

Forecasting of Catfish Sales by Time Series Using the SARIMA method

Nandia Rani Woro Tri Handayani¹, Martinus Maslim², Paulus Mudjihartono³

^{1,2,3}Department of Informatics, Universitas Atma Jaya Yogyakarta, Indonesia

Email: ¹150708395@students.uajy.ac.id, ²martinus.maslim@uajy.ac.id, ³paulus.mudjihartono@uajy.ac.id

Abstrak. Sistem informasi yang mengotomatiskan proses bisnis, terutama dengan persyaratan khusus masih relevan saat ini. Clarias Makmur, sebuah usaha mikro di Indonesia yang membiakkan dan menjual ikan lele menggunakan sistem informasi ini untuk menjalankan penjualan, pengeluaran, modal, dan pelaporan mereka. Penjualan ikan lele sebagai makhluk hidup memiliki ciri khas tersendiri yang menunjukkan pola musiman yang unik. Sebuah model bernama SARIMA (Seasonal Autoregressive Integrated Moving Average) kemudian diusulkan untuk memprediksi penjualan. Lebih lanjut, sistem yang disebut SITRAN dibuat online supaya perusahaan mikro ini beroperasi secara fleksibel. Ada 400 data penjualan yang digunakan dalam metode ini untuk membuat model dan prediksi, sedangkan 100 data lainnya digunakan untuk menguji akurasi metode. Hasil penelitian menunjukkan bahwa SARIMAX (21,2,0) (1,0,0,12) adalah model terbaik yang ditemukan dalam percobaan dan yang memberikan RMSE terkecil.

Kata kunci: prediksi; penjualan ikan lele; pola musiman; SARIMA

Abstract. An information system that automates a business process, especially with specific requirement is still relevant these days. Clarias Makmur, a micro cooperation in Indonesia that breeds and sells catfish uses such an information system to carry out their sales, expenses, capital and reporting. The sales of catfish as a living creature have their own characteristics showing the unique seasonal pattern. A model named SARIMA (Seasonal Autoregressive Integrated Moving Average) is then proposed to predict the sales. Furthermore the system called SITRAN is made to be online for the cooperation to operate flexibly. There are 400 sales data used for the method to model and predict, while another 100 are used to test the method accuracy. The result shows that SARIMAX(21,2,0)(1,0,0,12) is the best model found in the experiment giving the smallest RMSE.

Keywords: prediction; catfish sales; seasonal pattern; SARIMA

1. Introduction

Survey done by Badan Pusat Statistik (Central Bureau of Statistics) abbreviated as BPS says that Indonesia economics indicates growth. This is especially on Gross Domestic Product of employment of the field of agriculture, forestry and fisheries that grows 3.91% in 2018 compared that in 2017 [1]. The economic growth also comes with a decrease in unemployment in 2018 by 0.57% according to the Indonesia Labor report in the IV Quarter of 2018 made by BPS [2]. Although the percentage is small, nevertheless the absolute number is relatively big. One of the efforts to suppress this unemployment number is empowering and developing a micro business [3]. Since Indonesia has the potential in natural resources, one of the potential businesses would be freshwater aquaculture, especially in that of catfish [4]. Catfish becomes the choice of many farmers because its simple nature adapting to its environment, even in environments with poor water conditions or even with limited water [5]. Other than that, Indonesia also has vast areas with enough water [6]. All of these conditions have been met by a business called "Clarias Makmur" which is a catfish farming business in Banjarnegara, Central Java, which has been ongoing since August 2018.

Clarias Makmur is a micro cooperation that sells catfish which using information technology in automating its business process. This information technology takes a form of an online information system which is intended to manage the data of transactions and sales. The system is required to be ubiquitous since the manager needs to be able to access it from anywhere.

The manager should be responsible of predicting the number of sales for the next whole month. Hence this sales data in future is essential. The prediction is done by using SARIMA method. This SARIMA term stands for Seasonal Autoregressive Integrated Moving Average, which is extension of ARIMA where seasonal factor added. In this research monthly data is used and fed to SARIMA.

