Analisis Keberlanjutan Pengguna Jala Menggunakan *Factor Analysis*

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-produknya, 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 analisis keberlanjutan pengguna dengan teknik *Factor Analysis* diinisiasi untuk mencari tahu faktor yang mempengaruhi keberlanjutan pengguna sejak satu bulan pertama menggunakan Jala. Teknik-teknik lain yang digunakan adalah *preprocessing* menggunakan *pandas*, pengelompokan data dengan algoritma *k-means*, dan *modelling* menggunakan *random forest*. Hasilnya, intensitas penggunaan fitur pakan, sampling, dan pencatatan kualitas air pada saat satu bulan pertama pengguna menggunakan Jala berpengaruh besar terhadap keberlanjutan penggunaan pada bulan-bulan berikutnya.**

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

I. PENDAHULUAN

Jala merupakan salah satu perusahaan yang bergerak di bidang akuakultur khususnya dalam pembudidayaan udang. Memiliki tujuan utama memudahkan para petambak -selanjutnya disebut sebagai pengguna-, Jala menghadirkan produk-produk berbasis teknologi agar dapat membantu pengguna 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 pengguna 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 pengguna 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 terdaftar 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* analisis keberlanjutan pengguna untuk mengidentifikasi indikator utama yang menentukan keberlanjutan pengguna baru sejak satu bulan pertama menggunakan sistem Jala. *Project* ini diberi nama Analisis Keberlanjutan Pengguna dengan teknik *Factor Analysis*.

Factor Analysis merupakan salah satu teknik dalam *data science* yang mengidentifikasi setiap fitur yang bersinggungan langsung dengan objek -dalam hal ini pengguna- dan menganalisisnya untuk dilihat intensitas pengaruhnya terhadap nilai *churn* [3]. 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* ini dapat dijadikan indikator awal dalam mencegah tingginya *churn* pengguna.

Publikasi ini akan menjelaskan hasil penerapan *Factor Analysis* keberlanjutan pengguna Jala sejak satu bulan pertama penggunaan produk. Penggunaan periode satu bulan pertama dalam penelitian didasarkan pada lama durasi kulturisasi yang rata-rata memakan waktu 30-60 hari. Dengan data yang tersedia serta penggunaan teknik *clustering* dan algoritma *Random Forest*, luaran yang dihasilkan akan berupa persentase nilai pengaruh setiap variabel.

II. LANDASAN TEORI

A. Ekstraksi Data

Ekstraksi 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 -pimpinan, *marketing*, bisnis manajer, dan seluruh karyawan Jala- yang membutuhkan informasi tersebut. Pada *software*, data disimpan dan dikelola secara sistematis menggunakan *database management system* (DBMS). MySQL merupakan salah satu DBMS 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 [4].

B. *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, dan inkonsisten [5]. Salah satu *tools* untuk melakukan *preprocessing* 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* [5].

C. *Clustering*

Clustering dalam *data science* merupakan suatu proses pengelompokkan data berdasarkan kriteria tertentu baik yang terlihat secara eksplisit maupun implisit [6]. Dalam proses *clustering*, algoritma akan melakukan pencarian pola dari suatu data kemudian akan mengelompokkannya berdasarkan pola tersebut. Salah satu algoritma yang terkenal untuk *clustering* 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 [6]. Penggunaan algoritma K-Means dalam penelitian ini dikarenakan penerapannya yang mudah, banyak digunakan, dan secara umum telah terbukti efektivitasnya.

