

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Bus DAMRI

Berawal dari jaman pendudukan Jepang di Indonesia tahun 1943, pada saat itu terdapat dua perusahaan angkutan yaitu Jawa Unyu Zigyosha yang mengkhususkan diri pada jasa angkutan barang dengan truk, gerobak/cikar, dan Zidosha Sokyoku mengkhususkan diri untuk angkutan penumpang dengan kendaraan bermotor/bus. Tahun 1945 setelah Indonesia merdeka, dibawah pengelolaan Kementerian Perhoebongan RI, Jawa Unyu Zigyosha berubah nama menjadi “Djawatan Pengankoetan” untuk angkutan barang dan Zidosha Sokyoku beralih menjadi “Djawatan Angkutan Darat” untuk angkutan penumpang. Pada tanggal 25 November 1946, keduanya digabungkan berdasarkan Makloemat Mentei Perhoebongan RI No.01/DAM/46 dibentuklah “Djawatan Angkoetan Motor Repoeblik Indonesia”, disingkat DAMRI. Tugas utama DAMRI adalah menyelenggarakan pengangkutan darat dengan bus, truk, dan angkutan bermotor lainnya (DAMRI, 2018).

#### 2.2 Definisi Penumpang

Kata *passenger* berasal dari bahasa Inggris. Arti *Passenger* Menurut Echols & Hassan (2005) dalam kamus Bahasa Inggris Indonesia “*passenger*” adalah penumpang dan Menurut Damardjati (1995), pengertian penumpang adalah setiap orang yang di angkut ataupun harus di angkut di dalam pesawat udara ataupun alat pengangkutan lainnya, atas dasar persetujuan dari

perusahaan barang dan jasa yang mereka dapat berupa seorang (individu) dan dapat pula sebagai suatu jasa perusahaan.

### 2.3 Peramalan (*Forecasting*)

Peramalan berasal dari kata ramalan yang artinya adalah suatu situasi atau kondisi yang diperkirakan akan terjadi pada masa yang akan datang. Sedangkan peramalan adalah bentuk kegiatannya. Peramalan adalah memperkirakan keadaan dimasa yang akan datang melalui pengujian keadaan dimasa lalu. Peramalan merupakan proses untuk memperkirakan berapa kebutuhan dimasa yang akan datang yang meliputi kebutuhan dalam ukuran kuantitas, kualitas, waktu dan lokasi yang di butuhkan untuk memenuhi permintaan barang atau jasa (Nasution, 1999). Peramalan memiliki peranan penting dalam sebuah perusahaan, dikarenakan peramalan merupakan dasar dari sebuah perencanaan, baik perencanaan penjualan, pembelian, produksi, maupun transportasi. Oleh karena itu pemilihan cara melakukan peramalan menjadi sangat penting, walaupun cara peramalan bukan satu-satunya faktor yang mempengaruhi akurasi atas meningkatnya jumlah penumpang bus.

Dalam melakukan peramalan terdapat berbagai cara. Jika berdasarkan sifat ramalan yang disusun, maka peramalan dibedakan menjadi dua macam (Makridarkis & Wheelwright, 1999):

#### 1. Peramalan kualitatif

Beberapa model *forecasting* yang digolongkan sebagai model kualitatif.

##### a. Dugaan Manajemen (*management estimate*)

Merupakan metode *forecasting* dimana *forecasting* semata-mata

berdasarkan pertimbangan manajemen. Metode ini cocok dalam situasi yang sangat sensitif terhadap intuisi dari satu atau sekelompok kecil orang yang karena pengalamannya mampu memberikan opini yang kritis dan relevan.

b. Riset Pasar (*market research*)

Merupakan metode *forecasting* berdasarkan hasil *Survey* pasar yang dilakukan oleh tenaga pemasar produk atau yang mewakilinya. Metode inimenjaring informasi dari pelanggan yang berkaitan dengan rencana peningkatan jumlah pengguna produk di masa yang akan datang.

c. Metode kelompok Terstruktur (*structured groups methods*)

Merupakan metode *forecasting* berdasarkan proses konvergensi dari opini beberapa orang atau ahli secara interaktif dan membutuhkan fasilitator untuk menyimpulkan hasil dari *forecasting*.

d. Analogi Historis (*historical analogy*)

Merupakan teknik *forecasting* berdasarkan pola data masa lalu dari produk yang disamakan secara analogi.