Several researchers use SARIMA in their forecasting projects to deal with series data. For example the consumption of electricity is one of the SARIMA forecasting projects. A group of researcher conducted forecasting of the University Tun Hussein Onn Malaysia electricity consumption a year of 2019 [7]. The data are taken from January 2009 to December 2018 over 120 observations. The result shows that SARIMA(0,1,1)(0,1,1) is the best model for the case. Another example of SARIMA forecasting is the Kenya's inflation rate prediction. Inflation rate data are taken from Kenya National Bureau of Statistics (KNBS) period 1981 to 2013. This forecasting shows the best model of SARIMA(0,1,0)(0,0,1) [8]. In meanwhile, Ref [9] shows the application of SARIMA for predicting monthly mean rainfall in Rayalaseema (India). This R application found that the best model is ARIMA(5,0,1)(2,0,0). Similarly, another group of researcher generate synthetic monthly rainfall at Sinu river watershed in Colombia [10]. They used SARIMA approach as well. The goal is that the result can be useful for backing up the policy of water usage in the area.

Forecasting the coffee production is another example of SARIMA application. Maulana et al predict the Indonesian coffee production by using data taken from Indonesia Central Agency on Statistics (BPS) period January 2009 to December 2013. It is found that the best model for forecasting is SARIMA(2,1,0)(1,1,1) [11]. A research conducted by Apriyanto says that the success of information systems based on size giving small project into a success rate of 40% while based on its complexity giving a low one has a success rate of 51% [12]. Clarias Makmur is an early business entity which is expected to have a large success rate to help the business continuity especially a small catfish aquaculture business. Based on the need of supplying catfish accurately for the some months ahead, Clarias Makmur needed to predict for the coming catfish sales monthly. Therefore, the application of an information system named SITRAN Clarias Makmur (Clarias Makmur Transaction Information System) developed in web based and python backend program.

The development method of this system is Rapid Application Development (RAD). RAD is an object oriented approach to system development which includes a development method and software [13]. The application of the RAD method is applicative, practical, and easy to adjust for the needs of the system with a small to moderate coverage compared to the waterfall method. This paper aims an implementation of the SARIMA forecasting for the catfish sales.

The organization of the rest of the paper is as follows. Section II presents related works; Section III presents the implementation of the system; Section IV provides the results of the prediction; and Section V summarizes the conclusions. Lastly, Section VI lists the references.

2. Related Works

There are many research that their goal is to develop information system such as e-commerce for clothing at Distro Bahana Shop [14]. This research aims at its conventional sales process business of that shop. Its transaction was still done on site back then. As a result, transaction processes, cash flow generation, and reporting are very time consuming. The owner wanted to promote their product online through an information system. The system is then designed for buyer user and administrator user. Administrator controls stock, sales, and some other administering stuff. These jobs are among of the most important things that information system should have so that the owner is significantly assisted.

The next research conducted by Sriyanto with the title (originally in Indonesian) "The Design of the Information System of Gold Shop" [15] proposed a such information system. In that research, one of the goals of developing information system is to increase work performance in term of selling-buying process and the final recapitulation of "Semar" gold shop. They face the problem in converting gold weight unit into currency that keeps changing. This problem slows

the administration process in generating debt-to-supplier book. The proposed system handles the gold buying from supplier and its conversion to currency, gold classification, customer transactions, and financial report. Eventually, the end users which are the employees of the shop themselves do these things assisted by the system. According to the research the mandatory aspect is that the information system should be able to generate transaction and financial report so that the owner can view report regularly.

Another related research proceeded by Fajar Firmansyah with the topic “Integration of the Internal Control in Cooperation Management Information System” [16] built a web based information system. Its goal is to internally control the standard operational procedures whether it has been compromised or not by the cooperation employees. It has also been integrated into the existing system with features of deposits, savings, and loan. To enhance the security of the SOP transactions, a sequential of approvals are made for the employees in order not to commit mischief. This security should have been applied to the developing system but since the catfish cattle business is still an emergent business, the security only applied to the transaction history done by the employees.

3. Implementation of the System

In developing the system of SITRAN, some tasks are performed carefully yet quickly. First task is to design a physical data model (PDM) which is finally become a database in the server. Fig. 1 shows the PDM of the system which consists of several tables and relations. There two important tables in the PDM. Table First, table **Sales** takes a central role. This table consists of sale transactions which contains total (sales total), and date of sale. The detail of the sale transaction is placed on table **Detail_sales**. The latter table comprises the number of catfish bought in a transaction and what size they are in.

