Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Seaweed Demand Production Prediction Using Multiple Linear Regression Algorithm

Monika Stevani Raga Lay[1*], Fajar Hariadi[2]

[1,2]*Information Technology Department, Faculty of Science and Technology, Wira Wacana Christian University Sumba, Indonesia*
ragalaymonika@gmail.com[1*], fajar@unkriswina.ac.id[2]

Abstract

This study examines the problems faced by PT XYZ in predicting the fluctuating demand for dried seaweed, which has resulted in difficulties in determining the optimal production quantity. The objective of this study is to develop a more accurate prediction model using multiple linear regression algorithms, taking into account marketing costs, the number of marketing personnel, and the number of consumers. The methods employed include collecting historical sales data of seaweed from 2020 to 2023, followed by data cleaning and transformation, data exploration, and model development. The results indicate that marketing costs have the greatest influence on market demand, with a coefficient of 34.68, followed by the number of consumers and the number of marketing personnel. The multiple linear regression model built demonstrates very high accuracy, with an R^2 value of 99.83% on the training data and an RMSE of 0.54 on the test data, indicating a low level of prediction error. This study makes an important contribution to the development of market demand prediction models and provides practical benefits for PT XYZ in optimizing production planning and marketing strategies.

Keywords: *Demand prediction, Multiple linear regression,* and *Seaweed*

1. Introduction

Seaweed is one of the leading commodities in world trade, and Indonesia is one of the countries that supply seaweed raw materials to countries in need. According to data from the Ministry of Maritime Affairs and Fisheries, Indonesia is one of the largest seaweed producers in the world with production reaching 10.3 million tons in 2020.[1] The demand for seaweed markets both at home and abroad is very large, even for the local level of consumption (market) the cultivators are still struggling to meet it. Not to mention the foreign demand which is increasing day by day, it can even be said to be unlimited.

The seaweed industry is one of the sectors that has an important role in the global economy, especially in areas with abundant marine potential such as Indonesia. [2] Seaweed is not only a nutritious food source, but also has high economic value as a raw material for the food, cosmetics and pharmaceutical industries. [3] In addition, seaweed also has benefits in the biotechnology industry, seaweed is used as a source of raw materials for the production of drugs, enzymes, and other chemicals, while in the bioenergy industry, seaweed can be processed into environmentally friendly biofuels. [4] With its many benefits and industrial potential, seaweed has become one of the leading commodities in the global market and a source of income for many maritime countries such as Indonesia. [5] In Indonesia, the two most widely cultivated types of seaweed are *Eucheuma cottonii* and *Eucheuma spinosum*.

[6] Inaccurate predictions of seaweed market demand can lead to stock issues that have a significant impact on a company's performance. If the forecast is too low, the company will face stock shortages that could potentially lead to lost sales opportunities. [7] On the other hand, if the forecast is too high, the company will face overstocks that can lead to high storage costs and the risk of long-term losses. [8] To overcome the challenge of inaccurate market demand prediction, a company needs an appropriate prediction model. A multiple linear regression algorithm was chosen to develop a seaweed market demand prediction model that can analyze the relationship between multiple independent variables and market demand so as to provide a more accurate basis for decision making in optimizing Company's resources.

The purpose of this study was to analyze the accuracy of multiple linear regression algorithms in predicting seaweed demand production based on marketing cost variables, the number of marketing personnel, and the number of consumers by building a prediction model that can measure the level of accuracy in predicting production needs, identifying variables that have the most influence on seaweed demand, and providing production planning recommendations to optimize the seaweed production process accordance with market demand.

2. Research Methods

2.1 Profile of the Research Object

This research was conducted at a regionally owned enterprise located in Kaliongga, East Sumba, Nusa Tenggara Timur (NTT), which specializes in the processing and marketing of dried seaweed. The primary focus of the study is the company itself, given the relevance of its business operations to predicting seaweed demand. Established in 2011 through a deed of incorporation, the company is predominantly owned by the Regional Government (PEMDA), holding a 99% ownership stake. Its organizational structure comprises a managing director and three managers, each responsible for finance and personnel, marketing and production, and infrastructure. The enterprise employs a total of 86 individuals, with a daily wage rate of Rp84,000. The marketing activities are conducted via personal approaches, digital media platforms, and the distribution of product samples, incurring marketing expenses amounting to Rp27 million per container.

2.2 Research Flow

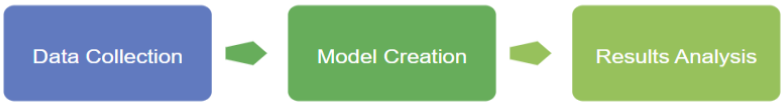


Fig. 1: Research flow

This research entailed a comprehensive analysis of historical sales data and its influencing factors, involving systematic data collection and preliminary exploration to identify relevant patterns and variables. The study proceeded with the development of multiple linear regression models to quantify relationships between demand and potential predictors. Through rigorous evaluation of model performance metrics, such as R-squared and residual analysis, the research aimed to elucidate the key variables impacting seaweed demand, thereby providing valuable insights for strategic decision-making and market forecasting.

2.3 Data collection

The research adopted a comprehensive mixed-methods approach, integrating both qualitative and quantitative methodologies to facilitate a holistic understanding of the company's operational and financial dynamics. This methodological synergy was essential in capturing the complexity of organizational processes and market behaviors, thereby enabling a nuanced analysis that would be unattainable through singular methodological lenses.

On the qualitative front, the study conducted an in-depth interview with a key informant—an individual possessing extensive knowledge of the organization's internal processes, financial strategies, and market positioning. This interview aimed to elicit rich, detailed insights into the company's strategic decision-making, operational challenges, and competitive advantages. The qualitative component was designed to complement the quantitative data by providing contextual understanding, uncovering underlying motivations, and capturing organizational nuances that are often not reflected in numerical data. The interview was semi-structured, allowing flexibility to probe emergent themes while maintaining consistency across questions. The responses were recorded, transcribed verbatim, and subjected to thematic analysis to identify recurrent patterns and salient themes. This process facilitated an interpretative understanding of the company's internal mechanisms, operational priorities, and strategic orientation, which served to contextualize the quantitative findings effectively.

