

# PERAMALAN HARGA BERAS DI TINGKAT PERDAGANGAN BESAR (GROSIR) DI INDONESIA MENGGUNAKAN METODE *SINGULAR SPECTRUM ANALYSIS* (SSA)

Sri Soerya Ningsih<sup>(1)</sup>, Rochdi Wasono<sup>(2)</sup> dan Indah Manfaati Nur<sup>(3)</sup>  
<sup>(1,2,3)</sup> Program Studi Statistika Universitas Muhammadiyah Semarang

Article history	Abstract
Submission :	<i>Rice is a staple food for the majority of the population in Indonesia.</i>
Revised :	<i>Stabilization of prices and supply is an important element in achieving</i>
Accepted :	<i>food security as one of the priorities of national development.</i>
<b>Keyword:</b>	<i>Therefore, it is necessary to forecast the price of rice so that it can</i>
Peramalan, MAPE, Harga	<i>help stabilize the price of staple food, because if the price of goods is</i>
Beras, SSA	<i>unstable it can cause inflation or deflation. In this research, rice price</i>
	<i>forecasting use the Singular Spectrum Analysis (SSA) method. The</i>
	<i>data used is the monthly rice price from January 2010 to July 2020.</i>
	<i>The accuracy of forecasting uses the Mean Absolute Percentage Error</i>
	<i>(MAPE). Forecasting results show that wholesale rice prices in</i>
	<i>Indonesia tend to increase every month and get an MAPE value of</i>
	<i>1.01%.</i>

## Pendahuluan

Beras merupakan bahan makanan pokok bagi mayoritas penduduk di Indonesia. Negara Indonesia merupakan produsen beras yang cukup besar di dunia, hal ini diungkapkan oleh Food and Agriculture Organization of the United Nations (FAO) dalam memonitoring pasar beras di dunia. Stabilisasi pasokan dan harga beras menjadi salah satu unsur penting dalam pencapaian ketahanan pangan sebagai salah satu prioritas pembangunan nasional[1]. Harga beras menjadi salah satu tolak ukur kestabilan perekonomian nasional maupun regional. Kebutuhan konsumen yang cenderung meningkat menyebabkan harga beras cenderung tidak stabil karena adanya keterkaitan antara barang akan membuat perubahan pada harga barang yang lainnya[2]. Menurut Badan Pusat Statistik (BPS) hasil panen disetiap tahunnya mengalami surplus atau lebih dari kebutuhan pangan harga beras tahunan di Indonesia. Namun di Indonesia masih tetap mengimpor beras tiap tahun, dengan tujuan untuk memastikan bahwa ketersediaan pangan selalu tercukupi. Impor beras nantinya akan mempengaruhi harga beras yang dihasilkan oleh petani lokal, sebab petani lokal masih menggunakan tenaga konvensional bahkan lahan yang digunakan bukan lahan milik sendiri melainkan sewa lahan. Imbasnya

akan mempengaruhi perekonomian dalam negeri. Data harga beras merupakan data deret waktu. Dimana dalam data deret waktu dapat dilakukan sebuah peramalan. Dalam suatu peramalan data deret waktu dengan melakukan identifikasi pola-pola data pada masa lalu guna membuat perkiraan pada masa yang akan datang. Secara umum terdapat tiga pola data deret waktu yaitu horizontal, *trend*, dan musiman.

Ada beberapa metode yang telah diusulkan sebelumnya tentang peramalan wisatawan, diantaranya peramalan produksi industri eropa menggunakan SSA, ARIMA, dan Holt-Winters[3], peramalan jumlah kunjungan wisatawan ke Indonesia melalui pintu masuk menggunakan metode singular spectrum analysis dan SARIMA[4], dan masih banyak yang lainnya. Beberapa metode peramalan mempunyai asumsi yang harus terpenuhi, terkadang jika asumsi tersebut tidak terpenuhi maka akan sulit untuk menggunakan metode tersebut. *Singular Spectrum Analysis* (SSA) merupakan suatu metode analisis deret waktu yang menggabungkan elemen-elemen dari analisis deret waktu klasik, statistik multivariat, geometri multivariat, sistem dinamis, dan pemrosesan[6]. Metode ini dapat digunakan pada beberapa deret waktu karena merupakan metode non-parametrik dimana asumsi pada

analisis deret waktu klasik tidak mendesak lagi untuk dilakukan. Sebagai metode non-parametrik dan bebas model, *Singular Spectrum Analysis* dapat digunakan pada beberapa deret waktu karena tidak memerlukan asumsi stasioneritas dan tidak memerlukan transformasi logaritma. Peramalan deret waktu adalah sebuah area partikel yang sangat penting dan *Singular Spectrum Analysis* dapat menjadi sangat efektif untuk peramalan. Terdapat algoritma dalam melakukan peramalan menggunakan *Singular Spectrum Analysis* (SSA), yaitu algoritma *Recurrent Forecasting* [6].

## Landasan Teori

### *Singular Spectrum Analysis* (SSA)

*Singular Spectrum Analysis* (SSA) adalah teknik baru dari analisis deret waktu. Metode ini tidak membuat asumsi statistik mengenai sinyal atau *noise* saat melakukan analisis dan menyelidiki sifat-sifat algoritma [8]. Dasar dari *Singular Spectrum Analysis* (SSA) adalah dekomposisi dengan menguraikan data deret waktu menjadi komponen-komponennya yaitu tren, musiman, siklis, dan *noise*[4]. Tahapan dalam melakukan analisis menggunakan *Singular Spectrum Analysis* (SSA) terbagi menjadi dua yaitu dekomposisi dan rekonstruksi[8].

