BAB II

TINJAUAN PUSTAKA

2.1 Gabah Kering Giling (GKG)

Gabah merupakan salah satu komoditi pertanian yang permintaan produksinya selalu meningkat dari waktu ke waktu. Buah dari hasil tanaman padi (Oryza Sativa Lineus) yang telah terlepas dari tangkainya tersebut terbagi menjadi 2 yaitu Gabah Kering Giling (GKG) dan Gabah Kering Panen (GKP). Gabah Kering Giling merupakan gabah yang mempunyai kandungan kadar air maksimum dengan besar 14,0 persen serta mempunyai hampa kotoran maksimum sebesar 3,0 persen (Badan Pusat Statistik,2020). Harga GKG di tingkat petani mengalami fluktuatif setiap tahunnya. Dari 1840 transaksi penjualan di 29 provinsi selama Agustus 2020 tercatat penjualan Gabah Kering Giling (GKG) sebesar 19,02 persen. Pada bulan Agustus 2019 rata-rata harga GKG di tingkat petani naik sebesar 1,65 persen, namun pada Agustus 2020 turun sebesar 1,01 persen atau Rp 5396,- per Kg.

2.2 Peramalan

Menurut (Heizer dan Barry,2015:112 dalam Indah dan Rahmadani, 2018) peramalan adalah seni dan juga ilmu pengetahuan yang digunakan dalam memprediksi peristiwa dimasa yang akan datang. Ada 3 tipe peramalan untuk membuat rancangan operasional dimasa mendatang, dintaranya yaitu :

1. Peramalan ekonomi (*economic forecast*) yaitu peramalan yang dapat menangani siklus bisnis dengan memprediksi tingkat inflasi, uang beredaar, dan indicator perencanaan lainnya.
2. Peramalan teknologi (*technological forecast*) yaitu peramalan yang berkaitan dengan tingkat perkembangan tenologi yang dapat membuat produk baru dan menarik yang membutuhkan peralatan baru
3. Peramalan permintaan (*demand forecast*) yaitu peramalan yang berkaitan dengan permintaan produk barang maupunjasa dari suatu perusahaan. peramalan akan memberikan informasi dan keakuratan terhadap permintaan sesungguhnya sehingga dapat membantu manajer dalam mengambil keputusan.

Rusdina (2014:97) dalam Indah dan Rahmadani (2018) juga menjelaskan bahwa peramalan yang bauk mempunyai kriteria penting, diantaranya yaitu :

1. Akurasi hasil ramalan diukur dari hasil kebiasaan dan konsisten \nya ramalan tersebut
2. Hasil peramalan dikatakan biasa jika hasil ramalan terlalu tinggi atau rendah dari kenyataan sebenarnya
3. Hasil ramalan dikatakan konsistn jika besar kesalahan dalam ramalan relatif kecil

2.3 *Time Series*

Time series atau runtun waktu merupakan serangkaian observasi data yang runtut dalam waktu. Metode *time series* merupakan peramalan yang memperkirakan variabel waktu dengan menggunakan analisa pola hubungan antar variabel. Yang harus diperhatikan dalam suatu data *time series* adalah pola atau tipe yang dimiliki oleh data tersebut. Menurut Hanke & Wichern (2005) dalam Setyowati (2018) terdapat empat pola data *time series* diantaranya yaitu :

1. Pola Horizontal

   Yaitu pola dengan kejadian yang tidak terduga serta mempunyai sifat acak, namun adanya pola tersebut akan mempengaruhi fluktuasi data *time series*

2. Pola Trend

   Merupakan pola data yang memiliki kecenderungan data mengalami kenaikan maupun penurunan dalam jangka yang panjang.

3. Pola Musiman

   Fluktuasi data yang terjadi secara periodik pada kurun waktu satu tahun (triwulanan, kuartalan, bulanan, mingguan maupun harian) disebut sebagai pola musiman

4. Pola Siklis

   Pola siklis merupakan fluktuasi data *time series* untuk waktu lebih dari satu tahun.

