# JALA

## Shrimp Analysis


M. Masdar Mahasin

```
Analisis Kelengkapan Data: Cycles
Total baris: 2617

Kolom dengan data yang hilang:
species_id: 814 baris (31.10%)
finished_at: 1 baris (0.04%)
remark: 1281 baris (48.95%)
initial_age: 48 baris (1.83%)
limit_weight_per_area: 7 baris (0.27%)
target_cultivation_day: 3 baris (0.11%)
target_size: 4 baris (0.15%)
ordered_at: 1523 baris (58.20%)
hatchery_id: 465 baris (17.77%)
total_seed_type: 242 baris (9.25%)
hatchery_name: 465 baris (17.77%)
pond_length: 6 baris (0.23%)
pond_width: 6 baris (0.23%)
pond_depth: 118 baris (4.51%)
```

```
Analisis Kelengkapan Data: Harvests
Total baris: 8087

Kolom dengan data yang hilang:
status: 263 baris (3.25%)
selling_price: 1793 baris (22.17%)

Informasi tambahan:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8087 entries, 0 to 8086
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   cycle_id       8087 non-null   float64
 1   updated_at     8087 non-null   object
 2   size           8087 non-null   float64
 3   created_at     8087 non-null   object
 4   weight         8087 non-null   float64
 5   id             8087 non-null   float64
 6   harvested_at   8087 non-null   object
 7   status         7824 non-null   object
 8   selling_price  6294 non-null   float64
dtypes: float64(5), object(4)
memory usage: 568.7+ KB
```

```
Analisis Kelengkapan Data: Mortalities
Total baris: 13221

Kolom dengan data yang hilang:

Informasi tambahan:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13221 entries, 0 to 13220
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              13221 non-null  int64
 1   cycle_id        13221 non-null  int64
 2   quantity        13221 non-null  int64
 3   recorded_at     13221 non-null  object
 4   created_at      13221 non-null  object
 5   updated_at      13221 non-null  object
 6   average_weight  13221 non-null  float64
dtypes: float64(1), int64(3), object(3)
memory usage: 723.2+ KB
```

Formula:

SR = (nP / nT) * 100%

Dimana:
- SR = Survival Rate (%)
- nP = Jumlah udang yang dipanen
- nT = Jumlah udang yang ditebar awal

| id | total_seed | total_harvested | survival_rate |
|---|---|---|---|
| 3458 | 566669 | 444548.02 | 78.449328 |
| 3459 | 566669 | 440387.88 | 77.715188 |
| 4036 | 172250 | 154350.00 | 89.608128 |
| 4038 | 350000 | 405078.84 | 115.736811 |
| 4039 | 210000 | 219808.62 | 104.670771 |
| ... | ... | ... | ... |
| 29619 | 70000 | 137250.00 | 196.071429 |
| 29659 | 75000 | 12000.00 | 16.000000 |
| 29679 | 26671 | 32187.00 | 120.681639 |
| 29873 | 125000 | 88230.00 | 70.584000 |
| 29874 | 125000 | 46305.00 | 37.044000 |

Survival Rate (SR) adalah persentase udang yang bertahan hidup dari awal penebaran hingga panen. Formula ini membandingkan jumlah udang yang berhasil dipanen dengan jumlah udang yang ditebar di awal siklus budidaya, kemudian dikalikan 100% untuk mendapatkan persentase. Semakin tinggi nilai SR, semakin baik tingkat kelangsungan hidup udang selama masa budidaya.

## Survival Rate Calculation

```
# Analisis SR detail : Data harvested tidak ada
# Cycle ID : 22269
detail_hasil = detail_sr(22269)
```

✓ 0.1s

```
Total seed : 362712
size and weight :
      size  weight
454  588.0     0.0
Total harvested shrimp : 0.0
Total mortality : 0
Survival rate : 0.00%
```

## Data Anomali

```
# Analisis SR detail : SR lebih dari 100%
# Cycle ID : 29619
detail_hasil = detail_sr(29619)
```

✓ 0.1s

```
Total seed : 70000
size and weight :
       size  weight
7828  215.0   150.0
7829  150.0   700.0
Total harvested shrimp : 137250.0
Total mortality : 0
Survival rate : 196.07%
```

Langkah-langkah pra-pemrosesan:
1. Hapus baris data jika:
   - total_harvested = 0
   - total_harvested = NaN

2. Batasi nilai survival_rate:
   - Jika survival_rate > 100, ubah menjadi 100. Hal ini dilakukan dengan asumsi bahwa Survival Rate maksimal adalah 100%

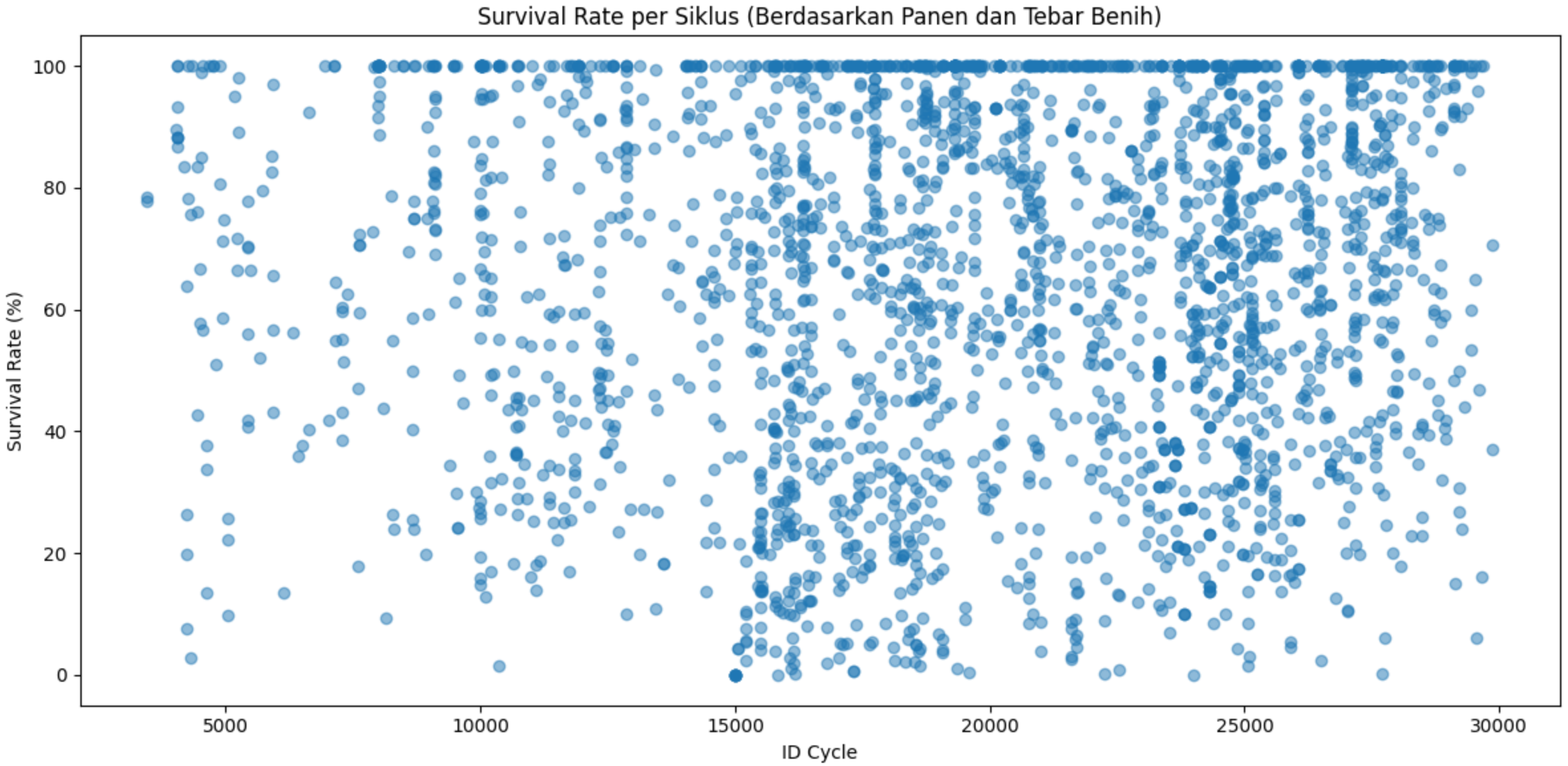
```
<class 'pandas.core.frame.DataFrame'>
Index: 2595 entries, 1939 to 1253
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               2595 non-null   int64
 1   total_seed       2595 non-null   int64
 2   total_harvested  2595 non-null   float64
 3   survival_rate    2595 non-null   float64
dtypes: float64(2), int64(2)
memory usage: 101.4 KB
```

