

# PERAMALAN WISATAWAN MANCANEGERA BERKUNJUNG KE BALI MENGUNAKAN METODE *SINGULAR SPECTRUM ANALYSIS* (SSA)

Dheanyta Alif Shafira <sup>(1)</sup>, Tiani Wahyu Utami <sup>(2)</sup> dan Prizka Rismawati Arum <sup>(3)</sup>

<sup>(1,2,3)</sup> Program Studi Statistika Universitas Muhammadiyah Semarang

Email : [dheanyta04@gmail.com](mailto:dheanyta04@gmail.com)

---

Article history	Abstract
Submission :	Foreign tourists are any people who visit a country outside of their place of residence. Data about foreign tourists is a time series data that can be used to forecast. The method used in predicting foreign tourist data is Singular Spectrum Analysis (SSA). Case studies in the application of the SSA are used to predict the number of foreign tourist visits to Bali. The advantage of SSA is getting a much smaller error value and getting more stable results. The purpose of this forecasting is to get the forecasting results using the Singular Spectrum Analysis (SSA) algorithm in forecasting the number of foreign tourists visiting Bali. Measurement of forecasting accuracy uses Mean Absolute Percentage Error (MAPE). The MAPE value is 10.79% with the forecast result tends to increase every month.
Revised :	
Accepted :	
<b>Keyword:</b>	
Bali, Foreign Tourists, SSA	

---

## Pendahuluan

Wisatawan Mancanegara adalah setiap orang yang mengunjungi suatu negara di luar tempat tinggalnya, di didorong oleh satu atau beberapa keperluan tanpa bermaksud memperoleh penghasilan di tempat yang di kunjungi dan lamanya kunjungan tidak lebih dari 12 bulan[1]. Keindahan alam di Indonesia, khususnya di Pulau Dewata Bali membuat banyak wisatawan mancaegara senantiasa berkunjung ke Bali. Pada tahun 2017, wisatawan mancanegara yang datang mengunjungi Pulau Dewata Bali sebanyak 5697739 pengunjung, meningkat sebanyak 372734 pengunjung atau sebanyak 6070734 pengunjung, meningkat sebesar 6,14% dari tahun 2017. Data jumlah kunjungan wisatawan mancanegara 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 menggunakan metode *Time Invariant Fuzzy Time Series*[2], Jaringan Syaraf Tiruan[3], *Holt-Winter* 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[5]. 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* [5].

$$\mathbf{X} = [X_1, \dots, X_p] = (x_{ij})_{L \times K} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1L} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2L} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3L} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{L1} & x_{L2} & x_{L3} & \dots & x_{LK} \end{bmatrix}$$

## 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 [7]. Dasar dari *Singular Spectrum Analysis* (SSA) adalah dekomposisi dengan menguraikan data deret waktu menjadi komponen-komponennya yaitu tren, musiman, siklis, dan *noise*[8]. Tahapan dalam melakukan analisis menggunakan *Singular Spectrum Analysis* (SSA) terbagi menjadi dua yaitu dekomposisi dan rekonstruksi[7].

### Dekomposisi

Pada tahap dekomposisi, parameter yang digunakan adalah *Window Length* (L) [8]. 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  $\mathbf{X}$  adalah [9] :

Lag vektor  $X_i$  adalah kolom dari matriks lintasan  $\mathbf{X}$ . Baris dan kolom dari  $\mathbf{X}$  bagian dari data deret waktu. Elemen  $(i,j)$  dari matriks  $\mathbf{X}$  adalah  $x_{ij} = x_{i+j-1}$  yang menghasilkan  $\mathbf{X}$  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  $\mathbf{X}$ . Didefinisikan  $\mathbf{S} = \mathbf{X}\mathbf{X}^T$  dan ditunjukkan dengan  $\lambda_1, \dots, \lambda_L$  *eigenvalue* dari  $\mathbf{S}$  sedangkan *singular value* dinotasikan dengan  $U_1, \dots, U_L$  merupakan *eigenvector* yang sesuai dengan *eigenvalues* pada matriks  $\mathbf{S}$ .

