

BAB II

TINJAUAN PUSTAKA

2.1 Uang Beredar

Pada awalnya fungsi uang hanya digunakan sebagai alat tukar saja, namun seiring dengan berkembangnya zaman, fungsi uang diperluas hingga dapat digunakan sebagai alat penyimpanan nilai (*store of value*), satuan hitung (*unit of account*) dan ukuran pembayaran yang tertunda (*standard for deferred payment*). Sukirno (Rosyidah dkk, 2017) mengatakan bahwa uang beredar adalah jumlah uang yang tersedia dalam perekonomian dan dapat digunakan untuk membiayai transaksi-transaksi yang dilakukan dalam masyarakat. Uang beredar juga didefinisikan sebagai kewajiban sistem moneter (Bank Sentral, Bank Umum, dan Bank Perkreditan Rakyat/BPR) pada sektor swasta domestik (tidak termasuk pemerintah pusat dan bukan penduduk).

Jika dilihat secara sempit uang beredar (M1) terdiri atas uang kartal dan uang giro. Uang kartal merupakan uang kertas dan uang logam yang beredar di masyarakat yang dikeluarkan dan diedarkan oleh Bank Indonesia yang berfungsi sebagai otoritas moneter. Bagi masyarakat luas, jenis uang ini lebih sering dikenal dengan sebutan uang tunai, yaitu uang yang ada di tangan masyarakat (di luar bank umum) dan siap dibelanjakan setiap saat, terutama untuk pembayaran-pembayaran dalam jumlah yang tidak terlalu besar. Sedangkan uang giral merupakan uang dalam rekening giro di bank umum yang penarikannya dapat dilakukan sewaktu-waktu hanya dengan menuliskan jumlah uang yang diinginkan pada selebar cek.

2.2 Analisis *Multivariate Time Series*

Muhtaram (Maghfiroh, 2018) menjelaskan bahwa analisis *time series* adalah salah satu metode peramalan secara matematis dengan menggunakan waktu sebagai acuan, kemudian membuat prediksi dengan menggunakan ekstrapolasi berdasarkan waktu untuk pola-pola tersebut. Tsay (Nabila, 2016) mengatakan bahwa beberapa kasus penelitian yang telah berkembang memerlukan suatu analisis *time series* yang mempertimbangkan berbagai *time series* secara simultan, analisis ini disebut dengan analisis *multivariate time series*. Analisis *multivariate time series* digunakan untuk menganalisis beberapa variabel yang memiliki keterkaitan dengan variabel lainnya yang saling berkaitan.

2.3 Stasioneritas Data

Salah satu prosedur yang dilakukan dalam analisis *time series* adalah dengan melakukan pengujian apakah data tersebut stasioner atau tidak. Sebuah deret dikatakan stasioner, jika seluruh moment dari deret tersebut (nilai tengah, varians dan kovarians) konstan sepanjang periode tertentu. Stasioner juga didefinisikan sebagai data yang tidak mengalami pertumbuhan atau penurunan secara signifikan, atau rata-rata dan varians setiap lag bernilai konstan untuk setiap waktu. Terdapat dua jenis stasioneritas data yaitu stasioner dalam *mean* dan stasioner dalam varians.

2.3.1 Stasioner dalam *mean*

Data *time series* akan dikatakan stasioner dalam *mean* jika $E(Y_t) = \mu$, hal ini mengartikan bahwa nilai *mean* konstan terhadap waktu. Data yang tidak stasioner dalam *mean* artinya $E(Y_t)$ dipengaruhi oleh waktu pengamatan yaitu $E(Y_t) = \mu_t$. Pengujian stasioneritas dalam *mean* dapat dilakukan dengan uji ADF (*Augmented Dickey-Fuller*) dan plot MCCF (*Matrix Cross Correlation Function*). Dikatakan stasioner jika nilai p-value pada uji ADF $< \alpha$ (5%) atau plot MCCF menunjukkan sedikitnya tanda + dan - . Jika data tidak stasioner dalam *mean*, maka dapat dilakukan pembedaan (*differencing*). *Differencing data time series* dapat dirumuskan seperti berikut:

$$W_t = Y_t - Y_{t-1} \quad (2.1)$$

dimana:

W_t = *differencing* orde ke-t

t = indeks waktu

Y_t = data pengamatan ke-t

Y_{t-1} = data pengamatan ke-(t-1)