| | id | total_seed | total_harvested | survival_rate |
|---|---|---|---|---|
| 1939 | 3458 | 566669 | 444548.02 | 78.449328 |
| 2277 | 3459 | 566669 | 440387.88 | 77.715188 |
| 1371 | 4036 | 172250 | 154350.00 | 89.608128 |
| 152 | 4038 | 350000 | 405078.84 | 100.000000 |
| 1301 | 4039 | 210000 | 219808.62 | 100.000000 |
| ... | ... | ... | ... | ... |
| 1785 | 29619 | 70000 | 137250.00 | 100.000000 |
| 424 | 29659 | 75000 | 12000.00 | 16.000000 |
| 2468 | 29679 | 26671 | 32187.00 | 100.000000 |
| 2076 | 29873 | 125000 | 88230.00 | 70.584000 |
| 1253 | 29874 | 125000 | 46305.00 | 37.044000 |

[2595 rows x 4 columns]

# Survival Rate Calculation

## Chart



Ada dataset mortality yang menunjukkan total udang mati, dari data ini kita bisa menganalisis Survival Rate juga.

Formula:

$$SR = ((nT - nM) / nT) * 100\%$$

Dimana:

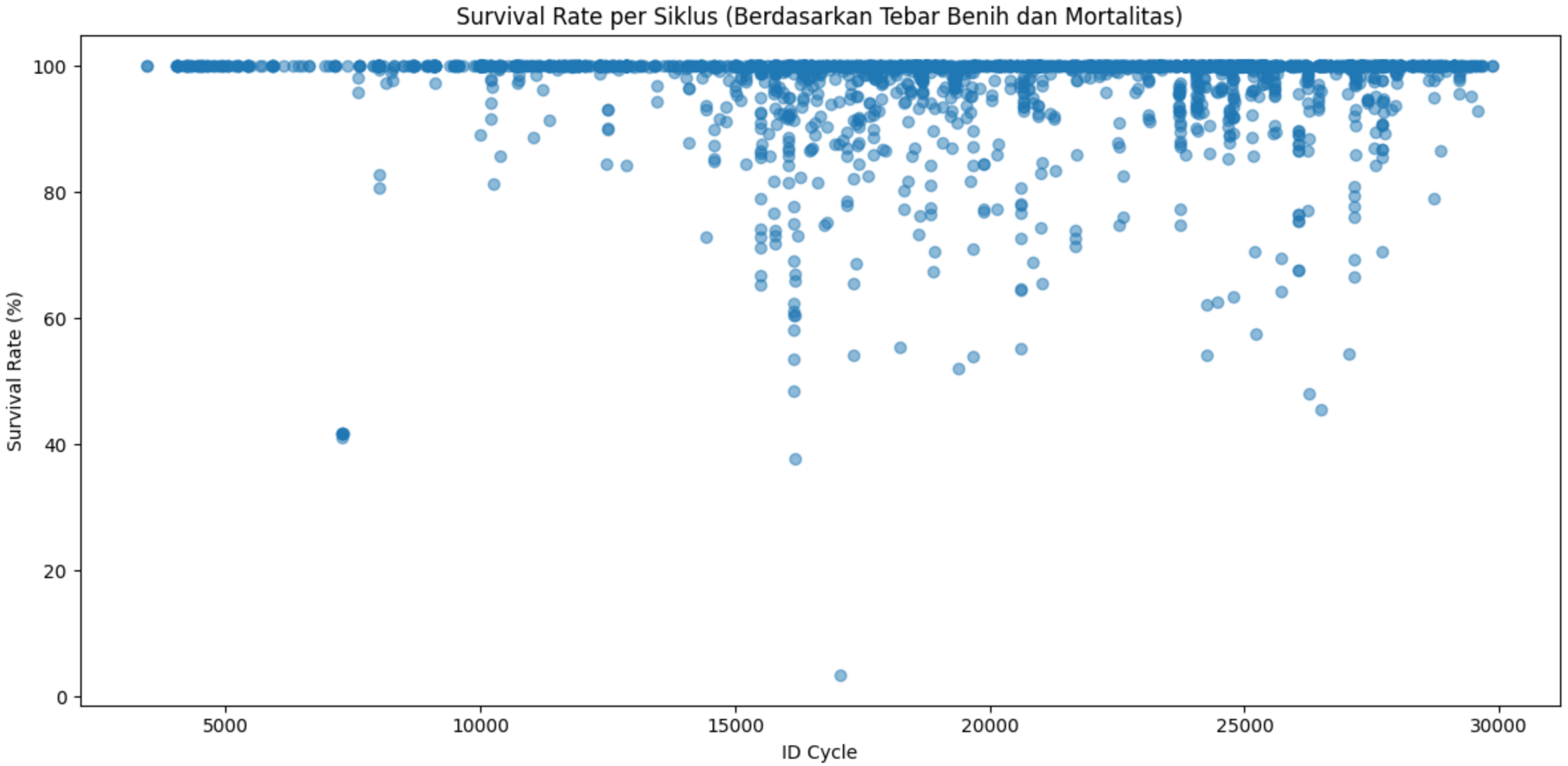
- SR = Survival Rate (%)
- nT = Jumlah udang yang ditebar awal
- nM = Jumlah udang yang mati (mortality)

Formula ini menghitung selisih antara jumlah udang yang ditebar di awal dengan jumlah udang yang mati selama masa budidaya, dibagi dengan jumlah tebar awal, kemudian dikalikan 100% untuk mendapatkan persentase. Metode ini menggunakan data mortalitas langsung, yang dapat memberikan estimasi survival rate yang lebih akurat sepanjang siklus budidaya. Semakin tinggi nilai SR, semakin baik tingkat kelangsungan hidup udang selama masa budidaya.

Perbedaan utama dengan metode sebelumnya adalah penggunaan data mortalitas langsung alih-alih data panen, yang memungkinkan pemantauan survival rate secara real-time selama siklus budidaya berlangsung.

| | id | total_seed | total_mortalities | survival_rate |
|---|---|---|---|---|
| count | 2609.000000 | 2.609000e+03 | 2609.000000 | 2609.000000 |
| mean | 19908.062093 | 2.201293e+05 | 5272.898045 | 97.465893 |
| std | 6013.475991 | 1.709823e+05 | 17692.657033 | 7.266867 |
| min | 3458.000000 | 1.000000e+02 | 0.000000 | 3.453370 |
| 25% | 16091.000000 | 9.600000e+04 | 0.000000 | 99.065656 |
| 50% | 20408.000000 | 1.956700e+05 | 0.000000 | 100.000000 |
| 75% | 24814.000000 | 3.000000e+05 | 1429.000000 | 100.000000 |
| max | 29874.000000 | 1.800000e+06 | 431324.000000 | 100.000000 |

Metode ini memiliki kelemahan, yaitu adanya indikasi data yang dilaporkan tidak akurat dan lengkap. Sehingga untuk saat ini, metode perhitungan Survival Rate melalui mortality kurang tepat untuk digunakan sebagai acuan.