2. Peramalan Kuantitatif

Yaitu peramalan berdasarkan atas dasar kuantitatif pada masa lampau. Metode kuantitatif sendiri dibedakan menjadi dua, yaitu metode deret berkala dan metode kausal.

a. Metode deret berkala (*time series*), yaitu metode kualitatif yang didasarkan atas penggunaan analisa pada hubungan antar variabel yang akan di perkirakan dengan variabel waktu. Data di masalalu pada analisa deret

waktu akan di pengaruhi empat komponen utama *Trend* (tren), *cycle* (siklus), *season* (musiman) dan *random* (acak).

b. Metode Kausal

Metode peramalan kausal mengembangkan model sebab-akibat antara permintaan yang diramalkan dengan variabel-variabel lain yang di anggap berpengaruh. Pada metode kausal ini dibagi menjadi tiga bagan yaitu :

1) Metode Kolerasi Regresi

- a. Peramalan penjualan
- b. Peramalan keuntungan
- c. Peramalan permintaan
- d. Peramalan keadaan ekonomi

Metode ini sangat cocok digunakan untu peramalan jangka pendek, data yang digunakan kumpulan dari data beberapa tahun.

2) Metode ekinimerik

- a. Peramalan penjualan menurut kelas produksi
- b. Peramalan keadaan ekonomi masyarakat yang meliputi permintaan, harga, dan penawaran.

Metode ini sangat cocok untuk peramalan jangka pendek dan panjang.

Data yang digunakan merupakan kumpulan data beberapa tahun.

3) Metode input output

- a. Peramalan penjualan menurut kelas produksi
- b. Peramalan keadaan ekonomi masyarakat yang meliputi permintaan, harga, dan penawaran.

Metode ini sangat cocok untuk peramalan jangka pendek dan panjang.

Data yang digunakan merupakan kumpulan data 10-15 tahun.

## 2.4 Metode Peramalan Artificial Neural Network

*Artificial Neural Network* (ANN) merupakan model matematis yang menyerupai cara kerja otak biologis. *Neural Network* terdiri dari sejumlah inti syaraf, hubungan antara inti dan aturan-aturan pembelajaran. Pembobot diberikan terhadap tiap hubungan menurut aturan pembelajaran tertentu. Hasil dari proses pembelajaran dalam jaringan tersimpan dalam bentuk bobot dalam tiap hubungan antara inti syaraf. ANN berusaha meniru struktur/arsitektur dan cara kerja otak manusia sehingga mampu menggantikan beberapa pekerjaan manusia. Pekerjaan seperti mengenali pola, prediksi, klasifikasi, pendekatan fungsi dan optimasi adalah pekerjaan-pekerjaan yang di harapkan bisa di selesaikan dengan ANN (Santosa,2007).

### 2.4.1 Jaringan Syaraf Tiruan (JST)

Jaringan syaraf tiruan (JST) atau *neural network* adalah metode komputasi yang meniru sistem jaringan saraf biologis. Metode ini menggunakan elemen perhitungan non-linear dasar yang disebut neuron yang diorganisasikan sebagai jaringan yang saling berhubungan, sehingga mirip dengan jaringan saraf manusia. Jaringan saraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran (Yani, 2005)

Jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada

otak manusia tersebut. Istilah buatan digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Jaringan syaraf tiruan memiliki karakteristik Pada hubungan *neuron* disebut arsitektur. *Neuron* adalah bagian terkecil dari jaringan syaraf tiruan yang berfungsi sebagai elemen proses. JST ini menarik karena memiliki kemampuan untuk belajar dari pengalaman serta mampu menyelesaikan permasalahan yang tidak terstruktur.

