



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Minyak Bumi

Minyak bumi atau dapat disebut minyak mentah merupakan senyawa organik dengan rantai hidrokarbon sulit terurai, namun memiliki arti dan fungsi yang besar bagi kehidupan manusia (Mardiah *et al*, 2003). Minyak bumi terdiri dari campuran kompleks berbagai hidrokarbon dan sebagian besar senyawa alkana yang terbentuk dari sisa renik tumbuhan dan hewan yang tertimbun selama berjuta tahun di dalam lapisan kerak bumi (Abdillah, 2017). Proses pembentukannya melibatkan suhu yang bisa di bilang sangat ekstrem. Minyak bumi disebut sebagai sumber daya alam yang tidak dapat diperbaharui karena proses pembentukannya yang sangat lama hingga berjuta-juta tahun (Abdillah, 2017).

Minyak bumi dapat diklasifikasikan sesuai dengan titik didih masing-masing. Dari perbedaan titik didih itu, bahan dasar minyak bumi di ubah menjadi berbagai material seperti aspal, bensin, bahkan hingga obat-obatan dan plastic. Karena Minyak bumi adalah sumber daya alam yang tidak dapat diperbaharui dan cukup lekat manfaatnya bagi kehidupan kita sehari-hari, minyak bumi merupakan sumber daya alam yang bernilai ekonomis cukup tinggi (Salamadian, 2018).

Minyak bumi didapatkan dari proses penambangan. Sebelum dilakukan penambangan, lokasi di identifikasi terlebih dahulu melalui beberapa cara, yaitu dengan

mempelajari struktur geologisnya, analisis sedimen, struktur sumber, karakter dan masih banyak lagi studi pendukung yang lain (OMC, 2015). Di Indonesia terdapat beberapa wilayah yang menghasilkan minyak bumi, seperti Riau yang menghasilkan 365 ribu barel per hari, Irian Jaya Barat menghasilkan 14 ribu barel per hari, Sumatera Selatan menghasilkan 10 ribu *barrel* per hari dan Jawa Timur menghasilkan 52 ribu barel per hari (Ningsih, 2014).

Produksi minyak bumi di Indonesia menunjukkan penurunan dapat dilihat pada gambar 1.3 dari bulan kebulan, per 30 Juni 2016 produksi minyak bumi sebesar 834,4 ribu barel per hari (bph). Sedangkan untuk konsumsi minyak bumi di Indonesia mengalami peningkatan pada tahun 2016 adalah 1,615 juta ribu barel per hari. Sampai data tahun 2018 defisit neraca minyak bumi di Indonesia kian melebar, untuk produksi minyak bumi sebesar 808,4 ribu barel per hari sedangkan untuk konsumsi sebesar 1,8 juta barel per hari (BG Global Company, 2019).

. Diantara negara penghasil minyak bumi, Indonesia menempati posisi ke-22 dengan produksi minyak bumi sebesar 911 ribu barel per hari dan Amerika Serikat menempati posisi pertama dengan produksi sebesar 13,9 juta barel per hari (IEA, 2020).

2.2 Machine Learning

Machine learning adalah bidang studi yang memberikan kemampuan pada komputer untuk belajar menemukan pola secara otomatis ataupun semi otomatis pada kumpulan data untuk menghasilkan performa yang lebih baik kedepannya (Witten & Frank, 2002). Algoritma *machine learning* membuat model matematika berdasarkan

data sampel atau dikenal dengan *training dataset* untuk membuat prediksi atau peramalan dan untuk memastikan efisiensi model, digunakan *test dataset* (Bishop, 2006). Ada beberapa macam jenis algoritma pembelajaran dalam *machine learning*, yaitu:

1. *Supervised learning*

Supervised learning adalah pembelajaran yang dapat menerapkan informasi yang telah ada pada data dengan memberikan label tertentu, misalnya data yang telah diklasifikasikan sebelumnya (terarah). Permasalahan *Supervised Learning* dapat dikelompokkan menjadi masalah regresi (*regression problem*) dan masalah klasifikasi (*classification problems*).