Survival Rate per Siklus (Berdasarkan Tebar Benih dan Mortalitas)
Survival Rate (%)
100
80
60
40
20
0
5000
10000
15000
20000
25000
30000
ID Cycle

# Average Daily Gain (ADG)

Average Daily Gain (ADG) adalah metrik penting dalam budidaya udang yang mengukur pertambahan berat harian rata-rata. ADG memberikan informasi berharga tentang laju pertumbuhan udang selama periode tertentu.

Untuk menghitung ADG, kita perlu terlebih dahulu menghitung Average Body Weight (ABW):

1. Formula ABW:

   ABW = Total berat udang (gram) / Jumlah udang sampel

2. Formula ADG:

   ADG = (ABW akhir - ABW awal) / Jumlah hari

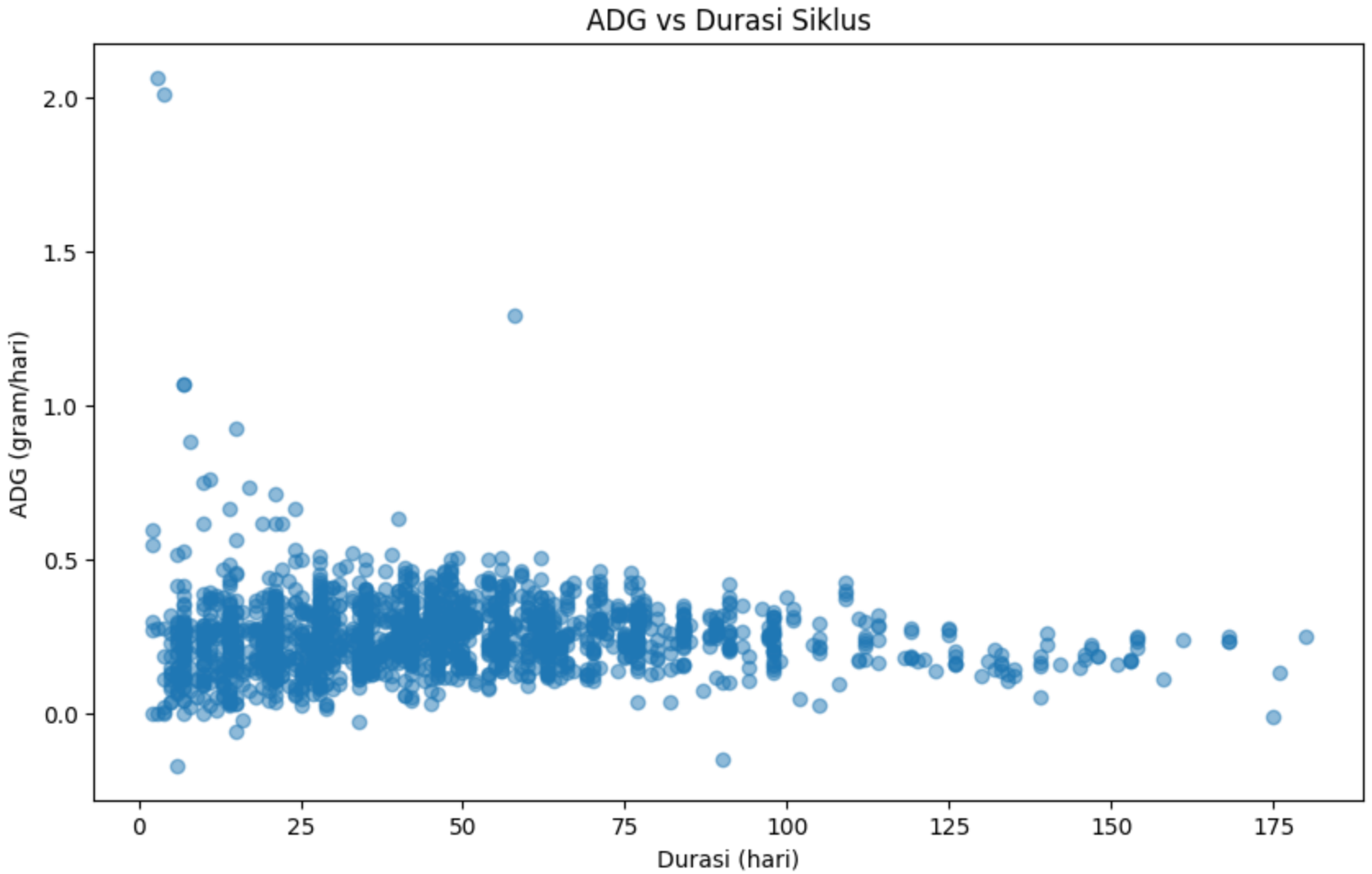
   Di mana:
   - ADG: Average Daily Gain (Pertambahan Berat Harian Rata-rata)
   - ABW akhir: Berat badan rata-rata pada akhir periode
   - ABW awal: Berat badan rata-rata pada awal periode
   - Jumlah hari: Selisih hari antara pengukuran awal dan akhir

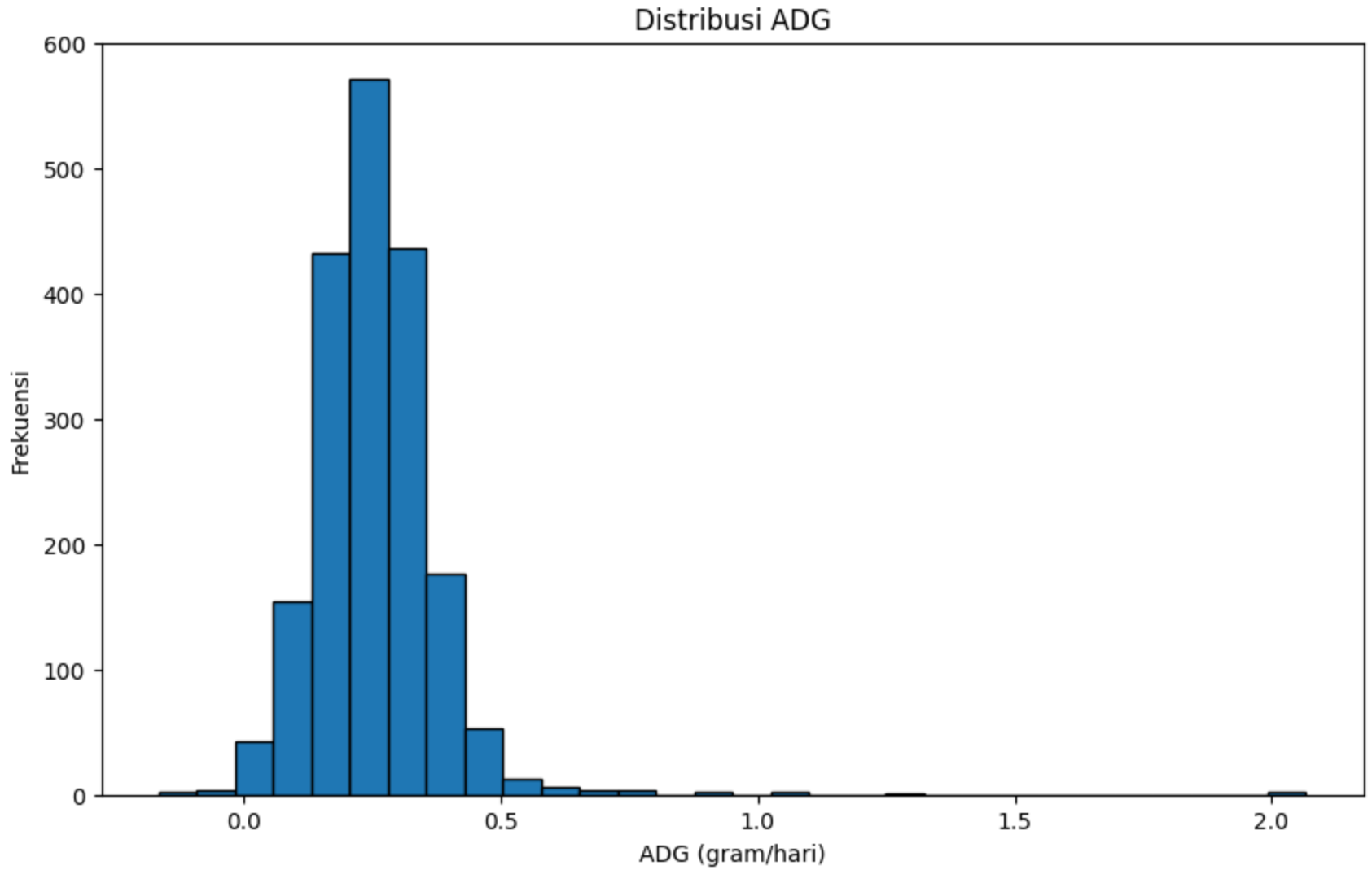
```
Outliers (duration_days > 200):
           cycle_id       ADG start_date   end_date  start_weight  end_weight  \
cycle_id
6126.0       6126.0  0.026316 2020-11-05 2021-06-02          3.50        9.00
18760.0     18760.0  0.034065 2020-01-26 2021-03-02          8.86       22.52
25134.0     25134.0  0.001425 2001-01-01 2023-11-11          5.72       17.62

           duration_days
cycle_id
6126.0               209
18760.0              401
25134.0             8349

Persentase outliers: 0.13%
```