Selanjutnya didefinisikan  $d = \text{rank } \mathbf{X} = \max \{i, \text{dimana } \lambda_i > 0\}$  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  $\mathbf{X}$  sebagai berikut

$$\mathbf{X} = X_1 + \dots + X_d$$

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

$$\mathbf{X} = X_1 + X_2 + \dots + X_d$$

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

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

### 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 [5]

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

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 [5]

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

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  $\mathbf{X}_{jk}$  membuat deret baru  $\widetilde{X}^{(k)} = (x_1^{(k)}, \dots, x_N^{(k)})$ . Oleh karena itu deret  $x_1, \dots, x_N$  adalah didekomposisi menjadi jumlah  $m$  deret yaitu :

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

### 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 [7].

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 [9].

### Reccurent 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 Pencapaian 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 [7]. Maka dari itu usaha untuk membuat nilai *error* seminimal mungkin dibutuhkan pada proses peramalan.

Salah satu tingkat akurasi 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 akurasinya 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 bali.bps.go.id berupa data jumlah kunjungan wisatawan mancanegara. Data jumlah kunjungan wisatawan mancanegara yang digunakan adalah data jumlah kunjungan wisatawan mancanegara pada bulan Januari 2010 sampai dengan Oktober 2019 dengan jumlah data sebanyak 118. Data periode Januari 2010 sampai dengan Oktober 2018 digunakan sebagai data pengamatan untuk model *training* dengan jumlah data sebanyak 106 data yang berfungsi untuk membentuk suatu model, sedangkan periode November 2018 sampai dengan Oktober 2019 digunakan sebagai data yang berfungsi untuk memvalidasi model yang terpilih dari data *testing* dengan jumlah data sebanyak 12 data.

### Variabel Penelitian dan Struktur Data

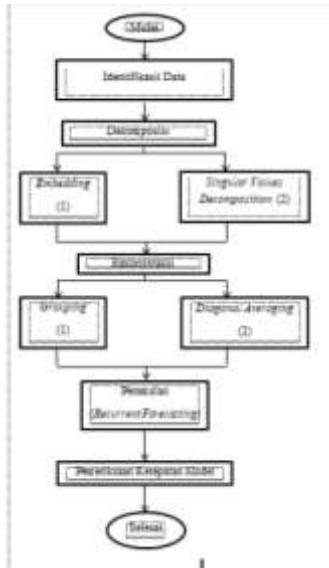
Tabel 1 Variabel data

Variabel	Keterangan	Sumber
$Y_{1,n}$	Jumlah Kunjungan Wisatawan Mancanegara Bulanan di Bali	Badan Pusat Statistik Provinsi Bali

Tabel 2 Struktur data

n	Bulan	Tahun	$Y_{1,n}$
1	1	2010	$Y_{1,1}$
2	2	2010	$Y_{1,2}$
:	:	:	:
6	6	2010	$Y_{1,6}$
:	:	:	:
12	12	2010	$Y_{1,12}$
:	:	:	:
118	11	2019	$Y_{1,118}$

## Langkah Penelitian



## Hasil Penelitian dan Pembahasan

### Analisis Deskriptif

Gambaran umum mengenai jumlah wisatawan mancanegara ke Bali dijelaskan menggunakan statistik deskriptif sebagai berikut :

Tabel 2 Statistik Deskriptif Kedatangan Wisatawan Mancanegara ke Bali periode Januari 2010 sampai Oktober 2019.

Rata-rata	Standar Deviasi	Minimum	Maksimum
350303	119706	179273	624366

Pada tabel diatas dapat dilihat bahwa rata-rata kunjungan wisatawan mancanegara ke Bali sebesar 350303 jiwa dengan standar deviasi sebesar 119706. Jumlah minimum kedatangan wisatawan mancanegara ke Bali mencapai 179273 jiwa dan jumlah maksimum kedatangan wisatawan mancanegara mencapai 624366 jiwa.