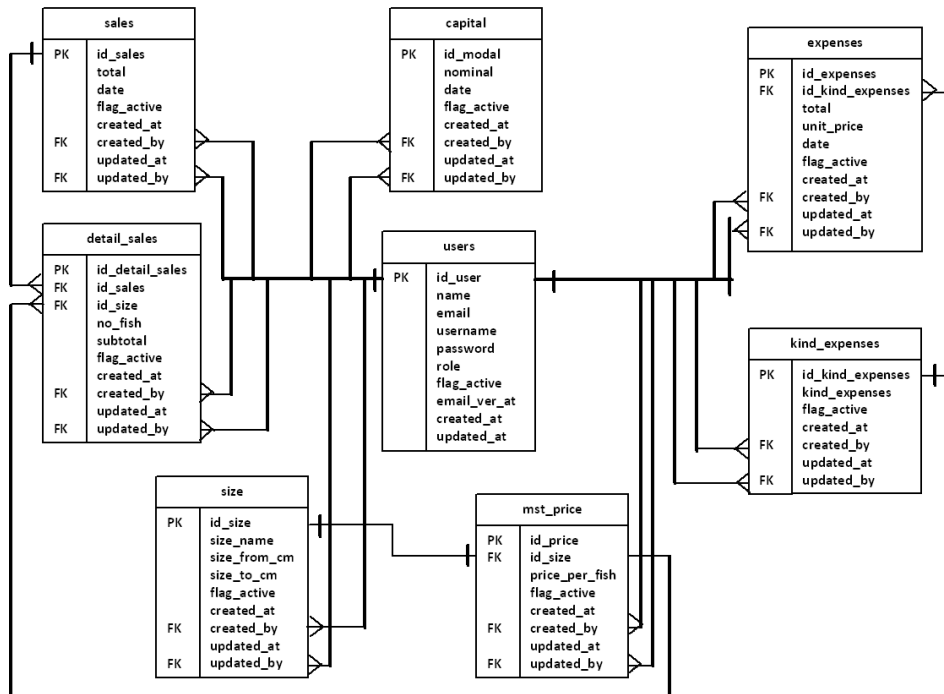


Figure 1. Physical Data Model of SITRAN

Model of SARIMA in SITRAN is implemented by making an API (Application Programming Interface). This API is made by coding a Python program and using Flask framework. Python becomes the first choice since this language has a library that can perform SARIMA forecasting. The library is statsmodels.api. Statsmodel.api is a python package that includes cross-sectional model and methods.

The cross-sectional is a study that assesses the connection of the extensive dataset of cases and the connection of inter-variables in certain point in time. The number of cases and variables is what makes inter-section analysis possible, that is, analysis between many cases and many variables. In this case, the connection itself is between catfish sales data and the months the sales take place. The sales data as a matter of fact, is monthly seasonal.

The use of SARIMA model is considered proper in calculating catfish sale data for that happens to have seasonal pattern, as shown on Fig. 2. Fig. 2 shows monthly raw data of catfish sales in sample of two years. The actual data is taken directly from the business person. It is obvious that this graph has a trend yet seasonal.

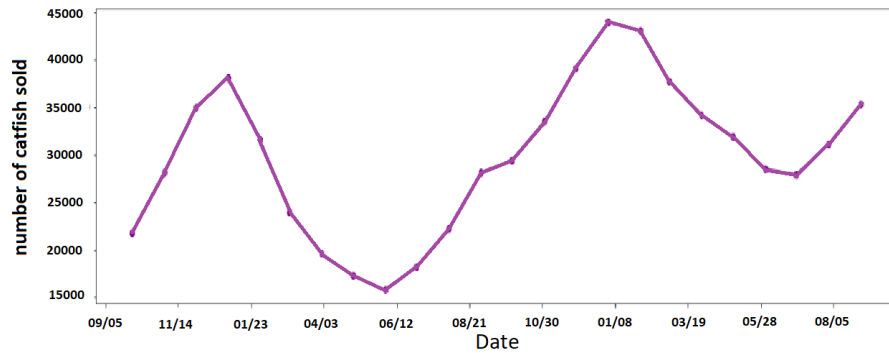


Figure 2. Catfish Sales data in two years

On the other hand, Fig. 3 shows the result of analysis of decomposition seasonal function belongs to statsmodels.api package. Graph 3b shows the trend factor. It is obvious that the trend is ascending. Furthermore, Graph 3c confirms the seasonal factor. Its peak happens in every Januaries while its lowest point happens in every Junes. Finally the observed graph as shown in Graph 3a summarizes the trend and the seasonal factors. Based on the analysis, sale data has an ascending trending pattern and also patterned seasonal. Hence SARIMA method is taken into account to be the choice method in this research.

