



*A Study by*  
Mutiara Kusuma Hapsari Raharjo

# Pengolahan Data Sensor Akselerometer MPU6050 pada eFeeder

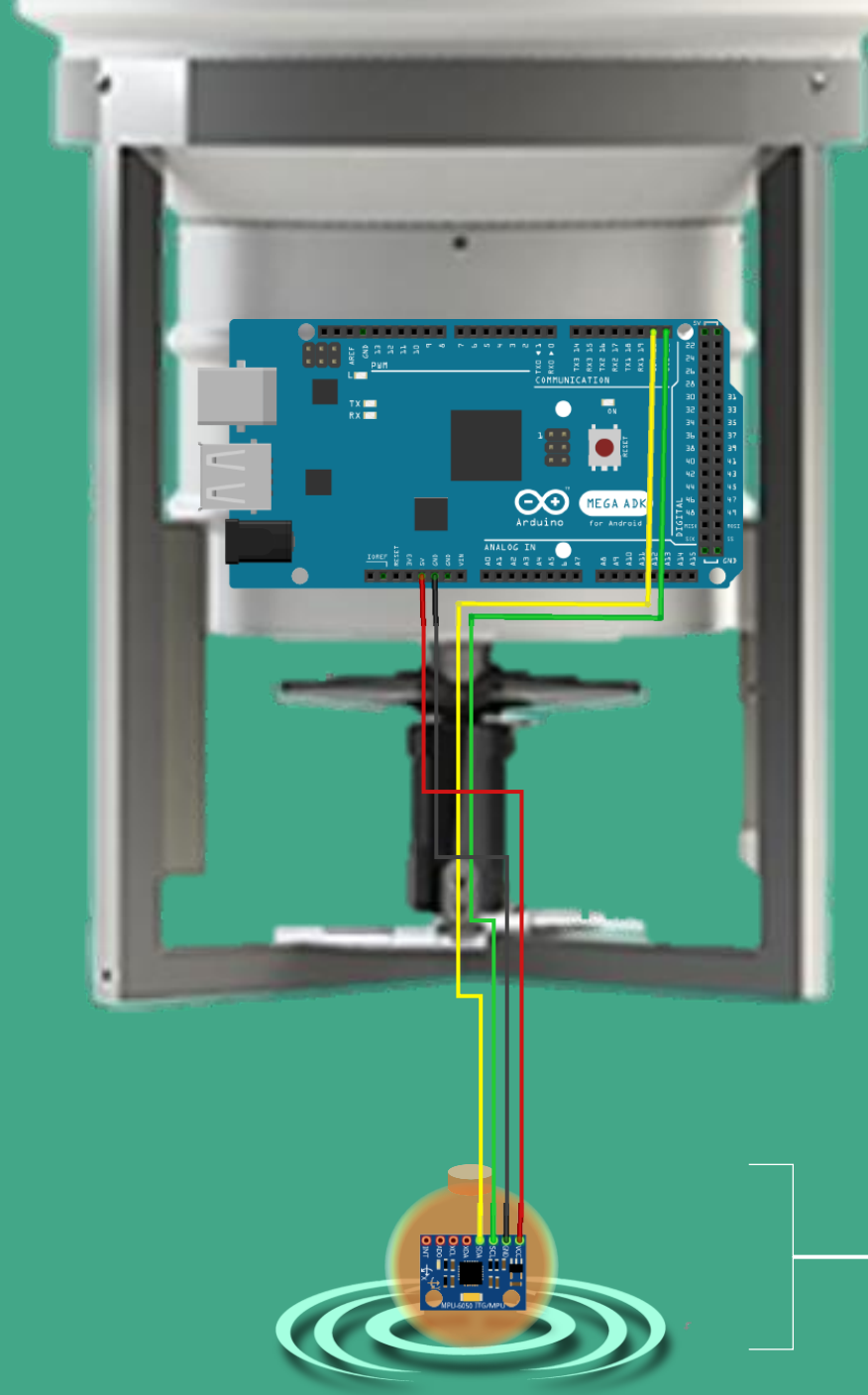
*A STUDY  
ON FINAL TEST eFishery DATA SCIENTIST CASE*



# Background

eFishery saat ini sedang mengembangkan sensor *auto Feeder*. Sensor ini terpasang dan mengirimkan sinyal aktivitas yang ada di kolam ikan secara *realtime* ke sistem.

Salah satu sensor yang digunakan adalah sensor *motion*.



Sensor motion yang digunakan adalah sensor *accelerometer and gyroscope motion*.

## **MPU6050 Sensor :**

- Accelerometer and Gyroscope motion sensor
- DoF = 6 Axes of sensing (3-axis accelerometer and 3-axis gyroscope sensor)
- Frequency up to 40Hz

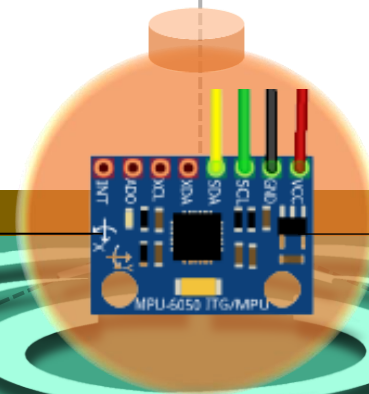
# Sensor Used for Data Collection

Gelombang air dari pergerakan ikan akan membuat sensor bergerak dan diubah dalam bentuk data akselerasi 3 sumbu.



Sensor dibiarkan mengapung pada permukaan air kolam.

**y**  
(up and down movement)



## **Sensor Motion MPU6050**

Frequency used = 25 Hz

DoF = 3 axis of accelerometer (X, Y, Z)

Data collection = Acceleration only

Data timestamp = Pseudo-Timestamp Data

**x**  
(left and right movement)

**z**  
(back and forth movement)

# Data Understanding

Data yang digunakan disimpan dalam google drive (*source* tertera) terbagi menjadi 7 file excel.

Kolom 'date' merupakan konversi dari data pseudo-timestamp pada data original.

Data original terdiri dari kolom "timestamp", "x", "y", "z", dan "label".

## Data Class

Nilai 0 : Ikan tidak sedang makan

Nilai 1 : Ikan sedang makan

Nilai tersebut tercantum pada kolom 'label' dan sudah dalam bentuk *numerical*.