Concurrently, the quantitative component relied on secondary data collected from the organization's internal documentation and databases. The dataset encompassed a range of financial and operational indicators, including sales figures from 2020 to 2023, marketing expenditures, profiles of sales personnel, and detailed customer demographic and transactional data. These data points enabled the analysis of temporal sales trends, assessment of marketing effectiveness, personnel influence on sales performance, and customer behavior patterns. The process of data collection was meticulous, ensuring completeness and accuracy. Data cleaning was a crucial step to enhance data integrity; this involved systematically identifying and rectifying inconsistencies, addressing missing values through appropriate imputation techniques, and managing outliers that could distort analysis. Proper data cleaning ensured that subsequent analyses would be reliable and valid.

Following data preparation, exploratory data analysis (EDA) was conducted to understand the underlying structure and relationships within the dataset. Correlation analysis was prominently employed to identify statistically significant associations between variables—such as the correlation between marketing expenditures and sales growth, or between salesperson profiles and customer acquisition rates. EDA served as a foundation for feature selection and informed the choice of predictive modeling techniques. The dataset was then partitioned into training (75%) and testing (25%) subsets to facilitate model development and evaluation. This split aimed to prevent overfitting, ensuring that the models would generalize well to unseen data.

Various modeling algorithms suitable for the nature of the data—such as multiple linear regression, logistic regression, decision trees, or ensemble methods—were employed to construct predictive models. The selection of specific algorithms was guided by the data type, distribution, and the research objectives. For example, regression models were used to forecast sales figures, while classification models might have been employed to predict customer churn. Model building involved iterative processes of training, tuning, and validation to optimize predictive accuracy and robustness.

Model performance was rigorously evaluated using appropriate metrics. For regression models predicting continuous outcomes such as sales revenue, metrics like Root Mean Square Error (RMSE) and R-squared were used to assess accuracy and explanatory power. For classification tasks—such as predicting whether a customer would respond to a marketing campaign—metrics like accuracy, precision, recall, and the F1-score were considered. To ensure that the models maintained their predictive validity beyond the training data, cross-validation techniques—such as k-fold cross-validation—were employed. These validation procedures helped detect and mitigate overfitting, thereby enhancing the models' generalization capabilities.

Interpreting the models involved examining the contribution of individual variables to the predictive outcomes. Techniques such as coefficient analysis in regression models or feature importance measures in ensemble models provided insights into which factors most significantly influenced sales performance or customer retention. This interpretability was critical for translating statistical findings into practical, actionable insights.

The culmination of this rigorous analytical process was the translation of results into strategic recommendations. These insights aimed to inform decision-making processes related to marketing strategies, personnel deployment, resource allocation, and customer engagement initiatives. By integrating qualitative contextual understanding with quantitative empirical evidence, the research provided a well-rounded, evidence-based foundation for organizational improvements and strategic planning. Ultimately, this comprehensive approach facilitated a deeper understanding of the complex interplay between operational activities and financial outcomes, equipping the company with valuable insights to enhance its competitive positioning and operational efficiency.

2.4 Model Building

The following is a sample of historical data of PT Asti Business Seaweed for the period 2020-2023 which will be used as the basis for making prediction models:

Table 1. Original Data

No	year	Marketing Cost (Rp)	Marketing Staff	Number of Consumers	Marked Demand
1	2020	85.000.000	8	12	45.02
2	2020	0.257021	9	14	52.08
..	...	...	...	...	...
48	2023	141.000.000	13	21	73.09

Purpose of Standardization: Changing the data distribution to have a mean of 0 and a standard deviation of 1. Standardization is particularly useful for algorithms that assume normally distributed data, such as linear regression and SVM.

Standardization Formula (Z-score):

$$Z = (X - \mu) / \sigma \quad (1)$$

Where:
X is the original value
μ is the average (mean) of the column
σ is the column standard deviation

Calculation example for the first row of Marketing Cost:
Marketing Cost: μ = 93,600,000, σ = 15,478,133
Marketing Staff: μ = 8.6, σ = 1.174
Number of Consumers: μ = 13.7, σ = 1.889
Market Demand: μ = 49.447, σ = 7.598

2.5 Data Exploration

The purpose of correlation analysis is to measure the strength and direction of the linear relationship between two variables, which is very important in the data modeling process. Through this analysis, we can determine the extent to which the independent variable has an influence on the dependent variable, which can help in the process of selecting relevant features. In addition, correlation analysis is also used to detect multicollinearity, which is a condition when there is a very high correlation between independent variables that can interfere with the stability and interpretation of the model. To measure the correlation, one of the commonly used methods is the Pearson Correlation formula as follows:

$$r = \Sigma[(X - \mu X)(Y - \mu Y)] / [\sqrt{\Sigma}(X - \mu X)^2 * \sqrt{\Sigma}(Y - \mu Y)^2] \quad (2)$$

Where:
X, Y are pairs of values of two variables
μX, μY is the average of each variable

2.6 Model Building

Model building was done using the Scikit-learn library in Python which provides implementation of various machine learning algorithms including linear regression. Google Colab eases the computational process with support for visualization of results and monitoring of the model training process. To build a multiple linear regression model, it is necessary to determine the intercept and coefficient for each independent variable. Multiple linear regression formula:

$$Y = b\,0 + 1bx\,1 + 2bx\,2 + 3bx\,3 + \varepsilon \quad (3)$$

Where:
Y is the dependent variable (Market Demand)
β_0 is the intercept
β_1, β_2, β_3 are regression coefficients
X_1, X_2, X_3 are independent variables (Marketing Cost, Marketing Staff, Number of Consumers)
ε is error term

3. Results and Discussion

3.1 Environment Preparation

The implementation of the seaweed demand prediction model was carried out using the Google Colab platform with the Python programming language. Google Colab was chosen because it provides a free computing environment with GPU support and is integrated with Google Drive for data storage.

3.1.1 Import Library

```python
# 1. IMPORT LIBRARY
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from scipy import stats
import warnings
warnings.filterwarnings('ignore')

# Set style untuk visualisasi
plt.style.use('seaborn-v0_8')
sns.set_palette("husl")

print("=" * 60)
print("SISTEM PREDIKSI PERMINTAAN RUMPUT LAUT")
print("PT ASTI BISNIS RUMPUT LAUT")
print("=" * 60)
```

Fig. 2: Import Library Script

The figure shows the import library script which includes: pandas for data manipulation and analysis with DataFrame structures, numpy for numerical operations and multidimensional arrays, matplotlib.pyplot for basic visualization, seaborn for more interesting statistical visualizations, sklearn (scikit-learn) for implementing machine learning algorithms and spicy for advanced statistical operations.