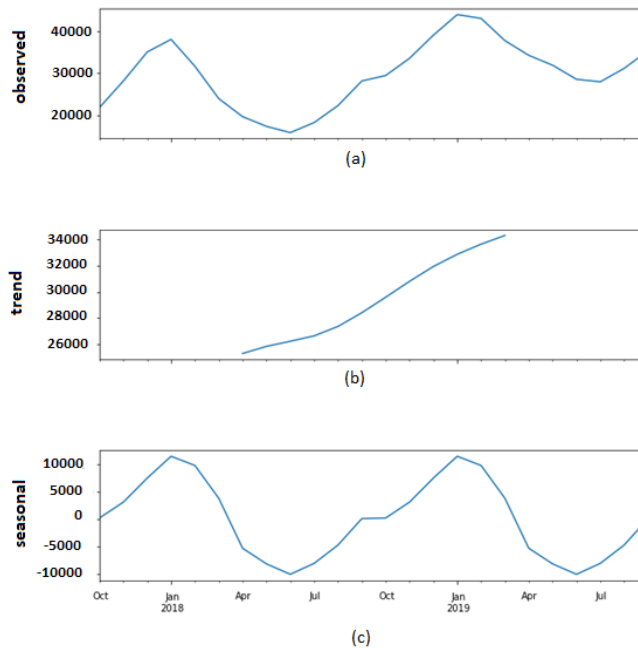


Figure 3. Decomposition seasonal results over the whole sale data

First step to build the SARIMA model is to create API which sends data to python web application. The applications then process these data to make SARIMA forecast model with the

code as shown in Code 1. The code shows the function called `getDataForecast()` which simply contains the query of the sales data.

```
Route::get('/dataRawForecast', 'HomeController@getDataForecast');

public function getDataForecast() {
    $data=DB::table('trn_penjualan')
        ->join('detail_penjualan',
            'trn_penjualan.id_penjualan', '=', 'detail_penjualan.id_penjualan')
        ->select(DB::raw("sum(detail_penjualan.jumlah_ikan)
            as jumlah_ikan, year(trn_penjualan.tanggal) year, month(trn_penjualan.tanggal)
            month"))
        ->whereraw('trn_penjualan.flag_active= "1"
            and detail_penjualan.flag_active = "1" and trn_penjualan.tanggal <
            (SELECT date_sub(last_day(CURRENT_DATE), interval 1 month))')
        ->groupby('year', 'month')
        ->orderby('year', 'asc')
        ->orderby('month', 'asc')
        ->get();
    return ($data);
}
```

Code 1. Getting sales data from MySQL database to web application

API is then called in python web application for the data to be processed using request package and json. The code how request package and json used to make a dataframe is shown in Code 2. Once API result is capture, json is used to convert them in a dataframe format.

```
URL='https://lele.learnme.web.id/api/dataRawForecast'
headers={'Accept':'application/json'}
r=requests.get(url=URL,headers=headers)
data=json.loads(r.text)
df = pd.io.json.json_normalize(data)
```

Code 2. Converting data resulted from API to dataframe

Subsequently data become a dataframe where index is the monthly and yearly period. These data are ready to proceed with SARIMAX function then. Moreover an iteration over the data is done for obtaining the fittest Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). The iteration for finding the best model performed by executing the following code as shown in Code 3.

```

rmse_min=100000000
p= range(0, 25)
d = q = range(0, 5)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]
#Loop for search model
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(df,order=param,seasonal_order=param_seasonal)
            results = mod.fit()
            pred = results.get_prediction(start=pd.to_datetime('2019-06-01'), dynamic=False)
            y_forecasted = pred.predicted_mean
            mse=mean_squared_error(y_truth, y_forecasted)
            rmse=np.sqrt(mse)
            mae=metrics.mean_absolute_error(y_truth, y_forecasted)
            if rmse<rmse_min:
                rmse_min=rmse
                print(param,param_seasonal)
                print('The Mean Squared Error is {}'.format(round(mse, 2)))
                print('The Root Mean Squared Error is {}'.format(round(rmse, 2)))
                print('The Mean Absolute Error is {}'.format(round(mae, 2)))
                print(y_truth)
                print(y_forecasted)
                print('\n')
        except:
            continue