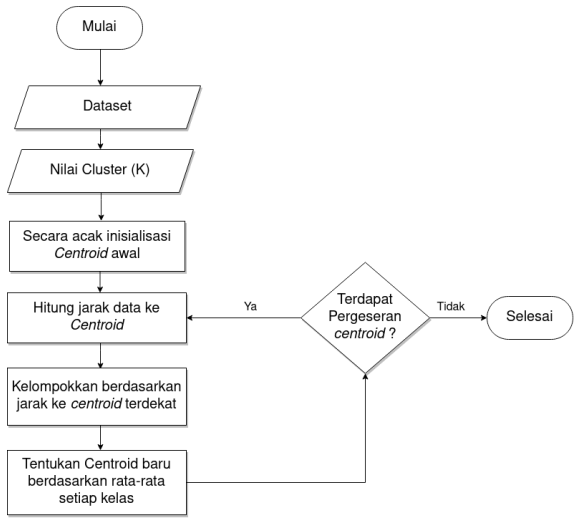
Gambar 1 menunjukkan *flowchart K-Means.* Algoritma *K-Means* akan dieksekusi dalam dua fase. Pada fase pertama, *K-Means* akan mengambil sampel secara acak sejumlah nilai K yang ditentukan sebagai *center* dari masing-masing kelas. Pada fase selanjutnya, algoritma akan menghitung jarak masing-masing data relatif terhadap *center*. Data yang terdekat dengan *center* akan menjadi anggota kelas dari *center* tersebut. Berikut adalah penjelasan detail dari setiap langkah yang terdapat pada *flowchart*.

- *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*.

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

x_i dan y_i merupakan objek x dan y ke-i. Sementara n adalah jumlah objek.

- Data yang terdekat dengan salah satu *centroid* kelas akan menjadi anggota dari kelas tersebut.
- Setelah semua data telah dikelompokkan, 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.



Gambar 1. *Flowchart* algoritma *K-Means*

D. *Random Forest*

Salah satu algoritma yang dapat digunakan untuk dua kasus sekaligus, baik regresi maupun klasifikasi adalah algoritma *Random Forest* [3]. 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 masukkan baru yang berbeda dari dataset *training*. Berikut adalah langkah-langkah saat menerapkan algoritma *random forest* [3].

- 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_{txy} 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 [3].

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 [3].

III. METODOLOGI PENELITIAN

Pengerjaan Analisis 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. Kriteria data dan algoritma yang digunakan dapat dilihat pada TABEL I.

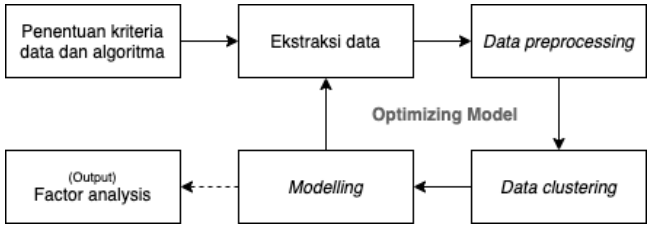
TABEL I KRITERIA DATA

No	Kriteria
1.	Seluruh data siklus yang tercatat hingga tanggal 22-11-2020
2.	Data dengan pengisian informasi pakan di atas 80% selama siklus berlangsung
3.	Data tambak yang memenuhi kriteria satu dan dua kemudian diagregasi dari hari pertama terdaftar hingga satu bulan selanjutnya dengan teknik penjumlahan dan rata-rata
4.	Identifikasi lanjut atau berhenti menggunakan Jala ditentukan dari persentase kelengkapan data pakan dalam satu bulan pertama menggunakan Jala. Kelengkapan data di atas 30% diasumsikan lanjut menggunakan Jala
5.	Data yang digunakan berasal dari seluruh data yang dimasukkan oleh pengguna melalui semua fitur pada website
6.	Data aktivitas pengguna pada website, seperti intensitas kunjungan setiap halaman dan jumlah klik pada masing-masing *button* atau *link* yang tersedia
7.	*Clustering* menggunakan Algoritma *K-Means* untuk melihat bagaimana data tersebar dan mengelompokkannya sesuai dengan persebaran tersebut.
8.	Algoritma *Random Forest* digunakan untuk menentukan tingkat signifikansi dari masing masing variabel yang diolah

Setelah itu, tahap pemrograman dimulai dengan mengekstraksi data dari MySQL database menggunakan SQL. Hasil dari *query* SQL akan disimpan dalam bentuk *file* dengan format *csv*. Data yang telah diekstraksi kemudian masuk ke tahap *preprocessing* untuk menghilangkan data yang tidak diperlukan dan menyelaraskan seluruhnya dengan format yang konsisten.

Tahap *preprocessing* dimulai dengan memuat data yang telah diekstrak dalam format *csv* ke *python* menggunakan *Pandas*. Kemudian dianalisis dengan beberapa cara, salah satunya adalah mengidentifikasi *missing value* dari setiap variabel. *Missing value* tersebut lalu diisi dengan berbagai metode diantaranya adalah metode rata-rata. Data yang sudah diolah kemudian disimpan kembali dalam bentuk *csv*.