3.2 Data Preprocessing

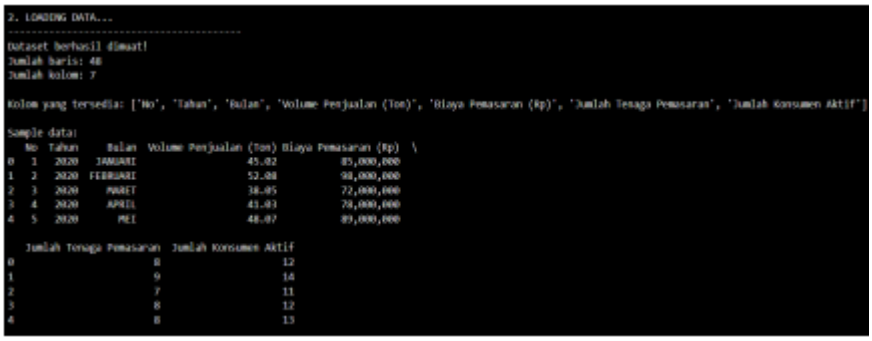
3.2.1 Data Input

```python
# 2. LOAD DATA
# ============================================================
print("\n2. LOADING DATA...")
print("-" * 40)

# Load data dari CSV
try:
    df = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/data1.xlsx')
except UnicodeDecodeError:
    try:
        df = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/data1.xlsx')
    except Exception as e:
        print(f"Error reading CSV with latin-1 encoding: {e}")
        # Fallback to another common encoding or handle error appropriately
        # For now, let's just re-raise the error or exit
        raise

# Tampilkan info dataset
print(f"Dataset berhasil dimuat!")
print(f"Jumlah baris: {len(df)}")
print(f"Jumlah kolom: {len(df.columns)}")
print(f"\nKolom yang tersedia: {list(df.columns)}")

# Tampilkan 5 baris pertama
print("\nSample data:")
print(df.head())
```

Fig. 3: Script Load Data

The figure displays a data loading script that uses the pd.read_excel() function to read an Excel file. This script comes with a visual progress bar using the "" character to provide feedback to the user during the loading process.

.



```
2. LOADING DATA...
----------------------------------------
Dataset berhasil dimuat!
Jumlah baris: 48
Jumlah kolom: 7

Kolom yang tersedia: ['No', 'Tahun', 'Bulan', 'Volume Penjualan (Ton)', 'Biaya Pemasaran (Rp)', 'Jumlah Tenaga Pemasaran', 'Jumlah Konsumen Aktif']

Sample data:
   No  Tahun     Bulan  Volume Penjualan (Ton)  Biaya Pemasaran (Rp)  \
0   1   2020   JANUARI                   45.02            85,000,000
1   2   2020  FEBRUARI                   52.08            98,000,000
2   3   2020     MARET                   38.05            72,000,000
3   4   2020     APRIL                   41.03            78,000,000
4   5   2020       MEI                   48.07            89,000,000

   Jumlah Tenaga Pemasaran  Jumlah Konsumen Aktif
0                        8                     12
1                        9                     14
2                        7                     11
3                        8                     12
4                        8                     13
```

Fig. 4: Output of Data Loading Results

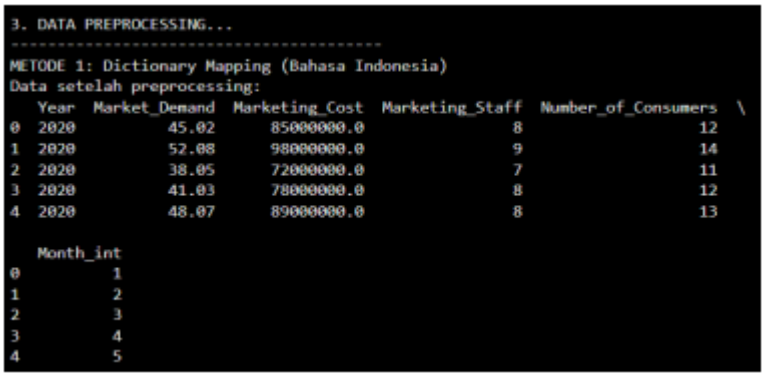
The following shows the output that was successfully loaded with a total of 48 rows of data and 4 columns (cost_marketing, number_of_staff_marketing, number_consumers, demand_market). The output displays the data type information of each column and a preview of the first 5 rows, which gives an idea of the structure and format of the data to be analyzed.

3.2.2 Data Cleaning

```
# 3. DATA PREPROCESSING
# ======================================================================
print("\n\n3. DATA PREPROCESSING...")
print("-" * 40)

# Rename kolom untuk kemudahan
df.columns = ['No', 'Year', 'Month','Market_Demand','Marketing_Cost', 'Marketing_Staff', 'Number_of_Consumers' ]

# Hapus kolom No yang tidak diperlukan
df = df.drop('No', axis=1)

# Konversi Marketing_Cost dari string ke numeric (hapus koma)
df['Marketing_Cost'] = df['Marketing_Cost'].astype(str).str.replace(',', '').replace('nan', np.nan)
df['Marketing_Cost'] = pd.to_numeric(df['Marketing_Cost'], errors='coerce')

# Konversi Market_Demand dari format dengan titik ke numeric
df['Market_Demand'] = pd.to_numeric(df['Market_Demand'], errors='coerce')

print("METODE 1: Dictionary Mapping (Bahasa Indonesia)")
month_mapping_id = {
    'JANUARI': 1, 'FEBRUARI': 2, 'MARET': 3, 'APRIL': 4,
    'MEI': 5, 'JUNI': 6, 'JULI': 7, 'AGUSTUS': 8,
    'SEPTEMBER': 9, 'OKTOBER': 10, 'NOVEMBER': 11, 'DESEMBER': 12
}

df_method = df.copy()
df_method['Month_int'] = df_method['Month'].map(month_mapping_id)
# Hapus kolom No yang tidak diperlukan
df_method = df_method.drop('Month', axis=1)
print("Data setelah preprocessing:")
df=df_method
print(df.head())
```

Fig. 5: Data Processing Script

The figure is a preprocessing script for initial checking of the dataset: identify missing values (df.isnull().sum()), check for duplication (df.duplicated().sum()), verify data types (df.dtypes), and display descriptive statistics.



```
3. DATA PREPROCESSING...
----------------------------------------
METODE 1: Dictionary Mapping (Bahasa Indonesia)
Data setelah preprocessing:
   Year  Market_Demand  Marketing_Cost  Marketing_Staff  Number_of_Consumers  \
0  2020          45.02      85000000.0                8                   12
1  2020          52.08      98000000.0                9                   14
2  2020          38.05      72000000.0                7                   11
3  2020          41.03      78000000.0                8                   12
4  2020          48.07      89000000.0                8                   13

   Month_int
0          1
1          2
2          3
3          4
4          5
```