*Data type* sudah dalam *numerical (float)* tidak ada yang masih berbentuk *object*.

	timestamp	x	y	z	label	date
0	1.657093e+12	-4.0	231.0	37.0	1.0	2022-07-06 07:40:02.000
1	1.657093e+12	7.0	232.0	33.0	1.0	2022-07-06 07:40:02.040
2	1.657093e+12	10.0	234.0	37.0	1.0	2022-07-06 07:40:02.080
3	1.657093e+12	4.0	242.0	43.0	1.0	2022-07-06 07:40:02.120
4	1.657093e+12	0.0	260.0	46.0	1.0	2022-07-06 07:40:02.160
...	...	...	...	...	...	...
28387	1.657094e+12	-1.0	243.0	42.0	0.0	2022-07-06 07:58:57.480
28388	1.657094e+12	-1.0	244.0	41.0	0.0	2022-07-06 07:58:57.520
28389	1.657094e+12	0.0	244.0	43.0	0.0	2022-07-06 07:58:57.560
28390	1.657094e+12	-2.0	246.0	44.0	0.0	2022-07-06 07:58:57.600
28391	1.657094e+12	-2.0	246.0	43.0	0.0	2022-07-06 07:58:57.640

28392 rows × 6 columns

Data Source :

[https://drive.google.com/file/d/1jIw6HjnFIHG5dDGafjCGed5nouBbKqRR/view?usp=share\\_link](https://drive.google.com/file/d/1jIw6HjnFIHG5dDGafjCGed5nouBbKqRR/view?usp=share_link)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28392 entries, 0 to 28391
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   timestamp    28392 non-null  float64
1   x            28392 non-null  float64
2   y            28392 non-null  float64
3   z            28392 non-null  float64
4   label        28392 non-null  float64
5   date         28392 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(5)
memory usage: 1.3 MB
```



# Data Understanding

Dalam data tidak ada *missing value (null)*. Dapat langsung diolah.

Data memiliki 6 *columns* dan 28392 *rows*.

Durasi pengambilan data terlihat dari kolom 'date' :

*Initial Time* = 2022-07-06 07:40:02

*End Time* = 2022-07-06 07:58:57.640000

Durasi pengambilan data = 00:18:55.640000

	timestamp	x	y	z	label	date
0	1.657093e+12	-4.0	231.0	37.0	1.0	2022-07-06 07:40:02.000
1	1.657093e+12	7.0	232.0	33.0	1.0	2022-07-06 07:40:02.040
2	1.657093e+12	10.0	234.0	37.0	1.0	2022-07-06 07:40:02.080
3	1.657093e+12	4.0	242.0	43.0	1.0	2022-07-06 07:40:02.120
4	1.657093e+12	0.0	260.0	46.0	1.0	2022-07-06 07:40:02.160
...	...	...	...	...	...	...
28387	1.657094e+12	-1.0	243.0	42.0	0.0	2022-07-06 07:58:57.480
28388	1.657094e+12	-1.0	244.0	41.0	0.0	2022-07-06 07:58:57.520
28389	1.657094e+12	0.0	244.0	43.0	0.0	2022-07-06 07:58:57.560
28390	1.657094e+12	-2.0	246.0	44.0	0.0	2022-07-06 07:58:57.600
28391	1.657094e+12	-2.0	246.0	43.0	0.0	2022-07-06 07:58:57.640

28392 rows × 6 columns

Data Source :

[https://drive.google.com/file/d/1jIw6HjnFIHG5dDGafjCGed5nouBbKqRR/view?usp=share\\_link](https://drive.google.com/file/d/1jIw6HjnFIHG5dDGafjCGed5nouBbKqRR/view?usp=share_link)

```
all_data.isnull().sum()
```

Data shape :

```
timestamp    0
```

(28392, 6)

