

BAB II

LANDASAN TEORI

2.1 DAYA LISTRIK

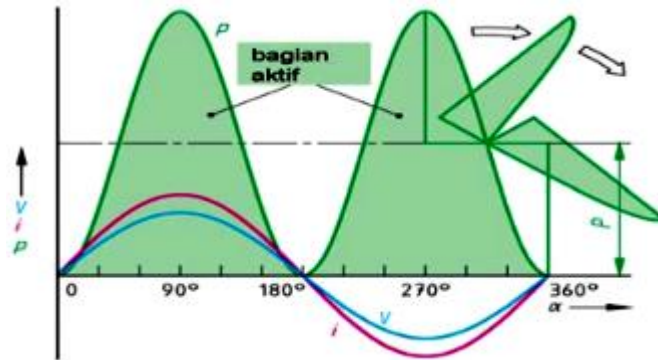
Daya listrik adalah laju hantaran energi listrik dalam rangkaian listrik. Daya memiliki arti sebagai energi per satuan waktu (Von Meier Alexander, 2006). Selain itu daya listrik merupakan suatu besarnya usaha yang dilakukan oleh sumber tegangan dalam 1 sekon. Jika dalam waktu t sekon sumber tegangan telah melakukan usaha sebesar W . Daya listrik bisa juga disebut suatu kekuatan yang dikandung dalam aliran arus dan tegangan listrik melalui hambatan dengan besaran tertentu. Daya pada suatu sistem tegangan bolak-balik (AC) dikenal dengan tiga macam yaitu, daya aktif (nyata) dengan simbol (P) satuannya adalah *Watt* (W), daya reaktif dengan simbol (Q) satuannya adalah *Volt Ampere Reactive* (VAR) dan daya semu dengan simbol (S) satuannya adalah *Volt Ampere* (VA). KWh bisa juga disebut kilowatt-jam atau *killojam-hour* merupakan ukuran satuan energi listrik yang dikirim oleh perlatan elektronik yang terhubung dan membutuhkan listrik serta diberi biaya. KWh adalah produk tenaga listrik dalam satuan kilowatt dikalikan dengan waktu dalam jam, bukan kilowatt per hour (Kw per h) (Hariyadi, 2015).

2.1.1 MACAM-MACAM JENIS DAYA LISTRIK

Ada 3 macam jenis daya listrik (Kusumandaru, 2015) yaitu :

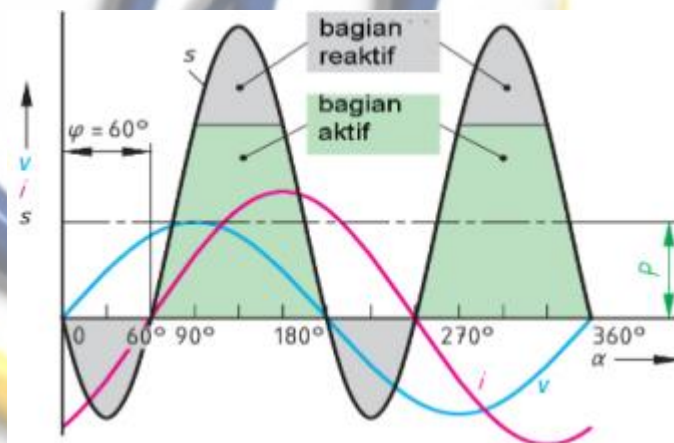
a. Daya Aktif

Daya aktif adalah daya yang sesungguhnya dibutuhkan oleh beban. Satuan daya aktif adalah W (*Watt*) dan dapat diukur dengan menggunakan alat ukur listrik *Wattmeter*. Daya aktif pada beban yang bersifat resistansi (R), dimana tidak mengandung induktor grafik gelombang tegangan (V) dan arus se fasa, sehingga besar daya sebagai perkalian tegangan dan arus menghasilkan dua gelombang yang keduanya bernilai positif. Besarnya daya aktif adalah P , sisa puncak dibagi menjadi dua untuk mengisi celah-celah kosong sehingga kedua rongga terisi oleh dua puncak yang mengisinya.