Penelitian ini menggunakan data dari Badan Pusat Statistik (BPS) terkait jumlah wisatawan mancanegara berkunjung ke Bali pada periode Januari 2010 hingga Oktober 2019. Data tersebut merupakan data deret waktu yang dibagi menjadi dua bagian:

- Data Training : Januari 2010 – Oktober 2018
- Data Testing : November 2018 – Oktober 2019

Peramalan jumlah wisatawan mancanegara berkunjung ke Bali dilakukan dengan menggunakan metode *Singular Spectrum Analysis* (SSA). Sebelum melakukan analisis data, terlebih dahulu dilakukakan *plotting* data jumlah wisatawan mancanegara berkunjung ke Bali sehingga terlihat pergerakan jumlah wisatawan selama periode tersebut. Gambar dapat dilihat pada gambar 1.



Gambar 1. Plot Jumlah Kunjungan Wisatawan

Pada gambar diatas, terlihat bahwa jumlah kunjungan wisatawan Bali mengandung komponen *trend* dan komponen musiman. 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 106 sehingga nilai L berkisar antara 2 hingga 53. 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	<b>12,09</b>	33,58	30,44	26,51	22,20

Diperoleh L dengan MAPE minimum adalah 10. Dengan cara yang sama dilakukan *tracking* di sekitar nilai 10 untuk mendapatkan nilai L yang paling tepat, hasilnya :

Tabel 4. Hasil MAPE *Window Length 2*

L	8	9	10	11	12
MAPE	15,29	16,06	12,09	<b>10,79</b>	11,25

Diperoleh nilai L dengan minimum adalah 11 yakni 10,79 persen. Dengan MAPE tersebut diharapkan bahwa hasil prediksi yang diperoleh tidak berbeda jauh dengan nilai data sebenarnya. Dengan L yang diambil adalah L = 11 berdasarkan hasil dari *trial and error*. Dengan L yang telah ditentukan, kemudian akan didapatkan nilai K = 96 untuk membentuk matriks dengan orde LxK. Dengan demikian matriks lintasan X yang dibangun dari deret waktu awal dapat dituliskan sebagai berikut :

$$X_{11 \times 96} = \begin{bmatrix} 179273 & 191926 & 192579 & \dots & 315909 \\ 191926 & 192579 & 184907 & \dots & 358065 \\ 192579 & 184907 & 203388 & \dots & 452423 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 199861 & 227251 & 209093 & \dots & 517889 \end{bmatrix}$$

### Singular Values Decomposition

Proses SVD dalam tahap ini menghasilkan *eigen value*, *eigen vector*, dan *Principal component* sehingga akan membantu dalam pengelompokan jumlah kunjungan wisatawan Bali ke dalam beberapa komponen utama yaitu komponen *trend*, komponen musiman, dan komponen *noise*. Matriks lintasan X didekomposisi pada langkah SVD menjadi 11 *eigen triple* yang terdiri dari 11 *eigen value*, 11 *eigen vector*, dan 0 vektor faktor.

#### a. Eigen Value ( $\sqrt{\lambda_i}$ )

Hasil *eigen value* diperoleh dari matriks lintasan yang didefinisikan dengan  $S = XX^T$

Table 5. Hasil *eigen value*

L	Eigen Value
1	11052538
2	786699,5
3	710684,7
4	425720,9
5	341024,9
⋮	⋮
11	127965,1

#### b. Eigen Vector ( $U_i$ )

Langkah selanjutnya adalah untuk mendapatkan *eigen vector*. Hal ini dilakukan

untuk mempermudah penentuan karakteristik data dari matriks lintasan.

$$U_{11 \times 11} = \begin{bmatrix} -0,28726 & -0,171764 & 0,364535 & \dots & -0,21189 \\ -0,28855 & -0,328817 & 0,314325 & \dots & 0,41912 \\ -0,29102 & -0,408681 & 0,127735 & \dots & -0,31064 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -0,31736 & 0,224253 & 0,391744 & \dots & -0,17625 \end{bmatrix}$$