```

Code 3. Iteration for finding best model with fittest MSE and RMSE

As shown in Code 3 model searching is done by applying parameter p (order of the autoregressive, range 0 to 25), parameter d (degree of the differencing, range 0 to 5) and q (order of moving average, range 0 to 5). Firstly, in experiment it is found that parameter d and q that exceed 5 will cause the python code mostly crashes due to the complexity. Afterward, code will pass two iteration for finding parameter order and seasonal respectively. Every trial of p , d and q is evaluated and determine whether the RMSE found minimum. It passes until the best model found.

4. Experiment Result

The next step is to find MSE and RMSE which is taken from four months: June 2019 to September 2019. The MSE and RMSE is calculated from comparing the actual data and forecast result during those four months. MSE is an average of squared differences between actual data and forecast data while RMSE is the squared-root of the MSE. The expected MSE or RMSE is a small value that reflects the size of the error of the forecast.

The following table (Table 1) is the result of the code execution for finding the best model. Column Model shows the p , d and q parameter which determine the model. They are evaluated in each iteration, or as in the rows of Table 1. It stops on RMSE of 88.72 with the model as shown on the model column.

Table 1. MSE/RMSE Results from various models

No	Model	MSE	RMSE	Perbandingan								Prediksi penjualan	
				Juni 2019		Juli 2019		Agustus 2019		19-Sep		Okt-2019	19-Nov
				Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi		
1	(0,0,0)(0,0,0,12)	950,990,000.0 0	30,838.1 3	28,500.00	-	27,900.0 0	-	31,100.0 0	-	35,300.0 0	-	-	-
2	(0,0,0)(0,1,0,12)	96,385,625.00	9,817.62	28,500.00	15,800.0 0	27,900.0 0	8,200.00	31,100.0 0	22,250.0 0	35,300.0 0	28,100.0 0	29,400.0 0	33,500.0 0
3	(0,0,0)(0,2,0,12)	18,874,531.25	4,344.48	28,500.00	23,700.0 0	27,900.0 0	27,300.0 0	31,100.0 0	33,375.0 0	35,300.0 0	42,150.0 0	37,000.0 0	38,900.0 0
4	(0,0,2)(0,1,0,12)	15,127,231.34	3,889.37	28,500.00	23,546.7 4	27,900.0 0	25,111.8 3	31,100.0 0	26,501.2 4	35,300.0 0	32,644.5 1	31,509.0 0	33,500.0 0
5	(0,1,0)(0,0,0,12)	9,865,625.00	3,140.96	28,500.00	31,850.0 0	27,900.0 0	28,500.0 0	31,100.0 0	27,900.0 0	35,300.0 0	31,100.0 0	31,100.0 0	31,100.0 0
6	(0,1,0)(0,1,0,12)	3,966,875.00	1,991.70	28,500.00	30,350.0 0	27,900.0 0	30,900.0 0	31,100.0 0	31,950.0 0	35,300.0 0	36,950.0 0	38,250.0 0	42,350.0 0
7	(0,1,1)(0,1,0,12)	3,200,346.14	1,788.95	28,500.00	30,317.4 7	27,900.0 0	30,524.2 8	31,100.0 0	31,407.4 9	35,300.0 0	36,886.4 3	38,186.0 0	42,286.0 0
8	(0,1,2)(0,1,0,12)	2,577,255.46	1,605.38	28,500.00	30,150.7 0	27,900.0 0	30,430.0 9	31,100.0 0	30,830.3 0	35,300.0 0	36,353.6 3	37,725.0 0	41,825.0 0
9	(0,1,4)(1,0,0,12)	1,718,998.24	1,311.11	28,500.00	27,060.1 2	27,900.0 0	26,436.7 1	31,100.0 0	30,737.8 0	35,300.0 0	33,709.3 1	35,778.0 0	37,353.0 0
10	(1,0,1)(1,0,0,12)	1,211,269.22	1,100.58	28,500.00	29,594.7 1	27,900.0 0	29,105.6 3	31,100.0 0	29,949.9 6	35,300.0 0	36,233.0 4	36,753.0 0	39,667.0 0
11	(2,0,2)(0,1,0,12)	911,033.32	954.48	28,500.00	27,337.2 1	27,900.0 0	27,490.8 9	31,100.0 0	31,372.2 9	35,300.0 0	36,731.9 7	38,275.0 0	42,234.0 0
12	(2,1,4)(0,0,0,12)	228,328.78	477.84	28,500.00	27,734.8 3	27,900.0 0	27,828.5 7	31,100.0 0	31,140.6 2	35,300.0 0	34,733.3 6	36,720.0 0	36,722.0 0
13	(2,1,4)(1,0,0,12)	79,955.33	282.76	28,500.00	28,282.2 0	27,900.0 0	28,357.3 7	31,100.0 0	31,100.7 4	35,300.0 0	35,551.3 9	37,432.0 0	38,243.0 0