```
x            0
```

```
y            0
```

```
z            0
```

```
label        0
```

```
date         0
```

```
dtype: int64
```

Durasi pengambilan data :

2022-07-06 07:40:02 - 2022-07-06 07:58:57.640000

0 days 00:18:55.640000



## Durasi Ikan Makan

fish eat duration :

0 days 00:01:43.960000

0 days 00:01:31.480000

0 days 00:01:14.840000

0 days 00:01:50.200000

0 days 00:01:12.760000

0 days 00:00:58.200000

0 days 00:00:56.120000

average eating duration : 0 days 00:01:21.080000

## Durasi Ikan Tidak Makan

fish not eat duration :

0 days 00:01:43.960000

0 days 00:01:31.480000

0 days 00:01:14.840000

0 days 00:01:50.200000

0 days 00:01:12.760000

0 days 00:00:58.200000

0 days 00:00:56.120000

average not eating duration : 0 days 00:01:21.080000

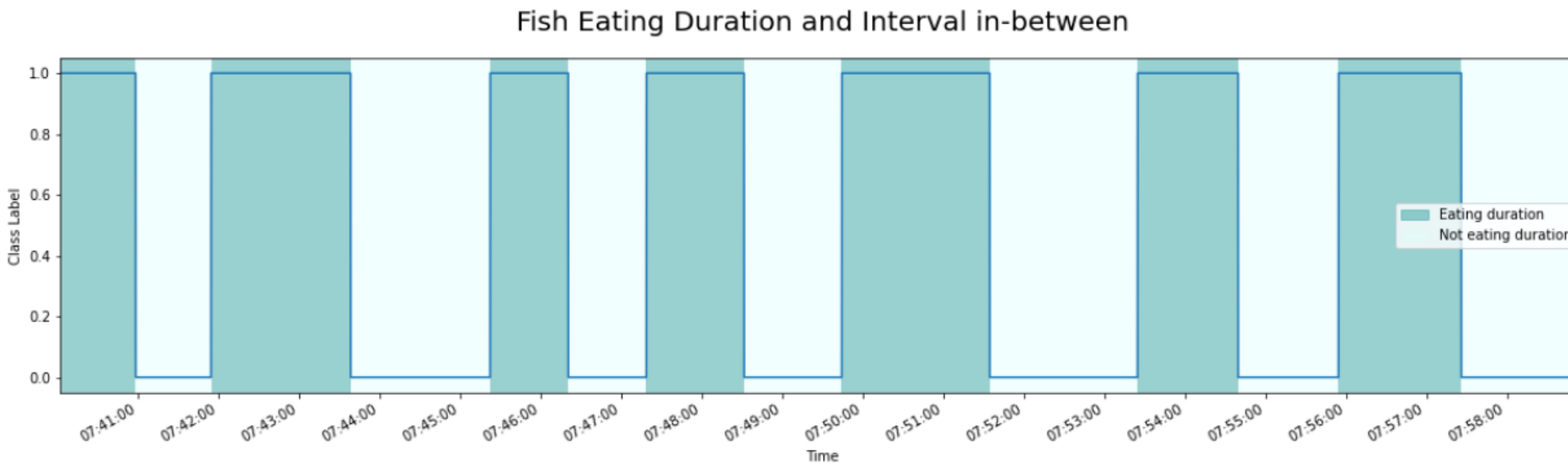
Dari hasil pengolahan data berdasarkan timestamp dengan label class 1.0 yaitu “ikan sedang makan”, didapatkan durasi makan ikan dari masing – masing file excel seperti di atas. Durasi makan ikan berbeda – beda dalam rentang 56 detik hingga 1 menit 43 detik. Dari total waktu yang dibutuhkan ikan untuk makan, rata – rata durasi makan ikan adalah 1 menit 21 detik.

Setelah dilakukan pengolahan data untuk timestamp dengan label class 0.0 yaitu “ikan tidak sedang makan”, didapatkan durasi “ikan tidak sedang makan” sama dengan durasi “ikan sedang makan”. Hal ini berarti ikan memiliki waktu interval antar kegiatan makan yang sama durasinya dengan durasi makan terakhir. Contoh jika kegiatan makan ikan terakhir berdurasi 1 menit 43 detik, maka kegiatan makan berikutnya akan terjadi setelah 1 menit 43 detik.

# Durasi Makan Ikan







Durasi pengambilan data :  
2022-07-06 07:40:02 - 2022-07-06 07:58:57.640000  
0 days 00:18:55.640000

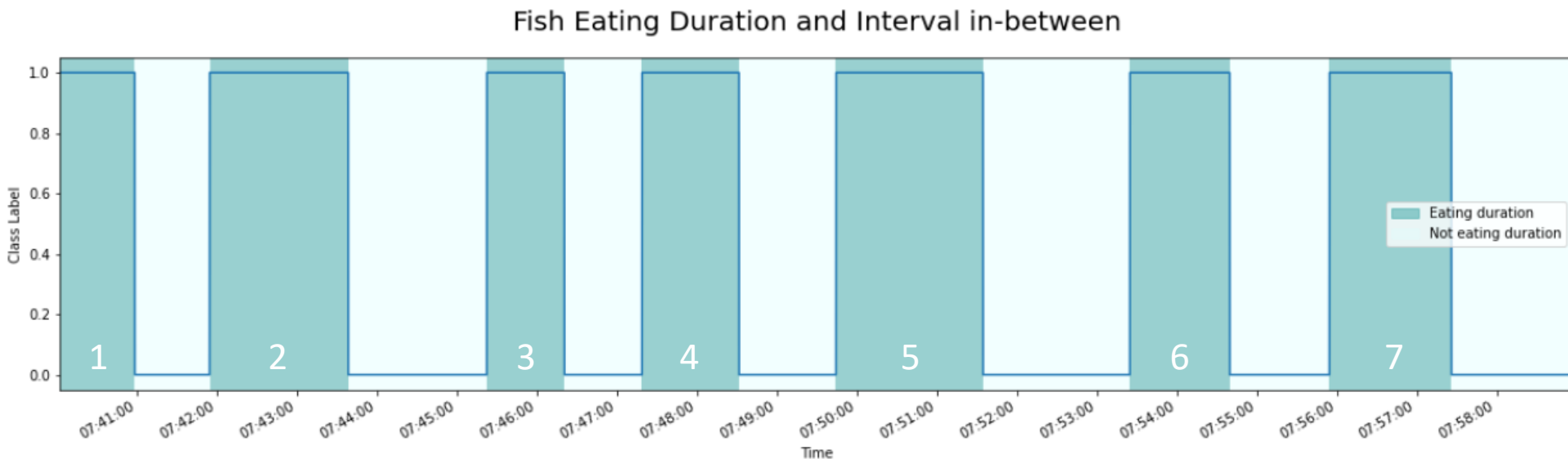