2. *Unsupervised learning*

Unsupervised learning adalah pembelajaran yang digunakan pada data yang tidak mempunyai informasi yang dapat diterapkan secara langsung (tidak terarah). Algoritma ini diharapkan mampu menemukan struktur tersembunyi pada data yang tidak berlabel. Permasalahan *unsupervised learning* dapat dikelompokkan menjadi *clustering problems* dan *association problems*.

3. *Semi-supervised learning*

Semi-supervised learning adalah pembelajaran yang digunakan untuk melakukan pembelajaran data berlabel dan tanpa label. Sistem yang menggunakan metode ini dapat meningkatkan efisiensi *output* yang dihasilkan. Permasalahan ini berada di antara *supervised learning* dan *unsupervised learning*.

4. *Reinforcement learning*

Reinforcement learning adalah pembelajaran yang mempunyai kemampuan untuk berinteraksi dengan proses belajar yang dilakukan, algoritma ini akan memberikan poin (*reward*) saat model yang diberikan semakin baik atau mengurangi poin (*punishment*) saat model yang dihasilkan semakin buruk.

Salah satu penerapannya adalah pada mesin pencari.

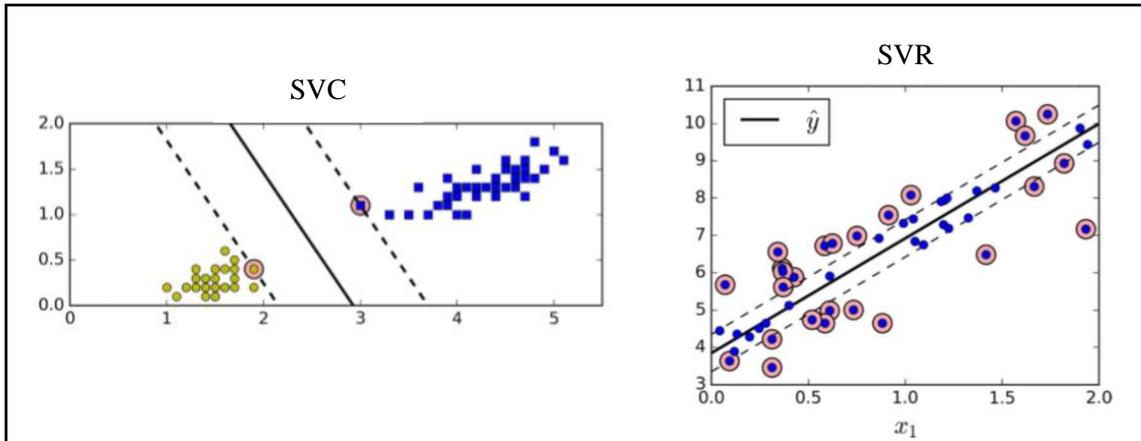
2.3 Regresi

Regresi adalah proses identifikasi relasi dan pengaruhnya pada nilai-nilai objek. Regresi bertujuan menemukan suatu fungsi yang memodelkan data dengan meminimalkan galat atau selisih antara nilai prediksi dengan nilai sebenarnya (Suyanto, 2018). Regresi umumnya digunakan untuk prediksi dan peramalan. Prediksi digunakan untuk memperkirakan nilai-nilai data bertipe apa saja dan kapan saja (masa lalu, sekarang, dan masa depan) (Suyanto, 2018). Digunakan kata prediksi karena penelitian bertujuan untuk memprediksi harga minyak bumi dengan menggunakan *dataset* harga minyak brent yang telah menjadi acuan untuk harga minyak dunia.