2.4 *Singular Spectrum Analysis* (SSA)

Menurut Golyandina (2001) *Singular Spectrum Analysis* merupakan teknik analisis *time series* yang mengkombinasikan analisis *time series* klasik, statistika multivariat, geometri multivariat, system dinamis dan pemprosesan sinyal. Terdapat banyak elemen statistik serta probabilistik yang digunakan dalam metode SSA diantaranya yaitu yang berhubungan dengan stasioneritas, ergodisitas, komponen utama dan teknik *bootstrap*. Metode SSA merupakan metode non parametrik yang relatif baru dan telah diaplikasikan pada *time series* yang berbeda mulai dari ekonomi sampai fisika. Darmawan (2015) menjelaskan bahwa metode SSA telah berusaha menguraikan data *time series* asli menjadi beberapa komponen kecil yang dapat ditafsirkan seperti komponen *trend*, komponen osilasi (gerakan berulang) dan *noise*. Dalam metode ini terdapat dua tahapan yaitu dekomposisi dan rekontruksi. Dalam tahap dekomposisi langkah yang perlu dilakukan yaitu *embedding* dan *singular value decomposition*. Kemudian pada tahap rekontruksi langkah yang perlu dilakukan yaitu tahap *grouping* dan *diagonal averaging*.

2.4.1 Dekomposisi

Pada Dekomposisi, parameter yang mempunyai fungsi penting dalam tahapan dekomposisi adalah *Windows Length* (L). Nilai L yang digunakan ditentukan melalui pengecekan *trial and error* karena belum ada metode khusus untuk menentukan nilai L. Tahap dekomposisi itu sendiri mempunyai tahapan-tahapan yang harus dilakukan yaitu *Embedding* dan *Singular Value Decomposittion* (SVD).

2.4.1.1 *Embedding*

Embedding merupakan proses merubah struktur deret waktu menjadi deret *multidimentional*. Misalnya $X = (X_1, X_2, \dots, X_N)$ merupakan data deret waktu sebanyak N dan tidak ada data yang hilang. Kemudian X akan ditransformasi ke dalam matriks lintasan berukuran $L\ x\ K$ dengan ketentuan $2 < L < \frac{N}{2}$ dan $K = N - L + 1$. Data X akan dipetakan menjadi *lag vector* $X_i = (x_i, \ldots, x_i + L - 1)^T$ untuk $i = 1,2, \ldots, K$. *Lag vector* X_i akan dibentuk matriks lintasan LxK.

$$\mathbf{X} = [X_1 : \ldots : X_K] = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_k \\ x_2 & x_3 & x_4 & \cdots & x_{K+1} \\ x_3 & x_4 & x_5 & \cdots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \cdots & x_N \end{bmatrix} \quad (2.1)$$

Matriks X tersebut disebut juga matriks Hankel yang merupakan semua elemen anti diagonalnya bernilai sama

2.4.1.2 *Singular Value Decomposition* (SVD)

Tahap selanjutnya dalam dekomposisi setelah *Embedding* adalah tahap *Singular Value Decomposition* (SVD). Untuk penentuan matriks singular dalam SSA dapat didefinisikan dengan $S = XX^T$. Jika $\lambda_1 \ldots \lambda_L$ merupakan nilai *eigen* dari matriks dimana $\lambda_1 \geq \cdots . \geq \lambda_L \geq 0$ dan $U_1 \ldots U_L$ merupakan *vector eigen* dari masing-masing nilai *eigen* serta *principal component* dinotasikan dengan $V_i = X^T U_i / \sqrt{\lambda_i}$ (i = 1,..., d) maka matriks lintasan X diperoleh sebagai berikut :

$$\mathbf{X} = X_1 + \ldots + X_d \text{ dimana : } X_i = \sqrt{\lambda_i} U_i {V_i}^T \quad (2.2)$$

