untuk digunakan pada algoritma *k-means* sesuai dengan data yang dimiliki. Gambar 9 menunjukkan grafik *elbow method* dari data yang digunakan.



Gambar 9 Hasil *elbow method*

Nilai dari *elbow method* tersebut kemudian kita masukkan ke dalam algoritma *k-means* sebagai *input* k dalam algoritma tersebut. Gambar 10 dan Gambar 11 menunjukkan penerapan *k-means* dan hasil klasifikasinya.

```
from sklearn.cluster import KMeans
kmeans_farm = KMeans(
    init="random",
    n_clusters=cluster_needed,
    n_init=10,
    max_iter=300
#     random_state=42
)
kmeans_farm.fit(pca_farm)
```

```
KMeans(init='random', n_clusters=3)
```

Gambar 10 Penerapan *k-means*

Gambar 11 Hasil klasifikasi *k-means*

4) Modelling

a) Persiapan data training dan fungsi Random Forest

Data yang telah dipersiapkan dibagi menjadi dua, yaitu data *training* dan data *split*. Pembagian ini dilakukan dengan menggunakan fungsi *train_test_split* seperti yang terdapat pada Gambar 12.

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split

x = model_data.drop(columns= ['is_use_jala'])
y = model_data.is_use_jala

x_train, x_test, y_train, y_test= train_test_split(x, y, train_size= 0.85)

scaler= StandardScaler()
classifier= RandomForestClassifier(n_estimators= 100)
model= Pipeline(steps= [('scaler', scaler), ('classifier', classifier)])
```

Gambar 12 Persiapan data training dan fungsi Random Forest

b) Menentukan hyperparameter terbaik

Hyperparameter berfungsi untuk mengoptimasi algoritma agar dapat memberikan hasil luaran yang akurat. Secara *default*, penentuan *hyperparameter* dilakukan secara manual dengan mengganti nilainya dan melakukan training. Akan tetapi, pada python, terdapat *library* yang mampu mencari nilai *hyperparameter* terbaik dengan menggunakan fungsi *GridSearchCV* seperti yang terdapat pada Gambar 13 dan Gambar 14

```
params= {'classifier__max_depth': [None,5, 10, 20, 30],
         'classifier__criterion': ['entropy'],
         'classifier__min_samples_split': [2,6,12],
         'classifier__min_samples_leaf': [1,5,9]}
```

Gambar 13 Inisialisasi pilihan nilai hyperparameter

```
selector= GridSearchCV(estimator= model, param_grid= params)
selector.fit(x_train, y_train)
selected_model= selector.best_estimator_
```

Gambar 14 Menerapkan GridSearchCV

c) Fungsi menyimpan dan memuat model

Melatih suatu model membutuhkan waktu yang cukup lama terlebih apabila data yang di-*training* sangat banyak. Oleh karena itu, diperlukan fungsi simpan dan muat model sehingga ketika model telah selesai dilatih dapat disimpan dan bisa langsung dimuat tanpa proses *training* lagi. Gambar 15 menunjukkan fungsi simpan dan muat model.

```
def save_model(selected_model, file_name):
    model_result= {
        'split': {'train': {'x': x_train, 'y': y_train},
                  'test': {'x': x_test, 'y': y_test}
                  },
        'model': selected_model
    }

    CURR_DIR = os.path.dirname(os.path.realpath('__file__'))

    with open(os.path.join(CURR_DIR, (file_name+'.pickle')), 'wb') as infile:
        pickle.dump(model_result, infile)
        infile.close()
    del infile
```

```
def load_model(file_name) :
    CURR_DIR = os.path.dirname(os.path.realpath('__file__'))

    with open(os.path.join(CURR_DIR, (file_name+'.pickle')), 'rb') as infile:
        result= pickle.load(infile)
        infile.close()

    del infile
    return result
```

Gambar 15 Menyimpan dan memuat model

d) Ekstrak nilai feature importances variabel

Gambar 16 menunjukkan baris kode yang berfungsi menghitung nilai kepentingan (*feature importances*) dari semua variabel yang dianalisis. Nilai inilah yang menjadi indikator pengaruh variabel terhadap keberlanjutan pengguna.

