

**PERAMALAN HARGA BERAS KUALITAS PREMIUM DI TINGKAT
PENGGILINGAN MENGGUNAKAN METODE ARIMA DAN SMA
(Studi Kasus: Data Harga Beras Kualitas Premium di Tingkat Penggilingan periode
Januari 2014 – Desember 2023)**

Zahra Khotimah, Febiola br Tarigan

11210940000003, 11210940000015

Mahasiswa Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam
Negeri Syarif Hidayatullah Jakarta

ABSTRAK

Indonesia merupakan negara dengan mayoritas masyarakatnya mengonsumsi nasi sebagai makanan pokok. Indonesia menjadi negara penghasil beras terbesar keempat global, dengan produksi 34 juta metrik ton pada musim 2022/2023. Beras sendiri pada tingkat penggilingan di Indonesia menurut kualitas dibagi menjadi tiga yakni beras Premium, Medium, dan Luar Kualitas/Rendah. Pada penelitian ini akan menggunakan data harga beras premium di tingkat penggilingan periode Januari 2014 – Desember 2023. Dengan menggunakan metode ARIMA, dihasilkan rata-rata prakiraan harga beras premium 12 bulan kedepan sebesar Rp11.107,45 dan nilai keakuratan MAPE sebesar 8.24%. Sedangkan metode SMA menghasilkan rata-rata harga beras premium 12 bulan kedepan sebesar Rp12.237,90 dan nilai keakuratan MAPE sebesar 3.52%.

Kata Kunci: Harga Beras, ARIMA, *Simple Moving Average*

I. PENDAHULUAN

Indonesia merupakan negara dengan mayoritas masyarakatnya mengonsumsi nasi sebagai makanan pokok. Berdasarkan data *United States Department of Agriculture* (USDA), Indonesia menjadi negara penghasil beras terbesar keempat global, dengan produksi 34 juta metrik ton pada musim 2022/2023. Namun, pada periode tersebut konsumsi beras Indonesia lebih besar dari produksinya yakni sebesar 35,7 juta metrik ton. Beras sendiri pada tingkat penggilingan di Indonesia menurut kualitas dibagi menjadi tiga yakni beras Premium, beras Medium, dan beras Luar Kualitas/Rendah. Menurut Permentan No. 31 Tahun 2017, beras Premium memiliki maksimum beras patah s.d. 15%, beras Medium sebesar 15,01% - 25% dan beras Luar Kualitas sebesar 25%.

Dengan menjadi makanan pokok masyarakat Indonesia, kestabilan harga beras sangat penting untuk diperhatikan karena apabila terjadi kenaikan harga maka akan memberatkan masyarakat. Berdasarkan data BPS, rata-rata harga beras Premium di tingkat penggilingan pada tahun 2023 sebesar Rp12.152,2/kg, beras Medium sebesar Rp11.724,5/kg, sedangkan beras Luar Kualitas sebesar Rp11.041,75/kg. Harga tersebut setiap bulannya mengalami kenaikan maupun penurunan harga, perubahan tersebut dapat terjadi karena berbagai faktor. Sebagai solusi, dapat dilakukan peramalan/prakiraan harga beras untuk beberapa bulan yang akan datang sehingga mampu membantu pemerintah dalam menetapkan kebijakan-kebijakan di masa mendatang.

Peramalan merupakan suatu pendekatan untuk menentukan sikap masa depan secara lebih baik dan rinci berdasarkan pengumpulan data historis dari periode sebelumnya hingga saat ini untuk meminimalkan tingkat kesalahan [1]. Ada berbagai macam metode yang dapat digunakan untuk melakukan peramalan, salah satunya dengan menggunakan suatu metode untuk data runtun waktu yaitu metode ARIMA (*Integrated Autoregressive Moving Average*). Model ARIMA dengan orde (p, d, q) merupakan model ARMA untuk data tidak stasioner hasil *differencing*. Pada tahun 2023, Virgania Sari dan Sylvia Ayu Hariyanto melakukan penelitian dalam peramalan harga beras premium bulanan di tingkat penggilingan menggunakan *Fuzzy Time Series Markov*. Penelitian tersebut menggunakan data harga beras premium dari Januari 2014 – Juli 2022 menghasilkan nilai keakuratan MAPE sebesar 0.81% dan nilai prakiraan harga beras premium pada bulan Agustus 2022 sebesar Rp9.627,99 [2].

Pada penelitian ini akan dilakukan peramalan harga beras premium dengan menggunakan data harga beras premium di tingkat penggilingan periode Januari 2014 – Desember 2023 menggunakan metode ARIMA. Sebagai perbandingan nilai keakuratan dari metode ARIMA, maka akan dilakukan juga menggunakan metode *Simple Moving Average* (SMA).

II. METODE PENELITIAN

Penelitian ini menggunakan data sekunder yang diperoleh dari Badan Pusat Statistik (BPS). Data yang digunakan adalah data *time series* harga beras kualitas premium di tingkat penggilingan yang mencakup periode bulanan dari Januari 2014 hingga Desember 2023 dengan jumlah 120 data dan satuan rupiah/kg.

	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
Jan	8.209	9.723	9.723	9.431	10.350	10.111	10.033	9.780	9.824	11.345
Feb	8.303	9.785	9.785	9.408	10.382	10.008	10.081	9.772	9.827	11.818

Mar	8.193	9.572	9.572	9.389	9.893	9.815	10.082	9.607	9.787	11.681
Apr	7.920	9.128	9.128	9.325	9.525	9.465	10.018	9.550	9.577	11.672
Mei	8.009	8.710	9.182	9.436	9.524	9.462	9.827	9.627	9.513	11.624
Jun	8.168	8.915	9.354	9.444	9.478	9.516	9.919	9.537	9.497	11.525
Jul	8.228	8.945	9.374	9.384	9.520	9.519	9.932	9.402	9.629	11.537
Ags	8.329	9.127	9.367	9.437	9.458	9.530	9.963	9.499	9.901	11.754
Sep	8.311	9.444	9.111	9.471	9.572	9.594	9.871	9.456	10.252	12.900
Okt	8.397	9.455	9.133	9.503	9.645	9.659	9.813	9.449	10.402	13.372
Nov	8.555	9.564	9.257	9.539	9.771	9.742	9.715	9.539	10.512	13.250
Des	9.018	9.664	9.342	9.860	9.818	9.838	9.788	9.673	10.954	13.348