$\sqrt{\lambda_i} U_i {V_i}^T$ disebut dengan eigentriple. Maka yang menjadi output dalam tahapan ini adalah *eigentriple* $\sqrt{\lambda_i} U_i {V_i}^T$ yaitu *eigen value, eigen vector,* dan *principal component*

2.4.2 Rekontruksi

Pada tahapan ini ada dua tahap yang harus dilakukan yaitu tahap *Grouping* dan juga *Diagonal Averaging*. Perameter yang mempunyai peran penting dalam tahapan rekontruksi adalah *grouping effect* (r) yang berfungsi untuk menentukan pola plot data. Hasil dari tahapan ini akan mendekatkan hasil peramalan dengan data asli maka dengan pengelompokan yang tepat serta dengan MAPE nilai ramal data asli dapat menghasilkan peramalan yang baik.

2.4.2.1 *Groupping*

Grouping merupakan tahapan penguraian matriks lintasan $L\ x\ K$ menjadi beberapa kelompok pola yang meliputi pola *trend*, pola musiman dan pola *noise*. Tahapan ini juga merupakan partisi dari indeks $\{I_1, \ldots, I_m\}$. Maka X akan bekorespondensi dengan kelompok $I = \{I_1, \ldots, I_m\}$ dengan persamaan sebagai berikut:

$$\mathbf{X} = X_{I1} + \ldots + X_{Im} \tag{2.3}$$

2.4.2.2 *Diagonal Averaging*

Dalam langkah ini matriks X_{Ij} yang diperoleh dari grouping disusun ulang menjadi data deret baru dengan panjang N. jika Y adalah matriks dengan ukuran LxK dengan elemen y_{Ij}, dimana $1 \leq i \ \leq L$ dan $1 \leq i \ \leq K$.misalkan L= min (L,K) dan K*= max (L,K), N=L+K-1. Dan misalkan juga $y^*_{Ij} = y_{Ij}$ jika L<K dan $y^*_{Ij} = y_{jI}$ selainnya. Menggunakan metode *Diagonal Averaging* Matriks Y

ditransformasikan ke bentuk series kembali, {y_1, …,y_N) dengan persamaan sebagai berikut :

$$y_k = \begin{cases} \frac{1}{k}\sum_{m=1}^{k} y^*_{m,k-m+1} & untuk\ 1 \le k \le L^* \\ \frac{1}{L^*}\sum_{m=1}^{L^*} y^*_{m,k-m+1} & untuk\ L^* \le k \le K^* \\ \frac{1}{N-k+1}\sum_{m=k-K^*+1}^{N-K^*+1} y^*_{m,k-m+1} & ntuk\ K^* \le k \le N \end{cases} \quad (2.4)$$

Sehingga X_{lj} akan menjadi deret $\widetilde{Y}^{(k)} = \widetilde{y_1}^{(k)}, \ldots, \widetilde{y_N}^{(k)}$ oleh sebab itu, deret asli akan menjadi jumlah dari m deret seperti berikut :

$$y_n = \sum_{k=1}^{m} \widetilde{y_n}^{(k)} \qquad (n = 1,2, \ldots N) \quad (2.5)$$

2.5 *Recurrent Forecasting*

Recurrent Forecasting merupakan teknik peramalan SSA yang paling umum digunakan. Sifat utama pada penguraian SSA adalah seri asli dapat memenuhi persamaan *Linier Recurrent Formula* (LRF). Berikut ini merupakan persamaan LRF :

$$f_n = a_1 f_{n-1} + \cdots + a_d f_{n-d} \quad (2.6)$$

Dimana a_i merupakan koefisien LRF dengan $i = 1,2, \ldots, d$

Jika seri asli f_n memenuhi persamaan 2.6 maka seri f_n merupakan penjumlahan dari komponen eksponensial, polynomial dan harmonic dalam jangka waktu yang singkat. Hal ini menyebabkan peramalan diwaktu yang akan datang dapat dilakukan menggunakan nilai LRF yang sama karena efek continuation yang diberikan telah sesuai untuk jangka waktu tidak panjang.