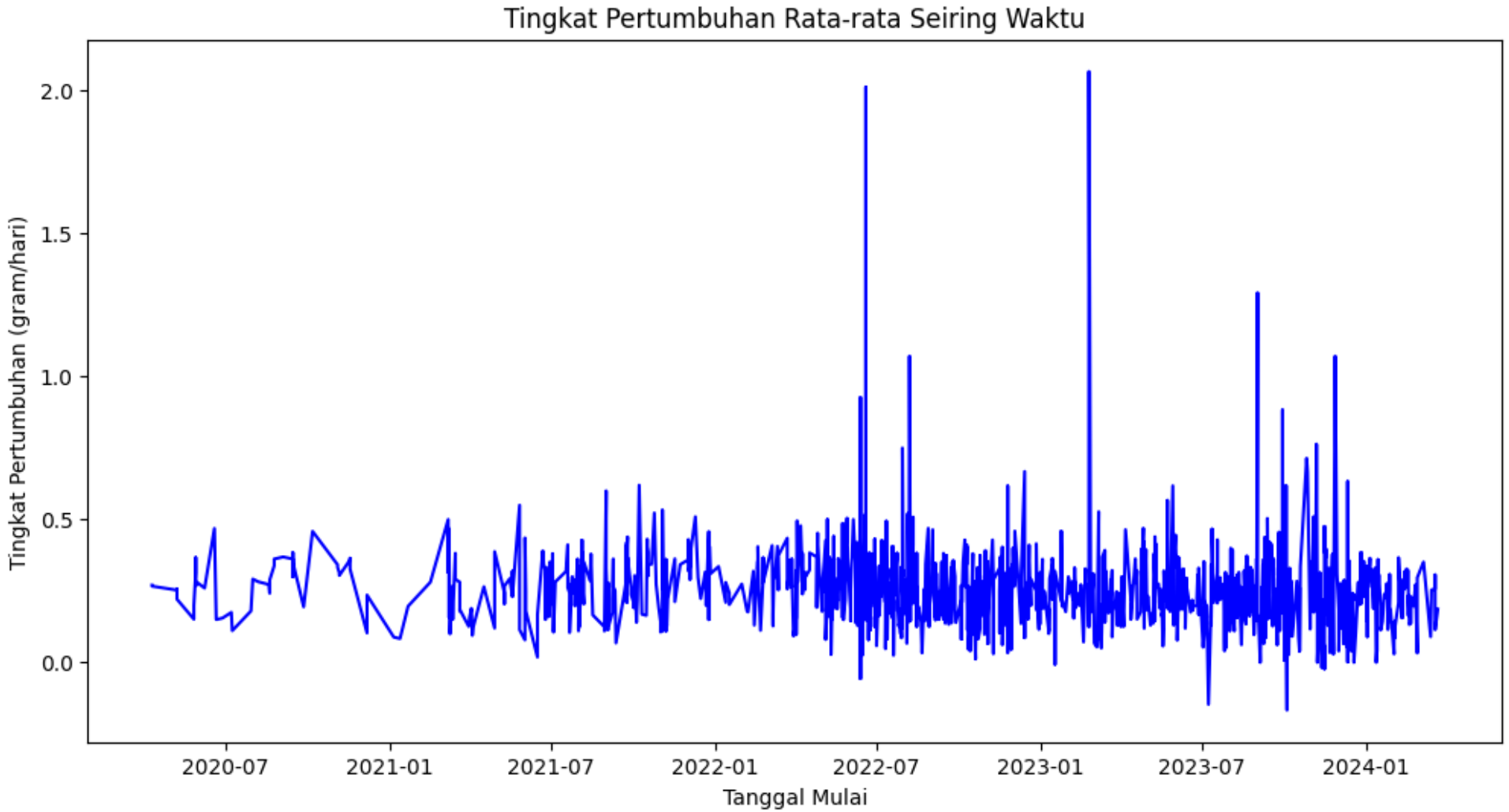
Salahsatu acuan data outlier adalah ketika terdeteksi durasi budidaya terlampau jauh dari waktu budidaya rata-rata. Hal ini penting dilakukan, karena terdapat data dengan durasi budidaya hingga 8349 hari, Dimana ini menandakan adanya anomali pada dataset.