#### c. Principal Component ( $V_i$ )

*Principal Component* dapat dituliskan sebagai berikut :

$$V_{11 \times 96} = \begin{bmatrix} -0,06420 & 0,0554705 & -0,061727 & \dots & 0,021222 \\ -0,05468 & 0,022853 & -0,077819 & \dots & -0,002054 \\ -0,065873 & -0,0166269 & -0,084186 & \dots & 0,027085 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -0,150121 & 0,188236 & -0,215709 & \dots & 0,0377866 \end{bmatrix}$$

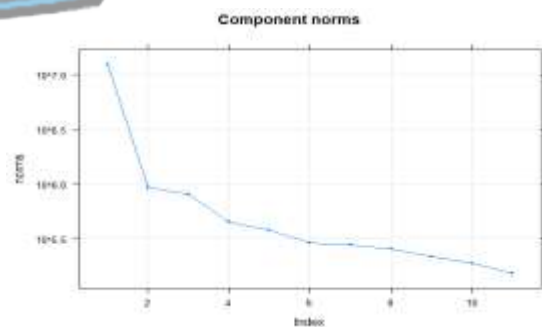
### Rekonstruksi

#### Grouping

Langkah awal dalam tahap rekonstruksi 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*.



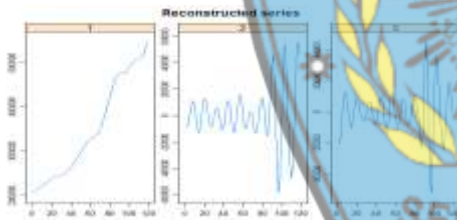
Gambar 2 Plot dari 11 *eigen value*

Berdasarkan gambar 2 dapat dilihat bahwa *eigen value* mulai menurun secara lambat pada *eigen triple* 4 sampai *eigen triple* 11 sehingga diidentifikasi sebagai komponen *noise*. Dengan demikian nilai parameter *grouping effect* (r) adalah r = 3 karena jumlah

*eigen triple* yang tidak mencerminkan *noise* pada plot dari *eigen value* adalah 3. Meskipun *eigen triple* yang mencerminkan *noise* sudah diidentifikasi, namun ada kemungkinan jumlah *eigen triple* yang mencerminkan *noise* sudah diidentifikasi, namun ada kemungkinan jumlah *eigen triple* yang mencerminkan *noise* bisa bertambah. Sisa *eigen triple* yang tidak berhubungan dengan *trend* dan musiman dari 3 *eigen triple* pertama akan dikelompokkan ke dalam kelompok *noise*.

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

Sesudah komponen *noise* berhasil dikelompokkan, langkah selanjutnya adalah melakukan pengelompokan terhadap *eigen triple* yang berhubungan dengan *trend* dan musiman. Jumlah *eigen triple* yang digunakan untuk mengidentifikasi komponen *trend* dan komponen musiman adalah 3 *eigen triple*. 3 *eigen triple* yang digunakan untuk mengidentifikasi komponen *trend* dan komponen musiman yang terdiri dari *eigen triple* 1, *eigen triple* 2, *eigen triple* 3. Plot dari deret yang direkonstruksi dapat digunakan untuk mengidentifikasi *eigen triple* yang berhubungan dengan *trend* dan musiman.



Gambar 3 Plot deret yang tidak termasuk komponen *noise*

Semua komponen yang bervariasi secara lambat pada plot dari deret yang direkonstruksi diidentifikasi sebagai komponen *trend*. Berdasarkan gambar 3, dapat dilihat bahwa deret yang direkonstruksi oleh *eigen triple* 1 memuat komponen yang bervariasi secara lambat, sehingga *eigen triple* 1 dikelompokkan ke dalam kelompok *trend*.

Sesudah *eigen triple* yang berhubungan dengan *trend* berhasil dikelompokkan, langkah selanjutnya adalah melakukan pengelompokan terhadap *eigen triple* yang berhubungan dengan musiman. Pengelompokan terhadap *eigen triple* yang berhubungan dengan musiman dilakukan berdasarkan kemiripan *eigen value* dari dua *eigen triple* yang berurutan. Pada plot

dari deret yang direkonstruksi oleh suatu *eigen triple* memiliki pola musiman yang sama dan periode musiman yang sama dengan deret yang direkonstruksi oleh *eigen triple* yang lainnya. *Eigen triple* yang menunjukkan komponen musiman pada *eigen triple* 2 hingga *eigen triple* 3.