Dalam mengolah data akan digunakan *software* RStudio dengan memanfaatkan beberapa *package* yang perlu di *install* terlebih dahulu.. Berikut ini merupakan tahapan-tahapan yang akan dilakukan dalam menganalisis data menggunakan metode ARIMA:

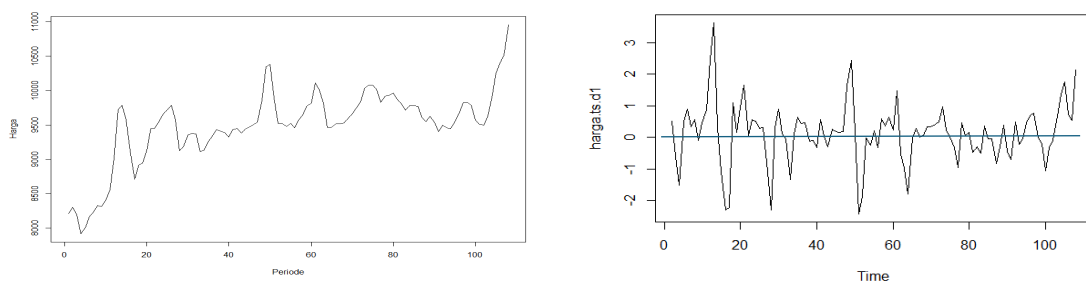
1. Mengumpulkan data sekunder, yaitu data harga beras kualitas premium di tingkat penggilingan dari Januari 2014 hingga Desember 2023 yang diperoleh dari Badan Pusat Statistik (BPS).
2. Membagi data menjadi dua bagian yaitu data *training* sebesar 70% - 90% dan data *testing* sebesar 10% - 30%. Pada penelitian ini, akan menggunakan data periode Januari 2014 – Desember 2022 sebagai data *training* dan data periode satu tahun terakhir yaitu Januari 2023 – Desember 2023 sebagai data *testing*. Nantinya, data *training* akan digunakan untuk membangun model ARIMA, sedangkan data *testing* untuk validasi model.
3. Membuat plot runtun waktu dari data *training*. Ini dilakukan untuk melihat pola harga beras dari data runtun waktu yang ada menggunakan suatu diagram atau grafik.
4. Melakukan pemeriksaan pada data, apakah data sudah stasioner atau belum. Untuk memeriksa kestasioneran dalam variansi dan juga mengatasinya yaitu dengan transformasi *Box-Cox*. Dan untuk memeriksa apakah data sudah stasioner dalam mean maka menggunakan uji *Kwiatkowski, Phillips, Schmidt and Shin* (KPSS) atau uji *Augmented Dickey Fuller* (ADF). Proses tidak stasioner dalam mean dapat diatasi dengan *de-trending* (trend deterministik) dan *differencing* (trend stokastik).
5. Melakukan identifikasi model yang paling cocok. Pada tahapan ini, akan dibuat plot *Autocorrelation Function* (ACF), *Partial Autocorrelation Function* (PACF), *Extended*

Autocorrelation Function (EACF), dan *Bayesian Information Criterion* (BIC). Model yang akan dihasilkan pada plot-plot tersebut, kemudian akan digunakan dalam mengestimasi model.

6. Mengestimasi parameter model. Beberapa metode estimasi diantaranya adalah metode *moment*, metode *least square*, dan metode *maximum likelihood*.
7. Pemeriksaan atau diagnosis apakah model memadai yaitu meliputi analisis residual dan overparameterisasi (*overfitting*). Pada analisis residual akan dibuat plot residualnya untuk memeriksa apakah ada indikasi heteroskedastisitas; uji normalitas dengan menggunakan histogram, QQ-plot, uji Shapiro-Wilk, dan Jarque-Bera; dan uji autokorelasi residual dengan Ljung Box untuk memeriksa apakah residual saling *independent* atau tidak.
8. Melakukan *forecasting* (peramalan) 12 bulan kedepan. Setelahnya kita gunakan data *testing* untuk mengevaluasi akurasi model dengan melakukan kriteria ketepatan peramalan menggunakan perhitungan nilai *Mean Square Error* dan *Mean Absolute Percentage Error* (MAPE).
9. Melakukan perbandingan ketepatan peramalan dengan menggunakan metode *Simple Moving Average* (SMA).

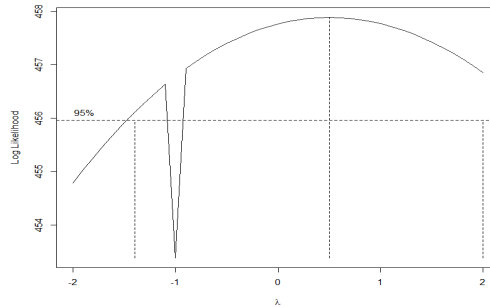
III. HASIL ANALISIS

1. Grafik runtun waktu



Gambar 1 Sebelum Stasioner (kiri) dan Setelah Stasioner (Kanan)

Grafik runtun waktu dibuat dengan menggunakan data *training* yakni data harga beras premium periode Januari 2014 – Desember 2022. Berdasarkan Gambar 1 bagian kiri merupakan grafik awal yang terlihat jelas tidak stasioner dalam varians dan mean. Dengan menggunakan transformasi *Box-Cox* untuk mengatasi kestasioneran dalam varians diperoleh λ sebesar 0.5. Maka $Y_t^\lambda = Y_t^{0.5} = \sqrt{Y_t}$ artinya data *training* akan diakarkan dengan fungsi `sqrt()`.