# Average Daily Gain (ADG)

Result

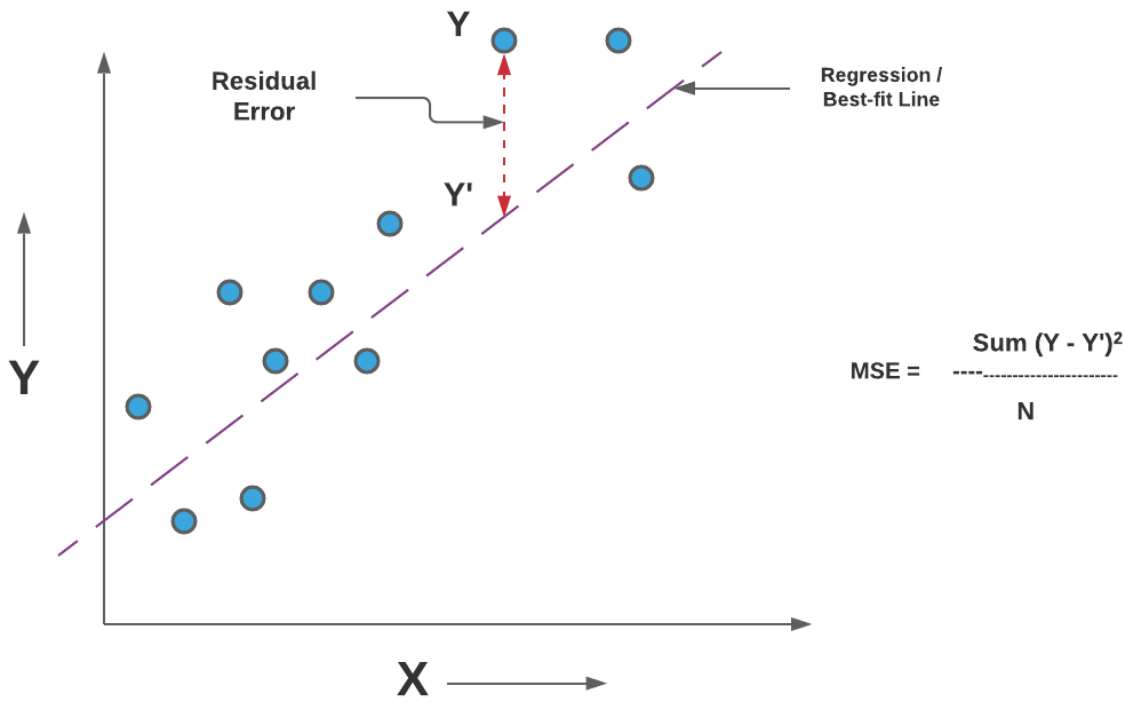



Distribusi ADG
Frekuensi
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
ADG (gram/hari)




Data ini cukup menarik, karena bisa dianalisis lebih dalam bahwa data ini menunjukkan bahwa semakin banyak data ADG yang diperolah, hal ini menunjukkan bahwa budidaya udang semakin berkembang massif. Selain itu, terlihat nilai ADG memiliki tren positif yang cukup baik.

# Predictive Modelling : Evaluation Metrics

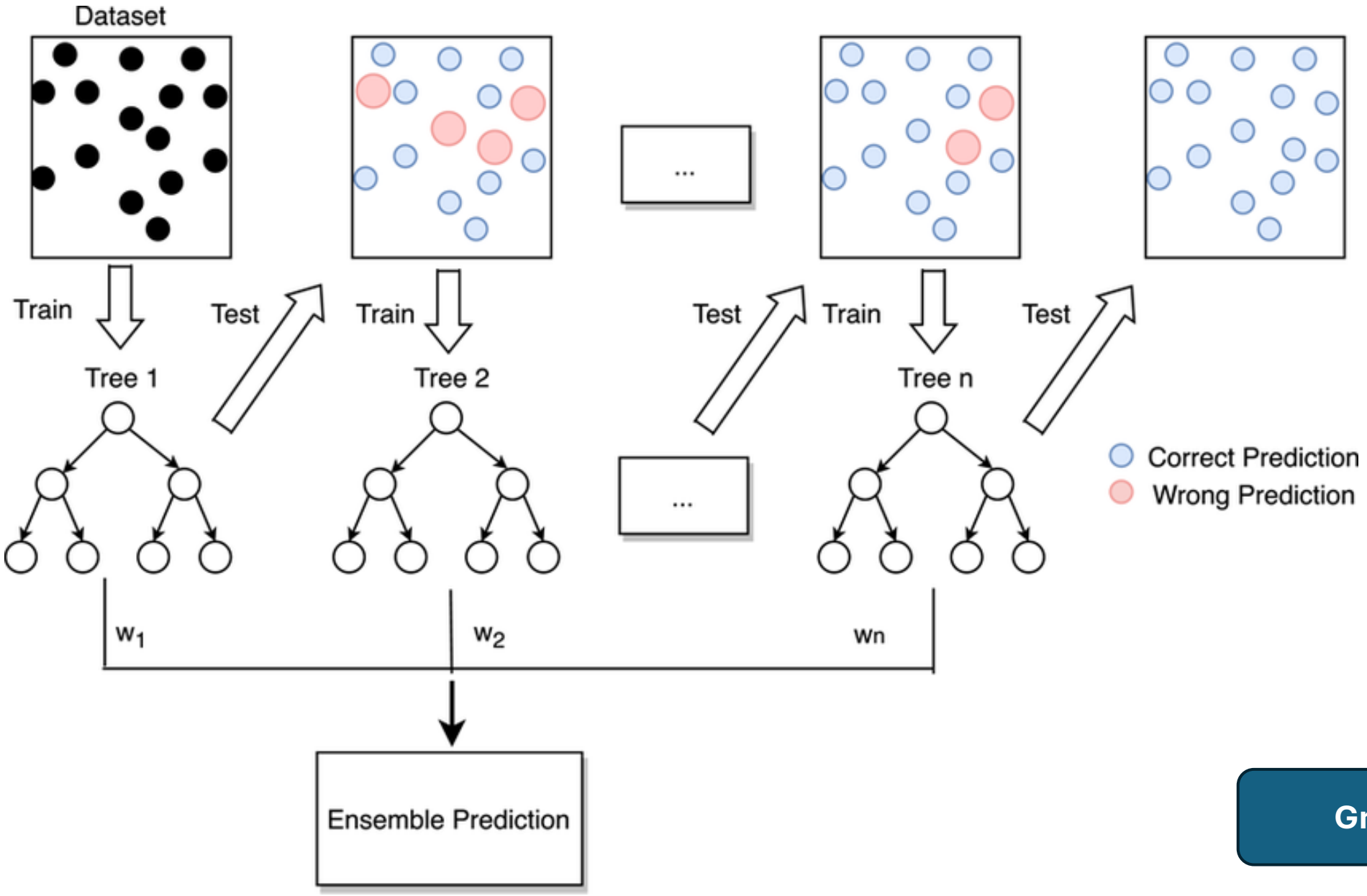



$$MSE = \frac{\text{Sum } (Y - Y')^2}{N}$$

Mean Squared Error (MSE)

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

R Squared Error (R2)