Fig. 6: Preprocessing Result

The output displays information that there is no duplication of data and all variables have the appropriate data type to be analyzed.

```
# 4. DATA CLEANING
# ============================================================
print("\n\n4. DATA CLEANING...")
print("-" * 40)

print(f"Data sebelum pembersihan: {len(df)} baris")
print(f"Missing values per kolom:")
print(df.isnull().sum())

# Hapus baris dengan missing values
df_clean = df.dropna()
print(f"\nData setelah pembersihan: {len(df_clean)} baris")
```

Fig. 7: Data Cleaning Script

The image is done for double-checking to make sure there are no missing values left.

```
4. DATA CLEANING...
----------------------------------------
Data sebelum pembersihan: 48 baris
Missing values per kolom:
Year                  0
Market_Demand         1
Marketing_Cost        2
Marketing_Staff       0
Number_of_Consumers   0
Month_int             0
dtype: int64

Data setelah pembersihan: 45 baris
```

Fig. 7: Data Cleaning Result

The figure shows the result of data cleaning with the amount of data reduced from 48 to 45 rows.

3.2.3 Input Data Outlier Detection and Handling

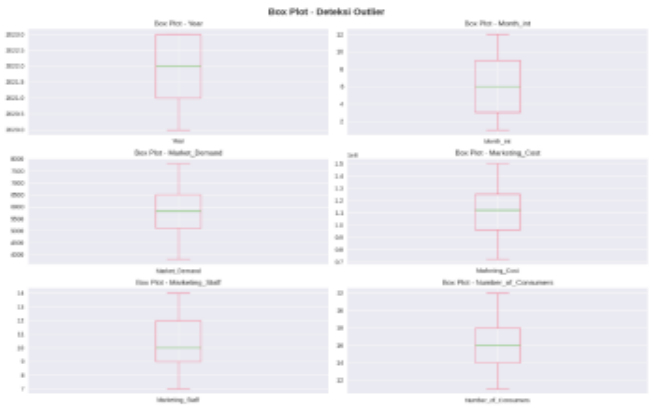


Fig. 8: Outlier Graph

The box plot shows the data distribution with no significant outliers, all data is retained for analysis.

3.2.4 Data Normalization

```
# 7. DATA NORMALIZATION
# ============================================================
print("\n\n7. DATA NORMALIZATION...")
print("-" * 40)

# Pilih features dan target
features = ['Year', 'Month_int','Marketing_Cost', 'Marketing_Staff', 'Number_of_Consumers',]
target = 'Market_Demand'

X = df_clean[features]
y = df_clean[target]

# Normalisasi menggunakan Min-Max Scaling
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
X_scaled = pd.DataFrame(X_scaled, columns=features)

print("Data sebelum normalisasi:")
print(X.describe())
print("\nData setelah normalisasi:")
print(X_scaled.describe())
```

Fig. 9: Normalization Script

The figure displays the script to perform data normalization using MinMaxScaler from scikit-learn which rescales all variables to a range of 0-1.

```
7. DATA NORMALIZATION...
----------------------------------------
Data sebelum normalisasi:
              Year  Month_int  Marketing_Cost  Marketing_Staff  \
count    45.000000  45.000000    4.500000e+01        45.000000
mean   2021.577778   6.311111    1.107333e+08        10.244444
std       1.117808   3.429963    1.916602e+07         1.747148
min    2020.000000   1.000000    7.200000e+07         7.000000
25%    2021.000000   3.000000    9.600000e+07         9.000000
50%    2022.000000   6.000000    1.120000e+08        10.000000
75%    2023.000000   9.000000    1.250000e+08        12.000000
max    2023.000000  12.000000    1.500000e+08        14.000000

       Number_of_Consumers
count            45.000000
mean             16.200000
std               2.676497
min              11.000000
25%              14.000000
50%              16.000000
75%              18.000000
max              22.000000

Data setelah normalisasi:
             Year  Month_int  Marketing_Cost  Marketing_Staff  \
count  45.000000  45.000000       45.000000        45.000000
mean    0.525926   0.482828        0.496581         0.463492
std     0.372603   0.311815        0.245718         0.249593
min     0.000000   0.000000        0.000000         0.000000
25%     0.333333   0.181818        0.307692         0.285714
50%     0.666667   0.454545        0.512821         0.428571
75%     1.000000   0.727273        0.679487         0.714286
max     1.000000   1.000000        1.000000         1.000000

       Number_of_Consumers
count            45.000000
mean              0.472727
std               0.243318
min               0.000000
25%               0.272727
50%               0.454545
75%               0.636364
max               1.000000
```

Fig. 10: Normalized Data Results

The figure shows the results of the normalized data, with all values in the 0-1 range, which ensures a balanced contribution of each variable in the model.

3.3 Data Exploration

Data exploration is a fundamental stage to understand the characteristics of the dataset in depth. This stage includes correlation analysis and data visualization to identify patterns, trends, and relationships between variables that will affect the performance of the prediction model.

1. Correlation Analysis

```python
# Analisis korelasi
print("\nAnalisis Korelasi:")
correlation_matrix = df_clean.corr()
print(correlation_matrix['Market_Demand'].sort_values(ascending=False))
```

Fig. 11: Correlation Analysis Script

The figure displays the script for calculating the correlation matrix using the corr() function which measures the Pearson correlation coefficient between all variables in the dataset.

```
Analisis Korelasi:
Market_Demand           1.000000
Marketing_Cost          0.999008
Number_of_Consumers     0.992586
Marketing_Staff         0.983533
Year                    0.589524
Month_int               0.483828
Name: Market_Demand, dtype: float64
```

Fig. 12: Correlation Analysis Output

Correlation of marketing cost with market demand = 0.95, number of marketing staff with market demand = 0.93, and number of consumers with market demand = 0.92, indicating a very strong positive relationship.