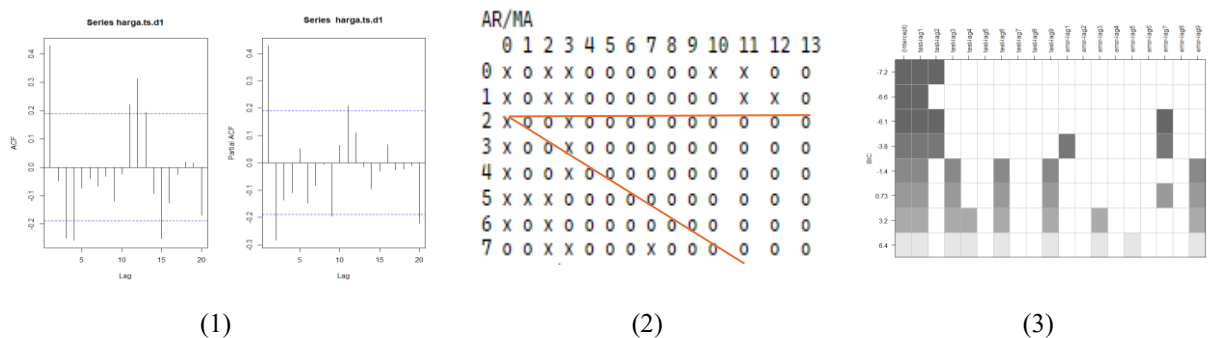
Gambar 2 Transformasi Box-Cox

Selanjutnya, akan diperiksa kestasioneran dalam mean dengan uji KPSS, dengan hipotesisnya:

$$H_0 : \text{stasioner} \text{ vs } H_1 : \text{Non - Stasioner}$$

Jika $p - \text{value} < \alpha$ (0.05) maka H_0 ditolak. Pada uji KPSS pertama dengan data yang sudah ditransformasi, didapat nilai $p - \text{value} = 0.01 < \alpha$ artinya data signifikan dan H_0 ditolak maka data tidak stasioner dan perlu *differencing*. Kemudian, dilakukan *differencing* dengan $d = 1$, hasil uji KPSS dengan *differencing* menghasilkan nilai $p - \text{value} = 0.1 > \alpha$ artinya data tidak signifikan dan H_0 diterima maka data sudah stasioner dalam mean dan tidak perlu dilakukan *differencing* lagi. Grafik setelah dilakukan transformasi dan *differencing* dapat dilihat pada Gambar 1 bagian kanan.

2. Membuat plot identifikasi model



Gambar 3 (1) ACF dan PACF, (2) EACF, (3) BIC

Plot PACF terputus pada lag ke-2 maka merupakan model $AR(2)$. Pada plot EACF daerah segitiga yang dibentuk oleh angka “o” menunjukkan model $ARMA(2,1)$. Dan pada plot BIC dengan nilai terendah sebesar -7.2 menunjukkan model $AR(2)$. Karena sebelumnya dilakukan *differencing* sebanyak satu kali maka orde $d = 1$,

dengan begitu kandidat model $ARIMA(p, d, q)$ yang didapatkan dari ketiga plot diatas menjadi: $ARIMA(2,1,0)$ dan $ARIMA(2,1,1)$

3. Mengestimasi Model

Bentuk persamaan estimasi model untuk $MA(q)$:

$$Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q}$$

Bentuk persamaan estimasi model untuk $AR(p)$:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + e_t$$

Berdasarkan kandidat model yang telah didapat sebelumnya yaitu model $ARIMA(2,1,0)$ dan $ARIMA(2,1,1)$, diperoleh:

Model $ARIMA(2, 1, 0)$

```
> ## Estimasi parameter model dari trans.harga.ts sebagai proses ARIMA(2,1,0)
> harga.arima.210 <- stats::arima(trans.harga.ts,order=c(2,1,0))
> harga.arima.210
```

```
Call:
stats::arima(x = trans.harga.ts, order = c(2, 1, 0))
```

```
Coefficients:
          ar1          ar2
      0.5878    -0.2957
s.e.  0.0946    0.0943
```

```
sigma^2 estimated as 0.6643: log likelihood = -130.15, aic = 266.3
```

Model $ARIMA(2, 1, 1)$

```
> ## Estimasi parameter model dari trans.harga.ts sebagai proses ARIMA(2,1,1)
> harga.arima.211 <- stats::arima(trans.harga.ts,order=c(2,1,1))
> harga.arima.211
```

```
Call:
stats::arima(x = trans.harga.ts, order = c(2, 1, 1))
```

```
Coefficients:
          ar1          ar2          ma1
      1.0331    -0.4915    -0.5099
s.e.  0.2722    0.1164    0.3227
```

```
sigma^2 estimated as 0.6519: log likelihood = -129.18, aic = 266.36
```

Ringkasan dan perbandingan hasil estimasi parameter model ARIMA dalam satu tabel.

	$ARIMA(2, 1, 0)$			$ARIMA(2, 1, 1)$		
	Estimasi	Std. Error		Estimasi	Std. Error	
Mean	-	-		-	-	
ϕ_1	0.5878	0.0946	Signifikan	1.0331	0.2722	Signifikan
ϕ_2	-0.2957	0.0943	Signifikan	-0.4915	0.1164	Signifikan

θ_1	-0.5099	0.3227	Tidak Signifikan
AIC	266.3	266.36	
σ^2	0.6643	0.6519	
Log Likelihood	-130.15	-129.18	