Algoritma *Recurrent Forecasting* adalah sebagai berikut :

1. Time Series Y_{N+M}= (y_1, ..., y_{N+M}) didefinisikan oleh

$$\mathrm{y_i} = \begin{cases} \overline{x}_i & \text{for } i=1,\dots,N \\ \sum_{j=1}^{L-1} a_j y_{i-j} & \text{for i=N+1,...,N+M} \end{cases} \quad (2.7)$$

2. Angka Y_{N+1}, ..., y_{N+M} akan membuat suatu istilah *M* dari *Recurrent Forecasting*.

   Recurrent Forecasting dilakukan dengan LRR menggunakan koefisien {a_j, j = 1, ..., *L-1*}.

   Berikut ini rumus yang mendefinisikan operator linear P_{Rec} : $R^L R^L$

$$P_{Rec}\, Y = \begin{pmatrix} \overline{Y} \\ R^T \overline{Y} \end{pmatrix} \quad (2.8)$$

$$\text{Dimana, } Z_i = \begin{cases} \widehat{X}_i & \text{for } i = 1, \dots, K \\ P_{Rec} Z_{i-1} & \text{for } i = K+1, \dots, K+M \end{cases} \quad (2.9)$$

   Disini matriks Z = [Z_1 : ... : Z_{K+M}] merupakan matriks lintasan dari Y_{N+M}. Oleh karena itu (2.9) dianggap bentuk vektor dari (2.8).

2.6 *Bootstrap-SSA Forecasting*

Bootstrap merupakan suatu proses pengambilan sampel (*resampling*) dengan pengembalian sampel hasil dari pengamatan menggunakan replikasi. Metode *Bootstrap* merupakan sebuah metode pendekatan non parametrik yang pertama kali diperkenalkan oleh Bradley Efron pada tahun 1979. Metode ini dgunakan untuk menaksir kuantitas statistik seperti rata-rata, *standar error*, dan bias suatu estimator atau untuk membentuk interval konfidensi. Resampling mempunyai 2 metode pendekatan yaitu pendekaatan *monte carlo* dan *boostrap*. Hanif (2017) menyebutkan bahwa simulasi *monte carlo* digunakan saat kondisi ideal yaitu model dari sinyal dan noise diketahui.karena kondisi tersebut jarang ditemui pada permalan kuantitatif, maka metode *bootstrap* yang lebih dominan digunakan pada peramalan kuantitatif yang membutuhkan teknik resampling.

Menurut Rahmani (2014) penerapan teknik *boostrap* menghasilkan nilai peramalan disertai batas atas dan bawah dari nilai peramalan. Selang kepercayaan yang terbentuk dapat memberikan informasi untuk mengambil keputusan. Teknik *Bootstrap* ini dapat diterapkan pada teknik peramalan SSA seperti teknik *bootstrap-recurrent forecasting* ketika diterapkan pada teknik peramalan *R-Forecasting* dan teknik *bootstrap-vector forecasting* ketika diterapkan pada teknik peramalan *V-Forecasting*

2.7 *Bootstrap Recurrent Forecasting*

Bootstrap-Recurrent Forecasting adalah penerapan teknik *bootstrap* pada SSA *R-Forecasting*. Golyandina, dkk. (2001) menyebutkan bahwa algoritma pada peramalan *R-Forcasting* erat kaitannya dengan penaksiran LRF. Penaksiran koefisien LRF yaitu a_1, ..., a_d menggunakan vektor yang diperoleh dari proses SVD. Koefisien LRF dapat dihitung dengan rumus :

$$A = (a_{L-1}, a_{L-2}, \dots, a_2, a_1)^T = \frac{1}{1-v^2}\sum_{i=1}^{r} \pi_i U_t^{\overline{V}} \quad (2.10)$$

Dengan, $$v^2 = \sum_{i=1}^{r} \pi_i^2 \quad (2.11)$$

$\pi_i = u_L$: komponen terakhir dari *eigen vector* **U**

$U^{\overline{V}} = (u_1, u_2, \dots, u_{L-1})^T$: *Eigen vector* **U** tanpa komponen terakhir.