Langkah selanjutnya adalah melakukan *clustering* menggunakan algoritma *K-Means*. Nilai dari K biasanya ditentukan secara manual, namun pada penelitian ini akan menggunakan metode *Elbow Method*. Dari hasil *clustering* 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. Pengulangan yang dimaksud berupa kembali ke tahap awal seperti ekstraksi dan *preprocessing* yang salah satu penyebabnya karena banyaknya nilai *outlier*. Gambar 2 adalah gambaran dari proses pengerjaan *project*.



Gambar 2. Proses 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 selama satu bulan pertama penggunaan. 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. *Data scientist* bertugas untuk mengeksekusi seluruh tugas yang telah didefinisikan di dalam PRD. Mulai dari ekstraksi data, *preprocessing*, *clustering*, hingga *modelling*. Komunikasi antara *data scientist* dan *lead project* berlangsung dua arah agar memastikan bahwa seluruh kualifikasi *project* dapat terpenuhi dengan benar.

B. Hasil

1) Ekstraksi Data

Ekstraksi data diambil dari database sistem yang menggunakan MySQL. Gambar 3 menunjukkan salah satu hasil *query* SQL yang merupakan data sampling. Secara umum untuk mengekstraksi data, sebagian besar menggunakan *query join* karena harus menggabungkan beberapa tabel dengan struktur yang kompleks. Hal tersebut ditunjukkan pada data sampling yang didapatkan dengan menggabungkan tabel sampling, siklus, dan tambak. Selain menggunakan *join*, *with clause* juga banyak digunakan. *With clause* berguna untuk melakukan *subquery* sehingga dapat membantu penggabungan banyak data dalam satu *query* SQL.

	cycle_id	pond_id	date	started_at	finished_at	doc	farm_name
1	64	1	2018-08-17	2018-07-20	2018-11-08	28	PW I
2	64	1	2018-08-24	2018-07-20	2018-11-08	35	PW I
3	64	1	2018-08-31	2018-07-20	2018-11-08	42	PW I
4	64	1	2018-09-07	2018-07-20	2018-11-08	49	PW I
5	64	1	2018-09-14	2018-07-20	2018-11-08	56	PW I
6	64	1	2018-09-21	2018-07-20	2018-11-08	63	PW I
7	64	1	2018-10-05	2018-07-20	2018-11-08	77	PW I

Gambar 3. Data sampling pengguna

2) Data Preprocessing

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 data dengan format *csv* yang dimuat di *python* menggunakan *Pandas*. Setelah data dimuat, salah satu langkah yang dilakukan adalah mengeluarkan data yang belum diperlukan. Pada Gambar 5 terdapat beberapa variabel dikeluarkan, seperti *farm_id, farm_name, register_month, province, district,* dan *is_use_jala*. Pengeluaran variabel-variabel tersebut bertujuan untuk memisahkan data yang bertipe kategorik dan numerik sehingga data dapat diolah untuk proses *clustering*.

```
farms_data.head()
```

	farm_id	farm_name	farm_age	register_year	register_month	user_count	province	district	total_pond	pond_area_mean	...	salinity_mean	or_ter
0	3	PW I	992	2018	3	15	JAWA TENGAH	PURWOREJO	4.0	1452.75	...	21.347203	
1	40	Mina Barokah	952	2018	4	1	JAWA TENGAH	PEKALONGAN	1.0	0.00	...	19.161319	
2	41	Tambak Banjarsari	950	2018	4	3	JAWA TENGAH	CILACAP	6.0	0.00	...	18.865838	
3	58	Irpan Sauri Siam	915	2018	5	1	NaN	NaN	1.0	0.00	...	0.000000	
4	61	Tambak Indra	908	2018	5	1	NaN	NaN	1.0	0.00	...	0.000000	