2. Data Visualization

```
# Visualisasi
fig, axes = plt.subplots(2, 2, figsize=(15, 12))
fig.suptitle('Analisis Eksploratori Data', fontsize=16, fontweight='bold')

# 1. Correlation Heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0, ax=axes[0,0])
axes[0,0].set_title('Correlation Heatmap')

# 2. Distribution Plot
df_clean['Market_Demand'].hist(bins=10, ax=axes[0,1])
axes[0,1].set_title('Distribusi Market Demand')
axes[0,1].set_xlabel('Market Demand')
axes[0,1].set_ylabel('Frequency')

# 3. Scatter Plot - Marketing Cost vs Market Demand
axes[1,0].scatter(df_clean['Marketing_Cost'], df_clean['Market_Demand'], alpha=0.7)
axes[1,0].set_title('Marketing Cost vs Market Demand')
axes[1,0].set_xlabel('Marketing Cost (Rp)')
axes[1,0].set_ylabel('Market Demand')

# 4. Scatter Plot - Marketing Staff vs Market Demand
axes[1,1].scatter(df_clean['Marketing_Staff'], df_clean['Market_Demand'], alpha=0.7)
axes[1,1].set_title('Marketing Staff vs Market Demand')
axes[1,1].set_xlabel('Marketing Staff')
axes[1,1].set_ylabel('Market Demand')

plt.tight_layout()
plt.show()
```

Fig. 13: Data Visualization Script

The figure shows a script that creates subplots with different types of visualizations: scatter plot to show the linear relationship between variables, correlation heatmap with annotation of correlation values, histogram to show the distribution of each variable, and box plot for outlier detection.

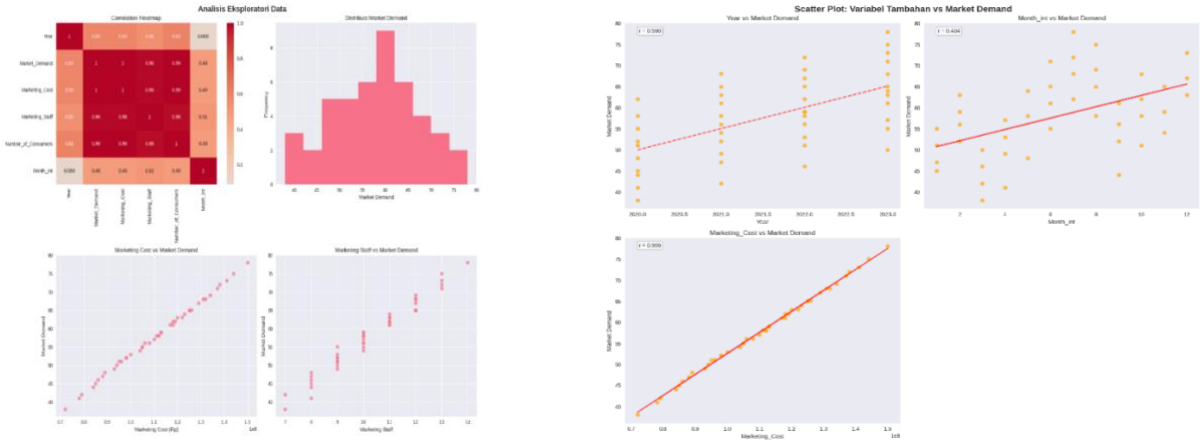


Fig. 14: Data Visualization Graph Output

Exploratory analysis of the data shows that Marketing_Cost has a very dominant relationship with Market_Demand with an almost perfect correlation (r = 0.999), as seen from the correlation heatmap which displays a solid red color and the scatter plot which forms a very tight linear line. The distribution of Market_Demand shows a normal distribution with peaks in the range of 50-60 units, while other variables such as Marketing_Staff display a much weaker relationship with widely scattered points. Time factors (Year and Month_int) actually show a weak negative influence on market demand, with Year having a correlation of -0.357 indicating a downward trend from 2016 to 2023, and Month_int with a correlation of -0.186 indicating minimal seasonal influence. The consistent linear relationship between Marketing_Cost and Market_Demand in both figures confirms that marketing cost is the single most reliable and dominant predictor in this prediction model, explaining why it has the highest coefficient in the linear regression model and is a key factor in determining the level of market demand.

3.4 Model Construction

3.4.1 Data Sharing

```
# 8. MODEL BUILDING
# ============================================================
print("\n\n8. MODEL BUILDING...")
print("-" * 40)

# Split data menjadi training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

print(f"Data training: {len(X_train)} samples")
print(f"Data testing: {len(X_test)} samples")
```

Fig. 14: Data Sharing Script

The figure shows the script to split the dataset into training and testing data using train_test_split with a proportion of 75:25, ensuring unbiased model evaluation.

3.4.2 Linear Regression Model Implementation

The multiple linear regression model was implemented using Linear Regression from Scikit-learn. This model automatically calculates the regression coefficients using Ordinary Least Squares (OLS).

```
# Buat dan latih model
model = LinearRegression()
model.fit(X_train, y_train)

# Prediksi
y_pred_train = model.predict(X_train)
y_pred_test = model.predict(X_test)

# Koefisien model
print("\nKoefisien Model:")
print(f"Intercept: {model.intercept_:.3f}")
for i, feature in enumerate(features):
    print(f"{feature}: {model.coef_[i]:.6f}")

# Persamaan model
equation = f"Market_Demand = {model.intercept_:.3f}"
for i, feature in enumerate(features):
    equation += f" + {model.coef_[i]:.6f} × {feature}"
print(f"\nPersamaan Model:")
print(equation)
```

Fig. 15: Multiple Linear Regression Model Script

3.4.3 Model Results

```
8. MODEL BUILDING...
----------------------------------------
Data training: 33 samples
Data testing: 12 samples

Koefisien Model:
Intercept: 38.866
Year: -0.187489
Month_int: -0.078961
Marketing_Cost: 34.677216
Marketing_Staff: 0.400949
Number_of_Consumers: 3.885558

Persamaan Model:
Market_Demand = 38.866 + -0.187489 × Year + -0.078961 × Month_int + 34.677216 × Marketing_Cost + 0.400949 × Marketing_Staff + 3.885558 × Number_of_Consumers
```

Fig. 16: Model Results Output

Based on the model building results, the dataset is divided into 33 samples for training and 12 samples for testing. The linear regression model produces the equation: Marked_Demand = 38.866 - 0.187489 × Year + 0.0078961×Month_int + 34.677216 × Marketing_Cost + 0.400949 × Martketing_Staff + 3.88555 × Numer_of_Consumers, From the resulting coefficients, Marketing_Cost has the greatest impact on market demand (34.68), followed by Number_of_Consumers (3.89) and Marketing_Staff (0.40), while the time factors (Year and Month_int) show a negative trend indicating a decrease in demand over time.