Gambar 2.1 Gelombang Daya Aktif Pada Beban Resistansi

Daya aktif pada beban impedansi (Z), beban impedansi pada suatu rangkaian disebabkan oleh beban yang bersifat resistansi (R) dan induktansi (L). Maka gelombang mendahului gelombang arus sebesar φ . Perkalian gelombang tegangan dan gelombang arus menghasilkan dua puncak positif yang besar dan dua puncak negatif yang kecil. Pergeseran sudut fasa bergantung seberapa besar nilai dari komponen induktor nya.



Gambar 2.2 Gelombang Daya Aktif Pada Beban Impedansi

b. Daya Reaktif

Daya reaktif adalah daya yang dibutuhkan untuk pembentukan medan magnet atau daya yang ditimbulkan oleh beban yang bersifat induktif. Satuan daya reaktif adalah *VAR* (*Volt.Amper Reaktif*). Untuk menghemat daya reaktif dapat dilakukan dengan memasang kapasitor pada rangkaian yang memiliki beban bersifat induktif. Hal serupa sering dilakukan pada pabrik-pabrik yang banyak menggunakan beban berupa motor-motor listrik.

c. Daya Semu

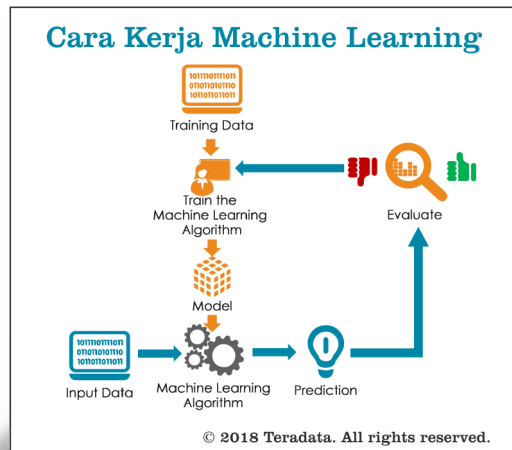
Daya semu adalah daya yang dihasilkan dari perkalian tegangan dan arus listrik. Daya nyata merupakan daya yang diberikan oleh PLN kepada konsumen. Satuan daya nyata adalah VA (*Volt.Ampere*). Beban yang bersifat daya semu adalah beban yang bersifat resistansi (R), contohnya adalah lampu pijar, setrika listrik, kompor listrik dan lain sebagainya. Peralatan listrik atau beban pada rangkaian listrik yang bersifat resistansi tidak dapat dihemat karena tegangan dan arus listrik se fasa perbedaan sudut fasa adalah 0° dan memiliki nilai faktor daya adalah 1.

2.2 MACHINE LEARNING

Machine learning adalah aplikasi dari disiplin ilmu kecerdasan buatan (*Artificial Intelligence*) yang menggunakan teknik statistika untuk menghasilkan suatu model otomatis dari sekumpulan data, dengan tujuan memberikan komputer kemampuan untuk "belajar". Pembelajaran mesin atau *machine learning* memungkinkan komputer mempelajari sejumlah data (*learn from data*) sehingga dapat menghasilkan suatu model untuk melakukan proses input-output tanpa menggunakan kode program yang dibuat secara eksplisit. Proses belajar tersebut menggunakan algoritma khusus yang disebut *machine learning algorithms*. Terdapat banyak algoritma *machine learning* dengan efisiensi dan spesifikasi kasus yang berbeda-beda. Dalam hal ini *machine learning* memiliki kemampuan untuk memperoleh data yang ada dengan perintah ia sendiri untuk melakukan tugas tertentu. Tugas yang dapat dilakukan oleh *machine learning* pun sangat beragam, tergantung dari apa yang ia pelajari (Dicoding Intern Team, 2020).

Secara fundamental cara kerja *machine learning* adalah belajar seperti manusia dengan menggunakan contoh-contoh dan setelah itu barulah dapat menjawab suatu pertanyaan terkait (Advernesia, 2018). Proses belajar ini menggunakan data yang disebut *train dataset*. Berbeda dengan program statis, *machine learning* diciptakan untuk membentuk program yang dapat belajar sendiri. Untuk memastikan efisiensi model yang terbentuk, data akan dibagi menjadi data pembelajaran (*train dataset*) dan data pengujian (*test dataset*). Pembagian data yang digunakan bervariasi tergantung algoritma yang digunakan. Pada umumnya *train*

dataset lebih banyak dari *test dataset*, misalnya dengan rasio 3:1 s. *Test dataset* digunakan untuk menghitung seberapa efisien model yang dihasilkan untuk melakukan klasifikasi atau prediksi kedepan yang disebut *test score*. Semakin banyak data yang digunakan, *test score* yang dihasilkan semakin baik.