Hal yang ingin dicapai dengan melatih JST adalah untuk mencapai keseimbangan antara kemampuan *memorisasi* dan *generalisasi*. Yang dimaksud kemampuan *memorisasi* adalah kemampuan JST untuk mengambil kembali secara sempurna sebuah pola yang dipelajari. Kemampuan *generalisasi* adalah kemampuan JST untuk menghasilkan respons yang bisa diterima terhadap pola-pola input yang serupa (namun tidak identik) dengan pola-pola yang sebelumnya telah dipelajari. Hal ini sangat bermanfaat bila pada suatu saat ke dalam JST itu diinputkan informasi baru yang belum pernah dipelajari, maka JST itu masih akan tetap dapat memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati (Puspita, 2006)

#### 2.4.2 Perhitungan Input ANN

Perhitungan yang diperlukan dalam menghitung nilai input dalam unit  $i$  pada jaringan pada waktu  $t$  dilakukan dengan persamaan :

$$\text{net}_i(t) = \sum w_{ij}(t) o_j(t) \quad (1.1)$$

Dimana :

$\text{net}_i(t)$  : jaringan *signal input* pada unit  $i$  pada jaringan

$w_{ij}(t)$  : nilai pembobot dari unit  $j$  ke unit  $i$

$o_j(t)$  : output dari unit  $j$  dalam jaringan

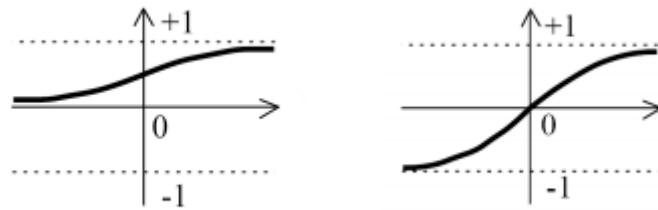
#### 2.4.3 Fungsi aktivasi

Fungsi aktivasi adalah salah satu parameter yang terpenting dalam jaringan saraf tiruan. Pemilihan fungsi aktivasi dapat berpengaruh pada performa jaringan saraf tiruan.

Ada beberapa fungsi aktivasi yang sering digunakan pada jaringan saraf tiruan (Siang, 2005: 26), yaitu :

##### 1) Fungsi *Sigmoid*

Terdapat 2 buah fungsi sigmoid yaitu *sigmoid biner (logsig)* dan *sigmoid bipolar (tansig)* yang akan diperjelas dengan gambar 2.1 terkait grafik fungsi *sigmoid biner* dan fungsi *sigmoid bipolar*.



Gambar 2.1 Grafik fungsi sigmoid biner (a) dan fungsi sigmoid bipolar (b) (Hermawan, 2006:52)

Sigmoid biner memiliki nilai (0,1) dan memiliki bentuk fungsi:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Dengan turunan

$$f'(x) = f(x)(1 - f(x))$$

Sedangkan pada sigmoid bipolar bentuk fungsinya mirip dengan fungsi sigmoid biner, tetapi dengan nilai interval (-1,1)

$$f(x) = \frac{2}{1 + e^{-x}} - 1$$

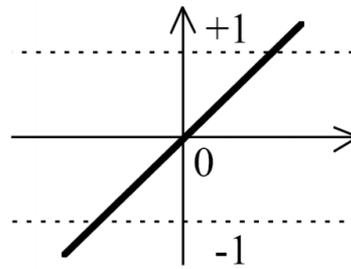
Dengan turunan

$$f'(x) = \frac{(1 + f(x))(1 - f(x))}{2}$$

## 2) Fungsi Identitas

Fungsi identitas memiliki nilai keluaran sembarang bilangan riil, (bukan hanya pada interval [0,1] atau [-1,1],

dimana  $f(x) = x$  untuk semua  $x$ . Berikut ditampilkan pada gambar 2.2 grafik fungsi identitas

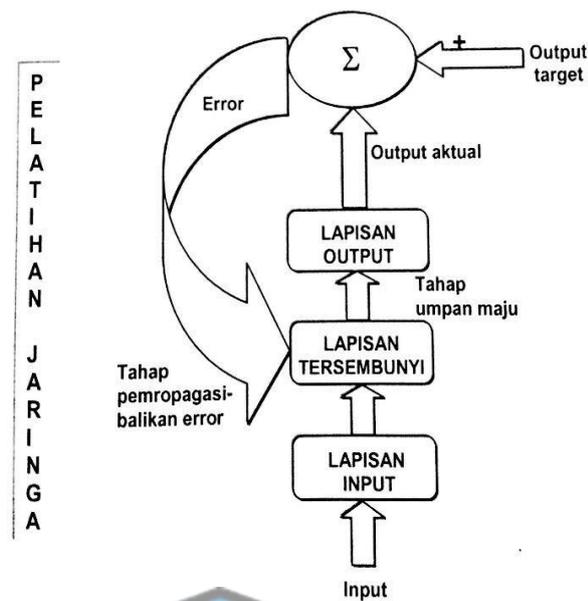


Gambar 2.2 Grafik fungsi identitas (Hermawan, 2006:54)