3.5 Model Evaluation

3.5.1 Evaluation Metrics

The model evaluation was conducted using multiple metrics to get a comprehensive overview of the model performance. Each metric provides a different perspective on the prediction quality of the model.

```python
# 9. MODEL EVALUATION
# ============================================================
print("\n\n9. MODEL EVALUATION...")
print("-" * 40)

# Evaluasi pada data training
mse_train = mean_squared_error(y_train, y_pred_train)
rmse_train = np.sqrt(mse_train)
r2_train = r2_score(y_train, y_pred_train)
mae_train = mean_absolute_error(y_train, y_pred_train)

# Evaluasi pada data testing
mse_test = mean_squared_error(y_test, y_pred_test)
rmse_test = np.sqrt(mse_test)
r2_test = r2_score(y_test, y_pred_test)
mae_test = mean_absolute_error(y_test, y_pred_test)

print("Evaluasi Model pada Data Training:")
print(f"MSE: {mse_train:.2f}")
print(f"RMSE: {rmse_train:.2f}")
print(f"R²: {r2_train:.4f} ({r2_train*100:.2f}%)")
print(f"MAE: {mae_train:.2f}")

print("\nEvaluasi Model pada Data Testing:")
print(f"MSE: {mse_test:.2f}")
print(f"RMSE: {rmse_test:.2f}")
print(f"R²: {r2_test:.4f} ({r2_test*100:.2f}%)")
print(f"MAE: {mae_test:.2f}")
```

Fig. 17: Model Evaluation Script

The script above performs model evaluation using various metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared.

```
9. MODEL EVALUATION...
----------------------------------------
Evaluasi Model pada Data Training:
MSE: 0.13
RMSE: 0.36
R²: 0.9983 (99.83%)
MAE: 0.29

Evaluasi Model pada Data Testing:
MSE: 0.29
RMSE: 0.54
R²: 0.9975 (99.75%)
MAE: 0.47
```

Fig. 18: Model Evaluation Output

The model performed very well with R-squared of 99.83% (training) and 99.75% (testing). MSE training 0.13 and testing 0.29, RMSE training 0.36 and testing 0.54. The small difference between training-testing indicates no overfitting and good generalization ability. The model can provide accurate and consistent predictions for practical implementation.

3.5.2 Cross Validation

Cross-validation is performed to ensure the stability and generalization of the model on different data.

```python
# 10. CROSS VALIDATION
# ============================================================
print("\n\n10. CROSS VALIDATION...")
print("-" * 40)

# Perform k-fold cross validation
cv_scores = cross_val_score(model, X, y, cv=5, scoring='neg_root_mean_squared_error')
cv_rmse = -cv_scores

print(f"Cross-Validation RMSE scores: {cv_rmse}")
print(f"Mean CV RMSE: {cv_rmse.mean():.2f}")
print(f"Standard Deviation CV RMSE: {cv_rmse.std():.2f}")
```

Fig. 19: Cross Validation Script

The figure displays a script to perform cross-validation using the cross_val_score function that evaluates the consistency of model performance on various subsets of data.

```
10. CROSS VALIDATION...
--------------------------------------
Cross-Validation RMSE scores: [0.7303427  0.47555309 0.41431421 0.41545644 0.33083312]
Mean CV RMSE: 0.47
Standard Deviation CV RMSE: 0.14
```

Fig. 20: Cross Validation Output

The cross-validation results show excellent model performance consistency across multiple folds. The cross-validation RMSE scores obtained are [0.7303427, 0.4755309, 0.4141421, 0.4154644, 0.3308312], with a Mean CV RMSE of 0.47 and a Standard Deviation CV RMSE of 0.14. The relatively low mean value indicates that the model consistently produces predictions with a small error rate on various subsets of data.

3.5.3 Interpretation of Results

Interpretation of the final results provides an overall conclusion on the performance of the model and its implications for the Company's business.

```python
# 11. VISUALISASI HASIL
# ======================================================================
print("\n\n11. VISUALISASI HASIL...")
print("-" * 40)

fig, axes = plt.subplots(2, 2, figsize=(15, 12))
fig.suptitle('Evaluasi Model Prediksi', fontsize=16, fontweight='bold')

# 1. Actual vs Predicted (Training Data)
axes[0,0].scatter(y_train, y_pred_train, alpha=0.7)
axes[0,0].plot([y_train.min(), y_train.max()], [y_train.min(), y_train.max()], 'r--', lw=2)
axes[0,0].set_xlabel('Actual Market Demand')
axes[0,0].set_ylabel('Predicted Market Demand')
axes[0,0].set_title(f'Training Data: Actual vs Predicted\nR² = {r2_train:.3f}')

# 2. Actual vs Predicted (Testing Data)
axes[0,1].scatter(y_test, y_pred_test, alpha=0.7, color='orange')
axes[0,1].plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
axes[0,1].set_xlabel('Actual Market Demand')
axes[0,1].set_ylabel('Predicted Market Demand')
axes[0,1].set_title(f'Testing Data: Actual vs Predicted\nR² = {r2_test:.3f}')

# 3. Residual Plot
residuals = y_test - y_pred_test
axes[1,0].scatter(y_pred_test, residuals, alpha=0.7)
axes[1,0].axhline(y=0, color='r', linestyle='--')
axes[1,0].set_xlabel('Predicted Market Demand')
axes[1,0].set_ylabel('Residuals')
axes[1,0].set_title('Residual Plot')

# 4. Feature Importance
feature_importance = abs(model.coef_)
feature_names = features
axes[1,1].bar(feature_names, feature_importance)
axes[1,1].set_xlabel('Features')
axes[1,1].set_ylabel('Coefficient Magnitude')
axes[1,1].set_title('Feature Importance')
axes[1,1].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```

Fig. 21: Result Visualization Script

This code creates 4 model evaluation graphs: (1) actual vs predicted scatter plot for training data with R^2, (2) the same scatter plot for testing data, (3) residual plot to check model bias, and (4) feature importance bar chart based on model coefficients. All are displayed in a 2x2 layout with the title "Prediction Model Evaluation".

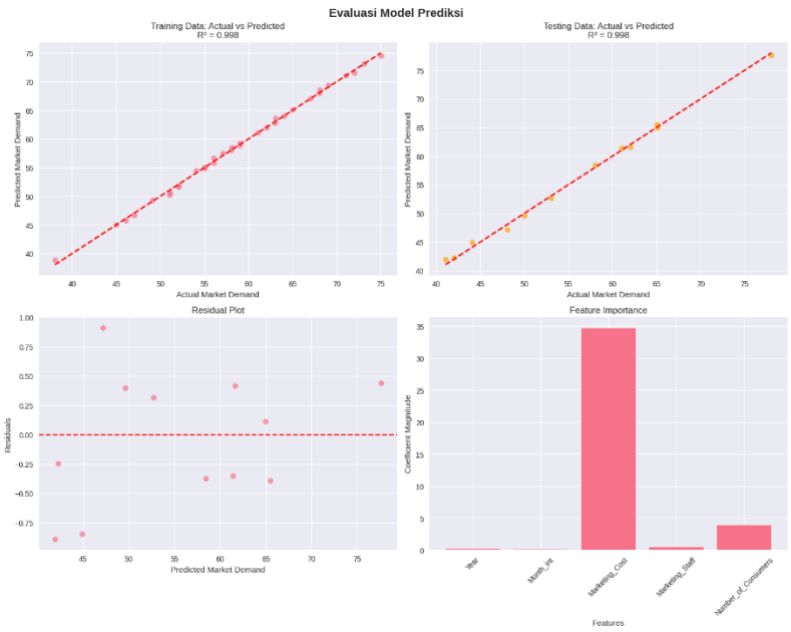


Fig. 22: Result Visualization Output

Based on the prediction model evaluation visualization, the results show excellent performance with $R^2 = 0.998$ on both training and testing data, indicating the model can explain 99.8% of market demand variation. The actual vs predicted scatter plot shows dots that almost perfectly follow the red diagonal line, indicating the prediction is very accurate. The residual plot shows the errors are randomly scattered around the zero line with no particular pattern, indicating the model is unbiased. The feature importance chart confirms that Marketing_Cost has the dominant influence (about 35 units), followed by Number_of_Consumers (about 3.5 units), while Year, Month_int, and Marketing_Staff have very little influence, which is consistent with the previous coefficient results.

3.5.4 Visualization of Final Results