2.4 Support Vector Regression (SVR)

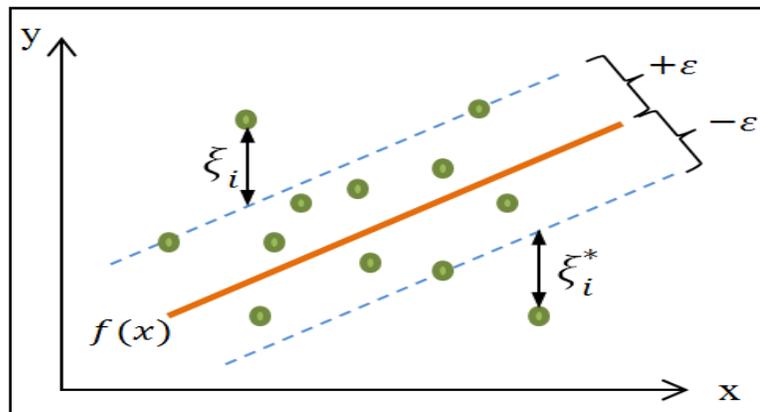
Support Vector Regression (SVR) merupakan versi untuk memecahkan permasalahan regresi dari *Support Vector Machine* (SVM) yang diperkenalkan oleh Vapnik, Steven Golowich dan Alex Smola pada tahun 1997 (Vapnik *et al*, 1997). Tujuan dari SVR adalah untuk menemukan sebuah fungsi sebagai suatu *hyperplane*

(garis pemisah) berupa fungsi regresi yang mana sesuai dengan semua *input* data dengan sebuah *error* dan membuat setipis mungkin (Smola, 2002). Ilustrasi fungsi *hyperplane* seperti pada gambar 2.1 berikut ini.



Gambar 2.1 SVC dan SVR (Géron, 2017)

Hal yang dilakukan SVR adalah memetakan set data ke ruang fitur dimensi tinggi *non-linear* dan menyelesaikan permasalahan regresi dalam ruang fitur dimensi ini (Lu dan Geng, 2011). Misalkan terdapat l data *training*, (x_i, y_i) , $i = 1, 2, \dots, l$ dengan data *input* $x = \{x_1, x_2, \dots, x_l\} \subseteq \mathfrak{R}$ dan $y = \{y_1, y_2, \dots, y_l\} \subseteq \mathfrak{R}$ dan l adalah banyaknya data training. SVR bertujuan menemukan suatu fungsi regresi $f(x)$ yang mempunyai deviasi paling besar ε dari target aktual y_1 untuk semua data latih.



Gambar 2.2 Hyperplane Support Vector Regression (Chanklan, 2018)

Ilustrasi *hyperplane* pada gambar 2.2. Fungsi regresi $f(x)$ dapat dinyatakan dengan formula sebagai berikut (Smola dan Schölkopf 2004):

$$f(x) = w^T \varphi(x) + b \quad \dots (2.1)$$

Dengan:

w = vector bobot

$\varphi(x)$ = fungsi yang memetakan x dalam dimensi yang lebih tinggi

b = bias

Untuk mendapatkan generalisasi yang baik untuk fungsi regresi $f(x)$, dilakukan dengan meminimalisir *norm* dari w dengan formulasi (Bermolen & Rossi, 2009):

$$\min\left\{\frac{1}{2} \|w\|^2\right\} \quad \dots (2.2)$$

$$\text{dengan kendala : } \begin{cases} y_i - w^T \varphi(x_i) - b \leq \varepsilon \\ w^T \varphi(x_i) - y_i + b \leq \varepsilon, i = 1, 2, \dots, l \end{cases}$$