#### 2.4.4 Backpropagation Network

Algoritma *backpropagation* merupakan bagian dari algoritma pembelajaran terawasi yang biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyi (J.J Siang,2012). Algoritma ini menggunakan *error* keluaran untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Saat perambatan maju, *neuron- neuron* diaktifkan dengan menggunakan fungsi aktivasi yang dapat didiferensiasikan seperti *sigmoid*.

Pelatihan pada *backpropagation* terdiri dari 3 fase (Siang, 2005: 100-101), yaitu (1) Propogasi maju, (2) Propogasi mundur, (3) Perubahan bobot. Untuk lebih jelasnya, algoritma pelatihan *backpropagation* dapat dilihat pada gambar 2.12 di bawah ini.



Gambar 2.3 Alur kerja *backpropagation* (Puspitaningrum, 2006: 127)

### Metode Peramalan Support Vector Regression

*Support Vector Regression* (SVR) merupakan cabang dari *Support Vector Machine* (SVM). SVR merupakan penerapan SVM untuk kasus *regresi*. Dalam kasus klasifikasi, yang merupakan keahlian SVM, output data berupa bilangan bulat atau diskrit, sedangkan untuk kasus *regresi* output data berupa bilangan riil atau kontinu. Dengan menggunakan konsep  *$\epsilon$ -insensitive loss function*, yang diperkenalkan oleh Vapnik, SVM bisa digeneralisasi untuk melakukan pendekatan fungsi (*function approximation*) atau *regresi* (Santosa, 2007).

Misalkan kita punya  $\ell$  set data *training*,  $(x_i, y_i)$ ,  $i = 1, \dots, \ell$  dengan data input  $x = (x_1, x_2, \dots, x_\ell)$  dan output yang bersangkutan  $y = \{y_1, \dots, y_\ell\} \subseteq \mathcal{R}$ . Dengan SVR, kita ingin menemukan suatu fungsi  $f(x)$  yang mempunyai deviasi paling besar  $\mathcal{E}$  dari target aktual  $y_i$  untuk semua data *training* maka, dengan SVR kita akan mendapatkan suatu tabung. Manakala nilai  $\mathcal{E}$  sama dengan 0 maka kita

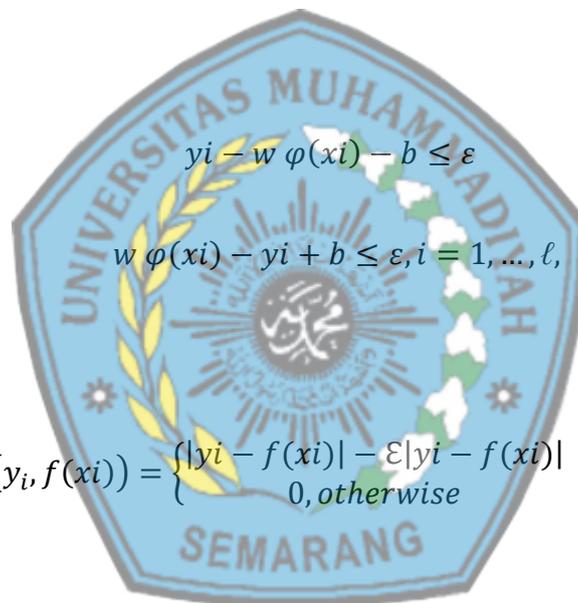
dapatkan suatu regresi yang sempurna. Misalkan kita mempunyai fungsi berikut sebagai garis regresi:

$$f(x) = w^T \varphi(x) + b \quad (1.12)$$

Dimana  $\varphi(x)$  menunjukkan suatu titik di dalam *feature space*  $F$  hasil pemetaan  $x$  di dalam *input space*. Koefisien  $w$  dan  $b$  diestimasi dengan cara meminimalkan fungsi resiko (*risk function*) yang didefinisikan dalam persamaan:

$$\min \frac{1}{2} \|w\|^2 + C \frac{1}{l} \sum_{i=1}^l L_{\varepsilon}(y_i, f(x_i)) \quad (1.13)$$