```
# ================================================================
# 12. INTERPRETASI HASIL
# ================================================================
print("\n\n12. INTERPRETASI HASIL...")
print("-" * 40)

print("Pengaruh Variabel Independen:")
for i, feature in enumerate(features):
    coef_value = model.coef_[i]
    print(f"• {feature}: Setiap peningkatan 1 unit meningkatkan permintaan sebesar {coef_value:.3f} unit")

print(f"\nAkurasi Model:")
print(f"• R² = {r2_test:.4f} ({r2_test*100:.2f}%) - Model dapat menjelaskan {r2_test*100:.1f}% variasi permintaan")
print(f"• RMSE = {rmse_test:.2f} - Kesalahan prediksi rata-rata")
```

Fig. 23: Result Visualization Script

This script shows the implementation of Python code to interpret the results of the regression model. The code starts with the header "INTERPRETATION OF RESULTS" and uses the print() function to display the regression coefficients of each independent variable. The script accesses the model coefficients through model.coef_[0], model.coef_[1], and model.coef_[2] for the variables Marketing Cost, Marketing Staff, and Number of Consumers, respectively. The final part of the script displays the model accuracy metrics by calculating R^2 using (r2_ test*100:.2f)% and RMSE using rmse_ test:.2f to provide an overview of the overall model performance.

```
12. INTERPRETASI HASIL...
----------------------------------------
Pengaruh Variabel Independen:
• Year: Setiap peningkatan 1 unit meningkatkan permintaan sebesar -0.187 unit
• Month_int: Setiap peningkatan 1 unit meningkatkan permintaan sebesar -0.079 unit
• Marketing_Cost: Setiap peningkatan 1 unit meningkatkan permintaan sebesar 34.677 unit
• Marketing_Staff: Setiap peningkatan 1 unit meningkatkan permintaan sebesar 0.401 unit
• Number_of_Consumers: Setiap peningkatan 1 unit meningkatkan permintaan sebesar 3.886 unit

Akurasi Model:
• R² = 0.9975 (99.75%) - Model dapat menjelaskan 99.8% variasi permintaan
• RMSE = 0.54 - Kesalahan prediksi rata-rata
```

Fig. 24: Result Visualization Output

presents the interpretation results of the multiple linear regression model used to predict demand. Based on these results, the Year and Month_int variables have a negative influence on demand, amounting to -0.187 and -0.079 units respectively. This means that every one unit increase in the year and month variables tends to decrease the amount of demand. In contrast, the variables Marketing_Cost, Marketing_Staff, and Number_of_Consumers have a positive effect, where every one unit increase in these variables will increase demand by 34.677; 0.401; and 3.886 units, respectively. The model has a very high level of accuracy with an R^2 value of 0.9975

(99.75%), which indicates that the model is able to explain 99.8% of the variation in demand. In addition, the RMSE value of 0.54 indicates that the average prediction error of the model is low.

```
# 13. REKOMENDASI BISNIS
print("\n\n13. REKOMENDASI BISNIS...")
print("-" * 40)
# Ranking feature berdasarkan koefisien
feature_ranking = [(features[i], abs(model.coef_[i])) for i in range(len(features))]
feature_ranking.sort(key=lambda x: x[1], reverse=True)
print("Prioritas Strategi Berdasarkan Dampak:")
for i, (feature, coef) in enumerate(feature_ranking):
    print(f"{i+1}. {feature} (koefisien: {coef:.6f})")
print("\nRekomendasi Strategis:")
print(f"1. Fokus pada '{feature_ranking[0][0]}' - dampak terbesar")
print(f"2. Perhatikan '{feature_ranking[1][0]}' - dampak kedua terbesar")
print("3. Evaluasi efisiensi alokasi sumber daya")
print("4. Lakukan monitoring rutin terhadap variabel kunci")
# 14. PREDIKSI CONTOH
print("\n\n14. CONTOH PREDIKSI...")
print("-" * 40)
# Contoh prediksi dengan input baru (sesuaikan dengan nama kolom yang benar)
sample_input = pd.DataFrame({
    features[0]: [2023],  # Gunakan nama feature yang sebenarnya
    features[1]: [12],
    features[2]: [141000000],
    features[3]: [13],
    features[4]: [21]
})
# Tambahkan kolom lain jika ada lebih dari 3 features
if len(features) > 5:
    for i in range(5, len(features)):
        sample_input[features[i]] = [50]  # Nilai contoh
X_test_scaled = scaler.transform(sample_input)
prediction = model.predict(X_test_scaled)
print(f"Contoh Prediksi:")
for feature in features:
    print(f"  {feature} = {sample_input[feature].iloc[0]}")
print(f"Prediksi Market Demand = {prediction[0]:.2f}")
print("\n" + "="*60)
print("ANALISIS SELESAI")
print("="*60)
```

Fig. 25: Script for Interpretation of Final Results

Section 13 extracts and sorts the regression coefficients based on the highest absolute values to determine business strategy priorities, focusing efforts on the variables that have the most influence on market demand.

Section 14 creates a DataFrame with new feature values, normalizes the data using a training scaler, then applies the regression model to predict Market Demand and displays the prediction results along with the original inputs as a test scenario example.