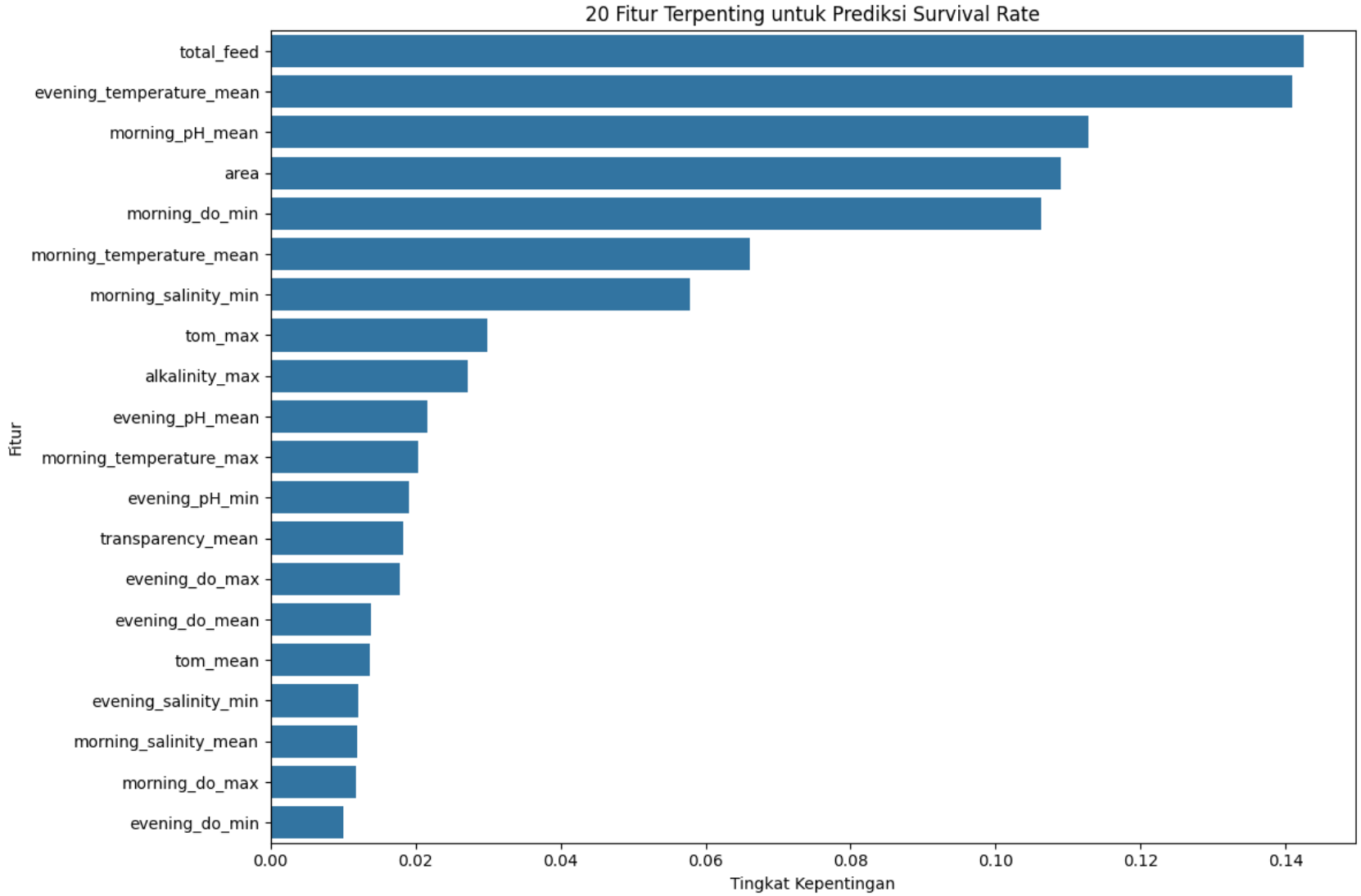


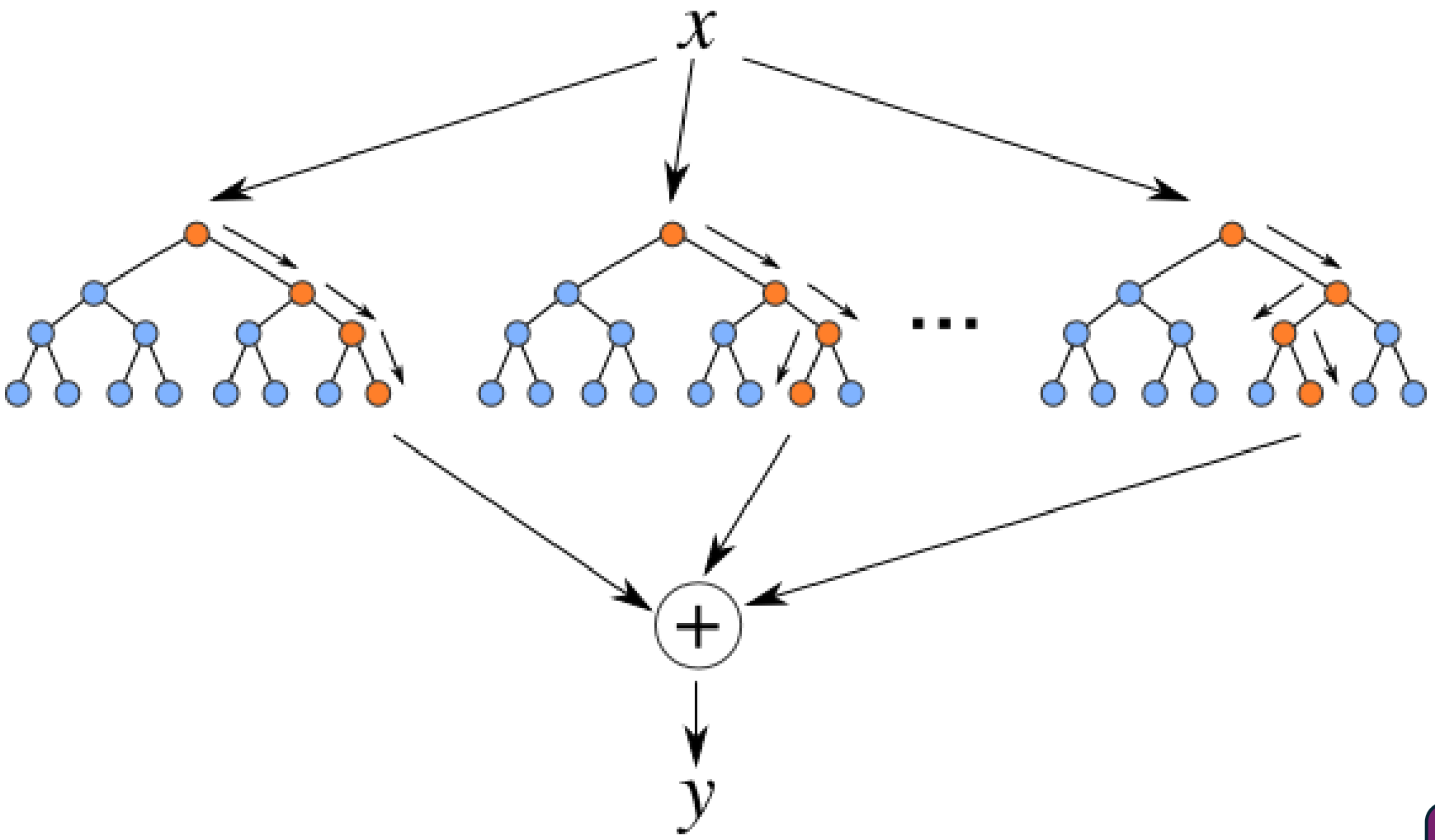

**Gradient Boosting**

# Predictive Modelling : Survival Rate

## Result

```
Random Forest:
Mean Squared Error: 220.60
R-squared Score: 0.74

XGBoost:
Mean Squared Error: 207.94
R-squared Score: 0.76

Linear Regression:
Mean Squared Error: 527.74
R-squared Score: 0.39

Ridge Regression:
Mean Squared Error: 557.99
R-squared Score: 0.35

Lasso Regression:
Mean Squared Error: 540.71
R-squared Score: 0.37

K-Nearest Neighbors:
Mean Squared Error: 304.88
R-squared Score: 0.65

Gradient Boosting:
Mean Squared Error: 182.77
R-squared Score: 0.79
```






Random Forest Regression

# Predictive Modelling : Average Body Weight

```
Mean Squared Error: 2.28
R-squared Score: 0.87

Kepentingan Fitur:
      feature  importance
0  total_feed    0.470635
2     density    0.317544
1   pond_area    0.211821
```