Subject to



$$y_i - w \varphi(x_i) - b \leq \varepsilon$$

$$w \varphi(x_i) - y_i + b \leq \varepsilon, i = 1, \dots, l,$$

Dimana

$$L_{\varepsilon}(y_i, f(x_i)) = \begin{cases} |y_i - f(x_i)| - \varepsilon & |y_i - f(x_i)| > \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (1.14)$$

Dengan :

$w$  = vektor bias

$x$  = nilai masukan atribut

$b$  = bias

Faktor  $\|w\|^2$  dinamakan *regularasi*. Meminimalkan  $\|w\|^2$  akan membuat suatu fungsi setipis (*flat*) mungkin, sehingga bisa mengontrol kapasitas fungsi (*function capacity*). Faktor kedua dalam fungsi tujuan adalah kesalahan empirik (*empirical error*) yang diukur dengan  $\varepsilon$ - *insensitive loss function*. Menggunakan

ide  $\epsilon$ - *insensitive loss function*, kita harus meminimalkan norm dan  $w$  agar mendapatkan generalisasi yang baik untuk fungsi regresi  $f$ . Karena itu kita perlu menyelesaikan problem optimasi berikut:

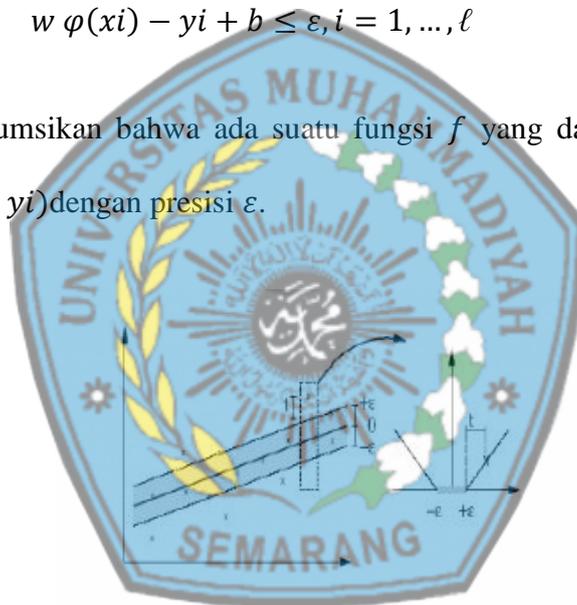
$$\text{Min} \frac{1}{2} \|w\|^2 \quad (1.15)$$

Subject to

$$y_i - w \varphi(x_i) - b \leq \epsilon$$

$$w \varphi(x_i) - y_i + b \leq \epsilon, i = 1, \dots, \ell \quad (1.16)$$

Kita asumsikan bahwa ada suatu fungsi  $f$  yang dapat mengaproksimasi semua titik  $(x_i, y_i)$  dengan presisi  $\epsilon$ .



Gambar 3.  $\epsilon$ - *insensitive loss function*. Semua titik di luar are berwarna dikenai pinalti

Dalam kasus ini kita asumsikan bahwa semua titik ada dalam rentang  $f \pm \epsilon$  (*feasible*). Dalam hal ketidak layakan (*infeasibility*), dimana ada beberapa titik yang mungkin keluar dari rentang  $f \pm \epsilon$ , kita bisa menambahkan *variable slack*  $t_m$  untuk mengatasi masalah pembatas yang tidak layak (*infeasible constraints*)

dalam problem optimasi. Selanjutnya problem optimasi di atas bisa di formulasikan sebagai berikut :

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} (t_i + t_i^*) \quad (1.17)$$