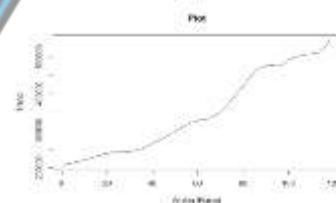
Dengan demikian sisa *eigen triple* yang tidak dikelompokkan ke dalam kelompok *trend* dan musiman dari 3 *eigen triple* pertama akan dikelompokkan ke dalam kelompok *noise*.

Tabel 6 *Eigen triple* dan Komponen yang Terkait

Jenis Komponen	<i>Eigen triple</i>
Musiman	2, 3
<i>Trend</i>	1
<i>Noise</i>	4,5,6,...,11

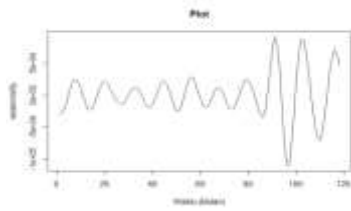
## *Diagonal Averaging*

Pada langkah terakhir dalam tahap rekonstruksi yaitu *diagonal averaging* masing-masing komponen dapat direkonstruksi dengan menggunakan masing-masing *eigen triple* 1. Gambar 4. menampilkan hasil dari komponen *trend* yang direkonstruksi oleh *eigen triple* tersebut.



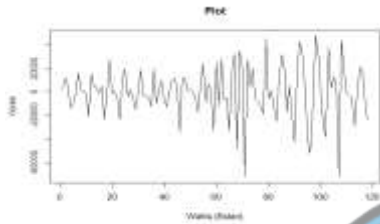
Gambar 4. Plot *Trend* yang Direkonstruksi

Pada langkah *diagonal averaging*, komponen musiman direkonstruksi oleh *eigen triple* 2, dan *eigen triple* 3. Gambar 5 menampilkan hasil dari komponen musiman yang direkonstruksi oleh keenam *eigen triple* tersebut.



Gambar 5 Plot Musiman yang Direkonstruksi

Pada langkah *diagonal averaging*, komponen *noise* direkonstruksi oleh *eigentriple* 4, *eigentriple* 5, *eigentriple* 6,..., *eigentriple* 11. Gambar 6 menampilkan hasil dari komponen *noise* yang direkonstruksi.



Gambar 6 Plot *Noise* yang Direkonstruksi

### Peramalan

Setelah komponen *trend*, musiman, dan *noise* telah dipisahkan, langkah selanjutnya adalah melakukan peramalan pada masing-masing komponen kecuali komponen *noise*. Peramalan yang digunakan untuk meramalkan komponen *trend* dan komponen musiman adalah algoritma *Recurrent Forecasting*. Nilai ramalan yang didapatkan dari masing-masing komponen terdiri dari nilai ramalan untuk data *training* dan nilai ramalan untuk data *testing*. Hasil peramalan dengan algoritma *Recurrent* yang diperoleh akan dibandingkan dengan data *testing* untuk validitas model. Model *Linear Recurrent Formula* (LRF) dapat dituliskan sebagai berikut :

$$g_i = \begin{cases} \tilde{y}_i & \text{untuk } i=1,\dots,106 \\ \sum_{j=1}^{L-1} a_j g_{i-j} & \text{untuk } i=107,\dots,118 \end{cases}$$

dengan nilai koefisien LRF  $a_j$  disajikan dalam tabel 7.