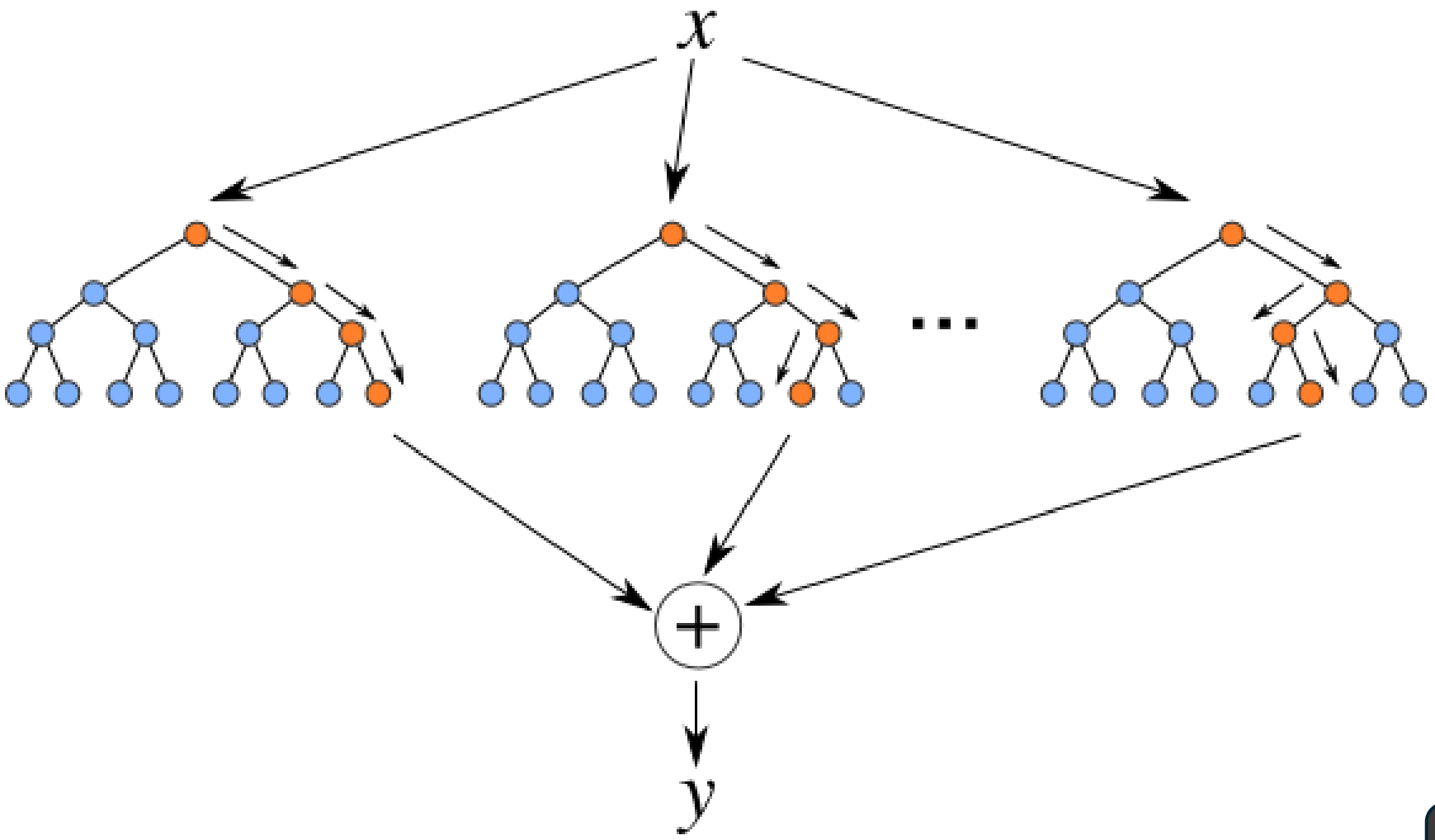
**Result**

```python
# Menggunakan model yang dimuat untuk prediksi
loaded_model = load('abw_prediction_model.joblib')
loaded_scaler = load('abw_prediction_scaler.joblib')

sample_data = np.array([[1000, 500, 20]])  # total_feed, pond_area, density
sample_data_scaled = loaded_scaler.transform(sample_data)
prediction = loaded_model.predict(sample_data_scaled)
print(f"\nPrediksi ABW menggunakan model yang ditraining : {prediction[0]:.2f}")
```

✓ 0.8s

```
Prediksi ABW menggunakan model yang ditraining : 17.62
```

**Testing Model**




Random Forest Regression

# Predictive Modelling : Average Body Weight

```
Mean Squared Error: 3089635331.58
R-squared Score: 0.88

Kepentingan Fitur:
         feature  importance
3    total_feed    0.670226
0  survival_rate    0.112368
2    total_seed    0.104171
1           ABW    0.080262
5       density    0.018617
4     pond_area    0.014357
```

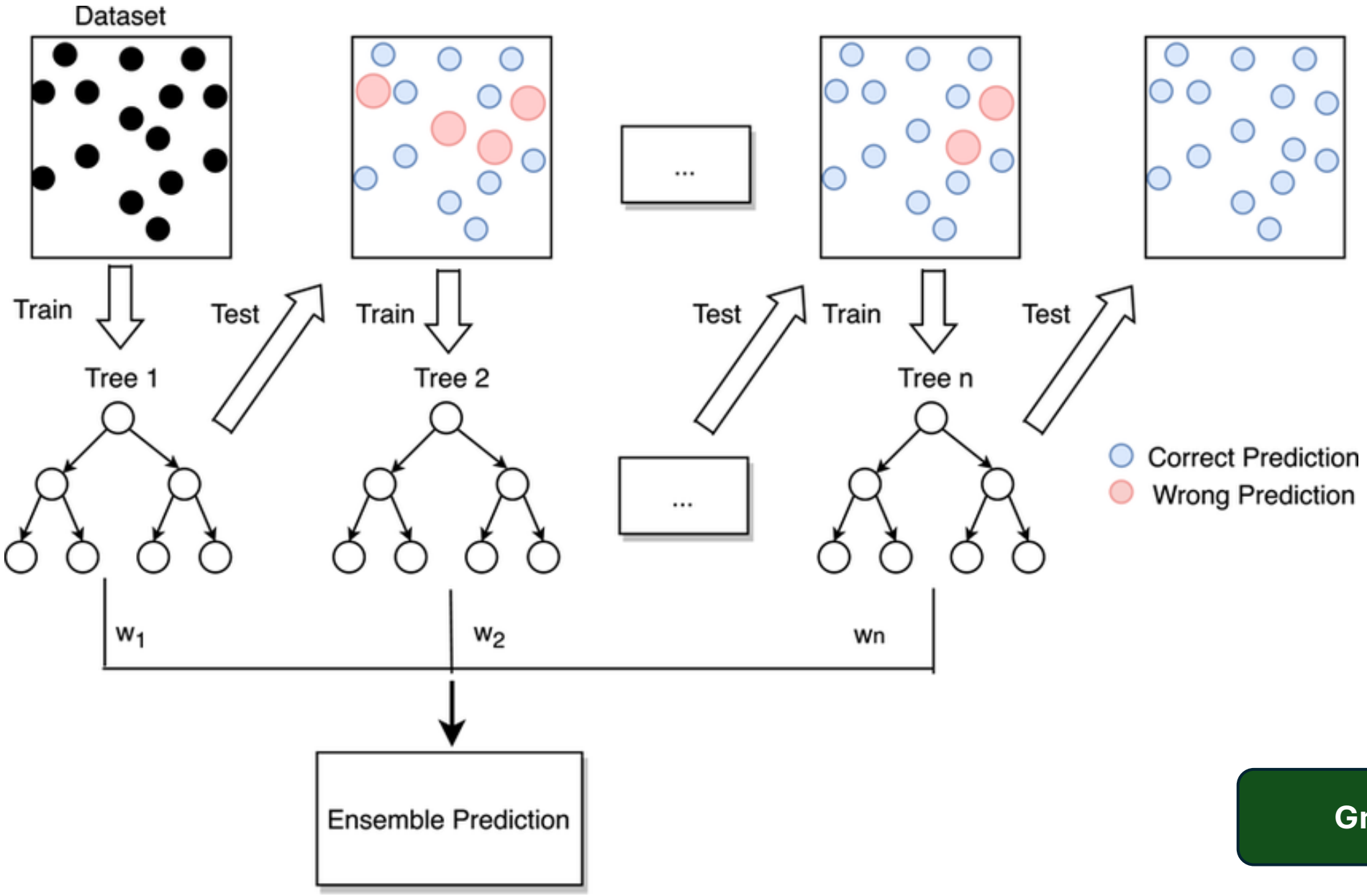
**Result**

```
# Contoh penggunaan model untuk prediksi
loaded_model = load('biomass_prediction_model.joblib')
loaded_scaler = load('biomass_prediction_scaler.joblib')

# survival_rate, ABW, total_seed, total_feed, pond_area, density
sample_data = np.array([[80, 15, 100000, 5000, 1000, 100]])
sample_data_scaled = loaded_scaler.transform(sample_data)
prediction = loaded_model.predict(sample_data_scaled)
print(f"\nPrediksi Biomass: {prediction[0]:.2f} kg")
```