5 rows × 45 columns

Gambar 4. Memuat data pada *Pandas*

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

	farm_age	user_count	total_pond	pond_area_mean	total_pond_area	seed_average	total_cycle	total_doc	ongoing_free	ongoing_pro	...	ph_mean	sa
0	992	15	4.0	1452.75	5811.0	1.475000e+05	4.0	912.0	0	0	...	8.033275	
1	952	1	1.0	0.00	0.0	1.000000e+05	1.0	120.0	0	0	...	8.257229	
2	950	3	6.0	0.00	0.0	1.833333e+05	6.0	720.0	0	0	...	8.171364	
3	915	1	1.0	0.00	0.0	2.500000e+05	1.0	120.0	0	0	...	0.000000	
4	908	1	1.0	0.00	0.0	1.000000e+05	1.0	120.0	0	0	...	0.000000	

Gambar 5. Menghapus data pada *Pandas*

b) Menyaring outlier data dan mengisi missing value

Dalam *preprocessing*, salah satu masalah yang sering ditemui adalah banyaknya data *outlier*. Data *outlier* merupakan data yang memiliki nilai berbeda dengan perbandingan yang sangat jauh terhadap sebagian besar data [7]. Salah satu teknik yang digunakan adalah dengan melakukan penyaringan *quantile*. Teknik ini dilakukan dengan menentukan *quantile* bawah dan atas kemudian mengambil nilai yang berada di rentang tersebut. Nilai quantile yang digunakan pada penelitian ini sebesar 0.05 dan 0.95.

Gambar 6 menunjukkan hasil sebelum dan sesudah penerapan *filter quantile*. Hasilnya menunjukkan sebelum menggunakan *filter quantile,* nilai maksimum dari variabel *shrimp_current_weight* atau berat udang berada di angka 8000 gr. Padahal rata-rata berat udang ada di angka 14.99 gr [8].

Sebelum

```
1 farms_data.shrimp_current_weight.max()
```

```
80000.0
```

Sesudah

```
1 farm_result.shrimp_current_weight.max()
```

```
16.222
```

Gambar 6. Hasil *filter quantile*

Tahapan selanjutnya adalah proses pengisian *missing value* yang diakibatkan *quantile filter*. Hal ini bisa terjadi karena penyaringan data *outlier* dilakukan pada setiap variabel sehingga jumlah data tersaring berbeda-beda dan menghasilkan *missing value*. Gambar 7 menunjukkan total *missing value* sebelum dan sesudah pengisian dengan metode rata-rata.

Sebelum

```
1 farm_result.isnull().sum()
```

farm_age	204
user_count	103
total_pond	103
pond_area_mean	203
total_pond_area	199
seed_average	196
total_cycle	87
total_doc	179
ongoing_free	52
ongoing_pro	12
sr_mean	103
sr_std	6
sr_max	103
sr_min	8

Sesudah

```
1 farm_result.isnull().sum()
```

farm_age	0
user_count	0
total_pond	0
pond_area_mean	0
total_pond_area	0
seed_average	0
total_cycle	0
total_doc	0
ongoing_free	0
ongoing_pro	0
sr_mean	0
sr_std	0
sr_max	0
sr_min	0

Gambar 7. Hasil pengisian *missing value*

3) Clustering

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

a) Standarisasi nilai setiap variabel

Setiap variabel memiliki rentang nilai yang berbeda-beda. Hal ini akan sangat sulit jika langsung dilakukan pengelompokkan. Oleh karena itu, diperlukan standarisasi nilai dengan metode statistik tertentu untuk mengubah rentang nilai yang berbeda-beda menjadi satu rentang nilai yang sama. Gambar 8 menunjukkan perbandingan antara rentang nilai sebelum dan sesudah penerapan standarisasi nilai menggunakan fungsi *StandardScaler* dari *library sklearn python*.

Sebelum

	farm_age	user_count	total_pond	pond_area_mean	total_pond_area
min	28.0	1.0	1.0	9.0	10.0
max	631.0	3.0	6.0	7500.0	19846.0

Sesudah

	StandardScaler Result
min	-1.645280
max	13.777293

Gambar 8. Hasil penerapan standarisasi nilai

b) Penerapan PCA (Principal Component Analysis)

PCA merupakan teknik menganalisis multivariabel (*multivariate technique*) yang bertujuan untuk memperkecil dimensi dari suatu data [9]. Dalam proses implementasi, penentuan jumlah dimensi yang akan dibentuk dari *PCA* ditentukan berdasarkan nilai *variance* dari masing-masing variabel. Gambar 9 menunjukkan perhitungan *variance* dari setiap variabel. Pada penelitian ini, nilai *variance* yang digunakan, yaitu kurang dari 0.7 sehingga penerapan *PCA* menghasilkan 11 variabel yang akan menjadi masukkan dalam proses selanjutnya.