2.3.2 Stasioner dalam varians

Data *time series* yang tidak stasioner dalam varians akan berubah sejalan dengan perubahan level $\text{var}(Y_t) = \sigma_t = cf(\mu_t)$. Pengujian stasioner terhadap varians dilakukan dengan uji *Box-cox*, dikatakan stasioner jika nilai *rounded-value* yang dihasilkan adalah 1. Untuk mengatasi data yang tidak stasioner dalam

varians dapat dilakukan dengan transformasi. Pada umumnya untuk menstabilkan varians, digunakan transformasi *Box-Cox* (Wei, 2006):

$$T(Y_t) = \frac{Y_t^\lambda - 1}{\lambda} \quad (2.2)$$

dimana:

$T(Y_t)$ = transformasi data ke-t

t = indeks waktu

λ = Nilai koefisien dari transformasi *Box-Cox*

Terdapat pula tabel kesetaraan transformasi, tabel tersebut merupakan tabel yang digunakan sebagai acuan untuk melakukan transformasi jika data tidak stasioner terhadap varians.

Tabel 2.1 Kesetaraan transformasi

Nilai λ	$T(X_t) =$
-1,0	$\frac{1}{X_t}$
-0,5	$\frac{1}{\sqrt{X_t}}$
0,0	$\text{Ln}(X_t)$
0,5	$\sqrt{X_t}$
1,0	X_t

2.4 Identifikasi Model

Identifikasi model dilakukan dengan melihat visualisasi plot dari MCCF dan MPCCF setelah data stasioner. Hal ini bertujuan untuk mengetahui model yang sesuai untuk diterapkan pada data yang akan dianalisis.

2.4.1 Matrix Cross Correlation Function (MCCF)

Matriks korelasi silang yang terbentuk pada sebuah vektor *time series* dengan observasi sebanyak n ($\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_n$) adalah sebagai berikut, (Wei, 2006)

$$\hat{\rho}(k) = [\hat{\rho}_{ij}(k)] \quad (2.3)$$

dengan $\hat{\rho}_{ij}(k)$ adalah persamaan matriks korelasi silang sampel pada komponen *series* ke- i dan ke- j yang dinyatakan dalam persamaan sebagai berikut:

$$\hat{\rho}_{ij}(k) = \frac{\sum_{t=1}^{n-k} (Y_{i,t} - \bar{Y}_i)(Y_{j,t+k} - \bar{Y}_j)}{\sqrt{(\sum_{t=1}^n (Y_{i,t} - \bar{Y}_i)^2 \sum_{t=1}^n (Y_{j,t+k} - \bar{Y}_j)^2)}} \quad (2.4)$$

dengan \bar{Y}_i dan \bar{Y}_j adalah nilai rata-rata sampel dari komponen *series* yang bersesuaian. Sebagai catatan, $\gamma_{ij}(k) \neq \gamma_{ij}(-k)$ untuk $i \neq j$ sehingga $\Gamma(k) \neq \Gamma(-k)$.

karena $\gamma_{ij}(k) = E[(Y_{i,t} - \bar{Y}_i)(Y_{j,t+k} - \bar{Y}_j)] = E[(Y_{j,t+k} - \bar{Y}_j)(Y_{i,t} - \bar{Y}_i)] = \gamma_{ij}(-k)$ didapatkan $\Gamma(k) \neq \Gamma(-k)$ dan $\rho(k) = \rho'(k)$. Bartlett dalam Wei (2006) melakukan penurunan varians dan kovarians dari besaran korelasi silang yang didapat dari sampel. Berdasarkan hipotesis yang mengatakan bahwa dua data *time series* Y_1 dan Y_j tidak berkorelasi, maka Bartlett menunjukkan persamaan sebagai berikut,

$$\text{Varians}[\hat{\rho}_{ij}(k)] \cong \frac{1}{n-k} [1 + 2 \sum_{s=1}^{\infty} \rho_{ii}(s)\rho_{jj}(s)], |k| > q \quad (2.5)$$

Ketika Y_i dan Y_j adalah deret yang bersifat *white noise*, maka selanjutnya akan diperoleh persamaan berikut:

$$\text{Cov}[\hat{p}_{ij}(k), \hat{p}_{ij}(k + s)] \cong \frac{1}{n-k} \quad (2.6)$$

$$\text{Var}[\hat{p}_{ij}(k)] \cong \frac{1}{n-k} \quad (2.7)$$

Untuk ukuran sampel yang besar, $(n-k)$ dalam persamaan di atas seringkali diganti menjadi n .