### Dekomposisi

Pada tahap dekomposisi, parameter yang digunakan adalah *Window Length* (L) [4]. Parameter ini berfungsi untuk menentukan banyaknya dimensi matriks lintasan. Nilai L ini merupakan dimensi dari matriks lintasan yang merupakan matriks dari perkalian *Hankel*. Penentuan dari nilai L dilakukan dengan proses pengecekan melalui *trial and error*. Dekomposisi mempunyai dua tahapan, yaitu *Embedding* dan *Singular Value Decomposition* (SVD).

### *Embedding*

Pada proses *Embedding* mengubah data deret waktu menjadi urutan Lag vektor dengan ukuran L dengan membentuk  $K=N-L+1$  lag vektor.

$$X_i = (x_i, \dots, x_{i+L-1})^T \quad (1 \leq i \leq K)$$

dengan ukuran L. Jika ditekan ukuran (dimensi) vektor  $X_i$ , maka akan disebut sebagai L-lag vektor. Matriks Lintasan dari  $X$  adalah :

$$X = [X_1; \dots; X_K] = (x_{ij})_{ij} = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_k \\ x_2 & x_3 & x_4 & \dots & x_{k+1} \\ x_3 & x_4 & x_5 & \dots & x_{k+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \dots & x_N \end{bmatrix}$$

Lag vektor  $X_i$  adalah kolom dari matriks lintasan  $X$ . Baris dan kolom dari  $X$  bagian dari data deret waktu. Elemen (i,j) dari matriks  $X$  adalah  $x_{ij} = x_{i+j-1}$  yang menghasilkan  $X$  bernilai sama dengan elemen "anti diagonals"  $i+j = \text{konstan}$ . (Oleh karena itu lintasan matriks yang terbentuk dinamakan matriks *Hankel*). Matriks yang terbentuk mendefinisikan kemiripan satu dengan yang lain antara matriks lintasan  $L \times K$  dan deret waktu.

### *Singular Value Decomposition* (SVD)

Pada tahap ini, akan menghasilkan *singular value decomposition* (SVD) dari matriks  $X$ . Didefinisikan  $S = XX^T$  dan ditunjukkan dengan  $\lambda_1, \dots, \lambda_L$  *eigenvalue* dari  $S$  sedangkan *singular value* dinotasikan dengan  $U_1, \dots, U_L$  merupakan *eigenvector* yang sesuai dengan *eigenvalues* pada matriks  $S$ .

Selanjutnya didefinisikan  $d = \text{rank } X = \max \{i, \text{dimana } \lambda_i > 0\}$  atau biasanya dengan  $d = L^* = \min \{K, L\}$ . Sedangkan *principal component* dinotasikan dengan  $V_i = X^T U_i / \sqrt{\lambda_i}$  ( $i = 1, \dots, d$ ). Sehingga didapatkan SVD dari matriks lintasan  $X$  sebagai berikut

$$X = X_1 + \dots + X_d$$

dimana  $X_i = \sqrt{\lambda_i} U_i V_i^T$ . Matriks  $X$  mempunyai rank 1, karena merupakan matriks elementer. Kumpulan dari  $(\sqrt{\lambda_i}, U_i, V_i^T)$  disebut *eigen triple* ke  $i$  dari SVD[5]. SVD dari matriks lintasan dapat ditulis dengan persamaan berikut :

$$X = X_1 + X_2 + \dots + X_d$$

$$X = \sqrt{\lambda_1} U_1 V_1^T + \sqrt{\lambda_2} U_2 V_2^T + \dots + \sqrt{\lambda_d} U_d V_d^T$$

$$X = \sum_{i=1}^d \sqrt{\lambda_i} U_i V_i^T$$

### Rekonstruksi

Parameter yang digunakan dalam tahap rekonstruksi adalah *grouping effect* ( $r$ ). Fungsi dari parameter ini adalah menentukan pola pada plot data. Sebelumnya pada tahap dekomposisi dengan penggunaan parameter  $L$ ,

dan menyajikan serangkaian seri awal yang telah dipisahkan dengan baik pada SVD maka *eigen triples* yang terbentuk akan membantu dalam penentuan parameter *grouping effect*. Hasil dari tahap rekonstruksi akan mendekati hasil peramalan dengan data asli. Oleh karena itu pengelompokan yang tepat dilakukan akan mendukung hasil peramalan dengan baik dengan menunjukkan nilai MAPE dari nilai ramal dengan data asli.

### Grouping

Pada persamaan yang telah terbentuk, matriks  $\mathbf{X}_i$  akan dipartisi ke  $m$  *disjoint subset*  $I_1, \dots, I_m$ . Jika  $I = \{i_1, \dots, i_p\}$  maka yang dihasilkan dari matriks  $\mathbf{X}_i$  sesuai dengan kelompok  $I$  yang didefinisikan sebagai  $\mathbf{X}_I = \mathbf{X}_{i_1} + \dots + \mathbf{X}_{i_p}$ . Matriks yang dihasilkan dihitung untuk pengelompokan  $I = I_1, \dots, I_m$  maka persamaan yang terbentuk adalah [6]

$$\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$$

Prosedur pemilihan set  $I_1, \dots, I_m$  dinamakan *eigen triple grouping*. Jika  $m=d$  dan  $I_j = \{j\}$ ,  $j = 1, \dots, d$  maka pengelompokan yang terbentuk disebut sebagai *elementary*.