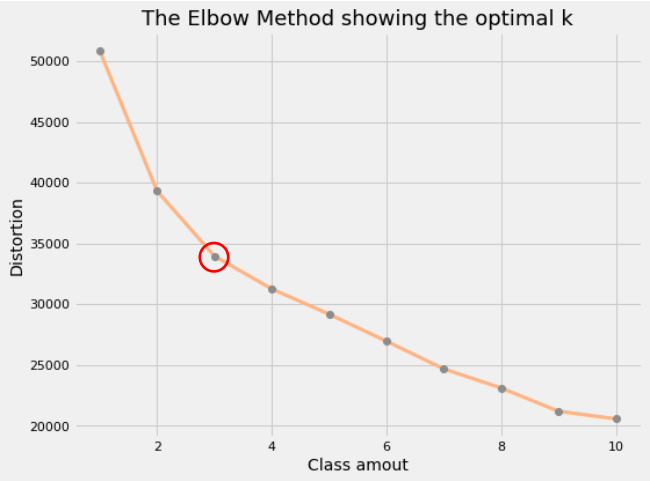
```
1 np.cumsum(pca.explained_variance_ratio_)
```

```
array([0.22065611, 0.31427913, 0.38032754, 0.42939096, 0.47596618,
       0.51924364, 0.5619235 , 0.59829995, 0.63167804, 0.66295968,
       0.68997068, 0.71476315, 0.7387266 , 0.76203991, 0.78368405,
       0.8035308 , 0.82273988, 0.84026495, 0.85704603, 0.87209149,
       0.88677885, 0.89986113, 0.91211676, 0.92374926, 0.93405429,
       0.94390165, 0.95313434, 0.96175222, 0.96984098, 0.97584282,
       0.98167678, 0.98713319, 0.99219709, 0.99542115, 0.99781692,
       1.        , 1.        , 1.        ])
```