```
# Load model
result = load_model('random_forest_model_1_web_data')

x_train= result['split']['train']['x']
y_train= result['split']['train']['y']
selected_model= result['model']
```

```
# List fitur-fitur penting

feature_importances= pd.DataFrame()
feature_importances['features']= x_train.columns
feature_importances['feature_importances']= selected_model['classifier'].feature_importances_.copy()
feature_importances= feature_importances.sort_values('feature_importances', ascending= False)
```

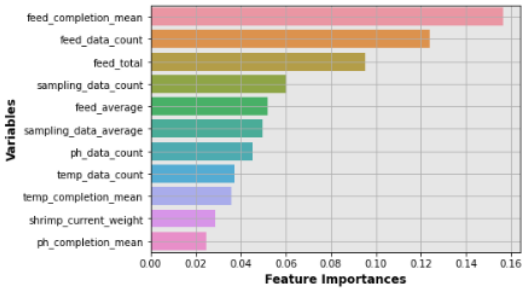
Gambar 16 Ekstrak *feature importances*

5) Factor Analysis (Output)

Hasil dari ekstraksi *feature importances* menjadi luaran *factor analysis*. Melalui luaran tersebut, akan terlihat seberapa besar pengaruh dari setiap variabel terhadap keberlanjutan pengguna. Terdapat dua garis besar hasil dari luaran ini, yaitu sebagai berikut.

a) Sepuluh besar faktor yang paling berpengaruh

Gambar 17 menunjukkan sepuluh teratas faktor yang berpengaruh terhadap keberlanjutan pengguna. *Feed_completion_mean* adalah persentase kelengkapan data pakan yang diisikan oleh pengguna selama proses kulturisasi. Dari nilai tersebut, menunjukkan bahwa semakin sering pengguna menggunakan fitur pengisian pakan kulturisasi, kemungkinan lanjut menggunakan produk Jala juga tinggi.



Gambar 17 Hasil sepuluh terbesar *Feature Importances*

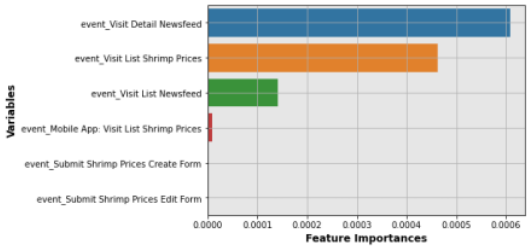
Penjelasan lima variabel selanjutnya adalah sebagai berikut.

- *feed_data_count*: Jumlah data pakan yang terisi
- *feed_total*: Total pakan yang tercatat
- *sampling_data_count*: Total jumlah data sampling yang tercatat
- *feed_average*: Rata-rata pemberian pakan yang tercatat
- *sampling_data_average*: Rata-rata jumlah data sampling yang tercatat

b) Pengaruh fitur media pada keberlanjutan pengguna

Fitur tambahan seperti fitur media yang dibuat khusus untuk mengedukasi pengguna ternyata masih belum berdampak besar pada keberlanjutan pengguna. Gambar 18 menunjukkan pengaruh fitur-fitur tersebut. Berikut adalah penjelasan dari masing-masing variabel.

- *even_Visit Detail Newsfeed*: Intensitas pengguna mengunjungi detail fitur media.
- *event_Visit List Shrimp Prices*: Intensitas pengguna mengunjungi halaman daftar harga udang
- *event_Visit List Newsfeed*: Intensitas pengguna mengunjungi halaman daftar media
- *event_Mobile App Visit List Shrimp Prices*: Intensitas pengguna mengunjungi halaman daftar harga udang menggunakan *mobile apps*
- *event_Submit Shrimp Prices Create Form*: Intensitas pengguna melakukan submit pembuatan form harga udang
- *event_Submit Shrimp Prices Edit Form*: Intensitas pengguna melakukan submit pengubahan form harga udang



Gambar 18 Hasil *Feature Importances* media

V. KESIMPULAN