Gambar 2.3 Cara Kerja Machine Learning

2.3 PYTHON

Python adalah bahasa pemrograman tinggi yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung (*interpretatif*) dengan metode orientasi objek (*Object Oriented Programming*) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan syntax. Sebagai bahasa pemrograman tinggi, python dapat dipelajari dengan mudah karena sudah dilengkapi dengan manajemen memori otomatis (*pointer*) (Advernesia, 2018).

Bahasa ini muncul pertama kali pada tahun 1991, dirancang oleh seorang bernama Guido van Rossum. Sampai saat ini python masih dikembangkan oleh *Python Software Foundation*. Bahasa python mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distronya sudah menyertakan Python di dalamnya.

2.4 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) dikembangkan oleh Boser, Guyon, dan Vapnik, pertama kali diperkenalkan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar metode SVM sebenarnya merupakan gabungan atau kombinasi dari teori-teori komputasi yang telah ada pada

tahun sebelumnya, seperti *marginhyperplane* (Duda dan Hart, 1973; Cover, 1965; Vapnik, 1964), kernel diperkenalkan oleh Aronszajn tahun 1950, *Lagrange Multiplier* yang ditemukan oleh Joseph Louis Lagrange pada tahun 1766, dan demikian juga dengan konsep-konsep pendukung lain.

Menurut Fachrurrazi (2011) SVM merupakan suatu teknik untuk melakukan prediksi, baik prediksi dalam kasus regresi maupun klasifikasi. Teknik SVM digunakan untuk mendapatkan fungsi pemisah (*hyperplane*) yang optimal untuk memisahkan observasi yang memiliki nilai variabel target yang berbeda (William, 2011). *Hyperplane* ini dapat berupa *line* pada *two dimension* dan dapat berupa *flat plane* pada *multiple dimension*.

2.5 SUPPORT VECTOR REGRESSION (SVR)

SVR merupakan bagian dari *Support Vector Machine* (SVM) yang diperkenalkan oleh Vapnik (Cortes & Vapnik, 1995). SVM adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam sebuah ruang fitur (*feature space*) berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan learning bias. Konsep SVM menggunakan konsep ε -insensitive loss function (Abe, 2010). SVM dapat digeneralisasi untuk melakukan pendekatan

$$f(x) = w^t \varphi(x) + b \dots \dots \dots (1)$$

fungsi yang dikenal dengan SVR. Konsep SVM menggunakan *hyperplane* tunggal pada ruang berdimensi banyak yang pada akhirnya partisi-partisi tersebut dapat diselesaikan secara non linear. *Hyperplane* yang optimum dapat diselesaikan dengan metode *quadratic programming*. Konsep SVR didasarkan pada *risk minimization*, yaitu untuk mengestimasi suatu fungsi dengan cara meminimalkan batasatas dari *generalization error*, sehingga SVR mampu mengatasi *overfitting*. Fungsi regresi dari metode SVR adalah sebagai berikut dengan w merupakan vector pembobot, $\varphi(x)$ merupakan fungsi yang memetakan x dalam suatu dimensi dan b merupakan bias. Untuk memaksimalkan *hyperplane* dengan meminimalkan nilai *loss function*.

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

Dimana

$$L_{\varepsilon}(y_i, f(x_i)) = \begin{cases} 0; & |y_i - f(x_i)| \leq \varepsilon \\ |y_i - f(x_i)| - \varepsilon; & \text{lainnya} \dots \dots \dots \end{cases} (3)$$

dengan L_{ε} merupakan ε - insensitive loss function, C dan ε merupakan parameter. Konsep dari kuadratik *loss function* adalah meminimumkan nilai sebagai berikut :

$$R(w, \xi, \xi^*) = \frac{1}{2} \|w\|^2 + C(\sum_{i=1}^n (\xi_i + \xi_i^*)) \dots \dots \dots (4)$$

Dengan batasan $w\varphi(x_i) + b - y_i \leq \varepsilon + \xi_i^*$; $y_i - w\varphi(x_i) - b \leq \varepsilon + \xi_i$ dan $\xi_i, \xi_i^* \geq 0$