Gambar 9. Nilai *variance* dari penerapan *PCA*

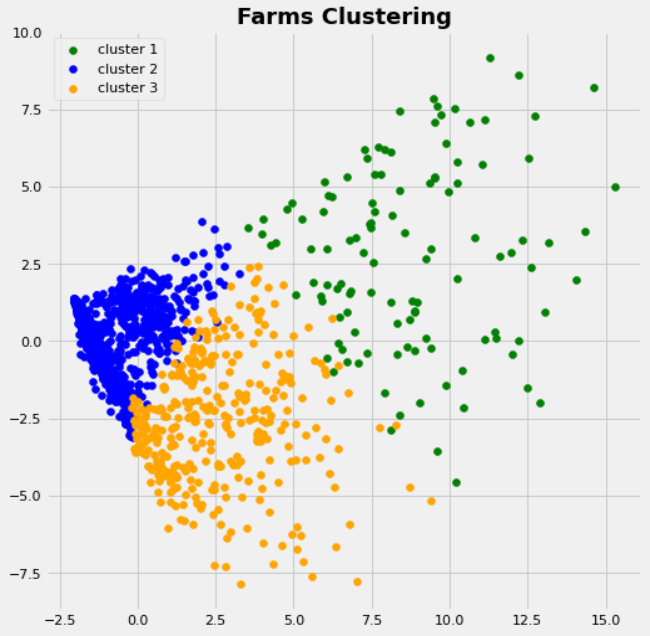
c) K-means dan Elbow Method

Elbow method berfungsi untuk membantu menentukan berapakah jumlah K yang optimal digunakan pada algoritma *K-Means* sesuai dengan data yang dimiliki. Proses penentuan *elbow method* dilakukan dengan melakukan *clustering* mulai dari satu hingga sejumlah n-1 kelas dengan n adalah total jumlah variabel yang digunakan -dalam hal ini 11 variabel-. Langkah selanjutnya adalah menghitung nilai SSE (*Sum Squared Error*) dari setiap kelas yang telah dikelompokkan. Pada umumnya, penentuan nilai K akan dilakukan secara manual dengan melakukan *plotting* hasil perhitungan setiap SSE. Setelah itu, kelas yang memiliki titik berbentuk sudut ditetapkan sebagai nilai K sebagaimana yang terdapat pada pada Gambar 10. Akan tetapi, pada *python* kita dapat menggunakan *library kneed* dengan fungsi *KneeLocator* untuk menentukan nilai K dari hasil perhitungan SSE. *Library* ini menerapkan algoritma "*kneedle*" yang merupakan salah satu teknik dalam menghitung *operating point* pada *system behaviour* [10].



Gambar 10 Hasil *elbow method*

Nilai K yang didapatkan adalah 3 dan menjadi masukkan pada proses *clustering*. Gambar 11 menunjukkan hasil *clustering* menggunakan K-Means.



Gambar 11. Hasil *clustering, K=3*

4) Modelling

Modelling akan dilakukan dengan dua versi data, data pertama adalah tanpa menggunakan variabel label -label hasil *K-Means Clustering*- dan yang kedua dengan variabel label. Tujuannya adalah untuk melihat hasil dari dua sisi yang berbeda, dari sisi tanpa penambahan variabel label dan dengan penambahan variabel tersebut.

a) Persiapan data training dan fungsi Random Forest

Terdapat dua jenis data *training* yang dipersiapkan untuk pembuatan model, yaitu data tanpa variabel label dan dengan variabel label. Masing-masing dari data tersebut kemudian akan dibagi menjadi dua bagian, yaitu data *training* dan data *test* dengan menggunakan *library sklearn, train_test_split*. Gambar 12 menunjukkan hasil pembagian data tersebut.

	num_x_train_data	num_x_test_data	num_y_train_data	num_y_test	total_variable
without_label	1624	287	1624	287	197
with_label	1624	287	1624	287	198

Gambar 12 Hasil pembagian data

keterangan:

- *num_x_train_data*: jumlah data *training* x (*independent variable*)
- *num_x_test_data*: jumlah data *test* x
- *num_y _train_data*: jumlah data *training* y (*dependent variable*)
- *num_y_test_data*: jumlah data *test* y
- *total_variable*: Total keseluruhan variabel.

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*. Gambar 13 menunjukkan *hyperparameter* yang dimasukkan ke dalam fungsi *GridSearchCV*. *Hyperparameter* paling optimum untuk data tanpa label dan dengan label ditunjukkan pada 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 *Hyperparameter* model

```
Pipeline(steps=[('scaler', StandardScaler()),
                ('classifier',
                 RandomForestClassifier(criterion='entropy', max_depth=30,
                                        min_samples_split=6))])
```

Gambar 14 Optimum *hyperparameter*

c) Fungsi menyimpan dan memuat model

Melatih suatu model membutuhkan waktu yang cukup lama terlebih apabila data yang dilatih 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. Terdapat dua model yang dihasilkan dari penelitian ini. Kedua model tersebut kemudian disimpan dalam format *.pickle*.

d) Ekstrak nilai feature importances variabel

Model yang berhasil disimpan dapat digunakan untuk mengekstrak nilai *feature importances* dari setiap variabel. *Feature Importances* adalah salah satu parameter yang didapatkan dari penerapan algoritma *Random Forest*. Parameter inilah yang menjadi penilaian pengaruh variabel terhadap keberlanjutan pengguna. Gambar 15 menunjukkan sepuluh variabel teratas dengan tingkat kepentingan tinggi.