### Diagonal Averaging

Pada tahap ini mengubah setiap matriks  $\mathbf{X}_{ij}$  dari persamaan menjadi data deret waktu baru dengan panjang  $N$  yang dimisalkan dengan  $\mathbf{Y}$  berikut :

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_k \\ y_{21} & y_{22} & \dots & y_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & \dots & y_N \end{bmatrix}$$

Jika  $\mathbf{Y}$  adalah matriks  $L \times K$  dengan elemen  $y_{ij}$ ,  $1 \leq i \leq L$ ,  $1 \leq j \leq K$ . Definisikan  $L^* = \min(L, K)$ ,  $K^* = \max(L, K)$  dan  $N = L + K - 1$ . Maka  $y_{ij}^* = y_{ij}$  jika  $L < K$  dan  $y_{ij}^* = y_{ji}$  untuk yang lain. Dengan membuat *diagonal averaging* mengubah matriks  $\mathbf{Y}$  menjadi deret waktu  $y_1, \dots, y_N$  dengan rumus [6]

$$\mathbf{Y}_k = \begin{cases} \frac{1}{k} \sum_{m=1}^k y_{m, k-m+1}^* & \text{untuk } 1 \leq k < L^* \\ \frac{1}{L^*} \sum_{m=1}^{L^*} y_{m, k-m+1}^* & \text{untuk } L^* \leq k \leq K^* \\ \frac{1}{N-k+1} \sum_{m=k-K^*+1}^{N-k+1} y_{m, k-m+1}^* & \text{untuk } K^* < k \leq N \end{cases}$$

Berdasarkan persamaan misal pada matriks  $\mathbf{Y}$ , dipilih  $k=1$  maka  $y_1 = y_{11}$ , untuk  $k=2$  maka  $y_2 = \left(\frac{y_{12} + y_{21}}{2}\right)$  untuk  $k=3$  maka  $y_3 = \left(\frac{y_{13} + y_{22} + y_{31}}{3}\right)$ , dan begitu seterusnya. Untuk catatan jika matriks  $\mathbf{Y}$  adalah matriks lintasan dari beberapa deret  $(z_1, \dots, z_N)$ , maka  $y_i = z_i$  untuk semua  $i$ .

Persamaan diterapkan pada hasil matriks  $\mathbf{X}_{Ik}$  membuat deret baru  $\widetilde{X}^{(k)} = (\widetilde{x}_1^{(k)}, \dots, \widetilde{x}_N^{(k)})$ . Oleh karena itu deret  $x_1, \dots, x_N$  adalah didekomposisi menjadi jumlah  $m$  deret yaitu :

$$x_n = \sum_{k=1}^m \widetilde{x}_n^{(k)} \quad (n = 1, 2, \dots, N)$$

### Peramalan Singular Spectrum Analysis (SSA)

Prinsip-prinsip peramalan dalam *Singular Spectrum Analysis (SSA)* memiliki sifat penting dari dekomposisi SSA jika seri asli  $f_n$  memenuhi rumus *linear recurrent formula (LRF)*

$$f_n = a_1 f_{n-1} + \dots + a_d f_{n-d}$$

dimana  $d$  ( $d \leq L$ ) adalah angka dari *nonzero singular values* dari matriks  $\mathbf{X}$ . Dari beberapa dimensi  $d$  dengan beberapa koefisien  $a_1, \dots, a_d$  kemudian untuk setiap  $N$  dan  $L$  terdapat banyak nilai singular  $d$  *nonzero* dalam SVD dari matriks lintasan  $\mathbf{X}$ . Oleh karena itu jika *window length*  $L$  dan  $K = N - L + 1$  lebih besar dari  $d$ , maka hanya dibutuhkan lebih banyak  $d$  pada matriks  $\mathbf{X}_i$  untuk merekonstruksi data [8].

Asumsikan tujuan yang ingin dicapai dengan menggunakan SSA adalah suatu komponen aditif tertentu  $F_N^{(1)}$  dapat diekstrak dari suatu deret  $F_N$ . Dalam algoritma ini, suatu *window length*  $L$  yang sesuai, SVD matriks lintasan diperoleh dari deret  $F_N$  dan *eigen triples*  $(\sqrt{\lambda}, U, V)$  dipilih yang sesuai dengan  $F_N^{(1)}$ . Pada langkah *diagonal averaging*, deret yang direkonstruksi  $F_N^{(1)}$  yang mengestimasi  $F_N^{(1)}$  akan diperoleh [4].

### Recurrent Forecasting

Algoritma *Recurrent Forecasting* dapat dirumuskan sebagai berikut:

1. Deret waktu  $\mathbf{Y}_{N+M} = (y_1, \dots, y_{N+M})$  didefinisikan dengan

$$y_i = \begin{cases} \bar{x}_i & \text{for } i=1, \dots, N \\ \sum_{j=1}^{L-1} a_j y_{i-j} & \text{for } i=N+1, \dots, N+M \end{cases}$$

2. Angka-angka  $\mathbf{Y}_{N+1}, \dots, \mathbf{y}_{N+M}$  membentuk istilah  $M$  dari *Recurrent Forecasting*.

Jadi, peramalan *Recurrent Forecasting* dilakukan dengan penggunaan langsung LRR dengan koefisien  $\{a_j, j = 1, \dots, L-1\}$ .

Definisikan operator linear  $\mathcal{P}_{\text{Rec}} : \mathbb{R}^L \mapsto \mathbb{R}^L$  dengan rumus

$$\mathcal{P}_{\text{Rec}} \mathbf{Y} = \begin{pmatrix} \bar{Y} \\ \mathbf{R}^T \bar{Y} \end{pmatrix}$$

dengan,

$$Z_i = \begin{cases} \hat{X}_i & \text{for } i = 1, \dots, K \\ \mathcal{P}_{\text{Rec}} Z_{i-1} & \text{for } i = K + 1, \dots, K + M \end{cases}$$

### Ketepatan Hasil Model Peramalan

Dalam melakukan suatu peramalan yang merupakan kegiatan memprediksi masa depan dengan menggunakan data di masa lampau, hasil yang akan didapatkan tidaklah sama dengan data sesungguhnya [8]. Maka dari itu usaha untuk membuat nilai *error* seminimal mungkin dibutuhkan pada proses peramalan.