Dengan menggunakan penyelesaian lagrange dalam bentuk :

$$L(w, b, \xi, \xi^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*) = \frac{1}{2} \|w\|^2 + C(\sum_{i=1}^n (\xi_i + \xi_i^*)) - \sum_{i=1}^n \alpha_i [w\varphi(x_i) + b - y_i + \varepsilon + \xi_i] - \sum_{i=1}^n \alpha_i^* [w\varphi(x_i) - b + \varepsilon + \xi_i^*] - \sum_{i=1}^n (\beta_i \xi_i + \beta_i^* \xi_i^*) \dots \dots (5)$$

Dengan menggunakan pendekatan Karush-Kuhn-Tuck didapatkan sebagai berikut:

$$Q(\alpha, \alpha^*) = -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i + \alpha_i^*)(\alpha_j + \alpha_j^*) K(x_i, x_j) - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \dots \dots \dots (6)$$

dengan batasan $\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0$; $0 \leq \alpha_i \leq C$; $0 \leq \alpha_i^* \leq C$ dimana $K(x_i, x_j)$ merupakan fungsi kernel. Penentuan nilai parameter C dan fungsi kernel sangat penting dalam menentukan tingkat keakuratan dari prediksi.

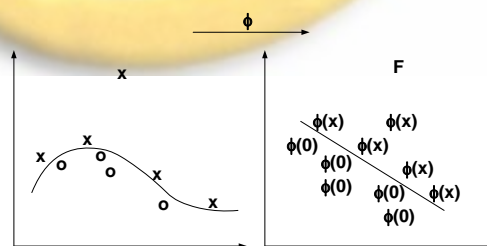
2.6 FUNGSI KERNEL

Beberapa metode dalam analisis *data mining* banyak menggunakan fungsi linear. Banyak kasus di dunia nyata merupakan kasus yang nonlinear, sehingga untuk mengatasinya dengan cara mentransformasi data kedalam dimensi ruang yang lebih tinggi. SVM dapat digunakan pada data nonlinear dengan menggunakan pendekatan kernel sehingga dapat dipisahkan secara linear pada *feature space* yang

baru. Fungsi kernel yang digunakan pada metode SVR adalah kernel linear, kernel polynomial, dan kernel *Radial Basis Function* (RBF) (Abe, 2010).

Metode kernel memberikan pendekatan alternatif dengan cara melakukan mapping data x dari input space ke *feature space* F melalui suatu fungsi ϕ sehingga $\phi : x \rightarrow \phi(x)$. Karena itu suatu titik x dalam *input space* menjadi $\phi(x)$ dalam *feature space*. Sering kali fungsi $\phi(x)$ tidak tersedia atau tidak bisa dihitung. Tetapi *dot product* dari dua vektor dapat dihitung baik dalam *input space* maupun dalam *feature space*. Dengan kata lain, sementara $\phi(x)$ mungkin tidak diketahui, *dot product* $\langle \phi(x_1), \phi(x_2) \rangle$ masih bisa dihitung dalam *feature space*. Untuk bisa menggunakan metode kernel, pembatas perlu diekspresikan dalam bentuk *dot product* dari vektor data x_i . Sebagai konsekuensi, pembatas yang menjelaskan permasalahan dalam klasifikasi harus diformulasikan kembali sehingga menjadi bentuk *dot product*.

Dalam *feature space* ini, *dot product* $\langle . \rangle$ menjadi $\langle \phi(x), \phi(x') \rangle$. Suatu fungsi kernel, $k(x, x')$, bisa untuk menggantikan *dot product* $\langle \phi(x), \phi(x') \rangle$. Kemudian di *feature space*, dibuatkan suatu fungsi pemisah yang linear yang mewakili fungsi yang nonlinear di *input space*. Gambar 2.4 mendeskripsikan suatu contoh *feature mapping* dari ruang dua dimensi ke *feature space* dua dimensi. Dalam *input space*, data tidak dapat dipisahkan secara linear, tetapi dapat dipisahkan di *feature space*. Karena itu dengan memetakan data ke *feature space* menjadikan tugas klasifikasi menjadi lebih mudah.