14	(7,0,1)(0,1,0,12)	40,440.67	201.10	28,500.00	28,381.17	27,900.00	27,675.80	31,100.00	30,841.26	35,300.00	35,474.43	35,870.00	40,413.00
15	(8,0,1)(0,1,0,12)	40,392.19	200.98	28,500.00	28,257.81	27,900.00	27,702.96	31,100.00	30,894.46	35,300.00	35,447.79	35,975.00	40,526.00
16	(14,0,4)(0,0,0,12)	35,505.84	188.43	28,500.00	28,338.03	27,900.00	28,154.58	31,100.00	31,011.10	35,300.00	35,092.45	38,827.00	39,256.00
17	(18,2,1)(0,0,0,12)	27,326.63	165.31	28,500.00	28,179.46	27,900.00	27,893.07	31,100.00	31,035.64	35,300.00	35,251.30	38,132.00	33,887.00
18	(19,2,1)(0,0,0,12)	17,874.34	133.69	28,500.00	28,524.60	27,900.00	27,848.48	31,100.00	31,249.60	35,300.00	35,085.86	38,155.00	34,186.00
19	(19,2,2)(0,0,0,12)	12,136.88	110.17	28,500.00	28,305.79	27,900.00	27,910.15	31,100.00	31,044.22	35,300.00	35,212.73	38,789.00	34,092.00
20	(21,2,0)(1,0,0,12)	7,870.93	88.72	28,500.00	28,338.11	27,900.00	27,861.00	31,100.00	31,159.43	35,300.00	35,285.04	31,535.00	28,533.00

Based on the result shown in Table 1, the best model is SARIMAX (21,2,0) (1,0,0,12). This model is ready to be applied on the sales forecasting in python application. Below is the code for providing forecast data resulted from the executed model. See Code 4. The forecast result is converted into an array to separate its data column and its index. Subsequently the array is further converted into json data form as the result of API call. As seen in the code of Figure 7, the json data converted from the API data which is lastly normalized. This normalized data becomes the data frame where in turns some fields data taken from it, such as the number of catfish and the date of transaction. Afterward, the date is set as a range of date according to the seasonal factor. Finally, a mean of the prediction computed and stored into the array in order that the program to traverse easily. Once the mean of the prediction obtained, that value will be sent as a prediction value.

```
@app.route('/forecastdata',methods=['GET'])
def forecast():
    URL = 'https://lele.learnme.web.id/api/dataRawForecast'
    headers = {'Accept': 'application/json'}
    r = requests.get(url=URL, headers=headers)
    data = json.loads(r.text)
    df = pd.io.json.json_normalize(data)

    month = df['month'].apply(str)
    year = df['year'].apply(str)
    y = month + '-01-' + year
    y = pd.to_datetime(y)
    x = df['jumlah_ikan'].apply(pd.to_numeric, errors='coerce')
    x_data = x.values
    y_date = list(pd.to_datetime(y))
    df = pd.DataFrame({'Date': y_date,
                      'Jumlah': x_data})
    df = df.set_index('Date')
    start = df.index[-3]
    end = start + relativedelta(months=+3)
    mod = sm.tsa.statespace.SARIMAX(df,order=(21,2,0),
                                   seasonal_order=(1, 0, 0, 12))
    results = mod.fit()
    pred = results.get_prediction(start=start, end=end, dynamic=True)

    #to array
    mean = round(pred.predicted_mean, 0)
    index = list(mean.index.values)
    index = np.asarray(index)
    value = mean.values
    mIndex = []
    for x in index:
        x = pd.to_datetime(x)
        mIndex.append(str(x.date()))

    arrMean = mean.values.tolist()
    mArrMean = []

    i = 0
    for x in arrMean:
        mData = [x, mIndex[i]]
        mArrMean.append(mData)
        i += 1

    #to dataframe
    dfMean = pd.DataFrame(data=mArrMean, columns=['mean', 'period'])
    dfMean=dfMean.to_json(orient='records')
    return dfMean
```

