

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Variabel Penelitian**

Penelitian ini menggunakan satu definisi variabel operasional yaitu rata-rata temperatur bumi periode tahun 1880 sampai dengan tahun 2012.

#### **3.2 Jenis dan Sumber Data**

Adapun data yang dipakai dalam penelitian ini adalah data sekunder yang diunduh dari sebuah situs yang beralamatkan [http://www.earth-policy.org/datacenter/xls/indicator8\\_2013\\_1.xlsx](http://www.earth-policy.org/datacenter/xls/indicator8_2013_1.xlsx) yang diakses pada tanggal 26 Agustus 2013. Data yang tersedia merupakan data rata-rata temperatur bumi dari tahun 1880-2012 (133 tahun, 133 data). Data rata-rata temperatur bumi lengkap dapat dilihat pada lampiran 1.

#### **3.3 Metode Pengumpulan Data**

Metode pengumpulan data yang digunakan dalam penelitian ini adalah *non-participant observer*, di mana peneliti hanya mengamati data yang sudah tersedia tanpa ikut menjadi bagian dari suatu sistem data.

#### **3.4 Metode Analisis**

##### **3.4.1 Metode ARIMA (Box-Jenkins)**

Metode ARIMA menggunakan pendekatan iteratif dalam mengidentifikasi suatu model yang paling tepat dari berbagai model yang ada. Model sementara

yang telah dipilih diuji lagi dengan data historis untuk melihat apakah model sementara yang terbentuk tersebut sudah memadai atau belum.

### 3.4.1.1 Stasioneritas Data

Data yang tidak stasioner memiliki rata-rata dan varian yang tidak konstan sepanjang waktu. Dengan kata lain, secara ekstrim data stasioner adalah data yang tidak mengalami kenaikan dan penurunan. Selanjutnya regresi yang menggunakan data yang tidak stasioner biasanya mengarah kepada regresi lancung. Permasalahan ini muncul diakibatkan oleh variabel (dependen dan independen) runtun waktu terdapat tren yang kuat (dengan pergerakan yang menurun maupun meningkat). Adanya tren akan menghasilkan nilai  $R^2$  yang tinggi, tetapi keterkaitan antar variabel akan rendah.

Model ARIMA mengasumsikan bahwa data masukan harus stasioner. Apabila data masukan tidak stasioner perlu dilakukan penyesuaian untuk menghasilkan data yang stasioner. Salah satu cara yang umum dipakai adalah metode pembedaan (*differencing*). Metode ini dilakukan dengan cara mengurangi nilai data pada suatu periode dengan nilai data periode sebelumnya.

### 3.4.1.2 Tahapan Metode ARIMA (Box-Jenkins)

Langkah-langkah dalam pembentukan model dalam metode ARIMA secara iteratif adalah sebagai berikut:

#### 1. Identifikasi Model

Identifikasi model bertujuan untuk menentukan (mengidentifikasi) model yang merupakan representasi data runtun waktu  $Z_1, Z_2, \dots, Z_n$ . Adapun langkah-langkah yang dilakukan adalah sebagai berikut.

- a. Menentukan mean dan variansi data runtun waktu.
- b. Menentukan FAK beserta  $2SE(\rho_k)$  dari data runtun waktu.
- c. Menentukan FAKP beserta  $2SE(\phi_{kk})$  dari data runtun waktu.
- d. Membandingkan FAK dan FAKP data runtun waktu dengan FAK dan FAKP teoretik.

Sebelum pemodelan dilakukan, hal berikut adalah mutlak diperlukan.

- a. Plot data untuk melihat kestasioneran data.
- b. Grafik dari distribusi frekuensi untuk melihat asumsi normalitas.
- c. Informasi lain (kemiringan, keruncingan, dll).

Jika  $E(\bar{Z}_t) = \bar{Z}, \bar{Z} \neq 0$ , maka model dituliskan sebagai  $\hat{Z}_t = \bar{Z}_t - \bar{Z}$  sehingga perlu diuji apakah  $\bar{Z} = 0$ . Hipotesis yang harus diuji adalah

$$H_0 : \bar{Z} = 0$$

$$H_1 : \bar{Z} \neq 0$$

Jika  $|\bar{Z}| < 2SE(\bar{Z})$ , maka  $H_0$  diterima ( $\bar{Z}$  tidak berbeda secara signifikan dengan nol).