Setelah dilakukan penggabungan file excel menjadi satu dataframe, didapatkan bahwa timestamp yang ada terjadi secara berkesinambungan dan membentuk urutan waktu yang utuh. Dari pengolahan data tersebut, didapatkan total durasi pengambilan data adalah selama 18 menit 55 detik dimulai dari pukul 07:40:02 hingga pukul 07:58:57.

Grafik di atas, menunjukkan kegiatan makan ikan berdasarkan waktu. Area yang lebih gelap pada grafik menunjukkan “ikan sedang makan” dan area yang terang menunjukkan “ikan tidak sedang makan”. Dapat dilihat dari durasi yang ditunjukkan oleh area gelap terang, bahwa semakin lama durasi makan yang dibutuhkan ikan, maka durasi tidak makan setelahnya pun akan semakin lama, begitupun sebaliknya.

# Durasi Makan Ikan







Durasi pengambilan data :

2022-07-06 07:40:02 - 2022-07-06 07:58:57.640000

0 days 00:18:55.640000

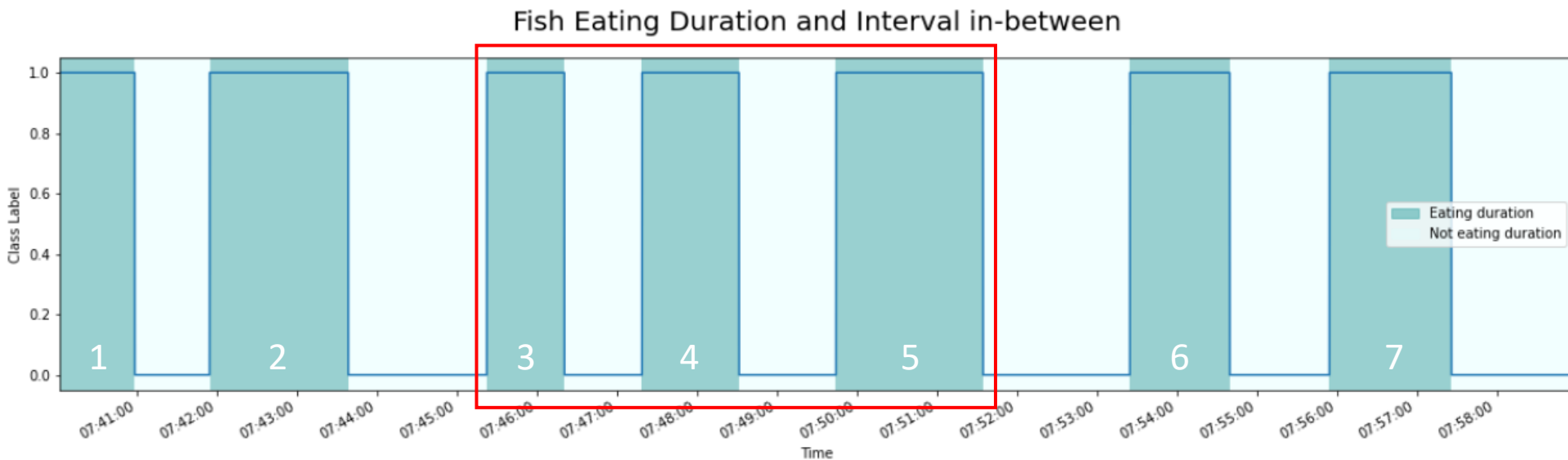
Dari durasi makan dan interval antar makan yang ada, dapat diperkirakan bagaimana jadwal pemberian pakan yang efektif dengan asumsi pakan yang diberikan adalah sebanyak 5 Kg pada setiap *event* makan.

Pertama dapat dilihat dari durasi makan yang berbeda – beda, hal ini menunjukkan respon ikan terhadap pakan yang diberikan. Semakin sempit durasi makan yang terjadi, berarti kondisi ikan sangat lapar dan semakin cepat 5 Kg pakan yang diberikan habis, seperti yang ditunjukkan oleh area 3 dan 6, dan semakin lama durasi makan, maka kondisi ikan tidak terlalu lapar dan semakin lama juga pakan yang diberikan habis, seperti yang ditunjukkan oleh area 2 dan 5.

Tetapi, dari grafik tersebut dapat terlihat ada *event* transisi dari *event* berdurasi cepat ke lambat, ditunjukkan oleh area 4 dan 7. Area ini tidak seluas area 2 dan 5, tetapi tidak sesempit area 3. Sehingga dapat disimpulkan area 3, 4, 5 merupakan *event* makan dari kondisi ikan sangat lapar, lapar, dan hampir kenyang. Hingga akhirnya pola ini pun muncul kembali setelah *time gap*.