Model $ARIMA(2,1,0)$ merupakan model yang terbaik karena memiliki nilai AIC (*Akaike's Information Criterion*) terendah. Parameter pada model tersebut sudah signifikan, maka tidak perlu ada parameter yang harus dibuang. Maka estimasi model $ARIMA(2,1,0)$ adalah:

$$\sqrt{Y_t} = 0.5878\sqrt{Y_{t-1}} - 0.2957\sqrt{Y_{t-2}} + e_t$$

Karena model hasil *differencing*, maka diperoleh:

$$\sqrt{Y_t} \sim ARI(2,1)$$

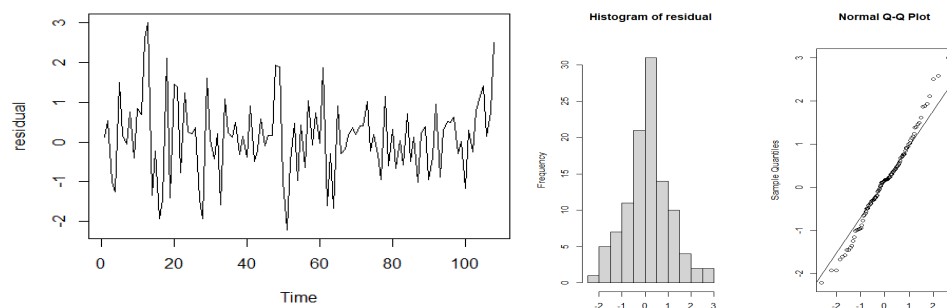
$$\Delta\sqrt{Y_t} \sim AR(2)$$

$$\Delta\sqrt{Y_t} = 0.5878\Delta\sqrt{Y_{t-1}} - 0.2957\Delta\sqrt{Y_{t-2}} + e_t$$

4. Melakukan diagnosis model

a) Analisis Residual

Untuk menganalisis residual dapat menggunakan plot residual, uji normalitas, dan uji autokorelasi residual. Plot residual dapat dilihat pada Gambar 4 dibawah. Plot menunjukkan residual konstan tanpa terlihat adanya trend tertentu. Pada gambar histogram grafik berbentuk seperti lonceng dan hanya memiliki satu puncak, serta gambar QQ Plot menunjukkan titik-titik tersebar disekitar garis maka dapat diindikasikan data berdistribusi normal.



Gambar 4 Plot Residual, Histogram, dan QQ-Plot

Pada uji Shapiro-Wilk dan uji Jarque-Bera hipotesisnya adalah:

$$H_0 : \text{Normal vs } H_1 : \text{Tidak Normal}$$

```
> shapiro.test(residual)

Shapiro-Wilk normality test

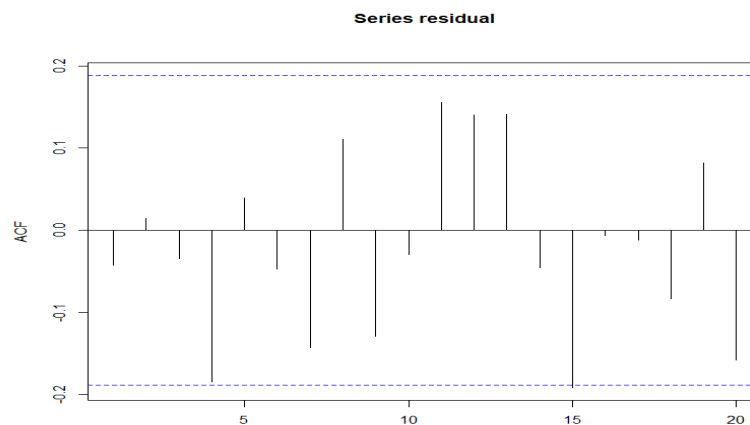
data:  residual
W = 0.98782, p-value = 0.4382

> jarque.bera.test(residual)

Jarque Bera Test

data:  residual
X-squared = 1.1899, df = 2, p-value = 0.5516
```

Jika $p - value < \alpha$ (0.05) maka H_0 ditolak, dan terlihat bahwa kedua uji tersebut menunjukkan nilai $p - value > \alpha$ maka H_0 diterima berarti uji tidak signifikan dan residual normal. Kemudian, dengan plot ACF kita dapat memeriksa autokorelasi residual, jika model fit maka residual akan mengikuti white noise, yaitu sampel residual akan saling *independent* atau autokorelasinya sama dengan nol.



Gambar 5 Series Residual

Uji autokorelasi residual juga dapat menggunakan uji Ljung-Box dengan hipotesis:

$$H_0 : \text{Independent vs } H_1 : \text{Tidak Independent}$$

```
> Box.test(residual, lag=15, type = c("Ljung-Box"))

Box-Ljung test

data:  residual
X-squared = 23.497, df = 15, p-value = 0.07415
```


Jika $p - value < \alpha$ (0.05) maka H_0 ditolak, dan terlihat bahwa uji tersebut menunjukkan nilai $p - value = 0.07415 > \alpha$ maka H_0 diterima berarti uji tidak signifikan dan residual *independent*.

b) Overparameterisasi Model

Overparameterisasi model adalah dengan membuat model yang lebih umum, yaitu model dengan satu order lebih tinggi. Jika model $ARIMA(2,1,0)$ fit, maka akan dimodelkan data sebagai proses $ARIMA(3,1,0)$. Model $ARIMA(3,1,0)$ akan terkonfirmasi jika estimasi parameter ϕ_3 tidak signifikan berbeda dari nol dan estimasi parameter ϕ_1 dan ϕ_2 tidak berubah secara drastis.

```
> harga.arima.210

Call:
stats::arima(x = trans.harga.ts, order = c(2, 1, 0))

Coefficients:
      ar1      ar2
  0.5878 -0.2957
s.e.  0.0946  0.0943

sigma^2 estimated as 0.6643:  log likelihood = -130.15,  aic = 266.3
> harga.arima.310 <- stats::arima(trans.harga.ts, order=c(3,1,0))
> harga.arima.310

Call:
stats::arima(x = trans.harga.ts, order = c(3, 1, 0))

Coefficients:
      ar1      ar2      ar3
  0.5521 -0.2278 -0.1172
s.e.  0.0986  0.1099  0.0992

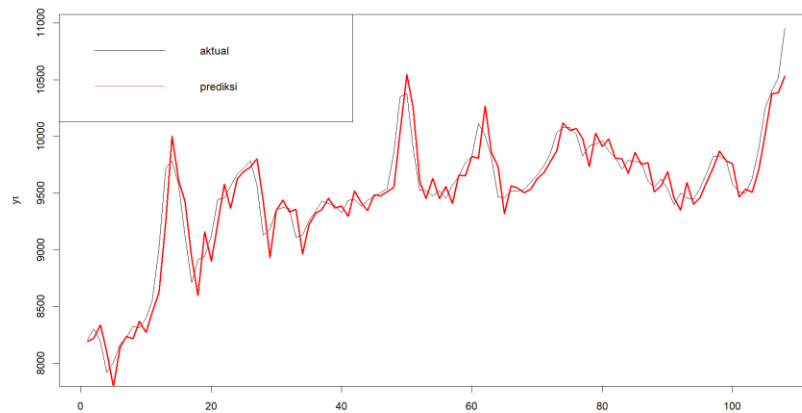
sigma^2 estimated as 0.6555:  log likelihood = -129.46,  aic = 266.91
```

Dapat terlihat bahwa estimasi parameter ϕ_3 tidak signifikan berbeda dari nol dan estimasi parameter ϕ_1 dan ϕ_2 tidak berubah secara drastis. Dengan demikian, model $ARIMA(2,1,0)$ fit.