Subject to

$$y_i - w^T \varphi(x_i) - b - t_i \leq \varepsilon, i = 1, \dots, \ell$$

$$w^T \varphi(x_i) - y_i + b - t_i^* \leq \varepsilon, i = 1, \dots, \ell$$

$$t_i, t_i^* \geq 0,$$

konstanta  $C > 0$  menentukan tawar-menawar (*trade off*) antara ketipisan fungsi (*flatness of function*)  $f$  dan batas atas deviasi lebih dari  $\varepsilon$  masih ditoleransi. Semua deviasi lebih besar daripada  $\varepsilon$  akan dikenakan penalti sebesar  $C$ . Dalam SVR,  $\varepsilon$  ekuivalen dengan akurasi dari aproksimasi kita terhadap data *training*. Nilai  $\varepsilon$  yang kecil terkait dengan nilai yang tinggi pada variabel *slack*  $t(i)$  dan akurasi aproksimasi yang tinggi. Sebaliknya, nilai yang tinggi untuk  $\varepsilon$  berkaitan dengan nilai  $t(i)$  yang kecil dan akurasi aproksimasi yang rendah. Menurut persamaan (1.16), nilai yang tinggi untuk variabel *slack* akan membuat kesalahan empirik mempunyai pengaruh yang besar terhadap faktor regulasi. Dalam empirik mempunyai pengaruh yang besar terhadap faktor regularisasi. Dalam SVR, *support vectors* adalah data *training* yang terletak pada dan di luar batas  $\varepsilon$  dari fungsi keputusan. Karena itu jumlah *support vectors* menurun dengan naiknya nilai  $\varepsilon$ . Dalam formulasi *dual*, problem optimasi dari SVR adalah sebagai berikut :

$$\max \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha'_i)(\alpha_i - \alpha'_i K(x_i x_j))$$

$$\sum_{i=1}^l y_i (\alpha_i - \alpha'_i) - \varepsilon \sum_{j=1}^l (\alpha_i - \alpha'_i) \quad (1.18)$$

Subject to:

$$\sum_{i=1}^l (\alpha_i - \alpha'_i) = 0$$

$$0 \leq \alpha_i \leq C, i = 1, \dots, l$$

Dimana C didefinisikan oleh user,  $K(x_i, x_j)$  adalah *dot-product kernel* yang didefinisikan sebagai  $K(x_i, x_j) = \varphi^T(x_i) \varphi(x_j)$ . Dengan menggunakan *Langre multiplier* dan kondisi optimalitas, fungsi regresi secara eksplisit dirumuskan sebagai :

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha'_i) K(x_i, x) + b \quad (1.19)$$

Dimana  $K(x_i, x)$ , didefinisikan melalui fungsi kernel  $k$ . Persamaan (1.18) adalah program kuadratik (QP).

### 2.5.1 Metode Kernel

Seperti yang sudah dijelaskan pada sub bab sebelumnya, dikatakan bahwa pada SVR dibutuhkan sebuah metode yang memetakan data kedalam suatu *featur space*. Pemetaan itu dilakukan oleh sebuah metode yaitu metode kernel, atau fungsi kernel.

SVM nonlinear menggunakan pendekatan kernel untuk data yang kelasnya tidak terdistribusi secara linear. Kernel merupakan suatu fungsi yang memetakan fitur data dari dimensi awal (rendah) ke dalam fitur baru dengan dimensi yang lebih tinggi (Prasetyo, 2012). Macam –macam fungsi kernel antara lain:

- a. Linear

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y} \quad (1.20)$$

- b. Quadratic

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^2 \quad (1.21)$$

- c. Polinomial

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + c)^d \quad (1.22)$$

- d. Raial Basic Function (RBF)

$$\exp \left[ \frac{-1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|^2 \right] \quad (1.23)$$

- e. Sigmoid

$$K(\mathbf{x}, \mathbf{y}) = \tan(\sigma(\mathbf{x}, \mathbf{y}) + c) \quad (1.24)$$

Fungsi kernel yang dipilih harus tepat karena sangat penting untuk menentukan fitur baru dimana *hyperlane* akan dicari.