# Durasi Makan Ikan





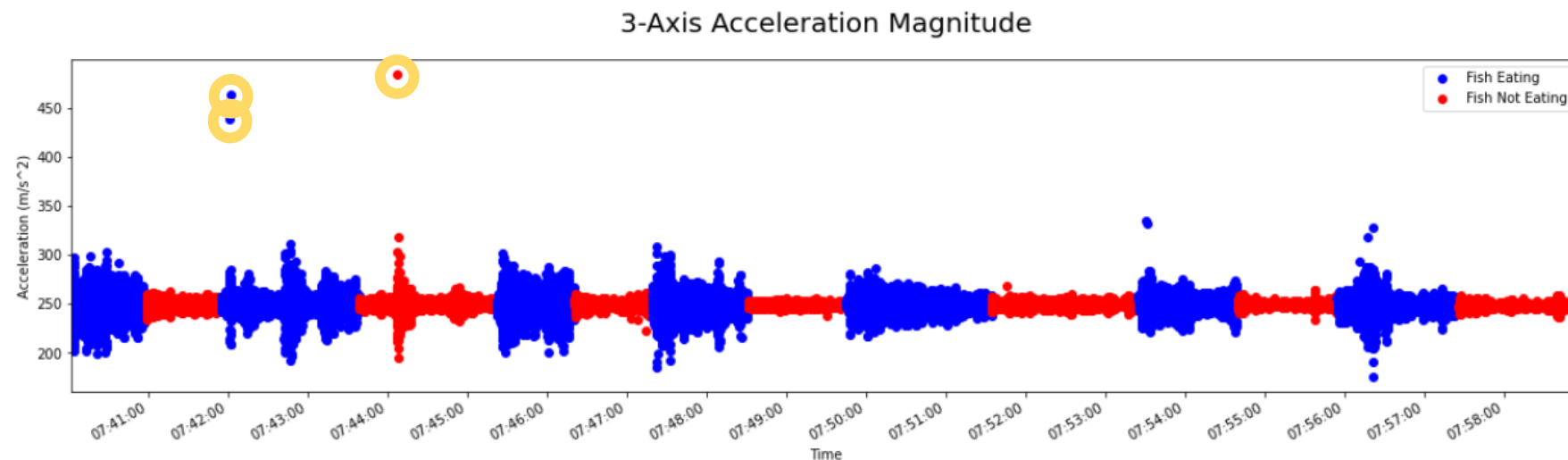
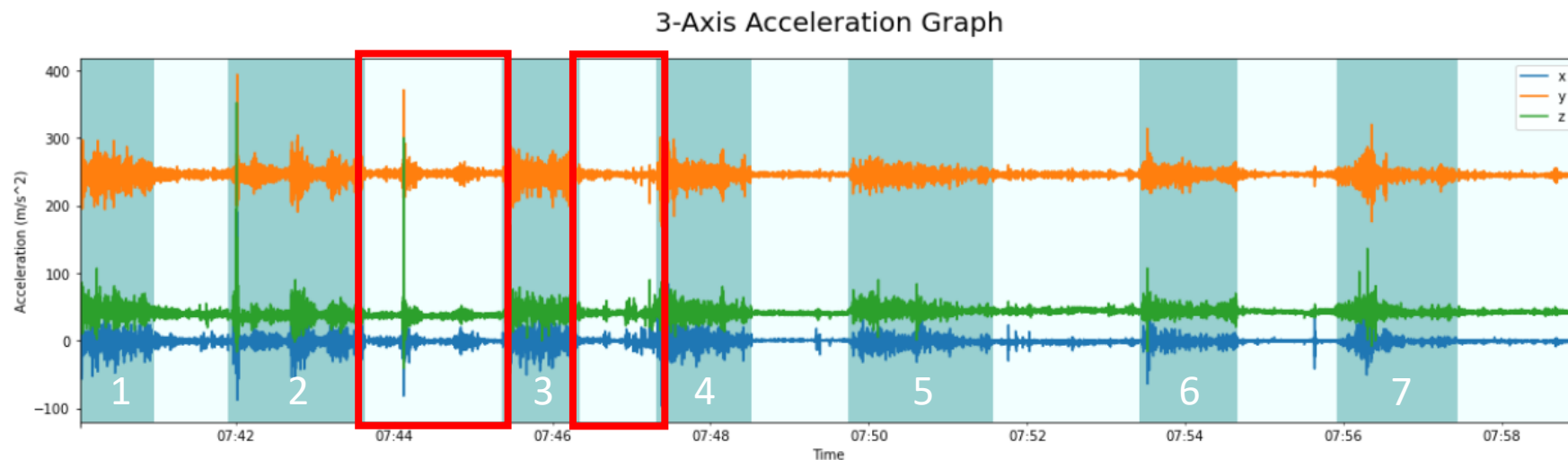
# Durasi Makan Ikan



Durasi pengambilan data :  
2022-07-06 07:40:02 - 2022-07-06 07:58:57.640000  
0 days 00:18:55.640000

Apabila penjadwalan pemberian pakan didasarkan atas data yang diberikan, dapat mengikuti pola ini (ditandai oleh kotak berwarna merah). Pola dimana pada saat ikan sangat lapar, dibutuhkan 3 kali event makan sebelum akhirnya ikan merasa kenyang dan terdapat *time gap* yang cukup lama sebelum akhirnya masuk ke pola selanjutnya yang akan berulang.

Dari data tersebut pun diketahui waktu tenang ikan setiap setelah *event* makan sama dengan durasi makan sebelumnya, sehingga penjadwalan makan dapat didasarkan dari waktu tersebut. Misalkan pada event makan di area 3 ikan membutuhkan waktu 1 menit untuk menghabiskan pakan dari pukul 07:46:30, karena *time gap* sama dengan durasi makan sebelumnya, maka eFeeder harus menjadwalkan pemberian pakan selanjutnya di pukul 07:47:30, 1 menit setelah waktu makan sebelumnya berakhir. Begitu seterusnya untuk *event* makan selanjutnya.

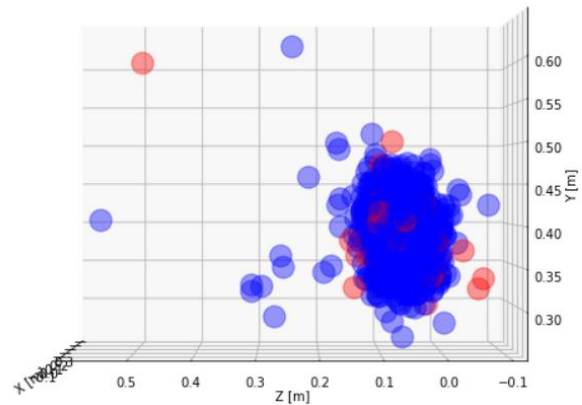
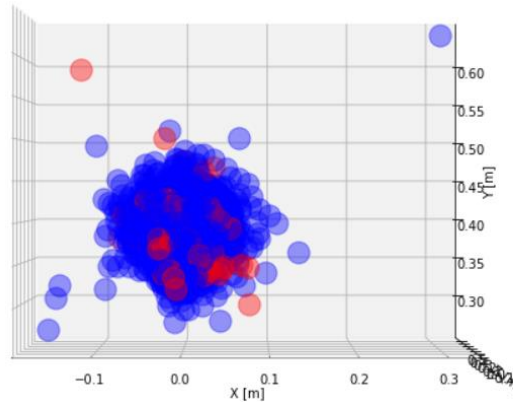
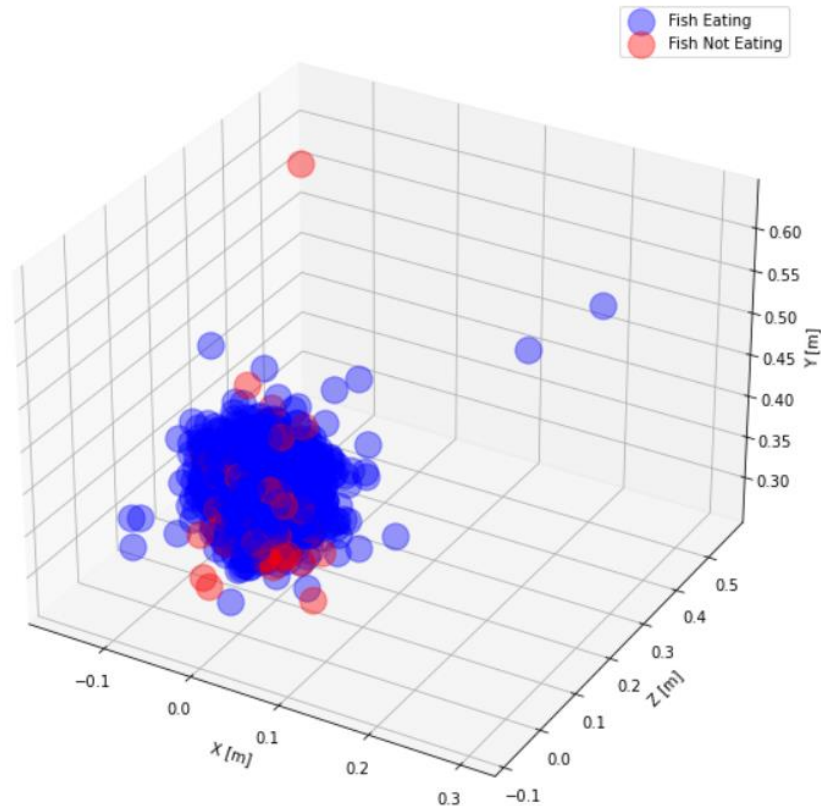


# Data Eksplorasi

eFishery

Berdasarkan grafik di atas, dapat dilihat adanya akselerasi x, y, z pada saat *event* makan yang secara signifikan berubah-ubah. Hal ini menunjukkan memang ikan bergerak secara masif yang terbaca oleh sensor dan membuat sensor mengalami peningkatan akselerasi. Tetapi terdapat peningkatan nilai *acceleration* pada saat ikan sedang tidak makan yang tidak mengikuti pola grafik yang lainnya (ditandai oleh kotak merah) dan beberapa *magnitude* akselerasi terbaca sangat tinggi nilainya (ditandai oleh lingkaran kuning) baik dari saat kondisi ikan sedang makan maupun saat ikan tidak makan, mengikuti nilai *acceleration* yang tinggi dari ketiga sumbu pada grafik di atasnya.