Matriks korelasi silang sampel dapat digunakan dalam mengidentifikasi orde model *moving average* (MA). Box dan Tiao (1981) dalam Wei (2006) memperkenalkan metode yang dapat meringkas hasil korelasi silang sampel. Metode ini memanfaatkan simbol (+), (-), dan (.) pada baris ke- i dan kolom ke- j pada matriks korelasi silang sampel, yaitu:

1. simbol (+) mengartikan bahwa nilai $\hat{p}_{ij}(k)$ mempunyai nilai yang lebih besar dari 2 kali nilai estimasi *standard error* (SE), hal ini menunjukkan adanya korelasi positif pada komponen (i,j)
2. simbol (-) mengartikan bahwa nilai $\hat{p}_{ij}(k)$ mempunyai nilai yang lebih kecil dari -2 kali nilai estimasi *standard error* (SE), hal ini menunjukkan adanya korelasi negatif pada komponen (i,j)
3. simbol (.) mengartikan bahwa nilai $\hat{p}_{ij}(k)$ terletak diantara ± 2 kali nilai estimasi *standard error* (SE), hal ini menunjukkan tidak adanya korelasi pada komponen (i,j) .

2.4.2 Matrix Partial Cross Correlation Function (MPCCF)

Partial autocorrelation function (PACF) digunakan dalam *time series* univariat untuk menentukan orde dalam model AR. Generalisasi dari konsep PACF pada bentuk vektor *time series* dilakukan oleh Tiao dan Box (1981) dalam Wei (2006), yang mengartikan bahwa matriks autoregresi parsial pada lag s dengan notasi $P(s)$, digunakan sebagai koefisien matriks terakhir ketika data diterapkan ke dalam *Vector Autoregressive* (VAR) orde s . $P(s)$ sama dengan $\Phi_{s,s}$ dalam regresi linier multivariat. Persamaan matriks autoregresi parsial diperoleh (Wei, 2006)

$$P(s) = \begin{cases} \Gamma'(1)[\Gamma'(0)]^{-1}, & s = 1 \\ \{\Gamma'(s) - c'(s)[A(s)]^{-1}b(s)\}[\Gamma'(0) - b'(s)[A(s)]^{-1}b(s)]^{-1}, & s > 1 \end{cases} \quad (2.8)$$

Jika model dari data diperoleh vector AR(p), maka:

$$P(s) = \begin{cases} \Phi_p, & s = p \\ 0, & s > p \end{cases} \quad (2.9)$$

sama seperti persamaan PACF pada data univariat *time series*, persamaan matriks parsial autoregresi ($P(s)$) juga memiliki sifat *cut-off* untuk model vektor AR.

Notasi $\Phi_{s,s}$ dalam regresi linier multivariat:

$$Y_{t+s} = \Phi_{s,1}Y_{t+s+1} + \Phi_{s,2}Y_{t+s+2} + \dots + \Phi_{s,s}Y_t + e_{s,t+s} \quad (2.10)$$

Heyse dan Wei (1985a,b) mendefinisikan PACF pada data univariat menjadi vektor *time series* (MPCCF) dan memperoleh matriks korelasi antara Y_t dan Y_{t+s} .

Persamaan matriks korelasi yang diperoleh sebagai korelasi antar vektor *residual* adalah sebagai berikut,

$$U_{s-1,t+s} = Y_{t+s} - \alpha_{s-1,1}Y_{t+s-1} - \dots - \alpha_{s-1,s-1}Y_{t+1}$$

$$= \begin{cases} Y_{t+s} - \sum_{k=1}^{s-1} \alpha_{s-1,k} Y_{t+s,k}, & s \geq 2 \\ Y_{t+1}, & s = 1 \end{cases} \quad (2.11)$$

dan

$$\begin{aligned} V_{s-1,t} &= Y_t - \beta_{s-1,1} Y_{t+1} - \dots - \beta_{s-1,s-1} Y_{t+s-1} \\ &= \begin{cases} Y_t - \sum_{k=1}^{s-1} \beta_{s-1,k} Y_{t+k}, & s \geq 2 \\ Y_t, & s = 1 \end{cases} \end{aligned} \quad (2.12)$$