Tabel 7. Koefisien *Linear Recurrent Formula* ( $a_j$ )

No.	$a_j$
1	0,301673
2	0,185170
3	0,045810
4	-0,073541

5	-0,153382
6	-0,141964
7	-0,034111
8	0,143466
9	0,313873
10	0,445119

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

- a. Data *training*

$$g_i = \tilde{y}_i$$

untuk  $i = 1, 2, \dots, 106$

- b. Data *testing*

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

$$g_{107} = 0,3016732 g_{106} + 0,1851703 g_{105} + \dots + 0,4451194 g_{97}$$

$$g_{108} = 0,3016732 g_{107} + 0,1851703 g_{106} + \dots + 0,4451194 g_{98}$$

$$g_{109} = 0,3016732 g_{108} + 0,1851703 g_{107} + \dots + 0,4451194 g_{99}$$

$$g_{110} = 0,3016732 g_{109} + 0,1851703 g_{108} + \dots + 0,4451194 g_{100}$$

$$g_{111} = 0,3016732 g_{110} + 0,1851703 g_{109} + \dots + 0,4451194 g_{101}$$

$$g_{112} = 0,3016732 g_{111} + 0,1851703 g_{110} + \dots + 0,4451194 g_{102}$$

$$g_{113} = 0,3016732 g_{112} + 0,1851703 g_{111} + \dots + 0,4451194 g_{103}$$

$$g_{114} = 0,3016732 g_{113} + 0,1851703 g_{112} + \dots + 0,4451194 g_{104}$$

$$g_{115} = 0,3016732 g_{114} + 0,1851703 g_{113} + \dots + 0,4451194 g_{105}$$

$$g_{116} = 0,3016732 g_{115} + 0,1851703 g_{114} + \dots + 0,4451194 g_{106}$$

$$g_{117} = 0,3016732 g_{116} + 0,1851703 g_{115} + \dots + 0,4451194 g_{107}$$

$$g_{118} = 0,3016732 g_{117} + 0,1851703 g_{116} + \dots + 0,4451194 g_{108}$$

Diambil satu model, model  $g_{107}$  adalah peramalan pada bulan November 2018, dimana  $g_{106}$  hingga  $g_{97}$  merupakan data jumlah



kunjungan wisatawan mancanegara pada bulan sebelum November 2018. Hasil peramalan untuk data ke-107 didapatkan dari penjumlahan nilai-nilai koefisien LRF dikalikan dengan data sebelumnya. Sehingga didapatkan sebagai berikut :

$$g_{107} = 0,3017 g_{106} + 0,1852 g_{105} + 0,0458 g_{104} - 0,0735 g_{103} - 0,1534 g_{102} - 0,1420 g_{101} - 0,0341 g_{100} + 0,1435 g_{99} + 0,3139 g_{98} + 0,4451 g_{97}$$

$$g_{107} = 0,3017 (517889) + 0,1852 (555903) + 0,0458 (573766) - 0,0735 (624366) - 0,1534 (544550) - 0,1420 (528512) - 0,0341 (516777) + 0,1435 (492678) + 0,3139 (452423) + 0,4451 (358065)$$

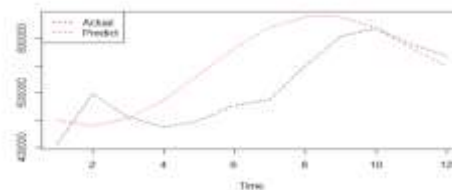
$$g_{107} = 450977$$

Setelah model SSA yang digunakan untuk melakukan peramalan telah dibentuk, langkah selanjutnya adalah melakukan peramalan pada data *training* berdasarkan model yang diperoleh. 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 10,79 %. Tabel 8 menyajikan hasil peramalan menggunakan algoritma *Recurrent Forecasting* pada metode SSA untuk 12 bulan ke depan.

Tabel 8. Perbandingan Hasil Peramalan Model SSA dengan Data Aktual

Tahun	Bulan	Hasil Ramalan	Data Aktual
2018	November	450977	406725
2018	Desember	439439	498819
2019	Januari	453109	455708
2019	Februari	487658	437537
2019	Maret	533966	449637
2019	April	581358	477125
2019	Mei	619688	488432
2019	Juni	640205	549751
2019	Juli	639765	604493
2019	Agustus	618766	618982
2019	September	582182	590565
2019	Oktober	548390	568067

Plot hasil ramalan menggunakan metode SSA dan data aktual ditampilkan pada gambar 7.