5. Melakukan *forecasting* dan mengevaluasi akurasi model

a) Peramalan (*Forecasting*)

Langkah terakhir dalam model ARIMA adalah melakukan peramalan dengan menggunakan model terbaik yang sudah didapatkan dalam tahap sebelumnya. Model ARIMA yang dipilih untuk peramalan adalah model $ARIMA(2,1,0)$. Sebelum melakukan *forecasting*, kita perlu melihat terlebih dahulu apakah hasil prediksi dari model $ARIMA(2,1,0)$ mendekati data aktual.



Gambar 6 Perbandingan Data Training dan Hasil Prediksi Model $ARIMA(2,1,0)$

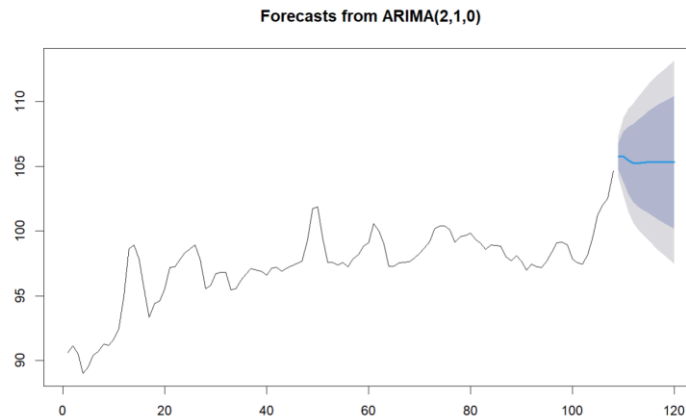
Pada Gambar 6 diatas menunjukkan hasil peramalan dari model $ARIMA(2,1,0)$ baik harga aktual maupun prediksi menunjukkan fluktuasi yang serupa dengan beberapa puncak dan lembah, mencerminkan bahwa prediksi cukup akurat dalam menangkap pola harga.

Selanjutnya akan dilakukan peramalan harga beras kualitas premium di tingkat penggilingan untuk 12 bulan kedepan. Dengan demikian, peramalan selama 12 bulan kedepan maka untuk $n = 109 - 120$.

```
> harga.forecast <- forecast(harga.arima.210, 12)
> harga.forecast
> plot(harga.forecast)
> prediksi <- harga.forecast$mean
> prediksi2 <- prediksi^2 # transformasi hasil prediksi ke dalam skala data asli
> prediksi2
```

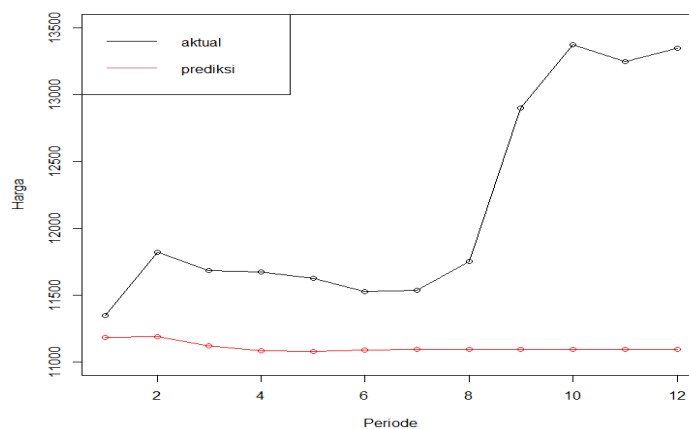
Hasil Peramalan $ARIMA(2, 1, 0)$			
Januari	11184.41	Juli	11092.94
Februari	11187.15	Agustus	11094.08
Maret	11120.37	September	11092.75
April	11080.40	Oktober	11091.63
Mei	11076.61	November	11091.37
Juni	11086.19	Desember	11091.54

Tabel diatas merupakan hasil peramalan harga beras kualitas premium di tingkat penggilingan 12 bulan kedepan.



Gambar 7 Forecasts

Selanjutnya akan disajikan grafik perbandingan aktual yang merupakan data *testing* dengan hasil peramalan yang telah diperoleh pada tabel diatas adalah sebagai berikut:



Gambar 8 Perbandingan Data Testing dan Hasil Peramalan

Dari peramalan yang telah dilakukan, dapat terlihat bahwa hasil peramalan harga secara konsisten berada di bawah harga aktual terutama pada periode bulan ke-8 hingga 12, menunjukkan bahwa hasil peramalan tersebut *underestimated* atau memperkirakan harga lebih rendah dari kenyataan. Dengan demikian, model menunjukkan performa yang baik pada data *training* tetapi kurang akurat pada data *testing*.

b) Pengujian Akurasi Hasil Peramalan

Pengujian akurasi hasil peramalan dilakukan dengan menghitung nilai MSE (*Mean Square Error*) dan MAPE (*Mean Absolute Percentage Error*).