Matriks koefisien regresi linier multivariat $\alpha_{s-1,k}$ dan $\beta_{s-1,k}$ diminimalisasi menjadi $E[|u_{s-1,t+s}|^2]$ dan $E[|v_{s-1,t}|^2]$. Diperoleh persamaan untuk generalisasi multivariat dengan meminimumkan persamaan di atas:

$$A(s) \alpha'(s) = c(s) \quad (2.13)$$

$$A(s) \beta'(s) = b(s)$$

$$\begin{bmatrix} \Gamma(0) & \Gamma'(1) & \dots & \Gamma'(s-2) \\ \Gamma(1) & \Gamma(0) & \dots & \Gamma'(s-3) \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma(s-2) & \Gamma(s-3) & \dots & \Gamma(0) \end{bmatrix} \begin{bmatrix} \alpha'_{s-1,1} \\ \alpha'_{s-1,2} \\ \vdots \\ \alpha'_{s-1,s-1} \end{bmatrix} = \begin{bmatrix} \Gamma(1) \\ \Gamma(2) \\ \vdots \\ \Gamma(s-1) \end{bmatrix} \quad (2.14)$$

untuk $s \geq 2$, didapatkan nilai $A(s)$, $b(s)$, dan $c(s)$ adalah sebagai berikut:

$$A(s) = \begin{bmatrix} \Gamma(0) & \Gamma'(1) & \dots & \Gamma'(s-2) \\ \Gamma(1) & \Gamma(0) & \dots & \Gamma'(s-3) \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma(s-2) & \Gamma(s-3) & \dots & \Gamma(0) \end{bmatrix}, \quad b(s) = \begin{bmatrix} \Gamma'(s-1) \\ \Gamma'(s-2) \\ \vdots \\ \Gamma'(1) \end{bmatrix}$$

$$c(s) = \begin{bmatrix} \Gamma(1) \\ \Gamma(2) \\ \vdots \\ \Gamma(s-1) \end{bmatrix}, \quad \alpha(s) = \begin{bmatrix} \alpha'_{s-1,1} \\ \alpha'_{s-1,2} \\ \vdots \\ \alpha'_{s-1,s-1} \end{bmatrix}, \quad \beta'(s) = \begin{bmatrix} \beta'_{s-1,s-1} \\ \beta'_{s-1,s-2} \\ \vdots \\ \beta'_{s-1,1} \end{bmatrix}$$

sehingga $\text{var}(\mathbf{u}_{s-1,t+s})$ dapat ditulis sebagai $V_u(s)$, $\text{var}(\mathbf{v}_{s-1,t})$ adalah $V_v(s)$ dan $\text{cov}(\mathbf{v}_{s-1,t}, \mathbf{u}_{s-1,t+s})$ $V_{vu}(s)$. Kemudian Heyse dan Wei (1985a, 1985b) mendapatkan

persamaan pada matriks autokorelasi lag parsial untuk lag s adalah sebagai berikut, (Wei, 2006)

$$P(s) = [\mathbf{D}_v(s)]^{-1} \mathbf{V}_{vu}(s) [\mathbf{D}_u(s)]^{-1} \quad (2.15)$$

dengan $\mathbf{D}_v(s)$ adalah matriks diagonal dimana elemen ke- i merupakan akar dari elemen diagonal ke- i dari $\mathbf{V}_v(s)$ dan $\mathbf{D}_u(s)$ yang didefinisikan sama untuk $\mathbf{V}_u(s)$.

Sama halnya pada plot MCCF, nilai MPCCF juga dinotasikan dalam bentuk simbol (+), (-) dan (.) dan mempunyai aturan perhitungan yang sama dengan nilai MCCF.

2.5 Model *Time Series Multivariate*

Terdapat beberapa model pada analisis multivariat, diantaranya adalah model *Vector Autoregressive* (VAR), model *Moving Average* (VMA), serta model *Vector Autoregressive Integrated Moving Average* (VARIMA).