Dari hasil *factor analysis* dan pemaparan terhadap seluruh stakeholder Jala, intensitas penggunaan sistem Jala pada satu bulan pertama sangat mempengaruhi keberlanjutan pengguna. Semakin sering pengguna menggunakan fitur pencatatan aktivitas pakan dan *sampling*, tingkat keberlanjutan pengguna ke bulan-bulan selanjutnya juga semakin tinggi. Dengan demikian diperlukan edukasi, kemudahan akses, dan kemudahan penggunaan fitur, terutama pada fitur pakan dan *sampling*. Sementara itu, efektivitas fitur media masih belum banyak berpengaruh terhadap keberlanjutan pengguna. Oleh karena itu, perlu untuk dipertimbangkan peningkatan pada fitur media. Mulai dari sisi konten, hingga kemudahan akses.

Project ini menjadi dasar untuk pengembangan project-project analisis keberlanjutan pengguna ke depan. Salah satu topik yang menarik untuk dikembangkan selanjutnya adalah *factor analysis* pengaruh keberadaan *mobile apps* untuk keberlanjutan pengguna.

REFERENSI

[1] [1] V. Arief Wardhany, H. Yuliandoko, M. U. Harun Al rasyid, and I. G. Puja Astawa, "Aplikasi Monitoring Dan Kontrol Tambak Udang Vanammei," vol. 11, pp. 37–42, 2020.

[2] [2] R. Govindaraju, T. Simatupang, and T. A. Samadhi, "Perancangan Sistem Prediksi Churn Pelanggan," *Tek. Inform.*, vol. 9, no. 1, pp. 33–42, 2008, [Online]. Available: http://puslit2.petra.ac.id/ejournal/index.php/inf/article/view/16893.

[3] [3] K. I. Satoto, R. R. Isnanto, R. Kridalukmana, and K. T. Martono, "Optimizing MySQL database system on information systems research, publications and community service," *Proc. - 2016 3rd Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2016*, pp. 1–5, 2017, doi: 10.1109/ICITACEE.2016.7892476.

[4] [4] V. Agarwal, "Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis," *Int. J. Comput. Appl.*, vol. 131, no. 4, pp. 30–36, 2015, doi: 10.5120/ijca2015907309.

[5] [5] N. Shi, X. Liu, and Y. Guan, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," *3rd Int. Symp. Intell. Inf. Technol. Secur. Informatics, IITSI 2010*, pp. 63–67, 2010, doi: 10.1109/IITSI.2010.74.

[6] [6] N. Amruthnath and T. Gupta, "Factor Analysis in Fault Diagnostics Using Random Forest," no. April, 2019, doi: 10.4172/2169-0316.1000278.

Factor Analysis Keberlanjutan Pengguna Jala Menggunakan Klasifikasi K-Means dan Algoritma Random Forest

ORIGINALITY REPORT

7%	7%	5%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	jurnal.ugm.ac.id Internet Source	1%
2	www.warse.org Internet Source	1%
3	journal.uii.ac.id Internet Source	1%
4	Sindhura Bonthu, Priscila Rodrigues Armijo, Tiffany Tanner, Qiuming Zhu. "Using Machine Learning to Improve Surgical Outcomes", 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019 Publication	1%
5	media.neliti.com Internet Source	1%
6	ojs.uajy.ac.id Internet Source	<1%
7	help.uii.ac.id Internet Source	<1%

8	text-id.123dok.com Internet Source	<1 %
9	www.coursehero.com Internet Source	<1 %
10	jtiik.ub.ac.id Internet Source	<1 %
11	www.scribd.com Internet Source	<1 %
12	duddyarisandi.wordpress.com Internet Source	<1 %
13	ml.scribd.com Internet Source	<1 %
14	peni1981.wordpress.com Internet Source	<1 %
15	portabs.blogspot.com Internet Source	<1 %
16	www.pure.ed.ac.uk Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		