Diasumsikan ada suatu fungsi $f(x)$ yang dapat mengaproksimasi semua titik (x_i, y_i) dengan presisi ε . Dalam kasus ini diasumsikan bahwa semua titik ada pada rentang $f(x) \pm \varepsilon$ (*feasible*). Dalam hal ketidaklayakan (*infeasible*), ada beberapa titik yang mungkin keluar dari rentang $f(x) \pm \varepsilon$, ditambahkan variable *slack* ξ, ξ^* untuk mengatasi permasalahan ketidaklayakan dalam problem optimasi pada formulasi 2.2. Selanjutnya, masalah optimasi diatas diformulasikan sebagai berikut:

$$\min\left\{\frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i, \xi_i^*)\right\} \quad \dots (2.3)$$

$$\text{dengan kendala } \begin{cases} y_i - w^T \varphi(x_i) - b \leq \varepsilon, i = 1, 2, \dots, l \\ w^T \varphi(x_i) - y_i + b \leq \varepsilon, i = 1, 2, \dots, l \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

Smola dan Schölkopf (2004) menjelaskan bahwa konstanta $C > 0$ menentukan *trade*

off antara tingkat *flatness* dari f dan batas atas deviasi yang lebih dari ε yang dapat ditoleransi. Masalah optimalisasi pada formulasi 2.3 dapat diselesaikan dengan bentuk formulasi *dual* (Smola dan Schölkopf 2004) sebagai berikut:

$$\max \begin{cases} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (a_i - a_i^*)(a_j - a_j^*) K(x_i, x_j) \\ \sum_{i=1}^l (a_i - a_i^*) y_i - \varepsilon \sum_{i=1}^l (a_i + a_i^*) \end{cases} \quad \dots (2.4)$$

dengan kendala $\sum_{i=1}^l (a_i - a_i^*) = 0$ dan $a_i - a_i^* \in [0, C]$

Fungsi regresi $f(x)$ pun dapat diformulasikan ulang secara eksplisit sebagai berikut:

$$f(x) = \sum_{i=1}^l (a_i - a_i^*) K(x_i, x) + b \quad \dots (2.5)$$

a_i^* dan a_i merupakan variabel *Lagrange multiplier*, $K(x_i, x)$ merupakan fungsi kernel dan b merupakan *bias*. Menurut (Smola dan Schölkopf 2004) solusi optimal untuk *bias* dapat dihitung dengan menggunakan KKT (*Karush-Khun-Tucker*) sebagai berikut:

$$a_i(\varepsilon + \xi_i - y_i - w^T \varphi(x_i) + b) = 0 \quad \dots (2.6)$$

$$a_i^*(\varepsilon + \xi_i^* + y_i - w^T \varphi(x_i) + b) = 0 \quad \dots (2.7)$$

$$(C - a_i)\xi_i = 0 \quad \dots (2.8)$$

$$(C - a_i^*)\xi_i^* = 0 \quad \dots (2.9)$$

Sehingga didapat:

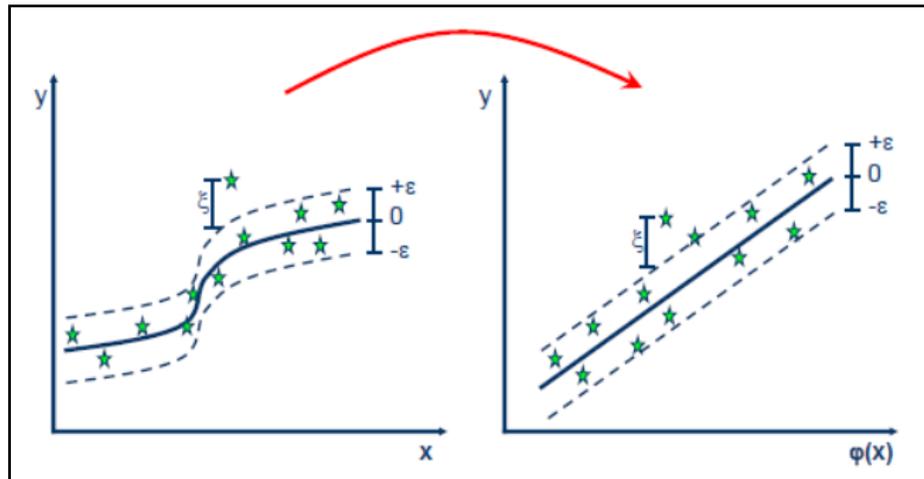
$$b = y_i - w^T \varphi(x_i) - \varepsilon \text{ untuk } 0 < a_i < C \quad \dots (2.10)$$

$$b = y_i - w^T \varphi(x_i) + \varepsilon \text{ untuk } 0 < a_i^* < C \quad \dots (2.11)$$