2.5.1 Model *Vector Autoregressive* (VAR)

Menurut Wei (Ulya, 2019) pemodelan deret waktu dengan menggunakan *Vector Autoregressive* adalah salah satu model peramalan untuk data deret waktu *multivariate* yang sering digunakan karena mudah dan fleksibel jika dibandingkan dengan model lainnya. Model VAR merupakan pengembangan dari model AR dengan melibatkan lebih satu variabel, dimana pada model VAR ini semua variabel dianggap sebagai variabel endogen dan saling berhubungan. Secara umum model VAR (p) ditulis sebagai berikut:

$$\mathbf{Y}_t = \Phi_1 \mathbf{Y}_{t-1} + \dots + \Phi_p \mathbf{Y}_{t-p} + \mathbf{e}_t \quad (2.16)$$

dimana:

Y_t : vektor $m \times 1$ dari variabel pada waktu ke- t

Y_{t-1} : vektor $m \times 1$ dari variabel pada waktu ke- $(t-1)$

Φ_i : matriks parameter *Autoregressive* berukuran $(m \times 1)$, $i = 1, 2, \dots, p$

e_t : vektor $m \times 1$ dari *residual* pada waktu ke- t

2.5.2 Model Moving Average (VMA)

Model *Moving Average* (VMA) merupakan pengembangan atau gabungan dari beberapa model MA yang membentuk sebuah vektor. Secara umum persamaan VMA (q) ditulis sebagai berikut:

$$Y_t = \alpha_t - \Theta_1 \alpha_{t-1} - \dots - \Theta_q \alpha_{t-q} \quad (2.17)$$

dimana:

Y_t : vektor $n \times 1$ dari variabel pada waktu ke- t

Y_{t-1} : vektor $n \times 1$ dari variabel pada waktu ke- $(t-1)$

Θ_i : matriks parameter *Moving Average* berukuran $(n \times 1)$, $i = 1, 2, \dots, q$

e_t : vektor $n \times 1$ dari *residual* pada waktu ke- t

2.5.3 Model Vector Autoregressive Integrated Moving Average

Model *Vector Autoregressive Integrated Moving Average* atau yang biasa disebut dengan VARIMA adalah bentuk lain dari model multivariat ARIMA. Secara umum model VARIMA (p, d, q) dapat dituliskan dalam bentuk (Wei, 2006)

$$\Phi_p(B)D(B)\dot{Y}_t = \Theta_q(B)e_t \quad (2.18)$$

dengan $\dot{Y}_t = (Y_{1,t}, Y_{2,t}, \dots, Y_{m,t})'$ merupakan vektor respon yang terkoreksi nilai rata-rata, $\Phi_p(B)$ dan $\Theta_q(B)$ merupakan suatu matriks koefisien AR(p) dan MA(q), $D(B)$ merupakan operator proses diferensi yang dinyatakan dengan $diag((1-B)^{d1}, (1-B)^{d2}, \dots, (1-B)^{dm})$, dan $e(t) \sim IIDN(0, \Omega)$. Model VARIMA juga dapat dituliskan dengan persamaan:

$$\dot{Y}_t = \Phi_1 \dot{Y}_{t-1} + \dots + \Phi_p \dot{Y}_{t-p} + e_t \Theta_{t-1} - \dots - \Theta_q e_{t-q} \quad (2.19)$$

Misalkan untuk empat data dengan orde p dan $q=1$, maka persamaan di atas dapat ditulis dalam bentuk matriks sebagai berikut:

$$\begin{bmatrix} Y_t^{(1)} \\ Y_t^{(2)} \\ Y_t^{(3)} \\ Y_t^{(4)} \end{bmatrix} = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \varphi_{13} & \varphi_{14} \\ \varphi_{21} & \varphi_{22} & \varphi_{23} & \varphi_{24} \\ \varphi_{31} & \varphi_{32} & \varphi_{33} & \varphi_{34} \\ \varphi_{41} & \varphi_{42} & \varphi_{43} & \varphi_{44} \end{bmatrix} \begin{bmatrix} Y_t^{(1)} \\ Y_t^{(2)} \\ Y_t^{(3)} \\ Y_t^{(4)} \end{bmatrix} + \begin{bmatrix} e_t^{(1)} \\ e_t^{(2)} \\ e_t^{(3)} \\ e_t^{(4)} \end{bmatrix} - \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} \\ \theta_{31} & \theta_{32} & \theta_{33} & \theta_{34} \\ \theta_{41} & \theta_{42} & \theta_{43} & \theta_{44} \end{bmatrix} \begin{bmatrix} e_t^{(1)} \\ e_t^{(2)} \\ e_t^{(3)} \\ e_t^{(4)} \end{bmatrix} \quad (2.20)$$

2.6 Uji Signifikansi Parameter

Pengujian signifikansi parameter model dilakukan secara parsial menggunakan uji t (Gujarati, 2003). Pengujian ini dilakukan untuk mengetahui parameter-parameter yang signifikan pada model dengan hipotesis awal (H_0) yaitu parameter tidak signifikan, sedangkan hipotesis alternatifnya (H_1) adalah parameter signifikan. Pengambilan keputusan dilakukan dengan membandingkan nilai p -value dari hasil pengujian dengan nilai signifikansi (α) sebesar 5% atau 0,05. Tolak H_0 apabila nilai p -value $< \alpha$, sehingga kesimpulan yang diambil ialah parameter signifikan, dan begitu pula sebaliknya.