### 2.5.2 Algoritma *Grid Search*

Algoritma *Grid Search* adalah algoritma yang membagi jangkauan parameter yang akan dioptimalkan dedalam grid dan melintasi semua titik untuk mendapatkan parameter yang optimal (Yasin *et al.*, 2014). Penggunaan algoritma *grid search* harus dipadu oleh beberapa metrik kerja, yang diukur dengan *cross-validation* pada data *training*. Sehingga disarankan untuk mencoba beberapa variasi pasangan parameter pada *hyperlane* SVR (HSU *et al.*, 2004). Pasangan

parameter hasil dari uji *cross-validation* dengan akurasi terbaik merupakan parameter yang optimal, parameter tersebut akan digunakan untuk membentuk model SVR yang akan digunakan untuk memprediksi data testing untuk serta mendapatkan generalisasi tingkat akurasi model.

*cross-validation* adalah pengujian standar yang dilakukan untuk memprediksi *error rate*. Data training dibagi secara random ke dalam beberapa bagian demi bagian, selanjutnya hitung rata-rata *error rate* untuk mendapatkan *error rate* keseluruhan dalam *cross-validation* dikenal validasi *leave-one-out* (LOO). Dalam LOO, data dibagi menjadi 2 subset. Subset 1 berisi N-1 data untuk *training* dan data satu sisanya untuk *testing* (Leidiyana, 2013)

$$CV = \sum_{i=1}^n (Y_i - \hat{Y}_{\neq i}) \quad (1.25)$$

Dengan :  $\hat{Y}_{\neq i}$  = Nilai penaksir  $Y_i$  dimana pengamatan ke-i dihilangkan dari proses penaksiran

$Y_i$  = Nilai aktual  $y$  pada pengamatan ke-i

Sehingga algoritma yang terbentuk sebagai berikut :

$x = [x_1, x_2, \dots, x_n]$  : sampel *training*

*Grid* : *Grid Search*

*LOO* : *Leave-one-out*

*CV* : jenis penentuan *error* dengan *Cross Validation*

*C* : *cost*

$\varepsilon$  : *epsilon*

$\gamma$  : *gamma*

1. Tentukan nilai-nilai parameter kernel ( $C$ ,  $\gamma$  dan  $\varepsilon$ ) dengan  $\varepsilon$  bilangan positif kecil
2. Pasangkan nilai masing-masing parameter
3. LOO membagi  $x$  menjadi 2 bagian, bagian 1 berisi  $n-1$  untuk *training* dan sisanya untuk *testing*
4. Hitung nilai CV untuk memprediksi *error* dari masing-masing pasangan yang terbentuk
5. Hitung rata-rata *error* dari masing-masing pasangan parameter kernel
6. Grid akan menentukan parameter yang optimal
7. Selesaikan dengan memilih *error* terkecil dari CV

## 2.5 Pengukuran Hasil Peramalan

Seperti yang sudah dijelaskan sebelumnya bahwa tidak ada peramalan yang memiliki tingkat akurasi yang tinggi 100%. Setiap peramalan pasti mengandung kesalahan. Oleh karena itu untuk mengukur sebuah metode peramalan, hal yang diperhitungkan adalah tingkat kesalahan yang terjadi. Semakin kecil kesalahan yang dihasilkan maka semakin baik peramalan tersebut.

Berdasarkan Heanke, Reitsch dan Wichern (2001) beberapa cara mengukur *error* dari peramalan adalah sebagai berikut:

1. *Mean Square Error* (MSE)

MSE dihitung dengan menjumlahkan kuadrat semua kesalahan peramalan pada setiap periode dan membaginya dengan jumlah periode peramalan.

Secara sistematis, MSE dirumuskan sebagai berikut:

$$MSE = \frac{\sum_{k=0}^n (A_t - F_t)^2}{n} \quad (1.26)$$

Dimana:

$A_t$  = permintaan aktual

$F_t$  = permintaan hasil ramalan

$n$  = jumlah data

2. *Mean Absolute Percentage Error* (MAPE).

MAPE merupakan cara perhitungan dengan melakukan perbandingan persentase perbedaan nilai rata-rata absolute antara nilai peramalan dengan nilai yang sebenarnya terjadi. Secara sistematis, MAPE dinyatakan sebagai berikut:

$$MAPE = \left(\frac{1}{n}\right) \sum_{i=1}^n \left| \frac{F_t - A_t}{A_t} \right| \quad (1.27)$$

Dimana:

$n$  = jumlah data

$F_t$  = permintaan hasil peramalan

$A_t$  = permintaan aktual