Salah satu ketepatan hasil peramalan dapat diukur dari nilai *Mean Absolute Percentage Error* (MAPE) yaitu rata-rata persentase kesalahan pertama dari beberapa periode. Tingkat keakuratan dapat dijelaskan dengan membandingkan nilai yang diproyeksikan dengan nilai aktual. Untuk melakukan peramalan dan untuk mengetahui akuratnya sebuah model maka nilai akurasi harus semakin kecil.

$$\text{MAPE} = \frac{\sum_{t=1}^n \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right|}{n} \times 100\%$$

### Metode Penelitian

#### Sumber Data

Data yang digunakan pada penelitian ini yaitu data sekunder yang bersumber dari bps.go.id berupa data rata-rata harga beras di tingkat perdagangan besar (grosir). Data harga beras yang digunakan bulan Januari 2010 sampai dengan Juli 2020 dengan jumlah data sebanyak 127. Data periode Januari 2010

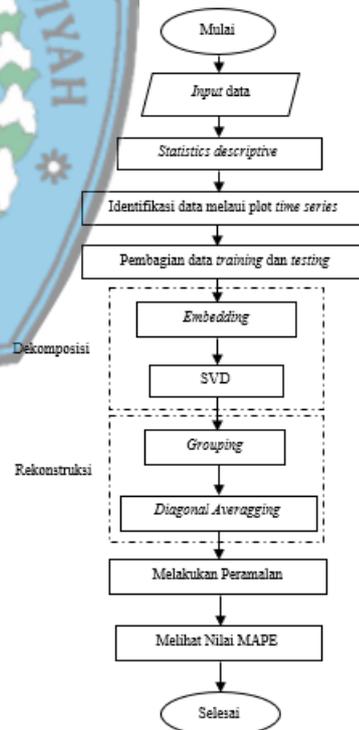
sampai dengan Juli 2019 digunakan sebagai data pengamatan untuk model *training* dengan jumlah data sebanyak 115 data yang berfungsi untuk membentuk suatu model, sedangkan periode Agustus 2019 sampai dengan Juli 2020 digunakan sebagai data untuk memvalidasi model yang terpilih dari data *testing* dengan jumlah data sebanyak 12 data.

### Struktur Data

Tabel 1 Struktur data

n	Bulan	Tahun	$Y_{1,n}$
1	1	2010	$Y_{1,1}$
2	2	2010	$Y_{1,2}$
6	6	2010	$Y_{1,6}$
:	:	:	:
127	7	2020	$Y_{1,127}$

### Langkah Penelitian



### Hasil Penelitian dan Pembahasan

#### Analisis Deskriptif

Gambaran umum mengenai harga beras di Indonesia dijelaskan menggunakan statistik deskriptif sebagai berikut :

Tabel 2 Statistik Deskriptif Harga Beras di Indonesia periode Januari 2010 – Juli 2020.

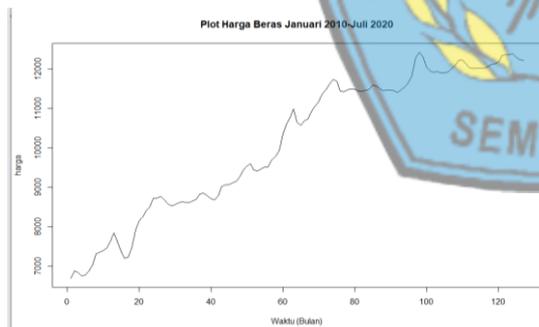
Rata-rata	Standar Deviasi	Minimum	Maksimum
10156	1788.124	6702	12414

Pada tabel diatas dapat dilihat bahwa rata-rata harga beras di Indonesia sebesar Rp 10156 dengan standar deviasi sebesar 1788.124. harga beras terendah pada bulan Januari 2010 sebesar Rp 6702 dan harga beras tertinggi pada bulan Februari 2018 sebesar Rp 12414.

Penelitian ini menggunakan data dari Badan Pusat Statistik (BPS) periode Januari 2010 hingga Juli 2020. Data tersebut merupakan data deret waktu yang dibagi menjadi dua bagian:

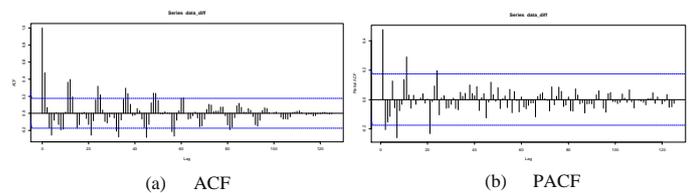
- Data *Training* : Januari 2010 – Juli 2019
- Data *Testing* : Agustus 2019 – Juli 2020

Peramalan harga beras di Indonesia dilakukan dengan menggunakan metode *Singular Spectrum Analysis* (SSA). Sebelum melakukan analisis data, terlebih dahulu dilakukakan *plotting* data harga beras di Indonesia sehingga terlihat pergerakan harga beras selama periode tersebut. Gambar dapat dilihat pada gambar 1.



Gambar 1. Plot Harga Beras

Pada gambar diatas, terlihat bahwa harga beras di Indonesia cenderung terjadi kenaikan. Hal ini menunjukkan bahwa data mengandung pola trend. Selain itu pada plot ACF dan PACF pada Gambar.2, terlihat nilai-nilai autokorelasi pada plot ACF signifikan pada lag ke 12, 24, 36, dan kelipatan 12. Hal ini mengindikasikan bahwa data mengandung pola musiman.



Gambar 2. Plot ACF dan PACF Harga Beras

Di dalam metode SSA, terdapat dua tahap yang digunakan untuk memisahkan komponen *trend*, komponen musiman dan komponen *noise* dari deret waktu awal, yaitu tahap dekomposisi dan tahap rekonstruksi.