MSE digunakan untuk mencari rata-rata perbedaan yang dikuadratkan antara nilai yang diprediksi dengan nilai yang diamati. Sedangkan, MAPE digunakan untuk menghitung rata-rata perbedaan absolut antara nilai perkiraan dan nilai aktual, dinyatakan dalam persen [3]. Perhitungan MSE dan MAPE dapat menggunakan formula berikut:

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2 \qquad MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right|$$

Y_t = nilai aktual pada periode t

\hat{Y}_t = nilai *forecast* pada periode t

Akurasi nilai MAPE kurang dari 20% dianggap hasil peramalan baik, nilai MAPE dengan rentang 20% - 50% dianggap hasil peramalan cukup baik, dan nilai MAPE lebih dari 50% dianggap hasil peramalan buruk.

```
> ## Bandingkan data testing dengan hasil prediksi
> ## Cara 1: hitung MSE (Mean Square Error)
> mean((harga.testing - prediksi2)^2) #MSE
[1] 1702535
>
> ## Cara 2: hitung MAPE (Mean Absolute Percentage Error)
> mean(abs(harga.testing - prediksi2)/harga.testing) #MAPE
[1] 0.08238411
```

Dari perbandingan data *testing* dengan hasil peramalan yang telah dilakukan, didapat nilai MSE sebesar 1702535 yang artinya menunjukkan adanya variasi besar antara harga aktual dan prediksi. Sedangkan, nilai MAPE sebesar $0.08238411 = 8.24\%$ yang menunjukkan bahwa hasil peramalan dianggap baik dan relatif akurat dalam memprediksi harga, dengan tingkat kesalahan rata-rata sekitar 8.24%.

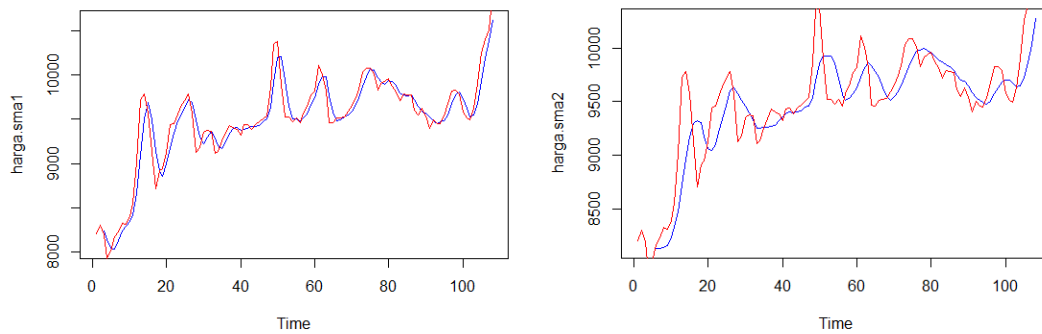
6. Membandingkan dengan Metode *Simple Moving Average* (SMA)

Simple Moving Average (SMA) adalah aritmatika *Moving Average* yang dihitung dengan menambahkan harga penutupan untuk sejumlah periode waktu dan kemudian dibagi sejumlah periode waktu tersebut.

Dalam SMA, data yang dimasukkan adalah sama bobot. Ini berarti bahwa setiap hari dalam kumpulan data memiliki tingkat kepentingan yang sama dan berbobot sama. Karena setiap hari baru berakhir, titik data yang tertua dibuang dan yang terbaru ditambahkan. SMA dengan n -period merupakan perhitungan rata-rata data dari sejumlah n periode ke belakang. Berikut adalah formula metode SMA:

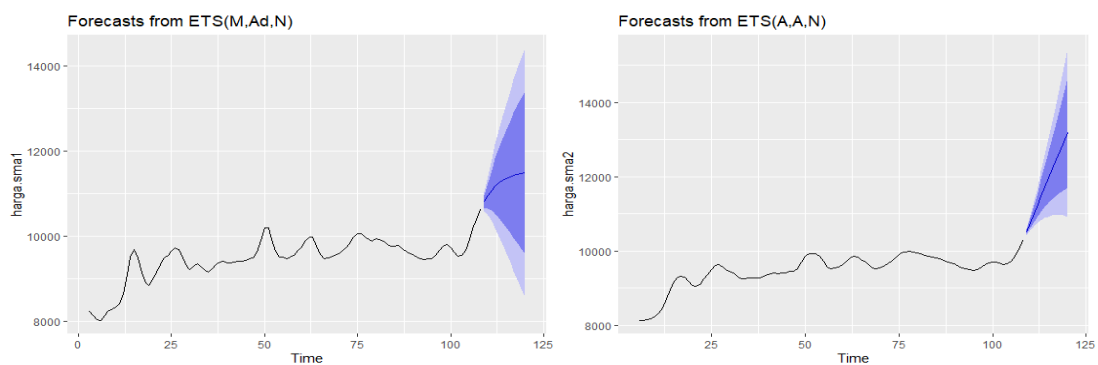
$$SMA_t(n) = \frac{P_t + \dots + P_{t-n+1}}{n}$$

Fungsi SMA pada RStudio tersedia di *package* ‘TTR’. Sama seperti metode ARIMA kita perlu terlebih dahulu membagi data sebagai data *training* dan data *testing*. Pada penelitian ini akan digunakan $n = 4$ dan $n = 6$ untuk selanjutnya dibandingkan nilai akurasi peramalannya.



Gambar 9 Perbandingan Data Training dan Hasil Prediksi dengan $n = 4$ (kiri) dan $n = 6$ (kanan)

Berdasarkan Gambar 9 diatas, dapat terlihat bahwa data *training* (garis merah) dengan hasil prediksi SMA (garis biru) memiliki keakuratan yang tinggi. Kedua garis menunjukkan tren naik secara umum dengan beberapa fluktuasi. Ini menunjukkan bahwa harga yang dianalisis mengalami peningkatan nilai dengan beberapa periode volatilitas. Namun, model SMA dengan 4 periode lebih akurat dalam menangkap pola harga.