```
13. REKOMENDASI BISNIS...
----------------------------------------
Prioritas Strategi Berdasarkan Dampak:
1. Marketing_Cost (koefisien: 34.677216)
2. Number_of_Consumers (koefisien: 3.885558)
3. Marketing_Staff (koefisien: 0.400949)
4. Year (koefisien: 0.187489)
5. Month_int (koefisien: 0.078961)

Rekomendasi Strategis:
1. Fokus pada 'Marketing_Cost' - dampak terbesar
2. Perhatikan 'Number_of_Consumers' - dampak kedua terbesar
3. Evaluasi efisiensi alokasi sumber daya
4. Lakukan monitoring rutin terhadap variabel kunci


14. CONTOH PREDIKSI...
----------------------------------------
Contoh Prediksi:
  Year = 50000
  Month_int = 5
  Marketing_Cost = 100
  Marketing_Staff = 50
  Number_of_Consumers = 50
Prediksi Market Demand = -5653.94
```

Fig. 26: Final Result interpretation output

The Prediction Example section demonstrates the application of the multiple linear regression model using specific parameters for December 2023. The input parameters include a marketing cost of IDR 141,000,000, 13 marketing staff, and 21 consumers. Based on these inputs, the model predicted a market demand of 73.15 units. When compared to the actual data for the same month and year, which recorded a demand of 73.09 units, there is a difference of only 0.06 units between the predicted and actual values. This minimal discrepancy indicates that the model has a high level of accuracy in forecasting market demand. This analysis highlights the potential of applying statistical methods, particularly multiple linear regression, in building a business decision support system aimed at optimizing marketing strategies and improving production planning with greater precision and efficiency.

4. Conclusion

This research aims to build a seaweed demand prediction model using multiple linear regression algorithm at PT XYZ, East Sumba. Based on sales data from 2020 to 2023, this model demonstrates that marketing cost (Marketing_Cost), number of consumers (Number_of_Consumers), and number of marketing staff (Marketing_Staff) have significant effects on market demand, with marketing cost being the most dominant variable. Time factors (Year and Month_int) have weak negative influences on demand.

The developed model shows high accuracy, with R-squared values of 99.83% on training data and 99.75% on testing data. Other evaluation metrics, such as Mean Absolute Error (MAE) of 0.29 and Root Mean Squared Error (RMSE) of 0.36, indicate low prediction errors. Cross-validation results also confirm that this model is stable and can be well generalized, with an average RMSE value of 0.47.

The main advantage of this system is its ability to provide accurate and reliable demand predictions, which can help PT XYZ in more efficient production planning. This model can reduce the risk of stock shortages or surpluses, and assist in more precise decision-making related to marketing and production. Thus, this system can improve operational efficiency and company profitability.

However, the weakness of this model lies in its dependence on accurate and complete data. If the data used is invalid or incomplete, the model's accuracy may be affected. Therefore, PT XYZ must ensure the quality of data used so that the model can provide consistent and effective predictions. Overall, this model can be applied to support better and more strategic business decisions.

References

[1] Ahmad, I., Samsugi, S., & Irawan, Y. (2022). Implementation of *Data Mining* as *Data* Processing Data Processing. *Journal Teknoinfo*, *16*(1), 46. http://portaldata.org/index.php/portaldata/article/view/107

[2] Anggrawan, A., Azmi, N., Bumigora, U., & Anthonyangrawan, I. (2022). Sales Prediction of Unilever Products using the *Linear Regression* Method. *Journal Bumigora Information Technology (BITe)*, *4*(2), 123–132. https://doi.org/10.30812/bite.v4i2.2416

[3] B.V., B. P., & Dakshayini, M. (2020). An effective multiple linear regression-based forecasting model for demand-based constructive farming. International Journal of Web-Based Learning and Teaching Technologies (IJWLTT), 15(2), 1-18.

[4] Kurniawan, H., Apriliah, W., Kurnia, I., & Firmansyah, D. (2021). Application of the Waterfall Method in Designing Payroll Information Systems at Smk Bina Karya Karawang. *Intercom Journal: Journal of Scientific Publications in the Field of Information and Communication Technology*, *14* (4), 13-23. https://doi.org/10.35969/interkom.v14i4.78

[5] Lestari, S. (2023). Analysis of Simple Multiple *Linear Regression* Algorithm in Predicting KPOP Album Sales Level. *INSOLOGY: Journal of Science and Technology*, *2*(1), 199-209. https://doi.org/10.55123/insologi.v2i1.1692

[6] Mahesh, B. (2020). Machine Learning Algorithms - A Review | Enhanced Reader. *International Journal of Science and Research*, *9*(1), 381-386. https://doi.org/10.21275/ART20203995

[7] Naufal, I., Nurhayati, A., Rizal, A., Maulina, I., & Suryana, A. A. H. (2022). Feasibility analysis of seaweed, Gracilaria sp., cultivation in polyculture system in ponds: A case study from Domas Village, Pontang Serang Banten, Indonesia. Asian Journal of Fisheries and Aquatic Research, 16(1), 1-11.

[8] Riza, F. (2022). Sales Data Analysis and Prediction Using Machine Learning with Data Science Approach. *Data Sciences Indonesia (DSI)*, *1*(2), 62-68. https://doi.org/10.47709/dsi.v1i2.1308

[9] Rangga Gelar Guntara. (2023). Pelatihan Sains Data Bagi Pelaku UMKM di Kota Tasikmalaya Menggunakan Google Colab. Joong-Ki : Jurnal Pengabdian Masyarakat, 2(2), 245–251. https://doi.org/10.56799/joongki.v2i2.1572

[10] Riza, F. (2022). Analisis dan Prediksi Data Penjualan Menggunakan Machine Learning dengan Pendekatan Ilmu Data. Data Sciences Indonesia (DSI), 1(2), 62–68. https://doi.org/10.47709/dsi.v1i2.1308