Sensor Position Based On Acceleration Data



Sensor Position

# Data Eksplorasi

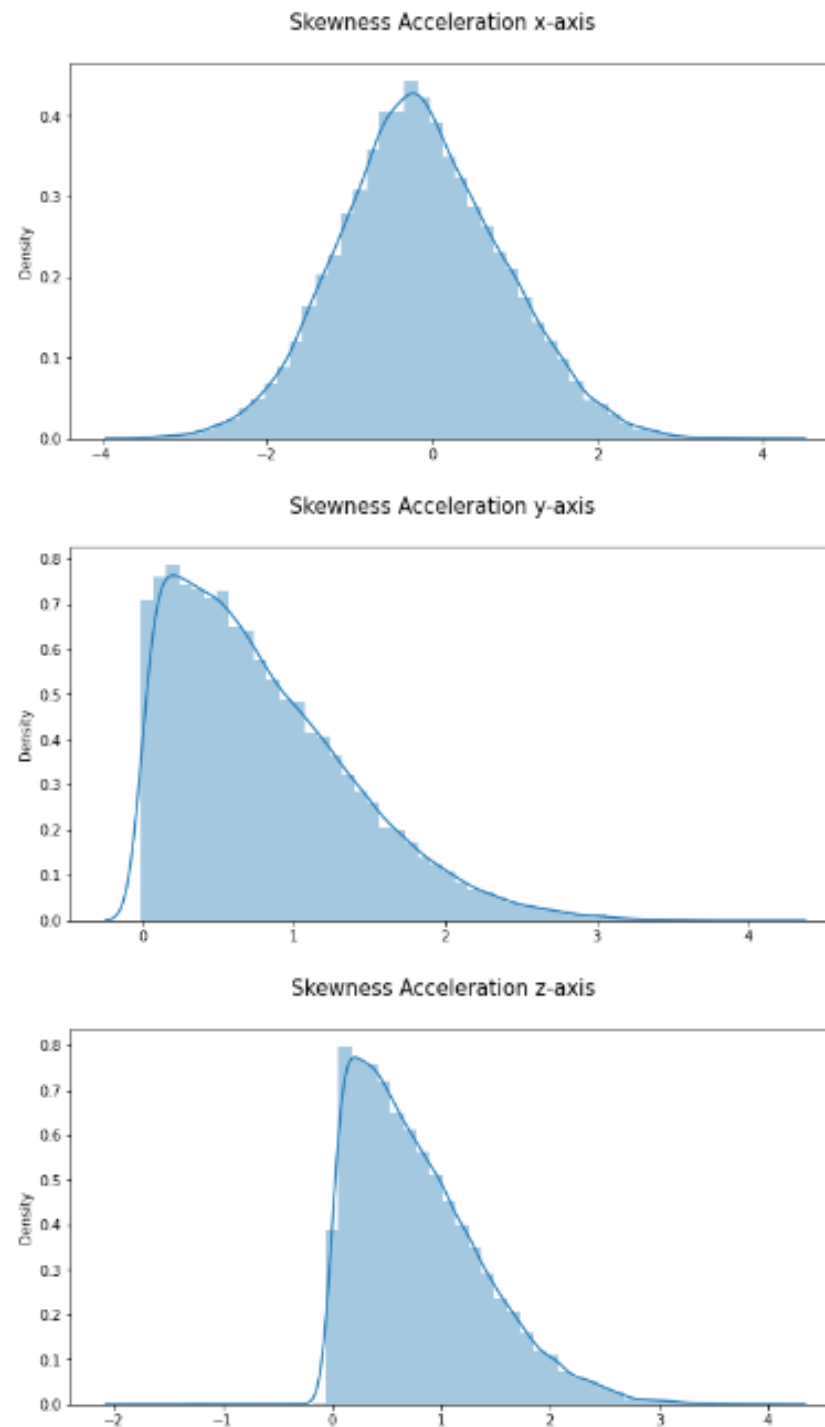
Posisi sensor didapatkan dari konversi akselerasi terhadap waktu dan disajikan pada grafik 3D di atas. Dari grafik tersebut terlihat bahwa memang posisi sensor pada saat *event* makan lebih berubah-ubah dibandingkan pada saat ikan tidak sedang makan. Hal ini dapat terlihat dari *dots* biru yang menutupi posisi *dots* merah sehingga tidak terlihat di grafik.

Posisi sensor ini menunjukkan bahwa sensor bergerak secara rata dari ketiga sumbu dan tidak condong ke arah satu sumbu saja. Berdasarkan representasi tersebut dapat dibayangkan kondisi sensor di kolam lebih terombang ambing dan tersundul oleh ikan pada saat *event* makan dibandingkan dengan saat kondisi ikan sedang tidak makan.

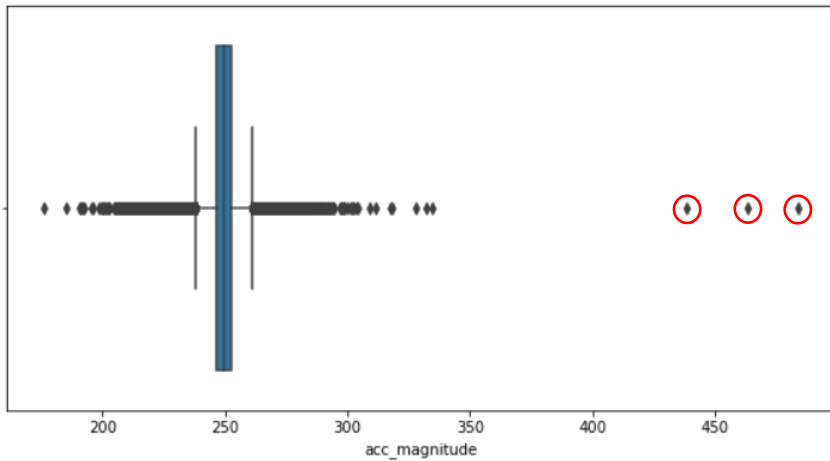
eFishery

Dari grafik *skewness* yang sudah dinormalisasi di samping, menguatkan bukti bahwa memang *acceleration* pada sumbu x lebih simetris dibandingkan *acceleration* yang terjadi pada sumbu y dan z. Dari grafik *acceleration* pada halaman sebelumnya terlihat bahwa ada nilai akselerasi pada sumbu y dan z yang sangat besar sehingga pada grafik terlihat sangat jauh dari rata-rata nilai akselerasinya. Dan memang akselerasi pada sumbu y dan z ini mayoritas bernilai positif dan lebih banyak data ke arah atas dibandingkan ke arah bawah mendekati 0.

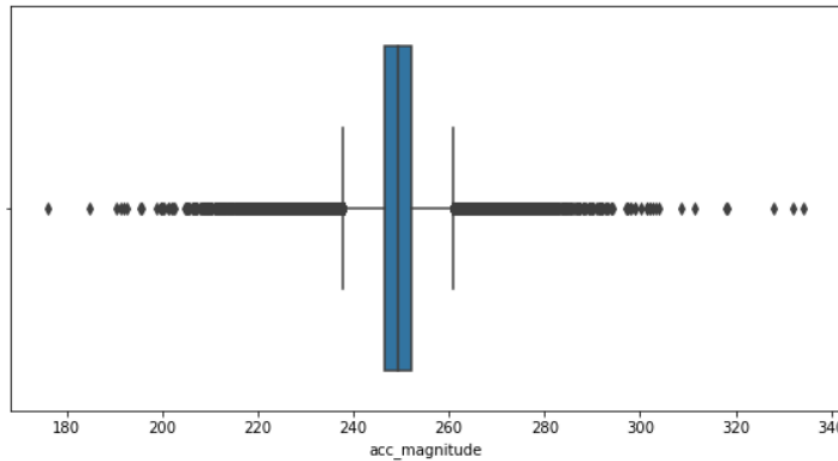