Code 4. Forecast code in Python

The forecasting result is implemented into website in the form of graph so that user is easy to understand. Users access the forecast result via dashboard which is even more comprehensible. Fig. 4 shows the connection of the forecast result which blue line is actual data while red line is the forecasted sales. There are some periods that have slight differences between actual and forecast data. Nevertheless, the differences are not that big so that it cannot be tolerated. On the contrary, the differences are acceptable thus the forecast is acceptable as well.

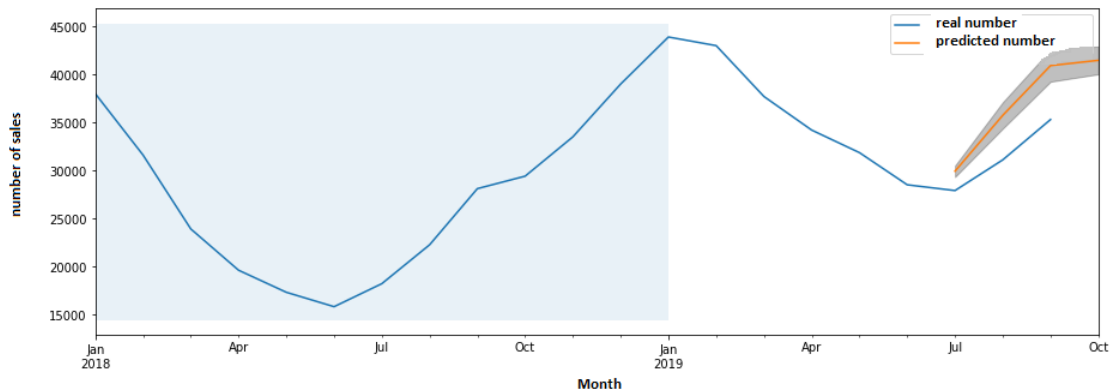


Figure 4. Graph of sales forecasting

Additionally, the trend of the predicting result shows the correct trend. However, the trend indicates a slight decrease at the end of the prediction where the real data do not exist yet. Since the real data have not come up yet, the validity cannot be confirmed. As it is true that the forecast formula includes the trend and seasonal factors, then the prediction value must also have been influenced by those factors. Hence, it can be predicted that the prediction also decreased when the trend decreasing. So, the limitation of the forecasting stands when the data test does not taken from the entire season. In this case, both trend and seasonal factors cannot be examined thoroughly.

5. Conclusions

Based on the analysis, design, implementation, and testing of SITRAN, it can be derived some conclusions. First, sales and expenses control that SITRAN does can fulfill the data and forecast requirements. It can also be operated online which escalate the flexibility and beneficial to the users. Second, SARIMA model implementation using python that has statmodels library for doing fish sales forecasting succeeds as desired. Third, SARIMAX model (21,2,0) (1,0,0,12) has proven to be the best model according to the experiment and yielded the acceptable forecasting model.

References

- [1] Badan Pusat Statistik, "Pertumbuhan Ekonomi Indonesia Triwulan IV-2018," No. 15/02/Th.XXII, 6 Februari 2019. [Online]. Available: <https://www.bps.go.id/pressrelease/2019/02/06/1619/ekonomi-indonesia-2018-tumbuh-5-17-persen.html>
- [2] Badan Pusat Statistik, "Keadaan Ketenagakerjaan Indonesia Agustus 2018," No. 92/11/Th. XXI, 05 November 2018. [Online]. Available: <https://www.bps.go.id/pressrelease/2018/11/05/1485/agustus-2018--tingkat-pengangguran-terbuka--tpt--sebesar-5-34-persen.html>
- [3] A. R. Fathurrohman, "Pengembangan usaha mikro melalui budidaya ikan lele sangkuriang," *EMPATI Jurnal Ilmu Kesejahteraan Sosial*, vol. 3, no. 1, pp. 61–70, 2018.
- [4] F. F. Riska, M. Primyastanto, and Z. Abidin, "Strategi pengembangan usaha budidaya ikan lele (*clarias sp.*) pada usaha perseorangan 'toni makmur' di kawasan agropolitan