✓ 0.1s

```
Prediksi Biomass: 198433.34 kg
```

**Testing Model**

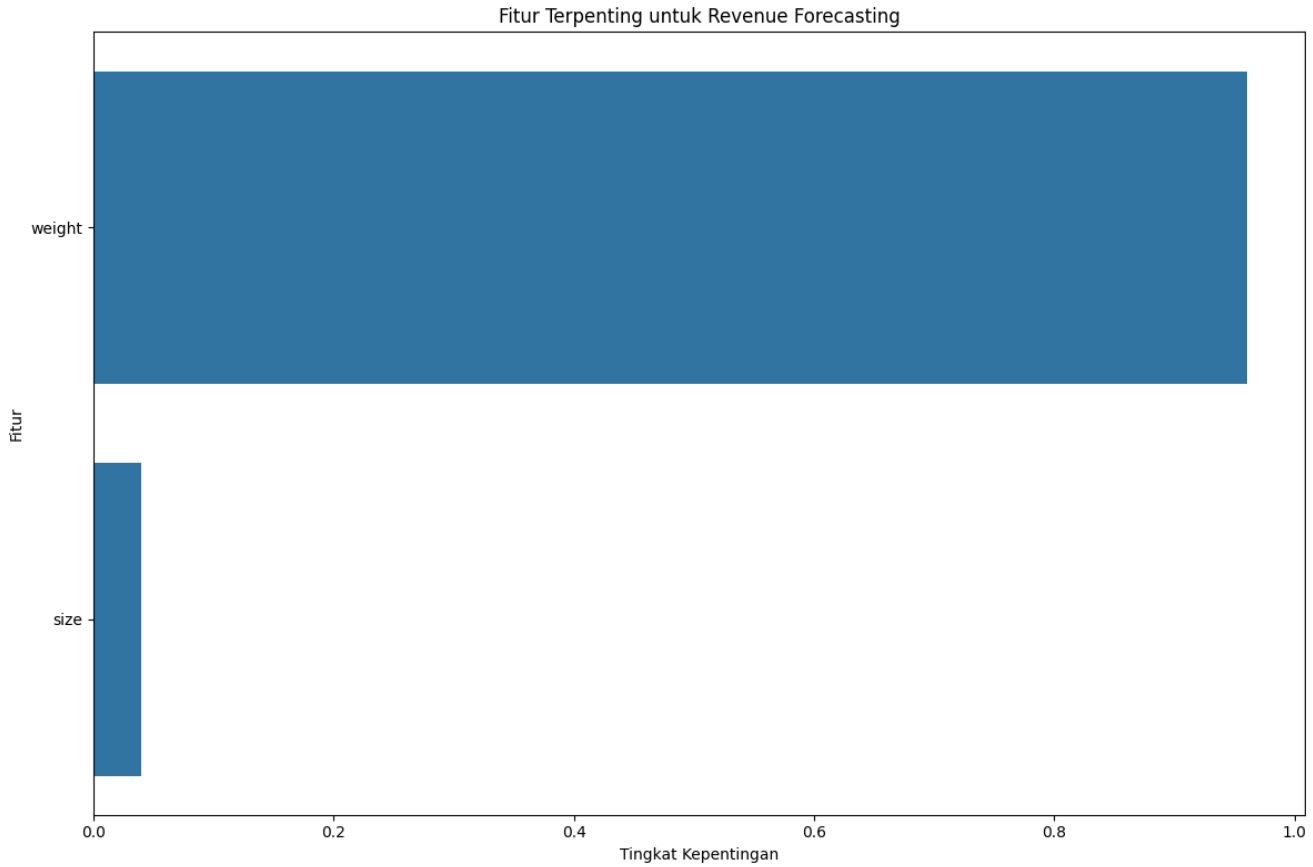



Gradient Boosting

# Predictive Modelling : Revenue

# Result

```
Ridge Regression:
Mean Squared Error: 3149341654670425.00
R-squared Score: 0.43

Lasso Regression:
Mean Squared Error: 3149023474356886.50
R-squared Score: 0.43

Random Forest:
Mean Squared Error: 2041782547250495.75
R-squared Score: 0.63

Gradient Boosting:
Mean Squared Error: 1735922799672199.25
R-squared Score: 0.69

K-Nearest Neighbors:
Mean Squared Error: 2482371330066496.50
R-squared Score: 0.55

XGBoost:
Mean Squared Error: 2979375034994621.50
R-squared Score: 0.46

Model terbaik: Gradient Boosting dengan R-squared Score: 0.69
```



# How to optimize Shrimp farming?

## Water Quality

Based on the dataset processed, the most Important thing for Water Quality ：

1. Temperature
2. pH
3. DO (Kandungan Oksigen)
4. Salinity

**Feeding**

Based on the dataset processed, the most Important thing for Feeding and Pond ：

1. Jumlah Pakan
2. FCR

**Pond**

3. Area
4. Density

# Future Works

**I plan to develop this project into a web-based application using Streamlit. An example of a similar implementation can be seen in my previous SAAS project: ChatMarket (chatmarket.streamlit.app)**

**Planned features include:**

- User-friendly web interface for data input and result visualization
- Real-time dashboard for shrimp cultivation monitoring
- Integration with RAG-LLM to be a Chatbot Shrimp Farming Consultant

**For detail data/processing, you can check in my github :**
https://github.com/mmasdar/shrimp-cultivation-monitoring

## Several Reference


1. Nayan, A. A. *et al*. A machine learning approach for early detection of fish diseases by analyzing water quality. *Trends Sci*. **18**, 1–11 (2021).
2. Islam, M. M., Kashem, M. A. & Uddin, J. Fish survival prediction in an aquatic environment using random forest model. *IAES Int*. *J*. *Artif*. *Intell*. **10**, 614–622 (2021).
3. Ayesha Jasmin, S., Ramesh, P. & Tanveer, M. An intelligent framework for prediction and forecasting of dissolved oxygen level and biofloc amount in a shrimp culture system using machine learning techniques. *Expert Syst*. *Appl*. **199**, 117160 (2022).
4. Orozco-Lugo, A. G. *et al*. Monitoring of water quality in a shrimp farm using a FANET. *Internet of Things (Netherlands)* **18**, 100170 (2022).
5. Venkateswarlu V, S. P. A. P. B. P. A study on water quality parameters in shrimp L. vannamei semi-intensive grow out culture farms in coastal districts of Andhra Pradesh, India. *Int*. *J*. *Fish*. *Aquat*. *Stud*. **7**, 394–399 (2019).
6. Ariadi, H., Fadjar, M., Mahmudi, M. & Supriatna. The relationships between water quality parameters and the growth rate of white shrimp (Litopenaeus vannamei) in intensive ponds. *AACL Bioflux* **12**, 2103–2116 (2019).