2.5 Kernel

Teknik pada *data mining* atau *machine learning* yang dikembangkan dengan asumsi kelinearan sangat banyak, sehingga untuk kasus-kasus yang *non-linear*, algoritma

yang dihasilkan terbatas. Dengan metode kernel (*Kernel Trick*) suatu data x di *input space* dipetakan ke *feature space* (ruang koordinat yang dimuati oleh fungsi) dengan dimensi yang lebih tinggi melalui φ (Sriyana, 2019).



Gambar 2.3 Kernel Trick (Sayad, 2010)

Pada gambar 2.3 menunjukkan secara eksplisit tentang hal yang dilakukan kernel untuk mempermudah algoritma dalam melakukan tugasnya, yaitu membuat data menjadi *linear* dengan memetakan *input space* ke *feature space* dimensi yang lebih tinggi. Ada beberapa jenis kernel dan formulanya sebagai berikut :

1. *Linear*

$$K(x, y) = x \cdot y \quad \dots (2.12)$$

2. *Polynomial*

$$K(x, y) = (\gamma(x \cdot y) + c)^d \quad \dots (2.13)$$

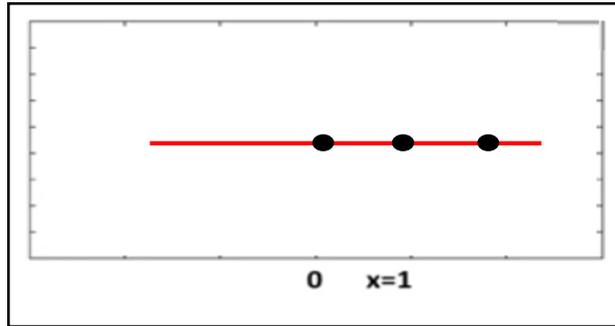
3. *Radial Basis Function (RBF)*

$$K(x, y) = \exp(-\gamma(\|x_i - x_j\|)^2) \quad \text{dengan } \gamma = \frac{1}{2\sigma^2} \quad \dots (2.14)$$

Dengan penggunaan kernel untuk kasus *non linearly separatable* pada ruang *input*

diharapkan menjadi *lineary separatable* pada *feature space*. Selanjutnya dapat digunakan *hyperplane* sebagai *decision boundary* secara efisien. Berikut akan diilustrasikan bagaimana setiap kernel memetakan masing-masing titik data.

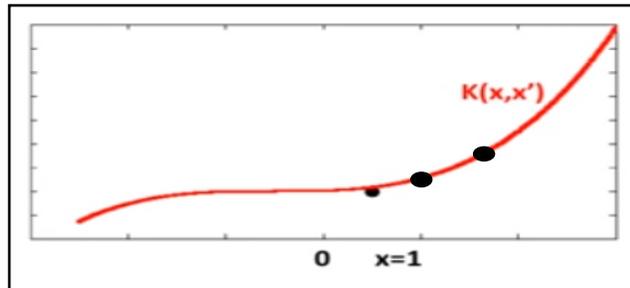
- Linear



Gambar 2.4 Linear Shape

Pada gambar 2.4 diberikan ilustrasi dimana memiliki input data 1 dimensi pada sumbu x dengan nilai 1. Untuk titik berikutnya $x=2$ akan berada pada garis yang sama dan disamping titik sebelumnya, dan seterusnya.