## Dekomposisi

### *Embedding*

Langkah pertama pada tahap dekomposisi yaitu tahap *embedding*. Pada proses ini dilakukan dengan mengubah deret waktu awal yang berdimensi satu menjadi deret multidimensi yang disebut dengan matriks lintasan. *Window Length* (L) merupakan parameter pada proses dekomposisi. Pada kasus ini jumlah data sebanyak 115 sehingga nilai L berkisar antara 2 hingga 58. Untuk memudahkan dalam pencarian L optimum yaitu dengan mencoba nilai L 10, 20, 30, 40, dan 50. Selanjutnya dipilih nilai L dengan MAPE minimum. Hasilnya sebagai berikut :

Tabel 3. Hasil MAPE *Window Length* 1

L	10	20	30	40	50
MAPE	2.91	1.83	1.01	1.29	5.39

Diperoleh L dengan MAPE minimum adalah 30. Dengan cara yang sama dilakukan *tracking* di sekitar nilai 30 untuk mendapatkan nilai L yang paling tepat, hasilnya :

Tabel 4. Hasil MAPE *Window Length* 2

L	28	29	30	31	32
MAPE	1.26	1.62	1.01	1.63	2.35

Diperoleh nilai L dengan minimum adalah 30 yakni 1.01 persen. Dengan MAPE tersebut diharapkan bahwa hasil prediksi yang diperoleh tidak berbeda jauh dengan nilai data sebenarnya. Dengan L yang diambil adalah L = 30 berdasarkan hasil dari *trial and error*. Dengan L yang telah ditentukan, kemudian akan didapatkan nilai K = 86 untuk membentuk matriks dengan orde LxK. Dengan demikian matriks lintasan **X** yang dibangun dari deret waktu awal dapat dituliskan sebagai berikut :

$$X = (x_{ij})_{i,j}^{30,86} = \begin{bmatrix} 6702 & 6888 & 6854 & \dots & 11571 \\ 6888 & 6854 & 6761 & \dots & 11494 \\ 6854 & 6761 & 6772 & \dots & 11449 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 8554 & 8606 & 8635 & \dots & 12021 \end{bmatrix}$$

### Singular Values Decomposition

Setelah didapat matriks  $X$ , tahap selanjutnya pembentukan matriks  $S = XX^T$  matriks ini berdimensi  $L \times L$  yang menghasilkan 3 komponen penting dalam tahap SVD yaitu *singularvalue* ( $\sqrt{\lambda_i}$ ), *eigenvector* ( $U_i$ ), dan *principal component* ( $V_i$ )

(a) *Singular value* ( $\sqrt{\lambda_i}$ )

Table 5. *Eigenvalue* dan *singular value*

No	<i>Eigenvalue</i>	<i>Singular value</i>
1	2.649052x10 <sup>11</sup>	514689.4616
2	1.486755x10 <sup>08</sup>	12193.2576
⋮	⋮	⋮
30	3.699792x10 <sup>04</sup>	291.4058

*Eigenvalue* dan *singular value* menunjukkan seberapa besar kontribusi komponen deret waktu terhadap proses pengelompokan data

(b) *Eigen Vector* ( $U_i$ )

Langkah selanjutnya adalah untuk mendapatkan *eigen vector*. Hal ini dilakukan untuk mempermudah penentuan karakteristik data dari matriks lintasan.

No	U1	U2	...	U30
1	-0.1693434	-0.261491504		-0.02847020
2	-0.1703291	-0.262820328		0.07682337
3	-0.1712682	-0.260668789		-0.11536422
⋮	⋮	⋮		⋮
30	-0.1939617	0.241548437		-0.09070846

(c) *Principal Component* ( $V_i$ )

No	V1	V2	...	V30
1	-42384.97	1867.99649	...	-33.0629796
2	-42730.27	1840.32712	...	25.6582661
3	-43048.19	1815.17293	...	-7.2144338
⋮	⋮	⋮	⋮	⋮
33	-50999.38	727.84498	...	-31.3107908

### Rekonstruksi

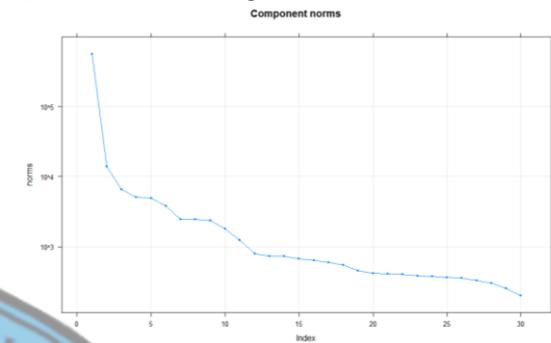
#### Grouping

Langkah awal dalam tahap rekosntruksi adalah melakukan pengelompokan terhadap *eigen triple* yang berhubungan dengan *trend*, *musiman*, dan *noise*. *Grouping effect* ( $r$ ) adalah

parameter yang digunakan pada tahap pengelompokan. *Grouping effect* digunakan untuk membatasi jumlah *eigen triple* yang akan digunakan untuk mengidentifikasi komponen *trend* dan *musiman*.

#### 1. Pengelompokan Komponen *Noise*

Nilai parameter  $r$  ditentukan berdasarkan jumlah *eigen triple* yang tidak mencerminkan *noise* pada plot dari *eigen value*. Pada plot dari *eigen value* yang menurun secara lambat mencerminkan komponen *noise*.