Apabila terdapat parameter yang tidak signifikan terhadap model, maka perlu dilakukan *restrict*. *Restrict* dilakukan secara bertahap dimulai dari parameter

yang memiliki nilai *p-value* tertinggi sampai semua parameter bernilai kurang dari α (5%) atau signifikan terhadap model.

2.7 Diagnostic Check Model

Pengujian diagnostik model dilakukan untuk menguji apakah residual memenuhi asumsi identik, independen dan berdistribusi normal multivariat.

2.7.1 Uji *white noise*

Pengujian residual *white noise* dilakukan untuk mengetahui apakah residual model saling independen antara satu dengan lainnya. Pengujian pada asumsi ini dapat menggunakan plot MACF dari residual, dimana bersifat *white noise* apabila tidak terdapat lag yang signifikan pada plot tersebut. Selain itu, dapat juga digunakan uji *Portmanteau* yang merupakan pengembangan dari uji Ljung Box dalam kasus *multivariate*. Pengambilan keputusan dilakukan dengan membandingkan nilai *p-value* pada beberapa lags hasil pengujian dengan nilai signifikansi (α) sebesar 5% atau 0,05. Model bersifat *white noise* apabila nilai *p-value* pada masing-masing lag tiap model $> \alpha$.

2.7.2 Uji *multivariate normal*

Pengujian residual *multivariate normal* digunakan untuk mengetahui apakah residual berdistribusi normal. Pengujian ini dilakukan menggunakan uji *Shapiro-Wilk*, dimana pengambilan keputusan dilakukan dengan membandingkan nilai *p-value* pada dari hasil pengujian dengan nilai signifikansi (α) sebesar 5%

atau 0,05. Residual pada model bersifat *multivariate normal* apabila nilai *p-value* yang dihasilkan bernilai $> \alpha$.

2.8 Support Vector Regression

Support Vector Regression (SVR) merupakan penerapan dari metode *Support Vector Machine* (SVM) dalam kasus regresi. Konsep SVM menggunakan konsep ϵ -insentive loss function yang dapat digeneralisasi untuk melakukan pendekatan fungsi yang dikenal dengan SVR (Gunn, 1998). Jika SVM digunakan untuk mencari *hyperplane* (fungsi pemisah) yang terbaik diantara 2 obyek yang tidak terbatas jumlahnya dengan cara memaksimalkan jarak (*margin*) antara dua obyek yang berbeda sedangkan SVR digunakan untuk menemukan suatu fungsi yang memiliki deviasi paling besar ϵ dari target aktual y_i .

Konsep SVR didasarkan pada *structural risk minimization*, yaitu untuk mengestimasi suatu fungsi dengan cara meminimalkan batas atas dari *generalization error*, sehingga SVR mampu mengatasi *overfitting*. Tujuan dari SVR adalah untuk mendapatkan suatu fungsi dengan tingkat kesalahan paling kecil sehingga menghasilkan suatu prediksi yang bagus. Ide dasar dari SVR yaitu dengan menentukan set data yang dibagi menjadi *set training (in sample)* dan *set testing (out sample)*. Kemudian dari set training tersebut ditentukan suatu fungsi regresi dengan batasan deviasi tertentu sehingga dapat menghasilkan prediksi yang mendekati dari target aktual.