Deret waktu yang digunakan pada *R-Forecasting* merupakan hasil rekonstruksi yang diperoleh dari *diagonal averaging* yang selanjutnya ditentukan M buah titik baru untuk diramalkan. Hasil peramalan didapatkan berdasarkan rumus berikut :

$$\tilde{F}_n \begin{cases} \bar{x}_i & untuk\ \ i=1,\dots,N \\ \sum_{j=1}^{L-1} a_j y_{i-j} & \text{untuk } \ i=N+1,\dots,N+M \end{cases} \tag{2.12}$$

2.8 Metode ESPRIT

Diberikan deret waktu $X_N = \{x_i\}_{i=1}^{N}$ dengan $y_i = s_i + p_i$, yang mana $S_N = \{s_i\}_{i=1}^{N}$ merupakan deret waktu yang diatur oleh *linier recurrence relation* (LRR) orde r (S_N yaitu sinyal) dan $P_N = \{p_i\}_{i=1}^{N}$ yaitu suatu residual (*noise*). Pada perkiraan keterpisahan sinyal dan residual, ada satu himpunan I dari banyak vektor eigen yang bersesuaian dengan sinyal. Apabila sinyal lebih dominan, maka $I = \{1, \dots, r\}$ dan subruang $\mathfrak{L}_r = \mathrm{span}\{\mathrm{U}_1, \dots, \mathrm{U}_r\}$ dapat dianggap estimasi dari sub ruang sinyal yang sebenarnya. Maka $\mathrm{Y} = \mathrm{U}_r = [\mathrm{U}_1, \dots, \mathrm{U}_r]$ dapat digunakan sebagai suatu estimasi dari Y.

Metode ESPRIT ini terjadi dalam estimasi akar sinyal sebagai nilai eigen dari suatu matriks D, yang mana $\underline{\mathrm{U}}\mathrm{D} \approx \overline{\mathrm{U}}_r$. Dengan mengestimasi akar sinyal, metode ESPRIT menyediakan estimasi dari parameter sinyal. Metode ESPRIT juga mampu mengestimasi parameter dari komponen deret waktu yang dapat dipisahkan apabila matriks U_r terdiri dari vektor eigen yang sesuai. Ada dua jenis metode ESPRIT yaitu yang pertama *least squares* (LS-ESPRIT) dan yang kedua *total least squares* (TLS-ESPRIT). Estimasi LS-ESPRIT dari matriks D merupakan $D = \underline{U}_r^{\dagger}\overline{U}_r = \left(\underline{U}_r^T\underline{U}_r\right)^{-1}\underline{U}_r^T\overline{U}_r$. Nilai eigen dari D tidak bergantung pada pilihan basis subruang $\mathfrak{L}_r = \mathrm{span}\{U_1, \dots, U_r\}$.

2.9 Akurasi Model Peramalan

Dalam mengukur rata-rata kesalahan peramalan yang paling sering digunakan adalah Mean Absolute Percentage Error (MAPE). Ukuran rata-rata kesalahan tersebut dinyatakan sebagai presentase dari nilai observasi yang relevan terlepas dari kesalahan peramalan yang positif yaitu melebihi perkiraan ataupun kesalahan peramalan negative yaitu dibawah perkiraan. Rumus persamaan MAPE dapat dilihat sebagai berikut :

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{|e_t|}{d_t} \qquad (2.13)$$

Ketentuan nilai MAPE peramalan adalah sebagai berikut :

Tabel 2.1 Ketentuan nilai MAPE

MAPE	Keterangan
< 10%	Sangat Baik
< 20%	Baik
< 30%	Cukup Baik
> 40%	Tidak Akurat

Sumber : (Lewis, 1997)