Gambar 10 Forecast from ETSA(A,A,N) untuk $n = 4$ (kiri) dan $n = 6$ (kanan)

Pada Gambar 10, menunjukkan peramalan harga beras untuk data yang dihaluskan dengan SMA dengan $n = 4$ dan $n = 6$. Garis hitam menunjukkan data historis, garis biru adalah ramalan titik, dan area berbayang menunjukkan interval

prediksi 80% dan 95%. Hasilnya menunjukkan tren peningkatan harga dengan ketidakpastian yang meningkat seiring waktu. Berikut adalah hasil peramalannya:

Hasil Peramalan SMA $n = 4$				Hasil Peramalan SMA $n = 6$			
Januari	10792.75	Juli	12369.28	Januari	10517.82	Juli	11974.79
Februari	11055.51	Agustus	12632.03	Februari	10760.65	Agustus	12217.62
Maret	11318.26	September	12894.79	Maret	11003.48	September	12460.45
April	11581.02	Oktober	13157.54	April	11246.31	Oktober	12703.28
Mei	11843.77	November	13420.30	Mei	11489.14	November	12946.11
Juni	12106.53	Desember	13683.05	Juni	11731.96	Desember	13188.94

Dengan $n = 4$ menghasilkan nilai akurasi MSE sebesar 255301.9 dan nilai MAPE sebesar $0.03522063 = 3.52\%$ menunjukkan hasil peramalan memiliki keakuratan yang baik. Sedangkan, dengan $n = 6$ menghasilkan nilai akurasi MSE sebesar 305678.4 dan nilai MAPE sebesar $0.0402283 = 4.02\%$ menunjukkan hasil peramalan yang dihasilkan juga memiliki keakuratan yang baik.

IV. KESIMPULAN DAN SARAN

1. Kesimpulan

Diperoleh beberapa kesimpulan sebagai berikut:

- 1) SMA dengan $n = 4$ sebagai periode terbaik menghasilkan nilai MSE yang lebih rendah sebesar 255301.9, dibandingkan dengan nilai MSE pada metode ARIMA dengan model terbaik adalah $ARIMA(2,1,0)$ sebesar 1702535.
- 2) SMA dengan $n = 4$ memiliki tingkat kesalahan peramalan yang lebih kecil sebesar 3.52% dibandingkan dengan $ARIMA(2,1,0)$ sebesar 8.24%.

Dengan demikian metode SMA dengan 4 periode lebih unggul dalam hal akurasi dan memiliki tingkat kesalahan yang lebih rendah dibandingkan dengan model $ARIMA(2,1,0)$. Oleh karena itu, SMA dengan 4 periode dapat dianggap sebagai model peramalan yang lebih baik untuk harga beras yang diteliti.

2. Saran

Berdasarkan hasil dan kesimpulan penelitian ini, diharapkan untuk penelitian lebih lanjut dapat mengkaji penggunaan metode lain yang mungkin memberikan hasil peramalan yang lebih akurat dan baik dibandingkan metode ARIMA dan SMA sehingga dapat

memberikan manfaat yang lebih besar dan berkelanjutan bagi ilmu pengetahuan dan sektor terkait.

REFERENSI

- [1] F. Ahmad, “PENENTUAN METODE PERAMALAN PADA PRODUKSI PART NEW GRANADA BOWL ST Di PT.X,” *JISI J. Integr. Sist. Ind.*, vol. 7, no. 1, p. 31, 2020, doi: 10.24853/jisi.7.1.31-39.
- [2] V. Sari and S. A. Hariyanto, “Peramalan Harga Beras Premium Bulanan Di Tingkat Penggilingan Menggunakan Fuzzy Time Series Markov Chain,” *J. Gaussian*, vol. 12, no. 3, pp. 322–329, 2023, doi: 10.14710/j.gauss.12.3.322-329.
- [3] E. Paga and A. Nugroho, “Perbandingan Metode Exponential Smoothing dan ARIMA untuk Prediksi Jumlah Mahasiswa Baru (Studi Kasus di FTI UKSW),” *Progresif J. Ilm. Komput.*, vol. 20, no. 1, p. 393, 2024, doi: 10.35889/progresif.v20i1.1829.

APPENDIX

1) Skrip R – Metode ARIMA

```
## Load package 'TSA' dan 'forecast'
## Install.packages("TSA")
## Package untuk import data 'readxl'
library(TSA)
library(forecast)
library(tseries)
library(readxl)

## Import data yang akan digunakan
data_harga <- read_excel("Projek ARW/harga.ts.xlsx")
View(data_harga)

## Ringkasan data harga beras premium
head(data_harga)
dim(data_harga)
n <- length(data_harga$Harga);n

## Bagi data menjadi dua bagian: data training dan testing (70-90%, 30-10%)
## Data training untuk membangun model ARIMA
## Data testing untuk validasi model (contoh: 1 tahun data terakhir digunakan sebagai testing: ~
4% data testing)
harga.training <- data_harga$Harga[1:(n-12)]
harga.testing <- data_harga$Harga[(n-11):n]

plot(harga.training)
## Gunakan perintah ts() untuk merubah harga bulanan menjadi data runtun waktu
```

```

harga.ts <- ts(harga.training)

## Buat plot runtun waktu dari data harga
plot(harga.ts, ylab = "Harga", xlab = "Periode")

## Cek perlu tidaknya melakukan transformasi data untuk mengatasi non-stasioner dalam variansi
bc <- BoxCox.ar(harga.ts)      # transformasi akar, lambda = 0.5
bc
trans.harga.ts <- sqrt(harga.ts)
plot(trans.harga.ts)

## Cek apakah data sudah stasioner dalam mean (uji adf atau uji kpss)
## Uji trend stokastik dengan uji KPSS (H0: stasioner vs H1: non-stasioner)
kpss.test(trans.harga.ts, null="Level")
#p-value (0.01) < 0.05, signifikan, Ho ditolak, artinya data tidak stasioner dan perlu differencing

## Plot diferensi ke-1 dari trans.harga.ts
harga.ts.d1 <- diff(trans.harga.ts, d = 1)
plot(harga.ts.d1)

## Uji stasioneritas dari diferensiasi ke-1 harga.ts.d1: (H0: stasioner vs H1: non-stasioner)
kpss.test(harga.ts.d1)
#p-value (0.1) > 0.05, tidak signifikan, Ho diterima, tidak ada indikasi non-stasioner dalam mean

##### TAHAP 1: IDENTIFIKASI MODEL

## Identifikasi Model dengan ACF dan PACF dari harga.ts.d1
par(mfrow=c(1,2))
acf(harga.ts.d1)
pacf(harga.ts.d1)
#kandidat model: AR(2) karena terputus pada lag ke-2, dan menjadi ARIMA(2,1,0)

## Identifikasi Model dengan EACF
eacf(harga.ts.d1)
#kandidat model: ARMA(2,1) menjadi ARIMA (2,1,1)

## Identifikasi Model dengan kriteria informasi BIC
par(mfrow=c(1,1))
res <- armasubsets(y=harga.ts.d1,nar=9,nma=9,y.name='test',ar.method='ols')
plot(res)
#kandidat model: AR(2) dilihat dari nilai BIC terendah, dan menjadi ARIMA(2,1,0)

### KESIMPULAN --> Didapat kandidat model: ARIMA(2,1,0) dan ARIMA(2,1,1)

##### TAHAP 2: ESTIMASI MODEL

## Estimasi parameter model dari trans.harga.ts sebagai proses ARIMA(2,1,0)
harga.arima.210 <- stats::arima(trans.harga.ts,order=c(2,1,0))
harga.arima.210