Kelebihan SVR dibanding regresi linier adalah regresi linier menghasilkan fungsi linier yang berupa garis lurus, sedangkan SVR menghasilkan trend data

yang bergelombang mengikuti jalur data yang terbentuk sehingga diharapkan hasil prediksi yang didapat lebih akurat. Fungsi regresi dari metode SVR akan sempurna apabila batas deviasinya sama dengan 0 sehingga dapat dituliskan sebagai berikut:

$$\mathbf{f}(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) + \mathbf{b} \quad (2.21)$$

dengan :

\mathbf{w} = vector pembobot

$\boldsymbol{\varphi}(\mathbf{x})$ = fungsi yang memetakan \mathbf{x} dalam suatu dimensi

\mathbf{b} = bias

Simbol $\boldsymbol{\varphi}(\mathbf{x})$ menunjukkan suatu titik di dalam *feature space* F yang merupakan hasil pemetaan \mathbf{x} di dalam *input space*. Koefisien \mathbf{w} dan \mathbf{b} di sini berfungsi untuk meminimalkan fungsi resiko. Dengan meminimalkan fungsi resiko tersebut akan membuat suatu fungsi menjadi setipis mungkin, sehingga kapasitas fungsi dapat terkontrol, hal ini dinamakan regularisasi. Koefisien \mathbf{w} dan \mathbf{b} diestimasi dengan cara meminimalkan fungsi resiko (*risk function*) yang didefinisikan dalam persamaan sebagai berikut:

$$R(f(\mathbf{x})) = \frac{C}{T} \sum_{i=1}^T L_{\varepsilon}(y_i, f(x_i)) + \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.22)$$

dimana

$$L_{\varepsilon}(y_i, f(x_i)) = \begin{cases} 0 & ; |y_i - f(x_i)| \leq \varepsilon \\ |y_i - f(x_i)| - \varepsilon & ; \text{lainnya} \end{cases} \quad (2.23)$$

Dengan L_{ε} merupakan ε -insensitive loss function, y_i adalah vektor dari nilai sebenarnya, C dan ε merupakan *hyper-parameter* yang sudah ditentukan.

Fungsi f diasumsikan dapat mengaproksimasi semua titik (x_i, y_i) dengan presisi ε jika semua titik berada dalam rentang $f \pm \varepsilon$ atau disebut *feasible*. Sedangkan *infeasible* merupakan kondisi dimana ada beberapa titik yang berada diluar rentang $f \pm \varepsilon$. Titik-titik yang *infeasible* bisa ditambahkan variabel *slack* ξ , ξ^* untuk mengatasi *infeasible constrain*. Sehingga optimasi persamaannya dapat ditransformasi dalam bentuk:

$$\min \frac{1}{2} \|w\|^2 + C \frac{1}{n} \sum_{t=1}^n (\xi_t + \xi_t^*) \quad (2.24)$$

dengan batasan:

$$\begin{aligned} w\varphi(\mathbf{x}_t) + b - y_t &\leq \varepsilon + \xi_t^* \\ y_t - w\varphi(\mathbf{x}_t) - b &\leq \varepsilon + \xi_t \\ \xi_t, \xi_t^* &\geq 0, \quad t = 1, 2, \dots, n \end{aligned}$$

Optimalisasi pada batasan tersebut bisa diselesaikan menggunakan primal lagrangian dalam bentuk sebagai berikut, (Hong, 2008)

$$\begin{aligned} L(w, b, \xi_t, \xi_t^*, \alpha_t, \alpha_t^*, \beta_t, \beta_t^*) &= \frac{1}{2} \|w\|^2 + C \left[\sum_{t=1}^n (\xi_t + \xi_t^*) \right] \\ &\quad - \sum_{t=1}^n \beta_t [w\varphi(x_t) + b - y_t + \varepsilon + \xi_t^*] \\ &\quad - \sum_{t=1}^n \beta_t^* [y_t - w\varphi(x_t) - b + \varepsilon + \xi_t^*] \\ &\quad - \sum_{t=1}^n (\alpha_t \xi_t + \alpha_t^* \xi_t^*) \end{aligned} \quad (2.25)$$

Persamaan diatas diminimalkan pada variabel primal w, b, ξ_t, ξ_t^* dan dimaksimalkan dalam bentuk *lagrangian multiplier nonnegative* $\alpha_t, \alpha_t^*, \beta_t, \beta_t^*$ seperti ditampilkan pada persamaan berikut (Hong, 2008)

$$\frac{\partial L}{\partial w} = w - \sum_{t=1}^n (\beta_t - \beta_t^*) \varphi(x_t) = 0 \quad (2.26)$$

$$\frac{\partial L}{\partial b} = \sum_{t=1}^n (\beta_t^* - \beta_t) = 0 \quad (2.27)$$

$$\frac{\partial L}{\partial \xi} = C - \beta_t - \alpha_t = 0 \quad (2.28)$$

$$\frac{\partial L}{\partial w} = C - \beta_t^* - \alpha_t^* = 0 \quad (2.29)$$

Sehingga kondisi Karush-Kuhn-Tucker terapkan untuk model regresi dan didapatkan persamaan *dual lagrangian* dengan fungsi kernel :

$$\mathbf{K}(\mathbf{x}_t, \mathbf{x}_u) = \varphi(\mathbf{x}_t) \varphi(\mathbf{x}_u) \quad (2.30)$$