Gambar 2.4 Kernel Map yang Mengubah Masalah Tidak Linear Menjadi Linear

Fungsi kernel yang biasanya dipakai dalam literatur SVM (Intan, 2019) adalah :

- a. Linear

Fungsi kernel linear didefinisikan sebagai :

$$k(x_i, x_j) = x_i^T x_j \dots\dots\dots(7)$$

Kernel linear adalah fungsi paling sederhana yang merupakan perkalian titik dari dua vektor.

b. Polynomial

Fungsi kernel polynomial memiliki derajat, dimana r dan d adalah parameter didefinisikan sebagai :

$$k(x_i, x_j) = y x_i^T x_j + r^d, y > 0 \dots\dots\dots(8)$$

c. *Radial Basis Function* (RBF)

Fungsi kernel RBF juga biasa disebut dengan fungsi kernel gaussian yang didefinisikan sebagai :

$$k(x_i, x_j) = \exp \left(-\gamma \|x_i - x_j\|^2 \right), \gamma > 0 \dots\dots\dots(9)$$

dimana γ adalah parameter positif untuk mengatur jarak.

d. *Tangent hyperbolic (sigmoid)*

Fungsi kernel sigmoid didefinisikan sebagai :

$$k(x_i, x_j) = \tanh(\gamma x_i, x_j + r) \dots\dots\dots(10)$$

Dimana

$$\tanh(a) = 2\sigma(a) - 1, \text{ dan } \sigma(a) = \frac{1}{1 + \exp(-a)} \dots\dots\dots(11)$$

Fungsi kernel mana yang harus digunakan untuk substitusi *dot product* di *feature space* sangat bergantung pada data. Biasanya metode *cross validation* digunakan untuk pemilihan fungsi kernel ini. Pemilihan fungsi kernel yang tepat adalah hal yang sangat penting karena fungsi ini akan menentukan *feature space* dimana fungsi klasiner akan dicari. Sepanjang fungsi kernelnya *legitimate*, SVM akan beroperasi secara benar meskipun kita tidak tahu seperti apa map yang akan digunakan.

Fungsi kernel yang *legitimate* diberikan oleh Teori Mercer (Cortes & Vapnik, 1995) dimana fungsi itu harus memenuhi syarat *continous and positive definite*. Lebih mudah menemukan fungsi kernel daripada mencari map ϕ seperti apa yang tepat untuk melakukan mapping dari *input space* ke *feature space*. Pada penerapan metode kernel, kita tidak perlu tahu map apa yang digunakan untuk satu per satu data, tetapi lebih penting mengetahui bahwa *dot product* dua titik di *feature space* bisa digantikan oleh fungsi kernel.

SVR merupakan penerapan SVM untuk kasus regresi. Perbedaan antara SVM dan SVR adalah penerapan SVM dilakukan untuk menemukan fungsi pemisah (*hyperplane*) terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua objek, misalkan objek tersebut akan termasuk dalam kelas -1 atau kelas +1 dengan cara memaksimalkan margin atau jarak antara dua set objek dari kelas yang berbeda. Sedangkan penerapan SVR dilakukan untuk menemukan suatu fungsi yang memiliki deviasi paling besar ε dari target aktual y_i .

2.7 PERHITUNGAN TINGKAT KESALAHAN PREDIKSI

Ada beberapa perhitungan matriks (DataTechNotes, 2019) yang dapat digunakan untuk mengevaluasi tingkat kesalahan prediksi dari kinerja model dalam analisis regresi yaitu :

- a. *Mean Absolute Error* (MAE) mewakili perbedaan antara nilai asli dan nilai prediksi yang diekstraksi dengan rata-rata perbedaan absolut atas kumpulan data.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}| \dots\dots\dots(12)$$

- b. *Mean Squared Error* (MSE) menunjukkan perbedaan antara nilai asli dan prediksi yang diekstrak dengan mengkuadratkan perbedaan rata-rata di atas kumpulan data.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2 \dots\dots\dots(13)$$

- c. *Root Mean Squared Error* (RMSE) adalah tingkat kesalahan menurut akar kuadrat MSE.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2} \dots\dots\dots(14)$$

- d. *R-squared* (koefisien determinasi) mewakili koefisien seberapa cocok nilai dibandingkan dengan nilai aslinya. Nilai dari 0 hingga 1 ditafsirkan sebagai persentase. Semakin tinggi nilainya, semakin baik modelnya.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2} \dots\dots\dots(15)$$