```



```

## Estimasi parameter model dari trans.harga.ts sebagai proses ARIMA(2,1,1)
harga.arima.211 <- stats::arima(trans.harga.ts,order=c(2,1,1))
harga.arima.211

## Gunakan perintah auto.arima() dari package 'forecast' - Secara otomatis untuk menentukan order
p, q, & d sekaligus memberikan estimasi parameter model
harga.auto <- auto.arima(trans.harga.ts)
harga.auto

##### TAHAP 3: DIAGNOSIS MODEL

residual <- rstandard(harga.arima.210)

## Plot residual
plot(residual)

## Cek normalitas dengan histogram dan QQ plot
par(mfrow=c(1,2))
hist(residual)
qqnorm(residual)
qqline(residual)

## Uji normalitas
shapiro.test(residual)
jarque.bera.test(residual)
#p-value kedua tes > 0.05, tidak signifikan, Ho diterima, residual normal

## Cek autokorelasi: individu
par(mfrow=c(1,1))
acf(residual)

## Uji autokorelasi dengan ljung.box: serentak
Box.test(residual,lag=15,type = c("Ljung-Box"))
#p-value > 0.05, tidak signifikan, Ho diterima, residual independent

## Diagnosis dengan overfitting
harga.arima.210
harga.arima.310 <- stats::arima(trans.harga.ts,order=c(3,1,0))
harga.arima.310

#----Apakah residual menunjukkan white noise?

##### TAHAP 4: FORECASTING (PRAKIRAAN) / MENGEVALUASI AKURASI MODEL

## Plot data sebenarnya (yt) dan hasil prediksi model ARIMA(2,1,0)
actual <- harga.ts

## Perintah fitted() digunakan untuk mengeluarkan hasil prediksi yt
fitted(harga.arima.210)
pred <- fitted(harga.arima.210)^2# transformasi balik ke data aktual

```

```

pred
par(mfrow=c(1,1))
plot(actual, ylab = "yt")
lines(pred, col = "red", lwd = 2)
legend("topleft", c("aktual", "prediksi"), col=c(1,2), lty = 1)

## Gunakan perintah forecast() untuk memvalidasi model
## Dengan cara memprediksi harga beras 12 bulan kedepan
harga.forecast <- forecast(harga.arima.210, 12)
harga.forecast
plot(harga.forecast)
prediksi <- harga.forecast$mean # simpan hasil prediksi 12 bulan kedepan ke objek prediksi
(prediksi dalam bentuk akar)
prediksi2 <- prediksi^2 # transformasi hasil prediksi ke dalam skala data asli
prediksi2

## Data testing
harga.testing <- data_harga$Harga[(n-11):n]

## Plot data sebenarnya (yt) dengan hasil peramalan
plot(harga.testing,type='o', ylab = "Harga", xlab = "Periode", ylim=c(11000,13500))
lines(ts(prediksi2),col='red',type='o')
legend("topleft", c("aktual", "prediksi"), col=c(1,2), lty = 1)

## Bandingkan data testing dengan hasil prediksi
## Cara 1: hitung MSE (Mean Square Error)
mean((harga.testing - prediksi2)^2) #MSE

## Cara 2: hitung MAPE (Mean Absolute Percentage Error)
mean(abs(harga.testing - prediksi2)/harga.testing) #MAPE
#MAPE = 8.24% (model prediksi yang baik < 20%)

```

2) Skrip R – Metode SMA

```

## Import data yang akan digunakan
library(TTR)
library(readxl)
library(forecast)
library(Metrics)
data_harga <- read_excel("Projek ARW/harga.ts.xlsx")
View(data_harga)

## Bagi data menjadi dua bagian: data training dan testing
n <- length(data_harga$Harga);n
harga.training <- data_harga$Harga[1:(n-12)]
harga.ts <- ts(harga.training)
harga.testing <- data_harga$Harga[(n-11):n]

## Untuk n = 4
harga.sma1 <- SMA(harga.ts, n = 4)

```

```

harga.sma1
plot(harga.sma1)
lines(harga.sma1, col = "blue")
lines(harga.ts, col = "red")

prediksi1 <- forecast(harga.sma1, h = 12)
prediksi1
autoplot(prediksi1)
pred1 <- prediksi1$mean
pred1

#Menghitung MSE dan MAPE
mean((harga.testing - pred1)^2) #MSE
mean(abs(harga.testing - pred1)/harga.testing) #MAPE

## Untuk n = 6
harga.sma2 <- SMA(harga.ts, n = 6)
harga.sma2
plot(harga.sma2)
lines(harga.sma2, col = "blue")
lines(harga.ts, col = "red")

prediksi2 <- forecast(harga.sma2, h = 12)
prediksi2
autoplot(prediksi2)
pred2 <- prediksi2$mean
pred2

#Menghitung MSE dan MAPE
mean((harga.testing - pred2)^2) #MSE
mean(abs(harga.testing - pred2)/harga.testing) #MAPE

```