Seperti yang dijelaskan sebelumnya bahwa model ARIMA hanya dapat diterapkan untuk runtun waktu yang stasioner. Oleh karena itu, pertama kali yang harus dilakukan adalah menyelidiki apakah data yang kita gunakan sudah stasioner atau belum. Jika data tidak stasioner, yang perlu dilakukan adalah memeriksa pada *differencing* beberapa data akan stasioner, yang menentukan berapa nilai  $d$ . Proses ini dapat dilakukan dengan menggunakan koefisien FAK (fungsi auto korelasi). Jika data sudah stasioner sehingga tidak dilakukan *differencing* terhadap data runtun waktu maka  $d$  diberi nilai 0.

Di samping menentukan  $d$ , pada tahap ini juga ditentukan berapa jumlah nilai lag residual ( $q$ ) dan nilai lag dependen ( $p$ ) yang digunakan dalam model. Alat utama yang digunakan untuk mengidentifikasi  $q$  dan  $p$  adalah FAK dan FAKP (fungsi auto korelasi parsial), dan *correlogram* yang menunjukkan plot nilai FAK dan FAKP terhadap lag.

## 2. Estimasi Parameter

Setelah beberapa model diidentifikasi, langkah selanjutnya adalah mengestimasi parameter yang ada pada model. Estimasi yang efisien yaitu estimasi yang meminimumkan kuadrat selisih antara nilai estimasi dengan nilai parameter sebenarnya. Untuk data yang cukup banyak, estimasi yang efisien adalah estimasi yang memaksimumkan fungsi Likelihood.

Diperlukan taksiran interval untuk estimasi parameter. Di sini perlu diuji apakah  $\hat{\theta}$  atau  $\hat{\phi}$  berbeda secara signifikan dengan nol atau tidak. Jika

$\hat{\theta} < 2SE(\hat{\theta})$ , maka  $\hat{\theta}$  tidak berbeda secara signifikan dengan nol. Begitu pula jika  $\hat{\phi} < 2SE(\hat{\phi})$ , maka  $\hat{\phi}$  tidak berbeda secara signifikan dengan nol.

### 3. Verifikasi Model

Verifikasi adalah pemeriksaan apakah model yang diestimasi cukup cocok dengan data yang ada. Jika terjadi penyimpangan yang cukup serius, maka model yang baru harus dirumuskan kembali. Langkah-langkah yang harus dilakukan pada tahap verifikasi ini adalah sebagai berikut.

#### a. Uji Keberartian Koefisien ( $\theta$ atau $\phi$ )

Hipotesis yang harus diuji adalah

$H_0$  : koefisien tidak berbeda secara signifikan dengan nol.

$H_1$  : koefisien berbeda secara signifikan dengan nol.

Adapun kriteria untuk uji keberartian koefisien adalah sebagai berikut.

- Tolak  $H_0$  jika  $|koef| > 2SE(koef)$  atau
- Tolak  $H_0$  jika  $P. Value < \alpha = 5\%$ .

#### b. Nilai Variansi Sesatan

Pilih model yang mempunyai variansi sesatan terkecil. Nilai variansi sesatan bisa langsung dilihat dari *output* Minitab 14 atau dihitung dengan menggunakan rumus  $\sigma_a^2 = \frac{SS-MS}{DF}$ , di mana

SS : Kuadrat jumlah (*Sum Square*)

MS : Kuadrat Rata-rata (*Mean Square*)

DF : Derajat Kebebasan (*Degree of Freedom*)

#### c. Uji Kecocokan (*lack of fit*)

Hipotesis yang harus diuji adalah

$H_0$  : model sesuai

$H_1$  : model tidak sesuai

Adapun kriteria untuk uji kecocokan adalah sebagai berikut.