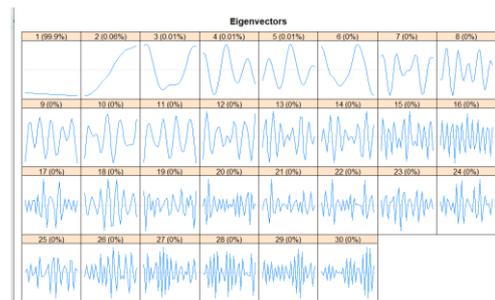


Gambar 3 Plot dari 30 *singular value*

Berdasarkan gambar 2 dapat dilihat bahwa *singular value* mulai menurun secara lambat pada *eigen triple* 11 sampai *eigen triple* 30 sehingga diidentifikasi sebagai komponen *noise*. Dengan demikian nilai parameter *grouping effect* ( $r$ ) adalah  $r = 11$ .

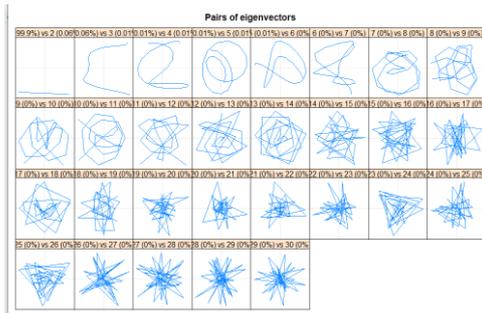
#### 2. Pengelompokan Komponen *Trend* dan Komponen *Musiman*

Sesudah komponen *noise* berhasil dikelompokan, langkah selanjutnya adalah melakukan pengelompokan terhadap *eigen triple* yang berhubungan dengan *trend* dan *seasonality*. Jumlah *eigen triple* yang digunakan untuk mengidentifikasi komponen *trend* dan komponen *seasonality* adalah 11 *eigen triple*. 11 *eigen triple* yang digunakan untuk mengidentifikasi komponen *trend* dan komponen *seasonality* terdiri dari *eigen triple* 1, *eigen triple* 2, *eigen triple* 3, ..., *eigen triple* 11.



Gambar 4 Plot 1D *Eigenvector*

Untuk mempermudah identifikasi pola musiman yang terbentuk sebaiknya dibuat grafik dua dimensi eigenvector berpasangan.



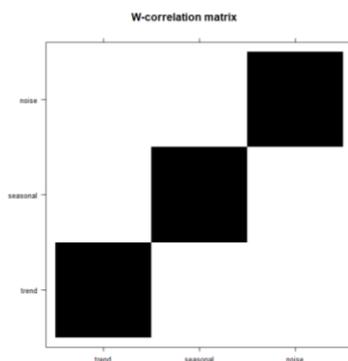
Gambar 5 Plot 2D Eigenvector

Semua komponen yang bervariasi secara lambat pada plot dari deret yang direkonstruksi diidentifikasi sebagai komponen *trend*. Berdasarkan gambar 3 dan 4, dapat dilihat bahwa *eigentriples* 1 dan 2 masuk dalam komponen *trend*, *eigentriples* ke 3-11 masuk dalam komponen *seasonal*, dan *eigentriples* 12-30 masuk dalam komponen *noise*.

Tabel 6 Eigentriple dan Komponen yang Terkait

Jenis Komponen	Eigentriple
Trend	1-2
Musiman	3-11
Noise	12-30

Kelompok yang terbentuk akan diidentifikasi keterpisahan komponen tersebut berdasarkan *w-correlation*.



Gambar 6 Plot w-correlation

Semakin pekat warna, maka akan semakin tinggi korelasinya.

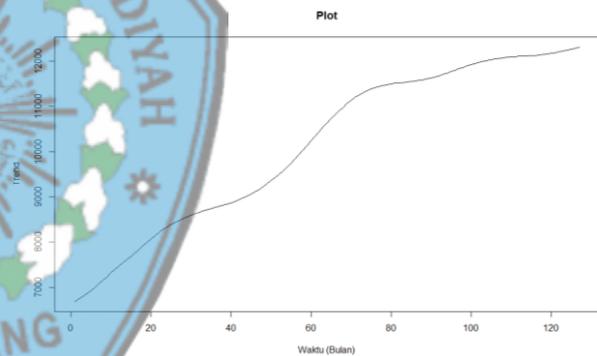
### Diagonal Averaging

Diagonal averaging merupakan tahap akhir dalam algoritma dasar SSA. Pada tahap ini, akan dibentuk data deret waktu baru berupa deret satu dimensi dengan panjang  $N$  berdasarkan kelompok yang telah dibentuk pada tahap sebelumnya. Setiap kelompok yang ada dibentuk rekonstruksi data berdasarkan pola kelompok tersebut.

Tabel 7 Hasil Diagonal Averaging

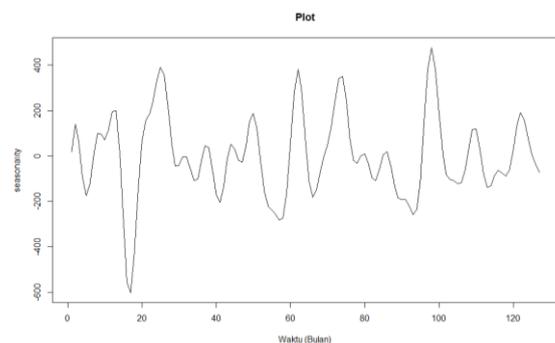
No	Hasil Rekonstruksi			Residual
	Trend	Seasonal	Diagonal Averaging	
1	6691.093	18.798	6710	-7.891
2	6743.226	140.673	6884	4.100
3	6795.646	66.495	6862	-8.141
4	6855.200	-92.399	6763	-1.801
⋮	⋮	⋮	⋮	⋮
127	12308.373	-71.379	12237	-23.993

Gambar 7 menampilkan hasil dari komponen *trend* yang direkonstruksi oleh *eigentriples* 1 dan 2



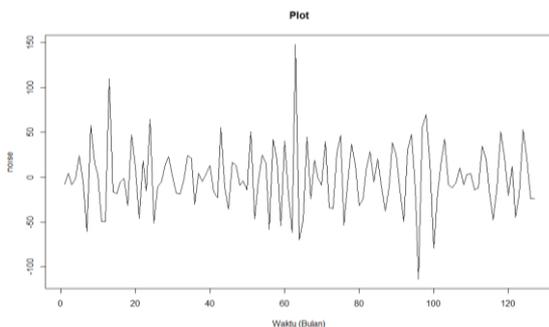
Gambar 7. Plot Trend yang Direkonstruksi

Pada langkah *diagonal averaging*, komponen musiman direkonstruksi oleh *eigentriples* 3 sampai 11.