- Desa Kauman Kecamatan Ngoro Kabupaten Jombang Jawa Timur,” *ESCOFIM Economic and Social of Fisheries and Marine Journal*, vol. 3, no. 1, pp. 48–54, 2015.
- [5] D. K. Dewi And J. H. Mulyo, “Analisis produksi budidaya ikan lele (*clarias gariepinus*): pendekatan fungsi produksi cobb douglas”, *Jurnal Perikanan Universitas Gadjah Mada*, vol. 17, no. 2, pp. 54–60, 2015.
- [6] S. R. Purwati, Zulkarnaini, and Hendrik, “Analisis usaha budidaya ikan lele dumbo (*clarias gariepinus*) menggunakan penerapan sistem bioflok dan sistem konvensional di Kelurahan Palas Kecamatan Rumbai Kota Pekanbaru Provinsi Riau,” *Jurnal Online Mahasiswa Fakultas Perikanan dan Ilmu Kelautan Universitas Riau*, vol. 5, no. 1, 2018.
- [7] S. E. Sim, K. G. Tay, A. Huong, and W. K. Tiong, "Forecasting electricity consumption using SARIMA method in IBM SPSS software", *Universal Journal of Electrical and Electronic Engineering*, vol. 6(5B), pp.103-114, 2019, [Online] doi: 10.13189/ujeee.2019.061614
- [8] S. W. Gikungu, A. G. Waititu, and J. M. Kihoro, “Forecasting inflation rate in Kenya using SARIMA model”, *American Journal of Theoretical and Applied Statistics*, vol. 4, no. 1, 2015, pp. 15-18, [Online] doi: 10.11648/j.ajtas.20150401.13
- [9] J. C. Ramesh Reddy, T. Ganesh, M. Venkateswaran, PRS Reddy, "Forecasting of monthly mean rainfall in Rayalaseema", *International Journal of Current Research and Review*, vol. 9, no. 24, December 2017, [Online] doi: 10.7324/IJCRR.2017.9244
- [10] L. Martínez-Acosta, J. P. Medrano-Barboza, Á. López-Ramos, J. F. R. López and Á. A. López-Lambrano, "SARIMA approach to generating synthetic monthly rainfall in the Sinú River watershed in Colombia", *Atmosphere*, vol. 11, pp. 1-16, 2020, [Online] doi:10.3390/atmos11060602
- [11] H. A. Maulana, K.W. Harahap, Adriyansyah, Rofiroh, F. Zainuddin, "Permodelan produksi kopi Indonesia dengan menggunakan Seasonal Autoregressive Integrated Moving Average (SARIMA) ", *Jurnal Saintika UNPAM*, vol. 2, no. 1, Juli 2019.
- [12] R. D. Apriyanto and H. P. Putro, “Tingkat kegagalan dan keberhasilan proyek sistem informasi di Indonesia,” *Seminar Nasional Teknologi Informasi dan Komunikasi 2018 (SENTIKA 2018)*, Yogyakarta, Indonesia, 2018, pp. 395–402.
- [13] R. Trimahardhika and E. Sutinah, “Penggunaan rapid development dalam perancangan sistem informasi perpustakaan,” *Jurnal Informatika*, vol. 4, no. 2, pp. 249-260, 2017.
- [14] W. Wulandari and A. Priyono, “E – commerce pakaian pada distro bahana shop,” *Jurnal TAM (Technology Acceptance Model)*, vol. 3, pp. 54–58, 2014.
- [15] S. Sriyanto, B. Purwanggono, and P. Nugroho, "Perancangan sistem informasi toko emas," *J@ti Undip: Jurnal Teknik Industri*, vol. 5, no. 1, pp. 15-24, Feb. 2012. [Online] doi: <https://doi.org/10.12777/jati.5.1.15-24>
- [16] F. Firmansyah, I M. Sukarsa, and P. W. Buana, “Integrasi kontrol internal pada sistem informasi manajemen koperasi,” *Jurnal Ilmiah Merpati Universitas Udayana*, vol. 6, no. 1, pp. 47–55, 2018.