Tanpa label

	variable	feature_importances
19	feed_data_count	0.195806
14	feed_completion_mean	0.131775
20	feed_total	0.099388
21	feed_average	0.054369
22	temp_data_count	0.039496
35	sampling_data_average	0.037696
15	temp_completion_mean	0.034365
25	salinity_data_count	0.030262
36	shrimp_growth_average	0.027693
24	ph_data_count	0.027341

Dengan label

	variable	feature_importances
14	feed_completion_mean	0.154033
19	feed_data_count	0.139287
20	feed_total	0.096742
21	feed_average	0.056419
34	sampling_data_count	0.051452
35	sampling_data_average	0.041714
22	temp_data_count	0.039535
37	shrimp_current_weight	0.028809
36	shrimp_growth_average	0.024064
24	ph_data_count	0.022345

Gambar 15. Hasil *feature importances*

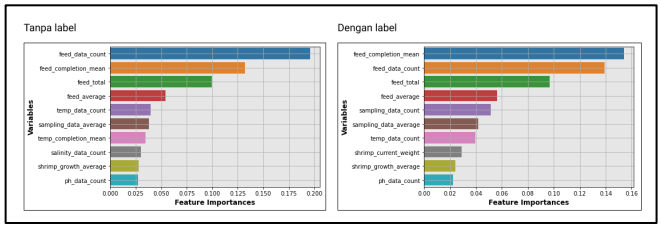
5) Factor Analysis (Output)

Hasil dari ekstraksi *feature importances* menjadi luaran *factor analysis*. Terdapat dua garis besar hasil dari luaran ini, yaitu sebagai berikut.

a) Sepuluh besar faktor yang paling berpengaruh

Gambar 16 menunjukkan perbandingan antara dua data yang digunakan sebagai masukkan algoritma *random forest*. Hasilnya menunjukkan nilai yang hampir sama satu sama lain, yaitu sepuluh faktor paling dominan berpengaruh terhadap keberlanjutan pengguna adalah variabel yang berkaitan dengan fitur pencatatan pakan, sampling, dan kualitas air. Berikut adalah pengelompokkan setiap variabel berdasarkan fiturnya.

- Pencatatan pakan: *feed_completion_mean, feed_data_count, feed_total, feed_average*.
- Pencatatan sampling: *sampling_data_average, shrimp_growth_average, sampling_data_count, shrimp_current_weight*
- Pencatatan kualitas air: *ph_data_count, temp_data_count, temp_completion_mean, salinity_data_count.*

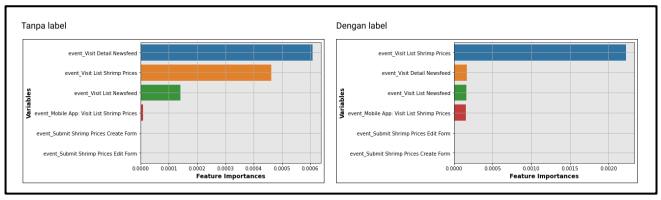


Gambar 16 Hasil sepuluh terbesar *Feature Importances*

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 17 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 17 Hasil *Feature Importances* media

V. KESIMPULAN

Dari hasil Analisis Keberlanjutan Pengguna 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, sampling, dan kualitas air, tingkat keberlanjutan pengguna ke bulan-bulan selanjutnya juga semakin tinggi. Dengan demikian diperlukan edukasi, kemudahan akses, dan kemudahan penggunaan fitur, terutama pada fitur pakan, sampling, dan pencatatan kualitas air. 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 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] 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] 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] N. Amruthnath and T. Gupta, "Factor Analysis in Fault Diagnostics Using Random Forest," no. April, 2019, doi: 10.4172/2169-0316.1000278.

[4] 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.

[5] 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.

[6] 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.

[7] Z. A. Bakar, R. Mohemad, A. Ahmad, and M. M. Deris, “A comparative study for outlier detection techniques in data mining,” *2006 IEEE Conf. Cybern. Intell. Syst.*, 2006, doi: 10.1109/ICCIS.2006.252287.

[8] J. Budidaya, P. Fakultas, and P. Universitas, “Produktivitas Udang Putih,” vol. 2, no. 1, pp. 48–53, 2006.

[9] J. A. López del Val and J. P. Alonso Pérez de Agreda, “Principal components analysis,” *Aten. Primaria*, vol. 12, no. 6, pp. 333–338, 1993, doi: 10.5455/ijlr.20170415115235.

[10] V. Satopää, J. Albrecht, D. Irwin, and B. Raghavan, “Finding a ‘kneedle’ in a haystack: Detecting knee points in system behavior,” *Proc. - Int. Conf. Distrib. Comput. Syst.*, pp. 166–171, 2011, doi: 10.1109/ICDCSW.2011.20.