Gambar 8 Plot Musiman yang Direkonstruksi

Pada langkah *diagonal averaging*, komponen *noise* direkonstruksi oleh *eigentriple* 12 sampai 31. Gambar 9 menampilkan hasil dari komponen *noise* yang direkonstruksi.



Gambar 9 Plot *Noise* yang Direkonstruksi

### Peramalan

Hasil *diagonal averaging* pada harga beras selanjutnya akan diolah dengan metode peramalan untuk mendapat hasil peramalan dengan metode *R-forecasting*. Setelah komponen *trend*, musiman, dan *noise* telah dipisahkan, selanjutnya adalah melakukan peramalan pada masing-masing komponen *trend* dan komponen *noise*. Hasil peramalan algoritma *Reccurent* yang diperoleh akan dibandingkan dengan data testing untuk validitas model. Model *Linear Reccurent Formula* (LRF) dapat dituliskan sebagai berikut:

$$g_i = \begin{cases} \tilde{y}_i & \text{untuk } i=1, \dots, 115 \\ \sum_{j=1}^{L-1} a_j g_{i-j} & \text{untuk } i=116, \dots, 127 \end{cases}$$

dengan nilai koefisien LRF  $a_j$  disajikan dalam tabel 8.

Tabel 8. Koefisien *Linear Reccurent Formula* ( $a_j$ )

$a_1$	$a_2$	...	$a_{29}$
0.082728103	0.026110023	...	0.133234890

Secara matematis, model SSA yang digunakan untuk meramalkan dapat dituliskan sebagai berikut :

- Data *training*  
 $g_i = \tilde{y}_i$   
 Untuk  $i = 1, 2, \dots, 115$
- Data *testing*

$$g_i = \sum_{j=1}^{L-1} a_j g_{i-j}$$

$$g_{116} = 0,08273g_{115} + 0,02611g_{114} + \dots + 0,13323g_{87}$$

$$g_{117} = 0,08273g_{116} + 0,02611g_{115} + \dots + 0,13323g_{88}$$

$$g_{118} = 0,08273g_{117} + 0,02611g_{116} + \dots + 0,13323g_{89}$$

$$g_{119} = 0,08273g_{118} + 0,02611g_{117} + \dots + 0,13323g_{90}$$

$$g_{120} = 0,08273g_{119} + 0,02611g_{118} + \dots + 0,13323g_{91}$$

$$g_{121} = 0,08273g_{120} + 0,02611g_{119} + \dots + 0,13323g_{92}$$

$$g_{122} = 0,08273g_{121} + 0,02611g_{120} + \dots + 0,13323g_{93}$$

$$g_{123} = 0,08273g_{122} + 0,02611g_{121} + \dots + 0,13323g_{94}$$

$$g_{124} = 0,08273g_{123} + 0,02611g_{122} + \dots + 0,13323g_{95}$$

$$g_{125} = 0,08273g_{124} + 0,02611g_{123} + \dots + 0,13323g_{96}$$

$$g_{126} = 0,08273g_{125} + 0,02611g_{124} + \dots + 0,13323g_{97}$$

$$g_{127} = 0,08273g_{126} + 0,02611g_{125} + \dots + 0,13323g_{98}$$

Kemudian diambil satu model, model  $g_{116}$  adalah peramalan pada bulan Agustus 2019, dimana  $g_{116}$  hingga  $g_{87}$  merupakan data harga beras pada bulan sebelum Agustus 2019. Hasil peramalan untuk data ke-116 didapatkan dari penjumlahan nilai-nilai koefisien LRF dikalikan dengan data sebelumnya. Sehingga didapatkan sebagai berikut:

$$g_{116} = 0,08273g_{115} + 0,02611g_{114} + \dots + 0,13323g_{87}$$

$$g_{116} = (0,08272 \times 12021) + (0,02611 \times 12009) - (0,01415 \times 12008) - (0,01200 \times 12019) + (0,02295 \times 12124) + (0,06068 \times 12222) + (0,06126 \times 12211) + (0,02450 \times 12106) - (0,01175 \times 12013) - (0,03273 \times 11926) - (0,03501 \times 11900) - (0,00881 \times 11899) + (0,01595 \times 11936) + (0,01179 \times 11907) - (0,01798 \times 11943) - (0,05470 \times 12035) - (0,07655 \times 12299) - (0,07668 \times 12414) - (0,07434 \times 12276) - (0,06410 \times 11838) - (0,05067 \times 11665) - (0,05413 \times 11552) - (0,07179 \times 11482) - (0,09629 \times 11411) - (0,13144 \times 11448) - (0,15529 \times 11465) - (0,12494 \times 11465) - (0,02489 \times 11449) + (0,13323 \times 11494)$$

$$g_{116} = 12098$$

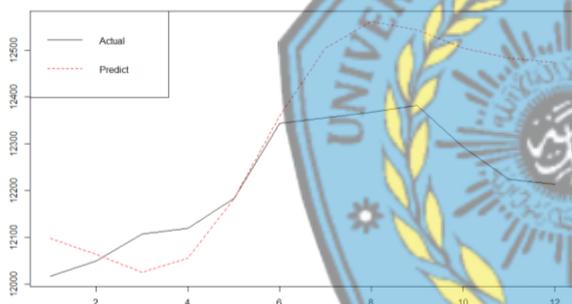
Setelah mendapat koefisien LRF, maka peramalan dapat dilakukan, dengan cara meramalkan data sebanyak 12 buah atau meramalkan harga beras 12 bulan. Akurasi hasil ramalan diukur dengan nilai MAPE, semakin kecil nilai MAPE maka semakin baik model dan akurat hasil ramalan yang diperoleh

dari suatu model. Nilai MAPE yang diperoleh dari hasil ramalan sebesar 1,01%.