*Skewness* pada sumbu y dan z lebih condong ke kanan yang berarti bernilai *skew* positif dengan hanya ada sedikit data yang ada pada area kurang dari nilai 0. *Skewness* pada sumbu y dan z ini pun lebih cenderung berkumpul di satu *range* saja, terlihat bahwa *density* dari 2 sumbu ini mencapai 0,8 yang berarti data pada kedua sumbu ini range nilai datanya tidak terlalu lebar dan modus yang muncul sangat banyak di *range* yang sangat sempit, sehingga grafik *skewness*nya terlihat lebih menumpuk ke arah vertikal dibandingkan terdistribusi secara horizontal.



Before Cleaning :



After Cleaning :



Terdapat 3 nilai akselerasi yang berbeda sangat signifikan dibandingkan data yang lainnya yang dapat dilihat pada *boxplot before cleaning* (ditandai oleh lingkaran merah). Maka dibutuhkan *cleaning* data untuk membersihkan data yang terlalu berbeda jauh untuk meningkatkan akurasi pada saat pemodelan.

*Boxplot* setelah *Cleaning* data disajikan di sebelah kanan pada *boxplot after cleaning*. Terlihat 3 nilai akselerasi yang melebihi  $350\text{m/s}^2$  sudah tidak ada. Data *after cleaning* inilah yang akan digunakan untuk pemodelan selanjutnya.

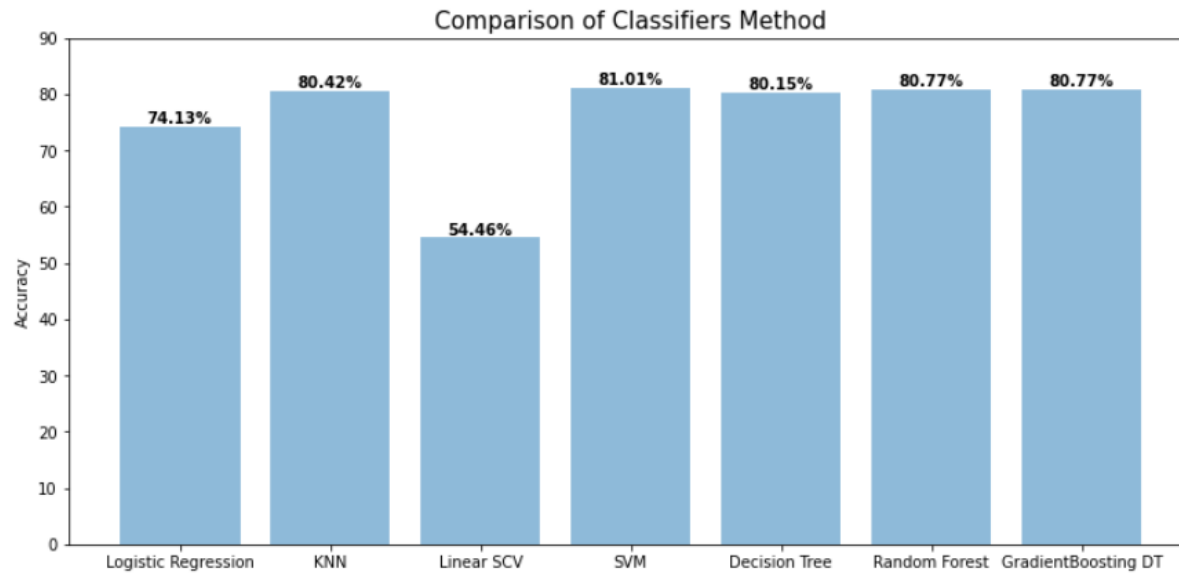
# Data Eksplorasi

eFishery



X\_train dimension : 22711  
y\_train dimension : 22711  
X\_test dimension : 5678  
y\_test dimension : 5678

	Accuracy	Error
Logistic Regression	74.13%	25.87%
KNN	80.42%	19.58%
Linear SVC	54.46%	45.54%
rbf SVM classifier	81.01%	18.99%
DecisionTree	80.15%	19.85%
Random Forest	80.77%	19.23%
GradientBoosting DT	80.77%	19.23%



Data *accelerometer* pada sumbu x, y, z dilakukan pemodelan menggunakan pemodelan *Logistic Regression*, *KNN*, *Linear SCV*, *rbf SVM classifier*, *DecisionTree*, *Random Forest*, *GradientBoosting DT*. Dari semua metode yang digunakan, akurasi tertinggi dihasilkan oleh pemodelan SVM dengan akurasi sebesar 81,01%. Nilai akurasi model sebesar 81,01% sudah termasuk nilai yang bagus untuk digunakan sebagai dasar pengklasifikasian data.