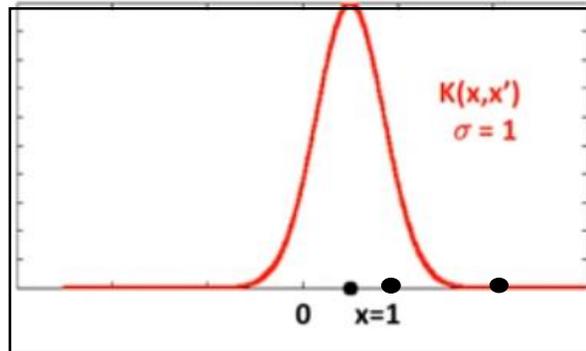
- Polynomial



Gambar 2.5 Polynomial Shape

Pada gambar 2.5 diberikan ilustrasi dimana memiliki input data 1 dimensi pada sumbu x dengan nilai 1 dan parameter pada polynomial $d=2$. Dapat dilihat, semakin besar nilai x maka akan menghasilkan nilai positif yang besar dan dipetakan lebih tinggi dari titik sebelumnya.

- Radial Basis Function (RBF)



Gambar 2.6 RBF Shape

Pada gambar 2.6 diberikan ilustrasi dimana memiliki input data 1 dimensi pada sumbu x dengan nilai 1 dan parameter pada polynomial $\sigma=1$. Dapat dilihat pada gambar nilai sigma menentukan lebar dari kurva, semakin besar sigma, maka curva akan semakin lebar dan menurun tingginya. Pada gambar diilustrasikan ada 2 buah titik didalam kurva. Titik didalam kurva itu nilanya akan dipengaruhi oleh kurva yang terbentuk.

2.6 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) merupakan metode yang dapat digunakan sebagai acuan untuk mengukur *error* yang dihasilkan peramalan. MAPE merupakan ukuran untuk menghitung *error* dengan membandingkan seberapa besar kesalahan peramalan dibandingkan dengan nilai sebenarnya. MAPE dipilih untuk pengujian *error* karena dapat memberikan hasil yang relatif akurat (Nugroho *et al*, 2015) dalam bentuk persentase *error* yang dihasilkan. Persamaan MAPE ditunjukkan pada persamaan 2.15 berikut:

$$MAPE = \frac{1}{n} \sum_{i=1}^l \left| \frac{y'_i - y_i}{y'_i} \right| \times 100 \quad \dots (2.15)$$

n adalah jumlah data, y adalah nilai prediksi, y' merupakan hasil aktual, dan l adalah banyaknya dimensi data. Penggunaan MAPE pada evaluasi hasil prediksi dapat menghindari pengukuran akurasi terhadap besarnya nilai aktual dan nilai prediksi untuk menghasilkan angka *minus*. MAPE juga memiliki Kriteria nilai yang berguna sebagai pengukur performa yang dilakukan berdasarkan kategori yang ditunjukkan pada Tabel 2.1. Tabel kriteria nilai MAPE ini menjadi salah satu alasan menggunakan metode perhitungan MAPE.

Tabel 2.1 Kriteria Nilai MAPE (Chang et al, 2007)

Nilai MAPE	Kriteria
<10%	Sangat Baik
10% - 20%	Baik
20%-50%	Cukup
>50%	Buruk

2.7 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) merupakan metode yang dapat digunakan untuk evaluasi model peramalan. Dalam kasus ini MAE dipilih karena memberikan bobot yang sama untuk seluruh data (Chai & Draxler, 2014). Berikut formula dari *Mean absolute Error* (MAE).

$$MAE = \frac{1}{n} \sum_{i=1}^l |f_i - y_i| \quad \dots (2.16)$$

dengan f_i adalah nilai prediksi dan y_i adalah nilai aktual dan n merupakan banyaknya data. Berdasarkan formula 2.16, MAE akan menghitung rata-rata *error* dengan memberi bobot yang sama untuk seluruh data dan formula MAE relatif mirip dengan MAPE.