Tabel 9. Perbandingan Hasil Peramalan Model SSA dengan Data Aktual

Bulan	Data Aktual	Hasil Peramalan
Agustus 2019	12018	12098
September 2019	12050	12064
Oktober 2019	12108	12026
November 2019	12120	12056
Desember 2019	12183	12183
Januari 2020	12343	12360
Februari 2020	12355	12504
Maret 2020	12368	12561
April 2020	12382	12543
Mei 2020	12293	12505
Juni 2020	12224	12484
Juli 2020	12213	12473

Plot hasil ramalan menggunakan metode SSA dan data aktual ditampilkan pada gambar 10.



Gambar 10. Perbandingan Hasil Peramalan dengan Data Aktual

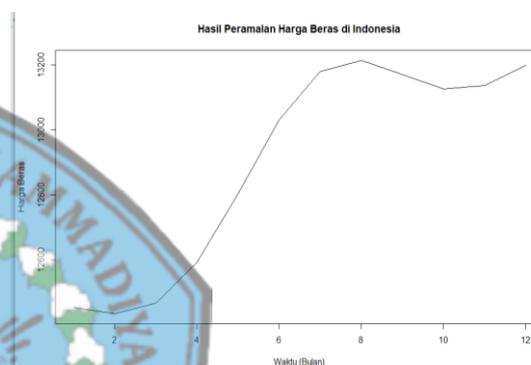
Tahap selanjutnya akan dilakukan peramalan Harga beras di Indonesia untuk 12 periode kedepan (periode Agustus 2020 hingga Juli 2021) dengan algoritma *Recurrent Forecasting* pada metode SSA.

Tabel 10 Hasil Peramalan Harga Beras di Indonesia

Periode	Hasil Ramalan
Agustus 2020	12454
September 2020	12435
Oktober 2020	12467
November 2020	12594
Desember 2020	12806
Januari 2021	13031

Februari 2021	13180
Maret 2021	13214
April 2021	13171
Mei 2021	13127
Juni 2021	13136
Juli 2021	13199

Dapat dilihat bahwa plot hasil peramalan memiliki plot yang sama meskipun dengan jumlah yang berbeda. Hal itu menunjukkan bahwa terjadi kenaikan harga setiap bulannya pada hasil peramalan diperoleh harga paling tinggi pada bulan Maret 2021 sebesar Rp 13214 dan harga terendah pada bulan September 2020 sebesar Rp 12435.



Gambar 11. Hasil Peramalan Harga Beras

## Simpulan dan Saran

### Simpulan

Dengan nilai MAPE yang didapatkan sebesar 1,01%. Hasil peramalan harga beras di tingkat perdagangan besar (grosir) di Indonesia didapatkan hasil harga beras memiliki harga tertinggi pada bulan Maret 2021 sebesar Rp 13214 dengan jumlah terendah pada bulan September 2020 sebesar Rp 12435.

### Saran

Penentuan parameter *window length* dan pembentukan *grouping* harus dicari seefisien mungkin agar dapat memberikan hasil peramalan yang terbaik. Selain itu diperlukan juga penelitian mengenai pengujian identifikasi pola sirkular (lingkaran) pada grafik *pairs of eigenvector* untuk penelitian selanjutnya agar proses *grouping* dapat dilakukan secara efektif.

### Daftar Pustaka

- [1] Bappenas. 2010. *Produksi Beras di Indonesia*.
- [2] Aryanti. 2001. *Kebutuhan Konsumen*

- Produk Beras.
- [3] Hassani, Heravi, dan Zhigljavsky. (2009). Peramalan Produksi Industri Eropa Menggunakan SSA, ARIMA, dan Holt-Winters.
  - [4] Ete, A. A. (2017). Forecasting the Amount of Foreign Tourist Travelers To Indonesia By Entrance Guide Using Singular Spectrum Analysis and Arima Forecasting the Amount of Foreign Tourist Travelers To Indonesia By Entrance Guide Using Singular Spectrum. Master Thesis. ITS.
  - [5] Widiarsi, N. . (2015). Analisis Komparasi *Holt Winter* Dan *Sarima* Pada Peramalan Statistik Wisatawan Asing Kraton Yogyakarta. 95–100.
  - [6] Zhigljavsky, A. (2011). *Singular Spectrum Analysis for Time Series*. In *International Encyclopedia of Statistical Science*. [https://doi.org/10.1007/978-3-642-04898-2\\_521](https://doi.org/10.1007/978-3-642-04898-2_521)
  - [7] Golyandina, N. (2001). *Analysis of Time Series Structure: SSA and Related Techniques*. *Journal of the American Statistical Association*, 97(460), 1207–1208. <https://doi.org/10.1198/jasa.2002.s239>
  - [8] Ischak, R. (2018). Peramalan Rata-Rata Harga Beras di Tingkat Penggilingan Menggunakan Model *Singular Spectrum Analysis* (SSA). 57–64