Pemodelan ini dilakukan dengan membagi data 80:20 untuk *training* dan *test dataset* sebanyak 22711 *train data* dan 5678 *test data*. Model dapat dengan 81,01% akurat mengklasifikasikan *test dataset* terhadap *event* makan yang terjadi berdasarkan data akselerasi dibandingkan dengan label sebenarnya.

## Data Eksplorasi





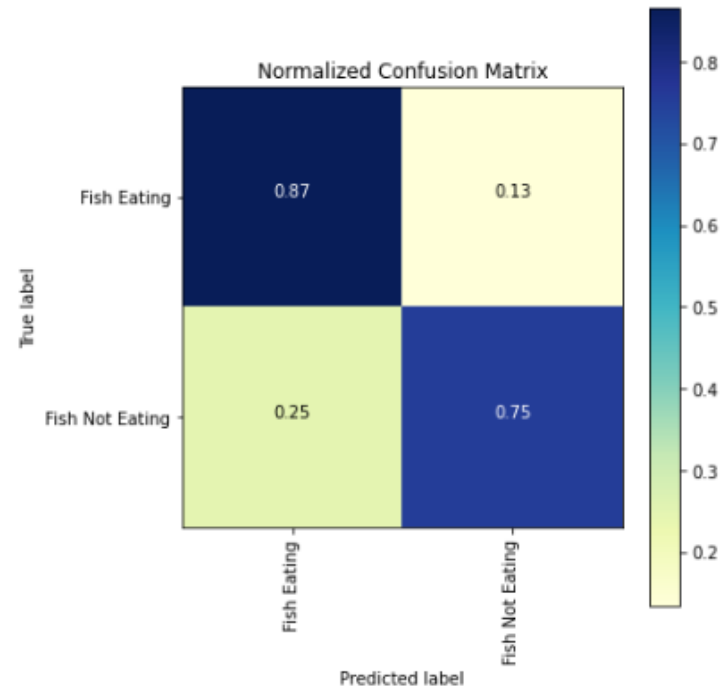
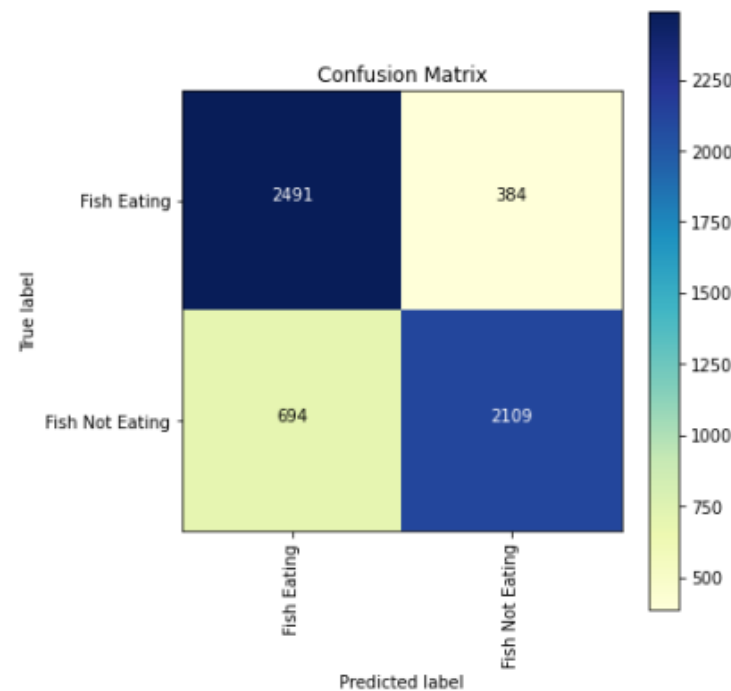
## SVM Classification Report and Confusion Matrix

Classification Report				
	precision	recall	f1-score	support
0.0	0.78	0.87	0.82	2875
1.0	0.85	0.75	0.80	2803
accuracy			0.81	5678
macro avg	0.81	0.81	0.81	5678
weighted avg	0.81	0.81	0.81	5678

Pemodelan menggunakan metode SVM menghasilkan akurasi tertinggi sebesar 81,01% yang tertulis 0,81 pada *classification report*.

Dari total sebanyak 5678 data yang dilakukan *test* pemodelan menggunakan metode SVM, terdapat 2491 *True Positive* data berlabelkan 'Fish Eating', 2109 *True Negative* data berlabelkan 'Fish Not Eating', 384 *False Positive* data berlabelkan 'Fish Eating', dan 694 *False Negative* data berlabelkan 'Fish Not Eating'.

Pelabelan data yang tidak sesuai ini disebabkan oleh nilai akselerasi yang bias yang tidak sesuai dengan pola yang dibaca dan dijadikan dasar oleh model. Seperti pada grafik akselerasi yang sebelumnya ditampilkan, terdapat pola pergerakan sensor pada saat ikan tidak sedang makan, sehingga ada kemungkinan pergerakan itu terbaca oleh model sebagai *event* makan.



## SVM Modelling

# Data Eksplorasi

eFishery

**Thank You,**  
**eFishery**

*Sincerely Yours,  
Mutiara Kusuma Hapsari Raharjo*



[https://gitlab.com/mutiara.kusuma.hapsari.raharjo/Final\\_Test\\_eFishery/-  
/blob/10f0d10af7d5428250a48ef6c5d6db8d65b922c2/final\\_test\\_efishery  
mutiara\\_khr.ipynb](https://gitlab.com/mutiara.kusuma.hapsari.raharjo/Final_Test_eFishery/-/blob/10f0d10af7d5428250a48ef6c5d6db8d65b922c2/final_test_efishery_mutiara_khr.ipynb)