Fungsi yang dapat digunakan untuk metode SVR adalah sebagai berikut:

1. Linier

$$\mathbf{K}(\mathbf{x}, \mathbf{x}^T) = \mathbf{x}^T \mathbf{x} \quad (2.31)$$

2. Polynomial

$$\mathbf{K}(\mathbf{x}, \mathbf{x}^T) = (\mathbf{x}^T \mathbf{x} + 1)^P \quad (2.32)$$

3. *Radial basis function* (RBF)

$$\mathbf{K}(\mathbf{x}_t, \mathbf{x}_u) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x}_t - \mathbf{x}_u\|^2\right) \quad (2.33)$$

Salah satu cara dalam menentukan parameter optimal pada model SVR adalah menggunakan optimasi *grid search*. Optimasi *grid search* merupakan kombinasi parameter yang diujikan pada suatu model SVR untuk mencari nilai *error* terkecil. Optimasi ini mengidentifikasi parameter optimal dalam data *training*, sehingga model tersebut mampu secara akurat memprediksi data *testing*. Proses ini dilakukan secara *trial and error* hingga didapatkan parameter optimal dengan RMSE terkecil.

2.9 Evaluasi Model

Evaluasi model dilakukan menggunakan beberapa kriteria, yaitu nilai *Root Mean Square Error* (RMSE), *Akaike Information Criterion* (AIC) dan *Mean Absolute Percentage Error* (MAPE).

2.9.1 AIC

Nilai AIC digunakan untuk pemilihan model terbaik pada model VAR, dimana model terbaik adalah model yang memiliki nilai AIC terkecil, berikut adalah persamaannya:

$$AIC(p) = n \ln(|S_p|) + 2pv^2 \quad (2.34)$$

dimana p adalah banyak parameter, v adalah banyaknya variabel, dan $|S_p|$ adalah determinan dari *residual sum of square* dan perkalian silangnya,

$$S_p = \sum_{t=p+1}^n (Y_t - \hat{\tau} - \hat{\Phi}_1 Y_{t-1} - \dots - \hat{\Phi}_p Y_{t-p}) \times (Y_t - \hat{\tau} - \hat{\Phi}_1 Y_{t-1} - \dots - \hat{\Phi}_p Y_{t-p})' \quad (2.35)$$

Dimana $\hat{\tau}$ merupakan vector konstan.

2.9.2 RMSE

Nilai RMSE digunakan untuk membandingkan fungsi kernel terbaik pada pemodelan SVR, dimana fungsi kernel terbaik merupakan fungsi kernel yang mempunyai nilai RMSE terkecil. Berikut adalah persamaan pada nilai RMSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2} \quad (2.36)$$

dimana n merupakan banyaknya data *out-sample*, Y_t menyatakan data *out-sample* ke- t , dan \hat{Y}_t merupakan data hasil ramalan ke- t

2.9.3 MAPE

Dalam suatu peramalan, nilai prediksi yang dihasilkan pasti mengandung kesalahan, sehingga nilai yang dihasilkan tidak mempunyai tingkat akurasi 100%. Semakin kecil tingkat kesalahan maka semakin baik prediksi yang dihasilkan. Dalam penelitian ini akan digunakan nilai MAPE sebagai pengukuran kesalahan hasil prediksi. Rumus untuk menghitung MAPE adalah sebagai berikut :

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \times 100 \quad (2.37)$$

dimana :

Y_t : data aktual pada periode t

\hat{Y}_t : nilai peramalan pada periode t

n : jumlah data observasi

Terdapat tabel kriteria MAPE yang menyatakan tingkat keakurasian dalam persen.

Tabel 2.2 Kriteria MAPE

MAPE	Pengertian
<10%	Kemampuan Peramalan Sangat Baik
<20%	Kemampuan Peramalan Baik
<30%	Kemampuan Peramalan Cukup Baik
>30%	Kemampuan Peramalan Tidak Akurat