- Tolak  $H_0$  jika  $\chi_{hit}^2 > \chi_{tabel}^2$  atau
- Tolak  $H_0$  jika  $P. Value < \alpha = 5\%$ .

Hal yang harus diperhatikan dalam tahap verifikasi adalah penggunaan prinsip *parsimony* terhadap model yang sedang diuji.

#### 4. Peramalan

Langkah terakhir dari pemodelan data runtun waktu adalah menentukan ramalan data-data yang belum terjadi berdasarkan pada data di masa lalu. Ramalan yang digunakan data runtun waktu adalah Ramalan Harapan Bersyarat yang memiliki sifat yang baik yaitu memiliki sesatan kuadrat rata-rata minimum, artinya jika terdapat nilai ramalan yang lain maka nilai ramalan tebakan memiliki sesatan yang kuadratnya mempunyai nilai harapan yang lebih besar. Pada dasarnya peramalan model runtun waktu seperti ini lebih cocok untuk peramalan dengan jangkauan sangat pendek.

##### 3.4.2 Metode Jaringan Syaraf Tiruan (JST) *Backpropagation*

Kusumadewi (2004) menjelaskan, *backpropagation* (propagasi balik) menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap *feedforward propagation* (propagasi maju) harus dikerjakan terlebih dulu.

*Input* yang digunakan dalam pelatihan ini adalah nilai rata-rata temperatur bumi periode tahun 1880 sampai dengan tahun 2012.

##### 3.4.2.1 Pelatihan Standar *Backpropagation*

Pelatihan Jaringan Syaraf Tiruan Propagasi Balik meliputi 3 fase. Ketiga fase tersebut dapat dijelaskan sebagai berikut (Siang, 2005) :

##### **Fase pertama : propagasi maju**

Selama propagasi maju, sinyal masukan ( $=x_i$ ) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit lapisan tersembunyi ( $=z_j$ ) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ( $=y_k$ ). Berikutnya, keluaran jaringan ( $=y_k$ ) dibandingkan dengan target yang harus dicapai ( $=t_k$ ).

Selisih  $t_k - y_k$  adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

### **Fase Kedua : Propagasi Mundur**

Berdasarkan kesalahan  $t_k - y_k$ , dihitung faktor  $\delta_k (k=1,2,\dots,m)$  yang dipakai untuk mendistribusikan kesalahan di unit  $y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor  $\delta_k$  di setiap unit di lapisan tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi pada lapisan di bawahnya. Demikian seterusnya hingga semua faktor  $\delta$  di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

### **Fase Ketiga : Perubahan Bobot**

Setelah semua faktor  $\delta$  dihitung, bobot semua garis dimodifikasi bersamanaan. Perubahan bobot suatu garis didasarkan atas faktor  $\delta$  neuron di lapisan atasnya. Sebagai contoh, perubahan garis yang menuju ke layar keluaran didasarkan atas  $\delta_k$  yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diizinkan.

Algoritma pelatihan untuk jaringan dengan satu layar tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut :

- a. **Langkah 0** : Inisialisasi bobot (ambil bobot awal dengan nilai *random* yang cukup kecil)

- b. **Langkah 1** : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9
- c. **Langkah 2** : Untuk setiap pasang data pelatihan lakukan langkah 3-8

Fase I : Propagasi maju (*feedforward propagation*)

- d. **Langkah 3** : Tiap unit masukan ( $x_i$ ,  $i=1,2,3,\dots,n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- e. **Langkah 4** : Tiap-tiap unit tersembunyi  $z_j$  ( $j = 1,2,\dots,p$ ) menjumlahkan sinyal-sinyal *input* terbobot:

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ij}$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*nya:

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

- f. **Langkah 5** : Tiap-tiap unit *output*  $y_k$  ( $k = 1,2,\dots,m$ ) menjumlahkan sinyal-sinyal *input* terbobot

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{jk}$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*nya:

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

Fase II : Propagasi mundur (*backpropagation*)

- g. **Langkah 6** : Tiap-tiap unit *output*  $y_k$  ( $k = 1,2,\dots,m$ ) menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi *error*nya

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k)$$

$\delta_k$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar di bawahnya (langkah 7)