Gambar 7. Perbandingan Hasil Peramalan dengan Data Aktual

Tahap selanjutnya akan dilakukan peramalan Wisatawan Mancanegara berkunjung ke Bali untuk 12 periode kedepan (periode November 2019 hingga Oktober 2020) dengan algoritma *Recurrent Forecasting* pada metode SSA. Hasil peramalan disajikan dalam tabel 9.

Tabel 9 Hasil Peramalan Wisatawan Mancanegara ke Bali

Periode	Hasil Ramalan
November 2019	524708
Desember 2019	518512
Januari 2020	532686
Februari 2020	564806
Maret 2020	607500
April 2020	650364
Mei 2020	683194
Juni 2020	698501
Juli 2020	694223
Agustus 2020	674580
September 2020	646261
Oktober 2020	618842

Secara visual, plot hasil peramalan jumlah kunjungan wisatawan mancanegara Bali ditunjukkan pada gambar 8.



Gambar 8. Hasil Peramalan Jumlah Kunjungan Wisatawan Mancanegara Bali

Dapat dilihat bahwa plot hasil peramalan memiliki plot yang sama meskipun

dengan jumlah yang berbeda. Hal itu menunjukkan bahwa setiap bulan Desember, jumlah kunjungan wisatawan mancanegara berada pada titik terendah dengan jumlah kunjungan paling sedikit setiap tahun, sebaliknya pada bulan Juni jumlah kunjungan wisatawan mancanegara berada pada titik tertinggi dengan jumlah kunjungan paling banyak setiap tahun. Untuk hasil peramalan yang didapatkan bahwa pada bulan Desember 2019 sebanyak 518512 wisatawan mancanegara berkunjung ke Bali dan pada bulan Juni 2020 sebanyak 698501 wisatawan mancanegara berkunjung ke Bali.

## Simpulan dan Saran

### Simpulan

Dengan nilai MAPE yang didapatkan sebesar 10,79%. Hasil peramalan jumlah kunjungan wisatawan mancanegara ke Bali dari bulan November 2019 hingga Oktober 2020 didapatkan hasil jumlah kunjungan wisatawan mancanegara berkunjung ke Bali memiliki jumlah kunjungan tertinggi pada bulan Juni 2020 sebanyak 698501 dengan jumlah terendah pada bulan Desember 2019 sebanyak 518512.

### Saran

Saran yang dapat diberikan untuk penelitian selanjutnya yaitu dapat menambah jumlah data agar tingkat akurasi yang didapatkan dapat lebih baik. Dapat juga mencari nilai  $L$  dan  $r$  yang lebih optimal.

## Daftar Pustaka

- [1] Badan Pusat Statistik. 2019. *Konsep dan Definisi Statistik Kunjungan Wisatawan Mancanegara*.
- [2] Elfajar, A. B., Setiawan, B. D., dan Dewi, C. (2017). Peramalan Jumlah Kunjungan Wisatawan Kota Batu Menggunakan Metode *Time Invariant Fuzzy Time Series*. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK)* Universitas Brawijaya, 1(2), 85–94.
- [3] Misriati, T. (2016). Peramalan Jumlah Kunjungan Wisatawan Mancanegara Ke Lombok Menggunakan Jaringan Syaraf Tiruan. *Seminar Nasional Ilmu Pengetahuan Dan Teknologi Komputer*

*Nusa Mandiri*, 13–17.

- [4] Widiarsi, N. . (2015). Analisis Komparasi *Holt Winter* Dan *Sarima* Pada Peramalan Statistik Wisatawan Asing Kraton Yogyakarta. 95–100.
- [5] 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)
- [6] 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>
- [7] Ischak, R. (2018). Peramalan Rata-Rata Harga Beras di Tingkat Penggilingan Menggunakan Model *Singular Spectrum Analysis (SSA)*. 57–64.
- [8] 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*.