Kemudian hitung suku perubahan bobot atau koreksi bobot (yang akan dipakai nanti untuk merubah bobot  $w_{jk}$ ) dengan laju percepatan  $\alpha$

$$\Delta w_{kj} = \alpha \delta_k z_j ; \quad k = 1, 2, \dots, m ; j = 0, 1, \dots, p$$

Hitung juga koreksi bias (yang akan dipakai nanti untuk merubah bobot  $w_{0k}$ )

$$\Delta w_{0k} = \alpha \delta_k$$

Kirimkan  $\delta_k$  ini ke unit-unit yang ada di lapisan bawahnya.

- h. **Langkah 7** : Tiap-tiap unit tersembunyi  $z_j$  ( $j = 1, 2, \dots, p$ ) menjumlahkan delta *inputnya* (dari unit-unit yang berada pada lapisan di atasnya)

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{jk}$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*

$$\delta_j = \delta_{net_j} f'(z_{net_k}) = \delta_{net_j} = \delta_{net_j} z_j (1 - z_j)$$

Kemudian hitung suku perubahan bobot atau koreksi bobot (yang akan dipakai nanti untuk merubah bobot  $v_{ij}$ )

$$\Delta v_{ji} = \alpha \delta_j x_i ; \quad j = 1, 2, \dots, p ; i = 0, 1, \dots, n$$

Hitung juga koreksi bias (yang akan dipakai nanti untuk merubah bobot  $v_{0j}$ )

$$\Delta v_{0j} = \alpha \delta_j.$$

### Fase III : Perubahan Bobot

- i. **Langkah 8** : Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit *output*  $y_k$  ( $k = 1, 2, \dots, m$ ):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} ; (j = 0, 1, \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi  $z_j$  ( $j = 1, 2, \dots, p$ ):



$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad ;(i = 0,1,\dots,n)$$

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

### 3.4.3 Pengukuran Kinerja

#### 3.4.3.1 Mean Absolute Percentage Error (MAPE)

Kriteria keakuratan ramalan menggunakan kedua metode tersebut ditentukan dengan menghitung nilai *Mean Absolute Percentage Error* (MAPE). Digunakan MAPE karena pada data pergerakan rata-rata temperatur bumi ukuran variabel peramalan merupakan faktor penting dalam mengevaluasi akurasi peramalan. Sehingga MAPE digunakan untuk menilai prestasi jaringan yang dilatih karena MAPE mengenal secara pasti signifikansi hubungan di antara data ramalan dengan data aktual melalui persentase dari data aktual serta indikator positif atau negatif pada galat (*error*) diabaikan. MAPE memberikan petunjuk seberapa besar kesalahan peramalan dibandingkan dengan nilai sebenarnya dari series tersebut. didapat dari persamaan di bawah ini :

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

dengan,

$y_t$  = nilai aktual pada waktu  $t$

$\hat{y}_t$  = nilai ramalan pada waktu  $t$

$n$  = jumlah ramalan.

#### 3.4.3.2 Komparasi Hasil Peramalan

Setelah nilai *Mean Absolute Percentage Error* (MAPE) dari kedua metode didapatkan, maka akan dilakukan komparasi terhadap nilai MAPE yang didapatkan pada periode *testing* (*out-sample*)

- Jika nilai  $MAPE_{ARIMA} < MAPE_{JST}$  maka metode ARIMA memiliki performa lebih baik dibandingkan metode JST *Backpropagation* karena

Oksendi Vitra Sihombing, 2013

Perbandingan Metode ARIMA (Box-Jenkins) Dengan Jaringan Saraf Tiruan (JST) Back Propagation Sebagai Metode Pemahaman Rata-Rata Temperatur Bumi

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

memiliki tingkat kesalahan yang dihasilkan oleh ARIMA relatif lebih kecil.

- Sebaliknya, jika  $MAPE_{ARIMA} > MAPE_{JST}$  maka metode ARIMA memiliki performa lebih buruk dibandingkan metode JST *Backpropagation* karena tingkat kesalahan yang dihasilkan oleh metode ARIMA relatif lebih besar.

